

Domain-Decomposed Graph Neural Network Surrogate Modeling for Ice Sheets

Adrienne M. Propp* Mauro Perego † Eric C. Cyr ‡ Anthony Gruber §
 Amanda Howard ¶ Alexander Heinlein ‖ Panos Stinis ** Daniel Tartakovsky ††

December 2, 2025

Abstract

Accurate yet efficient surrogate models are essential for large-scale simulations of partial differential equations (PDEs), particularly for uncertainty quantification (UQ) tasks that demand hundreds or thousands of evaluations. We develop a physics-inspired graph neural network (GNN) surrogate that operates directly on unstructured meshes and leverages the flexibility of graph attention. To improve both training efficiency and generalization properties of the model, we introduce a domain decomposition (DD) strategy that partitions the mesh into subdomains, trains local GNN surrogates in parallel, and aggregates their predictions. We then employ transfer learning to fine-tune models across subdomains, accelerating training and improving accuracy in data-limited settings. Applied to ice sheet simulations, our approach accurately predicts full-field velocities on high-resolution meshes, substantially reduces training time relative to a single global surrogate model, and provides a ripe foundation for UQ objectives. Our results demonstrate that graph-based DD, combined with transfer learning, provides a scalable and reliable pathway for training GNN surrogates on massive PDE-governed systems, with broad potential for application beyond ice sheet dynamics.

1 Introduction

Many important scientific phenomena are governed by complex, nonlinear partial differential equations (PDEs) posed on intricate, evolving geometries. High-fidelity numerical solvers for such systems, while accurate, are often prohibitively slow: a single simulation may take hours, days, or longer. Decision-oriented tasks such as uncertainty quantification (UQ), which require hundreds or thousands of model evaluations, can therefore become computationally intractable using traditional solvers alone. This has created a growing demand for efficient modeling techniques that not only accelerate simulations but also respect the underlying physics, scale to massive problems, and generalize across changes in mesh resolution, input parameters, and domain geometry.

In the present work, we address these challenges by developing a physics-inspired graph neural network (GNN)-based surrogate model, applied to the case of ice sheets. Ice sheet dynamics provide a compelling and high-impact testbed. These simulations involve coupled PDEs on intricate domains, representing vast regions across which physical processes vary significantly. Reliable predictions of ice movement and mass loss underpin critical policy decisions concerning coastal infrastructure, resource planning, and climate adaptation. However, the computational cost of traditional methods often precludes comprehensive UQ, limiting our ability to quantify risk and explore plausible scenarios. Efficient and accurate surrogate models are therefore essential in this domain.

While recent advances in machine learning (ML) offer several promising directions for data-driven surrogate modeling, GNNs are naturally suited to the setting of modeling physical systems on unstructured meshes.

*Institute for Computational and Mathematical Engineering, Stanford University, propp@stanford.edu

†Department of Scientific Machine Learning, Sandia National Laboratories, mperego@sandia.gov

‡Department of Scientific Machine Learning, Sandia National Laboratories, eccyr@sandia.gov

§Department of Scientific Machine Learning, Sandia National Laboratories, adgrube@sandia.gov

¶Advanced Computing, Mathematics and Data Division, Pacific Northwest National Laboratory

‖Delft Institute of Applied Mathematics, Delft University of Technology, Delft, Netherlands, a.heinlein@tudelft.nl

**Advanced Computing, Mathematics and Data Division, Pacific Northwest National Laboratory

††Department of Energy Science Engineering, Stanford University

MeshGraphNets [46] was among the first to demonstrate that GNNs can accurately learn physical dynamics on mesh-based PDE simulations while operating directly on irregular meshes and without requiring remeshing or post-processing. Alternative approaches, such as operator learning via DeepONets [16], have also shown strong performance for ice sheet modeling. However, because both the branch and trunk networks are tied to fixed sensor locations and domain geometry, standard DeepONets but must be retrained for each change in the computational domain and mesh, limiting their generalization and flexibility.

However, training surrogates at the scale of ice sheets, where tens of thousands of nodes represent millions of square kilometers, requires innovative techniques to ensure accuracy, generalization and computational efficiency. Transfer learning has emerged as a powerful tool in data-limited regimes [23, 35, 47], allowing models to be pre-trained on broad distributions and then fine-tuned on smaller, task-specific datasets. Separately, domain decomposition (DD) methods have long been a cornerstone of classical numerical simulations, reducing computational and memory costs as well as enabling concurrency of computation by partitioning the domain into smaller subdomains before solving [10, 48, 56, 59]. These ideas are now being revisited in the context of ML surrogates [18, 31], though they have not yet been extended to the case of GNNs. In our work, we combine transfer learning and domain decomposition to obtain major improvements in convergence speed, accuracy, and computational efficiency in our GNN surrogate.

While the modeling framework we propose has far-reaching implications beyond the context of ice sheet dynamics, we focus on predicting ice sheet velocity at a given time t from basal friction, bed topography (or elevation), and thickness at time t . Our main contributions include:

1. A physics-inspired GNN surrogate that accurately models ice sheet velocity on a large, unstructured mesh,
2. A transfer learning strategy that accelerates GNN training, and
3. A domain decomposition (DD) framework for GNNs.

Together, these contributions advance the development of efficient, accurate graph-based surrogate models for large-scale physical systems, and lay the groundwork for future UQ studies of ice sheet dynamics and other systems.

The remainder of the paper is organized as follows. Section 2 introduces the ice sheet model and governing equations. Section 3 describes our computational approach, including an overview of GNNs, our specific architecture, and our training setup. Sections 4 and 5 introduce our transfer learning and domain decomposition strategies, respectively. Section 6 presents results from our computational experiments, and Section 7 concludes with a discussion of implications and future directions, particularly for uncertainty quantification.

2 Physics-based model and data generation

As a case study, we demonstrate our surrogate modeling approach on Greenland’s Humboldt Glacier. Spanning nearly 2 million square kilometers, Humboldt Glacier is one of Greenland’s largest glaciers. Its massive size and complex dynamics make it an ideal test case for large-scale surrogate modeling of physical systems.

We use the MPAS-Albany Land Ice (MALI) ice-sheet model [21] to simulate glacier dynamics, governed by the equations of ice flow described below. MALI solves the governing equations using a finite-volume and finite-element discretization on unstructured meshes that are adaptively refined where higher accuracy is needed (see Figure 1). In our setup, the ice thickness and temperature equations are solved using a finite volume method with explicit forward-Euler time stepping on a Voronoi mesh of the Humboldt Glacier, while the velocity equations are solved on its dual Delaunay triangulation using low-order Lagrangian finite elements. Below, we briefly summarize the key equations and constitutive relationships governing ice thickness and velocity evolution. We refer the interested reader to [20] for additional details on the governing equations (and topics such as the temperature model and calving), and to [21, 63] for additional details on the Humboldt dataset setup.

2.1 Ice sheet model equations

The evolution of ice thickness $H(x, y, t)$ is governed by the continuity equation:

$$\partial_t H + \nabla \cdot (\bar{\mathbf{u}}H) = a_H, \quad (1)$$

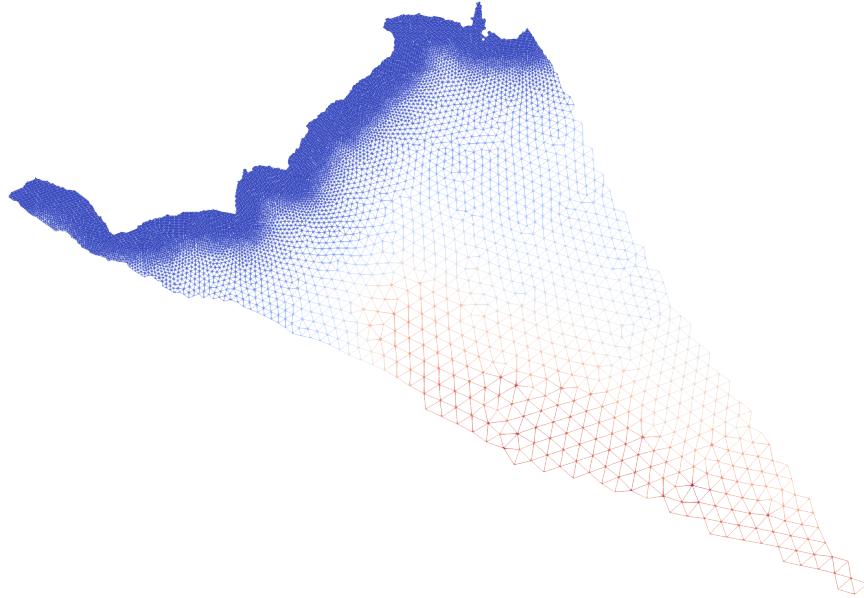


Figure 1: Dual Delaunay triangulation used as the graph representation of the Humboldt Glacier. In the MALI ice sheet model, ice thickness and temperature are discretized using a finite-volume method on a Voronoi mesh, while the velocity equations are solved on its dual Delaunay triangulation using low-order Lagrangian finite elements. We use this dual mesh as the node–edge structure for our GNN surrogate. Colors indicate cell area, with red indicating large cell area and blue indicating small cell area. Note that cell area decreases toward the glacier terminus where velocities and uncertainty are greatest, corresponding to finer mesh resolution in the finite element simulations.

where

$$\bar{\mathbf{u}} = \frac{1}{H} \int_l^s \mathbf{u} dz \quad (2)$$

is the depth-integrated velocity integrated between the glacier bed $z = l(x, y, t)$ and upper surface $z = s(x, y, t)$. The accumulation-ablation term a_H accounts for surface and basal mass balance, including both snowfall and melting. We adopt the convention that $z = 0$ corresponds to sea level, and define the bedrock surface $b(x, y)$ such that $l(x, y, t) = b(x, y)$ for grounded ice and $l(x, y, t) = -\frac{\rho}{\rho_w} H(x, y, t)$ for floating ice, where ρ and ρ_w are the densities of ice and seawater, respectively.

At the ice-sheet scale, ice behaves as an incompressible, shear-thinning fluid, with velocity $\mathbf{u}(x, y, z)$ satisfying Blatter-Pattyn approximation [2, 42]. This model simplifies the full nonlinear Stokes equations for shallow ice sheets by neglecting small terms in the vertical momentum balance and strain-rate tensor, yielding

$$-\nabla \cdot (2\eta(\mathbf{u}) \mathbf{D}(\mathbf{u})) = -\rho g \nabla s. \quad (3)$$

Here, ρ is the ice density, g is acceleration due to gravity, and η is the effective viscosity of ice, defined as:

$$\eta = \frac{1}{2} A(T)^{-q} D_e(\mathbf{u})^{q-1}, \quad (4)$$

with effective strain rate

$$D_e(\mathbf{u}) = \sqrt{u_x^2 + v_y^2 + u_x v_y + \frac{1}{4} (u_y + v_x)^2 + \frac{1}{4} (u_z^2 + v_z^2)}. \quad (5)$$

In (4), $A(T)$ is the temperature-dependent rate factor. We assume $q = 1/3$, which is standard practice and corresponds to Glen’s flow law exponent $n = 3$ [6, 12]. The modified strain-rate tensor $\mathbf{D}(\mathbf{u})$ is defined as:

$$\mathbf{D}(\mathbf{u}) = \begin{bmatrix} 2u_x + v_y & \frac{1}{2}(u_y + v_x) & \frac{1}{2}u_z \\ \frac{1}{2}(u_y + v_x) & u_x + 2v_y & \frac{1}{2}v_z \end{bmatrix}, \quad (6)$$

where u and v are the horizontal components of the velocity \mathbf{u} , and subscripts denote partial derivatives. Note that the Blatter-Pattyn model does not solve for the vertical component of the velocity, but it can be recovered from the the incompressibility condition $\nabla \cdot \mathbf{u} = 0$. Also, note that viscosity depends on the temperature T , which satisfies a heat equation (see [21] for more information).

Boundary conditions, particularly those governing the sliding of ice along the glacier bed, are of primary importance in ice sheet modeling. In this work, we model basal sliding using Budd's law:

$$2\eta(\mathbf{u})\mathbf{D}(\mathbf{u})\mathbf{n} = \mu N |\mathbf{u}|^{q-1} \mathbf{u}, \quad (7)$$

where $N = \max(\rho g H - \rho_w g z, 0)$ is the effective pressure at the bed, and $\mu(x, y)$ is a spatially varying basal friction coefficient. The field μ is a major source of uncertainty in ice sheet models, as it cannot be measured directly. Quantifying uncertainty in μ is therefore essential for producing reliable projections of glacial dynamics. Accelerating this analysis is one of the central motivations of this work. In the next section, we discuss how to approximate the probability distribution of μ .

We adopt the Mono-Layer Higher-Order (MOLHO) approximation [8], as implemented in MALI [19], which solves the Galerkin weak form of the Blatter-Pattyn (or higher-order) equations using a depth-separable velocity ansatz:

$$\mathbf{u}(x, y, z) = \Phi(z)\mathbf{u}_b(x, y) + (1 - \Phi(z))\mathbf{u}_s(x, y), \quad (8)$$

where

$$\Phi(z) = \left(\frac{s - z}{H} \right)^{\frac{1}{q}+1}. \quad (9)$$

Here, \mathbf{u}_b and \mathbf{u}_s denote velocities at the bed and upper surface of the ice-sheet, respectively. The shape functions $\Phi(z)$ and $(1 - \Phi(z))$ are also used to define the test functions of the weak formulation.

2.2 Basal Friction Parameter Distribution

The basal friction field $\mu(x, y)$ plays a central role in ice sheet dynamics but cannot be observed directly. In practice, μ is typically estimated by solving a PDE-constrained optimization problem [13, 36, 39, 44, 45] to identify the basal friction field that minimizes the misfit between observed surface velocities and those predicted by the physical model. To ensure positivity, we work with the transformed variable $p = \log(\mu)$.

We seek a probability distribution for p that reflects uncertainties in the observations, model, and estimated parameter itself. Classical Bayesian inference methods, such as Markov Chain Monte Carlo (MCMC), become intractable for such high-dimensional inverse problems. Instead, we adopt the Laplace approximation, which has been successfully applied in ice sheet modeling [3, 27, 50] after being first proposed in this context by [25]. The Laplace approximation constructs a quadratic (Gaussian) approximation of the log-posterior distribution in a neighborhood of the maximum a posteriori (MAP) estimate p_{MAP} , which can be obtained from the PDE-constrained optimization approach mentioned earlier. Although computing the MAP point is itself nontrivial, the resulting Gaussian approximation provides a tractable and scalable representation of posterior uncertainty.

Under this approximation, the posterior distribution of p is

$$p \sim \mathcal{N}(p_{\text{MAP}}, \Sigma^{\text{post}}). \quad (10)$$

Samples can then be generated as:

$$p = p_{\text{MAP}} + L\omega, \quad \omega \sim \mathcal{N}(0, I), \quad (11)$$

where L satisfies $LL^T = \Sigma^{\text{post}}$. In practice, neither L nor Σ^{post} is computed explicitly. Instead, we compute the matrix-free operation $L\omega$, allowing sampling from the posterior without assembling dense covariance matrices. The corresponding basal friction realizations are then obtained as $\mu = \exp(p)$. Further details on this approach can be found in [25].

We adopt a Gaussian prior on p of the form:

$$p \sim \mathcal{N}(0, \mathcal{A}^{-2}), \quad (12)$$

where \mathcal{A} is a precision operator, the infinite-dimensional analogue of a precision matrix (or inverse covariance matrix) in finite-dimensional Gaussian distributions. In PDE-based priors, the precision operator is a differential operator that imposes smoothness. Specifically, we define \mathcal{A} as:

$$\mathcal{A} := \begin{cases} -\gamma \Delta p + \delta p & \text{in } \Gamma_b, \\ -\gamma \nabla p \cdot \mathbf{n} - \xi p & \text{on } \partial\Gamma_b. \end{cases} \quad (13)$$

This represents a second-order elliptic operator with Neumann-type boundary conditions, promoting smooth friction fields while permitting sharp variations where supported by data. The notation \mathcal{A}^{-2} denotes the corresponding covariance operator, a convention common in PDE-constrained inverse problems. This construction yields a prior with the desired Sobolev regularity and spatial correlation structure. We set $\gamma = 8.976 \text{ km}^2$, $\delta = 8.865 \times 10^{-3}$, and $\xi = 0.1987 \text{ km}^{-1}$, yielding a prior with approximately unit marginal variance and a correlation length of 90km (see [62]).

Observational data enter through the likelihood term, which depends on the observed surface velocity \mathbf{u}_s . We assume $\mathbf{u}_s^{\text{obs}}$ is observed with covariance Σ^{obs} , inflated to account for both measurement uncertainty and model error. This setup allows us to generate ensembles of basal friction fields that capture physically plausible variability and propagate this uncertainty into the resulting ice flow simulations.

2.3 Data generation

To generate training data for our surrogate model, we sample basal friction fields $\mu(x, y)$ from the posterior distribution described in Section 2.2 and evolve the ice sheet forward from year 2007 to year 2100 using the MALI ice sheet model. Our setup builds on earlier work, e.g. [16], but incorporates several key improvements that provide more realistic physics and substantially richer training data.

First, the MALI ice sheet model used here includes physical processes absent from [16], including calving, basal melting, and temperature evolution. Second, we adopt Budd's nonlinear sliding law, which is more realistic than the linearized form used previously. Third, the Humboldt Glacier mesh resolution is roughly eight times finer than in [16], allowing for the representation of steep gradients near the terminus. Finally, our basal friction probability distribution is constrained by both data and physics, while [16] imposed a simple squared-exponential Gaussian process prior. Together, these advances yield substantially higher-fidelity simulation data and provide a more challenging and representative test case for surrogate modeling.

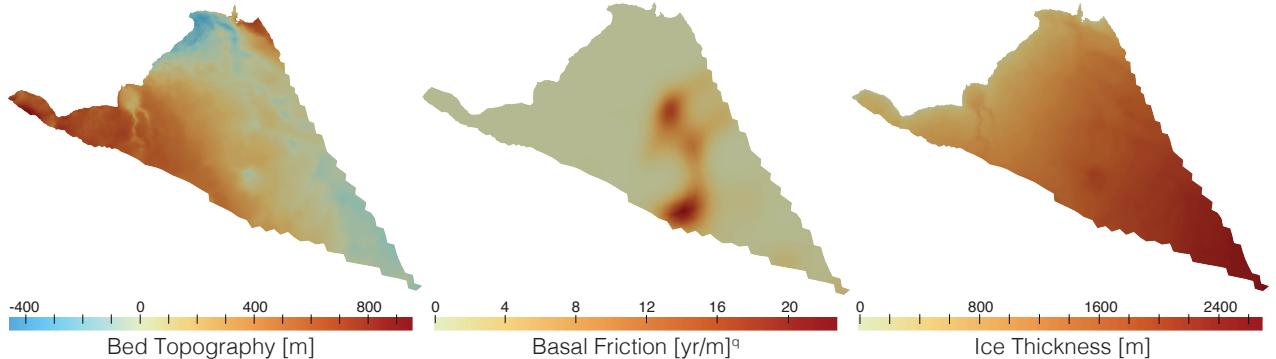


Figure 2: Sample realizations of the input features for our GNN surrogate, plotted on the Humboldt Glacier. Left: bed topography [m]; Center: basal friction [$\text{yr}/\text{m}]^q$, where the value of q accounts for the nonlinearity of the sliding law; Right: ice thickness [m].

3 Computational approach

In this section, we describe the components of our surrogate modeling framework, including an overview of graph neural networks (GNNs), the specific bracket-based GNN architecture we adopt, the construction of the computational graph, our training protocol, and the fine-tuning and domain-decomposition strategies we used to improve training efficiency and generalization. Throughout, we emphasize the design choices motivated by the physics of ice-sheet flow and the limitations of standard message-passing GNNs when applied to large-scale scientific simulations.

3.1 Graph neural networks (GNNs)

Many problems in computational physics, including ice-sheet modeling, involve solving PDEs on irregular geometries and unstructured meshes. Most standard ML architectures assume that data lie on a regular, Euclidean

grid. Applying them to unstructured FEM meshes requires remeshing or interpolation, which can introduce error, lose resolution, and increase computational cost. GNNs offer a natural alternative: they are designed for relational data, support permutation invariance, and operate directly on unstructured meshes [65]. These properties make GNNs a promising tool for surrogate modeling of ice dynamics and other physical systems [46, 55], where geometric flexibility and the ability to incorporate new meshes or subdomains are important.

The message-passing GNN modeling framework, proposed in [11], represents each node with a feature vector containing physical variables such as ice thickness and velocity. Although message-passing GNNs can perform predictions at the graph, edge, or node level, those designed for physical simulations primarily conduct node-level prediction through iterative aggregation and feature transformation operations on neighboring nodes. A typical GNN layer updates the node features according to:

$$h_i^{(l+1)} = \sigma \left(W^{(l)} \cdot \text{AGG}(\{h_j^{(l)} : j \in \mathcal{N}(i)\} \cup \{h_i^{(l)}\}) \right), \quad (14)$$

where $h_i^{(l)}$ denotes the representation of node i at layer l , $\mathcal{N}(i)$ denotes the neighborhood of node i , $W^{(l)}$ is a matrix of learnable weights, AGG is an aggregation operator (typically sum or mean), and σ is a nonlinear activation function. This framework enables the network to capture localized spatial interactions while aggregating broader contextual information across multiple layers. A reframing of this algorithm in the context of sparse linear algebra is detailed in [38].

For large-scale domains with heterogeneous physical behavior, GNNs offer several important computational advantages. GNNs perform learning through local message-passing operations that are applied uniformly across all nodes and edges. Because the same learned functions operate on node and edge features regardless of the underlying connectivity, model parameters can be transferred seamlessly across different graphs, as long as the feature dimensions are consistent. This enables flexible training strategies such as training on subgraphs, fine-tuning on related regions, and incorporating new data even when it arises from a different mesh or a modified domain geometry (see Section 5 for our DD approach). This flexibility is difficult to achieve with other deep learning architectures. For example, the DeepONet model developed in [16] must be re-trained for each new ice sheet or mesh configuration, limiting its usefulness.

Despite these advantages and successes across diverse modeling tasks (e.g.,[32, 33, 37, 57]), GNNs suffer from the notable drawback known as oversmoothing, where repeated averaging causes node features to become indistinguishable after only a small number of layers [34, 51, 64, 65]. Oversmoothing arises from the spectral properties of the graph Laplacian and the diffusive nature of standard message passing operators [5, 34]. As information propagates through successive layers, high-frequency components of the signal are damped. While the smoothing of node features can be advantageous for applications like node classification¹ [34], it ultimately leads to a loss of fine-grained spatial detail essential for capturing complex physical phenomena. For problems like ice sheet modeling, where preserving sharp gradients in quantities such as ice thickness, basal friction, and velocity fields is essential for accurate predictions, this presents a major challenge.

Various mitigations for oversmoothing have been proposed: for example, adding skip connections (such as the “multimesh” edge hierarchy introduced by [33]) to propagate long-range information with fewer message-passing steps; group normalization and stochastic edge-dropping, permitting GNNs to go deeper in node classification tasks [66]; and architectures that incorporate physically informed inductive biases to explicitly separate conservative and diffusive processes [1, 52]. We build on this last line of work by adopting the bracket-based GNN architecture of [14], which reformulates message passing through the lens of bracket-based partial differential equations. This physics-inspired framework mitigates oversmoothing and provides a principled mechanism for incorporating physical constraints and conservation laws directly into the GNN architecture. This ultimately yields more physically consistent and stable surrogate models for physical problems like ice sheet dynamics.

3.2 Bracket-based GNN architecture

We use the Hamiltonian bracket-based GNN architecture introduced in [14] as the core of our surrogate model, adapting it to the setting of large-scale physics simulations. Key to this is a reformulation of GNN message passing as a Hamiltonian dynamical system, where information propagates according to energy-conserving dynamics rather than diffusive averaging operations. This approach naturally avoids feature oversmoothing by preventing homogenization during message passing.

At a high level, the architecture processes information in three phases:

¹This is because classification tasks can be easier when the features of nodes in the same cluster are more similar.

1. **Encoding phase.** Raw node-edge feature pairs are lifted to a higher-dimensional latent space: $(\mathbf{q}', \mathbf{p}') = \mathbf{x}' \mapsto \mathbf{x} = (\mathbf{q}, \mathbf{p})$;
2. **Physics-inspired message-passing phase.** Latent features \mathbf{x} evolve to pseudo-time $T > 0$ (analogous to depth) according to an autonomous graph neural ODE:

$$\dot{\mathbf{x}} = F_\theta(\mathbf{x}), \quad (15)$$

where F_θ is constrained to generate a bracket-based dynamical system. Depending on the choice of bracket, this yields conservative, dissipative, or metriplectic flows, providing guarantees on stability and preventing oversmoothing.

3. **Decoding phase.** The final latent state $\mathbf{x}(T)$ is mapped back to physical space to producing node-level predictions.

The key differentiator between this architecture and standard GNNs lies in its message-passing phase: instead of a discrete stack of message-passing layers, depth is treated as a discretization of continuous time. Features are thus evolved according to a vector field F_θ defined in terms of an algebraic bracket. This bracket arises from variational considerations and enforces structural rigidity on the propagation of information, guaranteeing useful properties such as dynamical stability and the existence of a global invariant. Depending on the desired behavior, [14] implements four options for the message-passing dynamics, each conferring strict guarantees on a learnable notion of energy or entropy: Hamilton's least-action principle (conservative), a generalized gradient flow (totally dissipative), a double-bracket system (partially dissipative), or a metriplectic system (thermodynamically complete). The present work employs the energy-conserving Hamiltonian bracket proposed in [14], which performed best across our experiments.

To describe precisely how this works, recall that a (noncanonical) Hamiltonian system describes the evolution of a state variable \mathbf{x} via Hamilton's equations:

$$\dot{\mathbf{x}} = \mathbf{L}(\mathbf{x}) \nabla E(\mathbf{x}), \quad (16)$$

where E is a Hamiltonian function (i.e., total energy) of the state and \mathbf{L} is a skew-symmetric matrix field with additional structure.² In the present graph setting, the Hamiltonian E governs the latent node-edge feature pairs $\mathbf{x} = (\mathbf{q}, \mathbf{p}) \in \mathbb{R}^{(V+E) \times N_f}$ whose components exist in feature spaces equipped with state-varying inner product matrix fields $\mathbf{A}_0(\mathbf{q}) \in \mathbb{R}^{V \times V}$ and $\mathbf{A}_1(\mathbf{q}) \in \mathbb{R}^{E \times E}$. These inner products play the role of Riemannian metrics on the node/edge feature spaces and are specifically chosen to incorporate the influence of graph attention, which we discuss further below.

We define the combined node-edge inner product matrix field \mathbf{A} (for each \mathbf{q}) as:

$$\mathbf{A}(\mathbf{x}) = \text{diag}(\mathbf{A}_0(\mathbf{q}), \mathbf{A}_1(\mathbf{q})) \in \mathbb{R}^{(V+E) \times (V+E)}. \quad (17)$$

It then follows that the \mathbf{A} -gradient³ of any function $E(\mathbf{x}) \in \mathbb{R}^{V+E}$ is given by

$$\nabla^A E(\mathbf{x}) = \mathbf{A}(\mathbf{x})^{-1} \nabla E(\mathbf{x}), \quad (18)$$

and the \mathbf{A} -adjoint of any linear operator $\mathbf{L}(\mathbf{x}) \in \mathbb{R}^{(V+E) \times (V+E)}$ is given by

$$\mathbf{L}(\mathbf{x})^* = \mathbf{A}(\mathbf{x})^{-1} \mathbf{L}(\mathbf{x})^\top \mathbf{A}(\mathbf{x}), \quad (19)$$

(c.f. [14]). Choosing the particular expressions:

$$\mathbf{L}(\mathbf{x}) = \begin{pmatrix} 0 & -d_0^* \\ d_0 & 0 \end{pmatrix} \quad \text{and} \quad E(\mathbf{x}) = \frac{1}{2}(|\mathbf{q}|^2 + |\mathbf{p}|^2), \quad (20)$$

in terms of the graph gradient or incidence matrix $d_0 \in \mathbb{R}^{E \times V}$ of edges on nodes and its \mathbf{A} -adjoint $d_0^* = \mathbf{A}_0^{-1} d_0^\top \mathbf{A}_1$, it follows that the evolution equations $\dot{\mathbf{x}} = \mathbf{L}(\mathbf{x}) \nabla^A E(\mathbf{x})$, or:

$$\begin{pmatrix} \dot{\mathbf{q}} \\ \dot{\mathbf{p}} \end{pmatrix} = \begin{pmatrix} 0 & -d_0^* \\ d_0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{A}_0^{-1} & 0 \\ 0 & \mathbf{A}_1^{-1} \end{pmatrix} \begin{pmatrix} \mathbf{q} \\ \mathbf{p} \end{pmatrix} = \begin{pmatrix} -d_0^* \mathbf{A}_1^{-1} \mathbf{p} \\ d_0 \mathbf{A}_0^{-1} \mathbf{q} \end{pmatrix},$$

²Specifically, one usually enforces the Jacobi identity, a complicated PDE in the entries of \mathbf{L} , which is not explicitly incorporated in this parameterization.

³Here, we mean the gradient with respect to the derivative operator induced by the inner product involving \mathbf{A} .

generate Hamiltonian dynamics on the latent space containing \mathbf{x} . This can be seen from the fact that $\mathbf{L}^* = -\mathbf{L}$ is skew-adjoint for all \mathbf{x} , which guarantees that the instantaneous energy rate \dot{E} satisfies:

$$\begin{aligned}\dot{E}(\mathbf{x}) &= (\dot{\mathbf{x}}, \nabla^A E(\mathbf{x}))_{\mathbf{A}} \\ &= (\mathbf{L}(\mathbf{x}) \nabla^A E(\mathbf{x}), \nabla^A E(\mathbf{x}))_{\mathbf{A}} \\ &= -(\nabla^A E(\mathbf{x}), \mathbf{L}(\mathbf{x}) \nabla^A E(\mathbf{x}))_{\mathbf{A}} \\ &= 0,\end{aligned}\tag{21}$$

where $(\cdot, \cdot)_{\mathbf{A}} = \langle \cdot, \mathbf{A} \cdot \rangle$ denotes the \mathbf{A} -inner product on \mathbb{R}^{V+E} . The algebraic bracket $\{E, F\} = (\nabla E, \mathbf{L} \nabla F)_{\mathbf{A}}$ therefore guarantees that information flows along level sets of the Hamiltonian E , generalizing this consequence of classical mechanics to the graph setting and ensuring that node/edge feature updates align with Hamilton's principle of least action. We emphasize that the Hamiltonian E is a function on the latent features in this case, meaning that energy conservation in the original variables is not imposed. Rather, these latent Hamiltonian dynamics guarantee a global invariant on the layer-wise (i.e., discrete time) propagation, serving to stabilize information flow during message passing and prevent oversmoothing of the latent features.

It remains to explain how the learnable inner products $\mathbf{A}_0, \mathbf{A}_1$ implement graph attention. Letting $N_f > 0$ denote the latent dimension of the node features (e.g, ice thickness, basal friction, etc.), the entries of \mathbf{A}_1 are defined in terms of “query vectors” \mathbf{Wq}_i and “key vectors” \mathbf{Kq}_i coming from learnable linear embeddings $\mathbf{W}, \mathbf{K} \in \mathbb{R}^{N_h \times N_f}$, whose images are contained in a “hidden feature space” \mathbb{R}^{N_h} . For edge $\alpha = (i, j)$, we have:

$$[\mathbf{A}_1(\mathbf{q})]_{\alpha\alpha} = \exp(\mathbf{Wq}_i \cdot \mathbf{Kq}_j + \mathbf{Wq}_j \cdot \mathbf{Kq}_i).\tag{22}$$

Notice that $\mathbf{A}_1 \in \mathbb{R}^{E \times E}$ is diagonal (hence symmetric) with strictly positive entries in the space of graph edges, leading to a valid inner product on \mathbb{R}^E . However, it can also be represented with nodal indices by defining a matrix field $\mathbf{a}_1(\mathbf{q}) \in \mathbb{R}^{V \times V}$ with entries $[\mathbf{a}_1(\mathbf{q})]_{ij} = [\mathbf{A}_1(\mathbf{q})]_{\alpha\alpha}$, in which case it remains symmetric but contains off-diagonal terms. The nodal inner product \mathbf{A}_0 is then defined as the sum of incident edge weights:

$$[\mathbf{A}_0(\mathbf{q})]_{ii} = \sum_{j \in \mathcal{N}(i)} [\mathbf{a}_1(\mathbf{q})]_{ij} = \sum_{j \in \mathcal{N}(i)} \exp(\mathbf{Wq}_i \cdot \mathbf{Kq}_j + \mathbf{Wq}_j \cdot \mathbf{Kq}_i).\tag{23}$$

Again, it can be seen that \mathbf{A}_0 is diagonal with strictly positive entries. Moreover, it follows that the usual node-level attention (with symmetrized numerator) is recoverable via the expression $\text{att}(\mathbf{q}_i, \mathbf{q}_j) = [\mathbf{A}_0(\mathbf{q})^{-1} \mathbf{a}_1(\mathbf{q})]_{ij}$, and it can be shown that the “attention Laplacian” $\Delta = d_0^* d_0$ satisfies the following graph attention network (GAT)-like update:

$$[\Delta \mathbf{q}]_i = \sum_{j \in \mathcal{N}(i)} \text{att}(\mathbf{q}_i, \mathbf{q}_j) (\mathbf{q}_i - \mathbf{q}_j).\tag{24}$$

Importantly, the attention Laplacian Δ incorporates both topological information coming from the graph domain as well as metric information coming from the nodal representations, as opposed to the usual graph Laplacian which only accounts for connectivity. Therefore, the differential operators computed with the inner products $\mathbf{A}_0, \mathbf{A}_1$ naturally mimic key properties of graph attention while maintaining the conservation properties required for stable, long-term physical simulations.

3.3 Data preparation and graph generation

Each MALI simulation generates 93 annual snapshots of ice sheet state variables. We treat each snapshot as an independent training sample, and we randomly select 20 simulations for validation and 20 for testing.

As described in Section 2, MALI discretizes the thickness and temperature equations on a Voronoi mesh and solves the velocity equations on its dual Delaunay triangulation using low-order Lagrangian finite elements. We adopt this dual triangulation as the graph on which our GNN surrogate operates: the Delaunay nodes correspond to velocity degrees of freedom, and the FEM connectivity defines graph edges (Figure 1). This yields a fixed graph with 18,544 nodes and 54,962 edges.

Operating directly on this graph has two main advantages. First, it is naturally compatible with the unstructured meshes used in modern ice sheet models. Because GNNs act on node-edge structures and are agnostic to spatial ordering, no remeshing, interpolation, or other geometric preprocessing is required if the domain or mesh were to change. This stands in contrast to surrogate models based on regular grids, such as GNNs, which

require substantial data transformation. Second, the graph-based formulation enables transferability: a GNN trained on one domain or mesh can be fine-tuned on a related graph. For dynamic, plastic systems such as ice sheets, where domains and meshes may evolve, this provides a major computational advantage. We return to this point in Section 4.

Each node is assigned a seven-dimensional feature vector consisting of ice thickness, bed topography (or elevation), basal friction, and two Boolean indicators for grounded versus floating ice. Figure 2 shows a sample of these features. The model outputs are the x - and y - components of the velocity, which we also report in magnitude for visualization.

We normalize all scalar node and edge features using z-score normalization, except for the computed edge-wise distances, which use a min-max normalization to avoid distortion. For a given node feature Z , the normalized value z' is

$$z' = \frac{z - \mu_\Omega(Z)}{\sigma_\Omega(Z)}, \quad (25)$$

where μ_Ω and σ_Ω denote the global mean and standard deviation of Z over the full domain Ω . This centers each feature and places it on a comparable scale without distorting relative differences between data points.

Geometric information enters through edge-wise distances between node coordinates. These distance features are scaled to the fixed range [0,1] using min-max normalization,

$$d' = \frac{d - \min_\Omega d}{\max_\Omega d - \min_\Omega d}, \quad (26)$$

to preserve their relative magnitudes while keeping them numerically well-conditioned.

Global statistics $\mu_\Omega, \sigma_\Omega, \min_\Omega$ and \max_Ω are computed once and reused across all experiments, including pre-training and fine-tuning on different subgraphs, to ensure consistency. Training is performed on normalized features and targets, while all reported test metrics are computed using de-normalized velocity predictions and ground truth.

3.4 Training details

We learn the parameter set θ of the GNN surrogate by minimizing the mean squared error (MSE) between predicted and simulated velocities:

$$\mathcal{L}(\theta) = \frac{1}{N_\beta N_\tau} \sum_{i=1}^{N_\beta} \sum_{j=1}^{N_\tau} \sum_{\mathbf{x} \in \Omega} (\mathbf{u}_\theta(\mathbf{x}, t^j; \beta_i) - \mathbf{u}(\mathbf{x}, t^j; \beta_i))^2, \quad (27)$$

where we train on N_τ randomly selected time steps t^j , $j = 1, \dots, N_\tau$, for N_β independent basal friction fields β_i , $i = 1, \dots, N_\beta$. MSE is the standard choice for GNN regression problems and heavily penalizes large errors, which is desirable in our setting: the largest absolute errors typically occur in fast-flow regions near the glacier terminus, where accurate predictions are most critical. The encoder, decoder, internal attention and bracket parameters of the GNN are all learned as part of θ . We train the network using mini-batch gradient descent on a cluster equipped with NVIDIA A100 40GB GPUs.

We deliberately do not include any physics-based regularization terms in the loss (for example, penalties enforcing mass conservation residuals). The GNN architecture itself encodes useful structural biases, although this primarily serves to stabilize information flow in the latent dynamics and only implicitly encourages physically plausible behavior. Combined with our rich feature set, this is sufficient to produce stable and accurate predictions.

The physics-inspired bracket-based GNN architecture we use, based on [14], is highly configurable, allowing for different choices of brackets, ODE integrators, and encoding/decoding strategies. Our experiments indicate that architectural variations of this type are far less impactful than feature design and hyperparameter tuning (e.g. the learning rate scheduler), though they do affect training efficiency. We provide additional details on model parameters and runtimes in Section A. Figure 3 summarizes the overall training pipeline, including the modifications that implement our transfer-learning and domain decompositon strategies. These techniques are discussed in detail in Sections 4 and 5, respectively.

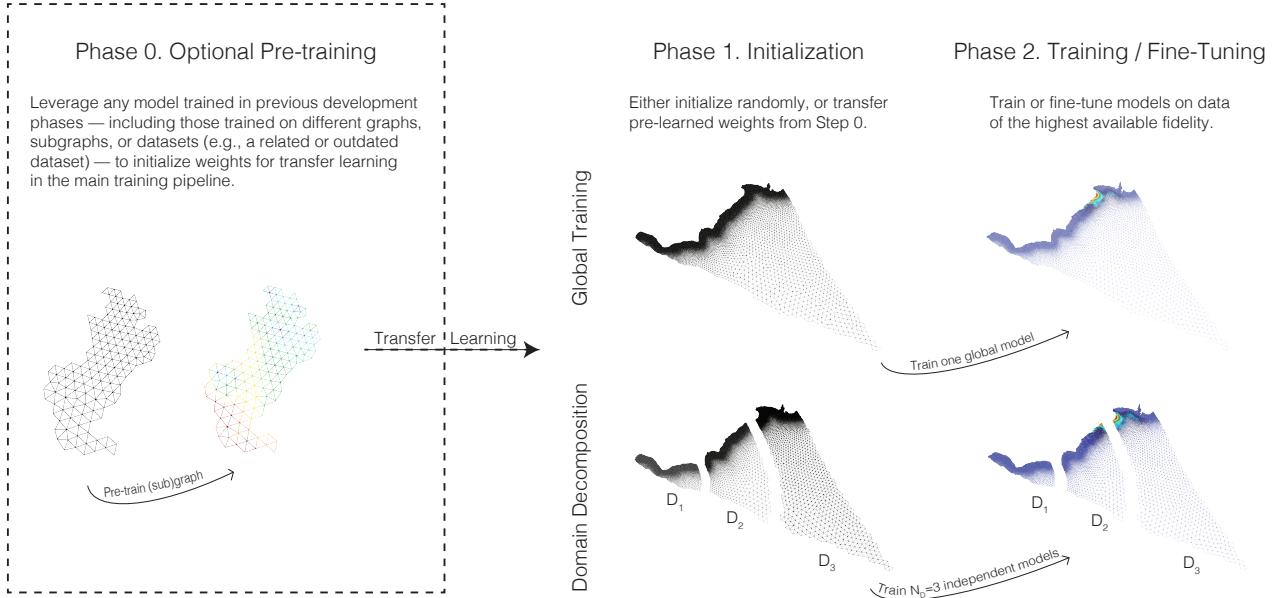


Figure 3: Overview of training pipeline, with different strategies considered. Phase 0 is an optional pre-training step that utilizes work already completed during model development, or abundant data pertaining to another related graph (e.g. a subgraph of the ice sheet of interest, or a completely different ice sheet).

4 Transfer learning

A key advantage of a graph-based surrogate modeling strategy is the ability to learn the attention mechanism on one part of the domain and reuse this knowledge elsewhere on the graph. This idea, commonly referred to as transfer learning or fine-tuning, can substantially reduce training time relative to training from scratch, and it enables surrogate modeling in regimes where high-fidelity data are scarce.

Transfer learning has been widely successful in improving predictive performance when only limited high-fidelity training data are available, even if the bulk of the training has been performed on coarser or lower-dimensional data [47]. Transferring weights from previously trained models is also known to accelerate convergence on new tasks [23], even when the learning problems differ significantly. In the context of GNNs, several transfer strategies are possible, but one of the most intuitive and effective is to pre-train on low-fidelity (or auxiliary) data and then fine-tune on a smaller set of high-fidelity samples [4]. This is the general strategy we adopt.

At a high level, our fine-tuning procedure consists of three steps:

1. **Pre-training.** Learn parameters θ_0^* on an initial graph G_0 , which may be a subgraph of the full domain or a related graph from a different simulation setup.
2. **Parameter transfer.** Initialize a new model on graph G_1 . All learnable parameters θ (encoder, decoder, and attention/bracket weights) are initialized from the pre-trained model on G_0 . If G_1 differs from G_0 , (e.g. a different mesh), all graph-dependent operators (e.g. incidence matrices d_0 , Laplacians, etc.) are recomputed from the topology of G_1 .
3. **Fine tuning.** Continue training the new model with parameters θ_1 on G_1 using the available high-fidelity data.

This workflow can be applied whether the pre-trained model lives on a different graph or on the same graph but with different data (e.g. with a different mesh resolution or a different representation of the basal friction field). Transfer learning is therefore particularly attractive in settings where the training data may be improved over time, for example, as new forward models, inversion algorithms, or observations become available. A model initially trained on outdated or coarse data can still be extremely valuable: it provides a strong initialization that both accelerates training on the new data and reduces the amount of additional data required for training.

Operationally, step 2 above involves extracting the learnable attention functions \mathbf{A}_0 and \mathbf{A}_1 (see (22) and (23) in Section 3.2) from the pre-trained model and using them, together with the encoder and decoder weights, to initialize the new model. Step 3 then fine-tunes this model on the target graph. In principle, one could restrict the update to a low-rank correction or to a subset of layers; in this work, we allow all parameters to adapt during the fine-tuning stage. Figure 3 illustrates one concrete realization of this strategy, in which we pre-train on a subdomain of the glacier and then use the resulting weights to provide a “warm start” for training a global model or additional subdomain models.

5 Domain decomposition

A key contribution of this work is the incorporation of domain decomposition (DD) into the GNN training pipeline as a mechanism for improving both computational efficiency and accuracy. At a high level, our DD approach consists of partitioning the full domain into subdomains, training an independent model on each, and then aggregating the independent subdomain predictions. DD has been extensively studied in classical numerical analysis [10, 48, 56, 59]. However, DD for neural networks, and for GNNs in particular, is still in its relative infancy. Recent work has started to blend DD and machine learning [18, 31], including both DD for ML and ML-enhanced DD, but the consideration of GNNs is restricted to the learning of parameters for classical DD solvers. Similarly, [58] use GNNs to learn DD parameters, but does not consider DD as a tool for training GNN surrogates themselves. The discrete nature of graph-based learning, combined with the flexibility of attention mechanisms, suggests that DD for GNNs offers unique potential.

In classical numerical simulations, DD yields iterative methods that can be viewed as a hybrid between basic iterative solvers and direct solvers. For large problems, a direct solve becomes infeasible due to superlinear growth in computational and memory cost. Standard iterative methods avoid this cost but often suffer from deteriorating convergence as the global problem size increases. DD methods strike a balance: each iteration splits the problem into subdomain solves, where direct or approximate solvers can be used efficiently, then couples them via coupling at the interfaces or overlap between the subdomains. This yields better convergence than basic iterative schemes and, with an appropriate coarse level, can even achieve numerical scalability, where iteration counts are essentially independent of global problem size. Crucially, DD localizes the computational work, enabling parallel scalability on modern high-performance architectures.

Localization is also one of the main benefits of DD in the context of neural networks. Localization can arise through the model architecture — for example, via mixtures of experts or locally connected networks [9, 17, 24, 26, 28, 40, 53] — or through the data, as in convolutional models trained on image patches [15, 30, 43, 61]. Beyond computational gains, localization has an interesting effect on learning dynamics. Neural networks exhibit a spectral bias, or frequency principle [49]: high-frequency functions are typically learned more slowly than low-frequency ones, in part due to the global nature of commonly used activation functions [22]. By decomposing the domain, DD effectively localizes these functions, allowing high-frequency behavior to be represented and learned within smaller subproblems. This can help mitigate spectral bias [9, 40].

In the present work, we employ a combination of model and data decomposition. On the data side, we decompose the domain in the form of the underlying graph (mesh), partitioning the ice sheet into subgraphs. On the model side, we train localized GNN surrogates on each subgraph and then aggregate their predictions into a global field. In this respect, our work shares some characteristics with localized CNN approaches for image decomposition [30], where a specialized model is trained on each part of a decomposed image. However, while [30] considers an image classification task using CNNs and aggregates submodel predictions into a single label using another model, we address a node-wise regression problem on an irregular graph. Other related works [43, 61] for CNNs learn a single global model via weight sharing, whereas our approach explicitly trains separate local surrogates and combines them at inference time.

These design choices motivate the developments in the following subsections. Section 5.1 describes how we construct physically coherent subdomains through a spectral clustering-based partitioning of the graph, and Section 5.2 details how we train and aggregate the subdomain models to obtain a consistent global prediction.

5.1 Partitioning of the graph

A central challenge in DD is how to partition the domain. In classical methods, this is typically done using geometric information or structure, or via graph partitioning algorithms [29, 7]. In the context of NNs, this topic still remains much less explored. Although recent work, e.g. [60], offers some guidance, the question of how best

to partition a graph for DD on GNNs remains relatively unexplored. For our purposes, an effective partition must satisfy three criteria: subdomains must be contiguous, approximately balanced in size, and aligned with meaningful physical variation in the data. Although we leave a thorough treatment of this topic for future work, we make a first step towards addressing it using a spectral-clustering approach that blends spatial, feature, and target similarity with a modified k -means procedure to encourage approximately balanced, physically coherent subdomains.

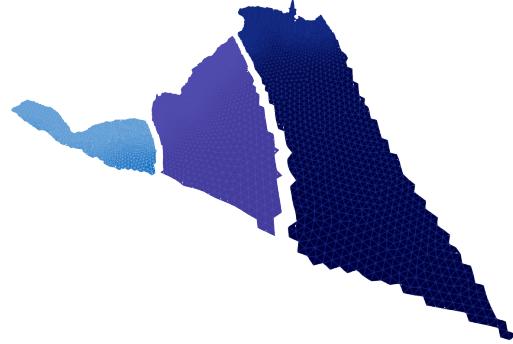


Figure 4: Subdomains produced by spectral clustering algorithm, for $k = 3$.

We partition the mesh by first building a symmetric weighted adjacency matrix W on the existing mesh edges so that contiguity is preserved and only neighboring cells interact. We borrow from the field of computer vision, and the work of [54] in particular, weighting each edge (u, v) with a Gaussian similarity that combines spatial proximity with similarity in both node features and target values, yielding higher affinity for nearby nodes with similar covariates and velocities. Concretely, letting $\Delta x_{uv} \in \mathbb{R}^2$ be the difference in spatial coordinates, $\Delta f_{uv} \in \mathbb{R}^3$ be the difference in features (thickness, friction, and bed topography), and $\Delta y_{uv} \in \mathbb{R}^2$ be the difference in velocity components (all computed on averages across training realizations), the resulting edge weight is:

$$W_{uv} = \exp\left(\frac{\|\Delta x_{uv}\|}{\sigma_x^2}\right) \exp\left(\frac{\|\Delta f_{uv}\|}{\sigma_f^2}\right) \exp\left(\frac{\|\Delta y_{uv}\|}{\sigma_y^2}\right), \quad (28)$$

yielding a sparse W supported on mesh edges. From W , we form the weighted graph Laplacian $L = D - W$ and compute its smallest nontrivial eigenvectors, which provide a spectral embedding in which Euclidean distances reflect relaxed cut objectives [41]. We then cluster the embedded points with a size-penalized k -means algorithm that softly balances subdomain sizes by adding a small bias to each cluster's squared distance proportional to its deviation from N/k points. This procedure promotes spatial coherence (because support is restricted to mesh edges), aligns boundaries with covariate and velocity structure, and avoids highly imbalanced partitions. In this work, we proceed with the clustering produced with $k = 3$, shown in Figure 4. Importantly, the subdomain boundaries of this partition roughly align with the direction of the velocity gradient, and ensure that each partition exhibits suitable variation in each feature.

Although overlapping subdomains are often beneficial in the classical DD setting, where they allow information about the PDE's nullspace to propagate across interfaces, we did not observe similar advantages in our setting. Prior work on DD with CNNs required either explicit overlap or a coarse global model to mitigate interface effects and ensure consistency across subdomains. In contrast, our GNN surrogate remains remarkably robust even with non-overlapping partitions: we do not see error growth near subdomain boundaries, nor do we observe the kinds of interface discontinuities one might expect when models are trained independently. This suggests that, because our GNN surrogate is fully data-driven and optimized under an MSE objective rather than solving a PDE directly, there is no analogous need to transport nullspace information across subdomains. We therefore adopt non-overlapping partitions based on mesh elements, which simplifies the training pipeline while retaining accuracy. Overlapping partitions could still be useful in regimes with extremely small neighborhoods or when the receptive field is severely truncated at partition boundaries, but such issues did not arise in our experiments.

Algorithm 1: Domain-decomposed training of N_D sub-domain GNNs

Require: Partitioned training sets $\{\mathcal{D}_i^{\text{train}}\}_{i=1}^{N_D}$;
 Global validation and test sets \mathcal{D}^{val} and $\mathcal{D}^{\text{test}}$;
 Normalization statistics $(\mu_x, \sigma_x, \mu_y, \sigma_y)$;
 Hyper-parameters opt (epochs, learning rate, warm start, ...)
Ensure : Trained weights $\{\Theta_i\}_{i=1}^{N_D}$, loss history, best global prediction on \mathcal{D}^{val}

for $i \leftarrow 1$ **to** N_D **do**

- $\mathcal{D}_i^{\text{train}} \leftarrow \text{NORMALIZEZ}(\mathcal{D}_i^{\text{train}}, \mu_x, \sigma_x, \mu_y, \sigma_y)$;
- Initialize Θ_i (optionally from pre-trained weights);

Normalize validation inputs (z-score): $\mathbf{X}^{\text{val}} \leftarrow \text{NORMALIZEZ}(\mathbf{X}^{\text{val}}, \mu_x, \sigma_x)$;

Define node-wise weights $w_{i,v}$ (for overlaps);

for $e \leftarrow 1$ **to** opt.num_epochs **do**

- for** $i \leftarrow 1$ **to** N_D **do**
- $\text{TRAINEPOCH}(\Theta_i, \mathcal{D}_i^{\text{train}})$;

Set aggregate prediction $\hat{\mathbf{Y}} \leftarrow 0$ and accumulation weights $\mathbf{W} \leftarrow 0$;

for $i \leftarrow 1$ **to** N_D **do**

- $\hat{\mathbf{Y}}_i \leftarrow \text{PREDICT}(\Theta_i, \mathbf{X}^{\text{val}}[\text{subgraph}_i])$;
- Undo normalization: $\hat{\mathbf{Y}}_i \leftarrow \hat{\mathbf{Y}}_i^{(z)} \sigma_y + \mu_y$;
- Add to aggregate: $\hat{\mathbf{Y}}[\text{subgraph}_i] \leftarrow \hat{\mathbf{Y}}[\text{subgraph}_i] + w_{i,v} \hat{\mathbf{Y}}_i$;
- Update weights: $\mathbf{W}[\text{subgraph}_i] \leftarrow \mathbf{W}[\text{subgraph}_i] + w_{i,v}$;

Weighted averaging for overlaps: $\hat{\mathbf{Y}} \leftarrow \hat{\mathbf{Y}} \oslash \mathbf{W}$ (element-wise);

Evaluate global validation loss; if improved, save $\{\Theta_i\}$ and log metrics;

After training, report final performance on held-out test set $\mathcal{D}^{\text{test}}$;

5.2 Training with subdomains

Having identified N_D suitable subdomains, we train N_D independent GNN surrogates using the procedure described in Algorithm 1. Training of the subdomain models closely mirrors the global training strategy described in Section 3.4, with three main differences: (i) reduced memory cost per subgraph, (ii) opportunities for parallelization across subdomains, and (iii) the need to aggregate subdomain predictions together at inference time.

For non-overlapping partitions, where each node belongs to exactly one subdomain, the aggregation step is trivial: the global prediction is obtained by placing each subdomain’s predictions onto its corresponding portion of the full graph, with no further processing required. For overlapping partitions, however, multiple subdomain models may produce predictions for the same node. In this case, we assign each node v a set of weights $\{w_{i,v}\}$, where $w_{i,v}$ specifies the contribution of subdomain i to the final aggregated prediction at node v . If $\hat{y}_{i,v}$ denotes the prediction from subdomain i at node v , the aggregated prediction is:

$$\hat{y}_v = \frac{\sum_{i=1}^{N_D} w_{i,v} \hat{y}_{i,v}}{\sum_{i=1}^{N_D} w_{i,v}}. \quad (29)$$

Setting $w_{i,v} = 1$ for all contributing subdomains yields a simple average over the predictions associated with node v . In the non-overlapping case, only a single subdomain contains node v , so the formula reduces to $\hat{y}_v = \hat{y}_{i,v}$; thus the aggregation step is entirely non-intrusive. Other weighting choices are possible, however, and could be designed to emphasize particular subdomains near boundaries or reflect local uncertainty estimates, for example. Figure 3 presents a schematic overview of the domain-decomposed training and aggregation process for the case of $N_D = 3$ non-overlapping subdomains.

6 Results

We evaluate the performance of our physics-inspired GNN surrogate and investigate how transfer learning and domain decomposition influence training efficiency and predictive accuracy. A high-level visual overview of our

training strategies (cold start, warm start, and warm start combined with DD) is provided in Figure 3. Figure 5 presents direct, side-by-side qualitative comparisons of their predictions on the same test snapshot.

We begin with the most challenging case as a baseline: training a single global model from scratch (“cold start”). With 25 basal friction fields and 40 snapshots per field (1000 samples total), the model successfully captures the large-scale structure of the velocity field. Even in this data-limited regime, errors concentrate near the grounding line and fast-flow terminus region, where velocities are highest and gradients are steepest. This can be seen by comparing the first and second rows in Figure 5.

Next, we assess “warm starts,” where a model is pre-trained on a subgraph and then fine-tuned globally. As shown in Figure 6, pre-training consistently accelerates convergence and reduces or maintains error across all data-regime conditions. When data are scarce (e.g. only 5-10 basal friction realizations), fine tuning provides particularly large gains. Compared to the cold start strategy, transfer learning plus fine-tuning achieves lower test error under both constrained training time (fixed epochs⁴) and constrained data (fixed training samples⁵). The second row of Figure 5 demonstrates that error magnitude decreases markedly relative to the cold-start model, especially near the terminus.

Finally, we combine transfer learning with domain decomposition. We partition the domain into subgraphs, pre-train on one region, and then transfer the learned parameters into the remaining subdomain models before fine-tuning. This strategy yields the fastest convergence (Figure 7) and the highest predictive accuracy (Figure 5, bottom row) of all methods we tested. Predictions from this combined transfer learning and DD strategy (Figure 5) show that errors near the terminus are nearly eliminated, with strong predictive accuracy even in regions of high velocity. The biggest improvements arise when pre-training uses the northeastern terminus region (rightmost subdomain in Figure 4), where velocity variability is greatest; initializing subdomain models from this high-complexity region transfers richer local structure and produces notably improved performance, as seen in Figure 7.

In summary, while all strategies eventually produce qualitatively correct predictions, transfer learning — and especially transfer learning combined with DD — dramatically improves both efficiency and accuracy. These results demonstrate the substantial computational savings and performance gains obtained by leveraging graph structure, localized dynamics, and warm-start initialization in large-scale surrogate modeling.

⁴This can be seen by comparing the error achieved by the dashed and solid lines at a given epoch.

⁵This can be seen by comparing dashed and solid lines of the same color.

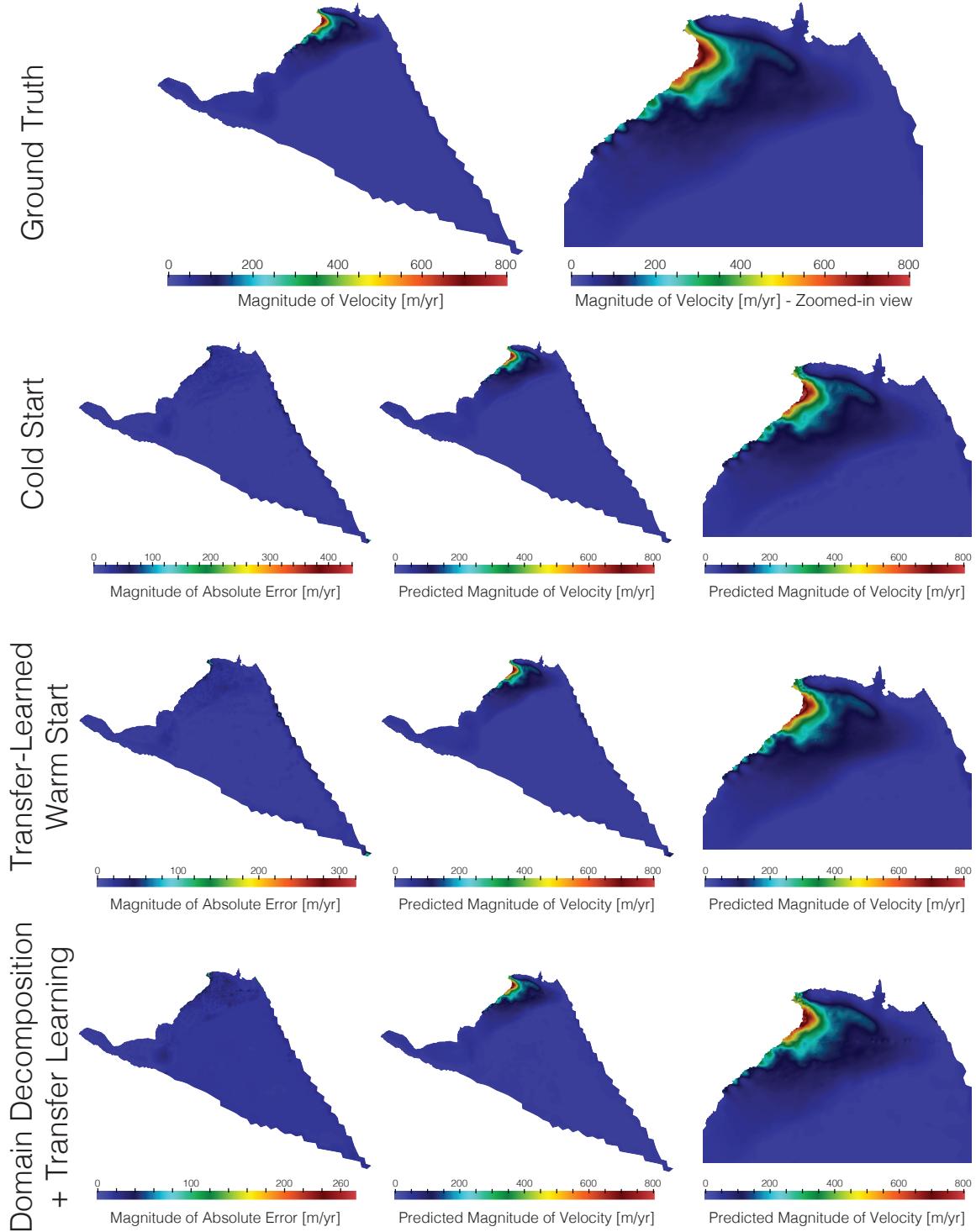


Figure 5: Comparison of three training strategies on the same test snapshot. Top row: ground-truth velocity magnitude (full domain and zoomed terminus view). Rows 2-4 show: (i) cold start, (ii) warm start (pre-trained on a subgraph and fine-tuned globally), and (iii) warm start plus DD, where subdomain models are pre-trained, fine-tuned, and stitched at inference. Columns display the magnitude of pointwise velocity error, full-field velocity prediction, and zoomed-in velocity prediction. The error color scale is intentionally not fixed across rows to highlight improvements: the overall magnitude of error decreases substantially from cold start to warm start to warm start plus DD, demonstrating progressively stronger accuracy, particularly in the terminus region.

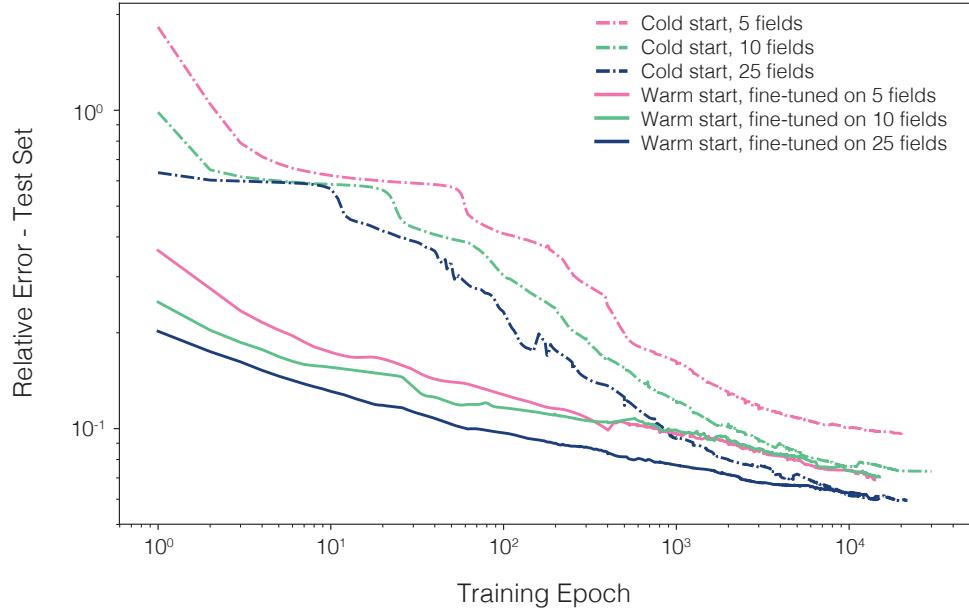


Figure 6: Relative test error for cold start and warm start strategies, trained on different numbers of basal friction fields. Warm start indicates fine-tuning from a model pre-trained on 40 friction fields for the southern interior region of the domain. Fine-tuning consistently accelerates training and lowers error compared to training from scratch.

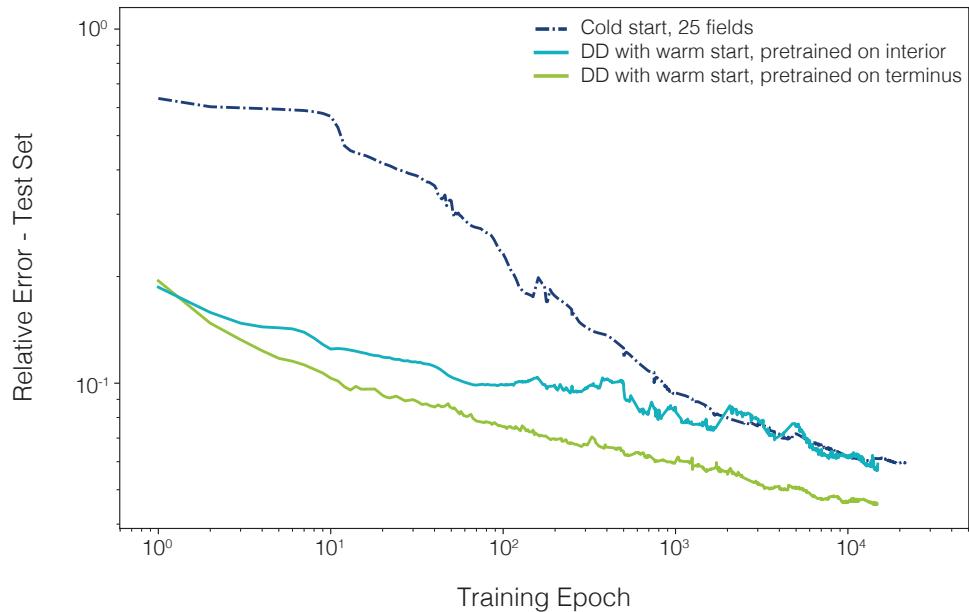


Figure 7: Relative test error for domain decomposition (DD) with warm start compared to global cold start. Pre-training on the terminus region (northeastern subgraph) yields faster convergence and lower error than pre-training on the interior (southernmost subgraph).

7 Discussion

We have presented a framework for scalable surrogate modeling of large-scale PDE-governed systems, motivated by the pressing need for efficient yet reliable projections of complex physical phenomena. Our approach combines GNNs, transfer learning, and domain decomposition to address the challenges inherent in training surrogates on large, unstructured meshes. By partitioning complex domains into manageable subgraphs, we enable training at scale while maintaining stability and physical accuracy.

Our results demonstrate that domain decomposition offers both computational and modeling benefits. Training on subdomains reduces memory requirements and training time, but it also provides a structural advantage: each subdomain forms a natural unit for knowledge transfer. Information learned in one region of the domain can be re-used and adapted in others, accelerating convergence and improving generalization. This perspective aligns with multifidelity learning, where inexpensive or partially aligned data sources are leveraged to boost performance on expensive tasks. In settings such as ice sheet modeling, where high-fidelity simulations remain prohibitively expensive for uncertainty quantification, these strategies offer a path toward practical, data-efficient surrogates.

Together, these elements lay the groundwork for a new generation of graph-based surrogates for large-scale physical systems. By exploiting domain structure and physics-inspired latent dynamics, we move closer to ML models that are not only computationally efficient but also scientifically trustworthy. More broadly, our results suggest that graph-based domain decomposition is a practical and versatile tool for building surrogates that remain accurate across heterogeneous regions, changing meshes, and evolving modeling objectives.

7.1 Towards foundation models for uncertainty quantification

A particularly promising direction, highlighted by our preliminary experiments on grounding-line flux, is the use of general-purpose surrogate models that can be lightly fine-tuned for downstream UQ tasks. When trained on many distinct basal-friction fields, our model learns a representation that captures the average velocity response across basal friction conditions, yielding accurate full-field predictions. This average-case accuracy emerges naturally from training on a shared latent structure across realizations. However, when the goal shifts from predicting mean behavior to capturing the variability induced by uncertain inputs, the baseline model alone may be insufficient. As illustrated by our grounding-line flux example (Figure 8, left), the untuned surrogate model correctly reproduces the true mean grounding line flux but fails to capture the spread across basal friction realizations: the predicted distribution is too narrow, even though the mean is correct.

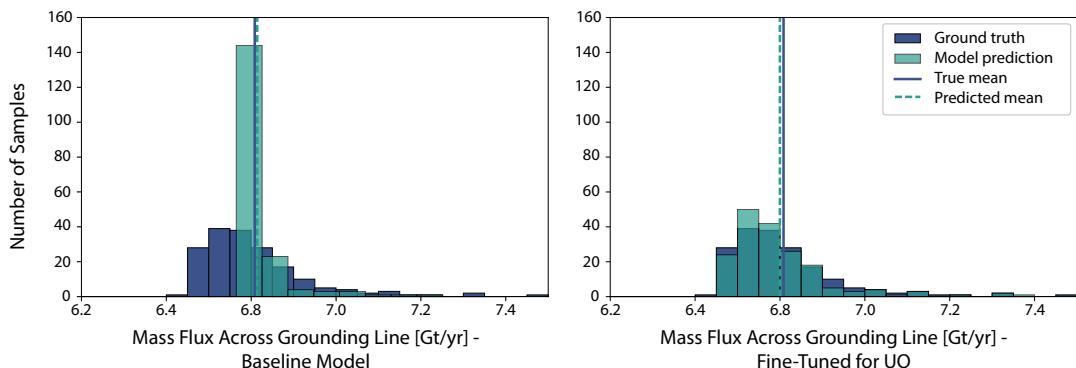


Figure 8: Histograms of the distribution of ice mass flux across grounding line at $t = 60$ years across 180 realizations of basal friction. Left: results using baseline high-performing model. Right: results after fine-tuning on curated UQ training set.

Introducing a targeted fine-tuning phase changes this picture. When we continue training on a small dataset in which each sample is a single snapshot drawn from a different basal-friction realization (rather than our usual training sets that include many time steps from the same realization), the surrogate not only maintains an accurate estimate of the mean flux but also closely recovers the full distribution induced by basal-friction uncertainty (Figure 8, right). These results suggest that fine-tuning on data designed to emphasize variability

in the input space may encourage the model to pick up UQ-relevant sensitivity directions without retraining from scratch.

These preliminary results indicate that foundation-model behavior is attainable in this context: a pre-trained, domain-decomposed surrogate can be efficiently adapted to a new UQ objective with only modest additional data and computation. Looking ahead, such capabilities could bring ice sheet modeling closer to the “foundation model” paradigms emerging in other scientific domains, where a single, broadly trained surrogate can be rapidly specialized for new scientific questions, forcing scenarios, or UQ objectives. Developing principled strategies for the design of fine-tuning datasets and objectives for UQ is an important focus of ongoing and future work.

Acknowledgments

This work performed in part at Sandia National Laboratories and Pacific Northwest National Laboratory was supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research through the SEA-CROGS project (PNNL Project No. 80278). Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy’s National Nuclear Security Administration under contract DE-NA0003525. Pacific Northwest National Laboratory is a multi-program national laboratory operated for the U.S. Department of Energy by Battelle Memorial Institute under Contract No. DE-AC05-76RL01830. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

A Additional model details

In Table 1, we summarize the architecture and hyperparameter choices for the BracketGraph GNN used as our surrogate model. Overall, we found the model to be relatively insensitive to many of these settings. For instance, increasing the number of attention heads from 2 to 4 or the hidden dimension from 20 to 50 produced results that were indistinguishable from the defaults reported here. By contrast, shrinking the encoder and decoder widths from 32 to 16 approximately doubled the final relative error, indicating greater sensitivity to these layers.

With the configuration in Table 1, the “off-line” training cost of the surrogate is roughly 30 hours. This can be reduced by employing explicit integration schemes such as forward Euler, although at the expense of some loss in accuracy. Once the surrogate is trained, inference takes approximately 15 ms per sample.

Table 1: Graph neural network (GNN) architecture and hyperparameters.

Parameter	Value
Integrator method	Implicit Adams-Bashforth-Moulton
Bracket type	Hamiltonian
Input node feature dimension (d_{in})	5
Hidden dimension (d_{hid})	20
Output feature dimension (d_{out})	2
Number of Neural ODE timesteps	2
Message-passing encoder/decoder width	32
Attention heads	2
Optimizer	Adam
Learning rate	1e-3
Gamma	0.95
Step size	250

References

- [1] S. BISHNOI, R. BHATTOO, S. RANU, AND N. M. A. KRISHNAN, *Enhancing the Inductive Biases of Graph Neural ODE for Modeling Dynamical Systems*, June 2024, <https://doi.org/10.48550/arXiv.2209.10740>, <http://arxiv.org/abs/2209.10740> (accessed 2025-08-01). arXiv:2209.10740 [cs].
- [2] H. BLATTER, *Velocity and stress fields in grounded glaciers: a simple algorithm for including deviatoric stress gradients*, Journal of Glaciology, 41 (1995), pp. 333–344.
- [3] D. J. BRINKERHOFF, B. S. TOBER, M. DANIEL, V. DEVAUX-CHUPIN, M. S. CHRISTOFFERSEN, J. W. HOLT, C. F. LARSEN, M. FAHNESTOCK, M. G. LOSO, K. M. F. TIMM, R. C. MITCHELL, AND M. TRUFER, *The demise of the world's largest piedmont glacier: a probabilistic forecast*, The Cryosphere, 19 (2025), pp. 2321–2353, <https://doi.org/10.5194/tc-19-2321-2025>, <https://tc.copernicus.org/articles/19/2321/2025/>.
- [4] D. BUTEREZ, J. P. JANET, S. J. KIDDLE, D. OGLIC, AND P. LIÓ, *Transfer learning with graph neural networks for improved molecular property prediction in the multi-fidelity setting*, Nature communications, 15 (2024), p. 1517.
- [5] C. CAI AND Y. WANG, *A Note on Over-Smoothing for Graph Neural Networks*, June 2020, <https://doi.org/10.48550/arXiv.2006.13318>, <http://arxiv.org/abs/2006.13318> (accessed 2025-08-01). arXiv:2006.13318 [cs].
- [6] K. M. CUFFEY AND W. S. B. PATERSON, *The physics of glaciers*, Academic Press, 2010.
- [7] K. DEVINE, E. G. BOMAN, R. HEAPHY, B. HENDRICKSON, AND U. V. CATALYUREK, *Zoltan data management services for parallel dynamic applications*, Computing in Science & Engineering, 4 (2002), pp. 90–97.
- [8] T. DIAS DOS SANTOS, M. MORLIGHEM, AND D. BRINKERHOFF, *A new vertically integrated mono-layer higher-order (molho) ice flow model*, The Cryosphere, 16 (2022), pp. 179–195.
- [9] V. DOLEAN, A. HEINLEIN, S. MISHRA, AND B. MOSELEY, *Multilevel domain decomposition-based architectures for physics-informed neural networks*, June 2023, <https://doi.org/10.48550/arXiv.2306.05486>, <http://arxiv.org/abs/2306.05486> (accessed 2023-07-01). arXiv:2306.05486 [cs, math].
- [10] V. DOLEAN, P. JOLIVET, AND F. NATAF, *An Introduction to Domain Decomposition Methods: Algorithms, Theory, and Parallel Implementation*, Society for Industrial and Applied Mathematics, Philadelphia, PA, Nov. 2015, <https://doi.org/10.1137/1.9781611974065>, <http://pubs.siam.org/doi/book/10.1137/1.9781611974065> (accessed 2024-05-14).
- [11] J. GILMER, S. S. SCHOENHOLZ, P. F. RILEY, O. VINYALS, AND G. E. DAHL, *Neural message passing for quantum chemistry*, in International conference on machine learning, Pmlr, 2017, pp. 1263–1272.
- [12] J. W. GLEN, *The creep of polycrystalline ice*, Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences, 228 (1955), pp. 519–538.
- [13] D. N. GOLDBERG, P. HEIMBACH, I. JOUGHIN, AND B. SMITH, *Committed retreat of smith, pope, and kohler glaciers over the next 30 years inferred by transient model calibration*, The Cryosphere, 9 (2015), pp. 2429–2446, <https://doi.org/10.5194/tc-9-2429-2015>, <https://tc.copernicus.org/articles/9/2429/2015/>.
- [14] A. GRUBER, K. LEE, AND N. TRASK, *Reversible and irreversible bracket-based dynamics for deep graph neural networks*, Advances in Neural Information Processing Systems, 36 (2023), pp. 38454–38484.
- [15] L. GU, W. ZHANG, J. LIU, AND X.-C. CAI, *Decomposition and Preconditioning of Deep Convolutional Neural Networks for Training Acceleration*, in Domain Decomposition Methods in Science and Engineering XXVI, S. C. Brenner, E. Chung, A. Klawonn, F. Kwok, J. Xu, and J. Zou, eds., Lecture Notes in Computational Science and Engineering, Cham, 2022, Springer International Publishing, pp. 153–160, https://doi.org/10.1007/978-3-030-95025-5_14.

- [16] Q. HE, M. PEREGO, A. A. HOWARD, G. E. KARNIADAKIS, AND P. STINIS, *A hybrid deep neural operator/finite element method for ice-sheet modeling*, Journal of Computational Physics, 492 (2023), p. 112428.
- [17] A. HEINLEIN AND T. KAPOOR, *Domain decomposition architectures and Gauss-Newton training for physics-informed neural networks*, Oct. 2025, <https://doi.org/10.48550/arXiv.2510.27018>, <http://arxiv.org/abs/2510.27018> (accessed 2025-11-07). arXiv:2510.27018 [math].
- [18] A. HEINLEIN, A. KLAWONN, M. LANSER, AND J. WEBER, *Combining machine learning and domain decomposition methods for the solution of partial differential equations—a review*, GAMM-Mitteilungen, 44 (2021), pp. Paper No. e202100001, 28, <https://doi.org/10.1002/gamm.202100001>, <https://mathscinet.ams.org/mathscinet-getitem?mr=4235094> (accessed 2022-12-01).
- [19] T. R. HILLEBRAND, M. J. HOFFMAN, H. K. HAN, M. PEREGO, A. O. HAGER, A. NOLAN, X. ASAY-DAVIS, S. F. PRICE, J. WATKINS, AND M. CARLSON, *Evolution of the antarctic ice sheet from 2000–2300 and beyond: model sensitivity and uncertainty analysis using mpas-albany land ice*, EGUsphere, 2025 (2025), pp. 1–51, <https://doi.org/10.5194/egusphere-2025-3942>, <https://egusphere.copernicus.org/preprints/2025/egusphere-2025-3942/>.
- [20] T. R. HILLEBRAND, M. J. HOFFMAN, M. PEREGO, S. F. PRICE, AND I. M. HOWAT, *The contribution of humboldt glacier, northern greenland, to sea-level rise through 2100 constrained by recent observations of speedup and retreat*, The Cryosphere, 16 (2022), pp. 4679–4700, <https://doi.org/10.5194/tc-16-4679-2022>, <https://tc.copernicus.org/articles/16/4679/2022/>.
- [21] M. J. HOFFMAN, M. PEREGO, S. F. PRICE, W. H. LIPSCOMB, T. ZHANG, D. JACOBSEN, I. TEZAUR, A. G. SALINGER, R. TUMINARO, AND L. BERTAGNA, *Mpas-albany land ice (mali): a variable-resolution ice sheet model for earth system modeling using voronoi grids*, Geoscientific Model Development, 11 (2018), pp. 3747–3780.
- [22] Q. HONG, J. W. SIEGEL, Q. TAN, AND J. XU, *On the Activation Function Dependence of the Spectral Bias of Neural Networks*, Sept. 2022, <https://doi.org/10.48550/arXiv.2208.04924>, <http://arxiv.org/abs/2208.04924> (accessed 2025-05-03). arXiv:2208.04924 [cs].
- [23] A. HOWARD, Y. FU, AND P. STINIS, *A multifidelity approach to continual learning for physical systems*, Machine Learning: Science and Technology, 5 (2024), p. 025042.
- [24] A. A. HOWARD, B. JACOB, S. H. MURPHY, A. HEINLEIN, AND P. STINIS, *Finite basis Kolmogorov-Arnold networks: domain decomposition for data-driven and physics-informed problems*, June 2024, <https://doi.org/10.48550/arXiv.2406.19662>, <http://arxiv.org/abs/2406.19662> (accessed 2024-07-08). arXiv:2406.19662 [physics].
- [25] T. ISAAC, N. PETRA, G. STADLER, AND O. GHATTAS, *Scalable and efficient algorithms for the propagation of uncertainty from data through inference to prediction for large-scale problems, with application to flow of the antarctic ice sheet*, Journal of Computational Physics, 296 (2015), pp. 348–368, <https://doi.org/https://doi.org/10.1016/j.jcp.2015.04.047>, <https://www.sciencedirect.com/science/article/pii/S0021999115003046>.
- [26] A. D. JAGTAP, E. KHARAZMI, AND G. E. KARNIADAKIS, *Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems*, Computer Methods in Applied Mechanics and Engineering, 365 (2020), p. 113028, <https://doi.org/10.1016/j.cma.2020.113028>, <https://www.sciencedirect.com/science/article/pii/S0045782520302127> (accessed 2024-07-23).
- [27] J. D. JAKEMAN, M. PEREGO, D. T. SEIDL, T. A. HARTLAND, T. R. HILLEBRAND, M. J. HOFFMAN, AND S. F. PRICE, *An evaluation of multi-fidelity methods for quantifying uncertainty in projections of ice-sheet mass change*, Earth System Dynamics, 16 (2025), pp. 513–544, <https://doi.org/10.5194/esd-16-513-2025>, <https://esd.copernicus.org/articles/16/513/2025/>.
- [28] A. D. J. . G. E. KARNIADAKIS, *Extended Physics-Informed Neural Networks (XPINNs): A Generalized Space-Time Domain Decomposition Based Deep Learning Framework for Nonlinear Partial Differential Equations*, Communications in Computational Physics, 28 (2020), pp. 2002–2041, <https://doi.org/https://doi.org/10.1007/s40006-020-00750-w>

- 10.4208/cicp.0A-2020-0164, http://global-sci.org/intro/article_detail/cicp/18403.html (accessed 2024-07-23).
- [29] G. KARYPIS AND V. KUMAR, *A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs*, SIAM Journal on Scientific Computing, 20 (1998), pp. 359–392, <https://doi.org/10.1137/S1064827595287997>, <https://pubs-siam-org.tudelft.idm.oclc.org/doi/10.1137/S1064827595287997> (accessed 2024-07-15). Publisher: Society for Industrial and Applied Mathematics.
- [30] A. KLAWONN, M. LANSER, AND J. WEBER, *A Domain Decomposition-Based CNN-DNN Architecture for Model Parallel Training Applied to Image Recognition Problems*, Feb. 2023, <https://doi.org/10.48550/arXiv.2302.06564>, <http://arxiv.org/abs/2302.06564> (accessed 2023-02-14). arXiv:2302.06564 [cs].
- [31] A. KLAWONN, M. LANSER, AND J. WEBER, *Machine learning and domain decomposition methods-a survey*, Computational Science and Engineering, 1 (2024), p. 2.
- [32] Y. KOO AND M. RAHNEMOONFAR, *Graph convolutional network as a fast statistical emulator for numerical ice sheet modeling*, Journal of Glaciology, 71 (2025), p. e15, <https://doi.org/10.1017/jog.2024.93>.
- [33] R. LAM, A. SANCHEZ-GONZALEZ, M. WILLSON, P. WIRNSBERGER, M. FORTUNATO, F. ALET, S. RAVURI, T. EWALDS, Z. EATON-ROSEN, W. HU, A. MEROSE, S. HOYER, G. HOLLAND, O. VINYALS, J. STOTT, A. PRITZEL, S. MOHAMED, AND P. BATTAGLIA, *Learning skillful medium-range global weather forecasting*, Science, (2023), <https://doi.org/10.1126/science.adl2336>, <https://www.science.org/doi/10.1126/science.adl2336> (accessed 2025-07-27). Publisher: American Association for the Advancement of Science.
- [34] Q. LI, Z. HAN, AND X.-M. WU, *Deeper insights into graph convolutional networks for semi-supervised learning*, in Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence, AAAI'18/IAAI'18/EAAI'18, AAAI Press, 2018.
- [35] S. LI, X. HAN, AND J. BAI, *Adaptergnn: Parameter-efficient fine-tuning improves generalization in gnns*, in Proceedings of the AAAI conference on artificial intelligence, vol. 38, 2024, pp. 13600–13608.
- [36] D. R. MACAYEAL, *A tutorial on the use of control methods in ice-sheet modeling*, Journal of Glaciology, 39 (1993), p. 91–98, <https://doi.org/10.3189/S0022143000015744>.
- [37] A. MERCHANT, S. BATZNER, S. S. SCHOENHOLZ, M. AYKOL, G. CHEON, AND E. D. CUBUK, *Scaling deep learning for materials discovery*, Nature, 624 (2023), pp. 80–85, <https://doi.org/10.1038/s41586-023-06735-9>, <https://www.nature.com/articles/s41586-023-06735-9> (accessed 2025-07-27). Publisher: Nature Publishing Group.
- [38] N. S. MOORE, E. C. CYR, P. OHM, C. M. SIEFERT, AND R. S. TUMINARO, *Graph neural networks and applied linear algebra*, SIAM review, 67 (2025), pp. 141–175.
- [39] M. MORLIGHEM, E. RIGNOT, H. SEROUSSI, E. LAROUR, H. BEN DHIA, AND D. AUBRY, *Spatial patterns of basal drag inferred using control methods from a full-stokes and simpler models for pine island glacier, west antarctica*, Geophysical Research Letters, 37 (2010), <https://doi.org/10.1029/2010GL043853>, <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2010GL043853>, <https://arxiv.org/abs/https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2010GL043853>.
- [40] B. MOSELEY, A. MARKHAM, AND T. NISSEN-MEYER, *Finite Basis Physics-Informed Neural Networks (FBPINNs): a scalable domain decomposition approach for solving differential equations*, July 2021, <https://doi.org/10.48550/arXiv.2107.07871>, <http://arxiv.org/abs/2107.07871> (accessed 2023-01-19). arXiv:2107.07871 [physics].
- [41] A. NG, M. JORDAN, AND Y. WEISS, *On spectral clustering: Analysis and an algorithm*, Advances in neural information processing systems, 14 (2001).

- [42] F. PATTYN, *A new three-dimensional higher-order thermomechanical ice sheet model: Basic sensitivity, ice stream development, and ice flow across subglacial lakes*, Journal of Geophysical Research: Solid Earth, 108 (2003).
- [43] J. PELZER, C. VERBURG, A. HEINLEIN, AND M. SCHULTE, *Few-shot learning by explicit physics integration: An application to groundwater heat transport*, arXiv preprint arXiv:2507.06062, (2025).
- [44] M. PEREGO, S. PRICE, AND G. STADLER, *Optimal initial conditions for coupling ice sheet models to earth system models*, Journal of Geophysical Research: Earth Surface, 119 (2014), pp. 1894–1917, <https://doi.org/https://doi.org/10.1002/2014JF003181>, <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1002/2014JF003181>, <https://arxiv.org/abs/https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1002/2014JF003181>.
- [45] N. PETRA, H. ZHU, G. STADLER, T. J. HUGHES, AND O. GHATTAS, *An inexact gauss-newton method for inversion of basal sliding and rheology parameters in a nonlinear stokes ice sheet model*, Journal of Glaciology, 58 (2012), p. 889–903, <https://doi.org/10.3189/2012JoG11J182>.
- [46] T. PFAFF, M. FORTUNATO, A. SANCHEZ-GONZALEZ, AND P. BATTAGLIA, *Learning mesh-based simulation with graph networks*, in International conference on learning representations, 2020.
- [47] A. M. PROPP AND D. M. TARTAKOVSKY, *Transfer learning on multi-dimensional data: A novel approach to neural network-based surrogate modeling*, Journal of Machine Learning for Modeling and Computing, 6 (2025).
- [48] A. QUARTERONI AND A. VALLI, *Domain Decomposition Methods for Partial Differential Equations*, Numerical Mathematics and Scientific Computation, Oxford University Press, Oxford, New York, May 1999.
- [49] N. RAHAMAN, A. BARATIN, D. ARPIT, F. DRAXLER, M. LIN, F. A. HAMPRECHT, Y. BENGIO, AND A. COURVILLE, *On the Spectral Bias of Neural Networks*, May 2019, <https://doi.org/10.48550/arXiv.1806.08734>, <http://arxiv.org/abs/1806.08734> (accessed 2023-02-25). arXiv:1806.08734 [cs, stat].
- [50] B. RECINOS, D. GOLDBERG, J. R. MADDISON, AND J. TODD, *A framework for time-dependent ice sheet uncertainty quantification, applied to three west antarctic ice streams*, The Cryosphere, 17 (2023), pp. 4241–4266, <https://doi.org/10.5194/tc-17-4241-2023>, <https://tc.copernicus.org/articles/17/4241/2023/>.
- [51] T. K. RUSCH, M. M. BRONSTEIN, AND S. MISHRA, *A survey on oversmoothing in graph neural networks*, arXiv preprint arXiv:2303.10993, (2023).
- [52] A. SANCHEZ-GONZALEZ, V. BAPST, K. CRANMER, AND P. BATTAGLIA, *Hamiltonian Graph Networks with ODE Integrators*, Sept. 2019, <https://doi.org/10.48550/arXiv.1909.12790>, <http://arxiv.org/abs/1909.12790> (accessed 2025-08-01). arXiv:1909.12790 [cs].
- [53] Y. SHANG, A. HEINLEIN, S. MISHRA, AND F. WANG, *Overlapping Schwarz preconditioners for randomized neural networks with domain decomposition*, Computer Methods in Applied Mechanics and Engineering, 442 (2025), p. 118011, <https://doi.org/10.1016/j.cma.2025.118011>, <https://www.sciencedirect.com/science/article/pii/S004578252500283X> (accessed 2025-04-25).
- [54] J. SHI AND J. MALIK, *Normalized cuts and image segmentation*, IEEE Transactions on pattern analysis and machine intelligence, 22 (2000), pp. 888–905.
- [55] K. SHUKLA, M. XU, N. TRASK, AND G. E. KARNIADAKIS, *Scalable algorithms for physics-informed neural and graph networks*, Data-Centric Engineering, 3 (2022), p. e24.
- [56] B. SMITH, P. BJORSTAD, AND W. GROPP, *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*, Cambridge University Press, Mar. 2004. Google-Books-ID: dxwRLu1dBioC.
- [57] J. M. STOKES, K. YANG, K. SWANSON, W. JIN, A. CUBILLOS-RUIZ, N. M. DONGHIA, C. R. MACNAIR, S. FRENCH, L. A. CARFRAE, Z. BLOOM-ACKERMANN, V. M. TRAN, A. CHIAPPINO-PEPE, A. H. BADRAN, I. W. ANDREWS, E. J. CHORY, G. M. CHURCH, E. D. BROWN, T. S. JAAKKOLA, R. BARZILAY, AND J. J. COLLINS, *A Deep Learning Approach to Antibiotic Discovery*,

- Cell, 180 (2020), pp. 688–702.e13, <https://doi.org/https://doi.org/10.1016/j.cell.2020.01.021>, <https://www.sciencedirect.com/science/article/pii/S0092867420301021>.
- [58] A. TAGHIBAKHSI, N. NYTKO, T. U. ZAMAN, S. MACLACHLAN, L. OLSON, AND M. WEST, *MG-GNN: Multigrid Graph Neural Networks for Learning Multilevel Domain Decomposition Methods*, Mar. 2023, <https://doi.org/10.48550/arXiv.2301.11378>, <http://arxiv.org/abs/2301.11378> (accessed 2025-07-27). arXiv:2301.11378 [cs].
- [59] A. TOSELLI AND O. WIDLUND, *Domain decomposition methods—algorithms and theory*, vol. 34 of Springer Series in Computational Mathematics, Springer-Verlag, Berlin, 2005, <https://doi.org/10.1007/b137868>, <https://mathscinet.ams.org/mathscinet-getitem?mr=2104179> (accessed 2022-12-01).
- [60] N. TRASK, A. HENRIKSEN, C. MARTINEZ, AND E. CYR, *Hierarchical partition of unity networks: fast multilevel training*, in Proceedings of Mathematical and Scientific Machine Learning, B. Dong, Q. Li, L. Wang, and Z.-Q. J. Xu, eds., vol. 190 of Proceedings of Machine Learning Research, PMLR, 15–17 Aug 2022, pp. 271–286, <https://proceedings.mlr.press/v190/trask22a.html>.
- [61] C. VERBURG, A. HEINLEIN, AND E. C. CYR, *DDU-Net: A Domain Decomposition-Based CNN for High-Resolution Image Segmentation on Multiple GPUs*, IEEE Access, 13 (2025), pp. 66967–66983, <https://doi.org/10.1109/ACCESS.2025.3561033>, <https://ieeexplore.ieee.org/document/10965628> (accessed 2025-04-26).
- [62] U. VILLA AND T. O’LEARY-ROSEBERRY, *A note on the relationship between pde-based precision operators and matérn covariances*, 2024, <https://arxiv.org/abs/2407.00471>, <https://arxiv.org/abs/2407.00471>.
- [63] J. WATKINS, M. CARLSON, K. SHAN, I. TEZAUR, M. PEREGO, L. BERTAGNA, C. KAO, M. J. HOFFMAN, AND S. F. PRICE, *Performance portable ice-sheet modeling with mali*, The International Journal of High Performance Computing Applications, 37 (2023), pp. 600–625, <https://doi.org/10.1177/10943420231183688>, <https://doi.org/10.1177/10943420231183688>.
- [64] X. WU, Z. CHEN, W. W. WANG, AND A. JADBABAIE, *A non-asymptotic analysis of oversmoothing in graph neural networks*, in The Eleventh International Conference on Learning Representations, 2023, <https://openreview.net/forum?id=CJd-BtnwtXq>.
- [65] Z. WU, S. PAN, F. CHEN, G. LONG, C. ZHANG, AND P. S. YU, *A comprehensive survey on graph neural networks*, IEEE transactions on neural networks and learning systems, 32 (2020), pp. 4–24.
- [66] K. ZHOU, X. HUANG, Y. LI, D. ZHA, R. CHEN, AND X. HU, *Towards deeper graph neural networks with differentiable group normalization*, in Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS ’20, Red Hook, NY, USA, 2020, Curran Associates Inc.