PROBABILISTIC DIFFUSIONNET: A GEOMETRY INFORMED PROBABILISTIC GENERATIVE SURROGATE FOR PDES

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose a probabilistic generative extension of the DiffusionNet architecture, widely used for surface learning tasks, by introducing latent random variables derived from a stochastic reformulation of the underlying diffusion process. The resulting probabilistic model can be used as a resolution-invariant and uncertainty-aware surrogate for the trace solution map of PDEs whose boundary conditions are determined by surface geometry. Such a surrogate can expedite and inform typical engineering design and optimisation processes that require computationally burdensome computational fluid dynamics (CFD) analysis pipelines. We demonstrate that the proposed architecture produces superior uncertainty quantification (UQ) performance on standard CFD datasets without sacrificing predictive accuracy, while enjoying lower computational cost compared to other prevalent geometry-informed PDE surrogates endowed with UQ capabilities.

1 INTRODUCTION

Across science and engineering it is often needed to produce solutions to non-linear partial differential equations (PDEs) with boundary conditions, as they describe the behaviour of a wide range of physical phenomena, from fluid dynamics and hydro-electromagnetics to relativistic gravitation. During the typical design process it is necessary to solve these PDEs for changing geometric conditions, which puts a time-consuming numerical solver in a loop. Machine learning promises to expedite the optimisation lifecycle by introducing models as surrogates for PDE solvers (Li et al., 2023d; Wu et al., 2024; Oldenburg et al., 2022; Sharp et al., 2022). For reliable application, a surrogate should to be: (i) ***invariant to the resolution*** of the chosen discretisation, and (ii) able to produce ***predictive uncertainty***. The latter is particularly important in engineering applications where the output of a surrogate is used to inform downstream decision-making.

Operator learning is a popular paradigm for building surrogate models that learn mappings between function spaces (Lu et al., 2019; Li et al., 2020a; Kovachki et al., 2023). Neural Operators, a type of operator learning, approximate a PDE's solution operator, enabling generalization across resolutions. However, methods like the Fourier Neural Operator (FNO) (Li et al., 2021) and Geometry-Informed Neural Operator (GINO) (Li et al., 2023d), when applied to problems with geometric boundary conditions, do not adequately leverage geometric information. This leads to overparameterised networks and inefficient uncertainty quantification, as they fail to incorporate useful geometric inductive biases.

In contrast, geometric deep learning architectures (Eliasof et al., 2021; Wu et al., 2024; Charles et al., 2017a; Bamberger et al., 2024) learn a mapping from 3D geometry to a PDE solution on the geometry's surface. These methods are parameter-efficient due to geometric inductive biases but can be overly sensitive to mesh discretization and struggle with global interactions. Recently, Sharp et al. (2022) introduced DiffusionNet (DN) to address both of these issues. DiffusionNet draws its strength for various surface learning tasks by using two simple geometric operations: (i) a diffusion layer with learnable diffusion time parameters for message-passing, and (ii) spatial gradient features according to manifold orientation for capturing anisotropy. Importantly, both these operations are agnostic to the choice of discretisation and sampling, and thus the resulting architecture is resolution invariant by construction, akin to a Neural Operator.

Contributions. In this work, we develop a probabilistic reformulation of DiffusionNet using a Fourier expansion of stochastic partial differential equations (SPDEs) (Holden et al., 1996; Walsh, 1986). This approach encodes uncertainty directly through the message-passing mechanism, avoiding the familiar limitations such as poor uncertainty quality, drop in accuracy, and lack of computational scalability of traditional Bayesian Neural Networks (Hernández-Lobato & Adams, 2015; Blundell et al., 2015) or post-hoc methods like Laplace approximation (MacKay, 1992; Ritter et al., 2018) and Conformal Predictions (Vovk et al., 2005). Instead, our reformulation allows us to build a probabilistic generative model, Probabilistic DiffusionNet (PDN), in the same spirit as a traditional variational auto-encoder that allows us to estimate PDE solutions given varying boundaries, with spatially correlated probabilistic outputs on those surfaces that reflect uncertainties. Our experiments show, firstly, that DiffusionNet is a powerful surrogate model competitive with state-of-the-art (SOTA) Neural Operators and PDE surrogates. We then show that PDN retains DiffusionNet's predictive performance while producing well-calibrated uncertainty estimates compared to standard post-hoc uncertainty quantification methods.

1.1 RELATED WORK

In this section we review the classes of models that have dominated the space of PDE surrogates, and approaches to endow them with uncertainty quantification. We provide an extended discussion on related works in Appendix A and F.

Geometric Deep Learning Graph Neural Graph Neural Networks (GNNs) (Scarselli et al., 2009; Sanchez-Gonzalez et al., 2020; Niepert et al., 2016; Li & Farimani, 2022) use message-passing to predict fields on mesh-structured data (Dalton et al., 2022; Pfaff et al., 2020; Horie & Mitsume, 2022; Brandstetter et al., 2022; Gladstone et al., 2024). Recently, Sharp et al. (2022) introduced DiffusionNet, which employs diffusion and gradient mechanisms for message-passing, building on the idea of GNNs as solutions to diffusion equations (Chamberlain et al., 2021).

For uncertainty quantification (UQ) in GNNs, methods include frequentist approaches like temperature-scaling (Guo et al., 2017; Wang et al., 2021), conformal predictions (Vovk et al., 2005; Huang et al., 2023a), ensembling (von Pichowski et al., 2024; Lin et al., 2022; Xu et al., 2022), as well as Bayesian methods (Lamb & Paige, 2020; Hasanzadeh et al., 2020). More recently, Lin et al. (2024b) and Bergna et al. (2024) extended GNNs by modeling stochastic diffusion/ODE processes to capture uncertainty. However, these methods require complex backpropagation through numerical solvers, making them computationally unsuitable for large-scale applications like CFD.

Neural Operators (NOs) (Kovachki et al., 2023; Lu et al., 2019; Pepe et al., 2025) are a popular method for learning function space maps. A key class of NOs, like FNO, capture long-range correlations using basis decompositions, with variants exploring alternative bases (Tripura & Chakraborty, 2023; Gupta et al., 2021; Bonev et al., 2023; Cao et al., 2024; Chen et al., 2024). GINO combines the graph-based GNO (Li et al., 2020a;b) for irregular domains with the FNO for regular domains. Other work has focused on alternative message-passing schemes (Li et al., 2023a; Alkin et al., 2024; Kissas et al., 2022).

For uncertainty quantification (UQ) in NOs, existing methods often require expensive post-training steps, such as when using the Laplace approximation (Magnani et al., 2022), generative models with Monte Carlo methods (Meng et al., 2022), or conformal prediction extensions (Ma et al., 2024b) Taking a different approach, Salvi et al. (2022) propose a Neural SPDE method similar to the ideas in Lin et al. (2024a) and Bergna et al. (2024), but lacks a direct UQ objective and requires slow numerical solvers. Our work combines the SPDE formalism with variational inference and the geometric inductive bias of DiffusionNet to create a scalable architecture for natural uncertainty quantification.

Point Cloud Methods Given the ubiquity of point-cloud representations in engineering and graphics applications, several methods (Charles et al., 2017a;b) have emerged that do not rely on connectivity or grid structures. Recent approaches (Guo et al., 2021; Xiao et al., 2024) adapt transformers to point-clouds, offering strong but computationally intensive PDE surrogates. To address the quadratic complexity of attention mechanisms, Cao (2021) propose the Galerkin Transformer, while Wu et al. (2024) introduce a Physics Attention layer. Both approaches scale linearly with the number of input points. While native uncertainty quantification remains unresolved, classical techniques like MC dropout and Laplace approximation are applicable.

2 BACKGROUND

2.1 PROBLEM SETTING

Our work addresses problems, such as design optimisation, where multiple geometric shapes are evaluated by solving systems of partial differential equations (PDEs). Formally, we consider a space of shapes $\mathcal{S}$, where each shape corresponds to a surface M_s embedded in $\mathbb{R}^3$. For each shape, we consider a PDE system:

$$\begin{cases} \mathcal{L}(u_s) = f & \text{in } \Omega_s \\ \mathcal{B}(u_s) = b & \text{on } \partial\Omega_s \end{cases} \tag{1}$$

where $u_s : \overline{\Omega}_s \to \mathbb{R}^{d_u}$ is the solution field and Ω_s is the domain surrounding the surface M_s. $\mathcal{L}$ and $\mathcal{B}$ are differential and boundary operators. We assume this PDE is well-posed, meaning a unique solution u_s exists for any valid shape s and boundary conditions b and forcing function f. In many applications—aerodynamics, heat transfer, structural mechanics—the quantities of interest are surface fields obtained via the trace operator:

$$u_s^\dagger = T_s(u_s) : M_s \to \mathbb{R}^{d_u}$$

We want to learn the solution map $F^\dagger$ which maps each pair (s, b_s) to a function $u_s^\dagger : M_s \to \mathbb{R}^{d_u}$ that solves the trace of the PDE system on that particular surface geometry, where $b_s : M_s \to \mathbb{R}^{d_b}$ represents boundary data or physical parameters on the surface.

A classic example of this problem is the study of fluid flow around a moving object. The governing partial differential equations (PDEs) are the Navier-Stokes equations, with the object's shape, s, representing the geometric domain. Of particular interest are surface quantities like pressure and wall-shear stress, which are integrated over the object's surface to calculate the aerodynamic drag. By developing an efficient mapping from the object's shape and other geometric parameters to these surface fields, it becomes possible to directly minimise drag. This optimisation problem has been extensively studied, for example by Wei et al. (2023) and Abbas et al. (2023).

The challenge in this setting, unlike traditional supervised learning on Euclidean spaces, is that the domain of $u_s^\dagger$ changes with each shape M_s. Because shapes in the dataset can vary in both geometry and topology, traditional neural networks—which require fixed input/output dimensions and cannot account for the geometric structure of each shape—are not directly applicable.

Supervised learning problem. In practice, each surface M_s is discretised as a triangular mesh $M_s^h = (\mathcal{V}_s, \mathfrak{F}_s, X_s)$ with $n_s = |\mathcal{V}_s|$ vertices having positions $X_s \in \mathbb{R}^{n_s \times 3}$ and face connectivity $\mathfrak{F}_s$ defining the triangulation. Functions on the surface become vectors at vertices, with inputs $\mathrm{b_s} \in \mathbb{R}^{\mathrm{n_s} \times \mathrm{d_b}}$ representing boundary data and outputs $\mathrm{u_s} \in \mathbb{R}^{\mathrm{n_s} \times \mathrm{d_u}}$ representing the solution field. While the continuous operator $F^\dagger$ maps between infinite-dimensional spaces, we learn from a discrete ground truth map $F^h : (M_s^h, \mathrm{b_s}) \mapsto \mathrm{u_s}$ obtained from numerical simulations. Henceforth, we will use the symbol $\mathrm{y_s} := \mathrm{u_s}$ to define the solution field to make it clear that this quantity is the target data. Given dataset $\mathcal{D} = \{(M_{s_i}^h, \mathrm{b_{s_i}}, \mathrm{y_{s_i}})\}_{\mathrm{i=1}}^{\mathrm{N}}$, we seek a neural network that processes functions on arbitrary meshes while respecting geometric structure and handling varying mesh sizes and topologies.

2.2 DIFFUSIONNET

DiffusionNet, as proposed by Sharp & Crane (2020), employs heat diffusion as a learnable convolution for surface analysis. It discretises the heat operator's action, $e^{\Delta_M t}$: $v(t) = e^{\Delta_M t} v(0)$ on a continuous signal $v(0)$ on an embedded surface, M. Here Δ_M represents the Laplace-Beltrami operator (LBO). The diffusion time, t, is a critical parameter that controls the scale of the convolution: a small t captures local feature dependencies, while a large t facilitates global message-passing.

On a mesh M_s^h, the LBO is approximated by the *lumped cotangent Laplacian* $\mathrm{L_s}$. Its first K eigenvectors and eigenvalues are denoted by $\Phi_s \in \mathbb{R}^{n_s \times K}$, $\Lambda_s = \mathrm{diag}(\lambda_1, \ldots, \lambda_K)$, respectively.

The *mass matrix* $\mathrm{A_s}$, derived from $\mathrm{L_s}$ is used to approximate the continuous $L^2(M_s)$ inner product. We refer to Sharp & Crane (2020) for details on how to compute these quantities.

For features $v \in \mathbb{R}^{n_s \times d_c}$ with channel dimension d_c, and learned diffusion times $t = (t_1, \ldots, t_{d_c})$, the diffusion operator is built by applying the heat operator, described above in continuum, channelwise:

$$\mathcal{P}_t(v) = \left[\Phi_s \exp(-\Lambda_s t_1)\Phi_s^T \mathrm{A_s v_1} \quad \cdots \quad \Phi_s \exp(-\Lambda_s t_{d_c})\Phi_s^T \mathrm{A_s v_{d_c}}\right] \tag{2}$$

where v_j is the j-the column of v and $\exp(-\Lambda_s t_j) = \mathrm{diag}(e^{-\lambda_1 t_j}, \ldots, e^{-\lambda_K t_j})$. This creates a learned spectral filter bank where each channel captures information at a different geometric scale, analogous to multi-scale convolutions in traditional CNNs but adapted to the non-Euclidean setting.

While the diffusion operator efficiently propagates feature information across the mesh, its inherently isotropic nature limits its effectiveness. To address this, DiffusionNet applies a learnable gradient operator $\Gamma_{\theta_g}(\cdot)$ to capture directional information from $\mathcal{P}_t(v)$. Since this operator is not a primary focus of our work, we refer the reader to Sharp et al. (2022) for a detailed discussion of how the gradients features are extracted. Finally, a pointwise MLP, $\mathcal{F}_w$, is applied to the concatenated features $[v \mid \mathcal{P}_t(v) \mid \Gamma_{\theta_g}(\mathcal{P}_t(v))]$. The combination of these three elements results in a single DiffusionNet block $\mathcal{B}_\theta$ with parameters $\theta = (t, \theta_g, w)$ given by

$$\mathcal{B}_\theta(v; M_s^h) := v + \mathcal{F}_w\left([v \mid \mathcal{P}_t(v) \mid \Gamma_{\theta_g}(\mathcal{P}_t(v))]\right) \tag{3}$$

The inclusion of M_s^h as a second input signifies that the block's operation is dependent on the geometric quantities required by the diffusion and gradient operators such as $\Phi_s, \Lambda_s, \mathrm{A_s}$.

The full DiffusionNet architecture is a composition of L such blocks, along with pointwise lifting and projection MLPs $\mathcal{L}_{w_{\text{lift}}} : \mathbb{R}^{n_s \times d_{\text{in}}} \to \mathbb{R}^{n_s \times d_c}$ and $\mathcal{Q}_{w_{\text{proj}}} : \mathbb{R}^{n_s \times d_c} \to \mathbb{R}^{n_s \times d_{\text{out}}}$. The complete map is defined as

$$D_\theta(v; M_s^h) := \mathcal{Q}_{w_{\text{proj}}} \circ \mathcal{B}_{\theta_L} \circ \cdots \circ \mathcal{B}_{\theta_1} \circ \mathcal{L}_{w_{\text{lift}}}(v). \tag{4}$$

DiffusionNet's effectiveness stems from its intrinsic operations, which adapt to the unique geometry of each surface. The spectral basis allows for multi-resolution processing while maintaining resolution invariance, i.e. models trained on one mesh resolution generalize to others.

To adapt DiffusionNet for physics applications, we augment the input field with various geometric and physical features such as: boundary conditions for the PDE (1), vertex coordinates and normals, heat kernel signatures (Sharp et al., 2022), and any relevant physical parameters. The specific features used in our experiments are detailed in Section 4.

Connection with Neural Operators Since the key components of DiffusionNet–diffusion and gradients–can be formulated in continuum, the architecture can learn PDE operators on complex geometries while maintaining geometric consistency and resolution invariance. This capability is similar to that of a Neural Operator. We discuss this connection in detail in Appendix A.

3 METHODOLOGY

Learning from finite training data inherently introduces uncertainty—we cannot perfectly recover F^h across all possible geometries, and our confidence should vary on unseen data. We therefore extend DiffusionNet to learn a distribution over possible mappings rather than a single approximation, enabling uncertainty quantification for predictions on novel geometries.

3.1 PROBABILISTIC DIFFUSIONNET

Our goal is to create a probabilistic model for surface fields on a mesh by leveraging the inductive biases of a geometric model architecture. To achieve this, we reformulate DiffusionNet's core mechanism—the deterministic diffusion process—into a stochastic one. This approach injects a noising mechanism directly into the underlying convolution operation, leading to a set of probabilistic operations that are both intuitive and interpretable. To define these operations, we first obtain a spectral representation of the solution to the *stochastic heat equation* in Theorem 3.1. We then use this representation to build a stochastic diffusion operator for carrying out a stochastic convolution (message-passing) layer that becomes the building block of a probabilistic model for surface fields on a mesh.

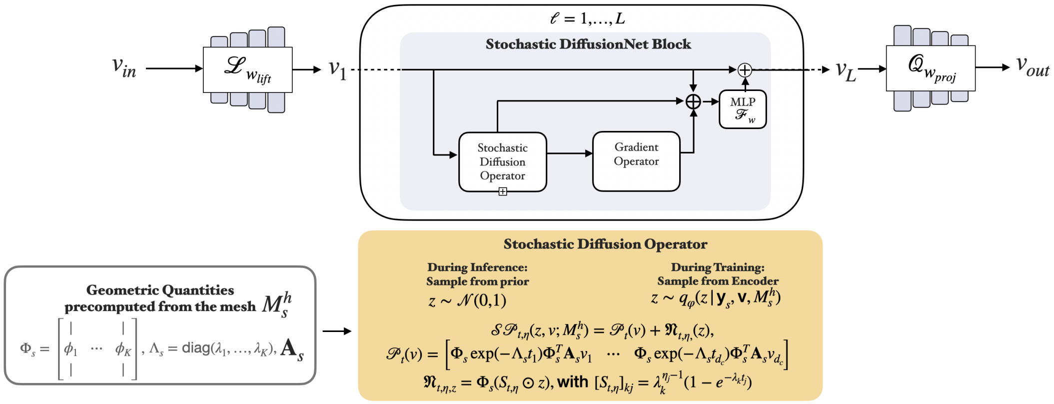


Figure 1: Architecture of the Stochastic DiffusionNet Block. Note that except the Stochastic Diffusion Operator, all other operations remain the same as DiffusionNet (Sharp et al., 2022).

A stochastic process, on an embedded surface M with LBO Δ_M, can be obtained as a randomly perturbed heat equation through the addition of spatial (coloured) noise:

$$\partial_t v(t) = \Delta_M v(t) + \mathcal{W}^Q, \quad v(0) = v_0 \tag{5}$$

where the spatial noise $\mathcal{W}^Q$ is an $L^2(M)$-valued Gaussian random field such that its covariance operator Q has LBO eigenvectors $\{\phi_k\}_{k=0}^{\infty}$ and eigenvalues $\{q_k\}_{k=0}^{\infty}$ (see, Prévôt & Röckner (2007)[Proposition 2.1.6] and Lototsky & Rozovskii (2009)[Thm. 3.2.15]). The following theorem characterizes the solution of (5). A more complete statement and proof is provided in Appendix B, where we also provide an upper bound on the expected squared norm of $v(t)$ under suitable initial conditions and choice of $\{\lambda_k, z_k\}_{k=0}^{\infty}$.

Theorem 3.1. *There exists a unique strong solution of* (5) *with the spectral representation:*

$$v(t) = \sum_{k=0}^{\infty} \left(e^{-\lambda_k t} \langle v_0, \phi_k \rangle_{L^2(M)} + \frac{\sqrt{q_k}}{\lambda_k} (1 - e^{-\lambda_k t}) z_k \right) \phi_k \,, \tag{6}$$

where $\{z_k\}_{k=0}^{\infty} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$*. Moreover, conditional on* v_0*,* $v(t)$ *is an* $L^2(M)$*-valued Gaussian random field.*

Building on this foundation, we now construct our stochastic variant of the diffusion operator $\mathcal{P}_t$. To do this, we first parameterise the noise covariance via $q_k = \lambda_k^{2\eta}$ where η controls the noise spectrum (see section G.5 for further details). This parameterisation avoids basis ambiguities as discussed in Lim et al. (2022); Zhang et al. (2024) (see Section A.1). For diffusion times $t = (t_1, \ldots, t_{d_c})$, noise parameters $\eta = (\eta_1, \ldots, \eta_{d_c})$ and random matrix $z \in \mathbb{R}^{K \times d_c}$ with $z_{kj} \sim \mathcal{N}(0,1)$, we can extend the diffusion operator $\mathcal{P}_t$ to a corresponding stochastic one by discretising Eq. (18) on the mesh M_s^h:

$$\mathcal{SP}_{t,\eta}(z, v; M_s^h) = \mathcal{P}_t(v) + \mathfrak{N}_{t,\eta,}(z), \tag{7}$$

where the noise perturbation term $\mathfrak{N}_{t,\eta,z} = \Phi_s(S_{t,\eta} \odot z)$, with $[S_{t,\eta}]_{kj} = \lambda_k^{\eta_j - 1}(1 - e^{-\lambda_k t_j})$, and $\odot$ denotes element-wise multiplication. In practice, we repeat the random variable across channels, which reduces z to a K-dimensional random vector. A Stochastic DiffusionNet block (see Figure 1) $\mathcal{SB}_\theta$ with parameters $\theta = (t, \eta, \theta_g, w)$ is then given by:

$$\mathcal{SB}_\theta(z, v; M_s^h) := v + \mathcal{F}_w\left([v \mid \mathcal{SP}_{t,\eta}(z, v) \mid \Gamma_{\theta_g}(\mathcal{SP}_{t,\eta}(z, v))]\right). \tag{8}$$

Probabilistic DiffusionNet (PDN), defines a hierarchical generative model encapsulated by the following generative process:

$$\begin{aligned} z_1, \ldots, z_L &\sim \mathcal{N}(0, I_K) \\ SD_\theta(z_{1:L}, v; M_s^h) &:= \mathcal{Q}_{w_{\text{proj}}} \circ \mathcal{SB}_{\theta_L} \circ \cdots \circ \mathcal{SB}_{\theta_1} \circ \mathcal{L}_{w_{\text{lift}}}(z_{1:L}, v) \end{aligned} \tag{9}$$

where $z_{1:L} = \{z_1, \ldots, z_L\}$ and $\theta = \{w_{\text{lift}}, \theta_1, \ldots, \theta_L, w_{\text{proj}}\}$.

The generative process above lets us define a marginal distribution:

$$p_{\boldsymbol{\theta}}(\mathrm{u}|\mathrm{v}_{\text{input}}, \mathrm{M}_{\mathrm{s}}^{\mathrm{h}}) = \int \delta(\mathrm{u} - \mathrm{SD}_{\theta}(\mathrm{z}_{1:\mathrm{L}}, \mathrm{v}; \mathrm{M}_{\mathrm{s}}^{\mathrm{h}})) \prod_{\ell=1}^{\mathrm{L}} \mathrm{p}(\mathrm{z}_{\ell}) \mathrm{d}\mathrm{z}_{1:\mathrm{L}} \tag{10}$$

of a surface field u on a mesh M_s^h, where $\delta(\cdot)$ denotes a Dirac delta distribution. The hierarchical structure induces uncertainty at multiple geometric scales, with each layer ℓ adding stochastic perturbations at scale t_ℓ and magnitude η_ℓ.

3.2 VARIATIONAL INFERENCE

Having defined a probabilistic generative model for fields on a mesh surface, we can use it to model the PDE solution fields obtained from simulation. Thus, we proceed to learn the parameters θ given the training dataset $\mathcal{D}$.

Observational model. We consider an observation model for the simulated solution field u_{s} given by the following measurement equation:

$$\mathrm{y}_{\mathrm{s}} = \mathrm{u} + \boldsymbol{\epsilon}, \tag{11}$$

where $\mathrm{u} \sim \mathrm{p}_{\theta}(\mathrm{u}|\mathrm{v}_{\text{input}}, \mathrm{M}_{\mathrm{s}}^{\mathrm{h}})$ (10) is the probabilistic surrogate model, and $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2 \mathbb{I}_{n_{s_i}})$ is a measurement noise. This measurement noise is added to capture some form of model discrepancy and to stabilise the training dynamics.

ELBO. Our probabilistic model is a parametrised latent-variable model, where z represents the random latent variable. Our goal is to maximize the marginal log-likelihood. After collecting the solution fields, inputs, latent variables and meshes corresponding to all shapes in $\mathrm{Y} = \{\mathrm{y}_{\mathrm{s}_{\mathrm{i}}}\}_{\mathrm{i}=1}^{\mathrm{N}}$, $\mathrm{V} = \{\mathrm{v}_{\text{input}_{\mathrm{i}}}\}_{\mathrm{i}=1}^{\mathrm{N}}$, $\mathcal{M} = \{M_{s_i}^h\}_{i=1}^N$, $Z = \{z^i\}_{i=1}^N$, we can write the marginal log-likelihood as:

$$\ln p_{\theta,\sigma}(\mathrm{Y}|\mathrm{V}, \mathcal{M}) = \ln \int p_{\theta,\sigma}(\mathrm{Y}|\mathrm{V}, \mathcal{M}, \mathrm{Z}) \mathrm{p}(\mathrm{Z}) \mathrm{d}\mathrm{Z}\,, \tag{12}$$

$$p_{\theta,\sigma}(\mathrm{Y}|\mathrm{V}, \mathcal{M}, \mathrm{Z}) = \prod_{i=1}^{N} \mathcal{N}(\mathrm{y}_{\mathrm{s}_{\mathrm{i}}}; \mathrm{SD}_{\boldsymbol{\theta}}(\mathrm{z}_{1:\mathrm{L}}^{\mathrm{i}}, \mathrm{v}_{\mathrm{i}}), \sigma^2)\,. \tag{13}$$

Since the integral cannot be computed analytically due to the non-linear relationship between y_{s} and z, we employ *variational inference* (Murphy, 2022). We introduce a family of distribution $q_{\varphi}(Z|\mathrm{Y}, \mathrm{V}, \mathcal{M})$, parametrised by φ, to approximate the posterior $p(Z|\mathcal{D})$. We optimize $(\theta, \sigma, \varphi)$ by maximizing the evidence lower bound (ELBO):

$$\mathcal{L}(\theta, \varphi) = \mathbb{E}_{q_{\varphi}}\left[p_{\theta,\sigma}(\mathrm{Y}|\mathrm{V}, \mathcal{M}, \mathrm{Z})\right] - \mathcal{D}_{KL}(q_{\varphi}(Z|\mathrm{Y}, \mathrm{V}, \mathcal{M}) \| \mathrm{p}(\mathrm{Z})) \leq \ln \mathrm{p}_{\theta,\sigma}(\mathrm{Y}|\mathrm{V}, \mathcal{M})\,. \tag{14}$$

Amortized encoder. We adopt an amortized approach for the variational approximation given by:

$$q_{\varphi}(Z|\mathrm{Y}, \mathrm{V}, \mathcal{M}) = \prod_{\mathrm{i}=1}^{\mathrm{N}} \prod_{\mathrm{l}=1}^{\mathrm{L}} \mathcal{N}(\mathrm{z}_{\ell}^{\mathrm{i}}; \mu_{\ell}^{\varphi}(\mathrm{y}_{\mathrm{s}_{\mathrm{i}}}, \mathrm{v}_{\text{input}_{\mathrm{i}}}, \mathrm{M}_{\mathrm{s}_{\mathrm{i}}}^{\mathrm{h}}), \Sigma_{\mathrm{l}}^{\varphi}(\mathrm{y}_{\mathrm{s}_{\mathrm{i}}}, \mathrm{v}_{\text{input}_{\mathrm{i}}}, \mathrm{M}_{\mathrm{s}_{\mathrm{i}}}^{\mathrm{h}})) \tag{15}$$

where μ_{ℓ}^{φ} and Σ_{ℓ}^{φ} are simple variants of DiffusionNet with an aggregation layer (see Section C for details). We use the *reparameterisation-trick* to obtain a Monte Carlo estimate of the gradient of the ELBO, to carry out its maximisation using a stochastic gradient descent algorithm.

Uncertainty quantification Once we have learned the optimal parameters $(\theta_\star, \sigma_\star, \varphi_\star)$ by maximizing (14), we may sample from the observational predictive distribution, for unseen shape s_*:

$$p_{\theta,\sigma}(\mathrm{y}_{\mathrm{s}_*}|\mathrm{v}_{\text{input}_*}, \mathrm{M}_{\mathrm{s}_*}) = \int \mathrm{p}_{\theta,\sigma}(\mathrm{y}_{\mathrm{s}_*}|\mathrm{v}_{\text{input}_*}, \mathrm{M}_{\mathrm{s}_*}^{\mathrm{h}}, \mathrm{z}^*) \mathrm{p}(\mathrm{z}^*) \mathrm{d}\mathrm{z}^*\,. \tag{16}$$

The variance of the predictive distribution (16) reflects the model's noise sensitivity, with shape-dependent modulation through $q_k(s) = \lambda_k(s)^{2\eta}$ ensuring higher variance for unusual geometric spectra. While this correlates with epistemic uncertainty patterns (higher on novel shapes), it is technically learned *heteroscedastic aleatoric uncertainty* since parameters are fixed post-training. True epistemic uncertainty would require treating θ, σ as random variables. Nevertheless, PDN's geometrically-structured predictive variance provides valuable confidence estimates, effectively warning when predictions extrapolate beyond the training distribution.

4 EXPERIMENTAL RESULTS

We benchmark our approach in two phases: first evaluating the accuracy of Probabilistic DiffusionNet (PDN) against state-of-the-art (SOTA) geometric deep learning models for PDE surrogacy, then comparing PDN against baseline UQ methods to assess predictive accuracy and uncertainty quality.

Datasets. We evaluate on two standard computational fluid dynamics datasets, both containing Reynolds-Averaged Navier-Stokes simulations on varying geometries: **ShapeNet car** (Umetani & Bickel, 2018), consisting of car meshes with 3,856 vertices, and **Ahmed bodies** (Li et al., 2023d), consisting of simplified automotive test shapes with meshes having $\sim 10^5$ vertices per shape. Following Li et al. (2023d), we apply the same preprocessing and test/train splits, learning a mapping from shape geometry to surface pressure distribution.

Baselines and Metrics. In Section 4.1, we compare PDN's accuracy against three categories of surrogate models: i) Neural operators that handle mesh data, including GNO, GINO, and Fengbo (Pepe et al., 2025); ii) Transformer-based approaches like Transolver (Wu et al., 2024) and Point Cloud Transformer (PCT); and iii) Graph-based methods, including GraphSage (Hamilton et al., 2017), MeshGraphNet (Pfaff et al., 2020), and the original DiffusionNet (DN). We evaluate predictive accuracy using the relative L^2 loss. We left out benchmarking some of the graph and transformer-based models for Ahmed bodies, due to the large mesh size. We report PDN's performance for an extended set of ShapeNet car dataset in Section G.1. In Section 4.2, we evaluate both accuracy using RMSE, due to our probabilistic output, and uncertainty quality using total miscalibrated area (MCAL), negative predictive log-likelihood (NLL), and interval score (IS). All metrics are computed using the uncertainty toolbox package Chung et al. (2021). Note that there is no single metric that captures every aspect of probabilistic predictions, so we resort to the collection of metrics as these are the most commonly used and there is a rich literature to support them. We discuss further details of the metrics and their choices in Section E. We compare PDN against GINO and Transolver, each enhanced with standard UQ methods: Monte Carlo Dropout (DO), Laplace Approximation (LA), and Model Ensembling (ME), with 10 models. As the code for the conformal prediction method proposed in Ma et al. (2024b) has not been released, we do not provide a comparison to it. In Section F we discuss the motivation behind choosing these UQ methods and their implementation. Finally in Section 4.3 we evaluate PDN's prediction quality on *out-of-distribution* (OOD) shapes by creating a different split of the **ShapeNet car** dataset based on geometric similarity.

Implementation. For all experiments we used 16 blocks and chose $K = 128$ modes, based on ablations (see Section G.2). To obtain the cotan-Laplacian, we used the robust Laplacian method proposed in Sharp & Crane (2020) and used the approach in Sharp et al. (2022) to approximate the gradient features. Note that these were obtained beforehand and cached. For the input field $\mathrm{v}_{\text{input}}$, we simply used the vertex positions X_s. In the Ahmed body dataset, the inlet velocity ρ changes between samples, so we repeated this scalar value across vertices to form the input $\mathrm{v}_{\text{input}} = [\mathrm{X}_\mathrm{s} | \rho 1_{\mathrm{n}_\mathrm{s}}]$, where $1_{\mathrm{n}_\mathrm{s}} \in \mathbb{R}^{\mathrm{n}_\mathrm{s}}$ is a vector of ones. We discuss further details of implementation including optimisation in Section D. We used `PyTorch` (Paszke et al., 2017) for implementation. Experiments ran on a single NVIDIA A100 GPU. The code is available at `retracted`.

4.1 PREDICTIVE ACCURACY

We compare the predictive performance of PDN against baseline methods across both datasets (Table 1). Our results demonstrate that both DiffusionNet (DN) and PDN achieve SOTA performance on the ShapeNet car dataset and outperform all other implementations on the larger Ahmed bodies dataset. These results validate that shape-dependent operators provide an effective geometric inductive bias for PDE surrogate modelling. Notably, PDN's performance closely matches that of the original DN implementation while adding UQ capabilities, and incurring a lower computational cost with a smaller parameter count, (see Table 2 and Section G.3). The minimal impact on predictive accuracy

Table 1: Performance of various methods for predicting the pressure field on a surface, evaluated using the relative L^2 (RL2) error. The values, shown in units of 10^{-2}, represent the average RL2 error across all samples in the test dataset, with lower values indicating higher accuracy. For PDN we used the mean of 100 samples from the predictive distribution (16).

The ShapeNet car dataset		**The Ahmed bodies dataset**	
Methods	↓ RL2	Methods	↓ RL2
PointNet (Charles et al., 2017a)	11	MeshGraphNet (Pfaff et al., 2020)	13.9
GraphSage (Hamilton et al., 2017)	10.5	UNet (with interpolation)	11.2
GNO (Li et al., 2020a)	18.8	FNO (with interpolation)	12.6
GeoFNO (Li et al., 2023b)	15.9	GINO (Li et al., 2023d)	8.31
GINO (Li et al., 2023d)	7.1	Fengbo (Pepe et al., 2025)	10.7
Fengbo (Pepe et al., 2025)	8.9	Transolver (Wu et al., 2024)	7.1
PCT (Guo et al., 2021)	7.7	DN	**5.4**
Transolver (Wu et al., 2024)	**6.2**	PDN	6.2
DN	6.3		
PDN	6.3		

when moving from DN to PDN, despite incorporating probabilistic elements, highlights the advantage of capturing uncertainty through the message-passing operation.

4.2 UNCERTAINTY QUANTIFICATION

In Table 2 we compare the quality of UQ. We evaluate models that create probabilistic predictions of scalar pressure fields for new shapes. Each model learns a distribution over possible pressure fields, allowing us to quantify its uncertainty by sampling candidate predictions. These samples are then used to calculate uncertainty quantification metrics, assessing the model's concentration, calibration, and coverage.

We furnished the average values of metrics across test samples. Moreover, excluding ME, we carried out additional repeats of the experiments and averaged the metrics across these. Since each metric captures only a certain aspect of the UQ, it is important that a method performs in balanced way across all these metrics. To highlight this aspect we ranked the performance of each of the competing method on individual metrics for each dataset. We then averaged these ranks across all the metrics and datasets. We noticed, based on the average rank, that TS-ME followed by PDN had the best average rank, indicating their dominance in terms of producing a balanced performance. We also noticed that for some methods, LA, DO, the performance can vary drastically across datasets and models. Notice how the PDN metrics are always well balanced across the datasets. This shows that PDN can reliably and robustly produce predictive uncertainties in a problem-agnostic manner. Fig. 2

Table 2: UQ metrics for surface pressure field prediction . Each metric is first averaged across the test set shapes, and then averaged across three repeats of the experiments. We also report the average rank (Rank) of each method across all metrics (a lowest value indicates most balanced performance considering all metrics)

		ShapeNet car				**Ahmed bodies**			
Methods	Rank↓	RMSE↓	NLL↓	MCAL↓	IS↓	RMSE↓	NLL↓	MCAL↓	IS↓
GINO-DO	6.50	6.59	2.44	**0.07**	10.01	32.74	5.28	0.24	121.83
GINO-LA	8.25	4.27	29.30	0.23	14.81	34.58	5.27	0.24	110.31
GINO-ME	5.38	3.87	**2.21**	0.13	6.35	28.82	4.65	0.29	126.69
TS-DO	5.06	**3.86**	2.47	0.11	9.57	21.17	4.62	0.30	119.55
TS-LA	6.13	4.02	5.84	0.09	11.19	25.01	5.29	0.26	100.66
TS-ME	**2.81**	4.06	2.31	0.11	**5.66**	24.95	3.26	**0.08**	**37.15**
DN-DO	8.25	4.61	2.76	0.21	20.55	26.30	4.92	0.30	190.18
DN-LA	5.25	4.18	7.52	0.09	12.32	20.79	4.72	0.23	67.01
DN-ME	4.13	5.11	2.36	0.1	8.8	27.35	**3.04**	0.1	50.50
PDN (Ours)	3.25	3.91	2.77	0.08	7.99	**18.28**	4.23	0.23	67.33

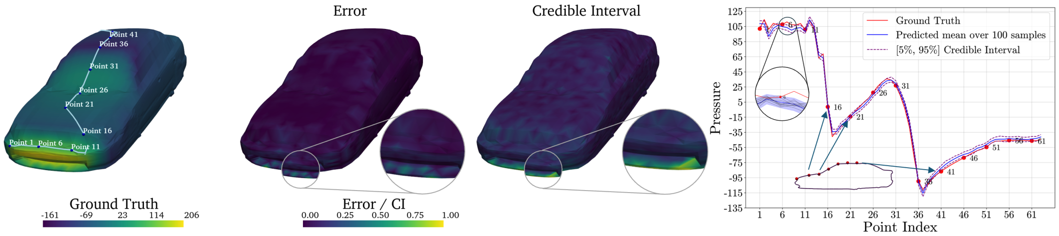


Figure 2: **Visualization of uncertainty estimates produced by PDN for a Shape-Net car**. a) Ground-truth pressure field and one sampled geodesic path of vertices from the front to the back of the car. b) Normalized absolute error between the predicted mean and the ground-truth field. c) Normalized Credible Range, i.e., difference between the 5th and 95th percentiles, of 100 samples from the predictive distribution, which correlates well, see the inset region, with the absolute error. d) Comparison of the predicted samples and ground-truth along the geodesic path points shown in a).

Table 3: UQ metrics for surface pressure field prediction (test set averages) on OOD shapes.

	ShapeNet car OOD			
Methods	RMSE↓	NLL↓	MCAL↓	IS↓
GINO-DO	7.65	5.67	**0.06**	<u>13.85</u>
TS-DO	5.99	<u>3.45</u>	0.30	42.74
DN-DO	5.68	**2.77**	0.12	16.57
PDN (Ours)	**5.54**	3.50	<u>0.09</u>	**10.22**

shows how PDN's uncertainty estimate correlates well with the prediction error for a ShapeNet car sample. Whilst model ensembling yields the strongest UQ performance, when combined with TS, it comes with the significant cost of training multiple (in our case 10) models on different subsets of the data, and so is infeasible for industrial scale applications. By contrast, PDN yields good UQ with no additional cost.

4.3 OUT-OF-DISTRIBUTION PREDICTION

For this experiment, we clustered the ShapeNet car dataset using spectral clustering (Shi & Malik, 2000), where we used a pairwise chamfer distance to measure similarity. This lead to discovering two clusters having 471 (used as training set) and 140 (used as test set) shapes respectively. We then compared the performance of PDN, GINO-DO, TS-DO and DN-DO (since these do not incur additional compute cost such as LA and ME methods) on this dataset using the same metrics used for previous UQ experiments. We furnish the UQ metrics in Table 3 and visualisations in Fig. 3. The above results reinforce PDN's strengths. All methods, except GINO-DO, show similar accuracy, with PDN having a slight edge. Crucially, PDN's coverage (IS) is significantly better, indicating it better handles distribution shifts. While GINO-DO shows good calibration, this is a result of overly wide prediction intervals, which negatively impacts its IS score.

5 CONCLUSIONS

DiffusionNet (DN) is an efficient graph neural network that uses a diffusion mechanism for message passing. We introduced Probabilistic DiffusionNet, an extension that introduces stochasticity within this message-passing operation of DiffusionNet by modifying the diffusion mechanism, yielding spatially correlated predictive distributions. This approach delivers superior uncertainty quantification on standard benchmarks while requiring substantially fewer parameters and computational resources than other popular alternatives.

For the DN/PDN architecture, one has to eigendecompose the cotan-Laplacian; despite efficient implementations, this can become a problem for large meshes. Future work may consider approximations to the eigendecomposition for efficiency. Future work may also consider applying the model to real-world applications requiring UQ such as design optimisation and active learning.

REFERENCES

Asad Abbas, Ashkan Rafiee, and Max Haase. Deepmorpher: deep learning-based design space dimensionality reduction for shape optimisation. *Journal of Engineering Design*, 34(3):254–270, 2023.

Benedikt Alkin, Andreas Fürst, Simon Lucas Schmid, Lukas Gruber, Markus Holzleitner, and Johannes Brandstetter. Universal physics transformers: A framework for efficiently scaling neural operators. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL `https://openreview.net/forum?id=oUXiNX5KRm`.

Herbert Amann. Parabolic equations on uniformly regular riemannian manifolds and degenerate initial boundary value problems. *Recent developments of mathematical fluid mechanics*, pp. 43–77, 2016.

John M Ball. Strongly continuous semigroups, weak solutions, and the variation of constants formula. *Proceedings of the American Mathematical Society*, 63(2):370–373, 1977.

Jacob Bamberger, Federico Barbero, Xiaowen Dong, and Michael Bronstein. Bundle neural networks for message diffusion on graphs. *arXiv preprint arXiv:2405.15540*, 2024.

Ali Behzadan and Michael Holst. Sobolev-Slobodeckij spaces on compact manifolds, revisited. *Mathematics*, 10(3):522, 2022.

Richard Bergna, Sergio Calvo Ordoñez, Felix Opolka, Pietro Lio, and José Miguel Hernández-Lobato. Uncertainty modeling in graph neural networks via stochastic differential equations. In *NeurIPS 2024 Workshop on Bayesian Decision-making and Uncertainty*, 2024. URL `https://openreview.net/forum?id=uwC1RRSqjz`.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pp. 1613–1622. JMLR.org, 2015.

Boris Bonev, Thorsten Kurth, Christian Hundt, Jaideep Pathak, Maximilian Baust, Karthik Kashinath, and Anima Anandkumar. Spherical Fourier neural operators: Learning stable dynamics on the sphere. In *International conference on machine learning*, pp. 2806–2823. PMLR, 2023.

Viacheslav Borovitskiy, Alexander Terenin, Peter Mostowsky, et al. Matérn Gaussian processes on Riemannian manifolds. *Advances in Neural Information Processing Systems*, 33:12426–12437, 2020.

Johannes Brandstetter, Daniel E Worrall, and Max Welling. Message passing neural PDE solvers. In *International Conference on Learning Representations*, 2022.

Edoardo Calvello, Nikola B Kovachki, Matthew E Levine, and Andrew M Stuart. Continuum attention for neural operators. *arXiv preprint arXiv:2406.06486*, 2024.

Qianying Cao, Somdatta Goswami, and George Em Karniadakis. Laplace neural operator for solving differential equations. *Nature Machine Intelligence*, 6(6):631–640, Jun 2024. ISSN 2522-5839. doi: 10.1038/s42256-024-00844-4. URL `https://doi.org/10.1038/s42256-024-00844-4`.

Shuhao Cao. Choose a transformer: Fourier or Galerkin. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL `https://openreview.net/forum?id=ssohLcmn4-r`.

Ben Chamberlain, James Rowbottom, Maria I Gorinova, Michael Bronstein, Stefan Webb, and Emanuele Rossi. GRAND: Graph neural diffusion. In *International conference on machine learning*, pp. 1407–1418. PMLR, 2021.

R. Qi Charles, Hao Su, Mo Kaichun, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3D classification and segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 77–85, 2017a. doi: 10.1109/CVPR.2017.16.

R. Qi Charles, Hao Su, Mo Kaichun, and Leonidas J. Guibas. PointNet: Deep learning on point sets for 3d classification and segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 77–85, 2017b. doi: 10.1109/CVPR.2017.16.

Isaac Chavel. *Eigenvalues in Riemannian geometry*, volume 115. Academic press, 1984.

Gengxiang Chen, Xu Liu, Qinglu Meng, Lu Chen, Changqing Liu, and Yingguang Li. Learning neural operators on Riemannian manifolds. *National Science Open*, 3(6):20240001, 2024.

Youngseog Chung, Ian Char, Han Guo, Jeff Schneider, and Willie Neiswanger. Uncertainty toolbox: an open-source library for assessing, visualizing, and improving uncertainty quantification. *arXiv preprint arXiv:2109.10254*, 2021.

Giuseppe Da Prato and Jerzy Zabczyk. *Stochastic equations in infinite dimensions*, volume 152. Cambridge university press, 2014.

David Dalton, Hao Gao, and Dirk Husmeier. Emulation of cardiac mechanics using graph neural networks. *Computer Methods in Applied Mechanics and Engineering*, 401:115645, 2022. ISSN 0045-7825. doi: https://doi.org/10.1016/j.cma.2022.115645. URL `https://www.sciencedirect.com/science/article/pii/S0045782522006004`.

Jingyang Deng, Xingjian Li, Haoyi Xiong, Xiaoguang Hu, and Jinwen Ma. Geometry-guided conditional adaptation for surrogate models of large-scale 3D PDEs on arbitrary geometries. In Kate Larson (ed.), *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pp. 5790–5798. International Joint Conferences on Artificial Intelligence Organization, 8 2024. doi: 10.24963/ijcai.2024/640. URL `https://doi.org/10.24963/ijcai.2024/640`. Main Track.

Vijay Prakash Dwivedi, Chaitanya K Joshi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Benchmarking graph neural networks. *Journal of Machine Learning Research*, 24 (43):1–48, 2023.

Moshe Eliasof, Eldad Haber, and Eran Treister. PDE-GCN: Novel architectures for graph neural networks motivated by partial differential equations. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 3836–3849. Curran Associates, Inc., 2021.

CM Elliott, M Hairer, and MR Scott. Stochastic partial differential equations on evolving surfaces and evolving Riemannian manifolds. *arXiv preprint arXiv:1208.5958*, 2012.

Lawrence C Evans. *Partial differential equations*, volume 19. American Mathematical Society, 2022.

Edwin Fong and Chris C Holmes. Conformal bayesian computation. *Advances in Neural Information Processing Systems*, 34:18268–18279, 2021.

Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: representing model uncertainty in deep learning. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pp. 1050–1059. JMLR.org, 2016.

Hongyang Gao and Shuiwang Ji. Graph u-nets. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2083–2092. PMLR, 09–15 Jun 2019. URL `https://proceedings.mlr.press/v97/gao19a.html`.

Rini Jasmine Gladstone, Helia Rahmani, Vishvas Suryakumar, Hadi Meidani, Marta D'Elia, and Ahmad Zareei. Mesh-based GNN surrogates for time-independent PDEs. *Scientific Reports*, 14(1):3394, Feb 2024. ISSN 2045-2322. doi: 10.1038/s41598-024-53185-y. URL `https://doi.org/10.1038/s41598-024-53185-y`.

Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007. doi: 10.1198/016214506000001437. URL `https://doi.org/10.1198/016214506000001437`.

Alexander Grigor'yan. *Heat kernel and analysis on manifolds*, volume 47. American Mathematical Soc., 2009.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pp. 1321–1330. JMLR.org, 2017.

Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R. Martin, and Shi-Min Hu. Pct: Point cloud transformer. *Computational Visual Media*, 7(2):187–199, Jun 2021. ISSN 2096-0662. doi: 10.1007/s41095-021-0229-5. URL `https://doi.org/10.1007/s41095-021-0229-5`.

Gaurav Gupta, Xiongye Xiao, and Paul Bogdan. Multiwavelet-based operator learning for differential equations. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL `https://openreview.net/forum?id=LZDiWaC9CGL`.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL `https://proceedings.neurips.cc/paper_files/paper/2017/file/5dd9db5e033da9c6fb5ba83c7a7ebea9-Paper.pdf`.

Zhongkai Hao, Zhengyi Wang, Hang Su, Chengyang Ying, Yinpeng Dong, Songming Liu, Ze Cheng, Jian Song, and Jun Zhu. GNOT: a general neural operator transformer for operator learning. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Arman Hasanzadeh, Ehsan Hajiramezanali, Shahin Boluki, Mingyuan Zhou, Nick Duffield, Krishna Narayanan, and Xiaoning Qian. Bayesian graph neural networks with adaptive connection sampling. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

José Miguel Hernández-Lobato and Ryan P. Adams. Probabilistic backpropagation for scalable learning of Bayesian neural networks. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pp. 1861–1869. JMLR.org, 2015.

Helge Holden, Bernt Øksendal, Jan Ubøe, and Tusheng Zhang. Stochastic partial differential equations. In *Stochastic partial differential equations*, pp. 141–191. Springer, 1996.

Masanobu Horie and Naoto Mitsume. Physics-embedded neural networks: graph neural pde solvers with mixed boundary conditions. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Kexin Huang, Ying Jin, Emmanuel Candes, and Jure Leskovec. Uncertainty quantification over graph with conformalized graph neural networks. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023a. URL `https://openreview.net/forum?id=ygjQCOyNfh`.

Yinan Huang, William Lu, Joshua Robinson, Yu Yang, Muhan Zhang, Stefanie Jegelka, and Pan Li. On the stability of expressive positional encodings for graphs. *arXiv preprint arXiv:2310.02579*, 2023b.

Michael Hutchinson, Alexander Terenin, Viacheslav Borovitskiy, So Takao, Yee Teh, and Marc Deisenroth. Vector-valued gaussian processes on Riemannian manifolds via gauge independent projected kernels. *Advances in Neural Information Processing Systems*, 34:17160–17169, 2021.

Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 4651–4664. PMLR, 18–24 Jul 2021. URL `https://proceedings.mlr.press/v139/jaegle21a.html`.

Georgios Kissas, Jacob H. Seidman, Leonardo Ferreira Guilhoto, Victor M. Preciado, George J. Pappas, and Paris Perdikaris. Learning operators with coupled attention. *J. Mach. Learn. Res.*, 23 (1), January 2022. ISSN 1532-4435.

Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces with applications to PDEs. *Journal of Machine Learning Research*, 24(89):1–97, 2023.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

George Lamb and Brooks Paige. Bayesian graph neural networks for molecular property prediction, 2020. URL `https://arxiv.org/abs/2012.02089`.

Shibo Li, Xin Yu, Wei Xing, Robert Kirby, Akil Narayan, and Shandian Zhe. Multi-resolution active learning of Fourier neural operators. In *International Conference on Artificial Intelligence and Statistics*, pp. 2440–2448. PMLR, 2024.

Zijie Li and Amir Barati Farimani. Graph neural network-accelerated lagrangian fluid simulation. *Comput. Graph.*, 103(C):201–211, April 2022. ISSN 0097-8493. doi: 10.1016/j.cag.2022.02.004. URL `https://doi.org/10.1016/j.cag.2022.02.004`.

Zijie Li, Kazem Meidani, and Amir Barati Farimani. Transformer for partial differential equations operator learning. *Transactions on Machine Learning Research*, 2023a. ISSN 2835-8856. URL `https://openreview.net/forum?id=EPPqt3uERT`.

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

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Andrew Stuart, Kaushik Bhattacharya, and Anima Anandkumar. Multipole graph neural operator for parametric partial differential equations. *Advances in Neural Information Processing Systems*, 33:6755–6766, 2020b.

Zongyi Li, Nikola Borislavov Kovachki, Kamyar Azizzadenesheli, Burigede liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021.

Zongyi Li, Daniel Zhengyu Huang, Burigede Liu, and Anima Anandkumar. Fourier neural operator with learned deformations for PDEs on general geometries. *Journal of Machine Learning Research*, 24(388):1–26, 2023b. URL `http://jmlr.org/papers/v24/23-0064.html`.

Zongyi Li, Daniel Zhengyu Huang, Burigede Liu, and Anima Anandkumar. Fourier neural operator with learned deformations for pdes on general geometries. *Journal of Machine Learning Research*, 24(388):1–26, 2023c.

Zongyi Li, Nikola Kovachki, Chris Choy, Boyi Li, Jean Kossaifi, Shourya Otta, Mohammad Amin Nabian, Maximilian Stadler, Christian Hundt, Kamyar Azizzadenesheli, et al. Geometry-informed neural operator for large-scale 3d PDEs. *Advances in Neural Information Processing Systems*, 36: 35836–35854, 2023d.

Derek Lim, Joshua Robinson, Lingxiao Zhao, Tess Smidt, Suvrit Sra, Haggai Maron, and Stefanie Jegelka. Sign and basis invariant networks for spectral graph representation learning. *arXiv preprint arXiv:2202.13013*, 2022.

Qi Lin, Shuo Yu, Ke Sun, Wenhong Zhao, Osama Alfarraj, Amr Tolba, and Feng Xia. Robust graph neural networks via ensemble learning. *Mathematics*, 10(8), 2022. ISSN 2227-7390. doi: 10.3390/math10081300. URL `https://www.mdpi.com/2227-7390/10/8/1300`.

Xixun Lin, Wenxiao Zhang, Fengzhao Shi, Chuan Zhou, Lixin Zou, Xiangyu Zhao, Dawei Yin, Shirui Pan, and Yanan Cao. Graph neural stochastic diffusion for estimating uncertainty in node classification. In *Forty-first International Conference on Machine Learning*, 2024a.

Xixun Lin, Wenxiao Zhang, Fengzhao Shi, Chuan Zhou, Lixin Zou, Xiangyu Zhao, Dawei Yin, Shirui Pan, and Yanan Cao. Graph neural stochastic diffusion for estimating uncertainty in node classification. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024b.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL `https://openreview.net/forum?id=Bkg6RiCqY7`.

Sergey V Lototsky and Boris L Rozovskii. Stochastic partial differential equations driven by purely spatial noise. *SIAM Journal on Mathematical Analysis*, 41(4):1295–1322, 2009.

Lu Lu, Pengzhan Jin, and George Em Karniadakis. DeepONet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators. *arXiv preprint arXiv:1910.03193*, 2019.

George Ma, Yifei Wang, and Yisen Wang. Laplacian canonization: A minimalist approach to sign and basis invariant spectral embedding. *Advances in Neural Information Processing Systems*, 36: 11296–11337, 2023.

George Ma, Yifei Wang, Derek Lim, Stefanie Jegelka, and Yisen Wang. A canonization perspective on invariant and equivariant learning. *arXiv e-prints*, pp. arXiv–2405, 2024a.

Ziqi Ma, David Pitt, Kamyar Azizzadenesheli, and Anima Anandkumar. Calibrated uncertainty quantification for operator learning via conformal prediction. *Transactions on Machine Learning Research*, 2024b. ISSN 2835-8856. URL `https://openreview.net/forum?id=cGpegxy12T`.

David J. C. MacKay. A practical Bayesian framework for backpropagation networks. *Neural Comput.*, 4(3):448–472, May 1992. ISSN 0899-7667. doi: 10.1162/neco.1992.4.3.448. URL `https://doi.org/10.1162/neco.1992.4.3.448`.

Emilia Magnani, Nicholas Krämer, Runa Eschenhagen, Lorenzo Rosasco, and Philipp Hennig. Approximate bayesian neural operators: Uncertainty quantification for parametric pdes. *arXiv preprint arXiv:2208.01565*, 2022.

Xuhui Meng, Liu Yang, Zhiping Mao, José del Águila Ferrandis, and George Em Karniadakis. Learning functional priors and posteriors from data and physics. *J. Comput. Phys.*, 457(C), May 2022. ISSN 0021-9991. doi: 10.1016/j.jcp.2022.111073. URL `https://doi.org/10.1016/j.jcp.2022.111073`.

Kevin P. Murphy. *Probabilistic Machine Learning: An introduction*. MIT Press, 2022. URL `http://probml.github.io/book1`.

Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. Learning convolutional neural networks for graphs. In *International conference on machine learning*, pp. 2014–2023. PMLR, 2016.

Jan Oldenburg, Finja Borowski, Alper Öner, Klaus-Peter Schmitz, and Michael Stiehm. Geometry aware physics informed neural network surrogate for solving Navier–Stokes equation (gapinn). *Advanced Modeling and Simulation in Engineering Sciences*, 9(1):8, Jun 2022. ISSN 2213-7467. doi: 10.1186/s40323-022-00221-z. URL `https://doi.org/10.1186/s40323-022-00221-z`.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

Alberto Pepe, Mattia Montanari, and Joan Lasenby. Fengbo: a Clifford neural operator pipeline for 3d PDEs in computational fluid dynamics. In *The Thirteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=VsxbWTDHjh`.

Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter Battaglia. Learning mesh-based simulation with graph networks. In *International conference on learning representations*, 2020.

Claudia Prévôt and Michael Röckner. *A concise course on stochastic partial differential equations*, volume 1905. Springer, 2007.

Hippolyt Ritter, Aleksandar Botev, and David Barber. A scalable Laplace approximation for neural networks. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=Skdvd2xAZ`.

Paul Rosa, Slava Borovitskiy, Alexander Terenin, and Judith Rousseau. Posterior contraction rates for Matérn gaussian processes on Riemannian manifolds. *Advances in Neural Information Processing Systems*, 36:34087–34121, 2023.

Walter Rudin. *Fourier analysis on groups*. Courier Dover Publications, 2017.

Cristopher Salvi, Maud Lemercier, and Andris Gerasimovics. Neural stochastic PDEs: Resolution-invariant learning of continuous spatiotemporal dynamics. *Advances in Neural Information Processing Systems*, 35:1333–1344, 2022.

Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter W. Battaglia. Learning to simulate complex physics with graph networks. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009. doi: 10.1109/TNN.2008.2005605.

Nicholas Sharp and Keenan Crane. A Laplacian for nonmanifold triangle meshes. *Computer Graphics Forum*, 39(5):69–80, 2020. doi: https://doi.org/10.1111/cgf.14069. URL `https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.14069`.

Nicholas Sharp, Souhaib Attaiki, Keenan Crane, and Maks Ovsjanikov. DiffusionNet: Discretization agnostic learning on surfaces. *ACM Transactions on Graphics (TOG)*, 41(3):1–16, 2022.

Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.

Eugenio Sinestrari. On the abstract cauchy problem of parabolic type in spaces of continuous functions. *Journal of mathematical analysis and applications*, 107(1):16–66, 1985.

Tapas Tripura and Souvik Chakraborty. Wavelet neural operator for solving parametric partial differential equations in computational mechanics problems. *Computer Methods in Applied Mechanics and Engineering*, 404:115783, 2023. ISSN 0045-7825. doi: https://doi.org/10.1016/j.cma.2022.115783. URL `https://www.sciencedirect.com/science/article/pii/S0045782522007393`.

Nobuyuki Umetani and Bernd Bickel. Learning three-dimensional flow for interactive aerodynamic design. *ACM Trans. Graph.*, 37(4), July 2018. ISSN 0730-0301. doi: 10.1145/3197517.3201325. URL `https://doi.org/10.1145/3197517.3201325`.

Jan von Pichowski, Vincenzo Perri, and Ingo Scholtes. Enhancing graph neural networks with random graph ensembles. In *The Third Learning on Graphs Conference*, 2024. URL `https://openreview.net/forum?id=N8tCUSzUFM`.

Vladimir Vovk, Alex Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Springer-Verlag, Berlin, Heidelberg, 2005. ISBN 0387001522.

John B Walsh. An introduction to stochastic partial differential equations. In *École d'Été de Probabilités de Saint Flour XIV-1984*, pp. 265–439. Springer, 1986.

Xiao Wang, Hongrui Liu, Chuan Shi, and Cheng Yang. Be confident! towards trustworthy graph neural networks via confidence calibration. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA, 2021. Curran Associates Inc. ISBN 9781713845393.

Tobias Weber, Emilia Magnani, Marvin Pförtner, and Philipp Hennig. Uncertainty quantification for fourier neural operators. In *ICLR 2024 Workshop on AI4DifferentialEquations In Science*, 2024.

Zhen Wei, Pascal Fua, and Michaël Bauerheim. Automatic parameterization for aerodynamic shape optimization via deep geometric learning. In *AIAA AVIATION 2023 Forum*, pp. 3471, 2023.

Haixu Wu, Huakun Luo, Haowen Wang, Jianmin Wang, and Mingsheng Long. Transolver: A fast transformer solver for PDEs on general geometries. *arXiv preprint arXiv:2402.02366*, 2024.

Zipeng Xiao, Zhongkai Hao, Bokai Lin, Zhijie Deng, and Hang Su. Improved operator learning by orthogonal attention. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Zhao Xu, Carolin Lawrence, Ammar Shaker, and Raman Siarheyeu. Uncertainty propagation in node classification. In *2022 IEEE International Conference on Data Mining (ICDM)*, pp. 1275–1280, 2022. doi: 10.1109/ICDM54844.2022.00167.

Steve Zelditch. *Eigenfunctions of the Laplacian on a Riemannian manifold*, volume 125. American Mathematical Soc., 2017.

Bohang Zhang, Lingxiao Zhao, and Haggai Maron. On the expressive power of spectral invariant graph neural networks. *arXiv preprint arXiv:2406.04336*, 2024.

Maksim Zhdanov, Max Welling, and Jan-Willem van de Meent. Erwin: A tree-based hierarchical transformer for large-scale physical systems. In *International Conference on Machine Learning (ICML)*, 2025.

Probabilistic DiffusionNet: A geometry informed probabilistic generative neural operator. Supplementary material

A ADDITIONAL DETAILS ABOUT DIFFUSIONNET

A.1 CONNECTION WITH NORM

In Chen et al. (2024), the authors generalise Fourier Neural Operators (FNOs) to an arbitrary Riemannian manifolds (M, g) by replacing the Fourier transform with projection onto the LBO basis $\{\phi_k\}$. The integral transformation layer of Chen et al. (2024) $\mathcal{K}_{\mathcal{T}} \in C(L^2(M)^{d_c}; L^2(M)^{d_c})$ is defined by

$$[\mathcal{K}_{\mathcal{T}}(M)(\mathrm{v})]_{\mathrm{c}} = \sum_{\mathrm{k}=0}^{\infty} \sum_{\mathrm{r}=1}^{\mathrm{d_c}} \mathcal{T}_{\mathrm{k}}^{\mathrm{cr}} \langle \phi_{\mathrm{k}}, \mathrm{v_r} \rangle_{\mathrm{L^2(M)}} \phi_{\mathrm{k}}\,, \quad \mathrm{c} \in [\mathrm{d_c}]\,, \tag{17}$$

where $\mathcal{T} \in \ell^\infty(\mathbb{N}_0)^{c \times c}$ is a bounded multiplier (in practice, one restricts to $\mathcal{T}$ such that $\mathcal{T}_k = 0$ for all $k \geq K$). The resulting architecture with lifting, projection, and residual skips is called Neural Operator on Riemannian Manifolds (NORM). The projection on to the LBO basis only corresponds to a Fourier transform in the strict sense if the manifold M is a locally compact abelian group Rudin (2017) .

In special cases, we recover familiar neural operators:

- If $M = \mathbb{T}^d = [0, 2\pi]^d/\sim$ is the flat torus, then $\mathcal{K}_{\mathcal{T}}(M)$ becomes the convolution layer in FNOs since the LBO basis is given by $\{\phi_k\} = \{e^{i\langle k, \cdot\rangle}\}_{k \in \mathbb{Z}^d}$ Zelditch (2017)[Sec. 4.3] ,
- If $M = S^2$ is the 2-sphere, then $\mathcal{K}_{\mathcal{T}}(M)$ is the convolution operator in Spherical Neural Operators Bonev et al. (2023) since the LBO basis $\{\phi_k\}$ consists of the spherical harmonics Zelditch (2017)[Sec. 4.4] .

The diffusion operator $\mathcal{P}_t$ introduced in Section 2.2, in continuum, is a special case of (17) with spectral multiplier

$$\mathcal{T}_k^{cr} = e^{-\lambda_k t_r} \delta_{cr}\,,$$

where $\delta_{cr} = 0$ if $c \neq r$ and 1 otherwise.

While NORM provides a general framework for neural operators on manifolds, our primary focus is inductive learning across different shapes. This introduces an important consideration: eigenfunctions in a given eigenspace are defined only up to orthogonal transformations—particularly, when the eigenvalue is simple (multiplicity one), they are defined only up to a sign flip. This phenomenon creates the "basis ambiguity problem" well-documented in spectral graph neural networks Huang et al. (2023b); Lim et al. (2022); Zhang et al. (2024).

To address ambiguities when learning across shapes, researchers have proposed several strategies:

- Use a restricted class of multipliers like $\mathcal{P}_{\mathrm{t}}$ which are invariant to sign flips, and more general, orthogonal transformations (see Huang et al. (2023b); Lim et al. (2022); Zhang et al. (2024)),
- Apply an eigenfunction canonisation procedure Ma et al. (2023; 2024a), or
- Randomly apply an orthonormal transformation to eigenfunctions in the same eigenspace (effectively flipping signs in the simple eigenvalue case) Dwivedi et al. (2023)[App. E.1].

Therefore, the general spectral multiplier parameterisation in (17) requires additional considerations to be suitable for inductive learning across shapes. For this reason, we do not benchmark against this architecture.

DiffusionNet uses the restricted diffusion operator $\mathcal{P}_{\mathrm{t}}$, which is naturally immune to the basis ambiguity issue. Our experimental results demonstrate that the restricted nature of $\mathcal{P}_{\mathrm{t}}$ does not compromise performance in practice, especially when complemented with the anisotropic gradient operator, which provides additional discriminative power.

A.2 CONNECTION WITH GINO

In Li et al. (2023d)[Sec. 2], the authors introduce a problem setting analogous to ours (cf. Section 2.1), addressing the challenge of learning the solution map of a boundary value problem from geometry to solution. Their approach, Geometry-Informed Neural Operator (GINO), consists of three components: i) a graph neural operator (GNO) encoder Li et al. (2020a), ii) an FNO latent on a rectangular grid, and iii) a GNO decoder. A similar strategy was employed in Li et al. (2023c), albeit with different encoders and decoders that are diffeomorphic mappings between manifolds. The advantage of moving to a rectangular grid is to leverage the speed of fast Fourier transforms to perform global convolutions.

The encoder of GINO maps a function on a general manifold to a function on a rectangular latent domain, and the decoder performs the opposite operation (though not the true inverse of the encoder). Given embedded manifolds $\mathcal{D}_1 \subset \mathbb{R}^d$ and $\mathcal{D}_2 \subset \mathbb{R}^d$ and function spaces $\mathcal{F}_1(\mathcal{D}_1)^{d_1}$ and $\mathcal{F}_2(\mathcal{D}_2)^{d_2}$, a local linear GNO layer $\mathcal{G}_\theta : \mathcal{F}_1(\mathcal{D}_1)^{d_1} \to \mathcal{F}_2(\mathcal{D}_2)^{d_2}$ is defined by:

$$\mathcal{G}_\theta(v)(x) = \int_{B_r^{(d)}(x)\cap\mathcal{D}_1} k_\theta(x,y)v(y)\mu_{\mathcal{D}_1}(dy)\,, \quad x \in \mathcal{D}_2\,,$$

where $B_r^{(d)}(x) = \{y \in \mathbb{R}^d : |x-y|_{\mathbb{R}^d} < r\}$ is a ball of radius $r \in (0,\infty]$ in $\mathbb{R}^d$, $k_\theta \in C(\mathbb{R}^d \times \mathbb{R}^d; \mathbb{R}^{d_2\times d_1})$ is a parameterised matrix of kernels, and $\mu_{\mathcal{D}_1}$ is a measure on $\mathcal{D}_1$.

A crucial feature of GINO is that the kernel k_θ is defined on the entire embedding space $\mathbb{R}^d$, which allows the decoder to be interpreted as mapping to functions defined throughout $\mathbb{R}^d$. This design makes GINO particularly well-suited to learn the extended solution map discussed in Li et al. (2023d)[Sec. 2]. The trace solution map $F^\dagger$ is not explicitly discussed in Li et al. (2023d). While the experiments in Li et al. (2023d) (ShapeNet car and Ahmed body) focused entirely on surface predictions—effectively learning the trace solution map to functions on Lebesgue measure-zero subsets of the embedding space $\mathbb{R}^3$—the ability to decode to functions throughout the entire volume has been recognised and utilised in subsequent works (see, e.g., Table 3 in Wu et al. (2024)).

In contrast to GINO, DiffusionNet and Probabilistic DiffusionNet are inherently constrained to the surface for every shape input $s \in \mathcal{S}$. We believe this architectural constraint introduces a strong geometric inductive bias, enabling DiffusionNet to achieve competitive performance with a significantly smaller parameter count than GINO while focusing precisely on the surface of interest for the trace solution map $F^\dagger$. By working directly on the manifold rather than extending to the ambient space, DiffusionNet exploits the intrinsic geometry of the problem.

B PROOF OF THEOREM 3.1

Let (M,g) denote a smooth Riemannian manifold embedded in $\mathbb{R}^3$ with the associated pullback embedded metric g. Denote by $W^{\mathfrak{s},2}(M)$, $\mathfrak{s} \in \mathbb{R}$, the corresponding scale of L^2-Sobolev spaces. We note that if $\mathfrak{s} > 1$, then $W^{\mathfrak{s},2}(M) \subset C(M)$ Behzadan & Holst (2022)[Thm. 9.14]. Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a complete probability space supporting an $L^2(M)$-valued Gaussian random field $\mathcal{W}^Q : \Omega \times M \to \mathbb{R}$ is such that its covariance operator $Q \in \mathcal{L}(L^2(M), L^2(M))$ has LBO eigenvectors $\{\phi_k\}_{k=0}^\infty$ and eigenvalues $\{q_k\}_{k=0}^\infty$ (see, Prévôt & Röckner (2007)[Proposition 2.1.6] and Lototsky & Rozovskii (2009)[Thm. 3.2.15]).

Definition B.1. A measurable stochastic process $v : \Omega \times [0,T] \to L^2(M)$ is called a weak solution of (5) on the interval $[0,T]$ if $\mathbb{P}$-a.s., $v \in C([0,T]; L^2(M))$ and $\mathbb{P}$-a.s., for all $t \in [0,T]$,

$$\langle v(t), \phi\rangle_{L^2(M)} = \langle v_0, \phi\rangle_{L^2(M)} + \int_0^t \langle v(s), \Delta_M \phi\rangle_{L^2(M)} ds + \langle \mathcal{W}^Q, \phi\rangle_{L^2(M)} t\,.$$

If, in addition, $\mathbb{P}$-a.s., $v \in L^1([0,T]; W^{2,2}(M))$, then we say v is a strong solution.

Remark B.2. *If v is a strong solution, then it follows that*

$$v(t) = v_0 + \int_0^t \Delta_M v(s) ds + \mathcal{W}^Q t\,,$$

where the equality is understood in $L^2(M)$.

Theorem B.3. *There exists a unique strong solution of (1) with the following properties:*

(i) The solution admits the representation

$$v(t) = \sum_{k=0}^{\infty} \left(e^{-\lambda_k t} \langle v_0, \phi_k \rangle_{L^2(M)} + \frac{\sqrt{q_k}}{\lambda_k} (1 - e^{-\lambda_k t}) z_k \right) \phi_k \,, \tag{18}$$

where $\{z_k\}_{k=0}^{\infty} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$ *;*

(ii) If $\sum_{k=0}^{\infty} \lambda_k^{s-2} q_k < \infty$ *for some* $s \geq 2$*, then for each* $t > 0$*, there is a positive constant* C_t *that tends to infinity as* $t \downarrow 0$ *such that*

$$\mathbb{E}[|v(t)|^2_{W^{s,2}}] \leq C_t (\mathbb{E}[|v_0|^2_{L^2(M)}] + 1) \,;$$

(iii) Conditional on v_0*, the solution* $v(t)$ *is an* $L^2(M)$*-valued Gaussian random field with:*

- *Conditional mean:* $\mathbb{E}[v(t) \mid v_0] = \sum_{k=0}^{\infty} e^{-\lambda_k t} \langle v_0, \phi_k \rangle_{L^2(M)} \phi_k$
- *Conditional covariance operator:* $K(t, s \mid v_0) : L^2(M) \to L^2(M)$ *given by*

$$K(t, s \mid v_0) = \sum_{k=0}^{\infty} \frac{q_k}{\lambda_k^2} (1 - e^{-\lambda_k t})(1 - e^{-\lambda_k s}) (\phi_k \otimes \phi_k) \,.$$

Proof of Theorem 3.1 and B.3. The existence and uniqueness of a weak solution for initial data satisfying $\mathbb{P}$-almost surely $v_0 \in L^2(M)$ follows from classical theory, combining Ball (1977) with Grigor'yan (2009)[Thm. 4.9]. We remark that additional comprehensive treatments for more general equations with space-time white noise can be found in Da Prato & Zabczyk (2014)[Thm. 5.4], Prévôt & Röckner (2007)[Thm. 4.2.4], and Elliott et al. (2012)[Prop. 2.6].

The space-time maximal regularity theory and continuity with respect to initial conditions follow from standard parabolic theory Sinestrari (1985); Amann (2016) with summaries in Da Prato & Zabczyk (2014)[App. A]). When $\mathbb{P}$-almost surely $v_0 \in W^{2,1}(M)$, following [9, Ch. 7 Thm. 5], we obtain $\mathbb{P}$-almost surely $v \in L^2([0,T]; W^{2,2}(M))$, confirming that v is a strong solution.

By Ball (1977), the solution v is also a mild solution satisfying $\mathbb{P}$-a.s. for all $t \in [0,T]$:

$$v(t) = \exp(\Delta_M t) v_0 + \int_0^t \exp(\Delta_M (t-s)) \mathrm{d}s \mathcal{W}^Q \,.$$

Following Prévôt & Röckner (2007)[Prop. 2.1.6], the Gaussian random field $\mathcal{W}^Q$ admits the decomposition

$$\mathcal{W}^Q = \sum_{k=0}^{\infty} \sqrt{q_k} \phi_k z_k \,,$$

which converges in $L^2(\Omega; L^2(M))$. Substituting this into the mild solution formulation and using the eigenfunction expansion of the heat semigroup yields the representation (18), which proves (i).

Expectation solution estimates can be derived from analogous solution estimates in the deterministic theory. In particular, following Evans (2022)[Ch. 7 Thm. 5], one can obtain the variational bound:

$$\mathbb{E}\left[\sup_{t \leq T} |v(t)|^2_{W^{1,2}(M)} \right] + \mathbb{E} \int_0^T |v(t)|^2_{W^{2,2}} dt \leq C \left(\mathbb{E}[|v_0|^2_{W^{1,2}(M)}] + \mathbb{E}[|\mathcal{W}^Q|^2_{L^2(M)}] \right) \,,$$

where $\mathbb{E}[|\mathcal{W}^Q|^2_{L^2(M)}] = \sum_{k=0}^{\infty} q_k$, and hence $v \in L^2(\Omega; L^\infty([0,T]; W^{1,2})) \cap L^2(\Omega; W^{2,2}(M))$.

To establish the $W^{s,2}$ estimates for $s \geq 2$, we employ a truncation argument. Define the finite-dimensional approximation:

$$v_K(t) := \sum_{k=0}^{K} \left(e^{-\lambda_k t} \langle v_0, \phi_k \rangle_{L^2(M)} + \frac{\sqrt{q_k}}{\lambda_k} (1 - e^{-\lambda_k t}) z_k \right) \phi_k \,.$$

Let $\mathfrak{s} \geq 2$. Using the Bessel-potential characterisation of Sobolev spaces on M Behzadan & Holst (2022) and the fact that $\mathbb{E}[|z_k|^2] = 1$:

$$\mathbb{E}\left[|v_K(t)|^2_{W^{\mathfrak{s},2}}\right] = \mathbb{E}\left[\sum_{k=0}^{K}(1+\lambda_k)^{\mathfrak{s}}(v_K(t),\phi_k)^2_{L^2(M)}\right]$$

$$\leq 2\,\overbrace{\mathbb{E}\left[\sum_{k=0}^{K}(1+\lambda_k)^{\mathfrak{s}}e^{-2\lambda_k t}(v_0,\phi_k)^2\right]}^{I_K} + 2\overbrace{\sum_{k=0}^{K}(1+\lambda_k)^{\mathfrak{s}}\frac{q_k}{\lambda_k^2}(1-e^{-\lambda_k t})^2}^{II_K} .$$

By the power mean inequality, for all $\mathfrak{s} \geq 1$, $(1+\lambda_k)^{\mathfrak{s}} \leq 2^{\mathfrak{s}-1}(1+\lambda_k^{\mathfrak{s}})$. Since $\lambda_k \geq 0$, we have $(1-e^{-\lambda_k t})^2 \leq 1$ for all $t \geq 0$. Thus, since Q is trace class, and importantly, $\sum_{k=0}^{\infty}\lambda_k^{\mathfrak{s}-2}q_k < \infty$, we can bound II_K as:

$$II_K = \sum_{k=1}^{K}(1+\lambda_k)^{\mathfrak{s}}\frac{q_k}{\lambda_k^2}(1-e^{-\lambda_k t})^2 \leq \frac{2^{\mathfrak{s}-1}}{\lambda_1^2}\sum_{k=1}^{\infty}q_k + 2^{\mathfrak{s}-1}\sum_{k=1}^{\infty}\lambda_k^{\mathfrak{s}-2}q_k =: C_{II} < \infty ,$$

which establishes the uniform boundedness of II_K with respect to K.

To bound I_K, we define $f_{t,\mathfrak{s}} : \mathbb{R}_+ \to \mathbb{R}_+$ by $f_{t,\mathfrak{s}}(\lambda) = \lambda^{\mathfrak{s}}e^{-2\lambda t}$ and compute its derivative:

$$f'_{t,\mathfrak{s}}(\lambda) = \mathfrak{s}\lambda^{\mathfrak{s}-1}e^{-2\lambda t} - 2t\lambda^{\mathfrak{s}}e^{-2\lambda t} = \lambda^{\mathfrak{s}-1}e^{-2\lambda t}\left(\mathfrak{s} - 2t\lambda\right) .$$

This shows that $f_{t,\mathfrak{s}}$ attains its maximum at $\lambda = \frac{\mathfrak{s}}{2t}$ with maximum value $f_{t,\mathfrak{s}}\left(\frac{\mathfrak{s}}{2t}\right) = \left(\frac{\mathfrak{s}}{2te}\right)^{\mathfrak{s}}$.

Therefore,

$$\lambda_k^{\mathfrak{s}}e^{-2\lambda_k t} \leq \tilde{C}_t := \max\left(\max_{l:\lambda_l < \frac{\mathfrak{s}}{2t}}\lambda_l^{\mathfrak{s}}e^{-2\lambda_l t}, \left(\frac{\mathfrak{s}}{2te}\right)^{\mathfrak{s}}\right) , \quad \forall t > 0 , \quad k \in \mathbb{N}_0 ,$$

where we note that since $\lambda_k \to \infty$ as $k \to \infty$, there are only finitely many k' such that $\lambda_{k'} < \frac{\mathfrak{s}}{2t}$. Indeed, by Weyl's law Chavel (1984)[Ch. 1, Eq. 50], there exists a univeral constant $C > 0$ independent of M such that asymptotically $\lambda_k \sim \frac{C}{\nu(M)}k$, where $\nu(M)$ is the volume of M. Thus, for each $K \in \mathbb{N}$, we can bound I_K as:

$$I_K = \sum_{k=0}^{K}(1+\lambda_k)^{\mathfrak{s}}e^{-2\lambda_k t}(v_0,\phi_k)^2 \leq 2^{\mathfrak{s}-1}(1+\tilde{C}_t)\sum_{k=0}^{K}(v_0,\phi_k)^2 \leq 2^{\mathfrak{s}-1}(1+\tilde{C}_t)|v_0|^2_{L^2(M)} .$$

Therefore, for each $K \in \mathbb{N}$, defining

$$C_t = 2\max(2^{\mathfrak{s}-1}(1+\tilde{C}_t), C_{II}) ,$$

we have the uniform bound:

$$\mathbb{E}\left[|v_K(t)|^2_{W^{\mathfrak{s},2}}\right] \leq C_t(\mathbb{E}[|v_0|^2_{L^2(M)}] + 1) .$$

Passing to the limit as $K \to \infty$ and using the weak lower-semicontinuity of the norm, we obtain

$$\mathbb{E}\left[|v(t)|^2_{W^{\mathfrak{s},2}}\right] \leq \liminf_{K\to\infty}\mathbb{E}\left[|v_K(t)|^2_{W^{\mathfrak{s},2}}\right] \leq C_t(\mathbb{E}[|v_0|^2_{L^2(M)}] + 1) .$$

The linearity of the stochastic heat equation ensures that conditional on the initial data v_0 (which, itself, may be random and non-Gaussian), the solution v is a Gaussian process on $[0,T] \times M$. The conditional mean and covariance operator stated in part of the theorem follow directly from the spectral representation and the independence of the $\{z_k\}$ from v_0. This completes the proof of (iii), and the theorem. ☐

Remark B.4. *It is worth highlighting the recent seminal work on Gaussian processes on manifolds Borovitskiy et al. (2020); Hutchinson et al. (2021); Rosa et al. (2023). While these papers focus on interpolating functions or tensor fields on a fixed Riemannian manifold, our work specifically needs to address the more challenging problem of inference across different manifolds. Probabilistic DiffusionNet is constructed by composing GP stochastic diffusion layers with MLPs, enabling transfer learning between shapes. An interesting direction for future work would be to explore cross-manifold inference purely from a GP perspective.*

C ENCODER FOR AMORTISED INFERENCE

When employing amortised variational inference for fitting PDN, we approximate the intractable posterior distribution of the latent variables $\{z_i\}_{i=1}^N$ given observations of the solution fields $\{\mathrm{y_{s_i}}\}_{\mathrm{i=1}}^{\mathrm{N}}$, inputs $\{\mathrm{v_i}\}_{\mathrm{i=1}}^{\mathrm{N}}$ and meshes $\{M_{s_i}^h\}_{i=1}^N$, with a variational distribution q_φ defined in (15). The variational distribution is designed to be tractable and easy to sample to compute the ELBO (14).

There are a number of choices for $(\mu^\varphi, \Sigma^\varphi)$, and we provide an ablation study in Table 4. In particular, we consider the choices, where we drop the notation on the shape s_i for convenience:

1. Partial Diffusion (PD), in which for each mesh $i \in \{1, \dots, N\}$, channel $c \in \{1, \dots, d_c\}$, and layer $l \in \{1, \dots, L\}$,

$$\mu_l^\varphi := \mu_l^\varphi(\mathrm{y}, \mathrm{M^h}) = [\exp(-\Lambda\tau_c)\Phi^\top \mathrm{A}\mathcal{L}^\mu_{\mathrm{lift}}(\mathrm{y})]_{\mathrm{c=1}}^{\mathrm{d_c}}$$
$$\mathrm{diag}(\Sigma_\mathrm{l}^\varphi)(\mathrm{y}, \mathrm{M^h})(\mathrm{y}, \mathrm{M^h}) = \Big(\text{softplus}\Big[[\exp(-\Lambda\tau_c)\Phi^\top \mathrm{A}\mathcal{L}^\Sigma_{\mathrm{lift}}(\mathrm{y})]_{\mathrm{c=1}}^{\mathrm{d_c}}\Big]\Big)^2$$

   where Φ, Λ, and A denote the eigenvectors, eigenvalues, and mass matrix associated with the cotan-laplacian of M^h, as defined in Section 2.2. $\mathcal{L}^\mu_{\varphi_{\text{lift}}}, \mathcal{L}^\Sigma_{\varphi_{\text{lift}}} : \mathbb{R}^{d_{\text{in}}+d_{\text{out}}} \to \mathbb{R}^{d_c}$ are pointwise lifting layers with learnable parameters φ_{lift}, and $\tau_c \in \mathbb{R}_+$ is the learnable diffusion time associated with channel c. We use τ to distinguish the diffusion time parameter in the encoder from those of the base Diffusion-Net model. We call this *partial* diffusion block in the sense that y is projected into the truncated eigenspace of the Laplace matrix L^h, before the diffusion multiplier and a learnable linear transformation are applied. Notice that (PD) suffers from basis ambiguities in that, for a given eigenvalue, the eigenvectors in a given eigenspace are defined only up to an orthogonal transformation, which reduces to sign ambiguities for simple eigenvalues (c.f. Lim et al. (2022) and Section A.1). We discuss this issue further below.

2. Diffusion followed by an aggregation layer (DN+), in which

$$\mu_l^\varphi(\mathrm{y}, \mathrm{M^h}) = [\mathcal{E}^\mu_{\phi_l} \circ \mathcal{P}_l \circ \mathcal{L}^\mu_{\varphi_{\text{lift}}}](\mathrm{y})\,,$$
$$\mathrm{diag}(\Sigma_l^\varphi(\mathrm{y}, \mathrm{M^h}) = [\mathcal{E}^\Sigma_{\phi_l} \circ \mathcal{P}^\tau_l \circ \mathcal{L}^\Sigma_{\varphi_{\text{lift}}}](\mathrm{y})\,,$$

   where we use $\mathcal{P}^\tau + l$ to denote the diffusion operation ((2)) for layer $l \in \{1, \dots, L\}$ using diffusion times τ, once again to distinguish between the diffusion times in the encoder and those of the DN blocks. $\mathcal{E}^\mu_{\phi_l} \in C(\mathbb{R}^{K\times d_c}; \mathbb{R}^{K\times d_c})$, $\mathcal{E}^\Sigma_{\phi_l} \in C(\mathbb{R}^{K\times d_c}\mathbb{S}_+^{K\times d_c})$ are aggregation layers. In practice, we use Perceiver cross-attention modules (Jaegle et al., 2021) to aggregate (see, also, Calvello et al. (2024)).

Effect of PD's Basis Ambiguity: Under eigenvector sign ambiguity, where $\phi_k \to -\phi_k$ for some modes, the encoder parameters transform as $[\mu_z]_k \to -[\mu_z]_k$ while the standard deviation becomes $\tilde{\sigma}[k] = \text{softplus}(-x_k)$ where x_k denotes the pre-activation value. Using the identity $\text{softplus}(-x) = \text{softplus}(x) - x$, the standard deviation shifts additively. The variance, being the square of the standard deviation, transforms as:

$$\mathrm{diag}(\tilde{\Sigma}_l^\varphi)[k] = (\text{softplus}(x_k) - x_k)^2 \tag{19}$$

Consequently, samples from the encoder under the flipped basis follow $\tilde{z}_k \sim \mathcal{N}(-[\mu_z]_k, \mathrm{diag}(\tilde{\Sigma}_l^\varphi)[k]$. The stochastic diffusion operator incorporates the encoded latent variables as:

$$[\mathfrak{N}_{t,\eta,z}]_{ic} = \sum_{k=1}^K \lambda_k^{\eta_c - 1}(1 - e^{-\lambda_k t_c}) z_k [\Phi_s]_{ik} \tag{20}$$

When both encoder and decoder operate on the same eigenbasis within a forward pass, the sign ambiguity creates a compatible equivariance. Under the flipped basis, the noise term becomes:

$$[\tilde{\mathfrak{N}}_{t,\eta,\tilde{z}}]_{ic} = \sum_{k=1}^K \lambda_k^{\eta_c - 1}(1 - e^{-\lambda_k t_c}) \tilde{z}_k \tilde{\phi}_{k,i} \tag{21}$$

Table 4: Comparison of PD and DN+. We averaged the metric across all samples in the **test split** of the ShapeNet car dataset.

Method	Par (M)	↓RMSE	↓NLL	↓MCAL	↓IS
PD	1.8	**3.87**	**2.74**	0.09	8.02
DN+	7.3	4.03	3.21	**0.08**	**7.87**

Since $\tilde{z}_k$ has mean $-[\mu_z]_k$ and $\tilde{\phi}_k = -\phi_k$, the expected reconstruction satisfies:

$$\mathbb{E}[\tilde{z}_k \cdot \tilde{\phi}_k] = (-[\mu_z]_k) \cdot (-\phi_k) = [\mu_z]_k \cdot \phi_k \tag{22}$$

This demonstrates that the mean reconstruction in physical space remains invariant to eigenvector sign flips, as the sign changes in the latent encoding and eigenbasis cancel exactly.

While the mean reconstruction is preserved, the standard deviation of the stochastic component changes under basis ambiguity. The softplus parameterisation provides crucial stability properties. For small pre-activations $|x_k| \ll 1$, which are encouraged by KL regularisation, the softplus function exhibits near-symmetry: softplus$(x) \approx \log(2) + x/2$ and softplus$(-x) \approx \log(2) - x/2$. This yields standard deviations that differ by approximately x, creating bounded perturbations to the noise magnitude.

The variance of the SP operator's stochastic component scales as:

$$\text{Var}[\mathfrak{N}_{t,\eta,z}] \propto \lambda_k^{2(\eta_c-1)}(1 - e^{-\lambda_k t_c})^2 \cdot \text{Var}[z_k] \tag{23}$$

Under sign flips, while Var$[z_k]$ changes from $\text{diag}(\tilde{\Sigma}_l^{\varphi})[k]$ to $(\text{softplus}(x_k) - x_k)^2$, this variation remains bounded and continuous, preventing the extreme fluctuations that would destabilise training. The KL divergence term in the ELBO further regularises this behaviour by penalizing large deviations from unit variance, naturally driving the system toward regions where softplus$(x) \approx$ softplus$(-x)$.

The PD encoder thus achieves robust performance despite basis ambiguity through the interaction between the encoder's sign equivariance and the SP operator's multiplicative structure. The softplus nonlinearity ensures that variance perturbations under sign flips remain bounded, while the consistent mean reconstruction preserves the fidelity of the physical solution. This mechanism explains the empirical success of PD encoders, as the basis ambiguity effectively acts as a form of stochastic regularisation during training, with the model learning to operate in regimes where sign variations have minimal impact on the overall reconstruction quality.

(DN+) is not subject to such ambiguities. There are, of course, other possible variations in the encoder, such as choosing between data-amortisation, in which $\mu^{\varphi}, \Sigma^{\varphi}$ do not use geometric features b_i, geometry-amortisation, in which $\mu^{\varphi}, \Sigma^{\varphi}$ do not use the output field y_i, or no amortisation in which $\mu^{\varphi}, \Sigma^{\varphi}$ have no dependency on s_i. In Table 4, we apply data-amortisation, however we will explore variations in future work.

Ablation Study In Table 4, we present an ablation of the choices (a) and (b). We compare their performance with the same accuracy and UQ metrics. In both cases, we take 16 PDN layers. In the case of (PD) we repeat the latent across all channels, whereas in (DN+) we learn mean and variance parameters for each channel. Whilst both methods yield similar performance, (PD) has $\sim 4\times$ fewer parameters and so we report its results in the main text as PDN. We anticipate that for larger datasets (DN+) will be more expressive than (PD). In future work we will explore this prospect, as well as considerations for reducing the parameter footprint of (DN+).

D IMPLEMENTATION DETAILS

Preprocessing for (P)DN Following Li et al. (2023d), we used a standard Gaussian normalisation for the pressure fields. We also put all the meshes within a $[-1, 1]$ bounding box, again following Li et al. (2023d).

Optimisation for (P)DN We used the AdamW (Loshchilov & Hutter, 2019) with learning rate (LR) set to 10^{-3} and weight decay set to 10^{-4}. We used the One Cycle LR scheduler with the same hyperparameters as was used in Wu et al. (2024). We trained each model for 200 epochs. For PDN we evaluated the Monte Carlo estimate of the gradient of the ELBO using just a single sample from the encoder.

Architectural specifications of (P)DN For (P)DN we used $K = 128$ modes, and $L = 16$ layers throughout. We chose the number of channels to be the same size as the number of modes ($K = d_c = 128$), following Sharp et al. (2022).

Preprocessing/Optimisation/architectural setup for other baselines We made sure for every baseline we follow the same pressure field normalisation as described above. For optimisation and other geometric preprocessing we retained exactly the same settings as was originally proposed for that particular baseline in its original publication or released code. We did the same for the architecture. Specifically, while using Transolver as a base model, for the UQ experiments, we retained the exact same architecture as was used for the results in Table 3 of Wu et al. (2024). For GINO we used the same architecture as the “decoder" version in Table 2 & 3 in Li et al. (2023d), respectively.

E EXPLANATION OF THE METRICS

To measure predictive accuracy, we use the the relative L^2 norm which, for a given observation $y \in \mathbb{R}^{n_s \times d_{\text{out}}}$ and prediction $\widehat{y} \in \mathbb{R}^{n_s \times d_{\text{out}}}$ on a mesh M^h, is given by

$$\text{RL2}(y, \hat{y}) = \frac{\|y - \widehat{y}\|_2}{\|y\|_2}.$$

This is a metric is widely used in the neural operator literature, allowing us to compare (P)DN with other methods. However, for comparing UQ performance it is important to assess methods in a holistic way. Thus, we have used a number of metrics which we describe next.

We use the following notation: Given a test set mesh $i \in \{M_1^h, \ldots, M_{N_{\text{test}}}^h\}$ and vertices $X_i = [x_{i1}, \ldots, x_{in_i}]^\top \in \mathbb{R}^{n_i \times 3}$ where $n_i = |\mathcal{V}_i^h|$ is the number of vertices in M_i^h, let y_{ij} denote the evaluations of the observed output function at point $x_{ij} \in V_i^h$. For baseline models that support uncertainty quantification via sampling, $\widehat{\mu}_{ij} \in \mathbb{R}^{d_{\text{out}}}$ represents the empirical mean over samples, and $\widehat{\sigma}_{ij} \in \mathbb{R}_+^{d_{\text{out}}}$ denotes the empirical standard deviation. For convenience, we collect $y_i = \{y_{ij}\}_{j=1}^{n_i}$, $\widehat{\mu}_i = \{\mu_{ij}\}_{j=1}^{n_i}$, $\widehat{\sigma}_i = \{\sigma_{ij}\}_{j=1}^{n_i}$.

Note that the empirical mean, standard deviations and other credible intervals were all obtained using 100 samples from the predictive distributions of respective models.

Negative log likelihood (NLL) We used the Gaussian negative log likelihood of the predicted field given by

$$\text{NLL}(y_i, \widehat{\mu}_i, \widehat{\sigma}_i) = -\frac{1}{n_i} \sum_{j=1}^{n_i} \ln \mathcal{N}(y_{ij} | \widehat{\mu}_{ij}, \widehat{\sigma}_{ij}^2 I_{d_{\text{out}}}).$$

Total Miscalibration Area The Miscalibration Area (MCAL) measures the discrepancy between predicted confidence levels and the observed frequencies of events. In particular, let $\pi^{(p)}$ denote expected coverage levels for $p \in [0, 1]$. For each input $i \in [N_{\text{test}}]$ and channel $c \in [d_c]$, the observed coverage at level $\pi^{(p)}$ is:

$$\hat{\pi}_{i,c}^{(p)} = \frac{1}{n_i} \sum_{j}^{n_i} \mathbb{I}\left(\frac{|y_{ij,c} - \hat{\mu}_{ij,c}|}{\widehat{\sigma}_{ij,c}} \leq \Phi^{-1}\left(\frac{1 + \pi^{(p)}}{2} \right) \right)$$

where Φ is the cumulative distribution function of the standard normal and $\mathbb{I}(\cdot)$ is the indicator function. The miscalibration area is given by

$$\text{MCAL}(y_i, \widehat{\mu}_i, \widehat{\sigma}_i) = \frac{1}{d_c} \sum_{c=1}^{d_c} \int_0^1 |\widehat{\pi}_{i,c}^{(p)} - \pi^{(p)}| \, dp.$$

In practice, the integral may be approximated through the trapezoidal rule Chung et al. (2021).

Interval Score Interval score, proposed in Gneiting & Raftery (2007), is a *proper scoring rule* (see definition in Chung et al. (2021)) that is a summary statistic of overall performance of a distributional prediction. In particular the formula for IS is given by

$$\mathrm{IS}(y_i, L_i, U_i) = \frac{1}{n_i \cdot d_{\text{out}}} \sum_{j,c=1}^{n_i, d_{\text{out}}} \Big[(U_{ij,c} - L_{ij,c}) + 2 * \alpha * (L_{ij,c} - y_{ij,c}) * \mathbb{I}(y_{ij,c} < L_{ij,c}) \\ + 2 * \alpha * (y_{ij,c} - U_{ij,c}) * \mathbb{I}(y_{ij,c} > U_{ij,c}) \Big],$$

where α is a parameter representing the level of the prediction interval; U_i, L_i are the upper and lower bounds of the prediction interval, and $\mathbb{I}(\cdot)$ is the indicator function. We calculated U_i, L_i by drawing 100 samples of $\widehat{y}_i$, using a particular chosen α. We chose $\alpha = 0.025$ throughout.

In all probabilistic experiments, we generated 100 samples per prediction, and the reported results are averaged over all N_{test} shapes in the test set.

Why multiple UQ metrics? When extracting probabilistic predictions for UQ one ideally wants good coverage (smaller calibration error) so that we can associate regions (in the input space, for us the mesh vertices) of high uncertainty with erroneous predictions. However, poor predictive performance can also have good calibration since an underperforming model can have good coverage with large credible intervals. Thus it is necessary to also access *sharpness* which quantifies the concentration of the predictive distribution. The best models should be competitive in terms of both sharpness and calibration. Thus, given the inherent trade-off between sharpness and calibration we chose multiple metrics. We assessed calibration through the total miscalibrated area, coverage and sharpness through the interval score.

F DISCUSSION ON THE CHOICE AND IMPLEMENTATION OF CLASSICAL UQ METHODS

Choice of the base models Transolver (TS) and GINO have emerged as state-of-the-art approaches towards building CFD surrogates for industrial scale problems, based on recent benchmarking studies Li et al. (2023d); Pepe et al. (2025); Wu et al. (2024). Moreover, they propose contrasting, but foundational, approaches (e.g. attention mechanism for Transolver) towards building an integral transform layer, for achieving resolution invariance, that can serve as building blocks for similar architectures. UQ performance with these base models, thus would be indicative of UQ capabilities of similar base models. Finally, we have also considered DN as a base model to highlight the benefits of PDN.

Choice of UQ methods We have specifically chosen those UQ methods that have recently been used in the context of neural operators. Laplace Approximation have been applied recently to endow a base FNO model, operating on a regular domain, with UQ capability Weber et al. (2024); Magnani et al. (2022). Model ensembling for extracting uncertainty (Lakshminarayanan et al., 2017), in the context of FNOs on regular domain, has been explored recently in Li et al. (2024) towards building an active learning framework, and was shown to outperform other popular approaches for UQ. In addition to these, we have also considered Monte Carlo Dropout, which is arguably the most popular approach for carrying out UQ for any deep neural network.

We like to point out that we indeed try to implement the Conformal Prediction (CP) approach for NOs proposed in Ma et al. (2024b). However, we could not reproduce the results in Ma et al. (2024b). Due to the lack of access to the officially released code, we decided against includ CP in our benchmarking. However, note that: i) our results already show very good calibration out-of-the-box, at least for ShapeNet cars, and ii) CP can be easily added on top of our method as an additional calibration technique (see Fong & Holmes (2021)).

Laplace Approximation For the Laplace Approximation, we used a last-layer approximation. In particular, we fit the Hessian of a Negative Log-Likelihood loss, using a learned homoscedastic noise variance σ^2, with respect to the parameters θ_f of the final (linear) layer of each baseline model

$$\mathcal{L}(\theta_f) = -\sum_{i=1}^{N} \ln \mathcal{N}(y_i | \mu_{\theta_f}, \sigma^2),$$

where y_i is the observed output field i.e. the pressure field on the mesh vertices, and μ_{θ_f} is the corresponding model (DN, Transolver, GINO) prediction, and m is the total number of shapes in the training dataset.

We approximate the Hessian with the Fisher Information matrix, which is the outer product of the gradients. Following Ritter et al. (2018), we scale the Hessian and add a multiple of the identity to yield

$$H_{M,\tau} = M\mathbb{E}\left[\frac{\partial^2 \log \mathcal{L}(\theta_f)}{\partial \theta_f^2}\right] + \tau I$$

where the expectation is approximated through Monte-Carlo samples and M, τ are hyperparameters to tune on a validation set. In our case we choose $N = 10^8$, $\tau = 500$.

Monte Carlo Dropout Monte Carlo Dropout (DO) is one of the most widely used methods to carry out UQ for a deep neural network. We apply dropout both at the time of training and inference. In Gal & Ghahramani (2016), it was shown that training a neural network while applying dropout is equivalent to variational inference of the posterior distribution of the network weights. At test time, a forward pass through the model, with Dropout, is equivalent to a sample from the posterior predictive distribution of the network output, for us the pressure field. The role of a prior on weights is fulfilled by applying a L^2 regularisation: $p(\theta) \propto w_d \|\theta\|^2$, where w_d is a hyperparameter. In practise, the prior is introduced through controlling the weight-decay hyperparameter in AdamW (Loshchilov & Hutter, 2019).

For all the baselines, we train using the same Gaussian Negative Log-Likelihood loss, using a learned homoscedastic noise variance σ^2, a dropout probability of $p = 0.1$ and as weight-decay of $w_d = 10^{-4}$.

For Transolver DO was applied within the attention and projection layers, for GINO DO was applied on MLPs within the FNO blocks, and for DN this was applied to the MLPs within each diffusion block.

Model Ensemble Model Ensemble (ME) is a computationally intensive, yet conceptually simple approach for carrying out UQ. ME constitutes ensembling the prediction of the same base model, however trained with different parameter initialisations. The intuition behind this approach is that by exploring a basin of attraction, in the parameter space, through multiple local minima, one can obtain a region in the parameter space each of whose points will generate a model output that would correspondingly live within a “confidence" band, thus quantifying uncertainty.

In practise, as suggested in Lakshminarayanan et al. (2017), we trained all baselines using a a Gaussian Negative Log-Likelihood loss, with a heteroskedastic noise i.e. with a spatially-varying variance $\sigma_\theta^2 = \sigma_\theta^2(x)$, where x are the vertices of a mesh and θ are the model parameters. To generate the spatially-varying mean and variance, we used two heads, as was proposed in Li et al. (2024), for the last layer of all baselines. The resulting loss is given by

$$\mathcal{L}(\theta) = -\ln \mathcal{N}(y | \mu_\theta, \sigma_\theta^2).$$

The predictive distribution is then given by the following mixture distribution

$$\frac{1}{P}\sum_{p=1}^{P} \mathcal{N}(\mu_{\theta^{(p)}}, \sigma^2_{\theta^{(p)}}),$$

where $\theta^{(p)}$ is the learned parameter corresponding to the p-th model initialisation. We set $P = 10$ for all experiments.

G ADDITIONAL RESULTS

In total we performed three repeats of all the UQ experiments barring the ones including model ensemble (since this method already includes multiple repeats of optimization with the same model).

In Tables 5- 7, we report the UQ metrics summarized as mean across the test set samples. In Table 2, we furnish the average of these metrics, across all three repeats, for the methods that were repeated. Note the average ranks were calculated using the metrics furnished in Table 2 only.

In addition to the baselines mentioned in the main text, we have also considered a recently proposed sparse attention based architecture Erwin (Zhdanov et al., 2025). In particular, we used Erwin with $\approx 10^6$ parameters (Erwin small, EWS) and with $\approx 5 \times 10^6$ parameters (Erwin medium, EWM). We paired these models with DO based UQ. We found Erwin's performance to be noticeably poor for the Ahmed dataset, and thus we decided to not include Erwin's results on Ahmed. Since, we have only included Erwin's results for Shapenet, thus we have decided to furnish Erwin's results in the appendix only. For Erwin we did a single repeat of the experiments.

Table 5: **First Repeat:** UQ metrics for surface pressure field prediction.

	ShapeNet car				**Ahmed bodies**			
Methods	RMSE↓	NLL↓	MCAL↓	IS↓	RMSE↓	NLL↓	MCAL↓	IS↓
GINO-DO	6.61	**2.46**	**0.08**	10.05	33.51	5.26	0.25	127.96
GINO-LA	4.35	27.2	0.23	15.25	35.37	5.29	0.25	113.54
TS-DO	3.88	2.47	0.11	9.64	21.11	4.62	0.3	120.33
TS-LA	4.03	6.07	0.09	11.28	25.01	5.29	0.26	100.66
DN-DO	5.25	2.96	0.22	24.75	24.62	4.93	0.33	197.45
DN-LA	3.93	6.96	0.09	11.16	20.47	4.71	**0.23**	**65.17**
EWS-DO	3.82	2.54	0.11	8.74	–	–	–	–
EWM-DO	**3.66**	2.51	0.10	8.47	–	–	–	–
PDN (Ours)	3.87	2.74	0.09	**8.02**	**18.34**	**4.21**	0.24	68.28

Table 6: **Second Repeat:** UQ metrics for surface pressure field prediction.

	ShapeNet car				**Ahmed bodies**			
Methods	RMSE↓	NLL↓	MCAL↓	IS↓	RMSE↓	NLL↓	MCAL↓	IS↓
GINO-DO	6.61	**2.46**	**0.08**	10.05	33.51	5.26	0.25	127.96
GINO-LA	4.17	32.21	0.23	14.30	33.48	5.26	0.24	104.66
TS-DO	**3.88**	2.50	0.11	9.7	21.57	4.65	0.3	120.67
TS-LA	3.99	5.39	0.09	11.03	25.01	5.29	0.26	100.66
DN-DO	4.71	2.82	0.22	21.50	28.34	4.93	0.266	185.14
DN-LA	4.30	7.36	0.09	12.72	**20.28**	4.67	0.24	**66.31**
PDN (Ours)	3.93	2.74	0.09	**8.20**	18.45	**4.27**	**0.23**	66.85

Table 7: **Third Repeat:** UQ metrics for surface pressure field prediction.

	ShapeNet car				**Ahmed bodies**			
Methods	RMSE↓	NLL↓	MCAL↓	IS↓	RMSE↓	NLL↓	MCAL↓	IS↓
GINO-DO	6.56	**2.42**	**0.07**	9.94	31.2	5.32	**0.23**	109.57
GINO-LA	4.29	28.49	0.23	14.90	34.89	5.27	0.25	112.73
TS-DO	**3.83**	2.46	0.11	9.38	20.83	4.58	0.3	117.66
TS-LA	4.03	6.07	0.1	11.27	25.01	5.29	0.26	100.66
DN-DO	3.88	2.51	0.2	15.41	25.94	4.91	0.31	187.95
DN-LA	4.32	8.25	0.11	13.08	21.63	4.79	0.24	**69.55**
PDN (Ours)	3.93	2.83	0.08	**7.77**	**18.05**	**4.21**	**0.23**	66.86

G.1 RESULTS ON NON-WATERTIGHT SHAPENET DATASET SPLIT:

Although our proposed approach relies on the assumption of a compact manifold, we relax that assumption in practise as long as we can evaluate the Laplacian. In particular, we apply the method proposed in Sharp & Crane (2020) which can operate on non-manifold shapes. As a result, we were able to use the full ShapeNet car dataset (Umetani & Bickel, 2018) with 789 shapes for training and 100 shapes for test, using the same train-test split as was used in Wu et al. (2024). We report the results for PDN in Table 8. We like to point out that all models were trained using not just the surface pressure field, but also the velocity field in the domain, putting PDN at a disadvantage. Also note that we only report pressure drag and ingore the skin friction component. Nevertheless PDN shows

SOTA performance as it outperforms most baseline methods with a fraction of the parameters (c.f. Table 10).

Table 8: Performance comparison in terms of relative L^2 distance, between ground truth pressure field on the surface of the shape and its prediction, and relative L^2 error for drag coefficient C_d, using the full **ShapeNet car** dataset, used in Wu et al. (2024). Both training and test sets have non-manifold meshes. Except PDN all models were trained using both pressure and velocity information.

Methods	↓Relative L^2 for surface pressure	↓Relative L^2 for C_d
GraphSage (Hamilton et al., 2017)	0.1050	0.0270
PointNet (Charles et al., 2017a)	0.1104	0.0298
Graph U-Net (Gao & Ji, 2019)	0.11024	0.0226
MeshGraphNet Pfaff et al. (2020)	0.0781	0.0168
GNO (Li et al., 2020a)	0.0815	0.0172
GeoFNO Li et al. (2023b)	0.2378	0.0664
GINO (Li et al., 2023d)	0.0810	0.0184
Galerkin (Cao, 2021)	0.0878	0.0179
GNOT (Hao et al., 2023)	0.0798	0.0178
3D-GeoCA (Deng et al., 2024)	0.0779	0.0159
Transolver (Wu et al., 2024)	**0.0745**	**0.0103**
PDN (Ours)	0.0752	0.0168

G.2 ABLATIONS ON MODEL SIZE

In Table 9, we report an ablation study on the number of eigenmodes K and number of PDN layers L, evaluated on the ShapeNet car dataset. We find that there is a clear increase in performance across all UQ metrics as both hyperparameters increase. In the paper we report results for $L = 16$, $K = 128$.

Table 9: Ablations of the number of probabilistic diffusion blocks L and number of eigenmodes K in PDN. We averaged the metric across all samples in the **test set** of the ShapeNet car dataset.

Hyperparameters	↓RMSE	↓NLL	↓MCAL	↓IS
$L = 4, K = 128$	4.24	2.82	0.17	10.9
$L = 8, K = 128$	4.06	3.34	**0.09**	10.42
$L = 16, K = 128$	3.87	2.74	**0.09**	**8.02**
$L = 4, K = 256$	3.90	2.61	0.15	12.52
$L = 8, K = 256$	**3.76**	2.59	0.10	8.95
$L = 16, K = 256$	**3.76**	**2.48**	0.12	9.66

G.3 COMPUTATIONAL ASPECTS

In Table 10 we report the number of parameters of each model, alongside the allocated memory (GB) and average seconds per epoch during training. We found that for smaller meshes (ShapeNet cars, $\sim$ 3500 vertices) PDN, although being memory efficient, is slower than the baselines. However, as the size of meshes (Ahmed bodies, $\sim 100,000$ vertices) grow, PDN runs much faster than the baselines, making it ideal for larger meshes typically found in industrial CFD applications.

In Table 11 we furnish the total time incurred in calculating the operators and caching them, subsequently training and carrying out inference using PDN. Note that each prediction on a test geometry requires 100 forward passes of the model to generate random draws of the pressure field. It should be also be noted that the cotan-Laplacian of a closed 2-manifold is a PSD sparse matrix, and so is amenable to preconditioning methods to accelerate eigendecomposition. Indeed, we employ a Cholesky preconditioning which allows us to eigendecompose Laplacians for meshes of 10^6 nodes in < 6 seconds on GPU, rendering the method suitable for typical industrial applications.

Table 10: Efficiency comparison in terms of parameters, average seconds per epoch and memory (GB) allocated during a training epoch.

	The **ShapeNet car** dataset			The **Ahmed bodies** dataset		
Methods	Parameters (M)	Time (s/epoch)	Memory (GB)	Parameters (M)	Time (s/epoch)	Memory (GB)
GINO	366	**17.86**	18.69	366	237.2	**18.15**
TS	3.8	18.91	1.29	3.8	403.12	52.1
PDN (Ours)	**1.8**	34.6	**0.70**	**1.8**	**88.29**	22.34

Table 11: Computation times for PDN, comprising of Operator caching (OP Caching), incurred training time and inference time, considering 100 draws of the pressure field from the predictive distribution.

	The **ShapeNet car** dataset			The **Ahmed bodies** dataset		
Methods	OP Caching (hours)	Training (hours)	Inference (hours)	OP Caching (hours)	Training (hours)	Inference (hours)
PDN (Ours)	0.18	1.92	0.21	1.77	5	0.5

G.4 FURTHER ANALYSIS OF PDN'S OOD PREDICTIONS

To further probe PDN's out-of-distribution predictions we have carried out a comparison between PDN's predicted dispersion (which we define as sum of predictive variance over the surface) as obtained on 10 randomly drawn test shapes from an out-of-distribution versus the in-distribution splits of the Shapenet cars. We carry out the comparison using a scatter-plot (Fig. 3) between RMSE and dispersion values for each of the test shapes. The scatter-plot in clearly reveals that the uncertainty grows with out-of-distribution shapes.

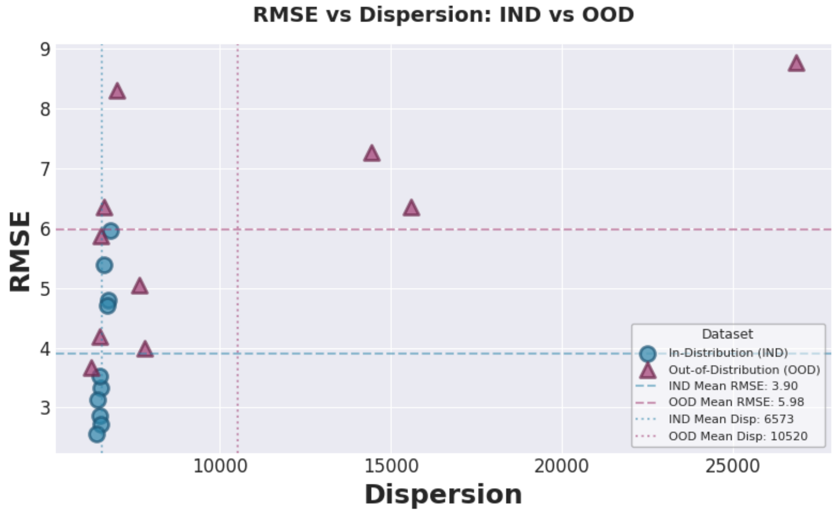


Figure 3: Comparison of PDN's performance (RMSE) versus prediction uncertainty (dispersion) for in-distribution (IND, blue circles) and out-of-distribution (OOD, red triangles) test sets. Dashed and dotted lines indicate mean RMSE and dispersion values, respectively. OOD samples exhibit both higher error rates and greater variation in dispersion, highlighting PDN's ability to increase uncertainty on unseen data distributions.

G.5 CONVERGENCE ANALYSIS OF THE NOISE SCALES η

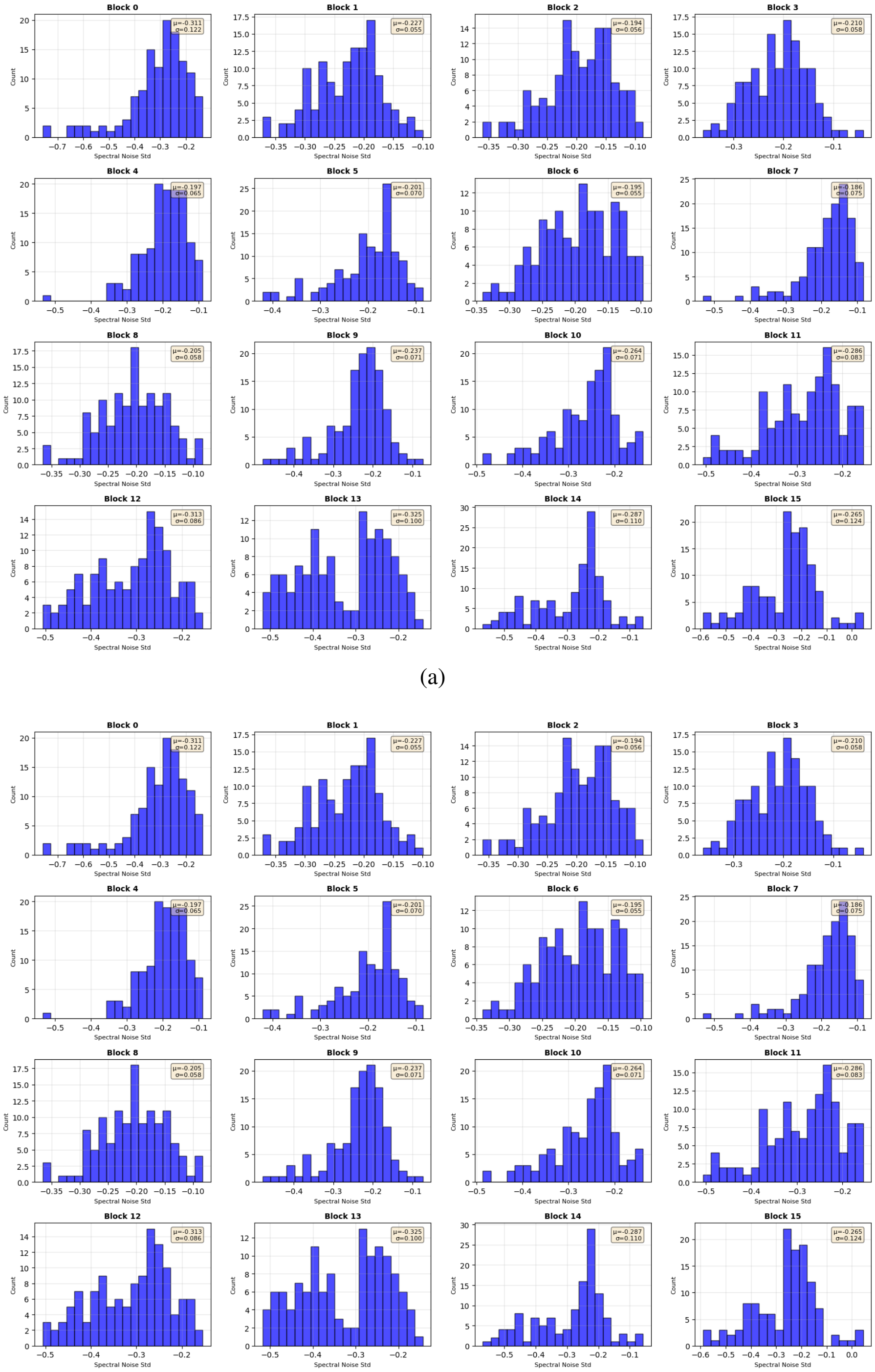
In Fig. 4 we have shown a plot of the histogram (across channels) of learned values of η. Note that we initialised training as $\eta_c = 1$, for each channel, across all blocks. The learned negative values of η, on average, across all channels, and per block, reveal that PDN has discovered a physically-motivated uncertainty structure. In the stochastic diffusion operator, the noise term $[\mathfrak{N}_{t,\eta,z}]_{kc} = \lambda_k^{\eta_c - 1}(1 - e^{-\lambda_k t_c}) z_{kc}$ is weighted by $\lambda_k^{\eta_c - 1}$, which with negative η_c creates inverse spectral weighting: low-frequency modes with small eigenvalues receive large noise amplitudes, while high-frequency modes with large eigenvalues are strongly suppressed. This learned pattern indicates the model expresses uncertainty primarily about large-scale, smooth features rather than fine details. Effectively, the stochastic diffusion operator, Eq. (7), has learned to incorporate a low-pass stochastic perturbation, with similar scale across the layers.

H BROADER IMPACT

This study presents a probabilistic extension for a popular graph neural network architecture, while providing solid theoretical footing for its interpretation as a neural operator for approximating PDE solutions on domain boundaries. The probabilistic extension allows us to provide uncertainty estimates without relying on post hoc methods which might struggle in the operator learning setting. Through robust benchmarking, we show that our approach has significant advantages and, together with our theoretical contributions, our work has the potential to advance computational modelling in scientific and engineering disciplines.

Given the importance of quantifying uncertainty for downstream predictions used for decision-making, how we are able to achieve that with a more efficient model, and how we provide theoretical grounding for further improvements, we recognise that our method could be employed in a wide range of high-stake applications and in sensitive contexts.

As the field adopts our contributions and advances, it will be essential to ensure that methods are used responsibly, particularly in applications where predictive reliability and interpretability are critical. Ongoing dialogue among machine learning researchers, domain experts, and the general public is essential to ensure that the development and deployment of such tools remain aligned with societal values and serve the public interest.



(a)

(b)

Figure 4: Distributions of learned noise scales η, across channels, for each block of PDN. (a) Shapenet car and (b) Ahmed datasets respectively.