

Space-Time Continuous PDE Forecasting using Equivariant Neural Fields

David M. Knigge^{*,1}, David R. Wessels^{*,1}, Riccardo Valperga¹, Samuele Papa¹, Jan-Jakob Sonke²,
 Efstratios Gavves^{†,1}, Erik J. Bekkers^{†,1}
¹University of Amsterdam ² Netherlands Cancer Institute
 d.m.knigge@uva.nl, d.r.wessels@uva.nl

Abstract

Recently, Conditional Neural Fields (NeFs) have emerged as a powerful modelling paradigm for PDEs, by learning solutions as flows in the latent space of the Conditional NeF. Although benefiting from favourable properties of NeFs such as grid-agnosticity and space-time-continuous dynamics modelling, this approach limits the ability to impose known constraints of the PDE on the solutions – e.g. symmetries or boundary conditions – in favour of modelling flexibility. Instead, we propose a space-time continuous NeF-based solving framework that - by preserving geometric information in the latent space - respects known symmetries of the PDE. We show that modelling solutions as flows of pointclouds over the group of interest G improves generalization and data-efficiency. We validated that our framework readily generalizes to unseen spatial and temporal locations, as well as geometric transformations of the initial conditions - where other NeF-based PDE forecasting methods fail - and improve over baselines in a number of challenging geometries.

1 Introduction

Partial Differential Equations (PDEs) are a foundational tool in modelling and understanding spatio-temporal dynamics across diverse scientific domains. Classically, PDEs are solved using numerical methods such as finite elements, finite volumes, or spectral methods. In recent years, Deep Learning (DL) methods have emerged as promising alternatives due to abundance of observed and simulated data as well as the accessibility to computational resources, with applications ranging from fluid simulations and weather modelling [51, 7] to biology [33].

* shared first author, † shared lead advising

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

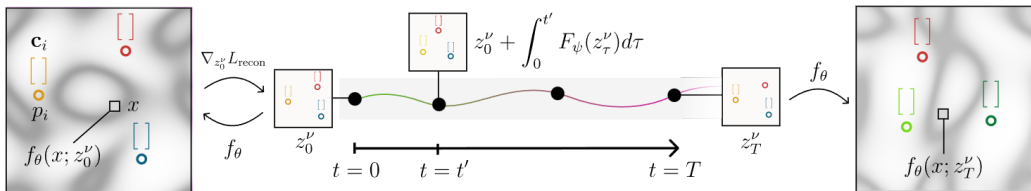


Figure 1: We propose to solve an equivariant PDE in function space by solving an equivariant ODE in latent space. Through our proposed framework, which leverages *Equivariant Neural Fields* f_θ , a field ν_t is represented by a set of latents $z_t^\nu = \{(p_i^\nu, \mathbf{c}_i^\nu)\}_{i=1}^N$ consisting of a *pose* p_i and context vector \mathbf{c}_i . Using meta-learning, the initial latent z_0^ν is fit in only 3 SGD steps, after which an equivariant neural ODE F_ψ models the solution as a latent flow.

The systems modelled by PDEs often have underlying symmetries. For example, heat diffusion or fluid dynamics can be modeled with differential operators which are rotation equivariant, e.g., given a solution to the system of PDEs, its rotation is also a valid solution ¹. In such scenarios it is sensible, and even desirable, to design neural networks that incorporate and preserve such symmetries to improve generalization and data-efficiency [12, 48, 4].

Crucially, DL-based approaches often rely on data sampled on a regular grid, without the inherent ability to generalize outside of it, which is restrictive in many scenarios [40]. To this end, [51] propose to use Neural Fields (NeFs) for modelling and forecasting PDE dynamics. This is done by fitting a neural ODE [11] to the conditioning variables of a conditional Neural Field trained to reconstruct states of the PDE [13]. However, this approach fails to leverage aforementioned known symmetries of the system. Furthermore, using neural fields as representations has proved difficult due to the non-linear nature of neural networks [13, 3, 35], limiting performance in more challenging settings. We posit that NeF-based modelling of PDE dynamics benefits from representations that account for the symmetries of the system as this allows for introducing inductive biases into the model that ought to be reflected in solutions. Furthermore, we show that through meta-learning [28, 45] the NeF backbone improves performance for complex PDEs by further structuring the NeF’s latent space, simplifying the task of the neural ODE.

We introduce a framework for *space-time continuous equivariant PDE solving*, by adapting a class of $SE(n)$ -Equivariant Neural Fields (ENFs) to PDE-specific symmetries. We leverage the ENF as representation for modelling spatiotemporal dynamics. We solve PDEs by learning a flow in the latent space of the ENF - starting at a point z_0 corresponding to the initial state of the PDE - with an equivariant graph-based neural ODE [11] we develop from previous work [5]. We extend the ENF to equivariences beyond $SE(n)$, by extending its weight-sharing scheme to equivalence classes for specific symmetries relevant to our setting. Furthermore, we show how meta-learning [14, 28, 45, 13], can not only significantly reduce inference time of the proposed framework, but also substantially simplify the structure of the latent space of the ENF, thereby simplifying the learning process of the latent dynamics for the neural ODE model. We present the following contributions:

- We introduce a framework for spatio-temporally continuous PDE solving that respects known symmetries of the PDE through equivariance constraints.
- We show that correctly chosen equivariance constraints as inductive bias improves performance of the solver - in terms of MSE - in spatio-temporally continuous settings, i.e. evaluated *off* the training grid and beyond the training horizon.
- We show how meta-learning improves the structure of the latent space of the ENF, simplifying the learning process, leading to better performance in solving PDEs.

We structure the paper as follows: in Sec. 2 we provide an overview of the mathematical preliminaries and describe the problem setting. Our proposed framework is introduced in Sec. 3. We validate our framework on different PDEs defined over a variety of geometries in Sec. 4, with differing equivariance constraints, showing competitive performance over other neural PDE solvers. We provide an in-depth positioning of our approach in relation to other work in Appx. A.

2 Mathematical background and problem setting

Continuous spatiotemporal dynamics forecasting. The setting considered is data-driven learning of the dynamics of a system described by continuous observables. In particular, we consider flows of fields, denoted with $\hat{\nu} : \mathbb{R}^d \times [0, T] \rightarrow \mathbb{R}^c$. We use $\hat{\nu}_t$ as a shorthand for $\hat{\nu}(\cdot, t)$. We assume the flow is governed by a PDE, and consider the Initial Value Problem (IVP) of predicting $\hat{\nu}_t$ from a given ν_0 . The dataset consists of field snapshots $\nu : \mathcal{X} \times \llbracket T \rrbracket \rightarrow \mathbb{R}^c$, in which $\llbracket T \rrbracket := 1, 2, \dots, T$ denotes the set of time points on which the flow is sampled and $\mathcal{X} \subset \mathbb{R}^d$ is a set of coordinate values. For each time point we are given a set of input-output pairs $[\mathcal{X}, \nu(\mathcal{X})]$ where $\nu(\mathcal{X}) \subset \mathbb{R}^c$ are the values of the field at those coordinates. Importantly, the location at which the field is sampled need not be regular, i.e., we do not require the training data to be on a grid or to be regularly spaced in time, nor need coordinate values be identical for train and test sets. Following [51], we distinguish between t_{in} - referring to values within the training time horizon $[0, T]$ - and t_{out} - analogously to values beyond T .

¹Assuming boundary conditions are symmetric, i.e. they transform according to the relevant group action.

Neural Fields in dynamics modelling. Conditional Neural fields (NeFs) are a class of coordinate-based neural networks, often trained to reconstruct discretely-sampled input continuously. More specifically, a conditional neural field $f_\theta : \mathbb{R}^n \rightarrow \mathbb{R}^d$ is a field –parameterized by a neural network with parameters θ – that maps input coordinates $x \in \mathbb{R}^n$ in the data domain alongside conditioning latents z to d -dimensional signal values $\nu(x) \in \mathbb{R}^d$. By associating a conditioning latent $z^\nu \in \mathbb{R}^c$ to each signal ν , a single conditional NeF $f_\theta : \mathbb{R}^n \times \mathbb{R}^c \rightarrow \mathbb{R}^d$ can learn to represent families \mathcal{D} of continuous signals such that $\forall \nu \in \mathcal{D} : f(x) \approx f_\theta(x; z^\nu)$. [51] propose to use conditional NeFs for PDE modelling by learning a continuous flow in the latent space of a conditional neural field. In particular, a set of latents $\{z_i^\nu\}_{i=1}^T$ are obtained by fitting a conditional neural field to a given set of observations $\{\nu_i\}_{i=1}^T$ at timesteps $1, \dots, T$; simultaneously, a neural ODE [11] F_ψ is trained to map pairs of temporally contiguous latents s.t. solutions correspond to the trajectories traced by the learned latents. Though this approach yields impressive results for sparse and irregular data in planar PDEs, we show it breaks down on complex geometries. We hypothesize that this is due to lack of a latent space that preserves relevant geometric transformations that characterize the symmetries of the systems we are modelling, and as such propose an extension of this framework where such symmetries are preserved.

Symmetries and weight sharing. Given a group G with identity element $e \in G$, and a set X , a *group action* is a map $\mathcal{T} : G \times X \rightarrow X$. For simplicity, we denote the action of $g \in G$ on $x \in X$ as $gx := \mathcal{T}(g, x)$, and call G -*space* a smooth manifold equipped with a G action. A group action is homomorphic to G with its group product, namely it is such that $ex = x$ and $(gh)x = g(hx)$. As an example, we are interested in the Special Euclidean group $\text{SE}(n) = \mathbb{R}^n \rtimes \text{SO}(n)$: group elements of $\text{SE}(n)$ are identified by a translation $t \in \mathbb{R}^n$ and rotations $\mathbf{R} \in \text{SO}(n)$ with group operation $gg' = (t, \mathbf{R}_\theta)(t', \mathbf{R}_{\theta'}) = (\mathbf{R}\mathbf{x}' + \mathbf{x}, \mathbf{R}\mathbf{R}_{\theta'})$; We denote by \mathcal{L}_g the left action of G on function spaces defined as $\mathcal{L}_g f(\mathbf{x}') = f(g^{-1}\mathbf{x}') = f(\mathbf{R}_\theta^{-1}(\mathbf{x}' - \mathbf{x}))$. Many PDEs are defined by *equivariant* differential operators such that for a given state ν : $\mathcal{L}_g \mathcal{N}[\nu] = \mathcal{N}[\mathcal{L}_g \nu]$. If the boundary conditions do not break the symmetry, namely if the boundary is symmetric with respect to the same group action, then a G -transformed solution to the IVP for some ν_0 corresponds to the solution for the G -transformed initial value. For example, laws of physics do not depend on the choice of coordinate system, this implies that many PDEs are defined by $\text{SE}(n)$ -equivariant differential operators. The *geometric deep learning* literature shows that models can benefit from leveraging the inherent symmetries or invariances present in the data by constraining the searchable function space through *weight sharing* [9, 25, 5]. Recall that in our framework we model flows of fields, solutions to PDEs defined by equivariant differential operators, with ordinary differential equations in the latent space of conditional neural fields. We leverage the symmetries of the system for two key aspects of the proposed method: first by making the relation between signals and corresponding latents equivariant; second, by using equivariant ODEs, namely ODEs defined by equivariant vector fields: if $\frac{dz}{d\tau} = F(z)$ is such that $F(gz) = gF(z)$, then solutions are mapped to solutions by the group action.

3 Method

We adapt the work of [51], and consider the following optimization problem ²:

$$\min_{\theta, \psi, z_\tau} \mathbb{E}_{\nu \in \mathcal{D}, x \in \mathcal{X}, t \in [T]} \|\nu_t(x) - f_\theta(x; z_t^\nu)\|_2^2, \quad \text{where} \quad z_t^\nu = z_0^\nu + \int_0^t F_\psi(z_\tau^\nu) d\tau, \quad (1)$$

with $f_\theta(x; z_t^\nu)$ a decoder tasked with reconstructing state ν_t from latent z_t^ν and F_ψ a neural ODE that maps a latent to its temporal derivative: $\frac{dz_\tau^\nu}{d\tau} = F_\psi(z_\tau^\nu)$, modelling the solution as flow in latent space starting at the initial latent z_0^ν - see Fig. 1 for a visual intuition.

Equivariant space-time continuous dynamics forecasting. A PDE defined by a G -equivariant differential operator - for which $\mathcal{L}_g \mathcal{N}[\nu] = \mathcal{N}[\mathcal{L}_g \nu]$ - are such that solutions are mapped to other solutions by the group action if the boundary conditions are symmetric. We would like to leverage this property, and *constrain* the neural ODE F_ψ such that the solutions it finds in latent space can be mapped onto each other by the group action. Our motivation for this is twofold: (1) it is natural for

²We highlight that [51] optimize latents z_t^ν , neural field f_θ , and ODE F_ψ using two separate objectives. We instead found that our framework is more stable under single-objective optimization.

our model to have, by construction, the geometric properties that the modelled system is known to possess - (2) to get more structured latent representations and facilitate the job of the neural ODE. To achieve this we first need the latent space Z to be equipped with a well-defined group action with respect to which $\forall g \in G, z \in Z : F_\psi(gz) = gF_\psi(z)$, and, most importantly, we need the relation between the reconstructed field and the corresponding latent to be equivariant, i.e.,

$$\forall g \in G, x \in \mathcal{X} : \mathcal{L}_g f_\theta(x; z'_t) = f_\theta(g^{-1}x; z'_t) = f_\theta(x; gz'_t). \quad (2)$$

Note that, somewhat imprecisely, we call this condition *equivariance* to convey the idea even though it is not, strictly speaking, the commonly used definition of equivariance for general operators. If we consider the decoder as a mapping from latents to fields, we can make the notion of equivariance of this mapping more precise. Namely

$$f(x) = D_\theta(z), D_\theta(z) : z'_t \mapsto f_\theta(\cdot; z'_t), f(g^{-1}x) = D_\theta(gz), D_\theta(gz) : gz'_t \mapsto f_\theta(g^{-1}\cdot; z'_t). \quad (3)$$

In Sec. 3.1 we describe the Equivariant Neural Field (ENF)-based decoder, which satisfies equation (2). Second, in Sec. 3.2 we outline the graph-based equivariant neural ODE. Sec. 3.3 explains the motivation for- and use of- meta-learning for obtaining the ENF backbone parameters. We show how the combination of equivariance and meta-learning produce much more structured latent representations of continuous signals (Fig. 3).

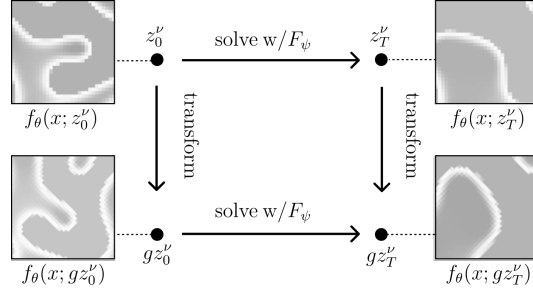


Figure 2: The proposed framework respects predefined symmetries of the PDE: a rotated solution $\mathcal{L}_g \nu_T$ may be obtained either by solving from latent z'_0 (top-left) and transforming the solution z'_T (top-right) to gz'_T (bottom-right) or transforming z'_0 to gz'_0 (bottom-left) and solving this.

3.1 Representing PDE states with Equivariant Neural Fields

We briefly recap ENFs here, referring the reader to [49] for more detail. We extend ENFs to symmetries for PDEs over varying geometries.

ENFs as cross-attention over bi-invariant attributes. Attention-based conditional neural fields represent a signal $\nu \in \mathcal{D}$ with a corresponding *latent set* z^ν [52]. This class of conditional neural fields obtain signal-specific reconstructions $\nu(x) \approx f_\theta(x; z^\nu)$ through a cross-attention operation between the latent set z^ν and input coordinates x . ENFs [49] extend this approach by imposing equivariance constraints w.r.t a group $G \subseteq \text{SE}(n)$ on the relation between the neural field and the latents such that transformations to the signal ν correspond to transformation of the latent z^ν (Eq. (2)). For this condition to hold, we need a well-defined action on the latent space Z of f_θ . To this end, ENFs define elements of the latent set z^ν as tuples of *pose* $p_i \in G$ and *context* $\mathbf{c}_i \in \mathbb{R}^d$, $z^\nu := \{(p_i, \mathbf{c}_i)\}_{i=1}^N$. The latent space is then equipped with a group action defined as $gz = \{(gp_i, \mathbf{c}_i)\}_{i=1}^N$. To achieve equivariance over transformations ENFs follow [5] where equivariance is achieved with convolutional *weight-sharing* over equivalence classes of points pairs x, x' . ENFs instead extend weight-sharing to cross-attention over *bi-invariant* attributes of z, x pairs.

Weight-sharing over bi-invariant attributes of z, x is motivated by Eq. 2, by which we have:

$$f_\theta(x; z) = f_\theta(gx; gz). \quad (4)$$

Intuitively, the above equation says that a transformation g on the domain of f_θ , i.e. $g^{-1}x$, can be undone by *also* acting with g on z . In other words, the output of the neural field f_θ should be *bi-invariant* to g -transformations of the pair z, x . For a specific pair $(z_i, x_m) \in Z \times X$, the term bi-invariant attribute $\mathbf{a}_{i,m}$ describes a function $\mathbf{a} : (z_i, x_m) \mapsto \mathbf{a}(z_i, x_m)$ such that $\mathbf{a}(z_i, x_m) = \mathbf{a}(gz_i, gx_m)$. Throughout the paper we use $\mathbf{a}_{i,m}$ as shorthand for $\mathbf{a}(z_i, x_m)$.

To parameterize f_θ , we can accordingly choose any function that is bi-invariant to G -transformations of z, x . In particular, for an input coordinate x_m ENFs choose to make f_θ a cross-attention operation between attributes $\mathbf{a}_{i,m}$ and the invariant context vectors \mathbf{c}_i :

$$f_\theta(x_m, z) = \text{cross_attn}(\mathbf{a}_{:,m}, \mathbf{c}_:, \mathbf{c}_:) \quad (5)$$

As an example, for $\text{SE}(n)$ -equivariance, we can define the bi-invariant simply using the group action: $\mathbf{a}_{i,m}^{\text{SE}(n)} = p_i^{-1}x_m = \mathbf{R}_i^T(x_m - x_i)$, which is bi-invariant by:

$$\forall g \in \text{SE}(n) : (p_i, x) \mapsto (gp_i, gx) \Leftrightarrow p_i^{-1}x \mapsto (gp_i)^{-1}gx = p_i^{-1}g^{-1}gx = p_i^{-1}x. \quad (6)$$

Bi-invariant attributes for PDE solving. As explained above, ENF is equivariant to $SE(n)$ -transformations by defining f_θ as a function of an $SE(n)$ -bi-invariant attribute $\mathbf{a}^{SE(n)}$. Although many physical processes adhere to roto-translational symmetries, we are also interested in solving PDEs that - due to the geometry of the domain, their specific formulation, and/or their boundary conditions - are not fully $SE(n)$ -equivariant. As such, we are interested in extending ENFs to equivariants that are not strictly (subsets of) $SE(n)$, which we show we can achieve by finding bi-invariants that respect these particular transformations. Below, we provide two examples, the other invariants we use in the experiments - including a "bi-invariant" \mathbf{a}^θ that is not actually bi-invariant to any geometric transformations, which we use to ablate over equivariance constraints - are in Appx. D.

The flat 2-torus. When the physical domain of interest is continuous and extends indefinitely, periodic boundary conditions are often used, i.e. the PDE is defined over a space topologically equivalent to that of the 2-torus. Such boundary conditions break $SO(2)$ symmetries; assuming the domain has periodicity π and none of the terms of this PDE depend on the choice of coordinate frame, these boundary conditions imply that the PDE is equivariant to periodic translations: the group of translations modulo π : $\mathbb{T}^2 \equiv \mathbb{R}^2/\mathbb{Z}^2$. In this case, periodic functions over x, y with periods π would work as a bi-invariant, i.e. using poses $p \in \mathbb{T}^2$, $\mathbf{a}^{\mathbb{T}^2} = \cos(2\pi(x_0 - p_0)) + \cos(2\pi(x_1 - p_1))$ - which happens to be bi-invariant to rotations by $\frac{\pi}{2}$ as well. Instead, since we do not assume any rotational symmetries to exist on the torus, we opt for a non-rotationally symmetric function:

$$\mathbf{a}_{i,m}^{\mathbb{T}^2} = \cos(2\pi(x_i^0 - p_i^0)) \oplus \cos(2\pi(x_i^1 - p_i^1)), \quad (7)$$

where \oplus denotes concatenation. This bi-invariant is used in experiments on Navier-Stokes over the flat 2-Torus.

The 2-sphere. In some settings a PDE may be symmetric only to rotations along a certain axes. An example is that of the global shallow-water equations on the two-sphere - used to model geophysical processes such as atmospheric flow [16], which are characterised by rotational symmetry only along the earth's axis of rotation due to inclusion of a term for Coriolis acceleration that breaks full $SO(3)$ equivariance. We use poses $p \in SO(3)$ parametrised by Euler angles ϕ, θ, γ , and spherical coordinates ϕ, θ for $x \in S^2$. We make the first two Euler angles coincide with the spherical coordinates and define a bi-invariant for rotations around the axis $\theta = \pi$.

$$\mathbf{a}_{i,m}^{SW} = \Delta\phi_{p_i, x_m} \oplus \theta_{p_i} \oplus \gamma_{p_i} \oplus \theta_{x_m}, \quad (8)$$

where $\Delta\phi_{p_i, x_m} = \phi_{p_i} - \phi_{x_m} - 2\pi$ if $\phi_{p_i} - \phi_{x_m} > \pi$ and $\Delta\phi_{p_i, x_m} = \phi_{p_i} - \phi_{x_m} + 2\pi$ if $\phi_{p_i} - \phi_{x_m} < -\pi$, to adjust for periodicity.

In summary, to parameterize an ENF equivariant with respect to a specific group we are simply required to find attributes that are bi-invariant with respect to the same group. In general we achieve this by using group-valued poses and their action on the PDE domain.

3.2 PDE solution as latent space flow

Let z_0^ν be a latent set that faithfully reconstructs the initial state ν_0 . We want to define a neural ODE F_ψ that map latents z_t^ν to their temporal derivatives $\frac{dz_t^\nu}{d\tau} = F_\psi(z_t^\nu)$ that is equivariant with respect to the group action: $gF_\psi(z_t^\nu) = F_\psi(gz_t^\nu)$. To this end, we use a *message passing neural network* (MPNN) to learn a flow of poses p_i and contexts \mathbf{c}_i over time. We base our architecture on PÖNITA [5], which employs convolutional weight-sharing over bi-invariants for $SE(n)$. For an in-depth recap of message-passing frameworks, we refer the reader to Appx. A. Since F_ψ is required to be equivariant w.r.t. the group action, any updates to the poses p_i should also be equivariant. [41] propose to parameterize an equivariant node position update by using a basis spanned by relative node positions $x_j - x_i$. In our setting, poses p_i are points on a manifold M equipped with a group action. As such, we analogously propose parameterizing pose updates by a weighted combination of logarithmic maps $\log_{p_i}(p_j)$, which intuitively describe the relative position between p_i, p_j in the tangent space $T_{p_i}M$, or the displacement from p_i to p_j . We integrate the resulting pose update over the manifold through the exponential map \exp_{p_i} . In the euclidean case $\log_{p_i}(p_j) = x_j - x_i$ and we get back node position updates per [41]. In short, the message passing layers we use consist of the

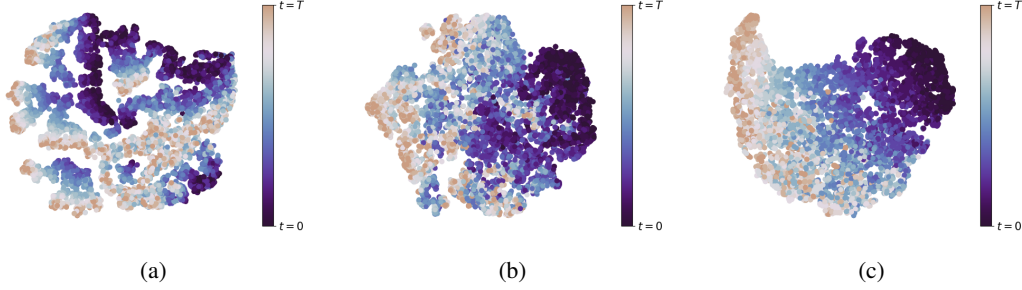


Figure 3: We show the impact of meta-learning and equivariance on the latent space of the ENF when representing trajectories of PDE states. Fig. 3a shows a T-SNE plot of the latent space of f_θ when z_t^ν is optimized with autodecoding, and no weight sharing over bi-invariants is enforced. Fig. 3b shows the latent space when meta-learning is used, but no weight sharing is enforced. Fig. 3c shows the latent space when z_t^ν are obtained using meta-learning and f_θ shares weights over $\mathfrak{a}^{\text{SE}(n)}$.

following update functions:

$$\mathbf{c}_i^{l+1} = \sum_{(p_j, \mathbf{c}_j) \in z^{\nu, l}} k^{\text{context}}(\mathbf{a}_{i,j}^l) \mathbf{c}_j^l, \quad p_i^{l+1} = \exp_{p_i^l} \left(\frac{1}{N} \sum_{(p_j^l, \mathbf{c}_j^l) \in z^{\nu, l}} k^{\text{pose}}(\mathbf{a}_{i,j}^l) \mathbf{c}_j^l \log_{p_i^l}(p_j^l) \right), \quad (9)$$

with $k^{\text{context}}, k^{\text{pose}}$ message functions weighting the incoming context and pose updates, parameterized by a two-layer MLP as a function of the respective bi-invariant.

3.3 Obtaining the initial latent z_0^ν

Until now we’ve not discussed how to obtain latents corresponding to the initial condition z_0^ν . An approach often used in conditional neural field literature is that of autodecoding [36], where latents z^ν are optimized for reconstruction of the input signal ν with SGD. Optimizing a NeF for reconstruction does not necessarily lead to good quality representations [35], i.e. using MSE-based autodecoding to obtain latents z_t^ν - as is proposed by [51] - may complicate the latent space, impeding optimization of the neural ODE F_ψ . Moreover, autodecoding requires many optimization steps at inference (for reference, [51] use 300-500 steps). [13] propose meta-learning as a way to overcome long inference times, as it allows for fitting latents in a few steps - typically three or four. We hypothesize that meta-learning may also structure the latent space - similar to the impact of equivariance constraints, since the very limited number of optimization steps requires efficient organization of latents z_t^ν around the (shared) initialization, forcing together the latent representation of contiguous states. To this end, we propose to use meta-learning for obtaining the initial latent z_0^ν , which is then unrolled by the neural ode F_ψ to find solutions z_t^ν .

3.4 Equivariance and meta-learning structure the latent space Z

As a first validation of the hypotheses that both equivariance constraints and meta-learning introduce structure to the latent space of f_θ , we visualize latent spaces of different variants of the ENF. We fit ENFs to a dataset consisting of solutions to the heat equation for various initial conditions (details in Appx. E). For each sample ν_t , we obtain a set of latents z_t^ν , which we average over the invariant context vectors $\mathbf{c}_i \in \mathbb{R}^c$ to obtain a single vector in \mathbb{R}^c invariant to a group action according to the chosen bi-invariant. Next, we apply T-SNE [47] to the resulting vectors in \mathbb{R}^c . We use three setups: (a) no meta-learning, θ and latents z_t^ν optimized for every ν_t separately using autodecoding [36], and no equivariance imposed (per Eq. 15), shown in Fig. 3a. (b) meta-learning is used to obtain θ, z_t^ν , but no equivariance imposed, shown in Fig. 3b and (c) meta-learning is used to obtain θ, z_t^ν and SE(2)-equivariance is imposed by weight-sharing over $\mathfrak{a}^{\text{SE}(n)}$ bi-invariants, shown in Fig. 3c. The results confirm our intuition that both meta-learning and equivariance improve latent-space structure.

Recap: optimization objective. We use a meta-learning inner-loop [28, 13] to obtain the initial latent z_0^ν under supervision of coordinate-value pairs $(x, \nu(x)_0)_{x \in \mathcal{X}}$ from ν_0 . This latent is unrolled for t_{train} timesteps using F_ψ . The obtained latents are used to reconstruct states z_t^ν along the trajectory

of ν , and parameters of f_θ, F_ψ are optimised for reconstruction MSE, as shown in the left-hand side of Eq. 1. See Appx. B for detailed pseudocode of this process.

4 Experiments

We intend to show the impact of symmetry-preservation in continuous PDE solving. To this end we perform a range of experiments assessing different qualities of our model on tasks with different symmetries. First, we investigate the **equivariance properties** of our framework by evaluating it against unseen geometric transformations of the initial conditions. Next, we assess **generalization** and **extrapolation** capabilities w.r.t. unseen spatial locations and time horizons inside and outside the time ranges seen during training respectively, **robustness** to partial test-time observations, and **data-efficiency**. As the continuous nature of NeF-based PDE solving allows, we verify these properties for PDEs defined over **challenging geometries**: the plane \mathbb{R}^2 , 2-torus \mathbb{T}^2 and the sphere S^2 and the 3D ball \mathbb{B}^3 . Architectural details and hyperparameters are in Appx. E. *Code is available on GitHub*.

Additionally, we validate our model on a benchmark of PDEs that exhibits **no transformation symmetries**: the CFDbench [32] benchmark. We include details on parameter counts, memory usage and runtimes of our model compared to baselines in Appx. F.

4.1 Datasets and evaluation

All datasets are obtained by randomly sampling disjoint sets of initial conditions for train and test sets, and solving them using numerical methods. Dataset-specific details on generation can be found in Appx E. **Heat equation on \mathbb{R}^2 and S^2** . The heat equation describes diffusion over a surface: $\frac{dc}{dt} = D\nabla^2 c$, where c is a scalar field, and D is the diffusivity coefficient. We solve it on the 2D plane where $\nabla^2 c = \frac{\partial^2 c}{\partial x_1^2} + \frac{\partial^2 c}{\partial x_2^2}$ - and on the 2-sphere S^2 where in spherical coordinates: $\nabla^2 c = \left(\frac{1}{\sin \theta} \frac{\partial}{\partial \theta} \left(\sin \theta \frac{\partial c}{\partial \theta} \right) + \frac{1}{\sin^2 \theta} \frac{\partial^2 c}{\partial \phi^2} \right)$. Although a relatively simple PDE, we find that defining it over a non-trivial geometry such as the sphere proves hard for non-equivariant methods. **Navier-Stokes on \mathbb{T}^2** . We solve 2D Navier Stokes [43] for an incompressible fluid with dynamics $\frac{dv}{dt} = -u\nabla v + v\Delta\mu + f, v = \nabla \times u, \nabla u = 0$, where u is the velocity field, v the vorticity, μ the viscosity and f a forcing term (see Appx. E). We create a dataset of solutions for the vorticity using Gaussian random fields as initial conditions. Due to the incompressibility condition, it is natural to solve this PDE with periodic boundary conditions corresponding to the topology of a 2-Torus \mathbb{T}^2 - implying equivariance to periodic translation. **Shallow-water on S^2** . The global shallow-water equations model large-scale oceanic and atmospheric flow on the globe, derived from Navier-Stokes under assumption of shallow fluid depth. The global shallow-water equations (see Appx. E) include terms for Coriolis acceleration, which makes this problem equivariant to rotation along the globe’s axis of rotation. We follow the IVP specified by [16], and create a dataset of paired vorticity-fluid height solutions. **Internally-heated convection in a 3D ball**. We solve the Boussinesq equation for internally heated convection in a ball, a model relevant for example in the context of the Earth’s mantle convection. It involves continuity equations for mass conservation, momentum equations for fluid flow under pressure, viscous forces and buoyancy, and a term modelling heat transfer. We generate initial conditions varying the internal temperature using $N(0, 1)$ noise and obtain solutions for the temperature defined over a regular spherical ϕ, θ, r grid. **CFDBench** [32] consists of a set of one-step solutions (pairs of input and output states ν_t, ν_{t+1}) for classic computational fluid dynamics (CFD) problems, with varying fluid properties, boundary conditions and geometries. The goal of this dataset is to assess generalizability of DL-based PDE solvers over problem parameters. This dataset does not exhibit transformation symmetries because of the absolute position of obstacles in the geometry and orientation of the flow.

Table 1: MSE \downarrow for heat equation on \mathbb{R}^2 .

	$t_{\text{IN TRAIN}}$	$t_{\text{OUT TRAIN}}$	$t_{\text{IN TEST}}$	$t_{\text{OUT TEST}}$
DINo [51]	5.92E-04	2.40E-04	3.85E-03	5.12E-03
Ours \mathbf{a}^{b}	6.23\pm1.01E-06	4.90\pm20.1E-06	2.19 \pm 0.32E-03	5.08 \pm 13.2E-04
Ours $\mathbf{a}^{\text{SE}(2)}$	1.18 \pm 0.45E-05	2.53 \pm 3.50E-05	1.50\pm0.77E-05	2.53\pm3.40E-05

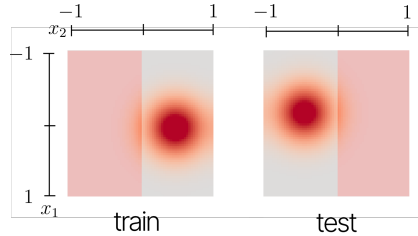


Figure 4: A train and test sample from the planar diffusion dataset. Initial conditions for train and test are spikes in disjoint subsets of \mathbb{R}^2 .

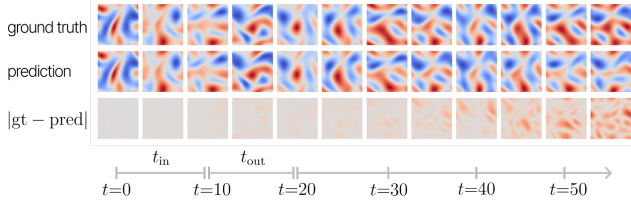


Figure 5: A Navier-Stokes test sample (top) and corresponding predictions from our model (bottom). We visualize predictions in the train horizon $t_{\text{in}} = [0, \dots, 9]$, $t_{\text{out}} = [10, \dots, 20]$ and beyond. The model remains stable well beyond the train horizon, but due to accumulated errors fails to capture dynamics beyond $t > 40$.

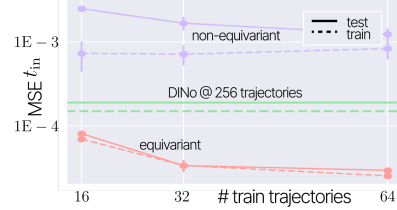


Figure 6: Test MSE t_{in} for increasing training set sizes for the heat equation over the sphere. Equivariant improves over non-equivariant. For reference we show performance of DINo [51] trained on 256 trajectories.

Evaluation. All reported MSE values are for predictions obtained given only the initial condition v_0 , with std over 3 runs. We evaluate two settings for train and test sets both: **generalization setting** with time evolution happening within the seen horizon during training (t_{in}); and, **extrapolation setting** with the time evolution happening outside the seen horizon during training (t_{out}). For both cases we measure the mean-squared error (MSE). To position our work relative to competitive data-driven PDE solvers, on the 2D-Navier-Stokes experiment we provide comparisons with a range of baselines. In most other settings these models cannot straightforwardly be applied, and we only compare to [51], to our knowledge the only other fully continuous PDE solving method in literature. For the Navier-Stokes and Internally-Heated Convection experiments, we compare with Transolver [50], which has shown SOTA results as DL-based PDE solving method for general geometries.

Equivariance properties - heat equation on the plane. To verify our framework respects the posed equivariance constraints, we create a dataset of solutions to the heat equation that *requires* a neural solver to respect equivariance constraints to achieve good performance. Specifically, for initial conditions we randomly insert a pulse of variable intensity in $x = (x_1, x_2) \in \mathbb{R}^2$ s.t. $-1 < x_1 < 1, 0 < x_2 < 1$ for the training data and $-1 < x_1 < 1, -1 < x_2 < 0$ for the test data. Intuitively, train and test sets contain spikes under different disjoint sets of roto-translations (see Fig. 4). We train variants of our framework with ($\mathbf{a}^{\text{SE}(2)}$, Eq. 6) and without (\mathbf{a}^0 , Eq. 15) equivariance constraints. In this dataset, we set $t_{\text{in}} = [0, \dots, 9]$, and evaluation horizon $t_{\text{out}} = [10, \dots, 20]$. Results in Tab. 1 show that the non-equivariant model, as well as the baseline [51] are unable to successfully solve test initial conditions, whereas the equivariant model performs well.

Robustness to subsampling & time-horizons - Navier-Stokes on the 2-Torus. We perform an experiment assessing the impact of equivariance constraints and meta-learning on robustness to sparse test-time observations of the initial condition. To this end, we train a model with (\mathbf{a}^{π^2} , Eq. 7), without (\mathbf{a}^0 , Eq. 15) equivariance constraints, and one with equivariance constraints and without meta-learning (AD \mathbf{a}^{π^2} , Eq. 7), on a fully-observed train set. The training horizon $t_{\text{in}} = [0, \dots, 9]$, and evaluation horizon $t_{\text{out}} = [10, \dots, 20]$. Subsequently, we apply the trained model to the problem of solving from sparse initial conditions v_0 , with observation rates where 50% and 5% of the initial condition is observed (Tab. 2). Approaches operating on discrete (CNODE [2]) and regular grids (FNO [29], G-FNO [20]) perform very well when evaluated on fully-observed regular grids, outperforming continuous approaches (ours, [51]). However, we note

Table 2: MSE \downarrow for Navier-Stokes on \mathbb{T}^2 .

	t_{in} TRAIN	t_{out} TRAIN	t_{in} TEST	t_{out} TEST
100% OF v_0 OBSERVED				
CNODE [2]	6.02E-02	3.35E-01	5.48E-02	3.17E-01
FNO	9.43E-05	2.11E-03	8.44E-05	1.60E-03
G-FNO	3.13E-05	3.49E-04	3.15E-05	3.52E-04
Transolver [50]	1.80E-02	4.85E-01	1.85E-02	4.90E-01
DINo [51]	8.20E-03	6.85E-02	1.11E-02	9.08E-02
Ours AD $\mathbf{a}^{\pi^2/\pi}$	5.60 \pm 0.43E-02	0.37 \pm 0.34E-01	6.75 \pm 0.62E-02	4.00 \pm 0.38E-01
Ours \mathbf{a}^0	1.41 \pm 1.83E-02	1.67 \pm 1.27E-01	2.60 \pm 3.16E-02	2.14 \pm 1.46E-01
Ours $\mathbf{a}^{\pi^2/\pi}$	1.45 \pm 0.08E-03	9.14 \pm 0.36E-03	1.57 \pm 0.09E-03	1.16 \pm 0.14E-02
50% OF v_0 OBSERVED				
CNODE [2]	1.38E-01	6.33E-01	1.52E-01	6.76E-01
FNO	3.31E-02	1.39E-01	3.20E-02	1.47E-01
G-FNO	2.75E-02	1.17E-01	2.32E-02	1.01E-01
Transolver [50]	4.69E-01	0.99E-01	4.76E-01	0.99E-01
DINo [51]	3.67E-02	2.81E-01	3.74E-02	2.83E-01
Ours AD $\mathbf{a}^{\pi^2/\pi}$	6.89 \pm 2.68E-02	3.95 \pm 2.18E-01	7.01 \pm 3.56E-02	4.01 \pm 2.29E-01
Ours \mathbf{a}^0	1.05 \pm 0.04E-02	1.45 \pm 0.01E-01	2.60 \pm 3.16E-02	2.14 \pm 1.46E-01
Ours $\mathbf{a}^{\pi^2/\pi}$	1.50\pm0.17E-03	8.97\pm1.57E-03	5.75\pm2.58E-03	5.03\pm2.62E-02
5% OF v_0 OBSERVED				
CNODE [2]	1.23E+00	2.14E+00	1.20E+00	4.35E+00
FNO	4.13E-01	7.70E-01	3.84E-01	7.07E-01
G-FNO	3.56E-01	7.09E-01	3.40E-01	6.47E-01
Transolver [50]	8.48E-01	1.23E+00	8.28E-01	1.24E+00
DINo [51]	3.67E-02	2.81E-01	3.94E-02	2.91E-01
Ours AD $\mathbf{a}^{\pi^2/\pi}$	6.89 \pm 2.68E-02	3.95 \pm 2.18E-01	7.01 \pm 3.56E-02	4.01 \pm 2.29E-01
Ours \mathbf{a}^0	7.31 \pm 1.37E-02	2.97 \pm 2.43E-01	7.96 \pm 1.65E-02	3.35 \pm 3.41E-01
Ours $\mathbf{a}^{\pi^2/\pi}$	3.19\pm1.07E-02	1.33\pm0.38E-01	3.44\pm1.43E-02	1.61\pm4.93E-01

Table 3: Zero-shot temporal superresolution on Navier-Stokes.

	t_{in} TEST			t_{out} TEST		
	$d\tau=1.0$ (train)	$d\tau=0.5$	$d\tau=0.25$	$d\tau=1.0$ (train)	$d\tau=0.5$	$d\tau=0.25$
DINo [51]	1.19E-02	3.85E-02	3.98E-02	8.82E-02	2.19E-01	2.22E-01
Ours	1.76E-03	1.86E-03	1.91E-03	1.42E-02	1.49E-02	1.52E-02

on fully-observed regular grids, outperforming continuous approaches (ours, [51]). However, we note

that all discrete/regular models greatly deteriorate in performance when observation rates decrease. Equivariance constraints and meta-learning clearly improve performance overall, achieving best performance in all sparse settings. Our proposed framework performs competitively to discrete baselines and other NeF based PDE solving methods [51] in the fully observed setting. To qualitatively assess long-term stability well-beyond the train horizon, we visualize test trajectory and the solution found by our model for $t_{\text{in}} = [0, \dots, 9]$, $t_{\text{out}} = [10, \dots, 20]$ and beyond in Fig. 5. We show error accumulation for solutions on 100% observed and 50% observed initial states in Appx. F. To validate the time-continuous nature of our model, we train a model with supervision at a resolution of $d\tau=1.0$, and subsequently evaluate on resolutions $d\tau=0.5$ and $d\tau=0.25$ (Tab. 3), showing that our framework does not accumulate significant error when deploying on higher temporal resolution.

Data-efficiency - Diffusion on the sphere. To assess the impact of equivariance on data efficiency, we vary the size of the training set of heat equation solutions from 16 to 64 trajectories and apply a model with ($\mathbf{a}^{\text{SO}(3)}$, Eq. 13) and without (\mathbf{a}^0 , Eq. 15) equivariance constraints. In this dataset, we set $t_{\text{in}} = [0, \dots, 9]$, and evaluation horizon $t_{\text{out}} = [10, \dots, 20]$. We visualize t_{in} test- and train MSE in Fig. 6. These results show the non-equivariant model overfitting the training set for smaller numbers of trajectories while unable to solve the PDE satisfactorily, whereas the equivariant model generalizes well even with only 16 training trajectories.

Super-resolution - Shallow-Water on the sphere. Due to their continuous nature, NeF-based approaches inherently support zero-shot super-resolution. In this setting, we generate a set of solutions for the global shallow-water equations over \mathbb{S}^2 at $2\times$ resolution, and apply mean-pooling with a kernel size of 2 to obtain a low-resolution dataset. We train a model that respects rotational symmetries along the rotation axis of the globe (\mathbf{a}^{SW} , Eq. 8) at train resolution, and evaluate the model by solving initial conditions at $2\times$ resolution (Tab. 4, Fig. 7). In this dataset, we set $t_{\text{in}} = [0, \dots, 9]$, and evaluation horizon $t_{\text{out}} = [10, \dots, 14]$. First, we note that our model has difficulty capturing the dynamics near t_{out} - and beyond the training horizon, i.e. $t \Rightarrow 9$ - we suspect because of accumulation of reconstruction errors impacting the ability of F_ψ to model the relatively volatile dynamics of these equations. This points to a drawback of NeF-based solvers: error accumulation starts with the reconstruction error on the initial condition. Ranging over our experiments, we found that this error can be reduced by increasing model capacity, at steep cost of computational complexity attributable to the global attention operator in the ENF backbone. Regarding super-resolution; the model is able to solve the high-resolution initial conditions without inducing significantly increased MSE - it does not produce significant artefacts in the process.

Challenging geometries - Internally heated convection in 3D ball. We show the value of inductive biases in modelling over a challenging geometry. We apply an equivariant model ($\mathbf{a}^{\mathbb{B}^3}$, Eq. 14) to a set of solutions to Boussinesq internally heated convection in a ball defined over a regular ϕ, θ, r -grid, where we set $t_{\text{in}} = [0, \dots, 9]$, and evaluation horizon $t_{\text{out}} = [10, \dots, 14]$. Results (Tab. 6, Fig. 8) for our equivariant model show good generalization compared to a non-equivariant baseline [51]. The significantly larger Transolver [50] model (see Appx. F) obtains better performance within the train time horizon on train and test

Table 4: MSE \downarrow on Shallow-Water equations on the sphere.

	$t_{\text{IN TRAIN}}$	$t_{\text{OUT TRAIN}}$	$t_{\text{IN TEST}}$	$t_{\text{OUT TEST}}$
TRAIN RESOLUTION				
DINo [51]	1.75E-04	1.36E-03	2.01E-04	1.37E-03
Ours \mathbf{a}^{SW}	9.94\pm0.41E-05	1.89 \pm 0.03E-03	1.09\pm1.14E-04	1.87 \pm 0.04E-03
ZERO-SHOT 2X SUPER-RESOLUTION				
DINo [51]	3.03E-04	2.03E-03	3.37E-04	2.03E-03
Ours \mathbf{a}^{SW}	1.58\pm0.02E-04	1.96\pm0.02E-03	1.61\pm0.01E-04	1.93\pm0.02E-03

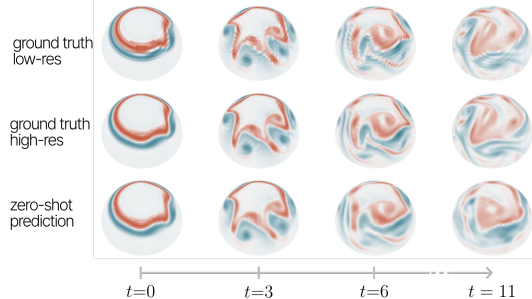


Figure 7: Test samples at train resolution (top), $2\times$ train resolution (middle) and corresponding predictions from our equivariant model (\mathbf{a}^{SW} Eq. 8 (bottom)). The model does not produce significant upsampling artefacts, but fails to capture dynamics outside the training horizon.

Table 5: MSE \downarrow on Internally-Heated Convection in the ball.

	$t_{\text{IN TRAIN}}$	$t_{\text{OUT TRAIN}}$	$t_{\text{IN TEST}}$	$t_{\text{OUT TEST}}$
100% OF ν_0 OBSERVED				
Transolver [50]	3.89E-04	1.88E-02	4.13E-04	2.09E-02
DINo [51]	2.94E-03	7.56E-02	3.06E-03	7.78E-02
Ours $\mathbf{a}^{\mathbb{B}^3}$	5.79 \pm 0.17E-04	7.72\pm0.55E-03	5.99 \pm 0.18E-04	7.97\pm0.46E-03
50% OF ν_0 OBSERVED				
Transolver [50]	4.39E-01	4.99E-01	4.38E-01	5.00E-01
DINo [51]	3.01E-03	1.06E-01	3.02E-03	1.13E-01
Ours $\mathbf{a}^{\mathbb{B}^3}$	6.27E-04	7.76E-03	6.63E-04	8.21E-03

sets, but overfits to this time horizon, generalizing poorly beyond. We additionally show results for sparsely observed input states, showing the limitations of non NeF-based solvers to generalize over irregular changing observation grids. We interpret these results as an indication of a marked reduction in solving-complexity and improved generalization when correctly accounting for a PDE’s symmetries.

Non-symmetric PDEs Lastly, we evaluate our model in the setting when solving PDEs that do not exhibit any global symmetries to assess whether the preservation of symmetries in the latent space of our model precludes application to non-symmetric PDEs. We train a translation-equivariant ($\mathfrak{a}^{\mathbb{R}^2}$) model, i.e. one with equivariance properties identical to a CNN, on the Cavity, Dam and Cylinder flows from CFDBench [32]. Comparing to the baselines set by the dataset authors, we see our method improves significantly over a number of classical baselines, despite no global transformational symmetries being present in the problems being solved.

5 Limitations & Future work

We’re interested in exploring the application of our ENF-based PDE solving framework to larger-scale, more complex problems. Throughout our experiments with ENFs we noticed that more complex signals, e.g. higher resolution PDEs, may be fit easily either by increasing the ENF hidden size or by increasing the number of latents used. However, either of these changes significantly impacts computational complexity, due to the calculation of attention coefficients in the latent space of the ENF for every latent-input coordinate pair. A possible way of addressing this is detailed in [49]; we can approximate the output of the attention operation through limiting the number of latents attended to (using k-nearest neighbours). This may open the door to modelling more complex, larger-scale dynamics than learned in present experiments.

6 Conclusion

We introduce a novel equivariant space-time continuous framework for solving partial differential equations (PDEs). Uniquely - our method handles sparse or irregularly sampled observations of the initial state while respecting symmetry-constraints and boundary conditions of the underlying PDE. We clearly show the benefit of symmetry-preservation over a range of challenging tasks, where existing methods fail to capture the underlying dynamics.

References

- [1] Ilze Amanda Auzina, Çağatay Yıldız, Sara Magliacane, Matthias Bethge, and Efstratios Gavves. Modulated neural odes. *Advances in Neural Information Processing Systems*, 36, 2024.
- [2] Ibrahim Ayed, Emmanuel De Bezenac, Arthur Pajot, and Patrick Gallinari. Learning the spatio-temporal dynamics of physical processes from partial observations. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3232–3236. IEEE, 2020.
- [3] Matthias Bauer, Emilien Dupont, Andy Brock, Dan Rosenbaum, Jonathan Richard Schwarz, and Hyunjik Kim. Spatial functa: Scaling functa to imagenet classification and generation. *arXiv preprint arXiv:2302.03130*, 2023.

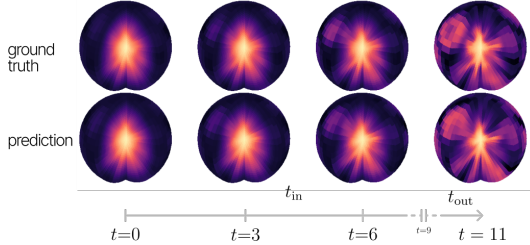


Figure 8: Test samples (top) and corresponding predictions from our model equivariant to S^2 -rotations in the ball. (Eq. 14)

Table 6: Test MSE \downarrow for CFDBench auto-regressive one forward propagation considering all properties for the cavity, dam and cylinder flows. Baselines taken from [32].

	CAVITY	DAM	CYLINDER
Identity	6.42E-02	1.69E-03	7.54E-02
Auto-FFN	6.42E-02	1.68E-03	7.54E-02
Auto-DeepONet	6.39E-02	1.64E-03	7.53E-02
Auto-EDeepONet	6.45E-02	1.49E-03	7.43E-02
Auto-DeepONetCNN	6.33E-02	1.68E-03	7.52E-02
FNO	2.61E-02	8.75E-05	1.15E-03
U-Net	1.58E-02	1.70E-03	5.49E-05
Ours $\mathfrak{a}^{\mathbb{R}^2}$	1.10E-02	8.19E-05	1.42E-05

- [4] Erik J Bekkers. B-spline cnns on lie groups. In *International Conference on Learning Representations*, 2019.
- [5] Erik J Bekkers, Sharvaree Vadgama, Rob D Hesselink, Putri A van der Linden, and David W Romero. Fast, expressive se (n) equivariant networks through weight-sharing in position-orientation space. *arXiv preprint arXiv:2310.02970*, 2023.
- [6] Johannes Brandstetter, Rob Hesselink, Elise van der Pol, Erik J Bekkers, and Max Welling. Geometric and physical quantities improve e (3) equivariant message passing. *arXiv preprint arXiv:2110.02905*, 2021.
- [7] Johannes Brandstetter, Rianne van den Berg, Max Welling, and Jayesh K Gupta. Clifford neural layers for pde modeling. *arXiv preprint arXiv:2209.04934*, 2022.
- [8] Johannes Brandstetter, Daniel Worrall, and Max Welling. Message passing neural pde solvers. *arXiv preprint arXiv:2202.03376*, 2022.
- [9] Michael M Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021.
- [10] Keaton J. Burns, Geoffrey M. Vasil, Jeffrey S. Oishi, Daniel Lecoanet, and Benjamin P. Brown. Dedalus: A flexible framework for numerical simulations with spectral methods. *Physical Review Research*, 2(2):023068, April 2020. doi: 10.1103/PhysRevResearch.2.023068.
- [11] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.
- [12] Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International conference on machine learning*, pages 2990–2999. PMLR, 2016.
- [13] Emilien Dupont, Hyunjik Kim, SM Eslami, Danilo Rezende, and Dan Rosenbaum. From data to functa: Your data point is a function and you can treat it like one. *arXiv preprint arXiv:2201.12204*, 2022.
- [14] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017.
- [15] Marc Finzi, Samuel Stanton, Pavel Izmailov, and Andrew Gordon Wilson. Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data. In *International Conference on Machine Learning*, pages 3165–3176. PMLR, 2020.
- [16] Joseph Galewsky, Richard K Scott, and Lorenzo M Polvani. An initial-value problem for testing numerical models of the global shallow-water equations. *Tellus A: Dynamic Meteorology and Oceanography*, 56(5):429–440, 2004.
- [17] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR, 2017.
- [18] Samuel Greydanus, Misko Dzamba, and Jason Yosinski. Hamiltonian neural networks. *Advances in neural information processing systems*, 32, 2019.
- [19] Xiaoxiao Guo, Wei Li, and Francesco Iorio. Convolutional neural networks for steady flow approximation. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 481–490, 2016.
- [20] Jacob Helwig, Xuan Zhang, Cong Fu, Jerry Kurtin, Stephan Wojtowytsch, and Shuiwang Ji. Group equivariant fourier neural operators for partial differential equations. *Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.*, 2023.
- [21] Quercus Hernández, Alberto Badías, David González, Francisco Chinesta, and Elías Cueto. Structure-preserving neural networks. *Journal of Computational Physics*, 426:109950, 2021.

- [22] Pengzhan Jin, Zhen Zhang, Aiqing Zhu, Yifa Tang, and George Em Karniadakis. Sympnets: Intrinsic structure-preserving symplectic networks for identifying hamiltonian systems. *Neural Networks*, 132:166–179, 2020.
- [23] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [24] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.
- [25] David M Knigge, David W Romero, and Erik J Bekkers. Exploiting redundancy: Separable group convolutional networks on lie groups. In *International Conference on Machine Learning*, pages 11359–11386. PMLR, 2022.
- [26] Miltiadis Miltos Kofinas, Erik Bekkers, Naveen Nagaraja, and Efstratios Gavves. Latent field discovery in interacting dynamical systems with neural fields. *Advances in Neural Information Processing Systems*, 36, 2023.
- [27] Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces. *arXiv preprint arXiv:2108.08481*, 2021.
- [28] Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li. Meta-sgd: Learning to learn quickly for few-shot learning. *arXiv preprint arXiv:1707.09835*, 2017.
- [29] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.
- [30] Yongtuo Liu, Sara Magliacane, Miltiadis Kofinas, and Efstratios Gavves. Graph switching dynamical systems. In *International Conference on Machine Learning*, pages 21867–21883. PMLR, 2023.
- [31] Yongtuo Liu, Sara Magliacane, Miltiadis Kofinas, and Efstratios Gavves. Amortized equation discovery in hybrid dynamical systems, 2024.
- [32] Yining Luo, Yingfa Chen, and Zhen Zhang. Cfdbench: A comprehensive benchmark for machine learning methods in fluid dynamics. *arXiv preprint arXiv:2310.05963*, 2023.
- [33] Philipp Moser, Wolfgang Fenz, Stefan Thumfart, Isabell Ganitzer, and Michael Giretzlehner. Modeling of 3d blood flows with physics-informed neural networks: Comparison of network architectures. *Fluids*, 8(2):46, 2023.
- [34] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.
- [35] Samuele Papa, David M Knigge, Riccardo Valperga, Nikita Moriaikov, Miltos Kofinas, Jan-Jakob Sonke, and Efstratios Gavves. Neural modulation fields for conditional cone beam neural tomography. *arXiv preprint arXiv:2307.08351*, 2023.
- [36] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 165–174, 2019.
- [37] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.
- [38] Adeel Pervez, Francesco Locatello, and Efstratios Gavves. Mechanistic neural networks for scientific machine learning. *arXiv preprint arXiv:2402.13077*, 2024.
- [39] Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter W Battaglia. Learning mesh-based simulation with graph networks. *arXiv preprint arXiv:2010.03409*, 2020.

- [40] Michael Prasthofer, Tim De Ryck, and Siddhartha Mishra. Variable-input deep operator networks. *arXiv preprint arXiv:2205.11404*, 2022.
- [41] Victor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E (n) equivariant graph neural networks. In *International conference on machine learning*, pages 9323–9332. PMLR, 2021.
- [42] Vincent Sitzmann, Eric Chan, Richard Tucker, Noah Snavely, and Gordon Wetzstein. Metasdf: Meta-learning signed distance functions. *Advances in Neural Information Processing Systems*, 33:10136–10147, 2020.
- [43] George Gabriel Stokes et al. On the effect of the internal friction of fluids on the motion of pendulums. 1851.
- [44] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in neural information processing systems*, 33:7537–7547, 2020.
- [45] Matthew Tancik, Ben Mildenhall, Terrance Wang, Divi Schmidt, Pratul P Srinivasan, Jonathan T Barron, and Ren Ng. Learned initializations for optimizing coordinate-based neural representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2846–2855, 2021.
- [46] Riccardo Valperga, Kevin Webster, Dmitry Turaev, Victoria Klein, and Jeroen Lamb. Learning reversible symplectic dynamics. In *Proceedings of The 4th Annual Learning for Dynamics and Control Conference*, volume 168 of *Proceedings of Machine Learning Research*, pages 906–916. PMLR, 23–24 Jun 2022.
- [47] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.
- [48] Maurice Weiler and Gabriele Cesa. General e (2)-equivariant steerable cnns. *Advances in neural information processing systems*, 32, 2019.
- [49] David R Wessels, David M Knigge, Samuele Papa, Riccardo Valperga, Efstratios Gavves, and Erik J Bekkers. Grounding continuous representations in geometry: Equivariant neural fields. *ArXiv Preprint arXiv:*, 2024.
- [50] 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.
- [51] Yuan Yin, Matthieu Kirchmeyer, Jean-Yves Franceschi, Alain Rakotomamonjy, and Patrick Gallinari. Continuous pde dynamics forecasting with implicit neural representations. *arXiv preprint arXiv:2209.14855*, 2022.
- [52] Biao Zhang, Jiapeng Tang, Matthias Niessner, and Peter Wonka. 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models. *ACM Transactions on Graphics (TOG)*, 42(4):1–16, 2023.
- [53] Maksim Zhdanov, David Ruhe, Maurice Weiler, Ana Lucic, Johannes Brandstetter, and Patrick Forré. Clifford-steerable convolutional neural networks. *arXiv preprint arXiv:2402.14730*, 2024.
- [54] David Zwicker. py-pde: A python package for solving partial differential equations. *Journal of Open Source Software*, 5(48):2158, 2020. doi: 10.21105/joss.02158. URL <https://doi.org/10.21105/joss.02158>.

A Related work

DL approaches to dynamics modelling In recent years, the learning of spatiotemporal dynamics has been receiving significant attention, either for modelling interacting systems [31, 30], scientific Machine Learning [51, 8, 7, 38, 26, 53], or even videos [1]. Most DL methods for solving PDEs attempt to directly replace solvers with mappings between finite-dimensional Euclidean spaces, i.e. through the use of CNNs [19, 2] or GNNs [39, 8] often applied autoregressively to an observed (discretized) PDE state. Instead, the Neural Operator (NO) [27] paradigm attempts to learn infinite-dimensional operators, i.e. mappings between function spaces, with limited success. Fourier Neural Operator (FNO) [29] extends this method by performing convolutions in the spectral domain. FNO obtains much improved performance, but due to its reliance on FFT is limited to data on regular grids.

Inductive biases in DL and dynamics modelling Geometric Deep Learning aims to improve model generalization and performance by constraining/designing a model’s space of learnable functions based on geometric principles. Prominent examples include Group Equivariant Convolutional Networks and Steerable CNNs [12, 4], generalizations of CNNs that respect symmetries of the data - such as dilations and continuous rotations [48, 15, 25]. Analogously, Graph Neural Networks (GNNs) [24] or Message Passing Neural Networks (MPNNs) [17] are a variant of neural network that respects set-permutations naturally found in graph data. They are typically formulated for graphs $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with nodes $i \in \mathcal{V}$ and edges \mathcal{E} . Typically nodes are embedded into a node vector f_i^0 , which is subsequently updated over multiple layers of *message passing*. Message passing consists of (1) computing messages $m_{i,j}$ over edges i, j from node j to i with the message function (taking into account edge attributes $a_{i,j}$: $m_{i,j} = \phi_m(f_i^l, f_j^l, a_{i,j})$) (2) aggregating incoming messages: $m_i = \sum_{j \in \mathcal{N}(i)} m_{i,j}$, (3) computing updated node features $f_i^{l+1} = \phi_u(f_i^l, m_i)$.

Recently, such methods have also been adapted for sparse physical data, e.g. for molecular property prediction [41, 6] - where the GNN is additionally required to respect transformation symmetries. [5] unifies these approaches to equivariance under the guise of *weight sharing* over equivalence classes defined by bi-invariant attributes of pairs of nodes i, j , a viewpoint we leverage in constructing the equivariant conditioning latent z_i^ν corresponding to a PDE state ν_t . In the context of dynamics modelling, equivariant architectures have been employed to incorporate various properties of physical systems in the modelling process, examples of such properties are the symplectic structure [22], discrete symmetries such as reversing symmetries [46] and energy conservation [18, 21].

Neural Fields in dynamics modelling Conditional Neural fields (NeFs) are a class of coordinate-based neural networks, often trained to reconstruct discretely-sampled input continuously. More specifically, a conditional neural field $f_\theta : \mathbb{R}^n \rightarrow \mathbb{R}^d$ is a field -parameterized by a neural network with parameters θ - that maps input coordinates $x \in \mathbb{R}^n$ in the data domain alongside conditioning latents z to d -dimensional signal values $f(x) \in \mathbb{R}^d$. By associating a conditioning latent $z^\nu \in \mathbb{R}^c$ to each signal ν , a single conditional NeF $f_\theta : \mathbb{R}^n \times \mathbb{R}^c \rightarrow \mathbb{R}^d$ can learn to represent families \mathcal{D} of continuous signals such that $\forall \nu \in \mathcal{D} : f(\mathbf{x}) \approx f_\theta(\mathbf{x}; \mathbf{z}^\nu)$. [13] showed the viability of using the latents \mathbf{z}^i as representations for downstream tasks (e.g. classification, generation) proposing a framework for *learning on neural fields*. This framework inherits desirable properties of neural fields, such as inherent support for sparsely and/or irregularly sampled data, and independence to signal resolution. [51] propose to use conditional NeFs for PDE modelling by learning a continuous flow in the latent space of a conditional neural field. In particular, a set of latents $\{\mathbf{z}_i^\nu\}_{i=1}^T$ are obtained by fitting a conditional neural field to a given set of observations $\{\nu_i\}_{i=1}^T$ at timesteps $1, \dots, T$; simultaneously, a neural ODE [11] F_ψ is trained to map pairs of temporally continuous latents s.t. solutions correspond to the trajectories traced by the learned latents. Though this approach yields impressive results for sparse and irregular data in planar PDEs, we show it breaks down on more challenging geometries. We hypothesize that this is due to a lack of a latent space that preserves relevant geometric transformation with respect to which systems we are modelling are symmetric, and as such propose an extension of this framework where such symmetries are preserved.

Obtaining Neural Fields representations Most NeF-based approach to representation or reconstruction use SGD to optimize (a subset of) the parameters of the NeF, inevitably leading to significant overhead in inference; conditional NeFs require optimizing a (set of) latents from initialization to reconstruct for a novel sample. Accordingly, research has explored ways of addressing this limitation. [42, 45] propose using Meta-Learning [14, 34] to optimize for an initialization for the NeF from which it is possible to reconstruct for a novel sample in as few as 3 gradient descent steps. [13] propose to meta-learn the NeF backbone, but fix the initialization for the latent \mathbf{z} and instead optimize the learning rate used in its optimization using Meta-SGD [28]. Recently, work has also explored the relation between initialization/optimization of a NeF and its value as downstream representation; [35] show that (1) using a shared NeF initialization and (2) limiting the number of gradient updates to the NeF improves performance in downstream tasks, as this simplifies the complex relation between a NeFs parameter space and its output function space. We combine these insights and make Meta-Learning part of our equivariant PDE solving pipeline, as it enables fast inference and we show it to simplify the latent space of the ENF, improving performance of the neural ODE solver.

B Pseudocode for optimization objective

See Alg. 1 for pseudocode of the training loop that we use, written for a single datasample for simplicity of notation. For simplicity, we further assume we're using an euler stepper to solve the neural ODE, but this can be replaced by any solver. For inference, this stratagem is identical, except we do not perform gradient updates to θ, ψ .

Algorithm 1 Optimization objective

```

Randomly initialize neural field  $f_\theta$ 
Randomly initialize neural ode  $F_\psi$ 
while not done do
  Sample initial states and coordinates  $\nu_0$ .
  Initialize latents  $z'_0 \leftarrow \{(p_i, \mathbf{c}_i)\}_{i=1}^N$ .
  for all step  $\in N_{\text{initial state opt}} = 3$  do
     $z'_0 \leftarrow z'_0 - \epsilon \nabla_{z'_0} \mathcal{L}_{\text{mse}}(f_\theta(\cdot, z'_0), \nu_0)$ 
  end for
  for all  $t \in [1, \dots, t_{\text{in}}]$  do
     $z'_t \leftarrow z'_0 + \int_0^t F_\psi(z'_\tau) d\tau$ 
  end for
  Update  $\theta, \psi$  per:
   $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}'_{\text{mse}}, \psi \leftarrow \psi - \eta \nabla_\psi \mathcal{L}'_{\text{mse}}$  with  $\mathcal{L}'_{\text{mse}} = (\{f_\theta(\cdot, z'_t), \nu_t\}_{t=0}^{t_{\text{in}}})$ 
end while

```

C Equivariant Neural Fields

ENF to reconstruct PDE states For ease of notation we denote \mathbf{P} and \mathbf{C} the matrices containing poses and corresponding appearances stacked row-wise, i.e. $\mathbf{P}_{i,:} = p_i^T$ and $\mathbf{C}_{i,:} = \mathbf{c}_i^T$. Furthermore, we denote \mathbf{A} as the matrix containing all bi-invariants $\mathbf{a}_{i,m}$ stacked row-wise, i.e. $\mathbf{A}_{i,:} = \mathbf{a}_{i,m}^T$:

$$f_\theta(\mathbf{x}; z^{\nu_t}) := \text{softmax} \left(\frac{\mathbf{Q}(\mathbf{A})\mathbf{K}^T(\mathbf{C})}{\sqrt{d_k}} + \mathbf{G}(\mathbf{A}) \right) \mathbf{V}(\mathbf{C}; \mathbf{A}), \quad (10)$$

where the softmax is applied over the latent set and with d_k the hidden dimensionality of the ENF. The query matrix \mathbf{Q} is constructed as $\mathbf{Q} = \mathbf{W}_q \gamma_q^T(\mathbf{A})$, γ_q a Gaussian RFF embedding [44], followed by a linear layer \mathbf{W}_q , i.e. \mathbf{Q} consists of the RFF embedded bi-invariants of the input coordinate x_m and each of the latent poses p_i stacked row-wise. The key matrix is given by a learnable linear transformation \mathbf{W}_k of the context vectors \mathbf{c}_i : $\mathbf{K} = \mathbf{W}_k \mathbf{C}^T$. The attention coefficients which result from the inner product of \mathbf{Q}, \mathbf{K} are weighted by a Gaussian window \mathbf{G} whose magnitude is conditioned on a distance measure on the relative distance between latent poses and input coordinates as: $\mathbf{G}_i = \sigma_{\text{att}}(\|p_i - \mathbf{x}\|^2)$, with σ_{att} a hyperparameter which determines the *locality* of each of the latents. Finally the value matrix is calculated as a learnable linear transformation \mathbf{W}_v of the appearances \mathbf{A} , conditioned through FiLM modulation [37] by a second RFF embedding of the relative poses split into scale- and shift modulations: $\mathbf{V} = \mathbf{W}_v \mathbf{A} \odot \mathbf{W}_{v_\alpha} \gamma_{v_\alpha}(\mathbf{A}) + \mathbf{W}_{v_\beta} \gamma_{v_\beta}(\mathbf{A})$. The latents z'_t are optimized for a single state ν_t , whereas the parameters θ of the ENF backbone - which consist of all the learnable parameters of the linear layers $\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v, \mathbf{W}_{v_\alpha}, \mathbf{W}_{v_\beta}$ used to construct $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ - are shared over all states.

The overall architecture consists of a linear layer $\mathbf{W} \mathbb{R}^c \rightarrow \mathbb{R}^d$ applied to $\mathbf{c}_i \in \mathbb{R}^c$, followed by a layernorm. After this, the cross attention listed above is applied, followed by three d -dim linear layers, the final one mapping to the output dimension \mathbb{R}^{out} .

Equivariance follows from sharing $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ over equivalence classes Note that the latent space of the ENF is equipped with a group action as: $gz'_t = \{(gp_i, \mathbf{a}_i)\}_{i=1}^N$. As an example, SE(2)-equivariance of the ENF follows from bi-invariance of the quantity \mathbf{a} used to construct \mathbf{Q} under the group action:

$$\forall g \in SE(n) : (p_i, \mathbf{x}) \mapsto (gp_i, g\mathbf{x}) \Leftrightarrow p_i^{-1}\mathbf{x} \mapsto (gp_i)^{-1}g\mathbf{x} = p_i^{-1}g^{-1}g\mathbf{x} = p_i^{-1}g^{-1}g. \quad (11)$$

And so, constructing the matrix containing the relative poses of bi-transformed poses and coordinates $(g\mathbf{P})^{-1}g\mathbf{x}$ as $((g\mathbf{P})^{-1}g\mathbf{x})_{i,:} = p_i^{-1}g^{-1}g\mathbf{x} = p_i^{-1}\mathbf{x}$, we trivially have:

$$\forall g \in SE(n) : (p_i, \mathbf{x}) \mapsto (gp_i, g\mathbf{x}) \Leftrightarrow \mathbf{Q}(\mathbf{A}) \mapsto \mathbf{Q}(g\mathbf{A}) = \mathbf{Q}(\mathbf{A}). \quad (12)$$

D Defining additional bi-invariant attributes

Other examples of the bi-invariant attributes that are used in the experiments section are listed here.

Full rotation symmetries on the 2-sphere For the global shallow water equations we defined \mathbf{a}^{SW} as an attribute that is bi-invariant only to rotations over globe’s axis, i.e. rotations over ϕ . In our experiments we also solve diffusion over the sphere, which is fully $\text{SO}(3)$ rotationally symmetric. To achieve equivariance to full 3d rotations, we take poses $p \in \text{SO}(3)$ parameterized by euler angles which act on points $x \in S^2$ parameterized by 3D unit vectors \mathbf{x} through 3D-rotation matrices, allowing us to calculate the bi-invariant $p^{-1}x$:

$$\mathbf{a}_{i,m}^{\text{SO}(3)} = \mathbf{R}_i \mathbf{x}_m. \quad (13)$$

This bi-invariant is used in our experiments for diffusion on the 2-sphere.

The 3D ball \mathbb{B}^3 . We experiment with Boussinesq equation for internally heated convection in a ball. The PDE is fully rotationally symmetric, but since the heat source K is at a fixed point (the center of the ball resp.), it is not symmetric to translations of the initial conditions within the ball. As such, we let $p \in \text{SO}(3) \times \mathbb{R}$ with ϕ, θ, γ, r s.t. $0 < r < 1$. The PDE is defined over spherical coordinates (ϕ, θ, r) , which we map to vectors in $\mathbf{x} \in \mathbb{R}^3$. We then use the following bi-invariant, which is only symmetric to rotations in $\text{SO}(3)$:

$$\mathbf{a}_{i,m}^{\mathbb{B}^3} = \mathbf{R}_i \mathbf{x}_m \oplus r p_i \oplus r x_m. \quad (14)$$

No transformation symmetries. A simple "bi-invariant" for this setting that preserves all geometric information is given by simply concatenating coordinates p with coordinates x :

$$\mathbf{a}_{i,m}^{\emptyset} = p_i \oplus x_m \quad (15)$$

Parameterizing the cross-attention operation in Eq. 5 as function of this bi-invariant results in a framework without any equivariance constraints. We use this in experiments to ablate over equivariance constraints and its impact on performance.

E Experimental Details

E.1 Dataset creation

For creating the dataset of PDE solutions we used py-pde [54] for Navier-Stokes and the diffusion equation on the plane. For the shallow-water equation and the diffusion equation on the sphere, as well as the internally heated convection in a 3D ball we used Dedalus [10].

Diffusion on the plane. For the diffusion equation on the plane we use as initial conditions narrow spikes centred at random locations in the left half of the domain for the train set, and in the right half of the domain for the test set. States are defined on a 64×64 grid ranging from -3 to 3. Initial conditions are randomly sampled uniformly between -2 and 2 for x and 0 and 2 for y in the training set and between -2 and 2 for x and -2 and 0 for y . A random value uniformly sampled between 5.0 and 5.5 is inserted at the randomly sampled location. We solve the equation with an Euler solver for 27 steps, discarding the first 7, with a timestep $dt = 0.01$. We generate 1024 training and 128 test trajectories.

Navier-Stokes on the flat 2-torus. For Navier-Stokes on the flat 2-torus we use Gaussian random fields as initial conditions and solve the PDE using a Crank-Nicholson method with timestep $dt = 1.0$ for 20 steps. The PDE is $\frac{dv}{dt} = -u \nabla v + v \Delta \mu + f$, $v = \nabla \times u$, $\nabla u = 0$, where u is the velocity field, v the vorticity, μ the viscosity and f a forcing term

$$\begin{aligned} \frac{dv}{dt} &= -u \nabla v + v \Delta \mu + f \\ v &= \nabla \times u \\ \nabla u &= 0, \end{aligned}$$

where u is the velocity field, v the vorticity, μ the viscosity and f a forcing term. We set viscosity to $1E-3$, resulting with our setup in a Reynolds number of $\sim \frac{1}{\nu} = 1000$. States are defined on a 64×64 grid. We generate 8192 training and 512 test trajectories.

Diffusion on the 2-sphere. For the diffusion dataset on the sphere, states are defined over a 128×64 ϕ, θ grid. Initial conditions are generated as a gaussian peak inserted at a random point on the sphere with $\sigma = 0.25$. The equation is solved for 20 timesteps with RK4 and $dt = 1.0$. We generate 256 training and 64 test trajectories.

Spherical shallow-water equations [16]. The global shallow-water equations are

$$\begin{aligned}\frac{du}{dt} &= -fk \times u - g\nabla h + \nu\Delta u \\ \frac{dh}{dt} &= -h\nabla \cdot u + \nu\Delta h,\end{aligned}$$

where $\frac{d}{dt}$ is the material derivative, k is the unit vector orthogonal to the surface of the sphere, u is the velocity field that is tangent to the spherical surface and h is the thickness of the fluid layer. The rest are constant parameters of the Earth (see [16] for details). As initial conditions we follow [16] and use basic zonal flow, representing a mid-latitude tropospheric jet, with a correspondingly balanced height field.

$$u(\phi) = \begin{cases} 0 & \text{for } \phi \leq \phi_0 \\ \frac{u_{\max}}{e_n} \exp\left[\frac{1}{(\phi-\phi_0)(\phi-\phi_1)}\right] & \text{for } \phi_0 < \phi < \phi_1 \\ 0 & \text{for } \phi \geq \phi_1 \end{cases}$$

Where $u_{\max} = 80\text{m.s}^{-1}$, $\phi_0 = \pi/7$, $\phi_1 = \pi/2 - \phi_0$, and $e_n = \exp[-4(\phi_1 - \phi_0)^2]$. With this initial zonal flow, we numerically integrate the balance equation

$$gh(\phi) = gh_0 - \int^{\phi} au(\phi') \left[f + \frac{\tan(\phi')}{a} u(\phi') \right] d\phi',$$

to obtain the height h . We then randomly generate small un-balanced perturbations h' to the height field

$$h'(\theta, \phi) = \hat{h} \cos(\phi) e^{-(\theta_2 - \theta/\alpha)^2} e^{-[(\phi_2 - \phi)/\beta]^2}$$

by uniformly sampling $\alpha, \beta, \hat{h}, \theta_2$, and ϕ_2 within a neighbourhood of the values use in [16]. States are defined on a 192×96 grid for the high-resolution dataset, which is subsequently downsampled by 2×2 mean pooling to a 96×48 grid. We generate 512 training trajectories and 64 test trajectories.

Internally-heated convection in the ball. The equations for the internally-heated convection system are listed here, they include thermal diffusivity (κ) and kinematic viscosity (ν), given by:

$$\begin{aligned}\kappa &= (\text{Ra} \cdot \text{Pr})^{-1/2} \\ \nu &= \left(\frac{\text{Ra}}{\text{Pr}} \right)^{-1/2}\end{aligned}$$

We set $\text{Ra} = 1e - 6$ and $\text{Pr} = 1$.

1. Incompressibility condition (continuity equation):

$$\nabla \cdot \mathbf{u} + \tau_p = 0$$

2. Momentum equation (Navier-Stokes equation):

$$\frac{\partial \mathbf{u}}{\partial t} - \nu \nabla^2 \mathbf{u} + \nabla p - \mathbf{r}T + \text{lift}(\tau_u) = -\mathbf{u} \times (\nabla \times \mathbf{u})$$

3. Temperature equation:

$$\frac{\partial T}{\partial t} - \kappa \nabla^2 T + \text{lift}(\tau_T) = -\mathbf{u} \cdot \nabla T + \kappa T_{\text{source}}$$

4. Shear stress boundary condition (stress-free condition):

$$\text{Shear Stress} = 0 \text{ on the boundary}$$

5. No penetration boundary condition (radial component of velocity at $r = 1$):

$$\text{radial}(\mathbf{u}(r = 1)) = 0$$

6. Thermal boundary condition (radial gradient of temperature at $r = 1$):

$$\text{radial}(\nabla T(r = 1)) = -2$$

7. Pressure gauge condition:

$$\int p dV = 0$$

The boundary conditions imposed are stress-free and no-penetration for the velocity field and a constant thermal flux at the outer boundary. These conditions are enforced using penalty terms (τ) that are lifted into the domain using higher-order basis functions.

States are defined over a $64 \times 24 \times 24 \phi, \theta, r$ grid. We use a SBDF2 solver which we constrain by $dt_{\min} = 1e - 4$ and $dt_{\max} = 2e - 2$. We evolve the PDE for 26 timesteps, discarding the first 6. We generate 512 training trajectories and 64 test trajectories.

E.2 Training details

We provide hyperparameters per experiment. We optimize the weights of the neural field f_θ , and neural ODE F_ψ with Adam [23] with a learning rate of 1E-4 and 1E-3 respectively. We initialize the inner learning rate that we use in Meta-SGD [28] for learning z^ν at 1.0 for p and 5.0 for \mathbf{c} . For the neural ODE F_ψ , we use 3 of our message passing layers in the architecture specified in [5], with a hidden dimensionality of 128. The std parameter of the RFF embedding functions $\gamma_q, \gamma_{v_\alpha}, \gamma_{v_\beta}$ (see Appx. C), is chosen per experiment. We run all experiments on a single A100. All experiments are ran 3 times.

Diffusion on the plane. We use 4 latents with $\mathbf{c} \in \mathbb{R}^{16}$. We set the hidden dim of the ENF to 64 and use 2 attention heads. We train the model for 1000 epochs. We set $\gamma_q = 0.05, \gamma_{v_\alpha} = 0.01, \gamma_{v_\beta} = 0.01$. We use a batch size of 8. The model takes approximately 8 hours to train.

Navier-Stokes on the flat 2-torus. We use 4 latents with $\mathbf{c} \in \mathbb{R}^{16}$. We set the hidden dim of the ENF to 64 and use 2 attention heads. We train the model for 2000 epochs. We set $\gamma_q = 0.05, \gamma_{v_\alpha} = 0.2, \gamma_{v_\beta} = 0.2$. We use a batch size of 4. The model takes approximately 48 hours to train.

Diffusion on the 2-sphere. We use 18 latents with $\mathbf{c} \in \mathbb{R}^4$. We set the hidden dim of the ENF to 16 and use 2 attention heads. We train the model for 1500 epochs. We set $\gamma_q = 0.01, \gamma_{v_\alpha} = 0.01, \gamma_{v_\beta} = 0.01$. We use a batch size of 2. The model takes approximately 12 hours to train.

Spherical shallow-water equations [16]. We use 8 latents with $\mathbf{c} \in \mathbb{R}^{32}$. We set the hidden dim of the ENF to 128, and use 2 attention heads. We train the model for 1500 epochs. We set $\gamma_q = 0.05, \gamma_{v_\alpha} = 0.2, \gamma_{v_\beta} = 0.2$. We use a batch size of 2. The model takes approximately 24 hours to train.

Internally-heated convection in the ball We use 8 latents with $\mathbf{c} \in \mathbb{R}^{32}$. We set the hidden dim of the ENF to 128, and use 2 attention heads. We train the model for 1500 epochs. We set $\gamma_q = 0.05, \gamma_{v_\alpha} = 0.2, \gamma_{v_\beta} = 0.2$. We use a batch size of 2. The model takes approximately 24 hours to train.

CFDBench [32] We use 25 latents with $\mathbf{c} \in \mathbb{R}^{16}$. We set the hidden dim of the ENF to 128, and use 1 attention head. We set $\gamma_q = 0.05, \gamma_{v_\alpha} = 0.1, \gamma_{v_\beta} = 0.1$. We train the model for 100 epochs on a single A100 GPU, taking about 16 hours.

Baselines As baseline models on Navier-Stokes we train FNO and GFNO [29] with 8 modes and 32 channels for 700 epochs (until convergence). We train CNODE [2] with 4 layers of size 64 for 300 epochs (until convergence). We train DINO on all experiments for 2000 epochs with an architecture as specified in [51]. For the IHC and shallow-water experiments, we increase the latent dim from 100 to 200, the number of layers for the neural ODE from 3 to 5, and the latent dim of the neural field decoder from 64 to 256, as per [51].

For the Navier-Stokes and Internally-Heated Convection experiments we additionally train the Transolver [50] model. We adapt the Transolver model in PyTorch from the official github repository, and use the same hyperparameter settings as in the original paper (resulting in a 7.1M parameter model) - modifying the training objective to be identical to the autoregressive one we use in our experiments (i.e. mapping from frame to frame). During training in the Navier-Stokes task we observed noticeable instabilities, with the method producing high-frequency artefacts in rollouts in this autoregressive setting. For sparsely subsampled initial conditions, performance deteriorates further - highlighting the benefit of NeF-based continuous PDE solving.

For experiments on internally-heated convection, due to the large size of the input frames, we were required to scale down the Transolver model size in order to be able to fit the model on our A100 GPU, from 256 to 64 hidden units, resulting in a 1.2M parameter model - somewhat comparable in size to the model we use in our experiments (889K). We train the model for 2000 epochs on an A100 GPU, taking approximately 30 hours.

F Additional results

Parameter counts, memory and time efficiency In order to compare parameter, memory and time efficiency of our method to other baselines, we provide details of our models compared to [51] in Tab. 7, and memory usage/inference time compared to the models we used as baselines in the Navier-Stokes experiments in 8. We note that although our method is not the most memory efficient - meta-learning requires a significant computational overhead - it is the fastest in inference settings when unrolling an unseen state. This is attributable to (1) the fact that we’re using a latent-space solver which operates on a drastically compressed representation of the PDE state, and use (2) meta-learning over auto-decoding, which means we only require 3 gradient descent steps to obtain the initial state compared to 300-500 typically used in auto-decoding [51].

Table 7: Parameter count for model used in Navier-Stokes, Shallow-Water and IHC experiments.

	# PARAMS f_θ	# PARAMS F_ψ	TOT # PARAMS
NAVIER-STOKES			
DINo [51]	333k	502k	835k
Ours	354k	531k	885k
SHALLOW-WATER			
DINo [51]	639k	902k	1.5M
Ours	356k	533k	889k
INTERNALLY-HEATED CONVECTION			
DINo [51]	639k	702k	1.3M
Ours	356k	533k	889k

Table 8: Parameter count and runtimes for models used in Navier-Stokes experiments. Note that training runtimes are measured per epoch, inference runtimes are measured for unrolling a single 20-step trajectory. GPU memory allocations are measured per trajectory.

	# PARAMS	TRAINING		INFERENCE	
		RUNTIME/EP	GPU MEM	RUNTIME/TRAJ	GPU MEM
FNO	541k	40.7s	0.35Gb	54.0ms	0.20Gb
GFNO	3.8M	200.6s	2.08Gb	423.1ms	1.28Gb
CNODE	1.8M	1480.0s	1.45Gb	170.4ms	0.71Gb
DINo [51]	835k	25.8s	0.23Gb	61.7ms	0.14Gb
Transolver [50]	7.1M	619.2s	9.42Gb	43.6ms	5.19Gb
Ours	885k	70.0s	0.70Gb	9.5ms	0.61Gb

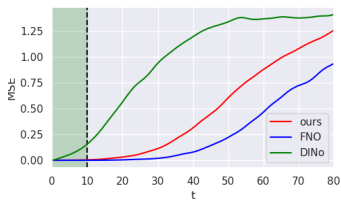


Figure 9: Error accumulation over long rollouts for Navier-Stokes test set with ν_0 100% observed, train horizon t_{in} is marked in green.

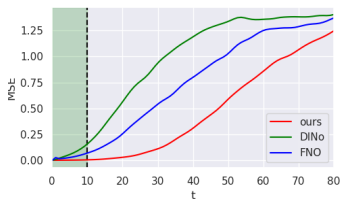


Figure 10: Error accumulation over long rollouts for Navier-Stokes test set with ν_0 50% observed, train horizon t_{in} is marked in green.

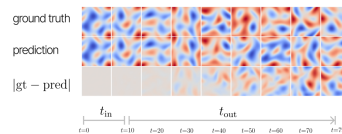


Figure 11: Visualization of an 80-step rollout for a Navier-Stokes test sample.

Error accumulation on long rollouts for Navier-Stokes In order to better empirically assess error accumulation of our method for long-term extrapolation, we provide experimental results for unrolling of Navier-Stokes for 80 timesteps. We apply our model both in a setting with fully observed initial conditions and sparsely observed (50%) initial conditions, and provide plots of accumulated MSE along these trajectories in Figs. 9,10, and a visualization of the solution and error in Fig. 11. We compare against DINo [51] and FNO [29], and show that we consistently improve over DINo in terms of long-term extrapolation. FNO achieves better extrapolation limits due to reduced error accumulation in the fully observed setting, but very rapidly deteriorates even within the train horizon with sparsely observed initial conditions, whereas our proposed approach loses very little in terms of extrapolation performance in this sparse setting.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope?

Answer: [Yes]

Justification: We feel the abstract and introduction accurately reflect the contributions made in the paper, the experiments in Sec. 4 clearly show that our framework achieves improved performance exactly because of the equivariance constraints.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: We discuss some limitations of the model’s performance, namely the inability to generalize well on t_{out} for the global shallow-water equations, and our hypothesis for the cause of this; namely limited reconstruction accuracy of the decoder on complex states ν .

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren’t acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory Assumptions and Proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The main claim of the paper; that we obtain equivariance by weight-sharing over bi-invariances is proven in Eq. 4.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental Result Reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The results should be reproducible based only on the contents of this paper and the references made in it. Code will be made available for a camera-ready version, which also includes code for generating the datasets of solutions and experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Code is added to the submission.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.

- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental Setting/Details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: These details are listed in Appx. E.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment Statistical Significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: std is given for experimental results over 3 runs.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments Compute Resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We provide details on the hardware on which we run the experiments, the approximate runtime per run. This should be enough information to reproduce.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code Of Ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: We respect the code of ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader Impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: The paper provides a way for improving PDE forecasting. This can have many positive societal impacts, e.g. improving weather forecasting for early warning, material sciences. However, the framework could provide erroneous predictions, without any indication of its confidence, which could have major negative consequences. Future work should look into associating confidence intervals to predictions, to get indications of the viability of results.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: We want the paper to be reproducible and as such we provide code. There are no major risks for misuse.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: Authors are original owners, or credit relevant owners as required.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. New Assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: Paper does not provide new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. Crowdsourcing and Research with Human Subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: Not applicable.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: Not applicable.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.