SINR: Equivariant Neural Vector Fields

David Ruhe [1 2 3] **Patrick Forré** [1 2]

Abstract

Several works have in recent years applied equivariant neural fields in, e.g., computer vision and scientific machine learning. However, those works usually restricted themselves to predicting scalar quantities and considered only a single type of symmetry. We study Steerable Implicit Neural Representations (SINRs) from a more general perspective, accommodating scalar, vector, or even multivector or tensor fields while considering equivariances with respect to translations, scaling, rotations, and reflections (or subgroups of these). Empirically, we explore promising use cases such as estimating vector fields, optical flow fields in 2D vision, or surface normals in 3D vision. In the 3D vision case, we furthermore apply a new way to obtain occupancy grids through a classification task, avoiding the typical expensive preprocessing steps required to obtain ground truth occupancy values. Our work highlights the versatility and potential of SINRs in advancing these domains.

1. Introduction

Implicit neural representations, also known as *neural fields* (NeFs) or coordinate-based neural networks, have made tremendous impact in several scientific domains. In computer vision, for example, they have advanced neural representation learning (Park et al., 2019; Mescheder et al., 2019; Mildenhall et al., 2022), differentiable rendering (Sitzmann et al., 2020; Lombardi et al., 2019), flow field representations (Li et al., 2021b), texture fields (Oechsle et al., 2019), and generative modeling (Zhang et al., 2023). In scientific deep learning, the seminal work by Raissi et al. (2019) caused a surge of interest in learning solutions to differential equations using neural networks directly. Other works in this direction include Wang (2021); Chen et al. (2023); Yin et al. (2023); Kawano et al. (2021). Neural fields also influence domains such as medical imaging (Molaei et al., 2023) and compression (Strümpler et al., 2022). Recently, neural fields have been used to implicitly represent and generate geometric surfaces (Berzins et al., 2024).

Related work Relatively little work has been conducted towards making these implicit neural representations agnostic to a chosen coordinate system or reference frame. An early high-impact work in computer vision is the neural descriptor field (Simeonov et al., 2022b;a). Later works include Chen et al. (2022); Wu et al. (2023). Recently, Knigge et al. (2024a); Wessels et al. (2024); Knigge et al. (2024b) propose equivariant neural fields for, e.g., PDE solutions. These works focus either on (1) interpreting the latent variable as a *latent point cloud*, (2) focus on 3D vision or PDE solving tasks, (3) specialize to a single type of symmetry (e.g., rotations), or (4) include only on scalar fields. This work presents a general theory of equivariant neural fields that builds on these works while addressing a broader spectrum of symmetries and fields.

Contributions In this work, we formalize equivariant neural fields as Steerable Implicit Neural Representations (SINR) in a general setting. That is, we start from the general transformation properties of fields, and then define how one can obtain these properties after conditioning on a latent variable. Specifically, we show that a neural network (MLP) becomes steerable if it is equivariant with respect to both the coordinate transformation and the latent variable. This framework accommodates scalar, vector, or even multivector or tensor fields while supporting equivariances with respect to translations, scaling, rotations, and reflections. This is in contrast to all previous works, which have focused solely on scalar fields, which have trivial output representations. Furthermore, those methods usually restrict to a single type of symmetry (e.g., rotations), while we consider multiple symmetries simultaneously.

SINRs are promising in several settings, which we explore empirically. First, the latent variable z can be considered as a chosen basis, which can then at test time be used to *steer* the output of the neural field. I.e., we can rotate, scale, or translate the output of the neural field simply by modifying

[*]Equal contribution [1]AMLab, Amsterdam, Netherlands [2]AI4Science Lab, Amsterdam, Netherlands [3]Anton Pannekoek Institute, Amsterdam, Netherlands. Correspondence to: David Ruhe <david.ruhe@gmail.com>.

Accepted as an extended abstract for the Geometry-grounded Representation Learning and Generative Modeling Workshop at the 41st International Conference on Machine Learning, ICML 2024, Vienna, Austria. Copyright 2024 by the author(s).

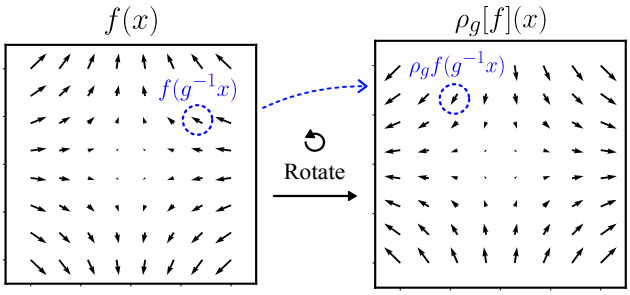

Figure 1. Visualization of a vector field under a rotation. We see that the response at x in the rotated field (highlighted with blue) can be computed by looking up the original vector at the inversely rotated coordinate $g^{-1}x$, and then applying the forward rotation ρ_g to the vector.

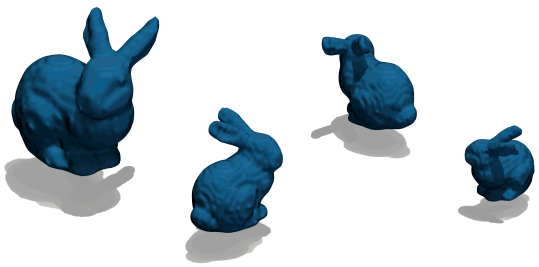

Figure 2. Synthesizing and composing scenes using several orientations, dilations, and translations of the latent variable z.

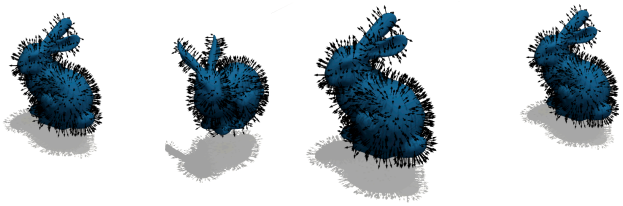

Figure 3. Visualization of the normal field under rotations, dilations, and translations of the latent variable z. Note that the normals also transform correctly.

the latent variable, which can be advantageous for efficient scene synthesis and composition. Second, the latent variable can be used to condition the output of the neural field on some other information. If done equivariantly, the neural field will respect the orientation of the latent variable. For example, the neural field can represent a solution to a PDE, conditioned on its initial conditions, boundary conditions, and so on, while respecting the symmetries of the PDE (Knigge et al., 2024b).

We explore these ideas in several use cases, such as approximating a synthetic flow, optical flow fields in 2D vision, or surface normals in 3D vision. In the 3D vision case, we furthermore apply a new way to obtain occupancy grids through a classification task, avoiding the typical expensive preprocessing steps required to obtain ground truth occupancy values. A notable advantage of our approach is that when z is encoded from e.g. point cloud or graph data, the equivariance is exact and does not suffer from gridding artifacts. Furthermore, since the neural field can be evaluated at any point in space, it can perform zero-shot super-resolution.

2. Method

2.1. SINR

In the following, we follow Weiler et al. (2018); Weiler & Cesa (2019). Let $\mathcal{F}$ be a set of smooth functions $f : \mathcal{X} \to \mathcal{Y}$, where we set $\mathcal{X} := \mathbb{R}^d$ and $\mathcal{Y} := \mathbb{R}^m$. As such, $f \in \mathcal{F}$ maps a d-dimensional coordinate to an m-dimensional output space (e.g. a vector field). We further assume that the output can be decomposed into several geometric quantities (e.g., scalars, vectors, tensors, multivectors). Let G be a specific fixed symmetry group acting on both input and output spaces such that for $g \in G$, $x \in \mathcal{X}$, and $y \in \mathcal{Y}$, we have $x \mapsto gx$ and $y \mapsto \rho_g^{\mathcal{Y}} y$. As a result, $\mathcal{F}$ also carries a G-action $\rho^{\mathcal{F}}$ defined as

$$\rho_g^{\mathcal{F}}[f](x) := \rho_g^{\mathcal{Y}} f\left(g^{-1}x\right) . \tag{1}$$

The intuition behind this definition becomes clear by studying, e.g., Weiler et al. (2018) (Figure 1). For completeness, we also include a visualization in Figure 1. Note that previous methods focused mainly on scalar fields, which have trivial (i.e., $\rho_g^{\mathcal{Y}} = \mathrm{id}$) representations.

We now introduce a map $h : \mathcal{Z} \to \mathcal{F}$ that maps a latent variable $z \in \mathcal{Z} := \mathbb{R}^n$ to a function $f \in \mathcal{F}$. $\mathcal{Z}$ also has a G-representation $\rho^{\mathcal{Z}}$ and can be thought of as an encoding of some geometric information (e.g., a point cloud) or a chosen basis. After parameterization, we efficiently share parameters across many different fields in an amortized way; each z producing a new f. We would like h to respect G-actions on z by making it equivariant, i.e., acting on z should induce Equation (1).

$$h\left(\rho_g^{\mathcal{Z}} z\right) = \rho_g^{\mathcal{F}}[h(z)] . \tag{2}$$

This then allows one to *steer* the resulting field through transforming z, leading to a *steerable implicit neural representation* (SINR).

In practice, mapping to the infinite-dimensional $\mathcal{F}$ is infeasible, so we *uncurry* the map h to receive a function $f : \mathcal{X} \times \mathcal{Z} \to \mathcal{Y}$ (overloading the notation a bit) with

$$f(x; z) := h(z)(x) . \tag{3}$$

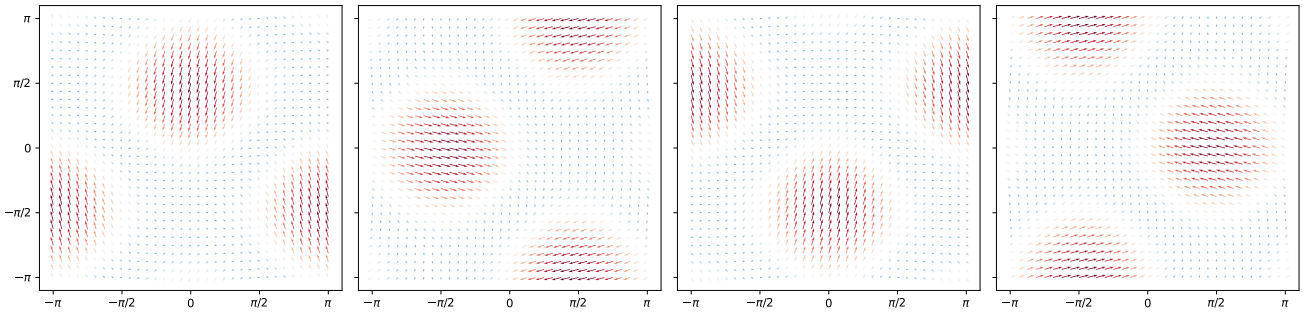

Figure 4. Implicit neural flow field under 90 degree rotations of the latent code z. Both the output grid and the flow vectors are transformed correctly (i.e., also rotated by 90 degrees). The MSE (all orientations) to the ground truth field is approx. 10^{-7} without explicitly training on rotated data.

Figure 5. The in- and output frames of the optical flow task which is to predict the flow field between the two frames.

By the following simple argument,

$$h\left(\rho_g^{\mathcal{Z}} z\right)(x) = f\left(x; \rho_g^{\mathcal{Z}} z\right) \tag{4}$$
$$= f\left(g g^{-1} x; \rho_g^{\mathcal{Z}} z\right) \tag{5}$$
$$\overset{!}{=} \rho_g^{\mathcal{Y}} f\left(g^{-1} x; z\right) \tag{6}$$
$$= \rho_g^{\mathcal{F}}\left[h(z)\right](x)\,, \tag{7}$$

we see that G-equivariance of h is equivalent to f being equivariant (commutes with the G-action) in both arguments. Note that $\mathcal{Z}$ must be sufficiently 'large' to make f maximally expressive. For example, for $G := O(d)$ with $\mathcal{Y} = \mathbb{R}^d$ then z must contain at least d vectors spanning $\mathbb{R}^d$ (Villar et al. (2021); Lemma 3). In addition, one wants to grow the latent space to encode as much features as possible.

2.2. Parameterizations

There are many ways to parameterize SINRs modulo the equivariance constraints with respect to the chosen symmetry group G. Examples of feedforward architectures (that consider various G) include Vector Neurons (Deng et al., 2021), Geometric Vector Perceptrons (Jing et al., 2021), E(3)-NN (Geiger & Smidt, 2022), Clifford Group Equivariant Neural Networks (Ruhe et al., 2023), E-MLP (Finzi et al., 2021), PONITA (Bekkers et al., 2024), and so on. In this explorative work, we keep things relatively simple. We start by considering $m = 1$, i.e., $f : \mathcal{X} \times \mathcal{Z} \to \mathbb{R}$. Let $z = (s, v, t)$ contain scalars, vectors $v \in \mathbb{R}^d$ that transform under $\mathrm{O}(d) \cdot \mathbb{R}_{>0} \subseteq \mathrm{GL}(d)$ and $t \in \mathbb{R}^d$ that transforms under $(\mathrm{O}(d) \cdot \mathbb{R}_{>0}) \ltimes \mathbb{R}^d$: the Euclidean group with scaling.

We can then featurize the input ϕ to our model as follows:

$$\phi(x, z) := \left(\langle \gamma^{-1}(x - t), \gamma^{-1} v\rangle, \|\gamma^{-1}(x - t)\|, s\right)\,, \tag{8}$$

where $\gamma := \|v\|$. Here, $\langle \cdot, \cdot \rangle$ denotes the (inner) product between x and all vectors in z, concatenating the results. In Appendix A we show the equivariance of this parameterization and explicitly show how the group G acts. We put

$$f_\theta(x; z) := \mathrm{MLP}_\theta\left(\phi(x, z)\right)\,, \tag{9}$$

where MLP_θ is a multilayer perceptron with parameters θ. Under this parameterization, we have

$$f_\theta(x; \rho_g^{\mathcal{Z}} z) = f_\theta\left(g^{-1} x; z\right) = \rho_g^{\mathcal{Y}} f_\theta(g^{-1} x; z)\,, \tag{10}$$

satifying the equivariance condition. For vector fields $f : \mathcal{X} \times \mathcal{Z} \to \mathcal{Y}$, where $\mathcal{Y} := \mathbb{R}^d$ only transforms under scaling and rotation, we can set

$$f_\theta(x; z) := \sum_{i=1}^{d} \mathrm{MLP}_\theta^i(\phi(x, z))\; v_i \tag{11}$$

We then have

$$f_\theta(x; \rho_g^{\mathcal{Z}} z) = \sum_{i=1}^{d} \mathrm{MLP}_\theta^i(\phi(g^{-1} x, z))\; \rho_g^{\mathcal{Z}} v_i \tag{12}$$
$$= \rho_g^{\mathcal{Y}} \sum_{i=1}^{d} \mathrm{MLP}_\theta^i(\phi(g^{-1} x, z))\; v_i \tag{13}$$

where MLP_θ^i is scalar-valued and we assume that v contains at least d linearly independent vectors. Using E(3)-NN (Geiger & Smidt, 2022) or Clifford Group Equivariant Neural Networks (Ruhe et al., 2023) we can extend this to higher-order fields.

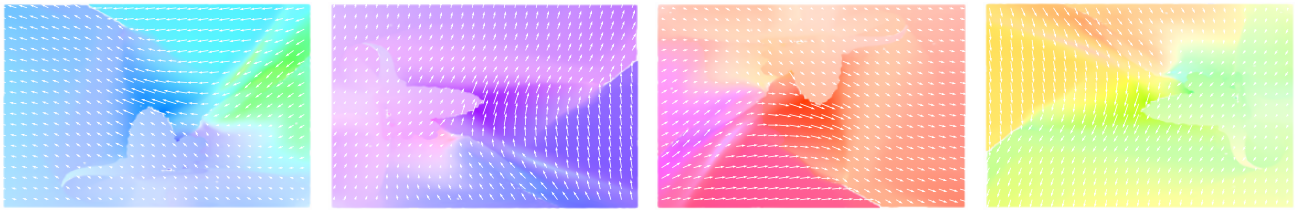

Figure 6. Visualization of the neural optical flow field between the frames shown in Figure 5 when the input frame is rotated by 90 degrees. The flow field is transformed correctly, i.e., also rotated by 90 degrees. The grid and flow vectors are orientation-independent. MSE with ground truth is in all orientations approx. 10^{-3} without explicitly training on rotated data.

3. Experiments

3.1. 3D Vision

We consider the task of estimating the surface occupancy and surface normals of a 3D object (Eslami et al., 2018; Chen & Zhang, 2019; Park et al., 2019). In particular, we focus on the Stanford Bunny mesh (Turk & Levoy, 1994). We then train a SINR to predict the surface occupancy and surface normals of the bunny. I.e., given a point $x \in [-1, 1]^3$, we have the ground-truth indicator

$$\mathbb{I}_{\text{bunny}}(x) = \begin{cases} 1 & \text{if } x \text{ is on the boundary} \\ 0 & \text{otherwise} \end{cases} . \tag{14}$$

We sample points from the surface of the bunny which can be done very efficiently. We also sample negative points from the unit cube. We sample these at a 50-50 ratio. Using a binary cross-entropy loss, we train the SINR to predict whether a point lies on the surface of the bunny. Note that this is a different task than predicting the occupancy grid, which is a common task in neural rendering. It avoids the need for expensive preprocessing steps to obtain ground truth occupancy values. I.e., sampling the mesh surface and the negative points can be done on-line during training. We also sample the ground truth surface normals directly from the mesh.

At test time, we sample a grid of points in the unit cube and predict the surface occupancy and surface normals. Postprocessing steps can then include mathematical morphology steps such as binary closing, and binary dilation for solidifying the surface. The surface can then be turned into a mesh using, e.g., Marching Cubes. In practice, we so far find that Marching Cubes already produces a good mesh directly from the SINR output (before postprocessing).

We set $z = (v, t)$ contain a (scaled) orthogonal system: $v = (e_1, e_2, e_3) \in \mathrm{O}(3) \cdot \mathbb{R}_{>0} \subseteq \mathrm{GL}(3)$ which is sometimes called the *conformal orthogonal group*. $t \in \mathbb{R}^3$ contains the translation.

At test time, we can then rotate, scale, or translate the output of the neural field simply by modifying the latent variable. We can therefore synthesize scenes with the bunny in different orientations, scales, and positions. See Figure 2.

Normal Estimation We now consider cases where the codomain transformation is not invariant. In particular, we consider the task of estimating the surface normals of the bunny (Ben-Shabat et al., 2019; Li et al., 2023; Wang et al., 2015). When we transform the coordinates, the surface normals also transform. This is depicted in Figure 3. We see that the bunny rotates, translates, and dilates, and the normals also transform correctly.

3.2. Fluid Flow

Recently, neural networks for fluid mechanics have gained significant popularity (e.g., (Brandstetter et al., 2022; Li et al., 2021a)). We consider a simple time-dependent synthetic flow field $u : \mathbb{R}^2 \times T \to \mathbb{R}^2$. Consider Appendix B for the exact equations used. We use as 'latent code' z the initial conditions at (0, 0, 0) with $G := \mathrm{O}(2)$. In more advanced settings, one would encode the initial conditions using a more sophisticated equivariant neural network. We now train a SINR to predict the flow field at a later time t using a MSE loss. In Figure 4, we visualize SINR output at $t = \pi$. We see that SINR is agnostic to the orientation of the initial condition, and generalizes over several orientations.

3.3. Optical Flow

Optical flow is a classic computer vision task that estimates the motion of objects in a scene. In modern computer vision, optical flow is often estimated using deep learning methods (Dosovitskiy et al., 2015; Ilg et al., 2017; Sun et al., 2018). We use the Middlebury dataset (Baker et al., 2011) to train a SINR to predict the optical flow field between two frames. We use the 'Dimetrodon' class which also provides ground truth flow. Given a coordinate $x \in \mathbb{R}^2$, we implicitly model the ground truth flow field $f(x) \in \mathbb{R}^2$.

Given the two frames in Figure 5, we output the flow field shown in Figure 6. We see that the SINR generalizes well over different orientations of the flow field. We use the orien-

tation of the input frame as the latent variable z, estimated by multiplying the pixel intensities with their respective coordinates and then spatially averaging.

4. Discussion

We discussed Steerable Implicit Neural Representations (SINRs) as a general framework for equivariant neural fields. We explored several use cases in computer vision or scientific machine learning. In the future, SINR can be extended in several ways. First and foremost, the conditioning scheme was explorative and simple. Encoding more complex input data into equivariant latent representations is a promising direction. Second, we can apply boundary conditions directly in the neural field, by using e.g. sinusoidal positional encodings or enforcing the boundary conditions in the loss function. Third, we can consider higher-order fields, such as tensors or multivectors. Fourth, one can consider more expressive equivariant neural networks. Fifth, one can consider settings with more complex symmetries, such as spacetime symmetries. Finally, SINRS can be explored in generative modeling, medical imaging, or compression. We hope that this work will inspire future research in these directions.

References

Baker, S., Scharstein, D., Lewis, J. P., Roth, S., Black, M. J., and Szeliski, R. A Database and Evaluation Methodology for Optical Flow. *International Journal of Computer Vision*, 92(1):1–31, 2011.

Bekkers, E., Vadgama, S. P., Hesselink, R. D., Linden, P. A. V. D., and Romero, D. W. Fast, Expressive SE(n) Equivariant Networks through Weight-Sharing in Position-Orientation Space. *ArXiv*, abs/2310.02970, 2024.

Ben-Shabat, Y., Lindenbaum, M., and Fischer, A. NestiNet: Normal Estimation for Unstructured 3d Point Clouds Using Convolutional Neural Networks. In *Computer Vision and Pattern Recognition (CVPR)*, pp. 10112–10120, 2019.

Berzins, A., Radler, A., Sanokowski, S., Hochreiter, S., and Brandstetter, J. Geometry-informed neural networks. *arXiv preprint arXiv:2402.14009*, 2024.

Brandstetter, J., Worrall, D. E., and Welling, M. Message Passing Neural PDE Solvers. In *International Conference on Learning Representations (ICLR)*, 2022.

Chen, H., Wu, R., Grinspun, E., Zheng, C., and Chen, P. Y. Implicit Neural Spatial Representations for Time-dependent PDEs. In *International Conference on Machine Learning (ICML)*, pp. 5162–5177, 2023.

Chen, Y., Fernando, B., Bilen, H., Nießner, M., and Gavves, E. 3d Equivariant Graph Implicit Functions. In *European Conference on Computer Vision (ECCV)*, pp. 485–502, 2022.

Chen, Z. and Zhang, H. Learning Implicit Fields for Generative Shape Modeling. In *Computer Vision and Pattern Recognition (CVPR)*, pp. 5939–5948, 2019.

Deng, C., Litany, O., Duan, Y., Poulenard, A., Tagliasacchi, A., and Guibas, L. J. Vector Neurons: A General Framework for SO(3)-Equivariant Networks. In *IEEE International Conference on Computer Vision (ICCV)*, pp. 12180–12189, 2021.

Dosovitskiy, A., Fischer, P., Ilg, E., Häusser, P., Hazirbas, C., Golkov, V., Smagt, P. v. d., Cremers, D., and Brox, T. Flownet: Learning Optical Flow with Convolutional Networks. In *IEEE International Conference on Computer Vision (ICCV)*, pp. 2758–2766, 2015.

Eslami, S. M. A., Rezende, D. J., Besse, F., Viola, F., Morcos, A. S., Garnelo, M., Ruderman, A., Rusu, A. A., Danihelka, I., Gregor, K., Reichert, D. P., Buesing, L., Weber, T., Vinyals, O., Rosenbaum, D., Rabinowitz, N., King, H., Hillier, C., Botvinick, M., Wierstra, D., Kavukcuoglu, K., and Hassabis, D. Neural scene representation and rendering. *Science*, 360(6394):1204–1210, 2018.

Finzi, M., Welling, M., and Wilson, A. G. A Practical Method for Constructing Equivariant Multilayer Perceptrons for Arbitrary Matrix Groups. In *International Conference on Machine Learning (ICML)*, pp. 3318–3328, 2021.

Geiger, M. and Smidt, T. E. e3nn: Euclidean Neural Networks. *arXiv*, abs/2207.09453, 2022.

Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., and Brox, T. Flownet 2.0: Evolution of Optical Flow Estimation with Deep Networks. In *Computer Vision and Pattern Recognition (CVPR)*, pp. 1647–1655, 2017.

Jing, B., Eismann, S., Suriana, P., Townshend, R. J. L., and Dror, R. O. Learning from Protein Structure with Geometric Vector Perceptrons. In *International Conference on Learning Representations (ICLR)*, 2021.

Kawano, M., Kumagai, W., Sannai, A., Iwasawa, Y., and Matsuo, Y. Group equivariant conditional neural processes. *arXiv preprint arXiv:2102.08759*, 2021.

Knigge, D. M., Wessels, D., Valperga, R., Papa, S., Gavves, S., and Bekkers, E. J. Equivariant Neural Fields For Symmetry Preserving Continous PDE Forecasting. In *ICLR Workshop on AI4DifferentialEquations In Science*, 2024a.

Knigge, D. M., Wessels, D. R., Valperga, R., Papa, S., Sonke, J.-J., Gavves, E., and Bekkers, E. J. Space-time

continuous pde forecasting using equivariant neural fields. *arXiv preprint arXiv:2406.06660*, 2024b.

Li, S., Zhou, J., Ma, B., Liu, Y.-S., and Han, Z. Neaf: Learning Neural Angle Fields for Point Normal Estimation. In *AAAI Conference on Artificial Intelligence (AAAI)*, pp. 1396–1404, 2023.

Li, Z., Kovachki, N. B., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A. M., and Anandkumar, A. Fourier Neural Operator for Parametric Partial Differential Equations. In *International Conference on Learning Representations (ICLR)*, 2021a.

Li, Z., Niklaus, S., Snavely, N., and Wang, O. Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes. In *Computer Vision and Pattern Recognition (CVPR)*, pp. 6498–6508, 2021b.

Lombardi, S., Simon, T., Saragih, J., Schwartz, G., Lehrmann, A., and Sheikh, Y. Neural volumes - learning dynamic renderable volumes from images. *ACM Transactions on Graphics*, 38(4):1–14, 2019.

Mescheder, L. M., Oechsle, M., Niemeyer, M., Nowozin, S., and Geiger, A. Occupancy Networks: Learning 3d Reconstruction in Function Space. In *Computer Vision and Pattern Recognition (CVPR)*, pp. 4460–4470, 2019.

Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., and Ng, R. Nerf - representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2022.

Molaei, A., Aminimehr, A., Tavakoli, A., Kazerouni, A., Azad, B., Azad, R., and Merhof, D. Implicit Neural Representation in Medical Imaging: A Comparative Survey. In *IEEE International Conference on Computer Vision (ICCV) - Workshops*, pp. 2373–2383, 2023.

Oechsle, M., Mescheder, L. M., Niemeyer, M., Strauss, T., and Geiger, A. Texture Fields: Learning Texture Representations in Function Space. In *IEEE International Conference on Computer Vision (ICCV)*, pp. 4530–4539, 2019.

Park, J. J., Florence, P. R., Straub, J., Newcombe, R. A., and Lovegrove, S. Deepsdf: Learning Continuous Signed Distance Functions for Shape Representation. In *Computer Vision and Pattern Recognition (CVPR)*, pp. 165–174, 2019.

Raissi, M., Perdikaris, P., and Karniadakis, G. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

Ruhe, D., Brandstetter, J., and Forré, P. Clifford Group Equivariant Neural Networks. In *Conference on Neural Information Processing Systems (NeurIPS)*, volume abs/2305.11141, 2023.

Simeonov, A., Du, Y., Lin, Y.-C., Garcia, A. R., Kaelbling, L. P., Lozano-Pérez, T., and Agrawal, P. Se(3)-Equivariant Relational Rearrangement with Neural Descriptor Fields. In *Conference on Robot Learning (CoRL)*, pp. 835–846, 2022a.

Simeonov, A., Du, Y., Tagliasacchi, A., Tenenbaum, J. B., Rodriguez, A., Agrawal, P., and Sitzmann, V. Neural Descriptor Fields: Se(3)-Equivariant Object Representations for Manipulation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6394–6400, 2022b.

Sitzmann, V., Martel, J. N. P., Bergman, A. W., Lindell, D. B., and Wetzstein, G. Implicit Neural Representations with Periodic Activation Functions. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

Strümpler, Y., Postels, J., Yang, R., Gool, L. V., and Tombari, F. Implicit Neural Representations for Image Compression. In *European Conference on Computer Vision (ECCV)*, pp. 74–91, 2022.

Sun, D., Yang, X., Liu, M.-Y., and Kautz, J. Pwc-Net: Cnns for Optical Flow Using Pyramid, Warping, and Cost Volume. In *Computer Vision and Pattern Recognition (CVPR)*, pp. 8934–8943, 2018.

Turk, G. and Levoy, M. Zippered polygon meshes from range images. In *International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pp. 311–318, 1994.

Villar, S., Hogg, D. W., Storey-Fisher, K., Yao, W., and Blum-Smith, B. Scalars are universal: Equivariant machine learning, structured like classical physics. In *Conference on Neural Information Processing Systems (NeurIPS)*, pp. 28848–28863, 2021.

Wang, R. Physics-Guided Deep Learning for Dynamical Systems: A survey. *ArXiv*, abs/2107.01272, 2021.

Wang, X., Fouhey, D. F., and Gupta, A. Designing deep networks for surface normal estimation. In *Computer Vision and Pattern Recognition (CVPR)*, pp. 539–547, 2015.

Weiler, M. and Cesa, G. General E(2)-Equivariant Steerable CNNs. In *Neural Information Processing Systems*, pp. 14334–14345, 2019.

Weiler, M., Geiger, M., Welling, M., Boomsma, W., and Cohen, T. 3d Steerable CNNs: Learning Rotationally Equivariant Features in Volumetric Data. In *Conference*

on Neural Information Processing Systems (NeurIPS), pp. 10402–10413, 2018.

Wessels, D., Knigge, D. M., Papa, S., Valperga, R., Vadgama, S., Gavves, S., and Bekkers, E. J. Grounding continuous representations in geometry: Equivariant neural fields. *arXiv preprint arXiv*, 2024.

Wu, R., Tie, C., Du, Y., Zhao, Y., and Dong, H. Leveraging SE(3) Equivariance for Learning 3d Geometric Shape Assembly. In *IEEE International Conference on Computer Vision (ICCV)*, pp. 14265–14274, 2023.

Yin, Y., Kirchmeyer, M., Franceschi, J.-Y., Rakotomamonjy, A., and Gallinari, P. Continuous PDE Dynamics Forecasting with Implicit Neural Representations. In *International Conference on Learning Representations (ICLR)*, 2023.

Zhang, B., Tang, J., Nießner, M., and Wonka, P. 3dshape2vecset: A 3d Shape Representation for Neural Fields and Generative Diffusion Models. *ACM Transactions on Graphics (TOG)*, 42(4):92:1–92:16, 2023.

A. Equivariance Derivation

We show that our parameterization is invariant with respect to the chosen symmetry group $G = (\mathrm{O}(d) \cdot \mathbb{R}_{>0}) \ltimes \mathbb{R}^d$ which contains translations, rotations, reflections, and scaling. Note that it suffices to show for $g \in G$: $\phi(gx, \rho_g^{\mathcal{Z}} z) = \phi(x, z)$, since we can reparameterize $x' := g^{-1}x$ to obtain $\phi(gx', \rho_g^{\mathcal{Z}} z) = \phi(x', z)$. Therefore, $\phi(x, \rho_g^{\mathcal{Z}} z) = \phi(g^{-1}x, z)$, showing the claim.

Let $z = (s, v, t)$ contain scalars, vectors v that transform under $\mathrm{O}(d) \cdot \mathbb{R}_{>0} \subseteq \mathrm{GL}(d)$ and t that transforms under $(\mathrm{O}(d) \cdot \mathbb{R}_{>0}) \ltimes \mathbb{R}^d \subseteq \mathrm{GL}(d)$: the Euclidean group with scaling.

We can then set the featurize the input ϕ to our model as follows

$$\phi(x, z) := \big(\langle \gamma^{-1}(x - t), \gamma^{-1}v\rangle, \|\gamma^{-1}(x - t)\|, s\big) \ , \tag{15}$$

with $\gamma := \|v\|$.

Then, $\rho_g^{\mathcal{Z}} z = (s, \alpha Rv, \alpha Rt + \beta)$ where $R \in \mathrm{O}(d)$, $\alpha \in \mathbb{R}_{>0}$, and $\beta \in \mathbb{R}^d$. As such,

$$\begin{aligned}
\phi(gx, \rho_g^{\mathcal{Z}} z) &= \big(\langle \alpha^{-1}\gamma^{-1}(\alpha Rx + \beta - \alpha Rt - \beta), \alpha^{-1}\gamma^{-1}\alpha Rv\rangle, \|\alpha^{-1}\gamma^{-1}(\alpha Rx + \beta - \alpha Rt - \beta)\|, s\big) && (16)\\
&= \big(\langle \gamma^{-1}\langle R(x - t), \gamma^{-1}Rv\rangle, \|\gamma^{-1}R(x - t)\|, s\big) && (17)\\
&= \big(\langle \gamma^{-1}(x - t), \gamma^{-1}v\rangle, \|\gamma^{-1}(x - t)\|, s\big) && (18)\\
&= \phi(x, z)\,, && (19)
\end{aligned}$$

showing the invariance of the featurization, leading to the equivariance of the model preditions (see main paper).

B. Flow Field Equations

The time-dependent field $u : \mathbb{R}^2 \times T \to \mathbb{R}^2$ is defined as

$$u_1(x, y, t) := \begin{pmatrix} \sin(t)(\sin(x) - \cos(y)) \\ \cos(t)(\cos(x) + \sin(y)) \end{pmatrix} \tag{20}$$

$$u_2(t) := \begin{pmatrix} 0.5\cos(t) \\ 0 \end{pmatrix} \tag{21}$$

$$u(x, y, t) := u_1(x, y, t) + u_2(t) \tag{22}$$

We use as 'latent code' the initial condition $z := (u_1(0, 0, 0), u_2(0))$.