Equivariant Neural Fields for Dark Matter Morphology Classification and Other Applications

Vishal Chandra
chandrav@umich.edu
University of Michigan
Ann Arbor, MI, USA

Alexander Rodríguez
alrodri@umich.edu
University of Michigan
Ann Arbor, MI, USA

1 Introduction

This project is on the topic of equivariant models for geometric understanding and classification. Equivariance is the property that transformations of the input should produce the *same* transformations in the output and/or latents of the model. This is a useful fact when reasoning about 3D. For example, even a young child knows where the parts of an object are when it is translated and rotated through space. This would be a segmentation problem with $SE(3)$ group invariance.

Similarly, by recent model architectures, we can embed this property into neural networks for various scientific and computer vision tasks. Two very common scientific applications of this property are in drug design and fluid dynamics, where 3D space as a medium is more important than the absolute location or orientation of objects. Another interesting application is in classifying gravitational lenses due to dark matter from telescope imagery. Here, we look at two more specific use cases.

It's somewhat intuitive that orientation is not strictly constrained in deep space, so that the dark matter whose gravity bends light in astronomical images can be in any variety of positions. To correctly identify the substructure of dark matter, our classification architecture can benefit greatly from being equivariant to rotation and orientation.

As a broader use case, with applications in all areas of science, these models can be used to solve PDEs on irregular domains like spheres. Here again, note that equivariant architectures require an abstract understanding of geometry in the model, free from coordinates. Thus, they can be much better suited to solving complex dynamical systems over irregular domains, as compared to a model which solves with no understanding of the underlying geometry.

On the motivation of this work – my deep learning background comes almost entirely from computer vision, which was the topic of my summer research. However, while there, I learned about equivariant learning for catalyst material screening in 3D. There were other projects ongoing as well about equivariant diffusion for 3D object generation from measurements. I found that the strong inductive priors of geometry made easier many of the inverse problems I am interested in. Thus, I chose to explore it as the topic of my project.

project page: vishalchandra.com/#cse598

2 Survey of Related Work

As noted in the introduction, equivariance has widespread applicability in scientific and computing domains. Here, we review recent literature specifically focused on applications of equivariant methods to 3D detection and reconstruction tasks. While this is the motivation for this survey, and for many of the works included, many other applications arise due to the properties of these methods. However, the commonality between these works remains in their approaches to novel-view synthesis and reasoning of 3D objects from incomplete measurements (often, 2D inputs).

2.1 Equivariance & Pioneering Work

Figure 1 shows a simple diagram known as an "equivariance map". Let the input mesh be called $\vec{x}$. This map simply states that

$$g(T^{\mathcal{Z}}(\vec{x})) = T^{\mathcal{X}}(g(\vec{x}))$$

Although $T^{\mathcal{Z}}$ and $T^{\mathcal{X}}$ are different operators in different domains, it is clear that they have the same qualitative effect on the object. Then, we can restate this more simply: rotating before or after the process g has the same effect. In this case, g is the rendering process. Since rotations and translations form mathematically groups, we often formally call these properties "special orthogonal" or "special euclidean" group equivariant, denoted as $SO(n)$ and $SE(n)$, respectively, where n is the dimensionality.

Initial work integrating group equivariance into deep learning was done by Cohen and Welling in [5]. This seminal work integrated $SO(2)$ and $SE(2)$ equivariance into convolutional neural networks (CNNs) for image classification tasks. This was done by implementing "G-convolutions" in place of ordinary ones to make the network more expressive with no increase to the number of parameters. The G-convolution is defined as

$$[f * \psi](g) = \sum_{y \in \mathbb{Z}^2} \sum_{k} f_k(y) \psi_k(g^{-1}y) \tag{1}$$

where the traditional "flip" of a convolution has been replaced by a generalized group operation. The proof of this expression as a function of only g (and hence its equivariance w.r.t. g) is discussed at length in the original paper.

Cohen and Welling showed the effectiveness of the "GCNN" on rotated verisons of the standard CIFAR10 and MNIST datasets. This sets the stage for equivariant learning across problem disciplines and across symmetry groups.

2.2 Equivariant Rendering

Some of the early works in 3D equivariant learning focus on neural rendering, as shown in the example in fig. 1. This seems like a natural problem, where a neural network responsible for rendering a body should render the exact transforms applied to the input. [6] takes this idea and bends it slightly – instead of changing the object rotation, the rendering should also be equivariant to camera perspective in the rendered scene. Object orientation and camera perspective are effectively equivalent for the rendered image. This is used as a *geometric prior* to the neural renderer to allow it to

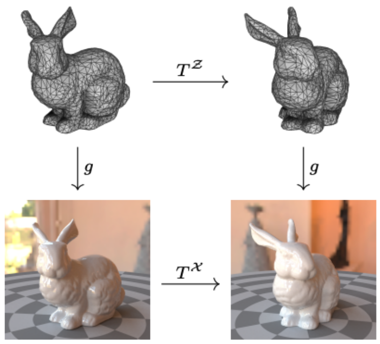

Figure 1: Rabbit Equivariant Map, taken from [6]

construct 3D from a single input image. Essentially, Dupont et al. argue that there is no need for a rendering to take on an explicit representation, so long as they transform in the same way (maintain equivariance to camera view). This is formalized as follows: let $f : X \to L, g : L \to X$ be an encoder and decoder between the image space X and the latent space L. If two input images $\mathbf{x}_1, \mathbf{x}_2$ are rotations of each other, then so should be their corresponding latents $\mathbf{z}_1 = f(\mathbf{x}_1), \mathbf{z}_1 = f(\mathbf{x}_2)$. To enforce this and achieve equivariance, we rotate both latents by the required amount and obtain $\widetilde{\mathbf{z}}_1, \widetilde{\mathbf{z}}_2$. In an equivariant network, decoding these rotated latents should yield the alternate image from which it was produced. As a loss:

$$\mathcal{L}_{\text{render}} = ||\mathbf{x}_1 - g(\widetilde{\mathbf{z}}_2)|| + ||\mathbf{x}_2 - g(\widetilde{\mathbf{z}}_1)||$$

Using this framework, [6] achieves real time inference and rendering of scenes from a single image. As a part of rendering, novel view synthesis is also possible under this equivariant architecture.

2.3 3D Reasoning from 2D

The novel view synthesis abilities of the neural renderer in [6] somewhat beg the question of whether 3D rendering can be benefited by equivariance architectures as well. Without geometric priors, constructing full shape estimates from few- or single-image inputs is nearly impossible, unless datasets are on a large enough scale to effectively learn this information implicitly. Klee et al. [10] introduce a method for $SO(3)$ object recognition from single-view inputs. Since the input is 2D, existing $SO(3)$ methods cannot be be applied because the group action g is not defined. To get around this, the work focuses on on the largest subgroup $I_{60} \subset SO(3)$, and is thus only approximately equivariant. The method uses a ResNet-style encoder to generate dense features from an input image, and then projects the features onto the vertices of an icosaheron. Then, the method applies a group convolution in the icosahedral domain, where the signal is learnable and the projected features act as the filter. This is a very similar formulation as in eq. 1, just with a realized group and group action. This leads to an approximation of SO(3) equivariance that allows reasoning in 3D. Surprisingly, the authors choose to compare their method against baselines with full 3D point cloud inputs, and still find that the reasoning enabled by this method leads to better performance, despite the much more constrained input modality.

2.4 Equivariant Transformers

Introduction of SE(3) Transformers The methods described in previous sections all revolve around clever loss design and embedding equivariance into more traditional neural architectures such as fully-connected networks and convolutional nets. However, with the rise of attention mechanisms and the success of transformers, it is reasonable to make these methods robust and aware of symmetry as well.

In their seminal work [8], Fuchs et al. present the SE(3)-equivariant transformer. This work builds on the work of Thomas et al. in [15] to build equivariance into the standard transformer architecture. The main contribution of this work is in developing a new, equivariant method of calculating attention weights. To do this, the authors combine features of Clebsh-Gordon coefficients, radial neural networks, and spherical harmonics into the computation. The emphasis is, as expected, on aggregating spherical and angular-invariant features into the attention given to a 3D point's neighbors. This makes intuitive sense, as if the whole coordinate frame shifts, a point should not change the weight it assigns to its neighbors. The authors also make an interesting note to relate the new $SO(3)$-equivariant method to prior work: with no weights at all, the attention mechanism degenerates to the same tensor convolution presented in [15]; with weights independent of spatial coordinates, it degenerates to the conventional attention mechanism.

The authors test the $SE(3)$-transformer in three tasks where equivariance is desireable – trajectory prediction, object recognition, and chemical property prediction. They find that it outperforms non-equivaraint methods in all three, though they do not seem to use any previous equivariant models as benchmarks.

Query-Based Transformers for Viewpoint Equivariance Building off of the work of [8], Chen et al. [4] introduce a viewpoint-equivariant transformer. This somewhat mirrors the interplay between object rotation and camera viewpoint in [6], but this is more complex considering the camera is not centered on any one object in the scene. Alongside [10], this is the second presented paper to deviate from now-standard $SO(3)$ and $SE(3)$ equivariances.

The novel contribution of this work is threefold: a geometry-based positional encoding as an input to the transformer, a point- and view-based query system to the equivariant transform (essentially, view-conditioning the transformer output), and a novel viewpoint equivariant loss. More specifically, the viewpoint loss is based on a Hungarian matching (assignment problem matching) between ground truth and predicted bounding boxes across viewpoints.

2.5 Equivariant Transformers in Alternative Domains

Unified Reconstruction & Rendering As there seems to be significant overlap between the tasks in the previous two sections, we examine a work that unifies the two tasks under a single method. Intuitively, rendering from few images and reconstructing shape seem to be very related tasks, if not the same in terms of geometric understanding. [19] extends equivariance to ray space, the space of light rays moving within the scene. In this domain, the Xu et al. construct a "light field" that is equivariant to coordinate system from multiple views and only relative data about camera positions. To do this they introduce new definitions for both convolution and attention in the ray space, and use to construct the equivariant light field transformer. In the work's experiments, the method is shown to be effective in both equivariant neural rendering and equivariant reconstruction. In practice, the mechanisms are slightly different because 3D reconstruction requires point sampling and a conversion from ray space to $\mathbb{R}^3$, and only employs cross attention between point and ray domains before producing an SDF. In contrast, equivariant rendering does not require this discretization until the final image is produced, and so can remain almost entirely in the ray space. After a ray space convolution, present in both pipelines, the rendering process employs a ray-to-ray equivariant transformer before transforming to the pixel domain.

Occupancy Field Transformer As a small addition to this section, it is also worth mentioning TF-ONet by Chatzipantazis et al. [3]. This network models 3D objects by their "occupancy field," or probability distribution over space. This recent work achieves better 3D reconstruction results than many previous methods, in part due to the novel approach of embedding local shape features and maintaining equivariance with respect to them. This is mainly included as another example of equivariance in an alternative domain.

2.6 Equivariant Neural Fields

Neural Fields [18] provides a detailed survey about neural field representations in computer vision and in other domains. From [18]: a *field* is a quantity that takes on a value at every position in space or time. For example, a sound is a field with quantity amplitude. An image is a field with quantity intensity. Many other quantities can be represented by fields, and *neural fields* (NFs) are those that are parameterized by a neural network. These can be written as

$$f_\theta : \mathbb{R}^d \to \mathbb{R}^c$$

Conditional Neural Fields Extending NFs, we can take a collection of neural fields $\mathcal{D}$, and represent each with a latent z_i by fitting a conditional model (CNF):

$$\mathcal{D} = \{f_i : \mathbb{R}^d \to \mathbb{R}^c\}_{i=1}^N$$
$$\forall x : f_i(x) \approx f_\theta(x; z_i)$$

As mentioned by [18], this enables novel approaches to solving tasks involving neural field tasks such as segmentation and classification through *learning with functa* [7]. What [7] proposes is meta-learning – learning on the set of learned latents that represent input data.

Equivariant Neural Fields The work by Wessels et al. in [17] extends CNFs to geometric reasoning by introducing equivariance. They call this steerability of the field – a common term meaning that rotating the input allows the output to be "steered". Specifically for ENFs, the authors impose steerability between the latents and the fields, so if one rotates, so does the other. The implementation of this is surprisingly simple. In large part, the authors look for a maximally informative and invariant replacement for geometric pairs (x_m, p_i), representing the coordinate and pose between two points. Maximally informative means the descriptor is unique for each ordered pair so long as they are not the same up to a group action. In symbols,

$$a(x_m, p_i) = a(x'_m, p'_i) \leftrightarrow \exists g \in G : gx_m = x'_m, gp_i = gp'_i$$

It turns out that $a(x_m, p_i) = p_i^{-1} x_m$ is a perfect candidate, since p_i is the group action g. [17] systematically replaces these ordered pairs with this invariant, leading to equivariance. Like some CNF architectures, ENFs are also based on spatial attention mechanisms and so fall under transformer methods. Because of the widespread applicability of neural fields, [17] performs a wide range of experiments from flood map segmentation to climate forecasting. The authors found that on geometrically symmetric tasks, ENFs were the state of the art method. ENFs also outperform [7] on some tasks involving complex geometry, but neither can beat tailor-made models.

ENFs for Partial Differential Equations In what appears to be a parallel work from the same group that published [17], there is another work that applies ENF to solving PDEs. In [11], Knigge et al. look at the problem of equivariant PDEs – that is, PDEs which themselves have some steerability. If these are modeled with some network F_ψ, their solutions can be found using neural field methods. Just as we saw how Conditional Neural Fields can model functions across time and space parameterized by a latent z_i, we can think of PDEs as *flows* in the latent space – a set of latents $\{z_i^\nu\}_{i=1}^\tau$ given observations $\{\nu_i\}_{i=1}^\tau$. Formally, the optimization problem is

$$\min_{\theta,\psi,z_\tau} \mathbb{E}_{\nu \in D, x \in \mathcal{X}, t \in [\![T]\!]} \left\| \nu_t(x) - f_\theta(x; z_t^\nu) \right\|_2^2, \tag{2}$$

$$\text{where} \quad z_t^\nu = z_0^\nu + \int_0^t F_\psi(z_\tau^\nu) d\tau \tag{3}$$

with $f_\theta(x; z_t^\nu)$ decodes ν_t from latent z_t^ν and F_ψ 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. Much of this paragraph is adapted from [11], referencing work from the pre-equivariance work [20]. Recall that ENFs extend from CNFs using the inverse of the group action p_i, so in solving PDEs, the method similarly looks at geometric pairs appearing in the implementation of f_θ and replaces them with the invariant $a(x_m, p_i)$.

As experiments, the authors evaluate PDEs on irregular domains, such as the heat equation on S^2 (the surface of a sphere), Navier-Stokes on $\mathbb{T}2$ (the 2-dimensional torus), and convection in a 3D ball. They also test superresolution on S^2. The method achieves state of the art on all tasks. Notably, they also generate a dataset specifically to test equivariance by adding a pulse to the initial conditions – any method that does not exhibit equivariance will fail to produce an accurate solution. This is a great resource for future equivariant PDE methods.

3 Proposed Directions & Problem Definitions

(1) **PI-ENF: Physics-Informed Equivariant Neural Field**. This idea borrows from the advancements in PINN, PINO, and PIDeepONet over their non physics-informed counterparts. As we saw in the workshop presentation of the ENF architecture, it originally emphasized solving equivariant differential equations – those equations with operators that are

equivariant to rotations and translations. However, the authors deliberately use only a single loss term to encourance smooth implicit latents in the model. It would be interesting to investigate how a physics-based loss might affect performance here. Would it improve performance on these equivariant processes? Would it impact the latent characteristics that the authors originally controlled for?

(2) **ENFs for Gravitational Lensing** In the spirit of applying the method to a more concrete scientific problem, rather than to general object detection or solving equations, it may be interesting to epxeriment with ENFs for gravitational lensing applications. Cheeramvelil et al. [9] showed the performance gains in applying equivariance to dark matter lensing problems, but used earlier architectures such as steerable CNNs and equivariant transformers. It will be very interesting to see if more sophisticated recent methods discover more areas of gravitational lensing in telescope data. Of course, faraway galaxy-scale images come in only one angle, so it is quite possible that earlier 2D equivariance architectures perform quite well in comparison to these 3D-focused methods.

(3) **VENF: Viewpoint Equivariant Neural Field for 3D Object Detection**. This idea combines [4] and [17] to apply ENFs similarly for robotics. The goal is to have the 3D output detections parameterized by the viewpoint, but to achieve this through the ENF framework rather than through a typical transformer architecture as in the original VEDet paper.

Of the three proposed research directions, it seems that (3) will be the most challenging to implement because of the complexity in [4]. To adapt each of those innovations to the domain of neural fields may be difficult, especially because ENFs are also based on transformers. Discerning the differences in paradigms may be difficult given the similar architecture. In contrast, (1) and (2) build significantly on two or more prior works, and the advancement is in combining those architectural pieces. For the following proposal details, we will address (1) and (2) and leave (3) as future work.

One advantage of all three proposed directions is that they seek to improve performance on existing tasks, which often already have established metrics and datasets. This gives some support to the research process.

4 Methods

4.1 Physics-Based Loss for ENFs

The intuition behind this method, which addresses proposed direction (1), is the same as many other physics-informed methods. Simply, if the current ENF system uses only data-based reconstruction loss to train the latents. In the case where we are trying to solve PDEs with this method, including a dynamics based loss will almost surely improve the solution accuracy by including that information in the structure of the learned latent space.

[14] explores the relationship between "true" and "latent" dynamics, like the integral equation 3, in greater detail than [11].

Suppose there is an autoencoder that relates input data $\mathbf{x} \in \mathcal{X}$ and latent representation $\mathbf{z} \in \mathcal{Z}$ by

$$\mathbf{x} = \psi_\theta(\mathbf{z}) = \psi_\theta(\phi_\theta(\mathbf{x})) \approx \psi_\theta(\psi_\theta^{-1}(\mathbf{x}))$$

Similarly, the true dynamics of x and latent dynamics of z are counterparts in the different spaces

$$\frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} = \mathbf{f}(\mathbf{x}(t)) \leftrightarrow \frac{\mathrm{d}\mathbf{z}(t)}{\mathrm{d}t} = \mathbf{h}(\mathbf{z}(t))$$

Then,

$$\mathbf{h}(\mathbf{z}(t)) = \frac{\mathrm{d}\mathbf{z}}{\mathrm{d}t} = \frac{\mathrm{d}\mathbf{z}}{\mathrm{d}\mathbf{x}}\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \nabla\phi_\theta(\mathbf{x}(t))^T\mathbf{f}(\mathbf{x}(t))$$

, allowing us to relate the original forcing function $\mathbf{f}$ to the latent forcing function $\mathbf{h}$ and impose physics based loss analogously to physics-informed neural networks [12].

Some difference in adapting to this method are worth noting. [14] studies latent space dynamics in an autoencoder context, introducing the learned encoder and decoder ϕ_θ and ψ_θ. In a neural fields context, the latents are generated by meta-learning on neural implicit representations to produce a latent-conditioned field $f_\theta(x; z)$, as discussed in 3.6. This is the analog of the encoder function whose Jacobian relates the true and latent forcing functions. Second, for ENFs h is approximated by a neural ODE as shown in eq. 3. It will be interesting to see if using the computed latent forcing function as a replacement for this neural ODE or if using it to impose a physics-based loss on latent collocation points leads to better performance.

It remains to be fully worked out how the gradient $\nabla\phi_\theta(\mathbf{x}(t))$ can be taken in the ENF architecture, which is a crucial piece of work.

4.2 Classification with ENFs

The associated blog post to [17] describes ways classification can be performed from different kinds of Neural Field models. To quote from there:

> Using NeFs in downstream tasks. For "conventional" NeFs, the weights θ_j are used as input to a downstream model that can operate on the computational graph of the neural field. For CNFs, the latent vectors z_j are used as representation instead, allowing the use of simple MLPs. In ENFs instead the latent point sets z_j are used as input to the downstream model, allowing for preservation of geometric information in the downstream task through the use of equivariant graph models.

Thus, to extend the work of [9] with ENFs, we may simultaneously or sequentially train an equivariant graph neural network (EGNN) classifier on the latent point cloud of the ENF model. The EGNN architecture introduced in [13] allows us to carry through the equivariance of the method to the classification, so that the transformation of the input propagates not only through the ENF but also through the equivariant classifier. In many cases, we want the classifier to be *invariant*, a special case of equivariance.

4.2.1 *Choice of Bi-invariant.*

The choice of bi-invariant is a critical piece of training the ENF. Since $SE(2)$ is the desired invariance in the classification problem, the bi-invariant is chosen accordingly. As described in [17]:

$$\mathbf{a}_{m,i}^{SE(2)} = \mathbf{R}_{\theta_i}(x_m - \mathbf{t}_i)$$

This is compared in an experiment using the translational bi-invariant rather than the roto-translational bi-invariant.

4.2.2 Downstream Classifiers. Once the latents are trained using reconstruction loss, a downstream classifier is trained with paired class data to classify from those latents. Since the structure of ENF latents is a point cloud in the field domain, a simple message passing neural network can be used, or, following the authors' tutorial notebook, the $P\Theta NITA$ architecture from [1] can be used for best performance. A later experiment is focused on ablating the perfomance between these two in the gravitational lensing use case.

4.3 Evaluation

Both [11] and [9] use relatively simple evaluation metrics. [11] uses point-wise MSE on the output solutions. [9] uses AUC and accuracy for classification, and RMSE, MSE, MAE to measure performance in regression agains the lensing mass. All metrics can be re-implemented easily.

4.4 Dataset

Both of [9][11] generate their own datasets and describe in detail how to recreate them.

- [11] uses a combination of `py-pde` and Daedalus to set up and solve complex PDEs for training and evaluation. The dataset shape is $(1024+128, 20, 64, 64)$ – 1024+128 (train/test split) trajectories of 20 states each, each state 64x64.
- [9] simulates dark matter gravitational lensing using custom code based on the `lenstronomy` package [2]. This dataset was obtained from the DeepLense Collaboration team [16] and was used in the experiments here.

4.5 Data Augmentation

$SE(2)$ invariance is the desired property for gravitational lensing problems. In order to ensure that these transforms exist in the data, each row of the DeepLense dataset is augmented with random rotation and translation copies, in order to better train the neural field. No augmentations are needed for the PDE dataset.

4.6 Computing Resources

[9] made mention of training on multiple A100 GPUs, which are available at any cloud provider today. There are grounds to believe that neural field-based methods will be significantly more costly to train, because of the neural implicit representations of functions requiring training before the model itself. However, this should also be within the margin of the provided computing resources. Neither ENF paper discusses the compute required.

5 Experiments

Here, we perform one main experiment: training the full ENF pipeline and downstream $P\Theta NITA$ classifier on Model I data used by [9]. This dataset is the base case in that work, without modifications that approximate meant to approximate either a Euclid or Hubble survey. This experiment has three parts: (1) fitting the base neural fields with reconstruction loss, to generate implicit representations of the image dataset, (2) fitting the ENF to generate latent point clouds for each neural field representation, and (3) training the $P\Theta NITA$ MLP as a classifier in the ENF latent space.

We find that step (1) of the experiment, fitting the neural fields, converges very quickly. Though this experiment was run for 30 epochs, the same as the authors of [17] used for STL-10 classification, the MSE only majorly improves over the first 3 epochs to the order of 1e−10 and improves only minorly after than that. The ENF fitting is not modified from default values in [17], given that it is tuned for the NIRs produced by step (1). Step (3) is again much quicker to converge than expected, reaching 100% accuracy in just 3 epochs rather than the prescribed 30 in [17].

The table below is partially reproduced from [9] to show existing methods compared to ENFs.

Model Name	Accuracy
ResNet50	96.86
C8Steerable CNN	99.02
Harmonic Net	90.95
Equivariant Transformer	92.413
Equivariant Neural Field	**100**

This work seems to be the first to reach 100% accuracy on this simulated dataset, and outperforms all those previously compared in [9].

6 Conclusion

Overall, in this work, we take the new equivariant neural field (ENF) architecture and propose three major developments that can be made for applications in the areas of physics, astronomy, and robotics. We pursue the first application in dark matter cosmology, and show that leveraging equivariant methods leads to the better performance, despite the higher computational cost associated with them. Here, we show that the ENF architecture proposed in [17] attains the best possible classification accuracy of dark matter substructure. More generally, in applications where compute is less of a concern than accuracy, as in offline physics and astrophysics, these methods are viable for the best possible results.

7 Future Work

There are three areas for related work:

(1) Ablations of downstream classifiers: it would be interesting to see how other classifiers perform on the generated latents from the ENF method. Given their relatively simple content, it is reasonable to assume that the latents do not contain so much geometric content so as to invalidate simpler MLP approached. It would be interesting to see the performance of methods like MLP and equivariant transformer on this latent dataset.
(2) Extension to other simulated datasets. DeepLense [16] provides 4 different simulated datasets, all tweaked to simulate different cosmological surveys. However, the Euclid survey in [9] shows higher accuracy by all models, so this model may continue to perform at the 100% mark.
(3) Other applications of ENFs are proposed in this report which should be pursued. Physics-informed ENFs are a promising avenue for PDE solving.

References

[1] Bekkers, E. J., Vadgama, S., Hesselink, R. D., van der Linden, P. A., and Romero, D. W. Fast, expressive $\mathrm{se}(n)$ equivariant networks through weight-sharing in position-orientation space, 2024.

[2] Birrer, S., Amara, A., and Refregier, A. Gravitational lens modeling with basis sets. *The Astrophysical Journal 813*, 2 (Nov. 2015), 102.

[3] Chatzipantazis, E., Pertigkiozoglou, S., Dobriban, E., and Daniilidis, K. Se(3)-equivariant attention networks for shape reconstruction in function space, 2023.

[4] Chen, D., Li, J., Guizilini, V., Ambrus, R., and Gaidon, A. Viewpoint equivariance for multi-view 3d object detection, 2023.

[5] Cohen, T. S., and Welling, M. Group equivariant convolutional networks, 2016.

[6] Dupont, E., Bautista, M. A., Colburn, A., Sankar, A., Guestrin, C., Susskind, J., and Shan, Q. Equivariant neural rendering, 2020.

[7] Dupont, E., Kim, H., Eslami, S. M. A., Rezende, D., and Rosenbaum, D. From data to functa: Your data point is a function and you can treat it like one, 2022.

[8] Fuchs, F. B., Worrall, D. E., Fischer, V., and Welling, M. Se(3)-transformers: 3d roto-translation equivariant attention networks, 2020.

[9] Geo Jolly Cheeramvelil, Sergei Gleyzer, M. W. T. Equivariant neural networks for signatures of dark matter morphology in strong lensing data, 2023.

[10] Klee, D., Biza, O., Platt, R., and Walters, R. Image to icosahedral projection for $\mathrm{SO}(3)$ object reasoning from single-view images, 2022.

[11] 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, 2024.

[12] Raissi, M., Perdikaris, P., and Karniadakis, G. E. Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations, 2017.

[13] Satorras, V. G., Hoogeboom, E., and Welling, M. E(n) equivariant graph neural networks, 2022.

[14] Sholokhov, A., Liu, Y., Mansour, H., and Nabi, S. Physics-informed neural ode (pinode): embedding physics into models using collocation points. *Scientific Reports 13*, 1 (June 2023).

[15] Thomas, N., Smidt, T., Kearnes, S., Yang, L., Li, L., Kohlhoff, K., and Riley, P. Tensor field networks: Rotation- and translation-equivariant neural networks for 3d point clouds, 2018.

[16] Toomey, M., and Gleyzer, S. Deeplense collaboration, 2024.

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

[18] Xie, Y., Takikawa, T., Saito, S., Litany, O., Yan, S., Khan, N., Tombari, F., Tompkin, J., Sitzmann, V., and Sridhar, S. Neural fields in visual computing and beyond, 2022.

[19] Xu, Y., Lei, J., and Daniilidis, K. Equivariant light field convolution and transformer, 2023.

[20] Yin, Y., Kirchmeyer, M., Franceschi, J.-Y., Rakotomamonjy, A., and Gallinari, P. Continuous pde dynamics forecasting with implicit neural representations, 2023.