Symmetric Models for Radar Response Modeling *

Colin Kohler KOHLER.C@NORTHEASTERN.EDU
Northeastern University

Nathan Vaska NATHAN.VASKA@LL.MIT.EDU
MIT Lincoln Laboratory

Ramya Muthukrishnan RAMYAMUT@MIT.EDU
Massachusetts Institute of Technology

Whangbong Choi WHANG@LL.MIT.EDU
MIT Lincoln Laboratory

Jung Yeon Park PARK.JUNGY@NORTHEASTERN.EDU
Northeastern University

Justin Goodwin JGOODWIN@LL.MIT.EDU
MIT Lincoln Laboratory

Rajmonda S. Caceres[†] RAJMONDA.CACERES@LL.MIT.EDU
MIT Lincoln Laboratory

Robin Walters[†] R.WALTERS@NORTHEASTERN.EDU
Northeastern University

Editors: Sophia Sanborn, Christian Shewmake, Simone Azeglio, Nina Miolane

Abstract

Many radar applications require complex radar signature models that incorporate characteristics of an object's shape and dynamics as well as sensing effects. Even though high fidelity, first-principles radar simulators are available, they tend to be resource intensive and do not easily support the requirements of agile and large-scale AI development and evaluation frameworks. Deep learning represents an attractive alternative to these numerical methods, but can have large data requirements and limited generalization ability. In this work, we present the Radar Equivariant Model (REM), the first SO(3)-equivariant model for predicting radar responses from object meshes. By constraining our model to the symmetries inherent to radar sensing, REM is able to achieve a high-level reconstruction of signals generated by a first-principles radar model and shows improved performance and sample efficiency.

[†] Equal advising

* Distribution Statement A. Approved for public release. Distribution is unlimited. This material is based upon work supported by the Under Secretary of Defense for Research and Engineering under Air Force Contract No. FA8702-15-D-0001. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Under Secretary of Defense for Research and Engineering.

1. Introduction

Efficient physics-based radar models are needed to simulate complex objects and scene variations for training and testing AI algorithms in data sparse or high consequence radar applications. High-resolution simulators like radar ray tracing models (Andersh et al., 2000) currently exist, but are computationally expensive to run as part of common AI engineering and evaluation workflows. Other approaches rely on domain expertise to carefully design and adapt object scatter locations or component representations (Jackson et al., 2008). However, because such approaches are human intensive, they do not scale well.

While machine learning methods represent an attractive alternative to these traditional methods, radar response data has several unique characteristics that make it challenging for existing machine learning methods to model. It exhibits spatio-temporal symmetries resulting from varying object shapes and the physical laws governing both the object dynamics and radar itself. Additionally, radar data is extremely sensitive to multi-path reflections, interference, ghost objects, clutter, and attenuation increasing the amount of variation in the data and introducing discontinuities.Current machine learning methods for reconstructing radar responses are limited and do not explicitly model the inherent geometric and physical symmetries of objects and scenes, i.e. rotational and scaling symmetries (Wheeler et al., 2017; Ouabi et al., 2021). Equivariant deep learning has emerged as a class of algorithms that can address some of these challenges by directly encoding symmetry and other physical priors into neural networks (Wang et al., 2021; Bronstein et al., 2021; Bogatskiy et al., 2020; Kohler et al., 2023), but its application to radar-like data has been lacking.

In this work, we present the Radar Equivariant Model (REM), a novel method for predicting the radar response from object meshes. REM compresses the input mesh down to a set of SO(3)-equivariant features using equivariant graph convolutions (GCN). Then, a series of spherical convolutions are performed on these features to decode this latent space onto the 2-sphere, resulting in a signal that is equivariant to 3D rotations. By evaluating this signal at the point on the 2-sphere specified by the radar line-of-sight, we can extract the radar response for a specific mesh and orientation. In this way, our method can be trained to simultaneously predict the full radar response for a given mesh in a fraction of the time of other methods. Our contributions are as follows. First, we propose REM, the first SO(3)-equivariant architecture for modeling radar signals. REM incorporates radar symmetries using an SO(3)-equivariant GCN encoder and spherical CNN decoder. REM simultaneously outputs the entire radar response over all viewing angles by encoding the response as a signal over the 2-sphere, leading to more geometric structure and efficiency. Second, we empirically demonstrate that our method is able to achieve a high-level reconstruction of signals generated by a first-principles radar model and shows improved performance, sample efficiency, and generalization to unseen viewing angles when compared to baselines. Finally, we present two novel mesh-to-radar datasets, *Basic Shapes* and *Frusta*, each consisting of 2000 meshes and their radar responses.

2. Related Works

Radar Modeling. There are a number of first-principles radar simulation tools of varying fidelity, from full-wave computationally-intense methods solving the integral form of Maxwell's electromagnetic equations for exact currents (Rao et al., 1982; Song et al., 1997; Engheta

et al., 1992; Taflove and Umashankar, 1987), to more simplified models such as physical optics (PO) approximations (Balanis, 2012; Paknys, 2016; Gibson, 2021). However, these methods are too computationally slow for many applications such as self-driving cars and weather monitoring, motivating data-driven solutions. There has been recent interest in using neural networks (Wheeler et al., 2017; Zheng and Wang, 2023) to replicate first-principles radar models efficiently. Wheeler et al. (2017) use a conditional variational autoencoder method to learn the radar response of automotive radars taking as input spatial raster representations of objects and the environment, while Zheng and Wang (2023) use physically-informed neural networks to solve the wave equation for the ground-penetrating radar. Our approach, in addition to focusing on far-field radar models and mesh representations of objects, differs from previous efforts by explicitly encoding symmetries of the radar function in the neural network.

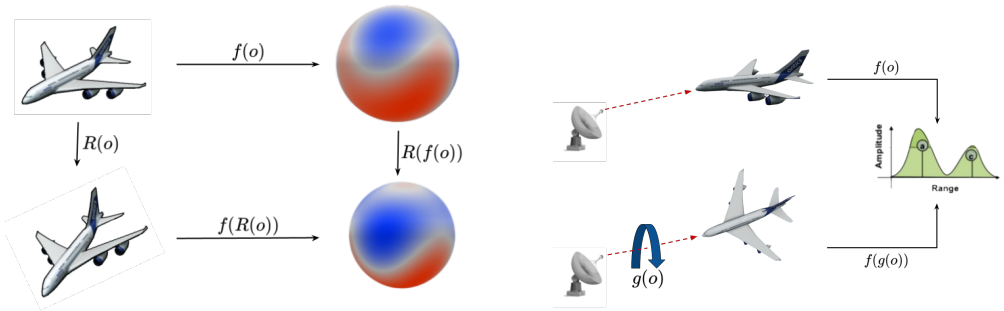
SO(3)-Equivariance. Many neural networks have incorporated equivariance to SO(3) for classifying shapes (Esteves et al., 2018; Cohen et al., 2018), classifying protein structures (Weiler et al., 2018), and predicting features of atomic systems (Thomas et al., 2018; Brandstetter et al., 2021; Liao and Smidt, 2022). Similar to our method, all these approaches use steerable kernel bases and perform equivariant operations such as convolutions or tensor products. However, none of these have been applied to radar modeling. Esteves et al. (2018); Cohen et al. (2018) model data over the sphere. For point cloud data, a commonly used approach is to use equivariant graph convolutions with message passing (Thomas et al., 2018; Fuchs et al., 2020), where the input features are representations of SO(3). Our approach uses a combination of previous methods for the novel application of radar response modeling. Our SO(3) equivariant GNN encoder maps mesh data onto SO(3) and the spherical CNN decoder maps SO(3) features onto the 2-sphere to represent the spherical radar response.

3. Background

Equivariance. A function is equivariant if it respects the symmetries of its input and output spaces. Specifically, a function $f : X \to Y$ is *equivariant* with respect to a symmetry group G, if it commutes with all transformations $g \in G$, $f(\rho_x(g)x) = \rho_y(g)f(x)$, where ρ_x and ρ_y are the *representations* of the group G that define how the group element $g \in G$ acts on $x \in X$ and $y \in Y$, respectively. An equivariant function is a mathematical way of expressing that f is symmetric with respect to G; if we evaluate f for various transformed versions of the same input, we should obtain transformed versions of the same output.

SO(3)-Equivariance. When dealing with physical systems in 3D, since the orientation of the coordinate frame is arbitrary, many task functions f should be equivariant to SO(3), the group of 3D rotations of the coordinates. In order to build neural networks that incorporate this sort of geometric reasoning, it is necessary to parameterize signals $f : \mathcal{X} \to \mathbb{R}$ where $\mathcal{X} = S^2$ or $\mathcal{X} = \mathrm{SO}(3)$ in a way that is both computationally efficient and easy to apply to rotations. Cohen et al. (2018) provide an effective solution using the truncated Fourier basis for signals defined over S^2 in terms of spherical harmonics Y_k^l and Wigner D-matrix coefficients D_{mn}^l. Writing $f : \mathrm{SO}(3) \to \mathbb{R}$ in terms of the D_{mn}^l and then truncating to a given frequency $l \leq L$, we arrive at the approximate representation $f(g) \approx \sum_{l=0}^{L} \sum_{m=0}^{2l+1} \sum_{n=0}^{2l+1} c_{mn}^l D_{mn}^l(g)$. Using this approximation, the SO(3) group convolution can be efficiently computed point-wise

in the Fourier domain using the convolution theorem. See Cohen et al. (2018) for additional details on the SO(3) group convolution.



(*a*) **SO(3)-Equivariance** (*b*) **SO(2)-Invariance**

Figure 1: **Symmetries of Radar Sensing.** Two symmetries of radar sensing that we incorporate into our equivariant model: (a) the SO(3)-equivariant mapping from the 3D mesh to the spherical radar signal, and (b) the invariance of the far field radar response to SO(2) rotations about the radar line-of-sight axis.

4. Symmetries of Radar Sensing

The radar signal prediction problem is a supervised learning task, where the goal is to predict the radar response for a given mesh and radar orientation. The mesh is defined as $o = (V, F)$, where V is the set of vertex coordinates in $\mathbb{R}^3$ and F defines the connectivity, and the radar line-of-sight is given by $e \in S^2$. The goal is to train a network f_ϕ to map $f_\phi \colon (o, e) \mapsto r$, where $r \in \mathbb{R}^{N_r}$ is the radar response signal intensity over range discretized into N_r distances, i.e., the range profile. In this work, we focus solely on far-field radar signal modeling, where the objects are a large distance from the radar origin point and so the radar response at a given time is a single scalar intensity with no spatial resolution.

The radar response for any given object is governed by Maxwell's equations. Invariances in Maxwell's equations imply certain symmetries in the mapping from input mesh o to radar response r, such as spatio-temporal shift or object rotations. We consider two symmetries in this work: (1) the rotational symmetry of the mesh and radar response, and (2) the invariance of the far-field radar response rotated about the radar line-of-sight axis.

The 3D rotational symmetries of the radar response function are illustrated in Figure 1(*a*). When the mesh is transformed by $R \in \mathrm{SO}(3)$, then the radar signal is also transformed by R. That is, the radar response function is SO(3)-equivariant, i.e., $f(R(o)) = R(f(o))$. The symmetry about the radar line-of-sight axis is illustrated in Figure 1(*b*). For the far-field radar signals in this work, the radar response does not have resolution across the plane perpendicular to the line of sight. That is, it does not change as the object rotates about the line-of-sight axis e of the radar. Formally, for some rotation $g \in \mathrm{SO}(2)$, $f(o, g(e)) = f(o)$. By computing the radar response using an SO(3)-equivariant model and encoding the output signal to the sphere $S^2 = \mathrm{SO}(3)/\mathrm{SO}(2)$, we combine the SO(3)-equivariance of the object system and the SO(2)-invariance of the far-field radar response.

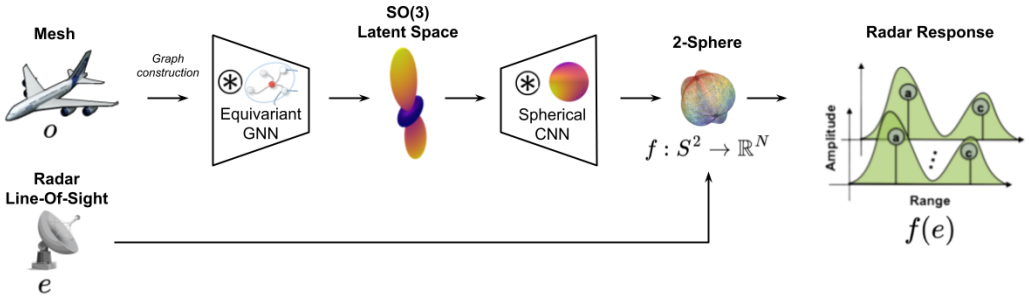


Figure 2: **Radar Equivariant Model.** High-level REM encoder-decoder architecture comprised of an equivariant GNN encoder and a spherical CNN decoder.

5. Method

We propose the Radar Equivariant Model (REM), an encoder-decoder architecture (Figure 2) for predicting the radar response for meshes. Encoder-decoder architectures such as U-Net (Zhou et al., 2018) and Transformers (Vaswani et al., 2017) are attractive methods for prediction tasks with high-dimensional inputs due to latent space compression. By compressing data into a compact, information-rich latent space, these models can efficiently learn representations that are valuable for various tasks. At a high level, REM utilizes an equivariant graph neural network to encode the mesh into an SO(3) latent space and then a spherical convolutional decoder to decompose the signal into spherical harmonics, allowing us to simultaneously predict the radar response for all possible viewing angles. REM utilizes the `e3nn` framework (Geiger et al., 2022; Geiger and Smidt, 2022) to ensure that the symmetries in Section 4 are respected. Additional details can be found in Appendix C.

5.1. Graph Construction

In order to utilize a graph-based encoder, we first must convert the object mesh into a graph with the appropriate node and edge features. The node features are the 3D position $\mathbf{x}_i \in V$, where V is the set of vertex coordinates indexed by i. Similarly to Brandstetter et al. (2021), the edge features are the spherical harmonic embedding of the relative positions $e_{ij} = Y^l(\mathbf{x}_j - \mathbf{x}_i)$, where Y^l: $\mathbb{R}^3 \to \mathbb{R}^{2l+1}$ are the spherical harmonics and i, j are vertices connected by an edge. In this paper, we do not include additional properties of the meshes such as texture, material, or sub-component types, leaving this for future work.

5.2. Encoder

In REM, we first use a geometric graph neural network to encode the local and global information of the mesh into Fourier space, thereby ensuring Euclidean and permutation transformations are correctly represented within our model. To accomplish this, we make use of anisotropic convolutions (Wiersma et al., 2022), which modify the standard graph convolution to take into account the mesh surface geometry. The kernel weights are different for each neighbor, depending on the relative position vector between nodes. For a source node i, the anisotropic convolution is $f_i' = \sigma\left(\sum_{j \in \mathcal{N}(i)} K(x_j - x_i) f_j\right)$, where x_i, x_j are are

the position coordinates for nodes i, j and $\mathcal{N}(i)$ denotes the neighborhood of node i. In order for this operation to be SO(3)-equivariant, the kernel K must satisfy the steerability constraint. We accomplish this by utilizing the `e3nn` library (Geiger et al., 2022), which uses tensor products and Clebsch-Gordan coefficients to construct K.

We build the encoder by stacking several encoding blocks, each of which contains a geometric graph convolution (GGCN) layer followed by a TopK pooling (Gao and Ji, 2019) layer. The pooling layers progressively reduce the size of the graph to encode higher-order features and the GGCN layers aggregate each node's first-order information. Pooling also greatly improves the computation efficiency of REM for large meshes. After each pooling layer, we recompute the spherical harmonics for the edges in the new graph. A final average global pooling layer reduces the entire mesh to a single node and outputs an SO(3)-equivariant feature vector, which is then used as the latent code and passed on to the decoder. This feature vector transforms as a direct sum of various irreducible representations of SO(3), which are components of the Fourier decomposition of signals over SO(3). Thus this feature vector may be viewed as a truncated Fourier space representation of a signal over SO(3).

5.3. Decoder

Once the input mesh has been projected into Fourier space using the encoder, we use operations that preserve the SO(3) symmetry of the latent representation. Specifically, we use SO(3)-equivariant group convolutions. Following Spherical CNN (Cohen et al., 2018), we apply non-linearities between convolutional layers by mapping the signal to the spatial domain, applying ReLU (Agarap, 2018), and then mapping back to the Fourier domain.

The output of the final layer of the decoder $f(o)$ represents an N-channel signal over S^2 in the Fourier domain. That is, $f(o)_n = (c^l_{nm}(o))^{2l+1,L}_{m=0,l=0}$ are the coefficients of the spherical harmonics up to frequency L for $1 \leq n \leq N$. To extract the radar response for a specific radar orientation, we apply the inverse Fourier transform and evaluate these N signals on the 2-sphere, at a position specified by the radar line-of-sight vector e. That is,

$$f(o, e) = \left(\sum_{l=0}^{L} \sum_{m=0}^{2l+1} Y^l_m(e) c^l_{nm}(o) \right)_{n=1}^{N}$$

where o is the input mesh and Y^l_m are the spherical harmonics of degree l.

6. Experiments

We perform a series of experiments to evaluate the performance and generalization of our method. First, we benchmark the performance of REM against several alternative approaches on two mesh-to-radar datasets. Second, we examine REM's ability to generalize from sparse radar response viewing angles to the full set of all possible radar orientations. Finally, we study the effect of the maximum signal frequency used in REM on prediction.

6.1. Datasets

Due to the expensive cost of collecting real-world radar data, we simulate radar responses for a variety of different meshes using a physical optics method to approximate the electromagnetic

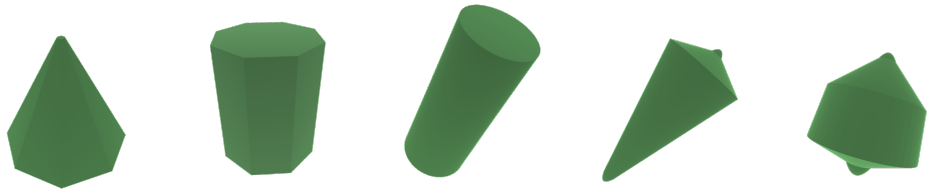


Figure 3: **Datasets.** Example meshes used in our experiments. Meshes (a) and (b) come from the *Basic Shapes* dataset and meshes (d) and (e) are from the *Frusta* dataset.

waves (Balanis, 2012). Additional details can be found in Appendix A. Specifically, we generate two mesh datasets: *Basic Shapes* and *Frusta*. The *Basic Shapes* dataset can be thought of as a collection of building block components that can be combined in various ways to form more complex 3D shapes. We generate 1000 pyramid and 1000 cylinder meshes that are composed of components like flat 2D plates and hemispheres in addition to the underlying shape. We also randomly vary radius and length parameters of the underlying shapes. The *Frusta* dataset represents a collection of complex 3D shapes, where each mesh is a combination of several different basic components stacked together. This dataset also consists of 2000 meshes. Each mesh is comprised of a mid-portion containing between one to three connected frusta of various sizes, and two end-caps that are randomly selected to be either a hemisphere or a flat plate. We also randomly vary the parameters used to generate these components. For both the *Basic Shapes* and the *Frusta* datasets, every mesh is evaluated over 360 orientations on the non-symmetric axes, $\theta \in [0, 2\pi]$, to acquire the associated radar responses. We will refer to this collection of orientations as the static pattern for a given mesh. We split both datasets into 1000 training and 1000 testing samples. Example meshes can be seen in Figure 3.

6.2. Baselines

We compare our method against two competitive baselines for processing meshes, including a non-equivariant graph convolutional network (GNN) and a transformer architecture. One key difference between both baselines and REM, is that REM simultaneously predicts the full radar response whereas the baselines predict a single response and therefor require multiple evaluations to compute the full response.

GNN (Kipf and Welling, 2017). We consider the widely used graph convolution architecture by Kipf and Welling (2017) using a similar graph input as the REM model. This baseline shares a GNN structure with REM, but without SO(3) equivariance. The node features are the concatenation of positions of the vertices and the radar line of sight, $n_i = (\mathbf{x}_i, e)$ for vertex i. The edge features are the length of the edges.

Transformer (Feng et al., 2018). Based on the approach introduced in Feng et al. (2018), we generate spatial and structural descriptors for each triangle in the mesh. We concatenate these representations to form a single embedded representation for each triangle. These

embeddings are combined with the line of sight and encoded with a transformer Vaswani et al. (2017) and decoded with a MLP to generate the predicted response. See Appendix C for additional baseline models and training details.

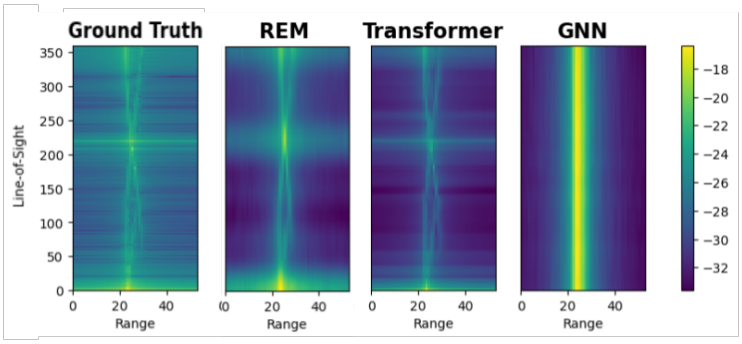


Figure 4: **Static Radar Pattern.** An illustrative example of the static radar predictions from REM and the baselines compared to a ground truth static pattern generated by the first-principles model. The static pattern shows the radar response for the mesh as the radar rotates around the object. For each of these line-of-sights, the radar response signal intensity, i.e. range profile, is discretized into 53 distances.

6.3. Radar Response Prediction

We report the performance of REM and baseline methods on the radar prediction task in Table 1 (See Appendix B for metric details). REM outperforms the baselines in almost all metrics for both the *Basic Shapes* and *Frusta* datasets. The transformer baseline does slightly outperform REM in terms of accurately predicting the locations of maxima and top k peaks (Maxima and Peak Bins). We attribute the increased performance of REM to both the equivariant layers encoding the symmetry present in the radar prediction problem and REM's ability to simultaneously predict the full radar response, encouraging generalization between viewing angles. Additionally, we see that both the GNN and REM models perform much better on the *Basic Shapes* dataset (compared to *Frusta*), implying that the size of the mesh has a large impact on the performance of a graph based encoder. Figure 4 displays the static pattern generated by each model using a mesh from the *Basic Shapes* dataset. Brighter yellow colors represent scattering responses resulting from the radar's interaction with the object at different line-of-sights. These responses stem from two types of scattering phenomena: highly reflective specular scattering, notably intense on flat surfaces perpendicular to the radar line-of-sight, and localized scattering originating from diffraction scattering at the object's edges. We observe that REM does a fairly good job in reconstructing the overall projected shape characteristics and correctly capturing both types of radar scattering responses. See Figure 8 and Figure 9 in Appendix E for additional qualitative results.

6.4. Generalization To Unseen Viewing Angles

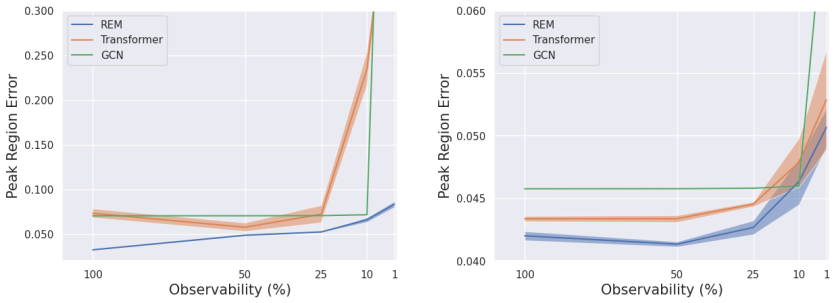
In real world datasets, unlike our simulated datasets, it is much more common to have a sparse radar response with only a few viewing angles. Data efficiency is thus very important,

Table 1: **Radar Response Prediction.** Comparison of radar prediction performance on the *Basic Shapes* and *Frusta* datasets.

Basic Shapes	MSE	Max Peak Value	Max Peak Bin	Peak Value	Peak Bin	Peak Region
GNN	0.35 ± 0.0	0.402 ± 0.0	1.17 ± 0.002	0.582 ± 0.0	1.08 ± 0.002	0.071 ± 0.002
Transformer	0.76 ± 0.335	0.418 ± 0.101	$\mathbf{0.867 \pm 0.012}$	0.558 ± 0.164	$\mathbf{0.636 \pm 0.014}$	0.073 ± 0.011
REM (ours)	$\mathbf{0.31 \pm 0.002}$	$\mathbf{0.352 \pm 0.002}$	1.056 ± 0.02	$\mathbf{0.516 \pm 0.004}$	0.787 ± 0.016	$\mathbf{0.032 \pm 0.001}$

Frusta	MSE	Maxima Value	Maxima Bin	Peak Value	Peak Bin	Peak Region
GNN	0.101 ± 0.0	0.256 ± 0.003	1.468 ± 0.001	0.369 ± 0.001	1.191 ± 0.001	0.046 ± 0.001
Transformer	0.099 ± 0.002	0.239 ± 0.004	$\mathbf{1.246 \pm 0.07}$	0.344 ± 0.004	$\mathbf{0.97 \pm 0.07}$	0.043 ± 0.002
REM (ours)	$\mathbf{0.094 \pm 0.002}$	$\mathbf{0.231 \pm 0.004}$	1.248 ± 0.08	$\mathbf{0.327 \pm 0.013}$	1.033 ± 0.07	$\mathbf{0.042 \pm 0.001}$

as it is necessary to be able to train a radar model with fewer samples. To evaluate how well REM is able learn with fewer samples and to generalize to unseen angles, we train our models on subsets of the radar response. Specifically, we train on 1%, 10%, 25%, 50%, and 100% of the full response, where we sample at regular intervals from $[0, 2\pi]$. Figure 5 plots the peak region prediction error against percentage of observability. We utilize the peak region prediction error as our metric for this experiment, as the main effect of reducing object observability is a decrease in prediction fidelity. A common failure mode is that instead of predicting two peaks in the response, the model might predict a single peak between the two true peaks, which is captured by a high peak region prediction error. Our results show that REM almost always outperforms the baseline methods with the only exception being on the *Frusta* dataset trained on 1% of the viewing angles, where the REM and the Transformer perform about the same.



(*a*) Basic Shapes (*b*) Frusta

Figure 5: **Viewing Angle Ablation.** Peak region prediction error when trained on datasets with restricted viewing angles. Shading shows standard error over 3 random seeds.

6.5. Effect of Maximum Frequency on Response Prediction

One important design choice of SO(3)-equivariant models is the maximum frequency, L, of the model. This maximum frequency limits the angular resolution of our model and effects the granularity of the predicted radar signal on the 2-sphere. Table 2 shows the effect of L on the model prediction metrics, when trained on the *Basic Shapes* dataset. We see that $L = 10$ has the best performance and the performance degrades as we decrease L and the

model is forced to infer more and more due to the restricted angular resolution[1]. We also note that the performance gap is greatest between $L = 4$ and $L = 6$. At higher frequencies, we see that maxima and peak metrics continue to improve, whereas the MSE starts to level off. This implies that the model can capture the general shape of the response even with lower L values, but higher L values are beneficial in order to accurately capture the peaks.

Table 2: **Maximum Frequency (L) Ablation.** Study examining the effect of the maximum frequency utilized in the REM model on the *Basic Shapes* dataset.

L	MSE	Max Peak Value	Max Peak Bin	Peak Value	Peak Bin	Peak Region
2	0.41 ± 0.007	0.453 ± 0.004	1.63 ± 0.12	0.81 ± 0.010	0.982 ± 0.035	0.082 ± 0.024
4	0.38 ± 0.003	0.425 ± 0.001	1.55 ± 0.10	0.72 ± 0.003	0.951 ± 0.034	0.071 ± 0.013
6	0.32 ± 0.005	0.401 ± 0.005	1.41 ± 0.04	0.65 ± 0.005	0.934 ± 0.025	0.045 ± 0.009
8	0.32 ± 0.004	0.382 ± 0.003	1.42 ± 0.07	0.63 ± 0.004	0.923 ± 0.021	0.035 ± 0.018
10	$\mathbf{0.31 \pm 0.002}$	$\mathbf{0.357 \pm 0.002}$	$\mathbf{1.12 \pm 0.02}$	$\mathbf{0.52 \pm 0.004}$	$\mathbf{0.847 \pm 0.016}$	$\mathbf{0.032 \pm 0.001}$

7. Limitations and Discussion

In this work, we present the Radar Equivariant Model, the first SO(3)-equivariant model for predicting radar responses from object meshes. By constraining our model to both equivariant input and output spaces, REM is able to achieve a high level reconstruction of signals generated by a first-principles radar model and shows improved performance and sample efficiency over other encoder-decoder models. Additionally, we present two novel mesh-to-radar datasets containing 2000 meshes, each alongside their full radar responses.

There are three primary limitations to this work. First, although REM's equivariant properties extend to the full SO(3) space, we currently have only evaluated REM on roll-symmetric objects. Additionally, the datasets considered in this work consist of primarily simpler shapes than one would encounter in the real world. Extending this work to non-roll symmetric object datasets, such as in *ShapeNet* (Chang et al., 2015), should better take advantage of our spherical representation and offer improved performance gains. Second, we utilize simulated radar response data in both datasets. While this simulated data should be comparative to real-world data due to our use of high-fidelity radar simulation software to generate the responses, there are complicating factors in real world data, such as material properties and multi-bounce effects that should be investigated. Finally, REM uses a high maximum frequency, L, in both the encoder and the decoder. Because the number of parameters is quadratic in L, both the depth of our model and the size of the input meshes are restricted.

8. Acknowledgements

This work is supported by the Under Secretary of Defense for Research and Engineering under Air Force Contract No. FA8702-15-D-0001 and NSF grants 2107256 and 2134178. We acknowledge the MIT SuperCloud, the Lincoln Laboratory Supercomputer Center, and the NEU Discovery HPC cluster for providing resources that have contributed to the research results in this work.

1. We limited to a maximum frequency of $L = 10$ due to limitations of the e3nn package.

References

Abien Fred Agarap. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*, 2018.

D. Andersh, J. Moore, S. Kosanovich, D. Kapp, R. Bhalla, R. Kipp, T. Courtney, A. Nolan, F. German, J. Cook, and J. Hughes. Xpatch 4: the next generation in high frequency electromagnetic modeling and simulation software. In *Record of the IEEE 2000 International Radar Conference [Cat. No. 00CH37037]*, pages 844–849, 2000. doi: 10.1109/RADAR.2000.851945.

Constantine A Balanis. *Advanced engineering electromagnetics*. John Wiley & Sons, 2012.

Alexander Bogatskiy, Brandon Anderson, Jan Offermann, Marwah Roussi, David Miller, and Risi Kondor. Lorentz group equivariant neural network for particle physics. In *International Conference on Machine Learning*, pages 992–1002. PMLR, 2020.

Johannes Brandstetter, Rob Hesselink, Elise van der Pol, Erik J. Bekkers, and Max Welling. Geometric and physical quantities improve E(3) equivariant message passing. *CoRR*, abs/2110.02905, 2021. URL https://arxiv.org/abs/2110.02905.

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.

E.C. Burt and T.G. Moore. High frequency rcs prediction theory, 1991.

Z. Chance, A. Kern, A. Burch, and J. Goodwin. Differentiable point scattering models for efficient radar target characterization, 2022. URL https://arxiv.org/abs/2206.02075.

Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

Taco Cohen, Mario Geiger, Jonas Kohler, and Max Welling. Spherical cnns. In *International Conference on Learning Representations*, 2018.

Nader Engheta, William D Murphy, Vladimir Rokhlin, and Marius S Vassiliou. The fast multipole method (fmm) for electromagnetic scattering problems. *IEEE Transactions on Antennas and Propagation*, 40(6):634–641, 1992.

Carlos Esteves, Christine Allen-Blanchette, Ameesh Makadia, and Kostas Daniilidis. Learning so (3) equivariant representations with spherical cnns. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 52–68, 2018.

Yutong Feng, Yifan Feng, Haoxuan You, Xibin Zhao, and Yue Gao. Meshnet: Mesh neural network for 3d shape representation, 2018.

Fabian Fuchs, Daniel Worrall, Volker Fischer, and Max Welling. Se (3)-transformers: 3d roto-translation equivariant attention networks. *Advances in neural information processing systems*, 33:1970–1981, 2020.

Hongyang Gao and Shuiwang Ji. Graph u-nets. In *international conference on machine learning*, pages 2083–2092. PMLR, 2019.

Mario Geiger and Tess Smidt. e3nn: Euclidean neural networks, 2022. URL https://arxiv.org/abs/2207.09453.

Mario Geiger, Tess Smidt, Alby M., Benjamin Kurt Miller, Wouter Boomsma, Bradley Dice, Kostiantyn Lapchevskyi, Maurice Weiler, Michał Tyszkiewicz, Simon Batzner, Dylan Madisetti, Martin Uhrin, Jes Frellsen, Nuri Jung, Sophia Sanborn, Mingjian Wen, Josh Rackers, Marcel Rød, and Michael Bailey. Euclidean neural networks: e3nn, April 2022. URL https://doi.org/10.5281/zenodo.6459381.

Walton C. Gibson. *Integral and Physical Optics Methods for RCS Computation*. CRC Press, 2021. doi: 10.1201/9781315161402-15.

Julie Ann Jackson, Brian D. Rigling, and Randolph L. Moses. Parametric scattering models for bistatic synthetic aperture radar. *2008 IEEE Radar Conference*, pages 1–5, 2008. URL https://api.semanticscholar.org/CorpusID:5787687.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=SJU4ayYgl.

Colin Kohler, Anuj Shrivatsav Srikanth, Eshan Arora, and Robert Platt. Symmetric models for visual force policy learning. *arXiv preprint arXiv:2308.14670*, 2023.

Yi-Lun Liao and Tess Smidt. Equiformer: Equivariant graph attention transformer for 3d atomistic graphs. In *The Eleventh International Conference on Learning Representations*, 2022.

Othmane-Latif Ouabi, Radmila Pribić, and Sorin Olaru. Stochastic complex-valued neural networks for radar. In *2020 28th European Signal Processing Conference (EUSIPCO)*, pages 1442–1446, 2021. doi: 10.23919/Eusipco47968.2020.9287425.

Robert Paknys. *Physical Theory of Diffraction*, pages 317–334. 09 2016. ISBN 9781118940563. doi: 10.1002/9781119127444.ch9.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

S. M. Rao, A. W. Glisson, and D. R. Wilton. Electromagnetic scattering by surfaces of arbitrary shape. *IEEE Transactions on Antennas and Propagation*, 30:409–418, May 1982. doi: 10.1109/TAP.1982.1142818.

J. Song, Cai-Cheng Lu, and Weng Cho Chew. Multilevel fast multipole algorithm for electromagnetic scattering by large complex objects. *IEEE Transactions on Antennas and Propagation*, 45(10):1488–1493, October 1997. doi: 10.1109/8.633855.

A Taflove and KR Umashankar. The finite-difference time-domain (fd-td) method for electromagnetic scattering and interaction problems. *Journal of Electromagnetic Waves and Applications*, 1(3):243–267, 1987.

Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint arXiv:1802.08219*, 2018.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Adv. in Neural Info. Proc. Sys.*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Rui Wang, Robin Walters, and Rose Yu. Incorporating symmetry into deep dynamics models for improved generalization. In *International Conference on Learning Representations (ICLR)*, 2021.

Maurice Weiler, Mario Geiger, Max Welling, Wouter Boomsma, and Taco S Cohen. 3d steerable cnns: Learning rotationally equivariant features in volumetric data. *Advances in Neural Information Processing Systems*, 31, 2018.

Timothy A. Wheeler, Martin Holder, Hermann Winner, and Mykel J. Kochenderfer. Deep stochastic radar models. *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 47–53, 2017.

Ruben Wiersma, Ahmad Nasikun, Elmar Eisemann, and Klaus Hildebrandt. Deltaconv: anisotropic operators for geometric deep learning on point clouds. *ACM Transactions on Graphics (TOG)*, 41(4):1–10, 2022.

Yikang Zheng and Yibo Wang. Ground-penetrating radar wavefield simulation via physics-informed neural network solver. *Geophysics*, 88(2):KS47–KS57, 02 2023.

Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++: A nested u-net architecture for medical image segmentation. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 20, 2018, Proceedings 4*, pages 3–11. Springer, 2018.

Appendix A. First-Principles RF Model

Due to the scarcity of available real-world and simulated radar data for training radar models, we simulate our own benchmark dataset. To generate ground truth data, we use the physical optics approximation method (Balanis, 2012), which provides a linear approximation of the more general and highly non-linear scattering formulation for electromagnetic waves. A simple operator that describes physical optics response across the illuminated section of an object for perfectly reflecting material as,

$$F(x,k) = \frac{ik}{2\pi}\int_{R^3} e^{-i2k\langle x,y\rangle}dy,$$

where the incident wave number is $k = 2\pi/\lambda$, λ is the wavelength, and the observation unit vector is,

$$x = (\sin\theta\cos\phi, \sin\theta\sin\phi, \cos\theta),$$

for $\theta \in [0,\pi]$ and $\phi \in [0,2\pi]$. We note that the approximation is valid only in high frequency regions such that $k << 2\pi/D$, where D is the length of the longest side of the object.

As we are particularly interested in far-field sensing, the physical optics approximation is useful as a fairly accurate and flexible simulation tool for training data generation. The simulation input is a parameterized mesh object (an example cross section shown in Figure 6(a)). The simulation calculates the the physical optics response for given observation line-of-sight and frequency and the total radar response of the object is equal to the sum of the individual triangle responses that are visible to the radar. Simulations for this work required generating the response across a linear set of frequencies to emulate a Linear Frequency Modulated (LFM) waveform, where the center frequency is $3e9$Hz and bandwidth is $4e8$Hz using circular polarized waves with orientation RL.

The Radar Cross Section (RCS) for each triangle is calculated using legacy software (Burt and Moore, 1991). The simulation produces a radar observation, $r \in \mathbb{R}^{N_r}$ (Figure 6(b)) for a given viewing angle. The observation is the normalized magnitude of the range-profile as described in Section III of Chance et al. (2022). All the normalized range profiles are then stacked across varying radar viewing angles to generate what is referenced as a radar static pattern (Figure 6(c)). Note that fixing the radar line-of-sight and rotating the object would generate the same radar static response.

Appendix B. Evaluation Metrics

A radar response at a given viewing angle or a range profile consists of a series of real-valued magnitudes corresponding to different detection ranges. Mean squared error can be used to determine the accuracy of a predicted response with respect to a first-principles model response. In addition, since many downstream signal processing and learning tasks are more dependent on the position and magnitude of the peaks than the overall shape of the range profile, we also assess the following more domain- and task-specific metrics:

- **k-Peak Value Error:** Given a predicted response and a ground truth object response, calculate the locations of up to k local maxima with the highest overall magnitude in each response. Associate individual local maxima between the two responses using

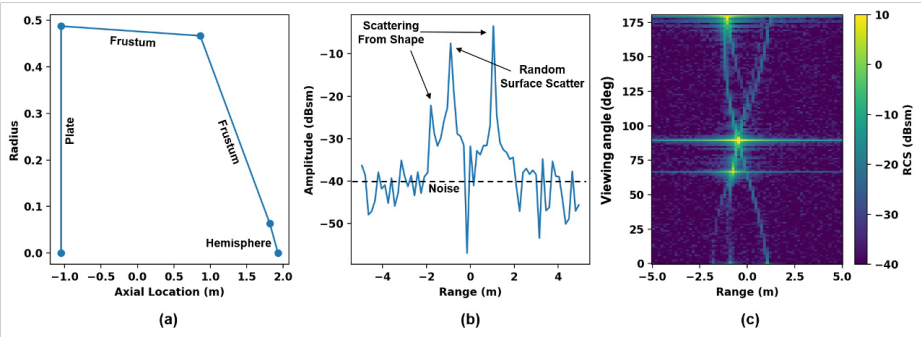


Figure 6: (a) Cross section of random 3D object mesh. (b) The range profile for 20° viewing angle. (c) The corresponding static radar pattern.

distance in number of range bins, and calculate the mean squared error between the magnitudes of associated local maxima.

- **k-Peak Range Bin Error:** Given a predicted response and a ground truth response, calculate the locations of the k local maxima with the highest overall magnitude. Associate individual local maxima between the two responses using distance in number of range bins, and calculate the mean squared error between the range bins of the associated local maxima.

- **Max Peak Value Error:** Same as k-Peak Magnitude Error, but instead of identifying k local maxima, select the highest overall magnitude.

- **Max Peak Range Bin Error:** Same as k-Peak Range Bin Error, but select the range bin of the highest magnitude point.

- **Max Peak Region Error:** Given a predicted object response and the corresponding ground truth response, select the highest magnitude range bin in the ground truth object response as the center of the region of interest. Extract the magnitude of the ground truth and predicted response at this range bin and the l range bins on either side of the center range bin. Compute the mean squared error between the magnitude of the ground truth and predicted response in each of the l range bins in the region, and average across the bins to get the total error.

The peak error metrics measure the model's ability to simulate responses with multiple peaks, while the maxima error metrics measure performance around the largest peak. Using these metrics allows us to more clearly identify significant performance improvements relative to common radar downstream tasks, where non-peak regions of the range profile are usually smoothed out. For our experiments, we ran these metrics with $k = 2$ and $l = 10$.

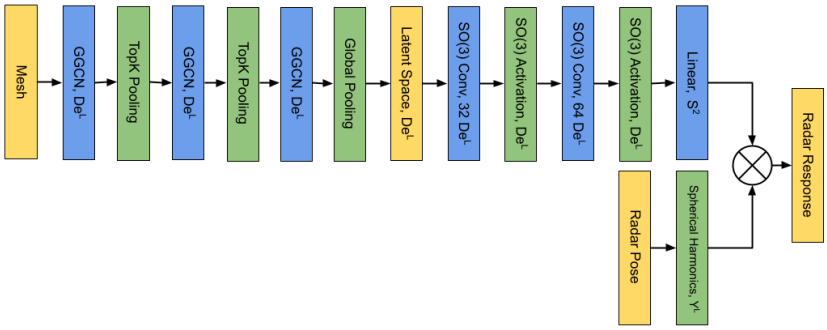


Figure 7: **REM Model Architecture.** Low-level model schematics for the REM model. De^L represents the natural decomposition of a signal in SO(3) for D^L, where D^L are the Wigner D-matrices with a maximum frequency of L.

Appendix C. Network and Training Details

REM. REM's encoder architecture consists of 3 GGCN layers and 2 TopK pooling layers with $k = 0.5$. The same maximum frequency ($L = 10$) is used to compute the spherical harmonics for the edge features for all datasets. The latent code is 16 vectors representing the natural decomposition for a signal over SO(3), $z = 16 \times L^2\,\mathrm{SO}(3)$, with $L = 10$. The decoder consists of 2 SO(3) convolutions with the same maximum frequency of the latent space ($L = 10$). REM is implemented in PyTorch (Paszke et al., 2019) and the *e3nn* library (Geiger et al., 2022). We train using the Adam optimizer(Kingma and Ba, 2014) with the best learning rate and its decay chosen to be 10^{-3} and 0.95 respectively. On the *Basic Shapes* dataset, we use a batch size of 64 and on the *Frusta* dataset we use a batch size of 8. The reduction in batch size is primarily due to the increased size of the meshes in the *Frusta* dataset.

GNN. The GNN consists of 4 layers, each with a graph convolution followed by batch normalization and a ReLU non-linearity. All graph convolutions use kernel size 5 and have 32 output channels. Thus, the GCN outputs a 32-dimensional embedding for every node. Node embeddings are max-pooled to form a global graph embedding, which is then processed by a 2-layer MLP with 64 hidden channels to predict the radar response for the shape.
We train using the Adam optimizer(Kingma and Ba, 2014) with the best learning rate and its decay were chosen to be 10^{-4} and 0.95 respectively. For the GCN architecture, we use a batch size of 256 for both datasets.

Transformer. The Transformer architecture consists of a series of embedding layers, a multi-headed transformer, and a MLP decoder. First, a MLP layer is used to generate a line of sight embedding representing the viewing geometry of the model, and an additional trainable embedding is generated to represent a classification embedding. The classification, orientation, and triangle representation embeddings are placed in sequence, and a standard transformer encoder architecture as introduced in Vaswani et al. (2017) is applied to the

sequence of embeddings to generate an encoded representation. The encoded classification token is then fed to a final MLP layer to generate the predicted response corresponding to the input. We train using the Adam optimizer(Kingma and Ba, 2014) with the best learning rate and its decay were chosen to be 50^{-5} and 0.95 respectively. For the transformer architecture, we use a batch size of 4 for both datasets.

C.1. Loss Function

Most downstream signal processing steps using radar observation data are more concerned with the position and magnitude of the peaks in the range profile than the overall shape of this profile. However, the majority of the range profile is dominated by sidelobe values (lower magnitude values different from the peaks (see Figure 6(b))for an illustration). This can cause adverse effects on peak predictions. For example, utilizing a naive MSE loss can cause poor performance due to the majority of the loss coming from the sidelobe effects. In practice, this results in uninformative low magnitude signals that can impact performance of downstream applications. In this work, for all NN models, we use a weighted-softmax MSE to produce higher losses at the peaks and lower losses at the sidelobe values to encourage the model to learn the more important features,

$$l(x) = \sum_{i=1}^{N} \text{softmax}(x_i)(x_i - \hat{x}_i)^2.$$

Appendix D. Model Consistency

One advantage of equivariant models is that they are robust-by-design against perturbations to the inputs (within their defined symmetries). In our domain, this means that REM is invariant to rotations of the object mesh or the radar orientation. To demonstrate this property, we examine two types of equivariance error: numerical error and functional error. To calculate numerical error, we apply 1000 random transformations to a sampled mesh and compare the model outputs to the model output of the non-transformed mesh. The numerical error is then the average difference between the transformed outputs and the non-transformed output. To calculate the functional error, we apply a random transformation to every sample (mesh, radar line-of-sight) in the test dataset and calculate the average mean-squared error between the predicted response and the ground truth response. Note, because we are randomly transforming both the mesh and the radar line-of-sight with the same transformation, the radar response does not change. We report the results in Table 3. REM has almost no numerical error (we report results to the third decimal place) whereas the baselines have a substantial amount of both numerical and functional error, highlighting the performance guarantees of REM.

Appendix E. Example Radar Response Predictions

We provide some qualitative performance results of the various models, by visualizing the radar learned responses as static patterns. Figure 8 and Figure 9 show the true static patterns alongside the predictions for REM and the baselines.

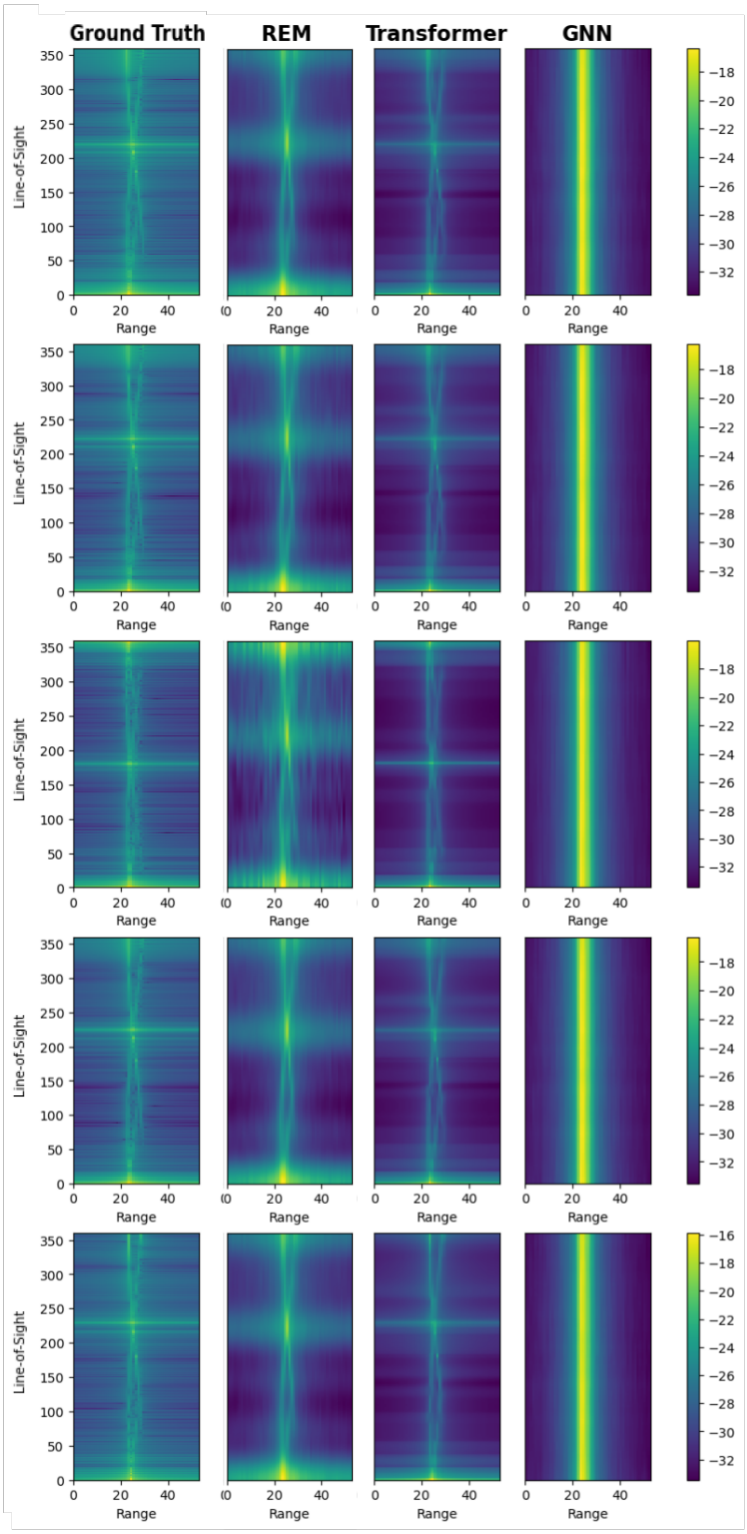


Figure 8: **Basic Shapes Predictions.** Static pattern predictions on the *Basic Shapes* dataset. Each row displays the ground truth static pattern for a randomly sampled mesh alongside the model predictions.

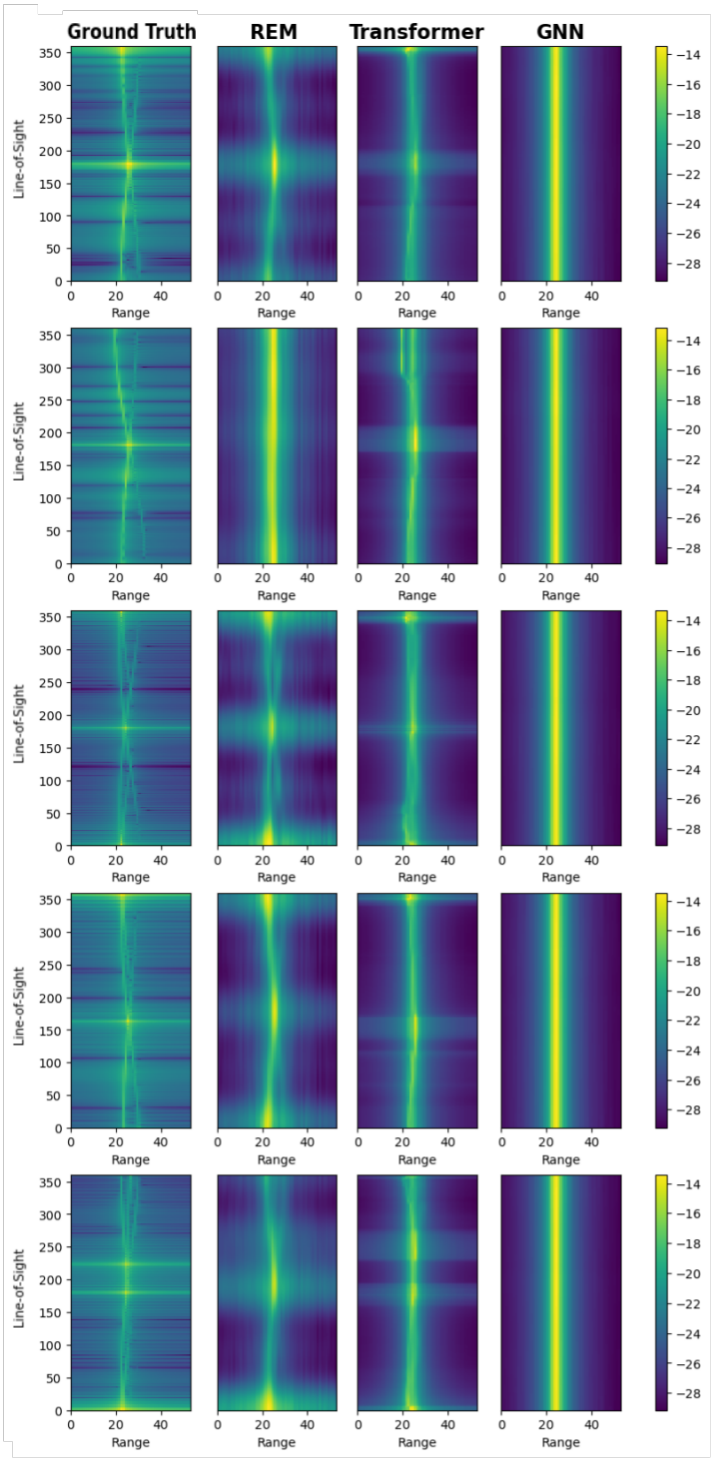


Figure 9: **Frusta Predictions.** Static pattern predictions on the *Frusta* dataset. Each row displays the ground truth static pattern for a randomly sampled mesh alongside the model predictions.

Table 3: **Equivariance Error.** The two types of equivariance error: numerical and functional. Numerical error measures the amount of variance in predicted response when randomly transforming the input mesh and functional error measures the effect of randomly transforming both the mesh and radar line-of-sight.

Basic Shapes	Numerical Error (MSE)	Functional Error (MSE)
GNN	0.853	> 2000
Transformer	0.246	1.064
REM	0.0	0.0

Appendix F. Zero-Shot Generalization To Unseen Meshes

In addition to generalizing to unseen viewing angles, we also test our model's ability to generalize to other unseen meshes. In this zero-shot generalization experiment, we take models trained on the *Basic Shapes* dataset and test them on the *Frusta* dataset (and vice versa). We report these results in Table 4. Interestingly, while both graph-based models are completely unable to generalize and produce large MSEs, the transformer is able to at least produce predictions near the target responses. We attribute this behavior to the difference in the sizes of the graphs with the *Basic Shapes* dataset meshes consisting of 20 nodes whereas meshes from the *Frusta* dataset consist of 250 nodes. Specifically, we hypothesize that the global pooling, which compresses the remaining nodes at the end of the encoder, causes this large accumulation of error.

Table 4: **Zero-Shot Generalization.** Comparison of zero-shot generalization performance from the *Basic Shapes* dataset to the *Frusta* dataset and vice versa. We report the mean and standard deviation over three random seeds.

MSE	**Basic Shapes → Frusta**	**Frusta → Basic Shapes**
GNN	inf	1063 ± 958
Transformer	$\mathbf{1.793 \pm 0.440}$	$\mathbf{0.188 \pm 0.002}$
REM	1011 ± 213	102 ± 13.24

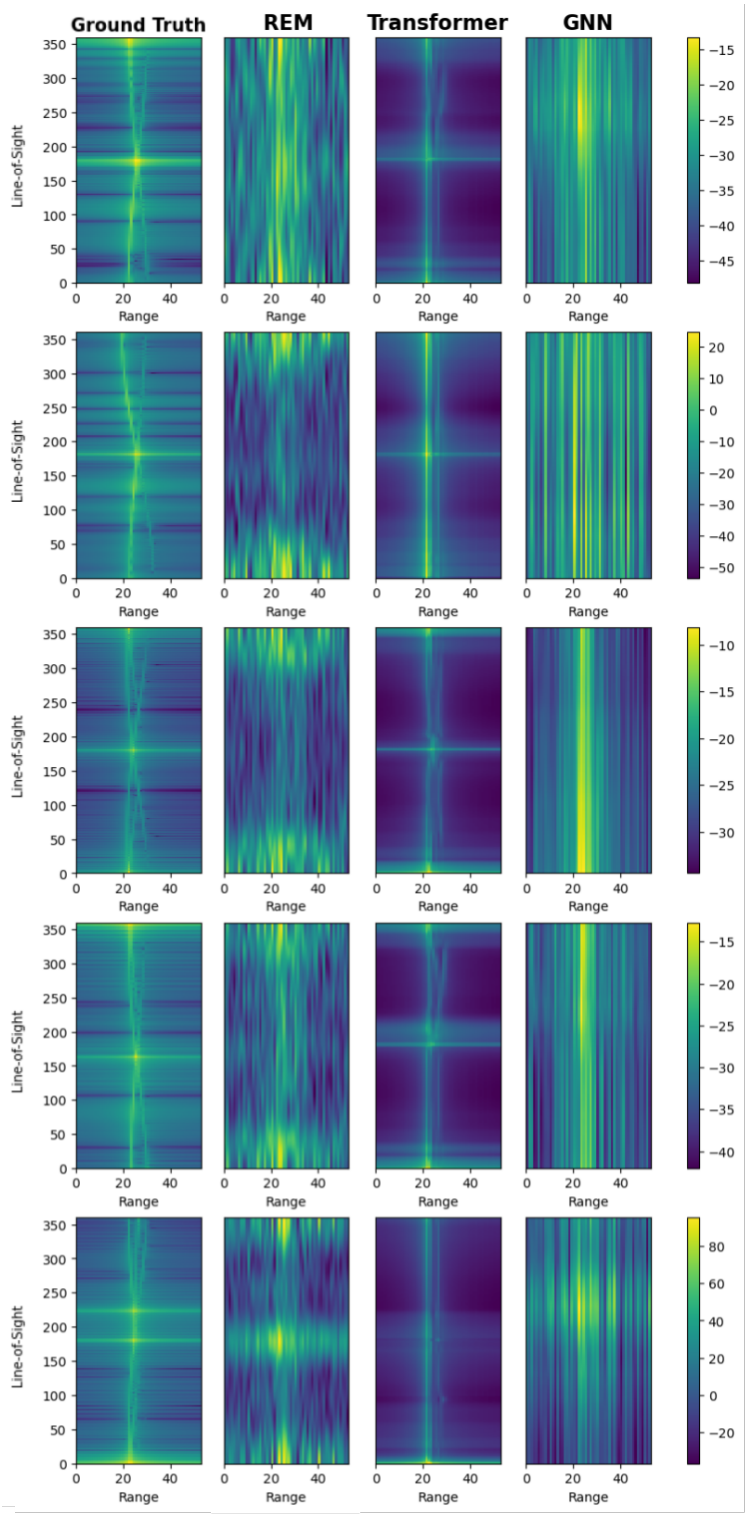


Figure 10: **Zero-Shot Predictions.** Zero-shot static pattern predictions on the *Frusta* dataset. Each row displays the ground truth static pattern for a randomly sampled mesh alongside the model predictions.