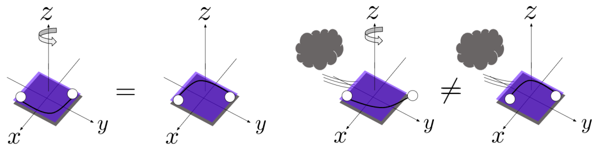
Regularizing Towards Soft Equivariance Under Mixed Symmetries

Hyunsu Kim[1] **Hyungi Lee**[1] **Hongseok Yang**[2 1 3] **Juho Lee**[1 4]

Abstract

Datasets often have their intrinsic symmetries, and particular deep-learning models called equivariant or invariant models have been developed to exploit these symmetries. However, if some or all of these symmetries are only approximate, which frequently happens in practice, these models may be suboptimal due to the architectural restrictions imposed on them. We tackle this issue of approximate symmetries in a setup where symmetries are mixed, i.e., they are symmetries of not single but multiple different types and the degree of approximation varies across these types. Instead of proposing a new architectural restriction as in most of the previous approaches, we present a regularizer-based method for building a model for a dataset with mixed approximate symmetries. The key component of our method is what we call equivariance regularizer for a given type of symmetries, which measures how much a model is equivariant with respect to the symmetries of the type. Our method is trained with these regularizers, one per each symmetry type, and the strength of the regularizers is automatically tuned during training, leading to the discovery of the approximation levels of some candidate symmetry types without explicit supervision. Using synthetic function approximation and motion forecasting tasks, we demonstrate that our method achieves better accuracy than prior approaches while discovering the approximate symmetry levels correctly.



(a) Equivariant trajectory **(b)** Soft-equivariant trajectory

Figure 1: Illustrative example of a system with mixed symmetries with soft equivariances.

1. Introduction

Exploiting symmetries in a dataset is one of the key principles for building an effective deep-learning model. A popular approach for implementing this principle is to restrict the architecture of a neural network in the model so that the model has desired symmetries by construction. The approach has been highly successful, leading to a range of effective so-called equivariant or invariant models (Bronstein et al., 2021), such as CNNs (Cohen & Welling, 2016; Cohen et al., 2019; 2018) and GNNs (Kipf & Welling, 2016; Veličković et al., 2018), that cover different types of symmetries, such as translation invariance.

In practice, however, the symmetries implied in data are often approximate, partially due to measurement noises or unexpected external effects. For such scenarios, models that are equivariant or invariant by construction may be suboptimal due to their architectural restrictions. Moreover, while most of the previous works assume a single type of symmetry, many real-world data come with mixed symmetries, that is, multiple types of symmetries may exist in data. Equivariant models assuming symmetries of just a single type cannot easily be combined to model such mixed symmetries. Even more, those mixed symmetries may be approximate, so different types of symmetries may exhibit different approximation levels. As an example, imagine we want to model the trajectory of a golf ball in 3D space as in Figure 1. The trajectory is $\mathrm{O}(3)$ equivariant, or there are mixed symmetries w.r.t. $\mathrm{O}^x(2)$, $\mathrm{O}^y(2)$, and $\mathrm{O}^z(2)$. Now assume that a wind is blowing along the y-axis. While the trajectory is still $\mathrm{O}^y(2)$-equivariant, it is only approximately equivariant to $\mathrm{O}^x(2)$ and $\mathrm{O}^z(2)$. An $\mathrm{O}(3)$-equivariant model by design would be too restrictive in this case, and a model equivariant only with respect to

[1] Kim Jaechul Graduate School of AI, KAIST, Daejeon, South Korea [2] School of Computing, KAIST, Daejeon, South Korea [3] Discrete Mathematics Group, Institute for Basic Science (IBS), Daejeon, South Korea [4] AITRICS, Seoul, South Korea. Correspondence to: Juho Lee <juholee@kaist.ac.kr>, Hongseok Yang <hongseok.yang@kaist.ac.kr>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

$\mathrm{O}^y(2)$ would miss soft equivariance along x and z axes.

In this paper, we tackle the modeling problem under mixed approximate symmetries, i.e., there are multiple types of symmetries with varying degrees of approximations across the types. Instead of building models symmetric by design, we propose a *regularizer-based* method, where an unconstrained model is regularized toward equivariance. The regularizer is attached for each potential symmetry type expected to be implied in data, and the degree of equivariance approximation of the type is captured by the strength of the regularizer for it - its regularization coefficient. Since it is almost impossible to know the degree of approximations in advance, the regularization coefficients must be carefully tuned to capture the approximation levels correctly. Our method, without explicit supervision, can automatically tune the coefficients during training and, thus, automatically discover the varying degrees of equivariance approximations (from prescribed candidate groups) in the mixed-symmetry settings.

We are not the first to study approximate symmetries. However, the existing works mostly rely on architectural restrictions in relaxed forms (Finzi et al., 2021a; van der Ouderaa et al., 2022; Wang et al., 2022). Moreover, they do not consider multiple types of symmetries with different approximation levels. In contrast, our method does not impose architectural restrictions on a model but solely relies on the equivariance regularizers. As we will show later, the regularizer-based method is especially useful in the mixed-symmetry settings, while the existing works are not straightforwardly extended to those settings.

We experimentally evaluated our method with a synthetic function-approximation task and a motion forecasting task. Our method could correctly discover degrees of approximations of different symmetry types in a relative term and achieve better test accuracy.

We summarize our contributions below:

- We tackle the problem where we have multiple types of (approximate) symmetries with different levels of equivariance/invariance errors.
- We propose a novel method regularizing an unrestricted model with (approximate) symmetry constraints, and present an algorithm that can automatically identify approximation levels of different symmetry types during training.
- We demonstrate the effectiveness of our approach on synthetic and real-world tasks with multiple types of (approximate) symmetries.

2. Backgrounds

We start with a review on the formalization of symmetries of neural networks in terms of groups. We also review so called residual pathway prior (Finzi et al., 2021a), a recent proposal for handling approximate symmetries.

2.1. Group Representation and Equivariance

A *representation* of a group G on a Euclidean space $\mathbb{R}^n$ is a function ρ from G to the general linear group on $\mathbb{R}^n$ (i.e., the group of invertible $n \times n$ matrices with matrix multiplication as group composition) such that ρ preserves the composition operator of the group. When we have representations of a group G in Euclidean spaces $\mathcal{X}$ and $\mathcal{Y}$, denoted $\rho_\mathcal{X}$ and $\rho_\mathcal{Y}$, we say that a function $f : \mathcal{X} \to \mathcal{Y}$ is *G-equivariant* if for all $g \in G$ and $x \in \mathcal{X}$, we have

$$f\big(\rho_\mathcal{X}(g)(x)\big) = \rho_\mathcal{Y}(g)\big(f(x)\big). \tag{1}$$

Intuitively, this condition means that f does not actively use information that can be altered by group elements g. The convolution layers in CNNs are a leading example that is equivariant with respect to the translation group (in an ideal setup where the images are defined over the entire plane $\mathbb{R}^2$). A range of neural-network architectures that ensure equivariance (including equivariant multilinear perceptions to be explained next) have been developed because they usually generalize better than non-equivariant counterparts.

2.2. Equivariant Multilayer Perceptrons

Equivariant Multilayer Perceptrons (EMLPs) (Finzi et al., 2021b) are models that are guaranteed to be equivariant with respect to a given group G and its representation ρ. As its name indicates, an EMLP is identical to a standard multilayer perceptron except for one thing: its weights and biases are not network parameters, but they are constructed out of other parameters. This further parameterization of weights and biases ensure that all the linear layers of the EMLP are equivariant by construction.

To describe the linear layers of an EMLP formally, we need to recall a few facts. First, when G has representations ρ on $\mathbb{R}^n$ and ρ' on $\mathbb{R}^m$, the set of G-equivariant linear maps from $\mathbb{R}^n$ to $\mathbb{R}^m$ forms a vector space. Thus, it has an orthonormal basis $\mathcal{B} = \{M_1, \ldots, M_d\}$ where the M_i's are $m \times n$ matrices representing G-equivariant linear maps and when reshaped to vectors via stacking columns (i.e., $\mathrm{vec}(M_1), \ldots, \mathrm{vec}(M_d)$), the matrices become orthonormal vectors of $(m \times n)$ dimension. Second, the set of vectors v in $\mathbb{R}^m$ that are invariant with respect to G and ρ' (i.e., $\rho'(g)(v) = v$ for all $g \in G$) forms a subspace of $\mathbb{R}^m$. So, this subspace also has an orthonormal basis. The linear layers of EMLP are defined in terms of these two bases.

Assume that the l-th layer of the network has n_l input nodes and n_{l+1} output nodes. Formally, each linear layer l of an EMLP is an affine map $\mathrm{Linear}_{\mathrm{EMLP}} : \mathbb{R}^{n_l} \to \mathbb{R}^{n_{l+1}}$ defined as follows:

$$\begin{aligned} \mathrm{Linear}_{\mathrm{EMLP}}(x) &= Wx + b, \\ \mathrm{vec}(W) &= Q\theta, \quad b = R\beta, \end{aligned} \tag{2}$$

where $\mathrm{vec}(W)$ is the vector obtained by stacking the columns of the matrix W, Q is a fixed matrix with $(n_{l+1} \times n_l)$ rows and d columns, and R is a fixed matrix with n_{l+1} rows and r columns. The columns of the matrix Q when reshaped to $n_{l+1} \times n_l$ matrices via unstacking form an orthonormal basis of the space of G-equivariant linear maps from $\mathbb{R}^{n_l}$ to $\mathbb{R}^{n_{l+1}}$. Similarly, the columns of the other matrix R form the basis of the subspace of G-invariant vectors in $\mathbb{R}^{n_{l+1}}$. The parameters to be trained are $\theta \in \mathbb{R}^d$ and $\beta \in \mathbb{R}^r$, the coefficients combining the orthonomal basis.

2.3. Residual Pathway Prior

The Residual Pathway Prior (RPP) (Finzi et al., 2021a) is a recent proposal for learning an approximately-equivariant neural network. It is based on the idea of combining equivariant and non-equivariant transformations together in each network layer. Concretely, it is the following variant of the EMLP, which adds a standard linear layer, called residual pathway, to each equivariant linear layer of the EMLP:

$$\begin{aligned} \mathrm{Linear}_{\mathrm{RPP}}(x) &= Wx + b, \\ \mathrm{vec}(W) &= QQ^{\intercal}\mathrm{vec}(W_1) + \mathrm{vec}(W_2), \\ b &= RR^{\intercal}b_1 + b_2, \end{aligned} \tag{3}$$

where Q and R are from the equations in (2), and $Q^{\intercal}\mathrm{vec}(W_1)$ and $R^{\intercal}b_1$ correspond to θ and β in the same equations, respectively. Note that $\mathrm{Linear}_{\mathrm{RPP}}(x)$ is the sum of the EMLP's linear layer on x and $W_2x + b_2$. The residual pathway refers to the latter part.

The parameters of an RPP are trained with the following ℓ_2-regularization, which comes from the prior distributions on those parameters:

$$\begin{aligned} &\mathcal{R}^{\mathrm{RPP}}(W_1, b_1, W_2, b_2) \\ &= \frac{\|\mathrm{vec}(W_1)\|^2 + \|b_1\|^2}{2\sigma_1^2} + \frac{\|\mathrm{vec}(W_2)\|^2 + \|b_2\|^2}{2\sigma_2^2}, \end{aligned} \tag{4}$$

with σ_2 being substantially smaller than σ_1, which encourages that residual layers play only a minor role for inference.

3. Equivariance Regularizer

In this section, we present our equivariance regularizer, the key conceptual contribution of the paper. We assume that a collection of groups $G_1, G_2, \ldots, G_K$ are given, capturing different types of symmetries, and also that these groups come with representations for input and output spaces of all linear layers. The latter assumption enables us to talk about G_k-equivariant linear or affine maps for all layers. In our presentation, we fix a layer l, and describe how our regularizers constrain network parameters at that layer. For notational simplicity, we omit the layer indices from the parameters, unless required to be specified.

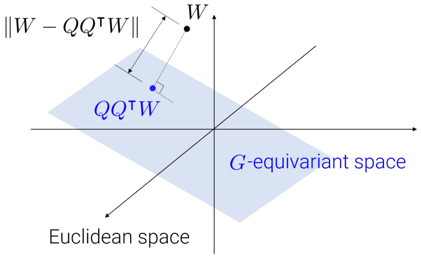


Figure 2: The projection-based equivariance regularizer for a group G measures the distance $\|W - QQ^{\intercal}W\|$, where W is either $\mathrm{vec}(W)$ or b in the standard linear layer and Q is an orthonormal basis of the space of G-equivariant matrices or G-invariant vectors.

3.1. Projection-Based Equivariance Regularizer

For every $k = 1, \ldots, K$, write Q_k and R_k for the matrices from the equations in (2); the columns of Q_k form an orthonormal basis of G_k-equivariant linear maps from $\mathbb{R}^{n_l}$ to $\mathbb{R}^{n_{l+1}}$ after being reshaped into $n_{l+1} \times n_l$ matrices, and the columns of R_k form an orthonormal basis of n_{l+1}-dimensional G_k-invariant vectors in $\mathbb{R}^{n_{l+1}}$.

Our Projection-based Equivariance Regularizer (PER) for a group G_k is defined by

$$\begin{aligned} \mathcal{R}_k^{\mathrm{PER}}(W, b) = &\frac{\lambda_k}{2}\|\mathrm{vec}(W) - Q_k Q_k^{\intercal}\mathrm{vec}(W)\|^2 \\ &+ \frac{\lambda_k}{2}\|b - R_k R_k^{\intercal} b\|^2, \end{aligned} \tag{5}$$

where W and b are parameters of the l-th layer of the network, and λ_k is a regularization coefficient for the group G_k. Modulo the reshaping into the vector form, the term $Q_k Q_k^{\intercal}\mathrm{vec}(W)$ is the projection of W (expressing a linear map from $\mathbb{R}^{n_l}$ to $\mathbb{R}^{n_{l+1}}$) to the space of G_k-equivariant linear maps expressed as $n_{l+l} \times n_l$ matrices. Thus, the first summand measures the ℓ_2-distance from W to the space of G_k-equivariant linear maps. Similarly, the second summand uses the projection of the bias term and measures the ℓ_2-distance from b to the space of G_k-invariant vectors. This regularizer can be a part of a learning objective during training, so that the training moves the parameters W and b towards the space of the G_k-equivariant linear maps or G_k-invariant vectors. An advantage of this regularizer-based approach for enforcing symmetries is that we can easily combine multiple regularizers for different groups simply by adding them to the objective function. Concretely, in our setup of K different groups, we can use the following regularizer for the parameters of the l-th layer:

$$\mathcal{R}^{\text{PER}}(W, b) = \sum_{k=1}^{K} \mathcal{R}_k^{\text{PER}}(W, b) \tag{6}$$

The regularization coefficients λ_k control the strength of enforcing different types of symmetries formalized by different groups $G_1, \ldots, G_K$. Ideally, these parameters are set according to the approximation levels of different symmetry types. However, we don't know the approximation levels in advance. In the next section, we explain how to infer such parameters during training without explicit supervision.

An implicit assumption under the regularizer $\mathcal{R}^{\text{PER}}(W, b)$ is that the ℓ_2-distance measures how much the symmetry with respect to G_k is violated by the corresponding parameters of the network. The following proposition supports that assumption, showing that minimizing the ℓ_2 distances indeed minimizes the equivariance error.

Proposition 3.1. *Let f be an S-layer MLP with the weight matrix $W^{(l)}$ and the bias term $b^{(l)}$ at each layer l. Assume that the activation functions of f are G-equivariant and L-Lipchitz continuous. Also, assume a constant $U > 0$ such that $\|x\| < U$ for every $x \in \mathcal{X}$, and the operator norms $\|\rho_{\mathcal{X}}(g)\|_{\text{op}}$ and $\|\rho_{\mathcal{Y}}(g)\|_{\text{op}}$ for any $g \in G$ are also bounded by U. Then, there exists a constant $C > 0$ depending on S, L, and U only, such that for all $\{(W^{(l)}, b^{(l)})\}_{l=1,\ldots,S}$, if the operator norm $\|W^{(l)}\|_{\text{op}}$ and the ℓ_2 norm $\|b^{(l)}\|$ are bounded by U for every l, we have*

$$\begin{aligned}\sup_{x\in\mathcal{X}, g\in G} &\|\rho_{\mathcal{Y}}(g)f(x) - f(\rho_{\mathcal{X}}(g)x)\| \\ &\leq C \cdot \sum_{l=1}^{S} \Big(\|\text{vec}(W^{(l)}) - Q_k^{(l)} Q_k^{(l)\intercal} \text{vec}(W^{(l)})\| \\ &\qquad + \|b^{(l)} - R_k^{(l)} R_k^{(l)\intercal} b^{(l)}\| \Big).\end{aligned} \tag{7}$$

The proof of a refined version of this proposition is given in Appendix A. According to Proposition 3.1, the equivariance error of a model is bounded by the ℓ_2-distances of parameters to equivariance subspaces, and the minimum equivariant error is achieved when the ℓ_2-distances are zero, which happens when the value of the regularizer is zero.

Our methodology presents a comparable functionality to RPP, yet it allows the model to discover soft equivariant weights with a reduced number of parameters. The distinctions between RPP and PER have been visually depicted in Appendix B.

3.2. Adjustment of Hyperparameters of Groupwise Equivariance Regularizers

The regularization coefficients $\lambda_1, \ldots, \lambda_K$ in (6) play an important role of controlling the strengths of groupwise equivariance constraints that we impose on the model. We empirically observed that a better model is learned when these regularization coefficients for different groups (and hence the strengths of regularization for these groups) are correlated with the approximation levels of symmetries for those groups in a dataset. That is, if $(\lambda_1^*, \ldots, \lambda_K^*)$ are the coefficents leading to the best model with the lowest validation error after training, then a smaller λ_k^* value means weaker symmetry (more approximation error) for the group G_k, and a larger λ_k^* means more exact symmetry for the group G_k.

Based on this observation, we propose an automatic tuning procedure that could discover the approximation levels of different symmetry types (formalized by different groups and captured through the regularizers) in a data-driven way. Given an S-layer MLP f, let $\mathcal{R}_k^{\text{PER}}(f) = \sum_{\ell=1}^{S} \mathcal{R}_k^{\text{PER}}(W^{(l)}, b^{(l)})$. We first initialize all the regularization coefficients with the same value, and in the early stage of training, adjust the coefficients $\{\lambda_k\}_{k=1,\ldots,K}$ based on the magnitudes of the corresponding regularizers $\{\mathcal{R}_k^{\text{PER}}(f)\}_{k=1,\ldots,K}$ with the following formula:

$$\lambda_k^* = \lambda_k \left(\frac{\min\{\mathcal{R}_k^{\text{PER}}(f)\}_{k=1,\ldots,K}}{\mathcal{R}_k^{\text{PER}}(f)} \right)^{\gamma}, \tag{8}$$

where γ is a scaling factor calibrating how much the approximation difference will be reflected in the coefficients. We empirically confirmed that setting $\gamma \in [2, 5]$ gives reasonable results.

3.3. Extension of EMLP for Mixed Symmetries

Unlike our method which can conveniently combine multiple regularizers for mixed symmetries, it is not straightforward to extend existing (approximately) equivariant models for mixed symmetry settings. Here, as a baseline, we describe a naïve extension of EMLP for our setup which assumes multiple types of symmetries formalized by groups $G_1, \ldots, G_K$. Assume the model is equivariant to the first L groups $G_1, \ldots, G_L$ and softly equivariant for the rest. For $G_1, \ldots, G_L$, we first compute a joint subspace by solving the set of equivariance constraints for L groups and denote the corresponding bases Q_1 and R_1. Similarly, we compute a joint subspace for all groups $G_1, \ldots, G_K$ and denote the bases Q_2 and R_2. A Mixed EMLP (MEMLP) is defined as

$$\begin{aligned}&\text{Linear}_{\text{MEMLP}}(x) = W_1 x + b_1 + W_2 x + b_2 \\ &\text{vec}(W_q) = Q_q \theta_q, \;\; b = R_q \beta_q \text{ for } q = 1, 2.\end{aligned} \tag{9}$$

Here, both $W_1 x + b_1$ and $W_2 x + b_2$ are equivariant to $G_1, \ldots, G_L$, so the overall model is equivariant to them.

Table 1: Test MSE for the moment of inertia task. EMLP and RPP are built with $\mathrm{O}(3)$ and MEMLP is built with $\mathrm{O}(3)$-EMLP and $\mathrm{O}^{(\mathrm{ax})}$-EMLP where $\mathrm{ax} \in \{x, y, z\}$.

Equiv group	MLP	$\mathrm{O}(2)$EMLP	$\mathrm{O}(3)$EMLP	RPP	MEMLP	PER
$\mathrm{O}(3)$	4.25 ± 0.17	-	1.13 ± 0.36	2.66 ± 1.43	-	$\mathbf{0.27} \pm \mathbf{0.23}$
$\mathrm{O}^x(2)$	2.84 ± 0.12	1.75 ± 0.63	62.36 ± 41.10	2.06 ± 1.12	3.38 ± 0.92	$\mathbf{0.25} \pm \mathbf{0.16}$
$\mathrm{O}^y(2)$	2.78 ± 0.12	1.73 ± 0.11	29.11 ± 14.06	1.72 ± 0.61	2.87 ± 0.31	$\mathbf{0.56} \pm \mathbf{0.48}$
$\mathrm{O}^z(2)$	2.69 ± 0.10	1.56 ± 0.17	46.32 ± 10.18	1.86 ± 1.13	2.75 ± 0.29	$\mathbf{0.32} \pm \mathbf{0.26}$
-	6.81 ± 0.23	-	10.65 ± 2.08	4.16 ± 0.49	-	$\mathbf{0.34} \pm \mathbf{0.28}$

On the other hand, since $W_1 x + b_1$ is not equivariant to $G_{L+1}, \ldots, G_K$, the overall model is only softly equivariant to them. The level of soft equivariance is controlled by the prior variances for W_1 and W_2, as in the case of RPP.

4. Experiments

To demonstrate the effectiveness of our method, especially for its utility in discovering mixed symmetries from data, we compare ours to (approximately) equivariant baseline models for a synthetic function approximation task and a real-world motion forecasting task. The baselines we are comparing against include EMLP, RPP, and MEMLP described in § 3.3. The network architectures used for those models including our model in common have four layers with the gated nonlinearities and bilinear layers as described in Finzi et al. (2021b;a). Throughout all the experiments, to see the net effect of the abilities of the models capturing equivariances, we controlled the sizes of the competing models so that all of them have similar number of parameters.

Additional information regarding the experiments, such as the specific hyperparameters employed and the data preprocessing details applied, can be found in Appendix F. Furthermore, Appendix C provides insightful recommendations for efficient initializations of neural networks in the PER settings. Additionally, Appendix G presents supplementary experiments conducted to assess the robustness of our method.

4.1. Synthetic Function-Approximation Task

4.1.1. The moment of Inertia Function

We generate a synthetic dataset having mixed symmetries by adding a perturbation to a symmetric function that computes the moment of inertia. Given the masses and positions of five particles, denoted by $(m_{1:5}, \mathbf{x}_{1:5}) := (m_i, \mathbf{x}_i)_{i=1}^5$, the moment of inertia is computed as follows:

$$\mathcal{I}(m_{1:5}, \mathbf{x}_{1:5}) := \sum_{i=1}^{5} m_i (\mathbf{x}_i^\intercal \mathbf{x}_i \mathbf{I} - \mathbf{x}_i \mathbf{x}_i^\intercal). \tag{10}$$

The moment-of-inertia function is equivariant with respect to group $\mathrm{O}(3)$, which consists of rotations and reflections. That is, for a group element $g \in \mathrm{O}(3)$, $\rho(g)\mathcal{I}(m_{1:5}, \mathbf{x}_{1:5}) = \mathcal{I}(\rho(g)(m_{1:5}, \mathbf{x}_{1:5}))$. Here g acts on each position $\mathbf{x}_i$, that is, $\rho(g)(m_{1:5}, \mathbf{x}_{1:5}) = (m_i, g\mathbf{x}_i)_{i=1}^5$ where g in $g\mathbf{x}_i$ is represented as a 3×3 matrix. The output of the function $M = \mathcal{I}(m_{1:5}, \mathbf{x}_{1:5})$ is a 3×3 matrix, and g acts on M as $\rho(g)(M) = gMg^{-1}$.

Table 2: Test MSE for the CosSim task. EMLP and RPP are built with $(\mathrm{SO}(3), \mathrm{S}(3))$ and MEMLP is built with $(\mathrm{SO}(3), \mathrm{S}(3))$-EMLP and $\mathrm{SO}(3)$ or $\mathrm{S}(3)$EMLP EMLP. Sub-EMLP stands for either $\mathrm{SO}(3)$ or $\mathrm{S}(3)$ and EMLP stands for $(\mathrm{SO}(3), \mathrm{S}(3))$-EMLP. All values are in a scale of $\times 10^{-1}$.

Inv group	MLP	Sub-EMLP	EMLP	RPP	MEMLP	PER
$\mathrm{SO}(3), \mathrm{S}(3)$	0.41 ± 0.03	-	1.10 ± 0.02	1.10 ± 0.03	-	$\mathbf{0.32} \pm \mathbf{0.02}$
$\mathrm{SO}(3)$	0.46 ± 0.07	$\mathbf{0.39} \pm \mathbf{0.30}$	2.54 ± 0.10	2.57 ± 0.10	2.56 ± 0.10	0.44 ± 0.03
$\mathrm{S}(3)$	0.69 ± 0.04	2.14 ± 0.11	2.14 ± 0.11	2.18 ± 0.09	2.18 ± 0.09	$\mathbf{0.65} \pm \mathbf{0.09}$
-	3.76 ± 0.32	-	3.76 ± 0.32	3.84 ± 0.04	-	$\mathbf{0.66} \pm \mathbf{0.13}$

To generate data, we draw $\mathbf{x}_{1:5} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $m'_{1:5} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1)$, and then compute $m_i = \mathrm{softplus}(m'_i)$. We then compute the moment of inertia with (10) and add five different types of errors to the output. Let $\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{z}} \in \mathbb{R}^3$ be the orthonormal basis vectors of the x, y and z axes, respectively. The five types of errors and the corresponding approximate symmetries are as follows:

1. $\mathbf{0}$ (no error), $\mathrm{O}(3)$-equivariant.
2. $-\mathcal{I}\hat{\mathbf{x}}\hat{\mathbf{x}}^\intercal$, $\mathrm{O}^x(2)$ equivariant, soft $\mathrm{O}(3)$ equivariant.
3. $-\mathcal{I}\hat{\mathbf{y}}\hat{\mathbf{y}}^\intercal$, $\mathrm{O}^y(2)$ equivariant, soft $\mathrm{O}(3)$-equivariant.
4. $-\mathcal{I}\hat{\mathbf{z}}\hat{\mathbf{z}}^\intercal$, $\mathrm{O}^z(2)$ equivariant, soft $\mathrm{O}(3)$ equivariant.
5. $-0.3\mathcal{I}(\hat{\mathbf{x}}\hat{\mathbf{x}}^\intercal - \hat{\mathbf{y}}\hat{\mathbf{y}}^\intercal + \hat{\mathbf{z}}\hat{\mathbf{z}}^\intercal)$, soft $\mathrm{O}(3)$-equivariant.

For the baselines, we consider $\mathrm{O}(3)$EMLP, $\mathrm{O}(3)$RPP, and $\mathrm{O}^{(\mathrm{axis})}$-$\mathrm{O}(3)$EMLP which is equivariant to $\mathrm{O}^{(\mathrm{axis})}$ and softly equivariant to $\mathrm{O}(3)$, where $\mathrm{axis} \in \{x, y, z\}$ is chosen according to the symmetry in the data. Our model, denoted by PER, regularizes an MLP with equivariance regularizers for the groups $(\mathrm{O}^x(2), \mathrm{O}^y(2), \mathrm{O}^z(2))$.

4.1.2. The CosSim Function

Another synthetic function-approximation task we consider is the CosSim function which computes the average cosine similarity between three particles. Given the positions of three particles in 3D space, denoted by $\mathbf{x}_{1:3} := \{\mathbf{x}_i\}_{i=1}^3$ with each $\mathbf{x}_i \in \mathbb{R}^3$, the CosSim function computes

$$\begin{aligned} &\mathrm{AvgCS}(\mathbf{x}_{1:3}) \\ &= \frac{\mathrm{CS}(\mathbf{x}_1, \mathbf{x}_2) + \mathrm{CS}(\mathbf{x}_2, \mathbf{x}_3) + \mathrm{CS}(\mathbf{x}_1, \mathbf{x}_3)}{3}, \end{aligned} \tag{11}$$

where $\mathrm{CS}(\mathbf{a}, \mathbf{b}) := \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\| \|\mathbf{b}\|}$. The AvgCS function is invariant to both $\mathrm{SO}(3)$ and $\mathrm{S}(3)$ where $\mathrm{SO}(3)$ is a rotation group in $\mathbb{R}^3$ and $\mathrm{S}(3)$ is a scaling group in $\mathbb{R}^3$. That is, for a group element $g \in \mathrm{SO}(3)$ or $g \in \mathrm{S}(3)$, $\mathrm{AvgCS}(\rho(g)\mathbf{x}_{1:3}) =$

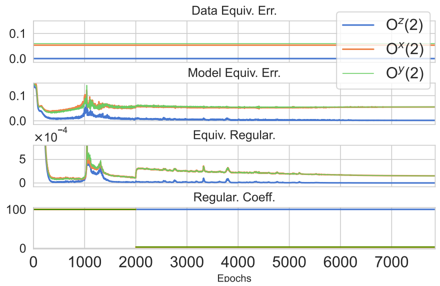


Figure 3: The training progress of PER on a dataset equivariant to $\mathrm{O}^z(2)$ and softly equivariant to $\mathrm{O}^x(2)$ and $\mathrm{O}^y(2)$. From top to bottom: data equivariance error (defined in Equation 12), model equivariance error, the values of the equivariance regularizers $(\mathcal{R}_k^{\mathrm{PER}}(f))_{k=1}^3$, and the regularization coefficients $(\lambda_k)_{k=1}^3$. The coefficients are adjusted automatically at epoch 2000.

$\mathrm{AvgCS}(\mathbf{x}_{1:3})$, where $\rho(g)\mathbf{x}_{1:3} = \{g\mathbf{x}_i\}_{i=1}^3$. Similarly to the inertia task, to generate data, we draw $\mathbf{x}_{1:3} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mathbf{0}, \mathbf{I})$, compute (11), and inject four different types of errors.

1. $\mathbf{0}$ (no error), $\mathrm{SO}(3)$ and $\mathrm{S}(3)$ invariant.
2. $\frac{-\sum_{i=1}^3 \|\mathbf{x}_i\|}{3}$, $\mathrm{SO}(3)$-invariant, soft $\mathrm{S}(3)$-invariant.
3. $\frac{-\sum_{i=1}^3 |\mathbf{x}_i \cdot \hat{\mathbf{x}}|}{\sum_{j=1}^3 (|\mathbf{x}_j \cdot \hat{\mathbf{y}}| + |\mathbf{x}_j \cdot \hat{\mathbf{z}}|)}$, soft $\mathrm{SO}(3)$ invariant, $\mathrm{S}(3)$ invariant.
4. $\frac{-\sum_{i=1}^3 \|\mathbf{x}_i\|}{3} + \frac{\sum_{i=1}^3 |\mathbf{x}_i \cdot \hat{\mathbf{x}}|}{\sum_{j=1}^3 (|\mathbf{x}_j \cdot \hat{\mathbf{y}}| + |\mathbf{x}_j \cdot \hat{\mathbf{z}}|)}$, soft $\mathrm{SO}(3)$ and $\mathrm{S}(3)$ invariant.

For the baselines, we consider $(\mathrm{SO}(3), \mathrm{S}(3))$EMLP, $(\mathrm{SO}(3), \mathrm{S}(3))$RPP, $\mathrm{SO}(3)$-$\mathrm{S}(3)$MEMLP (equivalent to $\mathrm{SO}(3)$ and softly equivalent to $\mathrm{S}(3)$), and $\mathrm{S}(3)$-$\mathrm{SO}(3)$MEMLP (equiv to $\mathrm{S}(3)$ and softly equiv to $\mathrm{SO}(3)$). For computing the basis of the joint equivariant subspace of $\mathrm{SO}(3)$ and $\mathrm{S}(3)$, we solve for the conjunction of equivariance constraints for two groups, as we explained in § 3.3.

4.1.3. Analysis of the results

Overall results. We summarize the results for the moment of inertia task in Table 1 and the results for the CosSim task in Table 2. For both tasks, PER significantly outperforms baselines, across all error types having different types of approximate equivariance. From below, we empirically show that this is because PER correctly captures the approximate equivariance in data and adjusts the regularization coefficients accordingly.

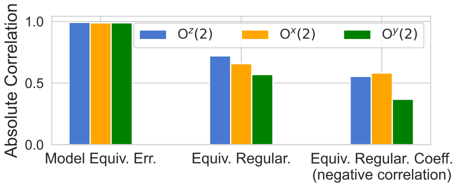


Figure 4: Absolute Pearson correlation coefficients between data equivariance errors and (model equivariance error, the values of equivariance regularizers, the regularization coefficients), measured across 13 datasets of different degrees and types of approximate equivariances.

Discovery of approximate equivariance. We check whether our model correctly learns the degree of approximate equivariance implied in the dataset. For instance, in the moment of inertia task, when data is perturbed by the Item 3, our model should be able to detect that the data is $\mathrm{O}^y(2)$ equivariant and softly $\mathrm{O}(3)$ equivariant. Figure 3 illustrates the progress of equivariance errors, values of regularization, and their coefficients during training. Here, the data is perturbed by the error type Item 4, so it is $\mathrm{O}^z(2)$ equivariant and softly equivariant to $\mathrm{O}^x(2)$ and $\mathrm{O}^y(2)$. As we can see in the figure, our model captures the difference between the equivariance error levels and adjusts the regularization coefficients at the epoch 2000. Here, our model lowers the regularization coefficients for $\mathrm{O}^x(2)$ (to 2.91) and $\mathrm{O}^y(2)$ (to 3.07) while keeping the coefficient for $\mathrm{O}^z(2)$ (to 100.0). As a result, the model trained with the adjusted regularization could correctly match the equivariance errors assumed in the data.

To further demonstrate that our model indeed captures the equivariance error levels from data, we measure the Pearson correlation coefficients between the (model equivariance errors, the values of the equivariance regularizers $(\mathcal{R}_k^{\mathrm{PER}}(f))_{k=1}^3$, the regularization coefficients $(\lambda_k)_{k=1}^3$), and the equivariance errors assumed in the data. Here, the model equivariance error is measured as a Monte Carlo approximation of the following expectation of equivariance error (scaled) of a model f:

$$\mathbb{E}_{\mathbf{x}\in\mathcal{X}, g\in G}\left[\frac{\|\rho_{\mathcal{Y}}(g)f(\mathbf{x}) - f(\rho_{\mathcal{X}}(g)\mathbf{x})\|}{\|\rho_{\mathcal{Y}}(g)f(\mathbf{x})\|\|f(\rho_{\mathcal{X}}(g)\mathbf{x})\|}\right]. \tag{12}$$

We measure the correlations across 13 different types of datasets with varying error types and scales, and summarize the result in Figure 4 (the specific values for each sample are written in Appendix D). The model equivariance error is highly correlated with the data equivariance error, indicating that the model correctly captures the equivariance errors implied in the data. Equivariance regularizers and their coefficients are also correlated with the data equivariance error, supporting our claim that the automatic tuning procedure in our method can discover the approximate equivariance (from prescribed candidate groups) in a data-driven way.

4.2. Motion Forecasting Task

4.2.1. Task Description

The goal of this task is to predict the future positions of a moving vehicle given past positions. The position of the vehicle is represented with a 3D coordinate (x, y, z). We collect the trajectories from Waymo Open Motion Dataset (WOMD) (Ettinger et al., 2021) containing trajectories of vehicles moving on roads. We use 16,814 trajectories for training, 3,339 trajectories for validation, and 3,563 trajectories for testing. Each trajectory consists of $T = 6$ past positions $\mathbf{x}^{(1:T)} := \{\mathbf{x}^{(t)}\}_{t=1}^{T}$ and $T = 6$ future positions $\mathbf{y}^{(1:T)} := \{\mathbf{y}^{(t)}\}_{t=1}^{T}$ to be predicted, and the positions are measured at a frequency of 2.5Hz. We assess the performance of the models trained for this task using the Average Distance Error (ADE) defined as follows:

$$\text{ADE}(\mathbf{y}^{(1:T)}, \hat{\mathbf{y}}^{(1:T)}) = \frac{1}{T}\sum_{t=1}^{T}\|\mathbf{y}^{(t)} - \hat{\mathbf{y}}^{(t)}\|, \tag{13}$$

where $\mathbf{y}^{(1:T)}$ and $\hat{\mathbf{y}}^{(1:T)}$ are predicted and ground-truth future trajectories, respectively.

In principle, the trajectory of a moving vehicle is equivariant to the rotations along the z-axis. Therefore, an $\text{O}^z(2)$-equivariant model is expected to perform better than non-equivariant models. Indeed, on the WOMD dataset, Assaad et al. (2022) reported that an $\text{O}^z(2)$-equivariant transformer works better than a non-equivariant transformer. However, they also reported that on the same task, the $\text{O}^z(2)$-equivariant transformer performs *worse* than a soft $\text{O}^z(2)$-equivariant transformer. In our experiment, we attempt to see why this is the case and also find out what other types of (approximate) symmetries the dataset might exhibit. To this end, we compare $\text{O}^z(2)$-EMLP, $\text{O}(3)$-EMLP, $\text{O}(3)$-RPP, $\text{O}^z(2)$-RPP, $\text{O}^z(2)$-$\text{O}(3)$ MEMLP, and MLP with $(\text{O}^x(2), \text{O}^y(2), \text{O}^z(2))$ PER.

4.2.2. Normalization Methods

Typically, for a regression problem, we preprocess the inputs either by normalizing or scaling them. However, we find that training with trajectories with such typical preprocessing performs poorly, due to high variance across trajectories. Hence, before the actual normalization, we first do *centering* for each trajectory to bring it near the origin. Given a i-th trajectory $\mathbf{x}_i^{(1:T)}$, the centering is defined as

$$\text{centering}(\mathbf{x}_i^{(1:T)}) = (\mathbf{x}_i^{(t)} - \bar{\mathbf{x}}_i) := \mathbf{c}_i^{(1:T)}, \tag{14}$$

where $\bar{\mathbf{x}}_i := \sum_{t=1}^{T}\mathbf{x}_i^{(t)}/T$.

Even after centering, we still suffer from varying scales of the coordinates (the values of the z-axis are significantly smaller than the values of the other axes because most vehicles run on horizontal roads). To resolve this, we may normalize each coordinate separately but it might also break the symmetry implied in the data. Hence, we consider three different types of normalization schemes where each scheme induces different (approximate) symmetry, and compare the models on the datasets preprocessed with them. The goal of the experiment is to show that our method can capture different types of symmetries induced by the normalizations and thus perform robustly across datasets. Examples of trajectory for each normalization are visually compared in Appendix E.

Scale-aware normalization. Assume we have N trajectories in the training set. Let $\boldsymbol{\mu} \in \mathbb{R}^3$ and $\boldsymbol{\sigma} \in \mathbb{R}^3_+$ be the element-wise mean and standard deviation of the trajectories in the training set,

$$\boldsymbol{\mu} = \sum_{i=1}^{N}\sum_{t=1}^{T}\frac{\mathbf{c}_i^{(t)}}{NT}, \quad \boldsymbol{\sigma} = \left(\sum_{i=1}^{N}\sum_{t=1}^{T}\frac{(\mathbf{c}_i^{(t)} - \boldsymbol{\mu})^{\odot 2}}{NT}\right)^{\odot\frac{1}{2}}, \tag{15}$$

where $\odot$ denotes the element-wise exponentiation. Given $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$, the first normalization scheme is defined as

$$\text{normalize}(\mathbf{c}_i^{(1:T)}) = \left((\mathbf{c}_i^{(t)} - \boldsymbol{\mu}) \oslash \boldsymbol{\sigma}\right)_{t=1}^{T}, \tag{16}$$

where $\oslash$ denotes the element-wise division. We call this normalization a *scale-aware* normalization, since it adjusts the data for each coordinate separately so that all the (x, y, z) coordinates have similar scales.

Symmetry-aware normalization. Note that the scale-aware normalization breaks the rotation symmetry because it scales each coordinate with a different value. In that case, we may lose the benefits of utilizing the rotation equivariance in a model. In the second normalization scheme, instead of element-wise scaling, we use the total standard deviation for the scaling:

$$m = \sum_{i=1}^{N}\sum_{t=1}^{T}\frac{\mathbf{1}_3^{\intercal}\mathbf{c}_i^{(t)}}{3NT}, \quad s^2 = \sum_{i=1}^{N}\sum_{t=1}^{T}\frac{\|\mathbf{c}_i^{(t)} - m\mathbf{1}_3\|^2}{3NT}$$

$$\text{normalize}(\mathbf{c}_i^{(1:T)}) = \left((\mathbf{c}_i^{(t)} - \boldsymbol{\mu})/s\right)_{t=1}^{T}, \tag{17}$$

where $\mathbf{1}_3 = [1, 1, 1]^{\intercal}$. We call this normalization *symmetry-aware* since the rotation symmetry of the resulting trajectory is not broken by the normalization.

Symmetry-scale-aware normalization. While the symmetry-aware normalization preserves the rotation symmetry, it still has the problem of a small z-scale in the training set. To further resolve this, as the third scheme, we modify the centering step as follows,

$$\text{centering}(\mathbf{x}_i^{(1:T)}) = (\mathbf{x}_i^{(t)} - \boldsymbol{\alpha} \otimes \bar{\mathbf{x}}_i)_{t=1}^{T}, \tag{18}$$

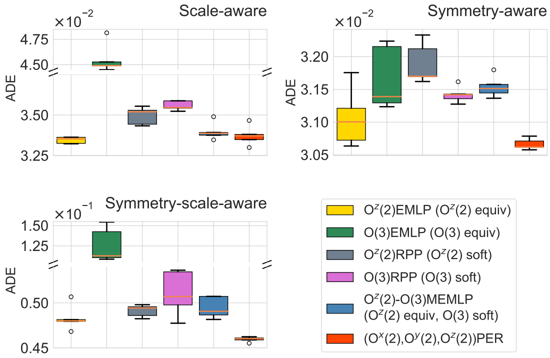


Figure 5: Test ADE results for WOMD dataset.

where $\otimes$ denotes the element-wise multiplication and $\boldsymbol{\alpha} \in \mathbb{R}^3$ is a scaling factor. We set $\boldsymbol{\alpha} = (1, 1, 0.993)$, so the values for the z-axis remain similar to the other axes after centering. Then we normalize the centered data as in the symmetry-aware normalization. Since the values of the z-axis were similar to those of other axes, even after the scaling, the values of the three coordinates have a similar scale. We call this scheme *symmetry-scale-aware* since it is both scale-aware and preserves rotation symmetry.

4.2.3. Analysis of the results

We expect that the scale-aware normalization breaks the $\mathrm{O}^z(2)$ equivariance because it normalizes the x and y axes with different scales, but the degree of approximate equivariance would not be serious because the x axis and the y axis have similar (but still different) scales. Indeed, Figure 5 shows that the models (approximately) equivariant to $\mathrm{O}^z(2)$, $\mathrm{O}^z(2)$-EMLP, $\mathrm{O}^z(2)$-$\mathrm{O}(3)$ MEMLP, and PER, perform better than the others. Interestingly, as can be seen in Figure 6, PER discovers that the data has soft $\mathrm{O}^z(2)$ equivariance, which coincides with our expectation that the scale-aware normalization mildly breaks the $\mathrm{O}^z(2)$ equivariance. Note that $\mathrm{O}^z(2)$-EMLP exhibits a tiny equivariance error. This is due to a numerical error in calculating the equivariant basis Q and R in Equation 2.

Even though the symmetry-aware normalization does not break the $\mathrm{O}(3)$ equivariance, the dataset itself has soft $\mathrm{O}(3)$ equivariance due to the gravity acting on the vehicles. However, the significantly small scale of z-coordinates in the symmetry-aware normalization causes a model to underestimate the $\mathrm{O}^x(2)$ and $\mathrm{O}^y(2)$ equivariance. Consequently, the small equivariance error discovered by PER led to the best performance. As shown in Figure 6, while $\mathrm{O}^z(2)$-EMLP captures only large equivariance errors on $\mathrm{O}^x(2)$ and $\mathrm{O}^y(2)$, PER captures small equivariance errors on $\mathrm{O}(3)$.

For the symmetry-scale-aware scheme, PER shows the best performance. As in the scale-aware, all models perform well except for the $\mathrm{O}(3)$-EMLP. Together with the captured equivariance errors in Figure 6, they explain the symmetry-scale-aware scheme is also softly $\mathrm{O}(3)$ equivariant. Whereas the element-wise scaling causes soft $\mathrm{O}(3)$ equivariance in the scale-aware, in the symmetry-scale-aware scheme, mainly the gravity acting on the vehicles along the z-axis results in the soft $\mathrm{O}(3)$ equivariance of the data. Moreover, the relatively large equivariance errors on $\mathrm{O}^z(2)$ (blue) helped the performance of PER, which was a coherent result with Assaad et al. (2022).

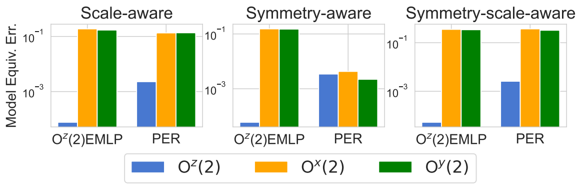


Figure 6: The model equivariance errors captured by $\mathrm{O}^z(2)$-EMLP and our algorithm.

To summarize, for all three normalization, PER robustly outperforms the baselines, and discovers reasonable soft symmetries.

5. Related Work

The translation equivariance of CNN and permutation equivariance of GNN are the most popular examples of symmetry built in the neural networks. Recently, there have been several works designing neural networks having desirable group equivariance. EMLP (Finzi et al., 2021b) is a framework that builds an MLP equivariant to various groups. LieConv (Finzi et al., 2020) is a variant of CNN targeting equivariance for Lie groups. Another variant called G-CNN (Cohen & Welling, 2016) is equivariant w.r.t. 90-degree rotations, reflections, and translation.

Most softly equivariant models impose architectural restrictions for the soft equivariance. RPP (Finzi et al., 2021a) build a soft equivariant model via a residual layer added to the equivariant linear layer, where the degree of equivariance is determined by the prior variances assigned for the equivariant layer and the residual pathway. Relaxed group convolution (Wang et al., 2022) implements a softly equivariant CNN by interpolating multiple conv operations with different weights, and the number of convolutions determines the degree of equivariance. Relaxed G-steerable group convolution (Wang et al., 2022) introduces spatial-location-dependent weights that replace the weights in the G-steerable CNN. Relaxed G- and G-steerable CNNs use group-action-based regularizers to restrict the relaxation.

There are some previous works allowing automatic symmetry discovery from data (Dehmamy et al., 2021; Krippendorf & Syvaeri, 2021). However, to our knowledge, ours is the first to discover the varying degrees of approximate equivariance across multiple groups under mixed symmetry settings.

6. Conclusion

In this paper, we tackle the learning problems under mixed symmetries, where a dataset contains multiple types of symmetries with different levels of equivariance errors. While previous methods focused on a single type of symmetries and bake in the equivariance constraint to the architecture as an inductive bias, ours take a regularizer-based approach, where a model without any equivariance constraint is regularized towards it using a projection-based regularization. One notable advantage is that it can automatically detect the levels of equivariance errors and adapt to those error levels by controlling the regularization coefficients. This is done during the training without any explicit supervision. Using a synthetic function approximation task and real-world motion forecasting task, we demonstrate that our proposed model could indeed capture mixed symmetries, identify the different level of equivariance errors, and predicts better than the existing methods. In this paper, we mainly focused on MLP architectures, so extending our framework to arbitrary neural network architectures such as CNNs, RNNs, or transformers (Vaswani et al., 2017) would be an interesting future research direction.

Acknowledgements

This work was partially supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2019-0-00075, Artificial Intelligence Graduate School Program (KAIST)), Artificial Intelligence Innovation Hub (No.2022-0-00713), and National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF2021M3E5D9025030). HY was supported by the Engineering Research Center Program through the National Research Foundation of Korea (NRF) funded by the Korean Government MSIT (NRF-2018R1A5A1059921) and also by the Institute for Basic Science (IBS-R029-C1). We are grateful to Seongho Keum, who helped us throughout the process of building up this work.

References

Assaad, S., Downey, C., Al-Rfou, R., Nayakanti, N., and Sapp, B. Vn-transformer: Rotation-equivariant attention for vector neurons, 2022. 7, 8

Bronstein, M. M., Bruna, J., Cohen, T., and Veličković, P. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021. 1

Cohen, T. and Welling, M. Group equivariant convolutional networks. In *Proceedings of The 33rd International Conference on Machine Learning (ICML 2016)*, 2016. 1, 8

Cohen, T., Geiger, M., Köhler, J., and Welling, M. Spherical CNNs. *arXiv preprint arXiv:1801.10130*, 2018. 1

Cohen, T., Weiler, M., Kicanaoglu, B., and Welling, M. Gauge equivariant convolutional networks and the icosahedral CNN. In *Proceedings of The 36th International Conference on Machine Learning (ICML 2019)*, 2019. 1

Dehmamy, N., Walters, R., Liu, Y., Wang, D., and Yu, R. Automatic symmetry discovery with Lie algebra convolutional network. In *Advances in Neural Information Processing Systems 34 (NeurIPS 2021)*, 2021. 9

Ettinger, S., Cheng, S., Caine, B., Liu, C., Zhao, H., Pradhan, S., Chai, Y., Sapp, B., Qi, C. R., Zhou, Y., Yang, Z., Chouard, A., Sun, P., Ngiam, J., Vasudevan, V., McCauley, A., Shlens, J., and Anguelov, D. Large scale interactive motion forecasting for autonomous driving : The waymo open motion dataset. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, 2021. 7

Finzi, M., Stanton, S., Izmailov, P., and Wilson, A. G. Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data. In *Proceedings of The 37th International Conference on Machine Learning (ICML 2020)*, 2020. 8

Finzi, M., Benton, G., and Wilson, A. G. Residual pathway priors for soft equivariance constraints. In *Advances in Neural Information Processing Systems 34 (NeurIPS 2021)*, 2021a. 2, 3, 5, 8

Finzi, M., Welling, M., and Wilson, A. G. A practical method for constructing equivariant multilayer perceptrons for arbitrary matrix groups. In *Proceedings of The 38th International Conference on Machine Learning (ICML 2021)*, 2021b. 2, 5, 8

Glorot, X. and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2010, Chia Laguna Resort, Sardinia, Italy, May 13-15, 2010*, 2010. 12

He, K., Zhang, X., Ren, S., and Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, 2015. 12

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 15

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2016. 1

Krippendorf, S. and Syvaeri, M. Detecting symmetries with neural networks. *Mach. Learn. Sci. Technol.*, 2021. 9

Loshchilov, I. and Hutter, F. SGDR: stochastic gradient descent with warm restarts. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017. 15

Puny, O., Atzmon, M., Smith, E. J., Misra, I., Grover, A., Ben-Hamu, H., and Lipman, Y. Frame averaging for invariant and equivariant network design. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*, 2022. 16

van der Ouderaa, T. F. A., Romero, D. W., and van der Wilk, M. Relaxing equivariance constraints with non-stationary continuous filters, 2022. 2

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, 2017. 9

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lió, P., and Bengio, Y. Graph attention networks. In *International Conference on Learning Representations (ICLR)*, 2018. 1

Wang, R., Walters, R., and Yu, R. Approximately equivariant networks for imperfectly symmetric dynamics, 2022. 2, 8

A. Proof of Proposition 3.1

Proof. We define notations V and $\dagger$ where V is a vectorized form of the weight W and $\dagger$ converts the vector form of a matrix back to the matrix form. i.e. $V = \mathrm{vec}(W)$ and $W = V^\dagger$. Also, we can utilize those identities by the definitions of Q_w and Q_b.

$$\rho_\mathcal{Y}(g)(Q_wQ_w^\top V)^\dagger = (Q_wQ_w^\top V)^\dagger \rho_\mathcal{X}(g), \qquad\qquad \rho_\mathcal{Y}(g)Q_bQ_b^\top b = Q_bQ_b^\top b. \tag{19}$$

Now we first prove Proposition 3.1 where f is a linear function. By the triangle inequality,

$$\begin{aligned}
&\|\rho_\mathcal{Y}(g)(W\mathbf{x}+b) - W\rho_\mathcal{X}(g)\mathbf{x} - b\| && (20)\\
&= \|\rho_\mathcal{Y}(g)W\mathbf{x} - \rho_\mathcal{Y}(g)(Q_wQ_w^\top V)^\dagger\mathbf{x} + \rho_\mathcal{Y}(g)b - \rho_\mathcal{Y}(g)Q_bQ_b^\top b && (21)\\
&\qquad - W\rho_\mathcal{X}(g)\mathbf{x} + (Q_wQ_w^\top V)^\dagger\rho_\mathcal{X}(g)\mathbf{x} - b + Q_bQ_b^\top b\| && (22)\\
&\le \|\rho_\mathcal{Y}(g)W\mathbf{x} - \rho_\mathcal{Y}(g)(Q_wQ_w^\top V)^\dagger\mathbf{x} + \rho_\mathcal{Y}(g)b - \rho_\mathcal{Y}(g)Q_bQ_b^\top b\| && (23)\\
&\qquad + \|W\rho_\mathcal{X}(g)\mathbf{x} - (Q_wQ_w^\top V)^\dagger\rho_\mathcal{X}(g)\mathbf{x} + b - Q_bQ_b^\top b\| && (24)\\
&\le \|\rho_\mathcal{Y}(g)\big(W - (Q_wQ_w^\top V)^\dagger\big)\mathbf{x}\| + \|\rho_\mathcal{Y}(g)(b - Q_bQ_b^\top b)\| && (25)\\
&\qquad + \|\big(W - (Q_wQ_w^\top V)^\dagger\big)\rho_\mathcal{X}(g)\mathbf{x}\| + \|b - Q_bQ_b^\top b\|. && (26)
\end{aligned}$$

We can split out $\rho(g)$ and $\mathbf{x}$ by using the operator norms and the operator norm is bounded by Frobenius norm.

$$\begin{aligned}
\|\rho_\mathcal{Y}(g)\big(W - (Q_wQ_w^\top V)^\dagger\big)\mathbf{x}\| &\le \|\rho_\mathcal{Y}(g)\|_{\mathrm{op}}\|W - (Q_wQ_w^\top V)^\dagger\|_F\|\mathbf{x}\|, && (27)\\
\|\rho_\mathcal{Y}(g)(b - Q_bQ_b^\top b)\| &\le \|\rho_\mathcal{Y}(g)\|_{\mathrm{op}}\|b - Q_bQ_b^\top b\|, && (28)\\
\|\big(W - (Q_wQ_w^\top V)^\dagger\big)\rho_\mathcal{X}(g)\mathbf{x}\| &\le \|W - (Q_wQ_w^\top V)^\dagger\|_F\|\rho_\mathcal{X}(g)\mathbf{x}\|. && (29)
\end{aligned}$$

Therefore, the G-equivariance error is bounded as follows:

$$\begin{aligned}
&\sup_{\mathbf{x},g}\|\rho_\mathcal{Y}(g)(W\mathbf{x}+b) - W\rho_\mathcal{X}(g)\mathbf{x} - b\| && (30)\\
&\le \big(\sup_g\|\rho_\mathcal{Y}(g)\|_{\mathrm{op}}\sup_\mathbf{x}\|\mathbf{x}\| + \sup_{\mathbf{x},g}\sup_g\|\rho_\mathcal{X}(g)\mathbf{x}\|\big)\|V - Q_wQ_w^\top V\| && (31)\\
&\qquad + \big(\|\rho_\mathcal{Y}(g)\|_{\mathrm{op}} + 1\big)\|b - Q_bQ_b^\top b\| && (32)\\
&= C_1\|V - Q_wQ_w^\top V\| + C_2\|b - Q_bQ_b^\top b\|. && (33)
\end{aligned}$$

The norm of $\mathbf{x}$ are supposed to be bounded since we have a finite dataset. Now we are looking at when f is a non-linear function whose activation σ is G-equivariant and L-lipchitz continuous. The equivariant activation σ has

$$\rho_\mathcal{Y}(g)\sigma(f(\mathbf{x})) = \sigma(\rho_\mathcal{Y}(g)f(\mathbf{x})) \tag{34}$$

for any function f. Hence, the G-equivariance error

$$\|\rho_\mathcal{Y}(g)\sigma(W\mathbf{x}+b) - \sigma(W\rho_\mathcal{X}(g)\mathbf{x} - b)\| = \|\sigma(\rho_\mathcal{Y}(g)(W\mathbf{x}+b)) - \sigma(W\rho_\mathcal{X}(g)\mathbf{x}+b)\|. \tag{35}$$

Since σ is L-lipchitz continuous,

$$\|\sigma(\rho_\mathcal{Y}(g)(W\mathbf{x}+b)) - \sigma(W\rho_\mathcal{X}(g)\mathbf{x}+b)\| \le L\|\rho_\mathcal{Y}(g)(W\mathbf{x}+b) - W\rho_\mathcal{X}(g)\mathbf{x} - b\|. \tag{36}$$

The r.h.s is the equivariance error when f is a linear function, which is bounded by Equation 33.

Lastly, we show the case when f is a two-layer MLP. More-than-two-layered MLPs can be shown in the same way. The G-equivariance error is

$$\begin{aligned}
&\|\rho^{(2)}(g)(W^{(2)}\sigma(W^{(1)}\mathbf{x}+b^{(1)})+b^{(2)}) - W^{(2)}\sigma(W^{(1)}\rho^{(0)}(g)\mathbf{x}+b^{(1)}) - b^{(2)}\| && (37)\\
&= \|\rho^{(2)}(g)(W^{(2)}\sigma(W^{(1)}\mathbf{x}+b^{(1)})+b^{(2)}) - W^{(2)}\rho^{(1)}(g)\sigma(W^{(1)}\mathbf{x}+b^{(1)}) - b^{(2)} &&\\
&\qquad + W^{(2)}\rho^{(1)}(g)\sigma(W^{(1)}\mathbf{x}+b^{(1)}) + b^{(2)} - W^{(2)}\sigma(W^{(1)}\rho^{(0)}(g)\mathbf{x}+b^{(1)}) - b^{(2)}\|. && (38)
\end{aligned}$$

This is bounded by an addition of two equivariance errors by triangle inequality

$$\begin{aligned}
&\|\rho^{(2)}(g)(W^{(2)}\sigma(W^{(1)}\mathbf{x}+b^{(1)})+b^{(2)})-W^{(2)}\rho^{(1)}(g)\sigma(W^{(1)}\mathbf{x}+b^{(1)})-b^{(2)}\| \\
&\qquad+\|W^{(2)}\rho^{(1)}(g)\sigma(W^{(1)}\mathbf{x}+b^{(1)})+b^{(2)}-W^{(2)}\sigma(W^{(1)}\rho^{(0)}(g)\mathbf{x}+b^{(1)})-b^{(2)}\| \qquad (39)\\
&\leq \|\rho^{(2)}(g)(W^{(2)}\mathbf{x}'+b^{(2)})-W^{(2)}\rho^{(1)}(g)\mathbf{x}'-b^{(2)}\| \\
&\qquad+\|W^{(2)}\|_{\text{op}}\|\rho^{(1)}(g)\sigma(W^{(1)}\mathbf{x}+b^{(1)})-\sigma(W^{(1)}\rho^{(0)}(g)\mathbf{x}+b^{(1)})\|, \qquad (40)
\end{aligned}$$

where $\mathbf{x}'=\sigma(W^{(1)}\mathbf{x}+b^{(1)})$. The first term of Equation 40 is the equivariance error of the linear function where the input is the output of the first layer. Besides, the second term involves the equivariance error of the non-linear function. Overall, the equivariance error of the two-layer MLP is bounded as

$$\begin{aligned}
&\sup_{\mathbf{x},g}\|\rho^{(2)}(g)(W^{(2)}\sigma(W^{(1)}\mathbf{x}+b^{(1)})+b^{(2)})-W^{(2)}\sigma(W^{(1)}\rho^{(0)}(g)\mathbf{x}+b^{(1)})-b^{(2)}\| \qquad (41)\\
&\leq \big(\sup_{g}\|\rho^{(2)}(g)\|_{\text{op}}\sup_{\mathbf{x}}\|\mathbf{x}'\|+\sup_{\mathbf{x},g}\|\rho^{(1)}(g)\mathbf{x}'\|\big)\|V^{(2)}-Q_{w^{(2)}}Q_{w^{(2)}}^{\intercal}V^{(2)}\| \qquad (42)\\
&\qquad+\big(\sup_{g}\|\rho^{(2)}(g)\|_{\text{op}}+1\big)\|b^{(2)}-Q_{b^{(2)}}Q_{b^{(2)}}^{\intercal}b^{(2)}\| \qquad (43)\\
&\qquad+L\|W^{(2)}\|_{\text{op}}\big(\sup_{g}\|\rho^{(1)}(g)\|_{\text{op}}\sup_{\mathbf{x}}\|\mathbf{x}\|+\sup_{\mathbf{x},g}\|\rho^{(0)}(g)\mathbf{x}\|\big)\|V^{(1)}-Q_{w^{(1)}}Q_{w^{(1)}}^{\intercal}V^{(1)}\| \qquad (44)\\
&\qquad+L\|W^{(2)}\|_{\text{op}}\big(\sup_{g}\|\rho^{(1)}(g)\|_{\text{op}}+1\big)\|b^{(1)}-Q_{b^{(1)}}Q_{b^{(1)}}^{\intercal}b^{(1)}\| \qquad (45)\\
&\leq C_1^{(2)}\|V^{(2)}-Q_{w^{(2)}}Q_{w^{(2)}}^{\intercal}V^{(2)}\|+C_2^{(2)}\|b^{(2)}-Q_{b^{(2)}}Q_{b^{(2)}}^{\intercal}b^{(2)}\| \qquad (46)\\
&\qquad+C_1^{(1)}\|V^{(1)}-Q_{w^{(1)}}Q_{w^{(1)}}^{\intercal}V^{(1)}\|+C_2^{(1)}\|b^{(1)}-Q_{b^{(1)}}Q_{b^{(1)}}^{\intercal}b^{(1)}\|. \qquad (47)
\end{aligned}$$

In terms of the mathematical induction, the bound of the equivariance error of more-than-two-layered MLPs can be derived as follows:

$$\begin{aligned}
&\sup_{\mathbf{x},g}[\|\rho^{(S)}(g)(W^{(S)}\sigma(f'(\mathbf{x}))+b^{(S)})-W^{(S)}\sigma(f'(\rho^{(0)}(g)\mathbf{x}))-b^{(S)}\|] \qquad (48)\\
&=\sup_{\mathbf{x},g}\|\rho^{(S)}(g)(W^{(S)}\sigma(f'(\mathbf{x}))+b^{(S)})-W^{(S)}\rho^{(1)}(g)\sigma(f'(\mathbf{x}))-b^{(S)} \\
&\qquad+W^{(S)}\rho^{(S-1)}(g)\sigma(f'(\mathbf{x}))+b^{(S)}-W^{(S)}\sigma(f'(\rho^{(0)}(S)\mathbf{x}))-b^{(S)}\| \qquad (49)\\
&\leq\sup_{\mathbf{x},g}\|\rho^{(S)}(g)(W^{(S)}\sigma(f'(\mathbf{x}))+b^{(S)})-W^{(S)}\rho^{(1)}(g)\sigma(f'(\mathbf{x}))-b^{(S)}\| \\
&\qquad+L\|W^{(S)}\|_{\text{op}}\sup_{\mathbf{x},g}\|\rho^{(S-1)}(g)f'(\mathbf{x})-f'(\rho^{(0)}(S)\mathbf{x})\|, \qquad (50)
\end{aligned}$$

where S is the number of layers and f' is a $(S-1)$-layered MLP. □

B. RPP vs. PER

Illustrated in Figure 7.

C. Weight Initializations for PER

Since our method does not restrict the parameter space, we can freely choose desirable strategies of weight initialization according to prior knowledge about a given task.

C.1. Standard

Obviously we can utilize the well-known initializations of neural networks such as Glorot initialization (Glorot & Bengio, 2010) and He initialization (He et al., 2015).

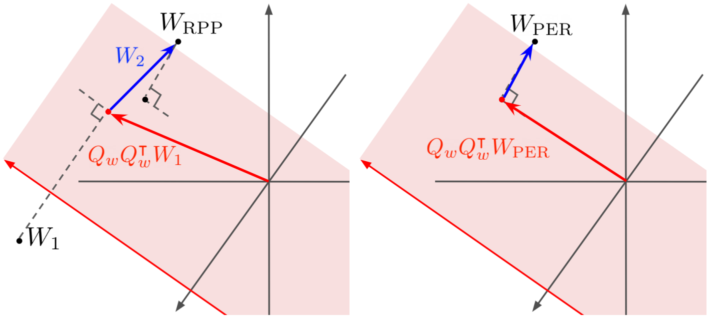


Figure 7: Comparison of parameterization bewteen RPP (left) and PER (right). W_1 and W_2 are the parameters of RPP. W_1 explains equivariance by projecting onto the equivariant space and W_2, called a residual path, captures the difference between approximate equivariance desired in dataset and strict equivariance of $Q_w Q_w^\intercal W_1$. On the other hand, W_{PER} does not require additional parameters because it is already close to the equivariant space due to the regularizer.

C.2. Soft

This initialization mimics the initial weights of the RPP model. The structure of the RPP models consists of addition of weights projected on the equivariant space $QQ^\intercal \text{vec}(W_1)$ and small weights $\text{vec}(W_2)$ acting as a perturbation of the equivariant weights.

$$\text{vec}(W_{\text{RPP}}) = QQ^\intercal \text{vec}(W_1) + \text{vec}(W_2), \qquad \text{vec}(W_1) \sim \mathcal{N}(0, \sigma^2 \mathbf{I}), \qquad \text{vec}(W_2) \sim \mathcal{N}(0, \epsilon\sigma^2 \mathbf{I}), \tag{51}$$

where $0 < \epsilon << 1$ and σ is determined by selected types of initialization such as Glorot and He. Thus, our model can be initialized with the added distribution as follows:

$$\text{vec}(W_{\text{PER}}) \sim \mathcal{N}(0, \sigma^2 QQ^\intercal + \epsilon\sigma^2 \mathbf{I}). \tag{52}$$

C.3. Half Soft

The degree of approximate equivariance is determined by the perpendicular distance from the equivariant space and the perpendicular distance is determined by an amount of the complementary direction of the equivariant space. i.e. the approximate equivariance degree of weight W is determined by $\tilde{Q}\tilde{Q}^\intercal \text{vec}(W)$ because $\text{vec}(W) = QQ^\intercal \text{vec}(W) + \tilde{Q}\tilde{Q}^\intercal \text{vec}(W)$, where $\tilde{Q}$ is the complementary basis of Q. Therefore, we can control the equivariance of the initial weights with a scaling factor λ as follows:

$$\text{vec}(W_{\text{PER}}) \sim \mathcal{N}(0, (1-\lambda)\sigma^2 QQ^\intercal + \lambda\sigma^2 \mathbf{I}) = \mathcal{N}(0, \sigma^2 QQ^\intercal + \lambda\sigma^2 \tilde{Q}\tilde{Q}^\intercal). \tag{53}$$

The case when $\lambda = 0$ corresponds to the initial weights of EMLP and the case when $\lambda = 1$ corresponds to the initial weights of MLP. We chose $\lambda = 0.5$ to locate the model in the middle between EMLP and MLP.

D. Samples for Measuring Correlation

Experiments for measuring the Pearson correlation are listed in Table 3.

E. Comparison of trajectory between the normalizations

3 example trajectories (red, green, and blue) for each normalization are described in Figure 8.

F. Experimental Details

F.1. Dataset Description

Information for each dataset is summarized in Table 4.

Table 3: Samples for measuring the correlation with the data equivariance error. $\epsilon_1 = -\mathcal{I}\hat{x}\hat{x}^\intercal$, $\epsilon_2 = -\mathcal{I}\hat{y}\hat{y}^\intercal$, $\epsilon_3 = -\mathcal{I}\hat{z}\hat{z}^\intercal$, and $\epsilon_4 = -\mathcal{I}\hat{x}\hat{x}^\intercal + \mathcal{I}\hat{y}\hat{y}^\intercal - \mathcal{I}\hat{z}\hat{z}^\intercal$

	Data Equiv. Err.			Equiv. Regular. Coeff.			Model Equiv. Err.			Equiv. Regular.		
Noise	$\mathrm{O}^z(2)$	$\mathrm{O}^x(2)$	$\mathrm{O}^y(2)$	$\mathrm{O}^z(2)$	$\mathrm{O}^x(2)$	$\mathrm{O}^y(2)$	$\mathrm{O}^z(2)$	$\mathrm{O}^x(2)$	$\mathrm{O}^y(2)$	$\mathrm{O}^z(2)$	$\mathrm{O}^x(2)$	$\mathrm{O}^y(2)$
0	7.22E-08	7.23E-08	7.38E-08	1.00E+02	9.80E+01	9.34E+01	9.17E-04	7.72E-04	7.60E-04	1.46E-07	1.32E-07	1.31E-07
$0.3\epsilon_1$	5.89E-02	7.05E-08	5.85E-02	1.35E+01	1.00E+02	1.32E+01	5.30E-02	4.27E-03	5.42E-02	7.06E-05	9.95E-07	7.01E-05
$0.6\epsilon_1$	1.05E-01	7.11E-08	1.04E-01	1.22E+01	1.00E+02	1.29E+01	7.56E-02	8.09E-03	7.95E-02	1.19E-04	4.78E-06	1.18E-04
$0.9\epsilon_1$	1.41E-01	6.90E-08	1.40E-01	1.97E+01	1.00E+02	1.98E+01	1.36E-01	3.78E-03	1.34E-01	9.61E-05	2.20E-06	9.53E-05
$0.3\epsilon_2$	5.99E-02	5.34E-02	7.43E-08	7.54E+00	7.36E+00	1.00E+02	4.54E-02	5.06E-02	9.64E-03	1.85E-04	1.84E-04	1.57E-05
$0.6\epsilon_2$	1.08E-01	9.74E-02	7.22E-08	2.64E+01	2.62E+01	1.00E+02	8.13E-02	8.71E-02	1.73E-02	1.28E-04	1.27E-04	2.62E-05
$0.9\epsilon_2$	1.47E-01	1.34E-01	7.15E-08	9.61E+00	9.86E+00	1.00E+02	1.08E-01	1.10E-01	7.21E-03	2.44E-04	2.44E-04	1.96E-05
$0.3\epsilon_3$	7.04E-08	5.34E-02	5.96E-02	1.00E+02	3.36E+00	3.39E+00	1.67E-03	5.55E-02	5.56E-02	7.64E-07	1.41E-04	1.41E-04
$0.6\epsilon_3$	7.05E-08	9.75E-02	1.08E-01	1.00E+02	4.74E+00	4.96E+00	2.02E-03	9.61E-02	9.47E-02	2.87E-06	2.04E-04	2.05E-04
$0.9\epsilon_3$	6.88E-08	1.34E-01	1.47E-01	1.00E+02	4.19E+00	4.21E+00	1.87E-03	1.39E-01	1.40E-01	2.40E-06	2.65E-04	2.65E-04
$0.3\epsilon_4$	1.32E-01	5.91E-02	6.70E-02	1.79E+01	9.73E+01	1.00E+02	1.17E-01	5.13E-02	6.59E-02	9.04E-05	3.46E-05	3.39E-05
$0.6\epsilon_4$	2.50E-01	1.14E-01	1.31E-01	3.68E+01	9.62E+01	1.00E+02	1.96E-01	9.09E-02	1.06E-01	1.53E-04	5.79E-05	5.92E-05
$0.9\epsilon_4$	3.41E-01	1.58E-01	1.85E-01	3.29E+01	5.17E+01	1.00E+02	3.07E-01	1.39E-01	1.75E-01	1.86E-04	6.21E-05	5.86E-05

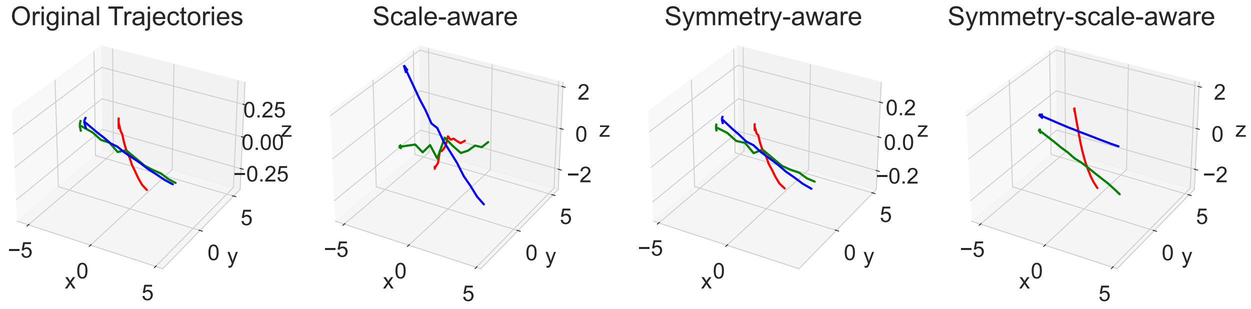


Figure 8: The scale-aware normalization strongly emphasizes the z-coordinates. The symmetry-aware normalization just scales down the whole coordinates but the scale of z-coordinates is still close to zero. The symmetry-scale-aware normalization scales down the whole coordinates while retaining the scale of z-coordinates.

F.2. Data Selection of WOMD

Trajectory Slicing The WOMD dataset contains maximum 91 points of a trajectory measured in 10Hz. We sliced and gathered first 24 points and dropped every even-numbered points so that the final trajectory contains only 12 points. The past 6 points and future 6 points are regarded as input and output, respectively.

Trajectory Selection We selected only a portion of the whole WOMD dataset. The training part of WOMD motion forecasting dataset consists of total 1,000 files of the `TFRecord` format. We used first 28 files as training set, next 6 files as validation set, and last 6 files as testing set only. Furthermore, we excluded all trajectory that doesn't move enough and move too far, we collected only trajectories satisfying the following conditions:

$$\|\mathbf{y}^{t=6} \cdot \hat{x} - \mathbf{y}^{t=1} \cdot \hat{x}\|_2 < 5 \tag{54}$$

$$\|\mathbf{y}^{t=6} \cdot \hat{y} - \mathbf{y}^{t=1} \cdot \hat{y}\|_2 < 5 \tag{55}$$

$$\|\mathbf{y}^{t=6} \cdot \hat{z} - \mathbf{y}^{t=1} \cdot \hat{z}\|_2 > 0.05, \tag{56}$$

where $\hat{x}, \hat{y}, \hat{z} \in \mathbb{R}^3$ are orthonormal basis vectors of x-,y-, and z-axes.

F.3. Details of Training

For all experiments, we used five different seeds to report performance results.

Architecture Description All architectures of the neural networks including EMLP, RPP, Mixed RPP, and our model are fixed with 4 layers and different width. Their widths were adjusted to set their number of parameters the same.

Table 4: Information of each dataset. $\mathbf{S}$ denotes a scalar and $\mathbf{V}$ denotes a vector in $\mathbb{R}^3$.

	Inertia	CosSim	WOMD
Training Samples	1,000	1,000	16,814
Validation Samples	1,000	1,000	3,339
Testing Samples	1,000	1,000	3,563
Input Representation	$5\mathbf{S} \oplus 5\mathbf{V}$	$3\mathbf{V}$	$6\mathbf{V}$
Output Representation	$\mathbf{V}^2$	$\mathbf{S}$	$6\mathbf{V}$

Hyperparameters See Table 5 for hyperparameter settings of our model and baseline methods for all experiments. Those hyperparameters are applied the same for all models. Additional hyperparameters of our model for each task are listed in Table 6.

Table 5: Common hyperparameter settings for each task.

Dataset	Mini-batch	Max Epochs	Learning Rate	Weight Decay	Width (RPP)
Inertia	500	8,000	0.001	2.0×10^{-4}	384 (270)
CosSim	200	10,000	0.0002	2.0×10^{-5}	128 (45)
WOMD (scale-aware)	256	750	0.0002	0	384 (269)
WOMD (symmetry-aware)	256	500	0.0002	0	384 (269)
WOMD (symmetry-scale-aware)	256	500	0.0002	0	384 (269)

Table 6: Hyperparameter setting of our model.

Dataset	Task Type	Initial λ	γ	Adjustment Epoch	Initialization	Mini-batch	Max Epochs
Inertia	$\mathrm{O}(3)$	100	2	2,000	Standard	500	8,000
	$\mathrm{O}^x(2)$	100	2	2,000	Standard	500	8,000
	$\mathrm{O}^y(2)$	100	2	2,000	Standard	500	8,000
	$\mathrm{O}^z(2)$	100	2	2,000	Standard	500	8,000
	Only Soft	100	2	2,000	Standard	500	8,000
CosSim	$\mathrm{SO}(3) \cup \mathrm{S}(3)$	0.005	2	2,500	Standard	200	10,000
	$\mathrm{SO}(3)$	0.1	2	2,500	Standard	200	10,000
	$\mathrm{S}(3)$	0.01	2	2,500	Standard	200	10,000
	Only Soft	0.005	2	2,500	Standard	200	10,000
WOMD	Scale-aware	0.2	5	125	Half Soft	128	500
	Symmetry-aware	0.3	5	100	Half Soft	128	500
	Symmetry-scale-aware	5	5	100	Half Soft	128	500

Extra Details We applied the cosine decaying of learning rate (Loshchilov & Hutter, 2017) and early stopping with 50 patience for stable training. The optimizers used in every tasks are ADAM (Kingma & Ba, 2015). All experiments were trained and evaluated on RTX 3090 devices.

G. Additional Experiments

G.1. Analysis of Adjustment of Hyperparameters

We share a part of the robustness analysis across different hyperparameters (initial coefficients λ, scaling factors γ, and the moments of the adjustment) required in the automatic tuning procedure described in § 3.2. As the results show in Table 7, we found that the performance of the model is not so sensitive to the choice of hyperparameters. For instance, for the initial value of lambda, we observed that the model would achieve similar performances provided that the ratio of the initial *Loss* over the initial $\lambda \cdot \mathcal{R}^{\text{PER}}$ is at a certain level. The situation for the scaling factor γ is similar: the final performance was consistent for the values arbitrarily chosen within the range [2, 5].

Table 7: Test MSE results across different hyperparameters required in the automatic PER-coefficients-tuning precedure described in § 3.2. λ is the initial coefficients and γ is the scaling factors.

(a) Inertia $\text{O}^z(2)$ task

loss/($\lambda \cdot \mathcal{R}^{\text{PER}}$)	Test MSE
0.00009	1.75±2.27
0.00037	0.32±0.26
0.00147	0.35±0.19
O^z(2)EMLP	1.56±0.17

(b) CosSim S(3) task

Loss/($\lambda \cdot \mathcal{R}^{\text{PER}}$)	Test MSE
0.0348	0.068±0.007
0.1741	0.065±0.009
0.8707	0.052±0.013
S(3)EMLP	0.21±0.11

(c) Inertia $\text{O}^z(2)$ task

γ	Test MSE
2	0.32±0.26
3	0.40±0.24
4	0.35±0.15
5	0.43±0.21
O^z(2)EMLP	1.56±0.17

(d) CosSim S(3) task

γ	Test MSE
2	0.065±0.009
3	0.044±0.004
4	0.044±0.004
5	0.044±0.004
S(3)EMLP	0.214±0.110

(e) Inertia $\text{O}^z(2)$ task (training epochs 8000)

Adjusted Epoch	Test MSE
1000	0.26±0.17
2000	0.32±0.26
3000	0.38±0.24
O^z(2) EMLP	1.56±0.17

(f) CosSim S(3) task (training epochs 2000)

Adjusted Epoch	Test MSE
300	0.044±0.003
500	0.065±0.009
700	0.046±0.004
S(3) EMLP	0.214±0.110

G.2. Comparison with Frame Averaging

Frame Averaging (FA) (Puny et al., 2022) is a framework that, in simple terms, trains a G-equivariant model f by taking the average over some group elements in G, called frames. FA is a flexible approach since it does not restrict the internal structure of the model f, unlike EMLP.

We ran additional experiments with FA on the fully-equivariant task same as the first row in Table 1. Table 8 shows the results. Although EMLP in the table used gated nonlinearity (GNL) due to its architectural restriction, FA does not need such a restriction, so the ”MLP w/ FA” row in Table 8 applied the frame averaging to the same setup as the MLP row (i.e., MLP with the Swish activation).

Our results confirmed, FA is indeed a more powerful baseline than EMLP. But note that our model (PER) performs better than MLP w/ FA here.

Table 8: Test MSE comparison with FA in the Inertia O(3) task

Model	Test MSE
MLP	4.25±0.17
EMLP	1.13±0.36
RPP	2.66±1.43
PER	**0.27±0.23**
MLP w/ FA	0.36±0.05

G.3. Simple Experiment Assuming Symmetries Are Unknown in the WOMD task

We explain an additional experiment where we mimic the situation of unknown symmetries by including various and sometimes wrong matrix groups as candidate groups and checking whether our method picks the correct groups. Table 9 shows the model equivariance error captured by the model when using all O(2), SL(2), and GL(2) PERs to train the motion forecasting task with symmetry-aware normalization (this task has symmetries with respect to $\text{O}^z(2)$ and $\text{O}^x(2)$). As shown in the tables, our method has appropriately captured the equivariance with respect to $\text{O}^z(2)$ and $\text{O}^x(2)$.

Table 9: (a) Captured equivariance errors across the prescribed regularizers with different groups (b) Change of Test MSE due to the additional regularizers ($\text{SL}^z(2)$, $\text{SL}^y(2)$, $\text{GL}^x(2)$, and $\text{GL}^y(2)$).

(a) Model equivariance errors

Regularized Groups	Model Equiv. Err.
$\text{O}^z(2)$	0.0007
$\text{O}^x(2)$	0.0006
$\text{SL}^z(2)$	0.2638
$\text{SL}^y(2)$	0.2302
$\text{GL}^x(2)$	0.2398
$\text{GL}^y(2)$	0.2089

(b) Test MSE

Models	Test MSE ($\times 10^{-2}$)
O(2) PER	3.07±0.01
O(2),SL(2),GL(2) PER	3.09±0.01