Relaxed Equivariance via Multitask Learning

Ahmed A. Elhag
University of Oxford
ahmed.elhag@cs.ox.ac.uk

T. Konstantin Rusch
ELLIS Institute Tübingen &
Max Planck Institute for Intelligent Systems &
Tübingen AI Center
tkrusch@tue.ellis.eu

Francesco Di Giovanni
Valence Labs
francesco.di.giovanni@valencelabs.com

Michael Bronstein
University of Oxford / AITHYRA
michael.bronstein@cs.ox.ac.uk

Abstract

Incorporating equivariance as an inductive bias into deep learning architectures to take advantage of the data symmetry has been successful in multiple applications, such as chemistry and dynamical systems. In particular, roto-translations are crucial for effectively modeling geometric graphs and molecules, where understanding the 3D structures enhances generalization. However, strictly equivariant models often pose challenges due to their higher computational complexity. In this paper, we introduce REMUL, a training procedure that learns *approximate* equivariance for unconstrained networks via multitask learning. By formulating equivariance as a tunable objective alongside the primary task loss, REMUL offers a principled way to control the degree of approximate symmetry, relaxing the rigid constraints of traditional equivariant architectures. We show that unconstrained models (which do not build equivariance into the architecture) can learn approximate symmetries by minimizing an additional simple equivariance loss. This enables quantitative control over the trade-off between enforcing equivariance constraints and optimizing for task-specific performance. Our method achieves competitive performance compared to equivariant baselines while being significantly faster (up to $10\times$ at inference and $2.5\times$ at training), offering a practical and adaptable approach to leveraging symmetry in unconstrained architectures.

1 Introduction

Equivariant machine learning models have achieved notable success across various domains, such as computer vision [1, 2], dynamical systems [3, 4], chemistry [5, 6], and structural biology [7]. For example, incorporating equivariance *w.r.t.* translations and rotations ensures the correct handling of complex structures like graphs and molecules [8–11]. Equivariant machine learning models benefit from this inductive bias by *explicitly* leveraging symmetries of the data during the architecture design. Typically, such architectures have highly constrained layers with restrictions on the form and action of weight matrices and nonlinear activations [12, 13]. This may come at the expense of higher computational cost, making it sometimes challenging to scale equivariant architectures, particularly those relying on spherical harmonics and irreducible representations [14–17]. On the other hand, equivariance constraints might limit the expressive power of the network, restricting its ability to act as a universal architecture [18].

Equivariant layers are not the only way to incorporate symmetries into deep neural networks. Several approaches have been proposed to either offload the equivariance restrictions to faster networks [19–23] or simplify the constraints by introducing averaging operations [24–27]. Nonetheless, while these approaches leverage unconstrained architectures, they often require additional networks or

Elhag et al., Relaxed Equivariance via Multitask Learning. *Proceedings of the Fourth Learning on Graphs Conference (LoG 2025)*, PMLR 338, Arizona State University, Phoenix, USA, December 10–12, 2025.

averaging techniques to achieve equivariance and may not rely solely on adjustments to the training protocol. To this aim, a widely adopted strategy to replace 'hard' equivariance (i.e., built into the architecture itself) with a 'soft' one, is *data augmentation* [28–36], whereby the training protocol of an arbitrary (unconstrained) network is augmented by assigning the same label to group orbits (e.g., rotated and translated versions of the input). In fact, recent works have shown that unconstrained architectures may offer a valid alternative, provided that enough data are available [37, 38].

Besides the challenges in computational cost and design, there are also tasks (especially in scientific applications of ML) that do not exhibit full equivariance, such as dynamical phase transitions [39, 40], polar fluids [41], molecular nanocrystals [42], and cellular symmetry breaking [43, 44]. For such tasks, fully-equivariant networks might be excessively constrained, which further motivates the design of a more flexible approach.

In this work, we present **REMUL**: **R**elaxed **E**quivariance via **Mu**ltitask **L**earning. REMUL is a training procedure that aims to learn approximate equivariance during training for unconstrained networks using a multitask approach with adaptive weights. We conduct a comprehensive evaluation of unconstrained models trained with REMUL, comparing their performance and computational efficiency to equivariant models. We consider Transformers and Graph Neural Networks (GNNs) and their roto-translational (E(3))-equivariant versions as our main baselines. Our contributions are:

- We formulate equivariance as a weighted multitask learning objective for unconstrained models, aiming to simultaneously learn the objective function and approximate the required equivariance associated with the data and the task.
- We demonstrate that by adjusting the weighting of the equivariance loss, we can modulate the extent to which a model exhibits equivariance, depending on the task's requirements. Specifically, tasks that demand full equivariance require a higher weight on the equivariance term, whereas tasks that require less strict equivariance can be managed with lower weights.
- Empirically, we show that Transformers and Graph Neural Networks trained with our multitask learning approach compete or outperform their equivariant counterparts.
- By leveraging the efficiency of Transformers, we achieve up to $10\times$ speed-up at inference and $2.5\times$ speed-up in training compared to equivariant baselines. This finding could provide motivations for the use of unconstrained models, which do not require equivariance in their design, potentially offering a more practical approach.
- We point out that the standard Transformer exhibits a more convex loss surface near the local minima compared to the Geometric Algebra Transformer [45], which can indicate further evidence of the optimization difficulties of equivariant networks.

2 Background

2.1 Symmetry Groups and Equivariant Models

Symmetry groups, a fundamental concept in abstract algebra and geometry, are a mathematical description of the properties of an object remaining unchanged (invariant) under a set of transformations. Formally, a symmetry group G of a set X is a group of bijective functions from X to itself, where the group operation is function composition.

Equivariant machine learning models are designed to preserve the symmetries associated with the data and the task. In geometric deep learning (GDL), the data is typically assumed to live on some geometric domain (e.g., a graph or a grid) that has an appropriate symmetry group (e.g., permutation or translation) associated with it. Equivariant models implement functions $f : X \to Y$ from input domain X to output domain Y that ensure the actions of a symmetry group G on data from X correspond systematically to its actions on Y, through the respective group representations ϕ and ρ. Formally, we say that:

Definition 1. *A function f is* equivariant *w.r.t. the group G if for any transformation $g \in G$ and any input $x \in X$,*

$$f(\phi(g)(x)) = \rho(g)(f(x)) \tag{1}$$

The group representations ϕ and ρ allow us to apply abstract objects (elements of the group G) on concrete input and output data, in the form of appropriately defined linear transformations. For

example, if $G = S_n$ (a permutation group of n elements, arising in learning on graphs with n nodes), its action on n-dimensional vectors (e.g., graph node features or labels) can be represented as an $n \times n$ permutation matrix.

A special case of equivariance is obtained for a trivial output representation $\rho = \mathrm{id}$:

Definition 2. *A function f is* invariant *w.r.t. the group G if for all $g \in G$, $x \in X$: $f(\phi(g)(x)) = f(x)$.*

2.2 Equivariance as a Constrained Optimization Problem

Consider a class of parametric functions f_θ from a hypothesis space $\mathcal{H}$, typically implemented as neural networks, whose parameters θ are estimated via a general training objective based on data pairs $(x, y) \sim q$:

$$\underset{\theta}{\text{minimize}} \quad \mathbb{E}_{(x,y)\sim q}\left[\mathcal{L}(f_\theta(x), y)\right] \tag{2}$$

Here, $\mathcal{L}$ represents the loss function that quantifies the discrepancy between the model's predictions $f_\theta(x)$ and the true labels y. The class of models is considered equivariant with respect to a group G if it satisfies the constraint in Eq. 1 for any input $x \in X$ and for any action $g \in G$.

Equivariance is typically achieved *by design*, by imposing constraints on the form of f_θ. Since f_θ is usually composed of multiple layers, ensuring equivariance implies restrictions on the operations performed in each layer, a canonical example being message-passing graph neural networks whose local aggregations need to be permutation-equivariant to respect the overall invariance to the action of the symmetric group S_n. As such, finding an equivariant solution to the minimization problem in Eq. 2 corresponds to solving the following constrained optimization:

$$\begin{aligned} \underset{\theta}{\text{minimize}} \quad & \mathbb{E}_{(x,y)\sim q}\left[\mathcal{L}(f_\theta(x), y)\right] \\ \text{subject to} \quad & f_\theta(\phi(g)(x)) = \rho(g) f_\theta(x),\ \forall g \in G,\ \forall x \in X \end{aligned} \tag{3}$$

In general, such optimization is challenging, leading to complex design choices to enforce equivariance that could ultimately restrict the class of minimizers and make the training harder. Additionally, for relevant tasks, the optimal solution only needs to be approximate equivariant [46–49] meaning that the extent to which a model needs to exhibit equivariance can vary significantly based on the specific characteristics of the data and the requirements of the downstream application. In light of these reasons, we require a flexible approach to incorporating equivariance into the learning process. To address this, we propose REMUL, a training procedure that replaces the hard optimization problem with a soft constraint, by using a multitask learning approach with adaptive weights.

3 REMUL Training Procedure

3.1 Equivariance as a New Learning Objective

Our main idea is to formulate equivariance as a multitask learning problem for an unconstrained model. We achieve that by *relaxing* the optimization problem in Eq. 3. Namely, once we introduce a functional $\mathcal{F}_{\mathcal{X},G}$ that measures the equivariance of a candidate function f_θ, we replace the constrained variational problem in Eq. 3 with

$$\underset{\theta}{\text{minimize}}\ \mathbb{E}_{(x,y)\sim q}\left[\alpha\mathcal{L}(f_\theta(x), y) + \beta\mathcal{F}_{\mathcal{X},G}(f_\theta(x), y)\right], \tag{4}$$

where $\alpha, \beta > 0$. This decomposition allows for tailored learning dynamics where the supervised loss specifically addresses the information from the dataset without constraining the solution f_θ, while the equivariance penalty $\mathcal{F}$ smoothly enforces symmetry preservation.

Empirical Formulation. Let $\mathcal{D}_n = \{(x_i, y_i)\}_{i=1}^n$ be a training sample of size n drawn i.i.d. from an underlying distribution P_{XY} on $\mathcal{X} \times \mathcal{Y}$. In conventional supervised settings, we define the empirical version of our optimization problem as:

$$\mathcal{L}_{\text{total}}(f_\theta, \mathcal{X}, \mathcal{Y}, G) = \alpha\widehat{\mathcal{L}}_{\text{obj}}(f_\theta, \mathcal{X}, \mathcal{Y}) + \beta\widehat{\mathcal{L}}_{\text{equi}}(f_\theta, \mathcal{X}, \mathcal{Y}, G), \tag{5}$$

where $\widehat{\mathcal{L}}_{\text{obj}}(f_\theta, \mathcal{X}, \mathcal{Y})$ is the empirical objective loss given by $\widehat{\mathcal{L}}_{\text{obj}}(f_\theta, \mathcal{X}, \mathcal{Y}) = \frac{1}{n}\sum_{i=1}^n \mathcal{L}(f_\theta(x_i), y_i)$, and $\widehat{\mathcal{L}}_{\text{equi}}(f_\theta, \mathcal{X}, \mathcal{Y}, G)$ represents our *augmented equivariance loss*,

specifically designed to enforce the model's adherence to the symmetry action of the group G. For a finite number of training samples n, we propose an empirical equivariant loss $\widehat{\mathcal{L}}_{\text{equi}}$ of the form:

$$\widehat{\mathcal{L}}_{\text{equi}}(f_\theta, \mathcal{X}, \mathcal{Y}, G) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{g\sim G}\Big[\ell(f_\theta(\phi(g)x_i),\ \rho(g)y_i)\Big] \tag{6}$$

here ℓ is a metric function, typically an L_1 or L_2 norm, that quantifies the discrepancy between $f(\phi(g)(x_i))$ and $\rho(g)(y_i)$, with $g \in G$ randomly-selected group elements drawn from a uniform distribution for each sample. In our implementation, we enhance computational efficiency by selecting a single group element per sample at each training step, which we found produces effective results. In addition, we show how the performance varies as we increase the number of group samples.

Characterizing the REMUL Trade-off. While REMUL is presented as a practical training procedure, it can be theoretically understood as a regularized optimization problem. The parameters α and β defined in Eq. 5 are weighting factors that balance the traditional objective loss with the equivariance loss, enabling practitioners to tailor the training process according to specific requirements of symmetry and generalization. The following proposition characterizes the properties of the empirical minimizer $f_{\alpha,\beta}$ and the underlying trade-offs between task performance and equivariance. The proof is provided in Appendix B.

Proposition 1. *Let $f_{\alpha,\beta} \in \arg\min_{f\in\mathcal{H}} \widehat{\mathcal{L}}_{\text{total}}(f;\alpha,\beta)$ be an empirical minimizer of the* REMUL *objective, and let $f^\star_{\text{obj}} \in \arg\min_{f\in\mathcal{H}} \widehat{\mathcal{L}}_{\text{obj}}(f)$ be an empirical minimizer for the objective loss alone. Then:*

(a) $f_{\alpha,\beta}$ is Pareto optimal for the bi-objective problem $(\min \widehat{\mathcal{L}}_{\text{obj}}(f), \min \widehat{\mathcal{L}}_{\text{equi}}(f))$.

(b) The following trade-off inequality holds:

$$\widehat{\mathcal{L}}_{\text{obj}}(f_{\alpha,\beta}) - \widehat{\mathcal{L}}_{\text{obj}}(f^\star_{\text{obj}}) \le \frac{\beta}{\alpha}\left(\widehat{\mathcal{L}}_{\text{equi}}(f^\star_{\text{obj}}) - \widehat{\mathcal{L}}_{\text{equi}}(f_{\alpha,\beta})\right). \tag{7}$$

Controlling Approximate Equivariance via β/α. Eq. 7 quantifies the empirical cost of enforcing equivariance, showing that any increase in primary task's loss beyond the unconstrained minimum $\widehat{\mathcal{L}}_{\text{obj}}(f^\star_{\text{obj}}))$ is bounded by the product of relative weight β/α and the achieved reduction in the equivariance loss (from $\widehat{\mathcal{L}}_{\text{equi}}(f^\star_{\text{obj}})$ down to $\widehat{\mathcal{L}}_{\text{equi}}(f_{\alpha,\beta})$). The ratio β/α serves as a lever to control the solution's properties: When $\beta/\alpha \to 0$, the objective prioritizes task performance, causing $f_{\alpha,\beta}$ to approximate $f^\star_{\text{obj}}$ (potentially sacrificing equivariance if $f^\star_{\text{obj}}$ lacks natural symmetry). In contrast, when $\beta/\alpha \to \infty$, the objective prioritizes equivariance, driving $\widehat{\mathcal{L}}_{\text{equi}}(f_{\alpha,\beta})$ toward zero (at the cost of task performance). Finally, at intermediate β/α values, the solution $f_{\alpha,\beta}$ represents a specific balance on the empirical Pareto frontier. REMUL thus allows learning a tunable degree of approximate equivariance, with larger β producing a more equivariant function, while smaller β produces a less equivariant function. This flexibility allows us to control the trade-off between model generalization & equivariance based on the task's requirements, as we demonstrate empirically in Section 6.

3.2 Adapting Penalty Parameters during Training

For simultaneously learning the objective and equivariance losses, we consider two distinct approaches to regulate the penalty parameters α and β: *constant* penalty and *gradual* penalty. The constant penalty assigns a fixed weight to each task's loss throughout the training process. In contrast, the gradual penalty dynamically adjusts the weights of each task's loss during training. For gradual penalty, we use the GradNorm algorithm introduced by [50], which is particularly suited for tasks that involve simultaneous optimization of multiple loss components, as it dynamically adjusts the weight of each loss during training. It updates the weights of the loss components based on the magnitudes of their gradients, *w.r.t* the last layer in the network, which is essential for the contribution of each loss. It also has a learning rate parameter η, that fine-tunes the speed at which the weights are updated, providing precise control over their convergence rates (see Algorithm 1 for details).

Equivariance with Data Augmentation. Standard data augmentation for enforcing equivariance typically involves augmenting the training data with pairs $(\phi(g)(x_i), \rho(g)(y_i))$ and training the

Algorithm 1 GradNorm Algorithm (one step)

Input: α, β, η, γ, $\widehat{\mathcal{L}}_{\text{obj}}$, $\widehat{\mathcal{L}}_{\text{equi}}$, and $\mathcal{W}$ (the weights of the last layer in the network)
$\mathcal{G}_{\text{obj}} = \|\nabla_{\mathcal{W}} \alpha \widehat{\mathcal{L}}_{\text{obj}}\|_2$, $\tilde{\mathcal{L}}_{\text{obj}} = \widehat{\mathcal{L}}_{\text{obj}} / \widehat{\mathcal{L}}_{\text{obj}}(0)$
$\mathcal{G}_{\text{equi}} = \|\nabla_{\mathcal{W}} \beta \widehat{\mathcal{L}}_{\text{equi}}\|_2$, $\tilde{\mathcal{L}}_{\text{equi}} = \widehat{\mathcal{L}}_{\text{equi}} / \widehat{\mathcal{L}}_{\text{equi}}(0)$
$\bar{\mathcal{G}} = \frac{\mathcal{G}_{\text{obj}} + \mathcal{G}_{\text{equi}}}{2}$, $r = \frac{\tilde{\mathcal{L}}_{\text{obj}} + \tilde{\mathcal{L}}_{\text{equi}}}{2}$
$r_\alpha = \frac{\tilde{\mathcal{L}}_{\text{obj}}}{r}$, $r_\beta = \frac{\tilde{\mathcal{L}}_{\text{equi}}}{r}$
$\widehat{\mathcal{L}}_{\text{g}} = |\mathcal{G}_{\text{obj}} - \bar{\mathcal{G}} \times [r_\alpha]^\gamma| + |\mathcal{G}_{\text{equi}} - \bar{\mathcal{G}} \times [r_\beta]^\gamma|$
$\alpha = \alpha - \eta \nabla_\alpha \widehat{\mathcal{L}}_{\text{g}}$
$\beta = \beta - \eta \nabla_\beta \widehat{\mathcal{L}}_{\text{g}}$
Return: α, β

model f_θ using only the original task loss $\mathcal{L}_{obj}$, i.e., minimizing $\sum_i \mathcal{L}(f_\theta(\phi(g)(x_i)), \rho(g)(y_i))$ over the augmented dataset. This implicitly encourages the network to learn symmetries by penalizing predictions on transformed data using the standard task objective. REMUL differs by introducing a separate, explicit equivariance loss term $\mathcal{L}_{\text{equi}}$ alongside the standard objective loss $\mathcal{L}_{\text{obj}}$ on the original data, as indicated in Eq. 5. The multitask framework with weights α, β allows *explicit control* over the balance between fitting the original data and enforcing the equivariance constraint.

4 Quantifying Learned Equivariance

Using group transformations to measure and assess the symmetries of ML models has been studied in several domains [51–55]. Inspired by the idea of frame-averaging [24–26], we introduce a metric to quantify the degree of equivariance exhibited by a function f, defined as:

$$E(f, G) = \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} \left\| \frac{1}{M} \sum_{i=1}^{M} \rho(g_i)(f(x)) - \frac{1}{M} \sum_{i=1}^{M} f(\phi(g_i)(x)) \right\|_2 \tag{8}$$

where $\| \cdot \|_2$ denotes an L_2 norm (for non-scalar function), and M is a large number of samples from G. (Proof in Appendix B). This error indicates the average deviation of a function f from perfect equivariance across the data distribution $\mathcal{D}$ (lower value means more equivariant function). We also compare to the standard measure that takes the average over the group of differences between $f(\phi(g)(x))$ and $\rho(g)(f(x))$,

$$E'(f, G) = \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} \frac{1}{M} \sum_{i=1}^{M} \| f(\phi(g_i)(x)) - \rho(g_i)(f(x)) \|_2 \tag{9}$$

We observed that both measures have very similar behavior in our experiments, where E and E' are near zero for equivariant models. Furthermore, as we discussed in Section 3, we demonstrate empirically that increasing the REMUL penalty weight β (Eq. 5) results in a lower equivariant error for E and E'.

5 Related Work

Equivariant ML Models. In the vision domain, group convolutions have proven to be a powerful tool for handling rotation equivariance for images and enhanced model generalization [56–59]. Similarly, the development of equivariant architectures with respect to roto-translations for geometric data has been an active area of research [3–5, 60]. Techniques that use spherical harmonics and irreducible representations have shown a large success in modeling graph-structured data, such as SE(3)-Transformers [15], Tensor Field Networks [14], and DimeNet [61]. More recently, [45] introduced an E(3) equivariant Transformer that employs geometric algebra for processing 3D point clouds.

Data Augmentation and Unconstrained Models. Alternatively, integrating transformations through data augmentation is a widely used strategy across multiple vision tasks, enhancing performance in

image classification [62–64], object detection [65–67], and segmentation [68–70]. For geometric data, [71] has adapted a Graph Neural Network architecture with data augmentation to process 3D molecular structures. In parallel, [72] introduced that Vision Transformers (ViTs) with a large amount of training data can achieve comparable performance with Convolutional Neural Networks (CNNs), obviating the need for explicit translation equivariance within the architecture. Recently, this has shown to be effective for handling geometric data [37, 38].

Learning Symmetries and Approximate Equivariance. Several studies have shown that the layers of CNN architectures can be approximated for a soft constraint [46, 73–77]. Conversely, [78] extends the Bayesian model selection approach to learning symmetries in image datasets. [79] introduced a parameter-sharing scheme to achieve permutations and shifts equivariances in Gaussian distributions. Recent works have relaxed the hard constrained models to a soft constraint by adding unconstrained layers in the architecture design [80, 81], canonicalization network [82], or explicit relaxation [83]. Additionally, [84] modified the loss of CNN for segmentation task. [85] introduced a method to learn equivariant representation using the group invariants, while [86] defined a regularizer that injects the equivariance in the latent space of the network by explicitly modeling transformations with additional learnable maps. In contrast, several works have started from pre-trained models [87, 88]. Furthermore, the EGNN framework [5] has been modified using an invariant function [89] or adversarial training procedure [90]. However, in our work, we start from completely *unconstrained* models, without imposing any equivariance constraints on the space of functions within the architecture. Moreover, we did not assume a specific class of models or introduce additional parameters, which increases the applicability of our method to various domains and makes it computationally efficient.

6 Experiments and Discussion

In this section, we aim to compare constrained equivariant models with unconstrained models trained with REMUL, our multitask approach. We are targeting three main questions: Can unconstrained models learn the approximate equivariance, how does that affect the performance & generalization, and what are their computational costs.

We evaluate our method on different tasks for geometric data: N-body dynamical system (Section 6.1), motion capture (Section 6.2), and molecular dynamics (Section 6.3). For unconstrained models, we apply REMUL to Transformers and Graph Neural Networks. We then compare against their equivariant counterparts: SE(3)-Transformer [15], Geometric Algebra Transformer [45], and Equivariant Graph Neural Networks [5] as well as unconstrained models with data augmentation. We consider learning the rotation group $SO(3)$ for REMUL and data augmentation and we subtract the center of mass for translation. We use the equivariance metric defined in Eq. 8 to analyze our results, and include the second metric in Appendix D. We also conduct a comparative analysis for the computational requirements of unconstrained models and equivariant models in Section 6.4. Lastly, we discuss the loss surfaces in Appendix A. Implementation details and additional experiments can be found in Appendix C & Appendix D.

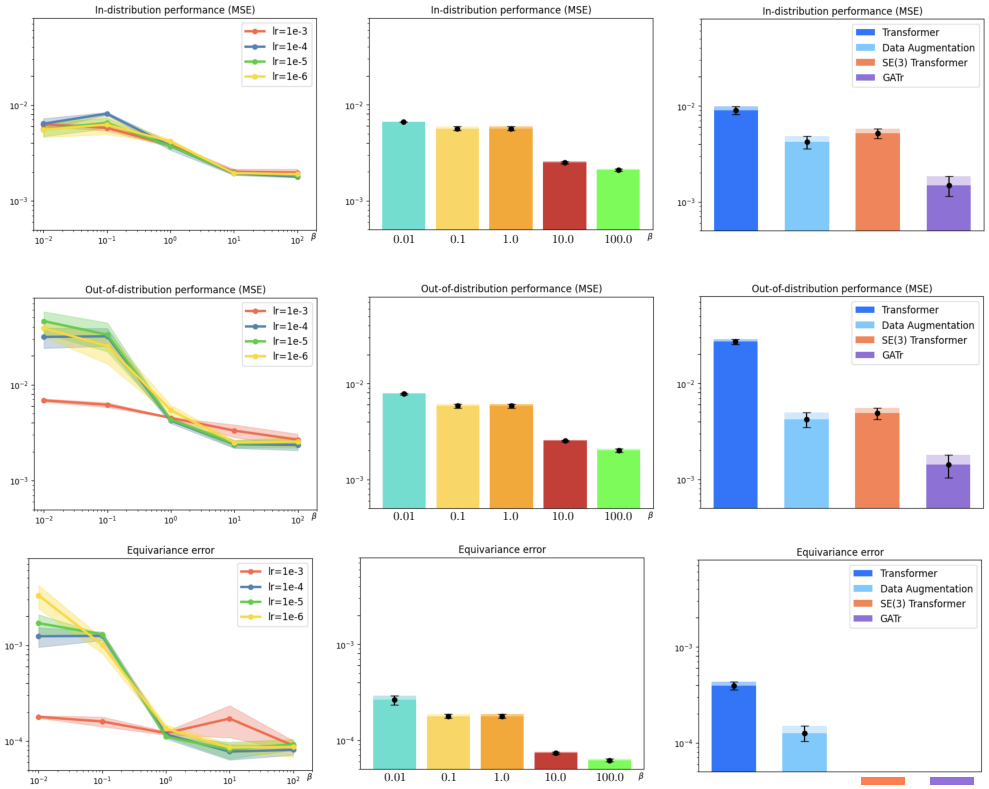
6.1 N-Body Dynamical System

To conduct ablation studies of our method, we utilized the dynamical system problem described by [45]. The task involves predicting the positions of particles after 100 Euler time steps of Newton's motion equation, given initial positions, masses, and velocities. This problem is equivariant under rotation and translation groups, implying that any rotation/translation of the initial states should rotate/translate the final states of the particles by the same amount. We conduct comparisons between Transformer trained with REMUL against two equivariant architectures: SE(3)-Transformer and Geometric Algebra Transformer (GATr). We use the same Transformer version and hyperparameters specified by [45]. Implementation details, including in-distribution and out-of-distribution settings, in Appendix C.2. Our results are presented in Figure 1 and Table 1.

Table 1: N-body dynamical system: MSE ($\times 10^{-3}$). First, Second.

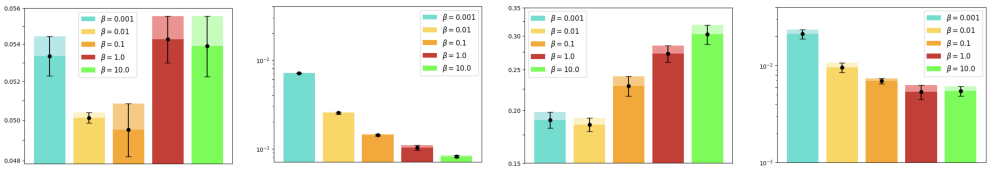
	In-dist.	OOD
SE(3)-Tr	$5.16_{\pm 0.70}$	$4.85_{\pm 0.78}$
GATr	$1.49_{\pm 0.43}$	$1.41_{\pm 0.46}$
Transformer	$8.99_{\pm 1.04}$	$27.06_{\pm 2.01}$
DA-Tr	$4.20_{\pm 0.79}$	$4.21_{\pm 0.91}$
REMUL-Tr	$1.94_{\pm 0.01}$	$1.83_{\pm 0.04}$

From Figure 1, we noticed that increasing the penalty parameter β of the equivariance loss significantly reduces the equivariance error in both constant and gradual settings (which results in a more



(a) REMUL: Gradual penalty (b) REMUL: Constant penalty (c) Baselines

Figure 1: N-body dynamical system. Each row represents a different evaluation scenario. Top: in-distribution performance, Middle: out-of-distribution performance, Bottom: equivariance error. The columns correspond to different architectures/ model conditions. (a) Transformer trained with REMUL (gradual penalty), (b) Transformer trained with a constant penalty, (c) Baselines (equivariant models, standard Transformer, and data augmentation). We conclude that Transformer architecture with high β reduces the equivariance error and improves the performance.



(a) MSE: Walking (b) Equiv. error: Walking (c) MSE: Running (d) Equiv. error: Running

Figure 2: Motion Capture dataset: Transformer trained with REMUL. We show a trade-off between model performance and equiv. error, where high penalty β gives less equiv. error (more equivariant model) but the best performance comes at an intermediate level of equivariance for both tasks.

equivariant model). Equivariant architectures demonstrate an equivariance error near zero, which is expected by their design. The performance behaves similarly; a higher penalty enhances model generalization for both in-distribution and out-of-distribution. Transformer with high β outperforms both data augmentation and SE(3)-Transformer across in-distribution and out-of-distribution and competes with GATr. We also observe that despite SE(3)-Transformer having a substantially lower equivariance error, its performance is slightly worse than Transformer trained with data augmentation. This highlights that equivariance, although improving generalization in this task, is only one aspect of understanding model performance. Lastly, the standard Transformer (without REMUL and data augmentation) exhibits the highest equivariance error and the lowest overall performance.

6.2 Motion Capture

We further illustrate a comparison on a real-world task, the Motion Capture dataset from [91]. This dataset features 3D trajectory data that records a range of human motions, and the task involves predicting the final trajectory based on initial positions and velocities. We have reported results for two types of motion: Walking (Subject #35) and Running (Subject #9). We adhered to the standard experimental setup found in the literature [3, 4, 92], employing a train/validation/test split of 200/600/600 for Walking and 200/240/240 for Running (additional details in Appendix C.3).

We apply our training procedure REMUL to the Transformer architecture and compare it with SE(3)-Transformer, Equivariant Graph Neural Operator (EGNO) [4], Geometric Algebra Transformer (GATr), standard Transformer, and Transformer trained with data augmentation. We also compare with Equivariant MLP [93], as well as two approximate equivariance architectures: Residual Pathway Priors (RPP) [80], and Projection-Based Equivariance Regularizer (PER) [94]. As these architectures are designed specifically on MLP and linear layers, we apply our method to a standard MLP with a similar number of parameters. Our results are presented in Table 2. For REMUL, we provide plots on how the performance and equivariance error change *w.r.t.* the penalty parameter β in Figure 2.

Table 2 indicates that when processing 3D positions related to human motions, both SE(3)-Transformer and GATr perform worse than the standard Transformer. This outcome is noteworthy because human motion often lacks full rotational symmetry, particularly along the vertical or gravity axis. In fact, as detailed in the Appendix D.5 (Table 9), our analysis of axis-specific equivariance errors for REMUL-Transformer confirms that the error is highest for rotations around the Z-axis. Consequently, imposing strict $SO(3)$ equivariance across all axes may not be beneficial and can be detrimental to performance. In contrast, a standard Transformer trained with REMUL has the best performance in both tasks. Following Figure 2, there is a noticeable trade-off: while higher β values reduce overall equivariance error, optimal task performance is often observed at an intermediate level of learned equivariance, where the model balances between being too rigid (fully equivariant) and too flexible (non-equivariant). This underscores that the optimal degree of symmetry is task-dependent and that REMUL's flexibility in learning approximate equivariance is advantageous for such real-world scenarios.

Table 2: Motion Capture dataset: MSE ($\times 10^{-2}$). REMUL procedure and data augmentation (DA) were applied to standard Transformer and MLP. First, Second. REMUL comes best in both tasks.

	Walking	Running
SE(3)-Tr	$10.85_{\pm 1.3}$	$42.13_{\pm 3.4}$
GATr	$10.06_{\pm 1.3}$	$32.38_{\pm 3.9}$
EGNO	$8.1_{\pm 1.6}$	$33.9_{\pm 1.7}$
Transformer	$5.21_{\pm 0.08}$	$20.78_{\pm 1.5}$
DA-Tr	$5.3_{\pm 0.18}$	$29.83_{\pm 1.4}$
REMUL-Tr	$4.95_{\pm 0.1}$	$18.5_{\pm 0.7}$
EMLP	$7.01_{\pm 0.46}$	$57.38_{\pm 8.39}$
RPP	$6.99_{\pm 0.21}$	$34.18_{\pm 2.00}$
PER	$7.48_{\pm 0.39}$	$33.03_{\pm 0.37}$
MLP	$6.80_{\pm 0.18}$	$39.56_{\pm 2.25}$
DA-MLP	$6.37_{\pm 0.04}$	$40.23_{\pm 0.94}$
REMUL-MLP	$6.04_{\pm 0.09}$	$32.57_{\pm 1.47}$

6.3 Molecular Dynamics

We also present a comparative analysis between constrained equivariant models and unconstrained models focusing on molecular dynamics, specifically predicting 3D molecule structures. We utilize the MD17 dataset [95], which comprises trajectories of eight small molecules. We use the same dataset split in [4, 92], allocating 500 samples for train, 2000 for validation, and 2000 for test. For this task, we selected the Equivariant Graph Neural Network (EGNN) architecture and its non-equivariant GNN counterpart, as presented in [5]. We then apply REMUL procedure as well as data augmentation to the GNN architecture. Both architectures have the same hyperparameters (more information is indicated in Appendix C.4). We also compare with GMN [92], EGNO [4], and HEGNN [96]. Our results are provided in Table 3. We illustrate how the performance and equivariance error of a GNN trained with REMUL vary across different molecules as a function of β in Figure 9 and Figure 10.

From the results presented in Table 3, GNN trained with REMUL outperforms EGNN in six out of eight molecules. Interestingly, a standard GNN, without data augmentation or REMUL, surpasses the performance of EGNN on multiple molecules, such as Aspirin and Toluene. In Figure 9 & Figure 10, we observe that the optimal performance of each molecule is attained at different values of the penalty parameter β. For instance, Malonaldehyde exhibits a direct correlation between model performance and equivariance, where a higher β yields better performance. Conversely, for most other molecules, there appears to be a pronounced trade-off where the best performance is achieved at a lower value of β. This is particularly evident with molecules like Aspirin, where a standard GNN architecture

Table 3: MD17 dataset: MSE ($\times 10^{-2}$). REMUL procedure and data augmentation (DA) were applied to GNN. First, Second.

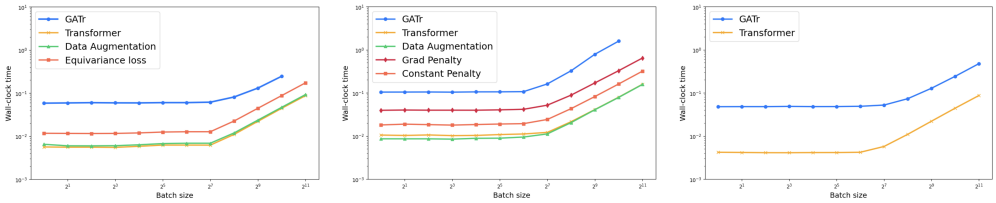
	Aspirin	Benzene	Ethanol	Malonaldehyde	Naphthalene	Salicylic	Toluene	Uracil
EGNN	$14.41_{\pm 0.15}$	$62.40_{\pm 0.53}$	$4.64_{\pm 0.01}$	$13.64_{\pm 0.01}$	$0.47_{\pm 0.02}$	$1.02_{\pm 0.02}$	$11.78_{\pm 0.07}$	$0.64_{\pm 0.01}$
GMN	$10.14_{\pm 0.03}$	$48.12_{\pm 0.4}$	$4.83_{\pm 0.01}$	$13.11_{\pm 0.03}$	$0.40_{\pm 0.01}$	$0.91_{\pm 0.01}$	$10.22_{\pm 0.08}$	$0.59_{\pm 0.01}$
EGNO	$9.18_{\pm 0.06}$	$48.85_{\pm 0.55}$	$4.62_{\pm 0.01}$	$12.80_{\pm 0.02}$	$0.37_{\pm 0.01}$	$0.86_{\pm 0.02}$	$10.21_{\pm 0.05}$	$0.52_{\pm 0.02}$
HEGNN	$9.94_{\pm 0.07}$	$59.93_{\pm 5.21}$	$4.62_{\pm 0.01}$	$12.85_{\pm 0.01}$	$0.37_{\pm 0.02}$	$0.88_{\pm 0.02}$	$10.56_{\pm 0.33}$	$0.54_{\pm 0.01}$
GNN	$9.26_{\pm 0.40}$	$26.13_{\pm 0.11}$	$4.26_{\pm 0.03}$	$18.45_{\pm 0.54}$	$0.54_{\pm 0.001}$	$1.02_{\pm 0.02}$	$9.93_{\pm 0.82}$	$0.70_{\pm 0.001}$
DA-GNN	$13.7_{\pm 0.04}$	$110.93_{\pm 5.3}$	$5.74_{\pm 0.02}$	$13.65_{\pm 0.02}$	$0.69_{\pm 0.001}$	$1.33_{\pm 0.04}$	$19.14_{\pm 0.001}$	$0.73_{\pm 0.002}$
REMUL-GNN	$9.28_{\pm 0.40}$	$25.95_{\pm 0.18}$	$4.02_{\pm 0.16}$	$13.59_{\pm 0.03}$	$0.54_{\pm 0.001}$	$0.99_{\pm 0.001}$	$9.38_{\pm 0.20}$	$0.67_{\pm 0.001}$

outperforms EGNN. We also plot the 3D structures of the eight molecules in Figure 12. Molecules such as Malonaldehyde, characterized by their symmetric components, might be ideally suited for equivariant design. However, this advantage does not apply to all molecules. Aspirin on the other side, might have an asymmetric structure and exhibit a range of interactions and dynamic states that equivariant models might simplify. Consequently, for such molecules, less equivariant models could potentially offer more accurate predictions.

Finally, while REMUL also achieves competitive performance relative to other equivariant models such as EGNO and HEGNN, it is important to note that these models incorporate architectural elements that enhance geometric properties in distinct ways and might not be directly comparable to a simple GNN. For example, EGNO employs additional Fourier features, while HEGNN leverages high-degree steerable features.

6.4 Computational Complexity

In this section, we report the computational time for the Geometric Algebra Transformer (GATr) and Transformer architectures. We selected models with an equivalent number of blocks and parameters for a fair comparison. Detailed configurations are provided in Appendix C.5. We measured the computational efficiency of each model by recording the time taken for both forward and backward passes during training, as well as inference time. For the Transformer's computations, we also considered all the cases of data augmentation and our training procedure with the equivariance loss. Figure 3 includes the wall-clock time as a function of batch size with a fixed number of nodes.



(a) Combined forward pass (b) Backward pass (c) Inference time

Figure 3: Computational time for GATr and Transformer architectures. GATr has the highest time in all scenarios. Inference times for all versions of the Transformer (standard and trained with equivariance loss and data augmentation) are the same.

In all comparisons, GATr architecture consistently required the highest time, being approximately ten times slower than Transformer architecture. This significant difference can be attributed to the calculations of multivectors in GATr's design. In the combined forward and backward passes, the addition of the equivariance loss increases the computation time of the standard Transformer as we calculate two model outputs at each step. However, it's still around $2.5\times$ faster than GATr, in the worst case of a gradual penalty. Furthermore, GATr reached its memory capacity earlier, hitting an out-of-memory issue at a batch size of 2^{11}. During inference, the computational speed for the Transformer trained with equivariance loss or data augmentation matches the standard Transformer, which results in an inference speed that is $10\times$ faster than GATr. Notably, while we include GATr as our equivariant baseline, GATr itself is computationally more efficient than many equivariant architectures such as SE(3)-Transformer and SEGNN, as indicated in [45].

7 Conclusion

We introduced REMUL, a simple and effective method for learning *approximately* equivariant functions using unconstrained architectures. By formulating equivariance as an explicit, tunable objective within a multitask learning framework, REMUL relaxes the often costly and rigid constraints of traditional equivariant models. We demonstrated empirically that unconstrained networks trained with REMUL can learn appropriate levels of symmetry, controlled by a hyperparameter β. This allows us to balance the benefits of the equivariance inductive bias against task-specific requirements and computational costs. Our method achieves competitive performance compared to constrained baselines on various geometric tasks, while offering significant speed advantages (up to $10\times$ faster inference, $2.5\times$ faster training).

Limitations and Future Directions. This work introduces a simple approach for understanding and analyzing unconstrained versus equivariant models, which significantly impact the field by enabling broader applicability and easier integration into existing frameworks. Building on these foundations, numerous additional ideas for extending our study present exciting opportunities for future research. For instance, as we noted earlier, α and β serve as additional hyperparameters that could be constant or automatically updated with GradNorm algorithm, we could explore more efficient learnable weights, such as [97, 98]. Another promising avenue is applying our method during the fine-tuning phase when leveraging pre-trained models for tasks that require equivariance [99, 100]. On the other side, further analysis is required to understand the theoretical guarantees of approximate equivariance offered by REMUL, such as how relaxing equivariance constraints affects the model's generalization bounds [47].

Acknowledgments

The authors would like to thank Joey Bose, Jacob Bamberger, Xingyue Huang, Emily Jin, Katarina Petrovic, and Scott Le Roux for their valuable comments on the early versions of the manuscript. This research is partially supported by EPSRC Turing AI World-Leading Research Fellowship No. EP/X040062/1, EPSRC AI Hub on Mathematical Foundations of Intelligence: An "Erlangen Programme" for AI No. EP/Y028872/1, and the Postdoc.Mobility grant P500PT-217915 from the Swiss National Science Foundation.

References

[1] Maurice Weiler, Fred A. Hamprecht, and Martin Storath. Learning steerable filters for rotation equivariant cnns. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 849–858, 2018. doi: 10.1109/CVPR.2018.00095. 1

[2] Hong-Xing Yu, Jiajun Wu, and Li Yi. Rotationally equivariant 3d object detection. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1446–1454, 2022. doi: 10.1109/CVPR52688.2022.00151. 1

[3] Jiaqi Han, Wenbing Huang, Tingyang Xu, and Yu Rong. Equivariant graph hierarchy-based neural networks. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=ywxtmG1nU_6. 1, 5, 8, 20

[4] Minkai Xu, Jiaqi Han, Aaron Lou, Jean Kossaifi, Arvind Ramanathan, Kamyar Azizzadenesheli, Jure Leskovec, Stefano Ermon, and Anima Anandkumar. Equivariant graph neural operator for modeling 3d dynamics. In *Proceedings of the 41st International Conference on Machine Learning*, 2024. 1, 8, 20, 21

[5] Victor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E(n) equivariant graph neural networks. In *Proceedings of the 38rd International Conference on Machine Learning*, 2021. 1, 5, 6, 8, 21, 23

[6] Johannes Brandstetter, Rob Hesselink, Elise van der Pol, Erik J Bekkers, and Max Welling. Geometric and physical quantities improve e(3) equivariant message passing. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=_xwr8gOBeV1. 1, 23

[7] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex

Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021. doi: 10.1038/s41586-021-03819-2. URL https://doi.org/10.1038/s41586-021-03819-2. 1

[8] Kristof Schütt, Oliver Unke, and Michael Gastegger. Equivariant message passing for the prediction of tensorial properties and molecular spectra. In *Proceedings of the 38th International Conference on Machine Learning*, 2021. URL https://proceedings.mlr.press/v139/schutt21a.html. 1

[9] Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges, 2021. URL https://arxiv.org/abs/2104.13478.

[10] Philipp Thölke and Gianni De Fabritiis. Equivariant transformers for neural network based molecular potentials. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=zNHzqZ9wrRB.

[11] Yi-Lun Liao, Brandon Wood, Abhishek Das*, and Tess Smidt*. EquiformerV2: Improved Equivariant Transformer for Scaling to Higher-Degree Representations. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=mCOBKZmrzD. 1

[12] Simon Batzner, Albert Musaelian, Lixin Sun, et al. E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature Communications*, 13:2453, 2022. doi: 10.1038/s41467-022-29939-5. URL https://doi.org/10.1038/s41467-022-29939-5. 1

[13] Ilyes Batatia, David Peter Kovacs, Gregor N. C. Simm, Christoph Ortner, and Gabor Csanyi. MACE: Higher order equivariant message passing neural networks for fast and accurate force fields. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=YPpSngE-ZU. 1

[14] 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, 2018. URL https://arxiv.org/abs/1802.08219. 1, 5, 23

[15] Fabian B. Fuchs, Daniel E. Worrall, Volker Fischer, and Max Welling. Se(3)-transformers: 3d roto-translation equivariant attention networks. In *Advances in Neural Information Processing Systems 34 (NeurIPS)*, 2020. 5, 6

[16] Yi-Lun Liao and Tess Smidt. Equiformer: Equivariant graph attention transformer for 3d atomistic graphs. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=KwmPfARgOTD.

[17] Shengjie Luo, Tianlang Chen, and Aditi S. Krishnapriyan. Enabling efficient equivariant operations in the fourier basis via gaunt tensor products, 2024. URL https://arxiv.org/abs/2401.10216. 1

[18] Chaitanya K. Joshi, Cristian Bodnar, Simon V Mathis, Taco Cohen, and Pietro Lio. On the expressive power of geometric graph neural networks. In *Proceedings of the 40th International Conference on Machine Learning*, 2023. URL https://proceedings.mlr.press/v202/joshi23a.html. 1

[19] Sékou-Oumar Kaba, Arnab Kumar Mondal, Yan Zhang, Yoshua Bengio, and Siamak Ravanbakhsh. Equivariance with learned canonicalization functions. In *NeurIPS 2022 Workshop on Symmetry and Geometry in Neural Representations*, 2022. URL https://openreview.net/forum?id=pVD1k8ge25a. 1

[20] Arnab Kumar Mondal, Siba Smarak Panigrahi, Sékou-Oumar Kaba, Sai Rajeswar, and Siamak Ravanbakhsh. Equivariant adaptation of large pretrained models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=m6dRQJw280.

[21] Justin Baker, Shih-Hsin Wang, Tommaso de Fernex, and Bao Wang. An explicit frame construction for normalizing 3d point clouds. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=SZ0JnRxi0x.

[22] George Ma, Yifei Wang, Derek Lim, Stefanie Jegelka, and Yisen Wang. A canonicalization perspective on invariant and equivariant learning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=jjcY92FX4R.

[23] Siba Smarak Panigrahi and Arnab Kumar Mondal. Improved canonicalization for model agnostic equivariance. In *CVPR 2024 Workshop on Equivariant Vision: From Theory to Practice*, 2024. URL https://arxiv.org/abs/2405.14089. 1

[24] Omri Puny, Matan Atzmon, Edward J. Smith, Ishan Misra, Aditya Grover, Heli Ben-Hamu, and Yaron Lipman. Frame averaging for invariant and equivariant network design. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=zIUyj55nXR. 1, 5, 23

[25] Alexandre Agm Duval, Victor Schmidt, Alex Hernández-García, Santiago Miret, Fragkiskos D. Malliaros, Yoshua Bengio, and David Rolnick. FAENet: Frame averaging equivariant GNN for materials modeling. In *Proceedings of the 40th International Conference on Machine Learning*, 2023. URL https://proceedings.mlr.press/v202/duval23a.html.

[26] Yuchao Lin, Jacob Helwig, Shurui Gui, and Shuiwang Ji. Equivariance via minimal frame averaging for more symmetries and efficiency. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=guFsTBXsov. 5

[27] Tinglin Huang, Zhenqiao Song, Rex Ying, and Wengong Jin. Protein-nucleic acid complex modeling with frame averaging transformer. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=Xngi3Z3wkN. 1

[28] Facundo Quiroga, Franco Ronchetti, Laura Lanzarini, and Aurelio F. Bariviera. *Revisiting Data Augmentation for Rotational Invariance in Convolutional Neural Networks*, page 127–141. Springer International Publishing, March 2019. ISBN 9783030154134. doi: 10.1007/978-3-030-15413-4_10. URL http://dx.doi.org/10.1007/978-3-030-15413-4_10. 2

[29] Gregory Benton, Marc Finzi, Pavel Izmailov, and Andrew Gordon Wilson. Learning invariances in neural networks. *arXiv preprint arXiv:2010.11882*, 2020.

[30] Aoming Liu, Zehao Huang, Zhiwu Huang, and Naiyan Wang. Direct differentiable augmentation search. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12199–12208, 2021. doi: 10.1109/ICCV48922.2021.01200.

[31] Yutong Bai, Jieru Mei, Alan Yuille, and Cihang Xie. Are transformers more robust than cnns? In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

[32] Jan E. Gerken, Oscar Carlsson, Hampus Linander, Fredrik Ohlsson, Christoffer Petersson, and Daniel Persson. Equivariance versus Augmentation for Spherical Images. In *Proceedings of the 39th International Conference on Machine Learning*, pages 7404–7421. PMLR, 2022. doi: 10.48550/arXiv.2202.03990.

[33] Guillermo Iglesias, Edgar Talavera, Ángel González-Prieto, Alberto Mozo, and Sandra Gómez-Canaval. Data augmentation techniques in time series domain: a survey and taxonomy. *Neural Computing and Applications*, 35(14):10123–10145, March 2023. ISSN 1433-3058. doi: 10.1007/s00521-023-08459-3. URL http://dx.doi.org/10.1007/s00521-023-08459-3.

[34] Evangelos Chatzipantazis, Stefanos Pertigkiozoglou, Kostas Daniilidis, and Edgar Dobriban. Learning augmentation distributions using transformed risk minimization. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=LRYtNj8Xw0.

[35] Mingle Xu, Sook Yoon, Alvaro Fuentes, and Dong Sun Park. A comprehensive survey of image augmentation techniques for deep learning. *Pattern Recognition*, 137:109347, 2023. ISSN 0031-3203. doi: 10.1016/j.patcog.2023.109347. URL https://www.sciencedirect.com/science/article/pii/S0031320323000481.

[36] Suorong Yang, Suhan Guo, Jian Zhao, and Furao Shen. Investigating the effectiveness of data augmentation from similarity and diversity: An empirical study. *Pattern Recognition*, 148:110204, 2024. ISSN 0031-3203. doi: 10.1016/j.patcog.2023.110204. URL https://www.sciencedirect.com/science/article/pii/S0031320323009019. 2

[37] Yuyang Wang, Ahmed A. Elhag, Navdeep Jaitly, Joshua M. Susskind, and Miguel Ángel Bautista. Swallowing the bitter pill: Simplified scalable conformer generation. In *Forty-first International Conference on Machine Learning*, 2024. 2, 6

[38] Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J. Ballard, Joshua Bambrick, Sebastian W. Bodenstein, David A. Evans, Chia-Chun Hung, Michael O'Neill, David Reiman, Kathryn Tunyasuvunakool, Zachary Wu, Akvilė Žemgulytė, Eirini Arvaniti, Charles Beattie, Ottavia Bertolli, Alex Bridgland, Alexey Cherepanov, Miles Congreve, Alexander I. Cowen-Rivers, Andrew Cowie, Michael Figurnov, Fabian B. Fuchs, Hannah Gladman, Rishub Jain, Yousuf A. Khan, Caroline M. R. Low, Kuba Perlin, Anna Potapenko, Pascal Savy, Sukhdeep Singh, Adrian Stecula, Ashok Thillaisundaram, Catherine Tong, Sergei Yakneen, Ellen D. Zhong, Michal Zielinski, Augustin Žídek, Victor Bapst, Pushmeet Kohli, Max Jaderberg, Demis Hassabis, and John M. Jumper. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, 630(8016):493–500, 2024. doi: 10.1038/s41586-024-07487-w. 2, 6

[39] Yongjoo Baek, Yariv Kafri, and Vivien Lecomte. Dynamical symmetry breaking and phase transitions in driven diffusive systems. *Physical Review Letters*, 118(3), January 2017. ISSN 1079-7114. doi: 10.1103/physrevlett.118.030604. URL http://dx.doi.org/10.1103/PhysRevLett.118.030604. 2

[40] Simon A. Weidinger, Markus Heyl, Alessandro Silva, and Michael Knap. Dynamical quantum phase transitions in systems with continuous symmetry breaking. *Physical Review B*, 96 (13), October 2017. ISSN 2469-9969. doi: 10.1103/physrevb.96.134313. URL http://dx.doi.org/10.1103/PhysRevB.96.134313. 2

[41] Calum J. Gibb, Jordan Hobbs, Diana I. Nikolova, Thomas Raistrick, Stuart R. Berrow, Alenka Mertelj, Natan Osterman, Nerea Sebastián, Helen F. Gleeson, and Richard. J. Mandle. Spontaneous symmetry breaking in polar fluids. *Nature Communications*, 15(1), July 2024. ISSN 2041-1723. doi: 10.1038/s41467-024-50230-2. URL http://dx.doi.org/10.1038/s41467-024-50230-2. 2

[42] Constantine Yannouleas and Uzi Landman. Erratum: Spontaneous symmetry breaking in single and molecular quantum dots [phys. rev. lett. 82, 5325 (1999)]. *Physical Review Letters*, 85(10):2220–2220, September 2000. ISSN 1079-7114. doi: 10.1103/physrevlett.85.2220. URL http://dx.doi.org/10.1103/PhysRevLett.85.2220. 2

[43] Nathan W. Goehring, Philipp Khuc Trong, Justin S. Bois, Debanjan Chowdhury, Ernesto M. Nicola, Anthony A. Hyman, and Stephan W. Grill. Polarization of PAR proteins by advective triggering of a pattern-forming system. *Science*, 334:1137–1141, 2011. doi: 10.1126/science.1208619. Epub 2011 Oct 20. 2

[44] Alexander Mietke, V. Jemseena, K. Vijay Kumar, Ivo F. Sbalzarini, and Frank Jülicher. Minimal model of cellular symmetry breaking. *Phys. Rev. Lett.*, 123:188101, Oct 2019. doi: 10.1103/PhysRevLett.123.188101. URL https://link.aps.org/doi/10.1103/PhysRevLett.123.188101. 2

[45] Johann Brehmer, Pim De Haan, Sönke Behrends, and Taco Cohen. Geometric algebra transformer. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=M7r2CO4tJC. 2, 5, 6, 9, 20

[46] Rui Wang, Robin Walters, and Rose Yu. Approximately equivariant networks for imperfectly symmetric dynamics. In *Proceedings of the 39th International Conference on Machine Learning*, 2022. 3, 6

[47] Mircea Petrache and Shubhendu Trivedi. Approximation-generalization trade-offs under (approximate) group equivariance. In *Advances in Neural Information Processing Systems*, 2023. 10

[48] Dominik S. Kufel, Jack Kemp, Simon M. Linsel, Chris R. Laumann, and Norman Y. Yao. Approximately-symmetric neural networks for quantum spin liquids, 2024. URL https://arxiv.org/abs/2405.17541.

[49] Matthew Ashman, Cristiana Diaconu, Adrian Weller, Wessel Bruinsma, and Richard E. Turner. Approximately equivariant neural processes, 2024. URL https://arxiv.org/abs/2406.13488. 3

[50] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. GradNorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *Proceedings of the 35th International Conference on Machine Learning*, 2018. URL https://proceedings.mlr.press/v80/chen18a.html. 4

[51] Clare Lyle, Mark van der Wilk, Marta Kwiatkowska, Yarin Gal, and Benjamin Bloem-Reddy. On the benefits of invariance in neural networks, 2020. URL https://arxiv.org/abs/2005.00178. 5

[52] Henry Kvinge, Tegan Emerson, Grayson Jorgenson, Scott Vasquez, Timothy Doster, and Jesse Lew. In what ways are deep neural networks invariant and how should we measure this? In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=SCD0hn3kMHw.

[53] Artem Moskalev, Anna Sepliarskaia, Erik J. Bekkers, and Arnold Smeulders. On genuine invariance learning without weight-tying. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

[54] Nate Gruver, Marc Anton Finzi, Micah Goldblum, and Andrew Gordon Wilson. The lie derivative for measuring learned equivariance. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=JL7Va5Vy15J.

[55] Till Speicher, Vedant Nanda, and Krishna P. Gummadi. Understanding the role of invariance in transfer learning. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=spJI4LSPIU. 5

[56] Taco Cohen and Max Welling. Group equivariant convolutional networks. In *Proceedings of the 33rd International Conference on Machine Learning*, 2016. URL https://proceedings.mlr.press/v48/cohenc16.html. 5

[57] Taco S. Cohen, Maurice Weiler, Berkay Kicanaoglu, and Max Welling. Gauge equivariant convolutional networks and the icosahedral cnn. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

[58] Maurice Weiler and Gabriele Cesa. General E(2)-Equivariant Steerable CNNs. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

[59] Wei-Dong Qiao, Yang Xu, and Hui Li. Scale-rotation-equivariant lie group convolution neural networks (lie group-cnns), 2023. URL https://arxiv.org/abs/2306.06934. 5

[60] Haiwei Chen, Shichen Liu, Weikai Chen, Hao Li, and Randall Hill. Equivariant point network for 3d point cloud analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 5

[61] Johannes Gasteiger, Janek Groß, and Stephan Günnemann. Directional message passing for molecular graphs. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=B1eWbxStPH. 5

[62] Luis Perez and Jason Wang. The effectiveness of data augmentation in image classification using deep learning, 2017. URL https://arxiv.org/abs/1712.04621. 6

[63] Hiroshi Inoue. Data augmentation by pairing samples for images classification, 2018. URL https://arxiv.org/abs/1801.02929.

[64] Fazle Rahat, M Shifat Hossain, Md Rubel Ahmed, Sumit Kumar Jha, and Rickard Ewetz. Data augmentation for image classification using generative ai, 2024. URL https://arxiv.org/abs/2409.00547. 6

[65] Barret Zoph, Ekin D. Cubuk, Golnaz Ghiasi, Tsung-Yi Lin, Jonathon Shlens, and Quoc V. Le. Learning data augmentation strategies for object detection. In *Computer Vision – ECCV 2020*, 2020. 6

[66] Hao Wang, Qilong Wang, Fan Yang, Weiqi Zhang, and Wangmeng Zuo. Data augmentation for object detection via progressive and selective instance-switching, 2019. URL https://arxiv.org/abs/1906.00358.

[67] Mate Kisantal, Zbigniew Wojna, Jakub Murawski, Jacek Naruniec, and Kyunghyun Cho. Augmentation for small object detection, 2019. URL https://arxiv.org/abs/1902.07296. 6

[68] Misgana Negassi, Diane Wagner, and Alexander Reiterer. Smart(sampling)augment: Optimal and efficient data augmentation for semantic segmentation. *Algorithms*, 15(5), 2022. ISSN 1999-4893. doi: 10.3390/a15050165. URL https://www.mdpi.com/1999-4893/15/5/165. 6

[69] X Chen, C Lian, L Wang, H Deng, T Kuang, SH Fung, J Gateno, D Shen, JJ Xia, and PT Yap. Diverse data augmentation for learning image segmentation with cross-modality annotations. *Medical Image Analysis*, 71:102060, 2021. doi: 10.1016/j.media.2021.102060. Epub 2021 Apr 20. PMID: 33957558; PMCID: PMC8184609.

[70] Xinyi Yu, Guanbin Li, Wei Lou, Siqi Liu, Xiang Wan, Yan Chen, and Haofeng Li. Diffusion-based data augmentation for nuclei image segmentation. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2023*, 2023. 6

[71] Weihua Hu, Muhammed Shuaibi, Abhishek Das, Siddharth Goyal, Anuroop Sriram, Jure Leskovec, Devi Parikh, and C. Lawrence Zitnick. Forcenet: A graph neural network for large-scale quantum calculations, 2021. URL https://arxiv.org/abs/2103.01436. 6

[72] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy. 6

[73] Tycho F.A. van der Ouderaa, David W. Romero, and Mark van der Wilk. Relaxing equivariance constraints with non-stationary continuous filters. In *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=5oEk8fvJxny. 6

[74] David W. Romero and Suhas Lohit. Learning partial equivariances from data. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=pNHT6oBaPr8.

[75] Lars Veefkind and Gabriele Cesa. A probabilistic approach to learning the degree of equivariance in steerable cnns, 2024. URL https://arxiv.org/abs/2406.03946.

[76] Zhiqiang Wu, Yingjie Liu, Hanlin Dong, Xuan Tang, Jian Yang, Bo Jin, Mingsong Chen, and Xian Wei. Sbdet: A symmetry-breaking object detector via relaxed rotation-equivariance, 2024. URL https://arxiv.org/abs/2408.11760.

[77] Daniel McNeela. Almost equivariance via lie algebra convolutions. In *NeurIPS 2023 Workshop on Symmetry and Geometry in Neural Representations*, 2023. URL https://openreview.net/forum?id=2sLBXyVsPE. 6

[78] Tycho F.A. van der Ouderaa, Alexander Immer, and Mark van der Wilk. Learning layer-wise equivariances automatically using gradients. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=bNIHdyunFC. 6

[79] Raymond A. Yeh, Yuan-Ting Hu, Mark Hasegawa-Johnson, and Alexander Schwing. Equivariance discovery by learned parameter-sharing. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, 2022. URL https://proceedings.mlr.press/v151/yeh22b.html. 6

[80] Marc Finzi, Gregory Benton, and Andrew G Wilson. Residual pathway priors for soft equivariance constraints. In *Advances in Neural Information Processing Systems*, 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/fc394e9935fbd62c8aedc372464e1965-Paper.pdf. 6, 8

[81] Stefanos Pertigkiozoglou, Evangelos Chatzipantazis, Shubhendu Trivedi, and Kostas Daniilidis. Improving equivariant model training via constraint relaxation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=tWkL7k1u5v. 6

[82] Hannah Lawrence, Vasco Portilheiro, Yan Zhang, and Sékou-Oumar Kaba. Improving equivariant networks with probabilistic symmetry breaking. In *ICML 2024 Workshop on Geometry-grounded Representation Learning and Generative Modeling*, 2024. URL https://openreview.net/forum?id=1VlRaXNMWO. 6

[83] Sékou-Oumar Kaba and Siamak Ravanbakhsh. Symmetry breaking and equivariant neural networks. In *NeurIPS 2023 Workshop on Symmetry and Geometry in Neural Representations*, 2023. URL https://openreview.net/forum?id=d55JaRL9wh. 6

[84] Kangcheng Lin, Bohao Huang, Leslie M. Collins, Kyle Bradbury, and Jordan M. Malof. A simple rotational equivariance loss for generic convolutional segmentation networks: preliminary results. In *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, pages 3876–3879, 2019. doi: 10.1109/IGARSS.2019.8898722. 6

[85] Mehran Shakerinava, Arnab Kumar Mondal, and Siamak Ravanbakhsh. Structuring representations using group invariants. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=vWUmBjin_-o. 6

[86] Sangnie Bhardwaj, Willie McClinton, Tongzhou Wang, Guillaume Lajoie, Chen Sun, Phillip Isola, and Dilip Krishnan. Steerable equivariant representation learning, 2023. URL https://arxiv.org/abs/2302.11349. 6

[87] Sourya Basu, Prasanna Sattigeri, Karthikeyan Natesan Ramamurthy, Vijil Chenthamarakshan, Kush R. Varshney, Lav R. Varshney, and Payel Das. Equi-tuning: Group equivariant fine-tuning of pretrained models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023. URL https://ojs.aaai.org/index.php/AAAI/article/view/25832. 6

[88] Jinwoo Kim, Dat Tien Nguyen, Ayhan Suleymanzade, Hyeokjun An, and Seunghoon Hong. Learning probabilistic symmetrization for architecture agnostic equivariance. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=phnN1eu5AX. 6

[89] Zinan Zheng, Yang Liu, Jia Li, Jianhua Yao, and Yu Rong. Relaxing continuous constraints of equivariant graph neural networks for physical dynamics learning, 2024. URL https://arxiv.org/abs/2406.16295. 6

[90] Jianke Yang, Robin Walters, Nima Dehmamy, and Rose Yu. Generative adversarial symmetry discovery. In *Proceedings of the 40th International Conference on Machine Learning*, 2023. 6

[91] CMU. Carnegie mellon motion capture database. http://mocap.cs.cmu.edu, 2003. 8, 20

[92] Wenbing Huang, Jiaqi Han, Yu Rong, Tingyang Xu, Fuchun Sun, and Junzhou Huang. Equivariant graph mechanics networks with constraints. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=SHbhHHfePhP. 8, 20, 21

[93] Marc Finzi, Max Welling, and Andrew Gordon Wilson. A practical method for constructing equivariant multilayer perceptrons for arbitrary matrix groups. In *Proceedings of the 38th International Conference on Machine Learning*, 2021. 8

[94] Hyunsu Kim, Hyungi Lee, Hongseok Yang, and Juho Lee. Regularizing towards soft equivariance under mixed symmetries. In *Proceedings of the 40th International Conference on Machine Learning*, 2023. 8

[95] Stefan Chmiela, Alexandre Tkatchenko, Huziel E. Sauceda, Igor Poltavsky, Kristof T. Schütt, and Klaus-Robert Müller. Machine learning of accurate energy-conserving molecular force fields. *Science Advances*, 3(5), May 2017. ISSN 2375-2548. doi: 10.1126/sciadv.1603015. URL http://dx.doi.org/10.1126/sciadv.1603015. 8, 21

[96] Jiacheng Cen, Anyi Li, Ning Lin, Yuxiang Ren, Zihe Wang, and Wenbing Huang. Are high-degree representations really unnecessary in equivariant graph neural networks? In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=MOncNVuGYN. 8

[97] Michael Crawshaw and Jana Košecká. Slaw: Scaled loss approximate weighting for efficient multi-task learning, 2021. URL https://arxiv.org/abs/2109.08218. 10

[98] Christian Bohn, Ido Freeman, Hasan Tercan, and Tobias Meisen. Task weighting through gradient projection for multitask learning, 2024. URL https://arxiv.org/abs/2409.01793. 10

[99] Sheheryar Zaidi, Michael Schaarschmidt, James Martens, Hyunjik Kim, Yee Whye Teh, Alvaro Sanchez-Gonzalez, Peter Battaglia, Razvan Pascanu, and Jonathan Godwin. Pre-training via denoising for molecular property prediction. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=tYIMtogyee. 10

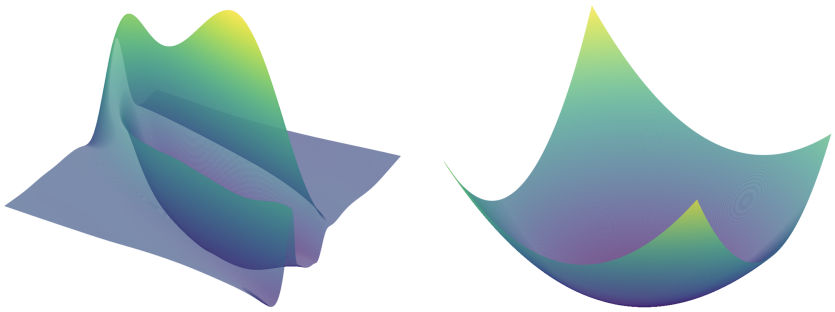
[100] Yuyan Ni, Shikun Feng, Xin Hong, Yuancheng Sun, Wei-Ying Ma, Zhi-Ming Ma, Qiwei Ye, and Yanyan Lan. Pre-training with fractional denoising to enhance molecular property prediction, 2024. URL https://arxiv.org/abs/2407.11086. 10

[101] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. In *Neural Information Processing Systems*, 2018. 18

A Loss Surface

In this section, we analyze the relative ease of training equivariant models compared to non-equivariant models by examining the loss surface around the achieved local minima for each model. We explore how each architecture influences the loss landscape when trained on the same task. However, due to the high dimensionality of parameter spaces in neural networks, visualizing their loss functions in three dimensions might be a significant challenge. We use the filter normalization method introduced by [101], which calculates the loss function along two randomly selected Gaussian directions in the parameters space, starting from the optimal parameters θ^* achieved at the end of training.

We visualize the loss surface of the Geometric Algebra Transformer (GATr) and Transformer in Figure 4, trained on the N-body dynamical system. We observe that the Transformer architecture exhibits a more favorable loss shape around its local minima, characterized by a convex structure. This might suggest that the optimization path for the Transformer is smoother and potentially easier to navigate during training, leading to more stable convergence. In contrast, the loss surface of GATr appears more erratic and rugged. This complexity in the loss landscape can indicate multiple local minima and a higher sensitivity to initial conditions or parameter settings. Such characteristics might complicate the training process, requiring more careful tuning of hyperparameters. We leave this for future work to analyze how the optimization path for each model behaves during training.



(a) Geometric Algebra Transformer (b) Transformer

Figure 4: Loss surface around local minima of trained models on N-body dynamical system.

B Proofs

B.1 Propositions

Proposition 1. Let $f_{\alpha,\beta} \in \arg\min_{f\in\mathcal{H}} \widehat{\mathcal{L}}_{\text{total}}(f;\alpha,\beta)$ be an empirical minimizer of the REMUL objective, and let $f^{\star}_{\text{obj}} \in \arg\min_{f\in\mathcal{H}} \widehat{\mathcal{L}}_{\text{obj}}(f)$ be an empirical minimizer for the objective loss alone. Then:

(a) $f_{\alpha,\beta}$ is Pareto optimal for the bi-objective problem $(\min \widehat{\mathcal{L}}_{\text{obj}}(f), \min \widehat{\mathcal{L}}_{\text{equi}}(f))$.

(b) The following trade-off inequality holds:

$$\widehat{\mathcal{L}}_{\text{obj}}(f_{\alpha,\beta}) - \widehat{\mathcal{L}}_{\text{obj}}(f^{\star}_{\text{obj}}) \leq \frac{\beta}{\alpha}\left(\widehat{\mathcal{L}}_{\text{equi}}(f^{\star}_{\text{obj}}) - \widehat{\mathcal{L}}_{\text{equi}}(f_{\alpha,\beta})\right). \tag{10}$$

Proof. Let $f_{\alpha,\beta}$ be an empirical minimizer of $\widehat{\mathcal{L}}_{\text{total}}(f;\alpha,\beta)$. By definition, for any $\tilde{f} \in \mathcal{H}$:

$$\alpha\,\widehat{\mathcal{L}}_{\text{obj}}(f_{\alpha,\beta}) + \beta\,\widehat{\mathcal{L}}_{\text{equi}}(f_{\alpha,\beta}) \leq \alpha\,\widehat{\mathcal{L}}_{\text{obj}}(\tilde{f}) + \beta\,\widehat{\mathcal{L}}_{\text{equi}}(\tilde{f}).$$

To show Eq. 10: Rearrange the optimality condition:

$$\alpha(\widehat{\mathcal{L}}_{\text{obj}}(f_{\alpha,\beta}) - \widehat{\mathcal{L}}_{\text{obj}}(\tilde{f})) \leq \beta(\widehat{\mathcal{L}}_{\text{equi}}(f') - \widehat{\mathcal{L}}_{\text{equi}}(f_{\alpha,\beta})).$$

Dividing by $\alpha > 0$:

$$\widehat{\mathcal{L}}_{\text{obj}}(f_{\alpha,\beta}) - \widehat{\mathcal{L}}_{\text{obj}}(\tilde{f}) \leq \frac{\beta}{\alpha}(\widehat{\mathcal{L}}_{\text{equi}}(\tilde{f}) - \widehat{\mathcal{L}}_{\text{equi}}(f_{\alpha,\beta})).$$

Setting $\tilde{f} = f^{\star}_{\text{obj}}$ yields Eq. 10.

For Pareto Optimality: Assume, for contradiction, that $f_{\alpha,\beta}$ is not Pareto optimal. Then there exists an $\widetilde{\widetilde{f}} \in \mathcal{H}$ such that $\widehat{\mathcal{L}}_{\text{obj}}(\widetilde{\widetilde{f}}) \leq \widehat{\mathcal{L}}_{\text{obj}}(f_{\alpha,\beta})$ and $\widehat{\mathcal{L}}_{\text{equi}}(\widetilde{\widetilde{f}}) \leq \widehat{\mathcal{L}}_{\text{equi}}(f_{\alpha,\beta})$, with at least one of these inequalities being strict. Since $\alpha > 0$ and $\beta > 0$, this would imply $\alpha\widehat{\mathcal{L}}_{\text{obj}}(\widetilde{\widetilde{f}}) + \beta\widehat{\mathcal{L}}_{\text{equi}}(\widetilde{\widetilde{f}}) < \alpha\widehat{\mathcal{L}}_{\text{obj}}(f_{\alpha,\beta}) + \beta\widehat{\mathcal{L}}_{\text{equi}}(f_{\alpha,\beta})$. This contradicts the assumption that $f_{\alpha,\beta}$ is a minimizer of $\widehat{\mathcal{L}}_{\text{total}}(f;\alpha,\beta)$. Therefore, $f_{\alpha,\beta}$ must be Pareto optimal. □

B.2 Equivariance Measure

We define the equivariance metric E to quantify the degree of equivariance exhibited by a function f, as:

$$E(f,G) = \frac{1}{|\mathcal{D}|}\sum_{x\in\mathcal{D}}\left\|\frac{1}{M}\sum_{i=1}^{M}\rho(g_i)(f(x)) - \frac{1}{M}\sum_{i=1}^{M}f(\phi(g_i)(x))\right\|_2 \tag{11}$$

Proof. Starting from Eq. 1: $f(\phi(g)(x)) = \rho(g)(f(x)$, the group integration of both sides *w.r.t.* the normalized Haar measure μ yields:

$$\int_G f(\phi(g)(x))\,d\mu(g) = \int_G \rho(g)(f(x))\,d\mu(g) \tag{12}$$

When G is a large or continuous group, as is the case in our work, the integrals over G may not be computable in closed form. Therefore, we approximate the integrals using a Monte Carlo approach with samples $\{g_i\}_{i=1}^{M}$ from G:

$$\int_G f(\phi(g)(x))\,d\mu(g) \approx \frac{1}{M}\sum_{i=1}^{M}f(\phi(g_i)(x)) \tag{13}$$

$$\int_G \rho(g)(f(x))\,d\mu(g) \approx \frac{1}{M}\sum_{i=1}^{M}\rho(g_i)(f(x)) \tag{14}$$

where M is a large number of samples from G.

Given the group averages, we can then define the equivariance error $E(f,G)$ as the average norm of the difference between these two averages over the data distribution $\mathcal{D}$:

$$E(f,G) = \frac{1}{|\mathcal{D}|}\sum_{x\in\mathcal{D}}\left\|\frac{1}{M}\sum_{i=1}^{M}\rho(g_i)(f(x)) - \frac{1}{M}\sum_{i=1}^{M}f(\phi(g_i)(x))\right\|_2 \tag{15}$$

with $\|\cdot\|_2$ denotes an L_2 norm (for non-scalar function).

□

C Implementation Details

C.1 Equivariance Loss

The empirical equivariance loss defined in Eq. 6, $\widehat{\mathcal{L}}_{\text{equi}}(f_\theta) = \frac{1}{n}\sum \mathbb{E}_{g\sim G}[\ell(f_\theta(\phi(g)x_i), \rho(g)y_i)]$, measures the consistency of the model's predictions on transformed inputs against the correspondingly transformed ground truth labels. It is distinct from a direct measure of functional equivariance, which compare $f_\theta(\phi(g)x_i)$ with $\rho(g)f_\theta(x_i)$ (the transformed prediction of the original input). While the latter directly assesses the equivariance of the function f_θ itself, our choice of $\widehat{\mathcal{L}}_{\text{equi}}$ offers a crucial advantage: it continuously anchors the learning process to the ground truth. To see this, let $f_\theta(x) = y(x) + \gamma(x)$, where $y(x)$ is the true label for input x and $\gamma(x)$ is the model's prediction

error. If we assume the ground truth data itself is perfectly equivariant, i.e., $y(\phi(g)x) = \rho(g)y(x)$, then the term minimized by $\widehat{\mathcal{L}}_{\text{equi}}$ (for a single instance, taking ℓ as an L_p norm) becomes:

$$\| \underbrace{f_\theta(\phi(g)x_i)}_{y(\phi(g)x_i)+\gamma(\phi(g)x_i)} - \underbrace{\rho(g)y_i}_{y(\phi(g)x_i)} \|_p = \|\gamma(\phi(g)x_i)\|_p.$$

Thus, minimizing $\widehat{\mathcal{L}}_{\text{equi}}$ directly minimizes the magnitude of the prediction error on transformed inputs. This helps prevent the model from "drifting" into solutions that might be equivariant but incorrect (i.e., $f_\theta(\phi(g)x_i) \approx \rho(g)f_\theta(x_i)$ but both are far from $\rho(g)y_i$). In contrast, a loss term based on functional equivariance, $\|f_\theta(\phi(g)x_i) - \rho(g)f_\theta(x_i)\|_p$, would simplify to $\|\gamma(\phi(g)x_i) - \rho(g)\gamma(x_i)\|_p$. While this term directly encourages the *error itself* to be equivariant, minimizing it alone does not guarantee that the error magnitude $\|\gamma(\cdot)\|_p$ is small. Our REMUL objective, by combining $\alpha\widehat{\mathcal{L}}_{\text{obj}}(f_\theta)$ (which minimizes $\|\gamma(x_i)\|_p$ on original data) with $\beta\widehat{\mathcal{L}}_{\text{equi}}(f_\theta)$ (which minimizes $\|\gamma(\phi(g)x_i)\|_p$ on transformed data, given ideal data equivariance), aims for both accuracy and consistency under transformations. The degree to which this also induces functional equivariance in f_θ (i.e., making $\|\gamma(\phi(g)x_i) - \rho(g)\gamma(x_i)\|_p$ small) is then assessed empirically using the equivariance metrics E and E' as shown in our experiments.

C.2 N-Body Dynamical System

Following the methodology outlined in [45], the dataset for the N-body system simulation encompasses four objects per sample. The center object is assigned a mass ranging from 1 to 10, whereas the other objects are uniformly positioned at a radius from 0.1 to 1.0 with masses between 0.01 and 0.1. We structured the datasets into two setups: in-distribution and out-of-distribution (OOD). Each sample in the in-distribution dataset is subjected to a random rotation within the range $[-10°, 10°]$. REMUL and data augmentation are trained with random rotations in the same range. Conversely, the OOD dataset is designed to evaluate the model's generalization capabilities by incorporating extreme rotational perturbations, specifically with angles set within the ranges $[-180°, -90°]$ and $[90°, 180°]$. We trained on 100 samples, and each of the validation, test, and OOD datasets contains 5000 samples. For models hyperparameters and training, we follow the same settings in [45], summarized in Table 4. For REMUL, initial $\alpha = 1$.

Table 4: Hyperparameters settings for N-body dynamical system.

Hyperparameters	Geometric Algebra Transformer	SE(3)-Transformer	Transformer
#attention blocks	10	-	10
#channels	128	8	384
#attention heads	8	1	8
#multivector	16	-	-
#layers	-	4	-
#degrees	-	4	-
#training steps	50000	50000	50000
#optimizer	Adam	Adam	Adam
#batch size	64	64	64
#lr	3×10^{-4}	3×10^{-4}	3×10^{-4}

C.3 Motion Capture

Motion Capture dataset by [91] features 3D trajectory data that records a range of human motions, and the task involves predicting the final trajectory based on initial positions and velocities. We have reported results for two types of motion: Walking (Subject #35) and Running (Subject #9).

Following the standard experimental setup in the literature on this task [3, 4, 92], we apply a train/validation/test split of 200/600/600 for Walking and 200/240/240 for Running. The interval between trajectories, $\Delta T = 30$ for both tasks. For model hyperparameters, we fine-tuned around the same in Table 4 and found it the best for each model except for the Geometric Algebra Transformer we increased the attention blocks to 12. We train each model for 2000 epochs with batch size $= 12$. For MLP comparisons, all models and baselines have the same number of layers and parameters. More details in Table 5. For REMUL, $\alpha = 1$.

Table 5: Hyperparameters settings for Motion Capture dataset.

Hyperparameters	Geometric Algebra Transformer	SE(3)-Transformer	Transformer
#attention blocks	12	-	10
#channels	128	8	384
#attention heads	8	1	8
#multivector	16	-	-
#layers	-	4	-
#degrees	-	4	-
#epochs	2000	2000	2000
#optimizer	Adam	Adam	Adam
#batch size	12	12	12
#lr	3×10^{-4}	3×10^{-4}	3×10^{-4}

Hyperparameters	Equivariant MLP	RPP	PER	standard MLP
#hidden dim	532	348	532	680
#layers	3	3	3	3

C.4 Molecular Dynamics

MD17 dataset [95] is a molecular dynamics benchmark that contains the trajectories of eight small molecules (Aspirin, Benzene, Ethanol, Malonaldehyde Naphthalene, Salicylic, Toluene, Uraci). We use the same dataset split in [4, 92], allocating 500 samples for train, 2000 for validation, and 2000 for test. The interval between trajectories, $\Delta T = 5000$. We selected the Equivariant Graph Neural Networks (EGNN) architecture and its non-equivariant version GNN, as introduced by [5]. The input for GNN architecture is the initial positions along with atom types. Both architectures have the same hyperparameters, details in Table 6. For REMUL, $\alpha = 1$.

Table 6: Hyperparameters settings for MD17 dataset.

Hyperparameters	
#layers	4
#hidden dim	64
#epochs	500
#optimizer	Adam
#batch size	200
#lr	5×10^{-4}

C.5 Computational Complexity

In the computational experiment of Geometric Algebra Transformer (GATr) and Transformer, we selected models with an equivalent number of blocks and parameters. GATr incorporates a unique design that includes a multivector parameter; we adjusted the Transformer architecture to match the parameter count of GATr. Both models have around 2.6M parameters, detailed configurations are provided in Table 7. SE(3)-Transformer gives out of memory for this setting. We selected a uniformly random Gaussian input with 20 nodes and 7 features dimension. We measured the computational efficiency of each model by recording the time taken for both forward and backward passes during training, as well as the inference time as a function of batch size. For each value, we took the average over 10 runs with Nvidia A10 GPU.

Table 7: Hyperparameters settings for Computational Complexity.

Hyperparameters	Geometric Algebra Transformer	Transformer
#attention blocks	12	12
#channels	128	168
#attention heads	8	8
#multivector	16	-

C.6 Compute Resources

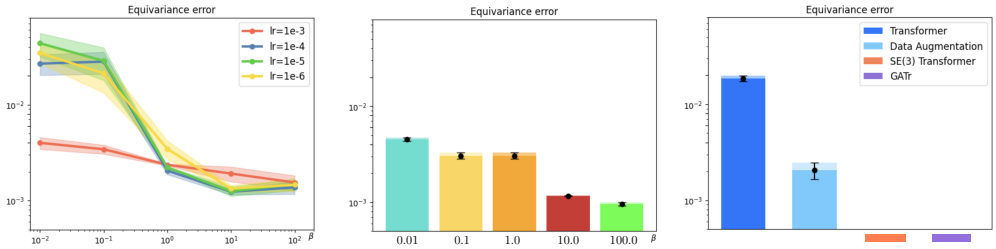
We ran all the experiments using a single Nvidia A10 GPU.

D Additional Experiments

This section provides additional experimental results to further validate our training procedure REMUL. We include:

- **Additional evaluations on the three tasks**: For the N-body dynamical system, Motion Capture, and Molecular Dynamics (MD17), we include the standard equivariance error E' (defined in Eq. 9 of the main paper). These results are consistent with our findings in the paper, using the E metric.
- **Performance on MD17:** We include detailed results showing performance and equivariance error trade-offs for REMUL applied to GNN architecture on individual molecules from the MD17 dataset (complementing Table 3 in the main paper).
- **Ablation on Group Sampling:** We conduct an ablation study investigating the impact of the number of group samples used during training for REMUL and data augmentation.
- **Convergence Speed Analysis:** We compare the convergence speed of REMUL against data augmentation by tracking training and validation MSE as a function of training steps.
- **Axis-Specific Equivariance Error on Motion Capture:** To further investigate the nature of symmetries in the Motion Capture dataset, we report the equivariance error around individual X, Y, and Z axes, separately.
- **Additional N-Body System Benchmark:** We evaluate REMUL (applied to a GNN model) on an additional N-body system benchmark, comparing it against several equivariant architectures.
- **Molecular Structures:** We provide 2D and 3D visualizations of the molecules from the MD17 dataset.

D.1 N-Body Dynamical System



(a) REMUL: Gradual penalty (b) REMUL: Constant penalty (c) Baselines

Figure 5: N-body dynamical system. The second equivariance measure E'. (a) Transformer trained with REMUL (gradual penalty), (b) Transformer trained with REMUL (constant penalty), and (c) Baselines: Equivariant models, standard Transformer, and data augmentation.

D.2 Number of Samples from the Symmetry Group

We conduct ablation studies on the number of samples required from the symmetry group during training. We compare our training procedure with data augmentation method. We selected the N-body dynamical system with the same training details and hyperparameters indicated in Appendix C.2. As shown in Figure 6, REMUL achieves better performance using fewer samples from the symmetry group compared to data augmentation.

D.3 Convergence Speed Analysis

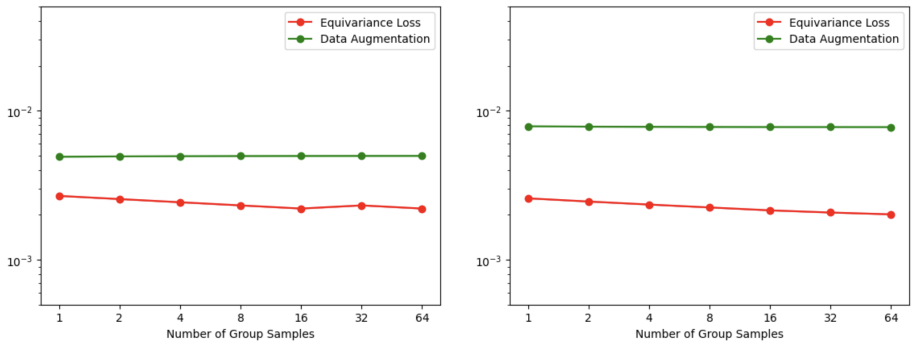
To assess the training efficiency of REMUL relative to data augmentation, we analyzed their convergence behavior on the N-body dynamical system. Both REMUL and DA were applied to a standard Transformer using the same experimental settings described in Appendix C.2. We report the Mean Squared Error (MSE) of the training and validation samples as a function of the training steps. The results, presented in Figure 7, indicate that REMUL achieves lower training and validation errors compared to data augmentation.

D.4 Additional Benchmark

To further assess REMUL's capabilities, we evaluated it on the N-body system benchmark from [5]. We applied REMUL to a Graph Neural Network (GNN) architecture, using the same hyperparameter configurations in [5] for a fair comparison. Table 8 compares our approach against several equivariant models, including EGNN [5], SEGNN [6], FA-GNN [24], and TFN [14]. The results demonstrate that REMUL achieves strong performance, outperforming EGNN and FA-GNN while being competitive with SEGNN, despite the latter incorporates more specialized geometric features.

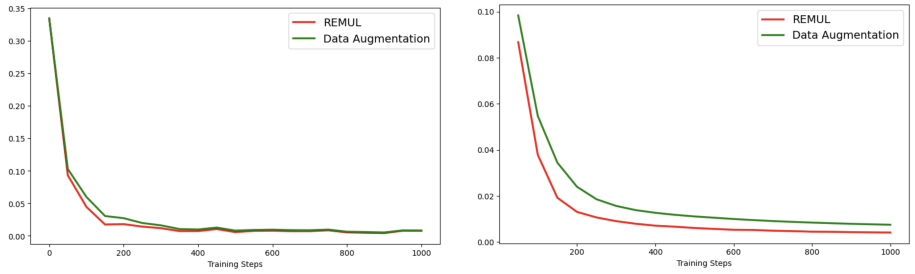
Table 8: Additional benchmark on N Body system.

	MSE
SE(3)-Tr	0.0244
TFN	0.0155
MPNN	0.0107
EGNN	0.0071
SEGNN	0.0043
FA-GNN	0.0057
REMUL-GNN	0.0046



(a) MSE: In-distribution (b) MSE: Out-of-distribution

Figure 6: Performance comparison of REMUL and data augmentation on N-body dynamical system, using different numbers of samples from the symmetry group.



(a) MSE: Training samples (b) MSE: Validation samples

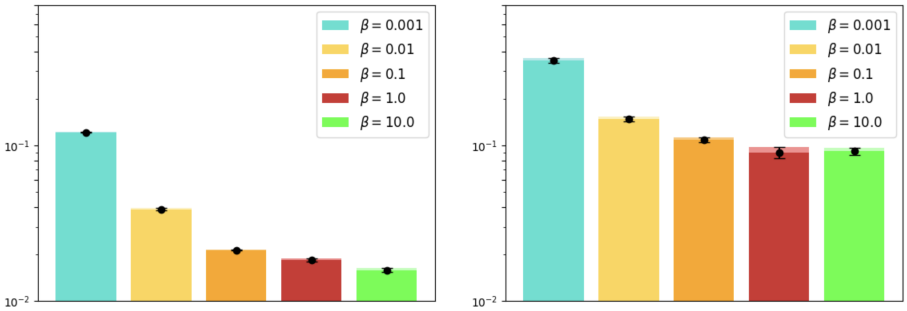
Figure 7: Comparison of convergence speed between REMUL and data augmentation on the N-body system.

D.5 Motion Capture

In the main paper (Section 6.2) we note that human motion may lack full SO(3) symmetry, particularly along the vertical (gravity) axis. To investigate this further, we measured the equivariance error E separately for rotations applied around X, Y, and Z axes. We use the best performing REMUL-Transformer models on Motion Capture dataset (specifically, $\beta = 0.1$ for the Walking task and $\beta = 0.01$ for the Running task). The results are presented in Table 9. For both Walking and Running tasks, the equivariance error associated with rotations around the Z-axis is higher than the errors for X-axis and Y-axis, which supports that the underlying dynamics in the Motion Capture exhibit a lesser degree of symmetry *w.r.t.* Z-axis, and aligns with our observation that models with relaxed equivariance (intermediate β) perform best on this task.

Table 9: Motion Capture: Equivariance error around different X, Y, and Z axis separately.

	Walking	Running
X	0.0047	0.026
Y	0.0034	0.031
Z	0.0084	0.042



(a) Equiv. error: Walking (b) Equiv. error: Running

Figure 8: Motion Capture: Transformer trained with REMUL. The second equivariance measure E'.

D.6 Molecular Dynamics

In the main paper (Section 6.3 and Table 3), we present the performance of REMUL applied to GNN architecture on the MD17 dataset. To provide more insights into how REMUL behaves across different molecular structures and how the equivariance penalty β affects task performance and equivariance error, we illustrate these relationships in Figures 9–11. For each molecule in the MD17 dataset, we trained a standard GNN using REMUL procedure with varying values of β. All experiments use the same training settings detailed in Appendix C.4.

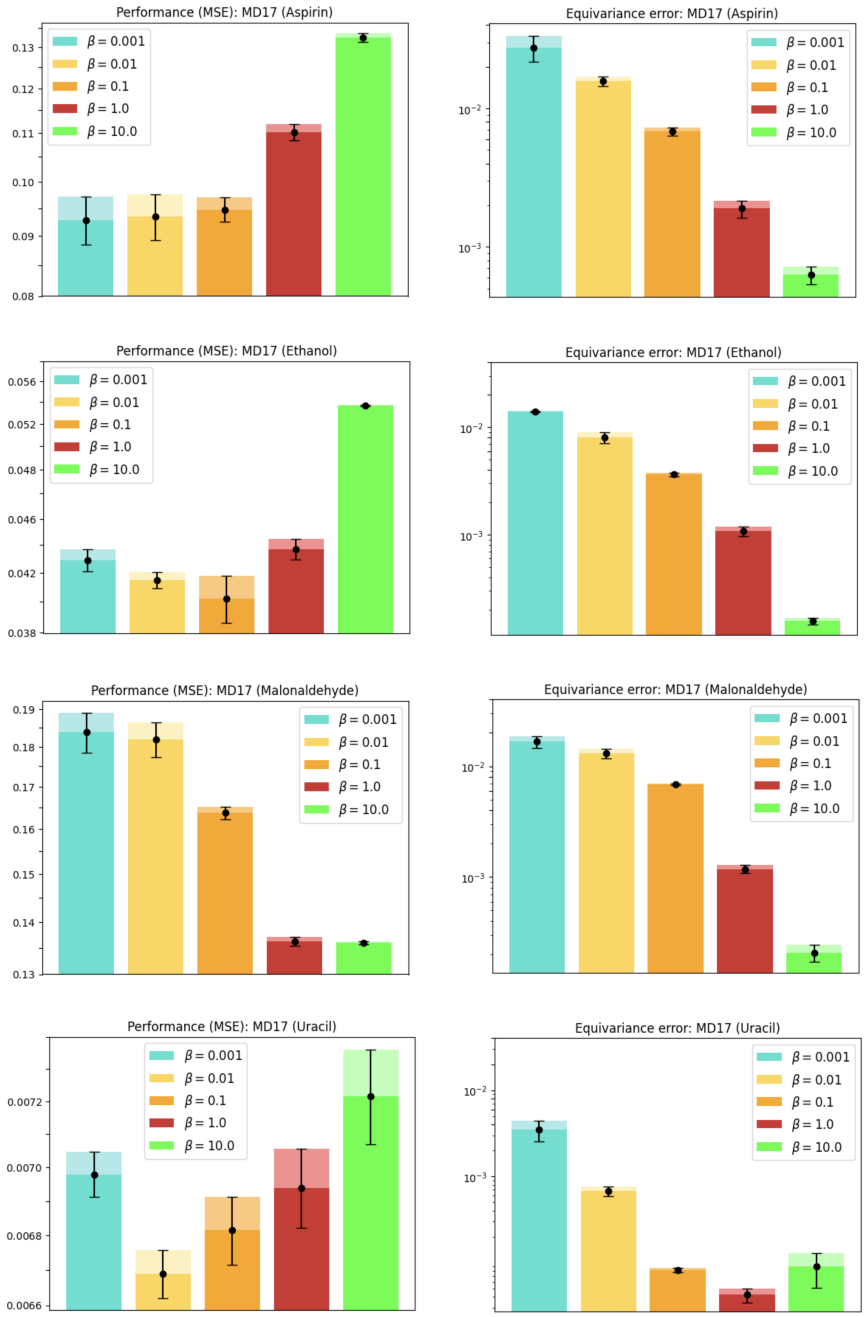


Figure 9: MD17 dataset: GNN trained with REMUL. The first column is model performance (MSE), and the second column is equivariance error E. Rows from top to bottom represent Aspirin, Ethanol, Malonaldehyde, and Uracil, respectively. The equivariance error decreases on all molecules with a higher value of β. In contrast, the required equivariance for best model performance varies for each molecule.

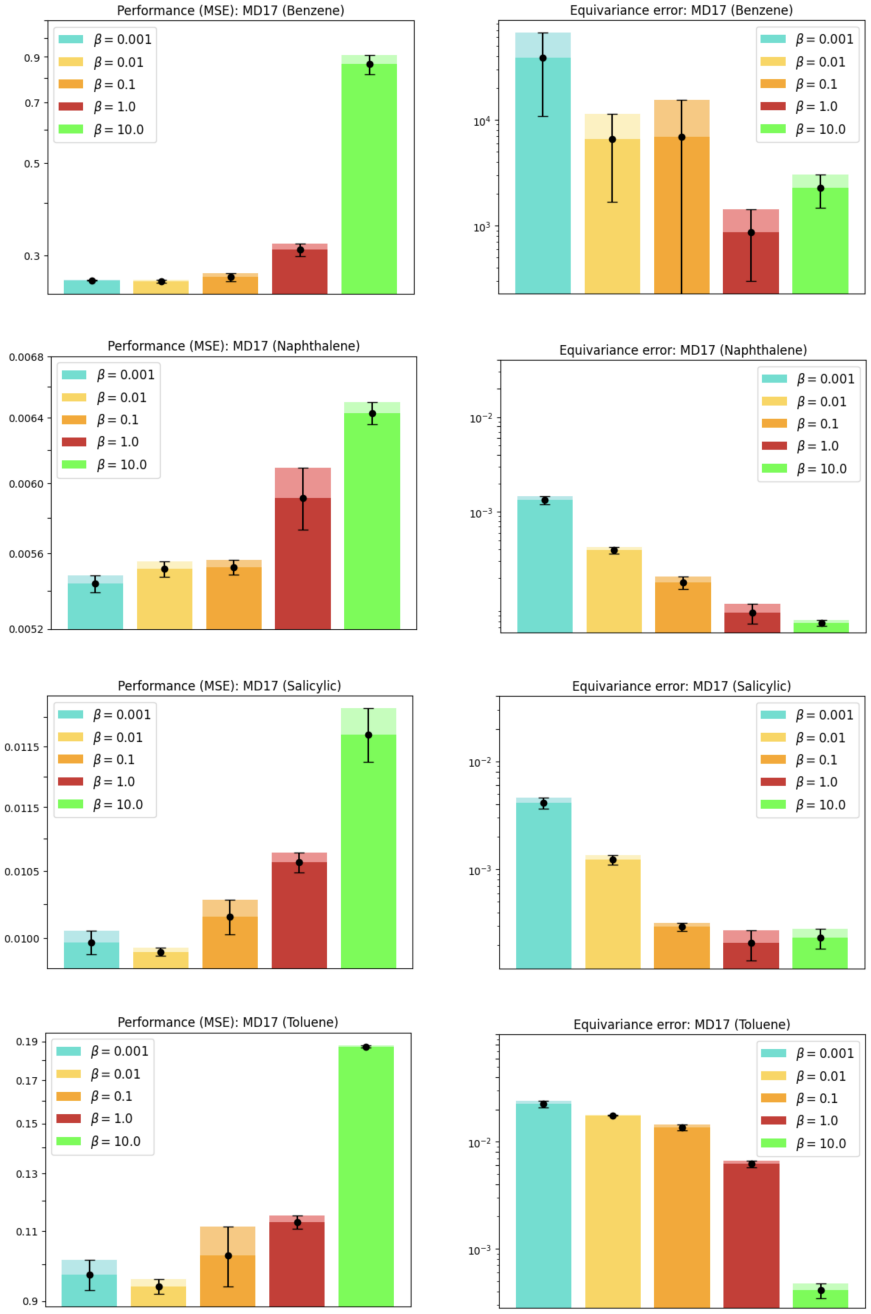


Figure 10: MD17 dataset: GNN trained with REMUL. The first column is model performance (MSE), and the second column is equivariance error E. Rows from top to bottom represent Benzene, Naphthalene, Salicylic, and Toluene, respectively.

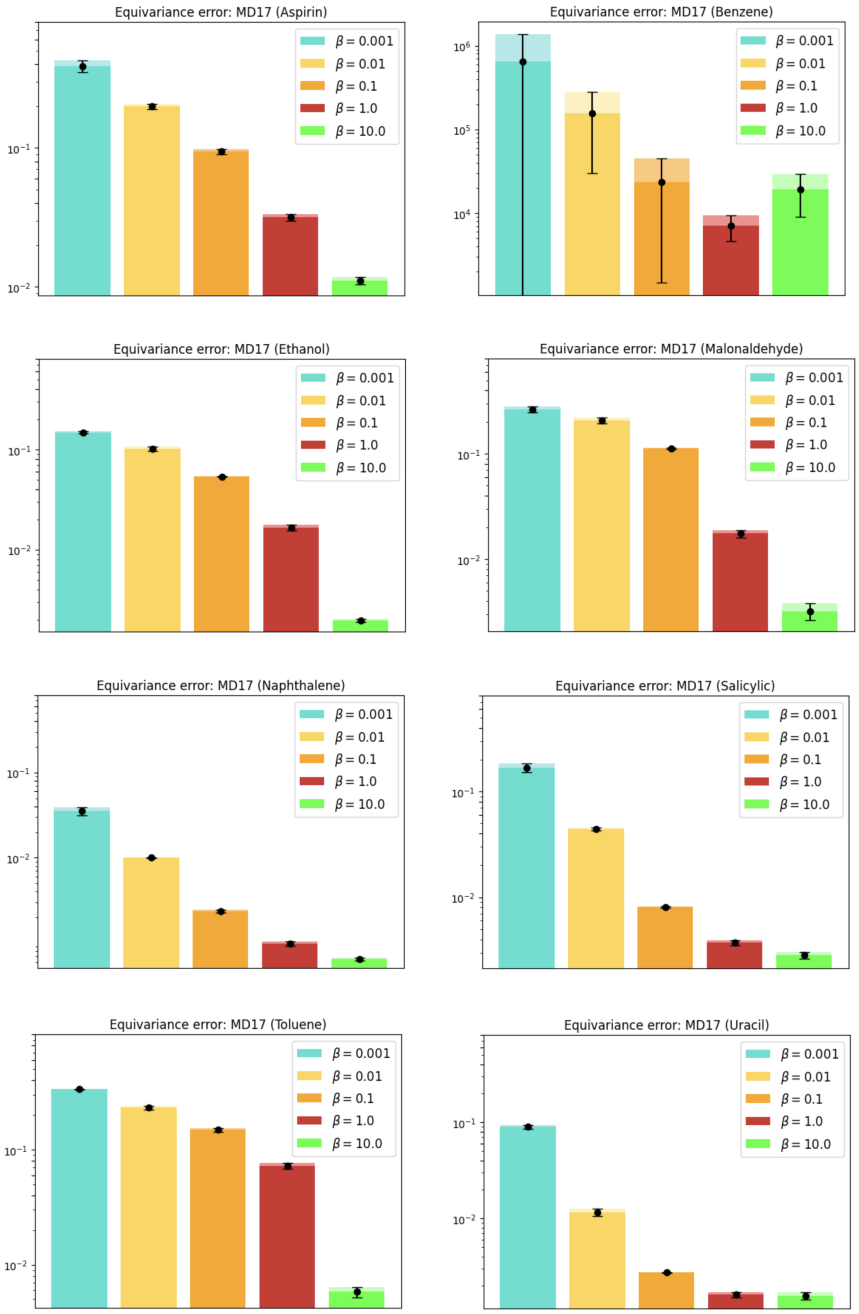
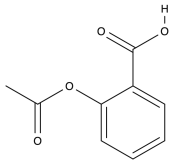
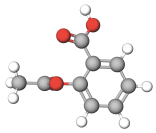
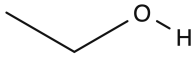
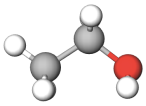


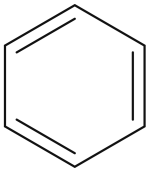
Figure 11: MD17 dataset: GNN trained with REMUL. The second equivariance measure E'.

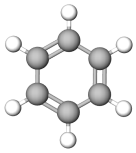

(a) Aspirin (2D)

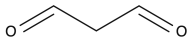

(b) Aspirin (3D)

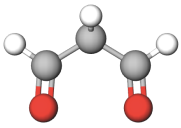

(c) Ethanol (2D)

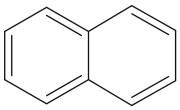

(d) Ethanol (3D)

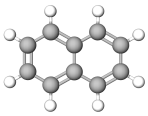

(e) Benzene (2D)

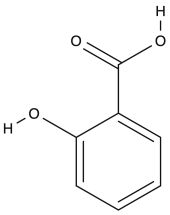

(f) Benzene (3D)

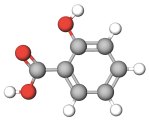

(g) Malonaldehyde (2D)

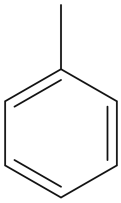

(h) Malonaldehyde (3D)

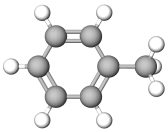

(i) Naphthalene (2D)

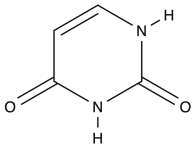

(j) Naphthalene (3D)


(k) Salicylic (2D)


(l) Salicylic (3D)


(m) Toluene (2D)


(n) Toluene (3D)


(o) Uracil (2D)

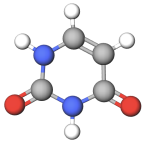

(p) Uracil (3D)

Figure 12: MD17 molecules structures.