Training Dynamics of Learning 3D-Rotational Equivariance

Anonymous Author(s)
Affiliation
Address
email

Abstract

We investigate how symmetry-agnostic models learn symmetries with data augmentation, by deriving a principled measure of equivariance error that, for convex losses, calculates the percent of total loss attributable to imperfections in learned symmetry. We focus our empirical investigation to 3D-rotation equivariance on high-dimensional molecular tasks (flow matching, force field prediction, denoising voxels) and find that models rapidly become nearly equivariant within 1k-10k training steps, a result robust to model and dataset size. This happens because learning 3D-rotational equivariance is an easier learning task, with a smoother and better-conditioned loss landscape, than the main prediction task. We then theoretically characterize learning dynamics for models that are nearly equivariant, as "stochastic equivariant learning dynamics". For 3D rotations, the loss penalty for non-equivariant models is small throughout training, so they may achieve lower test loss than equivariant models per GPU-hour unless the equivariant "efficiency gap" is narrowed.

1 Introduction

Machine learning modeling of molecules – generative modeling, property prediction, simulating dynamics, etc. – holds great potential for advancing scientific discovery and human health via therapeutics. Molecules are three-dimensional physical entities whose biochemical properties are invariant or equivariant to 3D rotations. To model these symmetries, two approaches are common: 1) use symmetry-respecting neural architectures, or 2) training symmetry-agnostic models with data augmentation, wherein training samples are randomly transformed by the symmetry group. This choice is made at the start of any molecular modeling project and can have a significant impact on engineering, training, and model performance, yet there has been a lack of clarity on when to prefer which approach.

3D-rotational equivariant architectures use sophisticated tensor operations to maintain equivariance (16), achieve loss scaling curves similar to non-equivariant models (2; 17), and are more parameter efficient than non-equivariant models on spherical image tasks (11). Yet they can be much slower (10x-100x) than non-equivariant models – though slowness is also partially from less optimized code and GPU kernels. (11; 6; 2), and they can be harder to optimize based on findings that breaking exact equivariance improves learning (24; 22; 3). We call this the *efficiency gap*, arising both from optimization speed (training steps per second) and ease (loss reduction per training step) Meanwhile, recent work achieve strong performance on molecular machine learning tasks using non-equivariant architectures with data augmentation (28; 1; 24; 10).

To answer "are symmetry-respecting architectures worth it?", one powerful principle is: *use the model that achieves better held-out loss*. In a fixed amount of GPU-hours, non-equivariant models could incur "unnecessary" equivariance error leading to higher loss, but equivariant models may

Submitted to the AI for Science workshop (NeurIPS 2025). Do not distribute.

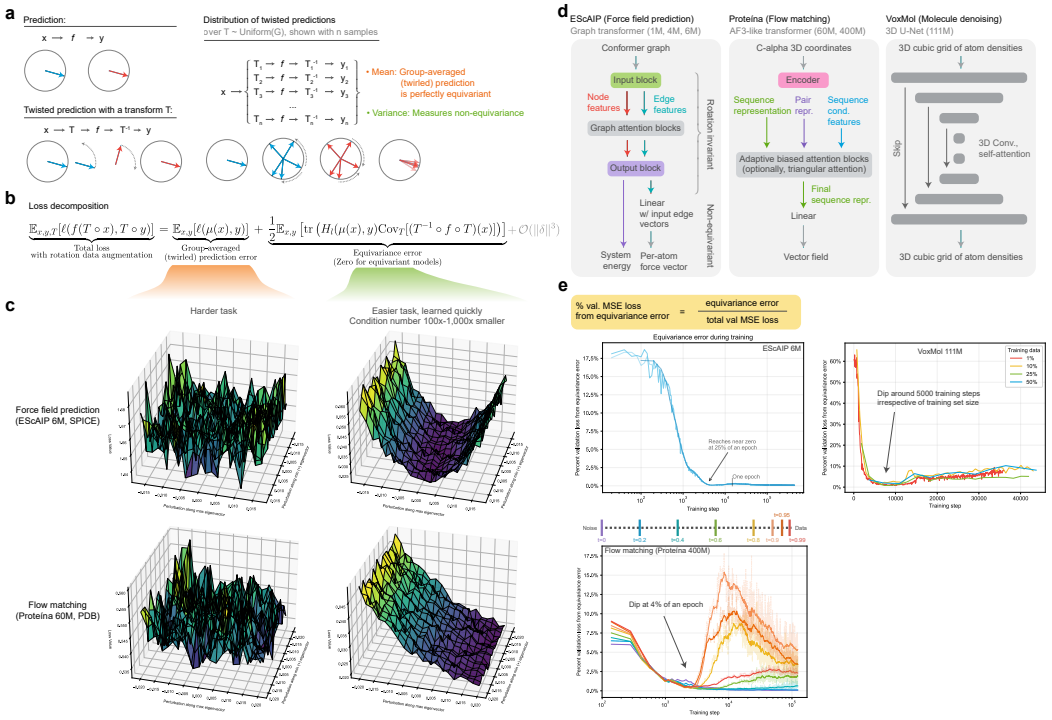

Figure 1: Overview of the paper. (a) Schematic of twisting and twirling, which underpin a principled measure of equivariance error. (b) Loss decomposition by Taylor expansion around the twirled prediction. (c) Loss landscapes for each loss component at early model checkpoints (step=500). (d) Architectures of three non-equivariant models studied here. (e) For MSE loss, the loss decomposition holds exactly, enabling computing the percent validation loss from equivariance error, which is plotted by training step in three settings.

achieve worse test loss due to the efficiency gap. In fact, we suggest the loss penalty vs. efficiency gap tradeoff is a general explanatory framework. This work focuses on equivariance, because on rotation-invariant tasks like property prediction, symmetry-respecting architectures are relatively uncontroversial (26; 9; 21): they have a minimal efficiency gap to symmetry-agnostic architectures as rotation-invariant features are informative and fast to compute, and standard deep learning operations easily preserve rotation invariance. In contrast, consider set permutation invariance where the symmetry-respecting architecture is the norm. This can be explained by observing that set transformers have minimal efficiency gap to symmetry-agnostic transformers, as set transformers simply ignore positional embeddings.

While it is possible to directly compare efficiency gaps to loss penalties from imperfect symmetry, this is easily confounded by implementation details. To provide a more fundamental insight, we instead isolate and quantify a key source of potential underperformance in symmetry-agnostic models. We develop tools to investigate: *what is the percent of a symmetry-agnostic model's loss that comes only from its failure to be perfectly equivariant?* (§2, Fig. 1A-B) In an idealized setting (ignoring efficiency differences), this characterizes the counterfactual error reduction if we had trained a symmetry-respecting model instead. In light of efficiency gaps for 3D-rotational equivariance, this metric quantifies how small the efficiency gap must become for equivariant models to outperform non-equivariant models.

In this work, we focus our empirical investigations to three high-dimensional ($\mathbb{R}^{3N} \to \mathbb{R}^{3N}$) molecular learning tasks satisfying 3D-rotational equivariance – flow matching, molecular dynamics force field prediction, and denoising voxelized atomic densities (§3, Fig. 1D). We decompose the total loss with data augmentation $\mathcal{L}(\theta) = \mathcal{L}_{\text{mean}}(\theta) + \mathcal{L}_{\text{equiv}}(\theta)$, where $\mathcal{L}_{\text{equiv}}$ captures all information about deviance from exact equivariance. In particular, for exact equivariant models, the loss relation is $\mathcal{L}(\theta) = \mathcal{L}_{\text{mean}}(\theta)$, i.e., $\mathcal{L}_{\text{equiv}} = 0$. We find:

(i) Equivariance error shrinks rapidly (1k-10k training steps; minutes). Models quickly become nearly equivariant, with equivariance error shrinking below 1% of the loss (Fig. 1E). This occurs because $\mathcal{L}_{\text{equiv}}$ is a significantly easier learning task than $\mathcal{L}_{\text{mean}}$: the loss landscape for $\mathcal{L}_{\text{equiv}}$ is significantly smoother and better conditioned (Fig. 1C). Strikingly, this is robust to model size, training set size, batch size, and optimizer: we find it with standard batch sizes as well as batch size 1, on training sets of 1M molecules to as small as 500 molecules, and on model sizes of 1M and 400M.

In §4, we theoretically characterize learning dynamics for nearly equivariant models. We analyze the relationships between the losses $\mathcal{L}_{\text{mean}}, \mathcal{L}_{\text{equiv}}$, the gradients $\nabla\mathcal{L}_{\text{mean}}, \nabla\mathcal{L}_{\text{equiv}}$, and the parameters $\theta = \theta_{\mathcal{E}} + \theta_{\mathcal{E}\perp}$ in the subspace of exactly equivariant functions and its orthogonal component. This analysis is not specific to 3D rotations.

(ii) Stochastic equivariant learning dynamics For nearly equivariant models, we can have $\nabla\mathcal{L}(\theta) \approx \nabla\mathcal{L}_{\text{mean}}(\theta)$. The learning gradients approximate the gradients of exactly equivariant models. Minibatch noise can cause fluctuations in $\mathcal{L}_{\text{equiv}}$, yet equivariance error remains small (¡10% of loss). During this phase, the parameters θ can be close to $\theta_{\mathcal{E}}$ – for the modern graph transformer architecture EScAIP, we prove that $\mathcal{L}_{\text{equiv}}$ has a globally-valid quadratic relationship with $\|\theta_{\mathcal{E}\perp}\|$.

2 Measuring Equivariance & Loss Decompositions

Let $f : \mathbb{R}^D \to \mathbb{R}^D$ be a learnable function and let G be a compact group, for instance of 3D rotations. We consider T as the matrix representation of the action of G on $\mathbb{R}^D$. A function f is G-equivariant if it commutes with all transformations $T \in G$, such that for any input $x \in \mathbb{R}^D$, we have $f(T(x)) = T(f(x))$, also written $(f \circ T)(x) = (T \circ f)(x)$. Rearranging, we observe that a perfectly equivariant function satisfies, for all x, T: $(T^{-1} \circ f \circ T)(x) = f(x)$. We call $(T^{-1} \circ f \circ T)(x)$ the *twisted prediction* for x, from the *twisted function* $T^{-1} \circ f \circ T$. To produce a twisted prediction on molecules, we sample a random rotation, use it to rotate the input molecule, pass this through the function, and un-rotate the output. The un-rotation step re-aligns the output to the "original frame" of the input molecule, which provides a canonical frame to compare the impact of different transformations on the output.

In contrast to a perfectly equivariant function, a non-equivariant function must have some distinct transformations T_1, T_2 where the twisted prediction is different: $(T_1^{-1} \circ f \circ T_1)(x) \neq (T_2^{-1} \circ f \circ T_2)(x)$. This property motivates analyzing the distribution of twisted predictions over a uniform distribution on the group, which is the usual choice for data augmentation. For a given x:

$$Z_x(T) \triangleq (T^{-1} \circ f \circ T)(x), \quad T \sim \text{Uniform}(G) \tag{1}$$

Its first central moment $\mu(x)$ is the group-averaged, or *twirled* prediction.

$$\mu(x) \triangleq \mathbb{E}_T[(T^{-1} \circ f \circ T)(x)] \tag{2}$$

By the twirling formula, $\mu(x)$ is perfectly G-equivariant (8). The second central moment of the twisted random variable is the covariance: $\text{Cov}_T(Z_x(T)) = \mathbb{E}_T\left[(Z_x(T) - \mu(x))(Z_x(T) - \mu(x))^\top\right]$. The total variance – the trace of the covariance matrix – is a natural measure of equivariance error:

$$\frac{1}{D}\mathbb{E}_{x,T}\left[\|(T^{-1} \circ f \circ T)(x) - \mu(x)\|^2\right] \tag{3}$$

2.1 Loss decomposition

Twisting and twirling provide machinery to understand a function's behavior around group actions. We can extend this machinery to analyze losses used to train models under random data augmentation, where each training point is randomly rotated. Let the data distribution $p(x, y)$ and loss function $l : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$ be invariant to G. That is, the joint data distribution $p(x, y)$ for any transformation $T \in G$ satisfies: $p(x, y) = p(T(x), T(y))$ and for any predictions z and targets y,

and for all $T \in G$: $l(T(z), T(y)) = l(z, y)$. These conditions imply that the loss-optimal model is equivariant, and that: $l((f \circ T)(x), T(y)) = l((T^{-1} \circ f \circ T)(x), y)$. The total loss over all data and transformations is:

$$\mathcal{L}(f) \triangleq \mathbb{E}_{x,y,T}\left[l((T^{-1} \circ f \circ T)(x), y)\right] \tag{4}$$

We perform a Taylor expansion of the total loss around the twirled prediction $\mu(x)$, and obtain terms involving central moments of the twisted random variable:

$$\mathcal{L}(f) = \underbrace{\mathbb{E}_{x,y}[l(\mu(x), y)]}_{\text{twirled prediction error}} + \underbrace{\frac{1}{2}\mathbb{E}_{x,y}\left[\text{tr}\left(\boldsymbol{H}_l(\mu(x), y)\text{Cov}_T[(T^{-1} \circ f \circ T)(x)]\right)\right]}_{\text{equivariance error}} + \mathcal{O}(\|\delta\|^3)$$

where $\delta = (T^{-1} \circ f \circ T)(x) - \mu(x)$, $\boldsymbol{H}_l(\mu, y)$ is the $D \times D$ Hessian matrix of the loss with respect to its first argument, and Cov_T is a $D \times D$ covariance matrix over the distribution of transformations T.

Proposition 1. *If $l(z, y) = \frac{1}{D}\|z - y\|^2$ is mean-squared error, then the total loss decomposes as:*

$\mathcal{L}(f) = \mathbb{E}_{x,y}[l(\mu(x), y)] + \frac{1}{D}\mathbb{E}_{x,T}\left[\|(T^{-1} \circ f \circ T)(x) - \mu(x)\|^2\right]$.

For MSE loss, our Taylor expansion reduces to a version of bias-variance decomposition. The equivariance error is identical to equation 3 because MSE loss places equal weight on all dimensions. These two terms are central objects of study, so we name them:

$$\mathcal{L}_{\text{mean}} \triangleq \mathbb{E}_{x,y}[l(\mu(x), y)] \tag{5}$$

$$\mathcal{L}_{\text{equiv}} \triangleq \frac{1}{D}\mathbb{E}_{x,T}\left[\|(T^{-1} \circ f \circ T)(x) - \mu(x)\|^2\right] \tag{6}$$

Percent of loss from equivariance error. Denoting model parameters as θ, under MSE loss, we can express the total loss exactly as $\mathcal{L}(\theta) = \mathcal{L}_{\text{mean}}(\theta) + \mathcal{L}_{\text{equiv}}(\theta)$. As all three terms are strictly non-negative, this implies: % MSE loss from equivariance error $= \frac{\mathcal{L}_{\text{equiv}}(\theta)}{\mathcal{L}(\theta)}$. We can further define a generalized measure of the percent of loss from equivariance error for any convex loss function with non-negative outputs. By Jensen's inequality, we have $\mathcal{L}_{\text{mean}}(\theta) \leq \mathcal{L}(\theta)$ and both terms are non-negative. Furthermore, the two terms are equal if and only if the model is exactly equivariant. This implies: % loss from equivariance error $= \frac{\mathcal{L}(\theta) - \mathcal{L}_{\text{mean}}(\theta)}{\mathcal{L}(\theta)}$.

3 Experiments

To gain insight into the empirical learning behavior of non-equivariant models, we apply our loss decomposition framework to three high-dimensional learning problems on 3D molecules, each with a distinct task and a modern non-equivariant model architecture. For each task, we follow the standard training procedure described in its original publication. Notably, all tasks use a mean-squared error loss, so our framework provides an exact decomposition of $\mathcal{L}(f)$ into $\mathcal{L}_{\text{mean}}$ and $\mathcal{L}_{\text{equiv}}$. We report both of these metrics, as well as the percentage of the total loss attributable to the model's lack of equivariance, on a validation set over the course of training. We provide complete details on methods in §E.

- **Neural Interatomic Potential (NNIP)**: We consider force prediction with EScAIP (24), a graph transformer architecture. The model predicts a 3D force vector for each atom based on density functional theory, mapping an input molecule with N atoms to an output in $\mathbb{R}^{3N}$. This task is physically equivariant to the special orthogonal group $SO(3)$ acting on atom coordinates in $\mathbb{R}^3$.
- **Probabilistic Flow Matching**: We study a generative modeling task with Proteína (10), a transformer-based architecture with similarities to AlphaFold3. The model learns to approximate the velocity field of a probability flow that transforms random noise into structured protein backbones. For a molecule with N alpha carbon atoms, the network maps noised atom coordinates and

a time $t \in [0, 1]$ to a velocity vector in $\mathbb{R}^{3N}$. The learning task is made rotationally equivariant through data augmentation, aligning it with $SO(3)$ acting on atom coordinates in $\mathbb{R}^3$.

- **Denoising Voxelized Atomic Densities**: We analyze a denoising autoencoder task with VoxMol (23; 20), a non-equivariant 3D convolutional neural network. Molecules are represented as densities in a cubic voxel grid. For a grid length g and a atom types, the input and output are tensors of shape $[g, g, g, a]$. This learning task is made rotationally equivariant through data augmentation using 16 axis-preserving 90-degree rotations of a cube, which do not introduce discretization artifacts due to aliasing. These rotations are a subset of the full octohedral group O.

3.1 Force field prediction with EScAIP

We trained EScAIP 6M on a subset of SPICE with 950k training examples used by (24) for 30 epochs with batch size 64. SPICE is a dataset with of small molecule 3D conformers with energies and forces computed by quantum-mechanical density functional theory (5). We varied model size from 1M, 4M and 6M, varied training set size from 950k, 50k, 5k, and 500 (with batch size 1), and varied the optimizer or learning rate. We observe the following:

- **Equivariance is learned early and quickly, in a manner robust to training set size, model size, and optimizer and learning rate.** The percent validation loss from equivariance error rapidly plummets in the first stage of training to under 0.1% within 1k-10k training steps (Fig. 2A-B). Notably, this speed is independent of epoch or training set size - with a 950k training set, this occurs 25% through the first epoch. Training with 500 datapoints with batch size 1, this occurs at the fourth epoch. The dip is least affected by changing model size (Fig. 2E), and most affected by the optimizer and learning rate (Fig. 2F).
- **Equivariance is learned quickly because its an easier learning task than the main prediction task.** The loss landscape (Fig. 1C) for the equivariance error is much smoother and better conditioned, with a 1,000x lower condition number, than the loss landscape for the twirled prediction error.
- **After a near-universal dip, percent loss from equivariance error can increase mildly.** In the default setting, the percent increases from 0.1% to 0.3%. This is explained by a plateau in the equivariance error while the twirled prediction error continues to decrease (Fig. 2C).
- **Typical models converge to being nearly equivariant, with percent validation loss from equivariance error under 0.1%.** The exception is training on 500 or 5k examples only: equivariance error continues to increase as training progresses, whereas equivariance error decreases in the long-term for larger training set sizes (Fig. 2D, Supp. Fig. 6).

3.2 Flow matching with Proteína

We trained Proteína at 60M without triangular attention and 400M with triangular attention on the full Protein databank (PDB) dataset with 225k training examples. We also trained models on 1% of the PDB with 2k examples and 0.1% with 200 examples. Flow matching trains a model jointly over t, flow matching time, ranging from $t = 0$ for noise and $t = 1$ for data. We measure metrics at $t = 0, 0.2, 0.4, 0.6, 0.8, 0.9, 0.95$, and 0.99, and use red colors for high t close to the data, and blue-purple colors for low t near noise in Figure 3. We observe the following:

- **The equivariance learning dip occurs early for all t, in a manner robust to training set size and model size**. Following the dip at 1k-10k training steps (Fig. 3A), low t (closer to noise) are more equivariant, while high t (closer to data) are less equivariant, with spikes to 10% validation loss from equivariance error for $t \in [0.8, 0.9, 0.95]$. This holds for the 400M model (Fig. 3E), and 60M model trained on 1% and 0.1% of the PDB (Fig. 3F-H). The dip occurs 4% through one epoch when trained on the full PDB, but occurs around epoch 53 when trained on 0.1% of the PDB.
- **After training, the model is approximately equivariant for all t, but less so around $t = 0.9$.** After one million training steps, the percent validation loss by t is plotted in Fig. 3B. The percent loss peaks at $t = 0.9$ at 6%, and is relatively lower at the extremes $t = 0.99$ at 3% and $t = 0$ at 0.04%. Task difficulty (measured by MSE loss) is harder at lower t (Fig. 3C), so $t = 0.9$ obtains low absolute equivariance error (Fig. 3D), but also low twirled prediction error.

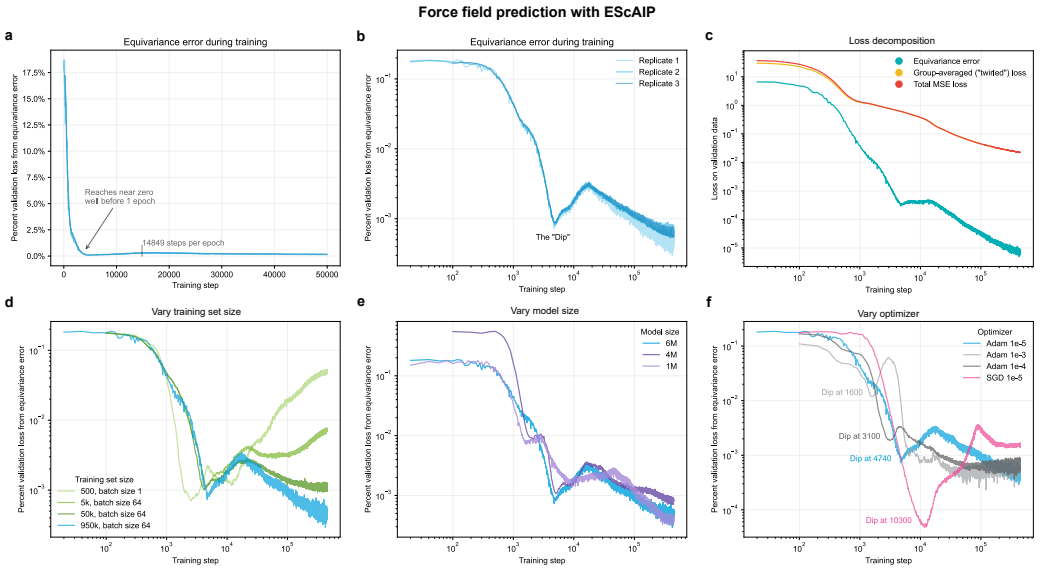

Figure 2: Training dynamics of learning equivariance in EScAIP (Force field prediction). (a-c) Validation losses and percent validation loss from equivariance error during training, early in training (a), with log-log axes (b), and decomposed into separate terms (c). (d-f) Impact of varying training set size (d), model size (e), and optimizer or learning rate (f).

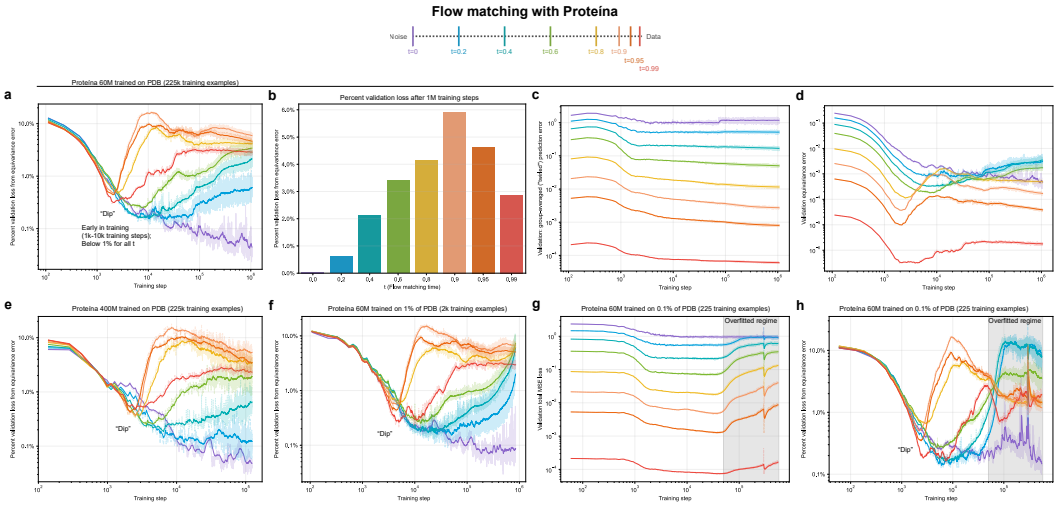

Figure 3: Training dynamics of learning equivariance in Proteína (Flow matching). Colors indicate flow matching time, with noise at $t = 0$ and data at $t = 1$. (a) Percent validation loss from equivariance error during training. (b) Bar plot of the percent validation loss from equivariance error, by flow matching time, at a final checkpoint after 1M training steps. (c-d) Validation losses by training step. (e-h) Impact of varying model size (e), training set size (f-h).

3.3 Denoising voxelized atomic densities with VoxMol

We trained VoxMol 111M on GEOM-drugs, a dataset of 3D structures of drug-like molecules with 1.1M training examples. We also trained models on 1% (11k), 10% (110k), 25% (275k), and 50% (550k) examples, and and models of varying size: full (111 M parameters), small (28 M), and tiny (7 M).. We observe:

- **The equivariance learning dip occurs early for all t, in a manner robust to training set and model size**. Across the training set sizes, all models rapidly reduce their percent validation loss

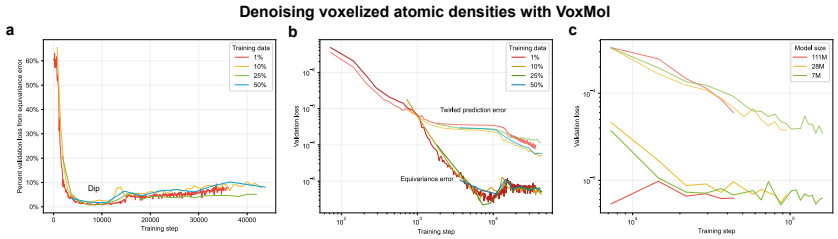

Figure 4: Training dynamics of learning equivariance in VoxMol (Denoising voxelized atomic densities). (a) Percent validation loss from equivariance error during training. (b-c) Validation losses by training step.

from equivariance error from an initial 60% to 3% or less within 1k-10k training steps (Fig. 4A-B). At 50k training steps, models have around 5-10% validation loss from equivariance error. Beyond 50k training steps, the twirled prediction error continues to decrease while the equivariance error plateaus, or decreases more slowly, below 1e-5 (Fig. 4C).

4 Learning Dynamics when $\mathcal{L}_{\text{equiv}} < \mathcal{L}_{\text{mean}}$

Our empirical results revealed a two-phase learning process, starting with a rapid initial reduction in equivariance error. What happens once the model is approximately equivariant, i.e., when $\mathcal{L}_{\text{equiv}} < \mathcal{L}_{\text{mean}}$? In this section, we investigate the implications this has on learning dynamics, focusing on three fundamental quantities illustrated in Figure 5: the relative magnitudes of the loss components ($\mathcal{L}_{\text{equiv}}$ vs. $\mathcal{L}_{\text{mean}}$), the norms of their respective gradients ($\|\nabla\mathcal{L}_{\text{equiv}}\|$ vs. $\|\nabla\mathcal{L}_{\text{mean}}\|$), and the model's parameter deviation from the subspace of perfectly equivariant functions ($\theta_{\mathcal{E}^\perp}$). By analyzing this interplay, we characterize the second stage of learning, which we call *stochastic equivariant learning dynamics*.

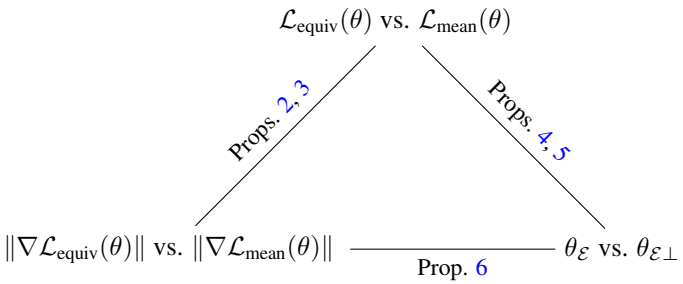

Figure 5: Diagram of theoretical relationships studied here.

We summarize our results as:

- Props. 2, 3: Under mild conditions, we prove lower bounds on the gradient purity in terms of the loss ratio. As the loss ratio shrinks, the worst possible gradient purity increases, so that learning gradients focus more on $\mathcal{L}_{\text{mean}}$.
- Props. 4, 5: We show that $\|\theta_{\mathcal{E}^\perp}\|$ has a quadratic relationship with $\mathcal{L}_{\text{equiv}}(\theta)$ for EScAIP, a modern graph transformer architecture.
- Prop. 6: We show that when $\|\theta_{\mathcal{E}^\perp}\|$ is small, $\|\nabla\mathcal{L}_{\text{equiv}}(\theta)\|$ cannot be too large.

Due to space constraints, we relegate our analysis to the appendix.

5 Discussion

In this work, we found that 3D-rotational equivariance is learned easily and quickly. We described a two-phase learning dynamic: initially, model rapidly learn equivariance. This occurs because learning equivariance is an easier task, with a smoother and better-conditioned loss landscape, than

the main prediction task. We then theoretically analyzed learning dynamics for nearly equivariant models. After training, the final percent loss from equivariance error is small for all models, but it is notably smaller for EScAIP at 0.006% than for Proteína and VoxMol (¡ 5%). While all of these loss penalties are small, and easily remedied by test-time postprocessing techniques like twirling or input frame canonicalization, this observation may also motivate research on architecture design to narrow this gap.

Intriguingly, equivariance is learned rapidly despite significant differences in model architectures. EScAIP is "nearly equivariant", as it becomes exactly equivariant with only a small change to its final linear head, yet its initial dip occurs just as quickly as Proteína and VoxMol, which are distant from being architecturally equivariant. It is also interesting that each model's latents learn (or fail to learn) to respect symmetries in different ways.

Our work establishes a principled and unified framework for quantifying equivariance error in relation to the loss. We focused our empirical study on 3D rotations, as this is a physically important symmetry group for biomolecules, but other symmetry groups may be easier or harder to learn. Looking forward, our framework could be used to study the learning dynamics of equivariance on other symmetry groups.

References

[1] 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, June 2024.

[2] Johann Brehmer, Sönke Behrends, Pim De Haan, and Taco Cohen. Does equivariance matter at scale?, 2025. URL https://openreview.net/forum?id=iIWeyfGTof.

[3] Diego Canez, Nesta Midavaine, Thijs Stessen, Arias Sebastian Fan, Jiapeng, and Alejandro Garcia. Effect of equivariance on training dynamics, July 2024.

[4] Steffen Dereich and Sebastian Kassing. Convergence of stochastic gradient descent schemes for lojasiewicz-landscapes. *CoRR*, abs/2102.09385, 2021. URL https://arxiv.org/abs/2102.09385.

[5] Peter Eastman, Benjamin P. Pritchard, John D. Chodera, and Thomas E. Markland. Nutmeg and spice: Models and data for biomolecular machine learning. *Journal of Chemical Theory and Computation*, 20(19):8583–8593, 2024. doi: 10.1021/acs.jctc.4c00794. URL https://doi.org/10.1021/acs.jctc.4c00794. PMID: 39318326.

[6] Ahmed A. Elhag, T. Konstantin Rusch, Francesco Di Giovanni, and Michael Bronstein. Relaxed equivariance via multitask learning, 2025. URL https://arxiv.org/abs/2410.17878.

[7] Fabian Fuchs, Daniel Worrall, Volker Fischer, and Max Welling. Se(3)-transformers: 3d roto-translation equivariant attention networks. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1970–1981. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/15231a7ce4ba789d13b722cc5c955834-Paper.pdf.

[8] William Fulton and Joe Harris. *Representation theory*. Graduate texts in mathematics. Springer, New York, NY, 1 edition, July 1999.

[9] Johannes Gasteiger, Muhammed Shuaibi, Anuroop Sriram, Stephan Günnemann, Zachary Ward Ulissi, C. Lawrence Zitnick, and Abhishek Das. Gemnet-OC: Developing graph neural networks for large and diverse molecular simulation datasets. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL https://openreview.net/forum?id=u8tvSxm4Bs.

[10] Tomas Geffner, Kieran Didi, Zuobai Zhang, Danny Reidenbach, Zhonglin Cao, Jason Yim, Mario Geiger, Christian Dallago, Emine Kucukbenli, Arash Vahdat, and Karsten Kreis. Proteina: Scaling flow-based protein structure generative models. In *International Conference on Learning Representations (ICLR)*, 2025.

[11] Jan Gerken, Oscar Carlsson, Hampus Linander, Fredrik Ohlsson, Christoffer Petersson, and Daniel Persson. Equivariance versus augmentation for spherical images. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 7404–7421. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/gerken22a.html.

[12] 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.

[13] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 852–863. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/076ccd93ad68be51f23707988e934906-Paper.pdf.

[14] Krzysztof Kurdyka. On gradients of functions definable in o-minimal structures. *Annales de l'institut Fourier*, 48(3):769–783, 1998. URL http://eudml.org/doc/75302.

[15] 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 (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=SCD0hn3kMHw.

[16] Shengjie Luo, Tianlang Chen, and Aditi S. Krishnapriyan. Enabling efficient equivariant operations in the fourier basis via gaunt tensor products. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=mhyQXJ6JsK.

[17] Scott Mahan, Davis Brown, Timothy Doster, and Henry Kvinge. What makes a machine learning task a good candidate for an equivariant network? In *ICML 2024 Workshop on Geometry-grounded Representation Learning and Generative Modeling*, 2024. URL https://openreview.net/forum?id=46vfUIfIo1.

[18] Oskar Nordenfors, Fredrik Ohlsson, and Axel Flinth. Optimization dynamics of equivariant and augmented neural networks. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=PTTa3U29NR.

[19] Ewa Nowara, Pedro O Pinheiro, Sai Pooja Mahajan, Omar Mahmood, Andrew Martin Watkins, Saeed Saremi, and Michael Maser. Nebula: Neural empirical bayes under latent representations for efficient and controllable design of molecular libraries. In *ICML 2024 AI for Science Workshop*, 2024.

[20] Ewa M. Nowara, Joshua Rackers, Patricia Suriana, Pan Kessel, Max Shen, Andrew Martin Watkins, and Michael Maser. Do we need equivariant models for molecule generation?, 2025. URL https://arxiv.org/abs/2507.09753.

[21] Ewa M Nowara, Joshua Rackers, Patricia Suriana, Pan Kessel, Max Shen, Andrew Martin Watkins, and Michael Maser. Do we need equivariant models for molecule generation? *arXiv preprint arXiv:2507.09753*, 2025.

[22] 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.

[23] Pedro O Pinheiro, Joshua Rackers, Joseph Kleinhenz, Michael Maser, Omar Mahmood, Andrew Martin Watkins, Stephen Ra, Vishnu Sresht, and Saeed Saremi. 3d molecule generation by denoising voxel grids. In *NeurIPS*, 2023.

[24] Eric Qu and Aditi S. Krishnapriyan. The importance of being scalable: Improving the speed and accuracy of neural network interatomic potentials across chemical domains. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=Y4mBaZu4vy.

[25] Saeed Saremi and Aapo Hyvärinen. Neural empirical bayes. *Journal of Machine Learning Research*, 20, 2019. ISSN 1532-4435.

[26] Nima Shoghi, Adeesh Kolluru, John R. Kitchin, Zachary Ward Ulissi, C. Lawrence Zitnick, and Brandon M Wood. From molecules to materials: Pre-training large generalizable models for atomic property prediction. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=PfPnugdxup.

[27] Carlos Vonessen, Charles Harris, Miruna Cretu, and Pietro Liò. Tabasco: A fast, simplified model for molecular generation with improved physical quality, 2025. URL https://arxiv.org/abs/2507.00899.

[28] 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.

A Appendix

A.1 Related Work

Prior work have measured learned equivariance with a wide variety of approaches (15; 13; 10; 24; 12; 20; 7), but to our knowledge, this work is the first to derive a measure of equivariance error that is interpreted as a percent of loss. Notably, many prior measures effectively estimate equivariance error as a pairwise deviation using only two samples per datapoint, whereas we estimate variance around a mean using enough samples of the twisted prediction as necessary to obtain stable estimates. (27) use the variance of the normalized twisted prediction, but this is not interpretable as a percent of loss. They study flow matching, but their metric conflates task difficulty, which gets easier as $t \to 1$, with equivariance error. We correct for this issue, and find that $t = 0.9$ is the most problematic time for non-equivariance, whereas they find $t = 0.5$ instead. (3) find that relaxing architectures from exact equivariance improves loss landscape conditioning and achieves better loss than perfectly equivariant architectures on image super-resolution and fluid dynamics modeling.

A.2 Parameter space decomposition

Here, we describe in greater detail (18)'s mathematical framework for analyzing the geometry of neural network parameters in terms of equivariant and non-equivariant parameter subspaces.

The foundation of this framework is the representation of a network's parameters in all of its linear layers as a point in a high-dimensional vector space, denoted $\mathcal{H}$. This captures the dominant set of learnable parameters when non-linearities are fixed. The space is formally constructed as the direct sum of the parameter spaces for each individual layer: $\mathcal{H} = \bigoplus_i \text{Hom}(X_i, X_{i+1})$. Specific network architectures are assumed to have parameters in an affine subspace $\mathcal{L} \subseteq \mathcal{H}$, referred to as the space of "admissible layers". This setup is shown by construction to be expressive and capable of describing many modern neural network architectures and operations, including fully connected layers, convolutions, residual connections, and attention layers.

To define equivariance for a multi-layer network, the framework supposes that the symmetry group G acts on all input, hidden, and output spaces $(X_0, X_1, ..., X_L)$ through a series of representations, ρ_i. With this setup, the set of all parameter configurations where each linear layer is individually equivariant forms a linear subspace of $\mathcal{H}$, denoted $\mathcal{H}_G$. This set is a linear subspace because the group actions $\rho_i(g)$ is a linear operator, which means any linear combination of equivariant linear maps remains equivariant. For instance in the setting of rotations on 3D molecules, consider a linear layer with matrix A with a rotation matrix R – if it is equivariant, we have $ARx = RAx$. If A and B are both equivariant to R, then $C = c_1A + c_2B$ is also equivariant to R: $RCx = R(c_1A + c_2B)x = (c_1A + c_2B)Rx = CRx$. $\mathcal{H}_G$ is thus a linear subspace that is closed under addition and scalar multiplication.

Algebraic manipulations show that $TC_ix = C_iTx$, using:

$$\begin{aligned} TC_ix &= T(c_1A_i + c_2B_i)x \\ &= c_1A_iTx + c_2B_iTx \\ &= (c_1A_i + c_2B_i)Tx \\ &= C_iTx \end{aligned}$$

This subspace's linearity follows from the group's actions being linear transformations.

The parameters that are both architecturally admissible and perfectly equivariant then lie in the intersection of these spaces, $\mathcal{E} = \mathcal{L} \cap \mathcal{H}_G$. It further follows that if non-linearities are equivariant, which is true for the common case of nonlinearities applied element-wise, then the entire neural network function is equivariant when its parameters are in $\mathcal{E}$.

This geometric structure guarantees that any admissible parameters θ in $\mathcal{L}$ can be uniquely decomposed via orthogonal projection into two components: $\theta = \theta_{\mathcal{E}} + \theta_{\mathcal{E}\perp}$. This is possible because $\mathcal{H}$ being an inner product space allows for a unique projection onto the tangent space of the subspace $\mathcal{E}$. The component $\theta_{\mathcal{E}}$ is the projection of the parameters onto the subspace of equivariant functions ($\mathcal{E}$), while $\theta_{\mathcal{E}\perp}$ is the component in the orthogonal complement of this subspace, representing deviation from perfect equivariance.

B Theoretical Analysis of Learning Dynamics when $\mathcal{L}_{\text{equiv}} < \mathcal{L}_{\text{mean}}$

B.1 Smaller loss ratios imply purer learning gradients

Under MSE loss, our loss decomposition also applies to gradients:

$$\nabla\mathcal{L}(\theta) = \nabla\mathcal{L}_{\text{mean}}(\theta) + \nabla\mathcal{L}_{\text{equiv}}(\theta) \tag{7}$$

Denote the relative loss ratio from equivariance error as:

$$\epsilon(\theta) \triangleq \frac{\mathcal{L}_{\text{equiv}}(\theta)}{\mathcal{L}_{\text{mean}}(\theta)} \tag{8}$$

This quantity is closely related to the percentage of total loss from equivariance error (which is $\frac{\epsilon(\theta)}{1+\epsilon(\theta)}$). As $\epsilon(\theta)$ shrinks, it is plausible that $\nabla\mathcal{L}_{\text{mean}}(\theta)$ can increasingly dominate $\nabla\mathcal{L}(\theta)$, so that we have $\nabla\mathcal{L}(\theta) \approx \nabla\mathcal{L}_{\text{mean}}(\theta)$.

We will formalize this gradient alignment in terms of $\epsilon(\theta)$ in a two-stage analysis. To gain theoretical insights into the optimization dynamics, we study the ideal, full-batch gradients including exact expectations over the symmetry group. First, we derive a general result that holds everywhere in parameter space but can be vacuous near critical points. Second, we show a result that holds near global optima. Importantly, we show both results in mild conditions that hold for typical deep neural networks. Together, these results show that broadly, when $\epsilon(\theta)$ becomes smaller, learning gradients on the total loss become increasingly pure towards the group-averaged prediction task, indicating that non-equivariant models increasingly adopt equivariant learning dynamics as their approximate equivariance improves.

Our first result relies only on a mild smoothness assumption on the loss ratio $\epsilon(\theta)$, a condition satisfied for typical neural networks.

Proposition 2. *Let $\epsilon(\theta)$ be M_ϵ-smooth. For the MSE loss, the approximation $\nabla\mathcal{L}(\theta) \approx \nabla\mathcal{L}_{mean}(\theta)$ holds with relative error bounded by:*

$$\frac{\|\nabla\mathcal{L}(\theta) - \nabla\mathcal{L}_{mean}(\theta)\|}{\|\nabla\mathcal{L}_{mean}(\theta)\|} \leq \epsilon(\theta) + \frac{\mathcal{L}_{mean}(\theta)}{\|\nabla\mathcal{L}_{mean}(\theta)\|}\sqrt{2M_\epsilon\epsilon(\theta)} \tag{9}$$

Proof. Provided in D.2 □

In well-behaved regions where the gradient norm $\|\mathcal{L}_{\text{mean}}(\theta)\|$ is large (i.e., where learning does not plateau or stall), when $\epsilon(\theta)$ becomes small, the learning gradient becomes increasingly pure at focusing on the group-averaged learning task. While this upper bound holds globally, it becomes less meaningful near saddle points of $\mathcal{L}_{\text{mean}}$ where the loss value can be large, but the gradient norm can become very small. In such situations, when $\epsilon(\theta) \neq 0$, the equivariance error gradient can assist in escaping these undesired saddle points or suboptimal local minima of $\mathcal{L}_{\text{mean}}$.

In the basin of attraction of global optima where $\mathcal{L}_{\text{mean}}(\theta) = 0$, we can derive another bound on the learning gradient purity. This bound also relies on mild assumptions satisfied by typical deep neural networks, and avoids the coefficient that explodes when $\|\mathcal{L}_{\text{mean}}(\theta)\| \to 0$.

Proposition 3. *Let the model f_θ be a deep neural network constructed from analytic activation functions, and let the data distribution $p(x, y)$ have compact support. In the basin of attraction of a global minimum θ^* where $\mathcal{L}_{mean}(\theta^*) = 0$, for the MSE loss, the approximation $\nabla\mathcal{L}(\theta) \approx \nabla\mathcal{L}_{mean}(\theta)$ holds with relative error bounded by:*

$$\frac{\|\nabla\mathcal{L}(\theta) - \nabla\mathcal{L}_{mean}(\theta)\|}{\|\nabla\mathcal{L}_{mean}(\theta)\|} \leq \sqrt{\frac{2M}{c}} \cdot \sqrt{\frac{\mathcal{L}_{equiv}(\theta)}{\mathcal{L}_{mean}(\theta)^\alpha}} \tag{10}$$

where M is the resulting smoothness constant of $\mathcal{L}_{equiv}(\theta)$, and $c > 0$, $\alpha \in [1, 2)$ are the constants of the Kurdyka-Łojasiewicz (KŁ) inequality that $\mathcal{L}_{mean}(\theta)$ is guaranteed to satisfy.

Proof. Provided in §D.3. □

Experimental validation. Our theory suggests that when the loss ratio is small, the gradient norm ratio is also small. We empirically investigated this and found strong log-log correlations of Pearson R = 0.75 over training in EScAIP, and R = 0.41 to 0.90 for Proteína at $t \geq 0.2$. The only exception was Proteína at $t = 0$, which had negative correlation of -0.32.

B.2 Parameter space decomposition

In the proceeding analysis, we adopt (18)'s mathematical framework for analyzing neural network parameters in terms of equivaraint and non-equivariant parameter subspaces, which enables expressing parameters into orthogonal components: $\theta = \theta_{\mathcal{E}} + \theta_{\mathcal{E}\perp}$. In this framework, the total parameter space of a neural network is shown to have a subspace $\mathcal{E}$ corresponding to perfectly equivariant functions. It is shown that under mild conditions, the total parameter space is an inner product space, and $\mathcal{E}$ is a linear subspace, which together enable the orthogonal decomposition $\theta = \theta_{\mathcal{E}} + \theta_{\mathcal{E}\perp}$. This framework is shown to apply to a broad class of modern neural network operations and architectures, including fully connected layers with non-linearities, convolutions, residual connections, and attention layers. It also includes a broad class of symmetry groups including $SO(3)$ and all groups studied in this work. We provide more detail in §A.2 and refer the interested reader to (18).

B.3 Relating equivariance error to the deviation from equivariant parameter subspace

We will study the relationship between $\mathcal{L}_{\text{equiv}}$ and $\|\theta_{\mathcal{E}\perp}\|$. In general for neural networks, $\mathcal{L}_{\text{equiv}}$ is a complex, highly non-linear function of $\theta = \theta_{\mathcal{E}} + \theta_{\mathcal{E}\perp}$. However, we know that $\mathcal{L}_{\text{equiv}}$ is non-negative, continuous, and equal to zero iff $\theta_{\mathcal{E}\perp} = 0$. By these properties, we know that if $\|\theta_{\mathcal{E}\perp}\|$ is small, then $\mathcal{L}_{\text{equiv}}$ is small. More formally, for any $\epsilon > 0$, there exists a $\delta > 0$ such that if the parameter deviation is small ($\|\theta_{\mathcal{E}\perp}\| < \delta$), then the equivariance error is also small ($\mathcal{L}_{\text{equiv}} < \epsilon$).

We will be able to make a stronger statement specifically for the EScAIP architecture, a modern graph transformer architecture that achieved strong results on NNIP energy and force prediction tasks (24). The EScAIP architecture uses rotation-invariant features derived from an input molecular graph. Its hidden representations for atoms and edges, denoted $\boldsymbol{h}$, are rotation-invariant throughout the network. Force prediction outputs a 3D force vector at each atom in a molecule. For a single atom with a set of 3D edge vectors E (the vectors pointing from one atom to another atom) in a molecule $\boldsymbol{x}$, EScAIP predicts force vectors as:

$$\begin{bmatrix} o_x \\ o_y \\ o_z \end{bmatrix} = \sum_{e \in E} \begin{bmatrix} e_x \cdot \mathbf{w_x}^\intercal \mathbf{h}(\mathbf{e}, \mathbf{x}) \\ e_y \cdot \mathbf{w_y}^\intercal \mathbf{h}(\mathbf{e}, \mathbf{x}) \\ e_z \cdot \mathbf{w_z}^\intercal \mathbf{h}(\mathbf{e}, \mathbf{x}) \end{bmatrix} \tag{11}$$

where $\boldsymbol{e} \in \mathbb{R}^3$, $\mathbf{h}(\mathbf{e}, \mathbf{x}) \in \mathbb{R}^h$ is the last hidden representation of the edge $\boldsymbol{e}$ in molecule $\boldsymbol{x}$, and $\boldsymbol{W} = [\mathbf{w_x}, \mathbf{w_y}, \mathbf{w_z}]$, where each $\mathbf{w} \in \mathbb{R}^h$, are the parameters for a linear head with no bias. The 3D edge vectors $\boldsymbol{e}$ are rotation-equivariant with respect to the input molecule, while the hidden representation $\mathbf{h}(\mathbf{e})$ is rotation-invariant to the input molecule, but composing these to form the output prediction generally breaks both invariance and equivariance.

In particular, force predictions are equivariant if and only if the scalar projections of the hidden features are independent of the coordinate axis, i.e., $\mathbf{w_x}^\intercal \mathbf{h}(\mathbf{e}, \mathbf{x}) = \mathbf{w_y}^\intercal \mathbf{h}(\mathbf{e}, \mathbf{x}) = \mathbf{w_z}^\intercal \mathbf{h}(\mathbf{e}, \mathbf{x})$, for all inputs. Under the mild assumption of a non-degenerate learned embedding function $\mathbf{h}(\mathbf{e}, \mathbf{x})$, such that the set of all possible hidden vectors spans the feature space, this condition holds if and only if the parameter vectors themselves are identical: $\mathbf{w_x} = \mathbf{w_y} = \mathbf{w_z}$. This condition defines the subspace $\mathcal{E}$ for the EScaIP architecture. Using this, we decompose $\boldsymbol{W} = \mathbf{W}_{\mathcal{E}} + \mathbf{W}_{\mathcal{E}\perp}$ with an equivariant part $\mathbf{W}_{\mathcal{E}} = [\bar{\boldsymbol{w}}, \bar{\boldsymbol{w}}, \bar{\boldsymbol{w}}] \in \mathcal{E}$ where $\bar{\boldsymbol{w}} = \frac{1}{3}(\mathbf{w_x} + \mathbf{w_y} + \mathbf{w_z})$, and a non-equivariant part $\mathbf{W}_{\mathcal{E}\perp} = [\boldsymbol{d}_x, \boldsymbol{d}_y, \boldsymbol{d}_z] \in \mathcal{E}\perp$ where $\boldsymbol{d}_x = \mathbf{w_x} - \bar{\boldsymbol{w}}$, and same for y, z.

With this setup, we can now establish that the equivariance error of the EScaIP architecture has a quadratic relationship with the magnitude of the parameter deviation from $\mathcal{E}$, the space of perfectly equivariant functions.

Theorem 4. *For the EScAIP architecture trained with mean-squared error loss on a non-degenerate dataset, for any fixed set of upstream parameters $\theta \setminus \boldsymbol{W}$, there exist positive constants $0 < \lambda_{min} \leq$*

λ_{max} *(which depend on the model architecture, data distribution, and other parameters* $\theta \setminus \boldsymbol{W}$*) such that:*

$$\lambda_{min} \cdot \|\mathbf{W}_{\mathcal{E}\perp}\|_F^2 \leq \mathcal{L}_{equiv}(\theta) \leq \lambda_{max} \cdot \|\mathbf{W}_{\mathcal{E}\perp}\|_F^2 \tag{12}$$

Proof. Provided in D.4. □

We can generalize the preceding analysis to a broader class of neural networks. Applying a Taylor expansion to $\mathcal{L}_{\text{equiv}}(\theta)$ for the neural net f on an input x, we have: $f(x;\theta_{\mathcal{E}} + \theta_{\mathcal{E}\perp}) = f(x;\theta_{\mathcal{E}}) + J_{\theta_{\mathcal{E}\perp}} f(x;\theta_{\mathcal{E}}) \cdot \theta_{\mathcal{E}\perp} + \mathcal{O}(\|\theta_{\mathcal{E}\perp}\|^2)$ where $J_{\theta_{\mathcal{E}\perp}} f(x;\theta_{\mathcal{E}})$ is the Jacobian of the network output with respect to parameter components $\theta_{\mathcal{E}\perp}$, evaluated at $\theta_{\mathcal{E}}$. The key structure, analogous to the EScAIP argument, is the decomposition of the neural net output into a purely equivariant term, and a term linear in $\theta_{\mathcal{E}\perp}$, as well as a remainder term in this setting. With this setup, for a broad class of neural network architectures, we can relate locally near $\mathcal{E}$ that $\mathcal{L}_{\text{equiv}}$ is quadratic in $\|\theta_{\mathcal{E}\perp}\|$ (Thm. 5), and its grad norm is linear in $\|\theta_{\mathcal{E}\perp}\|$ (Thm. D.6).

Theorem 5. *For any neural network whose parameters can be expressed as* $\theta = \theta_{\mathcal{E}} + \theta_{\mathcal{E}\perp}$ *with* $\theta_{\mathcal{E}} \in \mathcal{E}$ *and* $\theta_{\mathcal{E}\perp} \in \mathcal{E}\perp$*, and for equivariance error* $\mathcal{L}_{equiv}$ *defined by the variance of the output with respect to transformations, there exist positive constants* $0 < \lambda_{min} \leq \lambda_{max}$ *such that for a non-degenerate dataset, using* $\|\cdot\|$ *to denote* L_2*-norm:*

$$\lambda_{min}\|\theta_{\mathcal{E}\perp}\|^2 + \mathcal{O}(\|\theta_{\mathcal{E}\perp}\|^3) \leq \mathcal{L}_{equiv}(\theta) \leq \lambda_{max}\|\theta_{\mathcal{E}\perp}\|^2 + \mathcal{O}(\|\theta_{\mathcal{E}\perp}\|^3) \tag{13}$$

Proof. Provided in D.5. □

Theorem 6. *Under the same conditions as Thm. 5, the norm of the gradient of the equivariance loss with respect to the non-equivariant parameters is bounded by the deviation itself. Specifically, there exists a constant* C *such that:*

$$\|\nabla_{\theta_{\mathcal{E}\perp}} \mathcal{L}_{equiv}(\theta)\| \leq C \cdot \|\theta_{\mathcal{E}\perp}\|$$

Proof. Provided in D.6. □

C Supplementary Figures

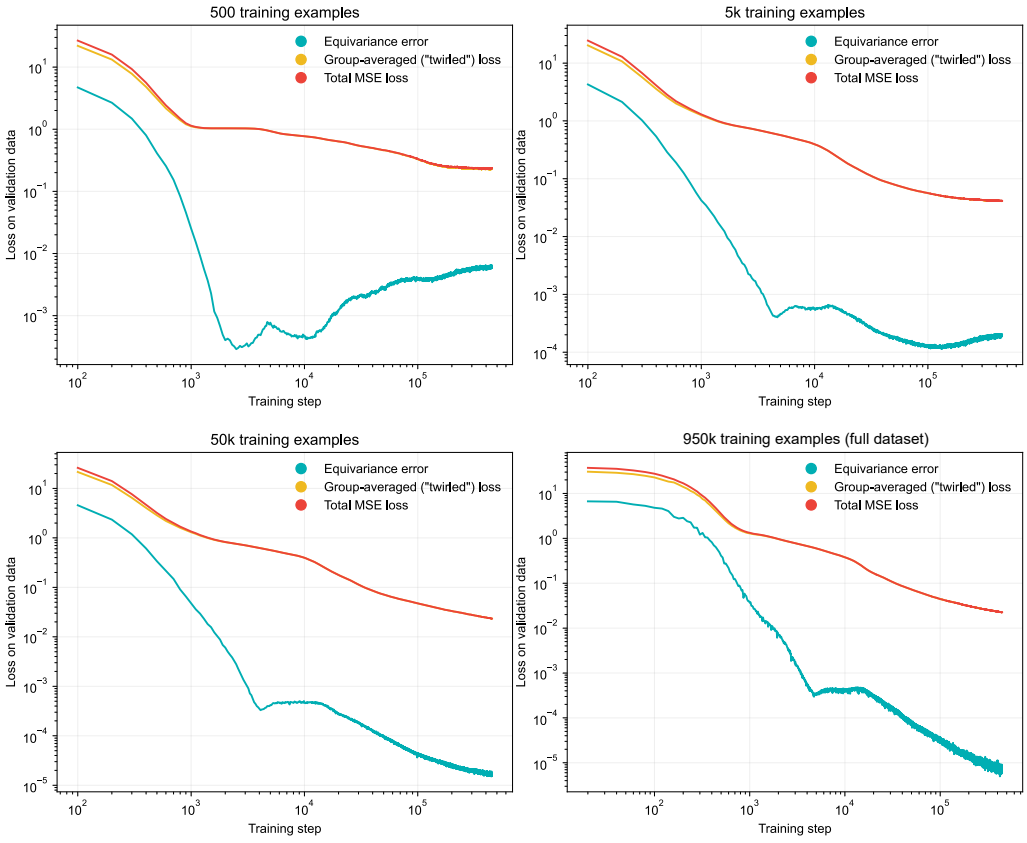

Figure 6: EScAIP: Validation loss curves over training, varied by training set size.

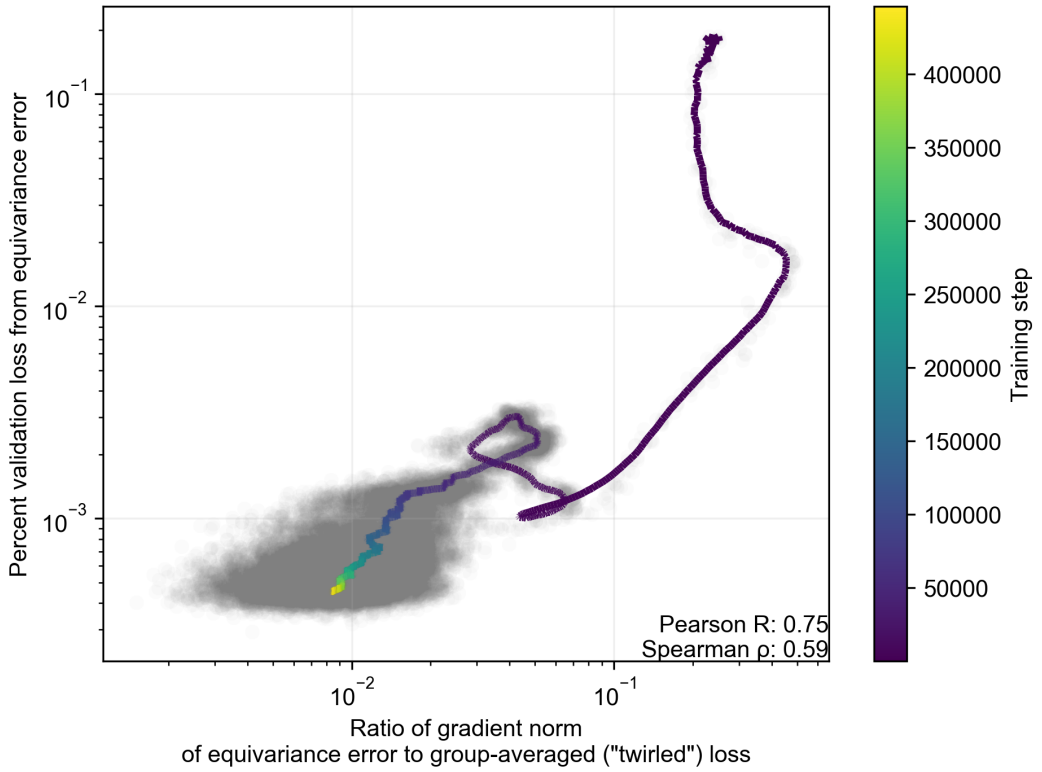

Figure 7: EScAIP: Percent validation loss from equivariance error vs. grad norm ratio, over training. Colored line indicates smoothed exponential moving average, colored by training step.

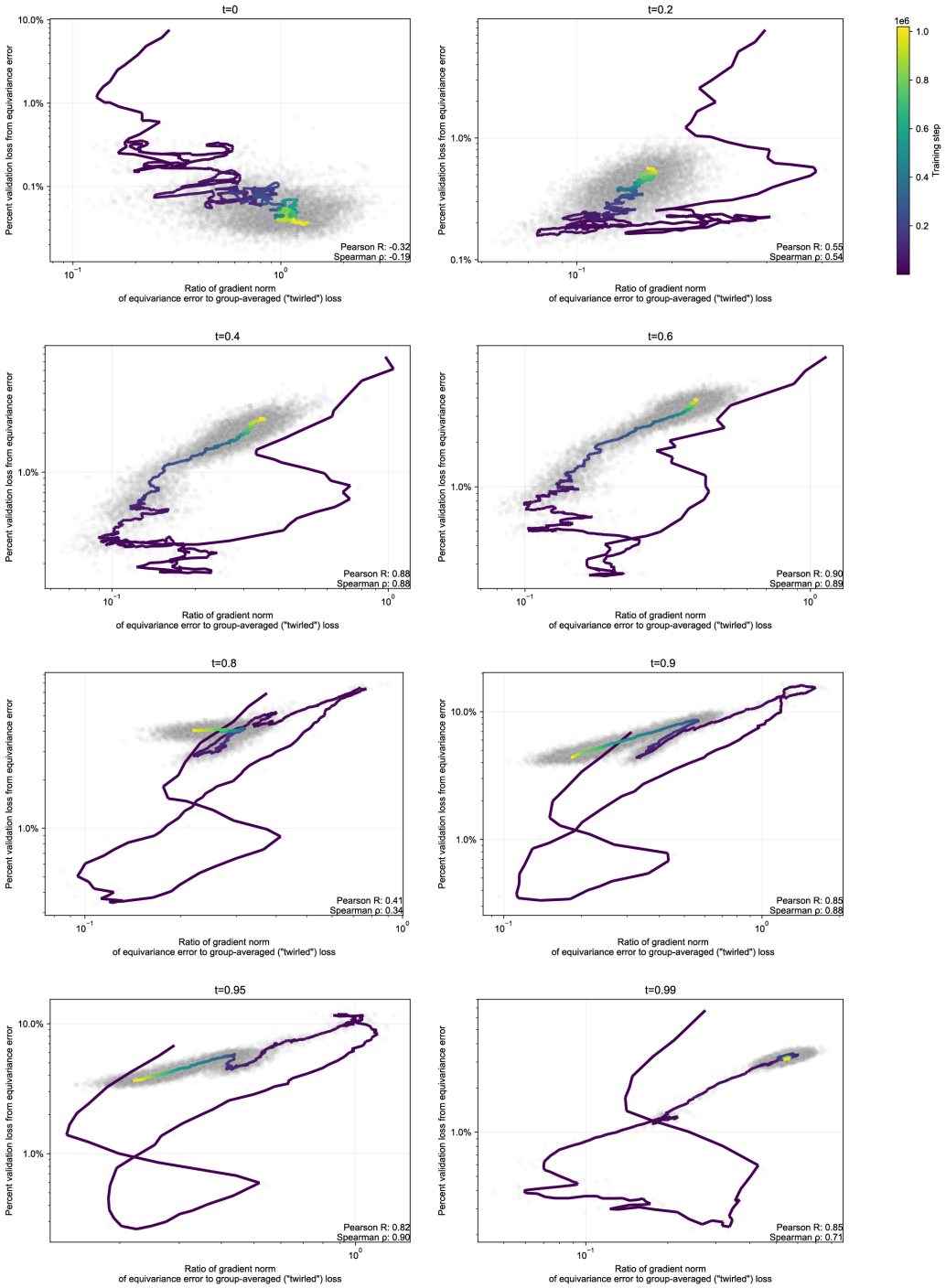

Figure 8: Proteína: Percent validation loss from equivariance error vs. grad norm ratio, over training, by flow matching time. Colored line indicates smoothed exponential moving average, colored by training step.

D Proofs

D.1 Proof of Proposition 1

Proposition. *If* $l(z, y) = \frac{1}{D}\|z - y\|^2$ *is mean-squared error, then the total loss decomposes as:*

$$\mathcal{L}(f) = \underbrace{\mathbb{E}_{x,y}[l(\mu(x), y)]}_{\textit{prediction error}} + \underbrace{\frac{1}{D}\mathbb{E}_{x,y}\left[\sum_{i=1}^{D} \textit{Var}_T[(T^{-1} \circ f \circ T)(x)_i]\right]}_{\textit{equivariance error}} \tag{14}$$

Proof. For mean-squared error, the Hessian is constant: $H_l(z, y) = \frac{2}{D}I$ where I is the $D \times D$ identity matrix. Furthermore, higher-order derivatives are zero, so the decomposition has no additional terms. The equivariance error simplifies as:

$$\frac{1}{2}\mathbb{E}_{x,y}\left[\text{tr}\left(\left(\frac{2}{D}I\right)\text{Cov}_T[\dots]\right)\right] = \frac{1}{D}\mathbb{E}_{x,y}\left[\text{tr}\left(\text{Cov}_T[\dots]\right)\right] \tag{15}$$

□

D.2 Proof of Proposition 2

Proposition. *Let $\epsilon(\theta)$ be M_ϵ-smooth. For the MSE loss, the approximation $\nabla\mathcal{L}(\theta) \approx \nabla\mathcal{L}_{mean}(\theta)$ holds with relative error bounded by:*

$$\frac{\|\nabla\mathcal{L}(\theta) - \nabla\mathcal{L}_{mean}(\theta)\|}{\|\nabla\mathcal{L}_{mean}(\theta)\|} \leq \epsilon(\theta) + \frac{\mathcal{L}_{mean}(\theta)}{\|\nabla\mathcal{L}_{mean}(\theta)\|}\sqrt{2M_\epsilon\epsilon(\theta)} \tag{16}$$

Proof. The total loss gradient is $\mathcal{L}(\theta) = (1 + \epsilon(\theta))\mathcal{L}_{\text{mean}}(\theta)$.

$$\begin{aligned}
\nabla\mathcal{L}(\theta) &= \nabla[(1 + \epsilon(\theta))\mathcal{L}_{\text{mean}}(\theta)] && (17)\\
&= \nabla\epsilon(\theta)\mathcal{L}_{\text{mean}}(\theta) + (1 + \epsilon(\theta))\nabla\mathcal{L}_{\text{mean}}(\theta) && (18)\\
\nabla\mathcal{L}(\theta) - \nabla\mathcal{L}_{\text{mean}}(\theta) &= \epsilon(\theta)\nabla\mathcal{L}_{\text{mean}}(\theta) + \mathcal{L}_{\text{mean}}(\theta)\nabla\epsilon(\theta) && (19)
\end{aligned}$$

Now, we bound the norm of this difference using the triangle inequality:

$$\|\nabla\mathcal{L}(\theta) - \nabla\mathcal{L}_{\text{mean}}(\theta)\| \leq \epsilon(\theta)\|\nabla\mathcal{L}_{\text{mean}}(\theta)\| + \mathcal{L}_{\text{mean}}(\theta)\|\nabla\epsilon(\theta)\| \tag{20}$$

Using the smoothness assumption that $\|\epsilon(\theta)\| \leq \sqrt{2M_\epsilon\epsilon(\theta)}$, we obtain the final result:

$$\frac{\|\nabla\mathcal{L}(\theta) - \nabla\mathcal{L}_{\text{mean}}(\theta)\|}{\|\nabla\mathcal{L}_{\text{mean}}(\theta)\|} \leq \epsilon(\theta) + \frac{\mathcal{L}_{\text{mean}}(\theta)}{\|\nabla\mathcal{L}_{\text{mean}}(\theta)\|}\sqrt{2M_\epsilon\epsilon(\theta)} \tag{21}$$

□

D.3 Proof of Proposition 3

Proposition 7. *Let the model f_θ be a deep neural network constructed from analytic activation functions, and let the data distribution $p(x, y)$ have compact support. In the basin of attraction of a global minimum θ^* where $\mathcal{L}_{mean}(\theta^*) = 0$, for the MSE loss, the approximation $\nabla\mathcal{L}(\theta) \approx \nabla\mathcal{L}_{mean}(\theta)$ holds with relative error bounded by:*

$$\frac{\|\nabla\mathcal{L}(\theta) - \nabla\mathcal{L}_{mean}(\theta)\|}{\|\nabla\mathcal{L}_{mean}(\theta)\|} \leq \sqrt{\frac{2M}{c}} \cdot \sqrt{\frac{\mathcal{L}_{equiv}(\theta)}{\mathcal{L}_{mean}(\theta)^\alpha}} \tag{22}$$

where M is the resulting smoothness constant of $\mathcal{L}_{equiv}(\theta)$, and $c > 0$, $\alpha \in [1, 2)$ are the constants of the Kurdyka-Łojasiewicz (KŁ) inequality that $\mathcal{L}_{mean}(\theta)$ is guaranteed to satisfy.

Proof. The network f_θ is a composition of analytic functions, making it analytic in θ. Further, the loss functions $\mathcal{L}_{\text{equiv}}, \mathcal{L}_{\text{mean}}$ preserve analyticity. Thus, both are also analytic functions of θ. $\mathcal{L}_{\text{equiv}}$ is thus M-smooth for some constant M in any compact parameter set. A foundational result states that any real-analytic function satisfies the Kurdyka-Łojasiewicz inequality (14; 4). From the M-smoothness of $\mathcal{L}_{\text{equiv}}(\theta)$, we have: $\|\nabla\mathcal{L}_{\text{equiv}}(\theta)\|^2 \leq 2M \cdot \mathcal{L}_{\text{equiv}}(\theta)$. From the KŁ condition on $\mathcal{L}_{\text{mean}}(\theta)$, we have: $\|\nabla\mathcal{L}_{\text{mean}}(\theta)\|^2 \geq c \cdot \mathcal{L}_{\text{mean}}(\theta)^\alpha$ for some constants $c > 0$ and $\alpha \in [1, 2)$ in the basin. The result follows from combining these properties. □

D.4 Proof of Proposition 4

Theorem. *For the EScAIP architecture trained with mean-squared error loss on a non-degenerate dataset, for any fixed set of upstream parameters $\theta \setminus \boldsymbol{W}$, there exist positive constants $0 < \lambda_{min} \leq \lambda_{max}$ such that:*

$$\lambda_{min} \cdot \|\mathbf{W}_{\mathcal{E}\perp}\|_F^2 \leq \mathcal{L}_{equiv}(\theta) \leq \lambda_{max} \cdot \|\mathbf{W}_{\mathcal{E}\perp}\|_F^2 \tag{23}$$

Remarks. The constants λ_{min} and λ_{max} depend on the model architecture, data distribution, and other parameters $\theta \setminus \boldsymbol{W}$.

Proof. For a molecule x, the k-th component of the predicted force vector decomposes into a sum of contributions from $\mathbf{W}_{\mathcal{E}}$ and $\boldsymbol{W}_{\mathcal{E}\perp}$:

$$o_k(x; \boldsymbol{W}) = \underbrace{\sum_{e\in E} e_k \cdot (\bar{\boldsymbol{w}}^T \mathbf{h}(\mathbf{e}))}_{o_{eq,k}(x;\mathbf{W}_{\mathcal{E}})} + \underbrace{\sum_{e\in E} e_k \cdot (\boldsymbol{d}_k^T \mathbf{h}(\mathbf{e}))}_{\Delta o_k(x;\boldsymbol{W}_{\mathcal{E}\perp})} \tag{24}$$

where the final hidden representation $\mathbf{h}$ depends on $\theta \setminus \boldsymbol{W}$, the set of upstream parameters. Recall the equivariance error from Proposition 1, and observe that the variance of $o_k = o_{eq} + \Delta o_k$ depends only on Δo_k, as o_{eq} is equivariant by construction. Thus, the equivariance error of the entire model, for a fixed set of upstream parameters and expressed as a function of the force prediction head parameters, is:

$$\mathcal{L}_{\text{equiv}}(\theta) = \mathbb{E}_{x,T}\left[\|\Delta o(Tx; \mathbf{W}_{\mathcal{E}\perp}) - \mathbb{E}_{T'}[\Delta o(T'x; \mathbf{W}_{\mathcal{E}\perp})]\|^2\right]$$

Now, let us denote: $g(T, x, \mathbf{W}_{\mathcal{E}\perp}) = T^{-1}\Delta o(Tx; \mathbf{W}_{\mathcal{E}\perp})$. Observe that this function g is linear in our deviation parameters $\mathbf{W}_{\mathcal{E}\perp}$. By vectorizing the $h \times 3$ parameter matrix $\mathbf{W}_{\mathcal{E}\perp}$ into a $3h \times 1$ column vector $\boldsymbol{p} = \text{vec}(\mathbf{W}_{\mathcal{E}\perp})$, we can express this linear relationship as a matrix-vector product, for some matrix $\boldsymbol{M}_{T,x}$ with shape $3 \times 3h$: $g(T, x, \mathbf{W}_{\mathcal{E}\perp}) = \boldsymbol{M}_{T,x}\boldsymbol{p}$. Similarly, the rotation-averaged prediction $\bar{g}(x; \mathbf{W}_{\mathcal{E}\perp}) = \mathbb{E}_T[g(T, x, \mathbf{W}_{\mathcal{E}\perp})]$ is also a linear function, so we associate it with the matrix $\bar{\boldsymbol{M}}_x$. The equivariance error term with these linear matrix forms is:

$$\mathbb{E}_{x,T}[\|g(T, x, \mathbf{W}_{\mathcal{E}\perp}) - \bar{g}(x, \mathbf{W}_{\mathcal{E}\perp})\|^2] = \boldsymbol{p}^\intercal \bar{\mathcal{Q}} \boldsymbol{p} \tag{25}$$

where the matrix $\bar{\mathcal{Q}} = \mathbb{E}_{x,T}[(\boldsymbol{M}_{T,x} - \bar{\boldsymbol{M}}_x)^\intercal(\boldsymbol{M}_{T,x} - \bar{\boldsymbol{M}}_x)]$. Finally, observe that $\bar{\mathcal{Q}}$ is positive definite, as the as equivariance error is strictly positive on a non-degenerate dataset whenever $\mathbf{W}_{\mathcal{E}\perp} \neq \mathbf{0}$. By the properties of a positive definite matrix, the quadratic form $\boldsymbol{p}^\intercal \bar{\mathcal{Q}} \boldsymbol{p}$ is lower-bounded by the smallest eigenvalue of $\bar{\mathcal{Q}}$, denoted $\lambda_{min}(\bar{\mathcal{Q}})$, which is positive. It is also upper bounded by the largest eigenvalue $\lambda_{max}(\bar{\mathcal{Q}})$. This establishes the quadratic relationship on the equivariance loss as stated in the theorem.

□

D.5 Proof of Proposition 5

Theorem. *For any neural network whose parameters can be expressed as $\theta = \theta_{\mathcal{E}} + \theta_{\mathcal{E}\perp}$ with $\theta_{\mathcal{E}} \in \mathcal{E}$ and $\theta_{\mathcal{E}\perp} \in \mathcal{E}\perp$, there exist positive constants $0 < \lambda_{min} \leq \lambda_{max}$ such that for a non-degenerate dataset:*

$$\lambda_{min}|\theta_{\mathcal{E}\perp}|_2^2 + \mathcal{O}(|\theta_{\mathcal{E}\perp}|_2^3) \leq \mathcal{L}_{equiv}(\theta) \leq \lambda_{max}|\theta_{\mathcal{E}\perp}|_2^2 + \mathcal{O}(|\theta_{\mathcal{E}\perp}|_2^3) \tag{26}$$

Proof. Applying a Taylor expansion to $\mathcal{L}_{\text{equiv}}(\theta)$ for the neural net f on an input x around equivariant parameters $\theta_{\mathcal{E}}$, we have:

$$f(x;\theta_{\mathcal{E}} + \theta_{\mathcal{E}\perp}) = f(x;\theta_{\mathcal{E}}) + J_{\theta_{\mathcal{E}\perp}} f(x;\theta_{\mathcal{E}})\theta_{\mathcal{E}\perp} + \mathcal{O}(\|\theta_{\mathcal{E}\perp}\|^2) \tag{27}$$

where $J_{\theta_{\mathcal{E}\perp}} f(x;\theta_{\mathcal{E}})$ is the Jacobian of the network output with respect to parameter components $\theta_{\mathcal{E}\perp}$, evaluated at $\theta_{\mathcal{E}}$. As before, the term $f(x;\theta_{\mathcal{E}})$ is equivariant by construction, and thus drops out of the equivariance error term. The term $J_{\theta_{\mathcal{E}\perp}} f(x;\theta_{\mathcal{E}})\theta_{\mathcal{E}\perp}$ is linear in $\theta_{\mathcal{E}\perp}$, which creates a quadratic dependence on $\theta_{\mathcal{E}\perp}$ in the variance term in $\mathcal{L}_{\text{equiv}}$.

The deviation from the twirled mean is the difference between the canonicalized prediction and its average over transformations. Let's expand this difference:

$$(T^{-1} \circ f \circ T)(x;\theta) - \mu(x;\theta) = (T^{-1} \circ f \circ T)(x;\theta) - \mathbb{E}_{T'}[(T'^{-1} \circ f \circ T')(x;\theta)] \tag{28}$$

Substituting the Taylor series and using the equivariance of $f(x;\theta_{\mathcal{E}})$:

$$\begin{aligned}
&= \left(f(x;\theta_{\mathcal{E}}) + [T^{-1}J\theta_{\mathcal{E}\perp} f(T(x);\theta_{\mathcal{E}})]\theta_{\mathcal{E}\perp} + \mathcal{O}(|\theta_{\mathcal{E}\perp}|^2)\right) \\
&\quad - \mathbb{E}_{T'}\left[f(x;\theta_{\mathcal{E}}) + [T'^{-1}J\theta_{\mathcal{E}\perp} f(T'(x);\theta_{\mathcal{E}})]\theta_{\mathcal{E}\perp} + \mathcal{O}(|\theta_{\mathcal{E}\perp}|^2)\right] &(29)\\
&= \left(T^{-1}J_{\theta_{\mathcal{E}\perp}} f(T(x);\theta_{\mathcal{E}}) - \mathbb{E}_{T'}[T'^{-1}J\theta_{\mathcal{E}\perp} f(T'(x);\theta_{\mathcal{E}})]\right)\theta_{\mathcal{E}\perp} + \mathcal{O}(|\theta_{\mathcal{E}\perp}|^2) &(30)
\end{aligned}$$

Let $\Delta J_{x,T} \triangleq T^{-1}J_{\theta_{\mathcal{E}\perp}} f(T(x);\theta_{\mathcal{E}}) - \mathbb{E}_{T'}[T'^{-1}J_{\theta_{\mathcal{E}\perp}} f(T'(x);\theta_{\mathcal{E}})]$. The expression becomes $\Delta J_{x,T} \cdot \theta_{\mathcal{E}\perp} + \mathcal{O}(\|\theta_{\mathcal{E}\perp}\|^2)$.

$$\begin{aligned}
\mathcal{L}_{\text{equiv}}(\theta) &= \frac{1}{D}\mathbb{E}_{x,T}\left[|\Delta J_{x,T} \cdot \theta_{\mathcal{E}\perp} + \mathcal{O}(|\theta_{\mathcal{E}\perp}|^2)|^2\right] &(31)\\
&= \frac{1}{D}\mathbb{E}_{x,T}\left[|\Delta J_{x,T} \cdot \theta_{\mathcal{E}\perp}|^2 + 2(\Delta J_{x,T} \cdot \theta_{\mathcal{E}\perp})^T\mathcal{O}(|\theta_{\mathcal{E}\perp}|^2) + |\mathcal{O}(|\theta_{\mathcal{E}\perp}|^2)|^2\right] &(32)
\end{aligned}$$

The orders of the terms are:

- $\|\Delta J_{x,T} \cdot \theta_{\mathcal{E}\perp}\|^2$ is $\mathcal{O}(\|\theta_{\mathcal{E}\perp}\|^2)$.
- The cross-term is $\mathcal{O}(\|\theta_{\mathcal{E}\perp}\|) \cdot \mathcal{O}(\|\theta_{\mathcal{E}\perp}\|^2) = \mathcal{O}(\|\theta_{\mathcal{E}\perp}\|^3)$.
- The final term is $(\mathcal{O}(\|\theta_{\mathcal{E}\perp}\|^2))^2 = \mathcal{O}(\|\theta_{\mathcal{E}\perp}\|^4)$.

We will study the leading term, which is quadratic in $\theta_{\mathcal{E}\perp}$, and subsume the remainder into $\mathcal{O}(\|\theta_{\mathcal{E}\perp}\|^3)$. As $\Delta J_{x,T}$ is a linear function, we can define a matrix $\bar{\mathcal{Q}}$ that represents the averaged outer product of the Jacobian deviations: $\bar{\mathcal{Q}} \triangleq \frac{1}{D}\mathbb{E}_{x,T}\left[(\Delta J_{x,T})^\intercal (\Delta J_{x,T})\right]$. The equivariance error can now be expressed concisely:

$$\mathcal{L}_{\text{equiv}}(\theta_{\mathcal{E}} + \theta_{\mathcal{E}\perp}) \approx \theta_{\mathcal{E}\perp}^T \bar{\mathcal{Q}}\theta_{\mathcal{E}\perp} \tag{33}$$

The matrix $\bar{\mathcal{Q}}$ is positive definite for a non-degenerate dataset when $\theta_{\mathcal{E}\perp} \neq 0$. Using the Rayleigh-Ritz theorem, this quadratic form is thus bounded by the smallest and largest eigenvalues:

$$\lambda_{\text{min}}\|\theta_{\mathcal{E}\perp}\|_2^2 \leq \theta_{\mathcal{E}\perp}^T \bar{\mathcal{Q}}\theta_{\mathcal{E}\perp} \leq \lambda_{\text{max}}\|\theta_{\mathcal{E}\perp}\|_2^2$$

Reincorporating the remainder term in our Taylor expression, we arrive at:

$$\lambda_{\text{min}}|\theta_{\mathcal{E}\perp}|_2^2 + \mathcal{O}(|\theta_{\mathcal{E}\perp}|_2^3) \leq \mathcal{L}_{\text{equiv}}(\theta) \leq \lambda_{\text{max}}|\theta_{\mathcal{E}\perp}|_2^2 + \mathcal{O}(|\theta_{\mathcal{E}\perp}|_2^3) \tag{34}$$

□

D.6 Proof of Proposition 6

Theorem 8. *Under the same conditions as the Taylor expansion theorem above, the norm of the gradient of the equivariance loss with respect to the non-equivariant parameters is bounded by the deviation itself. Specifically, there exists a constant C such that:*

$$\|\nabla_{\theta_{\mathcal{E}\perp}} \mathcal{L}_{equiv}(\theta)\| \leq C \cdot \|\theta_{\mathcal{E}\perp}\|$$

Proof. From previous theorems, we know $\mathcal{L}_{\text{equiv}}(\theta) \approx \boldsymbol{p}^\intercal \bar{\mathcal{Q}} \boldsymbol{p}$, where $\boldsymbol{p} = \text{vec}(\theta_{\mathcal{E}\perp})$. The gradient of a quadratic form is linear: $\nabla_{\boldsymbol{p}} \mathcal{L}_{\text{equiv}} = 2\bar{\mathcal{Q}}\boldsymbol{p}$. Taking norms, we get $\|\nabla_{\boldsymbol{p}} \mathcal{L}_{\text{equiv}}\| = \|2\bar{\mathcal{Q}}\boldsymbol{p}\| \leq 2\|\bar{\mathcal{Q}}\|\|\boldsymbol{p}\|$. Setting $C = 2\lambda_{max}$ or $2\|\bar{\mathcal{Q}}\|_2$ gives the result. □

E Code Availability, Methods & Experimental Details

Code repository for this project: ¡tbd¿ Our code repositories are minor modifications on the original codebases. We added callbacks to track metrics during training, added configuration files for controlling training, and added helper scripts for computing and plotting some metrics.

E.1 EScAIP

We trained EScAIP 6M on a subset of SPICE with 950k training examples used by (24) for 30 epochs with batch size 64. SPICE is a dataset with of small molecule 3D conformers with energies and forces computed by quantum-mechanical density functional theory (5). We varied model size from 1M, 4M and 6M, varied training set size from 950k, 50k, 5k, and 500 (with batch size 1), and varied the optimizer or learning rate. The model predicts a 3D force vector for each atom based on density functional theory, mapping an input molecule with N atoms to an output in $\mathbb{R}^{3N}$. This task is physically equivariant to the special orthogonal group $SO(3)$ acting on atom coordinates in $\mathbb{R}^3$.

We follow the same training recipe as the original repository, which does not use data augmentation. We suspect that data augmentation is not as important for EScAIP because it operates on rotation-invariant features.

For further details and configuration files, please refer to our code repository.

E.2 Proteína

We trained Proteína at 60M without triangular attention and 400M with triangular attention on the full Protein databank (PDB) dataset with 225k training examples. We also trained models on 1% of the PDB with 2k examples and 0.1% with 200 examples. Flow matching trains a model jointly over t, flow matching time, ranging from $t = 0$ for noise and $t = 1$ for data. We measure metrics at $t = 0, 0.2, 0.4, 0.6, 0.8, 0.9, 0.95$, and 0.99, and use red colors for high t close to the data, and blue-purple colors for low t near noise in Figure 3. The model learns to approximate the velocity field of a probability flow that transforms random noise into structured protein backbones. For a molecule with N alpha carbon atoms, the network maps noised atom coordinates and a time $t \in [0, 1]$ to a velocity vector in $\mathbb{R}^{3N}$. The learning task is made rotationally equivariant through data augmentation, aligning it with $SO(3)$ acting on atom coordinates in $\mathbb{R}^3$.

For further details and configuration files, please refer to our code repository.

E.3 VoxMol

Following (23), we represent each molecule using a 3D voxel grid by placing a continuous Gaussian density at each atom's position. Each atom type is assigned a distinct input channel, producing a 4D tensor of shape $[c \times l \times l \times l]$, where c denotes the number of atom types and l is the edge length of the voxel grid. The voxel values are normalized between 0 and 1.

The denoising task arises from the use of walk-jump sampling for generating molecules (25). This uses a two-step score-based sampling method. The "walk" phase involves running k steps of Langevin Markov chain Monte Carlo on a randomly initialized noisy voxel grid, simulating a stochastic trajectory along a manifold. The "jump" phase applies a denoising autoencoder (DAE) to

clean up the noisy sample using a forward pass of the trained model at step k. The DAE is trained on voxelized molecules corrupted with isotropic Gaussian noise, with a mean squared error (MSE) loss between prediction and ground truth. WJS provides a fast alternative to diffusion models by requiring only a single noise and denoise step (23; 19).

Architecture The VoxMol architecture is based on a 3D U-Net with convolutional layers spanning four resolution scales, and includes self-attention modules at the two coarsest levels (23). During training, data augmentation is performed by applying random rotations and translations to each sample. For further architectural and training details, refer to Pinheiro et al. (23).

Measuring whether latent representations learn to respect equivariance To evaluate whether VoxMol learns equivariant latent features, we analyze cosine similarity between latent embeddings under two scenarios.

First, we examine representations of the *same molecule under rotation*. Let $\mathbf{x}$ be a molecule and R_k a discrete rotation operator (e.g., 90° around an axis). Using the encoder $\phi(\cdot) \in \mathbb{R}^{C\times D\times H\times W}$, with $C = 512$ and spatial dimensions $8 \times 8 \times 8$, we define the spatially pooled latent vector:

$$\bar{\phi}(\mathbf{x}) = \frac{1}{DHW} \sum_{d,h,w} \phi(\mathbf{x})[:, d, h, w]$$

We then compute:

$$\text{sim}_{\text{same}} = \cos\left(\bar{\phi}(R_k(\mathbf{x})),\ R_k(\bar{\phi}(\mathbf{x}))\right)$$

This measures whether encoding a rotated molecule is equivalent to rotating the latent vector of the original input—a key signature of learned equivariance.

Second, to obtain a baseline, we compute cosine similarities between embeddings of *randomly selected different molecules*:

$$\text{sim}_{\text{diff}} = \cos\left(\bar{\phi}(\mathbf{x}_i),\ \bar{\phi}(\mathbf{x}_j)\right), \quad \text{with } \mathbf{x}_i \neq \mathbf{x}_j$$

We compute these metrics across 1000 molecules for various rotation angles along all three axes. Cosine similarities are calculated over the 512-dimensional latent vectors and visualized using violin plots to capture the distributional differences in Figure 9.

Findings. Cosine similarity between rotated versions of the same molecule tends to decrease as rotation angle increases, reflecting imperfect latent equivariance. While same-molecule embeddings remain more similar to each other than to embeddings of different molecules, the overlap between their distributions grows with rotation. This suggests that although the encoder partially preserves geometric structure, the latent space does not fully achieve rotation equivariance, indicating potential for improved regularization or architectural design.

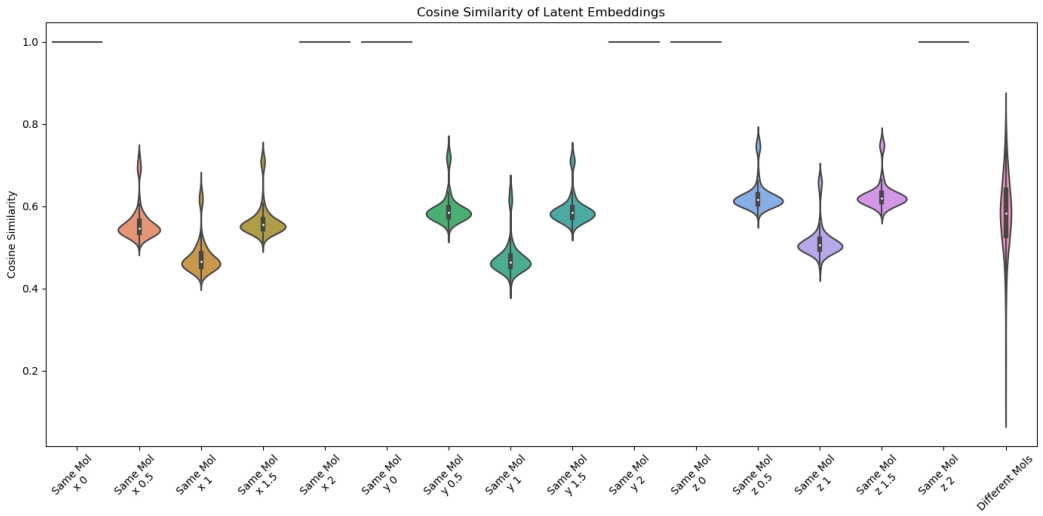

Figure 9: VoxMol: Cosine similarity of molecule latent representations with different rotations. x, y, z indicate rotation axes, and numbers 0, 0.5, 1, 1.5, 2 correspond to 0, 90, 180, 270, 360 degrees of rotation. The last column depicts cosine similarity between different molecules.

E.4 Metrics

To compute equivariance error, twirled prediction, error, percent MSE loss from equivariance error, and gradient norms, 10 rotations per sample were used in EScAIP and Proteína. This number was found to be sufficient to provide a stable signal for metrics which was robust to randomness and resampling. For EScAIP, these metrics were computed on the first four (fixed) validation batches with batch size of 16, for a total of 64 samples. For Proteína, these metrics were computed on the first eight (fixed) validation batches with batch size of 3, for a total of 24 samples. The total MSE loss on these subsets was indicative of the total validation MSE loss, indicating these sample sizes were sufficient to provide a stable and representative signal for these metrics.

To plot the loss landscape, we selected a subset of parameters in each architecture. For EScAIP, we used the final FFN (with a non-linearity) and the final linear head, for a combined total of 33k parameters. For Proteína, we used the final linear head with 1.5k parameters. We computed the Hessian of this parameter subset for the total MSE loss using one fixed training batch with ten rotations. We then performed eigendecomposition of the total MSE loss Hessian to find the eigenvectors for the largest positive eigenvalue, and minimum positive eigenvalue, which formed the two axes for plotting the loss landscape. We selected a step size approximately 2-3x the training step size at that checkpoint, which is estimated by multiplying the training learning rate with the total parameter gradient norm at that checkpoint. We then create a 2D grid of perturbations to the parameter subset, and compute $\mathcal{L}_{\text{mean}}$ and $\mathcal{L}_{\text{equiv}}$ at each point on the grid. Importantly, the axes and the step size are the same for both $\mathcal{L}_{\text{mean}}$ and $\mathcal{L}_{\text{equiv}}$.

To compute the condition numbers, we computed the Hessian of the same parameter subsets for $\mathcal{L}_{\text{mean}}$ and $\mathcal{L}_{\text{equiv}}$ separately, and performed eigendecomposition on them separately. We reported the condition number as the ratio between the largest positive eigenvalue and the minimum positive eigenvalue.

E.5 Loss Landscape Analysis

To better understand the initial dip, we studied loss landscapes for $\mathcal{L}_{\text{mean}}$ and $\mathcal{L}_{\text{equiv}}$ at early checkpoints (500 steps). We computed the Hessian of each loss on a training batch for a subset of 33k parameters including non-linear layers for EScAIP, 1.5k parameters in Proteína's linear head, and 6.9k parameters in a final layer of VoxMol. For EScAIP, we measured condition numbers of 1e9 for $\mathcal{L}_{\text{mean}}$ and 1e6 for $\mathcal{L}_{\text{equiv}}$ (1,000x smaller). For Proteína, we measured 2e10 for $\mathcal{L}_{\text{mean}}$ and 1e8 for $\mathcal{L}_{\text{equiv}}$ (100x smaller). For VoxMol, we measured 5e9 and 6e8 respectively (10x smaller). We calculate condition numbers for $\mathcal{L}_{\text{mean}}$ and $\mathcal{L}_{\text{equiv}}$ using the largest positive and smallest positive

eigenvalues for each loss. For loss landscape plotting, we chose two axes for plotting using the largest positive and smallest positive eigenvector on the total loss, and used the same step size and grid for $\mathcal{L}_{\text{mean}}$ and $\mathcal{L}_{\text{equiv}}$. In both models, we find that $\mathcal{L}_{\text{equiv}}$ has a substantially smoother loss landscape than $\mathcal{L}_{\text{mean}}$ (Fig. 1C).