Appendix

The appendix is organized as follows:

- In Sec. A, we provide additional visualizations following Fig. 1's format.
- In Sec. B, we review additional mathematical background.
- In Sec. C, we provide additional details on the method.
- In Sec. D, we provide clarifications on our design choices and provide examples.
- In Sec. E, we provide the complete proofs of the claims and lemma in the main paper.
- In Sec. F, we provide additional results on the human trajectory experiment.
- In Sec. G, we document the experimental details. Code is available at `https://github.com/ashiq24/soft-equivariance`

A. Additional Visualization following Fig. 1

Following the format of Fig. 1, we provide more visualization on the model weights, and with respect to different rotations in Fig. 3 and Fig. 4.

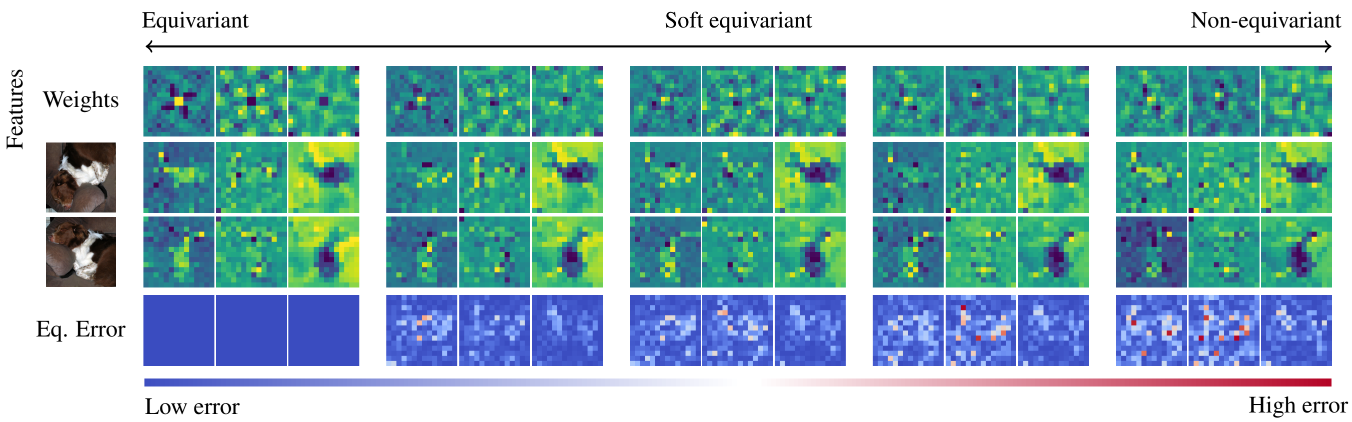


Figure 3. Visualization of the ViT [12] weights with our soft equivariance layer (w.r.t. 90° rotation) under different softness levels, along with the corresponding extracted features and the equivariance errors. Our tunable design allows the layers' weights to transition smoothly from perfectly equivariant to fully non-equivariant behavior in a controlled manner.

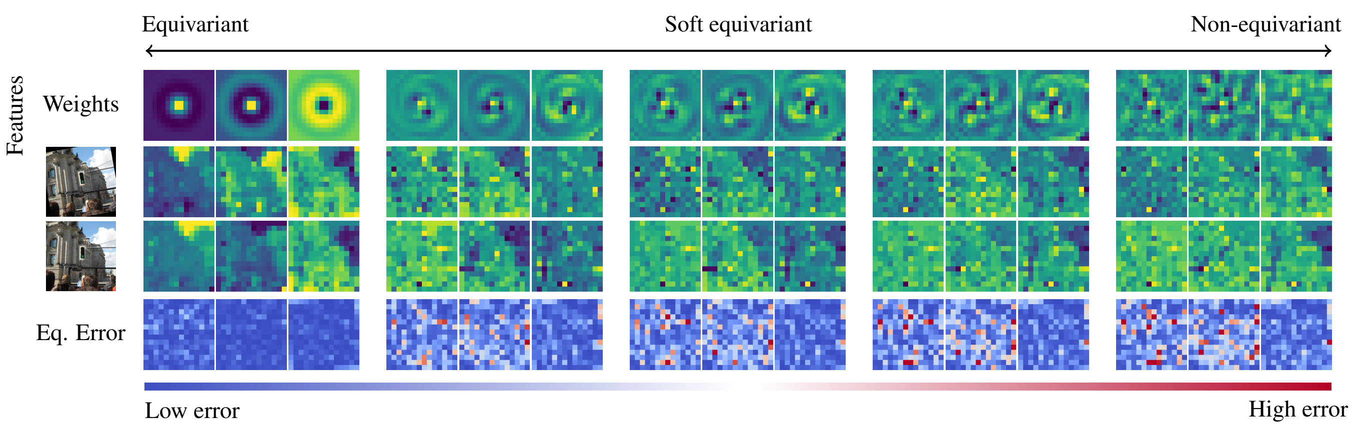


Figure 4. Visualization of the ViT [12] weights with our soft equivariance layer (w.r.t. 10° rotation) under different softness levels, along with the corresponding extracted features and the equivariance errors. Our tunable design allows the layers' weights to transition smoothly from perfectly equivariant to fully non-equivariant behavior in a controlled manner.

B. Mathematical Background

In this section, we review the mathematical preliminaries needed to construct the soft-invariant and soft-equivariant projection operator.

B.1. Groups

Group. A group is a set G together with a binary operation $\cdot : G \times G \to G$ that combines any two elements $a, b \in G$ to form another element denoted $a \cdot b$. And it satisfies the following four requirements:
- *Closure:* For all $a, b \in G$, the result of the operation $a \cdot b$ is also in G.
- *Associativity:* For all $a, b, c \in G$, $(a \cdot b) \cdot c = a \cdot (b \cdot c)$.
- *Identity element:* There exists an element $e \in G$ such that for every element $a \in G$, the equation $e \cdot a = a \cdot e = a$ holds.
- *Inverse element:* For each $a \in G$, there exists an element $b \in G$ such that $a \cdot b = b \cdot a = e$, where e is the identity element.

The binary operation is often referred to as the group product.

Generating Set. A subset $\mathbb{S} \subseteq G$ is called a generating set of the group G if every element of G can be expressed as product of the elements of the $\mathbb{S}$. So for any element $g \in G$, there exists a sequence of elements $s_1, s_2, \ldots, s_m$ from $\mathbb{S}$ such that $g = s_1^{n_1} \cdot s_2^{n_2} \cdot \ldots \cdot s_m^{n_m}$.

Word Metric. Given a generating set $\mathbb{S}$ of G, any element $g \in G$ can be written as a product $g = s_1^{n_1} \cdot s_2^{n_2} \cdots s_m^{n_m}$ with $s_i \in \mathbb{S}$. The *word length* of g with respect to $\mathbb{S}$ is defined as the minimum number of generator applications required to express g:

$$\ell_{\mathbb{S}}(g) \triangleq \min \left\{ \sum_{i=1}^{m} n_i \;\middle|\; g = s_1^{n_1} \cdots s_m^{n_m},\ s_i \in \mathbb{S} \right\}. \tag{30}$$

The *word metric* $d_{\mathbb{S}} : G \times G \to \mathbb{Z}_{\geq 0}$ is then defined as:

$$d_{\mathbb{S}}(g, g') \triangleq \ell_{\mathbb{S}}(g^{-1} g'), \tag{31}$$

which measures the minimum number of generator steps needed to reach g' from g. When $\mathbb{S}$ is clear from context, we write $d(\cdot, \cdot)$ and $\ell(\cdot)$ to denote word metric and word length, respectively.

Note: In general, the decomposition $g = s_1^{n_1} \cdot s_2^{n_2} \cdots s_m^{n_m}$ is not unique. For this, a fixed minimal-length word representation for each element $g \in G$ is chosen, which is called the 'canonical' word representation.

Group Action. The group action or the representation of a group G on space $\mathcal{X}$ is a map $\rho : G \to GL(\mathcal{X})$, where $GL(\mathcal{X})$ is the set of invertible linear transformations on $\mathcal{X}$, such that for any $g, h \in G$ and any $\boldsymbol{x} \in \mathcal{X}$, we have $\rho(e)\boldsymbol{x} = \boldsymbol{x}$ and $\rho(gh)\boldsymbol{x} = \rho(g)(\rho(h)\boldsymbol{x})$, where e is the identity element of the group G.

Discrete Rotation Group (Cyclic group C_n). Let r denote the planar rotation by angle $\frac{2\pi}{n}$. The set

$$C_n \triangleq \{\, r^k \mid k = 0, 1, \ldots, n-1 \,\}$$

With composition as the group product, it is a discrete (finite) group.

The generating set is $\{r\}$ since every element is r^k. The $2D$ linear representation ρ maps r^k to $2D$ rotation matrix $R(\frac{2\pi k}{n})$ where

$$R(\theta) \triangleq \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}.$$

The action on $\boldsymbol{x} \in \mathbb{R}^2$ is $\rho(r^k)\boldsymbol{x} = R\big(\frac{2\pi k}{n}\big)\boldsymbol{x}$.

Lie Group. A Lie group G is a smooth manifold equipped with a group structure such that the group operations—multiplication $(g, h) \mapsto gh$ and inversion $g \mapsto g^{-1}$ are smooth maps.

Lie Algebra. The Lie algebra $\mathfrak{g}$ associated with a Lie group G is the tangent space at the identity element e, denoted as $\mathfrak{g} = T_e G$. It is a vector space equipped with a bilinear operation $[\cdot, \cdot] : \mathfrak{g} \times \mathfrak{g} \to \mathfrak{g}$ called the Lie bracket. For matrix Lie groups (where $G \subseteq \mathrm{GL}(n, \mathbb{R})$), the bracket is defined as the matrix commutator:

$$[A, B] = AB - BA, \quad \forall A, B \in \mathfrak{g}. \tag{32}$$

Lie Algebra Basis. A basis for a Lie algebra $\mathfrak{g}$ of dimension d is a set of linearly independent elements $\{A_1, \ldots, A_d\}$ such that every element $A \in \mathfrak{g}$ can be uniquely represented as a linear combination $A = \sum_{i=1}^{d} c_i A_i$, where c_i are scalar coefficients.

Exponential Map. The exponential map $\exp : \mathfrak{g} \to G$ creates a mapping from the tangent space to the manifold. For matrix Lie groups, it is given by the matrix exponential:

$$\exp(A) = \sum_{k=0}^{\infty} \frac{A^k}{k!}. \tag{33}$$

The map satisfies $\exp(0) = I$ and $\frac{d}{dt}\exp(tA)|_{t=0} = A$. Crucially, for commutative algebras, $\exp(A+B) = \exp(A)\exp(B)$, though this does not hold generally for non-abelian groups like $\mathrm{SO}(3)$.

Group and Algebra Representations. A group representation is a homomorphism $\rho : G \to \mathrm{GL}(V)$ acting on a vector space V. This induces a Lie algebra representation $d\rho : \mathfrak{g} \to \mathfrak{gl}(V)$, derived by differentiating the group action at the identity:

$$d\rho(A) = \frac{d}{dt}\rho(\exp(tA))\bigg|_{t=0}. \tag{34}$$

This map preserves the bracket structure: $d\rho([A, B]) = [d\rho(A), d\rho(B)]$.

Case Studies: Rotations in $\mathbb{R}^2$ and $\mathbb{R}^3$.

We explicitly detail the geometry of 2D and 3D rotations used in the main paper.

The $\mathrm{SO}(2)$ Group. The $\mathrm{SO}(2)$ or the special orthogonal group in 2D consists of planar rotation matrices parameterized by an angle $\theta \in [0, 2\pi)$:

$$R(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}. \tag{35}$$

The corresponding Lie algebra, $\mathfrak{so}(2)$, consists of 2×2 skew-symmetric matrices. It is 1-dimensional, spanned by the generator A:

$$\mathfrak{so}(2) = \{tA \mid t \in \mathbb{R}\}, \quad \text{where } A = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}. \tag{36}$$

Since $\mathrm{SO}(2)$ is abelian, the exponential map is straightforward: $\exp(tA) = R(t)$. The standard representation acting on $\mathbf{x} \in \mathbb{R}^2$ yields the generator action:

$$d\rho(A)\mathbf{x} = A\mathbf{x} = \begin{pmatrix} -x_2 \\ x_1 \end{pmatrix}. \tag{37}$$

The $\mathrm{SO}(3)$ Group . The group of 3D rotations is non-abelian. Its Lie algebra, $\mathfrak{so}(3)$, consists of all 3×3 skew-symmetric matrices. A general element $A \in \mathfrak{so}(3)$ has the form:

$$A = \begin{pmatrix} 0 & -c & b \\ c & 0 & -a \\ -b & a & 0 \end{pmatrix}, \quad a, b, c \in \mathbb{R}. \tag{38}$$

Any such matrix can be written as $A = \theta K$, where θ is the rotation angle and K is a skew-symmetric matrix representing the unit rotation axis ($K^T = -K$ and $\|K\|_{\mathrm{F}} = \sqrt{2}$). The exponential map is given by Rodrigues' rotation formula:

$$\exp(\theta K) = I + \sin\theta K + (1 - \cos\theta)K^2. \tag{39}$$

For example, the generator for a rotation about the z-axis corresponds to $a = b = 0, c = 1$:

$$K_z = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \exp(\theta K_z) = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{40}$$

The Lie algebra action $d\rho(A)$ describes the infinitesimal velocity of a point $\mathbf{x}$ being rotated. It is computed via matrix-vector multiplication:

$$d\rho(A)\mathbf{x} = A\mathbf{x} = \begin{pmatrix} 0 & -c & b \\ c & 0 & -a \\ -b & a & 0 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} -cx_2 + bx_3 \\ cx_1 - ax_3 \\ -bx_1 + ax_2 \end{pmatrix}. \tag{41}$$

This linear action is geometrically equivalent to the cross product $\mathbf{v} \times \mathbf{x}$, where the vector $\mathbf{v} = (a, b, c)^\top$ corresponds to the axis of rotation scaled by the angular velocity.

Lie algebra basis. Since $\mathfrak{so}(3)$ is a 3-dimensional vector space, any element A can be uniquely expressed as a linear combination of three basis matrices, $A = aA_x + bA_y + cA_z$. These basis elements correspond to infinitesimal rotations about the standard axes:

$$A_x = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}, \quad A_y = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix}, \quad A_z = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}. \tag{42}$$

Real Schur Decomposition. For any real square matrix $\boldsymbol{M} \in \mathbb{R}^{n\times n}$, there exists an orthogonal matrix $\boldsymbol{U}$ such that $\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{U}^\top = \boldsymbol{M}$, where $\boldsymbol{\Sigma}$ is upper quasi-triangular. When $\boldsymbol{M}$ is normal, *i.e.*, $\boldsymbol{M}\boldsymbol{M}^\top = \boldsymbol{M}^\top\boldsymbol{M}$, $\boldsymbol{\Sigma}$ is block diagonal with 1×1 and 2×2 blocks where the 2×2 blocks correspond to pairs of complex conjugate eigenvalues. And the canonical form for a these 2×2 block is (see Theorem 2.5.8 by Horn and Johnson [24]):

$$\begin{pmatrix} a & b \\ -b & a \end{pmatrix},$$

where the associated eigenvalues are $a \pm ib$. The spectral norm of these 2×2 matrices models the scaling factor of the associated basis vectors in $\boldsymbol{U}$.

C. Method Details

In this section, we provide additional explanations and technical details of our proposed technique.

C.1. Multi-generator Equivariance

In practice, many Lie groups of interest are generated by multiple generators. For example, the special orthogonal group $SO(3)$ is generated by three generators corresponding to rotations about the x, y, and z axes. To design a soft equivariant layer for a group G with multiple Lie algebra generators $\{A_i\}_{i=1}^k$, we use the right singular vectors of the combined constraint (following Eq. (13)) as

$$\bar{\boldsymbol{L}} = \begin{bmatrix} \boldsymbol{L}_i \\ \vdots \\ \boldsymbol{L}_k \end{bmatrix} \in \mathbb{R}^{k\cdot d\cdot d'\times d\cdot d'}, \tag{43}$$

where, $\boldsymbol{L}_i = (d\rho_{\mathcal{X}}(A_i))^\top \otimes \mathbf{I}_{d'} - \mathbf{I}_d \otimes d\rho_{\mathcal{Y}}(A_i))$. The right singular vector of $\bar{\boldsymbol{L}}$ corresponding is then used to design $\boldsymbol{B}_{\text{eq}}$ following Eq. (15).

C.2. Details on Schur Equivariance prediction

We present two examples showing the implications of Lemma 1 and the constraints on the weights for exact equivariance.

Example: 4D to 3D Equivariant Map. Consider a map from a 4D input space $\mathcal{X}$ to a 3D output space $\mathcal{Y}$. Let the Schur decomposition of their respective Lie algebra representations be:

$$d\rho_{\mathcal{X}} = \boldsymbol{U}_{\mathcal{X}}\boldsymbol{\Sigma}_{\mathcal{X}}\boldsymbol{U}_{\mathcal{X}}^\top, \quad d\rho_{\mathcal{Y}} = \boldsymbol{U}_{\mathcal{Y}}\boldsymbol{\Sigma}_{\mathcal{Y}}\boldsymbol{U}_{\mathcal{Y}}^\top \tag{44}$$

The input and output Schur forms $\boldsymbol{\Sigma}_{\mathcal{X}}$ and $\boldsymbol{\Sigma}_{\mathcal{Y}}$ are block diagonal matrices with the following blocks:

$$\boldsymbol{S}_1 = \begin{pmatrix} a & b \\ -b & a \end{pmatrix},\ \boldsymbol{S}_2 = \begin{pmatrix} c & d \\ -d & c \end{pmatrix} \tag{45}$$

$$\boldsymbol{\Sigma}_{\mathcal{X}} = \operatorname{diag}(\boldsymbol{S}_1, \boldsymbol{S}_2) \tag{46}$$

$$\boldsymbol{T}_1 = \begin{pmatrix} c & d \\ -d & c \end{pmatrix},\ \boldsymbol{T}_2 = \begin{pmatrix} \lambda_1 \end{pmatrix} \tag{47}$$

$$\boldsymbol{\Sigma}_{\mathcal{Y}} = \operatorname{diag}(\boldsymbol{T}_1, \boldsymbol{T}_2) \tag{48}$$

The transformed weight (in Schur basis) matrix Θ' is a 3×4 matrix. To be equivariant, non-zero blocks Θ'_{IJ} can only exist where the representation blocks share eigenvalues, *i.e.*, $\boldsymbol{T}_I \simeq \boldsymbol{S}_J$. All other blocks must be zero [24] and the non-zero block must take the form of Eq. (22).

The final sparse structure for Θ' is

$$\Theta' = \left(\begin{array}{c|c} \mathbf{0}_{2\times 2} & \begin{pmatrix} \alpha & \beta \\ -\beta & \alpha \end{pmatrix} \\ \hline \mathbf{0}_{1\times 2} & \mathbf{0}_{1\times 2} \end{array}\right). \tag{49}$$

This equivariant design reduces the number of learnable parameters from $3\times 4 = 12$ in a standard dense layer to just 2, while guaranteeing the desired symmetry.

Example: 4D to 4D Equivariant Map. Consider a map from a 4D input space $\mathcal{X}$ to a 4D output space $\mathcal{Y}$. Let the Schur decomposition of their respective Lie algebra representations be:

$$d\rho_{\mathcal{X}} = \boldsymbol{U}_{\mathcal{X}}\boldsymbol{\Sigma}_{\mathcal{X}}\boldsymbol{U}_{\mathcal{X}}^\top, \quad d\rho_{\mathcal{Y}} = \boldsymbol{U}_{\mathcal{Y}}\boldsymbol{\Sigma}_{\mathcal{Y}}\boldsymbol{U}_{\mathcal{Y}}^\top \tag{50}$$

Let the block diagonal Schur forms $\mathbf{\Sigma}_{\mathcal{X}}$ and $\mathbf{\Sigma}_{\mathcal{Y}}$ be written as:

$$\mathbf{\Sigma}_{\mathcal{X}} = \mathrm{diag}(\boldsymbol{S}_1, \boldsymbol{S}_2), \quad \mathbf{\Sigma}_{\mathcal{Y}} = \mathrm{diag}(\boldsymbol{T}_1, \boldsymbol{T}_2) \tag{51}$$

We define the specific structure of these blocks using scalars a, b and c, d as follows:

$$\boldsymbol{S}_1 = \begin{pmatrix} a & b \\ -b & a \end{pmatrix}, \ \boldsymbol{S}_2 = \begin{pmatrix} c & d \\ -d & c \end{pmatrix} \tag{52}$$

$$\boldsymbol{T}_1 = \begin{pmatrix} a & b \\ -b & a \end{pmatrix}, \ \boldsymbol{T}_2 = \begin{pmatrix} c & d \\ -d & c \end{pmatrix} \tag{53}$$

The transformed weight matrix Θ' is a 4×4 matrix. According to Lemma 1 $\mathbf{W}'_{IJ}$ relating input block $\boldsymbol{S}_J$ to output block $\boldsymbol{T}_I$ can be non-zero only if the representations are equivalent ($\boldsymbol{T}_I \simeq \boldsymbol{S}_J$).

Here, we observe that $\boldsymbol{T}_1 \simeq \boldsymbol{S}_1$ and $\boldsymbol{T}_2 \simeq \boldsymbol{S}_2$, while cross-pairings (e.g., $\boldsymbol{T}_1$ vs $\boldsymbol{S}_2$) possess distinct eigenvalues. Consequently, the weight matrix allows for learnable parameters (following Eq. (22)) on the diagonal blocks but enforces zeros elsewhere:

$$\Theta' = \left(\begin{array}{c|c} \begin{pmatrix} \alpha_1 & \beta_1 \\ -\beta_1 & \alpha_1 \end{pmatrix} & \mathbf{0}_{2\times 2} \\ \hline \mathbf{0}_{2\times 2} & \begin{pmatrix} \alpha_2 & \beta_2 \\ -\beta_2 & \alpha_2 \end{pmatrix} \end{array} \right) \tag{54}$$

This constrains the layer to take a block-diagonal structure in Schur basis, reducing the number of learnable parameters from 16 to 4. We illustrate these examples in Fig. 5.

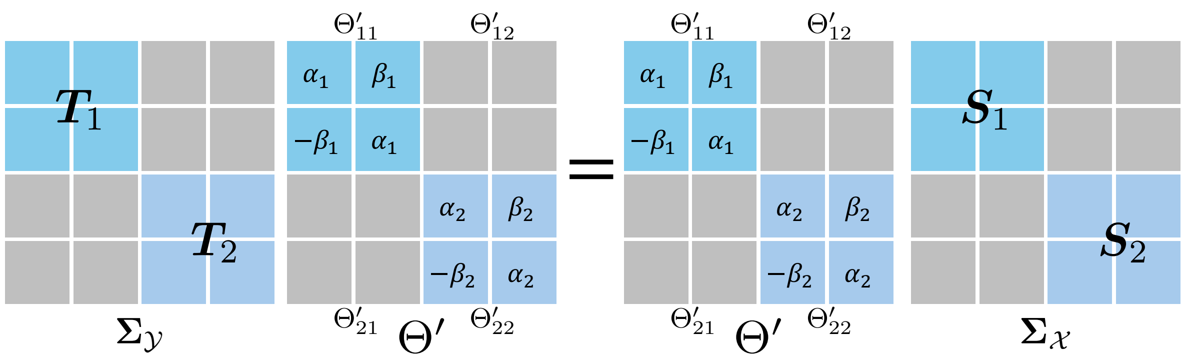


Figure 5. Visualization of condition of strict equivariance condition on the weight Θ' which is described in Lemma 1.

Schur equivariance projection for multiple generators. To design the Schur projection operator for a Lie algebra with multiple generators, we compose the projection operator for each generator, *i.e.*, $\boldsymbol{B}_{\texttt{Schur}} = \boldsymbol{B}^1_{\texttt{Schur}} \circ \boldsymbol{B}^2_{\texttt{Schur}} \circ \cdots \circ \boldsymbol{B}^{n_G}_{\texttt{Schur}}$, where $\boldsymbol{B}^i_{\texttt{Schur}}$ is the projection operator design from $d\rho_{\mathcal{X}}(A_i)$ and $d\rho_{\mathcal{Y}}(A_i)$. For commuting generators, this yields the exact projection operator. However, for non-commutative generators, this is an approximation of the true projection operator.

C.3. From hard thresholding to smooth cut-off

Instead of applying a sharp cut-off (threshold) to the singular values as in Eq. (9), an alternative design is to use a smooth transition, *i.e.*,

$$\boldsymbol{B}_{\texttt{inv}} \triangleq \sum_i \gamma_i \boldsymbol{u}_i \boldsymbol{u}_i^\top, \tag{55}$$

where $\gamma_i \triangleq 1$ if $\sigma_i < b$, else $\gamma_i \triangleq \exp(-\sigma_i^2/s^2)$. Here, $s \in \mathbb{R}^+$ is a hyperparameter controlling the decay rate.

Remarks. For the special case of shift invariance, the smooth cut-off in Eq. (55) is analogous to the concept of "transition band" in filter design [38, 65]. A sharp cut-off in the Fourier domain, *i.e.*, an ideal filter, produces ripples in the time domain, which is undesirable in practice.

Soft Schur equivariance projection. Instead of following a strict cut-off when designing the projection operator, we follow a smooth cut-off defined as:

$$\boldsymbol{W}'_{lk} = \begin{cases} \gamma_{lk}\Theta'_{lk} & \text{if } \boldsymbol{T}_l \not\simeq \boldsymbol{S}_k, \lambda_{\boldsymbol{S}_k} + \lambda_{\boldsymbol{T}_l} > b, \\ \mathtt{Sym}(\Theta'_{lk}) + \gamma_{lk}(\Theta'_{lk} - \mathtt{Sym}(\Theta'_{lk})) & \text{if } \boldsymbol{T}_l \simeq \boldsymbol{S}_k, \lambda_{\boldsymbol{S}_k} + \lambda_{\boldsymbol{T}_l} > b, \\ \Theta'_{lk} & \text{otherwise} \end{cases} \tag{56}$$

Where $\gamma_{lk} = \exp(-\frac{(\lambda_{\boldsymbol{S}_k} + \lambda_{\boldsymbol{T}_l})}{s^2})$ is a decay factor based on the eigenvalues of the corresponding representation blocks, with hyperparameters s.

D. Additional Clarifications

D.1. Properties of η-Soft Equivariance Metric

The relative equivariance error in Eq. (6) provides a metric that is unaffected by local rescaling of the function and the input. For linear layers, it is a scale-invariant measure: rescaling the weight ($\boldsymbol{W} \to \alpha\boldsymbol{W}$) or the input ($\boldsymbol{x} \to \beta\boldsymbol{x}$) does not change the relative equivariance error as

$$\frac{\|\alpha\boldsymbol{W}\rho_{\mathcal{X}}(g)\beta\boldsymbol{x} - \rho_{\mathcal{Y}}(g)\alpha\boldsymbol{W}\beta\boldsymbol{x}\|}{\|\alpha\boldsymbol{W}\|\|\beta\boldsymbol{x}\|} = \frac{\|\boldsymbol{W}\rho_{\mathcal{X}}(g)\boldsymbol{x} - \rho_{\mathcal{Y}}(g)\boldsymbol{W}\boldsymbol{x}\|}{\|\boldsymbol{W}\|\|\boldsymbol{x}\|}., \tag{57}$$

This property ensures that the metric captures the intrinsic equivariance error without being influenced by the scale of the weights or inputs.

Furthermore, we empirically validate the stability of this metric by analyzing a pre-trained ResNet-18 model. As illustrated in Fig. 6, the norm of the Jacobian of the logits with respect to the input does not vanish, and the relative equivariance error neither explodes nor collapses. This confirms the robustness and reliability of our proposed metric.

D.2. Example

Here we will show a worked-out example for constructing a projection operator for weights $\boldsymbol{W} \in \mathbb{R}^{3\times 3}$ working on 3D vectors. We aim to achieve approximate equivariance with respect to rotation about the z-axis, corresponding to the following Lie algebra generator:

$$A_z = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}. \tag{58}$$

Here we directly used the matrix form of A_z, *i.e.*, $d\rho(A_z) = A_z$

D.2.1. Using SVD

Following 13, we form the Kronecker product

$$\boldsymbol{L} = A_z^\top \otimes I_3 - I_3 \otimes A_z \;\in\; \mathbb{R}^{9\times 9}. \tag{59}$$

When written explicitly, we obtain:

$$\boldsymbol{L} = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{60}$$

The singular values of $\boldsymbol{L}$ are $\{0^{(\times 3)},\ 1^{(\times 4)},\ 2^{(\times 2)}\}$.

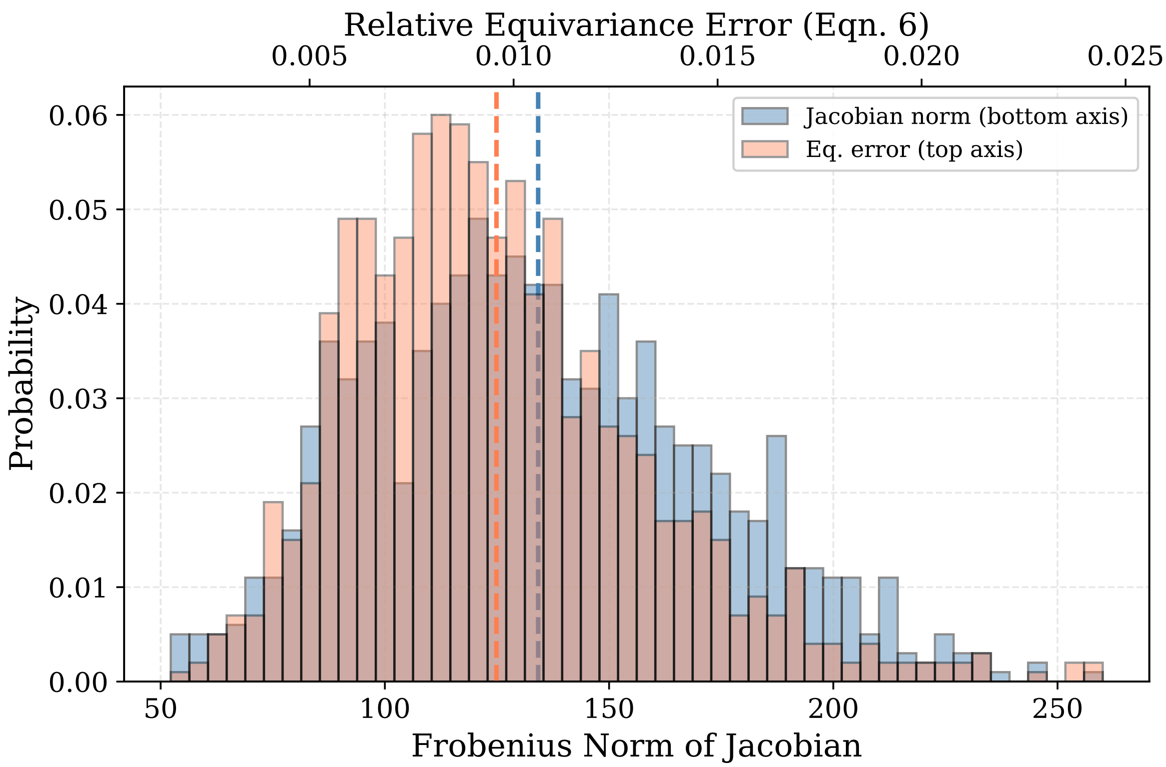


Figure 6. Distribution of the norm of the Jacobian and the relative equivariance metric (Eq. (6)) from a pre-trained ResNet-18 model on ImageNet. The relative equivariance error does not collapse or explode.

The 3-dimensional null space corresponds to singular values of 0, and yields the following 3-parameter family of exactly equivariant weights, i.e., exact equivariance:

$$W_{\text{eq}} = \begin{pmatrix} \alpha & \beta & 0 \\ -\beta & \alpha & 0 \\ 0 & 0 & \gamma \end{pmatrix}, \quad \alpha, \beta, \gamma \in \mathbb{R}. \tag{61}$$

With SVD $\boldsymbol{L} = \boldsymbol{U}^L \Sigma^L (\boldsymbol{V}^L)^\top$, the projection $\boldsymbol{B}_{\mathbf{eq}} = \sum_{i:\, \sigma_i < b} v_i^L\, (v_i^L)^\top$ keeps right singular vectors below cutoff b. For $b = 1.5$, the 7 vectors with $\sigma \in \{0, 1\}$ are retained, yielding a 7-dimensional projected subspace. Reshaping each of the 7 right singular vectors $v_i^L \in \mathbb{R}^9$ into 3×3 matrices $\boldsymbol{V}_i = \mathrm{reshape}(v_i^L)$, the projected weight can be written as:

$$\boldsymbol{W} = \sum_{i=1}^{7} a_i \boldsymbol{V}_i = a_1 \boldsymbol{V}_1 + a_2 \boldsymbol{V}_2 + a_3 \boldsymbol{V}_3 + a_4 \boldsymbol{V}_4 + a_5 \boldsymbol{V}_5 + a_6 \boldsymbol{V}_6 + a_7 \boldsymbol{V}_7, \tag{62}$$

where the 7 basis matrices are (3 from $\sigma = 0$, 4 from $\sigma = 1$):

$$\boldsymbol{V}_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \boldsymbol{V}_2 = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \boldsymbol{V}_3 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

$$\boldsymbol{V}_4 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \boldsymbol{V}_5 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}, \quad \boldsymbol{V}_6 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \quad \boldsymbol{V}_7 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}.$$

Here $\boldsymbol{V}_1, \boldsymbol{V}_2, \boldsymbol{V}_3$ span the exactly equivariant subspace ($\sigma = 0$), while $\boldsymbol{V}_4, \boldsymbol{V}_5, \boldsymbol{V}_6, \boldsymbol{V}_7$ correspond to $\sigma = 1$ directions that mildly break equivariance by coupling the xy-plane to the z-axis.

D.2.2. Schur decomposition approach

For this example A_z is already in real Schur form, $\boldsymbol{U}_{\mathcal{X}} = \boldsymbol{U}_{\mathcal{Y}} = I_3$. Furthermore, $\boldsymbol{\Sigma}_{\mathcal{X}} = \mathrm{diag}(\{\boldsymbol{S}_k\}_{k=1}^2) = \boldsymbol{\Sigma}_{\mathcal{Y}} = \mathrm{diag}(\{\boldsymbol{T}_l\}_{l=1}^2) = A_z$ with A_z having the following block diagonal structure:

$$A_z = \mathrm{diag}\left(\underbrace{\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}}_{S_1,\ \lambda_{S_1}=1}, \underbrace{(0)}_{\boldsymbol{S}_2,\ \lambda_{S_2}=0} \right). \tag{63}$$

The weight in Schur basis $\Theta' = \boldsymbol{U}_{\mathcal{Y}}^{\top} \Theta \boldsymbol{U}_{\mathcal{X}} = \Theta$ is partitioned into blocks. Following Lemma 1, each block Θ'_{lk} is governed by the Sylvester equation $T_l\, \Theta'_{lk} = \Theta'_{lk}\, S_k$ which can be summarized as follows:

Block	Size	$\lambda_{\boldsymbol{T}_l} + \lambda_{\boldsymbol{S}_k}$	$\boldsymbol{T}_l \simeq \boldsymbol{S}_k$?	Equivariant solution
Θ'_{11}	2×2	2	Yes	$\begin{pmatrix} \alpha & \beta \\ -\beta & \alpha \end{pmatrix}$
Θ'_{12}	2×1	1	No	$\mathbf{0}$
Θ'_{21}	1×2	1	No	$\mathbf{0}$
Θ'_{22}	1×1	0	Yes	γ

Schur projection with $b = 1.5$: Following Eq. (20), blocks with $\lambda_{\boldsymbol{T}_l} + \lambda_{\boldsymbol{S}_k} > 1.5$ are projected; blocks below this threshold are unconstrained. At $b = 1.5$:

- Θ'_{11} ($\lambda = 2 \geq 1.5$, $T_1 \simeq S_1$): symmetrized to $\begin{pmatrix} \alpha & \beta \\ -\beta & \alpha \end{pmatrix}$ form (2 degree of freedom) after projection.
- Θ'_{12}, Θ'_{21} ($\lambda = 1 < 1.5$): unconstrained (4 degree of freedom) ,*i.e.*, remain unchanged after projection.
- Θ'_{22} ($\lambda = 0 < 1.5$): unconstrained (1 degree of freedom).

Numerical Example: Schur projection in action Let the weights before projection be

$$\Theta = \begin{pmatrix} 2 & 3 & 1 \\ -1 & 4 & 2 \\ 3 & -1 & 5 \end{pmatrix}$$

.

Applying the Schur projection with cutoff $b = 1.5$:

- $\Theta'_{11} = \begin{pmatrix} 2 & 3 \\ -1 & 4 \end{pmatrix} \xrightarrow{\text{Sym}} \begin{pmatrix} 3 & 2 \\ -2 & 3 \end{pmatrix}$
- $\Theta'_{12} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$, $\Theta'_{21} = \begin{pmatrix} 3 & -1 \end{pmatrix}$: unchanged.
- $\Theta'_{22} = (5)$: unchanged.

The final projected weight matrix is

$$\boldsymbol{W} = \begin{pmatrix} 3 & 2 & 1 \\ -2 & 3 & 2 \\ 3 & -1 & 5 \end{pmatrix}.$$

Remarks. We note that both the SVD and the Schur decomposition approach give rise to the same projection operator. However, computing the projection operator via Schur decomposition is much more efficient.

E. Complete Proofs of the Claims and Lemmas

Reformulation of equivariance constraint. To study deviation of a function F from perfect equivariance at an input $\boldsymbol{x}$, we define the equivariance residual/error $R(g)$ under the action of group element $g \in G$:

$$R(g) = F(\rho_{\mathcal{X}}(g)\boldsymbol{x}) - \rho_{\mathcal{Y}}(g)F(\boldsymbol{x}). \tag{64}$$

For perfect equivariance, $\|R(g)\| = 0$ for all $g \in G$ and for soft equivariance the $\|R(g)\|$ is bounded for all $g \in G$. Applying the first order Taylor approximation from Eq. (3) around the identity element $e \in G$, we can rewrite the error as:

$$R(g) = R(e) + \sum_i t_i \left.\frac{\partial R(\exp(\sum_i t_i A_i))}{\partial t_i}\right|_{t=0} + O(\|A\|_{\mathfrak{g}}^2) \tag{65}$$

$$= \sum_i t_i \left.\frac{\partial R(\exp(\sum_i t_i A_i))}{\partial t_i}\right|_{t=0} + O(\|A\|_{\mathfrak{g}}^2), \qquad (\text{as } R(e) = 0) \tag{66}$$

where $g = \exp(A)$ with $A = \sum_i t_i A_i$ and $\|A\|_{\mathfrak{g}} \leq r_G$, where r_G is the injective radius (or injectivity radius) of G. Hence, the Taylor remainder is bounded by

$$O(\|A\|_{\mathfrak{g}}^2) \leq O(r_G^2) \leq \epsilon_G,$$

for some group-dependent constant ϵ_G. In the following proofs, we first bound the linear term and then add the residual term ϵ_G in the final estimate.

E.1. Proof of Claim 1

Claim 1. *For any compact and connected Lie group G with injective radius r_G and n_G number of generators, the function $F_{\mathtt{FC}}(\boldsymbol{x}, \boldsymbol{B}_{\mathtt{inv}}\theta)$ is η_b-soft invariant, i.e.,*

$$\frac{\|(\boldsymbol{B}_{\mathtt{inv}}\theta)^\top \boldsymbol{x} - (\boldsymbol{B}_{\mathtt{inv}}\theta)^\top \rho_{\mathcal{X}}(g)\boldsymbol{x}\|}{\|\boldsymbol{J}_{F_{\mathtt{FC}}(\cdot;\boldsymbol{w})}(\boldsymbol{x})\|_{\mathtt{F}}\|\boldsymbol{x}\|} \leq \eta_b, \forall g \in G \tag{10}$$

where $\eta_b = b\sqrt{n_G} r_G + \varepsilon_G$, $b \in \mathbb{R}^+$ is the cut-off value for the projection operator , and ε_G is the residual from the first-order Taylor approximation.

Proof. We apply the Taylor expansion of the equivariance error (Eq. (65)) on the function $F_{\mathtt{FC}} = \boldsymbol{w}^\top \boldsymbol{x}$ with $\boldsymbol{w} = \boldsymbol{B}_{\mathtt{inv}}\theta$. Following the convention established in Eq. (65), we work with the first-order terms and absorb the Taylor residual error ϵ_G into the final bound and obtain:

$$R(g) = \sum_i t_i \left.\frac{\partial\left(\boldsymbol{w}^\top \rho_{\mathcal{X}}(\exp(\sum_i t_i A_i))\boldsymbol{x} - \boldsymbol{w}^\top \boldsymbol{x}\right)}{\partial t_i}\right|_{t=0} \tag{67}$$

$$= \sum_i t_i \boldsymbol{w}^\top \left.\frac{\partial \rho_{\mathcal{X}}(\exp(\sum_i t_i A_i))\boldsymbol{x}}{\partial t_i}\right|_{t=0} \tag{68}$$

$$= \sum_i t_i \boldsymbol{w}^\top d\rho_{\mathcal{X}}(A_i)\boldsymbol{x} \quad (\text{using Lie algebra representation}). \tag{69}$$

For $F_{\mathtt{FC}}$ the exact invariance condition becomes:

$$R(g) = 0 \implies \sum_i t_i \boldsymbol{w}^\top d\rho_{\mathcal{X}}(A_i)\boldsymbol{x} = 0 \quad \forall t_i, \boldsymbol{x}, g. \tag{70}$$

This implies that if $\boldsymbol{w}$ is in the left null space $d\rho_{\mathcal{X}}(A_i)$ for all i, then the exact invariance condition is satisfied. We now derive the bound on the equivariance error when $\boldsymbol{w}$ is not in the null space of $d\rho_{\mathcal{X}}(A_i)$.

We denote the singular value decomposition (SVD) of each Lie algebra representation $d\rho_{\mathcal{X}}(A_i)$ as:

$$d\rho_{\mathcal{X}}(A_i) = \boldsymbol{U}^i \Sigma^i {\boldsymbol{V}^i}^\top, \tag{71}$$

where U^i and V^i are orthonormal matrices and Σ^i is a diagonal matrix with singular values $0 \leq \sigma_1^i \leq \sigma_2^i \leq \ldots$ arranged in increasing order.

We can express the invariance error norm, $\|R(g)\|$, as:

$$\|R(g)\| = \left\| \sum_i t_i \boldsymbol{w}^\top d\rho_{\mathcal{X}}(A_i)\boldsymbol{x} \right\| \tag{72}$$

$$\leq \sum_i |t_i||\boldsymbol{w}^\top d\rho_{\mathcal{X}}(A_i)\boldsymbol{x}| \tag{73}$$

$$= \sum_i |t_i||\boldsymbol{w}^\top U^i \Sigma^i {V^i}^\top \boldsymbol{x}| \tag{74}$$

$$= \sum_i |t_i||({\boldsymbol{U}^i}^\top \boldsymbol{w})^\top \Sigma^i ({\boldsymbol{V}^i}^\top \boldsymbol{x})| \tag{75}$$

$$= \sum_i |t_i| \left| {W^i}^\top \Sigma^i X^i \right| \tag{76}$$

where $W^i = {\boldsymbol{U}^i}^\top \boldsymbol{w}$ and $X^i = {\boldsymbol{V}^i}^\top \boldsymbol{x}$ are the coordinates of $\boldsymbol{w}$ and $\boldsymbol{x}$ in the left and right singular vector bases, respectively.
As b is the cut-off value of the projection of operator $\boldsymbol{B}_{\texttt{inv}}$, $W^i_j = 0$ for all $\sigma^i_j > b$ (true for all generators by definition). Therefore, we can write:

$$\|R(g)\| \leq \sum_i |t_i| \sum_{j:\sigma^i_j < b} \sigma^i_j |W^i_j||X^i_j| \quad \text{(as } \Sigma^i \text{ is a diagonal matrix)} \tag{77}$$

$$\leq \sum_i b|t_i| \sum_{j:\sigma^i_j < b} |W^i_j||X^i_j| \tag{78}$$

$$\leq \sum_i b|t_i|\|W^i\|\|X^i\| \quad \text{(by Cauchy-Schwarz)} \tag{79}$$

$$\leq \sum_i b|t_i|\|\boldsymbol{w}\|\|\boldsymbol{x}\| \quad \text{(as } U^i \text{ and } V^i \text{ are orthonormal } \|W^i\| = \|\boldsymbol{w}\|) \tag{80}$$

$$= b\|\boldsymbol{w}\|\|\boldsymbol{x}\| \sum_i |t_i| \tag{81}$$

For compact and connected Lie groups $\sqrt{\sum_i t_i^2} \leq r_G$ (assuming an orthonormal basis for the Lie algebra), where r_G is the injective radius of the group G [30]. Applying Cauchy-Schwarz inequality on $\sum_i |t_i|$ gives:

$$\sum_i |t_i| \leq \sqrt{n_G} \sqrt{\sum_i t_i^2} \leq \sqrt{n_G} r_G, \tag{82}$$

where n_G is the number of Lie algebra basis elements of the group G.

For the linear function $F_{\texttt{FC}}$, the norm of the Jacobian at $\boldsymbol{x}$ is $\|\mathbf{J}_{F_{\texttt{FC}}(;\boldsymbol{w})}(\boldsymbol{x})\|_{\texttt{F}} = \|\boldsymbol{w}\|$. Lastly, substituting the definition of $R(g)$ and $\boldsymbol{w}$, and reintroducing the Taylor residual error ϵ_G, we obtain the final bound on the equivariance error:

$$\|(\mathbf{B}\theta)^\top \mathbf{x} - (\mathbf{B}\theta)^\top \rho_{\mathcal{X}}(g)\mathbf{x}\| \leq b\|\boldsymbol{w}\|\|\mathbf{x}\|\sqrt{n_G} r_G + \epsilon_G \implies \frac{\|(\mathbf{B}\theta)^\top \mathbf{x} - (\mathbf{B}\theta)^\top \rho_{\mathcal{X}}(g)\mathbf{x}\|}{\|\mathbf{J}_{F_{\texttt{FC}}(;\boldsymbol{w})}(\mathbf{x})\|_{\texttt{F}}\|\mathbf{x}\|} \leq b\sqrt{n_G} r_G + \varepsilon_G. \tag{83}$$

Here, ε_G is the Taylor residual normalized by $\|\boldsymbol{w}\|\|\mathbf{x}\|$. □

E.2. Proof for Multiple Generators.

We now prove that the Claim 1 also holds for the projection operator $\boldsymbol{B}_{\texttt{inv}}$ designed using the combined SVD of multiple Lie algebra generators.

Proof. We denote

$$\boldsymbol{A} \triangleq [d\rho_{\mathcal{X}}(A_1) \mid d\rho_{\mathcal{X}}(A_2) \mid \cdots \mid d\rho_{\mathcal{X}}(A_{n_G})], \tag{84}$$

where, $A_i, \dots, A_{n_G}$ are the basis elements of the Lie algebra. We also denote the SVD of $\boldsymbol{A}$ as $\boldsymbol{A} = \boldsymbol{U}\Sigma\boldsymbol{V}^\top$.

We now define an extended input $\boldsymbol{z}$ as

$$\boldsymbol{z} \triangleq \begin{bmatrix} t_1\boldsymbol{x} \\ t_2\boldsymbol{x} \\ \vdots \\ t_{n_G}\boldsymbol{x} \end{bmatrix}, \tag{85}$$

□

i.e., $\boldsymbol{z}$ is n_G times repetition of the input $\boldsymbol{x}$ scaled by parameter associated with the Lie algebra generator.

Now following Sec. E.1 we express the invariance error as

$$\begin{aligned} \|R(g)\| &= \|\sum_i t_i \boldsymbol{w}^\top d\rho_{\mathcal{X}}(A_i)\boldsymbol{x}\| && (86)\\ &= \|\boldsymbol{w}^\top \sum_i t_i d\rho_{\mathcal{X}}(A_i)\boldsymbol{x}\| && (87)\\ &= \|\boldsymbol{w}^\top \boldsymbol{A}\boldsymbol{z}\| && (88)\\ &= \|\boldsymbol{w}^\top \boldsymbol{U}\Sigma\boldsymbol{V}^\top\boldsymbol{z}\| && (89)\\ &= \|(\boldsymbol{U}^\top\boldsymbol{w})^\top\Sigma(\boldsymbol{V}^\top\boldsymbol{z})\| && (90)\\ &= \|W^\top\Sigma Z\|. && (91) \end{aligned}$$

Following the condition of $\boldsymbol{B}_{\texttt{inv}}$, $W_j = 0$ for $\sigma_j > b$. We can rewrite the bound as

$$\begin{aligned} \|R(g)\| &\le \|\sum_{\sigma_j < b} bW_jZ_j\| && (92)\\ &\le b\|\boldsymbol{w}\|\|\boldsymbol{z}\| \quad \text{by Cauchy-Schwarz and } \|W\| = \|\boldsymbol{w}\|, \|Z\| = \|\boldsymbol{z}\| && (93)\\ &\le b\|\boldsymbol{w}\|\sqrt{\sum_i t_i^2}\|\boldsymbol{x}\| \quad \text{following the construction of } \boldsymbol{z} && (94)\\ &\le br_G\|\boldsymbol{w}\|\|\boldsymbol{x}\| \quad \text{using the definition of injective radius} && (95)\\ &\le b\sqrt{n_G}r_G\|\boldsymbol{w}\|\|\boldsymbol{x}\|. \quad \text{as } n_G \ge 1 && (96) \end{aligned}$$

Finally, following the definition of $R(g)$, and Jacobian of the linear function $F_{\texttt{FC}}$, we obtain

$$\frac{\|(\mathbf{B}\theta)^\top\mathbf{x} - (\mathbf{B}\theta)^\top\rho_{\mathcal{X}}(g)\mathbf{x}\|}{\|\mathbf{J}_{F_{\texttt{FC}}(;\boldsymbol{w})}(\mathbf{x})\|_{\texttt{F}}\|\mathbf{x}\|} \le b\sqrt{n_G}r_G + \varepsilon_G. \tag{97}$$

Thus, the designed projection operator via combined decomposition satisfies Claim 1.

Note: An alternative approach for designing a projection operator for groups with multiple generators is to design a projection operator $B_{\texttt{inv,i}}$ for each generator A_i and the final projection operator can be approximated via compositions $B_{\texttt{inv}} \approx B_{\texttt{inv,1}} \circ B_{\texttt{inv,2}} \circ \dots B_{\texttt{inv,n}_\texttt{G}}$. For commutative generators, this approximation is exact. For non-commutative generators, better approximations can be obtained by recursive application of these projection operators.

E.3. Proof of Claim 2

Claim 2. *For any compact and connected Lie group G with injective radius r_G and n_G generators, let $\boldsymbol{W}$ be defined as in (12). Then $F_{\texttt{FC}}(\boldsymbol{x}, \boldsymbol{W})$ is η_b-soft equivariant, i.e.,*

$$\frac{\|\boldsymbol{W}\rho_{\mathcal{X}}(g)\boldsymbol{x} - \rho_{\mathcal{Y}}(g)\boldsymbol{W}\boldsymbol{x}\|}{\|\boldsymbol{J}_{F_{\texttt{FC}}(\cdot;\boldsymbol{W})}(\boldsymbol{x})\|_{\texttt{F}}\,\|\boldsymbol{x}\|} \le \eta_b, \quad \forall\, g \in G, \tag{16}$$

where $\eta_b = b\sqrt{n_G\, d'}\, r_G + \varepsilon_G$, $b \in \mathbb{R}^+$ is the cut-off value of the projection operator $\boldsymbol{B}_{\texttt{eq}}$, d' is the output dimension, ε_G is the residual from the first-order Taylor approximation.

Proof. Applying the Taylor expansion of the equivariance error Eq. (65) in the function $F_{\texttt{FC}} = \boldsymbol{W}\boldsymbol{x}$ with $\texttt{vec}(\boldsymbol{W}) = \boldsymbol{B}'_{\texttt{eq}}\theta \in \mathbb{R}^{d'd}$ and retaining only the linear terms, we obtain:

$$
\begin{aligned}
R(g) &= \sum_i t_i \left. \frac{\partial \left(\boldsymbol{W}\rho_{\mathcal{X}}(\exp(\sum_i t_i A_i))\boldsymbol{x} - \rho_{\mathcal{Y}}(\exp(\sum_i t_i A_i))\boldsymbol{W}\boldsymbol{x}\right)}{\partial t_i} \right|_{t=0} && (98)\\
&= \sum_i t_i \left(\boldsymbol{W} d\rho_{\mathcal{X}}(A_i)\boldsymbol{x} - d\rho_{\mathcal{Y}}(A_i)\boldsymbol{W}\boldsymbol{x}\right) && (99)\\
&= \sum_i t_i \left(\boldsymbol{W} d\rho_{\mathcal{X}}(A_i) - d\rho_{\mathcal{Y}}(A_i)\boldsymbol{W}\right)\boldsymbol{x} && (100)\\
&= \sum_i t_i (\boldsymbol{x}^\top \otimes \boldsymbol{I}_{d'})\texttt{vec}(\boldsymbol{W} d\rho_{\mathcal{X}}(A_i) - d\rho_{\mathcal{Y}}(A_i)\boldsymbol{W}) \quad \big(\text{using the fact } \boldsymbol{M}\boldsymbol{z} = (\boldsymbol{z}^\top \otimes \boldsymbol{I}_{d'})\texttt{vec}(\boldsymbol{M})\big) && (101)\\
&= \sum_i t_i (\boldsymbol{x}^\top \otimes \boldsymbol{I}_{d'})\texttt{vec}(\boldsymbol{I}_{d'}\boldsymbol{W} d\rho_{\mathcal{X}}(A_i) - d\rho_{\mathcal{Y}}(A_i)\boldsymbol{W}\boldsymbol{I}_d) && (102)\\
&= \sum_i t_i (\boldsymbol{x}^\top \otimes \boldsymbol{I}_{d'})(d\rho_{\mathcal{X}}(A_i)^\top \otimes \boldsymbol{I}_{d'} - \boldsymbol{I}_d \otimes d\rho_{\mathcal{Y}}(A_i))\texttt{vec}(\boldsymbol{W}) \quad \big(\text{as } \texttt{vec}(\boldsymbol{A}\boldsymbol{B}\boldsymbol{C}) = \big(\boldsymbol{C}^\top \otimes \boldsymbol{A}\big)\,\texttt{vec}(\boldsymbol{B})\big). && (103)
\end{aligned}
$$

We denote $\boldsymbol{L}_i = d\rho_{\mathcal{X}}(A_i)^\top \otimes \mathbf{I}_{d'} - \mathbf{I}_d \otimes d\rho_{\mathcal{Y}}(A_i)$ and $\overline{\boldsymbol{x}} = \boldsymbol{x}^\top \otimes \mathbf{I}_{d'}$ and define the SVD of $\boldsymbol{L}_i$ as

$$
\boldsymbol{L}_i = \boldsymbol{U}^{\mathtt{L_i}} \Sigma^{\mathtt{L_i}} \boldsymbol{V}^{\mathtt{L_i}\top}, \tag{104}
$$

where $\mathbf{U}^{L_i}$ and $\mathbf{V}^{L_i}$ are orthogonal matrices and Σ^{L_i} is a diagonal matrix with singular values $0 \le \sigma_1 \le \sigma_2 \le \ldots$ arranged in increasing order. We can express the equivariance error norm as:

$$
\begin{aligned}
\|R(g)\| &= \left\| \sum_i t_i \overline{\boldsymbol{x}} \boldsymbol{L}_i \texttt{vec}(\boldsymbol{W}) \right\| && (105)\\
&\le \sum_i |t_i| \cdot \|\overline{\boldsymbol{x}} \boldsymbol{L}_i \texttt{vec}(\boldsymbol{W})\| && (106)\\
&= \sum_i |t_i| \|\overline{\boldsymbol{x}} \boldsymbol{U}^{\mathtt{L_i}} \Sigma^{\mathtt{L_i}} \boldsymbol{V}^{\mathtt{L_i}\top} \texttt{vec}(\boldsymbol{W})\| && (107)\\
&= \sum_i |t_i| \|(\boldsymbol{U}^{\mathtt{L_i}\top} \overline{\boldsymbol{x}}^\top)^\top \Sigma^{\mathtt{L_i}} (\boldsymbol{V}^{\mathtt{L_i}\top} \texttt{vec}(\boldsymbol{W}))\| && (108)\\
&= \sum_i |t_i| \|\overline{X}_i^\top \Sigma^{\mathtt{L_i}} \overline{W}_i\|, && (109)
\end{aligned}
$$

where $\overline{X}^i = (\boldsymbol{U}^{\mathtt{L_i}\top} \overline{\boldsymbol{x}}^\top) \in \mathbb{R}^{d'd \times d'}$ and $\overline{W}^i = \boldsymbol{V}^{\mathtt{L_i}\top} \texttt{vec}(\boldsymbol{W}) \in \mathbb{R}^{d'd}$ are the coordinates of $\overline{\boldsymbol{x}}$ and $\texttt{vec}(\boldsymbol{W})$ in the left and right singular vector bases, respectively. Here $\overline{X}^i$ is a matrix with d' columns, where $\overline{X}^i_c$ denotes its c-th column, and $\overline{X}^i_{c,j}$ denotes the j-th component of that column.

As b is the cut-off value of the projection of operator $\boldsymbol{B}_{\texttt{eq}}$, $\overline{W}^i_j = 0$ for all $\sigma^i_j > b$ (follows from the definition of the cut-off values and holds for all generators).

Therefore, we can write:

$$\|R(g)\| \leq \sum_i |t_i| \|\overline{X}^{i\top} \Sigma^{\mathtt{L_i}} \overline{W}^i\| \tag{110}$$

$$\leq \sum_i |t_i| \sum_{c=1}^{d'} \left\| \overline{X}_c^{i\top} \Sigma^{\mathtt{L_i}} \overline{W}^i \right\| \tag{111}$$

$$\leq \sum_i |t_i| \sum_{c=1}^{d'} \left\| \sum_{j:\sigma_j^i < b} \sigma_j^i \overline{X}_{c,j}^i \overline{W}_j^i \right\| \tag{112}$$

$$\leq \sum_i |t_i| \sum_{c=1}^{d'} \sum_{j:\sigma_j^i < b} \sigma_j^i |\overline{X}_{c,j}^i| |\overline{W}_j^i| \tag{113}$$

$$\leq \sum_i |t_i| b \sum_{c=1}^{d'} \sum_{j:\sigma_j^i < b} |\overline{X}_{c,j}^i| |\overline{W}_j^i| \tag{114}$$

$$\leq \sum_i |t_i| b \left(\sum_{c=1}^{d'} \sum_{j:\sigma_j^i < b} |\overline{X}_{c,j}^i|^2 \right)^{1/2} \left(\sum_{j:\sigma_j^i \leq b} |\overline{W}_j^i|^2 \right)^{1/2} \tag{115}$$

$$\leq \sum_i |t_i| b \|\overline{X}^i\|_{\mathtt{F}} \|\overline{W}^i\| \tag{116}$$

$$\leq \sum_i |t_i| b \|\overline{\boldsymbol{x}}\|_{\mathtt{F}} \|\mathtt{vec}(\boldsymbol{W})\| \quad \text{(as } \mathbf{U}^{L_i} \text{ and } \mathbf{V}^{L_i} \text{ are orthonormal).} \tag{117}$$

Now we compute $\|\overline{\boldsymbol{x}}\|_{\mathtt{F}}$ where $\overline{\boldsymbol{x}} = \boldsymbol{x}^\top \otimes \mathbf{I}_{d'} \in \mathbb{R}^{d' \times d'd}$:

$$\|\overline{\boldsymbol{x}}\|_{\mathtt{F}}^2 = \|\boldsymbol{x}^\top \otimes \mathbf{I}_{d'}\|_{\mathtt{F}}^2 \tag{118}$$

$$= \operatorname{tr}\left((\boldsymbol{x}^\top \otimes \mathbf{I}_{d'})(\boldsymbol{x}^\top \otimes \mathbf{I}_{d'})^\top \right) \tag{119}$$

$$= \operatorname{tr}\left((\boldsymbol{x}^\top \otimes \mathbf{I}_{d'})(\boldsymbol{x} \otimes \mathbf{I}_{d'}) \right) \tag{120}$$

$$= \operatorname{tr}\left((\boldsymbol{x}^\top \boldsymbol{x}) \otimes (\mathbf{I}_{d'} \mathbf{I}_{d'}) \right) \tag{121}$$

$$= \operatorname{tr}\left(\|\boldsymbol{x}\|^2 \otimes \mathbf{I}_{d'} \right) \tag{122}$$

$$= \|\boldsymbol{x}\|^2 \cdot \operatorname{tr}(\mathbf{I}_{d'}) \tag{123}$$

$$= d' \|\boldsymbol{x}\|^2. \tag{124}$$

Therefore, $\|\overline{\boldsymbol{x}}\|_{\mathtt{F}} = \sqrt{d'} \|\boldsymbol{x}\|$. Substituting the norms, we can rewrite the inequality as:

$$\|R(g)\| \leq \sum_i |t_i| b \sqrt{d'} \|\boldsymbol{x}\| \|\boldsymbol{W}\|_{\mathtt{F}} \tag{125}$$

as $\|\mathtt{vec}(\boldsymbol{W})\| = \|\boldsymbol{W}\|_{\mathtt{F}}$.

For compact and connected Lie groups $\sqrt{\sum_i t_i^2} \leq r_G$ (assuming an orthonormal basis for the Lie algebra), where r_G is the injective radius of the group G. Applying Cauchy-Schwarz inequality on $\sum_i |t_i|$ gives:

$$\sum_i |t_i| \leq \sqrt{n_G} \sqrt{\sum_i t_i^2} \leq r_G \sqrt{n_G}, \tag{126}$$

where n_G is the number of Lie algebra basis elements of the group G.

As the function $F_{\mathtt{FC}}$ is linear, the norm of the Jacobian at $\boldsymbol{x}$ is $\|\mathbf{J}_{F(;\boldsymbol{W})_{\mathtt{FC}}}(\boldsymbol{x})\|_{\mathtt{F}} = \|\boldsymbol{W}\|_{\mathtt{F}}$. Substituting the definition of

$R(g)$ and $\|\mathbf{J}_{F(;\boldsymbol{W})_{\text{FC}}}(\boldsymbol{x})\|$, and reintroducing the Taylor residual error ϵ_G, we obtain the final bound on the equivariance error:

$$\|\boldsymbol{W}\rho_{\mathcal{X}}(g)\boldsymbol{x} - \rho_{\mathcal{Y}}(g)\boldsymbol{W}\boldsymbol{x}\| \leq br_G\sqrt{n_G d'}\|\mathbf{J}_{F(;\boldsymbol{W})_{\text{FC}}}(\boldsymbol{x})\|_{\text{F}}\|\boldsymbol{x}\| + \epsilon_G \tag{127}$$

$$\implies \frac{\|\mathbf{B}'\Theta\rho_{\mathcal{X}}(g)\boldsymbol{x} - \rho_{\mathcal{Y}}(g)\mathbf{B}'\Theta\boldsymbol{x}\|}{\|\mathbf{J}_{F(;\boldsymbol{W})_{\text{FC}}}(\boldsymbol{x})\|_{\text{F}}\|\boldsymbol{x}\|} \leq br_G\sqrt{n_G d'} + \varepsilon_G. \tag{128}$$

Here, ε_G is the normalized Taylor residual. □

E.4. Proof of Lemma 1

Lemma 1. *The weight matrix Θ is equivariant if it satisfies the condition*

$$\boldsymbol{\Sigma}_{\mathcal{Y}}\Theta' - \Theta'\boldsymbol{\Sigma}_{\mathcal{X}} = \mathbf{0} \iff \boldsymbol{T}_l\Theta'_{lk} = \Theta'_{lk}\boldsymbol{S}_k \quad \forall l, k, \tag{19}$$

where Θ'_{lk} are blocks of Θ' corresponding to the blocks of $\boldsymbol{\Sigma}_{\mathcal{Y}}$ and $\boldsymbol{\Sigma}_{\mathcal{X}}$ of dimensions $\dim(\boldsymbol{T}_l) \times \dim(\boldsymbol{S}_k)$.

Proof. Following Eq. (100), we can write the condition for exact equivariance as for weight $\Theta \in \mathbb{R}^{d' \times d}$ as:

$$\|R(g)\| = 0 \implies \left\|\sum_i t_i \left(\Theta d\rho_{\mathcal{X}}(A_i) - d\rho_{\mathcal{Y}}(A_i)\Theta\right)\boldsymbol{x}\right\| = 0. \tag{129}$$

The above condition holds if

$$\|\Theta d\rho_{\mathcal{X}}(A_i) - d\rho_{\mathcal{Y}}(A_i)\Theta\|_{\text{F}} = 0 \quad \forall i. \tag{130}$$

As the condition is the same for all Lie algebra generators A_i, we drop the index A_i for readability. We define the Schur decomposition of the Lie algebra representations as:

$$d\rho_{\mathcal{X}} = \boldsymbol{U}_{\mathcal{X}}\boldsymbol{\Sigma}_{\mathcal{X}}\boldsymbol{U}_{\mathcal{X}}^\top, \quad d\rho_{\mathcal{Y}} = \boldsymbol{U}_{\mathcal{Y}}\boldsymbol{\Sigma}_{\mathcal{Y}}\boldsymbol{U}_{\mathcal{Y}}^\top, \tag{131}$$

where $\boldsymbol{U}_{\mathcal{X}}$ and $\boldsymbol{U}_{\mathcal{Y}}$ are orthonormal. And we denote $\Theta' = \boldsymbol{U}_{\mathcal{Y}}^\top\Theta\boldsymbol{U}_{\mathcal{X}}$ Substituting the Schur decomposition in the equivariance condition, we obtain:

$$\|\Theta d\rho_{\mathcal{X}} - d\rho_{\mathcal{Y}}\Theta\|_{\text{F}} = 0 \tag{132}$$

$$\implies \|\Theta\boldsymbol{U}_{\mathcal{X}}\boldsymbol{\Sigma}_{\mathcal{X}}\boldsymbol{U}_{\mathcal{X}}^\top - \boldsymbol{U}_{\mathcal{Y}}\boldsymbol{\Sigma}_{\mathcal{Y}}\boldsymbol{U}_{\mathcal{Y}}^\top\Theta\|_{\text{F}} = 0 \tag{133}$$

$$\implies \|\boldsymbol{U}_{\mathcal{Y}}^\top(\Theta\boldsymbol{U}_{\mathcal{X}}\boldsymbol{\Sigma}_{\mathcal{X}}\boldsymbol{U}_{\mathcal{X}}^\top - \boldsymbol{U}_{\mathcal{Y}}\boldsymbol{\Sigma}_{\mathcal{Y}}\boldsymbol{U}_{\mathcal{Y}}^\top\Theta)\boldsymbol{U}_{\mathcal{X}}\|_{\text{F}} = 0 \quad \text{(orthonormal matrix preserves norm)} \tag{134}$$

$$\implies \|\boldsymbol{U}_{\mathcal{Y}}^\top\Theta\boldsymbol{U}_{\mathcal{X}}\boldsymbol{\Sigma}_{\mathcal{X}}\boldsymbol{U}_{\mathcal{X}}^\top\boldsymbol{U}_{\mathcal{X}} - \boldsymbol{U}_{\mathcal{Y}}^\top\boldsymbol{U}_{\mathcal{Y}}\boldsymbol{\Sigma}_{\mathcal{Y}}\boldsymbol{U}_{\mathcal{Y}}^\top\Theta\boldsymbol{U}_{\mathcal{X}}\|_{\text{F}} = 0 \tag{135}$$

$$\implies \|\Theta'\boldsymbol{\Sigma}_{\mathcal{X}} - \boldsymbol{\Sigma}_{\mathcal{Y}}\Theta'\|_{\text{F}} = 0 \tag{136}$$

This condition holds when

$$\boldsymbol{\Sigma}_{\mathcal{Y}}\Theta' - \Theta'\boldsymbol{\Sigma}_{\mathcal{X}} = \mathbf{0}. \tag{137}$$

Now as $d\rho_{\mathcal{X}}$ and $d\rho_{\mathcal{Y}}$ are normal matrices, $\boldsymbol{\Sigma}_{\mathcal{X}}$ and $\boldsymbol{\Sigma}_{\mathcal{Y}}$ are block diagonal matrices *i.e.*, $\boldsymbol{\Sigma}_{\mathcal{X}} = \text{diag}(\{\boldsymbol{S}_k\}_{k=1}^p)$ and $\boldsymbol{\Sigma}_{\mathcal{Y}} = \text{diag}(\{\boldsymbol{T}_l\}_{l=1}^q)$, where $\{\boldsymbol{S}_k\}_{k=1}^p$ and $\{\boldsymbol{T}_l\}_{l=1}^q$ are sets of 1×1 or 2×2 Schur form blocks. We decompose Θ' into blocks Θ'_{lk} corresponding to the blocks of $\boldsymbol{\Sigma}_{\mathcal{X}}$ and $\boldsymbol{\Sigma}_{\mathcal{Y}}$ of size $\dim(\boldsymbol{T}_l) \times \dim(\boldsymbol{S}_k)$ *i.e.*, Θ'_{lk} is a submatrix of Θ' corresponding to the rows associated with block he $\boldsymbol{T}_l$ in $\boldsymbol{\Sigma}_{\mathcal{Y}}$ and columns associated with the block $\boldsymbol{S}_k$ in $\boldsymbol{\Sigma}_{\mathcal{X}}$.

As the 0 entries of $\boldsymbol{\Sigma}_{\mathcal{X}}$ and $\boldsymbol{\Sigma}_{\mathcal{Y}}$ do not contribute to the matrix multiplication, we can rewrite Eq. (137) as a blockwise condition:

$$\boldsymbol{T}_l\Theta'_{lk} - \Theta'_{lk}\boldsymbol{S}_k = \mathbf{0} \quad \forall l, k. \tag{138}$$

□

E.5. Proof of Claim 3

Claim 3. *For any Lie group G with normal Lie algebra representations with injective radius r_G and n_G generators. Let $\boldsymbol{W} = \boldsymbol{U}_{\mathcal{Y}} \boldsymbol{B}_{\texttt{Schur}} [\Theta'] \boldsymbol{U}_{\mathcal{X}}^{\top}$, then the function $F_{\texttt{FC}}(\boldsymbol{x}, \boldsymbol{W})$ is η_b-soft equivariant*

$$\frac{\|\boldsymbol{W}\rho_{\mathcal{X}}(g)\boldsymbol{x} - \rho_{\mathcal{Y}}(g)\boldsymbol{W}\boldsymbol{x}\|}{\|\boldsymbol{J}_{F_{\texttt{FC}}(\cdot;\boldsymbol{W})}(\boldsymbol{x})\|_{\texttt{F}}\|\boldsymbol{x}\|} \leq \eta_b, \forall g \in G \tag{23}$$

with $\eta_b = b\sqrt{n_G} r_G + \varepsilon_G$, b is the cut-off value of the Schur filter $\boldsymbol{B}_{\texttt{Schur}}$ and ε_G is the residual from the first-order Taylor approximation.

Proof. Applying the Taylor expansion of the equivariance error Eq. (65) for the function $F_{\texttt{FC}}(x, \boldsymbol{W}) = \boldsymbol{W}\boldsymbol{x}$ and retaining only the linear terms, we obtain:

$$R(g) = \sum_i t_i \left. \frac{\partial \left(\mathbf{W}\rho_{\mathcal{X}}(\exp(t_i A_i))\boldsymbol{x} - \rho_{\mathcal{Y}}(\exp(t_i A_i))\mathbf{W}\boldsymbol{x}\right)}{\partial t_i} \right|_{t=0} \tag{139}$$

$$= \sum_i t_i \left(\mathbf{W} d\rho_{\mathcal{X}}(A_i)\boldsymbol{x} - d\rho_{\mathcal{Y}}(A_i)\mathbf{W}\boldsymbol{x}\right) \tag{140}$$

$$= \sum_i t_i \left(\mathbf{W} d\rho_{\mathcal{X}}(A_i) - d\rho_{\mathcal{Y}}(A_i)\mathbf{W}\right)\boldsymbol{x}. \tag{141}$$

Now, we denote $\boldsymbol{W}^{i\prime} = {\boldsymbol{U}_{\mathcal{Y}}^{i}}^{\top} \boldsymbol{W} \boldsymbol{U}_{\mathcal{X}}^{i}$ where $\boldsymbol{U}_{\mathcal{X}}^{i}$ and $\boldsymbol{U}_{\mathcal{Y}}^{i}$ are the orthonormal matrices from the Schur decomposition of $d\rho_{\mathcal{X}}(A_i)$ and $d\rho_{\mathcal{Y}}(A_i)$ respectively. We denote $\boldsymbol{\Sigma}_{\mathcal{X}} = \operatorname{diag}(\{\boldsymbol{S}_k^i\}_{k=1}^{p})$ and $\boldsymbol{\Sigma}_{\mathcal{Y}} = \operatorname{diag}(\{\boldsymbol{T}_l^i\}_{l=1}^{q})$, where $\{\boldsymbol{S}_k^i\}_{k=1}^{p}$ and $\{\boldsymbol{T}_l^i\}_{l=1}^{q}$ are sets of 1×1 or 2×2 Schur form blocks corresponding to the Schur decomposition of $d\rho_{\mathcal{X}}(A_i)$ and $d\rho_{\mathcal{Y}}(A_i)$. By $\lambda_{\boldsymbol{S}_k^i}$ and $\lambda_{\boldsymbol{T}_l^l}$ we denote the norm of the highest eigenvalue of the blocks $\boldsymbol{S}_k^i$ and $\boldsymbol{T}_l^l$ respectively. We can express the equivariance error norm $|R(g)|$ to be upper-bounded by:

$$\sum_i |t_i| \|\boldsymbol{W} d\rho_{\mathcal{X}}(A_i) - d\rho_{\mathcal{Y}}(A_i)\boldsymbol{W}\|_2 \|\boldsymbol{x}\| \tag{142}$$

$$= \sum_i |t_i| \|(\boldsymbol{U}_{\mathcal{Y}}^i \boldsymbol{W}^{i\prime} {\boldsymbol{U}_{\mathcal{X}}^i}^{\top})(\boldsymbol{U}_{\mathcal{X}}^i \boldsymbol{\Sigma}_{\mathcal{X}}^i {\boldsymbol{U}_{\mathcal{X}}^i}^{\top}) - (\boldsymbol{U}_{\mathcal{Y}}^i \boldsymbol{\Sigma}_{\mathcal{Y}}^i {\boldsymbol{U}_{\mathcal{Y}}^i}^{\top})(\boldsymbol{U}_{\mathcal{Y}}^i \boldsymbol{W}^{i\prime} {\boldsymbol{U}_{\mathcal{X}}^i}^{\top})\|_2 \|\boldsymbol{x}\| \tag{143}$$

$$= \sum_i |t_i| \|\boldsymbol{W}^{i\prime} \boldsymbol{\Sigma}_{\mathcal{X}}^i - \boldsymbol{\Sigma}_{\mathcal{Y}}^i \boldsymbol{W}^{i\prime}\|_2 \|\boldsymbol{x}\| \quad \text{(spectral norm of orthonormal matrices is 1)} \tag{144}$$

$$\leq \sum_i |t_i| \|\boldsymbol{W}^{i\prime} \boldsymbol{\Sigma}_{\mathcal{X}}^i - \boldsymbol{\Sigma}_{\mathcal{Y}}^i \boldsymbol{W}^{i\prime}\|_{\texttt{F}} \|\boldsymbol{x}\| \tag{145}$$

$$= \sum_i |t_i| \left(\sum_{l,k} \|{\boldsymbol{W}^{i\prime}}_{lk} \boldsymbol{S}_k^i - \boldsymbol{T}_l^i {\boldsymbol{W}^{i\prime}}_{lk}\|_{\texttt{F}}^2\right)^{1/2} \|\boldsymbol{x}\| \tag{146}$$

$$= \sum_i |t_i| \left(\sum_{l,k:\lambda_{\boldsymbol{S}_k^i} + \lambda_{\boldsymbol{T}_l^i} \leq b} \|{\boldsymbol{W}^{i\prime}}_{lk} \boldsymbol{S}_k^i - \boldsymbol{T}_l^i {\boldsymbol{W}^{i\prime}}_{lk}\|_{\texttt{F}}^2\right)^{1/2} \|\boldsymbol{x}\| \quad (\text{as } \boldsymbol{B}_{\texttt{Schur}}, {\boldsymbol{W}^{i\prime}}_{lk}\boldsymbol{S}_k^i - \boldsymbol{T}_l^i {\boldsymbol{W}^{i\prime}}_{lk} = \boldsymbol{0} \text{ when } \lambda_{\boldsymbol{S}_k^i} + \lambda_{\boldsymbol{T}_l^i} > b) \tag{147}$$

$$\leq \sum_i |t_i| \left(\sum_{l,k:\lambda_{\boldsymbol{S}_k^i} + \lambda_{\boldsymbol{T}_l^i} \leq b} (\lambda_{\boldsymbol{S}_k^i} + \lambda_{\boldsymbol{T}_l^i})^2 \|{\boldsymbol{W}^{i\prime}}_{lk}\|_{\texttt{F}}^2\right)^{1/2} \|\boldsymbol{x}\| \quad (\text{ maximum scaling is bounded by the highest eigenvalues}) \tag{148}$$

$$\leq \sum_i |t_i| \left(\sum_{l,k:\lambda_{\boldsymbol{S}_k^i} + \lambda_{\boldsymbol{T}_l^i} \leq b} b^2 \|{\boldsymbol{W}^{i\prime}}_{lk}\|_F^2\right)^{1/2} \|\boldsymbol{x}\| \tag{149}$$

$$\leq \sum_i |t_i| b \left(\sum_{l,k} \|{\boldsymbol{W}^{i\prime}}_{lk}\|_{\texttt{F}}^2\right)^{1/2} \|\boldsymbol{x}\| \tag{150}$$

$$= \sum_i |t_i| b \|\boldsymbol{W}^{i'}\|_{\mathrm{F}} \|\boldsymbol{x}\| \tag{151}$$

$$= \sum_i |t_i| b \|\boldsymbol{W}\|_{\mathrm{F}} \|\boldsymbol{x}\| \quad \text{(as } \boldsymbol{U}_{\mathcal{X}}^i \text{ and } \boldsymbol{U}_{\mathcal{Y}}^i \text{ are orthonormal).} \tag{152}$$

For compact and connected Lie groups, applying the Cauchy-Schwarz inequality and using the injective radius property:

$$\sum_i |t_i| \leq \sqrt{n_G} \sqrt{\sum_i t_i^2} \leq \sqrt{n_G} r_G. \tag{153}$$

Substituting the definition of $R(g)$ and using $\|\mathbf{J}_{F_{\mathrm{FC}}(;\boldsymbol{W})}(\boldsymbol{x})\|_F = \|\boldsymbol{W}\|_F$, and reintroducing the Taylor residual error ϵ_G, we can state

$$\|\boldsymbol{B}_s \Theta \rho_{\mathcal{X}}(g)\boldsymbol{x} - \rho_{\mathcal{Y}}(g)\boldsymbol{B}_s\Theta\boldsymbol{x}\| \leq b r_G \sqrt{n_G} \|\boldsymbol{W}\|_F \|\boldsymbol{x}\| + \epsilon_G \implies \frac{\|\boldsymbol{B}_s \Theta \rho_{\mathcal{X}}(g)\boldsymbol{x} - \rho_{\mathcal{Y}}(g)\boldsymbol{B}_s\Theta\boldsymbol{x}\|}{\|\mathbf{J}_{F_{\mathrm{FC}}(;\boldsymbol{W})}(\boldsymbol{x})\|_F \|\boldsymbol{x}\|} \leq b\sqrt{n_G} r_G + \varepsilon_G. \tag{154}$$

Here, ε_G is the normalized Taylor residual. □

E.6. Proof of Lemma 2

Lemma 2. *Let G be a finite discrete group with generating set $\mathbb{S} = \{s_1, \ldots, s_k\}$, and $f : G \to \mathbb{R}$ be h-Lipschitz with respect to the word metric $d_{\mathbb{S}}$, then the first-order Taylor approximation of f at the identity element e defined as $\hat{f}(g) \triangleq f(e) + \sum_{i=1}^k n_{s_i} \Delta_{s_i} f(e)$ satisfies the following point-wise error bound:*

$$\left| f(g) - \hat{f}(g) \right| \leq 2h \cdot d_{\mathbb{S}}(e, g), \tag{25}$$

where n_{s_i} number of occurrence of s_i in the canonical word representation of g.

Proof. By definition of $\hat{f}$, the approximation error is:

$$\left| f(g) - \hat{f}(g) \right| = \left| f(g) - f(e) - \sum_{i=1}^k n_{s_i} \Delta_{s_i} f(e) \right|. \tag{155}$$

Applying the triangle inequality:

$$\left| f(g) - \hat{f}(g) \right| \leq \underbrace{|f(g) - f(e)|}_{\text{(I)}} + \underbrace{\sum_{i=1}^k n_i \left| \Delta_{s_i} f(e) \right|}_{\text{(II)}}. \tag{156}$$

Recall that the word metric $d_{\mathbb{S}}(g, g')$ counts the minimum number of generator from $\mathbb{S}$ needed to reach g' from g; in particular, for $g = s_1^{n_1} \cdots s_m^{n_m}$, the word metric can be expressed as $d_{\mathbb{S}}(e, g) = \sum_{i=1}^m n_i = \sum_{i=1}^k n_{s_i}$, where n_{s_i} is the total number of occurrence of s_i in $s_1^{n_1} \cdots s_m^{n_m}$ the canonical word representation of g .
Bounding (I). Since $d_{\mathbb{S}}(e, g) = \sum_{i=1}^k n_i$ and f is h-Lipschitz:

$$|f(g) - f(e)| \leq h \cdot d_{\mathbb{S}}(e, g). \tag{157}$$

Bounding (II). Each generator s_i satisfies $d_{\mathbb{S}}(e, s_i) = 1$, so by h-Lipschitz:

$$|\Delta_{s_i} f(e)| = |f(s_i \cdot e) - f(e)| \leq h \cdot d_{\mathbb{S}}(e, s_i) = h. \tag{158}$$

Summing over all generators:

$$\sum_{i=1}^k n_i \left| \Delta_{s_i} f(e) \right| \leq h \sum_{i=1}^k n_i = h \cdot d_{\mathbb{S}}(e, g). \tag{159}$$

Combining the two bounds:

$$\left| f(g) - \hat{f}(g) \right| \leq h \cdot d_{\mathbb{S}}(e, g) + h \cdot d_{\mathbb{S}}(e, g) = 2h \cdot d_{\mathbb{S}}(e, g), \tag{160}$$

which completes the proof.

□

Table 8. **Human Trajectory Prediction Results (ETH/UCY).** All metrics are lower is better.

Method	ETH						UNIV					
	ADE↓	aADE↓	cADE↓	FDE↓	aFDE↓	cFDE↓	ADE↓	aADE↓	cADE↓	FDE↓	aFDE↓	cFDE↓
Base [19]	4.67	4.80	4.73	**6.04**	**6.27**	**6.15**	7.90	7.92	7.91	8.16	8.17	8.16
EqAuto [10]	5.40	5.40	5.40	7.33	7.33	7.33	8.16	8.16	8.16	8.33	8.33	8.33
Ours	**4.50**	**4.66**	**4.58**	6.10	6.35	6.23	**7.82**	**7.88**	**7.85**	**8.04**	**8.09**	**8.07**

Method	ZARA1						ZARA2					
	ADE↓	aADE↓	cADE↓	FDE↓	aFDE↓	cFDE↓	ADE↓	aADE↓	cADE↓	FDE↓	aFDE↓	cFDE↓
Base [19]	3.57	3.65	3.61	**4.63**	4.73	4.68	3.18	3.15	3.17	3.83	3.76	3.79
EqAuto [10]	3.66	3.66	3.66	4.98	4.98	4.98	2.94	2.94	2.94	3.63	3.63	3.63
Ours	**3.39**	**3.41**	**3.40**	4.65	**4.68**	**4.67**	**2.91**	**2.91**	**2.91**	**3.60**	**3.60**	**3.60**

Method	HOTEL					
	ADE↓	aADE↓	cADE↓	FDE↓	aFDE↓	cFDE↓
Base [19]	5.83	5.85	5.84	6.64	6.71	6.67
EqAuto [10]	6.16	6.16	6.16	6.78	6.78	6.78
Ours	**5.69**	**5.69**	**5.69**	**6.26**	**6.25**	**6.26**

F. Additional Experiments and Results

F.1. Synthetic $O(5)$ benchmark

We follow the setup of [17] to create a synthetic benchmark for $O(5)$ invariant learning.

Task Description Given two vectors $x_1, x_2 \in \mathbb{R}^5$, the task is to predict the following $O(5)$ invariant quantity $f(x_1, x_2) = \sin(\|x_1\|) - \|x_2\|^3/2 + \frac{x_1^\top x_2}{\|x_1\|\|x_2\|}$. To induce a more challenging learning problem, we deviate from the strict invariant target function f and reformulate the target function as

$$\tilde{f}(x_1, x_2) = \sin(\|x_1\|) - \|x_2\|^3/2 + \frac{x_1^\top x_2}{\|x_1\|\|x_2\|} + \underbrace{\gamma \frac{|x_1^1| + |x_1^2| + |x_2^1| + |x_2^2|}{|x_1^3| + |x_1^4| + |x_2^3| + |x_2^4| + \varepsilon}}_{\text{non-invariant term}}, \tag{161}$$

where γ is a hyperparameter that controls the degree of deviation from strict invariance and x_j^k denotes the kth component of the vector x_j. When $\gamma = 0$, the target function is strictly invariant, and as γ increases, the target function becomes less invariant.

We present the results in Tab. 9, which show that our method outperforms the baselines, with EMLP-RPP [16] being the second-best.

Table 9. **Relative MSE (10^{-1}) for varying γ on Synthetic $O(5)$ benchmark.**

Model	$\gamma = 0.3$	$\gamma = 0.4$	$\gamma = 0.5$
EMLP [17]	<u>0.78</u>	0.88	1.00
EMLP-RPP [16]	0.79	<u>0.81</u>	<u>0.84</u>
MLP	1.21	1.20	1.29
Ours	**0.72**	**0.74**	**0.81**

F.2. Additional ablation studies and extended results

We evaluate the benefits of soft equivariant models we also report the performance of full equivariant models. However, fully equivariant pretrained models with architectures and parameter counts similar to ViT or DINOv2 are not publicly available. So we use our projection operator to create a full equivariant model from a pretrained non-equivariant model. We present the results in Tab. 10 on the classification and segmentation tasks following the setup of Sec. 5.2 and Sec. 5.3. We observe that the full equivariance constraint degrades the performance (see Tab. 1 and Tab. 4).

Table 10. Performance of full equivariant models

Task	**Dataset**	**ViT**	**DINOv2**	**ResN/Seg**
Class. (cAcc)	CIFAR-10	84.10	93.07	73.21†
	CIFAR-100	66.98	81.99	58.85†
Segmen. (cIoU)	Pascal VOC	63.45	68.87	29.20‡

†ResNet-50, ‡SegFormer

Table 11. Wang et al. [67] vs Ours.

Wang et al. [67]		**Ours**	
# Filter Bank	**cAcc (%)**	**Softness**	**cAcc (%)**
2	83.08	0.7	85.52
3	81.25	0.8	85.22
4	82.74	0.9	**85.11**

We also compare our method with the group convolution-based soft equivariant model proposed by Wang et al. [67] on the MNIST digits classification task. We follow the setup of Sec. 5.1 and report the results in Tab. 11. We observe that across different levels of softness, which is controlled by the number of filter banks in Wang et al. [67], our method maintains stable performance, outperforming the baseline.

We provide the full results on the human trajectory prediction task in Tab. 8. We observe that, across most datasets, our method outperforms both strict-equivariant and non-equivariant baselines on both standard metrics (ADE, FDE) and

Table 12. **Segmentation Performance on PASCAL VOC [15].** Results reported as mean $\pm$ std over multiple runs. eErr is in $\times 10^{-2}$.

Arch.	ViT [12]		DINOv2 [43]		Segformer [71]	
	cIoU↑	eErr↓	cIoU↑	eErr↓	cIoU↑	eErr↓
Base	72.99 ± 0.95	11 ± 01	87.97 ± 1.02	04 ± 00	63.67 ± 0.98	11 ± 00
Canon.	64.37 ± 0.75	19 ± 00	82.83 ± 0.18	38 ± 07	55.90 ± 1.26	24 ± 02
Ours	$\mathbf{73.25} \pm 0.38$	$\mathbf{10} \pm 00$	$\mathbf{88.61} \pm 0.18$	$\mathbf{03} \pm 00$	$\mathbf{63.87} \pm 0.67$	$\mathbf{10} \pm 00$

Table 13. **Performance on CIFAR10/100 across various backbones.** Results reported as mean $\pm$ std over multiple runs. iErr is in $\times 10^{-2}$.

	Arch.	ViT [12]		DINOv2 [43]		ResNet-50 [21]	
		cAcc↑	iErr↓	cAcc↑	iErr↓	cAcc↑	iErr↓
CIFAR10	Base	97.47 ± 0.05	06 ± 00	98.63 ± 0.06	04 ± 00	94.88 ± 0.17	21 ± 01
	Canon.	93.42 ± 0.22	13 ± 02	97.78 ± 0.22	$\mathbf{03} \pm 00$	91.00 ± 0.26	11 ± 01
	Ours	$\mathbf{97.49} \pm 0.14$	$\mathbf{06} \pm 00$	$\mathbf{98.70} \pm 0.05$	04 ± 00	$\mathbf{95.05} \pm 0.27$	$\mathbf{19} \pm 03$
CIFAR100	Base	85.63 ± 0.22	24 ± 00	91.62 ± 0.20	25 ± 01	79.78 ± 0.18	51 ± 02
	Canon.	78.41 ± 0.44	25 ± 02	88.64 ± 0.65	$\mathbf{16} \pm 01$	74.18 ± 1.02	$\mathbf{27} \pm 03$
	Ours	$\mathbf{85.67} \pm 0.28$	$\mathbf{23} \pm 01$	$\mathbf{91.90} \pm 0.12$	23 ± 00	$\mathbf{80.13} \pm 0.24$	45 ± 00

augmented metrics (aADE, aFDE). While 1 and 4 reports results for a single run, Tab. 13 and Tab. 12 report mean $\pm$ std over 5 random seeds following the exact setup of Sec. 5.2 and Sec. 5.3. We observe that our method consistently outperforms the baselines.

G. Experiment Details

G.1. Validating tunable softness level

We use a two-layer MLP trained with a batch size of 512 for 30 epochs. The MLPs take flattened images as input. The models are trained on the standard MNIST [33] training set with 10% held out for validation. We choose the best-performing model on the validation set for final evaluation on the test set. We use Adam optimizer with a learning rate of $1e^{-3}$ and a weight decay of $1e^{-4}$. The input images are padded to prevent the boundary artifact during rotation. We use the Adam optimizer with linear learning rate decay.

G.2. Image Classification

CIFAR-10 and CIFAR-100. We use the pretrained ViT-B/16 model from [12] as the backbone which is pretrained in ImageNet-1k dataset for classification task. We replace the final classification layer to match the number of classes in the CIFAR-10 and CIFAR-100 datasets. The ResNet-50 [21] backbone is also pretrained on ImageNet-1k, and only the final classification is replaced. The DINOv2 [43] is trained on the large-scale LVD-142M dataset. We add a single-layer classification head with the DINOv2 backbone.

For the Canon. [41] baseline we use a 3 layer C_{18} equivariant steerable (a 20° rotation equivariant) CNN architecture [9] as the canonicalization network. The model is trained from scratch with random initialization. We also use the prior loss as described in [41] to stabilize the canonicalization network. We use the same rotation group for designing our projection operators. See Sec. G.5 for details on the choice of the cutoff value for the projection operator.

All the models are finetuned for 30 epochs. Images are interpolated to 224×224 resolution to match the input size of the pretrained models. We use a larger learning rate of $1e^{-3}$ for the classification head and $1e^{-5}$ and $5e^{-5}$ for the ViT/DINOv2 and ResNet backbone, respectively. Models are trained with random rotation augmentation. We use the AdamW optimizer and a cosine learning rate scheduler. 20% of the training data is held out for validation. The best model on the validation set is used for final evaluation on the test set.

ImageNet-1k. We use publicly available pretrained ViT-B/16 [12], DINOv2-base [25, 43], and ResNet-50 [21] models. We directly evaluate the models on the ImageNet-1k validation set. We use the same architecture for the canonicalization module for the Canon. [41] baseline as described above. We use the training script from Pytorch Image Models (timm) library [68] for finetuning and evaluation. We train all the models on a single GPU with a batch size of 128 for 15 epochs with 4 warmup epochs. We use a cosine learning rate scheduler with a learning rate of $1e^{-4}$. We use standard data augmentation techniques from ImageNet training [12, 22], with mixup, cutmix, random erasing, and RandAugment. The hyperparameters are summarized in Tab. 14.

G.3. Semantic segmentation

We use the PASCAL VOC 2012 [15] dataset for the semantic segmentation task. Following the DINOv2 [43], we use a linear classifier over the pre-trained DINOv2-base and ViT-B/16 backbones. For SegFormer [71], we change the final classification layer of the decoder head to match the number of classes in the PASCAL VOC dataset. We use the same canonicalization network architecture as described above for the Canon. baseline. The SegFormer and DINOv2 models are trained with images of size 512×512 following their original setup. For ViT-B/16, we use an image size of 224×224. All the models are trained

Table 14. Training Hyperparameters for ImageNet-1k

Hyperparameter	**Value**
Optimization	
Optimizer	AdamW
Learning Rate	$1e-4$
Weight Decay	0.05-0.04
Epochs	15
Gradient Clipping	1.0
Learning Rate Schedule	
Scheduler	Cosine Annealing
Warmup Epochs	4
Warmup LR	$1e-6$
Layer-wise LR Decay	0.65
Augmentation	
Mixup α	0.8
CutMix α	1.0
Random Erasing	0.25
RandAugment	rand-m9-mstd0.5-inc1
Regularization	
Label Smoothing	0.1
Drop Path Rate	0.1
Model EMA	0.99

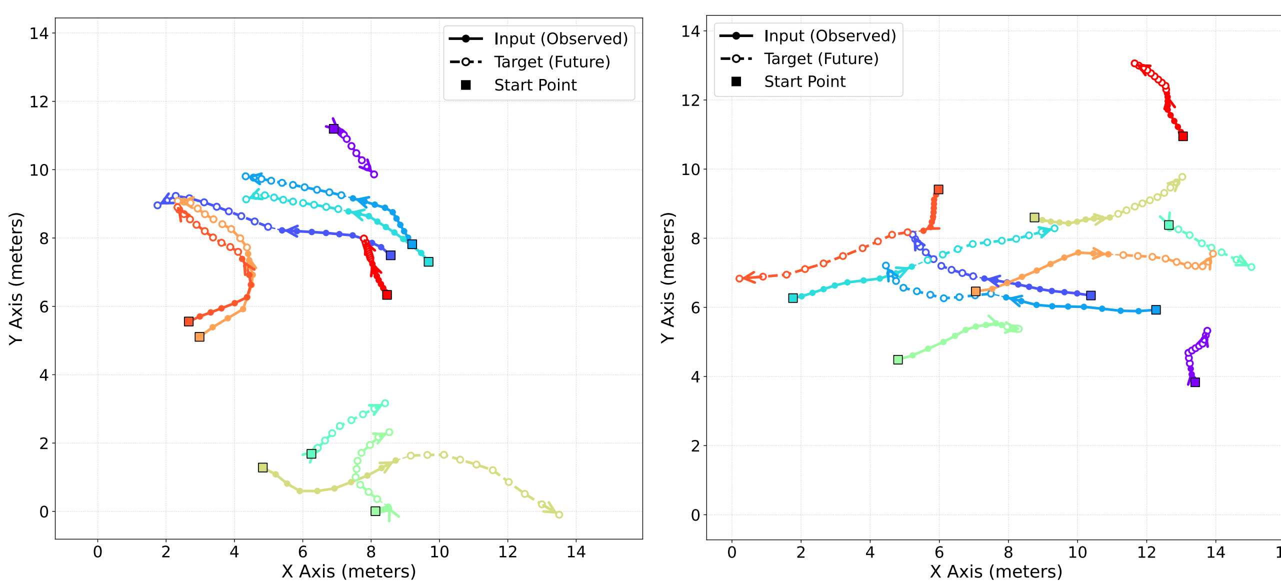
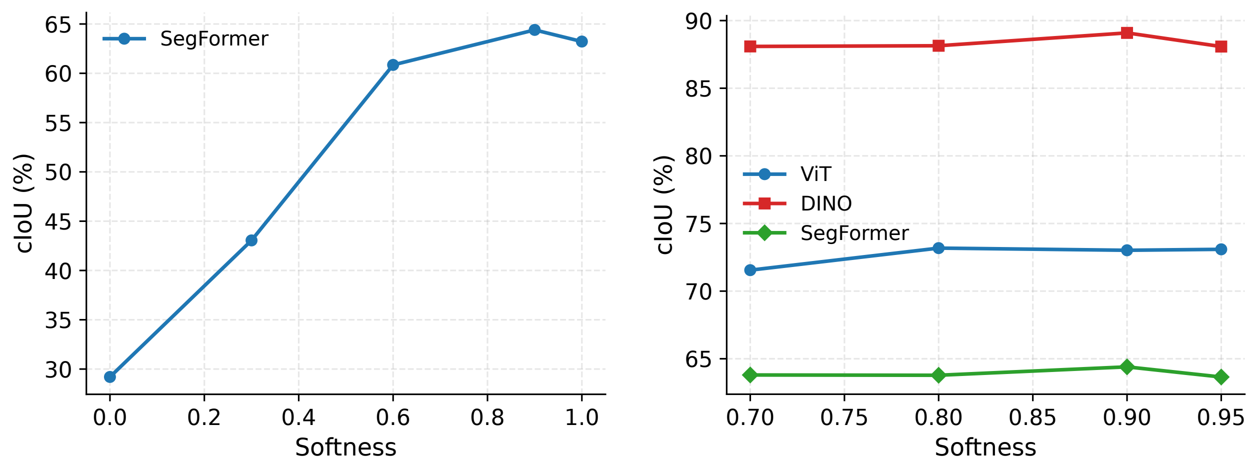


Figure 7. Sample trajectories from the ETH/UYC dataset. Solid circles indicate observed past positions, and hollow circles show the future. Square markers denote starting points. Each line with a distinct color corresponds to a different pedestrian in the scene.

for 100 epochs with a batch size of 32. We use a cosine annealing learning rate scheduler with 8 warmup epochs. The learning rate is set to $5e^{-5}$ for ViT-B/16, $1e^{-5}$ for DINOv2, and $1e^{-4}$ for SegFormer baselines. We use standard data augmentation techniques, including random scaling, cropping, horizontal flipping, and rotation.

(a) cIoU vs softness level for SegFormer (b) cIoU vs high softness level for different pretrained models



G.4. Human trajectory prediction

We use the human trajectory datasets from ETH [44] and UCY [34], consisting of five subsets with the number of subjects (humans) varying from $5 - 57$ across different scenes and datasets. We visualize two samples from the dataset in Fig. 7, illustrating the task and the dataset's complexity. We use an auto-regressive transformer model with 4 transformer layers, each with 4 attention heads. We follow the technique of Vector neurons [10] for designing an equivariant auto-regressive (EqAuto) transformer. Models are trained for 100 epochs with a learning rate of $5e^{-4}$ with a weight decay of $1e^{-5}$. The models are trained with mean squared error. The best-performing model on the validation set is used for the final evaluation on the test set. We use the Lie algebra representation of the $2D$ rotation group to design the projection operators. The combined ADE (cADE) and combined FDE (cFDE) are calculated as

$$\text{cADE} = \frac{\text{ADE} + \text{aADE}}{2}, \quad \text{cFDE} = \frac{\text{FDE} + \text{aFDE}}{2}, \tag{162}$$

G.5. Softness level selection

In Sec. 5.1, we show that when training from scratch, the level of 'softness' can be selected depending on the trade-off between performance and consistency.

However, when using pretrained models, smaller softness levels (i.e., more aggressive projection) can lead to a significant drop in performance because they deviate further from the pretrained weights. As we see in Fig. 8a, the performance of SegFormer drops significantly when the softness level is smaller than 0.6, and we do not see any benefit from pretraining. So, when adapting pretrained models, we use higher softness levels (> 0.7) to retain the benefits of pretraining. As we see in Fig. 8b, across all architectures, this range of softness levels leads to effective fine-tuning and better performance.

References

[1] 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. In *Proc. AAAI*, 2023. 2

[2] Erik J Bekkers, Maxime W Lafarge, Mitko Veta, Koen AJ Eppenhof, Josien PW Pluim, and Remco Duits. Roto-translation covariant convolutional networks for medical image analysis. In *Proc. MICCAI*, 2018. 2

[3] Gregory Benton, Marc Finzi, Pavel Izmailov, and Andrew G Wilson. Learning invariances in neural networks from training data. In *Proc. NeurIPS*, 2020. 1

[4] Georg Bökman, David Nordström, and Fredrik Kahl. Flopping for flops: Leveraging equivariance for computational efficiency. *arXiv preprint arXiv:2502.05169*, 2025. 2

[5] Anadi Chaman and Ivan Dokmanic. Truly shift-invariant convolutional neural networks. In *Proc. CVPR*, 2021. 2

[6] Siheng Chen, Aliaksei Sandryhaila, and Jelena Kovačević. Sampling theory for graph signals. In *Proc. ICASSP*, 2015. 2

[7] Siheng Chen, Rohan Varma, Aliaksei Sandryhaila, and Jelena Kovačević. Discrete signal processing on graphs: Sampling theory. *IEEE TSP*, 2015. 2

[8] Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International conference on machine learning*, pages 2990–2999. PMLR, 2016. 1, 2

[9] Taco S Cohen and Max Welling. Steerable CNNs. In *Proc. ICLR*, 2017. 2, 6, 27

[10] Congyue Deng, Or Litany, Yueqi Duan, Adrien Poulenard, Andrea Tagliasacchi, and Leonidas J Guibas. Vector neurons: A general framework for SO(3)-equivariant networks. In *Proc. ICCV*, 2021. 8, 26, 29

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proc. CVPR*, 2009. 2, 6

[12] 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 *Proc. ICLR*, 2021. 1, 2, 6, 7, 8, 9, 27

[13] Yuanqi Du, Limei Wang, Dieqiao Feng, Guifeng Wang, Shuiwang Ji, Carla P Gomes, Zhi-Ming Ma, et al. A new perspective on building efficient and expressive 3D equivariant graph neural networks. In *Proc. NeurIPS*, 2023. 2

[14] Ahmed A Elhag, T Konstantin Rusch, Francesco Di Giovanni, and Michael Bronstein. Relaxed equivariance via multitask learning. In *Proc. LOG*, 2025. 1, 2

[15] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The PASCAL visual object classes (VOC) challenge. *IJCV*, 2010. 2, 7, 8, 27

[16] Marc Finzi, Gregory Benton, and Andrew G Wilson. Residual pathway priors for soft equivariance constraints. In *Proc. NeurIPS*, 2021. 1, 2, 6, 26

[17] Marc Finzi, Max Welling, and Andrew Gordon Wilson. A practical method for constructing equivariant multilayer perceptrons for arbitrary matrix groups. In *Proc. ICML*, 2021. 2, 4, 26

[18] William Fulton and Joe Harris. *Representation theory: a first course*. Springer Science & Business Media, 2013. 5

[19] Francesco Giuliari, Irtiza Hasan, Marco Cristani, and Fabio Galasso. Transformer networks for trajectory forecasting. In *Proc. ICPR*, 2021. 2, 8, 26

[20] Jason Hartford, Devon Graham, Kevin Leyton-Brown, and Siamak Ravanbakhsh. Deep models of interactions across sets. In *Proc. ICML*, 2018. 2

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. CVPR*, 2016. 2, 6, 7, 27

[22] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proc. CVPR*, 2022. 27

[23] Elyssa Hofgard, Rui Wang, Robin Walters, and Tess Smidt. Relaxed equivariant graph neural networks. *arXiv preprint arXiv:2407.20471*, 2024. 2

[24] Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012. 4, 5, 12

[25] Hugging Face Model Hub. facebook/dinov2-base-imagenet1k-1-layer. `https://huggingface.co/facebook/dinov2-base-imagenet1k-1-layer`, 2025. Accessed: 2025-11-09. 6, 27

[26] Royina Karegoudra Jayanth, Yinshuang Xu, Ziyun Wang, Evangelos Chatzipantazis, Daniel Gehrig, and Kostas Daniilidis. Eqnio: Subequivariant neural inertial odometry. *arXiv preprint arXiv:2408.06321*, 2024. 2

[27] Sékou-Oumar Kaba, Arnab Kumar Mondal, Yan Zhang, Yoshua Bengio, and Siamak Ravanbakhsh. Equivariance with learned canonicalization functions. In *Proc. ICML*, 2023. 2, 6

[28] Hyunsu Kim, Hyungi Lee, Hongseok Yang, and Juho Lee. Regularizing towards soft equivariance under mixed symmetries. In *Proc. ICML*, 2023. 1, 2

[29] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *Proc. ICLR*, 2017. 1

[30] Wilhelm Klingenberg. *Riemannian geometry*. Walter de Gruyter, 1995. 18

[31] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 2, 6
[32] Soumyabrata Kundu and Risi Kondor. Steerable transformers for volumetric data. *arXiv preprint arXiv:2405.15932*, 2024. 2
[33] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 2002. 6, 27
[34] Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. Crowds by example. In *Computer graphics forum*, 2007. 7, 29
[35] Yi-Lun Liao and Tess Smidt. Equiformer: Equivariant graph attention transformer for 3d atomistic graphs. In *Proc. ICLR*, 2023. 2
[36] Iou-Jen Liu, Raymond A Yeh, and Alexander G Schwing. PIC: permutation invariant critic for multi-agent deep reinforcement learning. In *Proc. CORL*, 2020. 2
[37] Andrei Manolache, Luiz FO Chamon, and Mathias Niepert. Learning (approximately) equivariant networks via constrained optimization. *arXiv preprint arXiv:2505.13631*, 2025. 3
[38] Dimitris G Manolakis and Vinay K Ingle. *Applied digital signal processing: theory and practice*. Cambridge university press, 2011. 13
[39] Haggai Maron, Heli Ben-Hamu, Nadav Shamir, and Yaron Lipman. Invariant and equivariant graph networks. In *Proc. ICLR*, 2019. 2
[40] Daniel McNeela. Almost equivariance via lie algebra convolutions. *arXiv preprint arXiv:2310.13164*, 2023. 2, 3
[41] Arnab Kumar Mondal, Siba Smarak Panigrahi, Oumar Kaba, Sai Rajeswar Mudumba, and Siamak Ravanbakhsh. Equivariant adaptation of large pretrained models. In *Proc. NeurIPS*, 2023. 6, 7, 27
[42] Christopher Morris, Gaurav Rattan, Sandra Kiefer, and Siamak Ravanbakhsh. SpeqNets: Sparsity-aware permutation-equivariant graph networks. In *Proc. ICML*, 2022. 2
[43] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 2, 6, 7, 8, 27
[44] Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *Proc. ICCV*, 2009. 7, 29
[45] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proc. CVPR*, 2017. 1
[46] Md Ashiqur Rahman and Raymond A Yeh. Truly scale-equivariant deep nets with Fourier layers. In *Proc. NeurIPS*, 2024. 1
[47] Md Ashiqur Rahman and Raymond A Yeh. Group downsampling with equivariant anti-aliasing. In *Proc. ICLR*, 2025. 2
[48] Md Ashiqur Rahman, Robert Joseph George, Mogab Elleithy, Daniel Leibovici, Zongyi Li, Boris Bonev, Colin White, Julius Berner, Raymond A. Yeh, Jean Kossaifi, Kamyar Azizzadenesheli, and Anima Anandkumar. Pretraining codomain attention neural operators for solving multiphysics PDEs. In *Proc. NeurIPS*, 2024. 2
[49] Md Ashiqur Rahman, Chiao-An Yang, Michael N Cheng, Lim Jun Hao, Jeremiah Jiang, Teck-Yian Lim, and Raymond A Yeh. Local scale equivariance with latent deep equilibrium canonicalizer. In *Proc. ICCV*, 2025. 1
[50] Siamak Ravanbakhsh, Jeff Schneider, and Barnabas Poczos. Deep learning with sets and point clouds. In *Proc. ICLR workshop*, 2017. 2
[51] Renan A Rojas-Gomez, Teck-Yian Lim, Alex Schwing, Minh Do, and Raymond A Yeh. Learnable polyphase sampling for shift invariant and equivariant convolutional networks. In *Proc. NeurIPS*, 2022. 2
[52] Renan A Rojas-Gomez, Teck-Yian Lim, Minh N Do, and Raymond A Yeh. Making vision transformers truly shift-equivariant. In *Proc. CVPR*, 2024. 2
[53] David W Romero and Suhas Lohit. Learning partial equivariances from data. In *Proc. NeurIPS*, 2022. 1, 2
[54] Ashwin Samudre, Mircea Petrache, Brian D Nord, and Shubhendu Trivedi. Symmetry-based structured matrices for efficient approximately equivariant networks. *arXiv preprint arXiv:2409.11772*, 2024. 2
[55] Arne Schneuing, Charles Harris, Yuanqi Du, Kieran Didi, Arian Jamasb, Ilia Igashov, Weitao Du, Carla Gomes, Tom L Blundell, Pietro Lio, et al. Structure-based drug design with equivariant diffusion models. *Nature Computational Science*, 2024. 1
[56] Patrice Simard, Bernard Victorri, Yann LeCun, and John Denker. Tangent prop-a formalism for specifying selected invariances in an adaptive network. In *Proc. NeurIPS*, 1991. 1
[57] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proc. ICLR*, 2015. 1
[58] Ivan Sosnovik, Michał Szmaja, and Arnold Smeulders. Scale-equivariant steerable networks. In *Proc. ICLR*, 2020. 2
[59] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proc. CVPR*, 2015. 1
[60] Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint arXiv:1802.08219*, 2018. 2
[61] Tycho van der Ouderaa, David W Romero, and Mark van der Wilk. Relaxing equivariance constraints with non-stationary continuous filters. In *Proc. NeurIPS*, 2022. 1, 2
[62] Tycho van der Ouderaa, Alexander Immer, and Mark van der Wilk. Learning layer-wise equivariances automatically using gradients. In *Proc. NeurIPS*, 2023. 1, 2
[63] Elise Van der Pol, Daniel Worrall, Herke van Hoof, Frans Oliehoek, and Max Welling. MDP homomorphic networks: Group symmetries in reinforcement learning. In *Proc. NeurIPS*, 2020. 1

[64] Lars Veefkind and Gabriele Cesa. A probabilistic approach to learning the degree of equivariance in steerable CNNs. *arXiv preprint arXiv:2406.03946*, 2024. 2

[65] Martin Vetterli, Jelena Kovačević, and Vivek K Goyal. *Foundations of signal processing*. Cambridge University Press, 2014. 2, 13

[66] Dian Wang, Xupeng Zhu, Jung Yeon Park, Mingxi Jia, Guanang Su, Robert Platt, and Robin Walters. A general theory of correct, incorrect, and extrinsic equivariance. 2023. 2

[67] Rui Wang, Robin Walters, and Rose Yu. Approximately equivariant networks for imperfectly symmetric dynamics. In *Proc. ICML*, 2022. 2, 26

[68] Ross Wightman. Pytorch image models. `https://github.com/huggingface/pytorch-image-models`, 2019. 6, 27

[69] Daniel Worrall and Max Welling. Deep scale-spaces: Equivariance over scale. In *Proc. NeurIPS*, 2019. 1

[70] Zhiqiang Wu, Yingjie Liu, Hanlin Dong, Xuan Tang, Jian Yang, Bo Jin, Mingsong Chen, and Xian Wei. R2det: Exploring relaxed rotation equivariance in 2d object detection. *arXiv preprint arXiv:2408.11760*, 2024. 2

[71] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. SegFormer: Simple and efficient design for semantic segmentation with transformers. In *Proc. NeurIPS*, 2021. 2, 7, 8, 27

[72] Raymond A Yeh, Yuan-Ting Hu, and Alexander Schwing. Chirality nets for human pose regression. In *Proc. NeurIPS*, 2019. 2

[73] Raymond A Yeh, Alexander G Schwing, Jonathan Huang, and Kevin Murphy. Diverse generation for multi-agent sports games. In *Proc. CVPR*, 2019. 2

[74] Raymond A Yeh, Yuan-Ting Hu, Mark Hasegawa-Johnson, and Alexander Schwing. Equivariance discovery by learned parameter-sharing. In *Proc. AISTATS*, 2022. 2

[75] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan R Salakhutdinov, and Alexander J Smola. Deep sets. In *Proc. NeurIPS*, 2017. 2

[76] Richard Zhang. Making convolutional networks shift-invariant again. In *Proc. ICML*, 2019. 1, 2, 3

[77] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ADE20K dataset. *IJCV*, 2019. 7

[78] Yifan Zhou, Zeqi Xiao, Shuai Yang, and Xingang Pan. Alias-free latent diffusion models: Improving fractional shift equivariance of diffusion latent space. In *Proc. CVPR*, 2025. 2