# 3 Group Structures in Deep Neural Network

In this section, we apply group theory to analyze deep neural networks. Our central insight is that deep learning can be understood through the lens of symmetry—specifically, how networks build, break, and preserve symmetries to achieve their objectives. In Section 3.1, we prove that a deep linear network, which is the most primitive network, possesses the most universal group structure. As we add non-linear layers and regularization terms, we genuinely constrain group structure (Section 3.2). This symmetry-based framework reveals deep learning as a principled trade-off: networks must break sufficient symmetry to achieve expressivity while preserving appropriate structural constraints for generalization. After that, we study the symmetric structure in auto-encoder regularization in Section 3.3. By analyzing neural networks through their group-theoretic properties, we can gain a deeper understanding of why specific architectural choices succeed and develop principles for designing more effective architectures.

## 3.1 Deep Linear Networks

We begin by studying the simplest class of neural networks: linear networks. Despite their simplicity, linear networks exhibit a rich symmetry structure that illuminates fundamental aspects of overparameterization. We establish that the parameter space of a linear network is invariant by $\mathrm{GL}_n(\mathbb{R})$ (Definition 2.9) on each hidden layer $i$, leading to equivalence classes of parameters that realize identical input-output mappings.

**Definition 3.1** (Parameter Space and Linear Network)**.** *Consider an $L$-layer linear network with layer widths $n_0, n_1, \ldots, n_L \in \mathbb{N}$. The parameter space of this network is the space of all weight matrices:*

$$\mathcal{P} = \mathbb{R}^{n_L \times n_{L-1}} \times \mathbb{R}^{n_{L-1} \times n_{L-2}} \times \cdots \times \mathbb{R}^{n_1 \times n_0}.$$

*A point in the parameter space is represented as:*

$$W = (W_L, \ldots, W_1),$$

*where $W_i \in \mathbb{R}^{n_i \times n_{i-1}}$.*
*The linear network is the composed linear map:*

$$\Phi(W) = W_L W_{L-1} \cdots W_1 \in \mathbb{R}^{n_L \times n_0}.$$

**Remark 3.2** (Hypothesis Space Interpretation)**.** .
*In machine learning, a hypothesis space $\mathcal{H}$ is the set of all functions realizable by a model architecture. For the linear network above, we distinguish:*

- *Parameter space $\mathcal{P}$ : the domain of optimization algorithms (e.g., gradient descent operates here)*
- *Hypothesis space $\mathcal{H} = \mathrm{Im}(\Phi) \subset \mathbb{R}^{n_L \times n_0}$ : the set of input-output maps the network can express*

*The map $\Phi : \mathcal{P} \to \mathcal{H}$ sends each parameter configuration to its realized function. For linear networks, every hypothesis is a linear map $\mathbb{R}^{n_0} \to \mathbb{R}^{n_L}$, and the hypothesis space is*

$$\mathcal{H} = \left\{ W_L W_{L-1} \cdots W_1 : W_i \in \mathbb{R}^{n_i \times n_{i-1}} \right\} .$$

*When all layers are wide (i.e., $n_i \geq \min(n_0, n_L)$ for $i = 1, \ldots, L-1$ ), we have $\mathcal{H} = \mathbb{R}^{n_L \times n_0}$. Otherwise, $\mathcal{H}$ is the algebraic variety of matrices with rank at most $\min_i n_i$.*

*Note that the map $\Phi$ is generically non-injective, distinct parameter configurations can realize identical hypotheses.*

**Proposition 3.3** (Layer Group Structure)**.** *Let $\mathcal{G}_i \subseteq \mathrm{GL}_{n_i}(\mathbb{R})$ be subgroups for $i = 1, \ldots, L-1$. Then the direct product*

$$\prod_{i=1}^{L-1} \mathcal{G}_i = \mathcal{G}_1 \times \mathcal{G}_2 \times \cdots \times \mathcal{G}_{L-1} \tag{3.1}$$

*is a group under component-wise matrix multiplication.*

*Proof.* Denote

$$G = \prod_{i=1}^{L-1} \mathcal{G}_i$$

from Equation 3.1, we verify the four group axioms:

- Closure: Let $K = (K_1, \ldots, K_{L-1})$ and $K' = \left(K'_1, \ldots, K'_{L-1}\right)$ be elements of $G$ . Define their product component-wise:

  $$K \cdot K' := \left(K_1 K'_1, K_2 K'_2, \ldots, K_{L-1} K'_{L-1}\right) .$$

  Since each $\mathcal{G}_i$ is a subgroup of $\mathrm{GL}_{n_i}(\mathbb{R})$, we have $K_i, K'_i \in \mathcal{G}_i$, which implies $K_i K'_i \in \mathcal{G}_i$ by closure in $\mathcal{G}_i$. Therefore, $K \cdot K' \in G$

- Associativity: For $K, K', K'' \in G$, we have:

$$\begin{aligned}
(K \cdot K') \cdot K" &= (K_1 K'_1, \ldots, K_{L-1} K'_{L-1}) \cdot (K"_1, \ldots, K"_{L-1}) \\
&= ((K_1 K'_1) K"_1, \ldots, (K_{L-1} K'_{L-1}) K"_{L-1}) \\
&= (K_1 (K'_1 K"_1), \ldots, K_{L-1} (K'_{L-1} K"_{L-1})) \quad \text{(by associativity)} \\
&= (K_1, \ldots, K_{L-1}) \cdot (K'_1 K"_1, \ldots, K'_{L-1} K"_{L-1}) \\
&= K \cdot (K' \cdot K").
\end{aligned}$$

- Identity element:

  Let $I_i \in \mathcal{G}_i$ denote the identity matrix in $\mathrm{GL}_{n_i}(\mathbb{R})$ for each $i$.

  Define:

  $$e_G := (I_1, I_2, \ldots, I_{L-1}) \in G.$$

  For any $K = (K_1, \ldots, K_{L-1}) \in G$ :

  $$\begin{aligned} e_G \cdot K &= (I_1, \ldots, I_{L-1}) \cdot (K_1, \ldots, K_{L-1}) \\ &= (I_1 K_1, \ldots, I_{L-1} K_{L-1}) \\ &= (K_1, \ldots, K_{L-1}) = K, \end{aligned}$$

  and similarly $K \cdot e_G = K$.

- Inverse element:

  For any $K = (K_1, \ldots, K_{L-1}) \in G$, since each $K_i \in \mathcal{G}_i \subseteq \mathrm{GL}_{n_i}(\mathbb{R})$ is invertible, $K_i^{-1}$ exists and $K_i^{-1} \in \mathcal{G}_i$ (since $\mathcal{G}_i$ is a subgroup).

  Define:

  $$K^{-1} := \left(K_1^{-1}, K_2^{-1}, \ldots, K_{L-1}^{-1}\right) \in G.$$

  Then:

  $$\begin{aligned} K \cdot K^{-1} &= (K_1, \ldots, K_{L-1}) \cdot (K_1^{-1}, \ldots, K_{L-1}^{-1}) \\ &= (K_1 K_1^{-1}, \ldots, K_{L-1} K_{L-1}^{-1}) \\ &= (I_1, \ldots, I_{L-1}) = e_G, \end{aligned}$$

  and similarly $K^{-1} \cdot K = e_G$.

  Since all four group axioms are satisfied, $G$ is a group.

□

Using Proposition 3.3, which states that $\prod_{i=1}^{L-1} \mathcal{G}_i$ is a group, we define the group action (Definition 2.7) on network layers.

**Definition 3.4** (Layer Group and Layer Group Action)**.** *Let $\mathcal{G}_i \subseteq \mathrm{GL}_{n_i}(\mathbb{R})$ denote a subgroup of invertible $n_i \times n_i$ matrices for each hidden layer $i = 1, \ldots, L-1$.*

*Define The layer group:*

$$G = \prod_{i=1}^{L-1} \mathcal{G}_i. \tag{3.2}$$

*For $K = (K_{L-1}, \dots, K_1) \in G$, the layer group action $\alpha : G \times \mathcal{P} \to \mathcal{P}$ is defined by*

$$K \cdot W := \left(W_L K_{L-1}^{-1}, K_{L-1} W_{L-1} K_{L-2}^{-1}, \dots, K_1 W_1\right).$$

**Proposition 3.5.** *The action defined above is a (left) group action (Definition 2.7).*

*Proof.* Let $e$ be the identity element, $H, K \subset G$ as subgroups.

Check identity property: $e \cdot W = W$.

Check compatibility property: $(KH) \cdot W = K \cdot (H \cdot W)$ for $K, H \in G$ because matrix multiplications are associative.

□

**Proposition 3.6.** *In the Definition 3.4 , the linear group*

$$G = \prod_{i=1}^{L-1} \mathcal{G}_i$$

*preserves the linear map $\Phi(W)$ . Specifically, the map $G \times \mathcal{P} \to \mathcal{P}$ defined by $(K, W) \mapsto K \cdot W$ constitutes a left group action (Definition 3.4), and*

$$\Phi(K \cdot W) = \Phi(W) \quad \forall K \in G.$$

*The linear network is invariant under the* $\mathrm{GL}_{n_i}(\mathbb{R})$ *group.*

*Proof.* Using linear network $\Phi(\cdot)$ and $K = (K_{L-1}, \dots, K_1) \in G$ we defined before,

$$\Phi(K \cdot W) = \left(W_L K_{L-1}^{-1}\right)\left(K_{L-1} W_{L-1} K_{L-2}^{-1}\right) \cdots (K_1 W_1) = W_L W_{L-1} \cdots W_1 = \Phi(W).$$

□

## 3.2 Symmetry Reduction in Nonlinear Activation

Composing multiple linear networks yields another linear network; we show that deep linear networks exhibit $\mathrm{GL}_n(\mathbb{R})$ group, which is a large symmetry. However, this large symmetry corresponds to weak expressivity, as they can only represent linear transformations. Modern neural networks achieve universal approximation capabilities [33] by introducing nonlinear activation functions $\sigma : \mathbb{R} \to \mathbb{R}$ applied element-wise to hidden layers [48]. We argue that nonlinear activations generally reduce symmetries under linear transformations, i.e., the group equipped in activation is a proper subgroup of $\mathrm{GL}_n(\mathbb{R})$, often dramatically smaller than the full linear group. We also analyze the symmetry groups preserved by two widely-used activation functions: ReLU and sigmoid.

Consider a network with nonlinear activations $\sigma$:

$$\Phi_\sigma(W) = W_L \sigma\left(W_{L-1} \sigma\left(\cdots \sigma W_1\right)\right),$$

where $\sigma$ acts as a function composition on matrix $W_i$, $(\sigma \circ W)x = \sigma(Wx)$, applying $\sigma$ element-wise to the layer $Wx$.

Apply a layer group action (Definition 3.4) $K \in G$ (Equation 3.2):

$$\Phi_\sigma(K \cdot W) = W_L K_{L-1}^{-1} \sigma(K_{L-1} W_{L-1}(\cdots)).$$

To be invariant, we need:

$$\Phi_\sigma(K \cdot W) = \Phi_\sigma(W).$$

Unpack this equation:

$$\sigma\left(h\right) = K_i^{-1} \sigma\left(K_i h\right),$$

where $h$ is a consecutive sub-layer in the network, e.g., $h = W_i \sigma(W_{i-1}\sigma(\cdots \sigma W_1))$. This formula is equivalent to:

$$\sigma\left(K_i h\right) = K_i \sigma(h).$$

**Definition 3.7** (Residual Symmetry Group)**.** *Consider a neural network with activation function $\sigma : \mathbb{R}^n \to \mathbb{R}^n$ applied element-wise to hidden layers of dimension n. The residual symmetry group $\mathcal{G}_\sigma$ is the subgroup of* $\mathrm{GL}_n(\mathbb{R})$ *consisting of all invertible transformations that commute with the activation function:*

$$\mathcal{G}_\sigma = \left\{K \in \mathrm{GL}_n(\mathbb{R}) : \sigma(Kh) = K\sigma(h), \forall h \in \mathbb{R}^n\right\}.$$

**Proposition 3.8** (Residual Symmetry Group is a Subgroup)**.** *For any activation function $\sigma : \mathbb{R}^n \to \mathbb{R}^n$, the set $\mathcal{G}_\sigma$ is a subgroup of* $\mathrm{GL}_n(\mathbb{R})$.

*Proof.* We verify the subgroup criteria.

1. Identity: Since $\sigma(Ih) = \sigma(h) = I\sigma(h)$ for all $h \in \mathbb{R}^n$, we have $I \in \mathcal{G}_\sigma$.

2. Closure:Let $K_1, K_2 \in \mathcal{G}_\sigma$. Then for all $h \in \mathbb{R}^n$ :

$$\sigma\left(K_1 K_2 h\right) = K_1 \sigma\left(K_2 h\right) = K_1 K_2 \sigma(h),$$

3. Inverses: Let $K \in \mathcal{G}_\sigma$. For arbitrary $h \in \mathbb{R}^n$, set $h' = Kh$. Then:

$$\sigma\left(h'\right) = K\sigma\left(K^{-1}h'\right) \implies K^{-1}\sigma\left(h'\right) = \sigma\left(K^{-1}h'\right).$$

Since $h'$ ranges over all of $\mathbb{R}^n$, we have $K^{-1} \in \mathcal{G}_\sigma$.

□

This reveals that nonlinear activations eliminate $\mathrm{GL}_n(\mathbb{R})$ group to a proper subgroup. The specific subgroup depends on the activation function's properties. Crucially, this symmetry reduction breaks the closure property of linear transformations, thereby enabling the Universal Approximation Theorem [33].

### 3.2.1 Example 1: ReLU Activation

The ReLU activation is defined component-wise as $\mathrm{ReLU}(x) = \max(0, x)$. According to Definition 3.7, the residual group that preserves ReLU's action is:

$$\mathcal{G}_{ReLU} = \{K \in \mathrm{GL}_n(\mathbb{R}) : \mathrm{ReLU}(Kh) = K \cdot \mathrm{ReLU}(h), \forall h \in \mathbb{R}^n\}.$$

**Corollary 3.9.** *The residual group of ReLU is constructed by the semi-direct product (Definition 2.17):*

$$\mathcal{G}_{ReLU} = \mathcal{D}_n^+ \rtimes S_n \cong \mathcal{M}_n^+$$

*where $\mathcal{M}_n^+$ denotes the monomial matrices with positive entries (Definition 2.15), $\mathcal{D}_n^+$ denotes the positive diagonal matrix (Definition 2.14), $S_n$ is the permutation matrix (Definition 2.11).*

*Proof.* (Sketch) Because ReLU is component-wise and only zero out negative components, $\mathcal{D}_n^+$ and $S_n$ are both subgroup of $\mathcal{G}_{ReLU}$, but they do not commute. Also $\mathcal{D}_n^+ \cap S_n = \{e\} = I_n$. There are no other non-trivial subgroups.

Choose the homomorphism:

$$\varphi : S_n \longrightarrow \mathrm{Aut}\left(\mathcal{D}_n^+\right)$$

as conjugation:

$$\varphi(\sigma)(D) = P_\sigma D P_\sigma^{-1}.$$

where $P_\sigma$ is the permutation matrix corresponding to $\sigma$. This equation holds because permuting a diagonal matrix and then permuting it back still yields a diagonal matrix.

Thus we can define semi-direct product $\mathcal{G}_{ReLU} = \mathcal{D}_n^+ \rtimes S_n \cong \mathcal{M}_n^+$. In this case, $\mathcal{D}_n^+$ is a normal subgroup.

The semi-direct product $\mathcal{D}_n^+ \rtimes S_n$ consists of pairs $(d, \sigma)$ with multiplication:

$$(d_1, \sigma_1) \cdot (d_2, \sigma_2) = (d_1 \varphi(\sigma_1)(d_2), \sigma_1\sigma_2).$$

The representation $(d, \sigma) \mapsto DP_\sigma$ gives

$$\begin{aligned}(D_1 P_{\sigma_1})(D_2 P_{\sigma_2}) &= D_1 P_{\sigma_1} D_2 P_{\sigma_2} \\ &= D_1 \left(P_{\sigma_1} D_2 P_{\sigma_1}^{-1}\right) P_{\sigma_1} P_{\sigma_2} \\ &= D_1 (\varphi(\sigma_1)(D_2)) P_{\sigma_1} P_{\sigma_2}.\end{aligned}$$

Thus

$$\mathcal{G}_{ReLU} = \mathcal{D}_n^+ \rtimes S_n = \left\{DP : D \in \mathcal{D}_n^+, P \in S_n\right\}.$$

□

From a geometric perspective, ReLU networks preserve two types of symmetry:

1. Permutation symmetry: Weights within a layer can be reordered with corresponding weight adjustments.

2. Positive scaling symmetry: Weights can be scaled by $\alpha > 0$ with inverse scaling $1/\alpha$ in the subsequent layer.

However, ReLU breaks the flip symmetry (sign changes) and rotation symmetry present in the full $\mathrm{GL}_n(\mathbb{R})$ group.

### 3.2.2 Example 2: Sigmoid Activation

The sigmoid activation is defined component-wise as $\mathrm{sigmoid}(x) = \frac{1}{1+e^{-x}}$. The residual group (Definition 3.7) that preserves the sigmoid's action is:

$$\mathcal{G}_{sigmoid} = \{K \in \mathrm{GL}_n(\mathbb{R}) : \mathrm{sigmoid}(Kh) = K\,\mathrm{sigmoid}(h), \forall h \in \mathbb{R}^n\}\,.$$

**Conjecture 3.10.** *The residual group of the sigmoid consists only of permutations (Definition 2.11):*

$$\mathcal{G}_{sigmoid} = S_n.$$

*Explanation:* Since sigmoid acts component-wise, permuting coordinates before or after sigmoid does not matter. So $P \in \mathcal{G}_{sigmoid}$ for all permutation matrices $P$. Consider the linear case $\mathrm{linear}(x) = kx$, its residual $\mathrm{Cent}(\mathrm{linear}) = \mathrm{GL}_n(\mathbb{R})$. But $\mathrm{sigmoid}(x) = \frac{1}{1+e^{-x}}$ is very non-linear and only intersect $\mathrm{linear}(x)$ at isolated points. For residual, we need:

$$\mathrm{sigmoid}(Kh) = K\,\mathrm{sigmoid}(h).$$

Any small perturbation (except between the isolated intersection points) can make LHS not equal to RHS because of nonlinearity. The only safe operations are permutations that do not change values.

Comparing sigmoid with ReLU reveals a structural difference: Its bounded output range [0, 1] breaks scaling symmetry, thus has fewer symmetry constraints than ReLU networks, potentially requiring more parameters to achieve the same expressivity.

This symmetry analysis explains why batch normalization is particularly beneficial for sigmoid networks [28]: It reintroduces scaling flexibility that sigmoid inherently lacks, effectively expanding the learnable parameter space.

**Conjecture 3.11** (Impossibility of Translation Symmetry via Regularization)**.** *Let $\mathcal{G}_{reg} \subseteq \mathrm{GL}_n(\mathbb{R})$ be any residual symmetry group induced by regularization on a fully-connected neural network layer of dimension $n$. Then $\mathcal{G}_{reg}$ cannot realize discrete translation symmetry (which is the symmetry endowed by CNN [34]).*

*More precisely, there exists no subgroup $\mathcal{G}_{reg} \subseteq \mathrm{GL}_n(\mathbb{R})$ and no representation $\rho : (\mathbb{Z}^d, +) \to \mathcal{G}_{reg}$ that faithfully represents the discrete translation group $(\mathbb{Z}^d, +)$ acting on spatial coordinates.*

## 3.3 Symmetry Reduction in Auto-encoder

In this section, we demonstrate that the group action framework (Definition 2.7) can be extended to regularization, but the regularization acts globally rather than layer-wise. To align the group action in both the activation layer and regularization, we introduce the autoencoder. We examine how different regularization schemes determine the residual symmetry groups (Definition 3.7) of the learned representations.

### 3.3.1 Introduce to Auto-encoder

An auto-encoder is a type of artificial neural network designed to learn efficient data representations in an unsupervised manner [3].

**Definition 3.12** (Encoder and Decoder)**.** *Let $\mathcal{X} \subset \mathbb{R}^D$ be the input space, $\mathcal{Z} \subset \mathbb{R}^d$ be the latent space, $\mathcal{Y} \subset \mathbb{R}^D$ be the output space. An encoder is the map $f : \mathcal{X} \to \mathcal{Z}$; decoder is the map $g : \mathcal{Z} \to \mathcal{Y}$; reconstruction map is $r(x) = (g \circ f)(x)$.*

**Definition 3.13** (Autoencoder Architecture)**.** *An autoencoder consists of an encoder $f : \mathcal{X} \to \mathcal{Z}$ and decoder $g : \mathcal{Z} \to \mathcal{X}$, where $\mathcal{X} \subset \mathbb{R}^D$ is the data space and $\mathcal{Z} \subset \mathbb{R}^d$ is the latent space with $d \ll D$.*

*In general, both $f$ and $g$ are deep neural networks with multiple layers. For the theoretical analysis of group actions and symmetries, we focus on the single-layer case:*

$$f(x) = \sigma_f\left(Wx + b_z\right)$$
$$g(z) = \sigma_g\left(W'z + b_y\right).$$

*where $\sigma_f$ and $\sigma_g$ denotes the activation functions, $W \in \mathbb{R}^{d\times D}$, $W' \in \mathbb{R}^{D\times d}$ are the weight matrices, and $b_z \in \mathbb{R}^d$ and $b_y \in \mathbb{R}^D$ are the bias vectors, for encoder and decoder respectively.*

The training objective for auto-encoders involves finding parameters $\theta = \left\{W, W', b_z, b_y\right\}$ that minimize the reconstruction error over a training set $X \subset \mathbb{R}^D$ :

$$\mathcal{L}_{\mathrm{AE}}(\theta) = \frac{1}{|X|}\sum_{x\in X} L(x, g(f(x)))$$

where $L$ represents the reconstruction loss function, for example, MSE loss.

Similar to network structure, we define group action on the auto-encoder structure.

**Definition 3.14** (Group Action on Auto-encoders)**.** *Let $(G, \cdot)$ be a group with a group action $\rho : G \times \mathcal{Z} \to \mathcal{Z}$ on the latent space $\mathcal{Z}$, satisfying:*

1. *Identity: $\rho(e, z) = z$ for all $z \in \mathcal{Z}$.*

2. *Compatibility: $\rho\left(k_1, \rho\left(k_2, h\right)\right) = \rho\left(k_1 k_2, h\right)$ for all $k_1, k_2 \in G, h \in \mathcal{Z}$.*

*We denote $\rho(g, z) = g \cdot z$ for convenience.*
*The group $k \in G$ acts on autoencoder $(f, g)$ is defined as*

$$f_k(x) = k \cdot f(x), \quad \forall x \in \mathcal{X}$$
$$g_k(z) = g\left(k^{-1} \cdot z\right), \quad \forall z \in \mathcal{Z}.$$

### 3.3.2 Linear Auto-encoders

Consider a linear auto-encoder (LAE) where $\sigma_f = \sigma_g = \mathbf{id}$ and $b_z = b_y = 0$, yielding the reconstruction $g(f(x)) = W'Wx$. In the absence of regularization, we can insert any invertible transformation $K \in \mathrm{GL}_d(\mathbb{R})$ into the latent space and compensate in the decoder:

$$f'(x) = Kf(x) \tag{3.3}$$
$$g'(z) = g(K^{-1}z). \tag{3.4}$$

The reconstruction remains invariant under this transformation:

$$g'(f'(x)) = g(K^{-1}Kf(x)) = g(f(x)).$$

This invariance reveals that the transformation group $\mathrm{GL}_d(\mathbb{R})$ acts symmetrically on a linear loss function. This result is compatible with Section 3.1.

### 3.3.3 $L_2$ Regularization

Introducing $L_2$ regularization on both encoder and decoder weights:

$$\mathcal{L}(\theta) = \frac{1}{|X|} \sum_{x \in X} \left( L(x, g(f(x))) + \lambda \left( \|W\|_F^2 + \|W'\|_F^2 \right) \right),$$

breaks the full $\mathrm{GL}_d(\mathbb{R})$ symmetry, reducing it to the orthogonal group $O_d(\mathbb{R})$ (Definition 2.12). This symmetry reduction is formalized by the Transpose Theorem [35], which establishes that all critical points of the $L_2$-regularized linear auto-encoder satisfy $W' = W^T$.

### 3.3.4 Jacobian-based Regularization

We generalize $L_2$ regularization to any Jacobian-based regularization:

$$\mathcal{L}(f, g) = \frac{1}{|X|} \sum_{x \in X} \left[ L(x, g(f(x))) + \lambda_E \|J_f(x)\|_E + \lambda_D \|J_g(f(x))\|_D \right] \tag{3.5}$$

where $J_f(x) \in \mathbb{R}^{d \times D}$ and $J_g(f(x)) \in \mathbb{R}^{D \times d}$ denote the Jacobian matrices of the encoder and decoder respectively, and $\|\cdot\|_E, \|\cdot\|_D$ are matrix norms on these Jacobians.

For any invertible transformation $K \in \mathrm{GL}_d(\mathbb{R})$, we define the transformed encoder and decoder in the same way as the linear case (Equation 3.5):

$$f'(x) = Kf(x)$$
$$g'(z) = g(K^{-1}z).$$

Assuming $f$ and $g$ are linear, this transformation preserves the reconstruction: $g' \circ f' = g \circ f$, the linear loss term $L(x, g(f(x)))$ (Equation 3.3.1) satisfies $\mathrm{GL}_d(\mathbb{R})$ symmetry.

The Jacobian of composition $h = g \circ f$ is matrix multiplication:

$$J_h(x) = J_g(f(x)) \cdot J_f(x).$$

For encoder $f'(x) = Kf(x)$, denote $u = f(x)$, apply the linear map $\phi(u) = Ku$ get $f'(x) = (\phi \circ f)(x) = \phi(u)$.

Jacobian of the linear map $\phi(u) = Ku$ is

$$J_\phi(u) = K.$$

Apply the chain rule:

$$J_{f'}(x) = J_\phi(f(x)) \cdot J_f(x) = K \cdot J_f(x).$$

Similarly for decoder $g'(h) = g\left(K^{-1}h\right)$, denote $v = K^{-1}h$, apply linear map $\psi(h) = K^{-1}h$ get $g'(h) = (g \circ \psi)(h) = g(v)$.

The Jacobian of the linear map $\psi(h) = K^{-1}h$ is

$$J_\psi(h) = K^{-1}.$$

Apply the chain rule:

$$J_{g'}(h) = J_g(\psi(h)) \cdot J_\psi(h) = J_g\left(K^{-1}h\right) \cdot K^{-1}.$$

Evaluate at the transformed latent point $h = f'(x) = Kf(x)$ :

$$J_{g'}\left(f'(x)\right) = J_g\left(K^{-1}(Kf(x))\right) \cdot K^{-1} = J_g(f(x)) \cdot K^{-1}.$$

In conclusion, the Jacobians for the transformed encoder and decoder:

$$J_{f'}(x) = KJ_f(x), \quad J_{g'}(f'(x)) = J_g(f(x))K^{-1}.$$

Consequently, the regularization (Equation 3.3) terms become:

$$\sum_{x \in X} \left[\lambda_E \|KJ_f(x)\|_E + \lambda_D \|J_g(f(x))K^{-1}\|_D\right].$$

Our goal is to characterize the group of transformations $K$ that leave $\mathcal{L}$ (Equation 3.5) invariant for all choices of $f, g$, and $X$.

**Definition 3.15** (Isometry Group of Norm)**.** *The isometry group is the norm-preserving group. We define the left-isometry group of norm $\|\cdot\|_E$ as:*

$$\mathcal{G}_L(\|\cdot\|_E) = \{K \in \mathrm{GL}_d(\mathbb{R}) : \|KA\|_E = \|A\|_E, \forall A \in \mathbb{R}^{d\times D}\}.$$

*Similarly, the right-isometry group of norm $\|\cdot\|_D$ is:*

$$\mathcal{G}_R(\|\cdot\|_D) = \{K \in \mathrm{GL}_d(\mathbb{R}) : \|BK^{-1}\|_D = \|B\|_D, \forall B \in \mathbb{R}^{D\times d}\}.$$

The residual symmetry group of the loss $\mathcal{L}$ with Jacobian regularization is then given by the intersection:

$$\mathcal{G} = \mathcal{G}_L(\|\cdot\|_E) \cap \mathcal{G}_R(\|\cdot\|_D).$$

The residual symmetry group preserved by regularization depends critically on the choice of norm. We analyze two commonly used classes: unitarily invariant norms and entry-wise $\ell_p$ norms.

### Example 1: Unitarily Invariant Norms

**Definition 3.16** (Unitarily Invariant Norm)**.** *[22, Page 465] A norm $\|\cdot\|$ on $\mathbb{R}^{m\times n}$ is unitarily invariant if:*

$$\|UAV\| = \|A\|$$

*for all $U \in \mathrm{U}_m(\mathbb{R})$ and $V \in \mathrm{U}_n(\mathbb{R})$, where U is unitarily group (matrix). In real number, it is equivalent to say $U \in O_m(\mathbb{R})$ and $V \in O_n(\mathbb{R})$, where $O$ is orthogonal group (matrix).*

These norms depend only on singular values if $A = U\Sigma V^T$ by singular value decomposition (SVD).

The most commonly used unitarily invariant norm is the Schatten-$p$ norm.

**Definition 3.17** (Schatten-$p$ Norms [22])**.**

$$\|A\|_{S_p} = \left(\sum_{i=1}^{r} \sigma_i^p\right)^{1/p}$$

*where $\sigma_i$ are the singular values of $A$.*

Some special cases of Schatten-$p$ norms:

- $p = 2$ : Frobenius norm $\|A\|_F = \sqrt{\sum_{ij} a_{ij}^2}$.
- $p = 1$ : Nuclear norm $\|A\|_* = \sum_i \sigma_i$.
- $p = \infty$ : Spectral/operator norm $\|A\|_{\mathrm{op}} = \sigma_{\max}$ .

**Proposition 3.18.** *Schatten-$p$ norms $\subset$ Unitarily invariant norms [24].*

**Corollary 3.19.** *Use the isometry group in Definition 3.15:*

$$\mathcal{G}_L\left(\|\cdot\|_{S_p}\right)=\mathcal{G}_R\left(\|\cdot\|_{S_p}\right)=O_d(\mathbb{R}).$$

*Proof.* We prove $\mathcal{G}_L\left(\|\cdot\|_{S_p}\right) = O_d(\mathbb{R})$, similarly argument can be applied to $\mathcal{G}_R\left(\|\cdot\|_{S_p}\right)=O_d(\mathbb{R})$.

1. Prove $O_d \subseteq \mathcal{G}_L$.
   If $Q \in O_d(\mathbb{R})$, then for any $A \in \mathbb{R}^{d\times D}$ :

   $$\|QA\|_{S_p} = \|A\|_{S_p}$$

   by unitary invariance (Definition 3.16).

2. Prove $\mathcal{G}_L \subseteq O_d$.
   Suppose $K \in \mathcal{G}_L$, so $\|KA\|_{S_p} = \|A\|_{S_p}$ for all $A$.

   - Prove $K$ has unit norm columns.
     Take $A = e_i$ (standard basis vector, $d \times 1$ matrix). Then:

     $$\|Ke_i\|_{S_p} = \|e_i\|_{S_p}.$$

     For a vector, $\|v\|_{S_p} = \|v\|_2$ (the only singular value is $\|v\|_2$ ). Thus:

     $$\|Ke_i\|_2 = 1.$$

     So all columns of $K$ have unit norm.
   - Prove $K$ has orthogonal columns.
     Take $A = e_i + e_j$ for $i \neq j$ of Equation 1:

     $$\left\|K\left(e_i+e_j\right)\right\|_{S_p} = \left\|e_i+e_j\right\|_{S_p} = \sqrt{2}.$$

     The left side is:

     $$\left\|Ke_i+Ke_j\right\|_2 = \sqrt{\|Ke_i\|_2^2 + 2\left\langle Ke_i, Ke_j\right\rangle + \left\|Ke_j\right\|_2^2} = \sqrt{2+2\left\langle Ke_i, Ke_j\right\rangle}.$$

     Setting equal $\sqrt{2+2\left\langle Ke_i, Ke_j\right\rangle} = \sqrt{2}$, so $\left\langle Ke_i, Ke_j\right\rangle = 0$.

□

Thus, when using singular-values based norm, for example, Schatten-$p$ norm, for both encoder and decoder regularization, the residual symmetry group consists only of orthogonal transformations of the latent space:

$$\mathcal{G} = O_d(\mathbb{R}),$$

reducing the over-parameterization from the full $\mathrm{GL}_d(\mathbb{R})$ to a much smaller group $O_d(\mathbb{R})$. This result also encompasses the linear auto-encoder with $L_2$ regularization (Section 3.3.3) as the special case where $p = 2$ (Frobenius norm).

**Remark 3.20.** *While Schatten-$p$ regularization breaks $\mathrm{GL}_d(\mathbb{R})$ symmetry down to $O_d(\mathbb{R})$, a subtle caveat remains when encoder and decoder weights are separate.*

Based on Equation 3.5, consider a linear autoencoder with encoder $f(x) = Wx$ where $W \in \mathbb{R}^{d\times D}$, decoder $g(h) = W'h$ where $W' \in \mathbb{R}^{D\times d}$ with objective:

$$\mathcal{L}(\theta) = \frac{1}{|X|}\sum_{x\in X}\left(L(x, W'Wx) + \lambda_E \|W\|_{S_p} + \lambda_D \, \|W'\|_{S_p}\right)$$

The transformation

$$W \mapsto \alpha W, \quad W' \mapsto \frac{1}{\alpha}W'$$

for any $\alpha > 0$ preserves the reconstruction:

$$\frac{1}{\alpha}W' \cdot \alpha W x = W'Wx,$$

yet alters the regularization penalty:

$$\lambda_E \|\alpha W\|_{S_p} + \lambda_D \left\|\frac{1}{\alpha}W'\right\|_{S_p} = \lambda_E \alpha \|W\|_{S_p} + \lambda_D \frac{1}{\alpha}\, \|W'\|_{S_p}\,.$$

When $\lambda_E = \lambda_D = \lambda$, the network can minimize regularization by selecting $\alpha^* = \sqrt{\|W'\|_{S_p} / \|W\|_{S_p}}$, which yields the reduced penalty $2\lambda\sqrt{\|W\|_{S_p}\, \|W'\|_{S_p}}$. This imbalance allows the network to arbitrarily shift complexity between encoder and decoder while maintaining the same reconstruction error.

Contractive Autoencoders (CAE) [50] address this caveat by enforcing tied weights $W' = W^T$. Under this constraint, the scaling transformation becomes $W \mapsto \alpha W$ and $W' = W^T \mapsto \frac{1}{\alpha}W^T$, leading to regularization:

$$\lambda\left(\alpha + \frac{1}{\alpha}\right)\|W\|_{S_p},$$

which is minimized uniquely at $\alpha = 1$, thereby eliminating the scaling caveat.

### Example 2: Entry-wise $\ell_p$ Norm

**Definition 3.21** (Entry-wise $L_p$ Norm for Jacobian matrix [40])**.** *For a Jacobian matrix $J_f(x) \in \mathbb{R}^{d\times D}$ :*

$$\left\|J_f(x)\right\|_p = \left(\sum_{i=1}^{d}\sum_{j=1}^{D}\left|\frac{\partial f_i}{\partial x_j}(x)\right|^p\right)^{1/p} = \left\|\mathrm{vec}\left(J_f(x)\right)\right\|_p$$

*where* $\mathrm{vec}(A)$ *denotes the vectorization operator which stacks columns of a matrix into a single vector.*

This definition treats the matrix as a long vector of all entries and computes the $\ell_p$ norm.

A standard result([47, Page 45]) of $\mathrm{vec}(\cdot)$ and Kronecker product $\otimes$ is

$$\mathrm{vec}(AXB) = \left(B^T \otimes A\right)\mathrm{vec}(X).$$

This gives us two identities:

$$\begin{aligned}\mathrm{vec}(KA) &= (I \otimes K)\,\mathrm{vec}(A)\\ \mathrm{vec}\left(BK^{-1}\right) &= \left(K^{-T} \otimes I\right)\mathrm{vec}(B).\end{aligned}$$

In auto-encoder (Equation 3.3), under $K \in \mathrm{GL}_d(\mathbb{R})$ :

$$\begin{aligned}J_{f'}(x) &= KJ_f(x)\\ J_{g'}\left(f'(x)\right) &= J_g(f(x))K^{-1}.\end{aligned}$$

Apply identities (Equation 3.3.4) to encoder:

$$\left\|J_{f'}(x)\right\|_p = \left\|\mathrm{vec}\left(KJ_f(x)\right)\right\|_{\ell_p} = \left\|(I \otimes K)\,\mathrm{vec}\left(J_f(x)\right)\right\|_{\ell_p}.$$

To decoder:

$$\left\|J_{g'}\left(f'(x)\right)\right\|_p = \left\|\mathrm{vec}\left(J_g(f(x))K^{-1}\right)\right\|_{\ell_p} = \left\|\left(K^{-T} \otimes I\right)\mathrm{vec}\left(J_g(f(x))\right)\right\|_{\ell_p}.$$

The symmetry $K$ needs to satisfy

$$\|(I \otimes K)v\|_{\ell_p} = \|v\|_{\ell_p}$$

for all vectors $v$.

**Conjecture 3.22** (Residual Symmetry Group under $\ell_p$ Regularization)**.** *This conjecture is based on the Banach-Lamperti Theorem [39].*

*The residual symmetry group $\mathcal{G}_{\ell_p}$ acting on the latent space under entry-wise $\ell_p$ regularization is:*

- *For $p = 2$ (Euclidean regularization):*

$$\mathcal{G}_{\ell_2} = O_d(\mathbb{R}) = \left\{ K \in \mathrm{GL}_d(\mathbb{R}) : K^T K = I \right\}.$$

  *The symmetry group is the orthogonal group, including all rotations and reflections.*

- *For $p \in [1, \infty] \backslash \{2\}$:*

$$\mathcal{G}_{\ell_p} = \mathcal{M}_d^+ = \mathcal{D}_d^+ \rtimes S_d.$$

  *The symmetry group consists of monomial matrices, including permutations and positive scalings.*

These examples illustrate a fundamental principle: Regularizers that depend only on singular values preserve at most the orthogonal group, while regularizers that depend on specific matrix entries reduce symmetry to smaller subgroups, such as the permutation group. More precisely, to maintain a desired group structure $\mathcal{G}$ in the regularized loss function, we should select norms whose linear isometry groups coincide with $\mathcal{G}$. When regularizing both encoder and decoder, the transformation $K$ belongs to the intersection of both isometry groups, determining the final residual symmetry.