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. We prove that deep linear network, which is the most primitive network, possesses the most universal group structure. As we adding non-linear layers and regularization terms, we genuine constraining group structure. 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. By analyzing neural networks through their group-theoretic properties, we can better understand 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 rich symmetry structure that illuminates fundamental aspects of overparameterization. We establish that the parameter space of a linear network is invariant by $\mathrm{GL}_n(\mathbb{R})$ 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)**.** *Let $V_0, \ldots, V_L$ be finite-dimensional real vector spaces with $\dim V_i = n_i$. Let $\mathrm{Hom}(V, W)$ denotes the space of all linear maps from vector space V to vector space W. The parameter space of an L-layer linear network is*

$$\mathcal{P} = \prod_{i=1}^{L} \mathrm{Hom}\left(V_{i-1}, V_i\right).$$

Represent it in layer structure:

$$W = \left(W_L, \ldots, W_1\right), \quad W_i \in \mathrm{Hom}\left(V_{i-1}, V_i\right).$$

The realized linear map is the composite

$$\Phi(W) = W_L \circ W_{L-1} \circ \cdots \circ W_1 \in \mathrm{Hom}\left(V_0, V_L\right)$$

Next we need to define group action on layer structure.

Definition 3.2 (Group action on layer structure)**.** *Consider a layer group*

$$G = \prod_{i=1}^{L-1} \mathcal{G}\left(V_i\right),$$

each layer is the same group G *acts on layer V_i.*
For $K = (K_{L-1}, \ldots, K_1) \in G$, define action $\alpha : G \times \mathcal{P} \to \mathcal{P}$ by

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

By definition, K_i must be inevitable, so $\mathcal{G}(V_i) \subset \mathrm{GL}(V_i)$.

Proposition 3.3. *The action defined above is a (left) group action.*

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

Identity property: $e \cdot W = W$.

Compatibility property: $(KH) \cdot W = K \cdot (H \cdot W)$ for $K, H \in G$ because matrix multiplications associate.

□

Proposition 3.4. *The general linear group*

$$G = \prod_{i=1}^{L-1} \mathrm{GL}\left(V_i\right)$$

preserves 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, and

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

The linear network is invariant under GL *group.*

Proof.

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

□

3.2 Nonlinear Activation as Symmetry Reduction

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 [8] by introducing nonlinear activation functions $\sigma : \mathbb{R} \to \mathbb{R}$ applied element-wise to hidden layers [15]. We argue that nonlinear activations generally reduce symmetries under linear transformations, i.e. the group equipped in activation $\mathcal{G}_\sigma \subsetneq \mathrm{GL}_n(\mathbb{R})$ is a proper subgroup, 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 σ:

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

Apply a transformation $K \in G$:

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

To be invariant, we need:

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

Unpack this equation:

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

h is consecutive sub-layers in network, this formula is equivalent to:

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

Theorem 3.5 (Symmetry Reduction)**.** *The residual symmetry group $\mathcal{G}_\sigma$ of a network with activation function σ consists of transformations $K \in \mathrm{GL}_n(\mathbb{R})$ satisfying:*

$$\sigma(Kh) = K\sigma(h), \quad \forall h \in \mathbb{R}^n$$

is a centralizer subgroup:

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

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 Universal Approximation Theorem [8], which we will demonstrate in section 5.

3.2.1 Example 1: ReLU Activation

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

$$\mathrm{Cent}(\mathrm{ReLU}) = \{K \in \mathrm{GL}_n(\mathbb{R}) : \mathrm{ReLU}(Kz) = K \cdot \mathrm{ReLU}(z), \forall z\}.$$

Corollary 3.6. *The centralizer group of ReLU is constructed by semi-direct product:*

$$\mathrm{Cent}(\mathrm{ReLU}) = \mathcal{D}^+ \rtimes S_n \cong \mathcal{M}^+$$

where $\mathcal{M}^+$ denotes the monomial matrices (generalized permutation matrices) with positive entries, $\mathcal{D}^+$ denotes positive diagonal matrix, S_n is the symmetric group.

Explicitly:

$$\mathrm{Cent}(\mathrm{ReLU}) = \left\{DP : D \in \mathcal{D}^+, P \in S_n\right\}.$$

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

Choose the homomorphism:

$$\varphi : S_n \to \operatorname{Aut}\left(\mathcal{D}^+\right)$$

as conjugation:

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

where P_σ is the permutation matrix corresponding to σ. This equation holds because permute a diagonal matrix then permute it back still diagonal.

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

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

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

The representation $(d, \sigma) \mapsto D P_\sigma$ gives

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

Thus

$$\text{Cent(ReLU)} = \mathcal{D}^+ \rtimes S_n = \left\{DP : D \in \mathcal{D}^+, P \in S_n\right\}.$$

□

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

- Permutation symmetry: Neurons within a layer can be reordered with corresponding weight adjustments.
- Positive scaling symmetry: Neurons 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 $\text{sigmoid}(x) = \frac{1}{1+e^{-x}}$. The group that preserves sigmoid's action is:

$$\text{Cent}(\text{sigmoid}) = \{K \in \text{GL}_n(\mathbb{R}) : \text{sigmoid}(Kz) = K\,\text{sigmoid}(z), \forall z\}\,.$$

Corollary 3.7. *The centralizer group of sigmoid consists only of permutations:*

$$\text{Cent}(\text{sigmoid}) = S_n.$$

Explicitly:

$$\text{Cent}(\text{sigmoid}) = \{P : P \in S_n\}\,.$$

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

$$\text{sigmoid}(Kz) = K\,\text{sigmoid}(z).$$

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 which do not change values,

Comparing sigmoid with ReLU reveals a structural difference: It's 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 [7]: It reintroduces scaling flexibility that sigmoid inherently lacks, effectively expanding the learnable parameter space.

3.3 Regularization on Auto-encoder

The group action framework extends to regularization, though regularization acts globally rather than layer-wise. To rigorously define symmetry preservation under regularization, we introduce the autoencoder setting. We begin by presenting the general autoencoder formulation, then specialize to linear autoencoders, and finally examine how different regularization schemes determine the residual symmetry groups of the learned representations.

3.3.1 Auto-encoder

An auto-encoder is a type of artificial neural network designed to learn efficient data representations in an unsupervised manner[1].

Definition 3.8 (Encoder and Decoder)**.** *Let $\mathcal{X}$ be the input space, $\mathcal{Z}$ be the latent space, $\mathcal{Y}$ be the output space. Encoder is the map $f : \mathcal{X} \to \mathcal{Z}$; decoder is the map $g : \mathcal{Z} \to \mathcal{Y}$; reconstruction map is $r(x) = (g \circ f)(x)$.*

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

Definition 3.9 (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: $\rho(k_1, \rho(k_2, h)) = \rho(k_1 k_2, h)$ 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

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

For manifold learning problems mapping from $\mathbb{R}^D$ to $\mathbb{R}^d$ (where typically $d \ll D$), the encoder function f transforms an input $x \in \mathbb{R}^D$ into a hidden representation $z \in \mathbb{R}^d$:

$$z = f(x) = \sigma_f (Wx + b_z),$$

where σ_f denotes the activation function, $W \in \mathbb{R}^{d \times D}$ is the weight matrix, and $b_z \in \mathbb{R}^d$ is the bias vector.

The decoder function g reconstructs the original input from the hidden representation:

$$y = g(z) = \sigma_g \left(W'z + b_y\right),$$

where σ_g is the decoder's activation function, $W' \in \mathbb{R}^{D \times d}$ is the decoder weight matrix, and $b_y \in \mathbb{R}^D$ is the decoder bias vector.

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 $X \subset \mathbb{R}^D$:

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

where L represents the reconstruction loss function.

3.3.2 Linear Auto-encoders

Consider a linear auto-encoder (LAE) where $\sigma_f = \sigma_g = \mathrm{id}$ (identity functions), 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:

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

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 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:

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

breaks the full $\mathrm{GL}_d(\mathbb{R})$ symmetry, reducing it to the orthogonal group $O_d(\mathbb{R})$. This symmetry reduction is formalized by the Transpose Theorem [9], 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:

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

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:

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

Assuming f and g are linear, this transformation preserves the reconstruction: $g' \circ f' = g \circ f$, the linear loss term $L(x, g(f(x)))$ 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 transformed encoder and decoder:

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

Consequently, the regularization 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 $\mathcal{L}$ invariant for all choices of f, g, and X.

Definition 3.10 (Isometry Group of Norm)**.** *The isometry group is the norm-preserving group. We define the left-isometry group of norm $\|\cdot\|_E$ as:*

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

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

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

The residual symmetry group of the loss $\mathcal{L}$ 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 ℓ_p norms.

Example 1: Unitarily Invariant Norms

Definition 3.11 (Unitarily Invariant Norm[4] 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 SVD.

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

Definition 3.12 (Schatten-p Norms[4]).

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

where σ_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 $\|A\|_{\text{op}} = \sigma_{\max}$.

Proposition 3.13. *Schatten-p norms $\subset$ Unitarily invariant norms[6].*

Corollary 3.14.

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

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

Step 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.

Step 2: Prove $\mathcal{G}_L \subseteq O_d$.

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

2.1: 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.

2.2: Prove K has orthogonal columns.

Take $A = e_i + e_j$ for $i \neq j$:

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

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 as the special case where $p = 2$ (Frobenius norm).

Remark 3.15. *While Schatten-p regularization breaks* $\mathrm{GL}_d(\mathbb{R})$ *symmetry down to* $O_d(\mathbb{R})$*, a subtle loophole remains when encoder and decoder weights are separate.*

Consider a linear autoencoder with encoder $f(x) = Wx$ where $W \in \mathbb{R}^{d\times D}$, decoder $g(h) = W'h$ where $W' \in \mathbb{R}^{D\times d}$, and loss function:

$$\mathcal{L} = L\left(x, W'Wx\right) + \lambda_E\|W\|_{S_p} + \lambda_D\left\|W'\right\|_{S_p}.$$

The transformation

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

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

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

yet alters the regularization penalty:

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

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 $2\lambda\sqrt{\|W\|_{S_p}\left\|W'\right\|_{S_p}}$. This imbalance allows the network to arbitrarily shift complexity between encoder and decoder while maintaining the same reconstruction error.

Contractive Autoencoders (CAE) [16] address this vulnerability 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 loophole.
Example 2: Entry-wise ℓ_p Norm

Definition 3.16 (Entry-wise L_p Norm for Jacobian matrix [13])**.** *For a Jacobian matrix $J_f(x) \in \mathbb{R}^{d\times D}$:*

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

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

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

A standard result([14] 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, 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 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.

Theorem 3.17 (Banach, Lamperti [12]). *Let $T : \mathbb{R}^n \to \mathbb{R}^n$ be a linear operator. Then T preserves the ℓ_p norm (i.e., $\|Tv\|_{\ell_p} = \|v\|_{\ell_p}$ for all v) if and only if:*

- *For $p = 2 : T \in O_n(\mathbb{R})$ is an orthogonal matrix.*
- *For $p \in [1, \infty]\backslash\{2\} : T \in \mathcal{M}^+$ is a generalized permutation matrix.*

Geometrically, this means that ℓ_p regularization with $p \neq 2$ preserves only permutations and sign flips, while eliminating all continuous symmetries such as rotations or scalings.

These examples illustrate a fundamental principle: Regularizers that depend only on singular values preserve at most orthogonal group symmetry, while regularizers that depend on specific matrix entries reduce symmetry to smaller subgroups, for example 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 belong to the intersection of both isometry groups, determining the final residual symmetry.

3.4 Architecture

3.4.1 CNN as translation symmetry

Conjecture 3.18. *No combinations of regularization can perform translation symmetry(CNN).*