FINDING SYMMETRY IN NEURAL NETWORK PARAMETER SPACES

Anonymous authors
Paper under double-blind review

ABSTRACT

Parameter space symmetries, or loss-invariant transformations, are important for understanding neural networks' loss landscape, training dynamics, and generalization. However, identifying the full set of these symmetries remains a challenge. In this paper, we formalize data-dependent parameter symmetries and derive their infinitesimal form, which enables an automated approach to discover symmetry across different architectures. Our framework systematically uncovers parameter symmetries, including previously unknown ones. We also prove that symmetries in smaller subnetworks can extend to larger networks, allowing the discovery of symmetries in small architectures to generalize to more complex models.

1 INTRODUCTION

Parameter space symmetry, or loss-invariant transformation of parameters, influences various aspects of deep learning theory. Continuous symmetry connects groups to their orbits, revealing important topological properties such as the dimension (Zhao et al., 2023b) and connectedness (Zhao et al., 2023a) of the minimum. Parameter symmetry also influences training dynamics through the associated conserved quantities of gradient flow (Kunin et al., 2021) and by steering stochastic gradient descent towards certain favored solutions (Ziyin, 2024). Additionally, symmetry provides a tool to perform optimization within a loss level set, with successful applications in accelerating optimization (Armenta et al., 2023; Zhao et al., 2022) and improving generalization (Zhao et al., 2024). Other applications of parameter space symmetry include model compression (Ganev et al., 2022; Sourek et al., 2021) and reducing the search space for more efficient sampling in Bayesian neural networks (Wiese et al., 2023).

Despite the wide range of applications, our knowledge of parameter space symmetries is limited. In particular, known symmetries often cannot account for all loss-invariant parameter transformations. While several frameworks have been developed to unify known symmetries, whether the symmetries in current literature are complete remains an open question. Due to the lack of a systematic approach, current practice typically requires deriving symmetries from scratch for every new architecture, creating barriers for wider applications that leverage parameter symmetries.

In this paper, we present an automated approach to learn the symmetry groups and their group actions on the parameter space of neural networks. To define the search space, we formalize the definition for data-dependent symmetries and derive an infinitesimal version, which simplifies the automatic discovery architectures. Additionally, we learn the action maps directly using a neural network, which allows for learning nonlinear group actions. By including data-dependent and nonlinear group actions, our framework is capable of capturing a broader range of symmetries than previously considered.

While directly searching for symmetries in modern architectures with billions of parameters is prohibitively expensive, we show that large networks often inherit symmetries from their components or subnetworks. Identifying symmetries in small networks offers an efficient approach to uncovering many symmetries in larger networks. By analyzing small networks and extending their symmetries to larger ones, we sidestep the complexity of handling high-dimensional parameter spaces directly. This method not only reduces the computational cost of symmetry identification in large networks but also provides a systematic framework for leveraging small-scale symmetries to better understand more complex architectures.

In summary, our main contributions are:

- Formal definitions of data-dependent parameter symmetries and their infinitesimal form.
- An approach to identify symmetries in the parameter space of large networks from known symmetries in smaller subnetworks.
- A framework that discovers symmetry in neural network parameter spaces.
- Preliminary evidence of previously unknown symmetries that are data-dependent or act on non-contiguous layers.

2 Related Work

Parameter space symmetry. Parameter symmetries are loss-invariant transformations on neural network parameters, often in the form of group actions. Symmetry is present in many neural networks. Known symmetries include invertible linear transformations in linear networks, rescaling in homogeneous networks (Badrinarayanan et al., 2015; Du et al., 2018), radial rescaling in radial neural networks (Ganev et al., 2022), and translation in softmax and scaling in batchnorm functions (Kunin et al., 2021). In tanh neural networks (Chen et al., 1993), only permutation and sign flip symmetries preserve the loss function. ReLU networks, however, possess symmetries beyond the well-known rescaling (Grigsby et al., 2023). The existence and number of symmetries in most other architectures remain an open question.

Data-dependent symmetry. While the above symmetries leave the loss unchanged on all data, a relaxed definition, data-dependent symmetry, only requires loss invariance on a subset of data. Zhao et al. (2023b) found examples of such symmetries with nontrivial data dependency, although these symmetries are complicated, limited to minibatches of size one, and difficult to generalize across different architectures. This motivates an automated symmetry discovery framework, which, in principle, can find symmetries of arbitrary form in arbitrary architectures. The concept of a symmetry dependent on data has also appeared in adjacent fields. For example, (Moskalev et al., 2023) observe that learned data invariance in neural networks is strongly conditioned on data and breaks under data distribution drift; Sonoda et al. (2023) define a joint group action on data and parameters as part of a new proof of universal approximation theory.

Discovering and measuring symmetry. Various work explores learning continuous symmetries by identifying generators of Lie groups (Krippendorf & Syvaeri, 2020; Moskalev et al., 2022; Dehmamy et al., 2021; Yang et al., 2023b; Gabel et al., 2023), including cases with nonlinear group actions (Yang et al., 2023a; Shaw et al., 2024). We build on this approach to discover data-dependent group action in high-dimensional parameter spaces. While learning discrete symmetry (Zhou et al., 2021; Karjol et al., 2024) and distributions of symmetry (Benton et al., 2020; Romero & Lohit, 2022; Urbano & Romero, 2023) are also relevant, they are not the primary focus of this paper.

Extracted symmetry is often evaluated locally, by measuring function changes under infinitesimal symmetry transformations (Gruver et al., 2022) or by comparing tangent spaces of orbits under the learned group and the true symmetry group (Portilheiro, 2023). We adopt the local invariance of loss functions under symmetry transformation, similar to that defined in (Gruver et al., 2022; Moskalev et al., 2022), as the minimization objective in learning data-dependent group actions.

3 Parameter Space Symmetry

In this section, we provide a formal definition for data-dependent parameter symmetries. We then derive an alternative definition using Lie algebras, which is used to construct an automated framework for discovering parameter space symmetries in Section 5. Lastly, we provide examples of symmetries in common neural networks.

3.1 Data-dependent group action and symmetry

Let Θ be the space of parameters and $\mathcal{D}$ be the space of data. In this paper, we consider loss functions of the form $L : \Theta \times \mathcal{D} \to \mathbb{R}$, which map parameters and a single data point to a real number. By abuse of notation, we allow L to simultaneously process multiple data points. Specifically, we sometimes define $L : \Theta \times \mathcal{D}^d \to \mathbb{R}^d$ for $d \in \mathbb{N}$ data points.

Let G be a group. Consider a map a, which defines a map for every data batch of size $d \in \mathbb{Z}^+$:

$$\begin{aligned} a : \mathcal{D}^d &\to (G \times \Theta \to \Theta) \\ X &\mapsto (a_X : g, \theta \mapsto \theta'). \end{aligned} \tag{1}$$

The map a is a generalized group action on Θ if a_X is a group action for every data batch X, meaning that it satisfies the following axioms:

$$\begin{aligned} \text{identity:} \quad & a_X(I, \theta) = \theta, \qquad \forall X \in \mathcal{D}^d,\ \forall \theta \in \Theta. \\ \text{associative law:} \quad & a_X(g_2, a_X(g_1, \theta)) = a_X(g_2 g_1, \theta), \qquad \forall g_1, g_2 \in G,\ \forall X \in \mathcal{D}^d,\ \forall \theta \in \Theta. \end{aligned}$$

We introduce our first definition to formalize data-dependent symmetry. A group action a is *parameter space symmetry of L* if it additionally satisfies

$$\text{loss invariance:} \qquad L(a_X(g, \theta), X) = L(\theta, X), \qquad \forall g \in G,\ \forall X \in \mathcal{D}^d,\ \forall \theta \in \Theta.$$

A function L has a *G-symmetry* if there exists a loss-invariant group action a. We refer to G as a symmetry group of L. Additionally, the action a is termed a *data-dependent group action* or symmetry if the map (1) has a non-trivial dependency on X. That is, a is data-dependent if there exists $X_1, X_2 \in \mathcal{D}^d$, such that $a_{X_1} \neq a_{X_2}$.

3.2 Infinitesimal Symmetry

Next, we derive an infinitesimal version of parameter space symmetries. For the automatic symmetry discovery framework in Section 5, this definition allows us to learn the group elements and actions without computing the matrix exponential, which is expensive, during training. Proofs and additional examples can be found in Appendix A.

In this paper, we restrict the symmetry group G to be a linear group. That is, we assume there is a faithful representation $\rho : G \to \mathrm{GL}(n)$. The corresponding Lie algebra representation $d\rho : \mathfrak{g} \to \mathfrak{gl}(n)$ is the differential of ρ, mapping elements of the Lie algebra $\mathfrak{g}$ of G to the Lie algebra $\mathfrak{gl}(n)$ of $GL(n)$. If G is a subgroup of $\mathrm{GL}(n)$, then ρ is the inclusion map, and consequently, $d\rho$ is the inclusion of $\mathfrak{g}$ into $\mathfrak{gl}(n)$.

The following theorem shows that the derivative of the loss function L with respect to the parameters θ vanishes in the directions generated by the symmetry group's infinitesimal transformations. In other words, the loss function is invariant to small changes along these symmetric directions in parameter space.

Theorem 3.1. *Let $a : \mathcal{D}^d \to (G \times \Theta \to \Theta)$ be a parameter space symmetry of a loss function $L : \Theta \times \mathcal{D}^d \to \mathbb{R}^d$. Let $D_\theta L|_{\theta,X} : T_\theta\Theta \to \mathbb{R}^d$ be the derivative of L with respect to θ, and $D_g a_X|_{I,\theta} : \mathfrak{g} \to T_\theta\Theta$ be the derivative of $a_X(g, \theta)$ with respect to g. Then, for all $\theta \in \Theta$, $X \in \mathcal{D}^d$, and $h \in \mathfrak{g}$,*

$$(D_\theta L|_{\theta,X} \circ D_g a_X|_{I,\theta})(h) = 0. \tag{2}$$

Proof sketch. Consider a smooth curve $\gamma(t) = a_X(\exp(ht), \theta)$ in Θ, where $h \in \mathfrak{g}$ and $t \in \mathbb{R}$. Then, since L is invariant under a, $L(\gamma(t), X) = L(\theta, X), \forall t \in \mathbb{R}$. The result follows from differentiating both sides with respect to t at $t = 0$ and applying the chain rule. □

Equation 2 states that the gradient of the loss function L with respect to the parameters θ is orthogonal to the directions in parameter space generated by the infinitesimal symmetry transformations $D_g a_X\big|_{I,\theta}(h)$. This orthogonality implies that moving along these symmetric directions does not change the loss to first order, reflecting the invariance of L under the group action.

Assuming that $\Theta = \mathbb{R}^n$, then for a single data point ($d = 1$), we can write (2) in coordinates as

$$D_\theta L|_{\theta,X}\left(D_g a_X|_{I,\theta}(h)\right) = \sum_{i=1}^{n} \sum_{k=1}^{dim(\mathfrak{g})} \frac{\partial L}{\partial \theta_i} \left(D_g a_X\big|_{I,\theta}\right)_{ik} h_k = 0. \tag{3}$$

3.3 EXAMPLES

3.3.1 LINEAR ACTION OF MATRIX GROUPS

When $\Theta = \mathbb{R}^n$ and G is a subgroup of $\mathrm{GL}(n)$ with a linear, data-independent symmetry $a_x(g, \theta) = g\theta$ for all $x \in X$, (3) reduces to the equation in Theorem 3.1 in Moskalev et al. (2022). With $(D_g a)_{ijk} = \frac{\partial a_i}{\partial g_{jk}} = \delta_{ij}\theta_k$, we have

$$\left.\frac{dL(\exp(h \cdot t) \cdot \theta)}{dt}\right|_{t=0} = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} \frac{\partial L}{\partial \theta_i} \left(D_g a\big|_{I,\theta} \right)_{ijk} h_{jk} = \sum_{i=1}^{n}\sum_{k=1}^{n} \frac{\partial L}{\partial \theta_i} \theta_k h_{ik}. \tag{4}$$

Our symmetry acts on parameters instead of data, but otherwise this matches Theorem 3.1 in (Moskalev et al., 2022).

3.3.2 HOMOGENEOUS TWO-LAYER NEURAL NETWORK

We consider a homogeneous two-layer neural network with scalar weights for simplicity. Let parameter space $\Theta = \mathbb{R}^2$ and data space $X \in \mathbb{R}$. Consider the loss function

$$L : \Theta \times X \to \mathbb{R}, (w_1, w_2), x \mapsto w_2 \sigma(w_1 x)$$

with a homogeneous activation function $\sigma : \mathbb{R} \to \mathbb{R}$, i.e. $\sigma(\alpha x) = \alpha^c x$ for all $\alpha \in \mathbb{R}_{>0}$ and $x \in \mathbb{R}$, for some $c > 0$.

Let $G = (\mathbb{R}^\times, \times)$, and $\rho : G \to \mathrm{GL}_2, \alpha \mapsto \begin{pmatrix} \alpha & 0 \\ 0 & \alpha^{-c} \end{pmatrix}$. Then $a : \mathrm{GL}(2) \times \mathbb{R}^2 \to \mathbb{R}^2, \left(\rho(g), \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} \right) \mapsto \rho(g) \begin{pmatrix} w_1 \\ w_2 \end{pmatrix}$ is a symmetry of L.

4 BUILDING NEW SYMMETRIES FROM KNOWN ONES

One way to identify symmetries in a large network is by examining its components or subnetworks. Despite often having billions of parameters, neural networks typically consist of a limited set of functional families, such as fully connected layers, attention mechanisms, and activation functions. This modular view suggests a mechanism by which symmetries in networks with fewer layers might extend to those in deeper networks. Additionally, within similar types of networks, it may be possible to extrapolate symmetries found in narrower layers to wider ones.

By focusing on symmetries in small architectures and using them to infer symmetries in larger ones, we circumvent the complexity associated with direct handling of high-dimensional parameter spaces. This approach not only simplifies the discovery of symmetries in large-scale networks but also provides a systematic method for using symmetries in smaller subnetworks to understand those in more extensive architectures. We formalize this approach and discuss its limitations in the remainder of this section. Proofs can be found in Appendix B.

When the loss function L depends on a subset of the parameters solely through a subnetwork f, any symmetries that preserve f will also preserve the original network L:

Proposition 4.1. *Let $L : \Theta \times \mathcal{D}^d \to \mathbb{R}^d$ where the parameter space Θ is a product space $\Theta = \Theta_1 \times \Theta_2$. Suppose for some spaces S and T, there exist functions $h : \Theta_1 \times \mathcal{D}^d \to S$, $f : \Theta_2 \times S \to T$ and $j : (\Theta_1 \times T) \times \mathcal{D}^d \to \mathbb{R}^d$, such that for every $\theta = (\theta_1, \theta_2) \in \Theta$ and $X \in \mathcal{D}^d$, $L(\theta, X) = j\big((\theta_1, f\big(\theta_2, h(\theta_1, X)\big)\big), X\big)$. If $a : S \to (G \times \Theta_2 \to \Theta_2)$ is a G-symmetry of f, then there is an induced G-symmetry of L, $a' : \mathcal{D}^d \to (G \times \Theta \to \Theta)$, defined by $a'_X(g, (\theta_1, \theta_2)) = \big(\theta_1, a_{h(\theta_1, X)}(g, \theta_2)\big)$.*

The relationship between the functions in the proposition is described by the commutative diagram below, where $p_1 : \Theta \to \Theta_1$, $p_2 : \Theta \to \Theta_2$ are projections onto Θ_1 and Θ_2, $\mathrm{id}_1 : \Theta_1 \to \Theta_1$ and $\mathrm{id}_2 : \Theta_2 \to \Theta_2$ are identity maps, and $X \in \mathcal{D}^d$ represents a batch of data. Space S and T can be interpreted as intermediate feature spaces in the neural network. When L can be decomposed in this way, the function h does not depend on Θ_2, and the function j depends on Θ_2 only through the output of f. This effectively confines L's dependency on Θ_2 to the transformation defined by

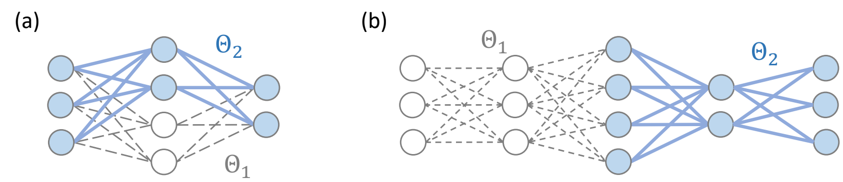

Figure 1: If a large network contains substructures with known symmetry, we can infer the same symmetry for the large network. (a) Symmetry from narrower networks. (b) Symmetry from shallower networks.

f, ensuring that any transformation on Θ_2 not altering the output of f will not affect the output of L. Consequently, symmetries identified in the smaller network f can be extrapolated to the larger network L.

$$\begin{array}{ccccccc} \Theta & & \xrightarrow{\quad L(\cdot, X) \quad} & & & & \mathbb{R}^d \\ \downarrow{\scriptstyle p_1 \times p_2 \times p_1} & & & & & & \uparrow{\scriptstyle j(\cdot, X)} \\ \Theta_1 \times \Theta_2 \times \Theta_1 & \xrightarrow{\text{id}_1 \times \text{id}_2 \times h(\cdot, X)} & & \Theta_1 \times \Theta_2 \times S & \xrightarrow{\text{id}_1 \times f(\cdot, \cdot)} & & \Theta_1 \times T \end{array}$$

We apply Proposition 4.1 to construct symmetries in larger networks from those in smaller ones in the next two corollaries. Specifically, we show that some symmetries are preserved as networks scale up through increasing the dimensionality of a layer or adding additional layers.

The first corollary describes how symmetries identified in narrower networks also apply to wider networks. A function $\sigma : \mathbb{R}^{h \times k} \to \mathbb{R}^{h \times k}$ is row-wise if, for any matrix $A \in \mathbb{R}^{h \times k}$ with rows $\{a_i \in \mathbb{R}^k\}_{i=1}^h$, the output matrix $\sigma(A)$ has rows $\{\sigma_{row}(a_i) \in \mathbb{R}^k\}_{i=1}^h$, where $\sigma_{row} : \mathbb{R}^k \to \mathbb{R}^k$ applies independently on each row of A. Element-wise functions are a special case of row-wise functions. For fully connected networks with row-wise activation functions, identifying a symmetry in one architecture suggests that the same symmetry will apply to wider versions of that architecture.

Corollary 4.2. *Consider a network parameter space $\Theta(m, h, n) = \mathbb{R}^{m \times h} \times \mathbb{R}^{h \times n}$ and data space $\mathcal{D}(n, k) = \mathbb{R}^{n \times k}$. Let $\sigma : \mathbb{R}^{h \times k} \to \mathbb{R}^{h \times k}$ be a row-wise function. Consider a function $L_{mnhk} : \Theta(m, h, n) \times \mathcal{D}(n, k) \to \mathbb{R}^{m \times k}$, defined as $L_{mnhk}((U, V), X) = U\sigma(VX)$ for $U \in \mathbb{R}^{m \times h}$, $V \in \mathbb{R}^{h \times n}$, and $X \in \mathbb{R}^{n \times k}$. If there is a G-symmetry of L_{mnhk}, then there is a G-symmetry of $L_{mnh'k}$ with any $h' > h$.*

The next corollary shows that symmetries of a subset of layers are also symmetries in the entire network.

Corollary 4.3. *Let $\Theta = \Theta_1 \times ... \times \Theta_l$ be a parameter space. Consider a list of spaces $V_0 = \mathcal{D}^d$, $V_l = \mathbb{R}^d$, and $V_1, ..., V_{l-1}$. Let $L : \Theta \times \mathcal{D}^d \to \mathbb{R}^d$ be a function defined recursively by $\{L_i\}_{i=1}^l$ with $L_i : \Theta_i \times V_{i-1} \to V_i$, such that $L = \phi_l$ where $\phi_i = L_i(\theta_i, \phi_{i-1}) \in V_i$ and $\phi_0 = X$. If for some $1 \leq i \leq l$, L_i has a G-symmetry, then L has a G-symmetry.*

Both corollaries can be proved by factoring the parameter space and defining corresponding functions that compose to L, before applying Proposition 4.1. The explicit forms of h, f, and j are deferred to Appendix B. Figure 1 shows the subset of parameters (Θ_2) that the symmetry applies to in the corollaries. These are the subnetworks where symmetries are assumed to be known and which the larger network inherits.

Note that this approach does not explore the emergence of new, more complex symmetries that may arise as the neural network scale up in size. Notably, there are cases where there exists a G symmetry over its input space, but group actions on individual layers are not loss-invariant (Kvinge et al. (2022)). Nevertheless, studying smaller and simpler networks remains a effective strategy to obtain a significant number of symmetries in larger networks, and is a first step in characterizing the complete set of symmetries in modern architectures.

In addition to obtaining symmetries from those in smaller networks, we can also get symmetries for a loss function over data batches with a certain size, if we know there is a symmetry for this function over larger data batches. Concretely, if there exists a group action that preserves loss for all data batches of size $d \in \mathbb{Z}^+$, then that group action preserves loss for all data batches of size $d' < d$.

Proposition 4.4. *Let $L_d : \Theta \times \mathcal{D}^d \to \mathbb{R}^d$ be a function that is applied pointwise on each of d data points in a data batch. If L_d admits a G-symmetry, then $L_{d'}$ admits a G-symmetry for all $d' < d$.*

5 Automatic Discovery of Parameter Symmetries

Formulating symmetries in the infinitesimal form makes them easier to learn using an automatic framework, as it defines a set of local conditions for a function to be a symmetry. Using the infinitesimal symmetry derived in Section 3.2, we construct an automated framework for discovering parameter space symmetries.

5.1 Enforcing Loss Invariance and Group Axioms

Given a function L, our goal is to find a symmetry a and a set of Lie algebra elements h corresponding to a symmetry group of L. We parameterize a using a neural network with learnable parameters, and set h to be learnable as well. We define the following loss terms that quantify the deviation from loss invariance and the group axioms (identity and associativity law):

$$\mathcal{L}_{\text{invariance}} = \mathbb{E}_{x,\theta} |D_\theta L|_{\theta,X} \circ D_g a_X|_{I,\theta}(h)| \tag{5}$$

$$\mathcal{L}_{\text{id}} = \mathbb{E}_{x,\theta} \|a_x(I,\theta) - \theta\|_2 \tag{6}$$

$$\mathcal{L}_{\text{assoc}} = \sum_{h_1,h_2 \in \mathfrak{g}} \mathbb{E}_{x,\theta} \left\| D_g a_X\big|_{I,\theta}(h_2) D_g a_X\big|_{I,\theta}(h_1) - \frac{1}{2} D_g a_X\big|_{I,\theta}([h_1,h_2]) \right\|. \tag{7}$$

The three loss terms bias the action towards being loss-invariant, preserving identity, and satisfying the associativity property. By minimizing $\mathcal{L}_{\text{Lie\_deriv}}$, we ensure that the learned symmetry a and the Lie algebra element h satisfy the infinitesimal symmetry condition (Theorem 3.1). Minimizing $\mathcal{L}_{\text{id}}$ enforces the identity axiom, ensuring that the action of the identity element leaves the parameters unchanged. Minimizing $\mathcal{L}_{\text{assoc}}$ enforces the associative axiom (derivation in Appendix A.2).

By focusing on the Lie algebras, we enforce the loss invariance and group structure at the infinitesimal level. This formulation allows us to avoid computing exponential maps.

5.2 Regularizations

To prevent the learned group action from becoming trivial, we encourage the infinitesimal action to be nonzero. On the other hand, we do not want it to grow infinitely large for training stability. Therefore, in implementation, we include the following regularization term to encourage the norm of the infinitesimal action to be around a fixed positive real number β:

$$\mathcal{L}_{\text{reg\_id}} = \min_{a,h} \mathbb{E}_\theta |\beta - \|D_g a_X|_{I,\theta}(h)\||. \tag{8}$$

When learning multiple generators simultaneously, we want them to be orthogonal. Following Yang et al. (2023b), we do this by including the following cosine similarity between each pair of the k generators in the loss function:

$$\mathcal{L}_{\text{reg\_h\_orth}} = \sum_{1 \le i < j \le k} \frac{h_i \cdot h_j}{\|h_i\| \|h_j\|}. \tag{9}$$

Finally, we encourage sparsity of h for easier interpretation, with the regularization term

$$\mathcal{L}_{\text{reg\_h\_sparse}} = \sum_{k,j} |h_{kj}|. \tag{10}$$

The final training objective is a weighted average of (6)-(11), with hyperparameters $\gamma_1, ..., \gamma_6 \in \mathbb{R}^+$:

$$\min_{h,a} \left(\gamma_1 \mathcal{L}_{\text{invariance}} + \gamma_2 \mathcal{L}_{\text{id}} + \gamma_3 \mathcal{L}_{\text{assoc}} + \gamma_4 \mathcal{L}_{\text{reg\_id}} + \gamma_5 \mathcal{L}_{\text{reg\_h\_orth}} + \gamma_6 \mathcal{L}_{\text{reg\_h\_sparse}} \right). \tag{11}$$

5.3 LEARNED DATA-INDEPENDENT SYMMETRIES

In the first set of tasks, we see if our method can learn generators for architectures with already known data-independent symmetries. We consider two-layer networks in the form of $L(W_1, W_2, X, Y) = \|W_2\sigma(W_1X) - Y\|^2$, where $W_2 \in \mathbb{R}^{m\times h}, W_1 \in \mathbb{R}^{h\times n}$ are parameters, $X \in \mathbb{R}^{n\times k}$, $Y \in \mathbb{R}^{m\times k}$ are data, and σ is a homogeneous activation function.

During training, we train the generators h and the group action a under objective (11). We parametrize a using a 4-layer MLP with hidden dimensions 64, 64, 64. The group aciton a takes a group element, parameter, and data as input and outputs transformed parameters. We use 10000 training samples, each containing a randomly generated set of parameters and data. We set the learning rate as 10^{-3} with decay 0.6 every 1000 steps, and the weights for the multi-objective loss as $\gamma_1 = 10$, $\gamma_2 = \gamma_4 = \gamma_5 = 1$, and $\gamma_6 = 0.1$.

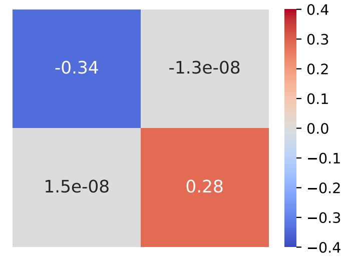

Figure 2: Generator for a two-layer linear MLP with scalar parameters and data.

As a proof of concept, we training a group action and a single generator $h \in \mathbb{R}^{2\times 2}$ for the two-layer architecture with $m = h = n = k = 1$ and σ being the identity function. Figure 2 visualizes the learned generator, which matches the expected generator that generates the rescaling group.

Note that, however, we do not impose constraints on the group action (in particular, not enforcing linear actions). Hence we do not expect the learned generators to look similar to the elements of the Lie algebra infinitesimal generators of the symmetry group in general. For example, the action a can be a composition of two function, the first transforming learned generators to the set of actual generators, and the second performing the group action. We find that our method can learn the generators and group actions for wider two-layer homogeneous architectures as well. More examples of learned generators for larger architectures can be found in Appendix C.

5.4 LEARNED DATA-DEPENDENT SYMMETRIES

As a more practical application of our framework, we attempt to uncover data-dependent symmetries from architectures where no continuous symmetry is known before. We apply our framework to learn generators and loss-invariant group actions for two-layer neural network with sigmoid and tanh activation function, as well as a three-layer neural network with skip connection.

Specifically, we aim to learn symmetries in the two-layer networks defined in the previous section, but replacing σ by sigmoid or tanh. Our objective is again to find a set of generators h and a group action a that minimizes (11). We use 10000 training samples, each containing a randomly generated set of parameters and data. We set the learning rate as 10^{-3} and the weights for the multi-objective loss as $\gamma_1 = 1$, $\gamma_2 = \gamma_4 = 10$, $\gamma_5 = 1$, and $\gamma_6 = 0.1$.

Figure 3 shows the learned generators for data-dependent symmetries in a two-layer sigmoid MLP with parameters dimensions $W_1 \in \mathbb{R}^{3\times 3}, W_2 \in \mathbb{R}^{3\times 1}$ and data $X \in \mathbb{R}^{3\times 1}, Y \in \mathbb{R}^{1\times 1}$. Figure 6 in the Appendix shows the training curve. Since sigmoid networks have no data-independent continuous symmetry, this set of symmetries are data-dependent, indicating that our method successfully learns data-dependent symmetries for this architecture.

Figure 4 shows the learned generators for data-dependent symmetries in a three-layer tanh MLP with parameters dimensions $W_1 \in \mathbb{R}^{2\times 2}, W_2 \in \mathbb{R}^{2\times 2}, W_3 \in \mathbb{R}^{2\times 1}$ and data $X \in \mathbb{R}^{1\times 2}, Y \in \mathbb{R}^{1\times 1}$. The generators indicate the existence of symmetries that act on non-contiguous layers, which has not been discovered in previous literature.

6 DISCUSSION

While our discovery framework suggests that there are previously unknown data-dependent symmetries in various neural network architectures, the existence and number of symmetries in neural network parameter spaces remain open questions. Whether the number of symmetries is affected

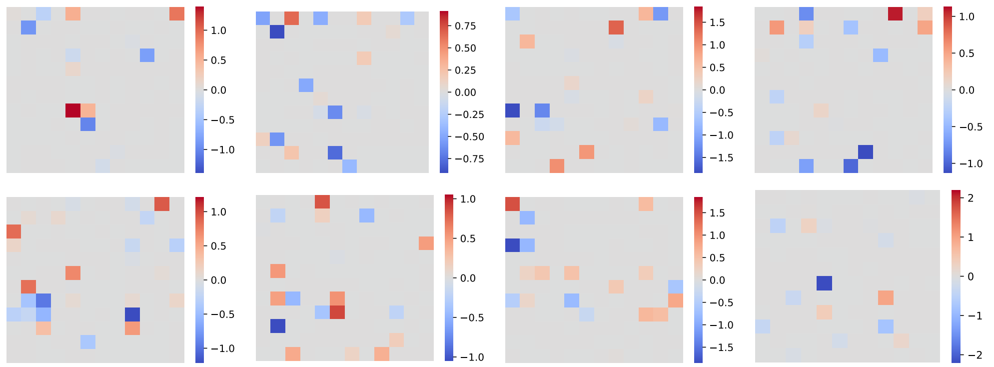

Figure 3: Learned generators for data-dependent symmetries in a two-layer sigmoid MLP with parameters dimensions $W_2 \in \mathbb{R}^{3\times 1}, W_1 \in \mathbb{R}^{3\times 3}$ and data $X \in \mathbb{R}^{1\times 3}, Y \in \mathbb{R}^{1\times 1}$.

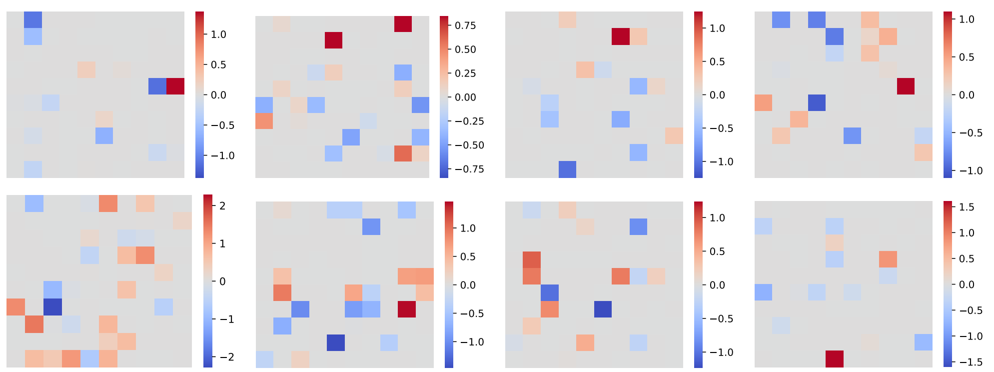

Figure 4: Learned generators for data-dependent symmetries in a three-layer tanh MLP with parameters dimensions $W_1 \in \mathbb{R}^{2\times 2}, W_2 \in \mathbb{R}^{2\times 2}, W_3 \in \mathbb{R}^{2\times 1}$ and data $X \in \mathbb{R}^{1\times 2}, Y \in \mathbb{R}^{1\times 1}$.

by existence of symmetry in data or changes during training are also interesting directions. Future work will examine the structure of learned symmetry, such as the dimension of Lie algebras.

REFERENCES

Marco Armenta, Thierry Judge, Nathan Painchaud, Youssef Skandarani, Carl Lemaire, Gabriel Gibeau Sanchez, Philippe Spino, and Pierre-Marc Jodoin. Neural teleportation. *Mathematics*, 11(2):480, 2023.

Vijay Badrinarayanan, Bamdev Mishra, and Roberto Cipolla. Symmetry-invariant optimization in deep networks. *arXiv preprint arXiv:1511.01754*, 2015.

Gregory Benton, Marc Finzi, Pavel Izmailov, and Andrew G Wilson. Learning invariances in neural networks from training data. *Advances in neural information processing systems*, 33:17605–17616, 2020.

An Mei Chen, Haw-minn Lu, and Robert Hecht-Nielsen. On the geometry of feedforward neural network error surfaces. *Neural computation*, 5(6):910–927, 1993.

Nima Dehmamy, Robin Walters, Yanchen Liu, Dashun Wang, and Rose Yu. Automatic symmetry discovery with lie algebra convolutional network. *Advances in Neural Information Processing Systems*, 34:2503–2515, 2021.

Simon S Du, Wei Hu, and Jason D Lee. Algorithmic regularization in learning deep homogeneous models: Layers are automatically balanced. *Neural Information Processing Systems*, 2018.

Alex Gabel, Victoria Klein, Riccardo Valperga, Jeroen SW Lamb, Kevin Webster, Rick Quax, and Efstratios Gavves. Learning lie group symmetry transformations with neural networks. In *Topological, Algebraic and Geometric Learning Workshops 2023*, pp. 50–59. PMLR, 2023.

Iordan Ganev, Twan van Laarhoven, and Robin Walters. Universal approximation and model compression for radial neural networks. *arXiv preprint arXiv:2107.02550v2*, 2022.

Elisenda Grigsby, Kathryn Lindsey, and David Rolnick. Hidden symmetries of relu networks. In *International Conference on Machine Learning*, pp. 11734–11760. PMLR, 2023.

Nate Gruver, Marc Anton Finzi, Micah Goldblum, and Andrew Gordon Wilson. The lie derivative for measuring learned equivariance. In *The Eleventh International Conference on Learning Representations*, 2022.

Pavan Karjol, Rohan Kashyap, Aditya Gopalan, and A. P. Prathosh. A unified framework for discovering discrete symmetries. In *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pp. 793–801. PMLR, 2024.

Sven Krippendorf and Marc Syvaeri. Detecting symmetries with neural networks. *Machine Learning: Science and Technology*, 2(1):015010, 2020.

Daniel Kunin, Javier Sagastuy-Brena, Surya Ganguli, Daniel LK Yamins, and Hidenori Tanaka. Neural mechanics: Symmetry and broken conservation laws in deep learning dynamics. In *International Conference on Learning Representations*, 2021.

Henry Kvinge, Tegan Emerson, Grayson Jorgenson, Scott Vasquez, Tim Doster, and Jesse Lew. In what ways are deep neural networks invariant and how should we measure this? *Advances in Neural Information Processing Systems*, 35:32816–32829, 2022.

Artem Moskalev, Anna Sepliarskaia, Ivan Sosnovik, and Arnold Smeulders. Liegg: Studying learned lie group generators. *Advances in Neural Information Processing Systems*, 35:25212–25223, 2022.

Artem Moskalev, Anna Sepliarskaia, Erik J Bekkers, and Arnold WM Smeulders. On genuine invariance learning without weight-tying. In *Topological, Algebraic and Geometric Learning Workshops 2023*, pp. 218–227. PMLR, 2023.

Vasco Portilheiro. Quantifying lie group learning with local symmetry error. In *NeurIPS 2023 Workshop on Symmetry and Geometry in Neural Representations*, 2023.

David W Romero and Suhas Lohit. Learning partial equivariances from data. *Advances in Neural Information Processing Systems*, 35:36466–36478, 2022.

Ben Shaw, Abram Magner, and Kevin R Moon. Symmetry discovery beyond affine transformations. *arXiv preprint arXiv:2406.03619*, 2024.

Sho Sonoda, Hideyuki Ishi, Isao Ishikawa, and Masahiro Ikeda. Joint group invariant functions on data-parameter domain induce universal neural networks. In *NeurIPS 2023 Workshop on Symmetry and Geometry in Neural Representations*, 2023.

Gustav Sourek, Filip Zelezny, and Ondrej Kuzelka. Lossless compression of structured convolutional models via lifting. In *International Conference on Learning Representations*, 2021.

Alonso Urbano and David W Romero. Self-supervised detection of perfect and partial input-dependent symmetries. *arXiv preprint arXiv:2312.12223*, 2023.

Jonas Gregor Wiese, Lisa Wimmer, Theodore Papamarkou, Bernd Bischl, Stephan Günnemann, and David Rügamer. Towards efficient mcmc sampling in bayesian neural networks by exploiting symmetry. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD): Research Track*, pp. 459–474, 2023.

Jianke Yang, Nima Dehmamy, Robin Walters, and Rose Yu. Latent space symmetry discovery. *arXiv preprint arXiv:2310.00105*, 2023a.

Jianke Yang, Robin Walters, Nima Dehmamy, and Rose Yu. Generative adversarial symmetry discovery. *International Conference on Machine Learning*, 2023b.

Bo Zhao, Nima Dehmamy, Robin Walters, and Rose Yu. Symmetry teleportation for accelerated optimization. *Advances in Neural Information Processing Systems*, 2022.

Bo Zhao, Nima Dehmamy, Robin Walters, and Rose Yu. Understanding mode connectivity via parameter space symmetry. In *UniReps: the First Workshop on Unifying Representations in Neural Models*, 2023a.

Bo Zhao, Iordan Ganev, Robin Walters, Rose Yu, and Nima Dehmamy. Symmetries, flat minima, and the conserved quantities of gradient flow. *International Conference on Learning Representations*, 2023b.

Bo Zhao, Robert M Gower, Robin Walters, and Rose Yu. Improving convergence and generalization using parameter symmetries. *International Conference on Learning Representations*, 2024.

Allan Zhou, Tom Knowles, and Chelsea Finn. Meta-learning symmetries by reparameterization. *International Conference on Learning Representations*, 2021.

Liu Ziyin. Symmetry leads to structured constraint of learning. In *International Conference on Machine Learning*. PMLR, 2024.

APPENDIX

A INFINITESIMAL SYMMETRY AND EXAMPLES

A.1 INFINITESIMAL FORMULATION FOR LOSS INVARIANCE

Theorem 3.1. *Let $a : \mathcal{D}^d \to (G \times \Theta \to \Theta)$ be a parameter space symmetry of a loss function $L : \Theta \times \mathcal{D}^d \to \mathbb{R}^d$. Let $D_\theta L|_{\theta,X} : T_\theta\Theta \to \mathbb{R}^d$ be the derivative of L with respect to θ, and $D_g a_X|_{I,\theta} : \mathfrak{g} \to T_\theta\Theta$ be the derivative of $a_X(g,\theta)$ with respect to g. Then, for all $\theta \in \Theta$, $X \in \mathcal{D}^d$, and $h \in \mathfrak{g}$,*

$$D_\theta L|_{\theta,X} \circ D_g a_X|_{I,\theta} \circ h = 0.$$

Proof. Since a is a symmetry of L, we have

$$L(a_X(g,\theta), X) = L(\theta, X), \quad \forall g \in G, \quad \forall \theta \in \Theta, \quad \forall X \in \mathcal{D}^d.$$

Consider a smooth curve $\gamma(t) = a_X(\exp(ht), \theta)$ in Θ, where $h \in \mathfrak{g}$ and $t \in \mathbb{R}$. Then, since L is invariant under a,

$$L(\gamma(t), X) = L(\theta, X), \quad \forall t \in \mathbb{R}.$$

Differentiating both sides with respect to t at $t = 0$, we get

$$\frac{d}{dt} L(\gamma(t), X)\bigg|_{t=0} = 0.$$

Applying the chain rule,

$$\frac{d}{dt} L(\gamma(t), X)\bigg|_{t=0} = D_\theta L|_{\theta,X}\left(\frac{d\gamma(t)}{dt}\bigg|_{t=0}\right).$$

Now, compute $\frac{d\gamma(t)}{dt}\big|_{t=0}$ using the chain rule:

$$\frac{d\gamma(t)}{dt}\bigg|_{t=0} = \frac{d}{dt} a_X(\exp(ht), \theta)\bigg|_{t=0} = D_g a_X|_{I,\theta}\left(\frac{d}{dt}\exp(ht)\bigg|_{t=0}\right).$$

Since $\exp$ is the exponential map from $\mathfrak{gl}(n)$ to $GL(n)$, and $h \in \mathfrak{gl}(n)$, we have

$$\frac{d}{dt}\exp(ht)\bigg|_{t=0} = h.$$

Therefore,

$$\frac{d\gamma(t)}{dt}\bigg|_{t=0} = D_g a_X|_{I,\theta}(h).$$

Putting it all together,

$$D_\theta L|_{\theta,X}\left(D_g a_X|_{I,\theta}(h)\right) = 0.$$

□

A.2 INFINITESIMAL FORMULATION FOR ASSOCIATIVITY AXIOM

In this section, we rewrite the associative axiom,

$$a_X(g_2, a_X(g_1, \theta)) = a_X(g_2 g_1, \theta), \tag{12}$$

into an infinitesimal form that uses Lie algebras but avoids the use of exponential maps. Below is a detailed derivation.

First, we consider infinitesimal group elements. Let $g_1 = \exp(\varepsilon h_1)$ and $g_1 = \exp(\varepsilon h_2)$, where ε is an infinitesimally small scalar, and $h_1, h_2 \in \mathfrak{g}$.

Expand the group multiplication to second order in ε using the Baker-Campbell-Hausdorff (BCH) formula:

$$g_2 g_1 \approx I + \varepsilon(h_1 + h_2) + \tfrac{1}{2}\varepsilon^2 [h_1, h_2].$$

Expand the right side of (12) to second order:

$$a_X(g_2 g_1, \theta) \approx \theta + \varepsilon D_g a_X\big|_{I,\theta}(h_1 + h_2) + \tfrac{1}{2}\varepsilon^2 D_g a_X\big|_{I,\theta}([h_1, h_2]).$$

Expand the left side of (12) to second order:

$$\begin{aligned} a_X(g_2, a_X(g_1, \theta)) &\approx a_X(I + \varepsilon h_2, \theta + \varepsilon\, D_g a_X|_{I,\theta}(h_1)) \\ &\approx \theta + \varepsilon D_g a_X\big|_{I,\theta}(h_2) + \varepsilon D_g a_X\big|_{I,\theta}(h_1) + \varepsilon^2 D_g a_X\big|_{I,\theta}(h_2) D_g a_X\big|_{I,\theta}(h_1). \end{aligned}$$

By associativity axiom, we expect the two sides to be equal. Since the first-order terms from both sides match, we equate the second-order terms. To enforce the associative axiom, we define the infinitesimal associative loss as the difference between the second-order terms from two sides:

$$L_{\text{assoc}} = \sum_{h_1, h_2 \in \mathfrak{g}} \mathbb{E}_{x,\theta} \left\| D_g a_X\big|_{I,\theta}(h_2) D_g a_X\big|_{I,\theta}(h_1) - \frac{1}{2} D_g a_X\big|_{I,\theta}([h_1, h_2]) \right\|.$$

This loss enforces that the commutator of the infinitesimal actions matches the Lie bracket of the Lie algebra, satisfying the associative property at the infinitesimal level.

A.3 Alternative Option for Discovery Objectives

A more straightforward training objective exponentiates the Lie algebra to obtain group elements, before enforcing loss invariance and group axioms:

$$\min_{h,a} L_{\text{invariance\_int}} + L_{\text{id\_int}} + L_{\text{assoc\_int}}$$

with

$$\begin{aligned} L_{\text{invariance\_int}} &= \mathbb{E}_{x,\theta,t} \| L\left(a_x(exp(ht), \theta), x\right) - L(\theta, x) \| \\ L_{\text{id\_int}} &= \mathbb{E}_{x,\theta} \| a_x(I, \theta) - \theta \| \\ L_{\text{assoc\_int}} &= \sum_{h_1, h_2 \in \mathfrak{g}} \mathbb{E}_{x,\theta} \left\| a_{\exp(h_1)X}(\exp(h_2), a_X(\exp(h_1), \theta)) - a_X(\exp(h_2)\exp(h_1), \theta) \right\|. \end{aligned}$$

Similarly to the infinitesimal version, this objective also directly enforces the necessary group structures. We adopt the infinitesimal formulation to avoid the computational overhead of evaluating exponential maps.

A.4 Homogeneous function properties

Proposition A.1 (Euler's homogeneous function theorem)**.** *Let $f : \mathbb{R} \to \mathbb{R}$ be a homogeneous function, i.e. $f(\alpha x) = \alpha^c x$ for all $\alpha \in \mathbb{R}_{>0}$ and $x \in \mathbb{R}$, for some $c > 0$. If f is differentiable at x, then $\frac{df}{dx} = c x^{-1} f(x)$.*

Proof. Using the definition of homogeneous function, the derivative of f at x is

$$\begin{aligned} \frac{df}{dx} &= \lim_{t \to 0} \frac{f(x+t) - f(x)}{t} \\ &= \lim_{t \to 0} \frac{f((1 + t x^{-1})x) - f(x)}{t} \\ &= \lim_{t \to 0} \frac{(1 + t x^{-1})^c f(x) - f(x)}{t} \\ &= \lim_{t \to 0} \frac{c(1 + t x^{-1})^{c-1} x^{-1} f(x)}{1} \\ &= c x^{-1} f(x) \end{aligned} \tag{13}$$

□

B BUILDING SYMMETRIES FROM KNOWN ONES

This section contains the proofs for results in Section 4.

Proposition 4.1. *Let $L : \Theta \times \mathcal{D}^d \to \mathbb{R}^d$ be a function, where the parameter space Θ is a product space $\Theta = \Theta_1 \times \Theta_2$, with spaces Θ_1, Θ_2. Suppose there exist functions $h : \Theta_1 \times \mathcal{D}^d \to S$, $f : \Theta_2 \times S \to T$, and $j : (\Theta_1 \times T) \times \mathcal{D}^d \to \mathbb{R}^d$, such that for every $\theta = (\theta_1, \theta_2) \in \Theta$ and $X \in \mathcal{D}^d$, $L(\theta, X) = j\big((\theta_1, f\big(\theta_2, h(\theta_1, X)\big)), X\big)$. If $a : S \to (G \times \Theta_2 \to \Theta_2)$ is a G-symmetry of f, then there is an induced G-symmetry of L, $a' : \mathcal{D}^d \to (G \times \Theta \to \Theta)$, defined by $a'_X(g, (\theta_1, \theta_2)) = \big(\theta_1, a_{h(\theta_1,X)}(g, \theta_2)\big)$.*

Proof. We need to show that a' satisfies the identity and associative law of a group action and preserves L.

Since a is a group action on Θ_2, it satisfies the identity axiom $a_{h(\theta_1,X)}(I, \theta_2) = \theta_2$. Applying this in the definition of a', we get $a'_X(I, (\theta_1, \theta_2)) = (\theta_1, a_{h(\theta_1,X)}(I, \theta_2)) = (\theta_1, \theta_2)$.

Since a is a group action on Θ_2, it satisfies the associative law $a_{h(\theta_1,X)}(g_2 g_1, \theta_2) = a_{h(\theta_1,X)}(g_2, a_{h(\theta_1,X)}(g_1, \theta_2))$, for all $g_1, g_2 \in G$. It follows that a' also satisfies the associative law: $a'_X(g_2 g_1, (\theta_1, \theta_2)) = (\theta_1, a_{h(\theta_1,X)}(g_2 g_1, \theta_2)) = (\theta_1, a_{h(\theta_1,X)}(g_2, a_{h(\theta_1,X)}(g_1, \theta_2))) = a'_X(g_2, a'_X(g_1, (\theta_1, \theta_2)))$

Finally, since a is a symmetry of f, we have $f(a_{h(\theta_1,X)}(g, \theta_2), h(\theta_1, X)) = f(\theta_2, h(\theta_1, X))$, for all $g \in G$. It follows that a' preserves the value of L: $L(a'_X(g, \theta), X) = j\big((\theta_1, f\big(a_{h(\theta_1,X)}(g, \theta_2), h(\theta_1, X)\big)), X\big) = j\big((\theta_1, f\big(\theta_2, h(\theta_1, X)\big)), X\big) = L(\theta, X)$. ☐

Corollary 4.2. *Consider a network parameter space $\Theta(m, h, n) = \mathbb{R}^{m\times h} \times \mathbb{R}^{h\times n}$ and data space $\mathcal{D}(n, k) = \mathbb{R}^{n\times k}$. Let $\sigma : \mathbb{R}^{h\times k} \to \mathbb{R}^{h\times k}$ be a row-wise function. Consider a function $L_{mnhk} : \Theta(m, h, n) \times \mathcal{D}(n, k) \to \mathbb{R}^{m\times k}$, defined as $L_{mnhk}((U, V), X) = U\sigma(VX)$ for $U \in \mathbb{R}^{m\times h}$, $V \in \mathbb{R}^{h\times n}$, and $X \in \mathbb{R}^{n\times k}$. If there is a G-symmetry of L_{mnhk}, then there is a G-symmetry of $L_{mnh'k}$ with any $h' > h$.*

Proof. The function $L_{mnh'k}$ can be decomposed into

$$\begin{aligned} U(\sigma(VX))_{ik} &= U_{ij}\sigma(VX)_{jk} \\ &= \sum_{j=1}^{h}\sum_{l=1}^{n} U_{ij}\sigma(V_{jl}X_{lk}) \\ &= \sum_{j=1}^{h}\sum_{l=1}^{n} U_{ij}\sigma(V_{jl}X_{lk}) + \sum_{j=h+1}^{h'}\sum_{l=1}^{n} U_{ij}\sigma(V_{jl}X_{lk}) \end{aligned} \tag{14}$$

Note that for all i, k, the first term depends only on the first h columns of U and first h rows of V, and the second terms depends only on the rest of the columns and rows of U and V. Denoting the first h columns of U as $U_{1:h}$, the rest of the columns of U as $U_{h+1:h'}$, the first h rows of V as $V_{1:h}$, and the rest of the rows of V as $V_{h+1:h'}$, we have

$$L_{mnh'k}((U, V), X) = L_{mnhk}((U_{1:h}, V_{1:h}), X) + L_{mn(h'-h)k}((U_{h+1:h'}, V_{h+1:h'}), X). \tag{15}$$

Let $\Theta_1 = \mathbb{R}^{m\times h} \times \mathbb{R}^{h\times n}$ and $\Theta_2 = \mathbb{R}^{m\times(h'-h)} \times \mathbb{R}^{(h'-h)\times n}$. Then $\Theta(m, h', n) = \Theta_1 \times \Theta_2$. Let $S = (\mathbb{R}^{m\times k} \times \mathcal{D}^d)$ and $T = \mathbb{R}^{m\times k} \times \mathbb{R}^{m\times k}$. Define the following three functions

$$\begin{aligned} &h : \Theta_1 \times \mathcal{D}^d \to (\mathbb{R}^{m\times k} \times \mathcal{D}^d) \\ &f : \Theta_2 \times (\mathbb{R}^{m\times k} \times \mathcal{D}^d) \to \mathbb{R}^{m\times k} \times \mathbb{R}^{m\times k} \\ &j : (\Theta_1 \times (\mathbb{R}^{m\times k} \times \mathbb{R}^{m\times k})) \times \mathcal{D}^d \to \mathbb{R}^{m\times k} \end{aligned} \tag{16}$$

by

$$\begin{aligned} h((U_{1:h}, V_{1:h}), X) &= (L_{mnhk}((U_{1:h}, V_{1:h}), X), X) \\ f((U_{h+1:h'}, V_{h+1:h'}), (Y, X)) &= \big(L_{mn(h'-h)k}((U_{h+1:h'}, V_{h+1:h'}), X), Y\big) \\ j\big(((U_{1:h}, V_{1:h}), (Y', Y)\big), X) &= Y' + Y. \end{aligned} \tag{17}$$

Then $L_{mnh'k}(\theta, X) = j\big((\theta_1, f\big(\theta_2, h(\theta_1, X)\big)), X\big)$ for all $\theta = (\theta_1, \theta_2) \in \Theta$ and $X \in \mathcal{D}^d$. Since L_{mnhk} has a symmetry, f has the same symmetry. By Proposition 4.1, $L_{mnh'k}$ also has the same symmetry. ☐

Corollary 4.3. *Let $\Theta = \Theta_1 \times ... \times \Theta_l$ be a parameter space. Consider a list of spaces $V_0 = \mathcal{D}^d$, $V_l = \mathbb{R}^d$, and V_1, ..., V_{l-1}. Let $L : \Theta \times \mathcal{D}^d \to \mathbb{R}^d$ be a function defined recursively by $\{L_i\}_{i=1}^l$ with $L_i : \Theta_i \times V_{i-1} \to V_i$, such that $L = \phi_l$ where $\phi_i = L_i(\theta_i, \phi_{i-1}) \in V_i$ and $\phi_0 = X$. If for some $1 \leq i \leq l$, L_i has a G-symmetry, then L has a G-symmetry.*

Proof. Define functions

$$\begin{aligned} &h : (\Theta_1 \times ... \times \Theta_{i-1} \times \Theta_{i+1} \times ... \times \Theta_l) \times \mathcal{D}^d \to V_{i-1} \\ &f : \Theta_i \times V_{i-1} \to V_i \\ &j : (\Theta_1 \times ... \times \Theta_{i-1} \times \Theta_{i+1} \times ... \times \Theta_l) \times V_i \times \mathcal{D}^d \to \mathbb{R}^d \end{aligned} \tag{18}$$

by

$$\begin{aligned} h((\theta_1, ..., \theta_{i-1}, \theta_{i+1}, ..., \theta_l), X) &= L_{i-1}(\theta_{i-1}, X), \quad \text{computed using } (\theta_1, ..., \theta_{i-1}) \\ f(\theta_i, \phi_{i-1}) &= L_i(\theta_i, \phi_{i-1}) \\ j((\theta_1, ..., \theta_{i-1}, \theta_{i+1}, ..., \theta_l), \phi_i, X) &= L_l(\theta_l, X), \quad \text{computed using } (\theta_l, ..., \theta_{i+1}) \text{ and } \phi_i. \end{aligned} \tag{19}$$

Then $L((\theta_1, ..., \theta_l), X) = j\big((\theta_1, ..., \theta_{i-1}, \theta_{i+1}, ..., \theta_l), f(\theta_i, h((\theta_1, ..., \theta_{i-1}, \theta_{i+1}, ..., \theta_l), X)), X\big)$ for all $\theta = (\theta_1, \theta_2) \in \Theta$ and $X \in \mathcal{D}^d$. By Proposition 4.1, if $f = L_i$ has a G-symmetry, L also has a G-symmetry. ☐

Proposition 4.4. *Let $L_d : \Theta \times \mathcal{D}^d \to \mathbb{R}^d$ be a function that is applied pointwise on each of d data points in a data batch. If L_d admits a G-symmetry, then $L_{d'}$ admits a G-symmetry for all $d' < d$.*

Proof. Suppose that L_d has a G-symmetry. Let $a : \mathcal{D}^d \to (G \times \Theta \to \Theta), X_d \mapsto (a_{X_d} : g, \theta \mapsto \theta')$ be the corresponding group action. Define $a' : \mathcal{D}^{d'} \to (G \times \Theta \to \Theta)$ by $X_{d'} \mapsto (a_{t(X_{d'})} : g, \theta \mapsto \theta')$, where $t : \mathcal{D}^{d'} \to \mathcal{D}^d$ appends $d - d'$ random data points to its input. Clearly, a' satisfies the identity and associate axiom and preserves loss. Therefore, a' is a G-symmetry of $L_{d'}$. ☐

C Additional Experiment Details

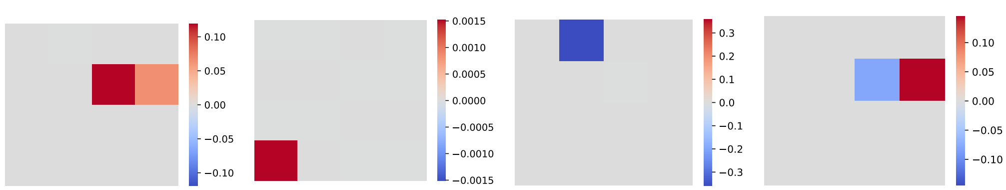

Figure 5: Learned generators for a two-layer linear MLP with parameters dimensions $W_2 \in \mathbb{R}^{1\times 2}, W_1 \in \mathbb{R}^{2\times 1}$ and data $X, Y \in \mathbb{R}$.

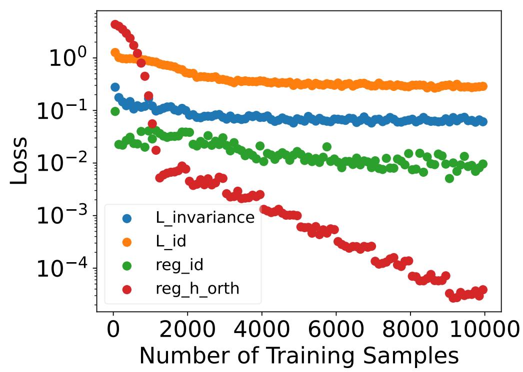

Figure 6: Training curve for learning data-dependent symmetry in a two-layer sigmoid MLP with parameters dimensions $W_2 \in \mathbb{R}^{3\times1}, W_1 \in \mathbb{R}^{3\times3}$ and data $X \in \mathbb{R}^{1\times3}, Y \in \mathbb{R}^{1\times1}$.