A Unified Framework for Discovering Discrete Symmetries

Pavan Karjol
Indian Institute of Science
Bangalore, India

Rohan Kashyap
Indian Institute of Science
Bangalore, India

Aditya Gopalan
Indian Institute of Science
Bangalore, India

Prathosh A.P.
Indian Institute of Science
Bangalore, India

Abstract

We consider the problem of learning a function respecting a symmetry from among a class of symmetries. We develop a unified framework that enables symmetry discovery across a broad range of subgroups including locally symmetric, dihedral and cyclic subgroups. At the core of the framework is a novel architecture composed of linear and tensor-valued functions that expresses functions invariant to these subgroups in a principled manner. The structure of the architecture enables us to leverage multi-armed bandit algorithms and gradient descent to efficiently optimize over the linear and the tensor-valued functions, respectively, and to infer the symmetry that is ultimately learnt. We also discuss the necessity of the tensor-valued functions in the architecture. Experiments on image-digit sum and polynomial regression tasks demonstrate the effectiveness of our approach.

1 Introduction

It is well known that machine learning tasks often exhibit natural symmetries. As a result, the function to be learnt, say in a classification or regression setting, possesses additional structure in terms being invariant or equivariant to the underlying symmetry. Being able to exploit symmetry structure in the training pipeline confers benefits such as improved sample complexity, added explainability, fewer model parameters and improved generalizability. A classic case in which symmetry is leveraged is the convolutional neural network (CNN) architecture [1] that intrinsically expresses equivariance to translations of input images in classification tasks.

A growing body of work has addressed the problem of incorporating known symmetries into the learning pipeline, either via augmenting data using the symmetry structure [2] or designing neural nets that inherently express functions with known symmetries [3, 4]. Consequently, it is known how to design architectures with n inputs that are, say, invariant to arbitrary permutations of the input variables, or equivalently, neural functions that are S_n-invariant where S_n is the group of permutations on n elements [5].

However, there are often settings in which the target function possesses a symmetry which is a priori *unknown*, but known to belong to a class of possible symmetries (subgroups of S_n). We are interested in the problem of discovering such an unknown symmetry automatically from data. Consider, for instance, data representing measured states of a system of multiple particles (e.g., positions, velocities, etc.), with the target function representing a physical quantity of interest depending on the state, such as potential energy. If only k of the n particles (whose identities are unknown) actually interact with each other (maybe because they are the only charged particles), then the net energy is invariant to permutations of the positions of this subset of particles alone. Here, the target function exhibits invariance with respect to the subgroup of permutations S_k associated to the position indices of these

Preprint. Under review.

k particles, which are not known upfront. On the other hand, the system's kinetic energy is unchanged under permutations of the subset of velocity parameters of the system state. In general, when the semantics of the target function and/or the input variables are unknown, then so is the underlying symmetry. A similar problem arises in computer vision as that of learning a classifier that can detect patterns or objects in an image while being invariant to local transformations or symmetries applied to specific regions or parts of the image [6, 7].

We consider the problem of learning a function $f : X \to Y$ given data $\left\{ \left(x^{(u)}, y^{(u)}\right) \right\}_{u=1}^{m}$, and given a collection of subgroups $\{G_1, G_2, \ldots\}$ of S_n[1], one of which f is invariant with respect to (i.e., $f \circ g \equiv f$ for every transformation g in some subgroup G_j). For a sufficiently rich collection of possible symmetry subgroups[2], we provide a unified and easy-to-use framework comprising of a parametric architecture together with algorithms to tune it and learn the underlying symmetry (subgroup). Our specific contributions are presented in the following subsection.

1.1 Contributions

- We introduce a general framework for discovering a variety of discrete symmetries. Our framework allows for efficiently learning functions that can be invariant to *any* locally symmetric, dihedral or cyclic subgroup using the same architecture.
- The unified architecture that forms the backbone of our framework is comprised of a novel combination of (learnable) linear and tensor-valued functions. We explicitly characterize the structure of both these transformations, in particular showing how they correspond to a variety of subgroups.

  To the best of our knowledge, this is the first unified framework to discover a wide range of discrete symmetries.
- Leveraging the specific structure of the linear transformations in our unified architecture, we devise an efficient training algorithm based on multi-armed bandits (for discrete optimization over matrices representing the learnable linear part) along with stochastic gradient descent (for continuous optimization over the nonlinear part). The bandit sampling allows for efficient search across the entire family of matrices associated to various symmetries, and, with our structural characterization, allows for interpretable results.

1.2 Related Work

1.2.1 Group Invariance

The utilization of symmetries in deep learning has garnered significant research interest in recent years [9, 10]. Within this context, [11] introduced G-equivariant neural networks as an extension of Convolutional Neural Networks (CNNs) to encompass a broader range of symmetries. In G-equivariant neural networks, the network layers demonstrate equivariance under the action of the group G, owing to the linear G-space structure of the representations. Furthermore, [12] establish convolution formulae in a more general setting, i.e., invariance under the action of any compact group and [13] delve into the application of G-CNNs on homogeneous spaces using equivariant linear maps.

1.2.2 Discrete Groups

The study of invariance to finite groups has received considerable attention in the existing literature. [4] proposed an approach that utilizes invariant polynomials to design G-invariant neural networks $f : X \to \mathbb{R}$, where X is a compact subset of R^n, achieved through a combination of a G-equivariant transformation block and the sum-product layer. They demonstrate the universality of their approach for larger and hierarchical subgroups of S_n. In a different approach, [3] introduced permutation-equivariant functions defined on sets using a decomposable representation expressed as $\rho\left(\sum_i \phi\left(x_i\right)\right)$. Motivated by these, we consider invariance under the action of subgroups of $G \leq S_n$, when the underlying subgroup is unknown.

[1]Restricting to subgroups of S_n is justified by the fact that any finite group is isomorphic to a subgroup of S_n for some n by Cayley's theorem [5].

[2]In general, if we consider all possible subgroups of S_n, then the problem of learning a specific symmetry is computationally intractable [8]

1.2.3 Automatic Symmetry Discovery

[10] presents a Lie algebra convolution network (L-conv) for constructing feedforward architectures that exhibit equivariance to arbitrary continuous groups. In a similar vein, [2] propose a different approach by parameterizing a distribution over training data augmentations, while [14] introduce a meta-learning framework that addresses symmetries through the reparameterization of network layers. Building upon the idea of establishing invariant symmetry-adapted data representations, [15] investigates the use of regularization on the representation matrix for unsupervised orbit learning.

2 Problem Setup and Proposed Solution

2.1 Mathematical Preliminaries

The group S_n is the set of all permutations on n elements along with the natural group multiplication (composition) and inverse operations. By a *symmetry* we mean a subgroup $G \leq S_n$; all groups used henceforth are assumed to be of this form. The group generated by an element g is $\langle g \rangle = \{g, g^2, g^3, \ldots\}$. We use $f \circ g$ to denote function composition: $(f \circ g)(x) = f(g(x))$.

Definition 2.1. Let $\mathcal{I} = \{i_1, \ldots, i_k\} \subset [n]$ be an index set with $i_1 < \cdots < i_k$.

- $\mathbb{Z}_{\mathcal{I}}$ is the locally cyclic group corresponding to $\mathcal{I}$, generated by the permutation $\pi \in S_n$ such that $\pi(i) = i_{\tau(j)}$ if $i = i_j$ and $\pi(i) = i$ otherwise. Here, $\tau(j) = (j \mod n) + 1$ denotes the cyclic shift operator.

- $D_{\mathcal{I}}$ is the locally dihedral group corresponding to $\mathcal{I}$, defined as $\{\pi, \pi^2, \ldots, \sigma\pi, \sigma\pi^2, \ldots\}$, where $\pi \in S_n$ is as defined above and $\sigma \in S_n$ is defined by $\sigma(i_l) = \sigma(i_{k-l+1})\ \forall l \in [k]$ (reflection about the center of $\mathcal{I}$).

- $S_{\mathcal{I}}$ is the locally symmetric group corresponding to $\mathcal{I}$, consisting of all permutations that move elements only within $\mathcal{I}$, i.e., $S_{\mathcal{I}} = \{\pi \in S_n : \pi(j) = j\ \forall j \notin \mathcal{I}\}$.

- $\mathbb{Z}_k = \mathbb{Z}_{\mathcal{I}}$; $\quad D_{2k} = D_{\mathcal{I}}$; $\quad S_k = S_{\mathcal{I}}$ with $\mathcal{I} = [k]$ (the first k elements of $[n]$).

Definition 2.2. Let $g \in S_n$. The action of g on $\mathbb{R}^n$ is the map $x \mapsto g \cdot x$ given by $(g \cdot x)_i = x_{g(i)}$ $\forall i \in [n]$.

Definition 2.3. The orbit of $x \in X$ under the action of group G is defined as $\mathcal{O}_G(x) = \{g \cdot x | g \in G\}$.

Definition 2.4. A function $f : X \to \mathbb{R}$ is said to be G-invariant, if $f(x) = f(g \cdot x), \forall g \in G, x \in X$.

Definition 2.5. Let $X, Y \subseteq \mathbb{R}^n$. A function $f : X \to Y$ is said to be G-equivariant, if for any $g \in G$, $\exists\, \tilde{g} \in G$, $f(g \cdot x) = \tilde{g} \cdot f(x), \forall x \in X$.

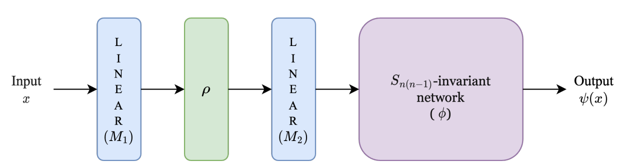

Figure 1: Proposed unified architecture for discovering symmetries, composed of linear transformations (M_1, M_2) and nonlinear functions (ρ, ϕ). ρ is explicitly fixed whereas M_1, M_2, ϕ are trainable. Theorem 4 guarantees that the architecture can express functions invariant to any locally symmetric, dihedral and cyclic. Here, ϕ is represented by a neural network and trained using gradient descent while M_1, M_2 are optimized using bandit sampling over a discrete space of matrices.

2.2 Problem statement

Let $X = [0,1]^n \subset \mathbb{R}^n$ denote the input (instance) domain. We frame the problem of symmetry discovery as follows: Given data $\left\{ \left(x^{(u)}, y^{(u)}\right) \right\}_{u=1}^{m}$ with $x^{(u)} \in X, y^{(u)} \in \mathbb{R}$, and a collection of subgroups $\mathcal{G} = \{G_1, G_2, \ldots\}$ of S_n, learn a function $f : X \to \mathbb{R}$ such that f is G-invariant for some $G \in \mathcal{G}$ with respect to the data.

2.3 Symmetry discovery framework

We aim to develop a framework for solving the symmetry discovery problem defined above in the problem statement, when the possible set of symmetries $\mathcal{G}$ can be *any* group of the form $\mathbb{Z}_{\mathcal{I}}$, $D_{\mathcal{I}}$ and $S_{\mathcal{I}}$, i.e., $\mathcal{G} = \cup_{\mathcal{I}\subseteq[n]}\{\mathbb{Z}_{\mathcal{I}}, D_{\mathcal{I}}, S_{\mathcal{I}}\}$. It is not a priori clear how to efficiently search over the function class $\mathcal{F}(\mathcal{G})$ – observe that $\mathcal{G}$ is an exponentially large (in n) set of subgroups.

Our solution strategy is based on finding a standard decomposition for any function ψ in the function class $\mathcal{F}(\mathcal{G})$. To this end, we first consider each type of subgroup individually and prove a structural decomposition of the form $\psi = \phi \circ \rho$ for any ψ which is invariant to that group. We then design a single decomposition of the form $\phi \circ M_2 \circ \rho \circ M_1$ that effectively integrates all the individual decompositions.

Our first result shows that any $\mathbb{Z}_k$-invariant function can be expressed as a composition of an S_k-invariant function and a specific tensor-valued function.

Theorem 1. *Let $\psi : [0,1]^k \to \mathbb{R}$ be $\mathbb{Z}_k$-invariant. There exists an S_k-invariant function $\phi : \mathbb{R}^k \to \mathbb{R}$ such that*

$$\psi = \phi \circ \rho, \tag{1}$$

where

$$\rho : [x_1, x_2, \ldots x_k]^T \mapsto [(x_1, x_2), (x_2, x_3), \ldots, (x_{k-1}, x_k), (x_k, x_1)]^T. \tag{2}$$

Proof. (Sketch) The $\mathbb{Z}_k$-invariant function ψ must assign the same value to every element of any $\mathbb{Z}_k$-orbit. We show that any such orbit $\mathcal{O}_{\mathbb{Z}_k}(x)$ can be uniquely associated with the corresponding S_k-orbit $\mathcal{O}_{S_k}(\rho(x))$. From this, it follows that by defining the S_k-invariant function ϕ to take the same value across any orbit of the form $\mathcal{O}_{S_k}(\rho(x))$ as ψ does across the orbit $\mathcal{O}_{\mathbb{Z}_k}(x)$ (and an arbitrary value across orbits not of the form $\mathcal{O}_{S_k}(\rho(x))$), we obtain the result.

We also assess the regularity conditions such as smoothness (C^∞) and continuity (C^0) of the ψ and ϕ function, and in this regard we state the following theorem.

Theorem 2. *The ϕ function is smooth (C^∞) whenever ψ function is C^∞. Similarly, the ϕ function is continuous (C^0) whenever ψ function is C^0.*

We now state the following lemma, to prove Theorem 2.

Lemma 3. The tensor-valued function ρ is a diffeomorphism between X and its image $\rho(X)$, where $X = [0,1]^k$.

Proof. To prove the claim, we need to endow $Y = \rho(X)$ with a topology. First, we observe that, for any $y = [(y_1, y_2), (y_2, y_3) \ldots, (y_k, y_1)]^T$ it can be written as a vector of the form $[y_1, y_2, y_2, y_3, y_3, \ldots y_k, y_k, y_1]^T \in \mathbb{R}^{2k}$. Thus we can employ subspace topology of the standard topology of $\mathbb{R}^{2k}$. It is obvious to see that ρ is bijective with ρ^{-1} defined as:

$$[(y_1, y_2), (y_2, y_3), \ldots, (y_k, y_1)]^T \mapsto [y_1, y_2, \ldots y_k]^T$$

Thus, since ρ and ρ^{-1} are smooth with respect to the subspace topology, ρ is a diffeomorphism. □

Proof. From 1, we have $\psi = \phi \circ \rho$ and thus, $\psi \circ \rho^{-1} = \phi$.

From Lemma 3, ρ^{-1} is smooth (C_∞) since ρ is a diffeomorphism. Thus, if ψ is a continuous function (C^0), then ϕ is composition of smooth function with a C^0 function which in turn implies composition of two C^0 functions. Thus ϕ is C^0. Similarly, if ψ is C^∞, then ϕ is a composition of C^∞ functions. Thus ϕ is C^∞. □

Results of the same form as Theorem 1 hold for ψ being a D_{2k}- or S_k-invariant function by replacing the definition of the function ρ with the appropriate definition in Table 1.

We now state our main result, which is a *single* canonical functional decomposition that includes functions invariant to all the subgroups of type $\mathbb{Z}_{\mathcal{I}}$, $S_{\mathcal{I}}$ and $D_{\mathcal{I}}$, in Theorem 4. The key idea is to introduce 'selection' matrices that appropriately reduce a general function to the specific type of subgroup as in Theorem 1 ($\mathbb{Z}_k$, D_{2k} or S_k).

Subgroup	S_k	$\mathbb{Z}_k$	D_{2k}
$\rho(x)$	$\begin{bmatrix} \vdots \\ (x_i, x_j) \\ \vdots \end{bmatrix}_{i,j\in[k],i\neq j}$	$\begin{bmatrix} \vdots \\ (x_i, x_{\tau(i)}) \\ \vdots \end{bmatrix}_{i\in[k]}$	$\begin{bmatrix} \vdots \\ (x_i, x_{\tau(i)}) \\ (x_{\tau(i)}, x_i) \\ \vdots \end{bmatrix}_{i\in[k]}$

Table 1: Subgroups of S_n and corresponding definitions of the tensor-valued function ρ, where τ is cyclic right shift by 1 element.

Theorem 4 (Unified symmetry discovery framework)**.** *Let $\mathcal{B}$ denote the class of all functions from $[0,1]^n \to \mathbb{R}$ of the form:*

$$x \mapsto \phi\left(\begin{bmatrix} (M_2 \circ \rho \circ M_1)(x) \\ (I - M_1)(x) \end{bmatrix}\right)$$

where,

- *$M_1 : \mathbb{R}^n \to \mathbb{R}^n$ and $M_2 : \mathbb{R}^{n(n-1)} \to \mathbb{R}^{n(n-1)}$ are linear transformations (i.e., matrices),*
- *ϕ is an $S_{n(n-1)}$-invariant function, and*
- *$\rho : \mathbb{R}^n \to \mathbb{R}^{n(n-1)}$ is a tensor-valued function $\rho : [x_1, \ldots, x_n]^T \mapsto \left[(x_i, x_j)_{i,j\in[n],i\neq j}\right]^T$.*

Let $\mathcal{I} = \{i_1, i_2, \ldots i_k\} \subseteq [n]$. Then, the following hold:

a) Any $S_{\mathcal{I}}$-invariant function belongs to $\mathcal{B}$. Moreover, the matrices M_1 and M_2 in its decomposition have the forms:

$$M_1[u, v] = \begin{cases} 1, & \text{if } u \in [k] \text{ and } v = i_u \\ 0, & \text{otherwise,} \end{cases} \tag{3}$$

$$M_2 = I_{n(n-1)\times n(n-1)}. \tag{4}$$

b) Any $\mathbb{Z}_{\mathcal{I}}$-invariant function belongs to $\mathcal{B}$. Moreover, M_1 is of the form as given in (3) *and M_2 is as follows:*

$$M_2[i, j] = \begin{cases} 1, & \text{if } i \in [k] \text{ and } (\rho \circ M_1)(x)[j] = (x_i, x_{\tau(i)}) \\ 0, & \text{otherwise}. \end{cases} \tag{5}$$

c) Any $D_{\mathcal{I}}$-invariant function belongs to $\mathcal{B}$. Moreover, M_1 is of the form as given in (3) *and M_2 is as follows:*

$$M_2[i, j] = \begin{cases} 1, & \text{if } i \in [2k] \text{ and } (\rho \circ M_1)(x)[j] = (x_i, x_{\tau(i)}) \\ 1, & \text{else if } i \in [2k] \text{ and } (\rho \circ M_1)(x)[j] = (x_i, x_{\tau(i)}) \\ 0, & \text{otherwise}. \end{cases} \tag{6}$$

Note that the function ρ above is the same as the one for S_k in Table 1 but with $k = n$.

Proof. (Sketch) We prove the result for $\mathcal{I} = [k]$, since for any other $\mathcal{I}$ (i.e., k indices), a simple modification for M_1 (composition with a suitable permutation matrix) works. From Theorem 1, we see that, the goal is to show that $\phi \circ M_2 \circ \rho \circ M_1$ (with ϕ being $S_{n(n-1)}$-invariant and ρ corresponding to S_n) is equivalent to $\phi \circ \rho$ (with ϕ being S_k-invariant (similarly for $\mathbb{Z}_k$ or D_{2k}) and ρ corresponding to S_k (similarly for $\mathbb{Z}_k$ or D_{2k})). This is achieved via appropriately choosing M_1 and M_2. The M_1 helps in selecting appropriate indices over which the subgroup acts and M_2 helps in identifying the broader category (symmetric, cyclic or dihedral) of the subgroup.

Figure 1 depicts the unified architecture stated in Theorem 4, along with the method to train it (described in Section 2.4).

We remark that Theorem 4 can be extended to express functions invariant to wider classes of subgroups. The following results offer a glimpse of how this can be achieved, for instance, for product groups.

Theorem 5 (Invariance to product groups)**.** *Let* $[n] = \bigcup_{j=1}^{L} \mathcal{I}_j$ *be a partition of* $[n]$, $G_i \in \{S_{\mathcal{I}_j}, D_{\mathcal{I}_j}, \mathbb{Z}_{\mathcal{I}_j}\}, \forall j \in [L]$ *and* $G = G_1 \times G_2 \times \cdots G_L$ *such that no two groups* G_i, G_j *are isomorphic. Let* ψ *be a G-invariant function, then there exists an* S_l*-invariant function* ϕ *and a specific tensor-valued function* ρ*, such that,*

$$\psi = \phi \circ \rho. \tag{7}$$

Proof. (Sketch) Let the ρ function be defined as the one outputting all the appropriate monomials of the form $x_i x_j^2$ corresponding to individual components of the product group G. Then ρ is injective and G-equivariant. Note that, here l equals to the total number of all the appropriate monomials. The remaining steps are similar to the ones of Theorem 1.

Corollary 5.1. *Let* $\sigma \in S_n$ *and* $G = \langle \sigma \rangle$ *such that whose disjoint cycles have unique lengths. Let* ψ *be a G-invariant function, then there exists an* S_l*-invariant function* ϕ *and a specific tensor-valued function* ρ*, such that,* $\psi = \phi \circ \rho$.

Proof. We use the fact that any permutation σ can be decomposed into disjoint cycles. Hence $G = \mathbb{Z}_{\mathcal{I}_1} \times \mathbb{Z}_{\mathcal{I}_2} \cdots \times \mathbb{Z}_{\mathcal{I}_L}$ with no two $\mathbb{Z}_{\mathcal{I}_k}, \mathbb{Z}_{\mathcal{I}_l}$ are isomorphic (because the lengths are different). Applying Theorem 5, we prove the claim.

2.4 Optimization for discovering symmetries

Having proposed, via Theorem 4, a common functional form ($\phi \circ M_2 \circ \rho \circ M_1$) for any function invariant to symmetries of type $\mathbb{Z}_{\mathcal{I}}$, $D_{\mathcal{I}}$ or $S_{\mathcal{I}}$, we turn to methods to fit the functional form to data (**??**) and discover the underlying symmetry.

A straightforward approach is to employ standard stochastic gradient descent (SGD)-type optimization jointly over ϕ, parameterized as a neural network, and M_1, M_2, parameterized as matrices in $\mathbb{R}^{n \times n}$ and $\mathbb{R}^{n(n-1) \times n(n-1)}$, respectively. However, in view of the discrete structure of M_1, M_2 prescribed explicitly by Theorem 4 (equations (3)-(6)), we resort to multi-armed bandit sampling to learn the best (M_1, M_2) pair in an 'outer loop', with SGD over ϕ running in the 'inner loop'. Specifically, each arm of the bandit corresponds to a (M_1, M_2) pair, and the reward for it is the negative of the loss that SGD over ϕ obtains for that pair. This approach is advantageous for two reasons: (i) It confers interpretability in the sense that the underlying symmetry can be directly read off from the M_1, M_2 which is ultimately learnt by the bandit outer loop, (ii) A bandit algorithm over (M_1, M_2) performs global optimization and avoids the potential pitfalls of using gradient descent that could get stuck in local optima.

Linear Thompson Sampling (LinTS)-based bandit optimization algorithm: Observe that although the space of matrices (M_1, M_2) guaranteed by Theorem 4 is discrete, it is still an exponentially large set. To enable efficient search over this set, we resort to using the linear parametric Thompson sampling algorithm (LinTS) [16]. In this strategy, whose pseudo code appears in Algorithm 1, each possible pair of matrices (M_1, M_2), denoting an arm of the bandit, is represented uniquely by a *binary* feature vector of an appropriate dimension d (described in detail below). The reward from playing an arm with feature vector a (which is the negative loss after optimizing for ϕ using SGD) is assumed to be linear in a with added zero-mean noise, i.e., $\exists \mu^\star \in \mathbb{R}^d$ such that the expected reward upon playing a is $a^\top \mu^\star$. LinTS maintains and iteratively updates a (Gaussian) probability distribution (lines 9, 12 and 13) over the unknown reward model $\mu^\star$, and explores the arm space by sampling from this probability distribution in each round (line 7).

Using LinTS for exploring across (M_1, M_2) is advantageous for several reasons. The chief one is that even though the arm set of binary vectors, representing all possible M_1, M_2 matrices, is exponentially large (of cardinality $O(3 \cdot 2^n)$), finding the arm maximizing the reward for a sampled vector μ (line 8) is a constant-time operation. Another reason to prefer LinTS as a search strategy is that it enjoys a rigorous guarantee on the probability of error in finding the best arm in a true linear model, as we show in Theorem 6 below.

Features for bandit arms: To specify the feature vector for each bandit arm, we employ one-hot encoding to represent the general subgroup category in the order given as, locally symmetric, dihedral, and cyclic respectively. An n-dimensional vector is utilized to represent the corresponding indices,

Algorithm 1: Linear Parametric Thompson Sampling for Subgroup Discovery

1 **Initialize:** $\mathcal{A} \subset \{0,1\}^d$ (arm set: binary feature vectors representing each pair of matrices (M_1, M_2)),
2 $B \leftarrow I_d$ (prior covariance),
3 $f \leftarrow 0 \in \mathbb{R}^d$, $\hat{\mu} \leftarrow 0 \in \mathbb{R}^d$ (prior mean),
4 $\nu > 0$ (variance inflation parameter),
5 T (time horizon).
6 **for** $t \in \{1, 2, \ldots, T\}$ **do**
7 Sample μ independently from $\mathcal{N}\left(\hat{\mu}, \nu^2 B^{-1}\right)$
8 $a \leftarrow \arg\max_{a' \in \mathcal{A}} \mu^\top a'$
9 $B \leftarrow B + aa^\top$
10 Fix matrices M_1, M_2 in the architecture as per a, and run SGD over ϕ with loss function $L(\phi) = \frac{1}{m} \sum_{u=1}^{m} \ell\left(y^{(u)}, (\phi \circ M_2 \circ \rho \circ M_1)\left(x^{(u)}\right)\right)$ to obtain $\tilde{\phi}$
11 Set reward from arm a: $\gamma \leftarrow -L(\tilde{\phi})$
12 $f \leftarrow f + a\gamma$
13 $\hat{\mu} \leftarrow B^{-1} f$
14 **end**
15 **return** $A_T = \arg\max_{a \in \mathcal{A}} a^\top \hat{\mu}$ (best arm for the estimated linear model)

where the indices pertaining to the subgroup category are set to 1, while the remaining indices are set to 0. Subsequently, this vector can be concatenated with a one-hot encoded representation of the subgroup category. For example, with $n = 10$, $G = \mathbb{Z}_{\mathcal{I}}$, and $\mathcal{I} = \{3, 5, 6, 8\}$ the overall feature vector is given as follows:

$$a = [0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1]^T.$$

The first n indices (in blue) above correspond to the actual indices, while the last three indices (in red) indicate the respective subgroup type.

Our next result is a performance guarantee for the LinTS algorithm (Algorithm 1), showing a bound on its probability of misidentifying the optimal arm in a linear reward model.

Theorem 6 (Error probability bound for LinTS)**.** *Let the set of arms $\mathcal{A} \subset \mathbb{R}^d$ be finite. Suppose that the reward from playing an arm $a \in \mathcal{A}$ at any iteration, conditioned on the past, is sub-Gaussian with mean[3] $a^\top \mu^\star$. After T iterations, let the guessed best arm A_T be drawn from the empirical distribution of all arms played in the T rounds, i.e., $\mathbb{P}[A_T = a] = \frac{1}{T}\sum_{t=1}^{T} \mathbf{1}\{a^{(t)} = a\}$ where $a^{(t)}$ denotes the arm played in iteration t. Then,*

$$\mathbb{P}[A_T \neq a^\star] \leq \frac{c \log(T)}{T},$$

where $c \equiv c\left(\mathcal{A}, \mu^\star, \nu\right)$ is a quantity that depends on the problem instance $(\mathcal{A}, \mu^\star)$ and algorithm parameter (ν).

Note that the rule for guessing the best arm A_T at the end of the time horizon is slightly different compared to that of Algorithm 1[line 15]. This result is derived by appealing to a standard reduction between cumulative regret and simple regret for the empirical distribution-based guessing rule [17]. This is then combined with a recent logarithmic bound for the cumulative regret for LinTS [18] on one hand, along with an inequality relating simple regret to the probability of misidentifying the best arm on the other, to obtain the result (the explicit form of c appears in the appendix). We are unaware of any prior result that bounds the identification error probability of linear parametric Thompson sampling, so this result may be of independent interest.

Alternative optimization algorithms: Instead of linear Thompson sampling and gradient descent, one could choose a variety of methods to optimize the unified architecture across the functions M_1, M_2 and ϕ, depending on practical considerations. We have already mentioned the possibility

[3]A random variable X is said to be sub-Gaussian with mean β if $\mathbb{E}[e^{t(X-\beta)}] \leq e^{t^2/2}$.

of using gradient-based optimization jointly across all three functions. On the other end, one can employ global optimization methods such as Bayesian optimization [19] for the continuous space of ϕ, along with multi-armed bandits for M_1, M_2 as we have done here. Of course, even the design of adaptive discrete sampling algorithms for finding the best M_1, M_2 is open to a wide variety of possibilities, including best arm identification algorithms for linear bandits [20], simulated annealing [21] and evolutionary algorithms [22], to name just a few.

3 Discussion

The work introduced by [23] can be considered as a specific instance of our work, when ρ is an identity function, in which the resulting architecture is a composition of an $S_{n(n-1)}$-invariant function and a linear transformation. In this section, we formally analyze the limitations associated with such an approach and establish the non-realizability of $\mathbb{Z}_k$-invariant functions using S_k-invariant functions and a linear transformation for $k \geq 3$.

Theorem 7. *Consider the following set of functions, for $k \geq 3$:*

$$\mathcal{A}_k = \left\{ \phi \circ M \middle| M \text{ is linear transformation from } \mathbb{R}^k \text{ to } \mathbb{R}^k \text{ and } \phi \text{ is } S_k - \text{invariant function} \right\}.$$

There exists a $\mathbb{Z}_k$-invariant function ψ such that $\psi \notin \mathcal{A}_k$.

Proof. (Sketch) We show the non-realizability of a $\mathbb{Z}_k$-invariant function which has a unique value for each orbit. We have, $\left|\mathcal{O}_{\mathbb{Z}_k}(x)\right| \leq k$. Suppose $\psi = \phi \circ M$, then M has to be invertible. Then, $\exists \tilde{x}$ such that $\left|\mathcal{O}_{S_k}(M\tilde{x})\right| = k!$, which leads to a contradiction.

We now conjecture a similar result for $\mathbb{Z}_k$-invariant functions for $n \geq k \geq 3$.

Conjecture 8. *Consider the following set of functions, for $n \geq 3$ and $k \leq n$,*

$$\mathcal{A}_n = \left\{ \phi \circ M \middle| M \text{ is a linear transformation and } \phi \text{ is } S_n - \text{invariant function} \right\}$$

Then, $\exists$ a $\mathbb{Z}_k$-invariant function ψ such that $\psi \notin \mathcal{A}_n$.

By employing tensor-valued functions as in Theorem 1, we gain additional flexibility, allowing us to overcome the above limitations.

Canonical form. The proposed architecture utilizes a common ϕ i.e., an $S_{n(n-1)}$-invariant network, while the work proposed in [23] requires ϕ be modified depending on the subgroup type. Moreover, our framework yields a canonical form for our overall architecture, as illustrated for the $\mathbb{Z}_\mathcal{I}$ subgroup, given as:

$$(\phi \circ M_2 \circ \rho \circ M_1)(x) = \mu \left(\sum_{i_l \in I} \eta \left(x_{i_l} x_{\tau(i_l)} \right) + C_1 \eta(0) \right),$$

where C_1 is a constant, and μ, η denote specific functions. This follows from the canonical form of ϕ as proved in [3]. Similar results can be obtained for $S_\mathcal{I}$ and $D_\mathcal{I}$ subgroups. This allows for a simple implementation of our architecture for various applications.

Handling non-divisors of n. We emphasize that the work proposed by [23] for learning $\mathbb{Z}_\mathcal{I}$ (or $D_\mathcal{I}$) symmetries is applicable only when $k|n$. In contrast, our framework allows for the discovery of subgroups of type $\mathbb{Z}_\mathcal{I}$ (or $D_\mathcal{I}$) for any $|\mathcal{I}\big| = k \leq n$, thus allowing a larger class of subgroups.

4 Experiments

We assess the performance of our proposed method in two representative tasks that have been considered in previous related work [4, 3, 23], one on synthetically generated data (polynomial regression) and the other on a real-world image dataset (image-digit sum).

4.1 Polynomial Regression

In this task, we conduct the model training to learn a G-invariant polynomial as studied in [4]. For example, with $n = 5, k = 4$; $f(x) = x_1x_2x_3x_4 + x_5$ is an S_4-invariant polynomial function. Note

Task	G	Accuracy
Polynomial Regression	$\mathbb{Z}_{\mathcal{I}}$	100
Polynomial Regression	$D_{\mathcal{I}}$	100
Image-Digit Sum	$S_{\mathcal{I}}$	100

Table (1.a): Accuracy (%)

G	$\mathbb{Z}_{\mathcal{I}}(5)$	$\mathbb{Z}_{\mathcal{I}}(7)$	$D_{\mathcal{I}}(5)$	$D_{\mathcal{I}}(7)$
$\mathbb{Z}_{\mathcal{I}}$	**4.2**	**6.1**	8.2	15.2
$D_{\mathcal{I}}$	4.7	7.9	**6.3**	**10.1**
$S_{\mathcal{I}}$	11.7	18.5	21.3	34.3
$M + H$-*INV*	12.3	-	23.2	-
SGD	14.4	17.7	26.5	34.4

Table (1.b): MAE ($\times 10^{-2}$)

Table (1): **(a)** Estimation accuracy (top 3) for subgroup discovery in polynomial regression and image-digit sum tasks. **(b)** Mean absolute error ($\times 10^{-2}$) for the regression task with $\mathbb{Z}_{\mathcal{I}}$ and $D_{\mathcal{I}}$ subgroups. The cardinality ($k = |\mathcal{I}|$) of the index set is given in braces. The first three rows display the top 3 bandit arm subgroups, with the actual subgroup results highlighted in bold. The $M + H$-INV (only applicable for $k|n$) represents the subgroup discovery method proposed by [23], which incorporates a composite of linear transformations and an H-invariant network. Here, $H \leq S_n$ is dependent on the underlying subgroup. The last row represents the proposed architecture entirely trained with SGD.

that we also study numerous polynomials of various degrees and give detailed definitions of the polynomials in the supplementary section. To examine the generalization abilities of the proposed method we use only 64 randomly generated points in $[0, 1]$ for training, whereas use 480 and 4800 points for validation and test sets respectively.

4.2 Image-Digit Sum

The goal of this task is to learn the function representing the sum of digit labels of k (out of n) images. An input is a set of n images of dimension 28×28 taken from MNISTm dataset ([24]). Using the proposed bandit setting, we discover the underlying subgroup (in this case $S_{\mathcal{I}}$). Note that, x_i is an image (or 2D matrix), instead of scalar element.

4.3 Results

Table (1.a) presents the accuracies achieved in subgroup discovery tasks for image-digit sum ($S_{\mathcal{I}}$) and polynomial regression ($\mathbb{Z}_{\mathcal{I}}$ and $D_{\mathcal{I}}$). The reported accuracies correspond to different values of k within the range $[n]$, where $n = 10$, and are based on randomly selected index sets $\mathcal{I}$. These accuracies indicate the successful identification of the underlying subgroup within the top 3 bandit arms, as determined by the final $\hat{\mu}$. The training process achieves this outcome within $T = O(n)$ iterations.

For the polynomial regression task, we also provide the mean absolute error (MAE) values for the top 3 bandit arms obtained. Notably, the MAE corresponding to the actual subgroup is the lowest, indicating successful discovery of the actual subgroup within the top 3. It is worth mentioning that the loss values observed for $\mathbb{Z}_{\mathcal{I}}$ and $D_{\mathcal{I}}$ subgroups are relatively close, as the only additional group symmetries are the reflections. In addition, we consider the proposed architecture entirely trained with SGD. Our results consistently demonstrate a significant performance improvement over the SGD method across all investigated subgroups in the polynomial regression tasks. Furthermore, we compare our approach with the subgroup discovery method proposed by [23], which combines linear transformations and an invariant network specifically designed for each subgroup type.

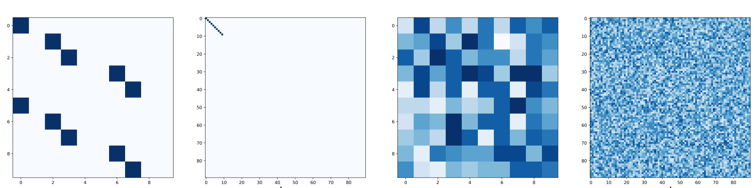

Figure 2: Visualization of the reference (bandit) M_1 (a) and M_2 (b) matrices, as well as those (c, d) obtained through training our method entirely using SGD for the task of polynomial regression of $\mathbb{Z}_{\mathcal{I}}$-invariant function, with $n = 10$ and $\mathcal{I} = \{0, 2, 3, 6, 7\}$.

4.4 Interpretability

Bandit sampling inherently yields interpretable outcomes, and an illustrative example (M_1, M_2) of this is demonstrated in Figure 2 (a, b). Conversely, training our method solely using SGD results in matrices that lack clear characterization of the underlying subgroup, as depicted in Figure 2 (c, d).

4.5 Limitations and Conclusion

This work introduces a novel framework for the discovery of discrete symmetry groups. We employ neural architectures trained using a combination of gradient descent and bandit sampling, resulting in interpretable outcomes. Through experiments on both synthetic and real-world datasets, we demonstrate the effectiveness of our approach. It is important to note that this work primarily focuses on theoretical aspects and serves as a proof of concept. In the future, we plan to explore similar approaches for addressing continuous groups and their corresponding applications.

5 Appendix

5.1 Multi-Armed Bandits

The Multi-Armed Bandit (MAB) framework is a classical approach for sequential decision-making problems, in which an agent $\mathcal{A}$ selects actions (arms) to minimize the total regret given by $R_T = T\lambda^* - \mathbb{E}\left[\sum_{t=1}^{T} R_t\right]$ where λ^* is the mean reward of the optimal arm.

Thompson sampling is a Bayesian approach to the multi-armed bandit problem. It works by sampling from a posterior distribution over the expected rewards of each arm, and then selecting the arm with the highest sampled reward. The posterior distribution is updated after each round of play, based on the observed rewards. In this setting, each arm (action) is associated with a context or feature vector x, and the goal is to learn a linear model that predicts the expected reward for each arm given its context. Let X_t be the context vector at time t, A_t be the chosen arm at time t, and R_t be the observed reward at time t. The algorithm assumes a prior distribution over the model parameters μ (e.g., multivariate Gaussian distribution). At each iteration, Thompson Sampling samples a parameter vector μ from the posterior distribution. Then, it estimates the expected reward for each arm by computing the inner product between the sampled μ and the corresponding context vector x. The arm with the highest estimated reward is chosen and pulled. After observing the reward, the posterior distribution is updated using Bayesian inference to obtain a new posterior distribution, taking into account the new data. This update process is typically performed using conjugate priors or approximate methods like Markov Chain Monte Carlo (MCMC) or variational inference. The algorithm continues to update the posterior distribution and select arms based on the sampled parameters, enabling it to learn the optimal policy in a contextual bandit setting.

Thompson Sampling has been proven to be asymptotically optimal, meaning that as $T \to \infty$, the regret of the algorithm is bounded by a logarithmic function of T. Formally, it has been shown that $\lim_{T\to\infty} \frac{R_T}{T} = 0$, where R_T represents the regret after T rounds. This result guarantees that over time, Thompson Sampling converges to the optimal arm and achieves maximum total reward. The logarithmic regret bound demonstrates the efficiency of the algorithm in balancing exploration and exploitation, leading to near-optimal performance in the long run.

5.2 Additional Experiments

Table 5: Estimation Accuracy $(\%)$

Task	G	Accuracy
Convex Area	$D_{\mathcal{I}}$	100
$S_{\mathcal{I}}$ *(4)*	$S_{\mathcal{I}}$	100

Table 5 presents the accuracies (top 3) achieved in subgroup discovery tasks on two tasks: (i) convex quadrangle area estimation. (ii) $S_{\mathcal{I}}$-invariant polynomial regression. The cardinality ($k = |\mathcal{I}|$) of the index set is given in braces.

Convex area estimation. In this task, we estimate the area of convex quadrilaterals which are invariant to cyclic shifts and reflections of the input coordinates, i.e., a $D_{\mathcal{I}}$-invariant function ($|\mathcal{I}| = 4$). The input is the (x, y) coordinates of the four points of the quadrilateral lying in $\mathbb{R}^{4\times 2}$. The training data consists of 256 examples (randomly generated convex quadrangles with their areas), while the validation dataset contains 1024 examples. Note that, the coordinates are randomly sampled from $[0, 2]$ and the area takes value in $(0, 1]$ respectively.

Polynomial regression. Here, we consider $S_{\mathcal{I}}$-invariant polynomial regression task. The training dataset consists of 64 randomly generated data points in $[0, 1]$, whereas 480 points were used for the validation set.

For all our experiments, we observe the subgroup discovery in $O(n)$ iterations. At each iteration, we run the model for 400 epochs (3 for image-digit sum) with batch size of 16 and decaying learning rate schedule on *NVIDIA A6000 GPU's*. We report the accuracy obtained across 5 trails with different index set I.

Table 6: Definition of Polynomials

INVARIANCE	POLYNOMIAL
$S_{\mathcal{I}}$ (4)	$x_1x_2x_3x_4 + x_5$
$\mathbb{Z}_{\mathcal{I}}$ (5)	$x_1x_2^2 + x_2x_3^2 + x_3x_6^2 + x_6x_7^2 + x_7x_1^2$
$\mathbb{Z}_{\mathcal{I}}$ (7)	$x_1x_2^2 + x_2x_3^2 + x_3x_6^2 + x_6x_7^2 + x_7x_9^2 + x_9x_{10}^2 + x_{10}x_1^2$
$D_{\mathcal{I}}$ (5)	$x_1x_2^2 + x_2x_3^2 + x_3x_6^2 + x_6x_7^2 + x_7x_1^2 + x_1x_7^2 + x_7x_6^2 + x_6x_3^2 + x_3x_2^2 + x_2x_1^2$
$D_{\mathcal{I}}$ (7)	$x_1x_2^2 + x_2x_3^2 + x_3x_6^2 + x_6x_7^2 + x_7x_9^2 + x_9x_{10}^2 + x_{10}x_1^2 + x_1x_{10}^2 + ... + x_2x_1^2$

Table (6): The exact definitions of the polynomials used in experiments is given in Table 6. For $\mathbb{Z}_{\mathcal{I}}$ and $D_{\mathcal{I}}$ the input is a vector in $[0,1]^{10}$ given as; $x = [x_1, x_2, ..., x_{10}]$ whereas for $S_{\mathcal{I}}$ it is a vector in $[0,1]^5$ given as; $x = [x_1, x_2, ..., x_5]$. In this example, the index set $\mathcal{I}$ is chosen to be $[1,2,3,4]$, $[1,2,3,6,7]$, and $[1,2,3,6,7,9,10]$ respectively.

Proposition 1 (Cayley's Theorem)**.** *Let G be a group, and let H be a subgroup. Let G/H be the set of left cosets of H in G. Let N be the normal core of H in G, defined to be the intersection of the conjugates of H in G. Then the quotient group G/N is isomorphic to a subgroup of $Sym(G/H)$. More specifically, it states that every group G is isomorphic to a subgroup of the symmetric group.*

6 Proof of Theorem 1

Theorem 1. *Let $\psi : [0,1]^k \to \mathbb{R}$ be $\mathbb{Z}_k$-invariant. There exists an S_k-invariant function $\phi : \mathbb{R}^k \to \mathbb{R}$ such that*

$$\psi = \phi \circ \rho, \tag{1}$$

where

$$\rho : [x_1, x_2, \dots x_k]^T \mapsto [(x_1, x_2), (x_2, x_3), \dots, (x_{k-1}, x_k), (x_k, x_1)]^T. \tag{2}$$

Proof. **Step 1**: First, we show that the $\rho : X \to \mathbb{R}^k$ is an injective function, where $X = [0,1]^k$. Suppose $\rho(x) = \rho(y)$, for some $x = [x_1, x_2, \dots x_k]^T$ and $y = [y_1, y_2, \dots y_k]^T$. Then,

$$[(x_1, x_2),\ (x_2, x_3),\ \dots,\ (x_k, x_1)]^T = [(y_1, y_2),\ (y_2, y_3),\ \dots,\ (y_k, y_1)]^T, \tag{8}$$

thus,

$$(x_1, x_2) = (y_1, y_2),\ (x_2, x_3) = (y_2, y_3),\ \dots,\ (x_{k-1}, x_k) = (y_{k-1}, y_k),\ (x_k, x_1) = (y_k, y_1). \tag{9}$$

Thus, we get, $x_i = y_i,\ \forall i \in [k]$. Hence, ρ is injective.

In addition, $\rho^{-1} : \rho(X) \to X$ is given by

$$\rho^{-1}\left([(x_1, x_2), (x_2, x_3), \dots (x_k, x_1)]^T\right) = [x_1, x_2, \dots x_k]^T \tag{10}$$

Step 2: It is obvious to see that ρ is a $\mathbb{Z}_k$-equivariant function, i.e.,

$$\rho(h \cdot x) = h \cdot \rho(x), \quad \forall h \in \mathbb{Z}_k \tag{11}$$

Step 3: We now show that, for any $g \in S_k$, $g \cdot \rho(x) \in$ Im(ρ) if and only if $g \in \mathbb{Z}_k$. In other words, only cyclic shifts of any vector $\rho(x)$ lie in the image of ρ.

From Step 2, we get that, if $g \in \mathbb{Z}_k$, then $g \cdot \rho(x) = \rho(g \cdot x)$. Thus, $g \cdot \rho(x) \in Im(\rho)$.

Suppose $g \cdot \rho(x) \in Im(\rho)$ for some $g \in S_k$. Since $\rho(x) \in Im(\rho)$, we have

$$\begin{aligned}
\rho(x) &= [(x_1, x_2),\ (x_2, x_3),\ \dots\ (x_k, x_1)]^T \\
g \cdot \rho(x) &= [(x_{g(1)}, x_{\tau(g(1))}), \quad (x_{g(2)}, x_{\tau(g(2))}), \quad \dots\ (x_{g(k)}, x_{\tau(g(k))})]^T \\
\rho^{-1}(g \cdot \rho(x)) &= [x_{g(1)},\ x_{g(2)},\ \dots \quad x_{g(k)}]^T \quad (g \cdot \rho(x) \in Im(\rho) \text{ and applying (10)}) \\
\rho(\rho^{-1}(g \cdot \rho(x))) &= [(x_{g(1)}, x_{g(2)}), \quad (x_{g(2)}, x_{g(3)}), \quad \dots \quad (x_{g(k)}, x_{g(1)})]^T \\
&= g \cdot \rho(x)
\end{aligned} \tag{12}$$

where τ is cyclic shift operator defined as $\tau(j) = (j \mod k) + 1$. Thus,

$$g(2) = \tau(g(1)),\ g(3) = \tau(g(2)) \dots \dots g(1) = \tau(g(k)) \tag{13}$$

Hence, g is a cyclic shift, i.e., $g \in \mathbb{Z}_k$

Step 4: Claim: The following map is injective:

$$\mathcal{O}_{\mathbb{Z}_k}(x) \mapsto \mathcal{O}_{S_k}\left(\rho(x)\right) \tag{14}$$

First we will show that, this map is well-defined. Suppose, $y \in \mathcal{O}_{\mathbb{Z}_k}(x)$, then $\mathcal{O}_{\mathbb{Z}_k}(y) = \mathcal{O}_{\mathbb{Z}_k}(x)$ and $y = h \cdot x$ for some $h \in \mathbb{Z}_k$.

$$\begin{aligned} \implies \mathcal{O}_{S_k}\left(\rho(y)\right) &= \mathcal{O}_{S_k}\left(\rho(h \cdot x)\right) \\ &= \mathcal{O}_{S_k}\left(h \cdot \rho(x)\right) && \text{(from step 2)} \\ &= \mathcal{O}_{S_k}\left(\rho(x)\right) && \text{(from the definition of orbit).} \end{aligned} \tag{15}$$

Hence, the map is well-defined.

Suppose, $\mathcal{O}_{S_k}\left(\rho(x)\right) = \mathcal{O}_{S_k}\left(\rho(y)\right)$ for some $x, y \in [0,1]^k$, then

$$\begin{aligned} \rho(y) &\in \mathcal{O}_{S_k}\left(\rho(x)\right) && \text{(from the definition of orbit)} \\ \rho(y) &= g \cdot \rho(x) && \text{(for some } g \in S_k) \\ g \cdot \rho(x) &\in \ Im\left(\rho\right) \\ g &\in \mathbb{Z}_k && \text{(from step 3)} \\ \rho(y) &= g \cdot \rho(x) = \rho\left(g \cdot x\right) && \text{(from step 2)} \\ y &= g \cdot x && \text{(from step 1)} \\ y &\in \mathcal{O}_{\mathbb{Z}_k}(x) \\ \mathcal{O}_{\mathbb{Z}_k}(y) &= \mathcal{O}_{\mathbb{Z}_k}(x). \end{aligned} \tag{16}$$

This implies that each $\mathcal{O}_{\mathbb{Z}_k}(x)$ orbit is uniquely mapped to $\mathcal{O}_{S_k}(\rho(x))$. From this, it follows that by defining the S_k-invariant function ϕ to take the same value across any orbit of the form $\mathcal{O}_{S_k}(\rho(x))$ as ψ does across the orbit $\mathcal{O}_{\mathbb{Z}_k}(x)$ (and an arbitrary value across orbits not of the form $\mathcal{O}_{S_k}(\rho(x))$), we obtain the result. □

7 Proof of Theorem 4

Proof. We will prove the result for $\mathbb{Z}_{\mathcal{I}}$-invariant function (part (b)). Similar steps hold for other variants. As stated in Theorem. 1, any $\mathbb{Z}_k$-invariant function ψ can be written as a composition of an S_k-invariant function and a specific non-linear function which is defined in (2). If we apply canonical form for S_k-invariant function as given by [3], we get,

$$\psi(x) = f_1\left(\sum_{i \in [k]} f_2\left(x_i, x_{\tau(i)}\right)\right), \tag{17}$$

for some functions f_1 and f_2.

Similarly any $\mathbb{Z}_{\mathcal{I}}$-invariant function ψ can be written as,

$$\psi(x) = f_1\left(\sum_{i \in \mathcal{I}} f_2\left(x_i, x_{\tau(i)}\right)\right), \tag{18}$$

Thus, the goal is show that, the function composition $\phi \circ M_2 \circ \rho \circ M_1$ has an equivalent form, for appropriately chosen M_1 and M_2. With M_1 chosen as in (3), we get,

$$\left(M_1 x\right)[i] = \begin{cases} x_i & \text{if } i \in I \\ 0 \end{cases} \tag{19}$$

Then applying the function ρ, we get that $\{(x_i, x_j) \mid i, j \in \mathcal{I},\ i \neq j\}$ will be the set of non-zero elements of the vector $\left(\rho \circ M_1\right)(x)$.

If we choose M_2 as stated in (5) for $\mathbb{Z}_{\mathcal{I}}$-invariant function, we obtain that $\left\{(x_i, x_{\tau(i)}) \mid i \in \mathcal{I}\right\}$ will be the set of non-zero elements of the vector $\left(M_2 \circ \rho \circ M_1\right)(x)$. Then, applying canonical form for $S_{n(n-1)}$-invariant function as given by [3], we get,

$$\left(\phi \circ M_2 \circ \rho \circ M_1\right)(x) = f_3\left(\sum_{i \in \mathcal{I}} f_4\left(x_i, x_{\tau(i)}\right) + L f_4(0)\right), \tag{20}$$

where L is constant and f_3 and f_4 are some functions. We observe that (18) and (20) have an equivalent form up to a bias term, which can subsumed in f_1 and f_2. Thus, we conclude that any $\mathbb{Z}_{\mathcal{I}}$-invariant function can be represented as a function composition of the form $\phi \circ M_2 \circ \rho \circ M_1$.

Remark 1. We provide the missing details of Theorem 4, elucidating the function composition $\phi \circ M_2 \circ \rho \circ M_1$. In this composition, the linear transformation M_1 plays a crucial role in selecting the relevant indices, associated with the index set $\mathcal{I}$, where the underlying subgroup operates. However, the remaining indices have to be passed to ϕ unchanged, similar to the results presented in [23].

Hence, ϕ is an $S_{n(n-1)}$-invariant function, where the invariance pertains to the appropriate $n(n-1)$ elements obtained from $M_2 \circ \rho \circ M_1$, while excluding the remaining indices. This can be expressed as follows:

$$\psi(x) = \phi\left(\begin{bmatrix} (M_2 \circ \rho \circ M_1)(x) \\ (I - M_1)(x) \end{bmatrix}\right).$$

Here, $S_{n(n-1)}$ acts upon the first $n(n-1)$ elements (out of the total $n(n-1)+n = n^2$ elements) and $I \in \mathbb{R}^{n\times n}$ is the identity matrix.

8 Proof of Theorem 5

Theorem 5 (Invariance to product groups)**.** *Let* $[n] = \bigcup_{j=1}^{L} \mathcal{I}_j$ *be a partition of* $[n]$*,* $G_i \in \{S_{\mathcal{I}_j}, D_{\mathcal{I}_j}, \mathbb{Z}_{\mathcal{I}_j}\}, \forall j \in [L]$ *and* $G = G_1 \times G_2 \times \cdots G_L$ *such that no two groups* G_i, G_j *are isomorphic. Let* ψ *be a* G*-invariant function, then there exists an* S_l*-invariant function* ϕ *and a specific tensor-valued function* ρ*, such that,*

$$\psi = \phi \circ \rho. \tag{7}$$

Proof. We provide the proof by example. Suppose $[n] = \mathcal{I}_1 \cup \mathcal{I}_2$ is the partition, where $\mathcal{I}_1 = \{1, 2 \ldots, k\}$ and $\mathcal{I}_2 = \{k+1, k+2 \ldots, n\}$ and $G = \mathbb{Z}_{\mathcal{I}_1} \times D_{\mathcal{I}_2}$.

Then appropriate ρ function is given by,

$$\begin{aligned}\rho : [x_1, x_2, \ldots, x_n]^T \mapsto [&(x_1, x_2),\ (x_2, x_3),\ \ldots\ , (x_k, x_1), \\ &(x_{k+1}, x_{k+2}),\ (x_{k+2}, x_{k+3}),\ \ldots\ , (x_n, x_{k+1}), \\ &(x_{k+1}, x_{k+2}),\ (x_{k+2}, x_{k+3}),\ \ldots\ (x_n, x_{k+1})]^T\end{aligned} \tag{21}$$

We claim that the ρ function is injective and G-equivariant.

We observe that the following maps (which are components of the function ρ) are injective as well as $\mathbb{Z}_{\mathcal{I}_1}$-equivariant and $D_{\mathcal{I}_2}$-equivariant respectively.

$$[x_1, x_2, \ldots, x_k]^T \mapsto [(x_1, x_2),\ (x_2, x_3),\ \ldots\ (x_k, x_1)]^T \tag{22}$$

$$\begin{aligned}[x_{k+1}, x_{k+2} \ldots, x_n]^T \mapsto [&(x_{k+1}, x_{k+2}),\ (x_{k+2}, x_{k+3}),\ \ldots\ , (x_n, x_{k+1}), \\ &(x_{k+1}, x_{k+2}),\ (x_{k+2}, x_{k+3}),\ \ldots\ (x_n, x_{k+1})]^T\end{aligned} \tag{23}$$

Therefore, ρ is injective and G-equivariant. The remaining steps follow a similar approach as the proof of Theorem 4. □

9 Proof of Theorem 6

Theorem 6 (Error probability bound for LinTS)**.** *Let the set of arms* $\mathcal{A} \subset \mathbb{R}^d$ *be finite. Suppose that the reward from playing an arm* $a \in \mathcal{A}$ *at any iteration, conditioned on the past, is sub-Gaussian with mean*[4] $a^\top \mu^\star$*. After* T *iterations, let the guessed best arm* A_T *be drawn from the empirical*

[4]A random variable X is said to be sub-Gaussian with mean β if $\mathbb{E}[e^{t(X-\beta)}] \leq e^{t^2/2}$.

distribution of all arms played in the T rounds, i.e., $\mathbb{P}[A_T = a] = \frac{1}{T}\sum_{t=1}^{T} \mathbf{1}\{a^{(t)} = a\}$ *where* $a^{(t)}$ *denotes the arm played in iteration* t*. Then,*

$$\mathbb{P}[A_T \neq a^\star] \leq \frac{c\log(T)}{T},$$

where $c \equiv c(\mathcal{A}, \mu^\star, \nu)$ *is a quantity that depends on the problem instance* $(\mathcal{A}, \mu^\star)$ *and algorithm parameter* (ν)*.*

Proof. Let $\Delta_a = \max_{\tilde{a}\in\mathcal{A}} \tilde{a}^\top \mu^\star - a^\top \mu^\star$ denote the gap in expected reward of an arm $a \in \mathcal{A}$, and let $a^\star$ be the optimal arm (thus $\Delta_{a^\star} = 0$). Let us define the LinTS algorithm's *cumulative* regret over T rounds as $R_T = \sum_{a\in\mathcal{A}} \Delta_a \mathbb{E}[N_T(a)]$, where $N_T(a) = \sum_{t=1}^{T} \mathbf{1}\left\{a^{(t)} = a\right\}$ denotes the total number of times action a is played in the time horizon $1, 2, \ldots, T$, and its *simple* regret for the guessed best arm after T rounds as $R_T^{\text{simp}} = \mathbb{E}[\Delta_{A_T}]$.

By a standard result [17, Prop. 33.2] relating the simple regret to the cumulative regret, when the guessed arm A_T is drawn according to the empirical distribution of plays as hypothesized, we have

$$R_T^{\text{simp}} = \frac{R_T}{T}. \tag{24}$$

We can also bound the simple regret from below as

$$R_T^{\text{simp}} \geq \Delta_{\min} \mathbb{P}[A_T \neq a^\star], \tag{25}$$

where $\Delta_{\min} = \min\{\Delta_a : a \in \mathcal{A}, \Delta_a > 0\}$ denotes the gap between the highest and second-highest expected reward across the arms.

It is also separately known [18, Thm. 3] that the cumulative regret of LinTS for a finite action set admits the upper bound

$$R_T \leq \kappa \log(T), \tag{26}$$

where $\kappa \equiv \kappa(\mathcal{A}, \mu^\star, \nu)$ is a quantity depending on the actions $\mathcal{A}$, true parameter $\mu^\star$ and algorithm parameter ν. Putting together (24), (25) and (26), we obtain

$$\mathbb{P}[A_T \neq a^\star] \leq \frac{\kappa\log(T)}{T\Delta_{\min}} \equiv \frac{c\log(T)}{T},$$

with $c = \frac{\kappa}{\Delta_{\min}}$, in the form as claimed. □

10 Proof of Theorem 7

Theorem 7. *Consider the following set of functions, for* $k \geq 3$*:*

$$\mathcal{A}_k = \left\{\phi \circ M \middle| M \textit{ is linear transformation from } \mathbb{R}^k \textit{ to } \mathbb{R}^k \textit{ and } \phi \textit{ is } S_k - \textit{invariant function}\right\}.$$

There exists a $\mathbb{Z}_k$*-invariant function* ψ *such that* $\psi \notin \mathcal{A}_k$*.*

Proof. Consider a $\mathbb{Z}_k$-invariant function ψ defined as follows:

$$\psi(x) \neq \psi(y) \text{ if } y \notin \mathcal{O}_{\mathbb{Z}_k}(x). \tag{27}$$

In other words, the above-defined function assigns a unique value to each orbit. Suppose $\psi = \phi \circ M$ for some S_k-invariant function ϕ and some linear transformation M. Since each orbit $\mathcal{O}_{\mathbb{Z}_k}(x)$ has a unique value and $\left|\mathcal{O}_{\mathbb{Z}_k}(x)\right| \leq k$, we have

$$\left|\psi^{-1}(\{c\})\right| \leq k \quad \text{for any } c \in \text{Im}(\psi). \tag{28}$$

The linear transformation M has a trivial null space, indicating that it has full rank and is bijective. Let $z \in \text{Im}(M)$ be such that all of its individual scalar components are unique. Such a vector exists in $\text{Im}(M)$ because M is full rank, i.e.,

$$Mx = z$$

for some $x \in \mathbb{R}^k$. Then,

$$\left|\mathcal{O}_{S_k}(z)\right| = k!. \tag{29}$$

Since $k \geq 3$, we have $k! > k$. Thus, from (28), we can see that this leads to a contradiction. □

References

[1] Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.

[2] Gregory Benton, Marc Finzi, Pavel Izmailov, and Andrew Gordon Wilson. Learning invariances in neural networks, 2020.

[3] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. *Advances in neural information processing systems*, 30, 2017.

[4] Piotr Kicki, Mete Ozay, and Piotr Skrzypczyński. A computationally efficient neural network invariant to the action of symmetry subgroups. *arXiv preprint arXiv:2002.07528*, 2020.

[5] David Steven Dummit and Richard M Foote. *Abstract algebra*, volume 3. Wiley Hoboken, 2004.

[6] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Semi-local affine parts for object recognition. In *British Machine Vision Conference (BMVC'04)*, pages 779–788. The British Machine Vision Association (BMVA), 2004.

[7] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2009.

[8] Danielle Ensign, Scott Neville, Arnab Paul, and Suresh Venkatasubramanian. The complexity of explaining neural networks through (group) invariants. *Theoretical Computer Science*, 808:74–85, 2020.

[9] Michael M Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021.

[10] 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.

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

[12] Risi Kondor and Shubhendu Trivedi. On the generalization of equivariance and convolution in neural networks to the action of compact groups. In *International Conference on Machine Learning*, pages 2747–2755. PMLR, 2018.

[13] Taco S Cohen, Mario Geiger, and Maurice Weiler. A general theory of equivariant cnns on homogeneous spaces. *Advances in neural information processing systems*, 32, 2019.

[14] Allan Zhou, Tom Knowles, and Chelsea Finn. Meta-learning symmetries by reparameterization. *arXiv preprint arXiv:2007.02933*, 2020.

[15] Fabio Anselmi, Georgios Evangelopoulos, Lorenzo Rosasco, and Tomaso Poggio. Symmetry-adapted representation learning. *Pattern Recognition*, 86:201–208, 2019.

[16] Shipra Agrawal and Navin Goyal. Thompson sampling for contextual bandits with linear payoffs. In *International conference on machine learning*, pages 127–135. PMLR, 2013.

[17] Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

[18] Taira Tsuchiya, Junya Honda, and Masashi Sugiyama. Analysis and design of thompson sampling for stochastic partial monitoring. *Advances in Neural Information Processing Systems*, 33:8861–8871, 2020.

[19] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando De Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2015.

[20] Tanner Fiez, Lalit Jain, Kevin G Jamieson, and Lillian Ratliff. Sequential experimental design for transductive linear bandits. *Advances in neural information processing systems*, 32, 2019.

[21] Rob A Rutenbar. Simulated annealing algorithms: An overview. *IEEE Circuits and Devices magazine*, 5(1):19–26, 1989.

[22] Eduardo Raul Hruschka, Ricardo JGB Campello, Alex A Freitas, et al. A survey of evolutionary algorithms for clustering. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 39(2):133–155, 2009.

[23] Pavan Karjol, Rohan Kashyap, and AP Prathosh. Neural discovery of permutation subgroups. In *International Conference on Artificial Intelligence and Statistics*, pages 4668–4678. PMLR, 2023.

[24] Gaëlle Loosli, Stéphane Canu, and Léon Bottou. Training invariant support vector machines using selective sampling. *Large scale kernel machines*, 2, 2007.

[25] Edward Wagstaff, Fabian B Fuchs, Martin Engelcke, Michael A Osborne, and Ingmar Posner. Universal approximation of functions on sets. *Journal of Machine Learning Research*, 23(151):1–56, 2022.

[26] Ryan L Murphy, Balasubramaniam Srinivasan, Vinayak Rao, and Bruno Ribeiro. Janossy pooling: Learning deep permutation-invariant functions for variable-size inputs. *arXiv preprint arXiv:1811.01900*, 2018.

[27] Taco Cohen. *Learning transformation groups and their invariants*. PhD thesis, PhD thesis, University of Amsterdam, 2013.

[28] Taco S Cohen and Max Welling. Steerable cnns. *arXiv preprint arXiv:1612.08498*, 2016.

[29] Jason Hartford, Devon Graham, Kevin Leyton-Brown, and Siamak Ravanbakhsh. Deep models of interactions across sets. In *International Conference on Machine Learning*, pages 1909–1918. PMLR, 2018.

[30] Risi Kondor, Zhen Lin, and Shubhendu Trivedi. Clebsch–gordan nets: a fully fourier space spherical convolutional neural network. *Advances in Neural Information Processing Systems*, 31, 2018.

[31] Sophia Sanborn, Christian Shewmake, Bruno Olshausen, and Christopher Hillar. Bispectral neural networks, 2023.

[32] Taco Cohen and Max Welling. Learning the irreducible representations of commutative lie groups, 2014.

[33] Maurice Weiler, Fred A. Hamprecht, and Martin Storath. Learning steerable filters for rotation equivariant cnns, 2018.

[34] Siamak Ravanbakhsh. Universal equivariant multilayer perceptrons. In *International Conference on Machine Learning*, pages 7996–8006. PMLR, 2020.

[35] Rui Wang, Robin Walters, and Rose Yu. Incorporating symmetry into deep dynamics models for improved generalization. *arXiv preprint arXiv:2002.03061*, 2020.

[36] Maurice Weiler and Gabriele Cesa. General e (2)-equivariant steerable cnns. *Advances in Neural Information Processing Systems*, 32, 2019.

[37] Siamak Ravanbakhsh, Jeff Schneider, and Barnabas Poczos. Equivariance through parameter-sharing. In *International conference on machine learning*, pages 2892–2901. PMLR, 2017.

[38] Pablo Gainza, Freyr Sverrisson, Frederico Monti, Emanuele Rodola, D Boscaini, MM Bronstein, and BE Correia. Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning. *Nature Methods*, 17(2):184–192, 2020.

[39] Harm Derksen and Gregor Kemper. Computational invariant theory. *Book manuscript*, 2001.

[40] Andrew W Senior, Richard Evans, John Jumper, James Kirkpatrick, Laurent Sifre, Tim Green, Chongli Qin, Augustin Žídek, Alexander WR Nelson, Alex Bridgland, et al. Improved protein structure prediction using potentials from deep learning. *Nature*, 577(7792):706–710, 2020.

[41] Brian L Trippe, Jason Yim, Doug Tischer, Tamara Broderick, David Baker, Regina Barzilay, and Tommi Jaakkola. Diffusion probabilistic modeling of protein backbones in 3d for the motif-scaffolding problem. *arXiv preprint arXiv:2206.04119*, 2022.

[42] Dan Raviv, Alexander M Bronstein, Michael M Bronstein, and Ron Kimmel. Symmetries of non-rigid shapes. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–7. IEEE, 2007.

[43] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[44] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5115–5124, 2017.

[45] Emanuele Rossi, Federico Monti, Yan Leng, Michael Bronstein, and Xiaowen Dong. Learning to infer structures of network games. In *International Conference on Machine Learning*, pages 18809–18827. PMLR, 2022.

[46] Alexander Bogatskiy, Brandon Anderson, Jan Offermann, Marwah Roussi, David Miller, and Risi Kondor. Lorentz group equivariant neural network for particle physics. In *International Conference on Machine Learning*, pages 992–1002. PMLR, 2020.

[47] Carlos Esteves. Theoretical aspects of group equivariant neural networks. *arXiv preprint arXiv:2004.05154*, 2020.

[48] Taco S Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical cnns. *arXiv preprint arXiv:1801.10130*, 2018.

[49] Carlos Esteves, Christine Allen-Blanchette, Ameesh Makadia, and Kostas Daniilidis. Learning so (3) equivariant representations with spherical cnns. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 52–68, 2018.

[50] Haggai Maron, Ethan Fetaya, Nimrod Segol, and Yaron Lipman. On the universality of invariant networks. In *International conference on machine learning*, pages 4363–4371. PMLR, 2019.

[51] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[52] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[53] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

[54] Lawrence Cayton. Algorithms for manifold learning. *Univ. of California at San Diego Tech. Rep*, 12(1-17):1, 2005.

[55] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, 10(5):1299–1319, 1998.

[56] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI conference on artificial intelligence*, 2017.

[57] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.