A Unified Framework for Discovering Discrete Symmetries

Pavan Karjol **Rohan Kashyap** **Aditya Gopalan** **Prathosh A.P.**
Department of Electrical Communication Engineering,
Indian Institute of Science, Bengaluru, Karnataka.

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, matrix-valued and non-linear 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 non-linear functions, respectively, and to infer the symmetry that is ultimately learnt. We also discuss the necessity of the matrix-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 (LeCun et al., 1995) that intrinsically expresses equivariance to translations of input images in classification tasks.

Proceedings of the 27$^{\text{th}}$ International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238.

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 (Benton et al., 2020) or designing neural nets that inherently express functions with known symmetries (Zaheer et al., 2017; Kicki et al., 2020). 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 (Dummit and Foote, 2004).

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 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 (Lazebnik et al., 2004; Felzenszwalb et al., 2009).

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 a collection of non-trivial subgroups[1] of S_n, one of which f is invariant with respect to (i.e., $f \circ g \equiv f$ for every transformation g in some subgroup of S_n). 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, matrix-valued and non-linear 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.

Note that, the goal of our paper is to propose a unified architecture for the discovering the underlying discrete subgroup. Thus, we argue that after the discovery of the *correct* symmetry using our framework, one could in practice utilize any off-the-shelf models (Kicki et al., 2020; Zaheer et al., 2017; Yang et al., 2023) to improve the model accuracy.

[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 (Dummit and Foote, 2004).

[2] In general, if we consider *all* subgroups of S_n, then the problem of learning a specific symmetry is known to be computationally intractable (Ensign et al., 2020).

1.2 Related Work

1.2.1 Group Equivariance

The utilization of symmetries in deep learning has garnered significant research interest in recent years (Bronstein et al., 2021; Dehmamy et al., 2021). Within this context, Cohen and Welling (2016) introduced G-equivariant neural networks as an extension of Convolutional Neural Networks (CNNs) to encompass a broader range of symmetries. Furthermore, Kondor and Trivedi (2018) establish convolution formulae in a more general setting, i.e., invariance under the action of any compact group and Cohen et al. (2019) 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. Kicki et al. (2020) proposed an approach that utilizes invariant polynomials to design G-invariant neural networks $f : X \to \mathsf{R}$, 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, Zaheer et al. (2017) 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.

1.2.3 Automatic Symmetry Discovery

Dehmamy et al. (2021) presents a Lie algebra convolution network (L-conv) for constructing feedforward architectures that exhibit equivariance to arbitrary continuous groups. Benton et al. (2020) propose a different approach by parameterizing a distribution over training data augmentations, while Zhou et al. (2020) 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, Anselmi et al. (2019) investigates the use of regularization on the representation matrix for unsupervised orbit learning.

Recently Yang et al. (2023) proposed LieGAN, which is based on generative adversarial approach to discover the underlying subgroup. However, most of the existing methods emphasize on continuous group symmetries. In this work, we propose a similar solution for discrete group symmetries. In particular, we demonstrate that a unified architecture can be used for arbitrary symmetry

discovery ($\{\mathsf{Z}_I, D_I, S_I\}$) using a multi-armed bandits setting which aids in identifying the exact symmetry learned as discussed in Section 2 and 4 respectively.

2 PROPOSED METHOD

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 $i_1 < \cdots < i_k$.

- Z_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_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_I is the locally symmetric group corresponding to $\mathcal{I}$, consisting of all permutations that move elements only within $\mathcal{I}$, i.e., $S_I = \{\pi \in S_n : \pi(j) = j\ \forall j \notin \mathcal{I}\}$.
- $\mathsf{Z}_k = \mathsf{Z}_I$; $D_{2k} = D_I$; $S_k = S_I$ with $\mathcal{I} = [k]$ (the first k elements of $[n]$).

Definition 2.2. Let $g \in S_n$. The action of g on 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 $f : X \to \mathsf{R}$ is said to be G-invariant, if $f(x) = f(g \cdot x), \forall g \in G, x \in X$.

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

2.2 Problem statement

Let $X = [0,1]^n \setminus E$ denote the input (instance) domain, where $E = \{[x_1, x_2 \ldots, x_n]^T \in [0,1]^n : x_i = x_j$ for some $i, j \in [n]$ with $i \neq j\}$. Note that the n-dimensional measure of the set E is zero. We frame the symmetry discovery problem as follows:

Given data $\left\{\left(x^{(u)}, y^{(u)}\right)\right\}_{u=1}^m$ with $x^{(u)} \in X, y^{(u)} \in \mathsf{R}$, and the collection of non-trivial subgroups $\mathcal{G} = \cup_{I\ [n], |I|>1} \{\mathsf{Z}_I, D_I, S_I\}$, we aim to learn a function $f : X \to \mathsf{R}$ such that f is G-invariant for some $G \in \mathcal{G}$ with respect to the data. Specifically, we wish to efficiently solve the following empirical risk minimization (ERM) problem,

$$\arg \min_{f \in \mathcal{F}(\mathcal{G})} \frac{1}{m} \sum_{u=1}^{m} \ell\left(y^{(u)}, f\left(x^{(u)}\right)\right), \tag{1}$$

where the hypothesis class $\mathcal{F}(\mathcal{G})$ is comprised of all functions that are G-invariant for some $G \in \mathcal{G}$, i.e., $\mathcal{F}(\mathcal{G}) = \{f : X \to \mathsf{R} : \exists G \in \mathcal{G}$ s.t. f is G-invariant$\}$, and ℓ stands for a loss function such as squared or absolute error loss.

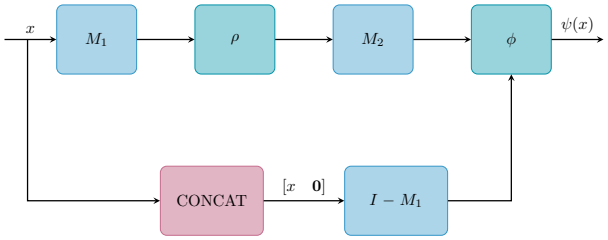

Figure 1: Proposed unified architecture for discovering symmetries, composed of linear transformations (M_1, M_2), matrix-valued (ρ) and non-linear function (ϕ). ρ is explicitly fixed whereas M_1, M_2 and ϕ are trainable. Theorem 3 guarantees that the architecture can express functions invariant to any 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.

2.3 Proposed framework

We aim to develop a framework for solving the symmetry discovery problem defined above in the problem statement. 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 Z_k-invariant function can be expressed as a composition of an S_k-invariant function and a specific matrix-valued function.

Theorem 1. *Let $\psi : [0,1]^k \to \mathsf{R}$ be Z_k-invariant. There exists an S_k-invariant function $\phi : [0,1]^{k\ \ 2} \to \mathsf{R}$ and $\rho : [0,1]^k \to [0,1]^{k\ \ 2}$, such that*

$$\psi = \phi \circ \rho, \tag{2}$$

where ρ is defined as,

$$\begin{bmatrix} x_1 \\ \vdots \\ x_k \end{bmatrix} \mapsto \begin{bmatrix} x_1 & x_2 \\ x_2 & x_3 \\ \vdots & \vdots \\ x_k & x_1 \end{bmatrix} \tag{3}$$

Proof. (Sketch) The Z_k-invariant function ψ must assign the same value to every element of any Z_k-orbit. We show that any such orbit $\mathcal{O}_{\mathsf{Z}_k}(x)$ 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 $\mathcal{O}_{\mathsf{Z}_k}(x)$ (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^1) and continuity (C^0) of the ψ and ϕ function, and in this regard we state the following theorem.

Theorem 2. *Under the same hypothesis of Theorem 1, the ϕ function is smooth (C^1) whenever ψ function is C^1. Similarly, the ϕ function is continuous (C^0) whenever ψ function is C^0.*

We state the following lemma, to prove Theorem 2.

Lemma 1. *The matrix-valued function ρ defined in (3) is a diffeomorphism between $[0,1]^k$ and its image $\rho([0,1]^k)$.*

The proof for Lemma 1 is given in the Appendix section.

Proof. From 2, we have $\psi = \phi \circ \rho$. Thus, $\psi \circ \rho^{\ -1} = \phi$. From Lemma 1, $\rho^{\ -1}$ is smooth (C^1) since ρ is a diffeomorphism. Thus, if ψ is a continuous function (C^0), then ϕ is composition of C^1 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^1, then ϕ is a composition of two C^1 functions. Thus ϕ is C^1. □

Results of the same form as Theorem 1 and Theorem 2 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 Z_l, S_l and D_l, in Theorem 3. The key idea is to introduce 'selection' matrices that appropriately reduce a general function

	S_k	Z_k	D_{2k}
$\rho(x)$	$\begin{bmatrix} \vdots & \\ x_i & x_i \\ \vdots & \end{bmatrix}_{i\in[k]}$	$\begin{bmatrix} \vdots & \\ x_i & x_{\tau(i)} \\ \vdots & \end{bmatrix}_{i\in[k]}$	$\begin{bmatrix} \vdots & \\ x_i & x_{\tau(i)} \\ x_{\tau(i)} & x_i \\ \vdots & \end{bmatrix}_{i\in[k]}$

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

to the specific type of subgroup as in Theorem 1 (Z_k, D_{2k} or S_k).

Theorem 3 (Unified symmetry discovery framework)**.** *Let $\mathcal{B}$ denote the class of all functions from $X \to \mathsf{R}$ of the form:*

$$x \mapsto \phi\left(\begin{bmatrix} (M_2 \circ \rho \circ M_1)(x) \\ (I - M_1)([x \quad \mathbf{0}]) \end{bmatrix}\right)$$

where,

- *M_1 and M_2 are matrices of size $n \times n$ and $n^2 \times n^2$ respectively.*
- *$\phi : [0,1]^{n(n+1)\ \ 2} \to \mathsf{R}$ is an S_{n^2}-invariant function where the invariance pertains to the initial n^2 rows out of a total of $n(n+1)$, and*
- *$\rho : X \to [0,1]^{n^2\ \ 2}$ is a matrix-valued function given as, $\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \mapsto \begin{bmatrix} \vdots & \\ x_i & x_j \\ \vdots & \end{bmatrix}_{i,j2[n]}$.*

Let $\mathcal{I} = \{i_1, i_2, \ldots i_k\} \subseteq [n]$ $(k > 1)$ and τ be the permutation (cyclic shift) as defined in 2.1. Then, the following hold:

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

$$M_1[u,v] = \begin{cases} 1, & \text{if } u \in [k] \text{ and } v = i_u \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

$$M_2[u,v] = \begin{cases} 1, & \text{if } u \in [k^2], \quad u = v \text{ and} \\ & (\rho \circ M_1)(x)[v] = (x_i, x_i) \\ & \text{for some } i \in \mathcal{I} \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

b) Any Z_l-invariant function belongs to $\mathcal{B}$. Moreover, M_1 is of the form as given in (4) and M_2 is as

follows:

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

c) Any D_l-invariant function belongs to $\mathcal{B}$. Moreover, M_1 is of the form as given in (4) and M_2 is as follows:

$$M_2[u,v] = \begin{cases} 1, & \text{if } u \in [k] \text{ and} \\ & (\rho \circ M_1)(x)[v] = (x_{i_u}, x_{\tau(i_u)}) \\ 1, & \text{else if } u \in [2k] \setminus [k] \text{ and} \\ & (\rho \circ M_1)(x)[v] = (x_{\tau(i_{u-k})}, x_{i_{u-k}}) \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

Proof. (Sketch) The goal is to show that $\phi \circ M_2 \circ \rho \circ M_1$ (with ϕ being S_{n^2}-invariant and ρ is as defined in the Theorem 3) is equivalent to $\phi \circ \rho$ (with ϕ being S_k-invariant and ρ is specific to the unknown subgroup, an example of which is given in Theorem 1). This is achieved via appropriately choosing M_1 and M_2 so that the elements of the form (x_i, x_j) specific to the subgroup are selected. 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.

Remark 1. While the domain of the function ϕ is defined as $[0,1]^{n(n+1)\ \ 2}$, it is worth noting that, when ϕ is post-composed with the transformation $M_2 \circ \rho \circ M_1$, the input to ϕ inevitably contains zeros at specific positions, which are contingent upon the selection matrices M_1 and M_2. Consequently, the S_{n^2}-invariance exhibited by ϕ effectively translates to permutation invariance with respect to the remaining indices (among the first n^2), namely the non-zero elements. Further elucidation on this aspect is presented in the Appendix section of this paper.

We further remark that Theorem 3 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 4 (Invariance to product groups)**.** *Let $[n] = \bigcup_{j=1}^{L} \mathcal{I}_j$ be a partition of $[n]$, $G_i \in \{S_{l_j}, D_{l_j}, \mathsf{Z}_{l_j}\}, \forall j \in [L]$ and $G = G_1 \times G_2 \times \cdots G_L$ such that no two groups G_i, G_j are isomorphic and only one of the component groups is of the type S_l. Let ψ be a G-invariant function, then there exists an S_l-invariant function ϕ and a specific matrix-valued function ρ, such that,*

$$\psi = \phi \circ \rho. \tag{8}$$

Proof. (Sketch) Let us define the function ρ, which maps to the appropriate elements of the form (x_i, x_j), corresponding to individual components of the product group G. It is important to note that ρ is both injective and G-equivariant. We denote the variable l (as in S_l-invariant function) to represent the total number of these appropriate elements. With this setup, we can demonstrate that each G-orbit can be uniquely associated with an S_{n^2}-orbit within the transformed space denoted as $Im(\rho)$. This mapping is analogous to the proof technique employed in Theorem 1.

Corollary 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 matrix-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 $G = \mathsf{Z}_{l_1} \times \mathsf{Z}_{l_2} \cdots \times \mathsf{Z}_{l_L}$ with no two $\mathsf{Z}_{l_k}, \mathsf{Z}_{l_l}$ are isomorphic (because the lengths are different). Applying Theorem 4, we prove the claim.

2.4 Optimization for discovering symmetries

Having proposed, via Theorem 3, a common functional form $(\phi \circ M_2 \circ \rho \circ M_1)$ for any function invariant to symmetries of type Z_l, D_l or S_l, 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 $\mathsf{R}^{n\ \ n}$ and $\mathsf{R}^{n^2\ \ n^2}$, respectively. However, in view of the discrete structure of M_1, M_2 prescribed explicitly by Theorem 3 (equations (4)-(7)), 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 3 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) (Agrawal and Goyal, 2013). 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., $\exists \mu^\star \in \mathsf{R}^d$ 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 5 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, 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 $n = 10$, $G = \mathsf{Z}_I$, 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 5 (Error probability bound for LinTS)**.** *Let the set of arms $\mathcal{A} \subset \mathsf{R}^d$ 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., $\mathsf{P}[A_T = a] = \frac{1}{T}\sum_{t=1}^{T} \mathbf{1}\{a^{(t)} = a\}$ where $a^{(t)}$ denotes the arm played in iteration t. Then,*

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

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).

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 (Lattimore and Szepesvári, 2020). This is then combined with a recent logarithmic bound for the cumulative regret for LinTS (Tsuchiya et al., 2020) 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 of using gradient-based optimization jointly across all three functions. On the other end, one can employ global optimization methods such as Bayesian optimization (Shahriari et al., 2015) 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 (Fiez et al., 2019), simulated annealing (Rutenbar, 1989) and evolutionary algorithms (Hruschka et al., 2009), to name just a few.

3 DISCUSSION

The work introduced by Karjol et al. (2023) 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^2}-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 Z_k-invariant functions using S_k-invariant functions and a linear transformation for $k \geq 3$.

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

Algorithm 1: Linear Parametric Thompson Sampling for Subgroup Discovery

Initialize: $\mathcal{A} \subset \{0,1\}^d$ (arm set: binary feature vectors representing each pair of matrices (M_1, M_2)),
$B \leftarrow I_d$ (prior covariance),
$f \leftarrow 0 \in \mathsf{R}^d, \hat{\mu} \leftarrow 0 \in \mathsf{R}^d$ (prior mean),
$\nu > 0$ (variance inflation parameter),
T (time horizon).
for $t \in \{1, 2, \ldots, T\}$ **do**
  Sample μ independently from $\mathcal{N}\left(\hat{\mu}, \nu^2 B^{-1}\right)$
  $a \leftarrow \arg\max_{a' \in \mathcal{A}} \mu^\top a'$
  $B \leftarrow B + aa^\top$
  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 $\tilde{\phi}$
  Set reward from arm a: $\gamma \leftarrow -L(\tilde{\phi})$
  $f \leftarrow f + a\gamma$
  $\hat{\mu} \leftarrow B^{-1} f$
end
return $A_T = \arg\max_{a \in \mathcal{A}} a^\top \hat{\mu}$ (best arm for the estimated linear model)

Theorem 6. *Consider the following set of functions, for $k \geq 3$:*

$$\mathcal{A}_k = \left\{ \phi \circ M \middle| M \in \mathsf{R}^{k \times k} \text{ (matrix), } \phi \text{ is } S_k\text{-invariant} \right\}.$$

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

Proof. (Sketch) We show the non-realizability of a Z_k-invariant function which has a unique value for each orbit. We have, $\left|\mathcal{O}_{\mathsf{Z}_k}(x)\right| \leq k$. 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 Z_k-invariant functions for $n \geq k \geq 3$.

Conjecture 1. *Consider the following set of functions, for $n \geq 3$ and $k \leq n$,*

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

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

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

3.1 Canonical form

The proposed architecture utilizes a common ϕ i.e., an S_{n^2}-invariant network, while the work proposed in Karjol et al. (2023) requires ϕ be modified depending on the subgroup type. Moreover, our framework yields a canonical form for our overall architecture, as illustrated for the Z_I subgroup, given as:

$$\phi\left(\begin{bmatrix} (M_2 \circ \rho \circ M_1)(x) \\ (I - M_1)([x \quad \mathbf{0}]) \end{bmatrix}\right) = \mu\left(\sum_{i_l \in I} \eta\left(x_{i_l}, x_{\tau(i_l)}\right) + C, Q\right),$$

where $C = \left(n^2 - k\right)\eta(0,0)$ (which is a constant), and μ, η denote specific functions and $Q = (I - M_1)[x \quad \mathbf{0}]$. This follows from the canonical form of ϕ as proved in Zaheer et al. (2017). Similar results can be obtained for S_I and D_I subgroups. This allows for a simple implementation of our architecture for various applications.

3.2 Handling non-divisors of n

We emphasize that the work proposed by Karjol et al. (2023) for learning Z_I (or D_I) symmetries is applicable only when $k|n$. In contrast, our framework allows for the discovery of subgroups of type Z_I (or D_I) for any $|\mathcal{I}| = k \leq n$, thus allowing a larger class of subgroups. To substantiate this contrast quantitatively, Table (2) presents the ratio between the number of locally cyclic (or dihedral) subgroups identifiable by the method in Karjol et al. (2023) (N_1) and our own (N_2). Notably, as n increases, this ratio tends towards zero, suggesting that method in Karjol et al. (2023) can only discern a negligible fraction of locally cyclic (or dihedral) subgroups. In contrast, our proposed approach circumvents this limitation, facilitating broader applicability across diverse group structures.

n	$\frac{N_1}{N_2}$
10	3×10^{1}
16	12×10^{2}
25	16×10^{4}
100	5.5×10^{86}

Table 2: Ratio of number of discovered subgroups using the method in Karjol et al. (2023) $\left(N_1 = \sum_{k=1,kjn}^{n} \binom{n}{k}\right)$ compared to our method $\left(N_2 = \sum_{k=1}^{n} \binom{n}{k}\right)$.

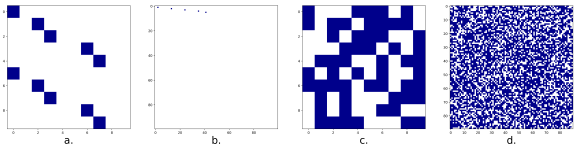

Figure 2: Visualization of the reference (bandit) M_1 (a) and M_2 (b) matrices, as well as those (c, d) obtained through training our method entirely using SGD for the task of polynomial regression of Z_I-invariant function, with $n = 10$ and $\mathcal{I} = \{0, 2, 3, 6, 7\}$.

4 EXPERIMENTS

We assess the performance of our proposed method in two representative tasks that have been considered in previous related work Kicki et al. (2020); Zaheer et al. (2017); Karjol et al. (2023), one on synthetically generated data (polynomial regression) and the other on a real-world image dataset (image-digit sum) [4]. We discuss additional experiments and potential applications in the appendix section.

4.1 Polynomial Regression

In this task, we conduct the model training to learn a G-invariant polynomial as studied in Kicki et al. (2020). For example, with $n = 5, k = 4$; $f(x) = x_1x_2x_3x_4 + x_5$ is an S_4-invariant polynomial function. Note 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 (Loosli et al. (2007)). Using the proposed bandit setting, we discover the underlying subgroup (in this case S_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_I) and polynomial regression (Z_I and D_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.

In Table (1.b), the top 3 bandit arms corresponds to the best three arms returned by the LinTS algorithm. We note that, in each case the top 3 results is the S_I, Z_I or D_I for the correct index set $\mathcal{I}$.

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 Z_I and D_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 Karjol et al. (2023), which combines linear transformations and an invariant network specifically designed for each subgroup type.

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 Effect of Label Noise and Data Size

To delve deeper into the efficacy of the proposed methodology, we conducted experiments involving la-

[4] While our theoretical results exclude the set E (as defined in the problem statement) from the input domain, we have opted not to do so in our experiments, considering that E is a set with measure zero.

Task	G	Accuracy
Polynomial Regression	Z_I	100
Polynomial Regression	D_I	100
Image-Digit Sum	S_I	100

Table (1.a): Accuracy (%)

G	$\mathsf{Z}_I(5)$	$\mathsf{Z}_I(7)$	$D_I(5)$	$D_I(7)$
Z_I	**4.2**	**6.1**	8.2	15.2
D_I	4.7	7.9	**6.3**	**10.1**
S_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 tasks with Z_I and D_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 Karjol et al. (2023), 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.

k \ ϵ	0	0.5	1
3	20	35	×
5	30	53	×
7	33	43	×
8	27	70	×

k \ N	8	16	64
3	36	33	20
5	41	32	30
7	49	41	33
8	×	31	27

k \ G	S_I	Z_I	D_I	*Overall*
3	85.1	68.3	86.1	77.1
5	92.8	86.6	92.7	89.9
7	93.9	93.2	96.2	93.6
8	94.3	92.4	95.2	93.8

Table 6: Bandit sampling iterations for identifying the underlying subgroup **(a)** with additive Gaussian noise, $\mathcal{N}(0, \epsilon * \text{stddev}(Y_{train}))$ across different k choices and **(b)** for reduced training data sizes of $N = 8, 16$ and 64. $\times$ indicates that the correct subgroup was not discovered within 80 bandit iterations and $*$ indicates results corresponding to those presented in the Table (1.a) and (1.b). **(c)** R^2 scores for subgroup types individually and combined.

bel noise and varying training sample sizes. Our findings reveal that the proposed approach demonstrates strong performance under conditions of low noise levels. However, its effectiveness diminishes notably in identifying the optimal subgroup when exposed to high levels of noise, as illustrated in Table (6.a). Similarly, our model exhibits favorable performance even with reduced sample sizes, except when the sample count drops to inadequately low levels, as depicted in Table (6.b). We attribute these instances of failure to insufficient information for distinguishing between various subgroups. Consequently, we intend to explore these observations further in subsequent research endeavors.

4.6 Linearity of the Reward Model

To validate the linearity of the reward model in the bandit algorithm, we present the R^2 scores in Table (6.c). These scores are derived from linear regression models that fit feature vectors (representing bandit arms) to the corresponding rewards, considering varying values of k such as $\{3, 5, 7, 8\}$ and $n = 10$. Higher R^2 values serve to affirm the accuracy of our proposed methodology. Notably, certain instances, such as when $k = 3$ (corresponding to Z_I), exhibit relatively lower scores. While this discrepancy may not be of significant concern as long as the linear reward model effectively distinguishes between subgroups, it raises potential issues regarding the applicability of linearity assumptions across a broader spectrum of subgroups beyond S_I, Z_I, and D_I. Addressing this concern may entail exploring alternative features, such as polynomial features, kernelized bandits, or generalized bandit models, to enhance the robustness of the reward model.

4.7 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.

Acknowledgments

We extend our sincere appreciation to Mohit Raju, from the Mathematics Department at the Indian Institute of Science, for validating certain proofs.

References

Agrawal, S. and Goyal, N. (2013). Thompson sampling for contextual bandits with linear payoffs. In *International conference on machine learning*, pages 127–135. PMLR.

Anselmi, F., Evangelopoulos, G., Rosasco, L., and Poggio, T. (2019). Symmetry-adapted representation learning. *Pattern Recognition*, 86:201–208.

Benton, G., Finzi, M., Izmailov, P., and Wilson, A. G. (2020). Learning invariances in neural networks.

Bronstein, M. M., Bruna, J., Cohen, T., and Veličković, P. (2021). Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*.

Carter, R. L. (1997). *Molecular symmetry and group theory*. John Wiley & Sons.

Cohen, T. and Welling, M. (2016). Group equivariant convolutional networks. In *International conference on machine learning*, pages 2990–2999. PMLR.

Cohen, T. S., Geiger, M., and Weiler, M. (2019). A general theory of equivariant cnns on homogeneous spaces. *Advances in neural information processing systems*, 32.

Dehmamy, N., Walters, R., Liu, Y., Wang, D., and Yu, R. (2021). Automatic symmetry discovery with lie algebra convolutional network. *Advances in Neural Information Processing Systems*, 34:2503–2515.

Dummit, D. S. and Foote, R. M. (2004). *Abstract algebra*, volume 3. Wiley Hoboken.

Ensign, D., Neville, S., Paul, A., and Venkatasubramanian, S. (2020). The complexity of explaining neural networks through (group) invariants. *Theoretical Computer Science*, 808:74–85.

Felzenszwalb, P. F., Girshick, R. B., McAllester, D., and Ramanan, D. (2009). Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645.

Fiez, T., Jain, L., Jamieson, K. G., and Ratliff, L. (2019). Sequential experimental design for transductive linear bandits. *Advances in neural information processing systems*, 32.

Hruschka, E. R., Campello, R. J., Freitas, A. A., et al. (2009). A survey of evolutionary algorithms for clustering. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 39(2):133–155.

Karjol, P., Kashyap, R., and Prathosh, A. (2023). Neural discovery of permutation subgroups. In *International Conference on Artificial Intelligence and Statistics*, pages 4668–4678. PMLR.

Kicki, P., Ozay, M., and Skrzypczyński, P. (2020). A computationally efficient neural network invariant to the action of symmetry subgroups. *arXiv preprint arXiv:2002.07528*.

Kondor, R. and Trivedi, S. (2018). 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.

Lattimore, T. and Szepesvári, C. (2020). *Bandit algorithms*. Cambridge University Press.

Lazebnik, S., Schmid, C., and Ponce, J. (2004). Semi-local affine parts for object recognition. In *British Machine Vision Conference (BMVC'04)*, pages 779–788. The British Machine Vision Association (BMVA).

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

Loosli, G., Canu, S., and Bottou, L. (2007). Training invariant support vector machines using selective sampling. *Large scale kernel machines*, 2.

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

Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., and De Freitas, N. (2015). Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175.

Tsuchiya, T., Honda, J., and Sugiyama, M. (2020). Analysis and design of thompson sampling for stochastic partial monitoring. *Advances in Neural Information Processing Systems*, 33:8861–8871.

Yang, J., Walters, R., Dehmamy, N., and Yu, R. (2023). Generative adversarial symmetry discovery.

Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R. R., and Smola, A. J. (2017). Deep sets. *Advances in neural information processing systems*, 30.

Zhou, A., Knowles, T., and Finn, C. (2020). Meta-learning symmetries by reparameterization. *arXiv preprint arXiv:2007.02933*.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Not Applicable]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Not Applicable]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Not Applicable]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Not Applicable]
   (b) The license information of the assets, if applicable. [Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Supplementary Materials

5 Appendix

The Appendix Section is organized as follows:-

- In Section 6, we provide an illustration of the proposed method for $G = \mathsf{Z}_I$ invariance with $n = 4$ and $\mathcal{I} = \{1, 2, 4\}$. We discuss multi-armed bandits and potential applications in Section 7 and 9 respectively.
- In Section 8, we discuss additional experiments for convex area estimation and polynomial regression tasks.
- In Section 10, we provide complete proofs for our results with additional theoretical results.

6 Illustration

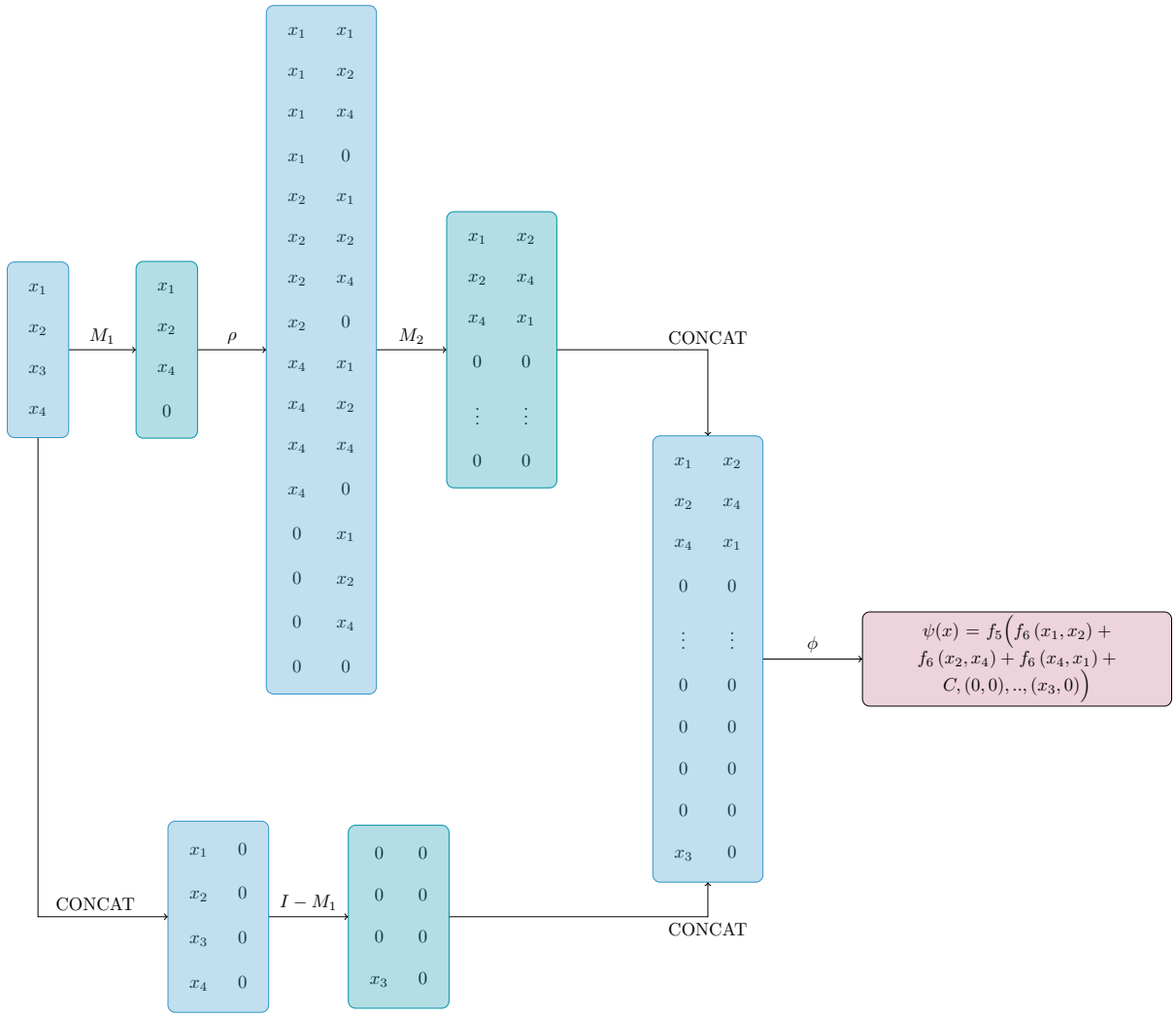

Figure 3: Illustration of the proposed method for $G = \mathsf{Z}_I$ with $n = 4$ and $\mathcal{I} = \{1, 2, 4\}$. Here f_5, f_6 denote some appropriate functions which will be approximated using neural networks.

7 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 $R_T = T\lambda \ - \mathsf{E}\left[\sum_{t=1}^{T} R_t\right]$ 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 $\lim_{T \,!\, 1} \frac{R_T}{T} = 0$, 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.

8 Additional Experiments

Table 7: Estimation Accuracy (%)

Task	G	Accuracy
Convex Area	D_I	100
S_I *(4)*	S_I	100

Table 7 presents the accuracies (top 3) achieved in subgroup discovery tasks on two tasks: (i) convex quadrangle area estimation. (ii) S_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_I-invariant function ($|\mathcal{I}| = 4$). The input is the (x, y) coordinates of the four points of the quadrilateral lying in $\mathsf{R}^{4\ \ 2}$. 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_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): The exact definitions of the polynomials used in experiments is given in Table 8. For Z_I and D_I the input is a vector in $[0, 1]^{10}$ given as; $x = [x_1, x_2, ..., x_{10}]$ whereas for S_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.

Table 8: Definition of Polynomials

INVARIANCE	POLYNOMIAL
S_I (4)	$x_1x_2x_3x_4 + x_5$
Z_I (5)	$x_1x_2^2 + x_2x_3^2 + x_3x_6^2 + x_6x_7^2 + x_7x_1^2$
Z_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_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_I (7)	$x_1x_2^2 + x_2x_3^2 + x_3x_6^2 + x_6x_7^2 + x_7x_9^2 + x_9x_{10}^2 + x_{10}x_1^2 + x_1x_{10}^2 + ... + x_2x_1^2$

9 Potential applications: Molecular Properties

In our research, we introduce a novel framework for the discovery of discrete invariance in functions, particularly concerning their behavior under a set of discrete symmetries. One compelling application of this framework emerges in the domain of molecular properties and their underlying symmetries. Consider a scenario where a collection of molecules exhibits a shared property, and it is hypothesized that this property is rooted in the presence of a common point group or discrete symmetry group (Carter (1997)). Our framework can serve as a powerful tool to discover this common point group and explore this hypothesis.

However, it is essential to note that the successful application of our framework necessitates the proper representation of molecules in terms of graphs or other suitable data structures. Additionally, we advocate the construction of backbone-invariant neural networks, such as ϕ, tailored to these data structures, specifically designed to withstand certain symmetry transformations (known as symmetry elements). This prerequisite forms a distinct yet intriguing avenue of research, wherein our framework for symmetry discovery plays a pivotal role. By leveraging our method, researchers can effectively tackle the challenging task of identifying and understanding the discrete symmetries that underlie molecular properties, promising significant advancements in the fields of chemistry, materials science, and drug discovery.

10 Complete Proofs and Additional Theoretical Results

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.*

10.1 Proof of Theorem 1

Theorem 1. *Let $\psi : [0,1]^k \to \mathsf{R}$ be Z_k-invariant. There exists an S_k-invariant function $\phi : [0,1]^{k^2} \to \mathsf{R}$ and $\rho : [0,1]^k \to [0,1]^{k^2}$, such that*

$$\psi = \phi \circ \rho, \tag{2}$$

where ρ is defined as,

$$\begin{bmatrix} x_1 \\ \vdots \\ x_k \end{bmatrix} \mapsto \begin{bmatrix} x_1 & x_2 \\ x_2 & x_3 \\ \vdots & \vdots \\ x_k & x_1 \end{bmatrix} \tag{3}$$

Proof. **Step 1**: First, we show that the $\rho : X \to \mathsf{R}^k$ 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,

$$\begin{bmatrix} x_1, & x_2 \\ x_2 & x_3 \\ \vdots & \vdots \\ x_k & x_1 \end{bmatrix} = \begin{bmatrix} y_1 & y_2 \\ y_2 & y_3 \\ \vdots & \vdots \\ y_k & y_1 \end{bmatrix}, \tag{9}$$

thus,

$$(x_1, x_2) = (y_1, y_2),\ (x_2, x_3) = (y_2, y_3),\ \ldots,\ (x_{k\ \ 1}, x_k) = (y_{k\ \ 1}, y_k),\ (x_k, x_1) = (y_k, y_1). \tag{10}$$

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

$$\rho^{\ -1}\left(\begin{bmatrix} x_1, & x_2 \\ x_2 & x_3 \\ \vdots & \vdots \\ x_k & x_1 \end{bmatrix}\right) = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{bmatrix}. \tag{11}$$

Step 2: It is obvious to see that ρ is a Z_k-equivariant function, i.e.,

$$\rho(h \cdot x) = h \cdot \rho(x), \quad \forall h \in \mathsf{Z}_k \tag{12}$$

Step 3: We now show that, for any $g \in S_k$, $g \cdot \rho(x) \in \ \mathrm{Im}(\rho)$ if and only if $g \cdot \rho(x) = h \cdot \rho(x)$ for some $h \in \mathsf{Z}_k$. In other words, any permutation (row wise) of $\rho(x)$ correspond to some cyclic shift of $\rho(x)$.

From Step 2, we get that, if $g \in \mathsf{Z}_k$, 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

$$\rho(x) = \begin{bmatrix} x_1, & x_2 \\ x_2 & x_3 \\ \vdots & \vdots \\ x_k & x_1 \end{bmatrix}$$

$$g \cdot \rho(x) = \begin{bmatrix} x_{g(1)} & x_{\tau(g(1))}) \\ x_{g(2)} & x_{\tau(g(2))}) \\ \vdots & \vdots \\ x_{g(k)} & x_{\tau(g(k))} \end{bmatrix} \tag{13}$$

$$\rho^{\ -1}(g \cdot \rho(x)) = \begin{bmatrix} x_{g(1)} \\ x_{g(2)} \\ \vdots \\ x_{g(k)} \end{bmatrix} \quad (g \cdot \rho(x) \in Im(\rho) \text{ and applying (11)})$$

$$\rho(\rho^{\ -1}(g \cdot \rho(x))) = g \cdot \rho(x) = \begin{bmatrix} x_{g(1)} & x_{g(2)} \\ x_{g(2)} & x_{g(3)} \\ \vdots & \vdots \\ x_{g(k)} & x_{g(1)} \end{bmatrix} \tag{14}$$

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

From eq. (13) and (14), (substituting $w = g(1)$), we get,

$$g \cdot \rho(x) = g \cdot \begin{bmatrix} x_1 & x_2 \\ x_2 & x_3 \\ \vdots & \vdots \\ x_k & x_1 \end{bmatrix} = \begin{bmatrix} x_w & x_{\tau(w)} \\ x_{\tau(w)} & x_{\tau^2(w)} \\ \vdots & \vdots \\ x_{\tau^{k-1}(w)} & x_{\tau^k(w)} \end{bmatrix}, \tag{15}$$

which is nothing but cyclic shift of $\rho(x)$. Thus, $g \cdot \rho(x) = h \cdot \rho(x)$ for some $h \in \mathsf{Z}_k$.

Step 4: Claim: The following map is injective:

$$\mathcal{O}_{\mathsf{Z}_k}(x) \mapsto \mathcal{O}_{S_k}(\rho(x)) \tag{16}$$

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

$$\begin{aligned} \implies \mathcal{O}_{S_k}(\rho(y)) &= \mathcal{O}_{S_k}(\rho(h \cdot x)) \\ &= \mathcal{O}_{S_k}(h \cdot \rho(x)) && \text{(from step 2)} \\ &= \mathcal{O}_{S_k}(\rho(x)) && \text{(from the definition of orbit).} \end{aligned} \tag{17}$$

Hence, the map is well-defined.

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

$$\begin{aligned} \rho(y) &\in \mathcal{O}_{S_k}(\rho(x)) && \text{(from the definition of orbit)} \\ \rho(y) &= g \cdot \rho(x) && \text{(for some } g \in S_k) \\ g \cdot \rho(x) &\in Im(\rho) \\ g &\in \mathsf{Z}_k && \text{(from step 3)} \\ \rho(y) &= g \cdot \rho(x) = \rho(g \cdot x) && \text{(from step 2)} \\ y &= g \cdot x && \text{(from step 1)} \\ y &\in \mathcal{O}_{\mathsf{Z}_k}(x) \\ \mathcal{O}_{\mathsf{Z}_k}(y) &= \mathcal{O}_{\mathsf{Z}_k}(x). \end{aligned} \tag{18}$$

This implies that each $\mathcal{O}_{\mathsf{Z}_k}(x)$ 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 $\mathcal{O}_{\mathsf{Z}_k}(x)$ (and an arbitrary value across orbits not of the form $\mathcal{O}_{S_k}(\rho(x))$), we obtain the result. □

10.2 Additional Theoretical Results

Theorem 7. *Let $\psi : X \to \mathsf{R}$ be D_{2k}-invariant. There exists an S_{2k}-invariant function $\phi : [0,1]^{2k^2} \to \mathsf{R}$ and $\rho : X \to [0,1]^{2k^2}$, such that*

$$\psi = \phi \circ \rho, \tag{19}$$

where ρ is defined as,

$$\begin{bmatrix} x_1 \\ \vdots \\ x_k \end{bmatrix} \mapsto \begin{bmatrix} x_1 & x_2 \\ x_2 & x_1 \\ x_2 & x_3 \\ x_3 & x_2 \\ \vdots & \vdots \\ x_k & x_1 \\ x_1 & x_k \end{bmatrix} \tag{20}$$

Proof. As discussed in Theorem 1, the goal is to map each of the D_{2k}-orbit in the input domain X uniquely to a S_{2k}-orbit in $\rho(X)$.

Step 1: First, we show that ρ is injective. Suppose $\rho(x) = \rho(y)$ for some

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{bmatrix}, y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_k \end{bmatrix}. \tag{21}$$

Then,

$$\begin{bmatrix} x_1 & x_2 \\ x_2 & x_1 \\ x_2 & x_3 \\ x_3 & x_2 \\ \vdots & \vdots \\ x_k & x_1 \\ x_1 & x_k \end{bmatrix} = \begin{bmatrix} y_1 & y_2 \\ y_2 & y_1 \\ y_2 & y_3 \\ y_3 & y_2 \\ \vdots & \vdots \\ y_k & y_1 \\ y_1 & y_k \end{bmatrix}. \tag{22}$$

Hence, $x_1 = y_1,\ x_2 = y_2,\ \ldots, x_k = y_k,$ Therefore, $x = y$, and thus, ρ is injective.

Step 2: ρ is equivariant function, i.e., for any $h \in D_{2k}$, we have $\rho(h \cdot x) = g \cdot \rho(x)$ for some $g \in S_{2k}$.

Step 3: Suppose $g \cdot \rho(x) \in \mathrm{Im}(\rho)$ for some $g \in S_k$, then $g \cdot \rho(x) = \rho(h \cdot x)$ for some $h \in D_{2k}$.

Case 1: If $(g \cdot \rho(x))[1] = \rho(x)[2u-1]$ for $u \in [k]$, then using the definition of $\rho(x)$ and since $g \cdot \rho(x) \in \mathrm{Im}(\rho)$, we get that,

$$g \cdot \rho(x)[3,1] = g \cdot \rho(x)[1,2].$$

Thus,

$$g \cdot \rho(x) = \begin{bmatrix} x_u & x_{\tau(u)} \\ x_{\tau(u)} & x_u \\ x_{\tau(u)} & * \\ * & x_{\tau(u)} \\ \vdots & \\ * & x_u \\ x_u & * \end{bmatrix},$$

where the '*' symbols represent values that we will discover next.

The uniqueness of x_i's (i.e., we exclude the set E from the input domain so that each of the x_i's are unique) leads to $g \cdot \rho(x)[3] = [x_{\tau(u)} \quad x_{\tau^2(u)}]$ (since, $g \cdot \rho(x)[2] = [x_{\tau(u)} \quad x_u]$). Thus, $g \cdot \rho(x)[4] = [x_{\tau^2(u)} \quad x_{\tau(u)}]$ and,

$$g \cdot \rho(x) = \begin{bmatrix} x_u & x_{\tau(u)} \\ x_{\tau(u)} & x_u \\ x_{\tau(u)} & x_{\tau^2(u)} \\ x_{\tau^2(u)} & x_{\tau(u)} \\ \vdots & \\ * & x_u \\ x_u & * \end{bmatrix}.$$

Continuing this process, we get,

$$g \cdot \rho(x) = \begin{bmatrix} x_u & x_{\tau(u)} \\ x_{\tau(u)} & x_u \\ x_{\tau(u)} & x_{\tau^2(u)} \\ x_{\tau^2(u)} & x_{\tau(u)} \\ \vdots & \\ x_{\tau^{k-1}(u)} & x_u \\ x_u & x_{\tau^{k-1}(u)} \end{bmatrix}.$$

Thus, $g \cdot \rho(x) = \rho([x_u, x_{\tau(u)}, x_{\tau^2(u)}, \ldots x_{\tau^{k-1}(u)}]^T) = \rho(h \cdot x)$ for some $h \in \mathsf{Z}_k$, then, $h \in D_{2k}$.

Case 2: If $(g \cdot \rho(x))[1] = \rho(x)[2u]$ with $u \in [k]$, then we get,

$$g \cdot \rho(x) = \begin{bmatrix} x_{\tau(u)} & x_u \\ x_u & x_{\tau(u)} \\ x_u & x_{\tau^{k-1}(u)} \\ x_{\tau^{k-1}(u)} & x_u \\ \vdots & \\ x_{\tau^2(u)} & x_{\tau(u)} \\ x_{\tau(u)} & x_{\tau^2(u)} \end{bmatrix}.$$

Thus, we obtain that:

- $g \cdot \rho(x) = \rho([x_{\tau(u)}, x_u, x_{\tau^{k-1}(u)}, \dots x_{\tau^2(u)}]^T)$.
- $g \cdot \rho(x) = \rho(\tilde{h} \cdot x)$ where $\tilde{h} \in D_{2k} \setminus \mathsf{Z}_k$ (i.e., reflection around the center followed by a cyclic shift).

To summarize, we now have the following:

- For any $h \in D_{2k}$, $\rho(h \cdot x) = g \cdot \rho(x)$ for some $g \in S_{2k}$ (from step 2).
- For any $g \in S_{2k}$, such that $g \cdot \rho(x) \in Im(\rho)$ (i.e., $g \cdot \rho(x) = \rho(y)$ for some $y \in X$),

$$g \cdot \rho(x) = \rho(h \cdot x), \text{ for some } h \in D_{2k}, \quad \text{(from step 3)}$$

Using this, we can show the mapping of orbits as discussed in Step 4 of the proof in Theorem 1. □

10.3 Exclusion of the Set E

As stated in the problem statement, the input domain is defined as $X = [0,1]^n \setminus E$, representing the input (instance) domain. Here, $E = \{[x_1, x_2, \dots, x_n]^T \in [0,1]^n : x_i = x_j \text{ for some } i, j \in [n] \text{ with } i \neq j\}$. The exclusion of this set is necessary for cases involving D_I invariance, which is an integral part of the overall framework. It should be noted that this exclusion is not required for Z_I-invariance or S_I-invariance.

The significance of excluding E in the context of D_I-invariance is illustrated by the following example:

$$x = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 1 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \end{bmatrix} \overset{\rho}{\mapsto} \rho(x) = \begin{bmatrix} 1 & 2 \\ 2 & 1 \\ 2 & 3 \\ 3 & 2 \\ 3 & 1 \\ 1 & 3 \\ 1 & 4 \\ 4 & 1 \\ 4 & 5 \\ 5 & 4 \\ 5 & 6 \\ 6 & 5 \\ 6 & 7 \\ 7 & 6 \\ 7 & 8 \\ 8 & 7 \\ 8 & 9 \\ 9 & 8 \\ 9 & 10 \\ 10 & 9 \\ 10 & 11 \\ 11 & 10 \\ 11 & 1 \\ 1 & 11 \end{bmatrix} \overset{g}{\mapsto} g \cdot \rho(x) = \begin{bmatrix} 2 & 1 \\ 1 & 2 \\ 1 & 4 \\ 4 & 1 \\ 4 & 5 \\ 5 & 4 \\ 5 & 6 \\ 6 & 5 \\ 6 & 7 \\ 7 & 6 \\ 7 & 8 \\ 8 & 7 \\ 8 & 9 \\ 9 & 8 \\ 9 & 10 \\ 10 & 9 \\ 10 & 11 \\ 11 & 10 \\ 11 & 1 \\ 1 & 11 \\ 1 & 3 \\ 3 & 1 \\ 3 & 2 \\ 2 & 3 \end{bmatrix} \overset{\rho^{-1}}{\longmapsto} \begin{bmatrix} 2 \\ 1 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \\ 1 \\ 3 \end{bmatrix} = y, \tag{23}$$

where $g \in S_{24}$. Here, $y = \tilde{h} \cdot x$ for some permutation $\tilde{h}$, but $\tilde{h} \notin D_{24}$. It is important to note that the elements x_i's are not unique (in this example, the value '1' is repeated twice), indicating that $x \in E$.

10.4 Proof of Theorem 3

Proof. We will prove the result for Z_I-invariant function (part (b)). Similar steps hold for other variants. As stated in Theorem. 1, any 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 (3). If we apply canonical form for S_k-invariant function as given by Zaheer et al. (2017), we get,

$$\psi(x) = f_1 \left(\sum_{i \mathsf{2} [k]} f_2 \left(x_i, x_{\tau(i)} \right) \right), \tag{24}$$

for some functions f_1 and f_2.

Similarly any Z_I-invariant function ψ can be written as (Karjol et al. (2023)),

$$\psi(x) = f_3 \left(\sum_{i \mathsf{2} I} f_4 \left(x_i, x_{\tau(i)} \right), Q \right), \tag{25}$$

for some functions f_3 and f_4, where $Q = (I - M_1)[x \quad \mathbf{0}]$.

Thus, the goal is show that, the function

$$x \mapsto \phi \left(\begin{bmatrix} (M_2 \circ \rho \circ M_1)(x) \\ (I - M_1)([x \quad \mathbf{0}]) \end{bmatrix} \right)$$

has an equivalent form, for appropriately chosen M_1 and M_2. With M_1 chosen as in (4), we get,

$$(M_1 x)[i] = \begin{cases} x_i & \text{if } i \in I \\ 0 & \end{cases} \tag{26}$$

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 $(\rho \circ M_1)(x)$.

If we choose M_2 as stated in (6) for Z_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 $(M_2 \circ \rho \circ M_1)(x)$. Then, applying canonical form for S_{n^2}-invariant function as given by Zaheer et al. (2017), we get,

$$\phi \left(\begin{bmatrix} (M_2 \circ \rho \circ M_1)(x) \\ (I - M_1)([x \quad \mathbf{0}]) \end{bmatrix} \right) = f_5 \left(\sum_{i \mathsf{2} I} f_6 \left(x_i, x_{\tau(i)} \right) + L f_4(0,0), Q \right), \tag{27}$$

where L is constant and f_5 and f_6 are some functions. We observe that (25) and (27) have an equivalent form up to a bias term, which can subsumed in f_3 and f_4. Thus, we conclude that any Z_I-invariant function can be represented as a function of the form, $x \mapsto \phi \left(\begin{bmatrix} (M_2 \circ \rho \circ M_1)(x) \\ (I - M_1)([x \quad \mathbf{0}]) \end{bmatrix} \right)$.

10.5 Proof of Lemma 1

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

$$\begin{bmatrix} y_1 & y_2 \\ y_2 & y_3 \\ \vdots & \vdots \\ y_k & y_1 \end{bmatrix} \mapsto \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_k \end{bmatrix}$$

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

10.6 Proof of Theorem 4

Theorem 4 (Invariance to product groups)**.** *Let* $[n] = \bigcup_{j=1}^{L} \mathcal{I}_j$ *be a partition of* $[n]$, $G_i \in \{S_{I_j}, D_{I_j}, \mathsf{Z}_{I_j}\}, \forall j \in [L]$ *and* $G = G_1 \times G_2 \times \cdots G_L$ *such that no two groups* G_i, G_j *are isomorphic and only one of the component groups is of the type* S_I*. Let* ψ *be a* G*-invariant function, then there exists an* S_l*-invariant function* ϕ *and a specific matrix-valued function* ρ*, such that,*

$$\psi = \phi \circ \rho. \tag{8}$$

Proof. Upon an analysis of different components of $\rho(x)$ corresponding to various component groups, it becomes evident that ρ is both injective and equivariant. Next, we need to establish the orbit mapping, similar to the proofs provided in Theorem 1 and Theorem 7.

Since, ρ is equivariant, it is sufficient to prove that, for any $g \in S_l$ such that $g \cdot \rho(x) \in \text{Im}(\rho)$ (i.e., $g \cdot \rho(x) = \rho(y)$ for some $y \in X$), we have:

$$g \cdot \rho(x) = \rho(h \cdot x) \tag{28}$$

for some $h \in G$.

Now, we proceed to show that permutations occur solely within the component groups. To do this, let's assume $g \cdot \rho(x) \in \text{Im}(\rho)$. Then, we can express it as:

$$g \cdot \rho(x) = \left(g_1 \cdot (\rho(x)[1:k_1]), \quad g_2 \cdot (\rho(x)[k_1+1:k_2]) \quad \ldots \quad g_L \cdot (\rho(x)[u:l])\right) \tag{29}$$

Here, $\rho(x)[i_1 : i_2]$ represents a portion of the vector $\rho(x)$ corresponding to a component group G_i. We'll now analyze the effects of the permutations on elements associated with different component subgroups G_i.

Without loss of generality, let $G_1 = D_{I_1}$, where $|\mathcal{I}_1|$ is the largest cardinality among component groups of type D_I.

Consider the first element of $g \cdot \rho(x)$.

Suppose $g \cdot \rho(x)[1] = \rho(x)[u] = [x_i \quad x_j]$ for some x_i and x_j:

$$\begin{aligned} &\text{Suppose, } g \cdot \rho(x)[1] = [x_i \quad x_j] \\ &\implies g \cdot \rho(x)[2] = [x_j \quad x_i] \end{aligned}$$

This implies that $\rho(x)[u]$ corresponds to some dihedral group $D_{I'}$. Continuing the analysis as done in step 3 of the proof of Theorem 7, we arrive at:

$$\begin{bmatrix} x_i & x_j \\ x_j & x_i \\ x_j & x_l \\ x_l & x_j \\ \vdots & \vdots \\ x_p & x_i \\ x_i & x_l \end{bmatrix} \tag{30}$$

We now have $2|\mathcal{I}_1|$ elements corresponding to a dihedral group. However, $|\mathcal{I}_1|$ is the largest cardinality among dihedral groups, and no two component subgroups are isomorphic. Hence, we conclude:

$$\mathcal{I}_1 = \mathcal{I}^0$$

Consequently, the permutations occur within the dihedral component, and we have:

$$(g \cdot \rho(x))[1:k_1] = g_1 \cdot (\rho(x)[1:k_1]), \quad \text{for some } g_1 \in G_1$$

Next, we consider the second-largest dihedral component and continue the analysis. Similarly, we can apply the same reasoning for groups of the type Z_l and S_l. This confirms the assertion presented in equation (29). Furthermore, based on Theorem 1, Theorem 7, and similar results for S_k, we obtain:

$$\begin{aligned} g \cdot \rho(x) &= (g_1 \cdot \rho(x)[1:k_1], \quad g_2 \cdot \rho(x)[k_1+1:k_2] \quad \dots \quad g_L \cdot \rho(x)[u:l]) \\ &= (\rho(h_1 \cdot x[1:l_1]), \quad \rho(h_2 \cdot x[l_1+1:l_2]) \quad \dots \quad \rho(h_L \cdot x[l_{L\ 1}:n])) \\ &= \rho(h \cdot x) \end{aligned}$$

for some $h = (h_1, h_2, \dots, h_L) \in G$ and appropriately chosen $l_1, l_2, \dots, l_{L\ 1}$. This aligns with the claim presented in equation (28).

□

10.7 Proof of Theorem 5

Theorem 5 (Error probability bound for LinTS)**.** *Let the set of arms $\mathcal{A} \subset \mathsf{R}^d$ 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*[5] *$a^\top \mu^\star$. 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., $\mathsf{P}[A_T = a] = \frac{1}{T}\sum_{t=1}^{T} \mathbf{1}\{a^{(t)} = a\}$ where $a^{(t)}$ denotes the arm played in iteration t. Then,*

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

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 $\Delta_a = \max_{\tilde{a} 2 A} \tilde{a}^\top \mu^\star - a^\top \mu^\star$ 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 $R_T = \sum_{a 2 A} \Delta_a \mathsf{E}[N_T(a)]$, where $N_T(a) = \sum_{t=1}^{T} \mathbf{1}\{a^{(t)} = a\}$ denotes the total number of times action a is played in the time horizon $1, 2, \dots, T$, and its *simple* regret for the guessed best arm after T rounds as $R_T^{\mathsf{simp}} = \mathsf{E}[\Delta_{A_T}]$.

By a standard result (Lattimore and Szepesvári, 2020, 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

$$R_T^{\mathsf{simp}} = \frac{R_T}{T}. \tag{31}$$

We can also bound the simple regret from below as

$$R_T^{\mathsf{simp}} \geq \Delta_{\min} \mathsf{P}[A_T \neq a^\star], \tag{32}$$

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 (Tsuchiya et al., 2020, Thm. 3) that the cumulative regret of LinTS for a finite action set admits the upper bound

$$R_T \leq \kappa \log(T), \tag{33}$$

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

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 (31), (32) and (33), we obtain

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

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

10.8 Proof of Theorem 6

Theorem 6. *Consider the following set of functions, for $k \geq 3$:*

$$\mathcal{A}_k = \Big\{\phi \circ M \Big| M \in \mathsf{R}^{k \quad k} \textit{(matrix)}, \ \phi \textit{ is } S_k\textit{-invariant}\Big\}.$$

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

Proof. Consider a Z_k-invariant function ψ defined as follows:

$$\psi(x) \neq \psi(y) \text{ if } y \notin \mathcal{O}_{\mathsf{Z}_k}(x). \tag{34}$$

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 $\mathcal{O}_{\mathsf{Z}_k}(x)$ has a unique value and $\left|\mathcal{O}_{\mathsf{Z}_k}(x)\right| \leq k$, we have

$$\left|\psi^{-1}(\{c\})\right| \leq k \quad \text{for any } c \in \mathrm{Im}(\psi). \tag{35}$$

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

$$Mx = z$$

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

$$\left|\mathcal{O}_{S_k}(z)\right| = k!. \tag{36}$$

Since $k \geq 3$, we have $k! > k$. Thus, from (35), we can see that this leads to a contradiction. □