Supplementary Materials

1 Appendix

1.1 Remarks regarding theoretical results

- To obtain the function ρ mentioned in Lemma 4.2, Theorem 4.3, and Theorem 4.4 of the main paper, we use the same technique presented in the proof of Corollary 4.1.1, i.e., $\rho = \psi E^{-1}$, where E is the corresponding homeomorphism.
- Lemma 4.2, Theorem 4.3, and Theorem 4.4 are special cases of Theorem 4.5. This claim directly follows once we specify the corresponding group actions. As such, the proofs of Lemma 4.2 and Theorem 4.4 already describe the required group actions. However, this is not obvious in Theorem 4.3. In that case, we observe the following:

$$\begin{aligned} x = [x_1, x_2 \dots x_n]^T \longmapsto Mx = &[0, 0, \overset{k}{\dots}, 0, \ x_{k+1}, x_{k+2}, \dots x_n, \\ &x_1, x_2, \dots x_k, \ 0, 0, \overset{n-k}{\dots}, 0]^T, \end{aligned} \tag{1}$$

where $M = \begin{bmatrix} I_{k\times k} & 0 \\ 0 & 0 \end{bmatrix}$ as mentioned in eq. (12) of the main paper.

Under the action of $S_k^{(0)}$ $\left(h \cdot x, \ \text{for some } h \in S_k^{(0)}\right)$, we get,

$$x \overset{h}{\mapsto} x' = [x_{h(1)}, x_{h(2)}, \dots, x_{h(k)}, x_{k+1}, \dots, x_n]^T \tag{2}$$

which corresponds to $\Big(g \cdot (Mx)$, for some $g \in S_{2n}^n\Big)$ $\Big(S_{2n}^n$ is the group of permutations of first n elements out of $2n$ elements$\Big)$,

$$\begin{aligned} Mx \overset{g}{\mapsto} Mx' = &[0, 0, \overset{k}{\dots}, 0, \ x_{k+1}, x_{k+2}, \dots, x_n, \\ &x_{g(1)}, x_{g(2)}, \dots, x_{g(k)}, \ 0, 0, \overset{n-k}{\dots}, 0]^T \end{aligned} \tag{3}$$

Similary, the action of S_{2n}^n on $R(M)$ (range of M) corresponds to the action of $S_k^{(0)}$ on X.

In the following subsections, we describe the architectures of various models and additional resutls considered in our experiments.

1.2 Image-Digit sum

1. **Ground truth**:- S_k-invariant neural network proposed by Zaheer et al. (2017). It consists of two networks, γ, and ρ. Each element in the input is passed through the γ network, followed by the sum operation. The result is then fed to the second network ρ. The network γ is a feed-forward network consisting of three *dense* layers with *tanh* activation, and the second network is a *dense* layer.
2. **LSTM**:- The LSTM network used for comparison in Zaheer et al. (2017). It consists of two *dense* layers, an *LSTM* layer followed by two *dense* layers. The activation used is *tanh* function.
3. **Proposed method**:- An S_n-invariant network follows a linear layer. The S_n-invariant network has the same architecture as the ground truth (the first approach) except the input layer.

1.3 Comparison between S_k-invariant networks with backbone as $S_k^{(0}$ and S_n

As specified in Section 5.2 of the main paper, any S_k-invariant network can be realized through either an $S_k^{(0)}$ or an S_n-invariant network and a linear layer when k is fixed. In general, we observed that the S_n-invariant network as the backbone does better than the $S_k^{(0)}$ network. We attribute this to the expressivity power of the linear transformation (based on its specific structure) when an S_n invariant network is used.

1.4 Symmetric Polynomial Regression

1. **Ground truth**:- $\mathbb{Z}_k$-invariant network implemented using the design described in Kicki et al. (2020). As discussed in Kicki et al. (2020), it is a composition of a $\mathbb{Z}_k$-equivariant network and a Sum-Product Layer. It then uses a Multi-Layer Perceptron to process the $\mathbb{Z}_k$-invariant representation of the input and thus predicts the polynomial output. The network is thus invariant under the action of the given permutation subgroup $\mathbb{Z}_k$.

2. **Simple-FC**:- This is an abbreviation of a fully-connected neural network without the Reynolds operator, i.e., group averaging for this baseline implementation (Derksen and Kemper (2001)).

3. **Conv-1D**:- This is an abbreviation of the 1D Convolutional neural network equipped with fully-connected layers.

4. **Proposed method**:- To discover the underlying subgroup, we use a $\mathbb{Z}_n$-invariant neural network with the addition of a linear layer. The architectural design of the $\mathbb{Z}_n$-invariant function is the same as the ground truth network.

The hyperparameters of the above models are given in Table. (6-9) of Kicki et al. (2020).

Table 1: Mean absolute errors (MAEs) $[10^{-2}]$ for $\mathbb{Z}_2 : \mathbb{Z}_{16}$

Method	Train	Validation	Test
Ground truth	0.578 ± 0.172	1.341 ± 0.278	1.257 ± 0.251
Proposed	6.301 ± 6.968	22.865 ± 4.259	22.274 ± 4.251
Conv-1D	1.47 ± 1.056	22.412 ± 0.67	21.675 ± 0.69
Simple FC	23.4 ± 3.31	22.4 ± 1.47	21.61 ± 1.538

Table 2: Mean absolute errors (MAEs) $[10^{-2}]$ for $\mathbb{Z}_8 : \mathbb{Z}_{16}$

Method	Train	Validation	Test
Ground truth	3.204 ± 3.911	15.007 ± 1.056	14.305 ± 1.038
Proposed	9.863 ± 4.341	38.073 ± 5.407	39.313 ± 5.118
Conv-1D	23.458 ± 5.909	49.848 ± 1.188	50.148 ± 1.029
Simple FC	20.845 ± 2.398	46.13 ± 2.455	45.92 ± 2.825

Table 3: Mean absolute errors (MAEs) $[10^{-2}]$ for $\mathbb{Z}_{16} : \mathbb{Z}_{16}$

Method	Train	Validation	Test
Ground truth	6.887 ± 1.314	16.678 ± 0.555	17.158 ± 0.595
Proposed	14.527 ± 1.718	39.694 ± 4.133	40.109 ± 4.166
Conv-1D	32.961 ± 10.766	67.710 ± 3.254	68.382 ± 3.133
Simple FC	46.131 ± 2.27	54.618 ± 1.338	51.642 ± 0.896

1.4.1 Additional results

The train, validation, and test errors for the $\mathbb{Z}_2 : \mathbb{Z}_{16}$, $\mathbb{Z}_8 : \mathbb{Z}_{16}$ and $\mathbb{Z}_{16} : \mathbb{Z}_{16}$ invariant functions are provided in Table. 1, 2 and 3 respectively. The details of the all the polynomials used in our work are presented in Table. 4.

Table 4: Definitions of the various polynomials used in the main paper.

INVARIANCE	**POLYNOMIAL**
$\mathbb{Z}_5 : \mathbb{Z}_{10}$	$x_1x_2^2 + x_2x_3^2 + x_3x_4^2 + x_4x_5^2 + x_5x_1^2$
$\mathbb{Z}_{10} : \mathbb{Z}_{10}$	$x_1x_2^2 + x_2x_3^2 + x_3x_4^2 + x_4x_5^2 + x_5x_6^2$ $+x_6x_7^2 + x_7x_8^2 + x_8x_9^2 + x_9x_{10}^2 + x_{10}x_1^2$
$\mathbb{Z}_2 : \mathbb{Z}_{16}$	$x_1x_2^2 + x_2x_1^2$
$\mathbb{Z}_4 : \mathbb{Z}_{16}$	$x_1x_2^2 + x_2x_3^2 + x_3x_4^2 + x_4x_1^2$
$\mathbb{Z}_8 : \mathbb{Z}_{16}$	$x_1x_2^2 + x_2x_3^2 + x_3x_4^2 + x_4x_5^2 + x_5x_6^2 + x_6x_7^2 + x_7x_8^2 + x_8x_1^2$
$\mathbb{Z}_{16} : \mathbb{Z}_{16}$	$x_1x_2^2 + x_2x_3^2 + x_3x_4^2 + x_4x_5^2 + x_5x_6^2 + x_6x_7^2 + x_7x_8^2 + x_8x_9^2 + x_9x_{10}^2$ $+x_{10}x_{11}^2 + x_{11}x_{12}^2 + x_{12}x_{13}^2 + x_{13}x_{14}^2 + x_{14}x_{15}^2 + x_{15}x_{16}^2 + x_{16}x_1^2$

References

Derksen, H. and Kemper, G. (2001). Computational invariant theory. *Book manuscript*.

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*.

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.