Supplemental Material of A New Approach to Construct Symmetry Invariant Neural Networks

1[st] Piotr Kicki 2[nd] Piotr Skrzypczyński
Institute of Robotics and Machine Intelligence
Poznan University of Technology
Poznań, Poland
{piotr.kicki;piotr.skrzypczynski}@put.poznan.pl

3[rd] Mete Ozay
meteozay@gmail.com

I. Mathematical definitions

A **group** is a non-empty set G with the binary operator $\circ : G \times G \to G$ called product, such that

1) $a, b \in G \implies a \circ b \in G$ (closed under product),
2) $a, b, c \in G \implies (a \circ b) \circ c = a \circ (b \circ c)$ (associative),
3) $\underset{e \in G}{\exists} \underset{a \in G}{\forall} a \circ e = e \circ a = a$ (existence of identity element),
4) $\underset{a \in G}{\forall} \underset{a^{-1} \in G}{\exists} a \circ a^{-1} = a^{-1} \circ a = e$ (existence of inverse element).

A **subgroup of the group** G is a non-empty subset $S \subset G$, which together with the product $\circ$, associated with the group G, forms a group.

A **permutation group** is a group whose elements are permutations.

II. Parameters Used in the Experiments

All experiments reported in the paper were performed using Nvidia GeForce GTX-1660 Ti with a learning rate equal to 10^{-3} and regularization parameter of the ℓ_2 regularization was set to 10^{-5}.

III. Architectures Considered in the Experiments

In this section, we describe all neural network based models that were used in the experiments for comparative analysis of architectures. It is worth noting that each of these neural networks uses the $\tanh$ activation function in its hidden layers, except the output layers and layers right before the G-invariant latent representation, which do not use activation functions.

FC G-avg is an abbreviation of a fully connected neural network aggregated by the group averaging operation defined by

$$f_R(x) = \frac{1}{|G|} \sum_{g \in G} f\left(g(x)\right), \tag{1}$$

where G is a finite group and $|G|$ denotes the size of the group (number of its elements). Hyperparameters of architectures of the networks used in experiments described in Section 4.C and 4.D of the main paper are given in Table I. For both architectures, an output of a network is an average of forward passes for all $g \in G$ acting on the input of the network, according to (1).

TABLE I
Hyperparameters (FC number of kernels) of the architectures of the FC G-avg networks used in both the $\mathbb{Z}_5$-invariant polynomial approximation and the convex quadrangle area estimation experiments described in Section 4.C and 4.D of the main paper.

Polynomial approximation	Area estimation
FC 89	Flatten
FC 192	FC 64
FC 32	FC 18
FC 1	FC 1

Conv1D G-avg is an abbreviation of a composition of a 1D convolutional neural network with a fully connected neural network and the group averaging defined in (1). It uses 1D convolutions to preprocess the input exploiting the knowledge about the group G, namely, it performs the cyclic convolution on the graph imposed by the group G – each kernel acts on a triplet of the selected vertex and its two neighbors in terms of group operation. Architectures of the networks used in experiments described in Section 4.C and Section 4.D of the paper are given in Table II. Similar to the FC G-avg, the output of a network is an average of the forward passes for all $g \in G$ acting on the input, according to (1). For the sake of implementation, the first and last elements of the input sequence are concatenated with the original input at the end and beginning respectively, in order to use a typical implementation of convolutional neural networks (as they normally do not perform cyclic convolution). For example, if the original input sequence looks like $[A\,B\,C\,D]$, then the network is supplied with sequence $[D\,A\,B\,C\,D\,A]$.

FC G-inv is an abbreviation of the G-invariant neural network equipped with a fully connected neural network implementing an f_{in} function proposed in the main paper. The general scheme of the G-invariant fully connected network architecture is described in Table III. Values of n, n_{in} and

TABLE II
HYPERPARAMETERS OF ARCHITECTURES OF THE CONV1D G-AVG NETWORKS USED IN BOTH $\mathbb{Z}_5$-INVARIANT POLYNOMIAL APPROXIMATION AND CONVEX QUADRANGLE AREA ESTIMATION EXPERIMENTS DESCRIBED IN SECTION 4.C AND 4.D OF THE MAIN PAPER.

POLYNOMIAL APPROXIMATION	AREA ESTIMATION
CONV1D LAYER: 32 KERNELS OF SIZE 3X1	CONV1D LAYER: 32 KERNELS OF SIZE 3X1
CONV1D LAYER: 118 KERNELS OF SIZE 1X1	CONV1D LAYER: 2 KERNELS OF SIZE 1X1
FLATTEN LAYER	FLATTEN LAYER
FC LAYER: 32 OUTPUT CHANNELS	FC LAYER: 32 OUTPUT CHANNELS
FC LAYER: 1 OUTPUT CHANNEL	FC LAYER: 1 OUTPUT CHANNEL

n_{mid} differ between experiments and are listed in Table IV.

TABLE III
HYPERPARAMETERS OF THE FC G-INV ARCHITECTURE PROPOSED IN THE MAIN PAPER.

LAYER	OUTPUT SIZE
INPUT	$n \times n_{in}$
FC	$n \times 16$
FC	$n \times 64$
FC	$n \times nn_{mid}$
RESHAPE	$n \times n \times n_{mid}$
$\Sigma\Pi$	n_{mid}
FC	32
FC	1

TABLE IV
VALUES OF n, n_{in} AND n_{mid} USED FOR DIFFERENT EXPERIMENTS.

EXPERIMENT (SECTION)	n	n_{in}	n_{mid}
4.C	5	1	64
4.D	4	2	2
4.E	4	2	2
4.F	4	2	{1, 2, 8, 32, 128}
4.G	5	1	8

Conv1D G-inv is an abbreviation of the G-invariant neural network equipped with a 1D convolutional neural network implementing an f_{in} function proposed in our paper. The general scheme of the G-invariant network architecture with a convolutional feature extractor is described in Table V. Values of n, n_{in} and n_{mid} differ between experiments and are the same as for the FC G-inv model (listed in Table IV), except the n_{mid} used in the experiment given in Section 4.C, where $n_{mid} = 118$.

Maron is an abbreviation of the G-invariant neural network architecture which is proved by [1] to be a universal approximator. In this case, one has to provide N_{inv} elements of the generating set of G-invariant polynomials to the fully connected neural network. The degree of those polynomials is at most $|G|$ (by the Noether theorem [2]). Thus, the generating set was obtained by applying the group averaging (see (1)) to all possible polynomials in $\mathbb{R}^{n \times n_i n}$ with degree up to $|G|$. It is worth to note that according to [1], a multiplication used to form the polynomials is approximated by a neural network,

TABLE V
HYPERPARAMETERS OF THE CONV1D G-INV ARCHITECTURE PROPOSED IN THE MAIN PAPER.

LAYER	OUTPUT SIZE
INPUT	$(n+2) \times n_{in}$
CONV1D 3X1	$n \times 32$
CONV1D 3X1	$n \times nn_{mid}$
RESHAPE	$n \times n \times n_{mid}$
$\Sigma\Pi$	n_{mid}
FC	32
FC	32
FC	1

whose architecture is presented in Table VI. The architecture of a multi-layer perceptron (MLP), which was applied on these polynomials, is presented in Table VII.

IV. DATASETS

A. Convex Quadrangle Area Estimation

The dataset used in the task of convex quadrangle area estimation consists of quadrangles with the associated area values. Each of these quadrangles is defined by 8 numbers, while the associated area is the label for supervised learning. Data generation procedure for the quadrangles consists of the following steps:

1) draw the value of the center of the quadrangle according to the uniform distribution,
2) generate n angles, in the range [0, $\frac{2\pi}{n}$],
3) add $\frac{2k\pi}{n}$ to the k-th angle, for $k \in \{0, 1, \ldots, n-1\}$,
4) draw uniformly the radius r,
5) draw uniformly n disturbances and add these values to the radius,
6) generate the x, y coordinates of vertices using generated angles and radii.
7) take an absolute value of those coordinates (we want to have the coordinates positive),
8) repeat steps 1–7 until obtained quadrangle is convex,
9) calculate the area of the obtained quadrangle using the Monte Carlo method.

Each of the training set and the validation set contains 256 examples, and 1024 examples were used in the test dataset.

B. G-invariant Polynomial Approximation

The dataset used in the tasks of G-invariant polynomial approximation consists of the input which is randomly gen-

TABLE VI
HYPERPARAMETERS OF THE MULTIPLICATION NETWORK USED IN THE MARON ARCHITECTURE PROPOSED BY [1].

POLYNOMIAL APPROXIMATION	AREA ESTIMATION
FC LAYER: 64 OUTPUT CHANNELS	FC LAYER: 32 OUTPUT CHANNELS
FC LAYER: 32 OUTPUT CHANNELS	FC LAYER: 1 OUTPUT CHANNEL
FC LAYER: 1 OUTPUT CHANNEL	

TABLE VII
HYPERPARAMETERS OF THE MLP NETWORK USED IN THE MARON ARCHITECTURE PROPOSED BY [1].

POLYNOMIAL APPROXIMATION	AREA ESTIMATION
FC LAYER: 48 OUTPUT CHANNELS	FC LAYER: 40 OUTPUT CHANNELS
FC LAYER: 192 OUTPUT CHANNELS	FC LAYER: 1 OUTPUT CHANNEL
FC LAYER: 32 OUTPUT CHANNELS	
FC LAYER: 1 OUTPUT CHANNEL	

TABLE VIII
EXACT FORMULAS OF THE POLYNOMIALS USED IN EXPERIMENTS GIVEN IN SECTION 4.C, 4.E AND 4.G IN THE MAIN PAPER.

INVARIANCE	POLYNOMIAL
$\mathbb{Z}_5$	$P_{\mathbb{Z}_5}(x) = x_1x_2^2 + x_2x_3^2 + x_3x_4^2 + x_4x_5^2 + x_5x_1^2$
$\mathbb{Z}_3$	$P_{\mathbb{Z}_3}(x) = x_1x_2^2 + x_2x_3^2 + x_3x_1^2 + 2x_4 + x_5$
S_3	$P_{S_3}(x) = x_1x_2x_3 + 2x_4 + x_5$
$S_3 \times S_2$	$P_{S_3 \times S_2}(x) = x_1x_2x_3 + x_4 + x_5$
D_8	$P_{D_8} = x_1x_2^2 + x_2x_3^2 + x_3x_4^2 + x_4x_1^2 + x_2x_1^2 + x_3x_2^2 + x_4x_3^2 + x_1x_4^2 + x_5$
A_4	$P_{A_4} = x_1x_2 + x_3x_4 + x_1x_3 + x_2x_4 + x_1x_4 + x_2x_3 + x_1x_2x_3 + x_1x_2x_4 + x_1x_3x_4 + x_2x_3x_4 + x_5$
S_4	$P_{S_4} = x_1x_2x_3x_4 + x_5$

erated and expected output, which is simply calculated using formulas listed in Table VIII. The generation procedure of the input draws samples from a uniform distribution between 0 and 1. For the experiments given in Section 4.C and 4.E, the number of samples used for training, validation and test set is 16, 480 and 4800 respectively. Only in the experiment given in Section 4.G, the number of samples in the training dataset was increased to 160, as the aim of the experiment was not analyzing the generalization properties, but analyzing the ability to adjust the weights to capture other invariances. To train and test models using datasets with similar statistical properties, the seed for the data generation was set to 444.

REFERENCES

[1] H. Maron, E. Fetaya, N. Segol, and Y. Lipman, "On the universality of invariant networks," *CoRR*, vol. abs/1901.09342, 2019. [Online]. Available: http://arxiv.org/abs/1901.09342

[2] H. P. Kraft and C. Procesi, "Classical invariant theory, a primer," 1996.