EQUIVARIANT AND INVARIANT REYNOLDS NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Invariant and equivariant networks are useful in learning data with symmetry, including images, sets, point clouds, and graphs. In this paper, we consider invariant and equivariant networks for symmetries of finite groups. Invariant and equivariant networks have been constructed by various researchers using Reynolds operators. However, Reynolds operators are computationally expensive when the order of the group is large because they use the sum over the whole group, which poses an implementation difficulty. To overcome this difficulty, we consider representing the Reynolds operator as a sum over a subset instead of a sum over the whole group. We call such a subset a Reynolds design, and an operator defined by a sum over a Reynolds design a reductive Reynolds operator. For example, in the case of a graph with n nodes, the computational complexity of the reductive Reynolds operator is reduced to $O(n^2)$, while the computational complexity of the Reynolds operator is $O(n!)$. We construct learning models based on the reductive Reynolds operator called equivariant and invariant Reynolds networks (ReyNets) and prove that they have universal approximation property. Reynolds designs for equivariant ReyNets are derived from combinatorial observations with Young diagrams, while Reynolds designs for invariant ReyNets are derived from invariants called Reynolds dimensions defined on the set of invariant polynomials. Numerical experiments show that the performance of our models is comparable to state-of-the-art methods.

1 INTRODUCTION

The universal approximation theorem in machine learning states that any continuous function can be approximated by a deep neural network, but in a practical situation, there are many cases where learning fails. It is believed that this is because the number of parameters used is too large to learn an appropriate model from a hypotheses set. Therefore, researchers developed task-specific models, such as convolutional networks for image input, and obtained better results. On the other hand, Zaheer et al. (2017) developed a model using permutation invariance and permutation equivariance, and obtained good experimental results as well as theoretical development. Such a model was generalized by Maron et al. (2019b) for subgroups of permutation groups. The problem with this model is that the shape of the parameter sharing needs to be calculated mathematically for each width of the equivariant layer. This mathematical difficulty is a major obstacle for practical applications of this model, which cannot have universality unless the width is sufficiently large Chen et al. (2019). Therefore, we take a different approach than they do. In this paper, we develope an equivariant and invariant deep neural network model for the action of permutation groups on higher-order tensor spaces using *Reynolds operators*. An important application of these model is the task of using sets, graphs and hypergraphs as input.

The central part of our idea is to use the equivariant Reynolds operator. The equivariant Reynolds operator has the property of transforming a function into an equivariant function, and by adapting this operator to a deep neural nets, we obtain a class of equivariant neural networks. Such attempts have been made in the past, for example byYarotsky (2021); Kicki et al. (2021); van der Pol et al. (2020); Mouli & Ribeiro (2020); Murphy et al. (2018), but the use of Reynolds operator causes computational problems when the order of the groups is large in the part that computes the equivariant Reynolds operator.

Therefore, we introduce the concept of Reynolds design, which is an analog of spherical design (Bannai & Bannai (2009)). Here, a t-spherical design is a finite subset $H \subset \mathbb{S}^{d-1}$ such that any polynomial f with degree at most t satisfies

$$\frac{1}{|\mathbb{S}^{d-1}|}\int_{\mathbb{S}^{d-1}} f(x)dx = \frac{1}{|H|}\sum_{x\in H} f(x). \tag{1}$$

Using the spherical design, the computation over the whole of $\mathbb{S}^{d-1}$ can be reduced to that over the small subset H. Similarly, a Reynolds design is a subset H of a group G that realizes

$$\frac{1}{|G|}\sum_{g\in G} f(g\cdot x) = \frac{1}{|H|}\sum_{g\in H} f(g\cdot x)$$

for a target function f, where $g\cdot x$ is the action of g to x. We call the operator defined on the right-hand side the reductive Reynolds operator, and use this operator instead of the Reynolds operator to construct equivariant and invariant networks. The advantage of this construction is that it preserves the universal approximation property of the construction using Reynolds operators. Also, in many cases, the Reynolds design can be much smaller than the original G. To formulate this fact, we use the space of higher order tensors. The space of higher order tensors is the space that reflects the adjacency matrices and tensors of graphs and hypergraphs, and it is through this space that models are defined when using equivariant models. For this space, we prove a representation theorem for equivariant functions. Technically speaking, we use the relation between Young diagrams and the representation of symmetric groups on higher-order tensor spaces (Sagan (2001)). As a result, in the case of graphs with n nodes, the order of the group is $n!$ but we have shown that there exist Reynolds designs of order $n(n-1)$. Thus, by using Reynolds designs, we can avoid the problem of computational complexity.

For the invariant network we follow the same structure as in the case of CNNs. In other words, we first extract the features in an equivariant network and then transform them in a deep neural network. At this point we can naively consider the following phenomenon: if we want to approximate $\frac{1}{n}(x_1^i + .. + x_n^i) = \frac{1}{|S_n|}\sum_{g\in S_n} g\cdot x_1^i$, we assume that we use Reynolds operator, so in effect we only need to approximate x_1^i. In order to formulate this phenomenon we introduce an invariant called the Reynolds dimension.Briefly, the Reynolds dimension is the number of variables required for the input of the functions before they are converted by the Reynolds operator, when the model had universality in the above construction. It is not clear what the Reynolds dimension is when taking a higher order tensor space as input, but our expectation is that it is independent of the dimension n. We prove that we can construct a Reynolds design when the Reynolds dimension is fixed. We also find that all input variables are not necessary when the Reynolds dimension is low, therefore we construct networks with reduced inputs (reduced ReyNets).

Using these models, we conduct experiments on synthetic data. In addition, as an implementation, we use the above representation theorem to develop the Corner MSE, which restricts the loss function for an equivariant function to two components. As a result, our method significantly outperform the baseline of FNN and Maron's model, confirming its consistency with the theoretical part.

2 Previous Work

Equivariance and Invariance. Various machine learning tasks aims to approximate a certain target map such as the labeling function in classification and regression. When symmetries exist behind data, the target map often have invariance or equivariance to the symmetries. In such cases, invariant or equivariant networks are effective and efficient to approximate the target map because the model complexity can be significantly reduced than neural networks without specific structure for the symmetries. Convolutional neural networks (CNNs) are well-known as a seminal equivariant model to translation symmetry (LeCun et al. (1989); Krizhevsky et al. (2012)). Inspired by the success of CNNs, various equivariant models have been proposed. Besides continuous symmetries such as translation, symmetries of finite groups often appear in many machine learning tasks. In the case where sets or point clouds are inputs, the target functions are typically invariant to the order of data points. Then, this function has invariance to the permutation group on data points (Qi et al. (2017); Zaheer et al. (2017)). In the case where graphs or hyper-graphs are inputs, the symmetry is represented by permutation on a tensor product space. Gori et al. (2005); Scarselli et al. (2008) introduced

graph neural networks. Various researchers generalized convolution to the setting of graphs motivated by CNNs (Bruna et al. (2013); Henaff et al. (2015); Kipf & Welling (2016); Defferrard et al. (2016); Levie et al. (2018)). Recently, Kondor et al. (2018); Maron et al. (2019a); Chen et al. (2019) investigate graph neural networks. Hartford et al. (2018) consider interaction between sets. Graham & Ravanbakhsh (2019) consider relational databases as generalization of graphs and provide equivariant models to handle relational databases. Maron et al. (2020) consider sets of symmetric elements as input data.

Universality. The expressive power of learning models is mathematically validated by universal approximation theorems. Many universal approximation theorems have been proved for different conditions. Invariant models with universal approximation property are provided for pointclouds networks and sets network (Qi et al. (2017); Zaheer et al. (2017)), networks with matrices and higher-order tensors as input (Hartford et al. (2018)), graph and hyper-graph networks (Maron et al. (2018); Nguyen & Maehara (2020)), and networks invariant to the action of finite groupMaron et al. (2019b). The universality of equivariant models for finite group are proved by Ravanbakhsh et al. (2017). On the other hand, Oono & Suzuki (2019) proved that graph neural networks of message passing type are over smoothing when many layers are stacked. There are also learning models with universality in other settings (Yarotsky (2021); Keriven & Peyré (2019); Maehara & NT (2019); Segol & Lipman (2019)).

3 Reynolds Operator and Reynolds Designs

In this section, we organize the mathematical framework for dealing with symmetric networks. The first objects to be considered are the following equivariant and invariant functions.

Definition 1 (Invariant / Equivariant Function)**.** *For a group G acting on $\mathbb{R}^N$ and $\mathbb{R}^M$, a function $f : \mathbb{R}^N \to \mathbb{R}^M$ is invariant if $f(g \cdot x) = f(x)$ holds for any $g \in G$ and any $x \in \mathbb{R}^N$, and equivariant if $f(g \cdot x) = g \cdot f(x)$ holds for any $g \in G$ and any $x \in \mathbb{R}^N$.*

The next operator, called the Reynolds operator, plays a central role in this paper.

Definition 2 (Reynolds Operator (cf. Mumford et al. (1994), Definition 1.5))**.** *For a group G, the followings are called the equivariant and invariant Reynolds operator respectively:*

$$\tau_G(f(-)) = \frac{1}{|G|}\sum_{g\in G} g^{-1} \cdot f(g \cdot -), \quad \gamma_G(f(-)) = \frac{1}{|G|}\sum_{g\in G} f(g \cdot -). \tag{2}$$

More generally, for a subset H in G[*1], we define τ_H and γ_H by replacing G by H in (2).

The equivariant Reynolds operator converts an arbitrary map to a equivariant map. Our idea is to use this operator to convert a fully connected deep neural network into an equivariant function. However, as we see in a moment, the computational complexity of Reynolds operators increases when the order of G is large, for example, for symmetric groups of order $n!$.

We define the following notion of Reynolds design to reduce the computational complexity. The concept of Reynolds design is first proposed in this paper.

Definition 3 (Reynolds Design)**.** *The Reynolds design H of a function f is a subset H of G that satisfies the following equation.*

$$\tau_G(f(-)) = \frac{1}{|G|}\sum_{g\in G} g^{-1} \cdot f(g \cdot -) = \frac{1}{|H|}\sum_{g\in H} g^{-1} \cdot f(g \cdot -).$$

In this case, τ_H is called the reductive Reynolds operator. Furthermore, the Reynolds design H of a set of functions $\mathcal{F}$ is defined by the property that the above equation for H holds for all $f \in \mathcal{F}$.

As an example, consider the following two cases. (1) When f is an invariant function In this case, $f = \tau_G(f)$ is valid. In other words, we can take $H = \{id\}$ as the Reynolds design. (2) When f is a power of one variable. Since $\tau_{S_n}\left(x_1^i\right) = \frac{1}{n}(x_1^i + \cdots + x_n^i)$, the number of terms in the summation

[*1] The subset H in G may not be a subgroup of G.

Young Diagram $\mathbf{k}$	(1,1,1)	(2,1)	(2,1)	(2,1)	(3)
Basis Tableau $\mathbf{T}$	1 2 3	1 2 3	1 3 2	2 3 1	1 2 3
Basis Tableau Vector $\mathbf{e_T}$	$\mathbf{e}_{123}$	$\mathbf{e}_{112}$	$\mathbf{e}_{121}$	$\mathbf{e}_{211}$	$\mathbf{e}_{111}$

Figure 1: Young diagrams and basis tableaux for $m = 3$, and the corresponding basis tableau vectors in $\mathbb{R}^{n^3} = \mathbb{R}^n \otimes \mathbb{R}^n \otimes \mathbb{R}^n$.

is only n, which can be written as an orbit by the cyclic group C_n of order n as $\tau_{S_n}\left(x_1^i\right) = \frac{1}{n}(x_1^i + \cdots + x_n^i) = \tau_{C_n}\left(x_1^i\right)$. Hence C_n is the Reynolds design of x_1^i.

As these examples show, the Reynolds design significantly reduces the computational complexity of the Reynolds operator. Our model is based on such a computational principle.

4 Representation Theorem for Equivariant Maps

In this section we see a representation theorem for equivariant functions by discussing a mathematical object called a Young diagram. An important application of the representation theorem is in the case of graph representations, where the Reynolds design H can be of order n^2 for graphs with n-nodes. As we will see later, this can be used to construct a universal equivariant/invariant graph model where the Reynolds operator requires only n^2 calculations.

4.1 Higher order tensors and adjacency matrices

Graphs and hypergraphs can be represented as adjacency matrices or higher order tensors when the ordering of their nodes is fixed. The main problem with using this representation for deep learning is that the trained model may produce different outputs depending on the permutation of node orderings. For example, in a graph classification problem, we cannot produce different classification results by replacing the ordering of the nodes. Therefore, we want the model to produce the same output. To avoid this problem, we define the appropriate action induced by node permutation in the adjacency matrix or higher order tensor. The action of S_n on tensors $\mathbf{X} \in \mathbb{R}^{n^m \times a}$ (the last index, denoted α represents feature depth) is defined by $(g \cdot \mathbf{X})_{i_1 \ldots i_l, \alpha} = \mathbf{X}_{g^{-1}(i_1) \ldots g^{-1}(i_l), \alpha}$. Note that when $m = 1$, this action represents the permutation of the index on the vector representation $(x_1, .., x_n)$ of the set $\{x_1, .., x_n\}$, and when $m = 2$, it represents the action of the node permutation on the adjacency matrix of the graph.

4.2 Basis and Young Diagrams

Young diagrams are combinatorial objects that are famous for having a one-to-one correspondence with irreducible representations of symmetric groups. We discussed representations of symmetric groups on the spaces of matrices and higher-order tensors, and since representations in general decompose into sums of irreducible representations, we can discuss these spaces in terms of Young diagrams. We prove a representation theorem for equivariant functions based on this mathematical background. Consider an example, where $e_{121} = e_1 \otimes e_2 \otimes e_1$, one of the bases of $\mathbb{R}^{n^3} = \mathbb{R}^n \otimes \mathbb{R}^n \otimes \mathbb{R}^n$. , is represented by a sequence of boxes with numbers in them. We consider representing $e_{121} = e_1 \otimes e_2 \otimes e_1$, one of the bases of $\mathbb{R}^{n^3} = \mathbb{R}^n \otimes \mathbb{R}^n \otimes \mathbb{R}^n$, by arranging boxes and putting numbers in them. The following procedure can be used to construct the representation by the young diagram. First, since the number of e_1 in e_{121} is two, we put two boxes in the first line, and since the number of e_2 in e_{121} is one, we put one box in the second line. Next, the tensor numbers where e_1 appears are 1 and 3, so put 1 and 3, respectively, in the box on the first line and the tensor number where e_2 appears is 2, so put 2 in the box on the second line. Thus we obtain the diagram in Figure

1 The general case of this procedure is as follows; For $e_{\mathbf{u}} = e_{u_1} \otimes e_{u_2} \otimes \cdots e_{u_n} \in \mathbb{R}^{n^m}$, first put as many boxes on the first row as there are e_1's appearing in $e_{\mathbf{u}}$. Then put as many boxes on the next line as there are e_2's appearing in $e_{\mathbf{u}}$. ...Put as many boxes on the last line there are e_n's appearing in $e_{\mathbf{u}}$. Next, put the tensor numbers where e_1 appears into each box in the first row, in decreasing order. Put the tensor numbers where e_2 appears into each box in the next row, in decreasing order. ... Put the tensor numbers in which e_n appears in each box of the last line in decreasing order. This procedure allows us to represent the basis of a tensor space as a combination of Young diagrams and numbers.

4.3 Basis Tableau and Representation Theorem

In order to justify the above procedure, we first define a Young diagram, which is the set of boxes in the above example. A Young diagram is a way to represent the division of a natural number m. Here, division means to express a natural number m as a sum of several non-negative integers.

Definition 4 (Young Diagram). *Let $D \in [m]$ be fixed. A vector $\mathbf{k} = (k_1, \ldots, k_D)$ of natural numbers $k_1, \ldots, k_D$ is called a Young diagram if it satisfies $m = k_1 + \ldots + k_D$ and $k_1 \geq \ldots \geq k_D$.*

This partition can be represented by a total of m boxes, consisting of D rows with k_i boxes in the d-th row. Here, each row is left-justified. Next, we define a basis tableau, which is a Young diagram with numbers in it.

Definition 5 (Basis Tableau). *Let $n \geq m$ and $\mathbf{k} = (k_1, \ldots, k_D)$ a Young diagram. A vector $\mathbf{T} = (\mathbf{t}_1, \ldots, \mathbf{t}_D)$ of vectors $\mathbf{t}_d = (t_{d,1}, \ldots, t_{d,k_d}) \in [n]^{k_d}$ is called basis tableau of depth D if it satisfies the following conditions:*

1. *(Different components) $t_{d,w} \neq t_{d',w'}$ for $(d, w) \neq (d', w')$.*
2. *(Row monotonicity) $t_{d,1} < t_{d,2} < \ldots < t_{d,k_d}$ for each $d \in [D]$.*
3. *(Partial column monotonicity) If $k_d = k_{d+1}$, then $t_{d,1} < t_{d+1,1}$ for $d = 1, \ldots, D-1$.*

Figure 1 shows an example of basis tableaux for $m = 3$. We denote the set of basis tableaux for m of depth D by $\mathcal{T}_{m,D}$, and the set $\bigcup_{1 \leq D \leq m} \mathcal{T}_{m,D}$ of basis tableaux with at most depth m by $\mathcal{T}_m$. In the above procedure, not all bases become basis tableaux, but only certain special bases become basis tableaux. We call such a basis a basis tableau vector. However, any basis can be converted to a basis tableau vector when moved by the action of the symmetric group. This fact is the core of the proof of Theorem 7.

Lemma 6 (Basis Tableau Vector). *Let $n \geq m$. For any $D \in [m]$ and $\mathbf{T} = [\mathbf{t}_1, \ldots, \mathbf{t}_D] \in \mathcal{T}_{m,D}$, then there exists an unique vector $\mathbf{e}_{\mathbf{T}} \in \mathbb{R}^{n^m}$, which gives the basis tableau $\mathbf{T}$ in the above procedure.*

Figure 1 shows an example of basis tableaux vectors when $m = 3$. Next, we review the linear map induced by higher-order tensors. For $\mathbf{X} \in \mathbb{R}^{n^m}$, $\mathbf{X}$ induces the linear map by

$$\hat{\mathbf{X}}_b : \mathbb{R}^b \ni (a_1, \ldots, a_b) \mapsto (a_1 \mathbf{X}, \ldots, a_b \mathbf{X}) \in \mathbb{R}^{n^m \times b}. \tag{3}$$

In the following, we mainly treat the case where $\mathbf{X} = \mathbf{e}_{\mathbf{T}}$ (i.e., $\hat{\mathbf{X}}_b = \hat{\mathbf{e}}_{\mathbf{T},b}$). Then, we have the following theorem.

Theorem 7 (Representation Theorem). *Let $n \geq m$ and $G = S_n$. For any equivariant continuous map $F : \mathbb{R}^{n^l \times a} \to \mathbb{R}^{n^m \times b}$, there exist continuous maps $F_{\mathbf{T}} : \mathbb{R}^{n^l \times a} \to \mathbb{R}^b$ indexed by basis tableaux $\mathbf{T} \in \mathcal{T}_m$ such that*

$$F = \sum_{D=1}^{m} \sum_{\mathbf{T} \in \mathcal{T}_{m,D}} \tau_{H_D}(\hat{\mathbf{e}}_{\mathbf{T},b} \circ F_{\boldsymbol{T}}),$$

where C_{n-i} denote the cyclic group of order $n-i$ on the set $\{i+1, .., n\}$ and

$$H_D := C_n \circ \cdots \circ C_{n-D+1} = \{\sigma_D \cdot \sigma_{D-1} \cdots \sigma_1 \mid \sigma_i \in C_{n-D+i} \ (i = n, n-1, \ldots, n-D+1)\}.$$

Furthermore, H_D is a Reynolds design of $\hat{\mathbf{e}}_{\mathbf{T},b} \circ F_{\boldsymbol{T}}$ for any $\boldsymbol{T} \in \mathcal{T}_{m,D}$.

4.4 COMPUTIONAL COMPLEXITY OF H_D

The most important application, the case of the graph, corresponds to the $m = 2$ case. In this case, the Young diagrams are $\mathbf{k} = (2)$ and $(1, 1)$, and the basis tableaux is also uniquely numbered in the above two. For the Reynolds design, H_1 is of order n and H_2 is of order $n(n-1)$, so the total computational complexity is $O(n^2)$. The number of summations that were originally required $n!$ times has been reduced to n^2 times, resulting in a significant reduction in the amount of calculation. We can see some experiments on these in the experiments section.

5 EQUIVARIANT REYNETS AND UNIVERSALITY

In this section, we define equivariant models using Theorem 7.

Definition 8 (Equivariant Reynolds Nets)**.** *We assume that $n \geq m$. For any basis tableaux $\mathbf{T} \in \mathcal{T}_m$, Let $\mathcal{N}_{\mathbf{T}} : \mathbb{R}^{n^l \times a} \to \mathbb{R}^b$ be a Multi layer perceptron. The map $\mathcal{E} : \mathbb{R}^{n^l \times a} \to \mathbb{R}^{n^m \times b}$*

$$\mathcal{E} = \sum_{D=1}^{m} \sum_{\mathbf{T} \in \mathcal{T}_{m,D}} \tau_{H_D}(\hat{\mathbf{e}}_{\mathbf{T},b} \circ \mathcal{N}_{\boldsymbol{T}}),$$

is called reduced Equivariant Reynolds Nets (equivariant ReyNets).

Theorem 7 guarantees the universal approximation property of this model.

Theorem 9 (Universality)**.** *We assume that $n \geq m$ and $G = S_n$. Let $F : \mathbb{R}^{n^l \times a} \to \mathbb{R}^{n^m \times b}$ be a continuous equivariant function. For any compact set $K \subset \mathbb{R}^{n^\ell \times a}$, there exists a equivariant Reynolds network that approximates F to an arbitrary precision on K. Namely, equivariant Reynolds nets are a universal approximator for equivariant functions.*

6 INVARIANT REYNETS AND UNIVERSALITY

In this section, we define the invariant models using the equivariant models defined in the previous section. First, we will define invariant Reynolds nets. Our construction is to first stack in an equivariant model, then take its orbit sum, and finally stack the MLP. Our main contribution in this section is that we can reduce the dimensionality of the input space of the MLP, which is the basis of the equivariate model.As an example, let us consider approximating the function $f = x_1^i + ... + x_n^i$. Since f can be written as $f = \Sigma(\tau_G(x^i))$ using the orbit sum Σ, only one variable is needed as input. We can generalize these properties and define an invariant called Reynolds dimension. Finally, we explain the universal approximation of invariant networks using Reynolds dimension. We will deal with the case where a general finite group G acts on an n-dimensional vector space or a set of n variables. Note that such formulations include the case of higher order tensor space as $G = S_n$.

Definition 10 (Invariant Reynolds Nets)**.** *A invariant Reynolds network (invariant ReyNet) is a function $\mathcal{I} : \mathbb{R}^{n^l \times a} \to \mathbb{R}$ defined as*

$$\mathcal{I} = \mathcal{M} \circ \Sigma \circ \mathcal{E},$$

where $\mathcal{E} : \mathbb{R}^{n^l \times a} \to \mathbb{R}^{n^m \times b}$ is a equivariant Reynolds network, Σ is the orbit sum[*2], *and $\mathcal{M}$ is a Multi-Layer Perceptron (MLP).*

We define the notion of generator of invariant polynomials, which is necessary to define Reynolds dimension.

Definition 11 (Generator of invariant polynomials)**.** *Let G be a finite group. A set of invariant polynomials $r_1(x_1, .., x_n), .., r_s(x_1, .., x_n)$ are called a generator of invariant polynomials if for any invariant polynomial $f(x_1, .., x_n)$, there is a polynomial $p(y_1, .., y_s)$ such that $f(x_1, .., x_n) = p(r_1(x_1, .., x_n), .., r_s(x_1, .., x_n))$ holds.*

The existence of such a generator is non-trivial for general group actions, but has been shown by Hilbert for finite groups or more generally linearly reductive groups Hilbert (1890). Under the preparation so far, we define the following Reynolds dimension.

[*2]The orbit sum $\Sigma : \mathbb{R}^{[n]^m \times b} \to \mathbb{R}^{[n]^m/G \times b}$ is defined by $\Sigma(\mathbf{X})_{G\cdot\mathbf{u},\beta} := \sum_{g \in G} x_{g\cdot\mathbf{u},\beta}$ for $\mathbf{X} = [x_{\mathbf{u},\beta}] \in \mathbb{R}^{[n]^m \times b}$, $\mathbf{u} \in [n]^m$, $\beta \in [b]$ and $G \cdot \mathbf{u} \in [n]^m/G$.

Definition 12 (Reynolds dimension)**.** *Let G be a finite group. The smallest natural number d satisfying the following property is the Reynolds dimension of the group G. There exist polynomial $h_1, .., h_s$ of d-variable and an index subset $\{j_1, \dots, j_d\} \subset [n]$ such that$\gamma_G(h_1(x_{j_1}, \dots, x_{j_d})), \dots, \gamma_G(h_s(x_{j_1}, \dots, x_{j_d}))$ is a generator of the invariant polynomials.*

Before discussing universality for invariant functions, let us review some notation. We define $\mathrm{Stab}_G([d])$ to be the set of elements of G for which $x_{j_1}, \dots, x_{j_d}$ are fixed. In addition, $[G/G']$ a complete system of representatives of G/G' for a subgroup $G' \subset G$ is a set of order $|G/G'|$ that satisfies $G = \bigcup_{a \in [G/G']} aG'$.

Proposition 13. *In the same situation as Definition 12, $[G/\mathrm{Stab}_G([d])]$ is a Reynolds design of h_i.*

Theorem 14 (Universality)**.** *We assume that $G = S_n$. Let d be the Reynolds dimension of G. Then, Reyonlds invariant nets constructed above for $\mathcal{E} : \mathbb{R}^{n^l \times a} \to \mathbb{R}^{n^d \times b}$ is a universal approximator for invariant functions $f : \mathbb{R}^{n^\ell \times a} \to \mathbb{R}^b$. More strongly, the input space of $\mathcal{E}$ can be replaced by a composite $\mathcal{E} \circ \mathcal{Z}$ with the the zero padding map*[*3] *$\mathcal{Z} : \mathbb{R}^{d \times a} \to \mathbb{R}^{n^\ell \times a}$.*

7 Reduction of input variables

So far, we have taken the input of $\mathcal{N}$ as an n^ℓ-dimensional space. However, when n or ℓ are large, this can be an obstacle to the calculation. Therefore, we take d-dimensional subspace as an input by Theorem 14. This is called the d-reduced equivariant (invariant) Reynolds network. For example, consider the case $\ell = 2$ and assume that the set of invariant polynomials $\mathbb{R}[\mathbf{x}]^{S_n}$ has Reynolds dimension 4 by the valuables $x_{1,1}, x_{1,2}, x_{2,1}, x_{2,2}$, then by Theorem 14, we can restrict the input space of $f_{\mathbf{T}}$ in our model to $x_{1,1}, x_{1,2}, x_{2,1}, x_{2,2}$ and still preserve universality for invariant functions. The reduced Reynolds networks give us the great benefit of generalization over n. In other words, by transferring $f_{\mathbf{T}}$ learned for small n to $f_{\mathbf{T}}$ for large n, we can obtain a model for large n. See Figure 3 for more details. When we consider a practical task, for example, the problem of classifying graphs, it is not a simple matter to determine how many Reynolds dimensions it has. However, this problem has been formulated mathematically as a problem in higher-order tensor spaces, and we expect that the Reynolds dimension will be independent of n in this case.

Conjecture 1. *Let G be an S_n acting on $\mathbb{R}^{n^\ell}$ as in Section 4. For arbitrary enough large n, the Reynolds dimension $d(n)$ of G is independent of n.*

8 Relationship with the parameter sharing models

The models of Zaheer et al. (2017) and Maron et al. (2018) are called parameter sharing models, which are defined by sharing parameters in matrices. Such models are obtained by computing the action of the group on the matrix parameters, and are essentially different from the proposed models, which are obtained from the action of the group on the functions. It is known that the parameters of a neural net do not correspond completely to the function, so the proposed method of looking at the action on the function seems to be more essential. In addition, although Maron et al. (2018) can be represented exactly by the proposed model (Theorem 7), it is a non-trivial question whether the proposed model can be represented exactly byMaron et al. (2018) .

9 Experiments

We evaluated our models compared to fully-connected neural nets (FNNs) and Maron et al. (2018) using synthetic datasets for equivariant and invariant tasks. Furthermore, we chose three graph benchmark datasets as real data for invariant tasks. Please refer to Appendix for implementation details of both experiments.

[*3]The zero padding map $\mathcal{Z} : \mathbb{R}^{d \times a} = \mathbb{R}^d \otimes \mathbb{R}^a \to \mathbb{R}^{n^\ell \times a} = \mathbb{R}^{n^\ell} \otimes \mathbb{R}^a$ is the linear map defined by $\mathcal{Z}((x_1, \dots, x_d) \otimes \mathbf{e}_\alpha) := (x_1, \dots, x_d, \underbrace{0, \dots, 0}_{n^\ell - d}) \otimes \mathbf{e}_\alpha$ for $\alpha = 1, 2, \dots, a$.

Table 1: Results of comparison to a baseline method

Task	Symmetry				Diagonal			
n	3	5	10	20	3	5	10	20
FNN	1.730e-4	9.180e-4	1.454e-3	3.0583	1.295e-4	2.655e-4	1.148e-4	1.081e-1
Maron et al. (2018)	6.600e-3	3.786e-3	9.294e-4	4.471e-3	2.065e-3	2.266e-3	4.098e-3	4.743e-4
ReyNet (ours)	2.147e-4	3.960e-4	1.408e-3	3.151e-3	1.007e-4	2.472e-4	6.635e-4	1.112e-4
4red-ReyNet (ours)	**8.544e-5**	**4.889e-5**	**7.529e-5**	**6.554e-5**	**6.947e-5**	**1.932e-5**	**5.568e-5**	**3.566e-5**
Task	Power				Trace			
n	3	5	10	20	3	5	10	20
FNN	2.586	3.091e+1	5.756e+1	6.268e+2	**2.821e-4**	**1.135e-3**	9.529e-3	5.149e-2
Maron et al. (2018)	4.036e-1	3.462e-1	7.062e-1	4.735e-1	1.241e-3	6.696e-3	3.663e-2	5.527e-2
ReyNet (ours)	3.798e-1	1.257	3.620	3.065	1.884e-3	2.949e-3	4.275e-2	5.338e-2
4red-ReyNet (ours)	**1.204e-1**	**1.330e-1**	**1.217e-1**	**1.165e-1**	3.491e-4	1.914e-3	**6.758e-3**	**1.220e-2**

9.1 SYNTHETHIC DATASETS

We created four types of synthetic datasets for the comparison. Given an input matrics data $\mathbf{A} \in \mathbb{R}^{n \times n}$, each task is defined by:

- **Symmetry**: projection onto the symmetric matrices $F(\mathbf{A}) = \frac{1}{2}(\mathbf{A} + \mathbf{A}^\top)$,
- **Diagonal**: diagonal extraction $F(\mathbf{A}) = diag(\mathbf{A})$,
- **Power**: computing each squared element $F(\mathbf{A}) = [A_{i,j}^2]$,
- **Trace**: computing the trace $F(\mathbf{A}) = tr(\mathbf{A})$,

where the task function F is equivariant in *symmetry*, *diagonal*, and *power*, and is invariant in *trace*. Following Maron et al. (2018), we sampled i.i.d. random matrices $\mathbf{A}$ with uniform distribution in $[0, 10]$; then we transformed it by the task function F. In our experiments, we provided $n \in \{3, 5, 10, 20\}$, and the size of training dataset and test dataset is 1000 respectively.

Objective Function. In terms of an objective function, we modified mean squared error (MSE) loss based on Theorem 7. In a straightforward manner, equivariant tasks are regression task so that MSE loss is selected to decrease the gap of all elements between an output matrix and ground truth matrix: $\ell_{std} \colon \mathbb{R}^{n \times n} \times \mathbb{R}^{n \times n} \to \mathbb{R}$. However, thanks to Theorem 7, ReyNet does not require calculating whole elements but the gap of 1st row of 1st column element and 1st row of 2nd column element: $\ell_{corner} \colon \mathbb{R}^{1 \times 2} \times \mathbb{R}^{1 \times 2} \to \mathbb{R}$. In this paper, we call the latter objective function as corner MSE loss.

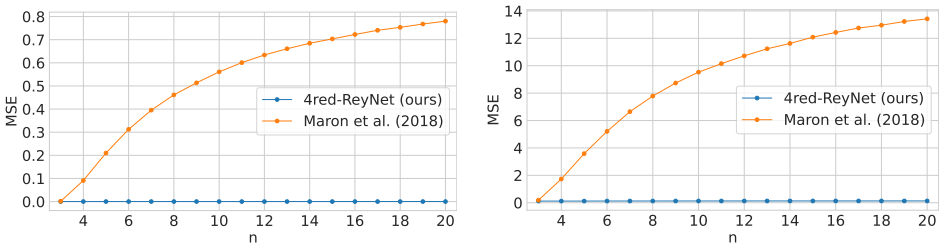
9.1.1 RESULTS

Table 1 shows the result of synthetic datasets, which is the average of MSE of five different seeds. As a result, our proposed models outperform Maron et al. (2018) overall. Interestingly, with regards to the *power* task, FNN and ReyNet are worse than other models. The function F of *power* task amplifies the output error to large value. Simultaneously, when the number of parameters of the model is very large along with the size of n, the output of the model becomes sensitive. Therefore, we considered that the error became large due to the combination with large neural networks and the *power* function.

Table 2: Comparison of Objective Function

n	mse	corner mse
3	1.438e-4	**1.249e-4**
5	**4.912e-5**	9.375e-5
10	1.157e-4	**6.487e-5**
20	1.608e-4	**8.537e-5**

Moreover, we validated the effect of an objective function. We trained 4r-ReyNets with standard MSE loss and Corner MSE loss respectively. Table 2 shows the result of using each objective function. Accordingly, Corner MSE loss achieved lower error than standard MSE loss except for $n = 5$ case. Therefore, using Corner MSE loss is a good choice for equivariant tasks.



(a) Symmetry (b) Power

Figure 2: Generalization Error of the Regression Task on two Synthetic Datasets.

Table 3: Graph Benchmark Results(Acc.)

	MUTAG	IMDB-BINARY	IMDB-MULTI
Maron et al. (2018)	0.822 ± 0.088	0.647 ± 0.075	0.411 ± 0.084
4red-ReyNet (ours)	0.853 ± 0.066	0.710 ± 0.061	0.476 ± 0.029

9.1.2 Generalization Error

Notably, our 4red-ReyNet does not depend on the size of inputs n. In order to test the generalization performance, we trained 4red-ReyNet with $n_{train} = 3$ dataset and then validated the MSE on $n_{test} \in \{3, 4, \ldots, 20\}$ datasets. The results are depicted in Figure 2. We can see that our 4-reduced ReyNet is generalized to the input size. Note that with regard to invariant tasks, we confirmed the model is not generalized to the tasks as Maron et al. (2018) has reported.

9.2 Graph Benchmark Datasets

We selected three benchmark datasets from the TU Dortmund data collection(Kersting et al., 2016): MUTAG dataset, IMDB-BINARY, and IMDB-MULTI. These datasets are provided for classification, thus we treated the experiments as invariant tasks. Since the size of these datasets are small, we run the 10-folds cross-validation score for each dataset.

9.2.1 Results

We report the average accuracy scores and standard deviations of 10 experiments in Table 3. As a result, our ReyNet outperforms the model of Maron et al. (2018) in the three datasets. With reference to practical use, our ReyNet has a limitation. Since our ReyNet requires calculating and storing the tensor at least $\mathcal{O}(n^2)$, it is difficult to run other TU Dortmund data collection (PTC, PROTEINS etc.) through lack of GPU memory. However, by combining our ReyNet with other methods (e.g. stochastic gradient descent from a uniform distribution on a Reynolds design), we believe that this limitation can be overcome.

10 Conclusion

We considered equivariant and invariant neural networks over higher order tensor spaces. The method of converting deep neural nets to equivariant neural nets using Reynolds operators had some difficulties in terms of computational complexity, which can be successfully avoided by using Reynolds designs. Then, we constructed equivariant Reynolds networks (equivariant ReyNets) based on the Reynolds designs and proved the universality. Similarly, we also introduced Reynolds designs induced by Reynolds dimension in the invariant case. Then, we also constructed invariant Reynolds networks (invariant ReyNets) and proved the universality as well. Furthermore, we showed that input variables of ReyNets can be reduced based on the Reynolds dimension. In the section of numerical experiments, we showed that equivariant and invariant ReyNets performs better than or comparable to existing models. Moreover, we observed that ReyNets with a few input variables can generalize well to the cases with more input variables.

REFERENCES

Eiichi Bannai and Etsuko Bannai. A survey on spherical designs and algebraic combinatorics on spheres. *European Journal of Combinatorics*, 30(6):1392–1425, 2009.

Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.

Zhengdao Chen, Soledad Villar, Lei Chen, and Joan Bruna. On the equivalence between graph isomorphism testing and function approximation with gnns. *arXiv preprint arXiv:1905.12560*, 2019.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29:3844–3852, 2016.

Marco Gori, Gabriele Monfardini, and Franco Scarselli. A new model for learning in graph domains. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 2, pp. 729–734. IEEE, 2005.

Devon Graham and Siamak Ravanbakhsh. Deep models for relational databases. *arXiv preprint arXiv:1903.09033*, pp. 6, 2019.

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

Mikael Henaff, Joan Bruna, and Yann LeCun. Deep convolutional networks on graph-structured data. *arXiv preprint arXiv:1506.05163*, 2015.

David Hilbert. Über die theorie der algebraischen formen. *Mathematische Annalen*, 1890.

Nicolas Keriven and Gabriel Peyré. Universal invariant and equivariant graph neural networks. *Advances in Neural Information Processing Systems*, 32:7092–7101, 2019.

Kristian Kersting, Nils M Kriege, Christopher Morris, Petra Mutzel, and Marion Neumann. Benchmark data sets for graph kernels. *URL http://graphkernels. cs. tu-dortmund. de*, 2016.

Piotr Kicki, Piotr Skrzypczyński, and Mete Ozay. A new approach to design symmetry invariant neural networks. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2021.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Risi Kondor, Hy Truong Son, Horace Pan, Brandon Anderson, and Shubhendu Trivedi. Covariant compositional networks for learning graphs. *arXiv preprint arXiv:1801.02144*, 2018.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

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

Ron Levie, Federico Monti, Xavier Bresson, and Michael M Bronstein. Cayleynets: Graph convolutional neural networks with complex rational spectral filters. *IEEE Transactions on Signal Processing*, 67(1):97–109, 2018.

Takanori Maehara and Hoang NT. A simple proof of the universality of invariant/equivariant graph neural networks. *arXiv preprint arXiv:1910.03802*, 2019.

Haggai Maron, Heli Ben-Hamu, Nadav Shamir, and Yaron Lipman. Invariant and equivariant graph networks. *arXiv preprint arXiv:1812.09902*, 2018.

Haggai Maron, Heli Ben-Hamu, Hadar Serviansky, and Yaron Lipman. Provably powerful graph networks. *arXiv preprint arXiv:1905.11136*, 2019a.

Haggai Maron, Ethan Fetaya, Nimrod Segol, and Yaron Lipman. On the universality of invariant networks. *arXiv preprint arXiv:1901.09342*, 2019b.

Haggai Maron, Or Litany, Gal Chechik, and Ethan Fetaya. On learning sets of symmetric elements. *arXiv preprint arXiv:2002.08599*, 2020.

S Chandra Mouli and Bruno Ribeiro. Neural networks for learning counterfactual g-invariances from single environments. In *International Conference on Learning Representations*, 2020.

David Mumford, John Fogarty, and Frances Kirwan. *Geometric invariant theory*, volume 34. Springer Science & Business Media, 1994.

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

Hoang Nguyen and Takanori Maehara. Graph homomorphism convolution. In *International Conference on Machine Learning*, pp. 7306–7316. PMLR, 2020.

Kenta Oono and Taiji Suzuki. Graph neural networks exponentially lose expressive power for node classification. In *International Conference on Learning Representations*, 2019.

Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 652–660, 2017.

Siamak Ravanbakhsh, Jeff Schneider, and Barnabas Poczos. Equivariance through parameter-sharing. In *International Conference on Machine Learning*, pp. 2892–2901. PMLR, 2017.

Bruce Sagan. *The symmetric group: representations, combinatorial algorithms, and symmetric functions*, volume 203. Springer Science & Business Media, 2001.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.

Nimrod Segol and Yaron Lipman. On universal equivariant set networks. *arXiv preprint arXiv:1910.02421*, 2019.

Elise van der Pol, Daniel Worrall, Herke van Hoof, Frans Oliehoek, and Max Welling. Mdp homomorphic networks: Group symmetries in reinforcement learning. *Advances in Neural Information Processing Systems*, 33, 2020.

Dmitry Yarotsky. Universal approximations of invariant maps by neural networks. *Constructive Approximation*, pp. 1–68, 2021.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan R Salakhutdinov, and Alexander J Smola. Deep sets. In *Advances in neural information processing systems*, pp. 3391–3401, 2017.

A PROOF OF THEOREMS AND PROPOSITIONS

A.1 PROOF OF LEMMA 6

Firstly, we give some notation about basis to state. Let $\mathbf{e}_1, \dots, \mathbf{e}_n$ the standard basis of $\mathbb{R}^n$. As we saw above, for $\mathbf{u} = (u_1, \dots, u_m) \in [n]^m$, we set

$$\mathbf{e}_{\mathbf{u}} := \mathbf{e}_{u_1,\dots,u_m} = \mathbf{e}_{u_1} \otimes \cdots \otimes \mathbf{e}_{u_m} \in \underbrace{\mathbb{R}^n \otimes \cdots \otimes \mathbb{R}^n}_{m} = \mathbb{R}^{n^m}. \tag{4}$$

For any basis tableau $\mathbf{T} \in \mathcal{T}_{m,D}$, natural numbers $\mathbf{u} = (u_1, \dots, u_m) \in [n]^m$ are given as $u_\ell := d \in [n] \quad \textit{if} \quad \ell \in \{\mathbf{t}_d\}$.[*4] Then, we define the map $\phi : \mathcal{T}_m \to [n]^m$ by $\phi(\mathbf{T}) := (u_1, .., u_m) \in [n]^m$. Then the basis tableau vector is defined by $\mathbf{e}_{\mathbf{T}} := \mathbf{e}_{\phi(\mathbf{T})} \in \mathbb{R}^{n^m}$, where the right hand side is defined by (4).

A.2 PROOF OF THEOREM 7

We introduce the following notion to obtain tableau-based representation of elements in $[n]^m$.

Definition 15 (Extended Tableau)**.** *Let $n \geq D$. Let $[n]^D_\#$ be the set of D different natural numbers at most n defined by $[n]^D_\# := \{[j_1, \dots, j_D] \in [n]^D \mid j_a \neq j_b$ for $a \neq b \in [D]\}$. Then, we call elements in $\tilde{\mathcal{T}}_{m,D} := [n]^D_\# \times \mathcal{T}_{m,D}$ extended tableaux with depth D.*

We denote the set $\bigcup_{1 \leq D \leq m} \tilde{\mathcal{T}}_{m,D}$ of extended tableaux with at most depth m by $\tilde{\mathcal{T}}_m$. The action $g \cdot [j_1, \dots, j_D] := [g \cdot j_1, \dots, g \cdot j_D]$ of G on $[n]^D_\#$ is well-defined. Then, we define the action of G on $\tilde{\mathcal{T}}_{m,D}$ by $g \cdot (\mathbf{j}, \mathbf{T}) := (g \cdot \mathbf{j}, \mathbf{T})$. In the following, we identify an extended tableau $([1, 2, \dots, D], \mathbf{T})$ with the basis tableau $\mathbf{T}$.

Next, we introduce some notation to represent elements in $[n]^m$ by extended tableaux. We first define the partial order $\succ$ of vectors of natural numbers that can have different dimensions. For $\mathbf{t} = [t_1, \dots, t_k] \in \mathbb{N}^k$ and $\mathbf{t}' = [t'_1, \dots, t'_{k'}] \in \mathbb{N}^{k'}$, we denote as $\mathbf{t} \succ \mathbf{t}'$ if either (i) $k > k'$ or (ii) $k = k'$ and $t_1 > t'_1$. We note that $\mathbf{t}_1 \succ \dots \succ \mathbf{t}_D$ holds for a basis tableau $\mathbf{T} = [\mathbf{t}_1, \dots, \mathbf{t}_D]$ by definition. Let $\mathbf{u} = [u_1, \dots, u_k] \in [n]^m$. We set $D(\mathbf{u}) := |\{\mathbf{u}\}|$. We define the multiplicity map $\mathrm{mult}_{\mathbf{u}} : \{\mathbf{u}\} \to [m]$ by $\mathrm{mult}_{\mathbf{u}}(u) := |\{\ell \in [m] \mid u_\ell = u\}|$. For $u \in \{\mathbf{u}\}$, we define $\mathbf{t}_u := [t_1, \dots, t_{k_u}]$, where $k_u := \mathrm{mult}_{\mathbf{u}}(u)$, $u_{t_1} = \cdots = \boldsymbol{u}_{t_{k_u}} = \mathbf{u}$, and $t_1 < \cdots < t_{k_u}$. Here, we note that either $\mathbf{t}_u \succ \mathbf{t}_{u'}$ or $\mathbf{t}_u \prec \mathbf{t}_{u'}$ holds for $u \neq u' \in \{\mathbf{u}\}$ by definition. Thus, there exist different natural numbers $j_1, \dots, j_D \subset [n]$ such that $\{j_1, \dots, j_D\} = \{\mathbf{u}\}$ and $\mathbf{t}_{j_1} \succ \dots \succ \mathbf{t}_{j_D}$.

Definition 16 (Tableau Representation)**.** *Let $n \geq m$. We define the map $\psi : [n]^m \ni \mathbf{u} \mapsto (\mathbf{j}, \mathbf{T}) \in \tilde{\mathcal{T}}_m$ called tableau representation by*

$$\psi(\mathbf{u}) := ([j_1, \dots, j_D], [\mathbf{t}_{j_1}, \dots, \mathbf{t}_{j_D}]) \in \tilde{\mathcal{T}}_{m,D}.$$

We define the map $\phi : \tilde{\mathcal{T}}_m \to [n]^m$ as follows: For any extended tableau $(\mathbf{j}, \mathbf{T}) \in \tilde{\mathcal{T}}_{m,D}$, natural numbers $\mathbf{u} = [u_1, \dots, u_m] \in [n]^m$ are given as $u_\ell := j_d \in [n] \quad \textit{if} \quad \ell \in \{\mathbf{t}_d\}$, and $\phi(\mathbf{j}, \mathbf{T}) := (u_1, .., u_m) \in [n]^m$.

Lemma 17. *When $n \geq m$, the tableau representation $\psi : [n]^m \to \tilde{\mathcal{T}}_m$ is bijective and $\psi(g \cdot \mathbf{u}) = g \cdot \psi(\mathbf{u})$ for $g \in H_D$.*

[Proof] . First, from the construction, ψ is injective and $\phi \circ \psi = \mathrm{id}_{[n]^m}$. Since $[n]^m$ and $\mathcal{T}$ are finite sets, if we show that ϕ is injective, then ψ is bijective. For the extended tableau $\mathbf{S}, \mathbf{T}$, assume that $\phi(\mathbf{S}) = \phi(\mathbf{T})$. Note that the set of row vectors $\{\mathbf{s}_1, ..., \mathbf{s}_d\}$ of $\mathbf{S}$ is uniquely determined from $\phi(\mathbf{S})$. Then, since the extended tableaux satisfy the order $\succ$ in which this goes between the row vectors, the extended tableau $\mathbf{S}$ having row vectors $\{\mathbf{s}_1, ..., \mathbf{s}_d\}$ is unique. Hence, we have $\mathbf{S} = \mathbf{T}$. ■

Definition 18 (Extended Tableau Vector)**.** *Let $n \geq m$. The extended tableau vector $\mathbf{e}_{\mathbf{j},\mathbf{T}}$ is defined by $\mathbf{e}_{\mathbf{j},\mathbf{T}} := \mathbf{e}_{\phi(\mathbf{j},\mathbf{T})}$.*

[*4] For a vector $\mathbf{t} = [t_1, \dots, t_k]$, we set $\{\mathbf{t}\} := \{t_1, \dots, t_k\}$. For example, when $\mathbf{t} = [1, 1]$, $\{\mathbf{t}\} = \{1\}$.

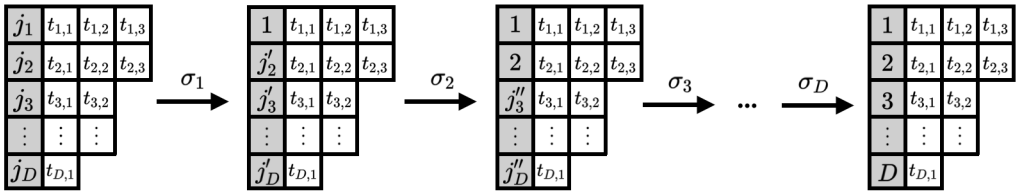


Figure 3: Reduction from an extended tableau to a basis tableau by permutations. An extended tableau $(\mathbf{j}, \mathbf{T})$ is converted to a basis tableau by multiplying $g = \sigma_D \cdot \sigma_{D-1} \cdots \sigma_1$.

For an extended tableau $(\mathbf{j}, \mathbf{T}) \in \tilde{\mathcal{T}}_m$, the linear map $\hat{\mathbf{e}}_{\mathbf{j},\mathbf{T},b} : \mathbb{R}^b \to \mathbb{R}^{n^m \times b}$ is defined by $\mathbf{X} = \mathbf{e}_{\mathbf{j},\mathbf{T}}$ in (3).

Lemma 19 (Normalization)**.** *Let $n \geq D$. For $\mathbf{j} \in [n]^D_{\#}$, there uniquely exists $g \in H_D$ such that $\mathbf{j} = g^{-1} \cdot [1, 2, \ldots, D] \in [n]^D_{\#}$. Hence, this correspondence $[n]^D_{\#} \ni \mathbf{j} \to g \in H_D$ is bijective.*

[Proof] . There uniquely exists $\sigma_1 \in C_n$ such that $\sigma_1(j_1) = 1$. Inductively, there uniquely exists $\sigma_d \in C_{n-d+1}$ such that $\sigma_d(\sigma_{d-1} \cdot \sigma_{d-2} \cdots \sigma_1(j_d)) = d$ for $d = 2, \ldots, D$. Then $g := \sigma_D \cdots \sigma_1 \in H_D$ satisfies $g \cdot \mathbf{j} = [1, 2, \ldots, D]$ by definition. ■

From Lemma 19, an extended tableau $(\mathbf{j}, \mathbf{T}) \in \tilde{\mathcal{T}}_{m,D}$ is uniquely represented by $(\mathbf{j}, \mathbf{T}) = g^{-1} \cdot ([1, \ldots, D], \mathbf{T}) = g^{-1} \cdot \mathbf{T}$ as in Figure 3, where $g \in H_D$ and we identified $\mathbf{T} \in \mathcal{T}_{m,D}$ with $([1, \ldots, D], \mathbf{T}) \in \tilde{\mathcal{T}}_{m,D}$ in the last equation. Thus, we have $\tilde{\mathcal{T}}_{m,D} = \bigcup_{g \in H_D} g^{-1}\mathcal{T}_{m,D}$, and $\tilde{\mathcal{T}}_m = \bigcup_{1 \leq D \leq m} \bigcup_{g \in H_D} g^{-1}\mathcal{T}_{m,D}$. From Lemma 17, for each $\mathbf{u} \in [n]^m$, there uniquely exists $g \in H_D$ and $\mathbf{T} \in \mathcal{T}_m$ such that $\psi(\mathbf{u}) = g^{-1} \cdot \mathbf{T}$ (or equivalently $\mathbf{u} = \psi^{-1}(g^{-1} \cdot \mathbf{T})$). In the following, we omit the bijective ψ for notational simplicity.

In the following, we prove Theorem 7. We can write $F = \sum_{\mathbf{u} \in [n]^m} f_{\mathbf{u}} \cdot \hat{e}_{\mathbf{u},b}$ by maps $f_{\mathbf{u}} : \mathbb{R}^{n^l \times a} \to \mathbb{R}^b$. Then since F is equivariant, we have $\sum_{\mathbf{u} \in [n]^m} f_{\mathbf{u}} \cdot \hat{e}_{\mathbf{u},b}(g \cdot x) = F(g \cdot x) = g \cdot F(x) = \sum_{\mathbf{u} \in [n]^m} f_{\mathbf{u}} \cdot \hat{e}_{g \cdot \mathbf{u},b}(x)$. This implies that

$$f_{\mathbf{u}} \cdot \hat{e}_{\mathbf{u},b}(g \cdot x) = f_{g^{-1} \cdot \mathbf{u}} \cdot \hat{e}_{\mathbf{u},b}(x). \tag{5}$$

Then, we obtain the following equations:

$$\begin{aligned}
F(x) &= \sum_{\mathbf{u} \in [n]^m} f_{\mathbf{u}} \cdot \hat{e}_{\mathbf{u},b}(x) \\
&= \sum_{1 \leq D \leq m} \sum_{g \in H_D} \sum_{\mathbf{T} \in \mathcal{T}_{m,D}} f_{g^{-1} \cdot \mathbf{T}} \cdot \hat{e}_{g^{-1} \cdot \mathbf{T},b}(x) \\
&= \sum_{1 \leq D \leq m} \sum_{\mathbf{T} \in \mathcal{T}_{m,D}} \sum_{g \in H_D} f_{g^{-1} \cdot \mathbf{T}} \cdot \hat{e}_{g^{-1} \cdot \mathbf{T},b}(x) \\
&= \sum_{1 \leq D \leq m} \sum_{\mathbf{T} \in \mathcal{T}_{m,D}} \sum_{g \in H_D} f_{\mathbf{T}} \cdot \hat{e}_{g^{-1} \cdot \mathbf{T},b}(g \cdot x) \\
&= \sum_{1 \leq D \leq m} \sum_{\mathbf{T} \in \mathcal{T}_{m,D}} \sum_{g \in H_D} g^{-1} \cdot (f_{\mathbf{T}} \cdot \hat{e}_{\mathbf{T},b}(g \cdot x)) \\
&= \sum_{1 \leq D \leq m} \sum_{\mathbf{T} \in \mathcal{T}_{m,D}} \sum_{g \in H_D} \frac{1}{|H_D|} g^{-1} \cdot (|H_D| f_{\mathbf{T}} \cdot \hat{e}_{\mathbf{T},b}(g \cdot x)) \\
&= \sum_{1 \leq D \leq m} \sum_{\mathbf{T} \in \mathcal{T}_{m,D}} \tau_{H_D}(|H_D| f_{\mathbf{T}} \cdot \hat{e}_{\mathbf{T},b})(x) \\
&= \sum_{1 \leq D \leq m} \sum_{\mathbf{T} \in \mathcal{T}_{m,D}} \tau_{H_D}(F_{\mathbf{T}} \cdot \hat{e}_{\mathbf{T},b})(x),
\end{aligned}$$

where the fourth equality follows from (5) and the last equality follows by putting $F_{\mathbf{T}} := |H_D| f_{\mathbf{T}}$. ■

B PROOF OF THEOREM 9

By Theorem 7, we have continuous maps $f_{\mathbf{T}} : \mathbb{R}^{n^l \times a} \to \mathbb{R}^b$ satisfying $F = \sum_{D=1}^{m} \sum_{\mathbf{T} \in \mathcal{T}_{m,D}} \tau_{H_D}(f_{\mathbf{T}} \circ \hat{\mathbf{e}}_{\mathbf{T},b})$, for standard Young tableaux $\mathbf{T} \in \mathcal{T}_{m,D}$. Since K is a compact set and S_n is a finite group, we may assume that K is closed under S_n-action by taking $\bigcup_{g \in S_n} g \cdot K$.Then for any ε, we have an MLP $\mathcal{N}_{\mathbf{T}}$ which approximates $f_{\mathbf{T}}$, namely $\|N_T - f_T\|_K < \varepsilon$ holds. Hence by the definition of our invariant model, we have

$$
\begin{aligned}
&\left\| F - \sum_{D=1}^{m} \sum_{\mathbf{T} \in \mathcal{T}_{m,D}} \tau_{H_D}\left(\mathcal{N}_T \cdot \hat{e}_{\mathbf{T},b}\right) \right\|_K \\
&= \| \sum_{D=1}^{m} \sum_{T \in \mathcal{T}_{m,D}} \tau_{H_D}\left(f_T \cdot \hat{e}_{\mathbf{T},b}\right) - \sum_{D=1}^{m} \sum_{T \in \mathcal{T}_{m,D}} \tau_{H_D}\left(N_T \hat{e}_{\mathbf{T},b}\right) \|_K \\
&\leq \sum_{D=1}^{m} \sum_{T \in \mathcal{T}_{m,D}} \| \tau_{H_D}\left(f_T \cdot \hat{e}_{\mathbf{T},b}\right) - \tau_{H_D}\left(N_T \cdot \hat{e}_{T,b}\right) \|_K \\
&\leq \sum_{D=1}^{m} \sum_{T \in \mathcal{T}_{m,D}} \| \tau_{H_D}\left(\left(f_T - N_T\right) \cdot \hat{e}_{T,b}\right) \|_K \\
&\leq \sum_{D=1}^{m} \sum_{T \in \mathcal{T}_{m,D}} \| \sum_{g \in H_D} \frac{1}{|H_D|} \left(\left(f_T - N_T\right) \cdot \hat{e}_{g^{-1} \cdot T,b}(g \cdot -) \right\|_K \\
&\leq \sum_{D=1}^{m} \sum_{T \in \mathcal{T}_{m,D}} \sum_{g \in H_D} \frac{1}{|H_D|} \left\| \left(f_T - N_T\right) \cdot \hat{e}_{g^{-1} \cdot T,b}(g \cdot -) \right\|_K \\
&\leq \sum_{D=1}^{m} \sum_{T \in \mathcal{T}_{m,D}} \sum_{g \in H_D} \frac{1}{|H_D|} \left\| \left(f_T - N_T\right) \cdot \hat{e}_{g^{-1} \cdot T,b}(-) \right\|_K \\
&\leq \sum_{D=1}^{m} \sum_{T \in \mathcal{T}_{m,D}} \sum_{g \in H_D} \frac{1}{|H_D|} \| \left(f_T - N_T\right) \|_K \\
&\leq \sum_{D=1}^{m} \sum_{T \in \mathcal{T}_{m,D}} \sum_{g \in H_D} \frac{1}{|H_D|} \varepsilon \\
&\leq m |\mathcal{T}_{m,D}| \varepsilon.
\end{aligned}
$$

By replacing ε, we obtain $\left\| F - \sum_{D=1}^{m} \sum_{\mathbf{T} \in \mathcal{T}_{m,D}} \tau_{H_D}\left(\mathcal{N}_T \cdot \hat{e}_{\mathbf{T},b}\right) \right\|_K < \varepsilon$. ■

C PROOF OF PROPOSITION 13 AND THEOREM 14

We use the following theorem.

Theorem 20 (Hilbert finiteness theorem (Hilbert (1890)))**.** *Let G be a finite group or, more generally, a linearly reductive group. In this case, there is always a generator of G-invariant polynomials.*

Let $f : \mathbb{R}^{n^\ell \times a} \to \mathbb{R}^b$ be a continuous invariant function. By replacing K with $\bigcup_{g \in G} g \cdot K$, we may assume that K is closed under the action of G. Then by the Stone-Weierstrass theorem, there exists a polynomial $\hat{f} : \mathbb{R}^{n^\ell \times a} \to \mathbb{R}^b$ which approximate f with arbitrary precision on K. Put $\tilde{f} = \gamma_G(\hat{f})$, then

$$
\left\| f(\mathbf{x}) - \gamma_G(\hat{f}(\mathbf{x})) \right\|_K = \frac{1}{|G|} \left\| |G| f(\mathbf{x}) - \sum_{g \in G} f(g \cdot \mathbf{x}) \right\|_K
$$

$$
\begin{aligned}
&\leq \frac{1}{|G|} \sum_{g \in G} \|f(\mathbf{x}) - \hat{f}(g \cdot \mathbf{x})\|_K \\
&= \frac{1}{|G|} \sum_{g \in G} \left\| f\left(g \cdot \mathbf{x}\right) - \hat{f}(g \cdot \mathbf{x}) \right\|_K \\
&\leq \frac{1}{|G|} \sum_{g \in G} \varepsilon = \varepsilon,
\end{aligned}
$$

where we used the property $f(\mathbf{x}) = f(g \cdot \mathbf{x})$ in the third equation. By Theorem 20, we have a generator of invariant polynomials $r_1, \ldots, r_s$. From the definition of generator, there exists a polynomial P and $\tilde{f}$ can be written in the form $\tilde{f}\left(x_1, \ldots, x_{n^l a}\right) = P(r_1\left(x_1, \ldots, x_{n^l a}\right), .., r_s\left(x_1, \ldots, x_{n^l a}\right))$. By the assumption of Reynolds dimension, $r_1\left(x_1, \ldots, x_{n^l a}\right), \ldots, r_s\left(x_1, \ldots, x_{n^l a}\right)$ are written as $\gamma_G\left(h_1\left(x_{j_1}, \ldots, x_{j_d}\right)\right), \ldots, \gamma_G\left(h_s\left(x_{j_1}, \ldots, x_{j_d}\right)\right)$ for some polynomials $h_1\left(x_{j_1}, \ldots, x_{j_d}\right), \ldots, h_s\left(x_{j_1}, \ldots, x_{j_d}\right)$ of d-variables.

Since H_d is a complete system of representative of $G/\operatorname{Stab}([d])$, we obtain the following decomposition:

$$
G = \bigcup_{g \in [G/\operatorname{Stab}_G([d])]} g \cdot \operatorname{Stab}_G([d]) = \bigcup_{g \in H_d} g \cdot \operatorname{Stab}_G([d]).
$$

Then, this induces the decomposition of Reynolds operators;

$$
\gamma_G = \gamma_{H_d} \circ \gamma_{\operatorname{Stab}([d])} : \mathbb{R}\left[x_1, .., x_{n^l a}\right] \to \mathbb{R}\left[x_1, .., x_{n^l a}\right]^{\operatorname{Stab}([d])} \to \mathbb{R}\left[x_1, \ldots, x_{n^l a}\right]^G,
$$

where $\mathbb{R}\left[x_1, .., x_{n^l a}\right], \mathbb{R}\left[x_1, .., x_{n^l a}\right]^{\operatorname{Stab}([d])}, \mathbb{R}\left[x_1, \ldots, x_{n^l a}\right]^G$ are the set of polynomials, $\operatorname{Stab}([d])$-invariant polynomials, invariant polynomials, respectively. This implies

$$
r_i = \gamma_G\left(h_i\left(x_{j_1}, \ldots, x_{j_d}\right)\right) = \gamma_{H_d}\left(\gamma_{\operatorname{Stab}([d])}(h_i\left(x_{j_1}, \ldots, x_{j_d}\right))\right) = \gamma_{H_d}\left(h_i\left(x_{j_1}, \ldots, x_{j_d}\right)\right).
$$

Here the last inequality follows from the fact that $h_i\left(x_{j_1}, \ldots, x_{j_d}\right)$ is a $\operatorname{Stab}_G([d])$-invariant polynomial. On the other hand, note that the invariant Reynolds operator is equal to the composition of the equivariant Reynolds operator and the orbit sum; $\gamma_G = \Sigma \circ \tau_G$. For the vector valued function $h = (h_1, .., h_s)$, by the universal approximation theorem of fully connected neural nets, we can take a fully connected neural net $\mathcal{Q}$ with which $\|\mathcal{Q} - h\|_K < \varepsilon$ holds. Let $\mathcal{N}$ be a fully connected neural net that approximates P above; $\|\mathcal{N} - P\|_K < \varepsilon$. Then

$$
\begin{aligned}
\mathcal{N} \circ \Sigma \circ \tau_{H_d} \circ Q(x_1, .., x_n) &\approx \mathcal{N} \circ \Sigma \circ \tau_{H_d}(h_1(x_1, .., x_n), .., h_s(x_1, .., x_n)) \\
&\approx \mathcal{N}(\gamma_G(h_1)(x_1, .., x_n), .., \gamma_G(h_s)(x_1, .., x_n)) \\
&= \mathcal{N}(r_1(x_1, .., x_n), .., r_s(x_1, .., x_n)) \\
&\approx P(r_1(x_1, .., x_n), .., r_s(x_1, .., x_n)) \\
&= \tilde{f}(x_1, .., x_n) \\
&\approx f(x_1, .., x_n).
\end{aligned}
$$

Here we denote the approximation by $\approx$. ■

D Implementation Details

In the experiments of synthetic datasets, we used two models; Reynolds Networks (ReyNets) and 4-reduced Reynolds Networks (4red-ReyNets), while only 4-reduced Reynolds Networks (4red-ReyNets) for the graph benchmark dataset. For equivariant tasks, we applied equivariant ReyNets. For invariant tasks, we implemented invariant ReyNets following the architecture of the invariant model proposed by Maron et al. (2018)[*5]; an equivariant ReyNet is followed by max pooling[*6], and fully-connected layers. The fully-connected layers consist of three layers and the number of the units are 512, 256, and 1 respectively. We adopted the ReLU function as activation. We used Adam optimizer and set learning rate as 1e-3 and weight decay as 1e-5. Batch size is 100. Note

[*5] Please refer to https://github.com/Haggaim/InvariantGraphNetworks/blob/master/models/invariant_basic.py#L14

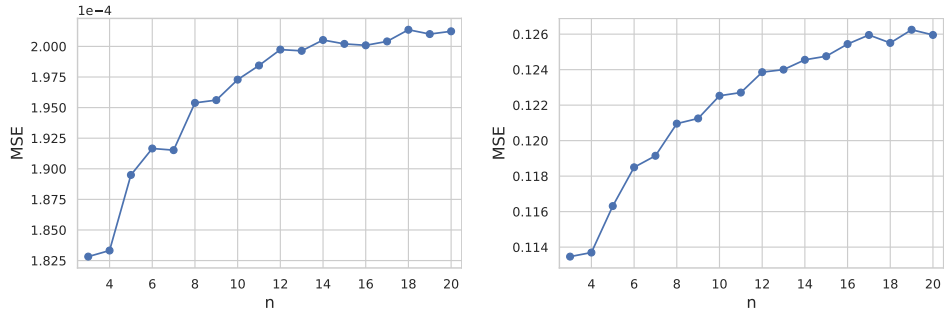
[*6] This operation outputs the max value of diagonal and non-diagonal elements of an input matrix.

that the models of Maron et al. (2018) are re-implemented by PyTorch in reference to the author's implementation in Tensorflow.

For the benchmark datasets, basically, we adopted the same architecture of ReyNet and Maron et al. (2018) as for the synthetic datasets. That is, the architecture consists of two parts; equivariant part and invariant parts (fully-connected layers). However, we set the output dimension of equivariant part of Maron et al. (2018) to the same as the input dimension although we can set any numbers (e.g. 256 dimension). This is because we aimed to evaluate the performance (capability) of equivariant part. If we set the dimension to 256 for example, it is difficult to distinguish whether the high-performance is thanks to equivariant part capability or high dimension representation.

E GENERALIZATION ERROR

Figure 4 shows the detail version of only 4red-ReyNets at Figure 2. In figure 3 our results seems an almost horizontal straight line, but this figure shows that the MSE also slightly increases as n increases.



(a) Symmetry (b) Power

Figure 4: Generalization Error

F RUNTIME SPEED

Figure 5 shows the runtime speed of our ReyNet compared with Maron et al. (2018). The left of Figure 5 shows the runtime speed when the number of input features changes while the size of the set does not. The right of Figure 5 shows the runtime speed when the size of the set changes while the number of input features does not. Given both figures, it seems that our ReyNet is much slower than Maron et al. (2018). However, as a result of the experiments, ReyNet outperforms Maron et al. (2018) in many cases. We expect that much more GPU resources would help ReyNet in the future.

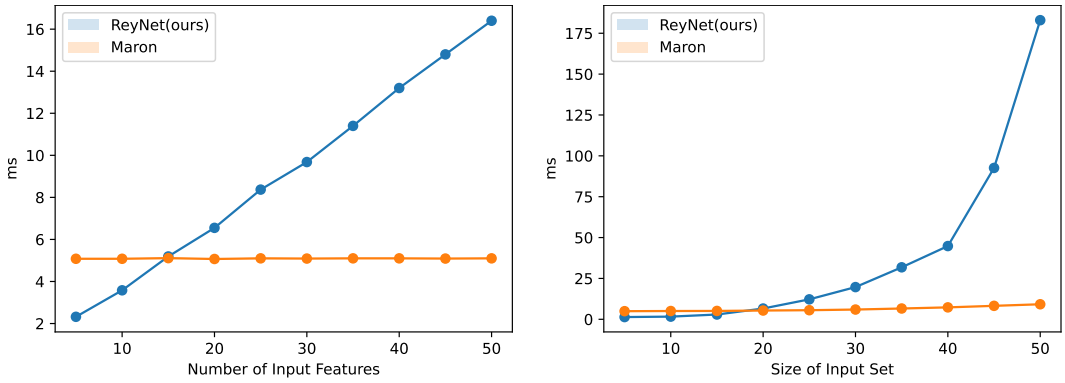


Figure 5: Runtime Speed