Graph Automorphism Group Equivariant Neural Networks

Edward Pearce–Crump [1] **William J. Knottenbelt** [1]

Abstract

Permutation equivariant neural networks are typically used to learn from data that lives on a graph. However, for any graph G that has n vertices, using the symmetric group S_n as its group of symmetries does not take into account the relations that exist between the vertices. Given that the actual group of symmetries is the automorphism group $\mathrm{Aut}(G)$, we show how to construct neural networks that are equivariant to $\mathrm{Aut}(G)$ by obtaining a full characterisation of the learnable, linear, $\mathrm{Aut}(G)$-equivariant functions between layers that are some tensor power of $\mathbb{R}^n$. In particular, we find a spanning set of matrices for these layer functions in the standard basis of $\mathbb{R}^n$. This result has important consequences for learning from data whose group of symmetries is a finite group because a theorem by Frucht (1938) showed that any finite group is isomorphic to the automorphism group of a graph.

1. Introduction

In many cases, the relationships that exist between certain entities can be structured in the form of a graph. These include the interactions between people in a social network (Leskovec et al., 2010), the bonds that connect atoms in a molecule (Gilmer et al., 2017), and the relationships between users and items in a recommendation system (Konstas et al., 2009). As a result, there is a high level of motivation to design neural network architectures that can offer new insights into data with this structure. A number of methods for learning from graphs exist in the machine learning literature. These include Graph Attention Networks (GATs) (Veličković et al., 2018), Graph Convolutional Networks (GCNs) (Kipf & Welling, 2017), and Graph Neural Networks (GNNs) (Gori et al., 2005; Scarselli et al., 2009), among others. In particular, many of these approaches use some form of equivariance to the symmetric group, which has since been characterised fully by a number of authors (Maron et al., 2019; Ravanbakhsh, 2020; Pearce-Crump, 2022). However, using neural networks that are equivariant to the symmetric group does not take into account the relations that exist between the vertices in a graph. Given that the actual group of symmetries of a graph is its automorphism group, we would like to construct neural networks that satisfy the stronger condition of being equivariant to this group instead.

In this paper, writing the automorphism group of a graph G having some n vertices as $\mathrm{Aut}(G)$, we give a full characterisation of all of the possible $\mathrm{Aut}(G)$-equivariant neural networks whose layers are some tensor power of $\mathbb{R}^n$ by finding a spanning set of matrices for the learnable, linear, $\mathrm{Aut}(G)$-equivariant layer functions between such tensor power spaces in the standard basis of $\mathbb{R}^n$. Our approach is similar to the one seen in Pearce-Crump (2022; 2023a;b), where they used the combinatorics of set partitions to characterise the learnable, linear, group equivariant layer functions between any two tensor power spaces of $\mathbb{R}^n$ for a number of important groups. However, instead of calculating the spanning set by studying set partitions, we obtain it by relating each spanning set element with the isomorphism class of a so-called *bilabelled graph*. In short, a (k, l)–bilabelled graph is a graph that comes with two tuples, one of length k and the other of length l, whose entries are taken from the vertex set of the graph (with repetitions amongst entries allowed). Consequently, by looking at the combinatorics of bilabelled graphs, we can determine the learnable, linear, $\mathrm{Aut}(G)$-equivariant layer functions between any two tensor power spaces of $\mathbb{R}^n$.

We leverage the work of Mančinska & Roberson (2020), who studied the problem of determining under what circumstances two graphs are quantum isomorphic. They built upon the work of Chassaniol (2019), who showed that a vertex-transitive graph has no quantum symmetries. We show how their results and methods can be applied instead for the purpose of learning from data that has an underlying symmetry to the automorphism group of a graph.

There are important consequences for performing such a characterisation. A famous theorem in algebraic graph theory known as Frucht's Theorem (Frucht, 1938) states that

[1]Department of Computing, Imperial College London, United Kingdom. Correspondence to: Edward Pearce–Crump <ep1011@ic.ac.uk>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

every finite group is isomorphic to the automorphism group of a finite undirected graph. As a result, for any finite group of our choosing, if we know the graph (having some n vertices) whose automorphism group is isomorphic to the group in question, then we will be able to characterise all of the learnable, linear, group equivariant layer functions between any two tensor power spaces of $\mathbb{R}^n$ for that group. In particular, we show that we can recover the diagram basis that appears in Godfrey et al. (2023) for the learnable, linear, S_n-equivariant layer functions between tensor power spaces of $\mathbb{R}^n$ in the standard basis of $\mathbb{R}^n$. We also show that we can determine characterisations for other groups, such as for D_4, considered as a subgroup of S_4, which, to the best of our knowledge, have been missing from the literature.

Furthermore, even if, for a given finite undirected graph, we are unable to determine the finite group that is isomorphic to the automorphism group of the graph, we know that we can calculate the learnable, linear, automorphism group equivariant linear layers using this method, which will be sufficient for performing learning in situations where we do not need to know what the actual automorphism group is explicitly.

The main contributions of this paper, which appear in Section 4 onwards, are as follows:

1. We are the first to show how the combinatorics underlying bilabelled graphs provides the theoretical foundation for constructing neural networks that are equivariant to the automorphism group of a graph G having n vertices where the layers are some tensor power of $\mathbb{R}^n$.
2. In particular, we find a spanning set for the learnable, linear, $\mathrm{Aut}(G)$-equivariant layer functions between such tensor power spaces in the standard basis of $\mathbb{R}^n$.
3. We show how our approach can be used to recover the diagram basis that appears in Godfrey et al. (2023) for the learnable, linear, S_n-equivariant layer functions between tensor power spaces of $\mathbb{R}^n$.

Notation: We let $[n]$ represent the set $\{1, \dots, n\}$ throughout this paper.

2. Graph Theory Essentials

We begin by recalling some of the fundamentals of graph theory. For more details, see any standard book on graph theory, such as Bollobas (1998).

Definition 2.1. A **graph** G is a tuple $(V(G), E(G))$ of sets, where $V(G)$ is a set of vertices for G and $E(G)$ is a subset of unordered pairs of elements from $V(G) \times V(G)$ denoting the undirected edges between the vertices of G.

We include the possibility that the graph has loops; however, we only allow at most one loop per vertex.

Definition 2.2. Let G be a graph having n vertices. The **adjacency matrix** of G, denoted by A_G, is the $n \times n$ matrix whose (i, j)–entry is 1 if vertex i is adjacent to vertex j in G, and is 0 otherwise. Note that A_G is a symmetric matrix because G has undirected edges, and the (i, i)-entry is 1 in A_G if and only if G has a loop at vertex i.

Definition 2.3. Let G be a graph. The **complement** of G, denoted by $\overline{G}$, is the graph having the same vertex set as G and the same loops as G, but now distinct vertices are adjacent in $\overline{G}$ if and only if they are not adjacent in G.

Example 2.4. The complete graph on n vertices, K_n, is the loopless graph where every vertex is adjacent to every other vertex.

Example 2.5. The cycle graph on n vertices, C_n, is the loopless graph where every vertex $i \in [n]$ is adjacent to $j = i \pm 1 \mod n$, where $j \in [n]$.

Definition 2.6. Let H and G be graphs. A **graph homomorphism** from H to G is a function $\phi : V(H) \to V(G)$ such that if i is adjacent to j in H, then $\phi(i)$ is adjacent to $\phi(j)$ in G.

Definition 2.7. Let H and G be graphs. A **graph isomorphism** from H to G is a graph homomorphism that is also a bijection.

Consequently, we get that

Definition 2.8. Let G be a graph. An **automorphism** of G is an isomorphism from G to G. The set of all automorphisms of G, written $\mathrm{Aut}(G)$, can be shown to be a group under composition of functions.

Remark 2.9. If G is a graph having n vertices, then $\mathrm{Aut}(G)$ is, in fact, a subgroup of S_n. Specifically, if $\sigma \in S_n$, then it is easy to show, viewing S_n as a subgroup of $GL(n)$, that

$$\sigma \in \mathrm{Aut}(G) \iff \sigma A_G = A_G \sigma \tag{1}$$

It is also clear to see that $\mathrm{Aut}(G) \cong \mathrm{Aut}(\overline{G})$.

Example 2.10. The automorphism group of the complete graph on n vertices, $\mathrm{Aut}(K_n)$, is the symmetric group S_n. Consequently, the automorphism group of the edgeless graph having n vertices, $\mathrm{Aut}(\overline{K_n})$, is also the symmetric group S_n.

Example 2.11. The automorphism group of the cycle graph on n vertices, $\mathrm{Aut}(C_n)$, is isomorphic to the dihedral group D_n of order $2n$.

Example 2.12. The automorphism group of two copies of the complete graph on two vertices, $\mathrm{Aut}(2K_2)$, is isomorphic to the dihedral group D_4 of order 8.

3. Graph Automorphism Group Equivariant Linear Layer Functions

Neural networks that are equivariant to the the automorphism group of a graph G having n vertices, $\mathrm{Aut}(G)$, can

be constructed by alternately composing linear and non-linear equivariant functions between layer spaces that are a tensor power of $\mathbb{R}^n$ (Lim & Nelson, 2022). These layer spaces are representations of $\mathrm{Aut}(G)$ in the following sense.

Recall first that any k-tensor power of $\mathbb{R}^n$, $(\mathbb{R}^n)^{\otimes k}$, is a representation of the symmetric group S_n, since the elements

$$e_I := e_{i_1} \otimes e_{i_2} \otimes \cdots \otimes e_{i_k} \tag{2}$$

for all $I := (i_1, i_2, \ldots, i_k) \in [n]^k$ form a basis of $(\mathbb{R}^n)^{\otimes k}$, and the action of S_n that maps a basis element of $(\mathbb{R}^n)^{\otimes k}$ of the form (2) to

$$e_{\sigma(I)} := e_{\sigma(i_1)} \otimes e_{\sigma(i_2)} \otimes \cdots \otimes e_{\sigma(i_k)} \tag{3}$$

can be extended linearly on the basis.

For any graph G having n vertices, as $\mathrm{Aut}(G)$ is a subgroup of S_n, we have that $(\mathbb{R}^n)^{\otimes k}$ is also a representation of $\mathrm{Aut}(G)$ that is given by the restriction of the representation of S_n to $\mathrm{Aut}(G)$.

We denote the representation of S_n by ρ_k. We will use the same notation for the restriction of this representation to $\mathrm{Aut}(G)$, with the context making clear that it is the restriction of the S_n representation.

Moreover, we have that

Definition 3.1. A map $\phi : (\mathbb{R}^n)^{\otimes k} \to (\mathbb{R}^n)^{\otimes l}$ is said to be **equivariant** to $\mathrm{Aut}(G)$ if, for all $\sigma \in \mathrm{Aut}(G)$ and $v \in (\mathbb{R}^n)^{\otimes k}$,

$$\phi(\rho_k(\sigma)[v]) = \rho_l(\sigma)[\phi(v)] \tag{4}$$

We denote the set of all *linear* $\mathrm{Aut}(G)$-equivariant maps between $(\mathbb{R}^n)^{\otimes k}$ and $(\mathbb{R}^n)^{\otimes l}$ by

$$\mathrm{Hom}_{\mathrm{Aut}(G)}((\mathbb{R}^n)^{\otimes k}, (\mathbb{R}^n)^{\otimes l}) \tag{5}$$

It can be shown that (5) is a vector space over $\mathbb{R}$. See Segal (2014) for more details. Note that (5) is a subspace of $\mathrm{Hom}((\mathbb{R}^n)^{\otimes k}, (\mathbb{R}^n)^{\otimes l})$, the vector space of all linear maps from $(\mathbb{R}^n)^{\otimes k}$ to $(\mathbb{R}^n)^{\otimes l}$.

Our goal is to calculate all of the weight matrices that can appear between any two layers of the graph automorphism group equivariant neural networks in question. It is enough to construct a spanning set of matrices for $\mathrm{Hom}_{\mathrm{Aut}(G)}((\mathbb{R}^n)^{\otimes k}, (\mathbb{R}^n)^{\otimes l})$, by viewing it as a subspace of $\mathrm{Hom}((\mathbb{R}^n)^{\otimes k}, (\mathbb{R}^n)^{\otimes l})$ and choosing the standard basis of $\mathbb{R}^n$, since any weight matrix will be a weighted linear combination of these spanning set matrices.

4. Characterisation Result using Bilabelled Graphs

In this section, we construct a spanning set of matrices for $\mathrm{Hom}_{\mathrm{Aut}(G)}((\mathbb{R}^n)^{\otimes k}, (\mathbb{R}^n)^{\otimes l})$ by relating each spanning set matrix with the isomorphism class of a so-called (k, l)–bilabelled graph. We leverage the work of Mančinska & Roberson (2020) throughout, but begin by stating a result found by Chassaniol (2019) which describes, in terms of a generating set of matrices, the category whose morphisms are the linear layer functions that we wish to characterise.

4.1. Chassaniol's Result

For each group G that is a subgroup of S_n, we can define the following category.

Definition 4.1. The category $\mathcal{C}(G)$ consists of objects that are the k-order tensor power spaces of $\mathbb{R}^n$, as representations of G, and morphism spaces between any two objects that are the vector spaces $\mathrm{Hom}_G((\mathbb{R}^n)^{\otimes k}, (\mathbb{R}^n)^{\otimes l})$.

The vertical composition of morphisms is given by the usual composition of linear maps, the tensor product is given by the usual tensor product of linear maps, and the unit object is the one-dimensional trivial representation of G.

Remark 4.2. We will sometimes write $\mathcal{C}(G)(k, l)$ for $\mathrm{Hom}_G((\mathbb{R}^n)^{\otimes k}, (\mathbb{R}^n)^{\otimes l})$, and reuse the notation $\mathcal{C}(G)$ for the set $\cup_{k,l=0}^{\infty} \mathcal{C}(G)(k, l)$.

Proposition 4.3. *The category $\mathcal{C}(G)$ is a strict $\mathbb{R}$-linear monoidal category.*

Proof. See the Technical Appendix. □

It will be useful to define a number of spider maps $M^{k,l}$ together with the swap map S.

Definition 4.4. For all non-negative integers k, l, the **spider map** $M^{k,l} \in \mathrm{Hom}((\mathbb{R}^n)^{\otimes k}, (\mathbb{R}^n)^{\otimes l})$ is defined as follows:

- if $k, l > 0$, then $M^{k,l}$ maps $e_i^{\otimes k}$ to $e_i^{\otimes l}$ for all $i \in [n]$ and maps all other vectors to the zero vector.
- if $k = 0$ and $l > 0$, then $M^{0,l}$ maps 1 to $\sum_i e_i^{\otimes l}$.
- if $k > 0$ and $l = 0$, then $M^{k,0}$ maps $e_i^{\otimes k}$ to 1 for all $i \in [n]$ and maps all other vectors to 0.
- if $k = 0$ and $l = 0$, then $M^{0,0} := (n)$.

The **swap map** $S \in \mathrm{Hom}((\mathbb{R}^n)^{\otimes 2}, (\mathbb{R}^n)^{\otimes 2})$ maps $e_i \otimes e_j$ to $e_j \otimes e_i$ for all $i, j \in [n]$.

For any graph G having n vertices, Chassaniol (2019) found the following generating set for the category $\mathcal{C}(\mathrm{Aut}(G))$:

Theorem 4.5 (Chassaniol (2019), Proposition 3.5)**.** *Let A_G denote the adjacency matrix of the graph G. Then*

$$\mathcal{C}(\mathrm{Aut}(G)) = \langle M^{0,1}, M^{2,1}, A_G, S \rangle_{+,\circ,\otimes,*} \tag{6}$$

where the right hand side denotes all matrices that can be generated from the four matrices using the operations of $\mathbb{R}$-linear combinations, matrix product, Kronecker product, and transposition.

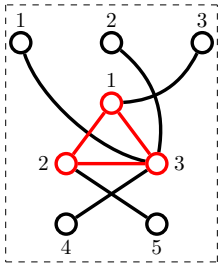


Figure 1. The $(2,3)$–bilabelled graph diagram that is associated with the isomorphism class of $\boldsymbol{K} := (K_3, (3,2), (3,3,1))$.

4.2. The Category of All Bilabelled Graphs

Mančinska & Roberson (2020) showed how Chassaniol's generating set for $\mathcal{C}(\mathrm{Aut}(G))$ can be improved by relating it to the combinatorics of bilabelled graphs.

Definition 4.6 (Bilabelled Graph)**.** A **$(\boldsymbol{k},\boldsymbol{l})$–bilabelled graph $\boldsymbol{H}$** is a triple $(H, \boldsymbol{k}, \boldsymbol{l})$, where H is a graph having some m vertices with labels in $[m]$, $\boldsymbol{k} := (k_1, \dots, k_k)$ is a tuple in $[m]^k$, and $\boldsymbol{l} := (l_1, \dots, l_l)$ is a tuple in $[m]^l$.

We call $\boldsymbol{k}$ and $\boldsymbol{l}$ the **input** and **output tuples** to $\boldsymbol{H}$ respectively, and we call H the **underlying graph** of $\boldsymbol{H}$. A vertex in the underlying graph H of $\boldsymbol{H}$ is said to be **free** if it does not appear in either of the input or output tuples.

In order to provide a diagrammatic representation of bilabelled graphs, we need the following definition.

Definition 4.7. Let $\boldsymbol{H_1} = (H_1, \boldsymbol{k}, \boldsymbol{l})$ be a (k,l)–bilabelled graph and let $\boldsymbol{H_2} = (H_2, \boldsymbol{k'}, \boldsymbol{l'})$ be another (k,l)–bilabelled graph. Then $\boldsymbol{H_1}$ and $\boldsymbol{H_2}$ are **isomorphic** as (k,l)–bilabelled graphs, written $\boldsymbol{H_1} \cong \boldsymbol{H_2}$, if and only if there is a graph isomorphism from H_1 to H_2 such that $k_i \mapsto k'_i$ for all $i \in [k]$ and $l_j \mapsto l'_j$ for all $j \in [l]$. We denote $\boldsymbol{H_1}$'s **isomorphism class** by $[\boldsymbol{H_1}]$.

Remark 4.8. An isomorphism between two (k,l)–bilabelled graphs can effectively be thought of as a relabelling of the vertices of the *same* underlying graph, resulting in an appropriate relabelling of the elements of the tuples themselves. Consequently, the isomorphism can be thought of as a relabelling of the same bilabelled graph.

With this in mind, we can represent the isomorphism class $[\boldsymbol{H}]$ for a (k,l)–bilabelled graph $\boldsymbol{H} = (H, \boldsymbol{k}, \boldsymbol{l})$ in diagrammatic form. We choose $\boldsymbol{H}$ as a class representative of $[\boldsymbol{H}]$, and proceed as follows: we draw

- l black vertices on the top row, labelled left to right by $1, \dots, l$,
- k black vertices on the bottom row, labelled left to right by $l+1, \dots, l+k$,
- the underlying graph H in red, with labelled red vertices and red edges, in between the two rows of black vertices, and
- a black line connecting vertex i in the top row to l_i, and a black line connecting vertex $l+j$ in the bottom row to k_j.

Note that if we drew a diagram for another representative of the same isomorphism class $[\boldsymbol{H}]$, by the definition of an isomorphism for (k,l)–bilabelled graphs, we would obtain exactly the same diagram as for the first class representative, except the labels of the underlying graph's vertices (and consequently the elements of the tuples) would be different. As a result, the diagram for the isomorphism class $[\boldsymbol{H}]$ is independent of the choice of class representative, and so we choose the class' label, $\boldsymbol{H}$, to draw the diagram throughout, unless otherwise stated. With the technicalities being understood, we often refer to the construction defined above as a (k,l)–bilabelled graph diagram *for $\boldsymbol{H}$ itself*, and use the same notation $\boldsymbol{H}$. We give an example of a $(2,3)$–bilabelled graph diagram in Figure 1.

We can define a number of **operations** on isomorphism classes of bilabelled graphs, namely, composition, tensor product, and involution, as follows.

Definition 4.9 (Composition of Bilabelled Graphs)**.** Let $[\boldsymbol{H_1}]$ be the isomorphism class of the (k,l)–bilabelled graph $\boldsymbol{H_1} = (H_1, \boldsymbol{k}, \boldsymbol{l})$, and let $[\boldsymbol{H_2}]$ be the isomorphism class of the (l,m)–bilabelled graph $\boldsymbol{H_2} = (H_2, \boldsymbol{l'}, \boldsymbol{m})$.

Then we define the **composition** $[\boldsymbol{H_2}] \circ [\boldsymbol{H_1}]$ to be the isomorphism class $[\boldsymbol{H}]$ of the (k,m)–bilabelled graph $\boldsymbol{H} = (H, \boldsymbol{k}, \boldsymbol{m})$ that is obtained as follows: drawing each isomorphism class as a diagram, we first relabel the red vertices in H_2 (and consequently the elements of $\boldsymbol{l'}, \boldsymbol{m}$) under the map $i \mapsto i'$, for all $i \in [V(H_2)]$. Then, we connect the top row of black vertices of $\boldsymbol{H_1}$ with the bottom row of black vertices of $\boldsymbol{H_2}$, and delete the vertices themselves. We are now left with l black lines that are edges between red vertices of the underlying graphs H_1 and H_2. Next, we contract these, forming set unions of the vertex labels where appropriate, and remove any red multiedges between red vertices that appear in the contraction to obtain the new underlying graph H. Note that we keep any loops that appear in the new underlying graph H. Finally, we relabel the vertex set of the new underlying graph H so that each vertex is labelled by an integer only, and consequently relabel the entries of the tuples $\boldsymbol{k}$ and $\boldsymbol{m}$ accordingly.

Since this operation has been defined on diagrams, it means that the operation is well defined on the isomorphism classes themselves. We give an example of this composition in Figure 2.

Remark 4.10. In order to compose two isomorphism classes of bilabelled graphs, note that only the number of bottom

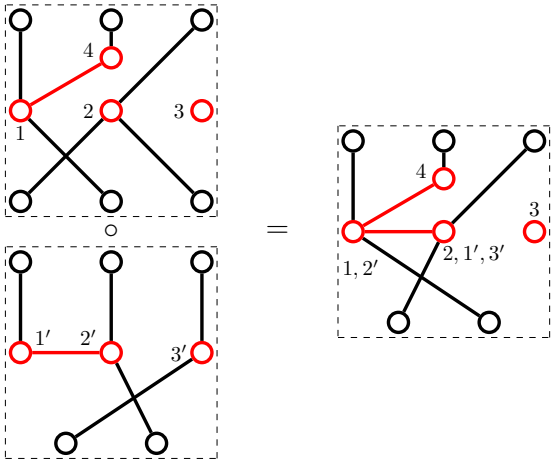


Figure 2. The composition $[\boldsymbol{H_2}] \circ [\boldsymbol{H_1}]$ of the $(2,3)$–bilabelled graph diagram for $\boldsymbol{H_1} = (H_1, (3', 2'), (1', 2', 3'))$ with the $(3,3)$–bilabelled graph diagram for $\boldsymbol{H_2} = (H_2, (2,1,2), (1,4,2))$, where H_1 is the graph having (relabelled) vertex set $[3']$ and edge set $\{(1', 2')\}$ and H_2 is the graph having vertex set $[4]$ and edge set $\{(1,4)\}$. The vertices of the resulting bilabelled graph diagram on the RHS would be relabelled. For example, $\{1, 2'\}$ could be relabelled as 1 and $\{2, 1', 3'\}$ could be relabelled as 2 to give the $(2,3)$–bilabelled graph diagram for $\boldsymbol{H} = (H, (2,1), (1,4,2))$, where H is the graph having vertex set $[4]$ and edge set $\{(1,4), (1,2)\}$.

row vertices in the diagram for $\boldsymbol{H_2}$ needs to be equal to the number of top row vertices in the diagram for $\boldsymbol{H_1}$. In particular, the number of vertices in the underlying graphs of $\boldsymbol{H_1}$ and $\boldsymbol{H_2}$ do *not* need to be the same.

Definition 4.11 (Tensor Product of Bilabelled Graphs)**.** Let $[\boldsymbol{H_1}]$ be the isomorphism class of the (k,l)–bilabelled graph $\boldsymbol{H_1} = (H_1, \boldsymbol{k}, \boldsymbol{l})$ and let $[\boldsymbol{H_2}]$ be the isomorphism class of the (q,m)–bilabelled graph $\boldsymbol{H_2} = (H_2, \boldsymbol{q}, \boldsymbol{m})$.

Then we define the **tensor product** $[\boldsymbol{H_1}] \otimes [\boldsymbol{H_2}]$ to be the isomorphism class of the $(k+q, l+m)$–bilabelled graph $(H_1 \cup H_2, \boldsymbol{kq}, \boldsymbol{lm})$ where $\boldsymbol{kq}$ is the $(k+q)$–length tuple obtained by concatenating $\boldsymbol{k}$ and $\boldsymbol{q}$, and likewise for $\boldsymbol{lm}$.

Definition 4.12 (Involution of Bilabelled Graphs)**.** Let $[\boldsymbol{H}]$ be the isomorphism class of the (k,l)–bilabelled graph $\boldsymbol{H} = (H, \boldsymbol{k}, \boldsymbol{l})$. Then we define the **involution** $[\boldsymbol{H}^*]$ to be the isomorphism class of the (l,k)–bilabelled graph $(H, \boldsymbol{l}, \boldsymbol{k})$.

We can form a category for the bilabelled graphs, as follows:

Definition 4.13 (Category of Bilabelled Graphs)**.** The **category of all bilabelled graphs** $\mathcal{G}$ is the category whose objects are the non-negative integers, and, for any pair of objects k and l, the morphism space $\mathcal{G}(k,l) := \mathrm{Hom}_{\mathcal{G}}(k,l)$ is defined to be the $\mathbb{R}$-linear span of the set of all isomorphism classes of (k,l)–bilabelled graphs.

The vertical composition of morphisms is the composition of isomorphism classes of bilabelled graphs given in Definition 4.9 that is extended to be $\mathbb{R}$-bilinear, the tensor product of morphisms is the tensor product of isomorphism classes of bilabelled graphs given in Definition 4.11 that is also extended to be $\mathbb{R}$-bilinear, and the unit object is 0.

Proposition 4.14. *The category of all bilabelled graphs, $\mathcal{G}$, is a strict $\mathbb{R}$–linear monoidal category.*

Proof. See the Technical Appendix. □

4.3. G-Homomorphism Matrices

Mančinska & Roberson (2020) established a relationship between the abstract and the concrete: namely, between isomorphism classes of (k,l)–bilabelled graphs and, for a fixed graph G having n vertices, matrices that are linear maps $(\mathbb{R}^n)^{\otimes k} \to (\mathbb{R}^n)^{\otimes l}$, which they termed G-homomorphism matrices. We express this relationship more formally in terms of functors and categories at the end of this section.

Definition 4.15 (G-Homomorphism Matrix)**.** Suppose that G is a graph having n vertices, and let $[\boldsymbol{H}]$ be the isomorphism class of the (k,l)–bilabelled graph $\boldsymbol{H} := (H, \boldsymbol{k}, \boldsymbol{l})$.

The **G-homomorphism matrix** of $[\boldsymbol{H}]$ is the $n^l \times n^k$ matrix where each (I,J)-entry is given by the number of graph homomorphisms from H to G such that $\boldsymbol{l}$ is mapped to I and $\boldsymbol{k}$ is mapped to J. We denote this matrix by $X_{\boldsymbol{H}}^G$.

Remark 4.16. Note that a G-homomorphism matrix $X_{\boldsymbol{H}}^G$ is independent of the choice of class representative for $[\boldsymbol{H}]$, and so we can refer to a G-homomorphism matrix *of* $\boldsymbol{H}$ *itself*, with the technicalities being understood. Also, any such matrix must have only real entries, by definition.

In Figure 3, we present an example that shows how to calculate the G-homomorphism matrix of $[\boldsymbol{H}]$ for the graph G having vertex set $V(G) = \{1,2,3\}$ and edge set $E(G) = \{(1,2)\}$ and for the $(1,1)$–bilabelled graph $\boldsymbol{H} = (H, (3), (1))$, where H is the graph having vertex set $V(H) = \{1,2,3,4\}$ and edge set $E(H) = \{(1,2), (3,4)\}$. To determine the (I,J)-entry of $X_{\boldsymbol{H}}^G$, we superimpose the values of I onto the top row of black vertices in $\boldsymbol{H}$, and the values of J onto the bottom row of black vertices in $\boldsymbol{H}$. For each red vertex that is connected with a set of black vertices, we look to update its label. If all of the black vertices in the set have the same label, then we update the red vertex with that label, otherwise we stop and immediately determine that the (I,J)-entry of $X_{\boldsymbol{H}}^G$ is 0. Assuming that these red vertices can and have been updated, we determine the number of possible mappings to G for the red vertices that have not been updated in $\boldsymbol{H}$ such that the overall mapping is a graph homomorphism. The total number of possibilities is the (I,J)-entry of $X_{\boldsymbol{H}}^G$.

We now describe some important examples of G-homomorphism matrices where G is a graph having n vertices throughout.

Example 4.17. If $\boldsymbol{A}$ is the $(1,1)$–bilabelled graph $(K_2, (1), (2))$, where K_2 is the complete graph on two vertices, then $X_{\boldsymbol{A}}^G$ is the adjacency matrix A_G of G.

Example 4.18. If $\boldsymbol{M}^{k,l}$ is the (k,l)–bilabelled graph $(K_1, \boldsymbol{k} = (1, \dots, 1), \boldsymbol{l} = (1, \dots, 1))$, where K_1 is the complete graph on one vertex, then $X_{\boldsymbol{M}^{k,l}}^G$ is the spider matrix $M^{k,l}$ that is given in Definition 4.4.

Example 4.19. If $\boldsymbol{S}$ is the $(2,2)$–bilabelled graph $(\overline{K_2}, (2,1), (1,2))$, where $\overline{K_2}$ is the edgeless graph on two vertices, then $X_{\boldsymbol{S}}^G$ is the swap map S that is also given in Definition 4.4.

For a fixed graph G having n vertices, we can form a category for the G-homomorphism matrices, as follows:

Definition 4.20 (Category of G-Homomorphism Matrices)**.** For a given graph G having n vertices, **the category of all G-homomorphism matrices**, $\mathcal{C}^G$, is the category whose objects are the vector spaces $(\mathbb{R}^n)^{\otimes k}$, and, for any pair of objects $(\mathbb{R}^n)^{\otimes k}$ and $(\mathbb{R}^n)^{\otimes l}$, the morphism space $\mathrm{Hom}_{\mathcal{C}^G}((\mathbb{R}^n)^{\otimes k}, (\mathbb{R}^n)^{\otimes l})$ is defined to be the $\mathbb{R}$-linear span of the set of all G-homomorphism matrices obtained from all isomorphism classes of (k,l)–bilabelled graphs.

The vertical composition of morphisms is given by the usual multiplication of matrices, the tensor product of morphisms is given by the usual Kronecker product of matrices, and the unit object is $\mathbb{R}$.

Proposition 4.21. *The category of all G-homomorphism matrices for a given graph G having n vertices, $\mathcal{C}^G$, is a strict $\mathbb{R}$–linear monoidal category.*

Proof. See the Technical Appendix. □

Mančinska & Roberson (2020) showed that the operations given on isomorphism classes of bilabelled graphs $\boldsymbol{H}$ correspond bijectively with the matrix operations on G-homomorphism matrices $X_{\boldsymbol{H}}^G$. This is stated more formally in the following three lemmas:

Lemma 4.22 (Mančinska & Roberson (2020), Lemma 3.21)**.** *Suppose that G is a graph having n vertices. Let $[\boldsymbol{H_1}]$ be the isomorphism class of the (k,l)–bilabelled graph $\boldsymbol{H_1} = (H_1, \boldsymbol{k}, \boldsymbol{l})$, and let $[\boldsymbol{H_2}]$ be the isomorphism class of the (l,m)–bilabelled graph $\boldsymbol{H_2} = (H_2, \boldsymbol{l'}, \boldsymbol{m})$. Then*

$$X_{\boldsymbol{H_2}}^G X_{\boldsymbol{H_1}}^G = X_{\boldsymbol{H_2} \circ \boldsymbol{H_1}}^G \tag{7}$$

Lemma 4.23 (Mančinska & Roberson (2020), Lemma 3.23)**.** *Suppose that G is a graph having n vertices. Let $[\boldsymbol{H_1}]$ be the isomorphism class of the (k,l)–bilabelled graph $\boldsymbol{H_1} = (H_1, \boldsymbol{k}, \boldsymbol{l})$, and let $[\boldsymbol{H_2}]$ be the isomorphism class of the (q,m)–bilabelled graph $\boldsymbol{H_2} = (H_2, \boldsymbol{q}, \boldsymbol{m})$. Then*

$$X_{\boldsymbol{H_1}}^G \otimes X_{\boldsymbol{H_2}}^G = X_{\boldsymbol{H_1} \otimes \boldsymbol{H_2}}^G \tag{8}$$

Lemma 4.24 (Mančinska & Roberson (2020), Lemma 3.24)**.** *Suppose that G is a graph having n vertices, and let $[\boldsymbol{H}]$ be the isomorphism class of the (k,l)–bilabelled graph $\boldsymbol{H} = (H, \boldsymbol{k}, \boldsymbol{l})$. Then*

$$(X_{\boldsymbol{H}}^G)^* = X_{\boldsymbol{H}^*}^G \tag{9}$$

where $(X_{\boldsymbol{H}}^G)^*$ is the transpose of the matrix $X_{\boldsymbol{H}}^G$.

Consequently, we obtain the following theorem, expressed in terms of functors and categories.

Theorem 4.25. *Suppose that G is a graph having n vertices. Then there exists a full, strict $\mathbb{R}$–linear monoidal functor*

$$\mathcal{F}^G : \mathcal{G} \to \mathcal{C}^G \tag{10}$$

that is defined on the objects of $\mathcal{G}$ by $\mathcal{F}^G(k) := (\mathbb{R}^n)^{\otimes k}$ and, for any objects k, l of $\mathcal{G}$, the map

$$\mathrm{Hom}_{\mathcal{G}}(k,l) \to \mathrm{Hom}_{\mathcal{C}^G}(\mathcal{F}^G(k), \mathcal{F}^G(l)) \tag{11}$$

is given by

$$[\boldsymbol{H}] \mapsto X_{\boldsymbol{H}}^G \tag{12}$$

for all isomorphism classes of (k,l)–bilabelled graphs.

Proof. See the Technical Appendix. □

4.4. A Spanning Set of Matrices for the Learnable, Linear, $\mathrm{Aut}(G)$-Equivariant Layer Functions

Compare the following proposition with Chassaniol's result, given in Theorem 4.5.

Proposition 4.26 (Mančinska & Roberson (2020), Theorem 8.4)**.** *We have that*

$$\mathcal{G} = \langle [\boldsymbol{M}^{0,1}], [\boldsymbol{M}^{2,1}], [\boldsymbol{A}], [\boldsymbol{S}] \rangle_{\circ, \otimes, *} \tag{13}$$

where $\boldsymbol{M}^{0,1}, \boldsymbol{M}^{2,1}, \boldsymbol{A}, \boldsymbol{S}$ are the bilabelled graphs defined in Examples 4.17, 4.18, and 4.19, and the operations $\circ, \otimes, *$ on bilabelled graphs are those that are given in Definitions 4.9, 4.11 and 4.12, respectively.

We have come to the main results of this paper, namely the following theorem and its corollary.

Theorem 4.27. *Suppose that G is a graph having n vertices. The vector space of all $\mathrm{Aut}(G)$-equivariant, linear maps between tensor power spaces of $\mathbb{R}^n$, $\mathrm{Hom}_{\mathrm{Aut}(G)}((\mathbb{R}^n)^{\otimes k}, (\mathbb{R}^n)^{\otimes l})$, when the standard basis of $\mathbb{R}^n$ is chosen, is spanned by all of the G-homomorphism matrices $X_{\boldsymbol{H}}^G$ that are obtained from all of the isomorphism classes of (k,l)–bilabelled graphs.*

Proof. The proof of this theorem is inspired by Mančinska & Roberson (2020), Theorem 8.5.

We know, by Chassaniol, that

$$\mathcal{C}(\mathrm{Aut}(G)) = \langle M^{0,1}, M^{2,1}, A_G, S \rangle_{+, \circ, \otimes, *} \tag{14}$$

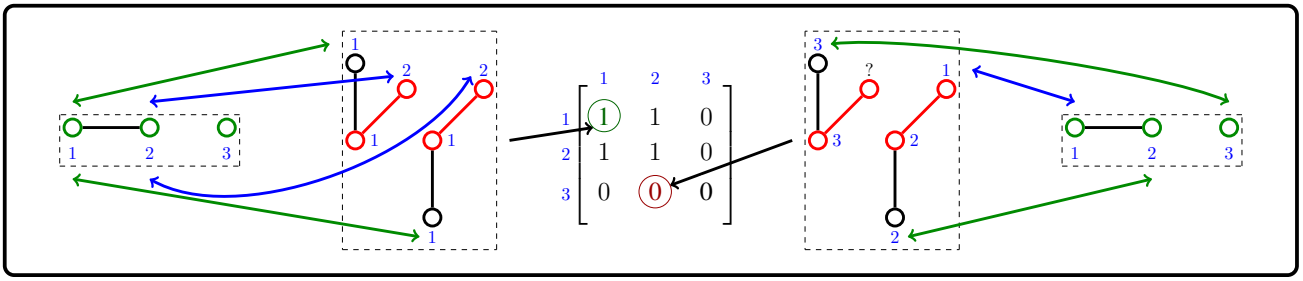


Figure 3. For the graph G having vertex set $V(G) = \{1, 2, 3\}$ and edge set $E(G) = \{(1, 2)\}$, we show how to calculate the (I, J)-entries of the G-homomorphism matrix $X_{\boldsymbol{H}}^G$ corresponding to the isomorphism class $[\boldsymbol{H}]$ of the $(1, 1)$–bilabelled graph $\boldsymbol{H} = (H, (3), (1))$ where H is the graph having vertex set $V(H) = \{1, 2, 3, 4\}$ and edge set $E(H) = \{(1, 2), (3, 4)\}$. On the left hand side, we calculate the $(1, 1)$-entry of $X_{\boldsymbol{H}}^G$. Since both of the black labelled vertices are being mapped to vertex 1 in G, we can superimpose 1 onto these vertices, and hence also onto the red labelled vertices that are connected with them, to determine where the other red vertices can be mapped to under a graph homomorphism. In this case, the only possible vertex in G that the other red vertices can be mapped to is 2. Hence there is only one possible graph homomorphism from H to G such that $1 \mapsto 1$ and $3 \mapsto 1$, and so the $(1, 1)$-entry of $X_{\boldsymbol{H}}^G$ is 1. On the right hand side, we calculate the $(3, 2)$-entry of $X_{\boldsymbol{H}}^G$. We follow the same approach by superimposing 3 in G onto the black vertex labelled 1 in $\boldsymbol{H}$, and 2 in G onto the black vertex labelled 3 in $\boldsymbol{H}$. We relabel the red vertices that the black vertices are connected with and see where the other red vertices can be mapped to under a graph homomorphism. Whilst one of these red vertices can only be mapped to 1 in G, the other red vertex cannot be mapped to any vertex in G, since the vertex 3 in G is not connected with any other vertex in G. Hence the number of graph homomorphisms from H to G such that $1 \mapsto 3$ and $3 \mapsto 2$ is 0, and so the $(3, 2)$-entry of $X_{\boldsymbol{H}}^G$ is 0.

By Examples 4.17, 4.18, and 4.19, we get that $\mathcal{C}(\mathrm{Aut}(G))$ is equal to

$$\{X_{\boldsymbol{H}}^G \mid [\boldsymbol{H}] \in \langle [\boldsymbol{M}^{0,1}], [\boldsymbol{M}^{2,1}], [\boldsymbol{A}], [\boldsymbol{S}] \rangle_{+,\circ,\otimes,*}\} \quad (15)$$

and so, by Proposition 4.26, we have that

$$\mathcal{C}(\mathrm{Aut}(G)) = \mathbb{R}\text{–span}\{X_{\boldsymbol{H}}^G \mid [\boldsymbol{H}] \in \mathcal{G}\} \quad (16)$$

Consequently, for any k, l, we get that

$$\mathcal{C}(\mathrm{Aut}(G))(k, l) = \mathbb{R}\text{–span}\{X_{\boldsymbol{H}}^G \mid [\boldsymbol{H}] \in \mathcal{G}(k, l)\} \quad (17)$$

As the LHS of (17) is equivalent notation for $\mathrm{Hom}_{\mathrm{Aut}(G)}((\mathbb{R}^n)^{\otimes k}, (\mathbb{R}^n)^{\otimes l})$, we obtain our result. □

Corollary 4.28. *For all non-negative integers l and k, if G is a graph having n vertices, then the weight matrix W that appears in an $\mathrm{Aut}(G)$-equivariant linear layer function from $(\mathbb{R}^n)^{\otimes k}$ to $(\mathbb{R}^n)^{\otimes l}$ must be of the form*

$$W = \sum_{[\boldsymbol{H}] \in \mathcal{G}(k,l)} \lambda_{\boldsymbol{H}} X_{\boldsymbol{H}}^G \quad (18)$$

for weights $\lambda_{\boldsymbol{H}} \in \mathbb{R}$.

Remark 4.29. In particular, Theorem 4.27 shows that $\mathcal{C}^G$ is isomorphic to $\mathcal{C}(\mathrm{Aut}(G))$ as categories, and that the objects of the category $\mathcal{C}^G$ come from representations of $\mathrm{Aut}(G)$.

Theorem 4.27 is especially powerful when it is combined with Frucht's Theorem (Frucht, 1938). Frucht's Theorem states that every finite group is isomorphic to the automorphism group of a finite undirected graph. Hence, for any finite group, by determining a graph (having n vertices) whose automorphism group is isomorphic to the group in question, we can determine the weight matrix that appears between any two layers that are some tensor power of $\mathbb{R}^n$ in a neural network that is equivariant to the group.

Remark 4.30. It is very important to note the following. If H is a group that is isomorphic to the automorphism group of a graph G having n vertices, then the spanning set that we obtain for $\mathrm{Hom}_H((\mathbb{R}^n)^{\otimes k}, (\mathbb{R}^n)^{\otimes l})$ using Theorem 4.27 depends on how the automorphism group is embedded in the symmetric group S_n, itself thought of as a subgroup of matrices in $GL(n)$ having chosen the standard basis of $\mathbb{R}^n$. The spanning set is determined not only by how the vertices of the underlying graph G are labelled (up to all automorphisms), but also by what the edges are in G.

For example, in the Technical Appendix, we show not only that D_4 has three embeddings in S_4, but also that each embedding can be obtained separately from two graphs that are the complement of each other, where both graphs have the same labelling of the vertices. All six instances (an embedding of D_4 coming from a graph with a certain labelling of its vertices) give rise to different, albeit related, spanning sets (in fact, bases) of $\mathrm{Hom}_{D_4}(\mathbb{R}^4, \mathbb{R}^4)$. Hence, we obtain a basis for each specific embedding and labelled graph that is chosen.

We have seen that we can find a spanning set for $\mathrm{Hom}_{\mathrm{Aut}(G)}((\mathbb{R}^n)^{\otimes k}, (\mathbb{R}^n)^{\otimes l})$ by considering all isomorphism classes of (k, l)–bilabelled graphs. However, with the following result, we can reduce the number of elements that appear in the spanning set by reducing the number of isomorphism classes that we need to consider.

Proposition 4.31. *Let $\boldsymbol{H}$ be a (k,l)–bilabelled graph whose underlying graph H contains a subset of free vertices that, whilst they may have edges amongst themselves, are entirely disconnected from any vertices that are in a connected component containing a non-free vertex.*

Then $X_{\boldsymbol{H}}^G$ is a scalar multiple of $X_{\boldsymbol{H'}}^G$, where $\boldsymbol{H'}$ is the (k,l)–bilabelled graph that is obtained from $\boldsymbol{H}$ by removing the subset.

Proof. See the Technical Appendix. □

We also have the following result that is useful for generating the isomorphism classes of (k,l)–bilabelled graphs.

Proposition 4.32 (Frobenius Duality)**.** *Suppose that we use Theorem 4.27 to obtain a spanning set for $\mathrm{Hom}_{\mathrm{Aut}(G)}((\mathbb{R}^n)^{\otimes k},(\mathbb{R}^n)^{\otimes l})$. Then we can immediately obtain a spanning set for $\mathrm{Hom}_{\mathrm{Aut}(G)}((\mathbb{R}^n)^{\otimes q},(\mathbb{R}^n)^{\otimes m})$, for any $q,m \geq 0$ such that $q+m=k+l$.*

Proof. See the Technical Appendix. □

Remark 4.33. In the Technical Appendix, we show that we can recover the diagram basis for $\mathrm{Hom}_{S_n}((\mathbb{R}^n)^{\otimes k},(\mathbb{R}^n)^{\otimes l})$ by looking at the homomorphism matrices for the complement of the complete graph on n vertices, $\overline{K_n}$.

This implies that, for a graph G having n vertices, the spanning set of $\mathrm{Hom}_{\mathrm{Aut}(G)}((\mathbb{R}^n)^{\otimes k},(\mathbb{R}^n)^{\otimes l})$ automatically includes the image, under the functor $\mathcal{F}^G$, of all set partitions of $[l+k]$ expressed as (k,l)–bilabelled graph diagrams, since $\mathrm{Aut}(G)$ is a subgroup of S_n.

We use all of the results given above to obtain a procedure for constructing the weight matrix for an $\mathrm{Aut}(G)$-equivariant linear layer function from $(\mathbb{R}^n)^{\otimes k}$ to $(\mathbb{R}^n)^{\otimes l}$ from isomorphism classes of (k,l)–bilabelled graphs. In the procedure, we create all $(q,0)$–bilabelled graph diagrams that are appropriate for the graph G, where $q=k+l$, and then use Frobenius duality and the functor $\mathcal{F}^G$ to obtain a spanning set of matrices for $\mathrm{Hom}_{\mathrm{Aut}(G)}((\mathbb{R}^n)^{\otimes k},(\mathbb{R}^n)^{\otimes l})$. To capture all of the $(q,0)$–bilabelled graph diagrams that are appropriate for G, we first need to calculate the length of the longest path m between any two (not necessarily distinct) vertices in G where each edge in G is traversed at most once. This is the longest path that needs to be mapped onto by a graph homomorphism from the underlying graph of a $(q,0)$–bilabelled graph diagram to G. We use m to generate the all of the $(q,0)$–bilabelled graph diagrams as follows. We focus on the **non-free red vertices** since they play a key role in how the entries of a spanning set matrix $X_{\boldsymbol{H}}^G$ are determined. We use the following ideas to generate all of the $(q,0)$–bilabelled graph diagrams from the set partitions of $[l+k]$ expressed as $(q,0)$–bilabelled graph diagrams.

- Between any pair of non-free red vertices, we must consider all paths of lengths 0 to $2m$ between them consisting solely of free red vertices, since such a path of length $2m$ is a double cover of the longest path in G.
- Starting with a single non-free red vertex that is not connected with any other red vertex, we must consider all paths of lengths 0 to m consisting solely of free red vertices from this non-free red vertex, since such a path of length m is a single cover of the longest path in G.

We also need to consider the impact of loops on $(q,0)$–bilabelled graph diagrams if G has loops (see Step 5 of the procedure). These points show that the set of $(q,0)$–bilabelled graph diagrams that we need to consider depends on the graph G.

Remark 4.34. In the Technical Appendix, we have provided a number of examples for how to calculate a spanning set of $\mathrm{Hom}_{\mathrm{Aut}(G)}((\mathbb{R}^n)^{\otimes k},(\mathbb{R}^n)^{\otimes l})$ using the procedure, for different graphs G and for low order tensor powers of $\mathbb{R}^n$.

5. Limitations and Feasibility

Given the current limitations of hardware, we recognize that there will be difficulties when implementing the neural networks that are discussed in this paper. Considerable engineering efforts will be needed to achieve the required scale due to the non-trivial task of storing high-order tensors and the weight matrices in memory. Kondor et al. (2018) demonstrated this by developing custom CUDA kernels to implement their tensor product-based neural networks. However, we anticipate that as computing power continues to improve, higher-order group equivariant neural networks will become more prominent in practical applications.

6. Conclusion

We are the first to show how the combinatorics underlying bilabelled graphs provides the theoretical background for constructing neural networks that are equivariant to the automorphism group of a graph having n vertices where the layers are some tensor power of $\mathbb{R}^n$. We found the form of the learnable, linear, $\mathrm{Aut}(G)$-equivariant layer functions between such tensor power spaces in the standard basis of $\mathbb{R}^n$ by finding a spanning set for the $\mathrm{Hom}_{\mathrm{Aut}(G)}$–spaces in which these layer functions live. However, given that the number of isomorphism classes of (k,l)–bilabelled graphs increases exponentially, both as the number of vertices in the graph G increases and as k and l increase, resulting in the number of spanning set elements increasing exponentially, it would be useful to find further ways of reducing the number of isomorphism classes of (k,l)–bilabelled graphs that we need to consider. We leave this to future work.

Procedure: Weight Matrix for an $\mathrm{Aut}(G)$-Equivariant Linear Layer Function from $(\mathbb{R}^n)^{\otimes k}$ to $(\mathbb{R}^n)^{\otimes l}$.

Calculate all $(q, 0)$–bilabelled graph diagrams that are appropriate for G where $q = l + k$, as follows. Let m be the maximum length of a path between any two vertices in G such that each edge in G is traversed at most once. Then

1. Calculate all set partitions of $\{1, \dots, q = l + k\}$, and express them as $(q, 0)$–bilabelled graph diagrams. For uniformity, we choose to order the blocks in a set partition by their size, in descending order. Each block B_i of size b_i corresponds to a spider diagram having a single red vertex and b_i black vertices connected to it with black edges. The vertices in the spider diagram for B_i are labelled with the elements of B_i in ascending order, from left to right. If $m = 0$, go to Step 6. Otherwise:

2. From all of the diagrams found in Step 1, create all possible $(q, 0)$–bilabelled graph diagrams that have only internal red edges between red vertices, as follows. For each diagram $\boldsymbol{H}$ found in Step 1:
   - Let c be the number of red vertices in $\boldsymbol{H}$. Let $e := \frac{c(c-1)}{2}$: this is the number of edges in the complete graph K_c having c vertices. Create all possible e length strings having values in $0 \to 2m$. Each position in the string represents a different pair of distinct red vertices in $\boldsymbol{H}$. Remove the all 0 string.
   - For each string, create a new $(q, 0)$–bilabelled graph diagram as follows: for each position in the string, if t is its value, then, assuming that the position represents the pair of red vertices v and w, insert t new edges between v and w into $\boldsymbol{H}$, adding in $t - 1$ new red vertices to make these t new edges possible.

3. Also, from all of the diagrams found in Step 1, create all possible $(q, 0)$–bilabelled graph diagrams that have only external red edges, as follows. For each diagram $\boldsymbol{H}$ found in Step 1:
   - Let c be the number of red vertices in $\boldsymbol{H}$. Create all possible c length strings having values in $0 \to m$, where each position in the string represents a red vertex in $\boldsymbol{H}$. Remove the all 0 string.
   - For each string, create a new $(q, 0)$–bilabelled graph diagram as follows: for each position in the string, if t is its value, then, assuming that the position represents the red vertex v, add t new edges outwards from v, adding in $t - 1$ new red vertices to make these t new edges possible.

4. From all of the diagrams found in Step 2, create all possible $(q, 0)$–bilabelled graph diagrams having both internal and external red edges, as follows. For each diagram $\boldsymbol{H_I}$ found in Step 2:
   - Let $\boldsymbol{H}$ be the $(q, 0)$–bilabelled graph diagram from Step 1 that $\boldsymbol{H_I}$ came from. Let c be the number of red vertices in $\boldsymbol{H}$, and label these vertices in $\boldsymbol{H_I}$ as $1, \dots, c$. Create an empty set named *originals*, and add only the vertices $1, \dots, c$ in $\boldsymbol{H_I}$ that are not connected to any other red vertex in $\boldsymbol{H_I}$.
   - If *originals* is empty, then no new diagrams come from $\boldsymbol{H_I}$. Otherwise, create all possible c length strings, indexed by the vertices $1, \dots, c$, allowing only the values in the positions in *originals* to range from $0 \to m$, with the rest being 0. Remove the all 0 string. Now follow the instructions given in Step 3.2.

5. Finally, if G has loops, then, for each $(q, 0)$–bilabelled graph diagram found in Steps 1–4:
   - If c is the number of red vertices in $\boldsymbol{H}$, label these vertices as $1, \dots, c$, and create all binary strings of length c. For each string, create a new $(q, 0)$–bilabelled graph diagram as follows: for each position i in the string, if the value is 1, then attach a loop to the red vertex labelled as i.

The weight matrix is obtained from the set of all $(q, 0)$–bilabelled graph diagrams found in Steps 1–5, as follows:

6. Apply Frobenius duality to each $(q, 0)$–bilabelled graph diagram $\boldsymbol{H}$ to obtain its form as a (k, l)–bilabelled graph diagram. This is the same as dragging the labelled black vertices in $\boldsymbol{H}$ into two rows of vertices such that the vertices in the top row are ordered $1, \dots, l$ and the vertices in the bottom row are ordered $l + 1, \dots, l + k$.

7. Apply the strict $\mathbb{R}$–linear monoidal functor $\mathcal{F}^G$ to each (k, l)–bilabelled graph diagram to obtain the spanning set matrices in $\mathrm{Hom}_{\mathrm{Aut}(G)}((\mathbb{R}^n)^{\otimes k}, (\mathbb{R}^n)^{\otimes l})$. Remove all duplicate matrices as well as the all zero matrices from this set. Weight each matrix that remains in the set by a parameter, and then add them all together to give the overall $\mathrm{Aut}(G)$-equivariant weight matrix.

Acknowledgements

This work was funded by the Doctoral Scholarship for Applied Research which was awarded to the first author under Imperial College London's Department of Computing Applied Research scheme. This work will form part of the first author's PhD thesis at Imperial College London.

Impact Statement

This paper presents work that is primarily a theoretical contribution; hence we do not expect profound societal impact in the short term. However, in the medium term, a number of applications may well emerge from the theory having high levels of impact.

References

Banica, T. and Speicher, R. Liberation of orthogonal Lie groups. *Advances in Mathematics*, 222(4):1461–1501, 2009. ISSN 0001-8708. URL https://doi.org/10.1016/j.aim.2009.06.009.

Bollobas, B. *Modern Graph Theory*. Springer, 1998.

Ceccherini-Silberstein, T., Scarabotti, F., and Tolli, F. *Representation Theory of the Symmetric Groups*. Cambridge University Press, 2010.

Chassaniol, A. Study of quantum symmetries for vertex-transitive graphs using intertwiner spaces. arXiv:1904.00455, 2019.

Cohen, T. and Welling, M. Group Equivariant Convolutional Networks. In Balcan, M. F. and Weinberger, K. Q. (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 2990–2999, New York, New York, USA, 20–22 Jun 2016. PMLR. URL https://proceedings.mlr.press/v48/cohenc16.html.

Finzi, M., Welling, M., and Wilson, A. G. G. A Practical Method for Constructing Equivariant Multilayer Perceptrons for Arbitrary Matrix Groups. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 3318–3328. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/finzi21a.html.

Frucht, R. Herstellung von Graphen mit vorgegebener abstrakter Gruppe. *Compos. Math.*, 6:239–250, 1938. ISSN 0010-437X.

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. Neural message passing for quantum chemistry. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1263–1272. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/gilmer17a.html.

Godfrey, C., Rawson, M. G., Brown, D., and Kvinge, H. Fast computation of permutation equivariant layers with the partition algebra. In *ICLR 2023 Workshop on Physics for Machine Learning*, 2023. URL https://openreview.net/forum?id=VXwts-IZFi.

Goodman, R. and Wallach, N. R. *Symmetry, Representations and Invariants*. Springer, 2009.

Gori, M., Monfardini, G., and Scarselli, F. A new model for learning in graph domains. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 2, pp. 729–734 vol. 2, 2005. doi: 10.1109/IJCNN.2005.1555942.

Gromada, D. Group-theoretical graph categories. *J Algebr Comb*, 55:591–627, 2022. URL https://doi.org/10.1007/s10801-021-01063-5.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=SJU4ayYgl.

Kondor, R., Lin, Z., and Trivedi, S. Clebsch–Gordan Nets: a Fully Fourier Space Spherical Convolutional Neural Network. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/a3fc981af450752046be179185ebc8b5-Paper.pdf.

Konstas, I., Stathopoulos, V., and Jose, J. M. On Social Networks and Collaborative Recommendation. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605584836. URL https://doi.org/10.1145/1571941.1571977.

Leskovec, J., Huttenlocher, D., and Kleinberg, J. Predicting Positive and Negative Links in Online Social Networks. In *Proceedings of the 19th International Conference on World Wide Web*, pp. 641–650, New York, NY, USA, 2010. Association for Computing Machinery. ISBN 9781605587998. URL https://doi.org/10.1145/1772690.1772756.

Lim, L.-H. and Nelson, B. J. What is an equivariant neural network?, 2022. arXiv:2205.07362.

Lovász, L. Operations with structures. *Acta Mathematica Academiae Scientiarum Hungarica*, 18:321–328, 1967. URL https://doi.org/10.1007/BF02280291.

Mac Lane, S. *Categories for the Working Mathematician*. Springer New York, NY, 1998. URL https://doi.org/10.1007/978-1-4757-4721-8.

Mančinska, L. and Roberson, D. E. Quantum isomorphism is equivalent to equality of homomorphism counts from planar graphs. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 661–672, 2020. URL https://doi.org/10.1109/FOCS46700.2020.00067.

Maron, H., Ben-Hamu, H., Shamir, N., and Lipman, Y. Invariant and Equivariant Graph Networks. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Syx72jC9tm.

Pearce-Crump, E. Connecting Permutation Equivariant Neural Networks and Partition Diagrams. arXiv:2212.08648, 2022.

Pearce-Crump, E. Brauer's Group Equivariant Neural Networks. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 27461–27482. PMLR, 23–29 Jul 2023a. URL https://proceedings.mlr.press/v202/pearce-crump23a/pearce-crump23a.pdf.

Pearce-Crump, E. How Jellyfish Characterise Alternating Group Equivariant Neural Networks. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 27483–27495. PMLR, 23–29 Jul 2023b. URL https://proceedings.mlr.press/v202/pearce-crump23b/pearce-crump23b.pdf.

Ravanbakhsh, S. Universal Equivariant Multilayer Perceptrons. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 7996–8006. PMLR, 2020. URL http://proceedings.mlr.press/v119/ravanbakhsh20a.html.

Sagan, B. E. *The Symmetric Group: Representations, Combinatorial Algorithms, and Symmetric Functions*. Springer, 2000.

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009. doi: 10.1109/TNN.2008.2005605.

Segal, E. Group Representation Theory. Course Notes for Imperial College London, 2014.

Velicković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. Graph Attention Networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJXMpikCZ.

Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R. R., and Smola, A. J. Deep Sets. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/f22e4747da1aa27e363d86d40ff442fe-Paper.pdf.

A. Proofs

To prove Propositions 4.3, 4.14, and 4.21, we first need to define a strict $\mathbb{R}$–linear monoidal category. We assume throughout that all categories are **locally small**, which means that the collection of morphisms between any two objects is a set. In fact, all of the categories that we consider throughout have morphism sets that are vector spaces. Hence, the morphisms between objects become linear maps.

Definition A.1. A category $\mathcal{C}$ is said to be **strict monoidal** if it comes with a bifunctor $\otimes : \mathcal{C} \times \mathcal{C} \to \mathcal{C}$, called the tensor product, and a unit object $\mathbb{1}$, such that, for all objects X, Y, Z in $\mathcal{C}$, we have that

$$(X \otimes Y) \otimes Z = X \otimes (Y \otimes Z) \tag{19}$$

$$(\mathbb{1} \otimes X) = X = (X \otimes \mathbb{1}) \tag{20}$$

and, for all morphisms f, g, h in $\mathcal{C}$, we have that

$$(f \otimes g) \otimes h = f \otimes (g \otimes h) \tag{21}$$

$$(1_{\mathbb{1}} \otimes f) = f = (f \otimes 1_{\mathbb{1}}) \tag{22}$$

where $1_{\mathbb{1}}$ is the identity morphism $\mathbb{1} \to \mathbb{1}$. We often use the tuple $(\mathcal{C}, \otimes_{\mathcal{C}}, \mathbb{1}_{\mathcal{C}})$ to refer to the strict monoidal category $\mathcal{C}$.

We can assume that all monoidal categories are **strict** (nonstrict monoidal categories would have isomorphisms where there are equalities in Definition A.1) owing to a technical result known as Mac Lane's Coherence Theorem. See Mac Lane (1998) for more details.

Definition A.2. A category $\mathcal{C}$ is said to be **$\mathbb{R}$–linear** if, for any two objects X, Y in $\mathcal{C}$, the morphism space $\mathrm{Hom}_{\mathcal{C}}(X, Y)$ is a vector space over $\mathbb{R}$, and the composition of morphisms is $\mathbb{R}$–bilinear.

Combining Definitions A.1 and A.2, we get

Definition A.3. A category $\mathcal{C}$ is said to be **strict $\mathbb{R}$–linear monoidal** if it is a category that is both strict monoidal and $\mathbb{R}$–linear, such that the bifunctor $\otimes$ is $\mathbb{R}$–bilinear.

Proof of Proposition 4.3. $\mathcal{C}(G)$ can immediately be seen to be a strict monoidal category by the associativity of the tensor product on vector spaces and by the associativity of the tensor product on linear maps between tensor spaces.

$\mathcal{C}(G)$ is $\mathbb{R}$–linear because the morphism space $\mathrm{Hom}_G((\mathbb{R}^n)^{\otimes k}, (\mathbb{R}^n)^{\otimes l})$, for any two objects $(\mathbb{R}^n)^{\otimes k}$ and $(\mathbb{R}^n)^{\otimes l}$ in $\mathcal{C}(G)$, is a vector space over $\mathbb{R}$ and the composition of morphisms is $\mathbb{R}$-bilinear because composition is $\mathbb{R}$-bilinear for linear maps on vector spaces. It is also clear that the bifunctor $\otimes$ is $\mathbb{R}$–bilinear since it is the standard tensor product for vector spaces. □

Proof of Proposition 4.14. $\mathcal{G}$ is a strict monoidal category because the bifunctor on objects reduces to the addition operation on natural numbers, which is associative, and the bifunctor on morphisms is the tensor product of isomorphism classes of bilabelled graphs given in Definition 4.11, which is associative because both the concatenation of tuples and the (set) unions of graphs are associative operations.

$\mathcal{G}$ is $\mathbb{R}$–linear because the morphism space between any two objects is by definition a vector space, and the composition of morphisms is $\mathbb{R}$-bilinear by definition. For the same reason, the bifunctor is also $\mathbb{R}$–bilinear. □

Proof of Proposition 4.21. This is effectively the same proof as the one that is given for Proposition 4.3, except we replace linear maps by matrices. □

In order to prove Theorem 4.25, we first need to recall the definition of a strict $\mathbb{R}$–linear monoidal functor.

Definition A.4. Suppose that $(\mathcal{C}, \otimes_{\mathcal{C}}, \mathbb{1}_{\mathcal{C}})$ and $(\mathcal{D}, \otimes_{\mathcal{D}}, \mathbb{1}_{\mathcal{D}})$ are two strict $\mathbb{R}$–linear monoidal categories.

A **strict $\mathbb{R}$–linear monoidal functor** from $\mathcal{C}$ to $\mathcal{D}$ is a functor $\mathcal{F} : \mathcal{C} \to \mathcal{D}$ such that

1. for all objects X, Y in $\mathcal{C}$, $\mathcal{F}(X \otimes_{\mathcal{C}} Y) = \mathcal{F}(X) \otimes_{\mathcal{D}} \mathcal{F}(Y)$

2. for all morphisms f, g in $\mathcal{C}$, $\mathcal{F}(f \otimes_{\mathcal{C}} g) = \mathcal{F}(f) \otimes_{\mathcal{D}} \mathcal{F}(g)$
3. $\mathcal{F}(\mathbb{1}_{\mathcal{C}}) = \mathbb{1}_{\mathcal{D}}$, and
4. for all objects X, Y in $\mathcal{C}$, the map
$$\mathrm{Hom}_{\mathcal{C}}(X, Y) \to \mathrm{Hom}_{\mathcal{D}}(\mathcal{F}(X), \mathcal{F}(Y)) \tag{23}$$
given by $f \mapsto \mathcal{F}(f)$ is $\mathbb{R}$–linear.

Proof of Theorem 4.25. Let G be a graph having n vertices. We show each of the four conditions of Definition A.4 in turn.

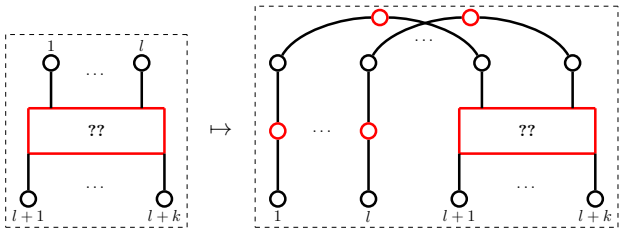
1. Let k, l be any two objects in $\mathcal{G}$. Then
$$\mathcal{F}^G(k \otimes l) = \mathcal{F}^G(k + l) = (\mathbb{R}^n)^{\otimes k+l} = (\mathbb{R}^n)^{\otimes k} \otimes (\mathbb{R}^n)^{\otimes l} = \mathcal{F}^G(k) \otimes \mathcal{F}^G(l) \tag{24}$$
2. This is Lemma 4.23.
3. It is clear from the statement of the theorem that $\mathcal{F}^G$ sends the unit object 0 in $\mathcal{G}$ to $\mathbb{R}$, which is the unit object in $\mathcal{C}^G$.
4. This is immediate from the definition of a G-homomorphism matrix.

It is clear that the functor $\mathcal{F}^G$ is full, once again by the definition of a G-homomorphism matrix. □

Proof of Proposition 4.31. Suppose that $\boldsymbol{H} = (H, \boldsymbol{k}, \boldsymbol{l})$. Let H_2 be the subgraph consisting of the subset of free vertices (including the edges that are solely between these vertices) that are entirely disconnected from any vertices that are in a connected component containing a non-free vertex. Define H_1 to be the graph that is H without H_2.

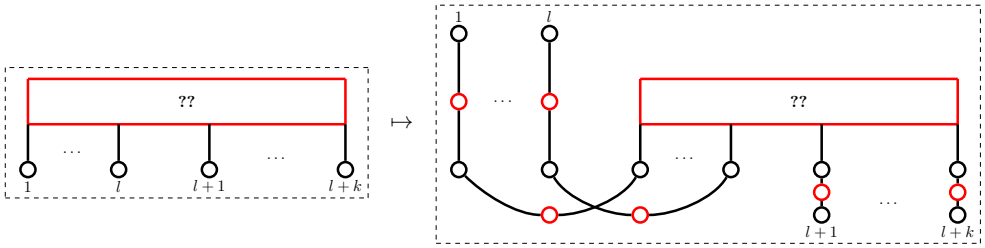
Then, by construction, H is a disjoint union of H_1 and H_2, as graphs, and, if we define $\boldsymbol{H_1} := (H_1, \boldsymbol{k}, \boldsymbol{l})$, then it is clear that
$$X_{\boldsymbol{H}}^G = c_{\boldsymbol{H_2}}^G X_{\boldsymbol{H_1}}^G \tag{25}$$
where $c_{\boldsymbol{H_2}}^G$ is a constant denoting the number of graph homomorphisms from H_2 to G, as required. □

Proof of Proposition 4.32. Firstly, there is an $\mathbb{R}$–linear isomorphism
$$\mathrm{Hom}_{\mathcal{G}}(k, l) \to \mathrm{Hom}_{\mathcal{G}}(l + k, 0) \tag{26}$$
that is given on bilabelled graph diagrams by



(27)

with inverse given by



(28)

Similarly, for any graph G having n vertices, there is an $\mathbb{R}$-linear isomorphism

$$\mathrm{Hom}_{\mathrm{Aut}(G)}((\mathbb{R}^n)^{\otimes k}, (\mathbb{R}^n)^{\otimes l}) \rightarrow \mathrm{Hom}_{\mathrm{Aut}(G)}((\mathbb{R}^n)^{\otimes l+k}, \mathbb{R}) \tag{29}$$

that is given by

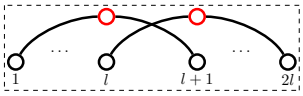
$$X_{\boldsymbol{H}}^G \mapsto \mathcal{F}^G([\boldsymbol{MM}_l]) \circ (Id^{\otimes k} \otimes X_{\boldsymbol{H}}^G) \tag{30}$$

where Id is the $n \times n$ identity matrix, with inverse given by

$$X_{\boldsymbol{H}}^G \mapsto (Id^{\otimes l} \otimes X_{\boldsymbol{H}}^G) \circ (\mathcal{F}^G([\boldsymbol{UU}_l]) \otimes Id^{\otimes k}) \tag{31}$$

Here, $\boldsymbol{MM}_l$ is the $(2l, 0)$–bilabelled graph diagram

(32)



and $\boldsymbol{UU}_l$ is its involution, as defined in Definition 4.12.

Since, for any graph G having n vertices, the functor $\mathcal{F}^G$ is strict monoidal, by Theorem 4.25, we get that the following diagram commutes:

$$\begin{tikzcd}
\mathrm{Hom}_{\mathcal{G}}(k,l) \arrow[r] \arrow[d] & \mathrm{Hom}_{\mathcal{G}}(l+k,0) \arrow[d] \\
\mathrm{Hom}_{\mathrm{Aut}(G)}((\mathbb{R}^n)^{\otimes k}, (\mathbb{R}^n)^{\otimes l}) \arrow[r] & \mathrm{Hom}_{\mathrm{Aut}(G)}((\mathbb{R}^n)^{\otimes l+k}, \mathbb{R})
\end{tikzcd} \tag{33}$$

Since $\mathcal{F}^G$ gives a bijective correspondence between all isomorphism classes of bilabelled graphs and the spanning set elements for $\mathrm{Hom}_{\mathrm{Aut}(G)}$, as shown in Theorem 4.27, we can use the commuting square to obtain a spanning set for $\mathrm{Hom}_{\mathrm{Aut}(G)}((\mathbb{R}^n)^{\otimes k+l}, \mathbb{R})$ from the spanning set for $\mathrm{Hom}_{\mathrm{Aut}(G)}((\mathbb{R}^n)^{\otimes k}, (\mathbb{R}^n)^{\otimes l})$.

Now, fixing $q, m \geq 0$ such that $q + m = k + l$, we can run the arrows of the commuting square in reverse to obtain a spanning set for $\mathrm{Hom}_{\mathrm{Aut}(G)}((\mathbb{R}^n)^{\otimes q}, (\mathbb{R}^n)^{\otimes m})$, as required. □

B. Recovery of the Characterisation of the Equivariant, Linear Maps for the Symmetric Group

We saw in Example 2.10 that the automorphism group of the complement of the complete graph, $\mathrm{Aut}(\overline{K_n})$, is the symmetric group S_n. As a result of applying Theorem 4.27 to this case, we can recover the diagram basis for $\mathrm{Hom}_{S_n}((\mathbb{R}^n)^{\otimes k}, (\mathbb{R}^n)^{\otimes l})$ for any non-negative integers k and l that first appeared in Godfrey et al. (2023).

To recover the diagram basis, we reduce the set of isomorphism classes of (k, l)–bilabelled graphs $\boldsymbol{H}$ that we need to consider in (17), as follows. Firstly, we only need to consider isomorphism classes of (k, l)–bilabelled graphs $\boldsymbol{H}$ whose underlying graphs H are edgeless, since if H has either some edge between two distinct vertices or a loop, then for there to be a graph homomorphism from H to $\overline{K_n}$, the graph $\overline{K_n}$ would need to have an edge between two distinct vertices or a loop. Secondly, we only need to consider isomorphism classes of (k, l)–bilabelled graphs $\boldsymbol{H}$ whose underlying graph H is edgeless, having at most n vertices, since the image under $\mathcal{F}^G$ of any (k, l)–bilabelled graph whose underlying graph is edgeless having more than n vertices would correspond to a scalar multiple of the image of a (k, l)–bilabelled graph whose underlying graph is edgeless having at most n vertices, Thirdly, we only need to consider isomorphism classes of (k, l)–bilabelled graphs $\boldsymbol{H}$ whose underlying graph H is edgeless having at most n vertices where no vertex is left *free* in $\boldsymbol{H}$, that is, there is no red vertex in H that is not attached by a black edge to some black vertex, by Proposition 4.31.

But this subset of isomorphism classes of (k, l)–bilabelled graphs is precisely all set partitions of $\{1, \dots, l + k\}$ having at most n blocks! By construction, the image of this subset under $\mathcal{F}^G$ is a spanning set of $\mathrm{Hom}_{S_n}((\mathbb{R}^n)^{\otimes k}, (\mathbb{R}^n)^{\otimes l})$ and, by a dimension count, the image, in fact, forms a basis, which is precisely the diagram basis of Godfrey et al. (2023).

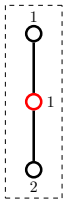
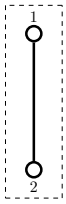
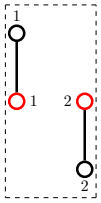
$(1,1)$–Bilabelled Graph Diagram $\boldsymbol{H}$	Set Partition Diagram	Standard Basis Element $X_{\boldsymbol{H}}^{\overline{K_4}}$
		$\begin{array}{c\|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 1 & 0 & 0 & 0 \\ 2 & 0 & 1 & 0 & 0 \\ 3 & 0 & 0 & 1 & 0 \\ 4 & 0 & 0 & 0 & 1 \end{array}$
		$\begin{array}{c\|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 1 & 1 & 1 & 1 \\ 2 & 1 & 1 & 1 & 1 \\ 3 & 1 & 1 & 1 & 1 \\ 4 & 1 & 1 & 1 & 1 \end{array}$

Figure 4. We use Theorem 4.27 to obtain the diagram basis of $\mathrm{Hom}_{S_4}(\mathbb{R}^4, \mathbb{R}^4)$ from all of the $(1,1)$–bilabelled graph diagrams.

C. Spanning Set Examples

Example C.1 (Diagram Basis for $\mathrm{Hom}_{S_n}(\mathbb{R}^n, \mathbb{R}^n)$). Suppose we take the complement of the complete graph having n vertices, $\overline{K_n}$. There is only one possible way to label the vertices of this graph, up to all automorphisms:

In Example 2.10, we saw that its automorphism group, $\mathrm{Aut}(\overline{K_n})$, is the symmetric group S_n.

Since the order of each tensor power space is 1, by the arguments in Section B, it is enough to consider the isomorphism classes of $(1,1)$–bilabelled graphs $\boldsymbol{H}$ whose underlying graph H is edgeless having at most n vertices where no vertex is left free in $\boldsymbol{H}$. Because only at most two vertices cannot be left free for $(1,1)$–bilabelled graphs $\boldsymbol{H}$, we only need to consider isomorphism classes of $(1,1)$–bilabelled graphs $\boldsymbol{H}$ whose underlying graph H is edgeless having at most two vertices.

This leads to two possibilities, namely $[\boldsymbol{H_1}]$ and $[\boldsymbol{H_2}]$, where $\boldsymbol{H_1} = (K_1, (1), (1))$ and where $\boldsymbol{H_2} = (\overline{K_2}, (1), (2))$. They are given in the left hand column of Figure 4. (Note, for example, that the bilabelled graph $(\overline{K_2}, (2), (1))$ is in the same isomorphism class as $\boldsymbol{H_2}$.)

In the right hand column of Figure 4, we show the corresponding G-homomorphism matrices for the case where $G = \overline{K_4}$, that is, $n = 4$, with the case for general n being very similar. Here we have used Definition 4.15 to find the entries of each of the matrices. To be clear, for general n, the (i,j)-entry of $X_{\boldsymbol{H_1}}^{\overline{K_n}}$, by definition, is equal to the number of graph homomorphisms from K_1 to $\overline{K_n}$ such that 1 is mapped to i and 1 is mapped to j, which is $\delta_{i,j}$. By contrast, the (i,j)-entry of $X_{\boldsymbol{H_2}}^{\overline{K_n}}$, is equal to the number of graph homomorphisms from $\overline{K_2}$ to $\overline{K_n}$ such that 2 is mapped to i and 1 is mapped to j, which is 1 for all $i, j \in [n]$.

In the middle column, we have also shown the equivalent set partition diagrams that appear in Pearce-Crump (2022) to highlight how the $(1,1)$–bilabelled graph diagrams are related to set partition diagrams in this case. This is also consistent with the procedure that is given in the orange box since the longest path in $\overline{K_n}$ is 0. Hence, only the set partitions of $\{1,2\}$, expressed as $(2,0)$–bilabelled graph diagrams that are then mapped under Frobenius duality to $(1,1)$–bilabelled graph diagrams, are needed to obtain a spanning set — which is actually a basis — of $\mathrm{Hom}_{S_n}(\mathbb{R}^n, \mathbb{R}^n)$.

Example C.2 (Basis for $\mathrm{Hom}_{D_4}(\mathbb{R}^4, \mathbb{R}^4)$). In Example 2.12, we said that the automorphism group of two copies of the complete graph on two vertices, $\mathrm{Aut}(2K_2)$, is isomorphic to the dihedral group D_4 of order 8.

There are three different ways to label the vertices of the graph $2K_2$, up to all automorphisms, namely:

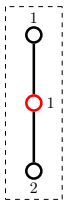
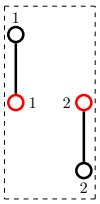
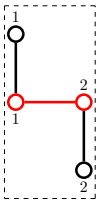
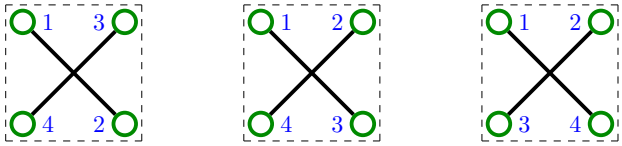
$(1,1)$–Bilabelled Graph Diagram $\boldsymbol{H}$	Standard Basis Element $X_{\boldsymbol{H}}^{A}$	Standard Basis Element $X_{\boldsymbol{H}}^{B}$	Standard Basis Element $X_{\boldsymbol{H}}^{C}$
	$\begin{array}{c\|c} & \begin{matrix}1&2&3&4\end{matrix} \\ \hline \begin{matrix}1\\2\\3\\4\end{matrix} & \begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}\end{array}$	$\begin{array}{c\|c} & \begin{matrix}1&2&3&4\end{matrix} \\ \hline \begin{matrix}1\\2\\3\\4\end{matrix} & \begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}\end{array}$	$\begin{array}{c\|c} & \begin{matrix}1&2&3&4\end{matrix} \\ \hline \begin{matrix}1\\2\\3\\4\end{matrix} & \begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}\end{array}$
	$\begin{array}{c\|c} & \begin{matrix}1&2&3&4\end{matrix} \\ \hline \begin{matrix}1\\2\\3\\4\end{matrix} & \begin{bmatrix}1&1&1&1\\1&1&1&1\\1&1&1&1\\1&1&1&1\end{bmatrix}\end{array}$	$\begin{array}{c\|c} & \begin{matrix}1&2&3&4\end{matrix} \\ \hline \begin{matrix}1\\2\\3\\4\end{matrix} & \begin{bmatrix}1&1&1&1\\1&1&1&1\\1&1&1&1\\1&1&1&1\end{bmatrix}\end{array}$	$\begin{array}{c\|c} & \begin{matrix}1&2&3&4\end{matrix} \\ \hline \begin{matrix}1\\2\\3\\4\end{matrix} & \begin{bmatrix}1&1&1&1\\1&1&1&1\\1&1&1&1\\1&1&1&1\end{bmatrix}\end{array}$
	$\begin{array}{c\|c} & \begin{matrix}1&2&3&4\end{matrix} \\ \hline \begin{matrix}1\\2\\3\\4\end{matrix} & \begin{bmatrix}0&1&0&0\\1&0&0&0\\0&0&0&1\\0&0&1&0\end{bmatrix}\end{array}$	$\begin{array}{c\|c} & \begin{matrix}1&2&3&4\end{matrix} \\ \hline \begin{matrix}1\\2\\3\\4\end{matrix} & \begin{bmatrix}0&0&1&0\\0&0&0&1\\1&0&0&0\\0&1&0&0\end{bmatrix}\end{array}$	$\begin{array}{c\|c} & \begin{matrix}1&2&3&4\end{matrix} \\ \hline \begin{matrix}1\\2\\3\\4\end{matrix} & \begin{bmatrix}0&0&0&1\\0&0&1&0\\0&1&0&0\\1&0&0&0\end{bmatrix}\end{array}$

Figure 5. Depending on how D_4 is embedded as a subgroup of S_4, we obtain a basis of $\mathrm{Hom}_{D_4}(\mathbb{R}^4, \mathbb{R}^4)$, where here D_4 refers to the specific embedding in S_4 that is obtained from the different labellings of the vertices of $2K_2$, considered up to all automorphisms.



We will refer to these graphs as A, B and C, respectively, throughout the rest of this example.

The automorphism group of each graph is isomorphic to D_4, but each automorphism group is a different embedding of D_4 in S_4. Said differently, while each automorphism group is isomorphic to D_4, the elements in each automorphism group are not the same when thought of as elements of S_4. Specifically, the group corresponding to the first diagram is $\langle (1423), (13)(24) \rangle$, whereas the second is $\langle (1432), (12)(34) \rangle$, and the third is $\langle (1342), (12)(34) \rangle$.

Consequently, when we would like to find a spanning set for $\mathrm{Hom}_{D_4}(\mathbb{R}^4, \mathbb{R}^4)$, we need to consider which embedding of D_4 in S_4 we are referring to, or, more simply, which labels we have used for the graph $2K_2$. As a result, we will obtain a spanning set (which we will show is actually a basis) for $\mathrm{Hom}_{D_4}(\mathbb{R}^4, \mathbb{R}^4)$, one for each embedding of D_4 in S_4. Note that these spanning sets, whilst different, will all be isomorphic under a change of basis that reorders the standard basis vectors in $\mathbb{R}^4$. This is a consequence of all of the automorphism groups being isomorphic to each other.

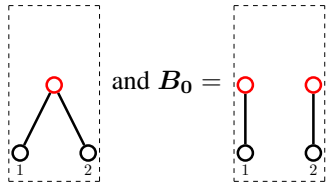
We present the bases for each instance of $2K_2$ in Figure 5.

We focus on finding the basis for the automorphism group of the graph A, noting that the approach for the other graphs is similar. We follow the procedure that is given in the orange box. We see that the longest path m between any two vertices in A is 1.

Step 1: We calculate all set partitions of $[l+k] = \{1, 2\}$ and express them as $(2,0)$–bilabelled graph diagrams, labelling only the black vertices.

They are given by

$$\boldsymbol{A_0} = \quad \text{and } \boldsymbol{B_0} = \tag{34}$$



Step 2: From $\boldsymbol{A_0}$ and $\boldsymbol{B_0}$, we calculate all possible $(2,0)$–bilabelled graph diagrams that have only internal red edges between red vertices.

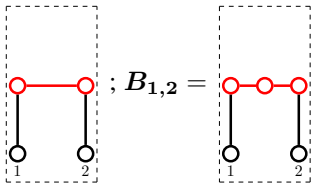
Since the number of red vertices c in $\boldsymbol{A_0}$ is 1, this implies that the number of edges in the complete graph K_1 is 0, and so we do not obtain any new $(2,0)$–bilabelled graph diagrams from $\boldsymbol{A_0}$.

However, for $\boldsymbol{B_0}$:

- The number of red vertices, c, in $\boldsymbol{B_0}$ is 2. Hence the number of edges in the complete graph K_2, e, is 1.
- We now create all possible length $e = 1$ strings having values in $0 \rightarrow 2m = 2$. Remove the all 0 string.

Hence we obtain the length one strings 1 and 2. We use each of these strings to create a new $(2,0)$–bilabelled graph diagram from $\boldsymbol{B_0}$, which we shall call $\boldsymbol{B_{1,1}}$ and $\boldsymbol{B_{1,2}}$, by inserting t new edges between the two red vertices, where t equals 1 and 2 respectively. Hence, $\boldsymbol{B_{1,1}}$ and $\boldsymbol{B_{1,2}}$ are given as follows:

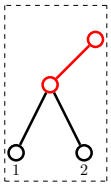
$$\boldsymbol{B_{1,1}} = \quad ; \boldsymbol{B_{1,2}} = \tag{35}$$



Step 3: From $\boldsymbol{A_0}$ and $\boldsymbol{B_0}$, we now calculate all possible $(2,0)$–bilabelled graph diagrams that have only external red edges between red vertices.

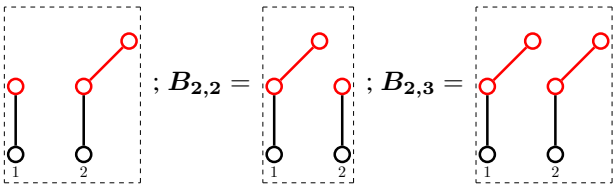
For $\boldsymbol{A_0}$, as the number of red vertices is 1, we create all length one strings having entries in 0 to $m = 1$, removing the all 0 string. We create a new $(2,0)$–bilabelled graph diagram from the string 1 by adding 1 new red edge outwards from the red vertex in $\boldsymbol{A_0}$, adding in a new red vertex to make this new red edge possible. Hence we obtain

$$\boldsymbol{A_2} = \tag{36}$$



For $\boldsymbol{B_0}$, as the number of red vertices is 2, we create all length two strings having entries in 0 to $m = 1$, removing the all 0 string. Hence we create three new $(2,0)$–bilabelled graph diagrams from the strings 01, 10 and 11, by adding 1 new red edge outwards from the red vertices in $\boldsymbol{A_0}$ corresponding to the ones in the string, adding in a new red vertex for each new red edge. Hence we obtain

$$\boldsymbol{B_{2,1}} = \quad ; \boldsymbol{B_{2,2}} = \quad ; \boldsymbol{B_{2,3}} = \tag{37}$$



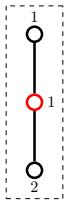
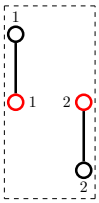
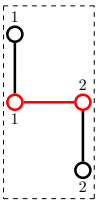
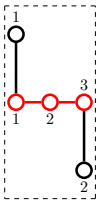
Step 4: We calculate all possible $(2,0)$–bilabelled graph diagrams that have external red edges from $\boldsymbol{B_{1,1}}$ and $\boldsymbol{B_{2,2}}$.

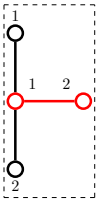
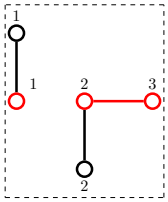
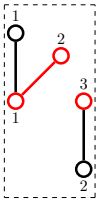
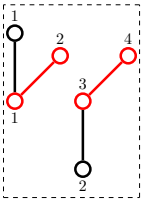
We do not create any new $(2,0)$–bilabelled graph diagrams in this step as the set *originals* is empty for $\boldsymbol{B_{1,1}}$ and $\boldsymbol{B_{2,2}}$.

Step 5: As A does not have any loops, we immediately move onto Step 6.

Step 6: We now apply Frobenius duality to each $(2,0)$–bilabelled graph diagram found in Steps 1 to 5 inclusive to obtain their form as $(1,1)$–bilabelled graph diagrams.

This is equivalent to dragging the black vertex labelled 1 up to the top row. At this stage, we choose to arbitrarily label the red vertices as well. Using the same names for the bilabelled graph diagrams, we obtain:

$\boldsymbol{A_0} =$  ; $\boldsymbol{B_0} =$  ; $\boldsymbol{B_{1,1}} =$  ; $\boldsymbol{B_{1,2}} =$  (38)

$\boldsymbol{A_2} =$  ; $\boldsymbol{B_{2,1}} =$  ; $\boldsymbol{B_{2,2}} =$  ; $\boldsymbol{B_{2,3}} =$  (39)

Step 7: We calculate the A-homomorphism matrices that correspond to the $(1,1)$–bilabelled graph diagrams given in Step 6. We remove all duplicate matrices as well as any all zero matrices, weight those that remain, and then add them together to obtain the weight matrix for an $\mathrm{Aut}(A) \cong D_4$-equivariant linear layer function from $\mathbb{R}^4$ to $\mathbb{R}^4$.

The A-homomorphism matrices that correspond to the $(1,1)$–bilabelled graph diagrams given in (38) are

$$\begin{array}{c} \\ 1\\2\\3\\4 \end{array}\begin{array}{c} \begin{array}{cccc}1&2&3&4\end{array}\\ \begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}\end{array} ; \begin{array}{c} \\ 1\\2\\3\\4 \end{array}\begin{array}{c} \begin{array}{cccc}1&2&3&4\end{array}\\ \begin{bmatrix}1&1&1&1\\1&1&1&1\\1&1&1&1\\1&1&1&1\end{bmatrix}\end{array} ; \begin{array}{c} \\ 1\\2\\3\\4 \end{array}\begin{array}{c} \begin{array}{cccc}1&2&3&4\end{array}\\ \begin{bmatrix}0&1&0&0\\1&0&0&0\\0&0&0&1\\0&0&1&0\end{bmatrix}\end{array} \text{ and } \begin{array}{c} \\ 1\\2\\3\\4 \end{array}\begin{array}{c} \begin{array}{cccc}1&2&3&4\end{array}\\ \begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}\end{array} \tag{40}$$

whereas the A-homomorphism matrices that correspond to the $(1,1)$–bilabelled graph diagrams given in (39) are

$$\begin{array}{c} \\ 1\\2\\3\\4 \end{array}\begin{array}{c} \begin{array}{cccc}1&2&3&4\end{array}\\ \begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}\end{array} ; \begin{array}{c} \\ 1\\2\\3\\4 \end{array}\begin{array}{c} \begin{array}{cccc}1&2&3&4\end{array}\\ \begin{bmatrix}1&1&1&1\\1&1&1&1\\1&1&1&1\\1&1&1&1\end{bmatrix}\end{array} ; \begin{array}{c} \\ 1\\2\\3\\4 \end{array}\begin{array}{c} \begin{array}{cccc}1&2&3&4\end{array}\\ \begin{bmatrix}1&1&1&1\\1&1&1&1\\1&1&1&1\\1&1&1&1\end{bmatrix}\end{array} \text{ and } \begin{array}{c} \\ 1\\2\\3\\4 \end{array}\begin{array}{c} \begin{array}{cccc}1&2&3&4\end{array}\\ \begin{bmatrix}1&1&1&1\\1&1&1&1\\1&1&1&1\\1&1&1&1\end{bmatrix}\end{array} \tag{41}$$

respectively.

Clearly, only the A-homomorphism matrices corresponding to $\boldsymbol{A_0}$, $\boldsymbol{B_0}$ and $\boldsymbol{B_{1,1}}$ are unique, and so they form a spanning set for $\mathrm{Hom}_{D_4}(\mathbb{R}^4, \mathbb{R}^4)$, for $D_4 = \mathrm{Aut}(A) = \langle (1423), (12)(34) \rangle$, considered as a subgroup of S_4.

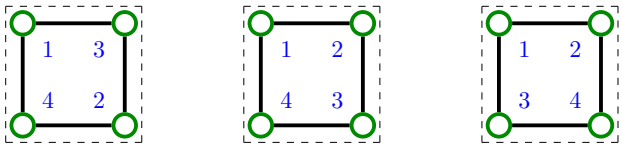
Hence the weight matrix that we obtain for an $\mathrm{Aut}(A) \cong D_4$-equivariant linear layer function from $\mathbb{R}^4$ to $\mathbb{R}^4$ is

$$\begin{array}{c} \\ 1\\2\\3\\4 \end{array}\begin{array}{c} \begin{array}{cccc}1&2&3&4\end{array}\\ \begin{bmatrix}\lambda_{1,2}&\lambda_{2,3}&\lambda_2&\lambda_2\\\lambda_{2,3}&\lambda_{1,2}&\lambda_2&\lambda_2\\\lambda_2&\lambda_2&\lambda_{1,2}&\lambda_{2,3}\\\lambda_2&\lambda_2&\lambda_{2,3}&\lambda_{1,2}\end{bmatrix}\end{array} \tag{42}$$

for weights $\lambda_1, \lambda_2, \lambda_3 \in \mathbb{R}$, where $\lambda_{i,j}$ means $\lambda_i + \lambda_j$.

Note that, for this example, each spanning set corresponding to an instance of $2K_2$ happens to be a basis of $\mathrm{Hom}_{D_4}(\mathbb{R}^4, \mathbb{R}^4)$. Indeed, for each instance of $2K_2$, the subset consisting of the first two matrices in each column in Figure 5 is a basis for $\mathrm{Hom}_{S_4}(\mathbb{R}^4, \mathbb{R}^4)$, by Example C.1, and since D_4 is a proper subgroup of S_4 and the spanning set for $\mathrm{Hom}_{D_4}(\mathbb{R}^4, \mathbb{R}^4)$ includes one further matrix that is not a linear combination of these two matrices, this implies that it must be a basis.

Note further, by Example 2.11, that we could have used the cycle graph C_4 to obtain automorphism groups that are isomorphic to D_4. In fact, if we label the cycle graphs as follows, we obtain



which are the complements of the three instances A, B, C of $2K_2$. This is clear since $C_4 = \overline{2K_2}$. In fact, in this case, we actually have that $\mathrm{Aut}(C_4) = \mathrm{Aut}(2K_2)$, considered as subgroups of S_4, for each of the three possible labellings of the vertices.

However, the basis for $\mathrm{Hom}_{D_4}(\mathbb{R}^4, \mathbb{R}^4)$ for each version of D_4 coming from the three instances $\overline{A}, \overline{B}, \overline{C}$ of C_4 is different, but related, to each of those for the three instances A, B, C of $2K_2$. We present these bases in Figure 6. Note that only the matrix corresponding to the third $(1,1)$–bilabelled graph diagram $\boldsymbol{B_{1,1}}$ has changed, so even though the embeddings of D_4 in S_4 are the same for a given labelling of the vertices, how those vertices are connected (resulting in either the square C_4 or its complement $2K_2$) affects the basis that we obtain for $\mathrm{Hom}_{D_4}(\mathbb{R}^4, \mathbb{R}^4)$. In fact, it is clear that, for each of $G = A, B, C$, we have that

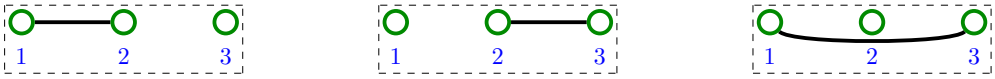
$$X^{\overline{G}}_{\boldsymbol{B_{1,1}}} = J - I - X^{G}_{\boldsymbol{B_{1,1}}} \tag{43}$$

where J is the all ones matrix, and I is the identity matrix. By Example 4.17, we see that this is, in fact, the equation

$$A_{\overline{G}} = J - I - A_G \tag{44}$$

which describes the commonly known result of how to obtain the adjacency matrix of the complement of a graph G from the adjacency matrix of the original graph G!

Example C.3 (Spanning Set for $\mathrm{Hom}_{S_2}(\mathbb{R}^3, \mathbb{R}^3)$). It is clear that, for a graph G having $n = 3$ vertices such that its automorphism group is S_2, there are three different ways to label the vertices of the graph, up to all automorphisms, namely:



We will refer to these graphs as A, B and C, respectively, throughout the rest of this example.

We focus on graph A, with the approach for the other graphs being similar. In this case, $S_2 = \mathrm{Aut}(A) = \{id, (12)\}$ as a subgroup of S_3.

Given that the longest path m between any two vertices in A is 1, and that $l = k = 1$, we see that, by following the procedure that is given in the orange box, Steps 1 to 6 inclusive are exactly the same as in Example C.2. Hence the $(1,1)$–bilabelled graph diagrams that we need to consider are those that are given in (38) and (39).

However, given that the graph A is different from the one that appears in Example C.2, the A-homomorphism matrices that correspond to the $(1,1)$–bilabelled graph diagrams given in (38) are now

$$\begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & 1 & 0 & 0 \\ 2 & 0 & 1 & 0 \\ 3 & 0 & 0 & 1 \end{array} ; \begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & 1 & 1 & 1 \\ 2 & 1 & 1 & 1 \\ 3 & 1 & 1 & 1 \end{array} ; \begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & 0 & 1 & 0 \\ 2 & 1 & 0 & 0 \\ 3 & 0 & 0 & 0 \end{array} \text{ and } \begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & 1 & 0 & 0 \\ 2 & 0 & 1 & 0 \\ 3 & 0 & 0 & 0 \end{array} \tag{45}$$

and the A-homomorphism matrices that correspond to the $(1,1)$–bilabelled graph diagrams given in (39) are now

$$\begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & 1 & 0 & 0 \\ 2 & 0 & 1 & 0 \\ 3 & 0 & 0 & 0 \end{array} ; \begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & 1 & 1 & 0 \\ 2 & 1 & 1 & 0 \\ 3 & 1 & 1 & 0 \end{array} ; \begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & 1 & 1 & 1 \\ 2 & 1 & 1 & 1 \\ 3 & 0 & 0 & 0 \end{array} \text{ and } \begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & 1 & 1 & 0 \\ 2 & 1 & 1 & 0 \\ 3 & 0 & 0 & 0 \end{array} \tag{46}$$

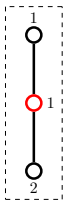
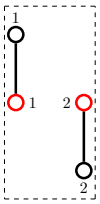
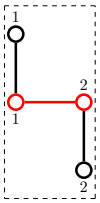
$(1,1)$–Bilabelled Graph Diagram $\boldsymbol{H}$	Standard Basis Element $X_{\boldsymbol{H}}^{\overline{A}}$	Standard Basis Element $X_{\boldsymbol{H}}^{\overline{B}}$	Standard Basis Element $X_{\boldsymbol{H}}^{\overline{C}}$
	$\begin{array}{c\|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 1 & 0 & 0 & 0 \\ 2 & 0 & 1 & 0 & 0 \\ 3 & 0 & 0 & 1 & 0 \\ 4 & 0 & 0 & 0 & 1 \end{array}$	$\begin{array}{c\|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 1 & 0 & 0 & 0 \\ 2 & 0 & 1 & 0 & 0 \\ 3 & 0 & 0 & 1 & 0 \\ 4 & 0 & 0 & 0 & 1 \end{array}$	$\begin{array}{c\|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 1 & 0 & 0 & 0 \\ 2 & 0 & 1 & 0 & 0 \\ 3 & 0 & 0 & 1 & 0 \\ 4 & 0 & 0 & 0 & 1 \end{array}$
	$\begin{array}{c\|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 1 & 1 & 1 & 1 \\ 2 & 1 & 1 & 1 & 1 \\ 3 & 1 & 1 & 1 & 1 \\ 4 & 1 & 1 & 1 & 1 \end{array}$	$\begin{array}{c\|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 1 & 1 & 1 & 1 \\ 2 & 1 & 1 & 1 & 1 \\ 3 & 1 & 1 & 1 & 1 \\ 4 & 1 & 1 & 1 & 1 \end{array}$	$\begin{array}{c\|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 1 & 1 & 1 & 1 \\ 2 & 1 & 1 & 1 & 1 \\ 3 & 1 & 1 & 1 & 1 \\ 4 & 1 & 1 & 1 & 1 \end{array}$
	$\begin{array}{c\|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 0 & 1 & 1 \\ 2 & 0 & 0 & 1 & 1 \\ 3 & 1 & 1 & 0 & 0 \\ 4 & 1 & 1 & 0 & 0 \end{array}$	$\begin{array}{c\|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 1 & 0 & 1 \\ 2 & 1 & 0 & 1 & 0 \\ 3 & 0 & 1 & 0 & 1 \\ 4 & 1 & 0 & 1 & 0 \end{array}$	$\begin{array}{c\|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 1 & 1 & 0 \\ 2 & 1 & 0 & 0 & 1 \\ 3 & 1 & 0 & 0 & 1 \\ 4 & 0 & 1 & 1 & 0 \end{array}$

Figure 6. Choosing the cycle graph C_4 instead to obtain an embedding of D_4 inside S_4, we obtain a basis of $\mathrm{Hom}_{D_4}(\mathbb{R}^4, \mathbb{R}^4)$, one for each possible labelling of the vertices of C_4, that is related to the basis obtained from the complement graph, $2K_2$.

All of the A-homomorphism matrices are unique except for those coming from $\boldsymbol{B_{1,2}}$ and $\boldsymbol{A_2}$, and so by removing the matrix corresponding to $\boldsymbol{A_2}$, we obtain a spanning set $\mathrm{Hom}_{S_2}(\mathbb{R}^3, \mathbb{R}^3)$, for $S_2 = \mathrm{Aut}(A) = \langle (id, (12)) \rangle$, as a subgroup of S_3. We label the remaining matrices in the order in which they are presented using the index set $\{1, \dots, 7\}$.

Hence the weight matrix that we obtain for an $\mathrm{Aut}(A) \cong S_2$-equivariant linear layer function from $\mathbb{R}^3$ to $\mathbb{R}^3$ is

$$\begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & \lambda_{1,2,4,5,6,7} & \lambda_{2,3,5,6,7} & \lambda_{2,6} \\ 2 & \lambda_{2,3,5,6,7} & \lambda_{1,2,4,5,6,7} & \lambda_{2,6} \\ 3 & \lambda_{2,5} & \lambda_{2,5} & \lambda_{1,2} \end{array} \tag{47}$$

for weights $\lambda_1, \dots, \lambda_7 \in \mathbb{R}$.

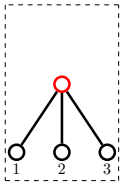
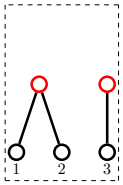
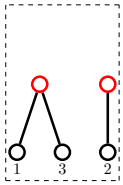
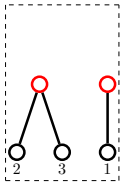
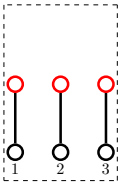
We can find the elements of the spanning set for $\mathrm{Hom}_{S_2}(\mathbb{R}^3, \mathbb{R}^3)$ for the other two embeddings of S_2 in S_3, given by $\mathrm{Aut}(B)$ and $\mathrm{Aut}(C)$. For $\mathrm{Aut}(B)$, we perform the permutation (13) on each index of the rows and columns of the spanning set matrices found for $\mathrm{Aut}(A)$, and for $\mathrm{Aut}(C)$, we perform the permutation (23) instead.

Example C.4 (Spanning Set for $\mathrm{Hom}_{S_2}((\mathbb{R}^3)^{\otimes 2}, \mathbb{R}^3)$). We refer to the same three graphs, A, B, C, that were given in Example C.3, and, once again, focus on graph A, with the approach for the other graphs being similar.

Recall that, in this case, $S_2 = \mathrm{Aut}(A) = \{id, (12)\}$, as a subgroup of S_3. We still have that the longest path in A is 1. We follow the procedure that is given in the orange box. However, in order to keep the A-homomorphism matrices in close proximity with the bilabelled graph diagrams that they correspond to, we have chosen to apply Step 6 and the first part of Step 7 during Steps 1 through 5 inclusive.

Step 1: We calculate all set partitions of $[l + k] = \{1, 2, 3\}$ and express them as $(3, 0)$–bilabelled graph diagrams, labelling only the black vertices.

They are given by

$\boldsymbol{A_0} =$  ; $\boldsymbol{B_0} =$  ; $\boldsymbol{C_0} =$  ; $\boldsymbol{D_0} =$  and $\boldsymbol{E_0} =$  (48)

and correspond, after Frobenius duality, to the A-homomorphism matrices

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array}\begin{array}{c} \begin{array}{ccc|ccc|ccc} 1,1 & 1,2 & 1,3 & 2,1 & 2,2 & 2,3 & 3,1 & 3,2 & 3,3 \end{array} \\ \left[\begin{array}{ccc|ccc|ccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right] \end{array} ; \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array}\begin{array}{c} \begin{array}{ccc|ccc|ccc} 1,1 & 1,2 & 1,3 & 2,1 & 2,2 & 2,3 & 3,1 & 3,2 & 3,3 \end{array} \\ \left[\begin{array}{ccc|ccc|ccc} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{array}\right] \end{array} ; \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array}\begin{array}{c} \begin{array}{ccc|ccc|ccc} 1,1 & 1,2 & 1,3 & 2,1 & 2,2 & 2,3 & 3,1 & 3,2 & 3,3 \end{array} \\ \left[\begin{array}{ccc|ccc|ccc} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{array}\right] \end{array} \tag{49}$$

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array}\begin{array}{c} \begin{array}{ccc|ccc|ccc} 1,1 & 1,2 & 1,3 & 2,1 & 2,2 & 2,3 & 3,1 & 3,2 & 3,3 \end{array} \\ \left[\begin{array}{ccc|ccc|ccc} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{array}\right] \end{array} \text{ and } \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array}\begin{array}{c} \begin{array}{ccc|ccc|ccc} 1,1 & 1,2 & 1,3 & 2,1 & 2,2 & 2,3 & 3,1 & 3,2 & 3,3 \end{array} \\ \left[\begin{array}{ccc|ccc|ccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}\right] \end{array} \tag{50}$$

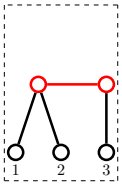
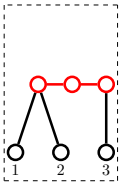
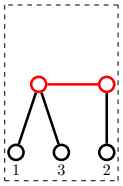
Step 2: We calculate all possible $(3, 0)$–bilabelled graph diagrams that have only internal red edges between red vertices from the five bilabelled graph diagrams given in Step 1.

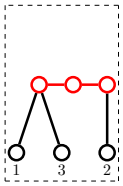
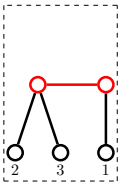
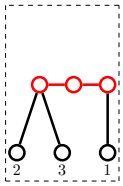
Clearly there are no new bilabelled graph diagrams coming from $\boldsymbol{A_0}$ in this step.

However, for $\boldsymbol{B_0}$, $\boldsymbol{C_0}$ and $\boldsymbol{D_0}$, calling these generically by $\boldsymbol{H}$, we have that

- The number of red vertices, c, in $\boldsymbol{H}$ is 2. Hence the number of edges in the complete graph K_2, e, is 1.
- We now create all possible length $e = 1$ strings having values in $0 \to 2m = 2$. Remove the all 0 string.

Hence we obtain the length one strings 1 and 2. We use these strings to create two new $(3, 0)$–bilabelled graph diagrams from each of $\boldsymbol{B_0}$, $\boldsymbol{C_0}$ and $\boldsymbol{D_0}$, by inserting t new edges between the two red vertices, where t equals 1 and 2, respectively. The new $(3, 0)$–bilabelled graph diagrams are given as follows:

$\boldsymbol{B_{1,1}} =$  ; $\boldsymbol{B_{1,2}} =$  ; $\boldsymbol{C_{1,1}} =$  (51)

$\boldsymbol{C_{1,2}} =$  ; $\boldsymbol{D_{1,1}} =$  ; $\boldsymbol{D_{1,2}} =$  (52)

They correspond, after Frobenius duality, to the A-homomorphism matrices

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array}\begin{array}{c} \begin{array}{ccc|ccc|ccc} 1,1 & 1,2 & 1,3 & 2,1 & 2,2 & 2,3 & 3,1 & 3,2 & 3,3 \end{array} \\ \left[\begin{array}{ccc|ccc|ccc} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \end{array} ; \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array}\begin{array}{c} \begin{array}{ccc|ccc|ccc} 1,1 & 1,2 & 1,3 & 2,1 & 2,2 & 2,3 & 3,1 & 3,2 & 3,3 \end{array} \\ \left[\begin{array}{ccc|ccc|ccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \end{array} ; \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array}\begin{array}{c} \begin{array}{ccc|ccc|ccc} 1,1 & 1,2 & 1,3 & 2,1 & 2,2 & 2,3 & 3,1 & 3,2 & 3,3 \end{array} \\ \left[\begin{array}{ccc|ccc|ccc} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \end{array} \tag{53}$$

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array}\begin{array}{c} \begin{array}{ccc|ccc|ccc} 1,1 & 1,2 & 1,3 & 2,1 & 2,2 & 2,3 & 3,1 & 3,2 & 3,3 \end{array} \\ \left[\begin{array}{ccc|ccc|ccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \end{array} ; \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array}\begin{array}{c} \begin{array}{ccc|ccc|ccc} 1,1 & 1,2 & 1,3 & 2,1 & 2,2 & 2,3 & 3,1 & 3,2 & 3,3 \end{array} \\ \left[\begin{array}{ccc|ccc|ccc} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \end{array} ; \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array}\begin{array}{c} \begin{array}{ccc|ccc|ccc} 1,1 & 1,2 & 1,3 & 2,1 & 2,2 & 2,3 & 3,1 & 3,2 & 3,3 \end{array} \\ \left[\begin{array}{ccc|ccc|ccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \end{array} \tag{54}$$

For $\boldsymbol{E_0}$, we have that

- The number of red vertices, c, in $\boldsymbol{E_0}$ is 3. Hence the number of edges in the complete graph K_3, e, is 3.
- We now create all possible length $e = 3$ strings having values in $0 \rightarrow 2m = 2$. Remove the all 0 string.

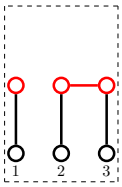
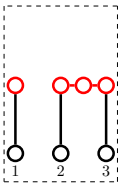
There are 26 such strings. If we label the red vertices in $\boldsymbol{E_0}$, without loss of generality, as 1, 2 and 3, from left to right, then we can let the indices in each string refer to possible connections between vertices 1 and 2, vertices 1 and 3, and vertices 2 and 3, written (12), (13) and (23), also from left to right and without loss of generality.

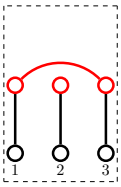
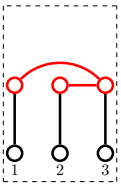
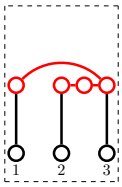
Table 1. All Possible Length 3 Strings With Entries in $0 \rightarrow 2m = 2$

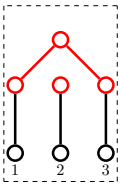
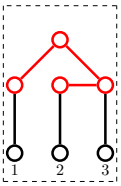
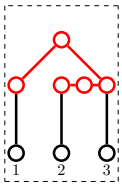
(12)(13)(23)	(12)(13)(23)	(12)(13)(23)
000	001	002
010	011	012
020	021	022
100	101	102
110	111	112
120	121	122
200	201	202
210	211	212
220	221	222

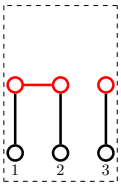
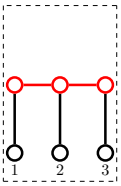
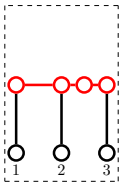
To create the 26 new $(3, 0)$–bilabelled graph diagrams, we now add in the edges (and new red vertices) between the labelled red vertices in $\boldsymbol{E_0}$, according to the entries in the string.

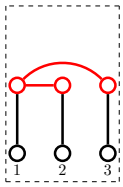
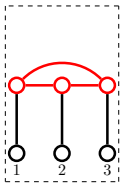
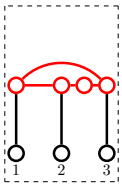
They are

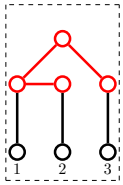
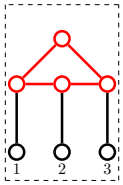
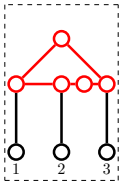
$\boldsymbol{E_{001}} =$  ; $\boldsymbol{E_{002}} =$  (55)

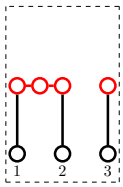
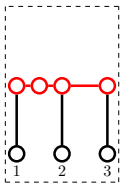
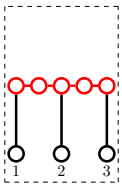
$\boldsymbol{E_{010}} =$  ; $\boldsymbol{E_{011}} =$  ; $\boldsymbol{E_{012}} =$  (56)

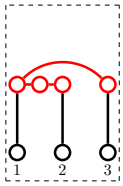
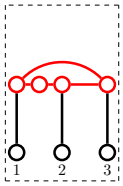
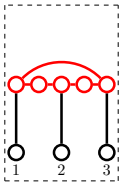
$\boldsymbol{E_{020}} =$  ; $\boldsymbol{E_{021}} =$  ; $\boldsymbol{E_{022}} =$  (57)

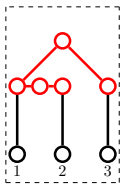
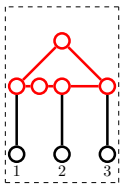
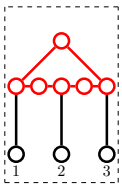
$\boldsymbol{E_{100}} =$  ; $\boldsymbol{E_{101}} =$  ; $\boldsymbol{E_{102}} =$  (58)

$\boldsymbol{E_{110}} =$  ; $\boldsymbol{E_{111}} =$  ; $\boldsymbol{E_{112}} =$  (59)

$\boldsymbol{E_{120}} =$  ; $\boldsymbol{E_{121}} =$  ; $\boldsymbol{E_{122}} =$  (60)

$\boldsymbol{E_{200}} =$  ; $\boldsymbol{E_{201}} =$  ; $\boldsymbol{E_{202}} =$  (61)

$\boldsymbol{E_{210}} =$  ; $\boldsymbol{E_{211}} =$  ; $\boldsymbol{E_{212}} =$  (62)

$\boldsymbol{E_{220}} =$  ; $\boldsymbol{E_{221}} =$  ; $\boldsymbol{E_{222}} =$  (63)

and correspond, after Frobenius duality, to the A-homomorphism matrices

$$\begin{array}{c|ccc|ccc|ccc} & 1,1 & 1,2 & 1,3 & 2,1 & 2,2 & 2,3 & 3,1 & 3,2 & 3,3 \\ \hline 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{array} ; \begin{array}{c|ccc|ccc|ccc} & 1,1 & 1,2 & 1,3 & 2,1 & 2,2 & 2,3 & 3,1 & 3,2 & 3,3 \\ \hline 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 3 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{array} \tag{64}$$

$$\begin{array}{c|ccc|ccc|ccc} & 1,1 & 1,2 & 1,3 & 2,1 & 2,2 & 2,3 & 3,1 & 3,2 & 3,3 \\ \hline 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 2 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} ; \begin{array}{c|ccc|ccc|ccc} & 1,1 & 1,2 & 1,3 & 2,1 & 2,2 & 2,3 & 3,1 & 3,2 & 3,3 \\ \hline 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} ; \begin{array}{c|ccc|ccc|ccc} & 1,1 & 1,2 & 1,3 & 2,1 & 2,2 & 2,3 & 3,1 & 3,2 & 3,3 \\ \hline 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \tag{65}$$

$$\begin{array}{c|ccc|ccc|ccc} & 1,1 & 1,2 & 1,3 & 2,1 & 2,2 & 2,3 & 3,1 & 3,2 & 3,3 \\ \hline 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 2 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} ; \begin{array}{c|ccc|ccc|ccc} & 1,1 & 1,2 & 1,3 & 2,1 & 2,2 & 2,3 & 3,1 & 3,2 & 3,3 \\ \hline 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} ; \begin{array}{c|ccc|ccc|ccc} & 1,1 & 1,2 & 1,3 & 2,1 & 2,2 & 2,3 & 3,1 & 3,2 & 3,3 \\ \hline 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \tag{66}$$

$$\begin{array}{c|ccc|ccc|ccc} & 1,1 & 1,2 & 1,3 & 2,1 & 2,2 & 2,3 & 3,1 & 3,2 & 3,3 \\ \hline 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 2 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} ; \begin{array}{c|ccc|ccc|ccc} & 1,1 & 1,2 & 1,3 & 2,1 & 2,2 & 2,3 & 3,1 & 3,2 & 3,3 \\ \hline 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} ; \begin{array}{c|ccc|ccc|ccc} & 1,1 & 1,2 & 1,3 & 2,1 & 2,2 & 2,3 & 3,1 & 3,2 & 3,3 \\ \hline 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \tag{67}$$

$$\begin{array}{c|ccc|ccc|ccc} & 1,1 & 1,2 & 1,3 & 2,1 & 2,2 & 2,3 & 3,1 & 3,2 & 3,3 \\ \hline 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} ; \begin{array}{c|ccc|ccc|ccc} & 1,1 & 1,2 & 1,3 & 2,1 & 2,2 & 2,3 & 3,1 & 3,2 & 3,3 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} ; \begin{array}{c|ccc|ccc|ccc} & 1,1 & 1,2 & 1,3 & 2,1 & 2,2 & 2,3 & 3,1 & 3,2 & 3,3 \\ \hline 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \tag{68}$$

$$\begin{array}{c|ccc|ccc|ccc} & 1,1 & 1,2 & 1,3 & 2,1 & 2,2 & 2,3 & 3,1 & 3,2 & 3,3 \\ \hline 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} ; \begin{array}{c|ccc|ccc|ccc} & 1,1 & 1,2 & 1,3 & 2,1 & 2,2 & 2,3 & 3,1 & 3,2 & 3,3 \\ \hline 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} ; \begin{array}{c|ccc|ccc|ccc} & 1,1 & 1,2 & 1,3 & 2,1 & 2,2 & 2,3 & 3,1 & 3,2 & 3,3 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \tag{69}$$

$$\begin{array}{c|ccc|ccc|ccc} & 1,1 & 1,2 & 1,3 & 2,1 & 2,2 & 2,3 & 3,1 & 3,2 & 3,3 \\ \hline 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} ; \begin{array}{c|ccc|ccc|ccc} & 1,1 & 1,2 & 1,3 & 2,1 & 2,2 & 2,3 & 3,1 & 3,2 & 3,3 \\ \hline 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} ; \begin{array}{c|ccc|ccc|ccc} & 1,1 & 1,2 & 1,3 & 2,1 & 2,2 & 2,3 & 3,1 & 3,2 & 3,3 \\ \hline 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \tag{70}$$

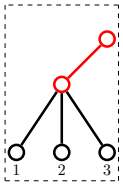
$$\begin{array}{c|ccc|ccc|ccc} & 1,1 & 1,2 & 1,3 & 2,1 & 2,2 & 2,3 & 3,1 & 3,2 & 3,3 \\ \hline 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} ; \begin{array}{c|ccc|ccc|ccc} & 1,1 & 1,2 & 1,3 & 2,1 & 2,2 & 2,3 & 3,1 & 3,2 & 3,3 \\ \hline 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} ; \begin{array}{c|ccc|ccc|ccc} & 1,1 & 1,2 & 1,3 & 2,1 & 2,2 & 2,3 & 3,1 & 3,2 & 3,3 \\ \hline 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \tag{71}$$

$$\begin{array}{c|ccc|ccc|ccc} & 1,1 & 1,2 & 1,3 & 2,1 & 2,2 & 2,3 & 3,1 & 3,2 & 3,3 \\ \hline 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} ; \begin{array}{c|ccc|ccc|ccc} & 1,1 & 1,2 & 1,3 & 2,1 & 2,2 & 2,3 & 3,1 & 3,2 & 3,3 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} ; \begin{array}{c|ccc|ccc|ccc} & 1,1 & 1,2 & 1,3 & 2,1 & 2,2 & 2,3 & 3,1 & 3,2 & 3,3 \\ \hline 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \tag{72}$$

Step 3: We calculate all possible $(3,0)$–bilabelled graph diagrams that have only external red edges between red vertices from the five bilabelled graph diagrams given in Step 1.

For $\boldsymbol{A_0}$, as the number of red vertices is 1, we create all length one strings having entries in 0 to $m = 1$, removing the all 0 string. As a result, we create a new $(3,0)$–bilabelled graph diagram from the string 1 by adding 1 new red edge outwards from the red vertex in $\boldsymbol{A_0}$, adding in a new red vertex to make this new red edge possible. Hence we obtain

$$\boldsymbol{A_2} = \tag{73}$$

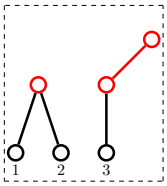


which, after Frobenius duality, corresponds to the A-homomorphism matrix

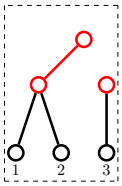
$$\begin{array}{c|ccc|ccc|ccc} & 1,1 & 1,2 & 1,3 & 2,1 & 2,2 & 2,3 & 3,1 & 3,2 & 3,3 \\ \hline 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \tag{74}$$

However, for $\boldsymbol{B_0}$, $\boldsymbol{C_0}$ and $\boldsymbol{D_0}$, calling these generically by $\boldsymbol{H}$, we know that the number of red vertices, c, in $\boldsymbol{H}$ is 2. As a result, we create all length two strings having entries in 0 to $m = 1$, removing the all 0 string. There are three such strings, 01, 10 and 11. Hence we create three new $(3,0)$–bilabelled graph diagrams using these strings from each of $\boldsymbol{B_0}$, $\boldsymbol{C_0}$ and $\boldsymbol{D_0}$, by adding 1 new red edge outwards from a red vertex that corresponds to a 1 in the string, adding in a new red vertex to make this new red edge possible. They are given by

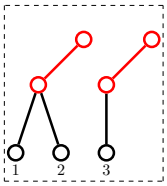
$$\boldsymbol{B_{2,1}} = \tag{75}$$



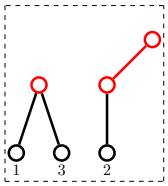
$$; \boldsymbol{B_{2,2}} =$$



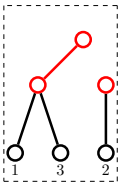
$$; \boldsymbol{B_{2,3}} =$$



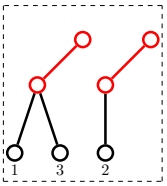
$$\boldsymbol{C_{2,1}} = \tag{76}$$

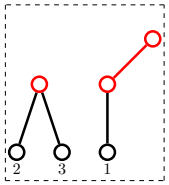
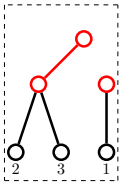
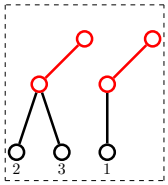


$$; \boldsymbol{C_{2,2}} =$$



$$; \boldsymbol{C_{2,3}} =$$



$\boldsymbol{D_{2,1}} =$  ; $\boldsymbol{D_{2,2}} =$  ; $\boldsymbol{D_{2,3}} =$  (77)

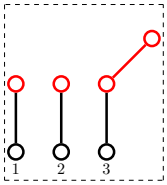
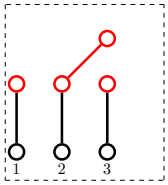
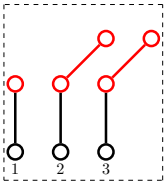
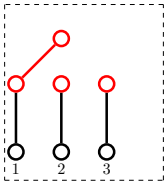
which, after Frobenius duality, correspond to the A-homomorphism matrices

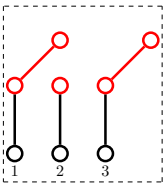
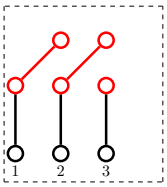
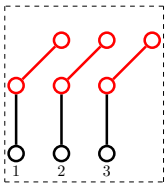
$$\begin{array}{c|ccccccccc} & 1,1 & 1,2 & 1,3 & 2,1 & 2,2 & 2,3 & 3,1 & 3,2 & 3,3 \\ \hline 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{array} ; \begin{array}{c|ccccccccc} & 1,1 & 1,2 & 1,3 & 2,1 & 2,2 & 2,3 & 3,1 & 3,2 & 3,3 \\ \hline 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} ; \begin{array}{c|ccccccccc} & 1,1 & 1,2 & 1,3 & 2,1 & 2,2 & 2,3 & 3,1 & 3,2 & 3,3 \\ \hline 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \tag{78}$$

$$\begin{array}{c|ccccccccc} & 1,1 & 1,2 & 1,3 & 2,1 & 2,2 & 2,3 & 3,1 & 3,2 & 3,3 \\ \hline 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \end{array} ; \begin{array}{c|ccccccccc} & 1,1 & 1,2 & 1,3 & 2,1 & 2,2 & 2,3 & 3,1 & 3,2 & 3,3 \\ \hline 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 2 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} ; \begin{array}{c|ccccccccc} & 1,1 & 1,2 & 1,3 & 2,1 & 2,2 & 2,3 & 3,1 & 3,2 & 3,3 \\ \hline 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \tag{79}$$

$$\begin{array}{c|ccccccccc} & 1,1 & 1,2 & 1,3 & 2,1 & 2,2 & 2,3 & 3,1 & 3,2 & 3,3 \\ \hline 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 2 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} ; \begin{array}{c|ccccccccc} & 1,1 & 1,2 & 1,3 & 2,1 & 2,2 & 2,3 & 3,1 & 3,2 & 3,3 \\ \hline 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 3 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{array} ; \begin{array}{c|ccccccccc} & 1,1 & 1,2 & 1,3 & 2,1 & 2,2 & 2,3 & 3,1 & 3,2 & 3,3 \\ \hline 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \tag{80}$$

For $\boldsymbol{E_0}$, we know that the number of red vertices, c, in $\boldsymbol{E_0}$ is 3. As a result, we create all length three strings having entries in 0 to $m = 1$, removing the all 0 string. There are seven such strings, 001, 010, 011, 100, 101, 110 and 111. Hence we create seven new $(3, 0)$–bilabelled graph diagrams using these strings from $\boldsymbol{E_0}$, by adding 1 new red edge outwards from a red vertex that corresponds to a 1 in the string, adding in a new red vertex to make this new red edge possible. They are given by

$\boldsymbol{E_{2,1}} =$  ; $\boldsymbol{E_{2,2}} =$  ; $\boldsymbol{E_{2,3}} =$  ; $\boldsymbol{E_{2,4}} =$  (81)

$\boldsymbol{E_{2,5}} =$  ; $\boldsymbol{E_{2,6}} =$  ; $\boldsymbol{E_{2,7}} =$  (82)

which, after Frobenius duality, correspond to the A-homomorphism matrices

$$\begin{array}{c|ccccccccc} & 1,1 & 1,2 & 1,3 & 2,1 & 2,2 & 2,3 & 3,1 & 3,2 & 3,3 \\ \hline 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 2 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 3 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \end{array} ; \begin{array}{c|ccccccccc} & 1,1 & 1,2 & 1,3 & 2,1 & 2,2 & 2,3 & 3,1 & 3,2 & 3,3 \\ \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 2 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 3 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \end{array} \tag{83}$$

$$\begin{array}{c|ccccccccc} & 1,1 & 1,2 & 1,3 & 2,1 & 2,2 & 2,3 & 3,1 & 3,2 & 3,3 \\ \hline 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 2 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 3 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \end{array} ; \begin{array}{c|ccccccccc} & 1,1 & 1,2 & 1,3 & 2,1 & 2,2 & 2,3 & 3,1 & 3,2 & 3,3 \\ \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \tag{84}$$

$$\begin{array}{c|ccccccccc} & 1,1 & 1,2 & 1,3 & 2,1 & 2,2 & 2,3 & 3,1 & 3,2 & 3,3 \\ \hline 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 2 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} ; \begin{array}{c|ccccccccc} & 1,1 & 1,2 & 1,3 & 2,1 & 2,2 & 2,3 & 3,1 & 3,2 & 3,3 \\ \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 2 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} ; \begin{array}{c|ccccccccc} & 1,1 & 1,2 & 1,3 & 2,1 & 2,2 & 2,3 & 3,1 & 3,2 & 3,3 \\ \hline 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 2 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \tag{85}$$

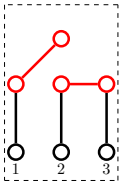
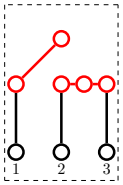
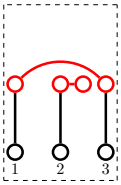
Step 4: We calculate all possible $(3, 0)$–bilabelled graph diagrams that have external red edges from the 32 bilabelled graph diagrams that were given in Step 2.

None of the bilabelled graph diagrams given in (51) and (52) will lead to new bilabelled graph diagrams, since each red vertex in each bilabelled graph diagram is connected with another red vertex in the diagram.

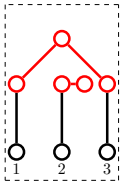
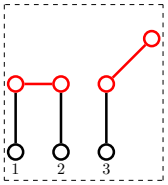
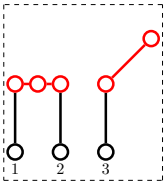
Only six bilabelled graph diagrams in (55) - (63) give rise to new bilabelled graph diagrams: they are $\boldsymbol{E_{001}}$, $\boldsymbol{E_{002}}$, $\boldsymbol{E_{010}}$, $\boldsymbol{E_{020}}$, $\boldsymbol{E_{100}}$, and $\boldsymbol{E_{200}}$, as they are the only bilabelled graph diagrams that have at least one (in this example, exactly one) red vertex that is not connected with any other vertex.

As each bilabelled graph diagram has three red vertices having only a single red vertex that is not connected with any other vertex, and since $m = 1$, this step produces exactly one string of length three, where each position in the string corresponds to a red vertex in the bilabelled graph diagram.

Hence, from $\boldsymbol{E_{001}}$, $\boldsymbol{E_{002}}$, and $\boldsymbol{E_{010}}$, we obtain

$\boldsymbol{E_{001,E}} =$  ; $\boldsymbol{E_{002,E}} =$  ; $\boldsymbol{E_{010,E}} =$  (86)

and from $\boldsymbol{E_{020}}$, $\boldsymbol{E_{100}}$, and $\boldsymbol{E_{200}}$, we obtain

$\boldsymbol{E_{020,E}} =$  ; $\boldsymbol{E_{100,E}} =$  ; $\boldsymbol{E_{200,E}} =$  (87)

These six bilabelled graph diagrams correspond, after Frobenius duality, to the following A-homomorphism matrices.

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array}\begin{array}{c} \begin{array}{ccc|ccc|ccc} 1,1 & 1,2 & 1,3 & 2,1 & 2,2 & 2,3 & 3,1 & 3,2 & 3,3 \end{array} \\ \left[\begin{array}{ccc|ccc|ccc} 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \end{array} ; \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array}\begin{array}{c} \begin{array}{ccc|ccc|ccc} 1,1 & 1,2 & 1,3 & 2,1 & 2,2 & 2,3 & 3,1 & 3,2 & 3,3 \end{array} \\ \left[\begin{array}{ccc|ccc|ccc} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \end{array} ; \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array}\begin{array}{c} \begin{array}{ccc|ccc|ccc} 1,1 & 1,2 & 1,3 & 2,1 & 2,2 & 2,3 & 3,1 & 3,2 & 3,3 \end{array} \\ \left[\begin{array}{ccc|ccc|ccc} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \end{array} \tag{88}$$

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array}\begin{array}{c} \begin{array}{ccc|ccc|ccc} 1,1 & 1,2 & 1,3 & 2,1 & 2,2 & 2,3 & 3,1 & 3,2 & 3,3 \end{array} \\ \left[\begin{array}{ccc|ccc|ccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \end{array} ; \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array}\begin{array}{c} \begin{array}{ccc|ccc|ccc} 1,1 & 1,2 & 1,3 & 2,1 & 2,2 & 2,3 & 3,1 & 3,2 & 3,3 \end{array} \\ \left[\begin{array}{ccc|ccc|ccc} 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \end{array} ; \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array}\begin{array}{c} \begin{array}{ccc|ccc|ccc} 1,1 & 1,2 & 1,3 & 2,1 & 2,2 & 2,3 & 3,1 & 3,2 & 3,3 \end{array} \\ \left[\begin{array}{ccc|ccc|ccc} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \end{array} \tag{89}$$

Step 5: There are no new bilabelled graph diagrams to consider in this step as A does not have any loops.

In total, this gives us 60 $(3, 0)$–bilabelled graph diagrams that are appropriate for A.

We have implicitly applied **Step 6** in calculating the A-homomorphism matrices during Steps 1 through 5 inclusive since we had to convert each $(3, 0)$–bilabelled graph diagram into a $(2, 1)$–bilabelled graph diagram first before applying the functor $\mathcal{F}^G$ to obtain each A-homomorphism matrix. Hence we move onto the second half of **Step 7**: we remove all duplicate matrices and the all zero matrices. We see that we are left with 31 matrices in the spanning set. After this, to obtain the weight matrix for an $\mathrm{Aut}(A) \cong S_2$-equivariant linear layer function from $(\mathbb{R}^3)^{\otimes 2}$ to $\mathbb{R}^3$, we would weight each of the 31 matrices in the spanning set and then add them together.

It is important to highlight that the vector space $\mathrm{Hom}((\mathbb{R}^3)^{\otimes 2}, \mathbb{R}^3)$, in which $\mathrm{Hom}_{S_2}((\mathbb{R}^3)^{\otimes 2}, \mathbb{R}^3)$ lives, is of dimension 27; hence, there must be linear dependencies amongst the 31 spanning set elements that we have found. However, determining these linear dependencies in general at the graph level *a priori* is left for future work.

We can also find the elements of the spanning set for $\mathrm{Hom}_{S_2}((\mathbb{R}^3)^{\otimes 2}, \mathbb{R}^3)$ for the other two embeddings of S_2 in S_3, given by $\mathrm{Aut}(B)$ and $\mathrm{Aut}(C)$. For $\mathrm{Aut}(B)$, we perform the permutation (13) on each index of the rows and columns of the spanning set matrices found for $\mathrm{Aut}(A)$, and for $\mathrm{Aut}(C)$, we perform the permutation (23) instead.

D. Adding Features and Biases

D.1. Features

We have assumed throughout that the feature dimension for all of the layers that appear in a graph automorphism group equivariant neural network is one. We can adapt all of the results that we have shown for the case where the feature dimension of the layers is greater than one.

Let G be a graph having n vertices, and suppose that an r-order tensor has a feature space of dimension d_r. We now wish to find a spanning set for

$$\mathrm{Hom}_{\mathrm{Aut}(G)}((\mathbb{R}^n)^{\otimes k} \otimes \mathbb{R}^{d_k}, (\mathbb{R}^n)^{\otimes l} \otimes \mathbb{R}^{d_l}) \tag{90}$$

in the standard basis of $\mathbb{R}^n$.

The spanning set can be found by adapting the result given in (17), and is

$$\{X^G_{\boldsymbol{H},i,j} \mid [\boldsymbol{H}] \in \mathcal{G}(k,l), i \in [d_l], j \in [d_k]\} \tag{91}$$

where now, if $\boldsymbol{H} := (H, \boldsymbol{k}, \boldsymbol{l})$, then $X^G_{\boldsymbol{H},i,j}$ is the $(n^l \times d_l) \times (n^k \times d_k)$ matrix that has (I, i, J, j)–entry given by the number of graph homomorphisms from H to G such that $\boldsymbol{l}$ is mapped to I and $\boldsymbol{k}$ is mapped to J, and is 0 otherwise.

D.2. Biases

Including bias terms in the layer functions of an $\mathrm{Aut}(G)$-equivariant neural network is also possible. If we consider a learnable linear layer in $\mathrm{Hom}_{\mathrm{Aut}(G)}((\mathbb{R}^n)^{\otimes k}, (\mathbb{R}^n)^{\otimes l})$, Pearce-Crump (2022) shows that the $\mathrm{Aut}(G)$-equivariance of the bias function, $\beta : ((\mathbb{R}^n)^{\otimes k}, \rho_k) \rightarrow ((\mathbb{R}^n)^{\otimes l}, \rho_l)$, needs to satisfy

$$c = \rho_l(g)c \tag{92}$$

for all $g \in \mathrm{Aut}(G)$ and for all $c \in (\mathbb{R}^n)^{\otimes l}$.

Since any $c \in (\mathbb{R}^n)^{\otimes l}$ satisfying (92) can be viewed as an element of $\mathrm{Hom}_{\mathrm{Aut}(G)}(\mathbb{R}, (\mathbb{R}^n)^{\otimes l})$, to find the matrix form of c, all we need to do is to find a spanning set for $\mathrm{Hom}_{\mathrm{Aut}(G)}(\mathbb{R}, (\mathbb{R}^n)^{\otimes l})$.

But this is simply a matter of applying Theorem 4.27, setting $k = 0$.