Imperial College London

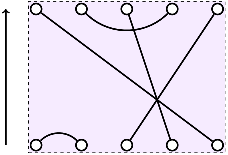
# Diagrammatic Mathematics for Equivariant Deep Learning Architectures

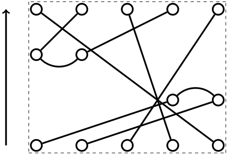
CaLISTA Workshop Geometry-Informed Machine Learning

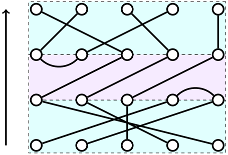
Edward Pearce-Crump

Imperial College London

# Diagrams, Diagrams, Diagrams!







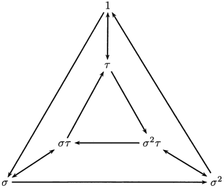
# Machine Learning, Symmetries and Groups

In machine learning, we would like to **develop principled approaches** for constructing neural networks.

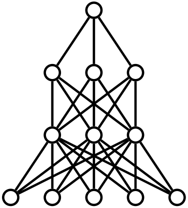
One important approach is to

$$A + B + C = C + A + B$$

**identify symmetries** that exist in data



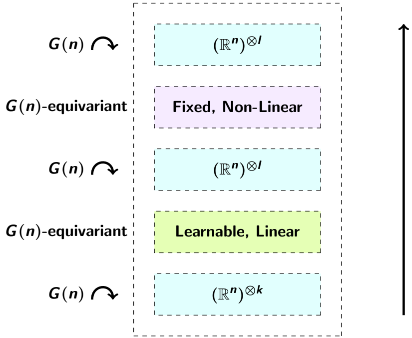

view the symmetries formally as **groups**



create neural network architectures that **take advantage** of these symmetries when performing learning.

The image of the Cayley Table of $S_3$ is attributed to Shiraki: `10.1093/ietfec/e88-a.10.2734`

# Group Equivariant Neural Networks




**Layer spaces:**

$$G(n) \subseteq GL(n)$$

$$\rho_k : G(n) \to GL((\mathbb{R}^n)^{\otimes k})$$

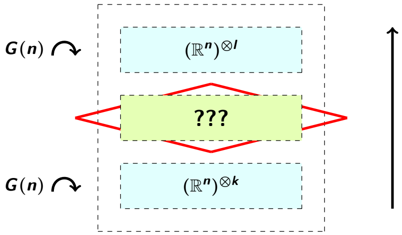
$$\rho_k(g)(v_1 \otimes \cdots \otimes v_k) := gv_1 \otimes \cdots \otimes gv_k$$

**Equivariance:**

$$\begin{array}{ccc} (\mathbb{R}^n)^{\otimes k} & \xrightarrow{\rho_k(g)} & (\mathbb{R}^n)^{\otimes k} \\ \downarrow \phi & & \downarrow \phi \\ (\mathbb{R}^n)^{\otimes l} & \xrightarrow{\rho_l(g)} & (\mathbb{R}^n)^{\otimes l} \end{array}$$

# Initial Research Question

**Question:** For different groups $G(n) \subseteq GL(n)$, what are the possible **weight matrices** that can appear in these neural networks?




**Goal:** To find a **basis** or **spanning set** of **matrices** for

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

# a) Symmetric Group $S_n$

Permutation equivariant neural networks have been studied by many authors:

1. **Zaheer et al.** (2017, arXiv:1703.06114): introduced the first permutation equivariant neural network, Deep Sets, for learning from sets in a permutation equivariant manner.
2. **Hartford et al.** (2018, arXiv:1803.02879): modelled interactions between different sets of objects using a permutation equivariant neural network.
3. **Maron et al.** (2019, arXiv:1812.09902): characterised all of the learnable, linear, equivariant layer functions when the layers are some tensor power of $\mathbb{R}^n$ for the symmetric group $S_n$ in the practical cases, by looking at fixed point equations representing the symmetric subspace.

# Notation

$[n]$ means the set of elements $\{1, \ldots, n\}$

$[n]^p$ is the $p$-fold Cartesian product set

$$\{1, \ldots, n\} \times \cdots \times \{1, \ldots, n\}$$

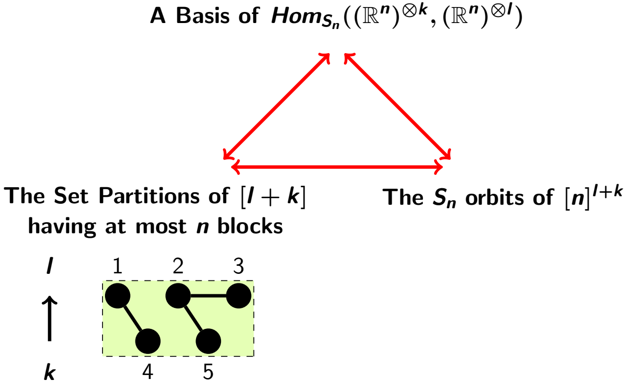
For example, if $n = 3$ and $p = 6$, then

$$(2, 1, 1, 3, 3, 2)$$

is an element of $[3]^6$.

# An Approach using Set Partitions

We showed that there exists a bijective correspondence between




Pearce-Crump (2022): Connecting Permutation Equivariant Neural Networks and Partition Diagrams, **arXiv:2212.08648**, to appear at ECAI 2024.

# Bijection between Basis and $S_n$ orbits

Any **linear** map $f : (\mathbb{R}^n)^{\otimes k} \to (\mathbb{R}^n)^{\otimes l}$ can be expressed in the basis of **matrix units** as

$$f = \sum_{I \in [n]^l} \sum_{J \in [n]^k} f_{I,J} E_{I,J}$$

We can show that $f$ is $S_n$-**equivariant** if and only if, for all $\sigma \in S_n$, $I \in [n]^l$ and $J \in [n]^k$,

$$f_{\sigma(I),\sigma(J)} = f_{I,J}$$

Defining an **action** of $S_n$ on $[n]^{l+k}$ by

$$\sigma(I, J) \coloneqq (\sigma(I), \sigma(J))$$

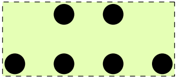
shows that the **orbits** of this action correspond **bijectively** with the **basis elements** of $\mathrm{Hom}_{S_n}((\mathbb{R}^n)^{\otimes k}, (\mathbb{R}^n)^{\otimes l})$. These orbits **partition** $[n]^{l+k}$ entirely.

Consider $n = 3$, $l = 2$ and $k = 4$, and let

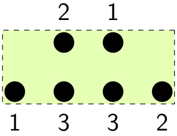
$$(I, J) = (2, 1 \mid 1, 3, 3, 2)$$

be the class representative of an orbit under $S_3 \curvearrowright [3]^{2+4}$.

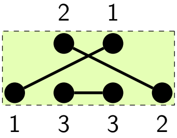
We obtain a $(k = 4, l = 2)$-set partition diagram having (at most) $n = 3$ blocks as follows:



Draw two rows of nodes




Label the nodes with the elements of $(I, J)$




Only connect nodes that have the same labels

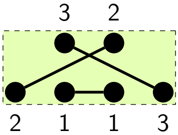
The correspondence is independent of the choice of class representative for the orbit: if instead of

$$(I, J) = (2, 1 \mid 1, 3, 3, 2)$$

we take

$$(I, J) = (3, 2 \mid 2, 1, 1, 3)$$

we obtain the same $(k = 4, l = 2)$-set partition diagram:




Also, each $(k, l)$-set partition diagram having at most $n$ blocks appears in the correspondence: calculate all possible labels to get the orbit!

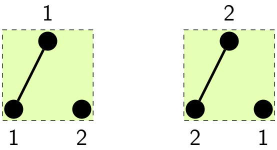
> The basis matrices correspond bijectively with all $(k, l)$-set partition diagrams having at most $n$ blocks!

# Example: $\mathrm{Hom}_{S_2}((\mathbb{R}^2)^{\otimes 2}, \mathbb{R}^2)$ ($n = 2, k = 2, l = 1$)

**Goal:** The weight matrix for an $S_2$-equivariant linear map from $(\mathbb{R}^2)^{\otimes 2}$ to $\mathbb{R}^2$.

**Step 1:** We need all ($k = 2, l = 1$)-set partition diagrams that have at most $n = 2$ blocks:

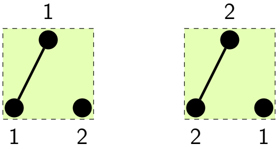
**Step 2:** For each diagram, calculate all possible labellings of the blocks of the diagram **without repeating labels**, and label each node with its block label. For the second diagram:



# Example: $\mathrm{Hom}_{S_2}((\mathbb{R}^2)^{\otimes 2}, \mathbb{R}^2)$ ($n = 2, k = 2, l = 1$)

**Step 3:** Convert these labels into $(I, J)$ pairs.

We see that




become

$$(1 \mid 1, 2) \quad \text{and} \quad (2 \mid 2, 1)$$

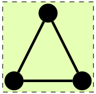
**Step 4:** For each diagram, add together the matrix units indexed by the $(I, J)$ pairs to form the associated basis matrix:

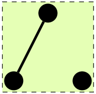
$$\begin{array}{c} \\ 1 \\ 2 \end{array}\begin{array}{cccc|cccc} & 1,1 & 1,2 & & 2,1 & 2,2 \\ \end{array}$$

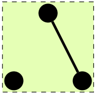
$$\begin{matrix} & {\scriptstyle 1,1} & {\scriptstyle 1,2} & {\scriptstyle 2,1} & {\scriptstyle 2,2} \\ {\scriptstyle 1} & 0 & 1 & 0 & 0 \\ {\scriptstyle 2} & 0 & 0 & 0 & 0 \end{matrix} + \begin{matrix} & {\scriptstyle 1,1} & {\scriptstyle 1,2} & {\scriptstyle 2,1} & {\scriptstyle 2,2} \\ {\scriptstyle 1} & 0 & 0 & 0 & 0 \\ {\scriptstyle 2} & 0 & 0 & 1 & 0 \end{matrix} = \begin{matrix} & {\scriptstyle 1,1} & {\scriptstyle 1,2} & {\scriptstyle 2,1} & {\scriptstyle 2,2} \\ {\scriptstyle 1} & 0 & 1 & 0 & 0 \\ {\scriptstyle 2} & 0 & 0 & 1 & 0 \end{matrix}$$

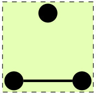
# Example: $\mathrm{Hom}_{S_2}((\mathbb{R}^2)^{\otimes 2}, \mathbb{R}^2)$ ($n = 2, k = 2, l = 1$)

Hence we obtain the following basis matrices:









$$\begin{array}{c} \\ 1 \\ 2 \end{array} \begin{array}{cccc} 1,1 & 1,2 & 2,1 & 2,2 \\ \end{array} \qquad \begin{array}{cccc} 1,1 & 1,2 & 2,1 & 2,2 \\ \end{array} \qquad \begin{array}{cccc} 1,1 & 1,2 & 2,1 & 2,2 \\ \end{array} \qquad \begin{array}{cccc} 1,1 & 1,2 & 2,1 & 2,2 \\ \end{array}$$

$$\begin{array}{c} 1 \\ 2 \end{array} \left[\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right] \quad \begin{array}{c} 1 \\ 2 \end{array} \left[\begin{array}{cc|cc} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{array}\right] \quad \begin{array}{c} 1 \\ 2 \end{array} \left[\begin{array}{cc|cc} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{array}\right] \quad \begin{array}{c} 1 \\ 2 \end{array} \left[\begin{array}{cc|cc} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{array}\right]$$

and so the weight matrix for an $S_2$-equivariant linear map from $(\mathbb{R}^2)^{\otimes 2}$ to $\mathbb{R}^2$ is of the form:

$$\begin{array}{c} \\ 1 \\ 2 \end{array} \begin{array}{c} \begin{array}{cccc} 1,1 & 1,2 & 2,1 & 2,2 \end{array} \\ \left[\begin{array}{cc|cc} w_1 & w_2 & w_3 & w_4 \\ w_4 & w_3 & w_2 & w_1 \end{array}\right] \end{array}$$

for weights $w_1, w_2, w_3, w_4 \in \mathbb{R}$.

# b) Orthogonal Group $O(n)$

Neural networks that are equivariant to other groups have been studied by a few authors:

1. **Finzi et al.** (2021, arXiv:2104.09459): constructed a **numerical algorithm** to find the weight matrices for three groups:
    - the orthogonal group $O(n)$,
    - the special orthogonal group $SO(n)$, and
    - the symplectic group $Sp(n)$

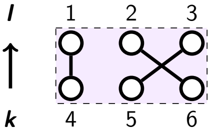
    Issue: their algorithm only works for small values of $n, k$ and $l$ as it is **constrained by memory** on larger values.
2. **Villar et al.** (2021, arXiv:2106.06610): characterised $O(n)$, $SO(n)$, and $Sp(n)$ invariant scalar functions $(\mathbb{R}^n)^{\otimes k} \to \mathbb{R}$ and equivariant vector functions $(\mathbb{R}^n)^{\otimes k} \to \mathbb{R}^n$ by **approximating with MLPs**.

We showed that there exists a bijective correspondence between

**A Spanning Set of $Hom_{O(n)}((\mathbb{R}^n)^{\otimes k}, (\mathbb{R}^n)^{\otimes l})$**

**The Set Partitions of $[l + k]$ whose blocks come in pairs**

**First Fundamental Theorem for $O(n)$ : A Spanning Set of $Hom_{O(n)}((\mathbb{R}^n)^{\otimes l+k}, \mathbb{R})$**




Pearce-Crump (2022): Brauer's Group Equivariant Neural Networks, **arXiv:2212.08630**, ICML 2023.

# Brauer's Invariant Argument

It can be shown that the linear map from $(\mathbb{R}^n)^{\otimes k}$ to $(\mathbb{R}^n)^{\otimes l}$

$$\sum_{I \in [n]^l, J \in [n]^k} c_{I,J} E_{I,J}$$

is **equivariant** to $O(n)$ if and only if the linear map from $(\mathbb{R}^n)^{\otimes (l+k)}$ to $\mathbb{R}$ which maps an element of the form

$$u(1) \otimes u(2) \otimes \cdots \otimes u(l) \otimes v(1) \otimes v(2) \otimes \cdots \otimes v(k)$$

to

$$\sum_{I \in [n]^l, J \in [n]^k} c_{I,J} \prod_{t=1}^{l} u_{i_t}(t) \prod_{r=1}^{k} v_{j_r}(r)$$

is **invariant** to $O(n)$.

# First Fundamental Theorem for $O(n)$

Suppose that $\mathbb{R}^n$ has associated with it a non-degenerate, symmetric, bilinear form $(\cdot,\cdot)$.

Pick the standard basis for $\mathbb{R}^n$, so that $(\cdot,\cdot)$ becomes the Euclidean inner product on $\mathbb{R}^n$.

If $f : (\mathbb{R}^n)^{\otimes(l+k)} \to \mathbb{R}$ is a polynomial function on elements in $(\mathbb{R}^n)^{\otimes(l+k)}$ of the form

$$u(1) \otimes u(2) \otimes \cdots \otimes u(l) \otimes v(1) \otimes v(2) \otimes \cdots \otimes v(k)$$

that is $O(n)$-invariant, then $f$ must be a polynomial of the Euclidean inner products

$$(u(i), u(j)), \quad (u(i), v(j)), \quad (v(i), v(j))$$

Hence, from Brauer's Invariant Argument, we get that

**Theorem: Spanning Set of Invariants $(\mathbb{R}^n)^{\otimes(l+k)} \to \mathbb{R}$ for $O(n)$**

The functions

$$(z(1), z(2))(z(3), z(4)) \ldots (z(l+k-1), z(l+k))$$

where $z(1), \ldots, z(l+k)$ is a permutation of

$$u(1), u(2), \ldots, u(l), v(1), v(2), \ldots, v(k)$$

form a spanning set of invariants $(\mathbb{R}^n)^{\otimes(l+k)} \to \mathbb{R}$ for $O(n)$.

# Example: $\mathrm{End}_{O(2)}((\mathbb{R}^2)^{\otimes 2})$ $(n = 2, k = 2, l = 2)$

**Goal:** The weight matrix for an $O(2)$-equivariant linear map from $(\mathbb{R}^2)^{\otimes 2}$ to $(\mathbb{R}^2)^{\otimes 2}$.

One possible method is the following:

**Step 1:** Calculate all functions of the form

$$(z(1), z(2))(z(3), z(4))$$

where $z(1), \ldots, z(4)$ is a permutation of

$$u(1), u(2), v(1), v(2)$$

These are:

$$(u(1), u(2))(v(1), v(2)) \quad (u(1), v(1))(u(2), v(2)) \quad (u(1), v(2))(u(2), v(1))$$

**Step 2:** Expand each product in the standard basis:

For example, for $(u(1), u(2))(v(1), v(2))$, this is

$$= [u_1(1)u_1(2) + u_2(1)u_2(2)][v_1(1)v_1(2) + v_2(1)v_2(2)]$$

$$= u_1(1)u_1(2)v_1(1)v_1(2) + u_1(1)u_1(2)v_2(1)v_2(2)$$
$$+ u_2(1)u_2(2)v_1(1)v_1(2) + u_2(1)u_2(2)v_2(1)v_2(2)$$

**Step 3:** Identify the coefficients $c_{I,J}$ from the invariant and reverse Brauer's Invariant Argument to obtain the spanning set matrix:

For the above: $c_{1,1|1,1} = 1$, $c_{1,1|2,2} = 1$, $c_{2,2|1,1} = 1$, $c_{2,2|2,2} = 1$, and all other coefficients are 0.

Hence the spanning set matrix is

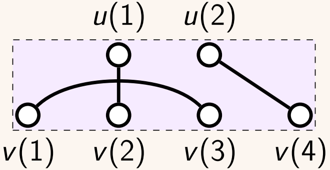
$$\begin{array}{c} \\ 1,1 \\ 1,2 \\ 2,1 \\ 2,2 \end{array} \begin{array}{c} \begin{array}{cccc} 1,1 & 1,2 & 2,1 & 2,2 \end{array} \\ \left[\begin{array}{cc|cc} 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{array}\right] \end{array}$$

**Key idea**: each function can be presented in the form of a $(k, l)$-**Brauer diagram**:

For example, if $l = 2$ and $k = 4$, then the function

$$(u(1), v(2))(u(2), v(4))(v(1), v(3))$$

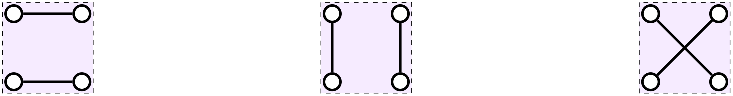
corresponds to the diagram




The set of all $(k, l)$-**Brauer diagrams** determines the spanning set matrices for $\mathrm{Hom}_{O(n)}((\mathbb{R}^n)^{\otimes k}, (\mathbb{R}^n)^{\otimes l})$!

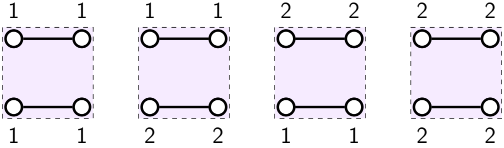
# Example: $\mathrm{End}_{O(2)}((\mathbb{R}^2)^{\otimes 2})$ $(n = 2, k = 2, l = 2)$

**Step 1:** We need all $(k = 2, l = 2)$–Brauer diagrams with $k = 2$ nodes at the bottom and $l = 2$ nodes at the top.



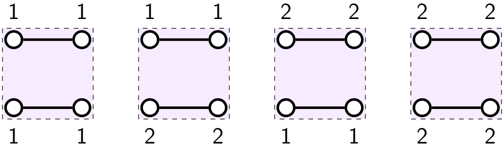
**Step 2:** For each diagram, calculate all possible labellings of the blocks of the diagram, where this time **we allow repeated block labels**, and label each node with its block label:

For example, for the first diagram, we have



**Step 3:** Convert these labels into $(I, J)$ pairs.

We see that



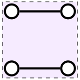
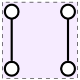
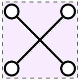

becomes

$$(1,1 \mid 1,1), \quad (1,1 \mid 2,2), \quad (2,2 \mid 1,1), \quad (2,2 \mid 2,2)$$

**Step 4:** Add together the matrix units indexed by these $(I, J)$ pairs to obtain the spanning set matrix:

| | 1,1 | 1,2 | 2,1 | 2,2 |
|---|---|---|---|---|
| 1,1 | 1 | 0 | 0 | 1 |
| 1,2 | 0 | 0 | 0 | 0 |
| 2,1 | 0 | 0 | 0 | 0 |
| 2,2 | 1 | 0 | 0 | 1 |

# Example: $\mathrm{End}_{O(2)}((\mathbb{R}^2)^{\otimes 2})$ $(n = 2, k = 2, l = 2)$

| Set Partition Diagram | Inner Products | Spanning Set Element |
|---|---|---|
|  | $(u(1), u(2))(v(1), v(2))$ | $\begin{array}{ccccc} & 1,1 & 1,2 & 2,1 & 2,2 \\ 1,1 & 1 & 0 & 0 & 1 \\ 1,2 & 0 & 0 & 0 & 0 \\ 2,1 & 0 & 0 & 0 & 0 \\ 2,2 & 1 & 0 & 0 & 1 \end{array}$ |
|  | $(u(1), v(1))(u(2), v(2))$ | $\begin{array}{ccccc} & 1,1 & 1,2 & 2,1 & 2,2 \\ 1,1 & 1 & 0 & 0 & 0 \\ 1,2 & 0 & 1 & 0 & 0 \\ 2,1 & 0 & 0 & 1 & 0 \\ 2,2 & 0 & 0 & 0 & 1 \end{array}$ |
|  | $(u(1), v(2))(u(2), v(1))$ | $\begin{array}{ccccc} & 1,1 & 1,2 & 2,1 & 2,2 \\ 1,1 & 1 & 0 & 0 & 0 \\ 1,2 & 0 & 0 & 1 & 0 \\ 2,1 & 0 & 1 & 0 & 0 \\ 2,2 & 0 & 0 & 0 & 1 \end{array}$ |

# Example: $\mathrm{Hom}_{O(2)}((\mathbb{R}^2)^{\otimes 2}, (\mathbb{R}^2)^{\otimes 2})$ $(n = 2, k = 2, l = 2)$

Hence the **weight matrix** for an $O(2)$–equivariant linear map from $(\mathbb{R}^2)^{\otimes 2}$ to $(\mathbb{R}^2)^{\otimes 2}$ must have the form

$$
\begin{array}{c} \\ 1,1 \\ 1,2 \\ 2,1 \\ 2,2 \end{array}
\begin{array}{c}
\begin{array}{cccc} 1,1 & 1,2 & 2,1 & 2,2 \end{array} \\
\left[\begin{array}{cc|cc}
w_{1,2,3} & 0 & 0 & w_1 \\
0 & w_2 & w_3 & 0 \\
\hline
0 & w_3 & w_2 & 0 \\
w_1 & 0 & 0 & w_{1,2,3}
\end{array}\right]
\end{array}
$$

where $w_{1,2,3} := w_1 + w_2 + w_3$.

# Diagrammatic Mathematics

Recall the main result for $O(n)$:

The set of all $(k, l)$-Brauer diagrams determines the spanning set matrices for $\text{Hom}_{O(n)}((\mathbb{R}^n)^{\otimes k}, (\mathbb{R}^n)^{\otimes l})$.

Recall the main result for $S_n$:

The set of all $(k, l)$-partition diagrams having at most $n$ blocks determines the basis matrices for $\text{Hom}_{S_n}((\mathbb{R}^n)^{\otimes k}, (\mathbb{R}^n)^{\otimes l})$.

Notice the following:

For each group $G(n)$, both the diagrams associated with $G(n)$ and the $\text{Hom}_{G(n)}$ spaces are very similar for any pair of values $k$ and $l$.

# Category Theory

**Idea**: form categories for both diagrams and representations!

Let us focus on $O(n)$ for now.

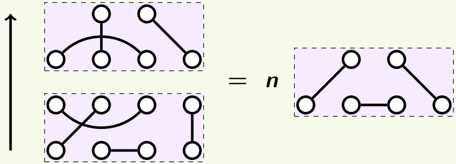
For the diagrams:

We can define a category $\mathcal{B}(n)$, called the **Brauer category**, to be:

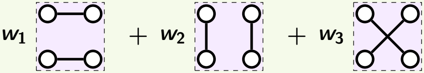
- **Objects**: $0, 1, 2, \ldots$
- **Morphisms**: for objects $k, l$, these are the set of all $(k, l)$-Brauer diagrams

Pearce-Crump (2023): Categorification of Group Equivariant Neural Networks, **arXiv:2304.14144**.

We need to provide a definition for the composition of any two morphisms.




We can also turn $\mathcal{B}(n)$ into an $\mathbb{R}$-**linear** category by allowing, for objects $k, l$, all possible formal linear combinations of the morphisms.



For the representations:

We can define the **tensor power representation category** $\mathcal{C}(O(n))$ to be:

- **Objects**: $\mathbb{R}, \mathbb{R}^n, (\mathbb{R}^n)^{\otimes 2}, \ldots$
- **Morphisms**: for objects $(\mathbb{R}^n)^{\otimes k}, (\mathbb{R}^n)^{\otimes l}$, this is the vector space $\mathrm{Hom}_{O(n)}((\mathbb{R}^n)^{\otimes k}, (\mathbb{R}^n)^{\otimes l})$
- **Composition**: is given by the usual composition of linear maps

**Q: Why do we care about this additional structure?**

Previously, for each $k, l$, we constructed a surjective linear map from

- the vector space spanned by the set of all $(k, l)$-Brauer diagrams to
- $\mathrm{Hom}_{O(n)}((\mathbb{R}^n)^{\otimes k}, (\mathbb{R}^n)^{\otimes l})$

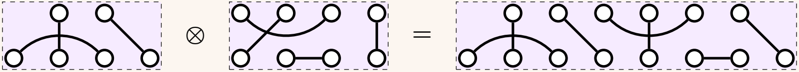
by mapping each $(k, l)$-Brauer diagram to a spanning set matrix and extending linearly.

We can now turn these linear maps into a full, $\mathbb{R}$-linear **functor** from $\mathcal{B}(n)$ to $\mathcal{C}(O(n))$:

- **Objects**: $k \mapsto (\mathbb{R}^n)^{\otimes k}$
- **Morphisms**: $(k, l)$-Brauer diagram $\mapsto$ spanning set matrix
- It can be shown that the functor respects the composition of morphisms!

But most importantly, there is **another composition operation** that we can define for each of these categories.

For morphisms in $\mathcal{B}(n)$:



For morphisms in $\mathcal{C}(O(n))$:

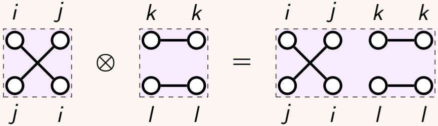
For any two $O(n)$-equivariant matrices, define the composition operation to be their tensor product (Kronecker product)!

This turns $\mathcal{B}(n)$ and $\mathcal{C}(O(n))$ into **strict monoidal categories**.

Crucially:

One can show that the functor from $\mathcal{B}(n)$ to $\mathcal{C}(O(n))$ becomes a **monoidal functor**!

**Informal argument**: the labelling of the blocks in each separate diagram in a tensor product is unaffected when the diagrams are horizontally concatenated, and vice versa!




$$E_{(i,j|j,i)} \otimes E_{(k,k|l,l)} = E_{(i,j,k,k|j,i,l,l)}$$

Note that this is true even if none, some or all of the values $1 \leq i, j, k, l \leq n$ are the same!

We can look to do the same thing for the symmetric group $S_n$:

For the diagrams:

We can define a category $\mathcal{P}(n)$, called the **Partition category**, to be:

- **Objects**: $0, 1, 2, \ldots$
- **Morphisms**: for objects $k, l$, this is the vector space given by the formal linear span of all $(k, l)$-partition diagrams
- **Vertical Composition**: Similar to Brauer category
- **Horizontal Composition**: Similar to Brauer category

For the representations:

We can define the **tensor power representation category** $\mathcal{C}(S_n)$ to be:

- **Objects**: $\mathbb{R}, \mathbb{R}^n, (\mathbb{R}^n)^{\otimes 2}, \ldots$
- **Morphisms**: for objects $(\mathbb{R}^n)^{\otimes k}, (\mathbb{R}^n)^{\otimes l}$, this is the vector space $\mathrm{Hom}_{S_n}((\mathbb{R}^n)^{\otimes k}, (\mathbb{R}^n)^{\otimes l})$
- **Vertical Composition**: is given by the usual composition of linear maps
- **Horizontal Composition**: tensor product of linear maps

But we hit a problem!

Although, for each $k, l$, the surjective linear map from

- the vector space spanned by the set of all $(k, l)$-partition diagrams to
- $\text{Hom}_{S_n}((\mathbb{R}^n)^{\otimes k}, (\mathbb{R}^n)^{\otimes l})$

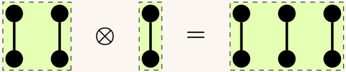
that is given by mapping

- each $(k, l)$-partition diagram having at most $n$ blocks to a basis matrix
- and all others to the zero matrix

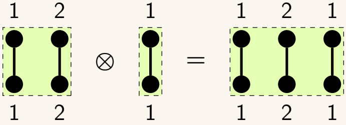
and extending linearly, can be turned into a functor from $\mathcal{P}(n)$ to $\mathcal{C}(S_n)$,

This functor is **not monoidal**!

Imagine we are looking at $S_3$-equivariance, and we consider:



Each of these diagrams, considered separately, is valid. But now consider the following labels:




Under the functor, the LHS becomes

$$E_{(1,2|1,2)} \otimes E_{(1|1)}$$

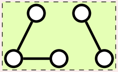
which equals

$$E_{(1,2,1|1,2,1)}$$

But the RHS under the functor becomes the zero matrix, as each block in the right hand diagram needs to be labelled with different values!

**New Basis:** Godfrey et al. (2023, arXiv:2303.06208) came up with a different basis for $\text{Hom}_{S_n}((\mathbb{R}^n)^{\otimes k}, (\mathbb{R}^n)^{\otimes l})$ that happens to be a fix for our monoidal functor problem!

**Idea:** Allow repeating block labels in a $(k, l)$–set partition diagram. We now draw these diagrams with white nodes to be consistent!



**Claim**:

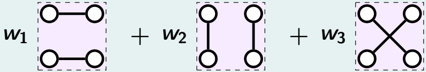
- The image of the set of all $(k, l)$–set partition diagrams having at most $n$ blocks under this labelling scheme still forms a basis of $\text{Hom}_{S_n}((\mathbb{R}^n)^{\otimes k}, (\mathbb{R}^n)^{\otimes l})$, and
- The resulting functor we get, from $\mathcal{P}(n)$ to $\mathcal{C}(S_n)$, is monoidal!

## Implications for Equivariant Neural Networks

- Under the **monoidal** functor, we can think of **diagrams as being the same as matrices**, and so each weight matrix can be thought of as being equivalent to a formal linear combination of diagrams: for $O(2)$, the weight matrix

| | 1,1 | 1,2 | 2,1 | 2,2 |
|---|---|---|---|---|
| 1,1 | $w_{1,2,3}$ | 0 | 0 | $w_1$ |
| 1,2 | 0 | $w_2$ | $w_3$ | 0 |
| 2,1 | 0 | $w_3$ | $w_2$ | 0 |
| 2,2 | $w_1$ | 0 | 0 | $w_{1,2,3}$ |

  is the same as




- Planar (non-crossing) diagrams correspond to Kronecker products of matrices!

# Application: Algorithm for Computing with Weight Matrices

Suppose that we would like to perform

$$Wv = y$$

where

- $W$ is an $O(n)$ (or $S_n$)-equivariant weight matrix
- $v$ is an input vector in $(\mathbb{R}^n)^{\otimes k}$, and
- $y$ is an output vector in $(\mathbb{R}^n)^{\otimes l}$.

A naive matrix multiplication implementation would take $O(n^{l+k})$ time.

**Q: Can we do any better?**

We want to use our observations to obtain a faster implementation!

Initially, we could look to

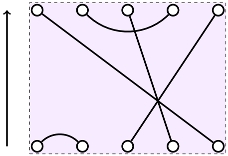
1. Express the weight matrix as a linear combination of diagrams.
2. Apply each diagram to the input vector in parallel.

But we haven't really done much at this stage!

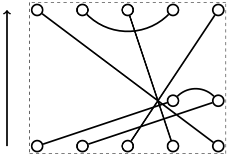
**Q: What is the best way to apply each diagram to the input vector?**

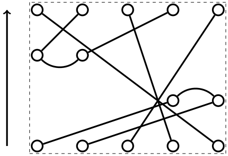
Pearce-Crump (2023): An Algorithm for Computing with Brauer's Group Equivariant Neural Network Layers **arXiv:arXiv:2304.14165**.

Imagine that one of the diagrams in our decomposition of an $O(3)$-equivariant weight matrix from $(\mathbb{R}^3)^{\otimes 5}$ to $(\mathbb{R}^3)^{\otimes 5}$ is

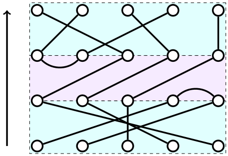


We can deform the diagram, as follows:



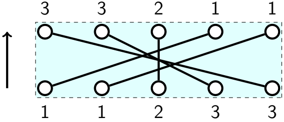


And with the final deformation we obtain a factoring of the original diagram into three new diagrams:



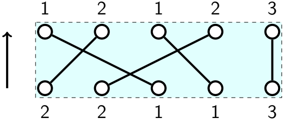
Each of these diagrams plays a unique role!

The bottom diagram is a permutation that changes how we access the coefficients in our original vector for input into the middle diagram. It can be viewed as having no cost!




For example, when we consider the coefficient at the $(3, 3, 2, 1, 1)$-entry of the vector that is the input to the middle diagram, it is the value $v_{1,1,2,3,3}$ in our original vector $v$ that is accessed.

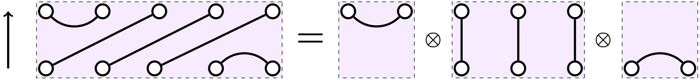
The top diagram is also a permutation that decides where the output from the middle diagram is stored in the final output vector. It can also be viewed as having no cost!




If $\alpha$ is the coefficient that is the output from the middle diagram in the $(2, 2, 1, 1, 3)$-entry, then it is stored in the final output vector at the $(1, 2, 1, 2, 3)$-entry.

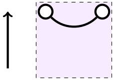
So we have reduced the calculation entirely to the middle diagram!

Moreover, the middle diagram is **planar**, so it can be decomposed horizontally:

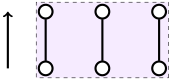


Under the monoidal functor, this becomes a Kronecker product of smaller matrices!

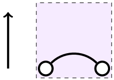
Each diagram performs a different function as a matrix:



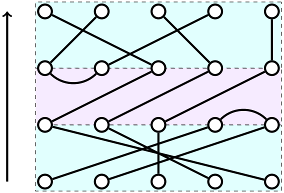
Copy Operation

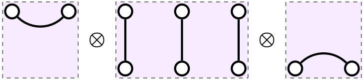


Identity Matrix



Tensor Contraction





**Step 0:** Input vector:

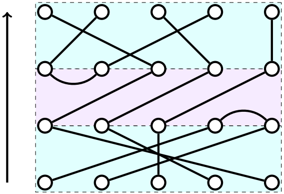
$$\sum_{l_1,l_2,l_3,l_4,l_5\in[3]} v_{l_1,l_2,l_3,l_4,l_5} e_{l_1,l_2,l_3,l_4,l_5}$$

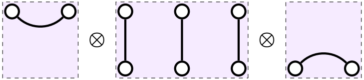
**Step 1:** Apply bottom permutation:

$$\sum_{l_1,l_2,l_3,l_4,l_5\in[3]} v_{l_1,l_2,l_3,l_4,l_5} e_{l_5,l_4,l_3,l_1,l_2}$$

**Step 2:** Apply tensor contraction:

$$\sum_{j\in[3]} \sum_{l_3,l_4,l_5\in[3]} v_{j,j,l_3,l_4,l_5} e_{l_5,l_4,l_3}$$





**Step 3:** Apply identity:

$$\sum_{j\in[3]}\sum_{l_3,l_4,l_5\in[3]} v_{j,j,l_3,l_4,l_5} e_{l_5,l_4,l_3}$$

**Step 4:** Apply copying:

$$\sum_{m\in[3]}\sum_{j\in[3]}\sum_{l_3,l_4,l_5\in[3]} v_{j,j,l_3,l_4,l_5} e_{m,m,l_5,l_4,l_3}$$

**Step 5:** Apply top permutation:

$$\sum_{m\in[3]}\sum_{j\in[3]}\sum_{l_3,l_4,l_5\in[3]} v_{j,j,l_3,l_4,l_5} e_{l_5,m,l_4,m,l_3}$$

This is the output vector!

# Time Complexity

By looking at the operations in the middle diagram:

For the orthogonal group $O(n)$, one can show that the time complexity is reduced from $O(n^{l+k})$ to $O(n^{k-1})$.

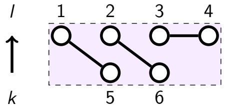
For the symmetric group $S_n$, one can show that the time complexity is reduced

- In the worst case, from $O(n^{l+k})$ to $O(n^k)$ (coming from a one vertex tensor contraction block)
- In the best case, the computation is free! (no tensor contraction blocks in the bottom row)

# Monoidal Diagram Categories and Equivariance

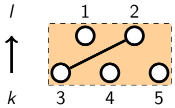
1. *Sp*($n$)**-equivariance:**

   Pearce-Crump (2023, arXiv:2212.08630): Brauer's Group Equivariant Neural Networks. ICML 2023.



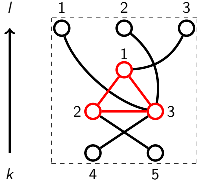

2. *SO*($n$)**-equivariance:**

   Pearce-Crump (2023, arXiv:2212.08630): Brauer's Group Equivariant Neural Networks. ICML 2023.




3. Aut($G$)**-equivariance,** $G$ **a graph:**

   Pearce-Crump and Knottenbelt (2023, arXiv:2307.07810): Graph Automorphism Group Equivariant Neural Networks. ICML 2024.



# Contact Details

**Edward Pearce-Crump**

Imperial College London

**e.pearcecrump@googlemail.com**

**ep1011@ic.ac.uk**

https://www.epearcecrump.co.uk