ICLR

Physics4ML Workshop

Neural Networks Learn Representation Theory: Reverse Engineering how Networks Perform Group Operations

Bilal Chughtai[1], Lawrence Chan[2], Neel Nanda[1]

[1]Independent, [2]UC Berkeley

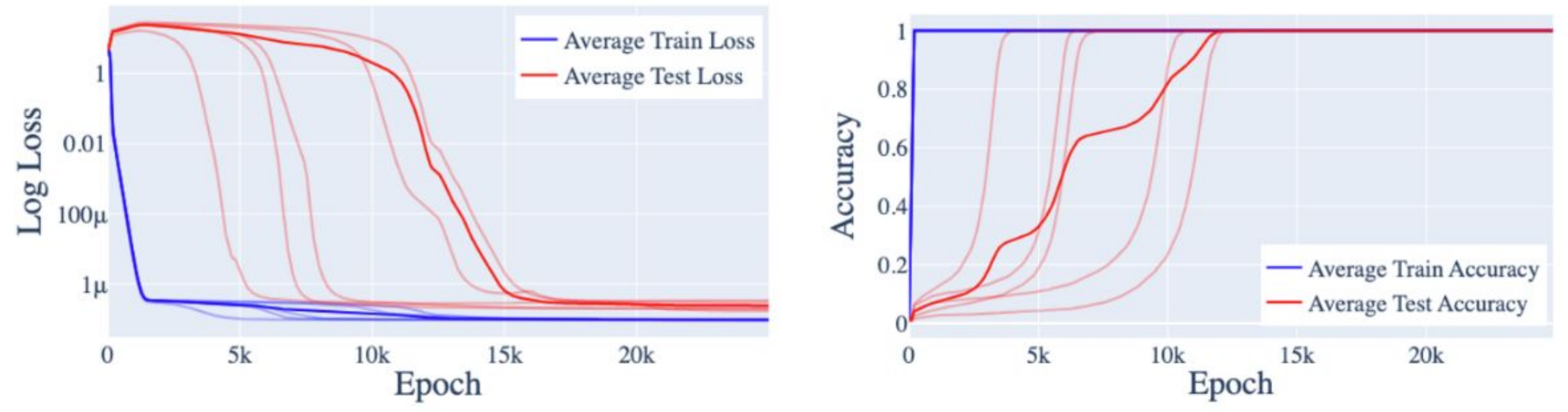
Stanford Existential Risks Initiative

Introduction

- We train models to perform **group composition**, demonstrating via **mechanistic interpretability** that networks consistently learn an **interpretable, representation theory** theoretic algorithm, across various different tasks (groups) and architectures.
- We use **progress measures** to track the development of this algorithm over training, and to understand grokking.
- We use this as an algorithmic test bed for the hypothesis of **universality** in mechanistic interpretability. We find convincing evidence for a form of weak universality, but against stronger forms.

Background

Grokking: Power et al. (2022) found that small models trained on algorithmic tasks such as modular addition, quickly memorised training data, and then after training for a much longer time suddenly generalise.

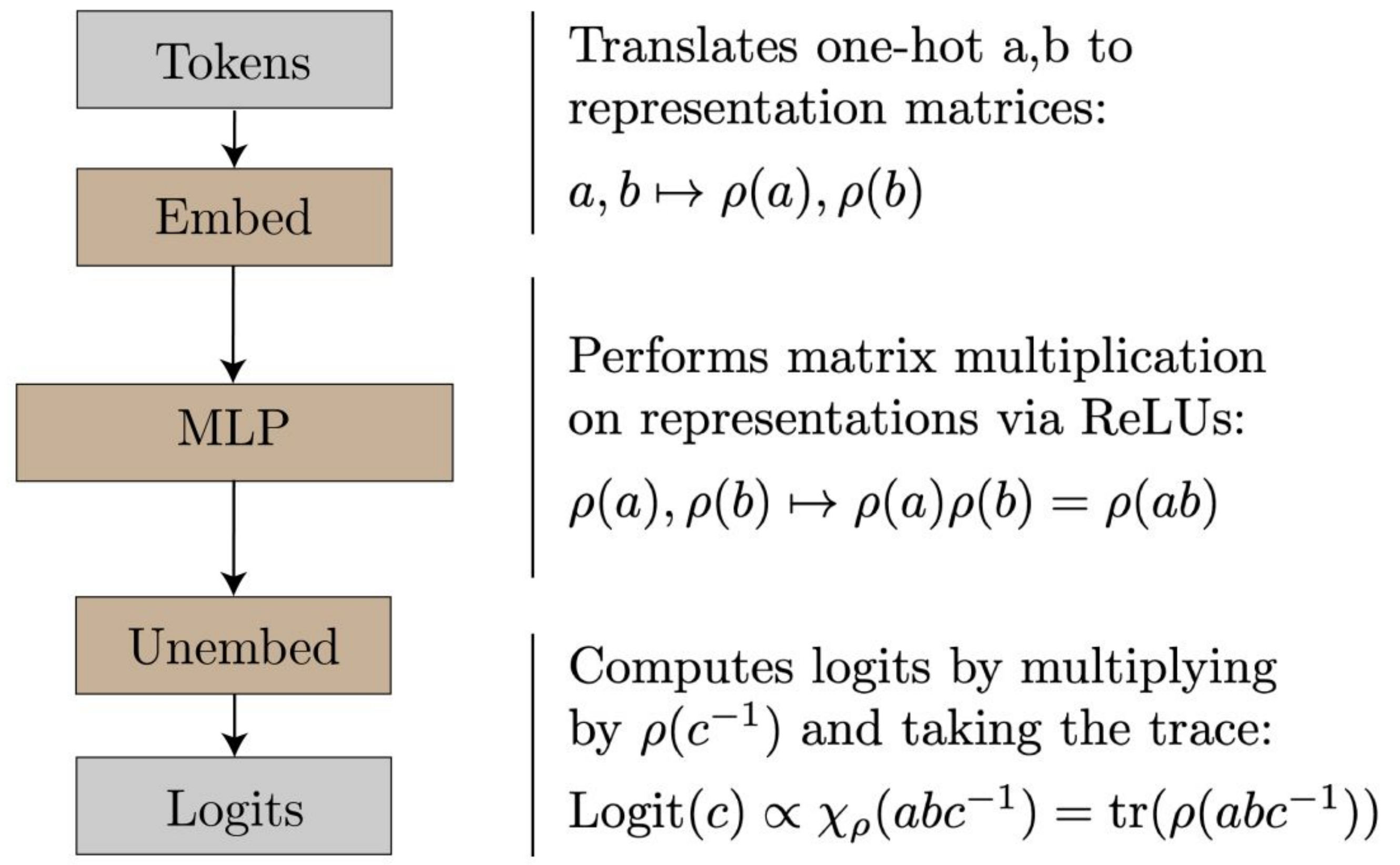


Mechanistic Interpretability (mech interp) is a sub-field attempting to reverse engineer neural networks. It claims neural networks are not an inscrutable mess, but learn human interpretable algorithms, which can be made legible through human effort. Our paper is heavily inspired by mech interp techniques.

Modular Addition: Nanda et al. (2023) were able to understand grokking by using mech interp to reverse engineer one layer Transformers trained to perform modular addition, finding a **fourier transform** and **trig identity** based algorithm.

Group Composition via Representations

- **Representation Theory** bridges group theory and linear algebra, and let's us think of group elements as matrices.
- Formally, a representation is a homomorphism $\rho : G \to GL(\mathbb{R}^d)$
- e.g. $C_n = \{r^k \mid r^n = e\}$ has a rep $\rho(r^k) = \begin{pmatrix} \cos \frac{2\pi k}{n} & -\sin \frac{2\pi k}{n} \\ \sin \frac{2\pi k}{n} & \cos \frac{2\pi k}{n} \end{pmatrix}$
- The task of modular addition reverse engineered by Nanda et al. (2023) is group composition on the cyclic group. We are able to directly generalise their algorithm to arbitrary groups using representation theory.



Translates one-hot a,b to representation matrices:

$a, b \mapsto \rho(a), \rho(b)$

Performs matrix multiplication on representations via ReLUs:

$\rho(a), \rho(b) \mapsto \rho(a)\rho(b) = \rho(ab)$

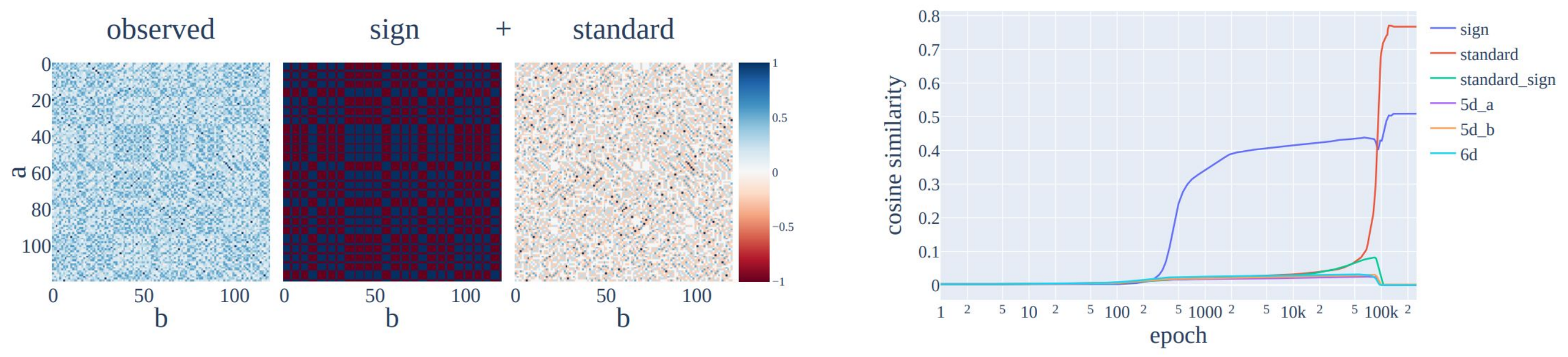
Computes logits by multiplying by $\rho(c^{-1})$ and taking the trace:

$\text{Logit}(c) \propto \chi_\rho(abc^{-1}) = \text{tr}(\rho(abc^{-1}))$

Reverse Engineering S_5

We first reverse engineer a network trained to perform S_5 composition, and find the GCR algorithm is learned, via four lines of evidence.

1. **Logit similarity** in **key representations.**

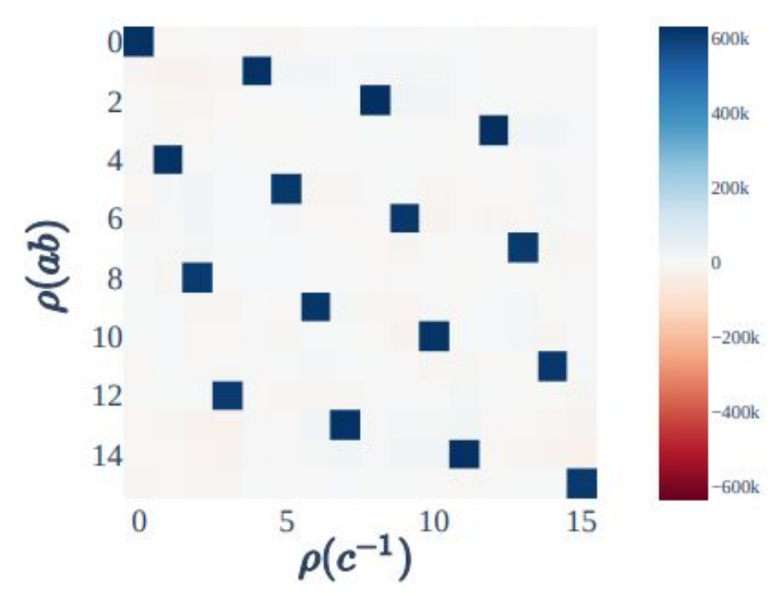


2. **Embeddings and unembeddings.**

	W_a	W_b	W_U
SIGN	6.95%	6.95%	9.58%
STANDARD	93.0%	93.0%	84.5%
RESIDUAL	0.00%	0.00%	5.96%

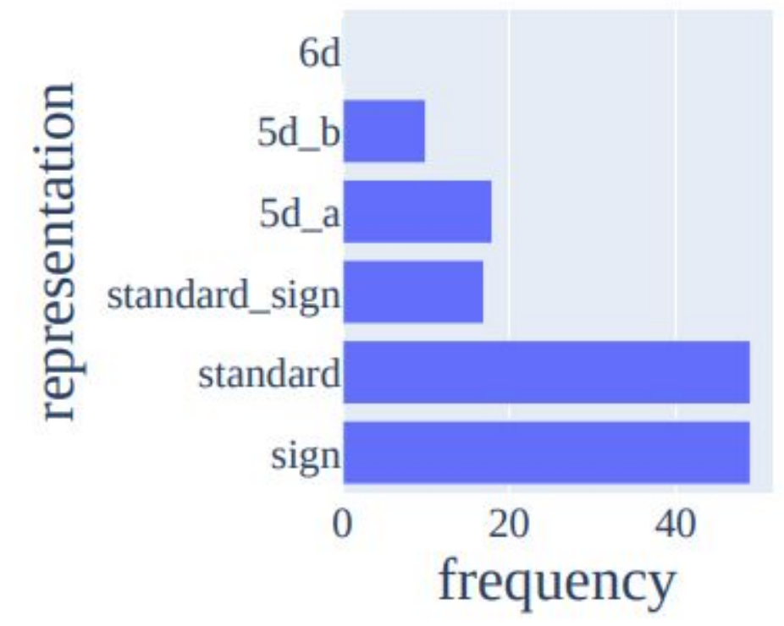
3. **MLP activations**, and the **map to logits**.

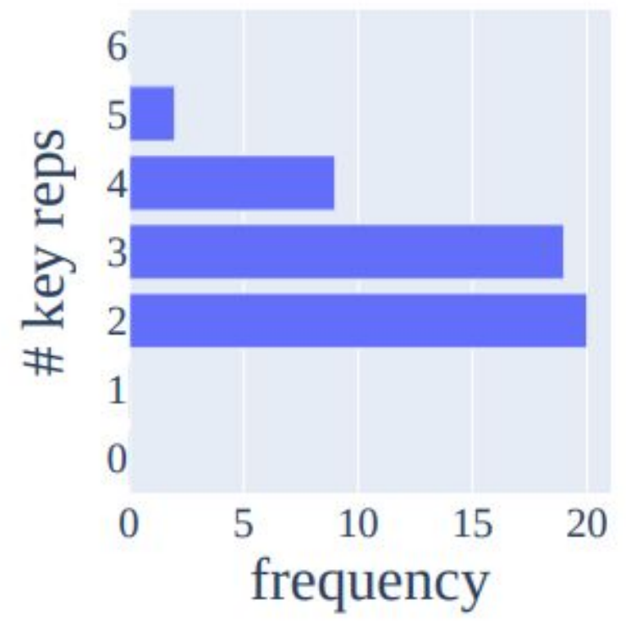
CLUSTER	$\rho(a)$	$\rho(b)$	$\rho(ab)$	RESIDUAL
SIGN	33.3%	33.3%	33.3%	0.00%
STANDARD	39.6%	37.1%	11.3%	12.1%

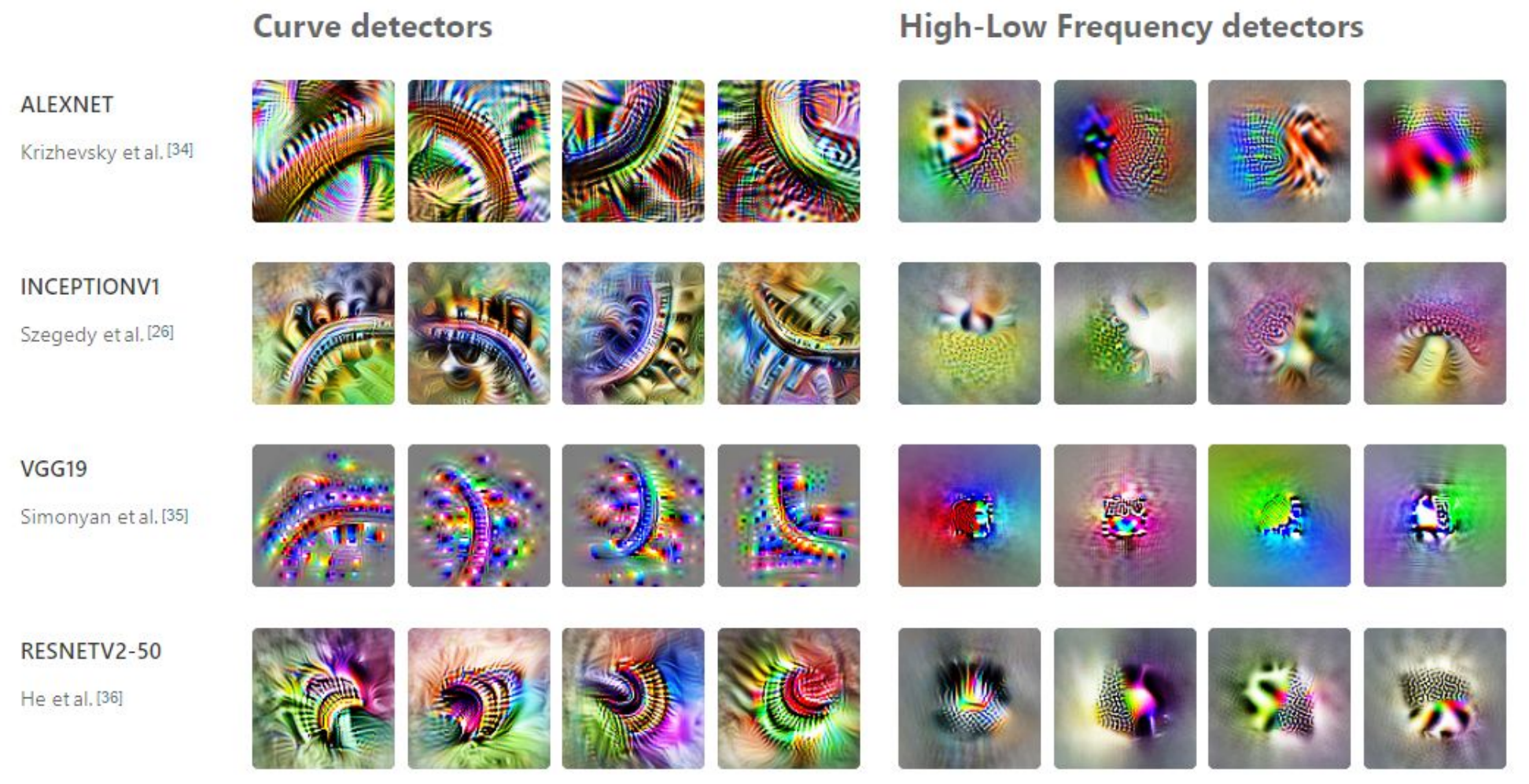


4. **Ablations**.

Universality







The universality hypothesis claims networks do not learn ad hoc and arbitrary algorithms, but canonical solutions, so **different models** will tend to learn similar features and circuits.

- Olah et al. (2020) demonstrated early layer neurons in vision models often learn similar features.
- We investigate universality **systematically,** by studying how networks solve the group composition task across different groups, random seeds, and architectures.
- We find networks **always** implement the GCR algorithm - convincing evidence for **weak universality**.
- However, the specific representations used vary, even if the architecture and data ordering is kept constant - evidence against **strong universality**.
- Interpreting a single network is insufficient to understand behaviour in general, **but** interpreting many networks may suffice give a periodic table of **universal features,** that in aggregate explain behaviour fully.