

Layerwise Development of Compositional Functional Representations Across Architectures

Jeyashree Krishnan and Ajay Mandyam Rangarajan

Apart Research AI Safety x Physics Grand Challenge Hackathon

The Challenge

AI Safety Question:

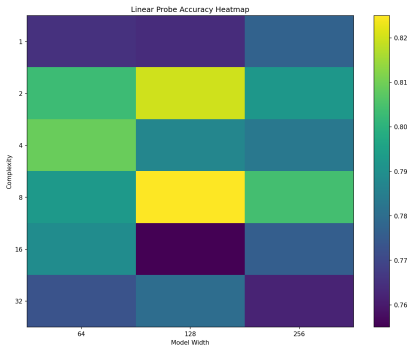
Can we explain where in a model
knowledge is stored?

Why does this matter?

- Trust depends on understanding
- Need to verify robust reasoning
- Critical for high-stakes domains

The Challenge: Complexity Scaling

How do models learn concepts?

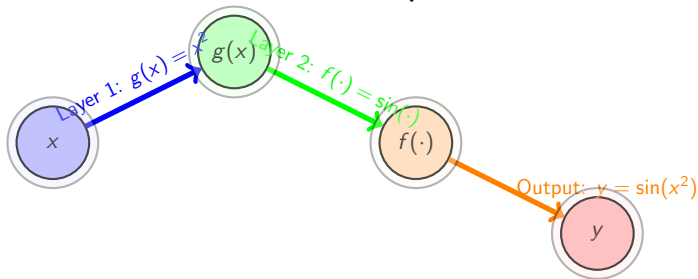


MLPs show gradual emergence across layers

Compositional Functions

What are compositional functions?

Modular Circuit Decomposition



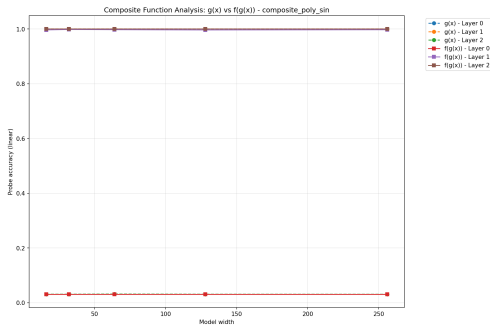
Example: $y = \sin(x^2)$ learned as $g(x) = x^2$ then $f(g) = \sin(g)$

Functions built from simpler components

Key Finding: MLPs

MLPs: Modular Decomposition

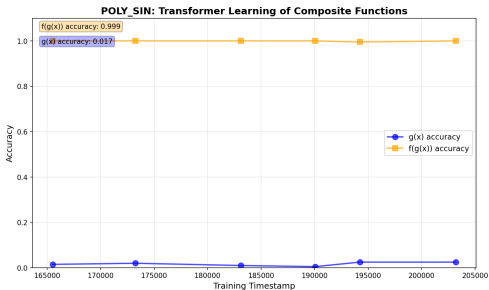
- Learn $g(x)$ first, then apply $f(\cdot)$
- Interpretable components



Key Finding: Transformers

Transformers: No Decomposition

- Perfect accuracy but no intermediate concepts
- Opaque mechanisms



Evidence: Quantitative Comparison

Model	Accuracy	Probe on $g(x)$	Modular
MLP	100%	High	Yes
Transformer	100%	Low ($< 5\%$)	No

Both achieve perfect accuracy, but only MLPs decompose

Future Work

Next Steps for AI Safety

- **Nonlinear Probes:** Beyond linear regression to uncover deeper structure
- **Real-world Tasks:** Physics simulations, program execution
- **Dynamic Probing:** Online monitoring during training
- **Safety Monitoring:** Alert systems for sudden capability acquisition
- **Scaling Laws:** How emergence scales with parameters and architecture

Goal: Build transparent, trustworthy, and governable AI systems

AI Safety Implications

The Problem:

- High performance $\neq$ Interpretability
- Transformers fail to decompose functions
- Opaque, potentially brittle mechanisms

The Solution:

- Architectural choices matter
- Need interpretable components
- Circuit emergence is architectural

Key Message: High performance doesn't guarantee interpretability.