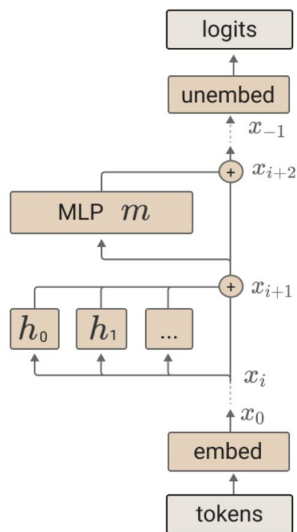


Interpretability of Language Models across Abstraction Levels

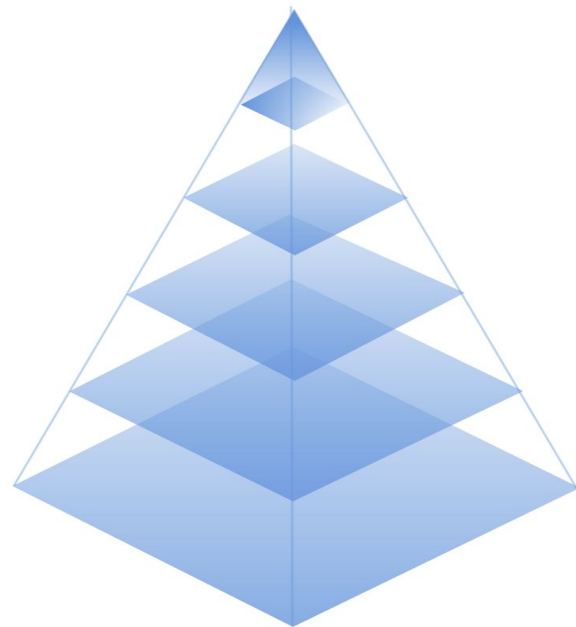


Oliver Eberle

Principal Researcher

TU Berlin / BIFOLD

Machine Learning Group



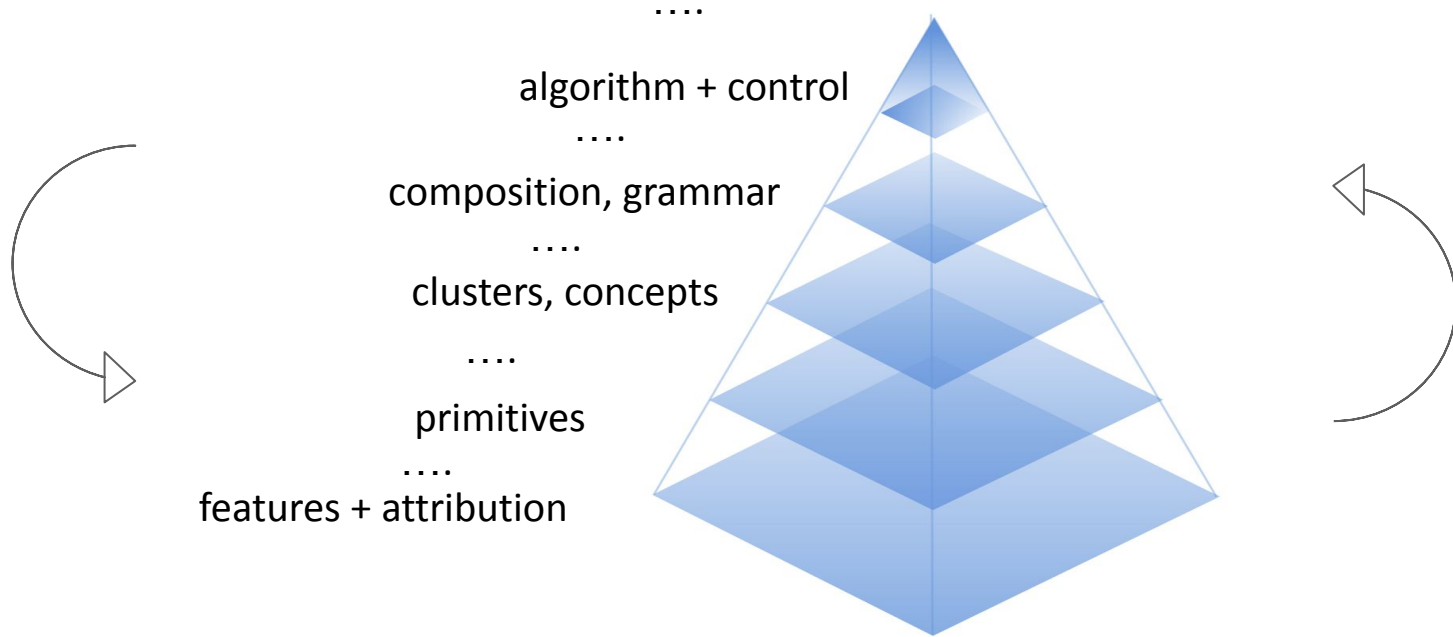
Impressive capabilities, opaque mechanisms

Status quo: Scale vs. Understanding

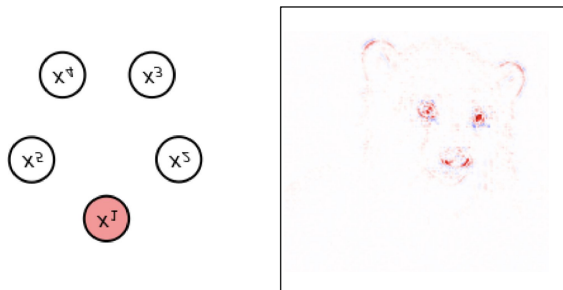
- The promise of the **Bitter Lesson** (Sutton, 2019) & **Reward is enough** (Silver et al., 2021):
 - scaling yields increasingly capable models
 - costly training
 - limits of available data and its quality
- **Scaling $\neq$ Understanding**
 - **No first-principles theory:** We cannot design algorithm-centric behavior from fundamentals.
 - Unknown **how gen AI implements problem solving**, i.e., how emergent algorithms are realized

Transparency, alignment, controllability, and efficiency suffer!

Going up – and down – the abstraction levels



XAI develops methods for input attribution



I.

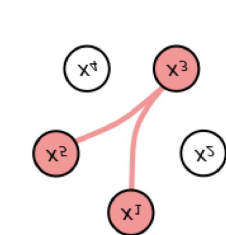
Attribution: *What features?*

2015

2020

2025

Input is not enough, extending to interactions and concepts



interactions



II.

Concepts & Interactions: *What groups emerge?*

Counterfactuals: *What would have to change?*

I.

Attribution: *What features?*

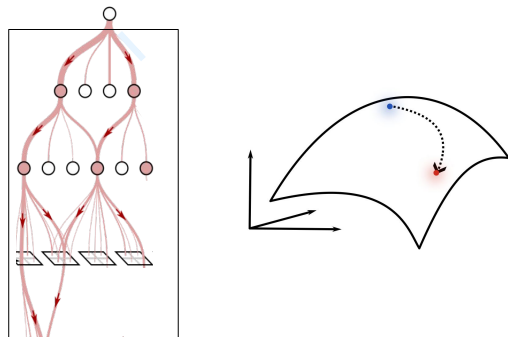
2015

2020

2025

Mechanistic interpretability to understand internal computations

III.



Interventions: *How to steer behavior?*

Circuits: *How do computations compose?*

II.

Concepts & Interactions: *What groups emerge?*

Counterfactuals: *What would have to change?*

I.

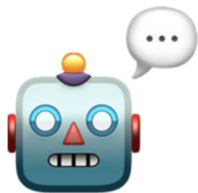
Attribution: *What features?*

2015

2020

2025

As models learned to talk, they started explaining themselves



Self-explanations: *How does the model explain itself?*

Interventions: *How to steer behavior?*

Circuits: *How do computations compose?*

Concepts & Interactions: *What groups emerge?*

Counterfactuals: *What would have to change?*

Attribution: *What features?*

2015

2020

2025

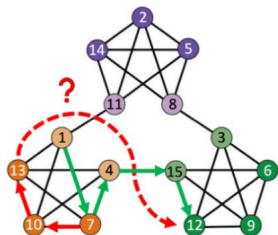
I. Algorithmic Evaluation and Understanding

Eberle et al., *Position: We Need An Algorithmic Understanding of Generative AI*. ICML Spotlight 2025

Lippl et al., *AlgoTrace: Algorithmic Primitives and Compositional Geometry of Reasoning in Language Models*. ICML 2026

Case Study: Graph-Based Multi-Step Reasoning, Goal-Directed Navigation

Motivation: Models perform poorly on graph navigation and multi-step planning tasks.



What is the shortest path from room 1 to room 12?

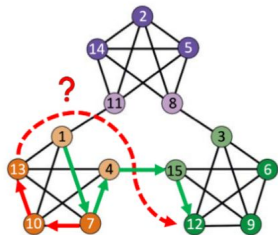
Answer: 1, 7, 10, 13, 12

Note: These tasks can be solved easily with existing algorithms; we would not need LLMs for it.

We are interested in them as it lets us study how these models learn to reason and try to solve structured problems; many interesting problems can be represented as inference from graphs.

Also we generate ground truth solution strategies for evaluation!

Case Study: Graph-Based Multi-Step Reasoning, Goal-Directed Navigation



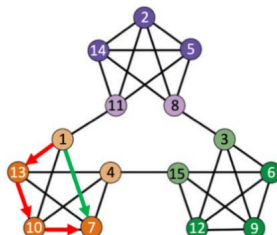
What is the shortest path from room 1 to room 12?

Answer: 1, 7, 10, 13, 12



GPT-4, temp=0, graph D, n-step

Path Hallucination



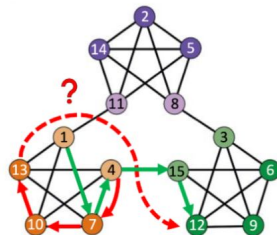
What is the shortest path from room 1 to room 7?

Answer: 1, 13, 10, 7



GPT-35, temp=1, graph D, 1-step

Detours



What is the shortest path from room 1 to room 12?

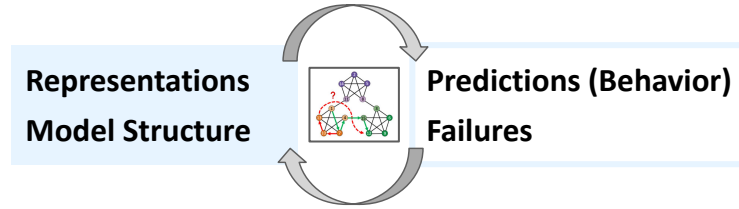
Answer: 1, 7, 4, 7, 13, 12



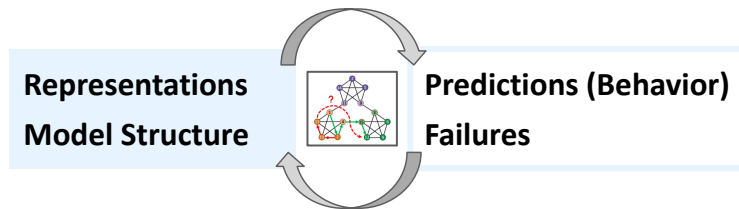
GPT-4, temp=0, graph D, n-step

Loops and
Hallucinations

Building algorithmic solutions: primitives, grammar, algorithms



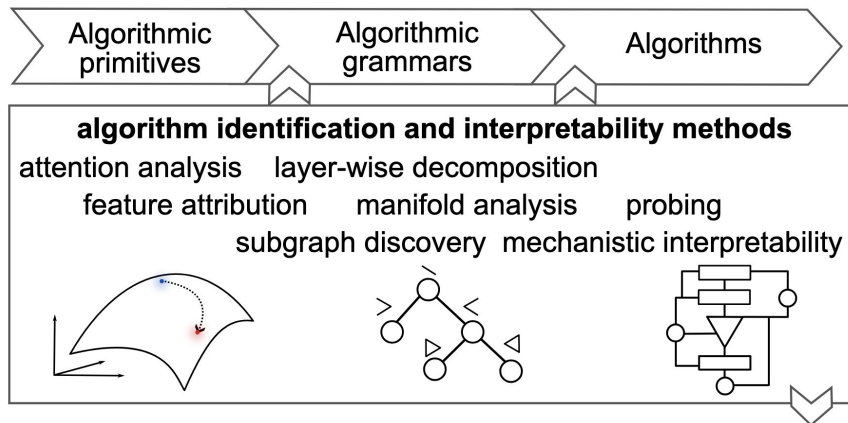
Building algorithmic solutions: primitives, grammar, algorithms



Basic operations necessary to implement an algorithm (vocabulary)

Compositions of primitives (grammar)

Finite set of rules for transforming inputs to outputs

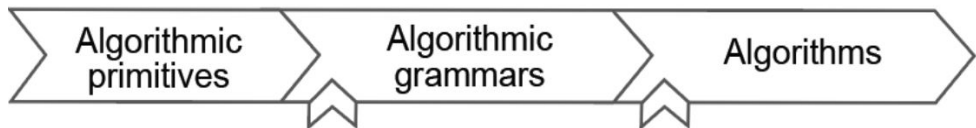


Case Study: Graph-Based Multi-Step Reasoning, Goal-Directed Navigation

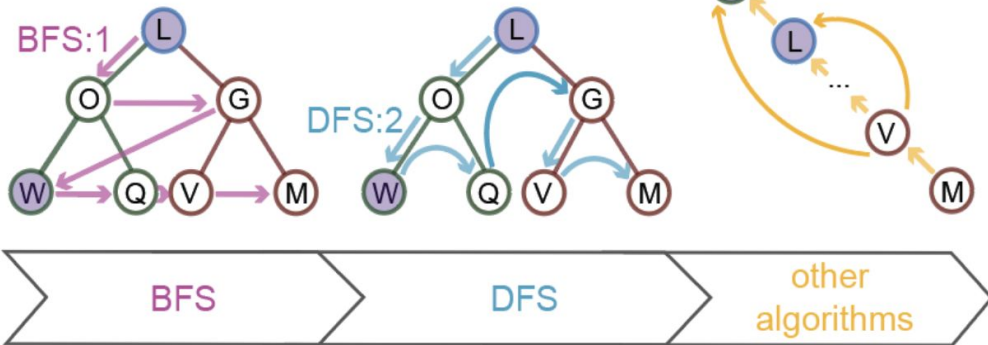
Setup:

- Llama-3.1-8/70B-Instruct
- Tree Graph (n=7)
- **Task:** Determine if there is a valid path from the lobby (L) to the goal node (here: W).

1. Algorithm Identification

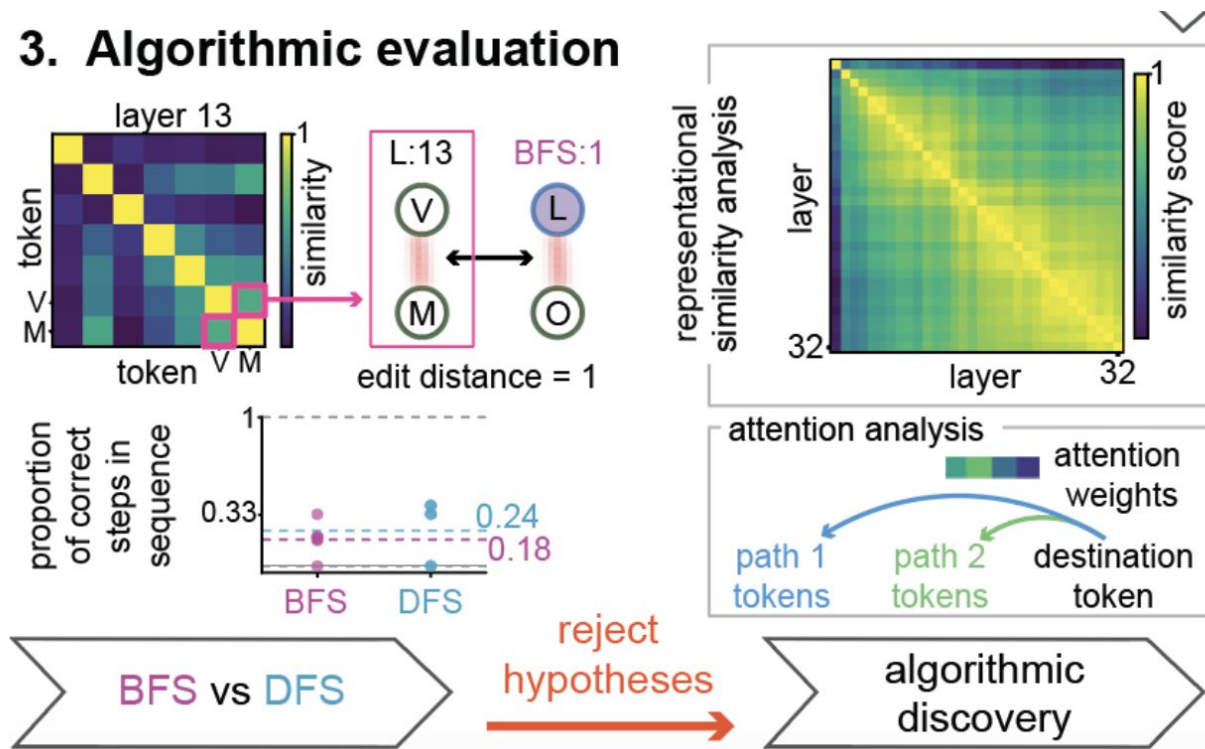


2. Hypotheses

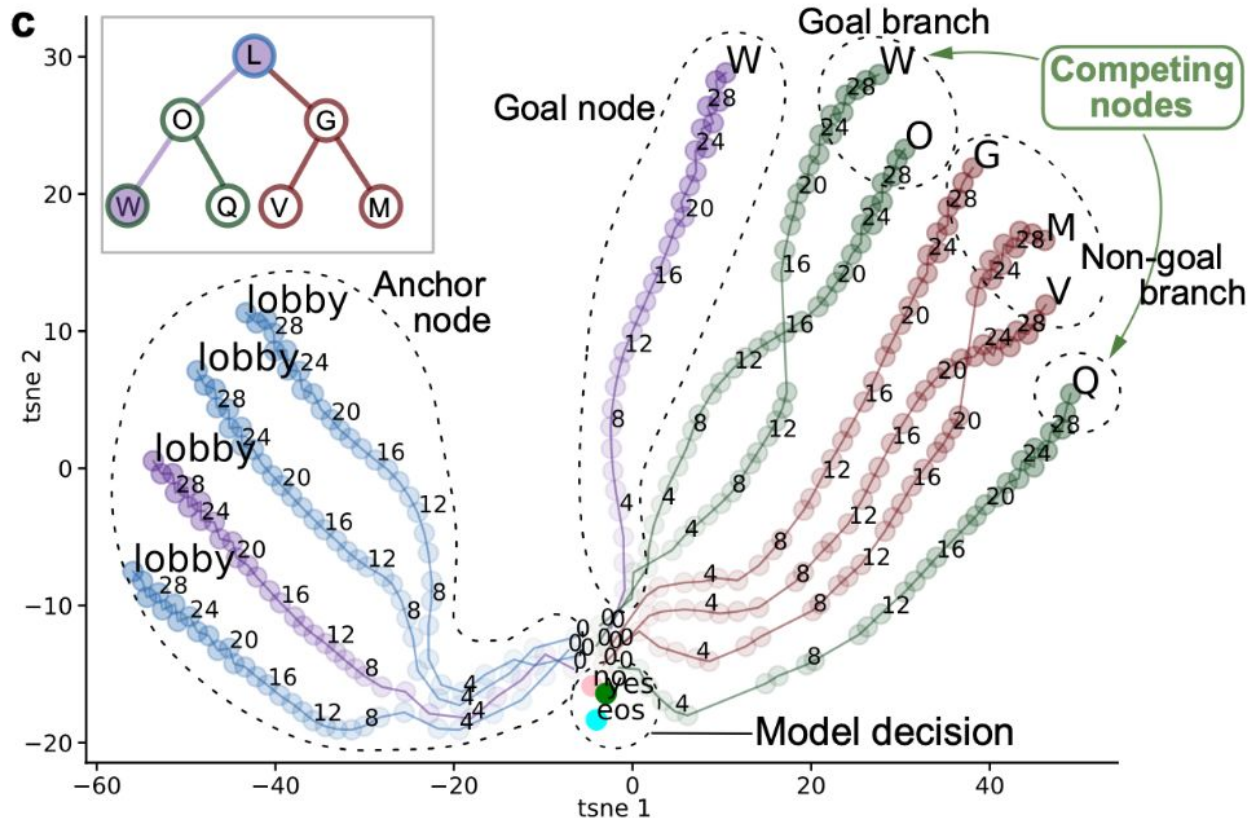


Evaluate Candidate Algorithms and Identify Implemented Algorithm

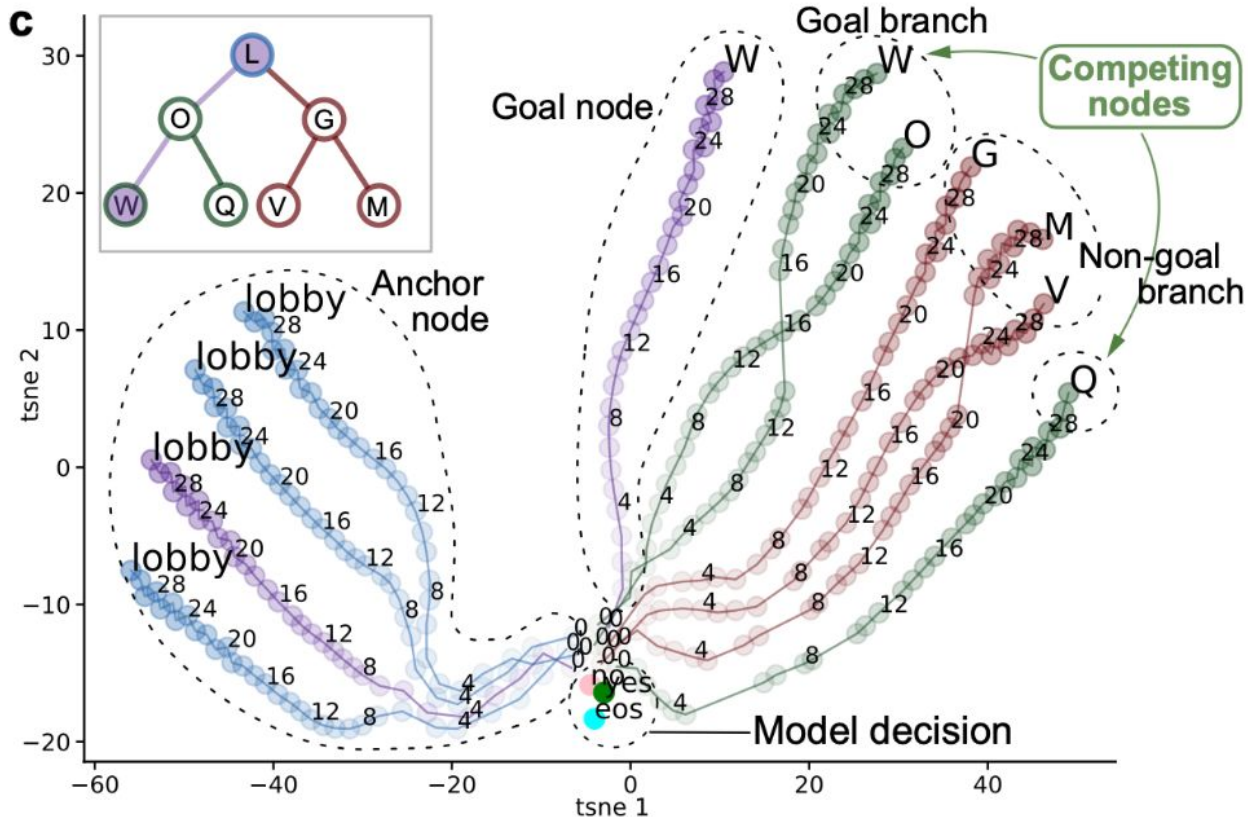
3. Algorithmic evaluation



Algorithm Discovery: Representational Analysis Reveals Policy-Dependent Search



Algorithm Discovery: Representational Analysis Reveals Policy-Dependent Search



Identified Algorithm

1. Incremental & competition-driven separation across layers
2. Leads to attention on goal token in later layers

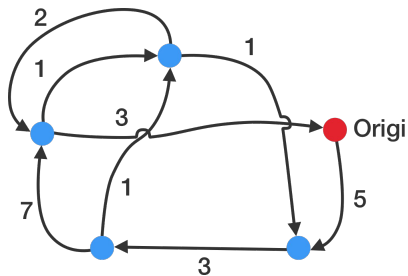


Policy-dependent search

instead of a classic algorithm

Decomposing LLM Reasoning in Algorithmic Primitives

Traveling Salesperson Problem (TSP)



*The distance between City 0 and City 2 is 8.
(...) The distance between City 2 and City 4 is 5. (...) What is the total distance of traveling from 0->2->4?*

chain-of-thought

0->2 has distance 8 and 2->4 has distance 5. The total distance is $8+5=13$.

Task capability

Compute total distance of a given path.

Algorithmic capability

Algorithmic primitives

Retrieve first node

Retrieve subsequent node

Retrieve edge attribute

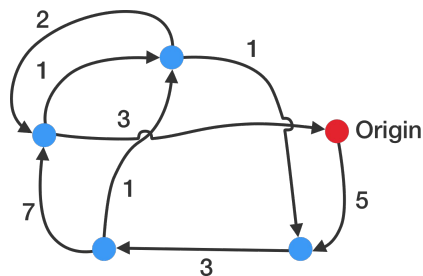
Compute sum

Algorithmic composition



CoT reveals distinct representational grouping

Traveling Salesperson Problem (TSP)



Cluster 0: Path initiation statements and recall statements.

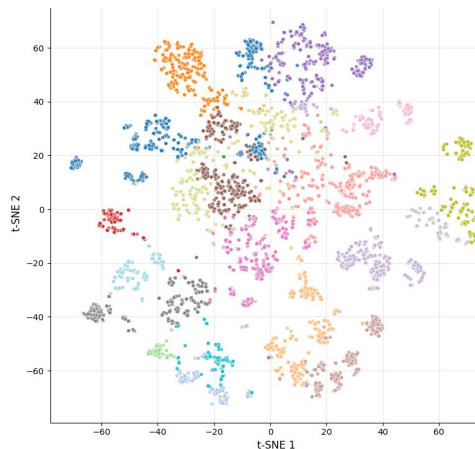
```
good . What about 0 -> 4 -> 2 -> 3 -> 1 -> 5 -> 0 . - 0 -> 4 = 14 , 4 -> 2 = 6 , total = 20 , 2 -> 3 = 6 , total = 26 , 3 -> 1 = 15 , total = 41 , 1 -> 5 = 14 , total = 55 , 5 -> 0 = 42 , total = 97 . What about 0 -> 4 -> 2 -> 5 -> 1 -> 3 -> 0 . - 0 -> 4 = 14 , 4 -> 2 = 6 , total = 20 , 2 -> 5 = 10 , total = 30 , 5 -> 1 = 14 , total = 44 , 1 -> 3 = 15 , total = 59 , 3 -> 0 = 19 , total = 78 . So total 78 , which is even better than 85 . Let 's check that route : 0 -> 4 = 14 , 4 -> 2 = 6 , 2 -> 5 = 10 , 5 -> 1 = 14 , 1 -> 3 = 15 , 3 -> 0 = 19 . Sum : 14 + 6 = 20 , 20 + 10 = 30 , 30 + 14 = 44 , 44 + 15 = 59 , 59 + 19 = 78 . Yes total = 78 . Now can we do even better ? Let 's try 0 -> 4 -> 2 -> 5 -> 3 -> 1 -> 0 . - 0 -> 4 = 14 , 4 -> 2 = 6 ( 20 ) , 2 -> 5 = 10 ( 30 ) , 5 -> 3 = 10 ( 40 ) , 3 -> 1 = 15 ( 55 ) , 1 -> 0 = 35 ( 90 ) . So total 90 . What
```

Cluster 3: Path representations.

```
> 1 -> 0 -> 3 -> 0 no . What about starting from 0 -> 4 -> 1 -> ... not sure . Maybe consider starting with 0 -> 4 -> 1 -> 3 -> 2 -> 5 -> 0 : - 0 -> 4 = 14 , 4 -> 1 = 17 , total = 31 , 1 -> 3 = 15 , total = 46 , 3 -> 2 = 6 , total = 52 , 2 -> 5 = 10 , total = 62 , 5 -> 0 = 42 , total = 104 . What about 0 -> 4 -> 1 -> 5 -> 2 -> 3 -> 0 : - 0 -> 4 = 14 , 4 -> 1 = 17 , total = 31 , 1 -> 5 = 14 , total = 45 , 5 -> 2 = 10 , total = 55 , 2 -> 3 = 6 , total = 61 , 3 -> 0 = 19 , total = 80 . So 80 is slightly higher than 78 . Maybe try 0 -> 4 -> 2 -> 5 -> 1 -> 0 -> 3 -> 0 is not valid . Maybe try
```

Cluster 4: Distance statements

```
cycle ) . We can try to solve TSP by enumerating possibilities or use intuition . Let 's list distances in matrix form : City 0 : - to 1 : 35 - to 2 : 33 - to 3 : 19 - to 4 : 14 - to 5 : 42 City 1 : - to 0 : 35 - to 2 : 25 - to 3 : 15 - to 4 : 17 - to 5 : 14 City 2 : - to 0 : 33 - to 1 : 25 - to 3 : 6 - to 4 : 6 - to 5 : 10 City 3 : - to 0 : 19 - to 1 : 15 - to 2 : 6 - to 4 : 44 - to 5 : 10 City 4 : - to 0 : 14 - to 1 : 17 - to 2 : 6 - to 3 : 44 - to 5 : 11 City 5 : - to 0 : 42 - to 1 : 14 - to 2 : 10 - to 3 : 10 - to 4 : 11 We can try to use nearest neighbor or use inspection . Let 's check potential candidate paths . Let 's try to find a cycle manually by trying to choose the shortest segments . Looks like the smallest edge is : between City 2 and City 3 : 6 , and also City 2 and City 4 : 6 . So
```



TSP: Identifying algorithmic reasoning strategies

Algorithmic composition



MC_00: Reasoning scaffold

is 14, 5 → 0 is 42. So best is 5 → 1 with 14. So pat
> 4 → 2 → 3 → 5 → 1, total: 36 + 14 = 50. Now at City
remaining is 0. So add edge: 1 → 0 is 35. Total = 50 +
total route: 0, 4, 2, 3, 5, 1, 0 with total = 85. Let's
find a better route manually. Let's try starting with 0 → 4 →
> 0, total 85. What if we try 0 → 4 → 2 → 1 → 5 → 3 → 0.
- 0 → 4 = 14 - 4 → 2 = 6, total = 20 - 2 → 1 = 25, t
> 5 = 14, total = 59 - 5 → 3 = 10, total = 69 - 3 → 0 =
So 88 > 85. What about 0 → 4 → 2 → 1 → 3 → 5 → 0. - 0 →
> 2 = 6, 2 → 1 = 25, total = 45, 1 → 3 = 15, total = 60,
total = 70, 5 → 0 = 42, total = 112. Not good. What about 0
> 5 → 0. - 0 → 4 = 14, 4 → 2 = 6, total = 20, 2 → 3 = 6
> 1 = 15, total = 41, 1 → 5 = 14, total = 55, 5 → 0 = 42, t
about 0 → 4 → 2 → 5 → 1 → 3 → 0. - 0 → 4 = 14, 4 → 2 =
2 → 5 = 10, total = 30, 5 → 1 = 14, total = 44, 1 → 3 = 15
> 0 = 19, total = 78. So total 78, which is even better than
that route: 0 → 4 = 14, 4 → 2 = 6, 2 → 5 = 10, 5 → 1 =
3 → 0 = 19. Sum: 14 + 6 = 20, 20 + 10 = 30, 30 + 14 = 44,
59 + 19 = 78. Yes total = 78. Now can we do even better? Let's

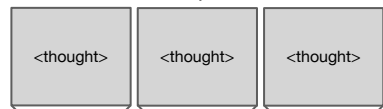
MC_06: Computing totals

> 0, total 85. What if we try 0 → 4
- 0 → 4 = 14, 4 → 2 = 6, total =
> 5 = 14, total = 59, 5 → 3 = 10, t
So 88 > 85. What about 0 → 4 → 2 → 1
> 2 = 6, 2 → 1 = 25, total = 45, 1 →
total = 70, 5 → 0 = 42, total = 112. No
> 5 → 0. - 0 → 4 = 14, 4 → 2 = 6,
> 1 = 15, total = 41, 1 → 5 = 14, total
about 0 → 4 → 2 → 5 → 1 → 3 → 0. -
2 → 5 = 10, total = 30, 5 → 1 = 14, t
> 0 = 19, total = 78. So total 78, whic
that route: 0 → 4 = 14, 4 → 2 = 6,
3 → 0 = 19. Sum: 14 + 6 = 20, 20 + 10
59 + 19 = 78. Yes total = 78. Now can we
> 2 → 5 → 3 → 1 → 0. - 0 → 4 = 14,
5 → 3 = 10 (40), 3 → 1 = 15 (55),
about 0 → 4 → 2 → 5 → 1 → 3 → 0. We

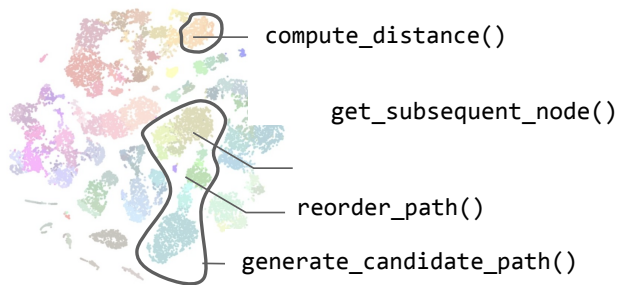
Code at <https://github.com/sflippl/algotracer>

Algorithmic understanding via top-down and bottom up interpretability

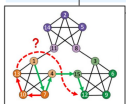
Top-down from behavior and reasoning traces



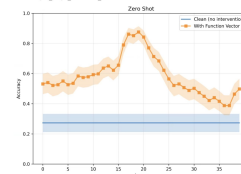
Algorithmic Primitive Clustering



Bottom-up from representations



Algorithmic compositions



Evaluation and Grounding

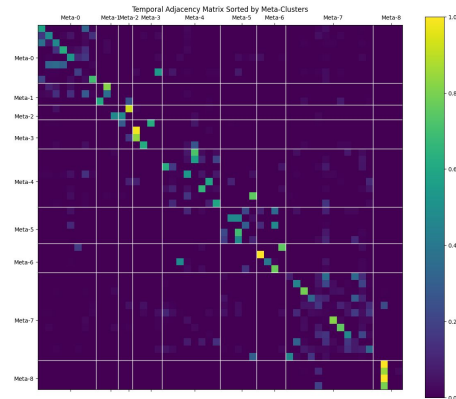
Hypothesis-Driven



Human Aligned

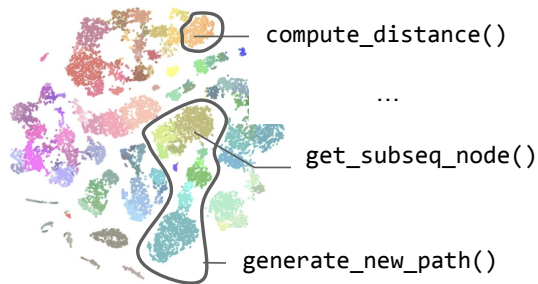


Analyzing reasoning at scale



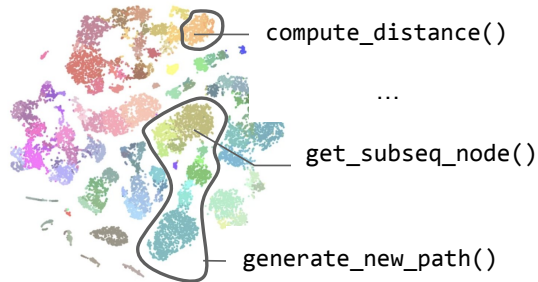
AlgoTrace: Algorithmic Understanding of Generative Processes

- a. Representational clusters map to
algorithmic primitives

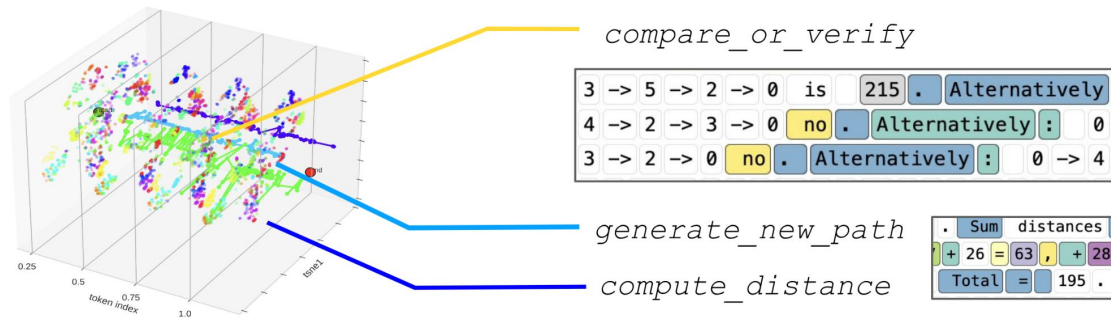


Latent Clusters Reveal Algorithmic Primitives

a. Representational clusters map to *algorithmic primitives*

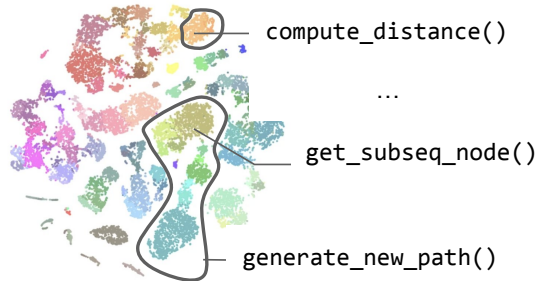


b. Compositions and Transitions

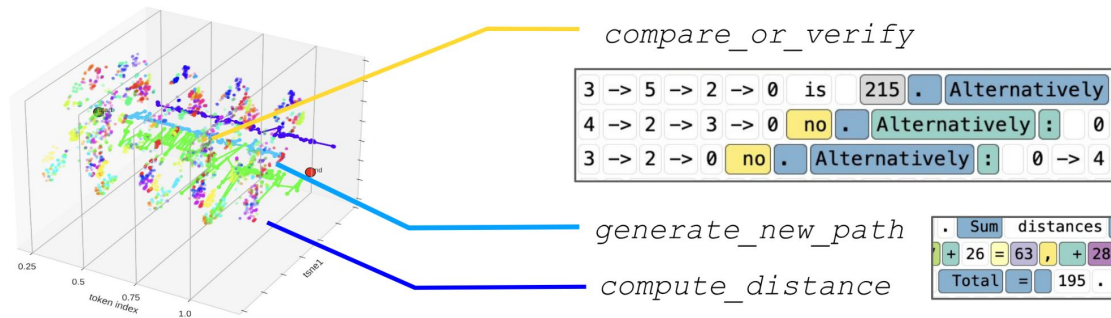


From Primitive Discovery to Causal Algorithmic Interventions

a. Representational clusters map to *algorithmic primitives*

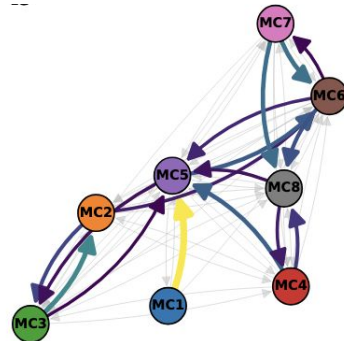


b. Compositions and Transitions



c. Control and intervene on process steps, then measure primitive expression in model output

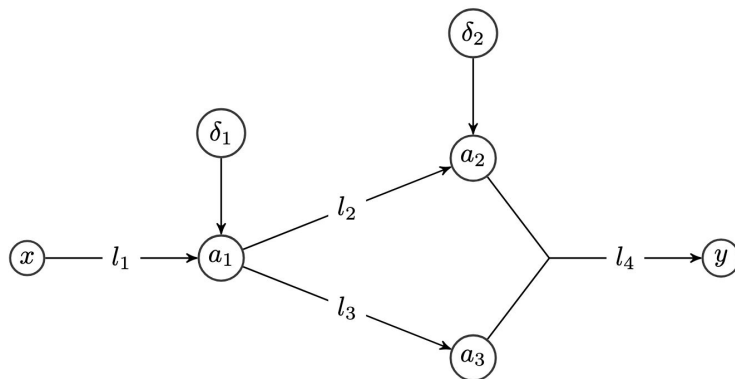
	%NN paths ↑	#Paths ↑	Dist. comp. ↓	Final ans. ↓	#Verif. ↑	#Comp. ↑
nearest_neighbor	+56.1%	+72.0%	-73.6%	-4.5%	+104.3%	+125.6%
generate_path	+4.8%	+143.9%	-76.0%	+18.4%	+25.0%	+12.2%
compute_distance	+22.2%	+36.5%	-86.5%	-47.4%	+198.9%	+79.3%
final_answer	+23.2%	+34.8%	-29.8%	-81.5%	+82.6%	-37.8%
compare_verify	+37.9%	+43.6%	-62.8%	-7.0%	+655.4%	+1103.7%
compare_verify2	+29.5%	+48.4%	-64.9%	-13.7%	+706.5%	+526.8%



II. Distributed and Sparse Interventions

Ernst et al., *Distributed Sparse Interventions in Language Models*. (under review, 2026)

Applying interventions leads to interaction effects

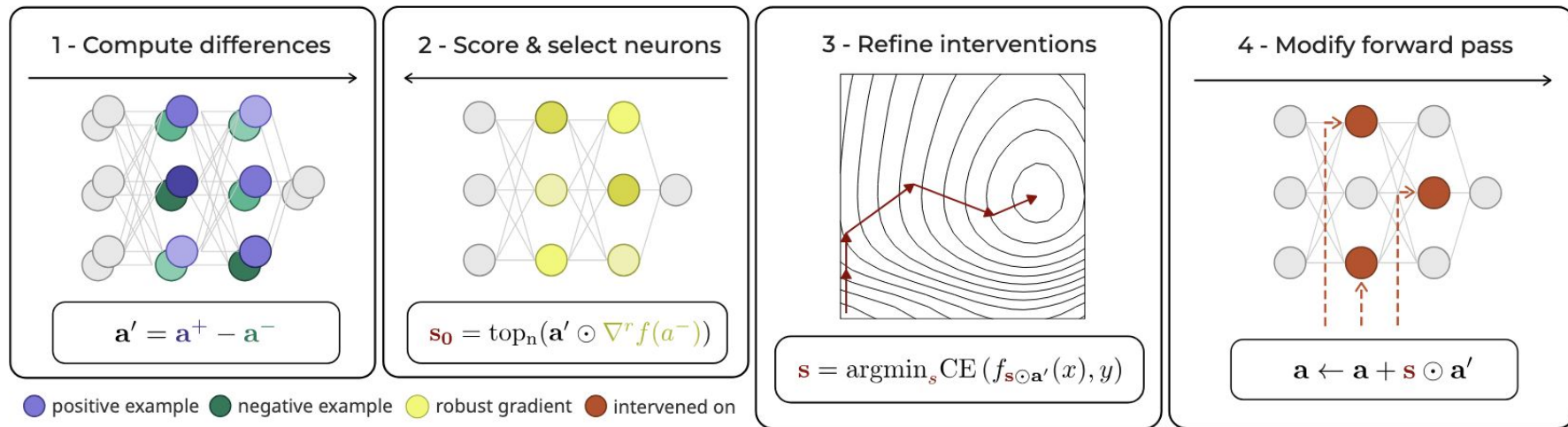


input x
layer l
activation a
output y
effect $\mathcal{E}$
intervention δ

$$\begin{aligned}\mathcal{E}(\delta_1, \delta_2) &= f(x|a_1 \leftarrow a_1 + \delta_1, a_2 \leftarrow a_2 + \delta_2) - f(x) \\ &= 2x\delta_1 + \delta_1^2 + x\delta_2 + \delta_1\delta_2 \\ &= \mathcal{E}(\delta_1) + \mathcal{E}(\delta_2) + \delta_1\delta_2.\end{aligned}$$

These interactions are typically not captured by classic intervention methods like function vectors as they rely on first-order approximations and linear patching.

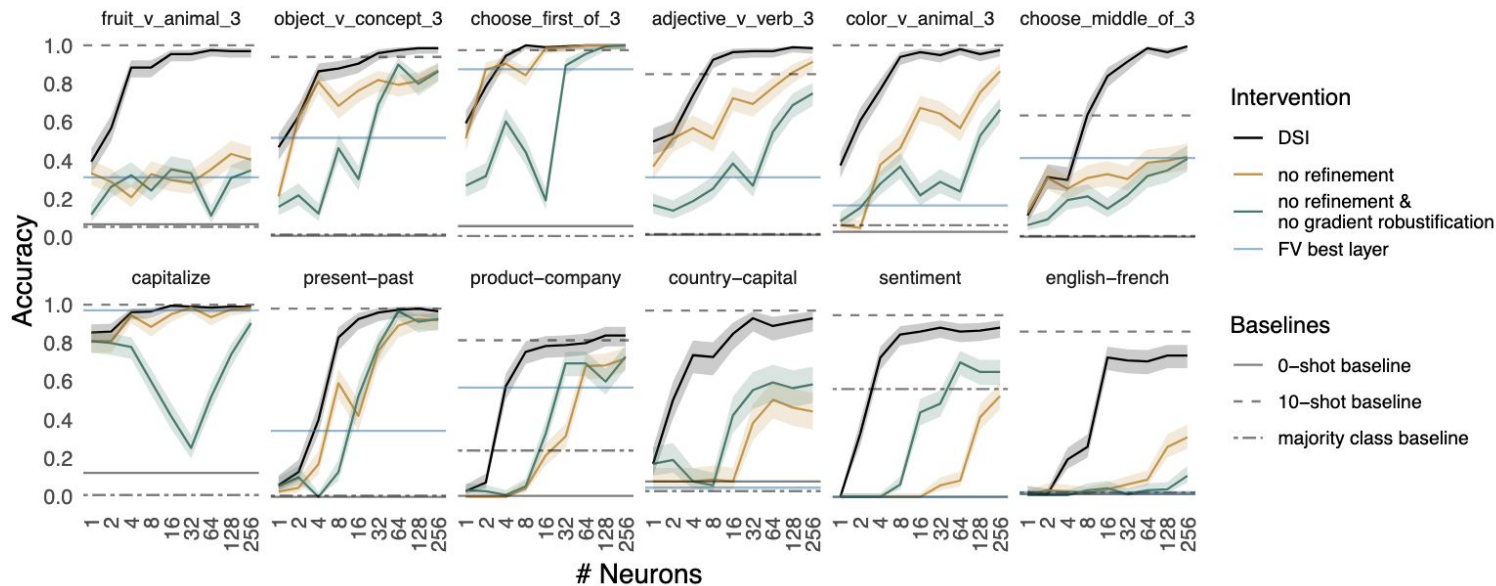
Distributed Sparse Interventions (DSI) mitigates this issue



DSI enables sparse steering

- (i) at the level of **individual neurons** instead of vector representations,
- (ii) does not require selecting a specific layer as it localizes neurons in a **distributed** manner,
- (iii) that can capture **interaction effects** across the entire model.

DSI improves over existing steering baselines, while intervening less



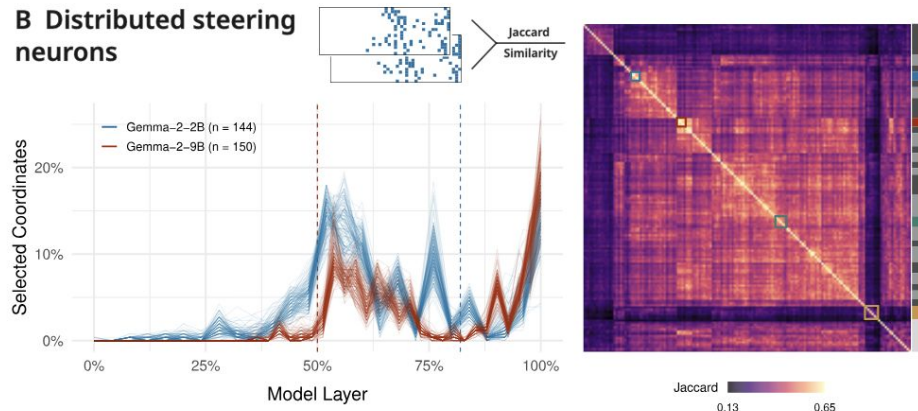
Improved steering performance of function vectors, at much lower intervention footprint.

DSI leads to interpretable neuron-level signatures of steering

A Steering performance on AxBench

model	DSI				Other Methods				
	64	256	1024	d model	$\dagger$ ReFT-r1	$\dagger$ DiffMean	$\dagger$ SAE	$\dagger$ PCA	$\dagger$ Probe
2B	0.34	0.51	0.60	0.64	0.51	0.18	0.15	0.08	0.09
9B	0.31	0.53	0.69	0.72	0.63	0.32	0.19	0.13	0.11

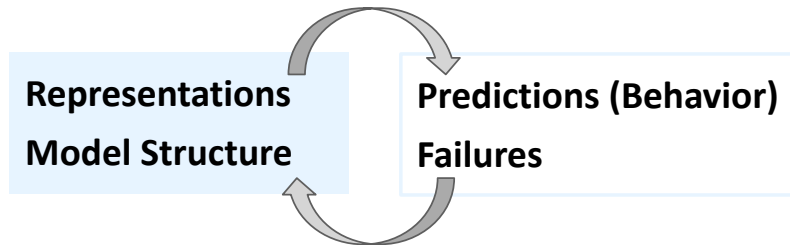
B Distributed steering neurons



C Shared neurons reveal related concepts

ID	Score	Concept
4	0.98	expressions of vulnerability and honesty about personal struggles
	0.94	expressions of personal experience and struggles related to judgment and stigma
	0.80	themes related to personal loss and health struggles
8	0.84	specific terms and phrases related to scientific protocols and methodologies
	1.16	phrases related to measurement processes and specific scientific terms
	1.22	technical terms and phrases related to scientific measurements and research methodologies
17	1.08	terms related to outreach and audience engagement
	1.18	expressions of commitment to collaboration and community responsibility
	0.92	terms related to social organizations and community involvement
26	1.54	specific punctuation or words that indicate enthusiasm or emphasis
	1.22	expressions of surprise or disbelief
	1.38	informal second-person pronouns and abbreviations commonly used in casual communication

Conclusion



A. What to explain?

- Decompositions into building blocks; guided by top-down/bottom-up label extraction

B. Bridging representations and structure?

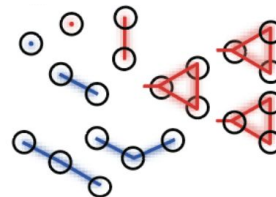
- Functional analysis of nodes, messages, and their compositions, methodological challenge!

C. Evaluate and ground explanations?

- Hypothesis-driven evaluation, match with known solutions or human processes, otherwise discovery!

From Scaling to Understanding

- Move from brute-force scaling → principled, algorithm-centric research
- Building a library of validated algorithmic primitives and mechanisms



Joint work with



BIFOLD



TECHNISCHE
UNIVERSITÄT
BERLIN

Maximilian Ernst
Aaron Peikert

Klaus-Robert Müller
Grégoire Montavon
Lorenz Linhardt



Mila



**Microsoft
Research**

Ida Momennejad Taylor Webb
Samuel Lippl Thomas McGee Po-Chen Kuo

oliver.eberle@tu-berlin.de

<https://www.linkedin.com/in/eberleoliver/>

<https://bsky.app/profile/eberleoliver.bsky.social>

