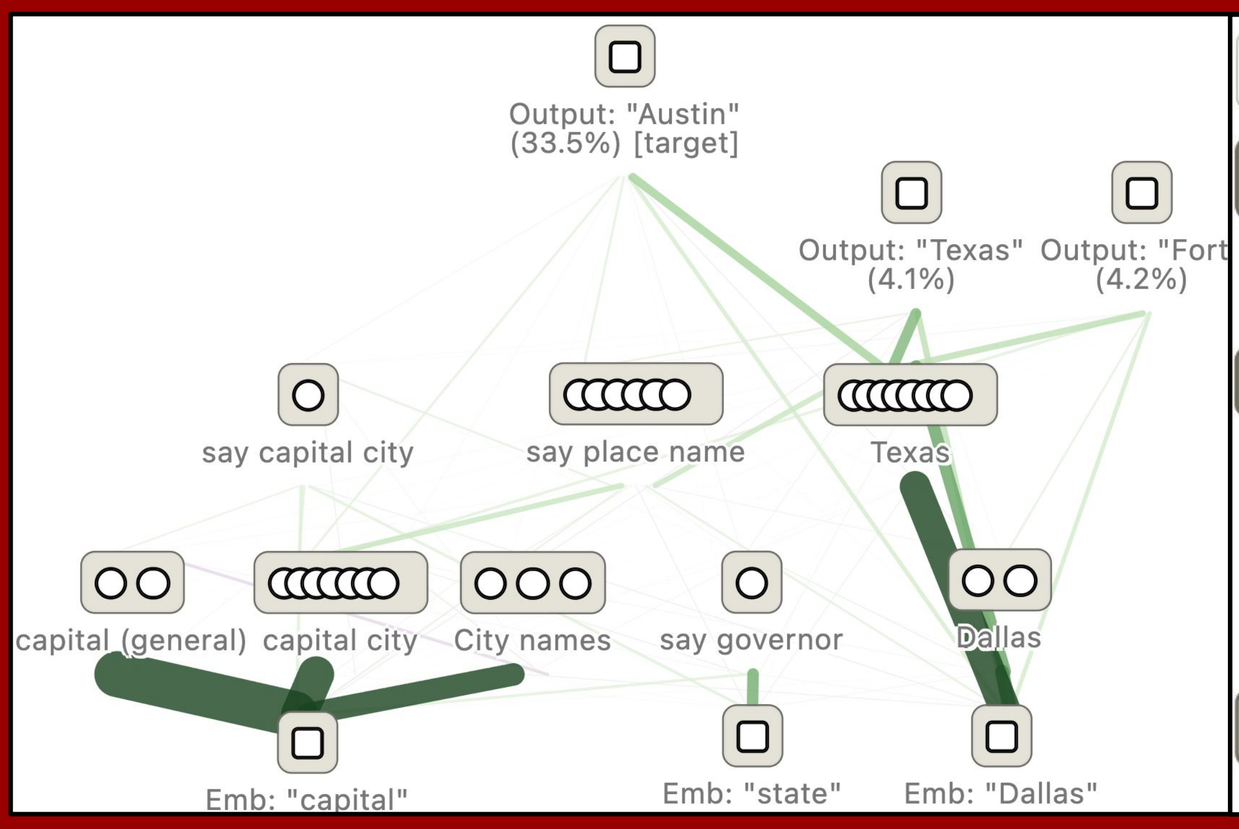
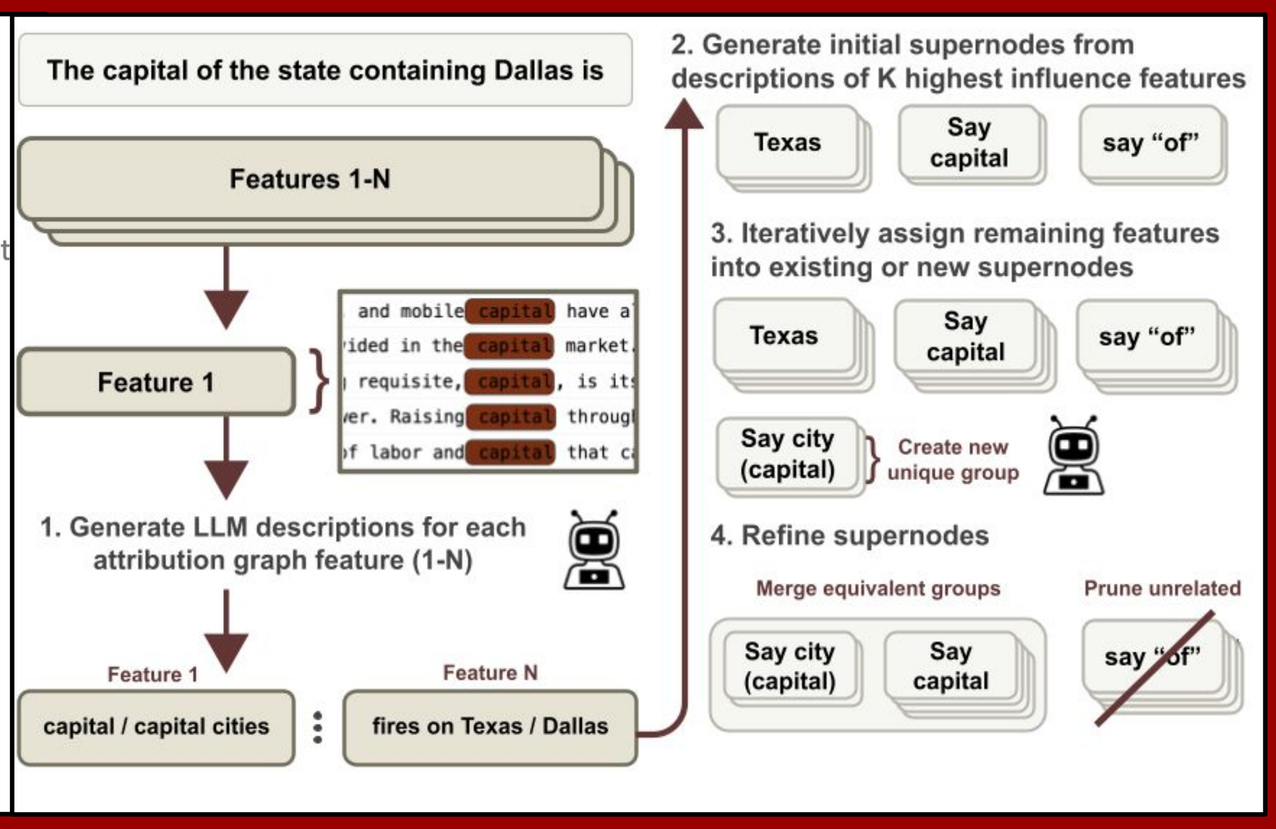
LLMs can Annotate Attribution Graphs

Ameen Patel*, Max Zhang*, Nathan Hu
*: Equal contribution





1 TL;DR

MAIN 3 TAKEAWAYS

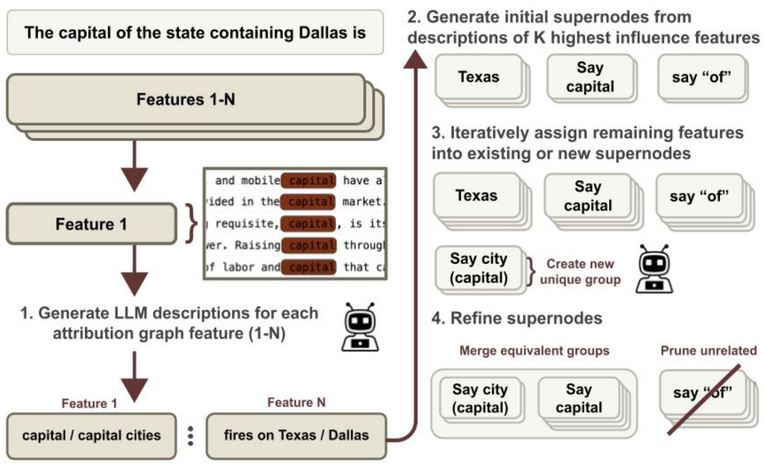
- We automate the slowest step of circuit tracing: grouping a model's features into meaningful concept groups (supernodes). An LLM does it instead of a human.
- The auto-generated groups are **as interpretable as human ones** and cost only a few cents per graph.
- It recovers the hidden intermediate step in **97 of 100** two-hop reasoning prompts, and an LLM judge can scan 1,000s of graphs to flag interesting ones.

2 Background / Motivating Question

- Circuit tracing reveals *how* a language model computes an answer internally as a graph of features.
- Annotating attribution graphs is manual and labor intensive (1-2 researcher hours)
- Question: what's the *simplest* way to automate grouping while still getting real insight?

3 Method

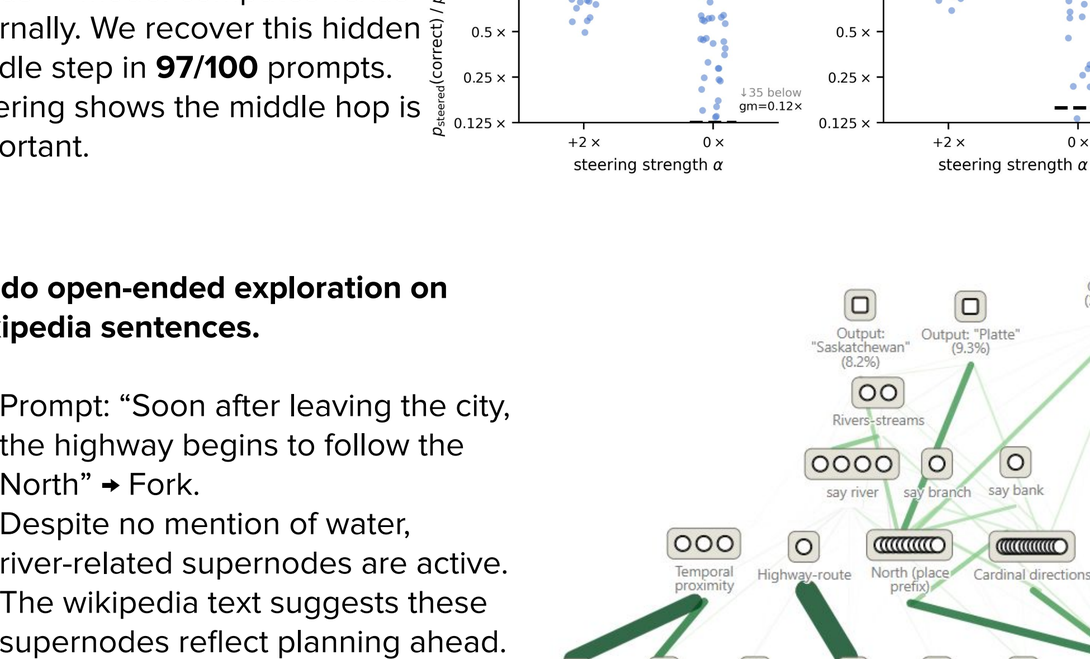
- LLM creates feature descriptions using top-activating text examples and sorts them into groups in a 3 pass format. See pipeline below.
- Cheap: **5 cents** on small graphs, **60 cents** on dense ones.
- We validate our groups using two automated interpretability metrics:
  - **Feature Detection Score:** Can an LLM identify the supernode feature?
  - **Text Detection Score:** Can an LLM identify the 5 text samples that activate a supernode feature?



Condition	Feature Detection	Text Detection	Features Grouped	# Groups	Feat./ group
Human-annotated	66.6%	81.8%	33.7	6.6	4.91
Ours (top 50)	80.8%	81.7%	26.9	7.1	3.98
Ours (top 100)	83.1%	82.1%	48.5	7.1	7.08
Ours (top 150)	82.8%	84.2%	68.1	7.5	9.43
Ours (top 200)	81.9%	81.7%	88.5	8.1	11.62
Ours (full)	93.4%	83.2%	207.5	8.3	26.42
Ours (full, no refine)	92.3%	82.1%	250.1	11.1	24.30

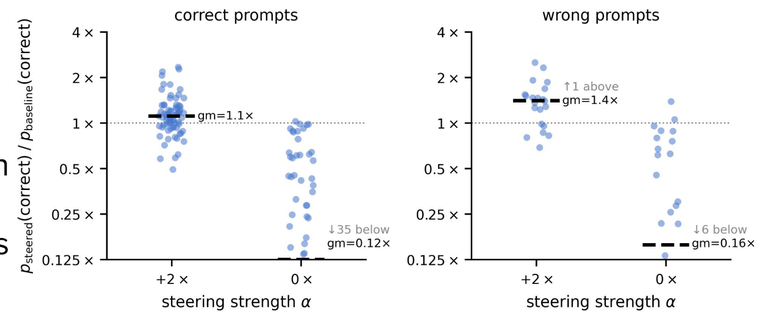
On 15 Neuronpedia graphs from Hanna et al., **our supernodes match or slightly exceed human-annotated ones** on both automated interpretability metrics of Feature Detection and Text Detection.

4 Experiments



Left: ours (top 100 features). **Right: Neuronpedia (human)**. Prompt: “Mexico:peso :: US:” The two-step analogy is recovered on both sides.

Two-hop reasoning (Capitals): "capital of the state containing Dallas" → model computes *Texas* internally. We recover this hidden middle step in **97/100** prompts. Steering shows the middle hop is important.



We do open-ended exploration on Wikipedia sentences.

- Prompt: “Soon after leaving the city, the highway begins to follow the North” → Fork.
- Despite no mention of water, river-related supernodes are active.
- The wikipedia text suggests these supernodes reflect planning ahead.

5 Limitations and Discussion

- Tested on one model (Gemma-2-2B) and one annotator LLM.
- Grouping is naive, ignoring where features sit and how they connect.
- Breaks on arithmetic: labels too coarse to capture known add/carry structure (only 1% of groups name a real mechanism).
- Still, even simple automation gives useful annotations — lots of low-hanging fruit left.

PAPER

CODE

ameen.patel.210@gmail.com, maxhzhang119@gmail.com, nathu@cs.stanford.edu