Tracking Equivalent Mechanistic Interpretations Across Neural Networks

Alan Sun
Carnegie Mellon University

Mariya Toneva
Max Planck Institute for Software Systems

Tracking Equivalent Mechanistic Interpretations Across Neural Networks

Alan Sun
Carnegie Mellon University

Mariya Toneva
Max Planck Institute for Software Systems

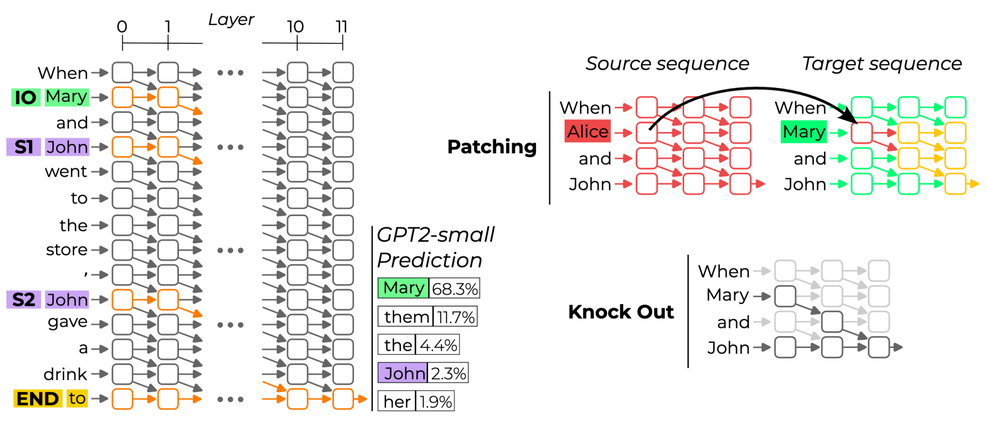
Can we tell whether two neural networks use the same underlying algorithm without having to fully interpret either one?

Why Interpretive Equivalence?

Fix some task and some model (usually a neural network)

Why Interpretive Equivalence?

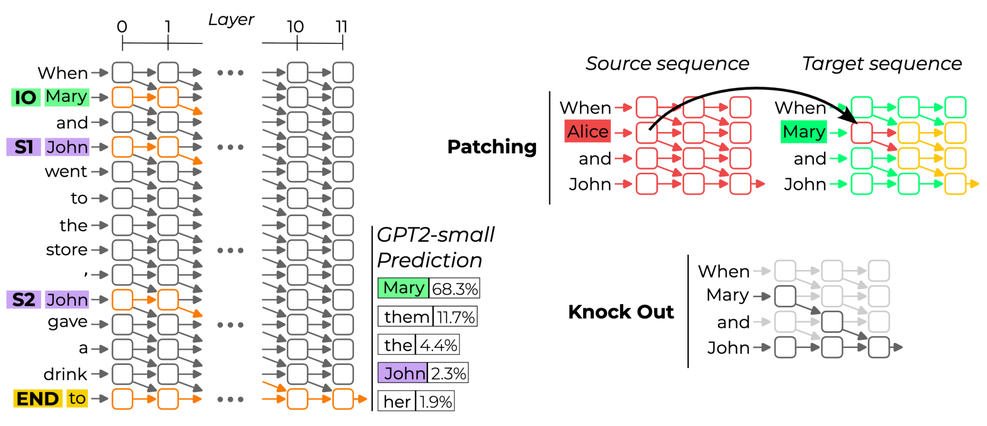
Fix some task and some model (usually a neural network)



✦ ***Mechanistic interpretability:*** recover the “algorithm” used to solve the task

Why Interpretive Equivalence?

Fix some task and some model (usually a neural network)

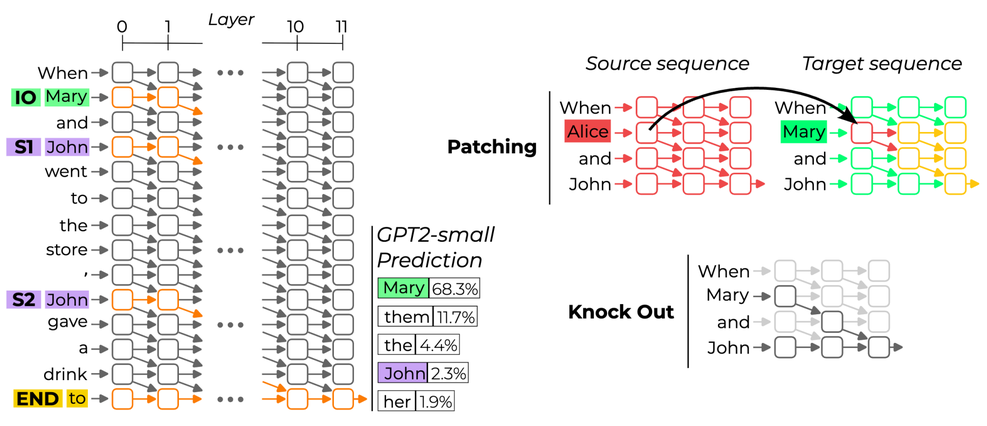


- ***Mechanistic interpretability:*** recover the “algorithm” used to solve the task
- ***Scalability issues***:

Why Interpretive Equivalence?

Fix some task and some model (usually a neural network)

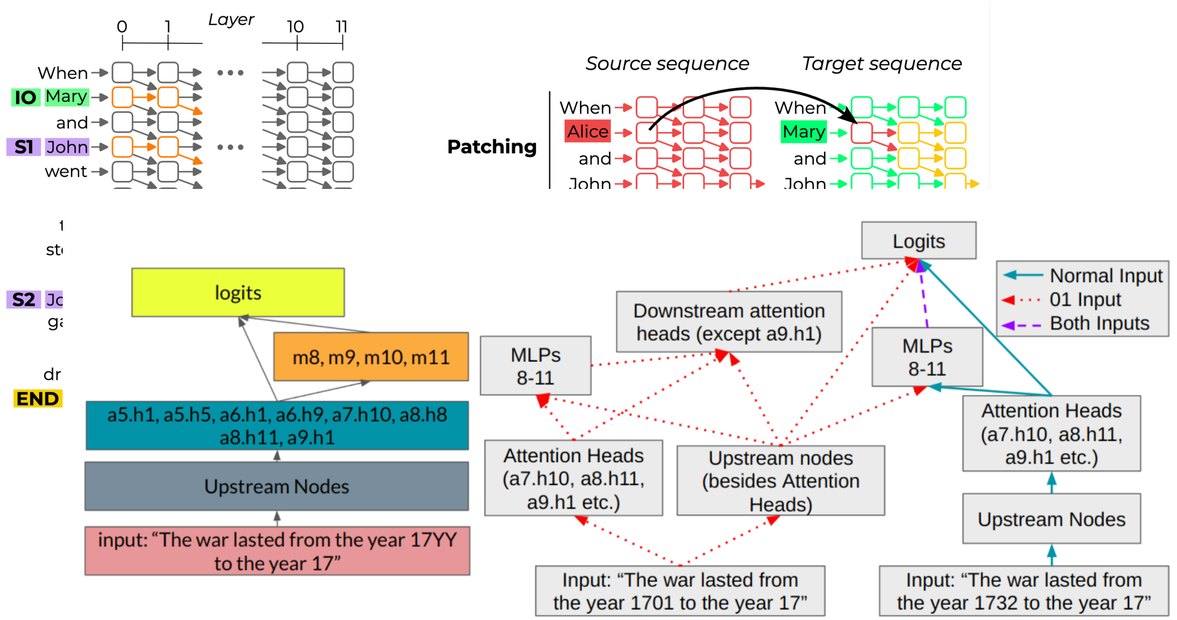
- ***Mechanistic interpretability:*** recover the “algorithm” used to solve the task
- ***Scalability issues***:
  - At least as hard as **manually constructing** an algorithm to solve the task
  - ***Compute-*** and ***labor-intensive***
  - Multiple circuits can elicit the same algorithm and vice versa



Why Interpretive Equivalence?

Fix some task and some model (usually a neural network)

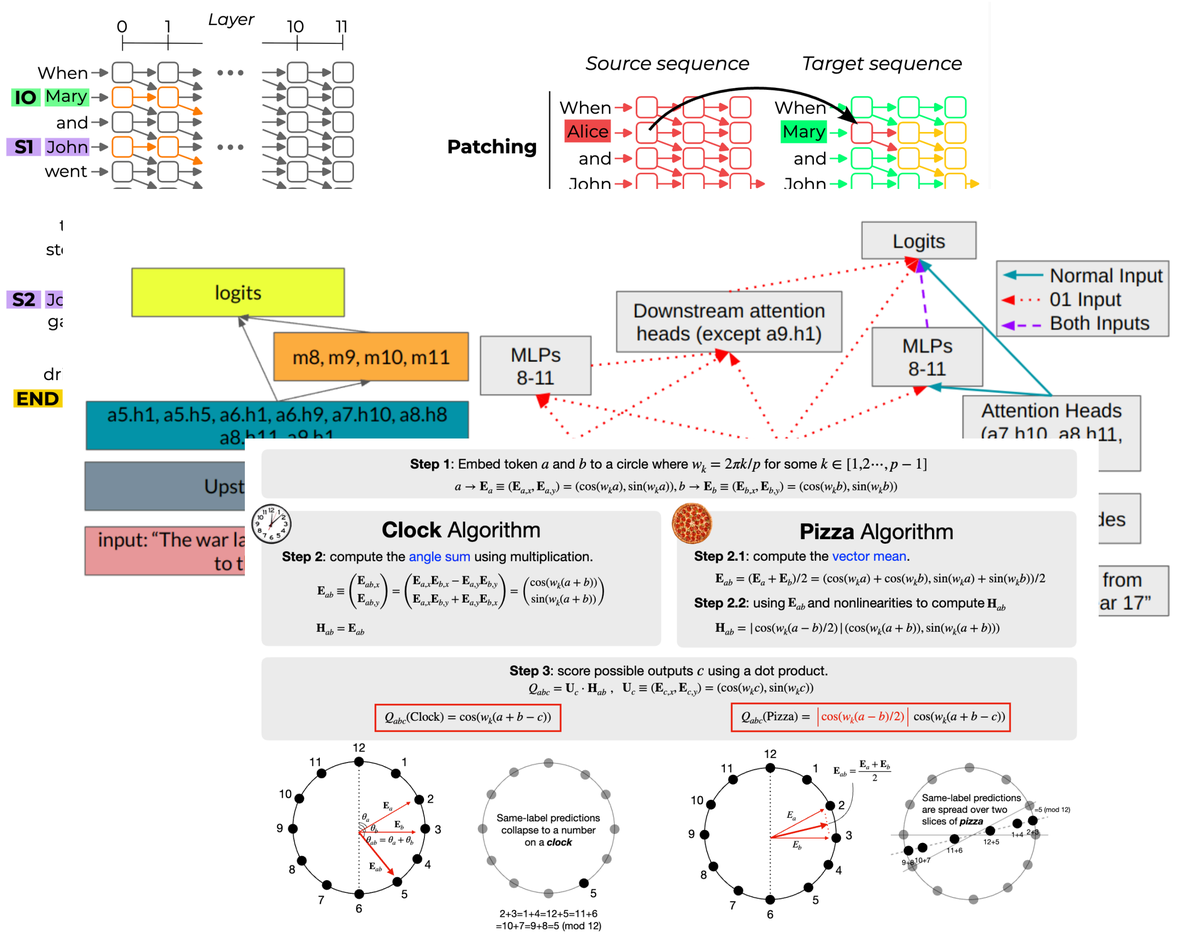
- ***Mechanistic interpretability:*** recover the “algorithm” used to solve the task
- ***Scalability issues***:
  - At least as hard as **manually constructing** an algorithm to solve the task
  - ***Compute-*** and ***labor-intensive***
  - Multiple circuits can elicit the same algorithm and vice versa



Why Interpretive Equivalence?

Fix some task and some model (usually a neural network)

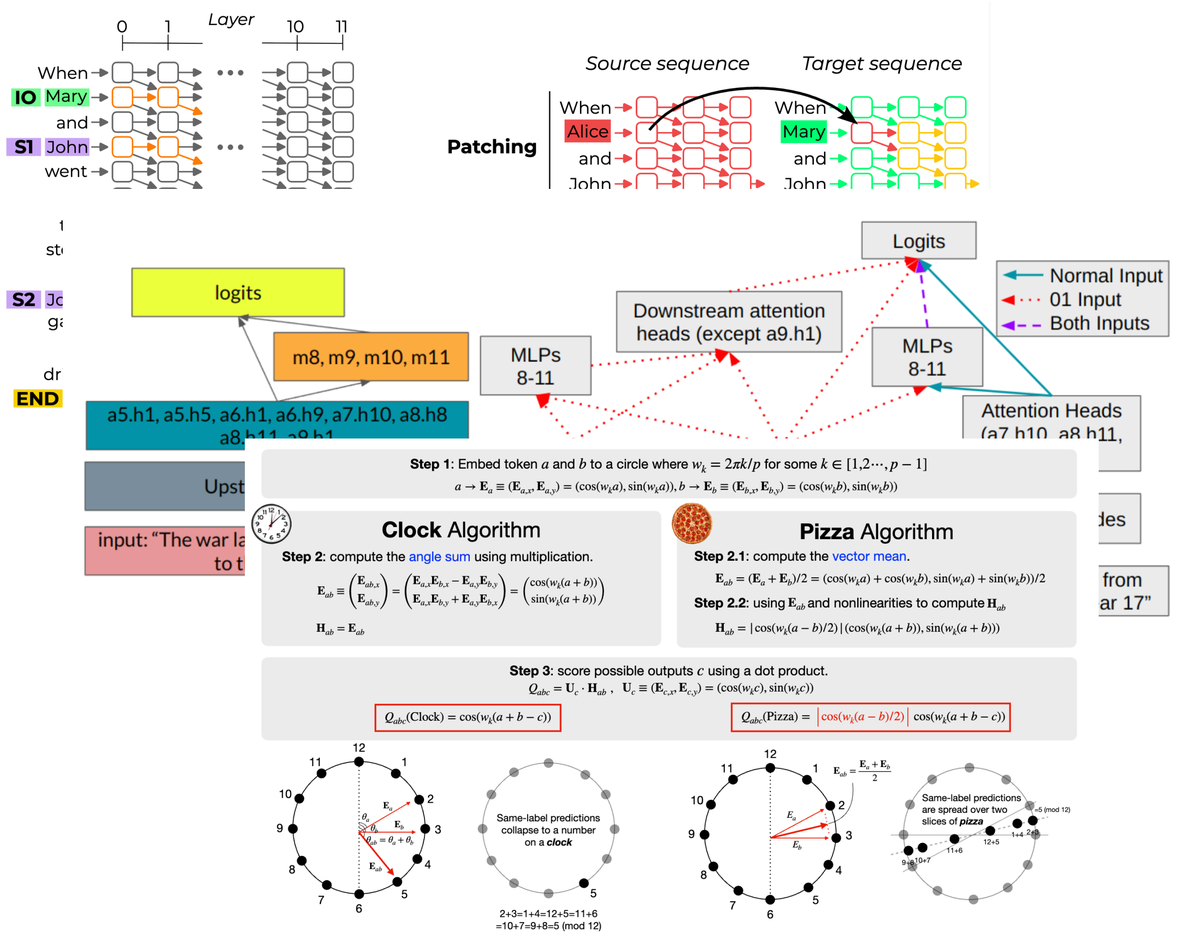
- ***Mechanistic interpretability:*** recover the “algorithm” used to solve the task
- ***Scalability issues***:
  - At least as hard as **manually constructing** an algorithm to solve the task
  - ***Compute-*** and ***labor-intensive***
  - Multiple circuits can elicit the same algorithm and vice versa



Why Interpretive Equivalence?

Fix some task and some model (usually a neural network)

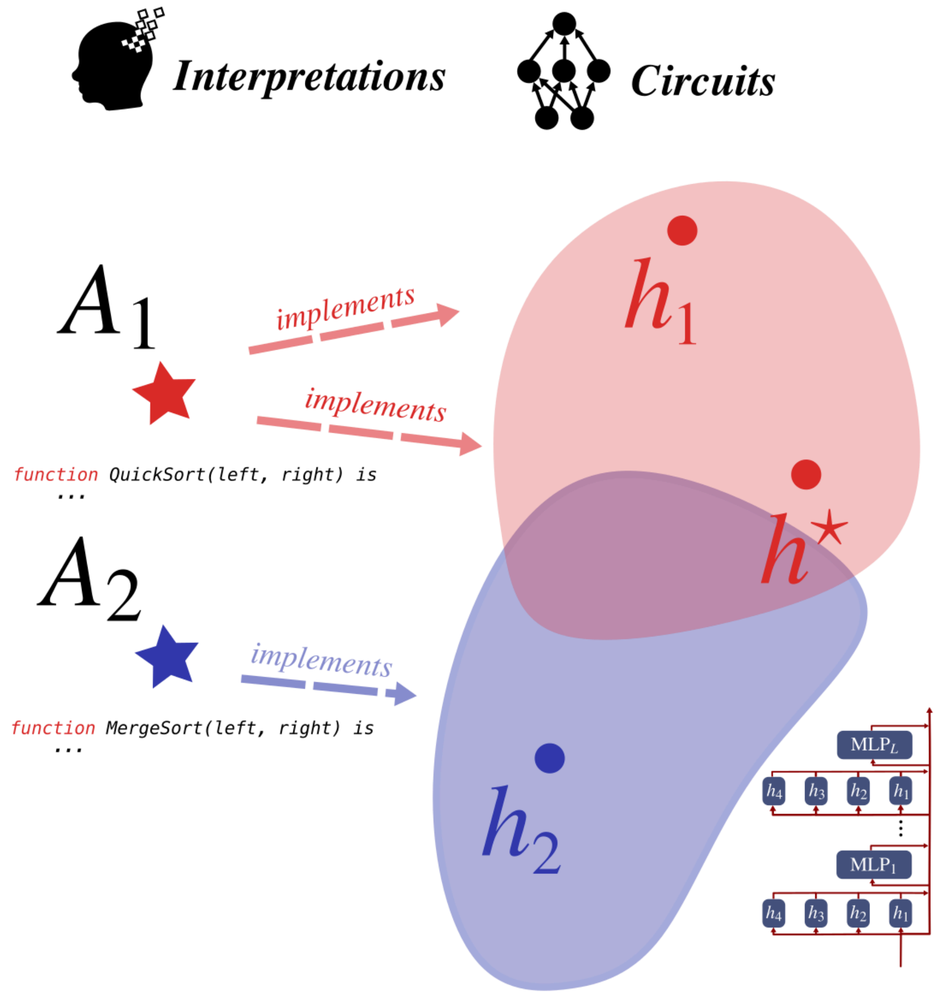
- ***Mechanistic interpretability:*** recover the “algorithm” used to solve the task
- ***Scalability issues***:
  - At least as hard as **manually constructing** an algorithm to solve the task
  - ***Compute-*** and ***labor-intensive***
  - Multiple circuits can elicit the same algorithm and vice versa
- We take a different approach: do two models share the same interpretations? If so, we can do ***reductions***



Key Insight

Interpretive Equivalence

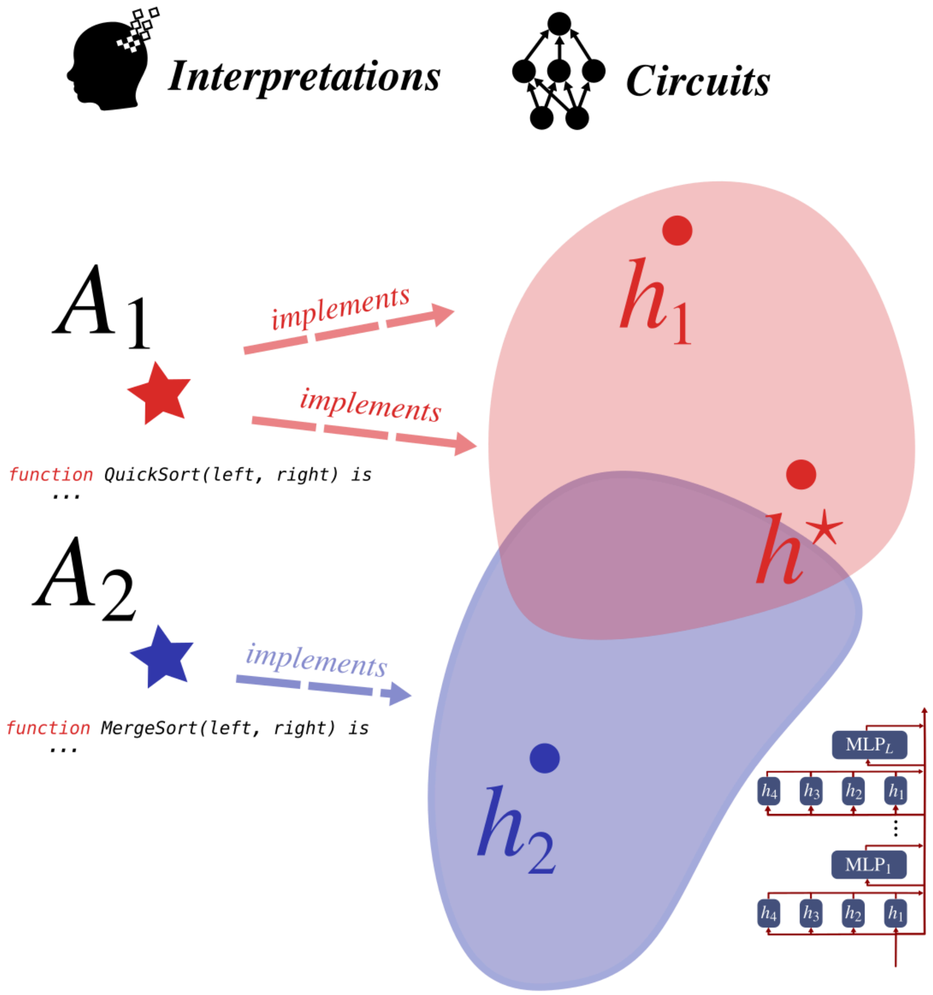
Two interpretations are equivalent <u>if and only if</u> all of their implementations are equivalent



Key Insight

Interpretive Equivalence

Two interpretations are equivalent <u>if and only if</u> all of their implementations are equivalent

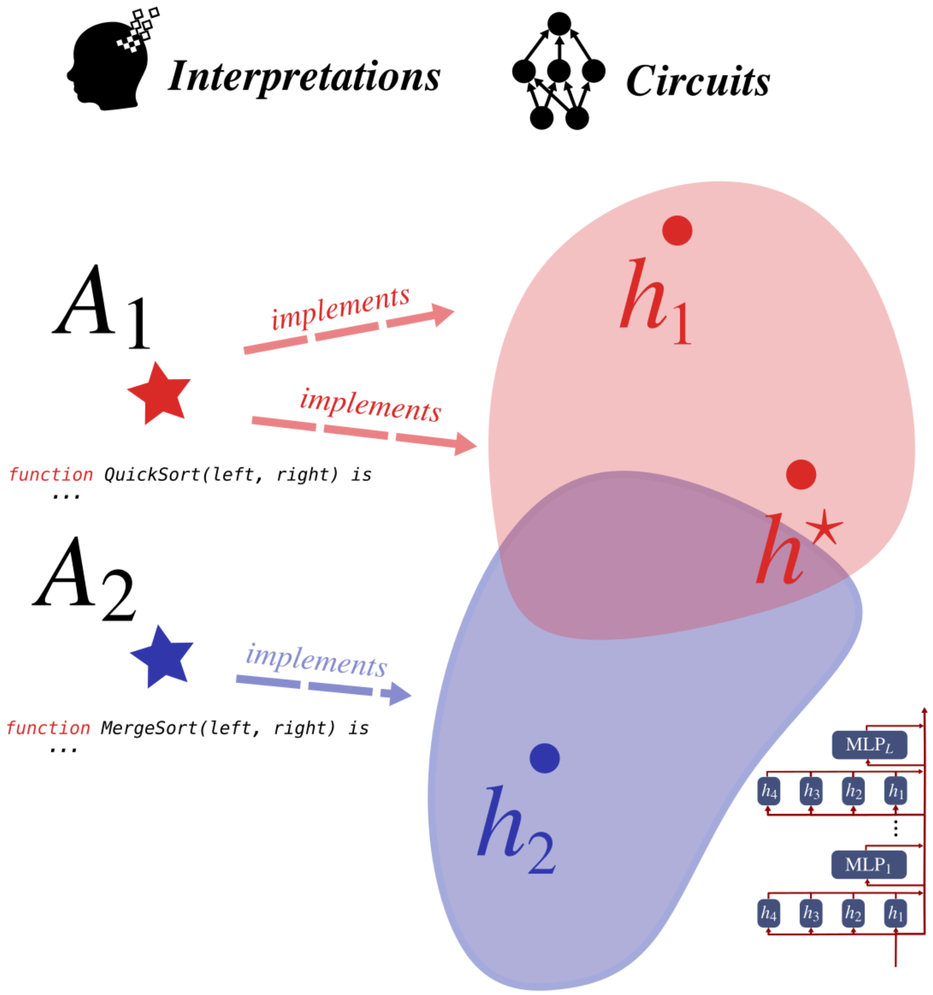


✦ Avoid writing down the mechanism explicitly

Key Insight

Interpretive Equivalence

Two interpretations are equivalent <u>if and only if</u> all of their implementations are equivalent



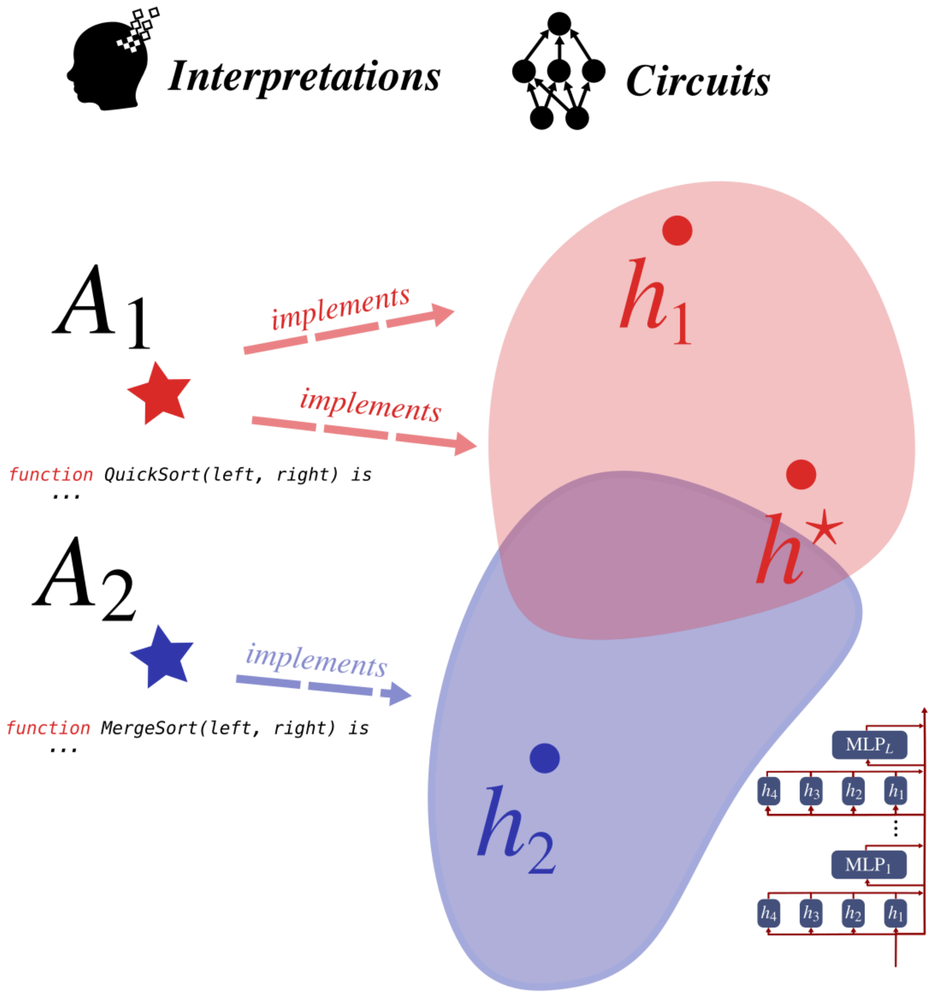
- ✦ Avoid writing down the mechanism explicitly
- ✦ Instead compare the space of implementations [weight and representation space]

Key Insight

Interpretive Equivalence

Two interpretations are equivalent <u>if and only if</u> all of their implementations are equivalent

- ✦ Avoid writing down the mechanism explicitly
- ✦ Instead compare the space of implementations [weight and representation space]
- ✦ Intuition:
  - ✦ If two model's use the same interpretation then appropriate perturbations of one can make it "look like" the other
  - ✦ "look like" : similarity in their representations



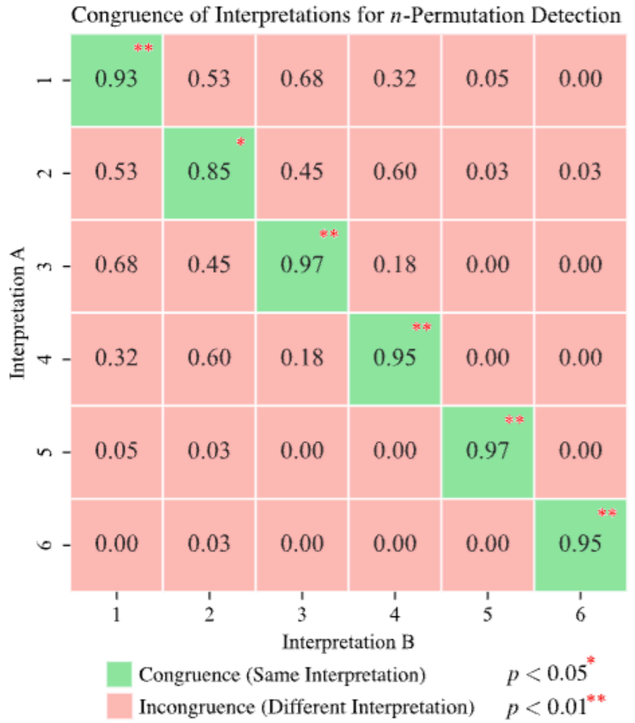
Experiments

If we cannot reliably tell the difference between two models' representations across perturbations, then they are **congruent**.

Experiments

If we cannot reliably tell the difference between two models' representations across perturbations, then they are **congruent**.

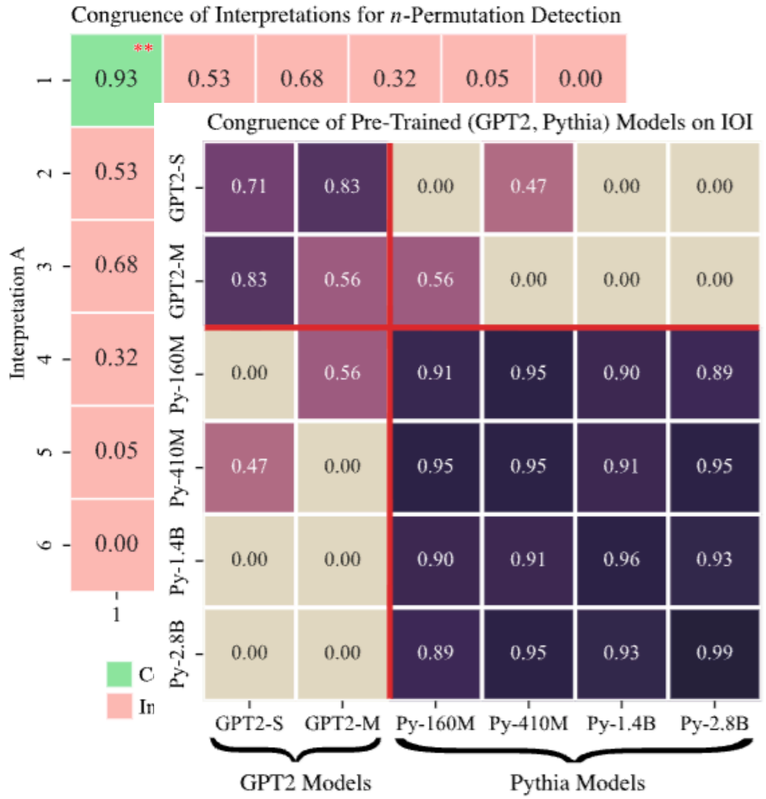
- **Toy task:** the method separates hand-designed algorithms known to be different



Experiments

If we cannot reliably tell the difference between two models' representations across perturbations, then they are **congruent**.

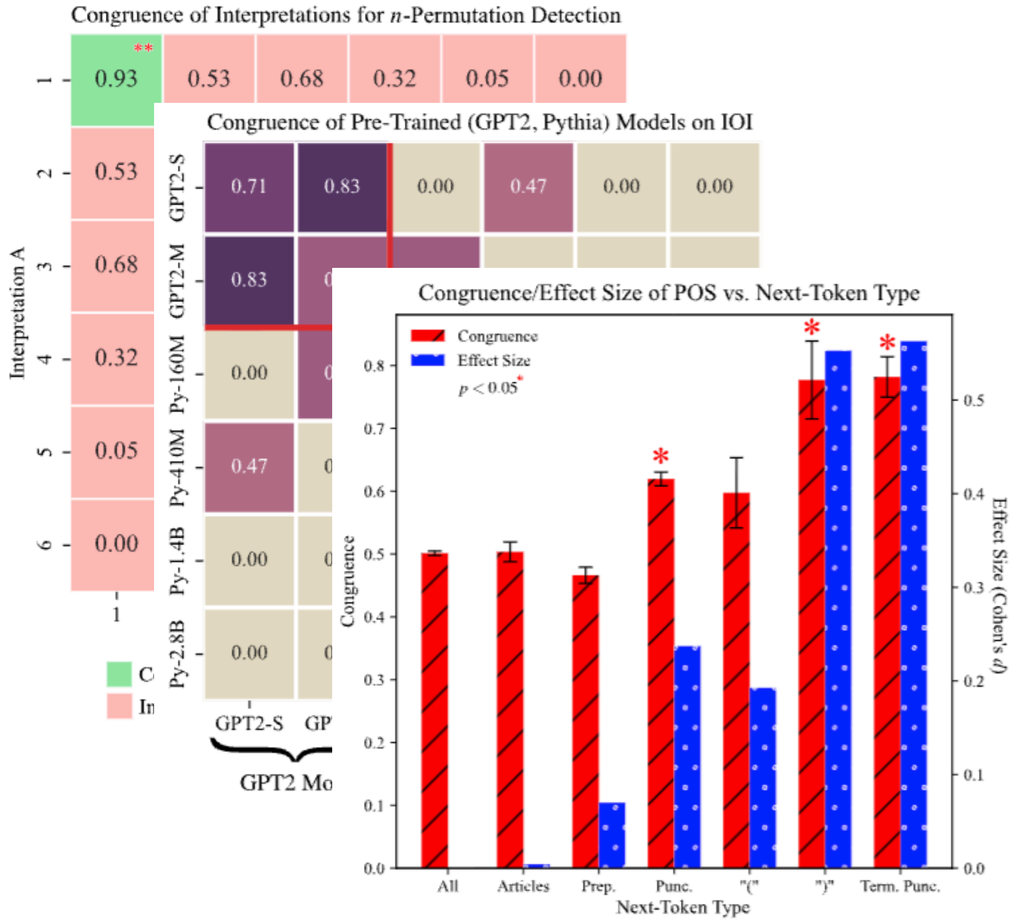
- **Toy task:** the method separates hand-designed algorithms known to be different
- **Across model scale:** on IOI task, GPT-2 models cluster together, Pythia models cluster together, matching prior circuit-level findings



Experiments

If we cannot reliably tell the difference between two models' representations across perturbations, then they are **congruent**.

- **Toy task:** the method separates hand-designed algorithms known to be different
- **Across model scale:** on IOI task, GPT-2 models cluster together, Pythia models cluster together, matching prior circuit-level findings
- **Across tasks:** some next-token predictions in GPT-2 align with a simpler syntactic task, parts-of-speech tagging



Takeaways

- ✦ Make mechanistic interpretability **more scalable**:
  - ✦ Reduction to **smaller** models
  - ✦ Reduction to **simpler** tasks
  - ✦ More **rigorous evaluation** of interpretability claims [see §4, 5, 6]
- ✦ **Theory:** rigorously define interpretive equivalence using causal abstraction and show bounds on congruity
- ✦ **Reusing interpretability results** instead of starting from scratch for every new model