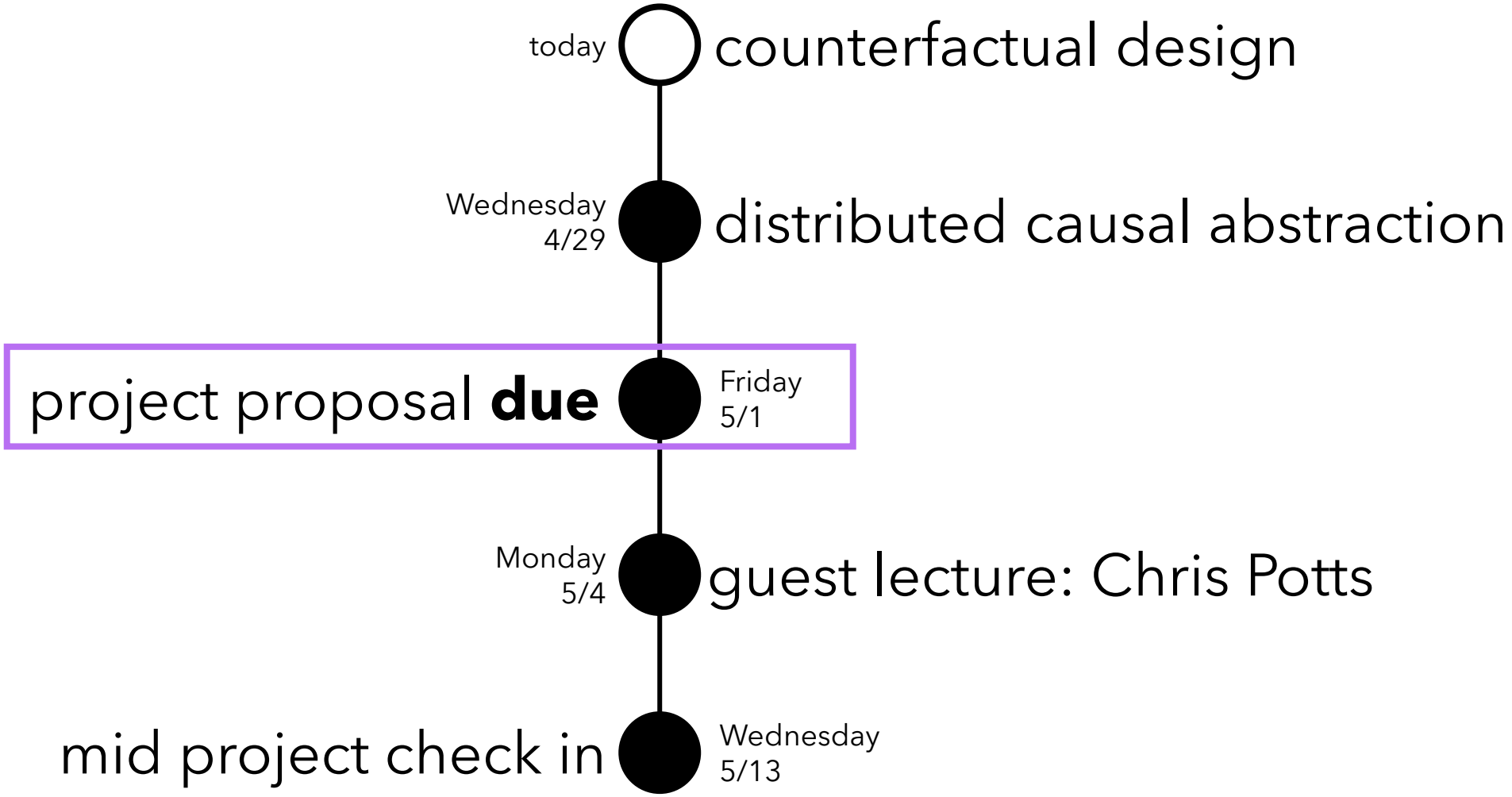
# designing counterfactuals for causal abstraction

CS 221M
Week 5, Lecture 9

# Course timeline



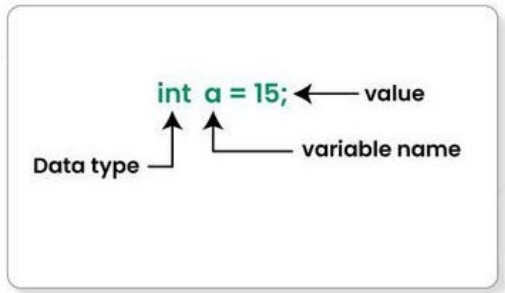
# Attendance form

- Participation policy
  You get full credit for submitting the attendance form, even if it's incomplete
- Just make sure it has your name!

tinyurl.com/cs221m-lecture9

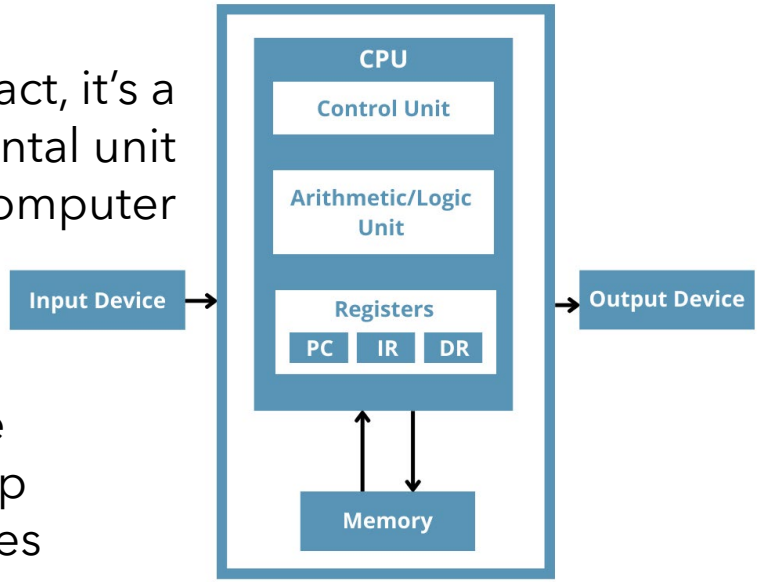
# Motivation: transformers use pointers

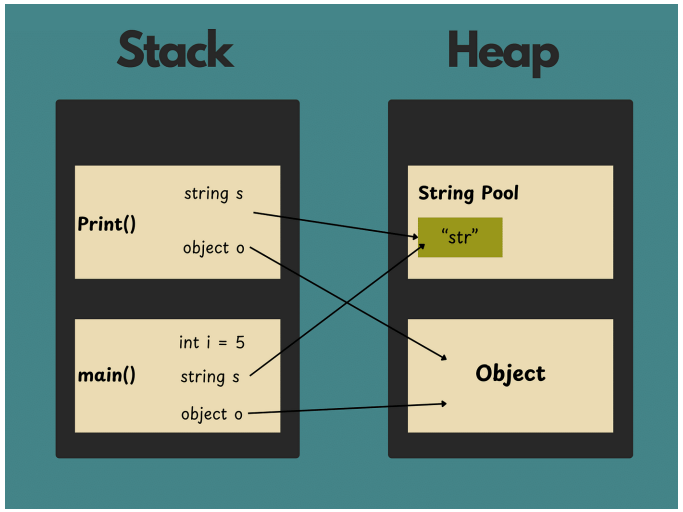
what does “memory” in transformers look like?




Assigning variables to values is a coding fundamental

In fact, it’s a fundamental unit of a computer



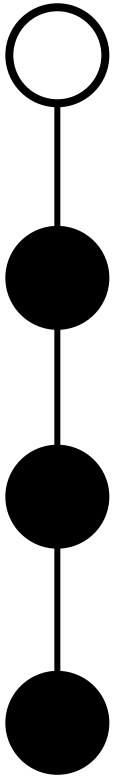

Computers use *pointers* to keep track of variables

**We’ll show that transformers do the same thing**

# Lecture overview

- example: indirect objects
- theory: designing counterfactuals
- example: MCQA
- example: entity binding

# Lecture overview



example: indirect objects

theory: designing counterfactuals

example: MCQA

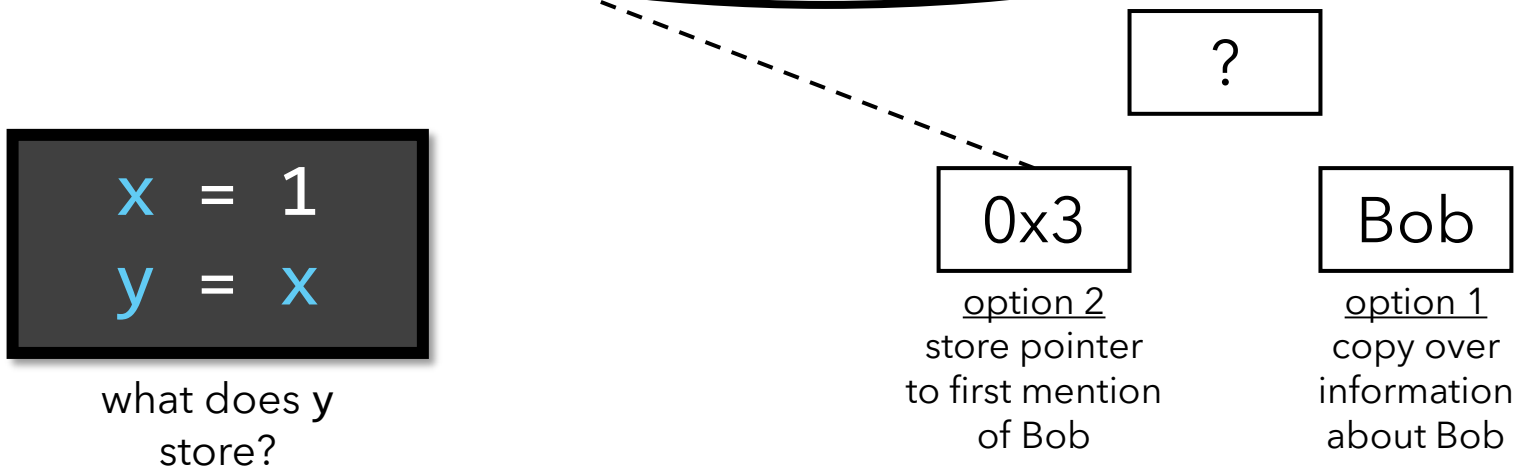
example: entity binding

# Indirect object identification (IOI)

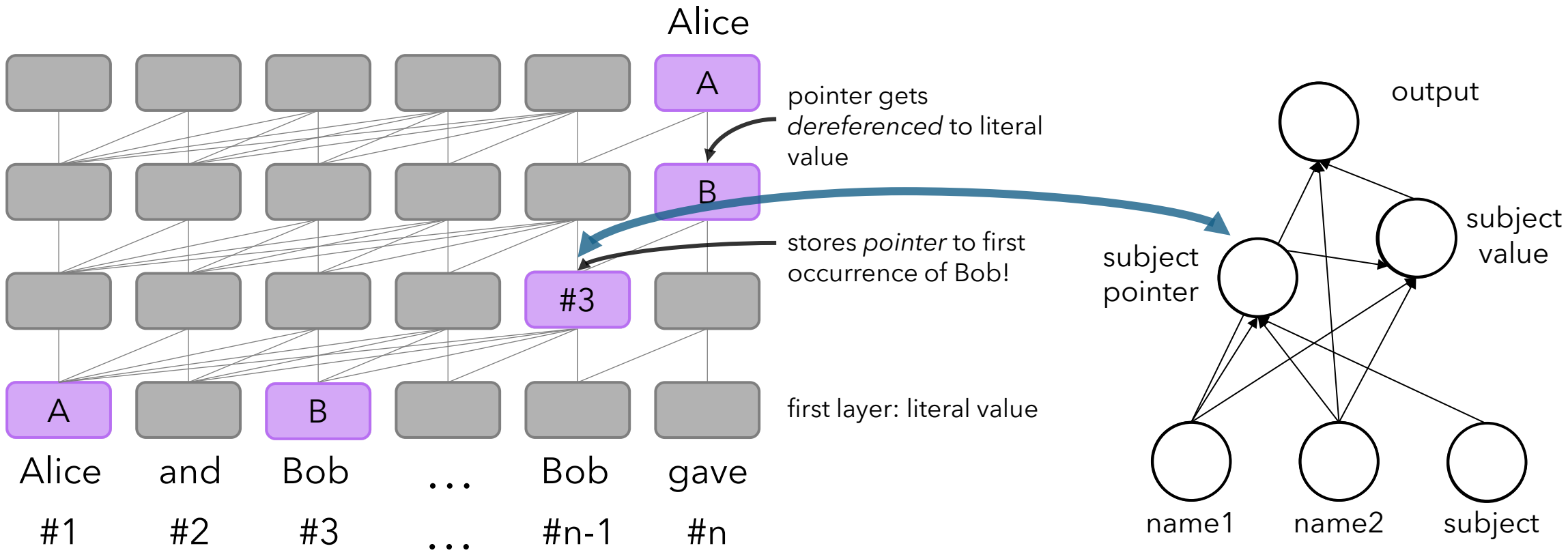
How do models represent anaphora?

# Tracking memory: indirect object

Alice and Bob went to the library. Bob gave a book to _____



# Tracking memory: indirect object

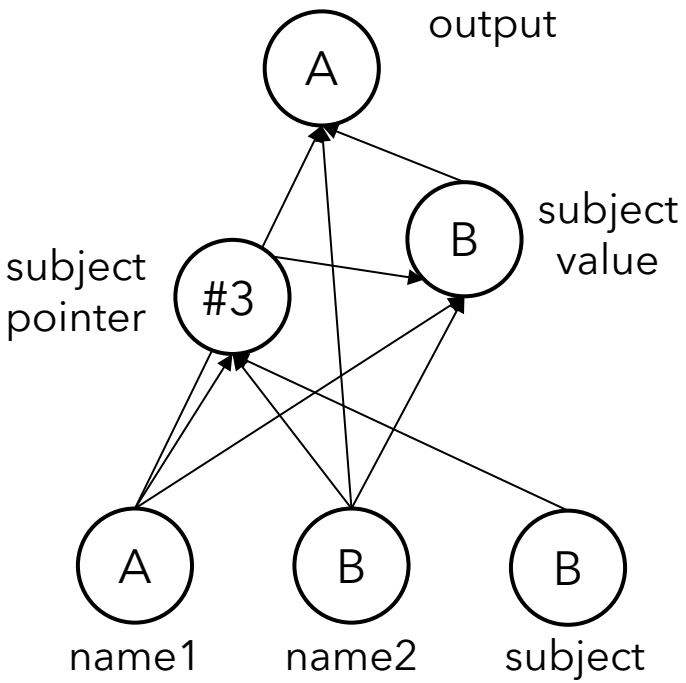



Alice and Bob went to the library. Bob gave a book to _____

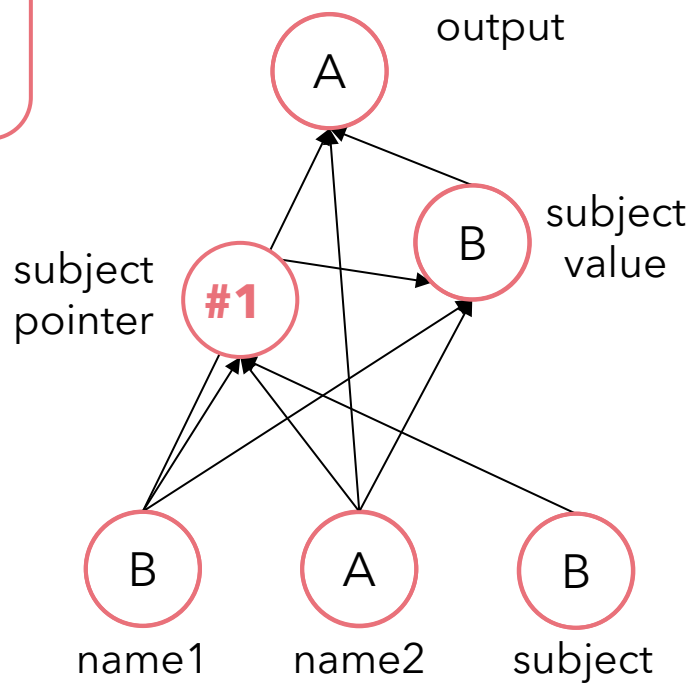
Wang et al. 2022 Interpretability in the Wild

# Tracking memory: indirect object

how do we change just the value of the subject pointer?

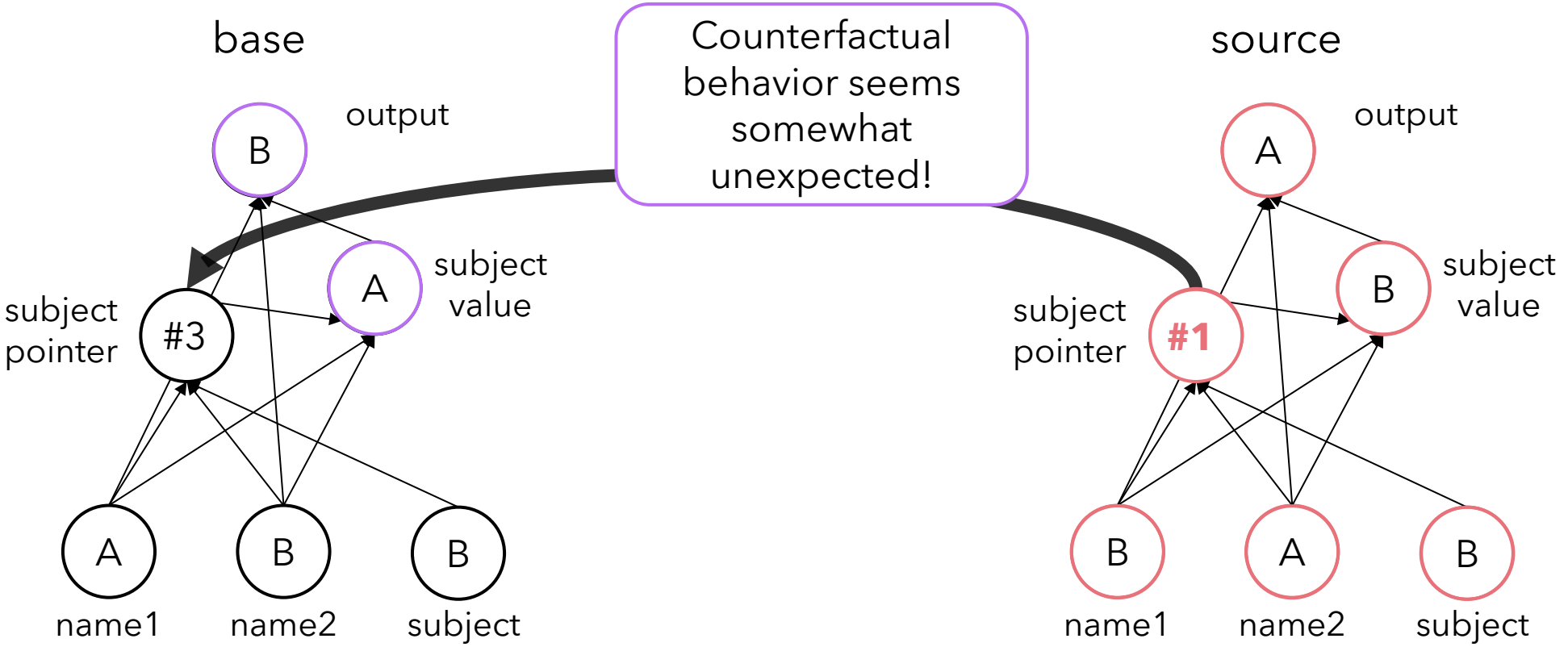



Alice and Bob went to the library.
Bob gave a book to _____




**Bob** and **Alice** went to the library.
Bob gave a book to _____

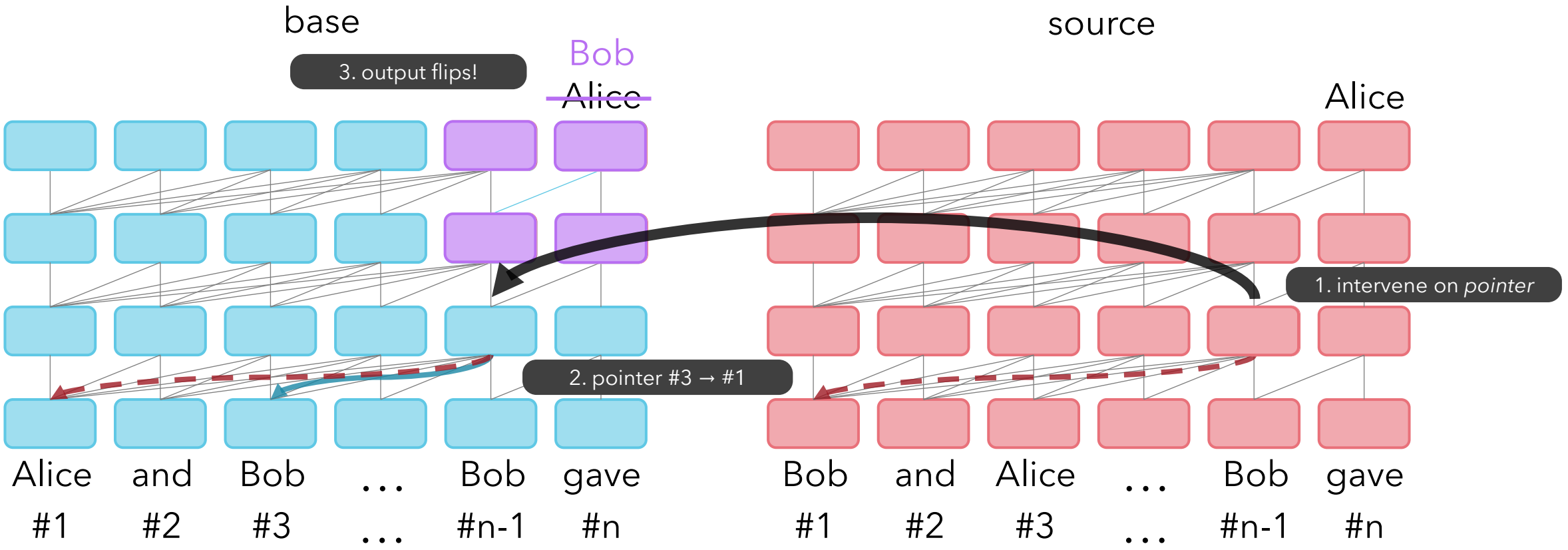
# Tracking memory: indirect object




Alice and Bob went to the library.
Bob gave a book to _____

**Bob** and **Alice** went to the library.
Bob gave a book to _____

# Tracking memory: indirect object

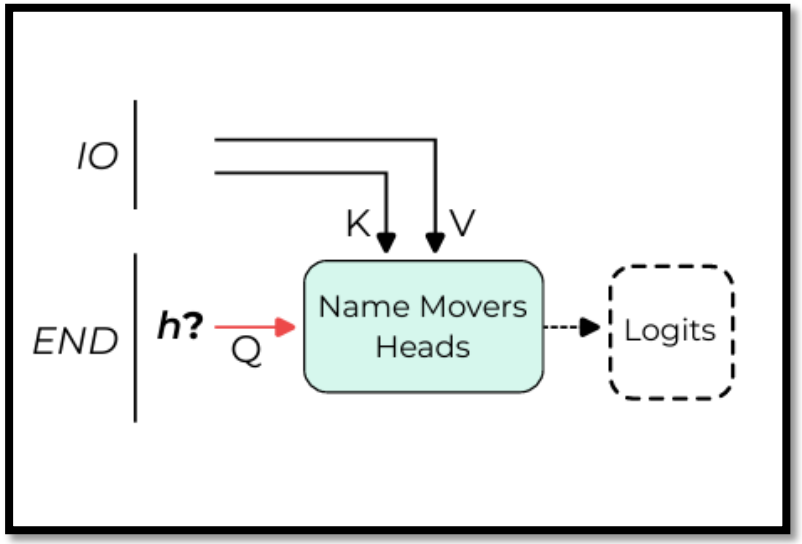



**Alice** and **Bob** went to the library.
Bob gave a book to _____

**Bob** and **Alice** went to the library.
Bob gave a book to _____

# Let's try it out!

exercise on workbench

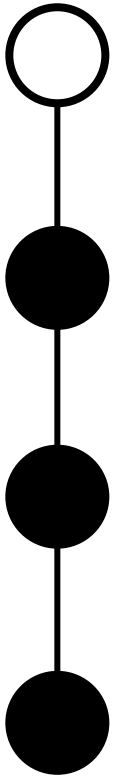


# Attendance form

- Participation policy
  You get full credit for submitting the attendance form, even if it's incomplete
- Just make sure it has your name!

tinyurl.com/cs221m-lecture9

# Lecture overview



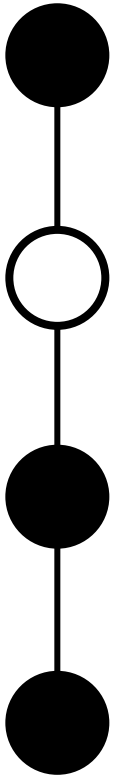
example: indirect objects

theory: designing counterfactuals

example: MCQA

example: entity binding

# Lecture overview

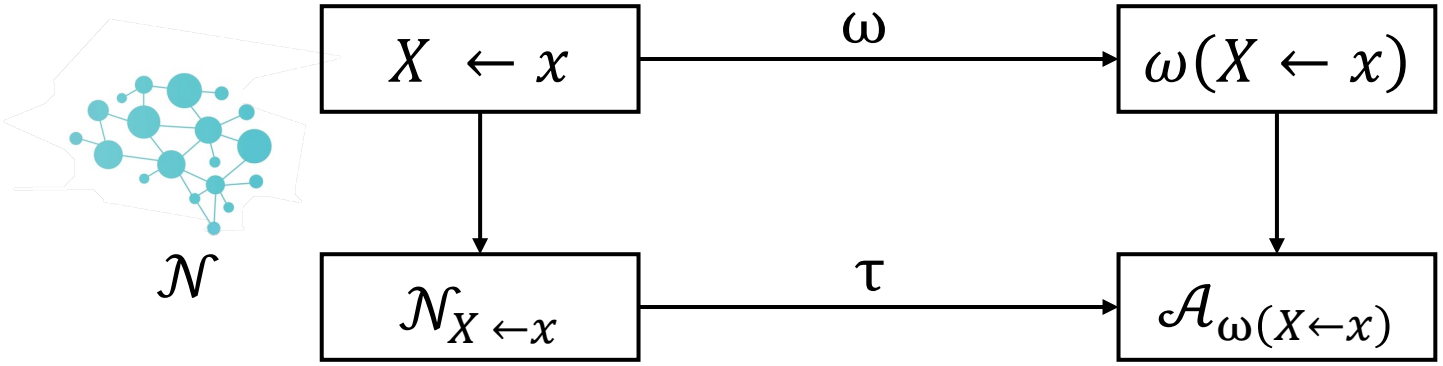


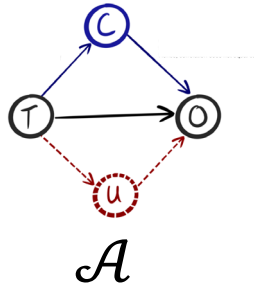
- example: indirect objects
- theory: designing counterfactuals
- example: MCQA
- example: entity binding

# Designing counterfactuals

Revisiting hierarchical equality

# Formalizing causal abstraction




$$\tau(\mathcal{N}_{X \leftarrow x}) = \mathcal{A}_{\omega(X \leftarrow x)}$$

with respect to a dataset of paired inputs,

$$\mathrm{IIA}(\omega, \tau; D) = \sum_{b,s \in D} \mathbb{1}\left[\tau\left(\mathcal{N}_{X \leftarrow x_s}(b)\right) = \mathcal{A}_{\omega(X \leftarrow x_s)}(b)\right]$$

how should we choose counterfactual pairs?

# Criteria for counterfactuals

how should we choose counterfactual pairs?

Change variable

b and s should have different values for x

$$\mathcal{A}(b)[X] \neq \mathcal{A}(s)[X]$$

Change value

Intervention should change output

$$\mathcal{A}_{X \to \mathcal{A}(s)[X]}(b) \neq \mathcal{A}(b)$$

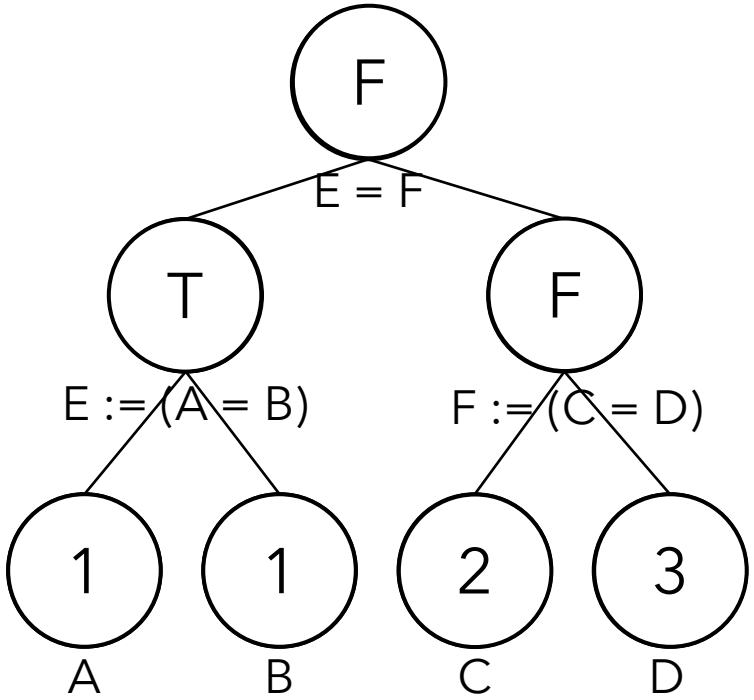
Rule out alternatives

Can't be explained by a different intervention

$$\mathcal{A}_{X \to \mathcal{A}(s)[X]}(b) \neq \mathcal{A}_{W \to \mathcal{A}(s)[W]}(b)$$

this is only with respect to $\mathcal{A}$!

# Revisiting hierarchical equality




Change variable

b and s should have different values for x

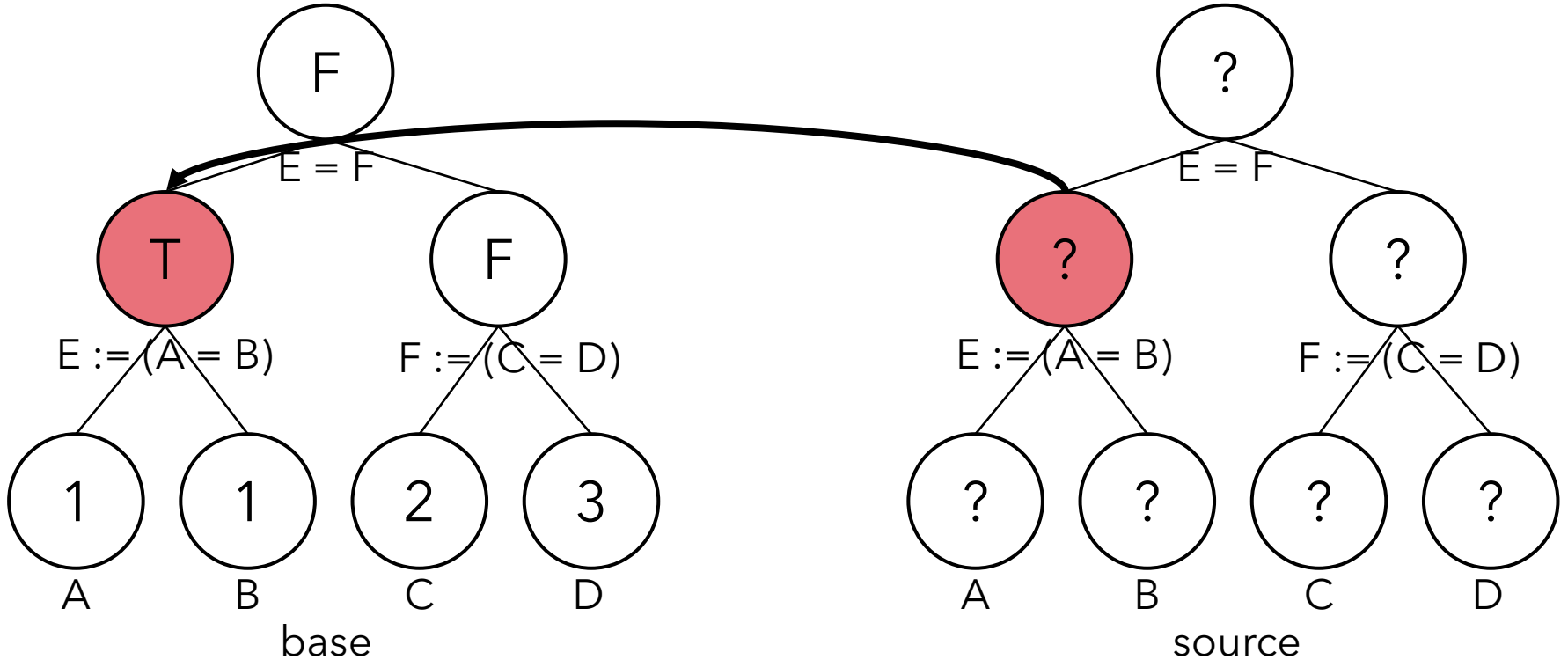
Change value

Intervention should change output

Rule out alternatives

Can’t be explained by a different intervention

# Revisiting hierarchical equality




Change variable

b and s should have different values for x

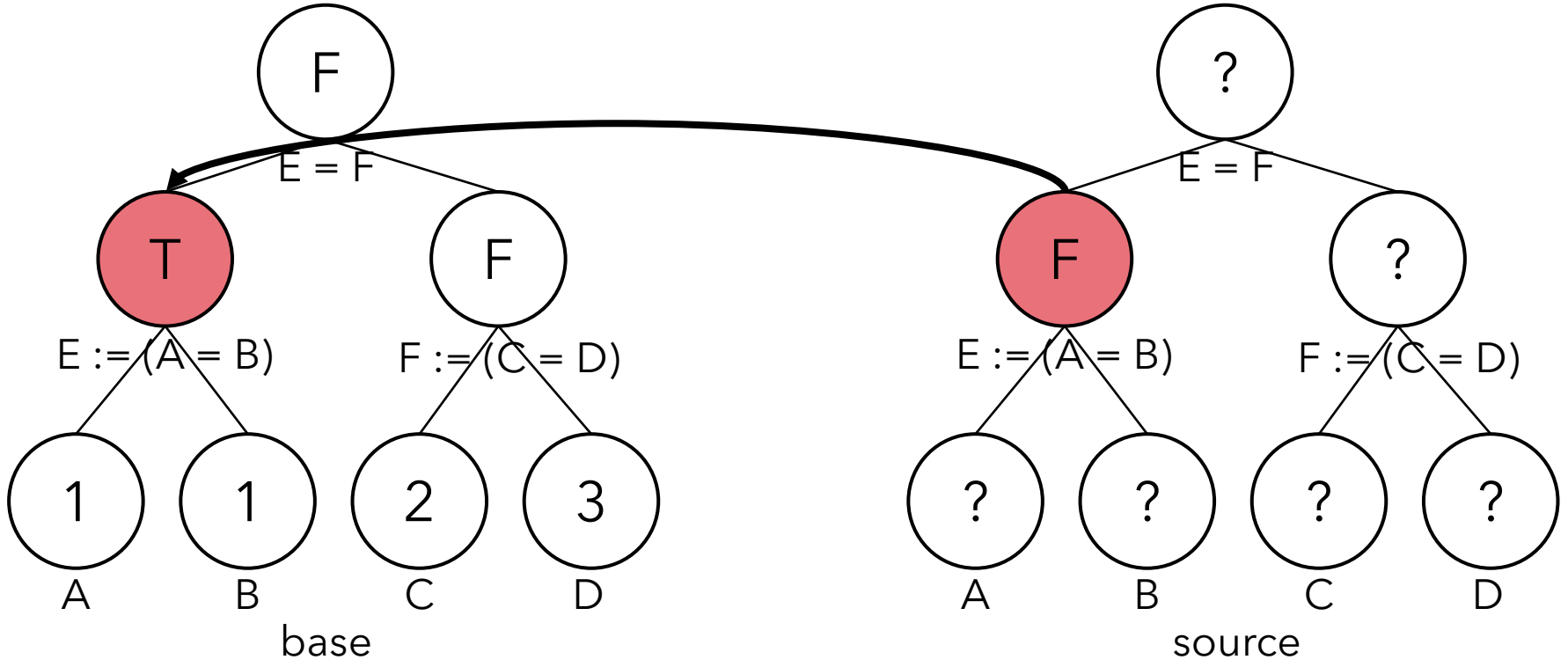
Change value

Intervention should change output

Rule out alternatives

Can't be explained by a different intervention

# Revisiting hierarchical equality




Change variable

b and s should have different values for x

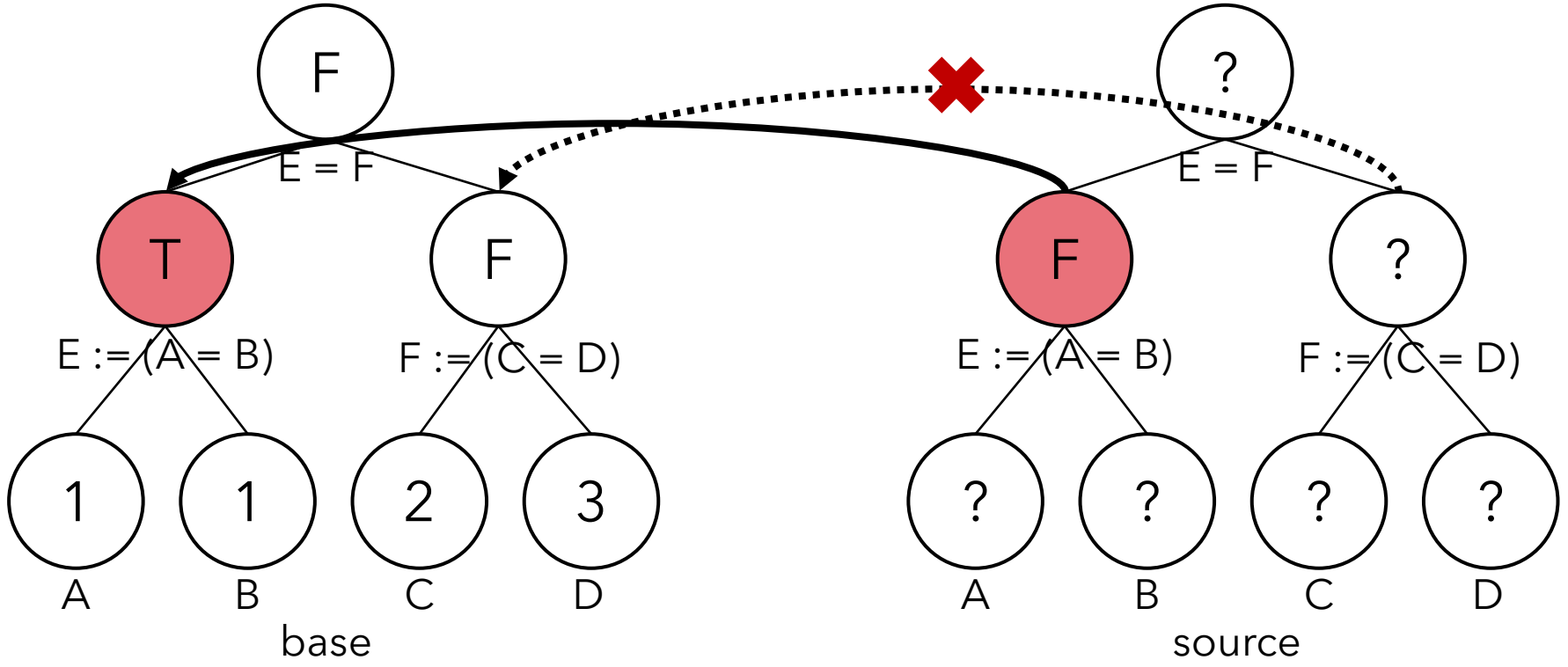
Change value

Intervention should change output

Rule out alternatives

Can't be explained by a different intervention

# Revisiting hierarchical equality




Change variable

b and s should have different values for x

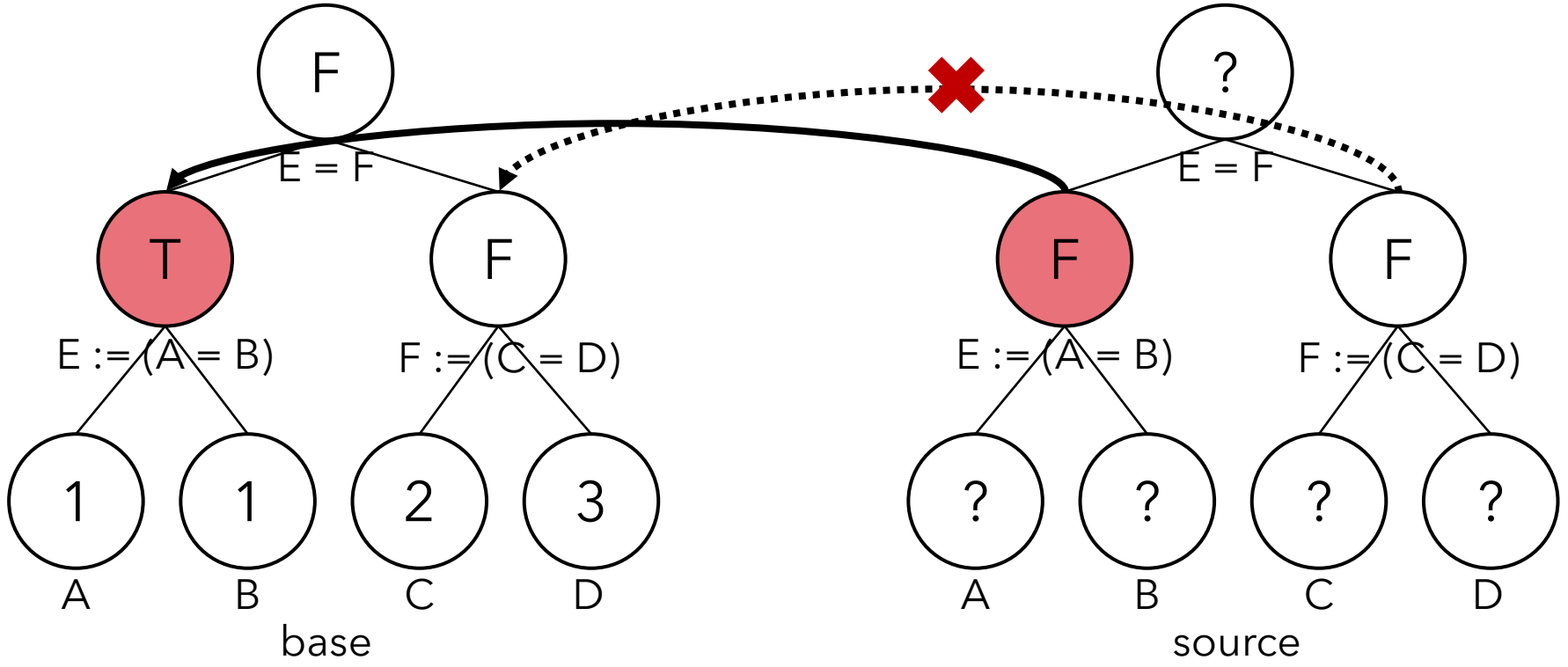
Change value

Intervention should change output

Rule out alternatives

Can't be explained by a different intervention

# Revisiting hierarchical equality




Change variable

b and s should have different values for x

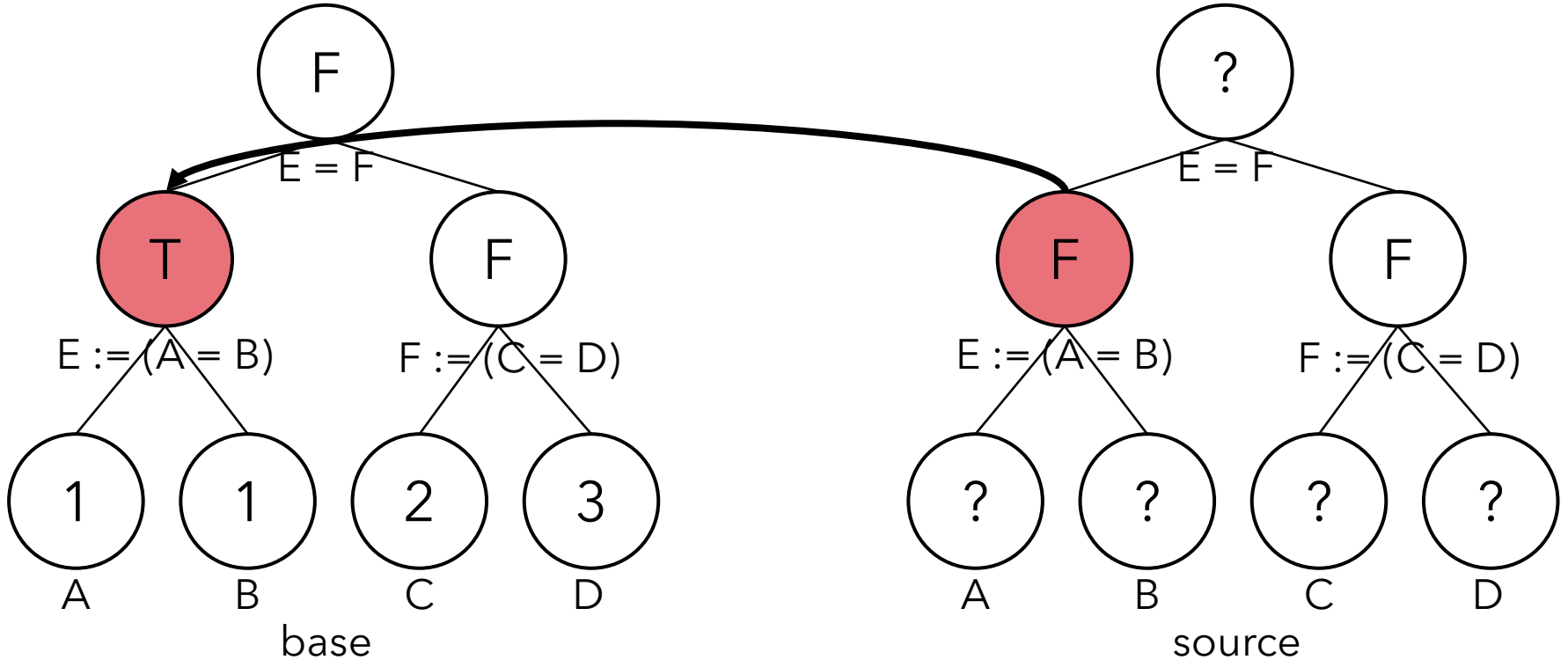
Change value

Intervention should change output

Rule out alternatives

Can't be explained by a different intervention

# Revisiting hierarchical equality




Change variable

b and s should have different values for x

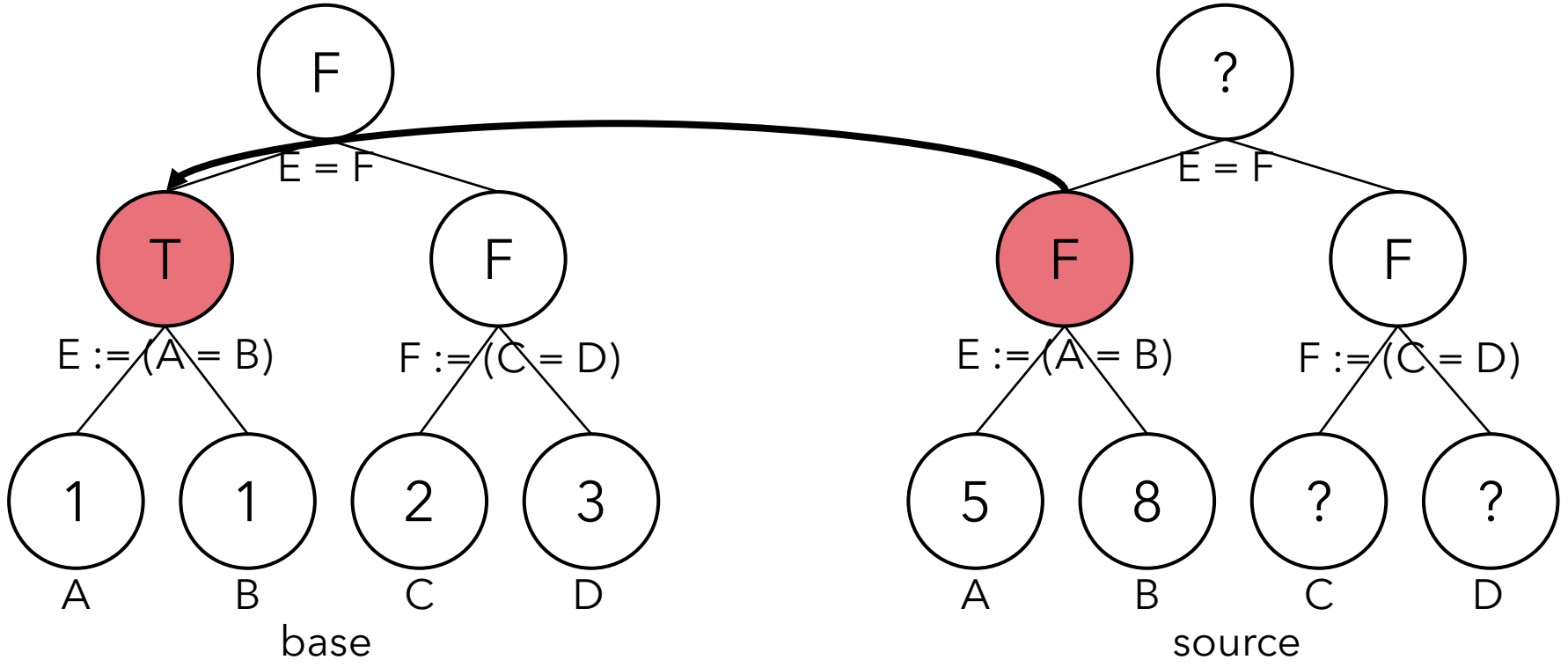
Change value

Intervention should change output

Rule out alternatives

Can't be explained by a different intervention

# Revisiting hierarchical equality




Change variable

b and s should have different values for x

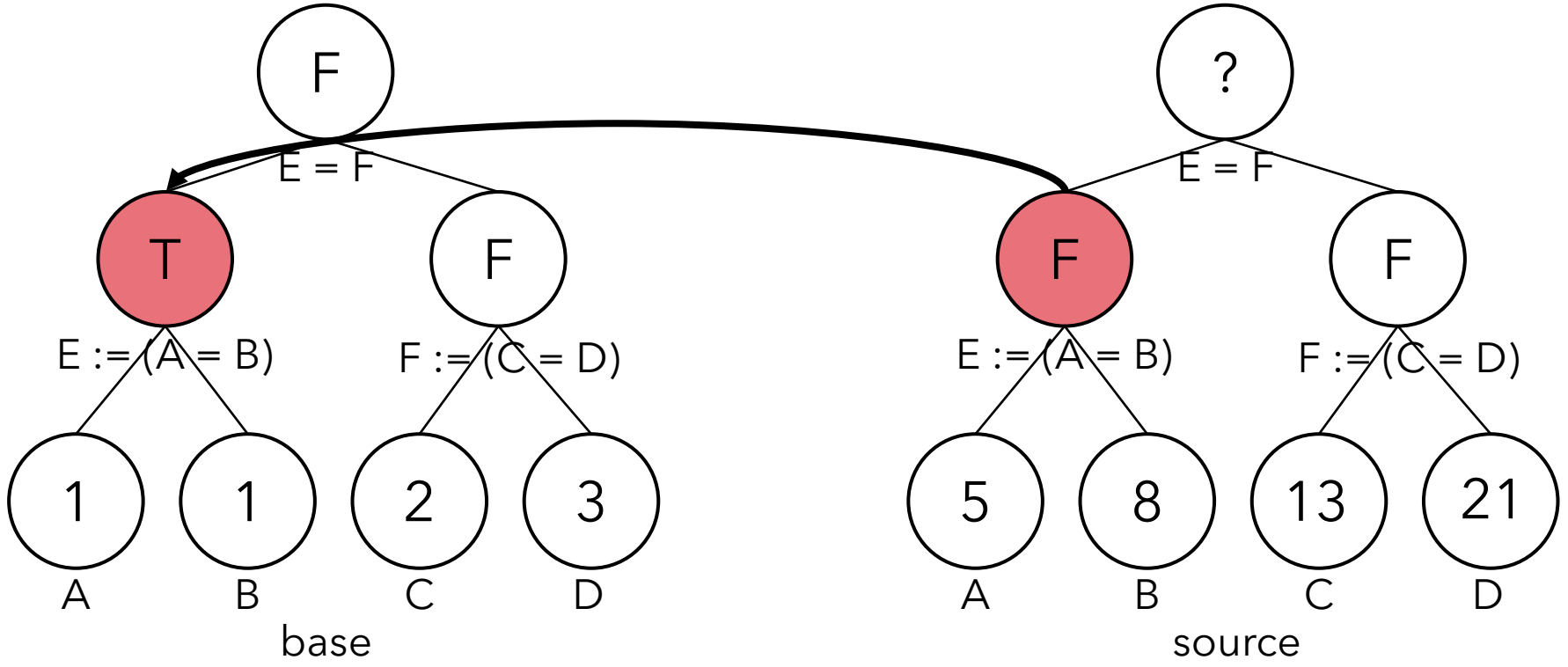
Change value

Intervention should change output

Rule out alternatives

Can't be explained by a different intervention

# Revisiting hierarchical equality




Change variable

b and s should have different values for x

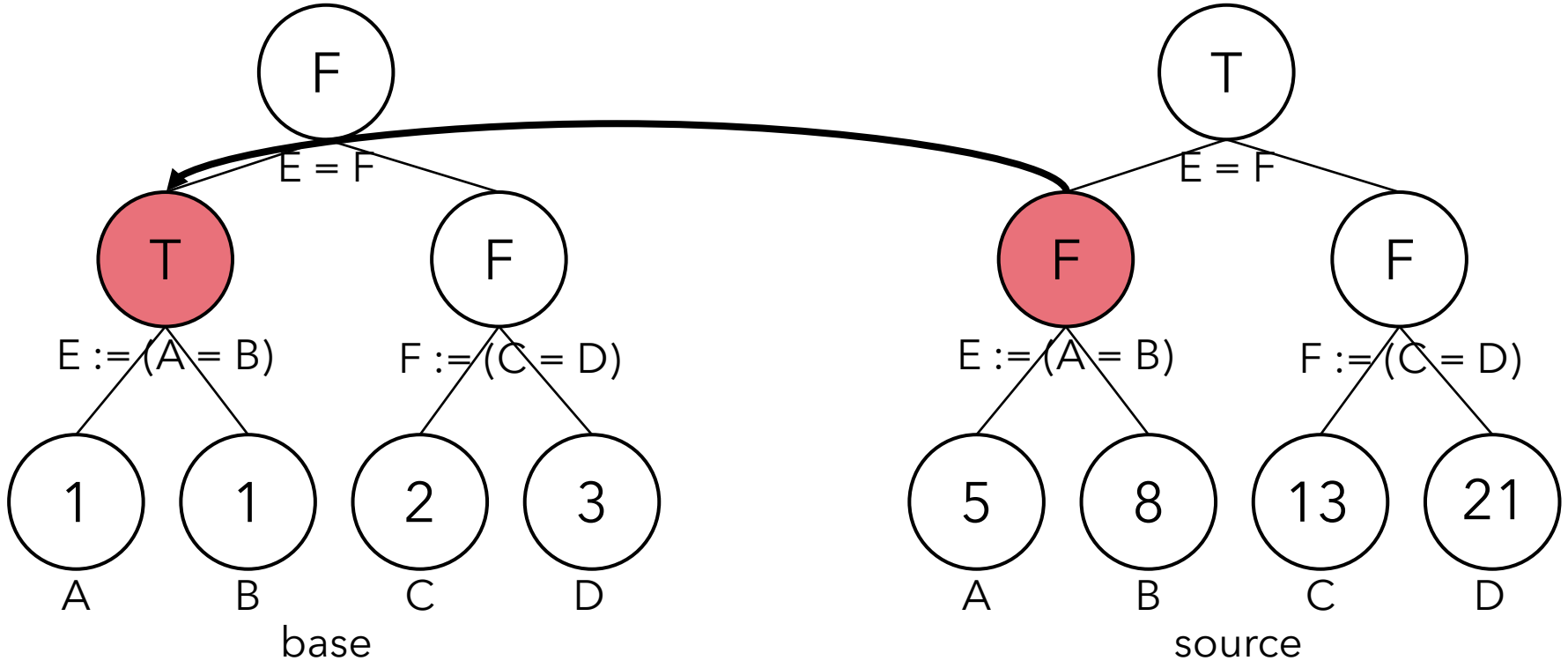
Change value

Intervention should change output

Rule out alternatives

Can't be explained by a different intervention

# Revisiting hierarchical equality




Change variable

b and s should have different values for x

Change value

Intervention should change output

Rule out alternatives

Can't be explained by a different intervention

# What happens if we try different pairs?

## random

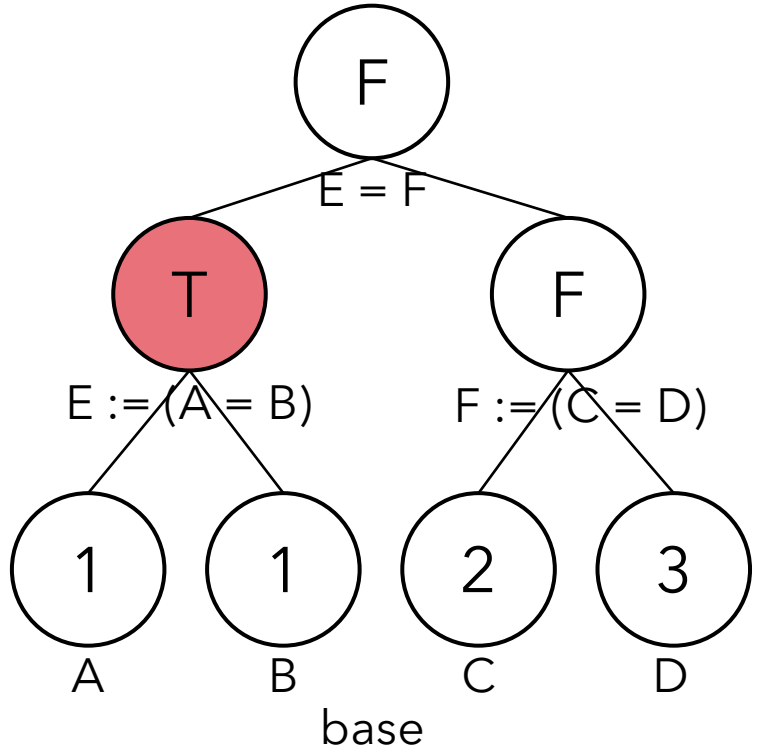
- Sample different pairs of inputs

## target E

- Sample inputs with different values for E

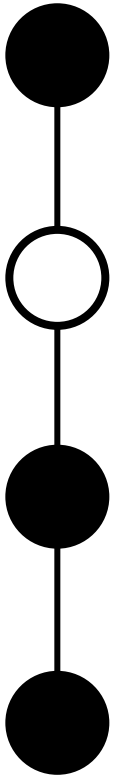
## isolate E

- Sample inputs with different values for E and the same value for F



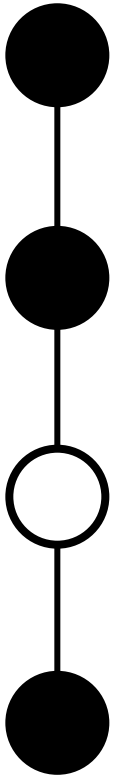
# What happens if we try different pairs?

# Lecture overview



- example: indirect objects
- theory: designing counterfactuals
- example: MCQA
- example: entity binding

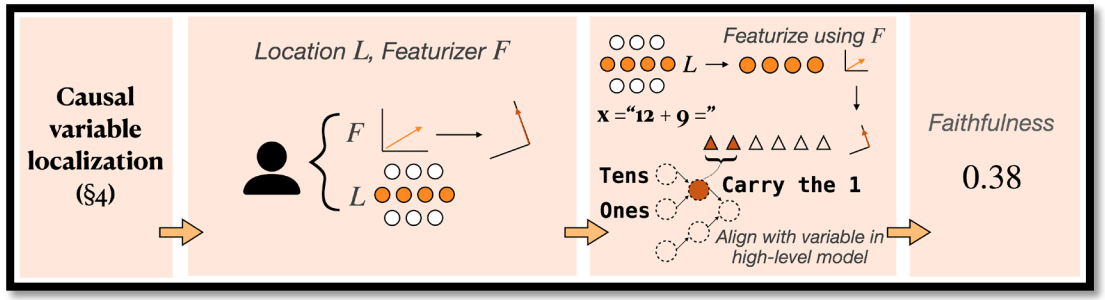
# Lecture overview



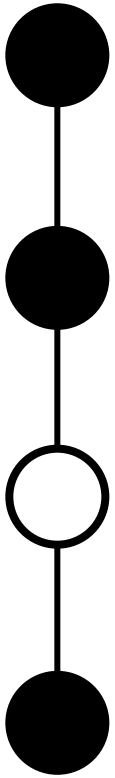
- example: indirect objects
- theory: designing counterfactuals
- example: MCQA
- example: entity binding

# MCQA example

exercise on workbench

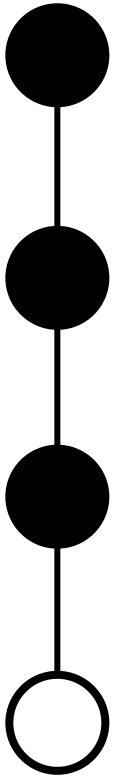


# Lecture overview



- example: indirect objects
- theory: designing counterfactuals
- example: MCQA
- example: entity binding

# Lecture overview



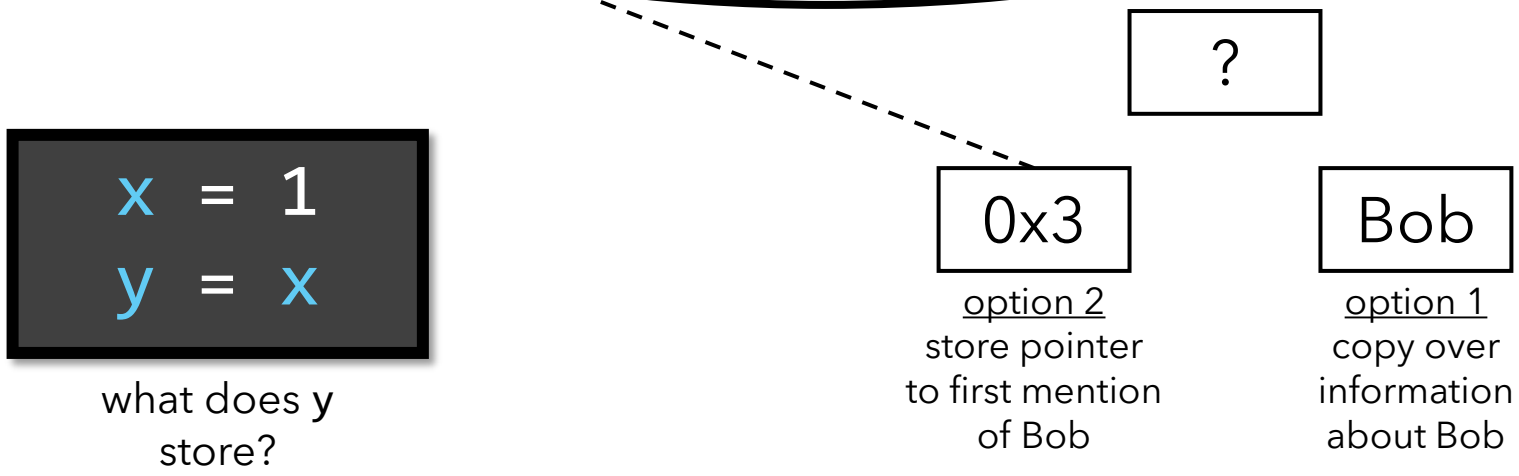
- example: indirect objects
- theory: designing counterfactuals
- example: MCQA
- example: entity binding

# Entity binding

How do models track relations between variables?

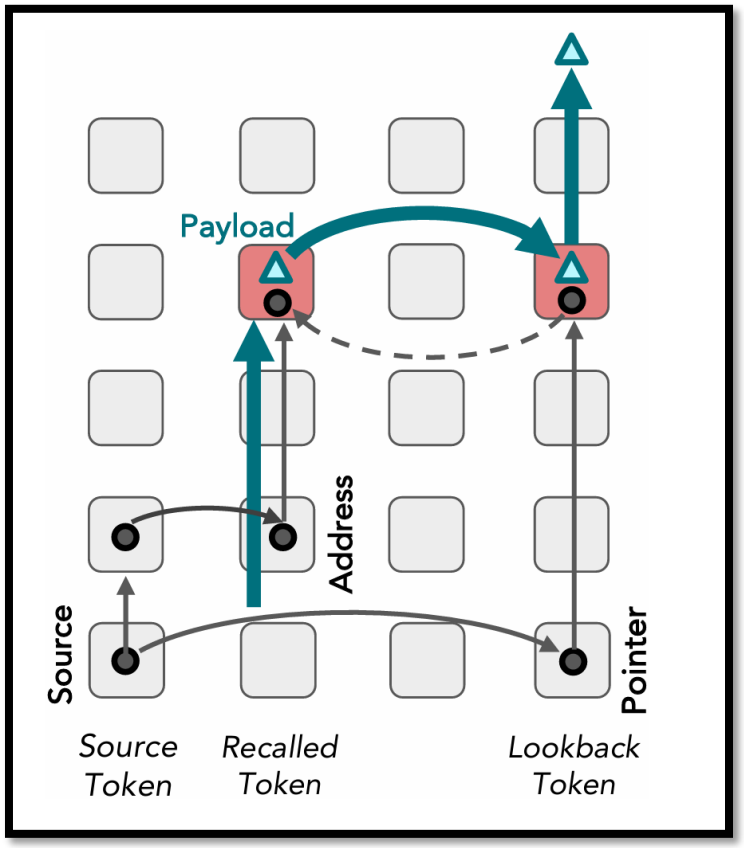
# Tracking memory: indirect object

Alice and Bob went to the library. Bob gave a book to _____



# Tracking memory: entity binding

Alices likes dogs. Bob likes cats. We know that Bob likes _____




0x3

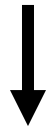
store pointer to first mention of Bob

Prakash et al. 2025 Language Models Use Lookbacks to Track Beliefs

# Tracking memory: entity binding

source

We know that Bob likes cats

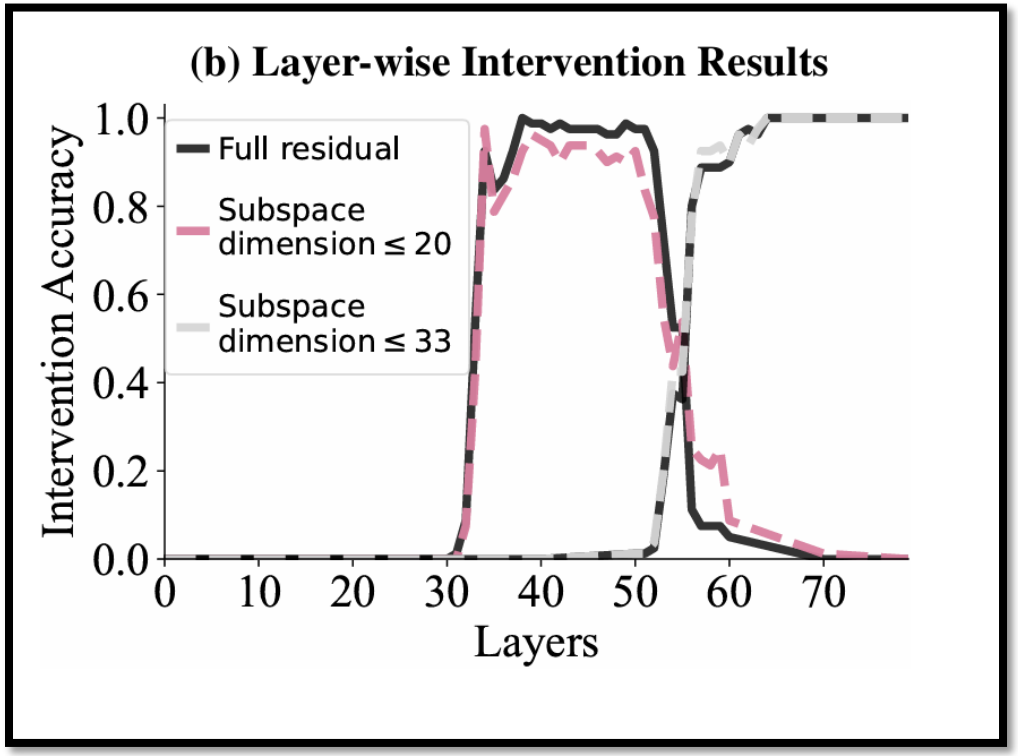


base

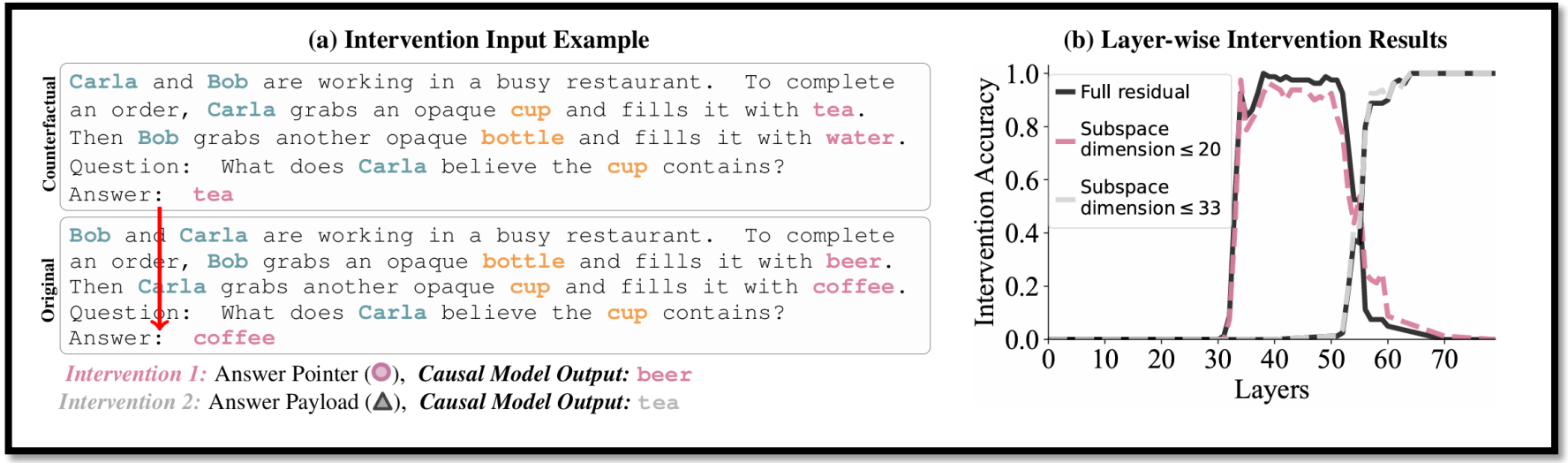
Alices likes dogs. Bob likes cats. We know that Bob likes ~~cats~~ dogs

# Let's try it out!

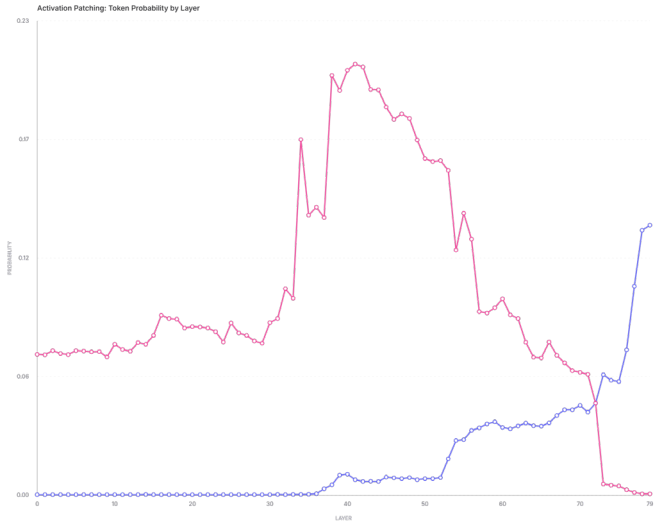
exercise on workbench



# Replicated lookback mechanism!








Prakash et al. 2025 Language Models Use Lookbacks to Track Beliefs

# What about multiple mechanisms?

- It's often the case that we want to test different hypotheses
- The *same* (base, source) pair could test different hypotheses
- Hypotheses predict *different* counterfactual outputs for the *same* intervention

# Mixture of mechanisms: entity binding

- Where is the pointer pointing?

Ann loves ale. Joe loves jam. Pete loves pie.
We know that Ann loves _____

| Positional | Lexical | Reflexive |
| --- | --- | --- |
| • referent stored at Ann's *position* | • referent stored at object closest to Ann | • referent stored at Ann's token |

Gur-Arieh et al. 2025 Mixing Mechanisms

# Mixture of mechanisms: entity binding

source

Joe loves ale. Ann loves pie. Pete loves jam.
We know that Ann loves _____

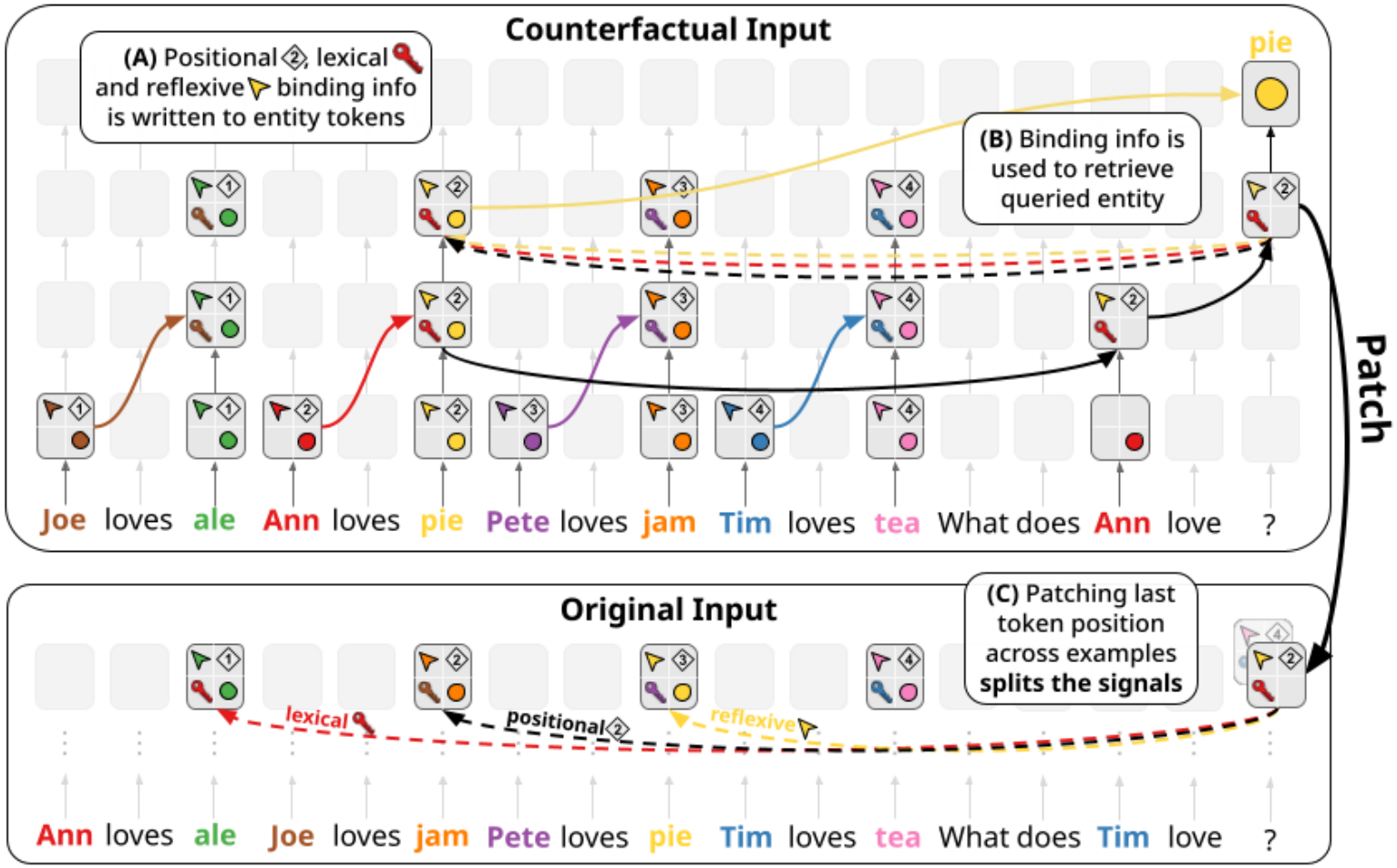
base

L P R
Ann loves ale. Joe loves jam. Pete loves pie.
We know that Ann loves _____

| Positional | Lexical | Reflexive |
| --- | --- | --- |
| • referent stored at Ann's *position* | • referent stored at object closest to Ann | • referent stored at Ann's token |

Gur-Arieh et al. 2025 Mixing Mechanisms

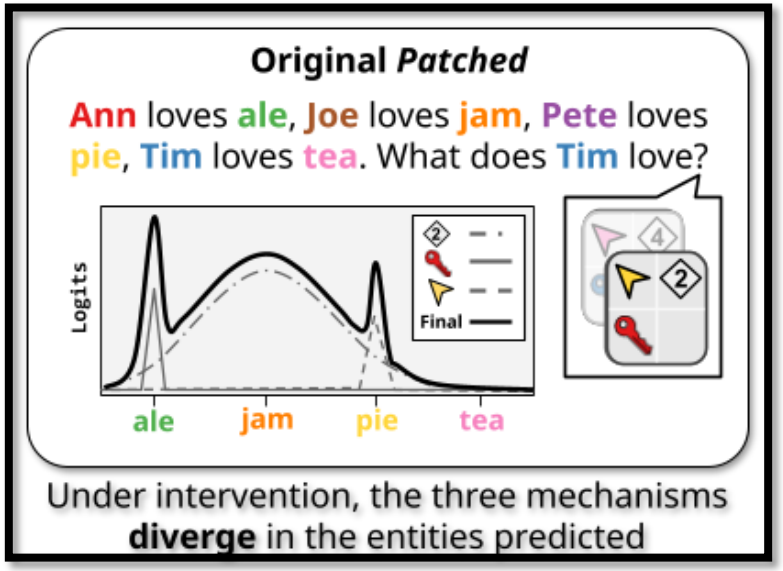
# Mixture of mechanisms: entity binding



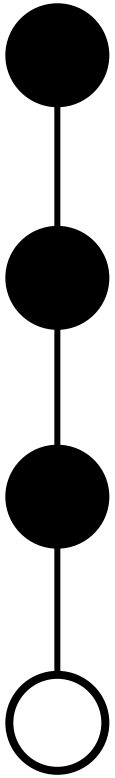

Gur-Arieh et al. 2025 Mixing Mechanisms

# Let’s try it out!

exercise on workbench



# Lecture overview



- example: indirect objects
- theory: designing counterfactuals
- example: MCQA
- example: entity binding

# Lecture overview

- example: indirect objects
- theory: designing counterfactuals
- example: MCQA
- example: entity binding

# Conclusion

## Testing hypothesis through counterfactual pairs

- Change variable value, change output, rule out alternatives

## The same counterfactual pair can distinguish different hypotheses

- Because they predict different outputs for the same intervention

## We apply causal abstraction through constructing counterfactuals

- This can help us discover algorithmic primitives, like pointer-reference mechanism in transformers!

# Course timeline