



Gary Marcus ✅
@GaryMarcus

AI is not hitting a wall.

But LLMs probably are (or at least a point of diminishing returns).

We need new approaches, and to diversify the which roads are being actively explored.

The algebraic mind

Relations between Variables



```
total_cost = item_cost * number_ordered
```

`total_cost` : $\mathbb{Z} \times \mathbb{Z} \rightarrow \mathbb{Z}$

`total_cost(7, 5) = 35`

`total_cost(7, 6) = 42`

Outline

1. Marcus' line of argument

2. Critique of his argument

3. Contemporary research

4. Discussion

Outline

1. Marcus' line of argument

- UQOTOM

2. Critique of his argument

- Marcus' critique on MLPs and backpropagation
- Marcus' proposed alternative

3. Contemporary research

4. Discussion

Outline

1. Marcus' line of argument

- **UQOTOM**

2. Critique of his argument

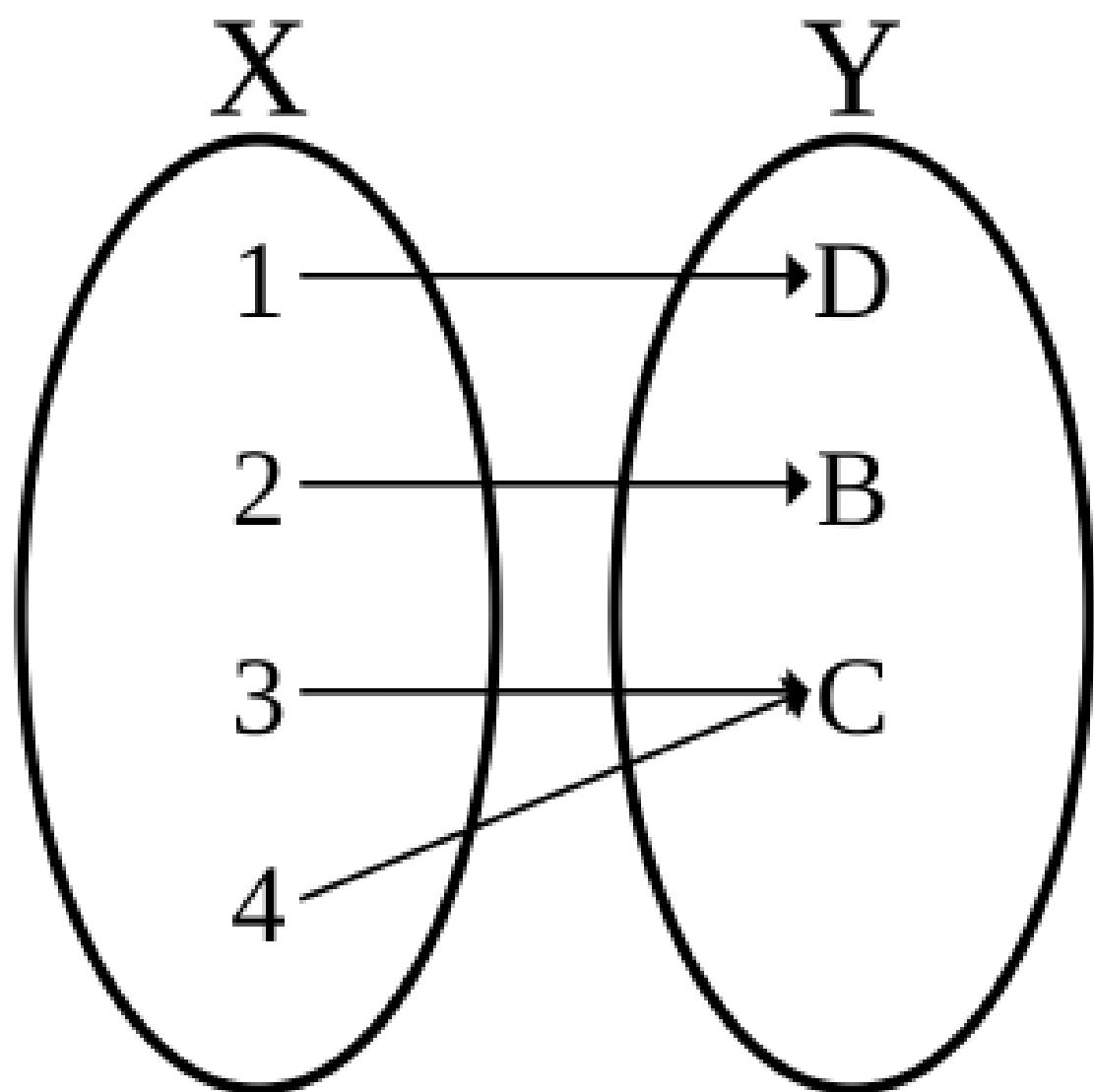
- Marcus' critique on MLPs and backpropagation
- Marcus' proposed alternative

3. Contemporary research

4. Discussion

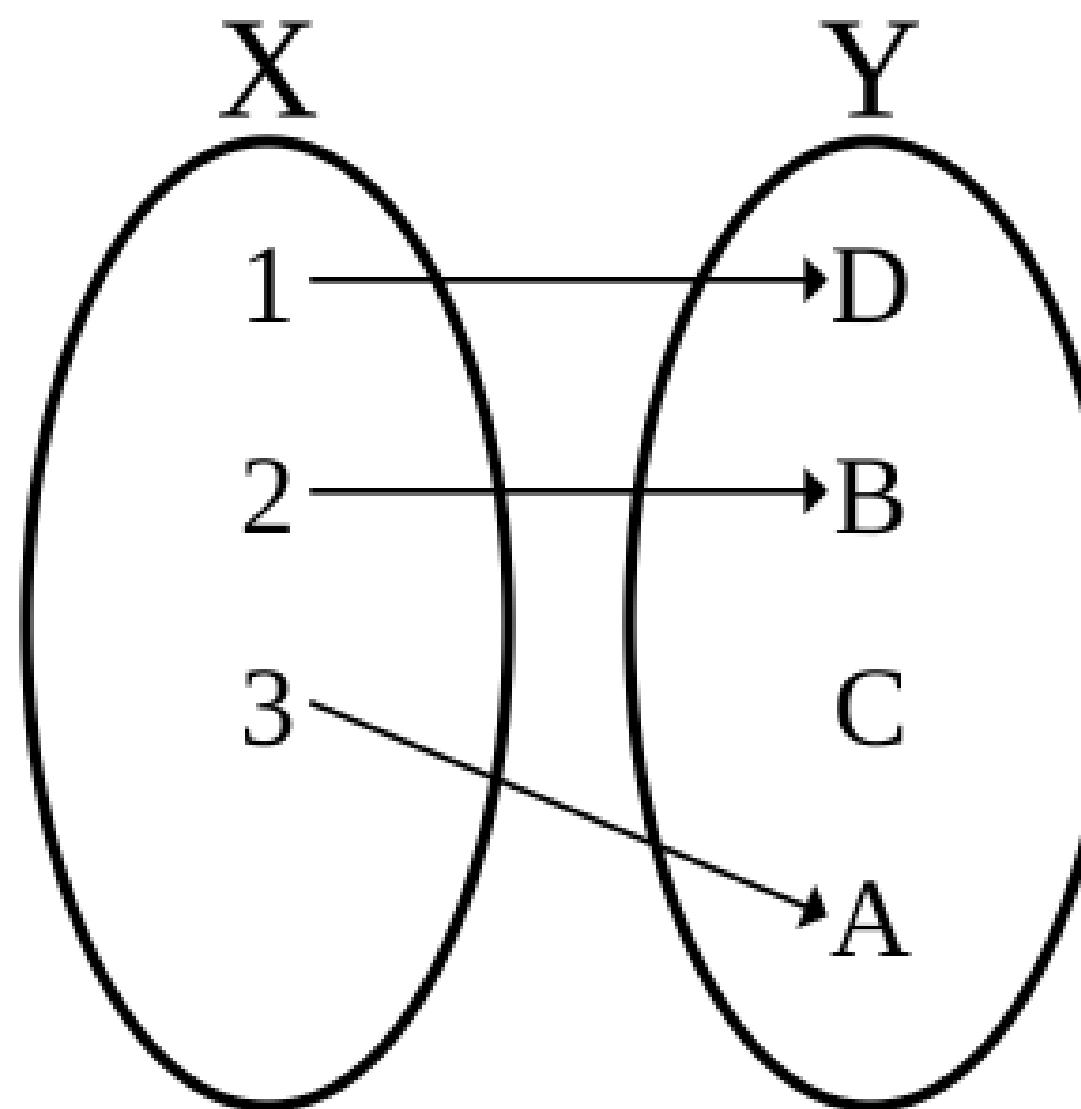
UQOTOM

Universally Quantified One-To-One Mappings



Universally Quantified

Function is defined for all elements in a set.

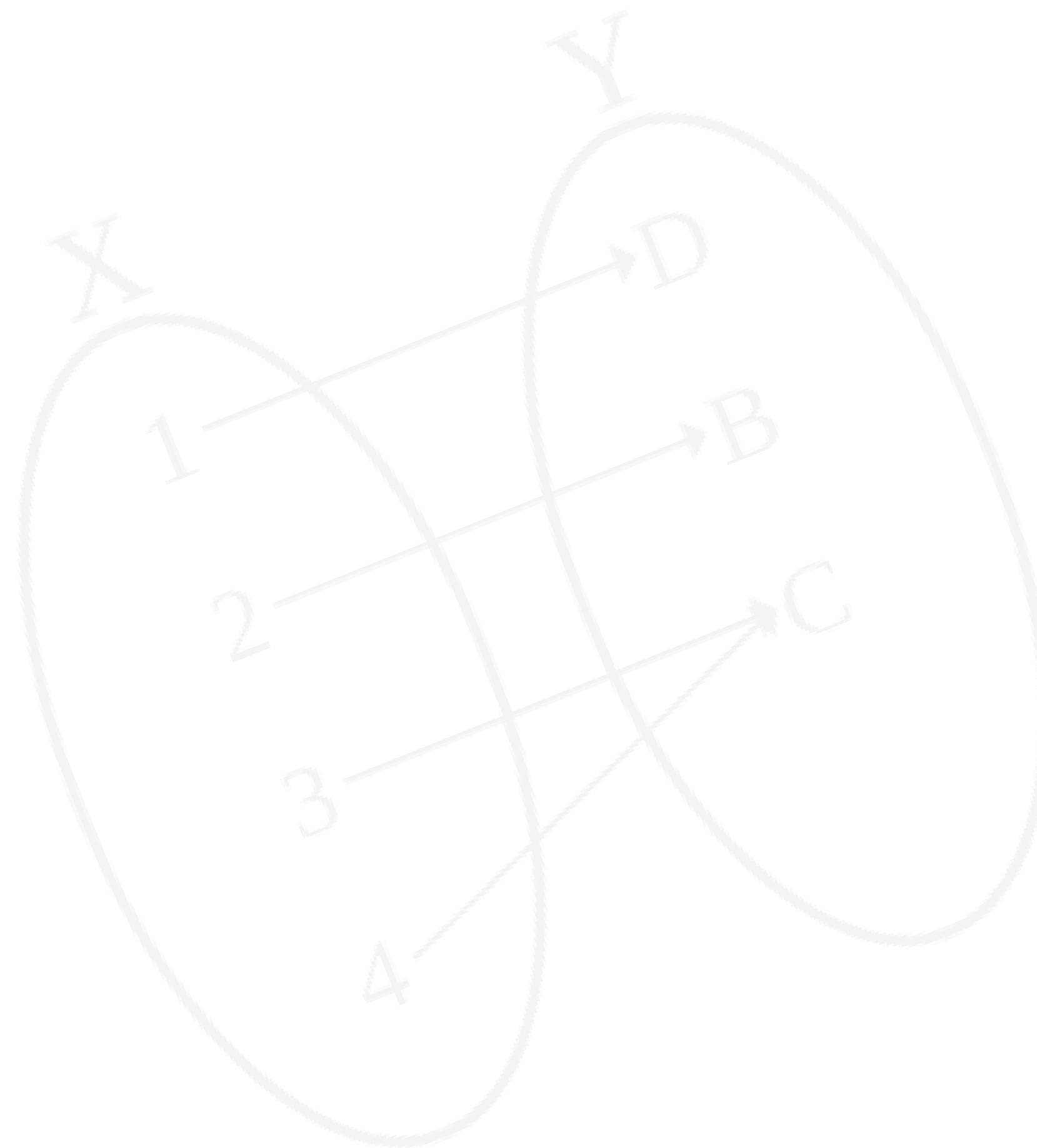


One-To-One

Each unique input has a unique output.

Quiz

UQOTOM or not?



Duplication

$$f : \Sigma^* \xrightarrow{\quad} \Sigma^*$$

$$w \mapsto ww$$

Multiplication

$$f : \mathbb{Q} \rightarrow \mathbb{Q}$$

$$q \mapsto \alpha \cdot q, \quad \alpha \in \mathbb{Q}$$

Identity

$\text{id} : \mathcal{X} \mapsto \mathcal{X}$

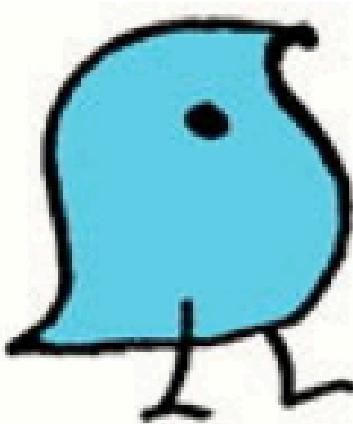
$x \mapsto x$

Live Test

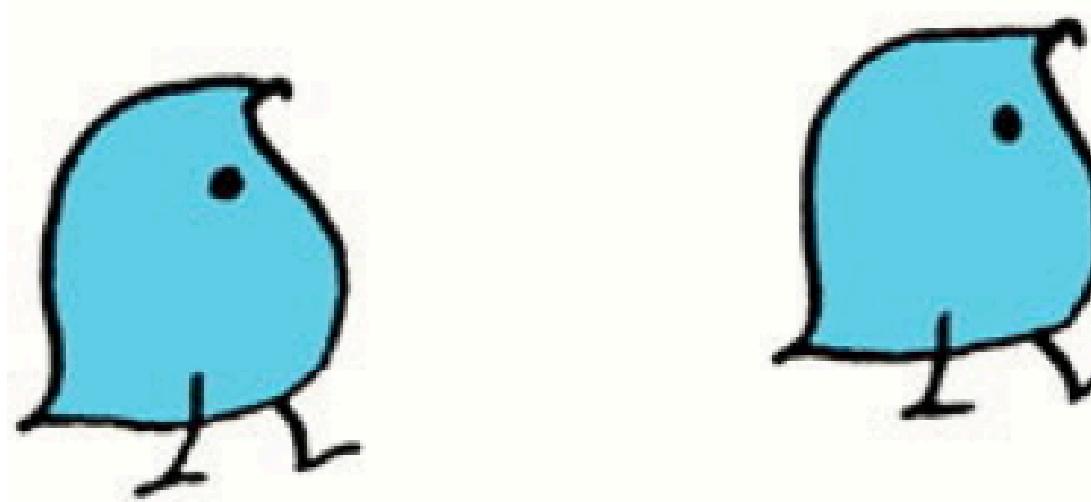
- “There is ample evidence, I think, that **people can generalize** universally quantified one-to-one mappings”

Input	Output
1010	1010
0100	0100
1110	1110
0000	0000
1111	?

Does anyone know what a **wug** is?



This is a Wug.



Now there is another one.
There are two of them.
There are two _____.[©]

Photo courtesy of Jean Berko Gleason
<https://thelanguagenerds.com/2025/what-is-a-wug/>

UQOTOM

Universally Quantified One-To-One Mappings

Marcus says:

"I do not mean to suggest that UQOTOM are the only mappings people compute."

- Page 36

*"Because **free generalization of UQOTOM** would **preclude memorization**, evidence that people [...] can freely generalize UQOTOM would be particularly **strong evidence** [...] that people [...] can perform **operations over variables**."*

- Page 36

Why are they relevant?

"Algebraic rules are not finite tables of memorized facts or relationships between specific instances but open-ended relationships that can be **freely generalized** to all elements within some class."

- Page 39 f.

Outline

1. Marcus' line of argument

- UQOTOM

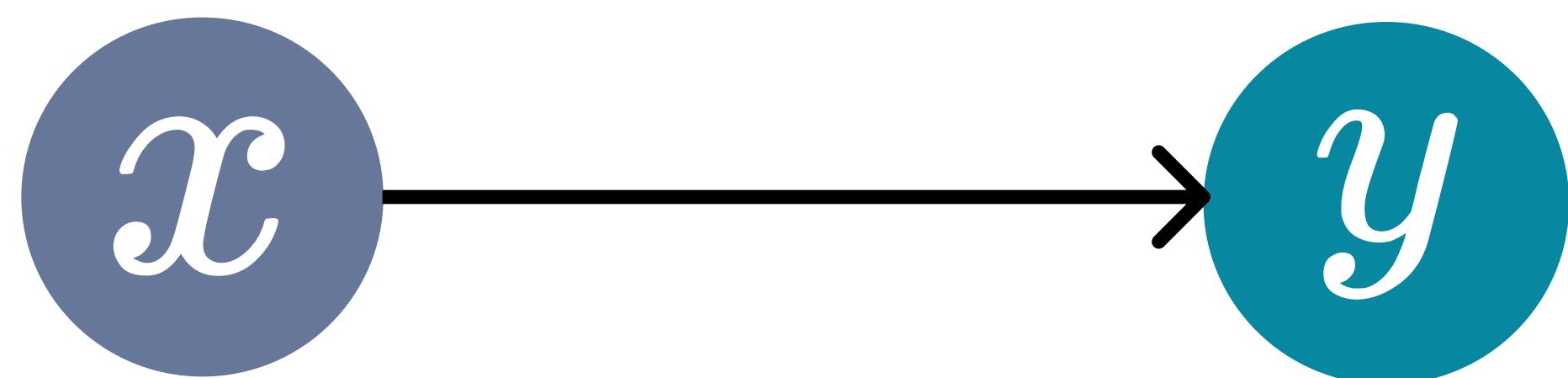
2. Critique of his argument

- Marcus' critique on MLPs and backpropagation
- Marcus' proposed alternative

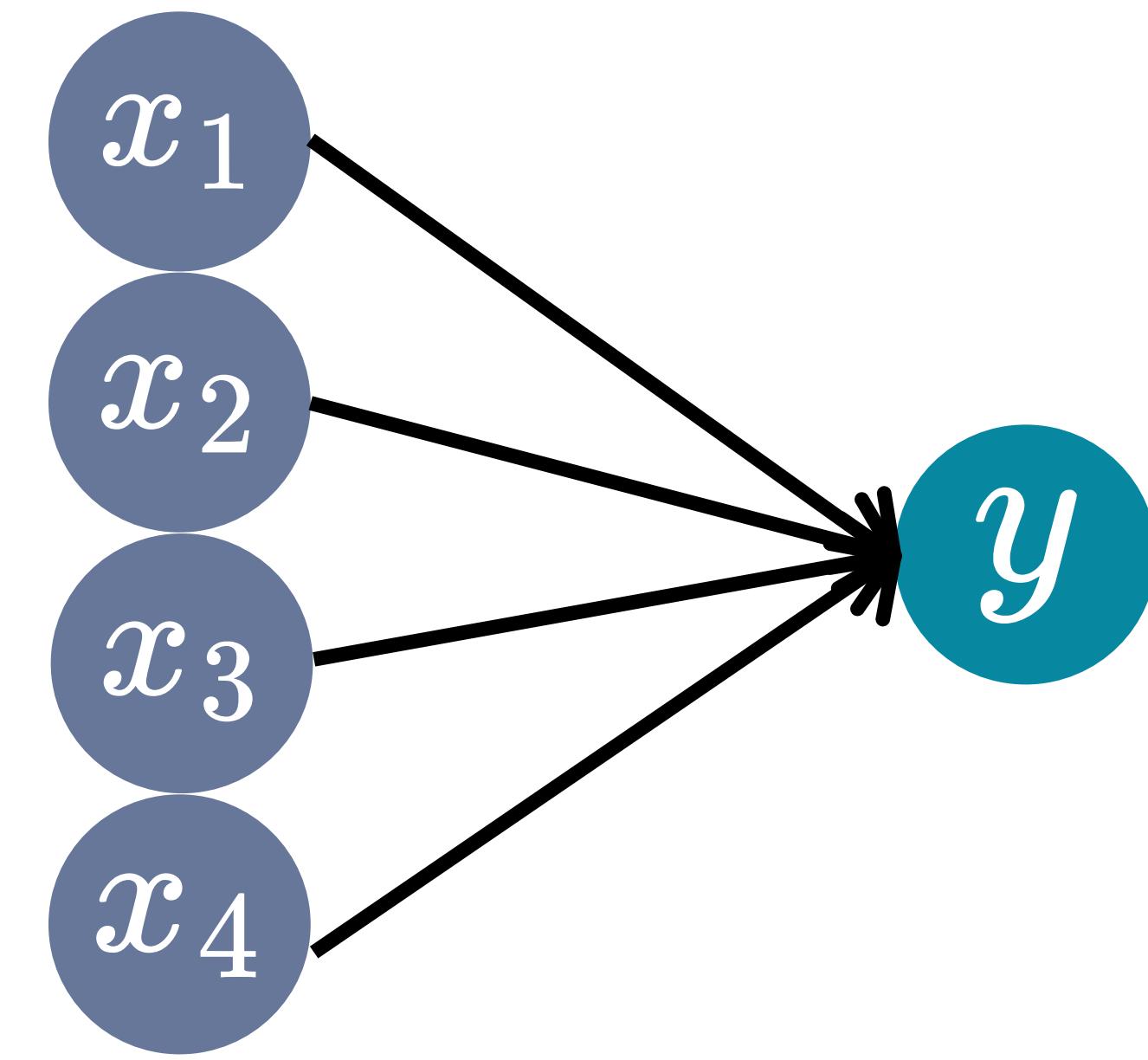
3. Contemporary research

4. Discussion

Two distinct types of MLPs

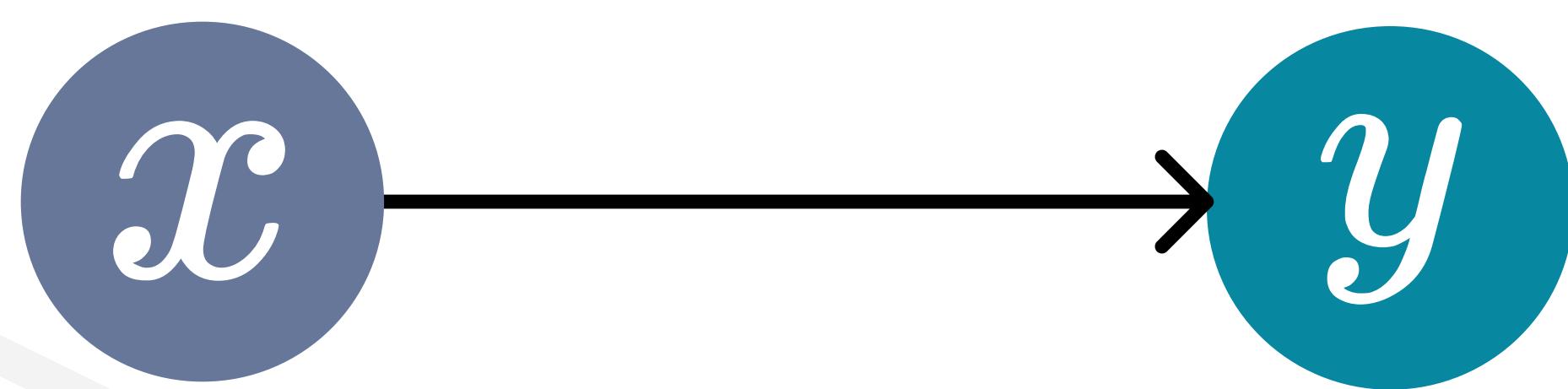


One-Node-Per-Variable Models

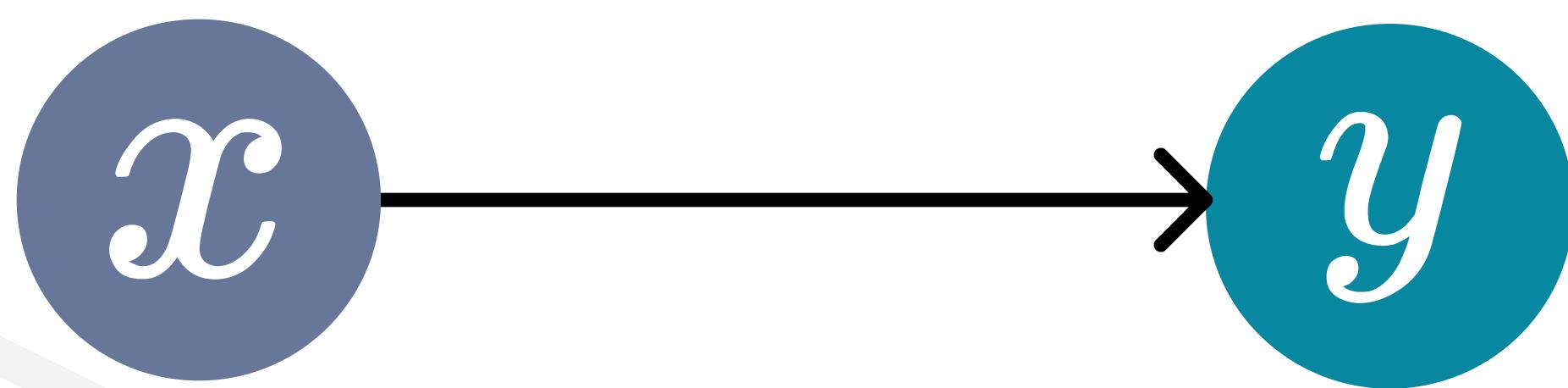


Many-Nodes-Per-Variable Models

One-Node-Per-Variable

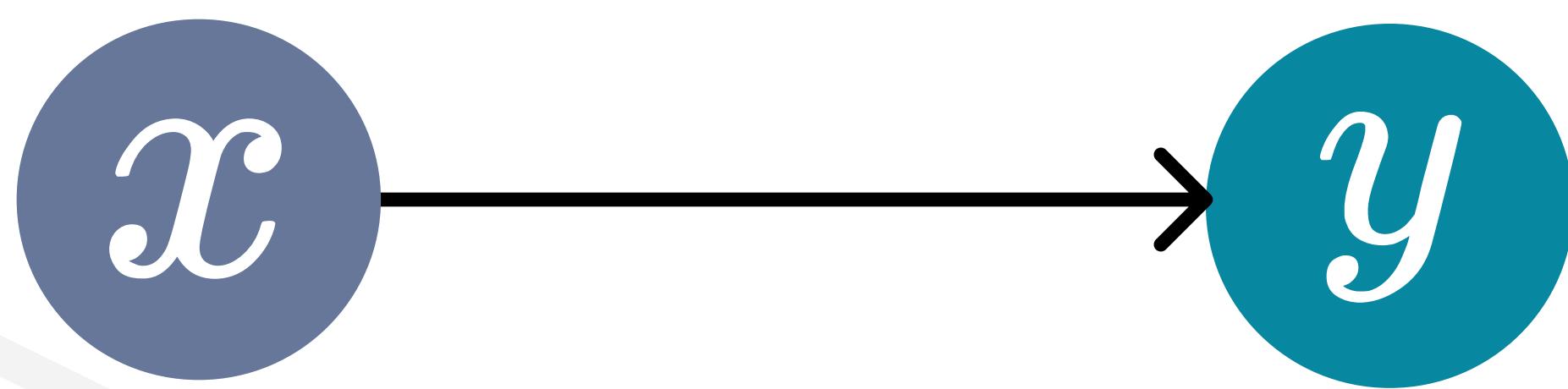


One-Node-Per-Variable



$$f(x) = wx + b$$

One-Node-Per-Variable



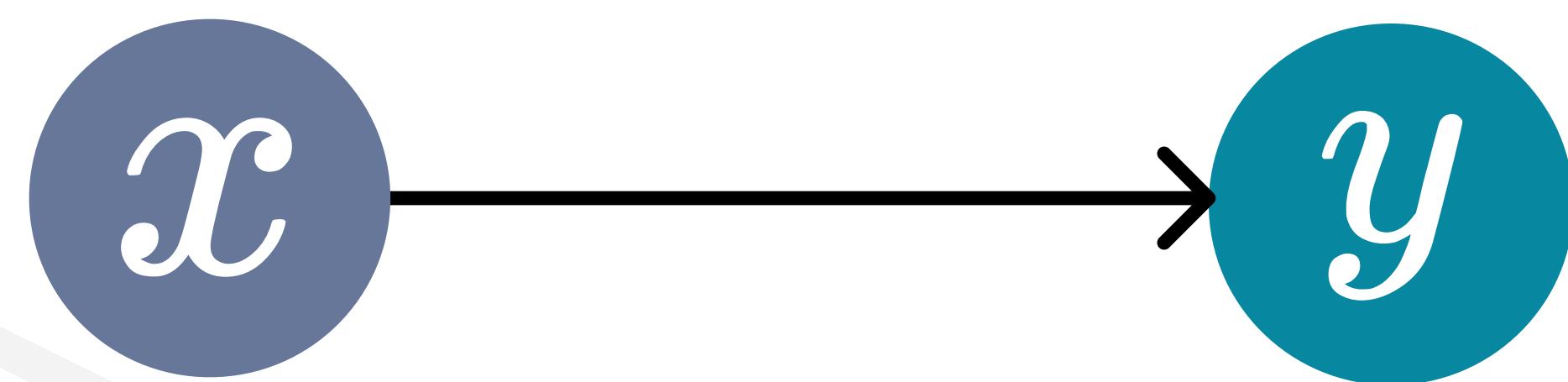
$$f(x) = wx + b$$

This is an UQOTOM!

One-Node-Per-Variable

*[...] I am claiming that some multilayer perceptrons [...] **implement** - rather than eliminate - **algebraic rules**.*

[...] models that allocate a single node to each variable have [...] no choice but to represent abstract relationships between variables, [...]"



$$f(x) = wx + b$$

This is an UQOTOM!

One-Node-Per-Variable

Linearity is necessary

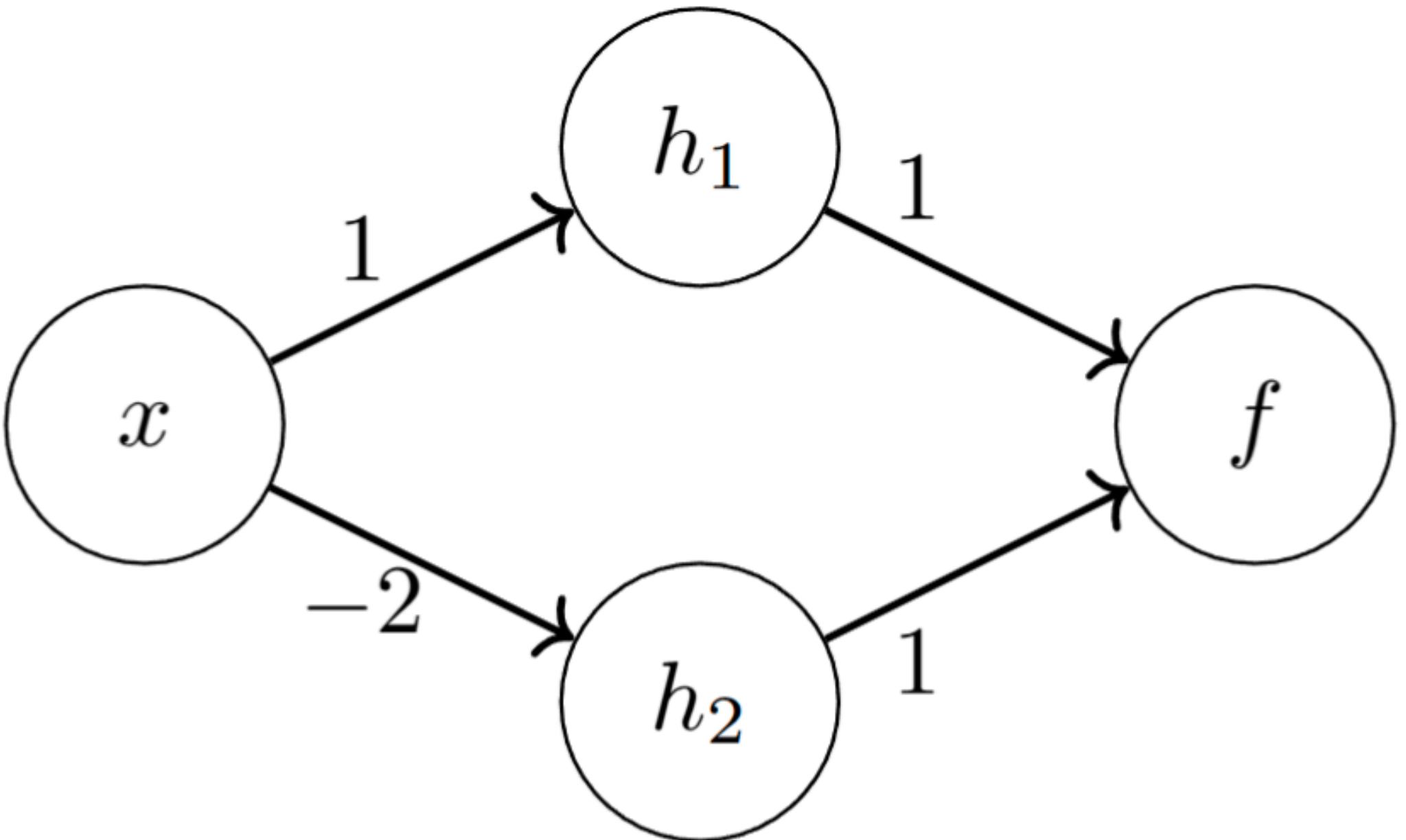
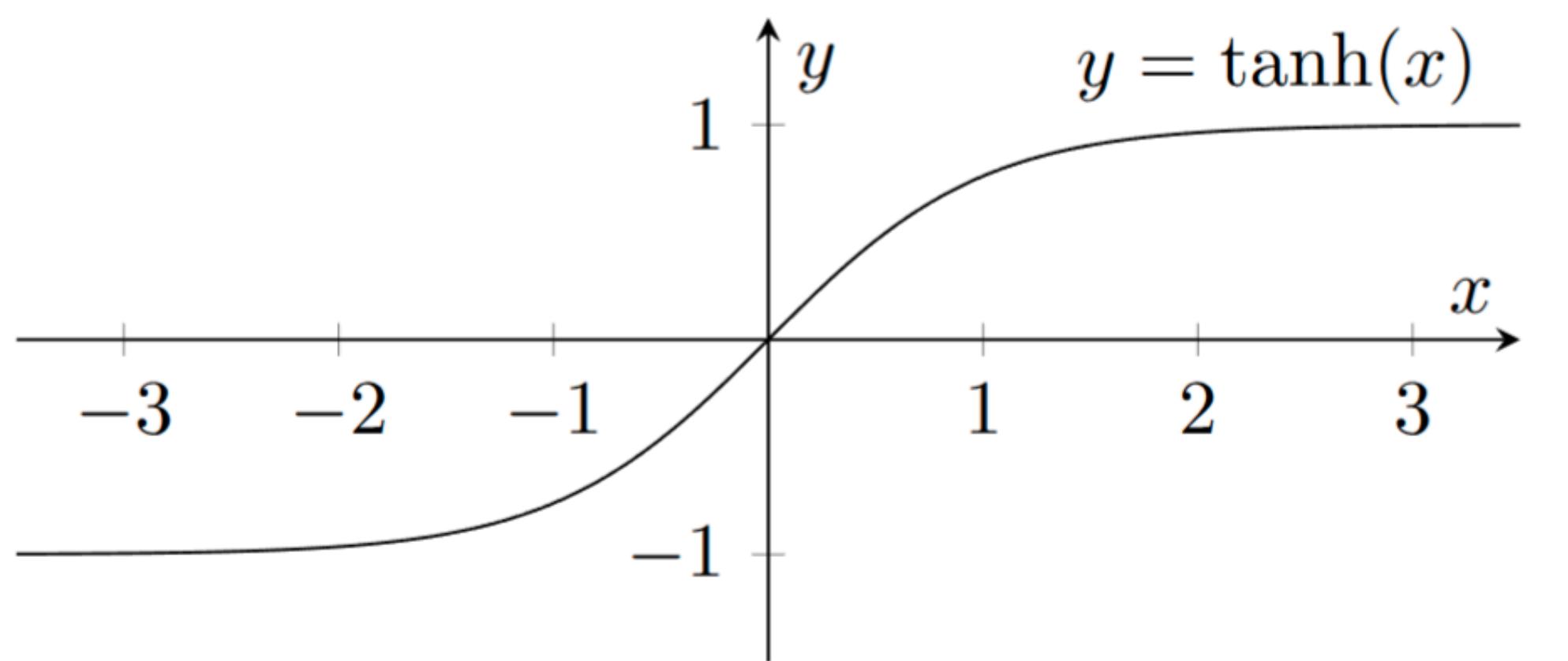


Figure 2: Plot of the function $y = \tanh(x)$.

One-Node-Per-Variable

Linearity is necessary

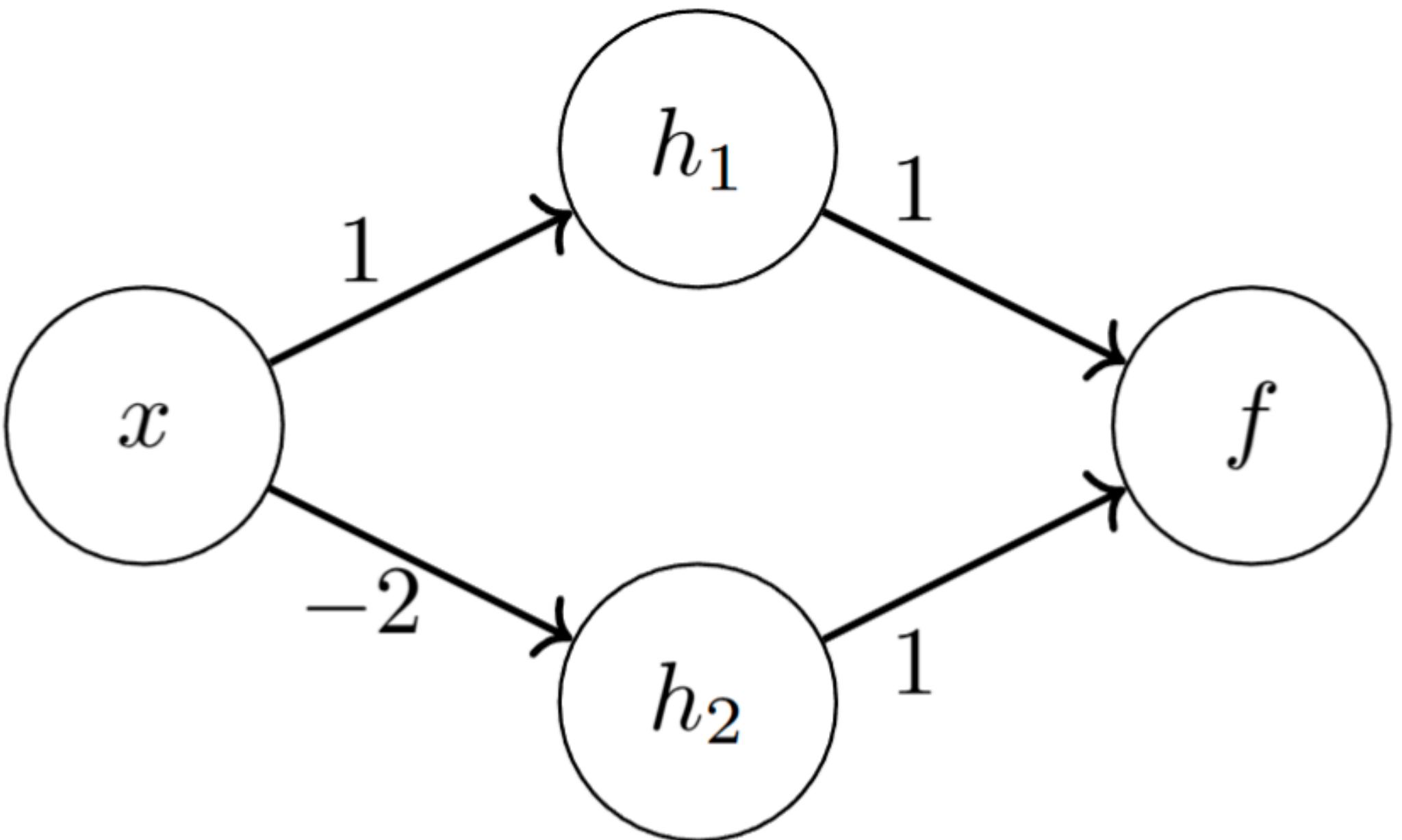
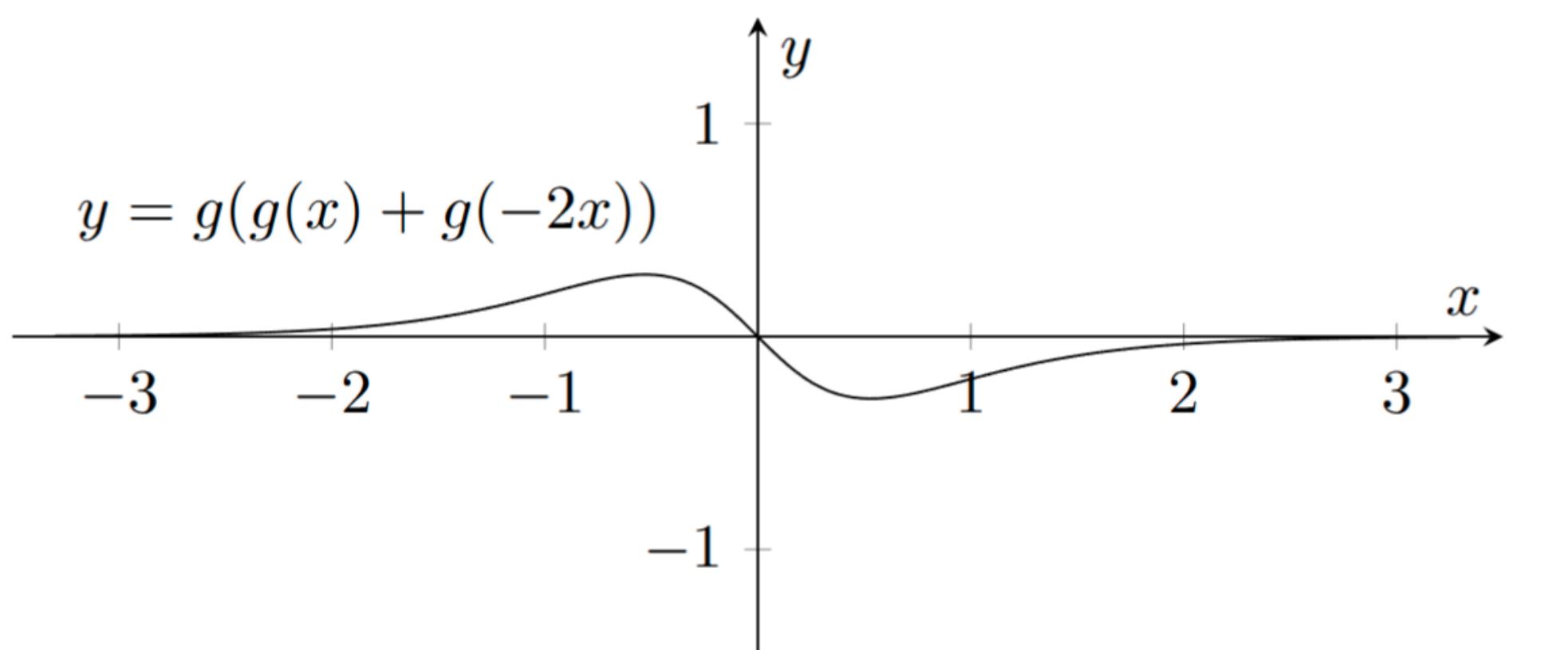
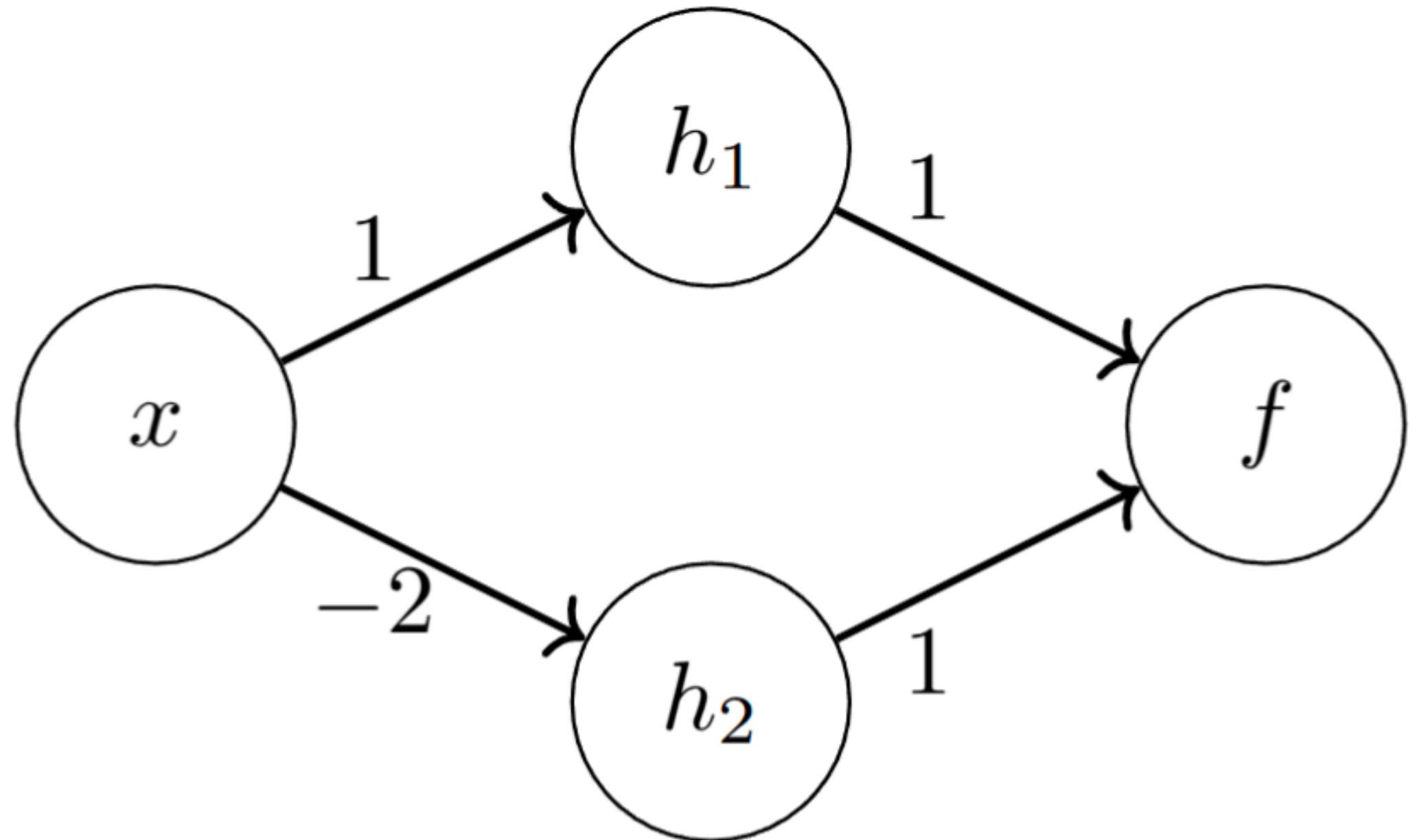
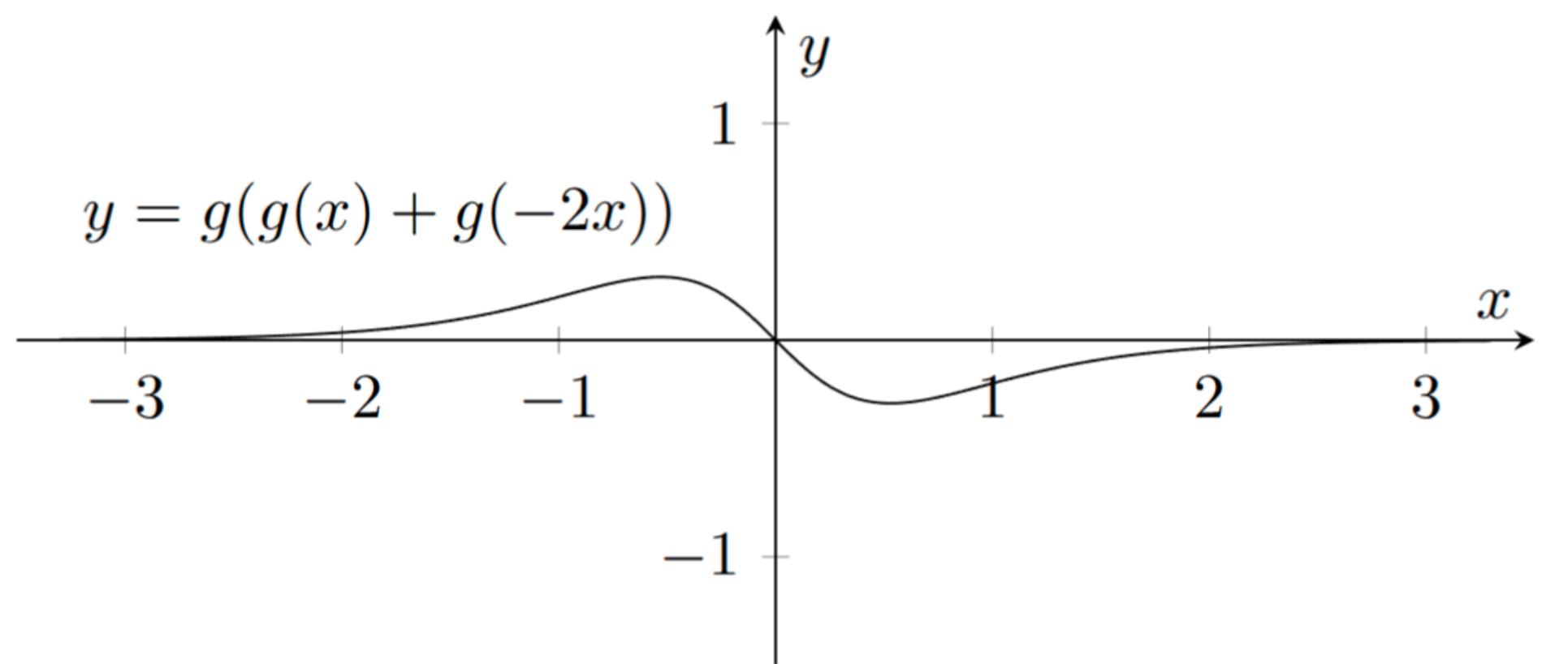


Figure 4: Plot of the function $y = g(g(x) + g(-2x))$ with $g(x) := \tanh(x)$.

One-Node-Per-Variable

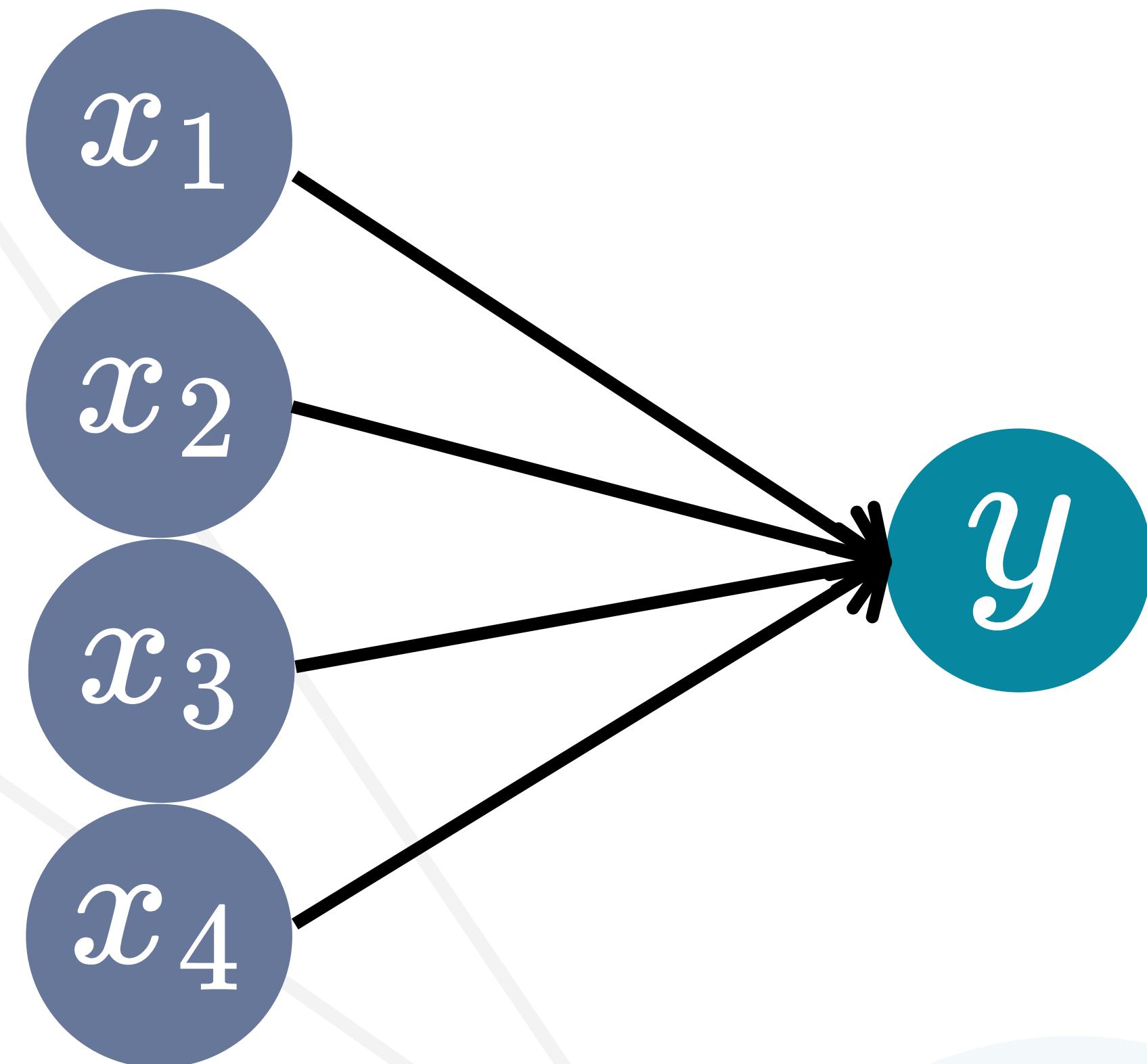
Linearity is necessary



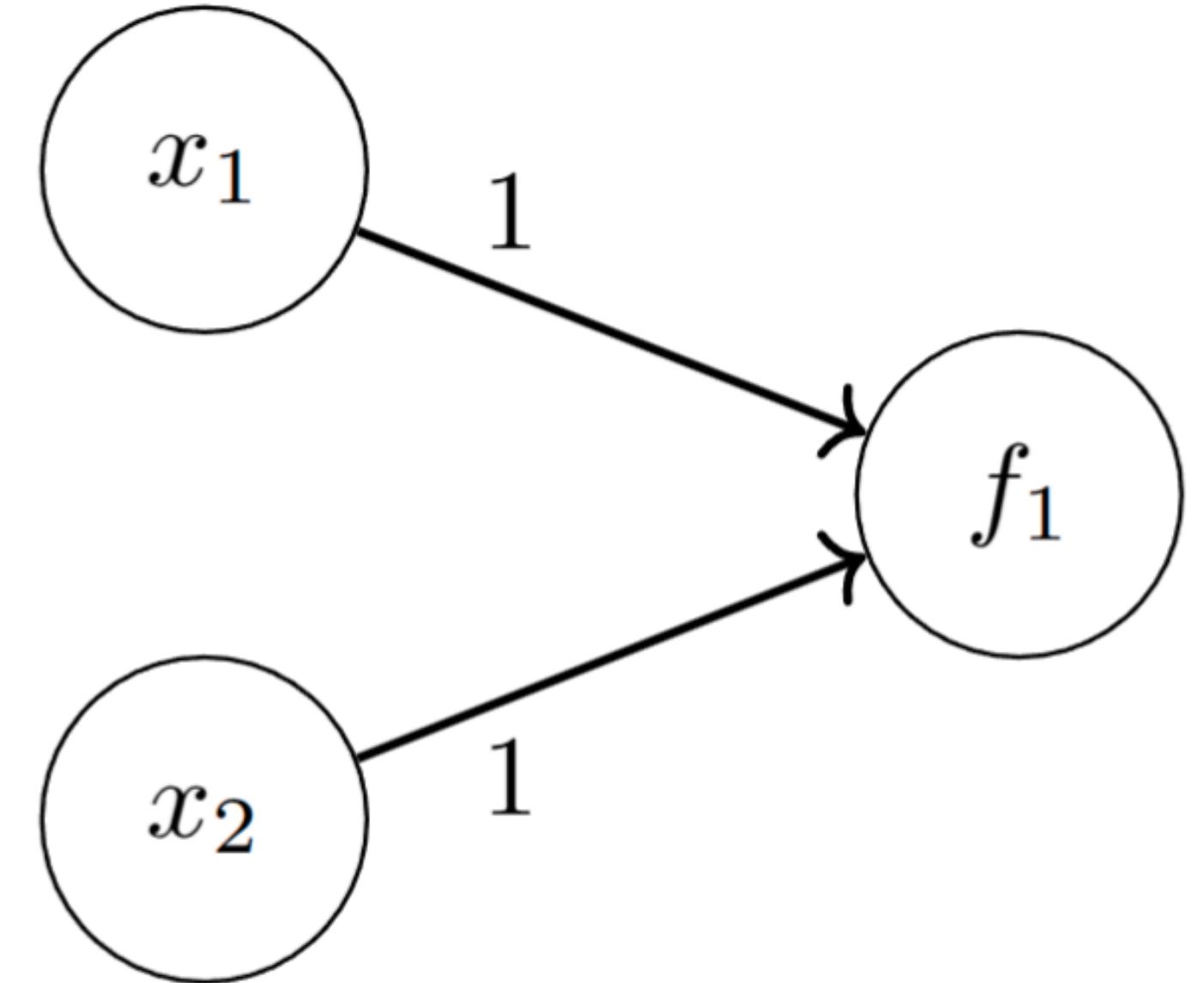
This is not an UQOTOM!

Figure 4: Plot of the function $y = g(g(x) + g(-2x))$ with $g(x) := \tanh(x)$.

Many-Nodes-Per-Variable



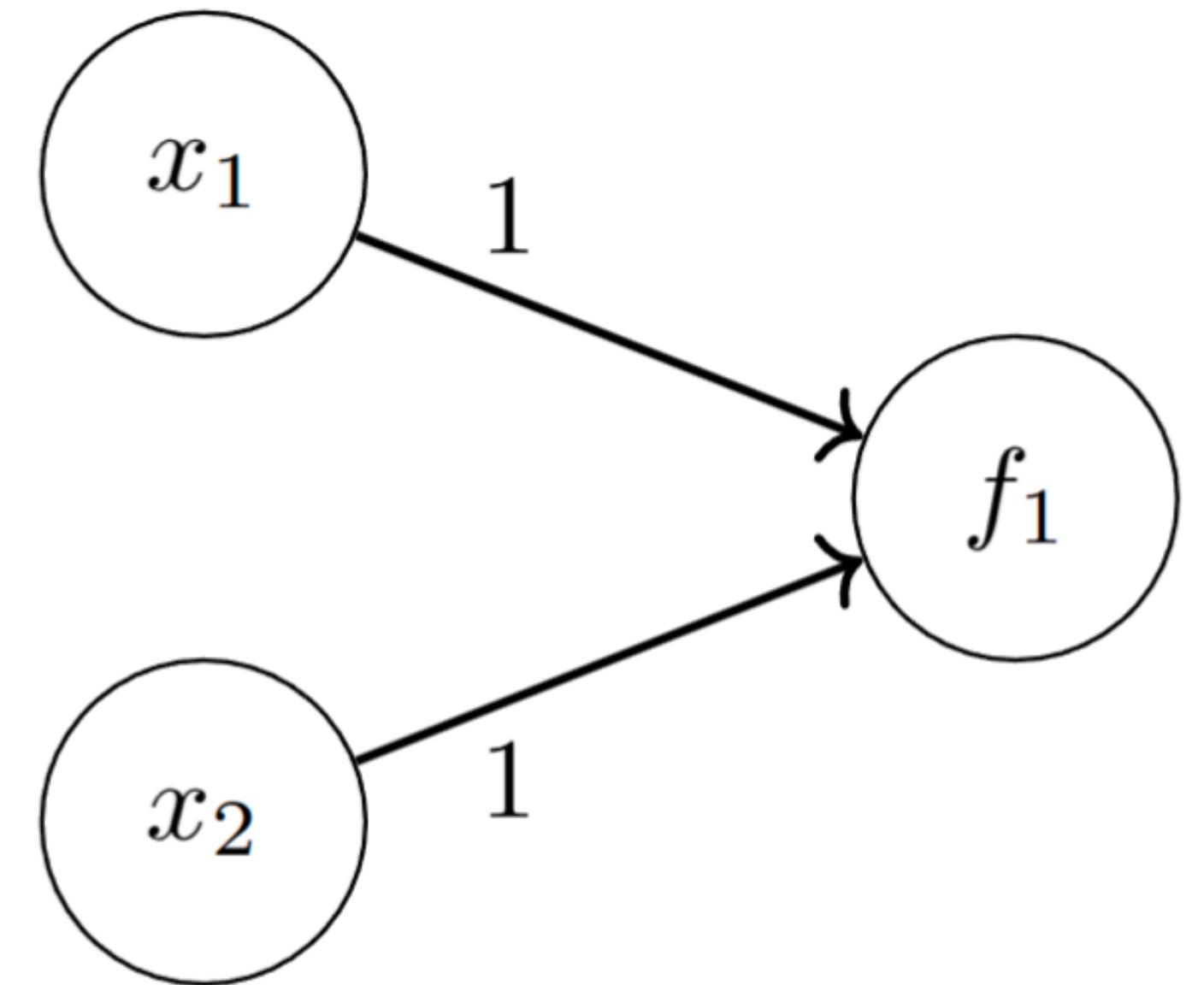
Many-Nodes-Per-Variable



$$f_1(x_1, x_2) = ?$$

Many-Nodes-Per-Variable

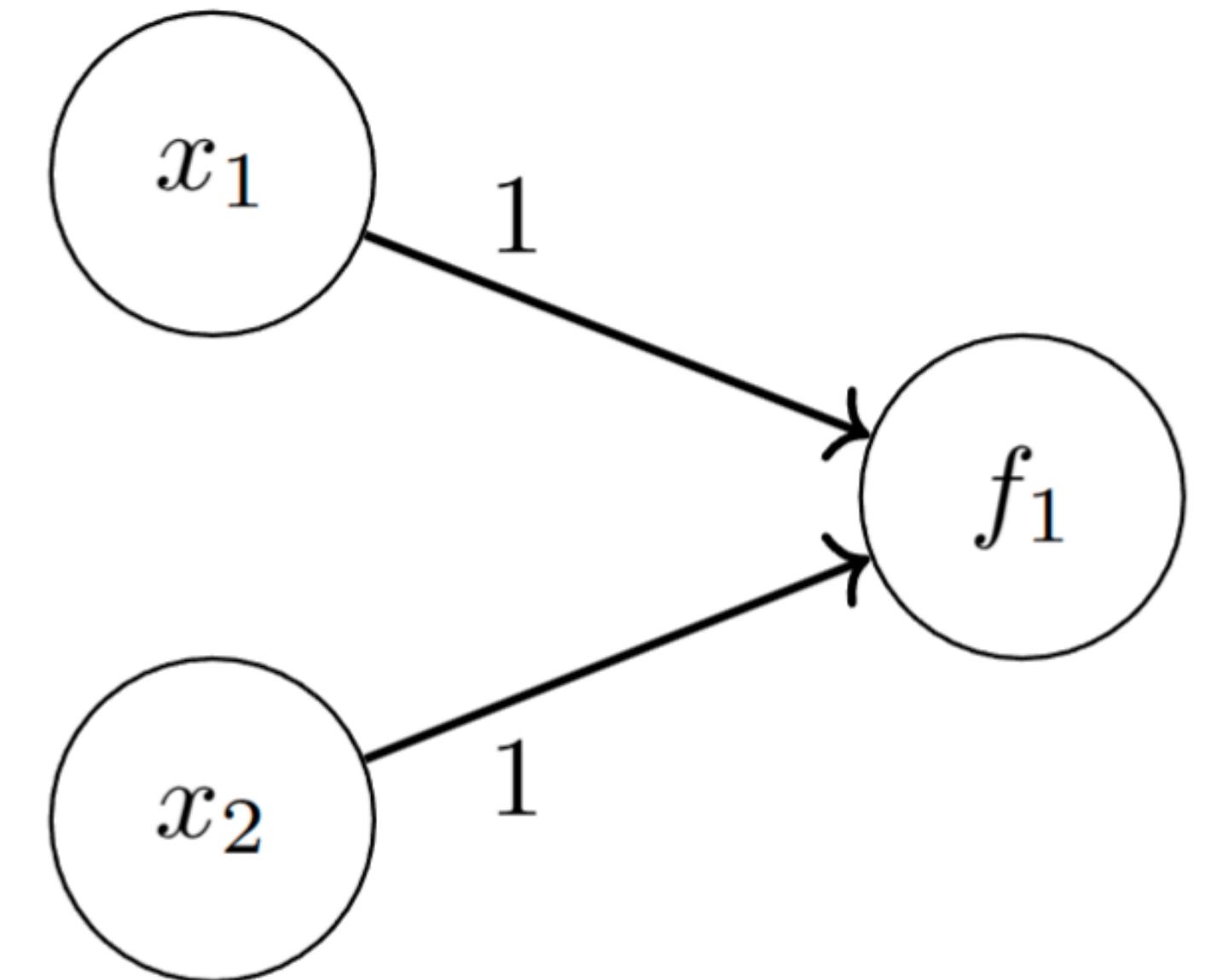
$$f_1(x_1, x_2) = x_1 + x_2$$



Many-Nodes-Per-Variable

They "[...] sometimes represent abstract relationships between variables and sometimes do not [...]"

$$f_1(x_1, x_2) = x_1 + x_2$$



This is not an UQOTOM!

Many-Nodes-Per-Variable

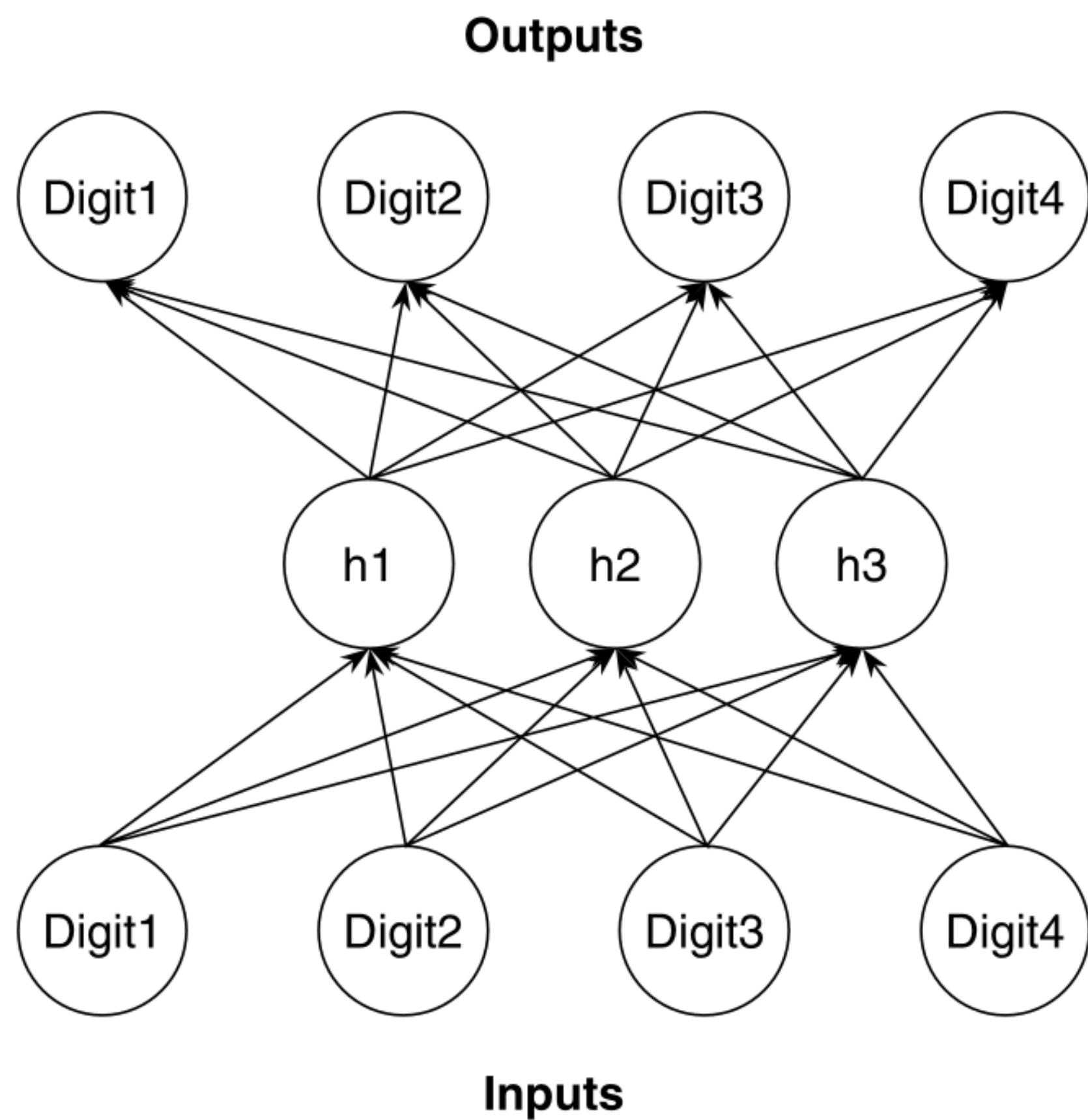
They “[...] can learn UQOTOMs, and they can learn arbitrary mappings. But what they learn depends on the nature of the learning algorithm.”

Input	Output
1010	1010
0100	0100
1110	1110
0000	0000
1111	?

$$f(x_1, x_2, x_3, x_4) = ?$$

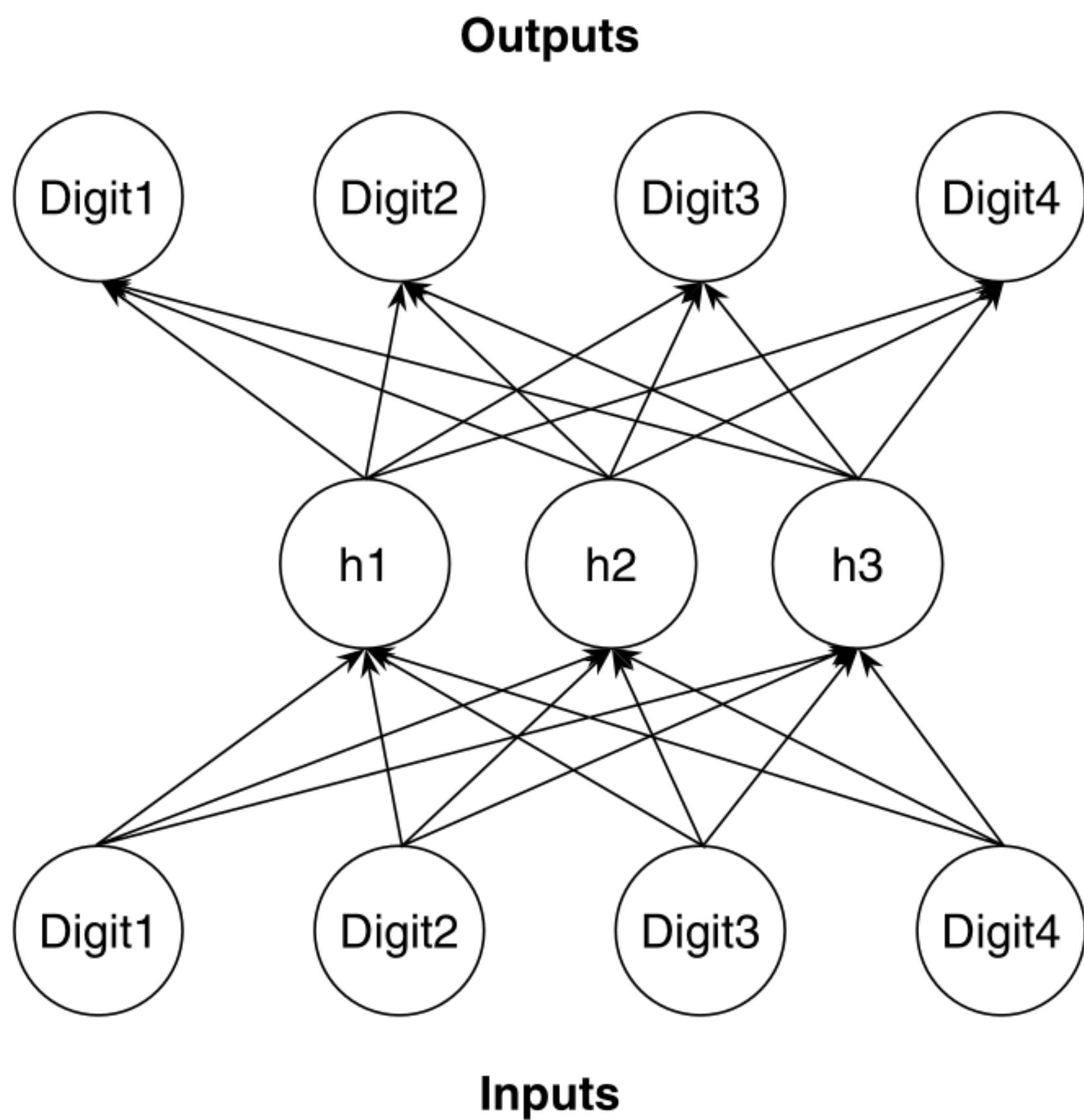
Can it generalize the identity function?

Many-Nodes-Per-Variable + Backpropagation



Input	Output
1010	1010
0100	0100
1110	1110
0000	0000
1111	?

Many-Nodes-Per-Variable + Backpropagation



Input	Output
1010	1010
0100	0100
1110	1110
0000	0000
1111	?

$$f(1, 1, 1, 1) = (1, 1, 1, 0)$$



Many-Nodes-Per-Variable + Backpropagation

“The flexibility in what [they] can represent leads to a flexibility in what they can learn.”

BUT:

*“Back-propagation [...] **does not allocate special status to UQOTOMs.**”*

It learns UQTOMS “only if it sees that UQOTOM illustrated with respect to each possible input and output node”

Input	Output
1010	1010
0100	0100
1110	1110
0000	0000
1111	?

$$f(1, 1, 1, 1) = (1, 1, 1, 0)$$



The Problem: Training Independence

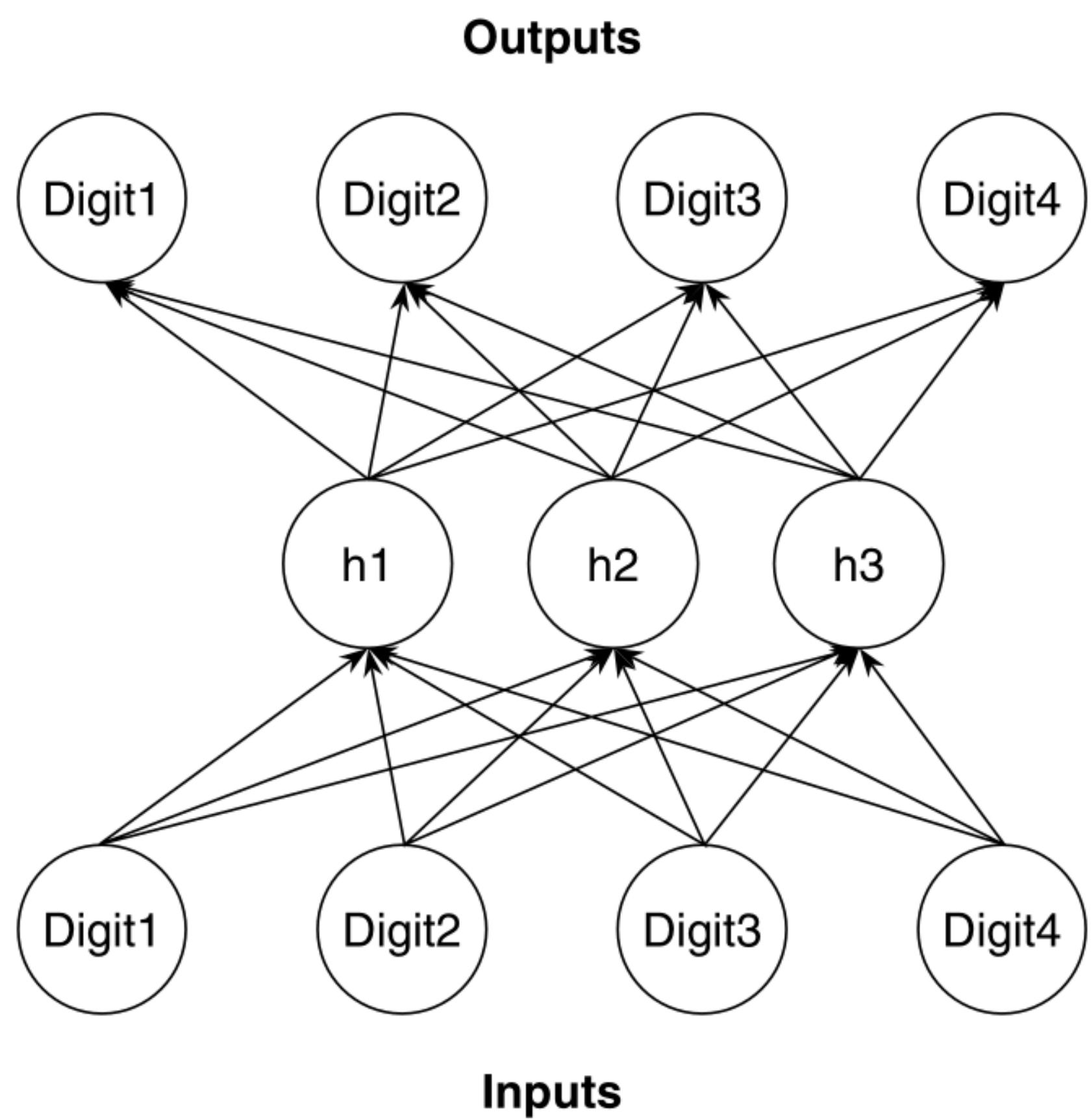
“The equations lead to two properties that I call **input independence** and **output independence** or, collectively, training independence”

The Problem: Training Independence

1. Input Independence:

“Input independence is about how the connections that **emanate from input nodes** are trained. First, when an input node is **always off** (that is, set to 0), the **connections** that emanate from it will **never change**.”

Input Independence



Input	Output
1010	1010
0100	0100
1110	1110
0000	0000
1111	?

The Problem: Training Independence

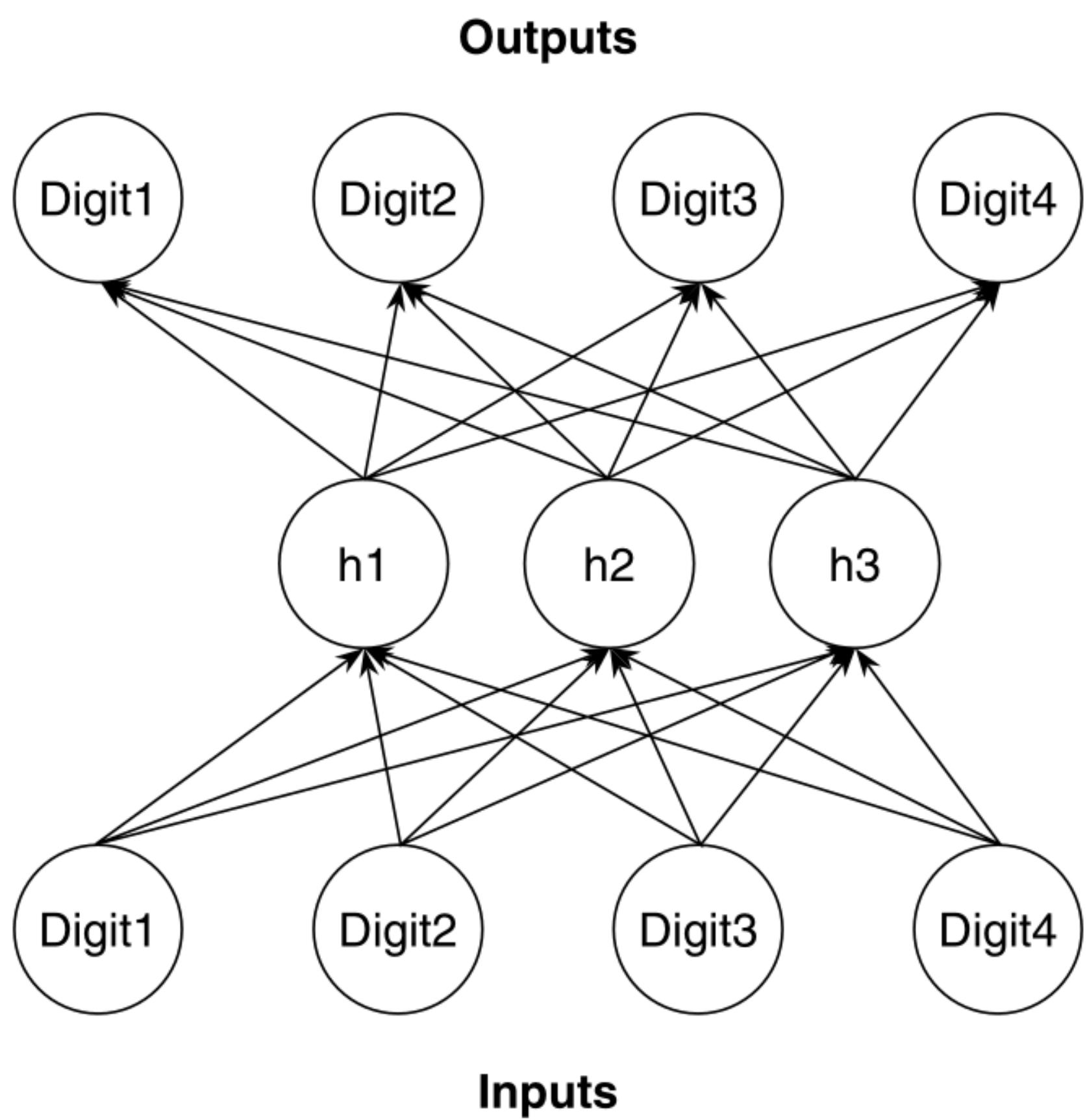
Input space

Training space

1. Output Independence:

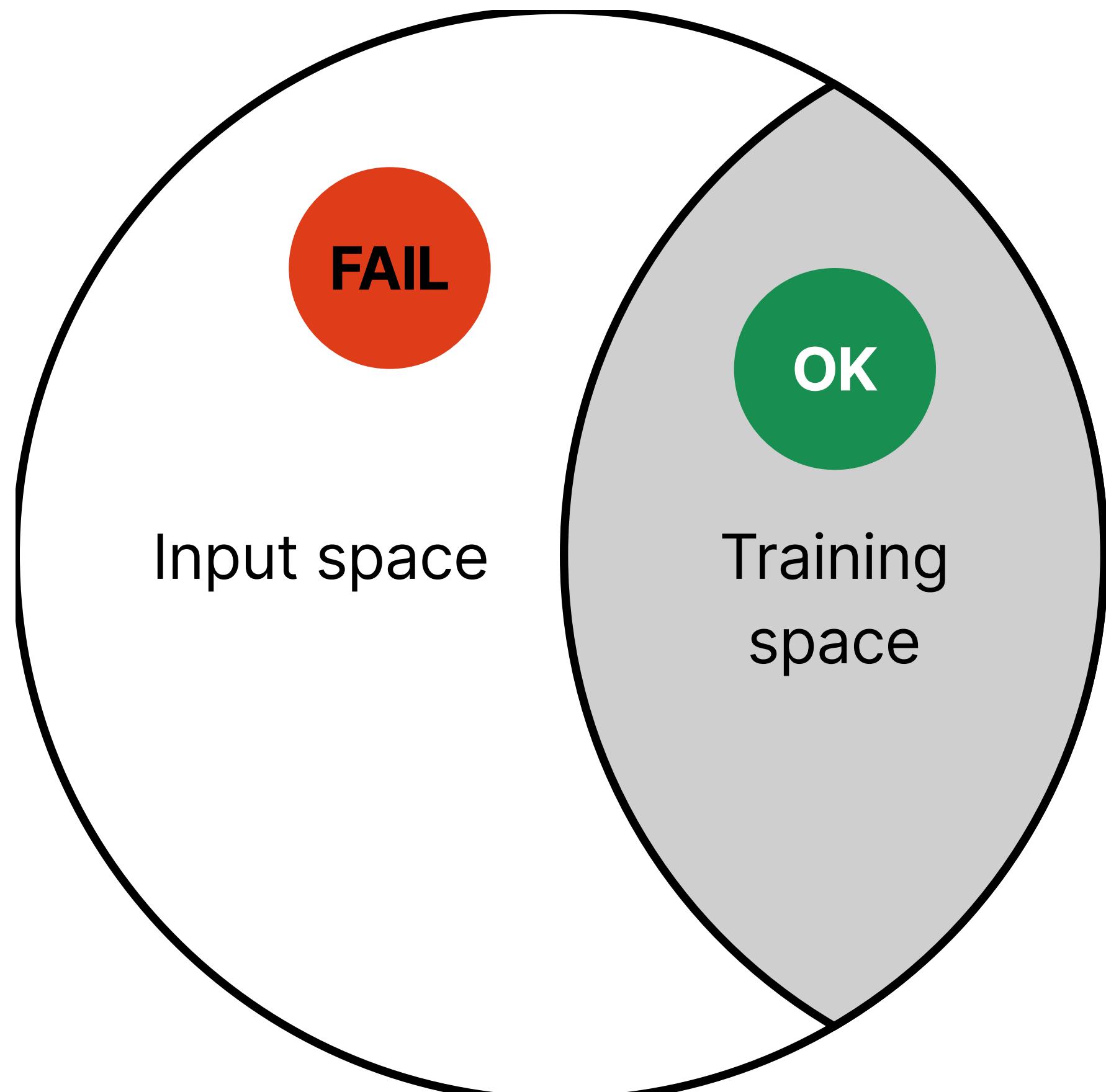
“Output independence is about the **connections that feed into the output units**. The equations that adjust the weights feeding an output unit j **depend** on the difference between the observed output for unit j and the target output for unit j but **not on the observed values or target values of any other unit.**”

Output Independence



Input	Output
1010	1010
0100	0100
1110	1110
0000	0000
1111	?

So what?



Backpropagation + MLP

*"The fact that [they] cannot generalize a **UQOTOM** function to a node that lies outside the training space follows from the equations that define back-propagation."*



Alternatives?

If this standard **architecture** is fundamentally **unsuited** for the task, what would a **better one** look like?

Outline

1. Marcus' line of argument

- UQOTOM

2. Critique of his argument

- Marcus' critique on MLPs and backpropagation
- Marcus' proposed alternative

3. Contemporary research

4. Discussion

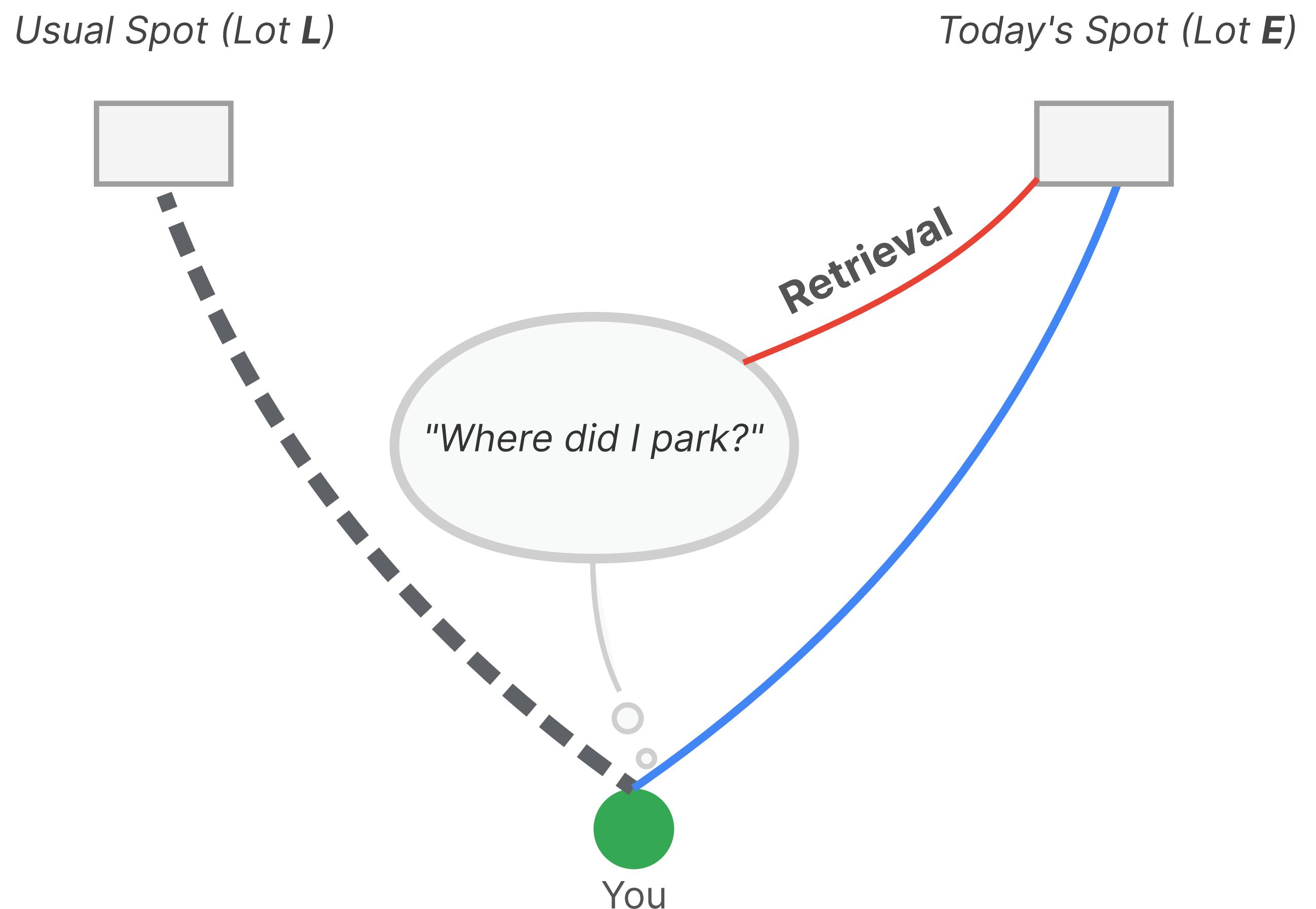
Why We Need Stored Bindings

The Car Park Analogy

- “Coming to work in the morning, I **park** my car in parking lot **E instead** of parking lot **L**, where I **usually park**... if I quickly think ‘Where did I park my car?’ I **remember**, and head correctly for lot E.... I remember to go to lot E on the basis of one occurrence.” - Jackendoff & Mataric, 1997

Marcus argues

This requires a mechanism for **stable**, yet **rapidly updatable, memory**—not just a fleeting activation pattern.



The Problem

Storing a Binding

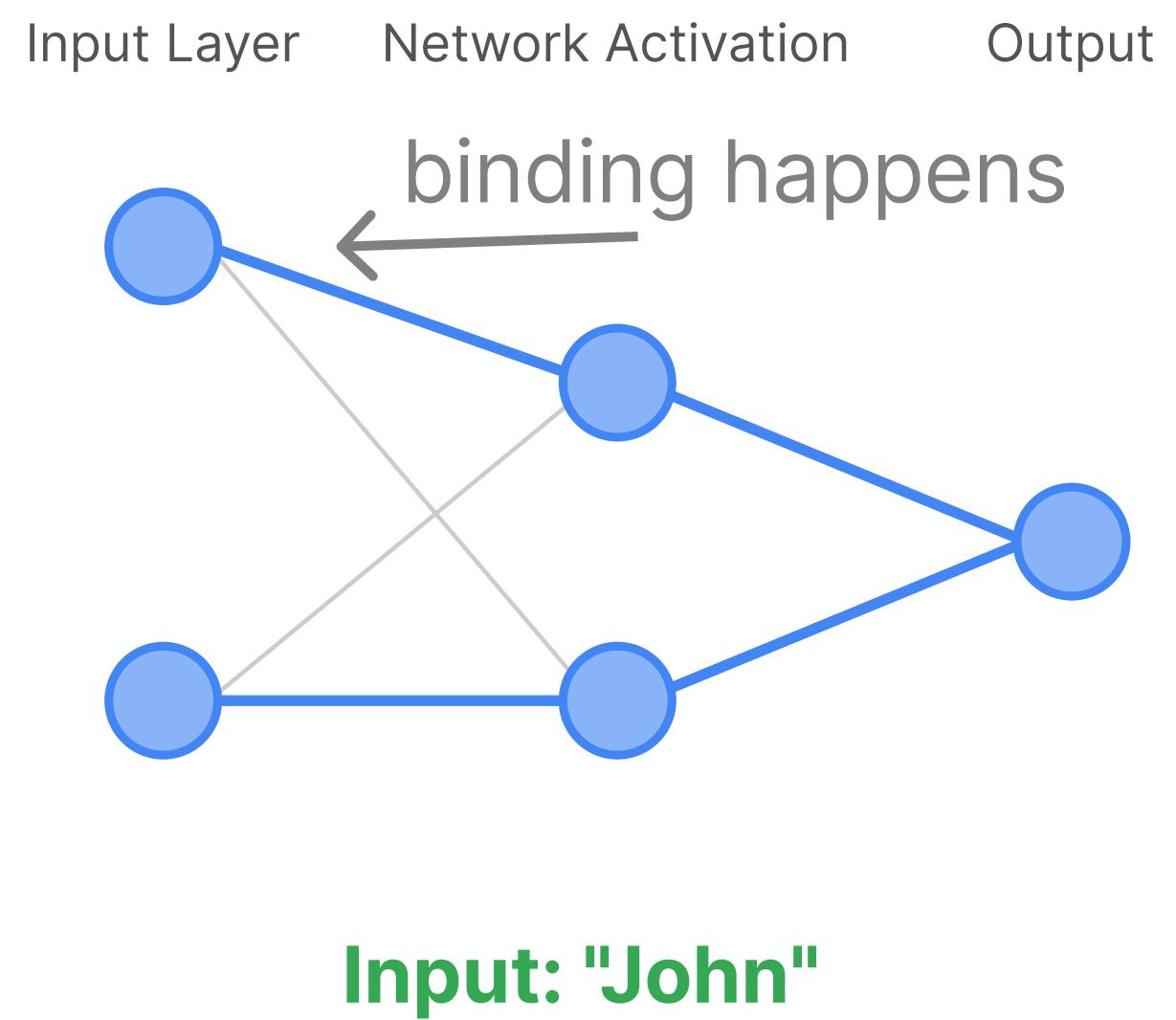
- Car Park Analogy:

We want to be able to '**store**' a **binding**.
like "**today's parking spot = E**"

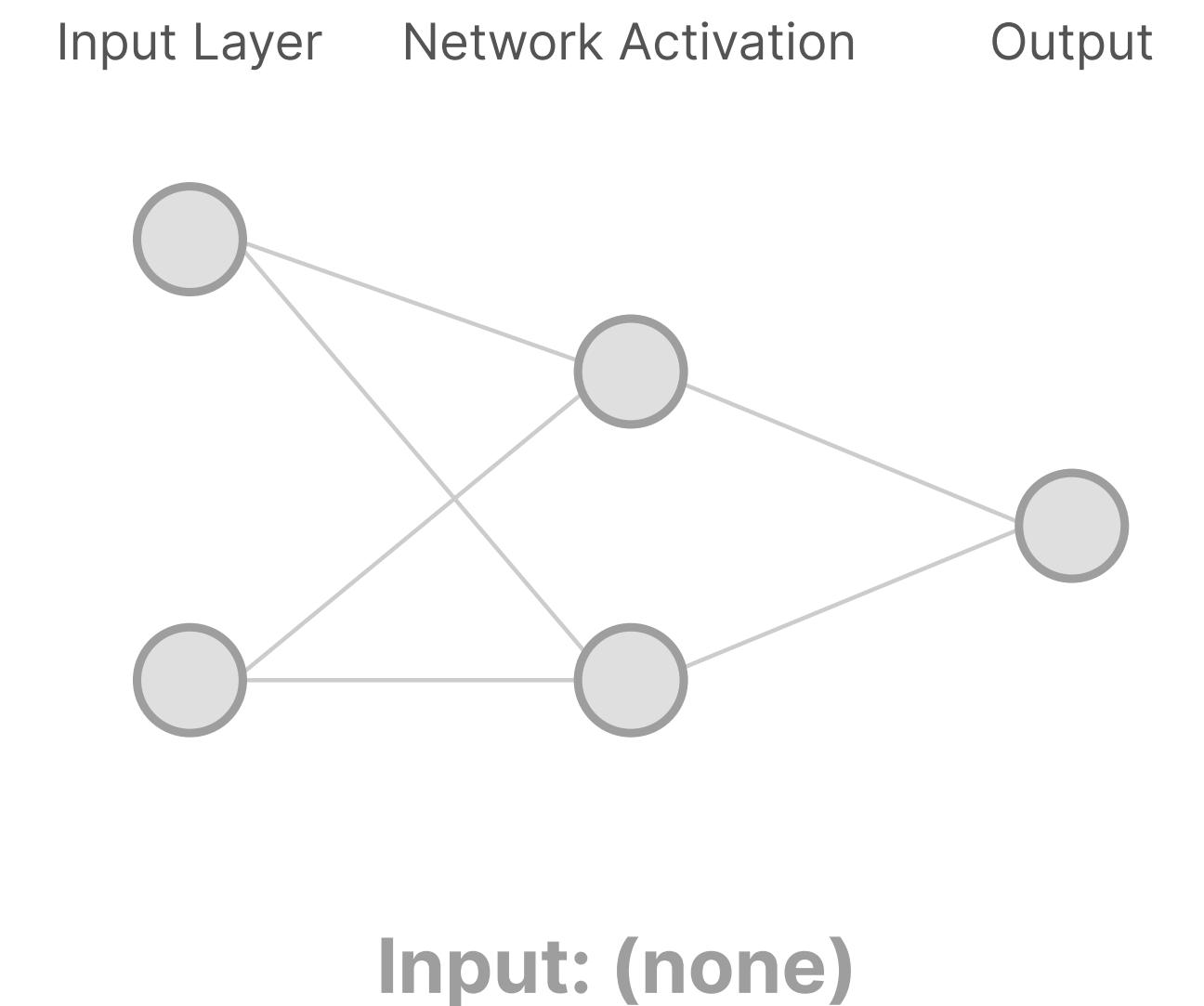
But

- In standard networks, a **binding** is just the **pattern of activation** caused by the **current input**. It disappears when the input is removed.

A. Input Present → Binding Active



B. Input Removed → Binding Vanishes



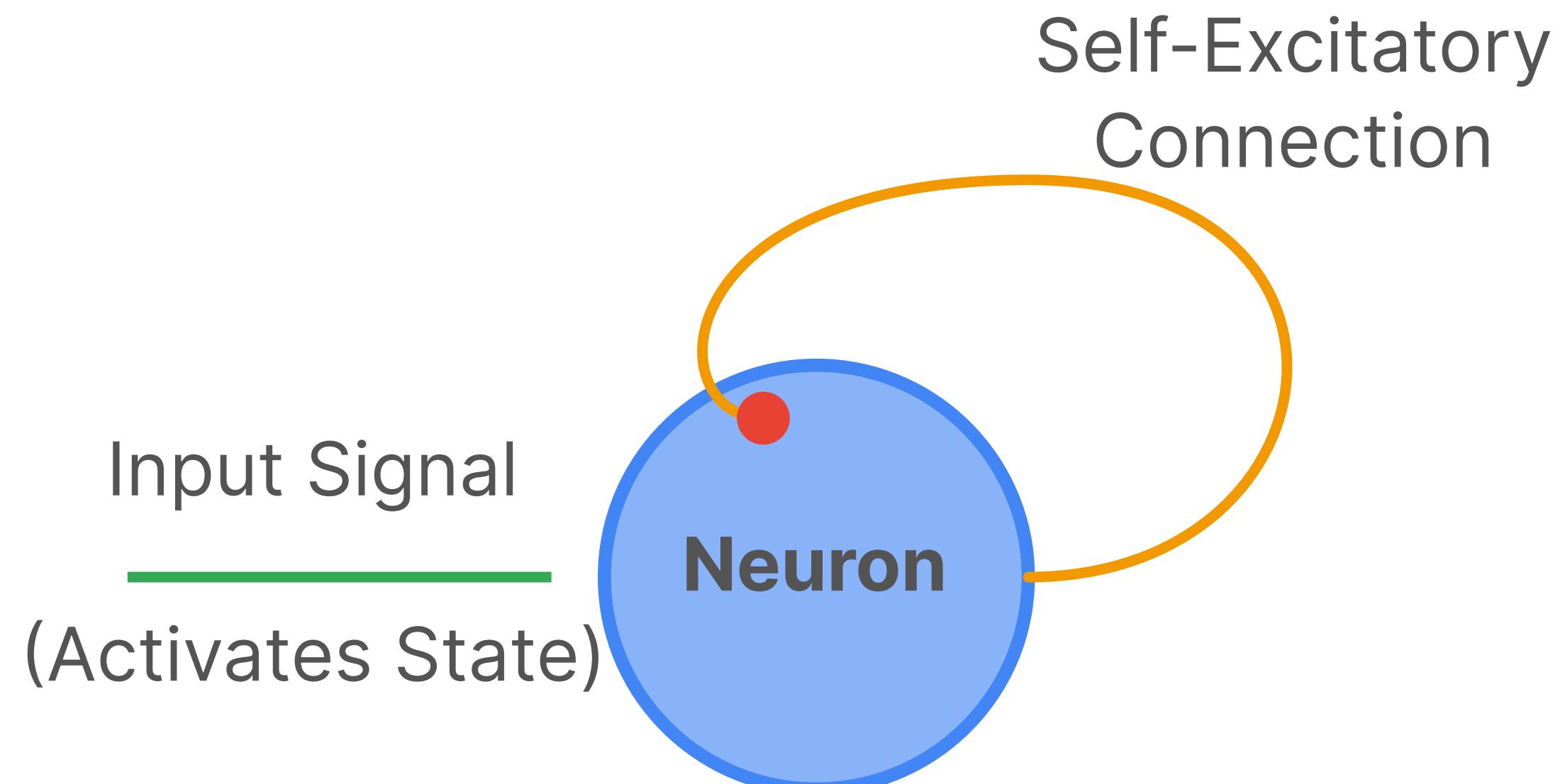
The pattern of activation (the "binding") only exists when the input is actively present.

Registers

Idea:

"One way to do this is to use devices that have **two or more stable states** [...] **digital computers** often make use of flip-flops—**binary or bistable devices** that can be set to either on or off and then **maintained** in that state **without further input.**"

"Registers **are central** to digital **computers**; my **suggestion** is that registers are **central** to **human cognition** as well."



Bistable Device (Register)

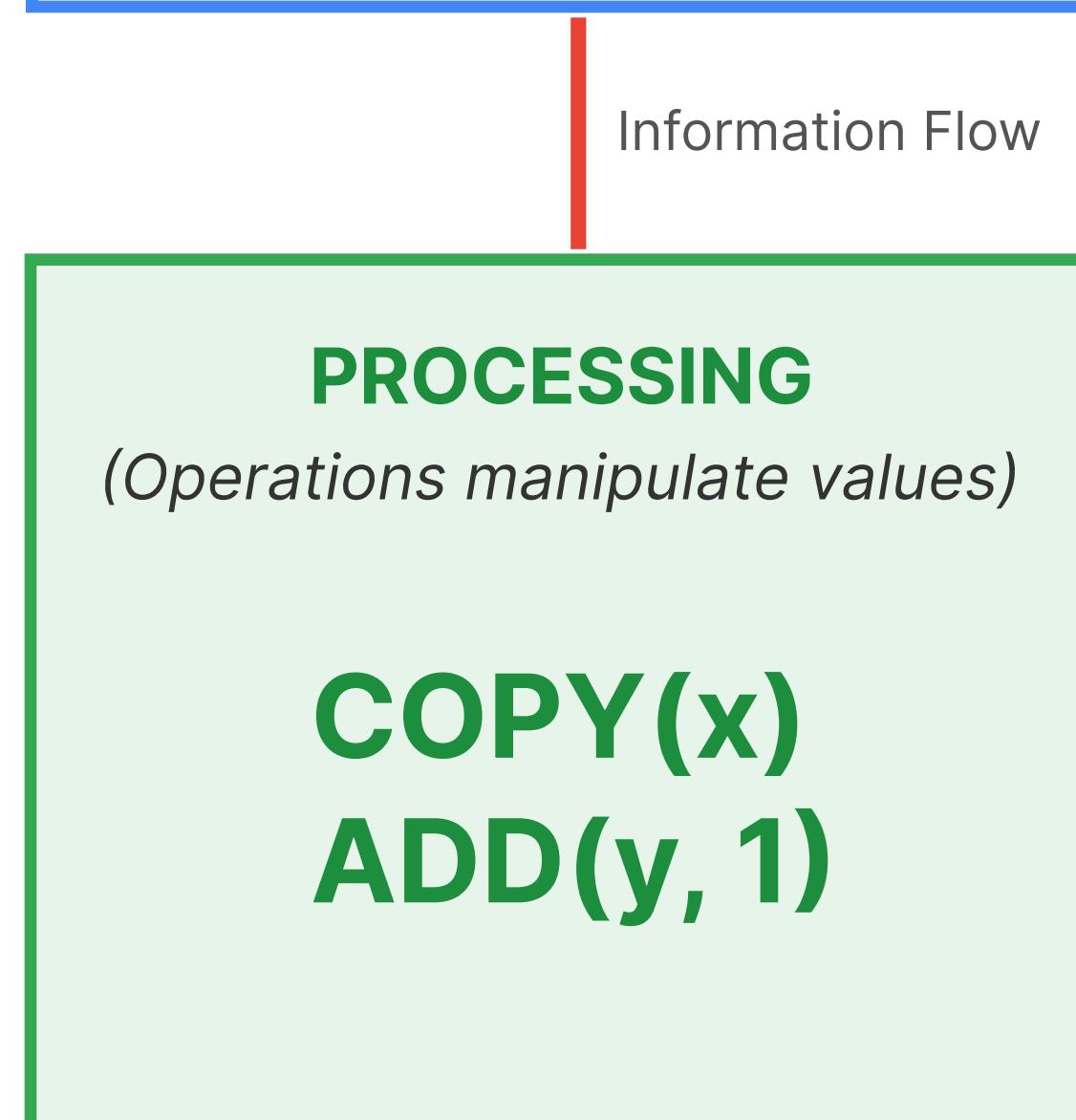
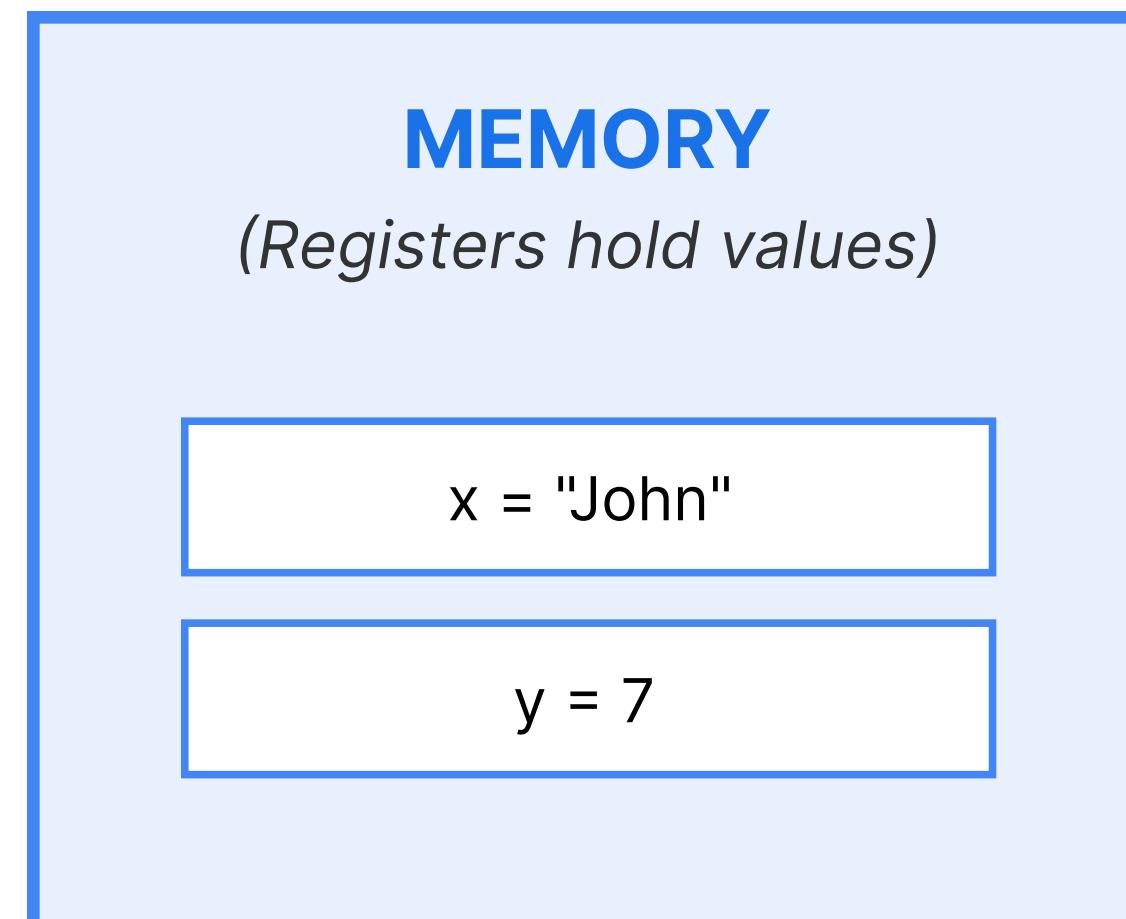
Can maintain an "ON" state.

Architectural Shift

Separating Memory & Processing

- By **using registers**, you create a **fundamentally different architecture**.
- **Registers provide a dedicated mechanism for memory** (storing values).
- An **operation reads from or writes to the register**.

*“... standard multilayer perceptrons **do not** make a **distinction** between **components for processing** and **components for memory**.”*



Recap + Case Study

The Framework

Test: UQOTOM generalization.

Critique: Standard MLPs fail due to "*Training Independence*."

Proposal: A system with registers + operations should pass *the test*.

Can Marcus idea hold up?

Does this **framework** correctly **predict** which real-world **models work** and which don't?

Case Study

The Experiment

The Question: Can 7-month-old **infants learn** an abstract, **algebraic-like rule?**

Learning Phase: Infants listened to sentences from **one of two grammars:**

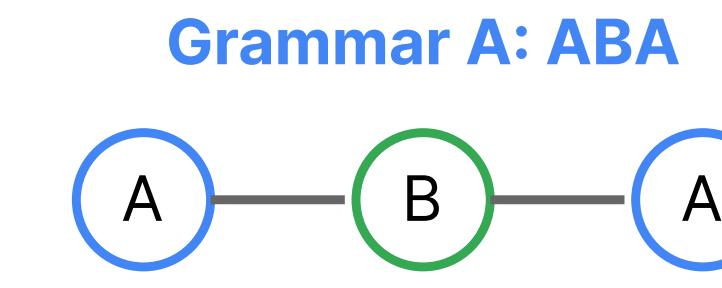
- Grammer A: **ABA** (e.g., ga ti ga)
- Grammer B: **ABB** (e.g., ga ti ti)

The Finding: Infants **generalize** the **abstract ABA rule**, showing they can **learn an UQOTOM**.

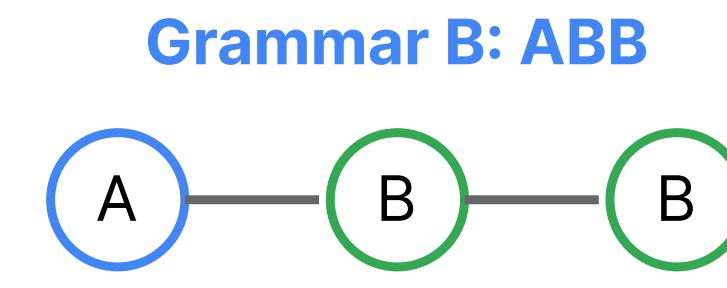
→ “*infants were indeed sensitive to the abstract structure*”

The Marcus et al. (1999) Experiment

1. Habituation Phase: Infants learn a grammar

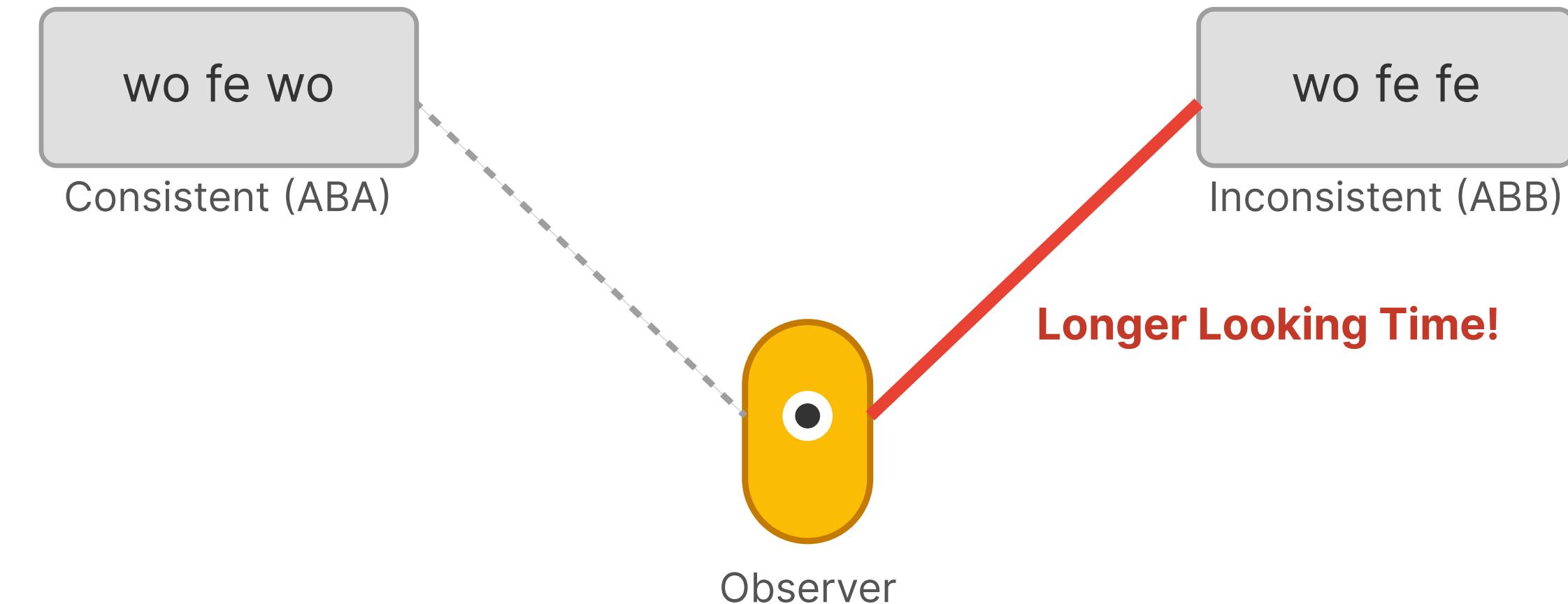


e.g., "ga ti ga"



e.g., "ga ti ti"

2. Test Phase (After learning ABA grammar)



Marcus's Analysis

Sorting the Models

After the (infant) **study was published**, a number of researchers **proposed connectionist models** to **explain** the results.

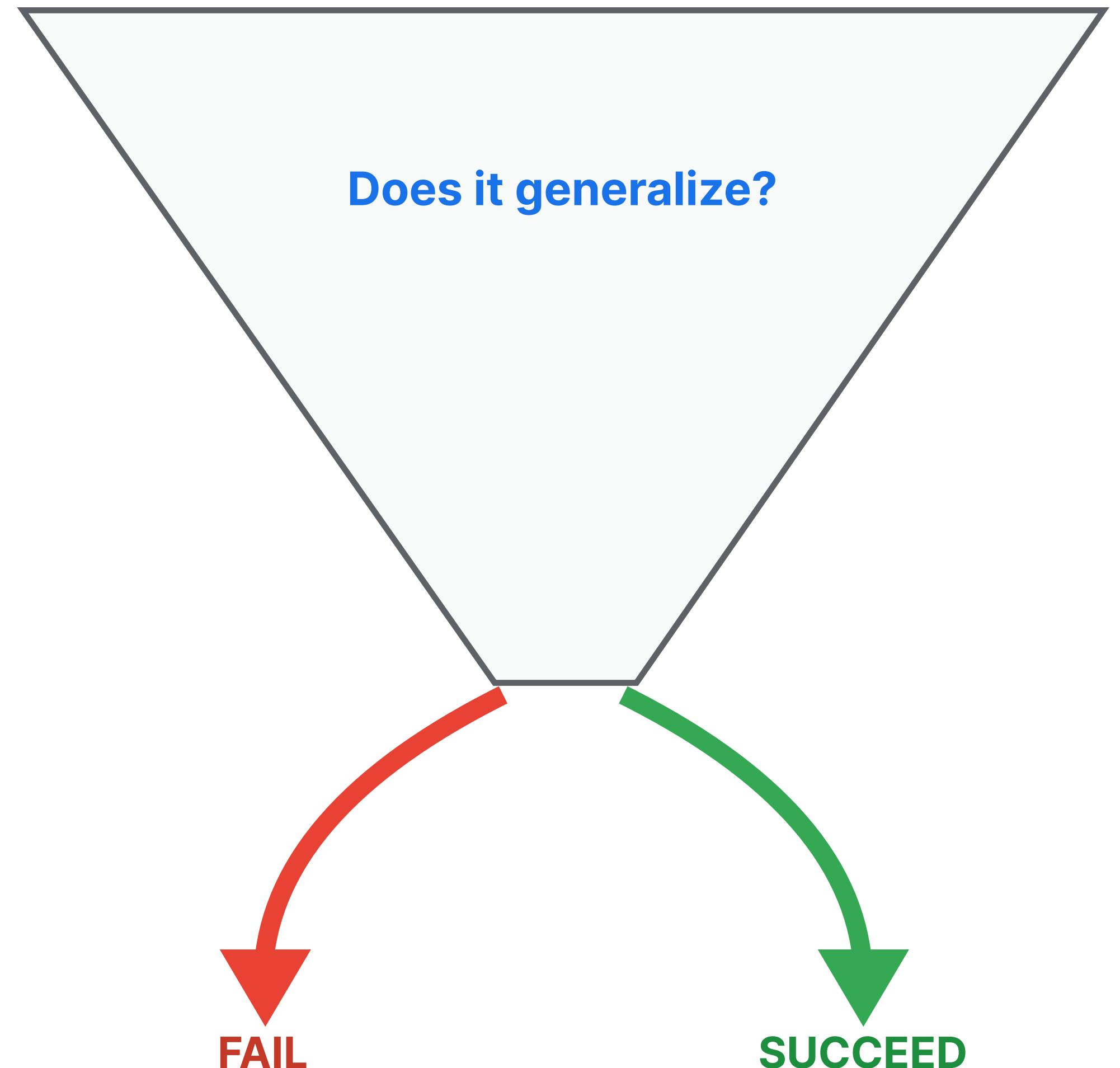
Marcus's Approach: He **analyzes** these models using the **framework** he developed.

Marcus's Analysis: Sorting the Models

SRN (Elman)

Shultz Model

Other Models...



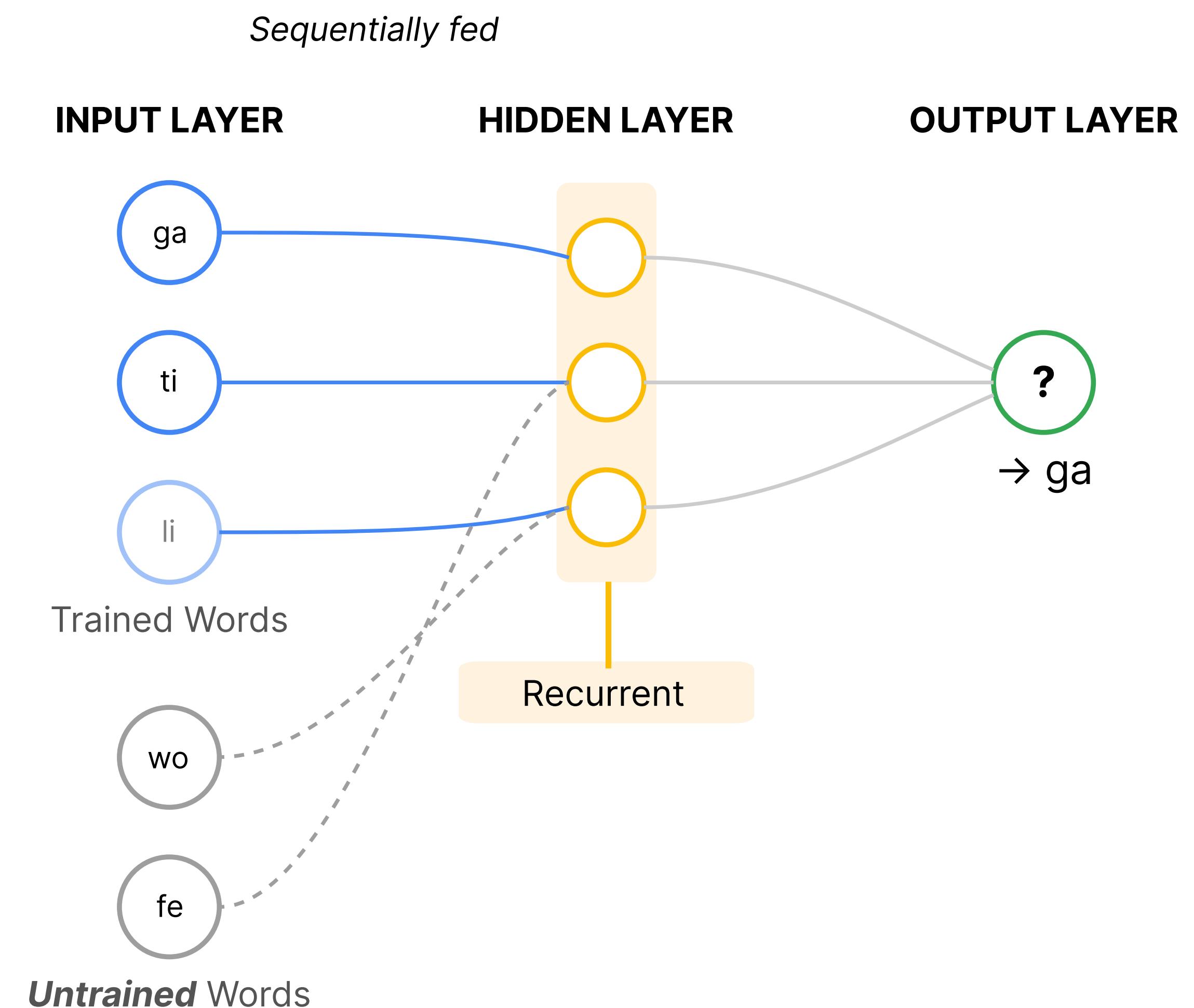
Models That Fail

The Standard SRN

The Model: A standard Simple Recurrent Network (Elman, 1990), where each word is a unique node.

The Result: It fails to generalize. → new words

The Reason: Training Independence.



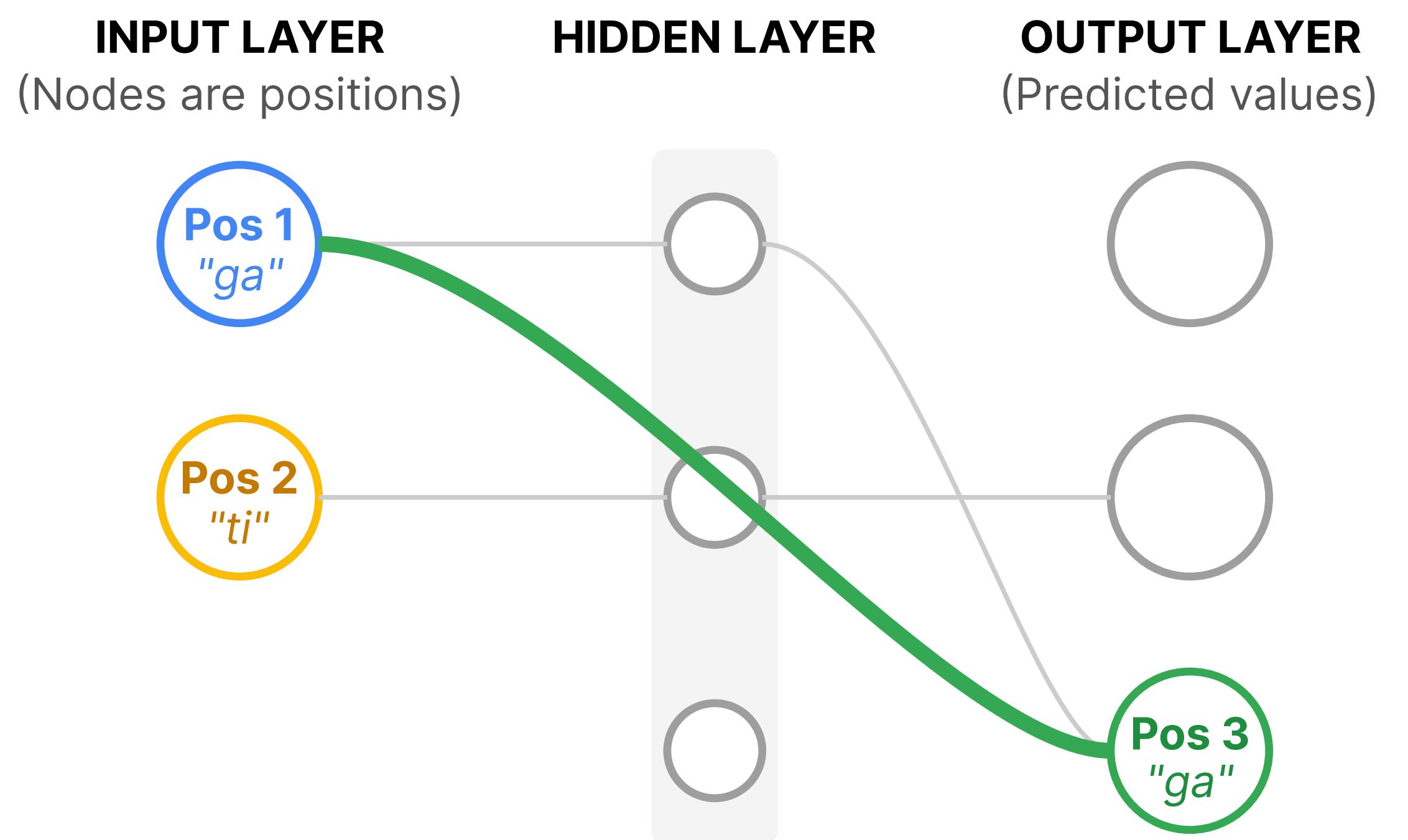
Models That Succeed (1)

Nodes as Variables

The Model: Shultz (1999)

Why it Works: The network learns to wire the "1st Word" node to the "3rd Word" node, **effectively creating a COPY operation** that works on any value.

→ **implementation** of ABA rule



Models That Succeed (2)

Registers & Comparators

The Model: Dominey & Ramus (2000)

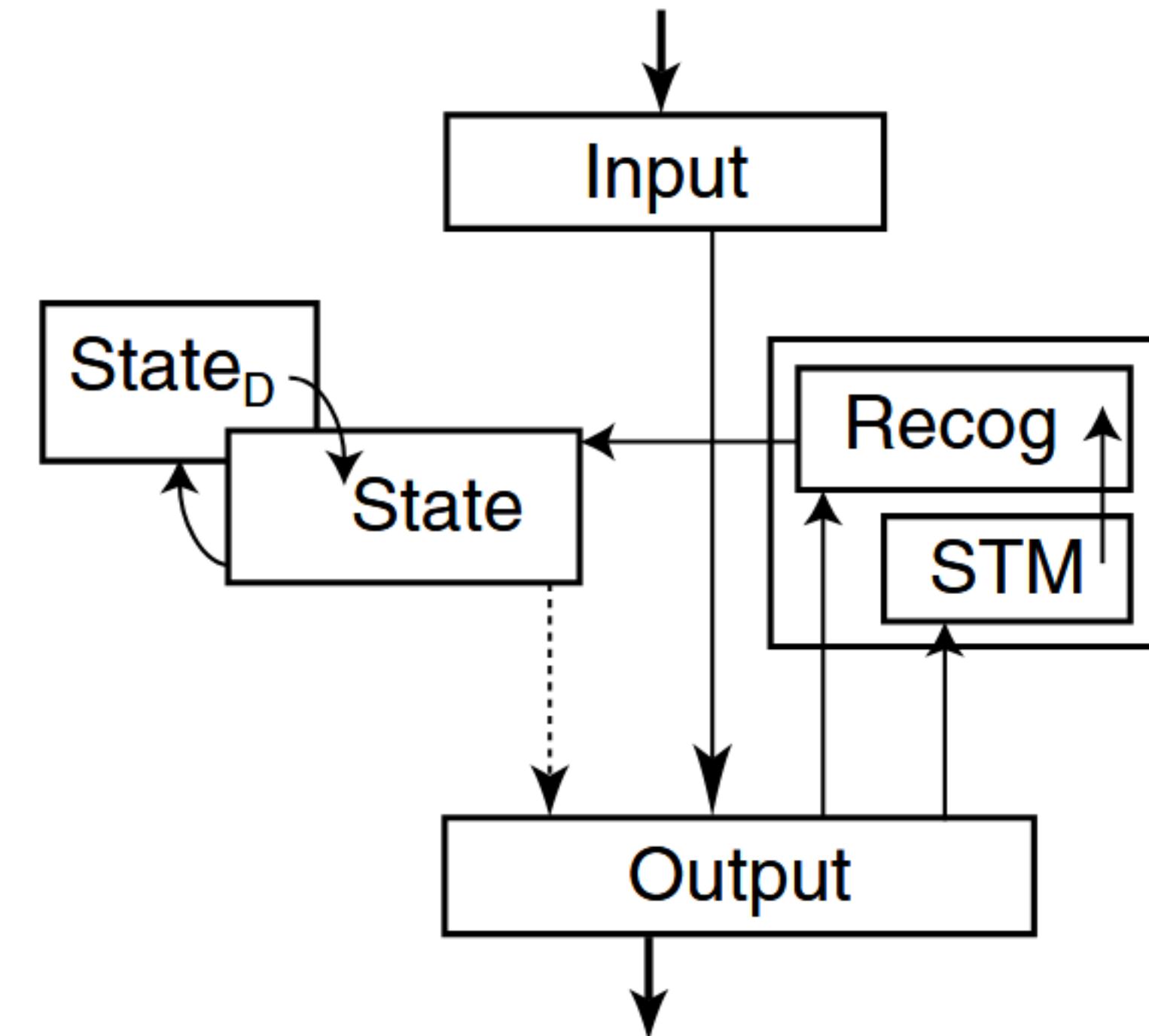
The Architecture:

The authors explicitly **tested two** versions:

The **successful** one **included** a short-term memory buffer (**a register**) and a mechanism to compare the current input to the value in memory (**an operation**).

The Authors' Own Conclusion:

The "**registerless**" version of the model **could not** capture our **results**"



Case Study

Conclusion

- Models that **Fail** lack a mechanism to **implement an operation over a variable** and are trapped by **Training Independence**.
- Models that **Succeed** work because they **explicitly build** in the machinery for **symbol manipulation** (nodes-as-variables, registers, etc.).

*"what makes the **successful** connectionist models work is a facility for **representing operations** that apply to all instances in a class"*

→ They aren't alternatives to rules; they are neural implementations of them.

Marcus' argument in a nutshell

- **Premise:** Many algebraic rules are UQOTOM
- **Premise:** Humans can learn/ generalize these rules, so our models should too
- **Conclusion:** Models should be able to represent and learn UQOTOM (even with restricted training data)
- **Claim:** MLPs don't generalize UQOTOM well outside the training space (based on observation, argued via training independence)
- **Conclusion:** Vanilla MLPs are not well suited for tasks that require a high amount of symbol manipulation
- **Suggestion:** Motivated by computer architecture and supported by case studies, he suggests the use of registers

Marcus' argument in a nutshell

- **Premise:** Many algebraic rules are UQOTOM
- **Premise:** Humans can learn/ generalize these rules, so our models should too
- **Conclusion:** Models should be able to represent and learn UQOTOM (even with restricted training data)
- **Claim:** MLPs don't generalize UQOTOM well outside the training space (based on observation, argued via training independence)
- **Conclusion:** Vanilla MLPs are not well suited for tasks that require a high amount of symbol manipulation
- **Suggestion:** Motivated by computer architecture and supported by case studies, he suggests the use of registers

Marcus' argument in a nutshell

- **Premise:** Many algebraic rules are UQOTOM
- **Premise:** Humans can learn/ generalize these rules, so our models should too
- **Conclusion:** Models should be able to represent and learn UQOTOM (even with restricted training data)
- **Claim:** MLPs don't generalize UQOTOM well outside the training space (based on observation, argued via training independence)
- **Conclusion:** Vanilla MLPs are not well suited for tasks that require a high amount of symbol manipulation
- **Suggestion:** Motivated by computer architecture and supported by case studies, he suggests the use of registers

Marcus' argument in a nutshell

- **Premise:** Many algebraic rules are UQOTOM
- **Premise:** Humans can learn/ generalize these rules, so our models should too
- **Conclusion:** Models should be able to represent and learn UQOTOM (even with restricted training data)
- **Claim:** MLPs don't generalize UQOTOM well outside the training space (based on observation, argued via training independence)
- **Conclusion:** Vanilla MLPs are not well suited for tasks that require a high amount of symbol manipulation
- **Suggestion:** Motivated by computer architecture and supported by case studies, he suggests the use of registers

Marcus' argument in a nutshell

- **Premise:** Many algebraic rules are UQOTOM
- **Premise:** Humans can learn/ generalize these rules, so our models should too
- **Conclusion:** Models should be able to represent and learn UQOTOM (even with restricted training data)
- **Claim:** MLPs don't generalize UQOTOM well outside the training space (based on observation, argued via training independence)
- **Conclusion:** Vanilla MLPs are not well suited for tasks that require a high amount of symbol manipulation
- **Suggestion:** Motivated by computer architecture and supported by case studies, he suggests the use of registers

Marcus' argument in a nutshell

- **Premise:** Many algebraic rules are UQOTOM
- **Premise:** Humans can learn/ generalize these rules, so our models should too
- **Conclusion:** Models should be able to **represent and learn UQOTOM** (even with restricted training data)
- **Claim:** MLPs don't generalize UQOTOM well outside the training space (based on observation, argued via training independence)
- **Conclusion:** Vanilla MLPs are **not well suited** for tasks that require **symbol manipulation**
- **Suggestion:** Motivated by computer architecture and supported by case studies, he **suggests** the use of **registers**

Outline

1. Marcus' line of argument

2. Critique of his argument

3. Contemporary research

4. Discussion

Critique

Are UQOTOM the Right Test?

Humans also generalize **non-UQOTOMs**.
UQOTOMs are a sub-class of problems.

→ Maybe **MLPs** are **better suited** for
these **messier, many-to-one mappings**
that are common in the real world.

Is This a Fair Fight?

Many important UQOTOMs (identity, concatenation) have
simple symbolic definitions.

→ The **main purpose** of MLPs is to **approximate functions** we
don't have a neat **formula** for. → **Why even inspect** whether
they learn these 'simple' functions given **symbol processing** can
inherently solve them better.

Input	Output
(1, 1)	1
(4, 2)	8
(3, 4)	12
(2, 3)	6
(3, 2)	?

Critique

Are UQOTOM the Right Test?

Humans also generalize **non-UQOTOMs**.
UQOTOMs are a sub-class of symbol manipulation problems.

→ Maybe **MLPs** are **better suited** for these **messier, many-to-one mappings** that are common in the real world.

Is This a Fair Fight?

Many important UQOTOMs (identity, concatenation) have **simple symbolic definitions**.

→ The **main purpose** of MLPs is to **approximate functions** we **don't** have a neat **formula** for. → **Why** even **inspect** whether they learn these 'simple' functions given **symbol processing** can **inherently** solve them **better**.

Input	Output
(1, 1)	1
(4, 2)	8
(3, 4)	12
(2, 3)	6
(3, 2)	?

Outline

1. Marcus' line of argument

2. Critique of his argument

3. Contemporary research

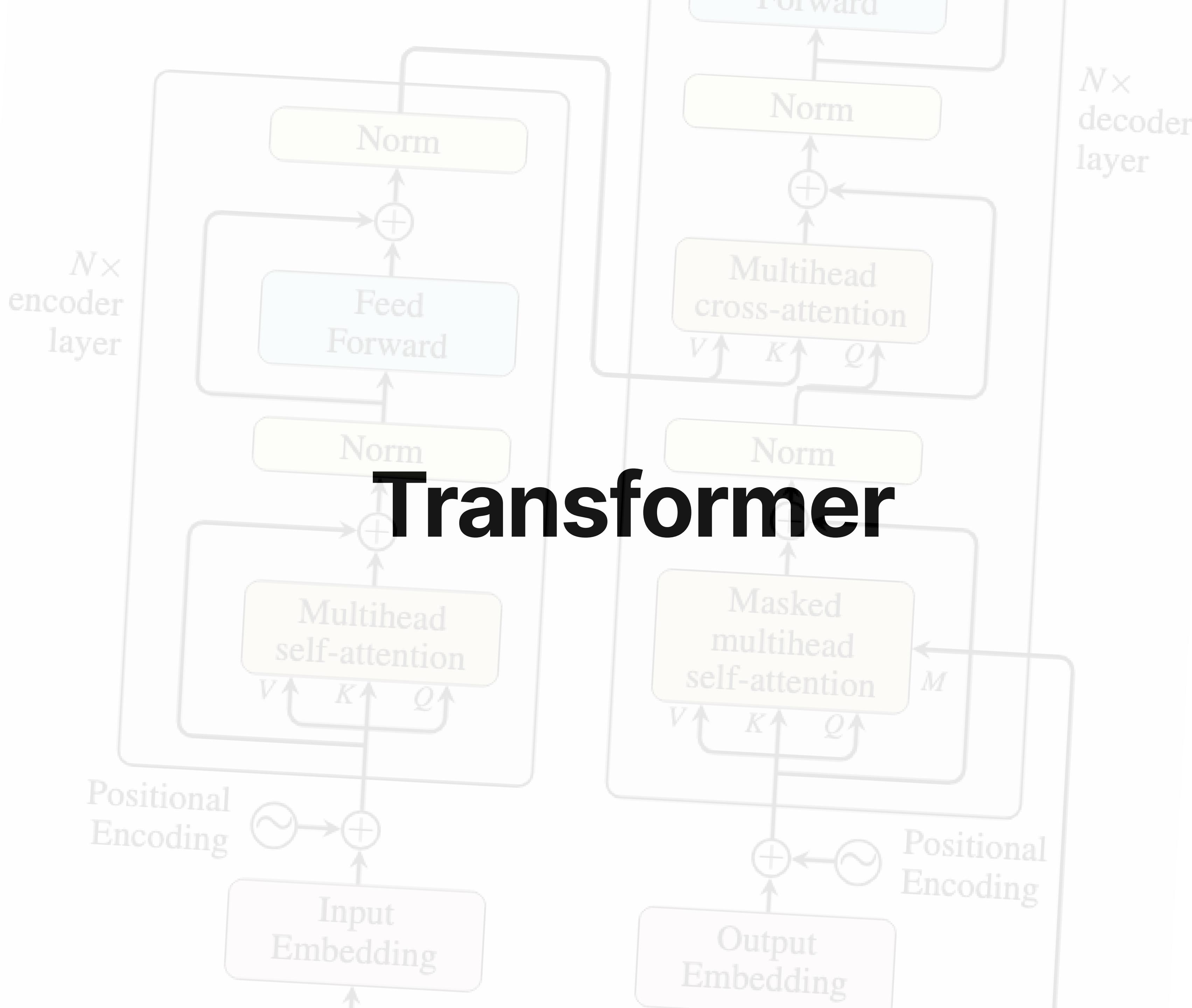
4. Discussion



Gemini
Claude

LLM

Transformer



Transformer = MLP?

Flaws of plain MLP

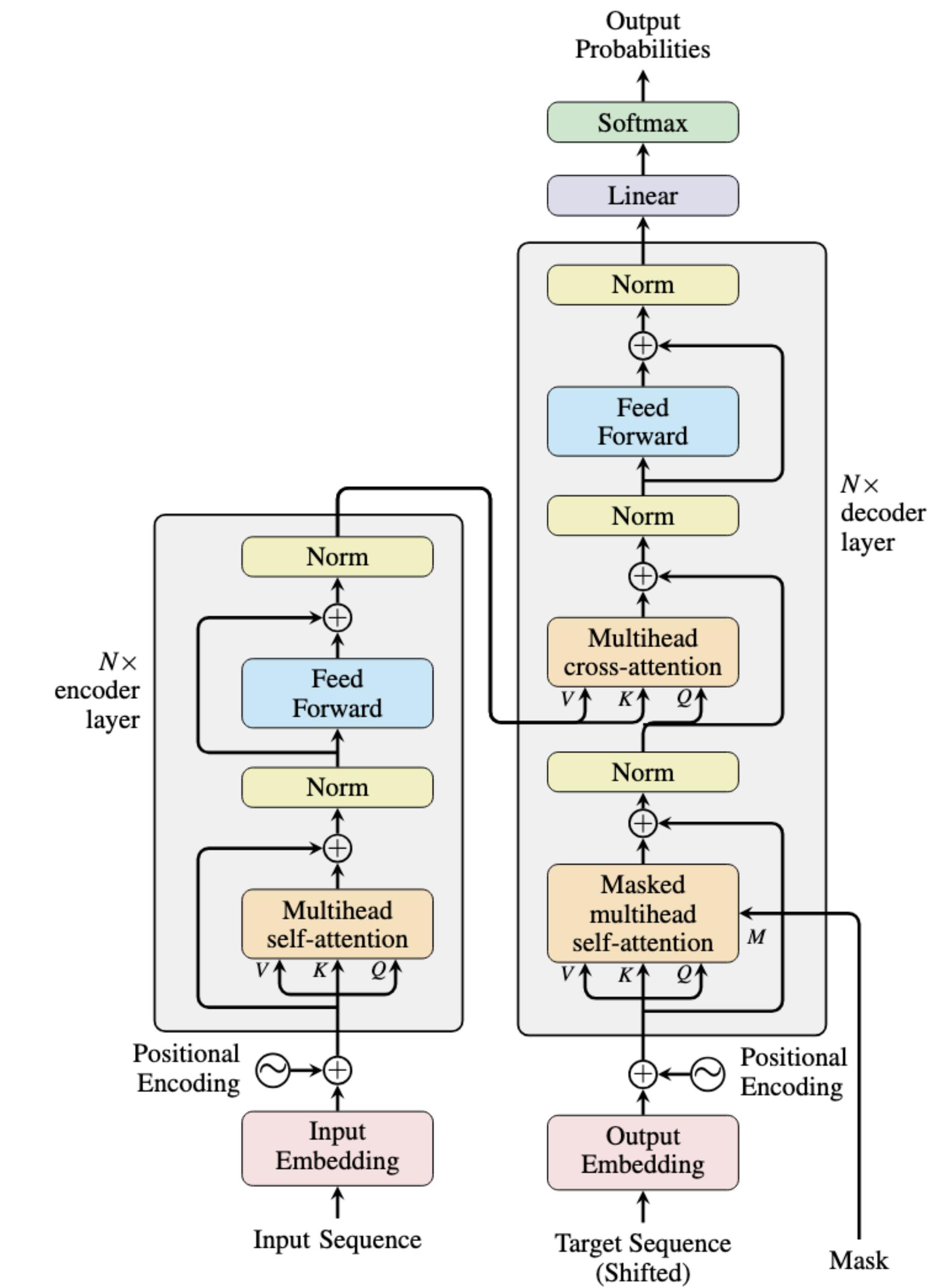
Fixed-Size Input and Output

No Concept of Word Order

No Sharing of Weights Across Positions

The Idea

Combine algorithmic processing with feedforward networks.



Transformer in a Nutshell

Architecture optimized for NLP

Reusing parameters, encoding position

Includes algorithmic operations

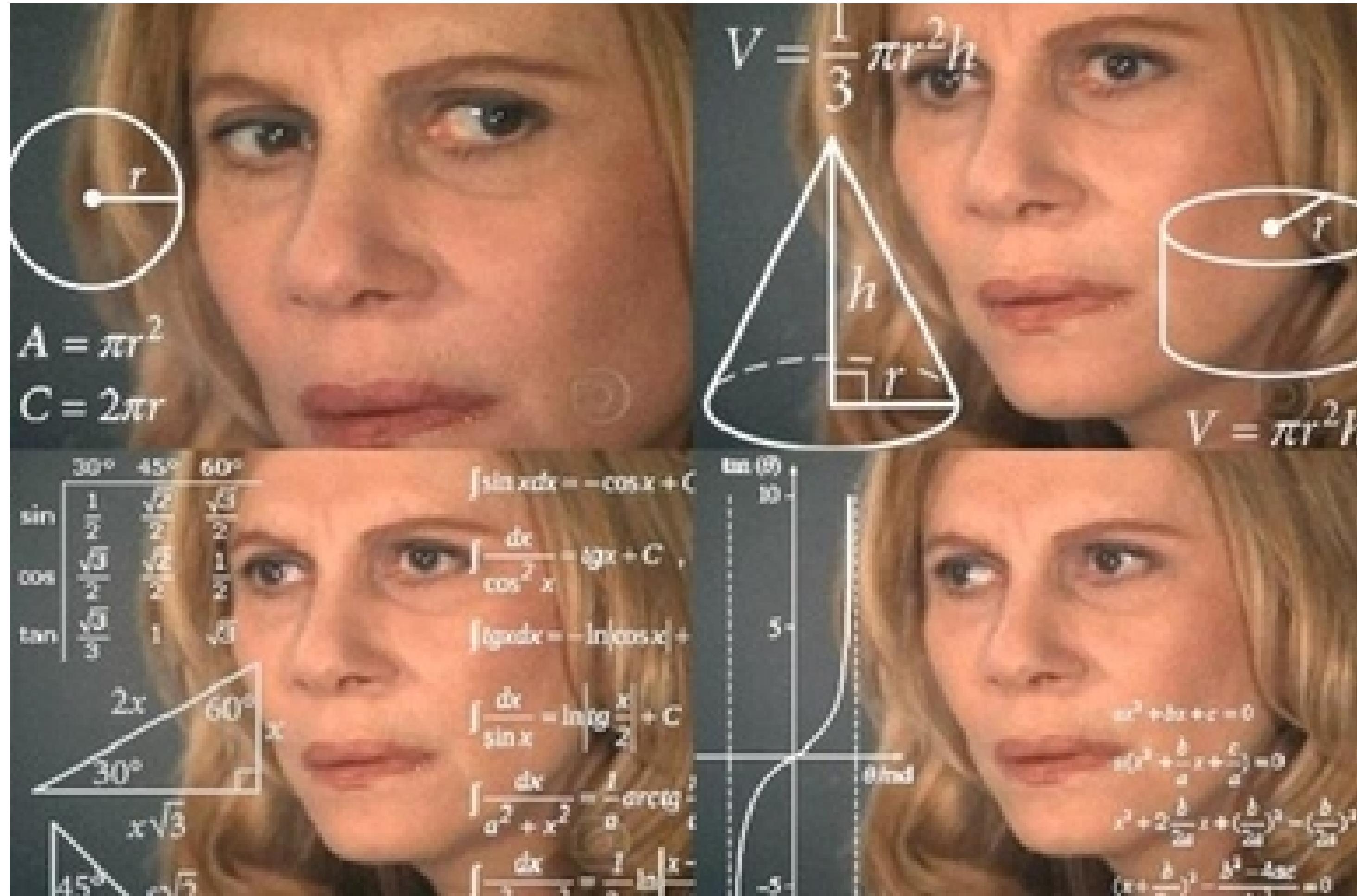
Iteratively generates tokens, ...

Main ingredient: MLP

Trained via backpropagation

Large Reasoning Model

Think before responding



All problems solved?

what is 8.8-8.11?



Gedankengang anzeigen ^

Calculating the Difference

I've just finished the simple subtraction calculation and arrived at the final answer. The difference between 8.8 and 8.11 is now clear, and I'm ready to present it.

$$8.8 - 8.11 = -0.31$$

Let's do the subtraction properly:

$$\begin{array}{r} 8.11 \\ - 8.80 \\ \hline 0.31 \end{array}$$

So, the absolute difference between 8.11 and 8.8 is **0.31**.

Step 2: Apply the correct sign.

Because our original problem was $8.8 - 8.11$ (subtracting a larger number), the result must be negative.

Therefore, the answer is **-0.31**.

The Illusion of Thinking: Understanding the Strengths and Limitations of Reasoning Models via the Lens of Problem Complexity

Parshin Shojaee*†
Maxwell Horton

Iman Mirzadeh*
Samy Bengio

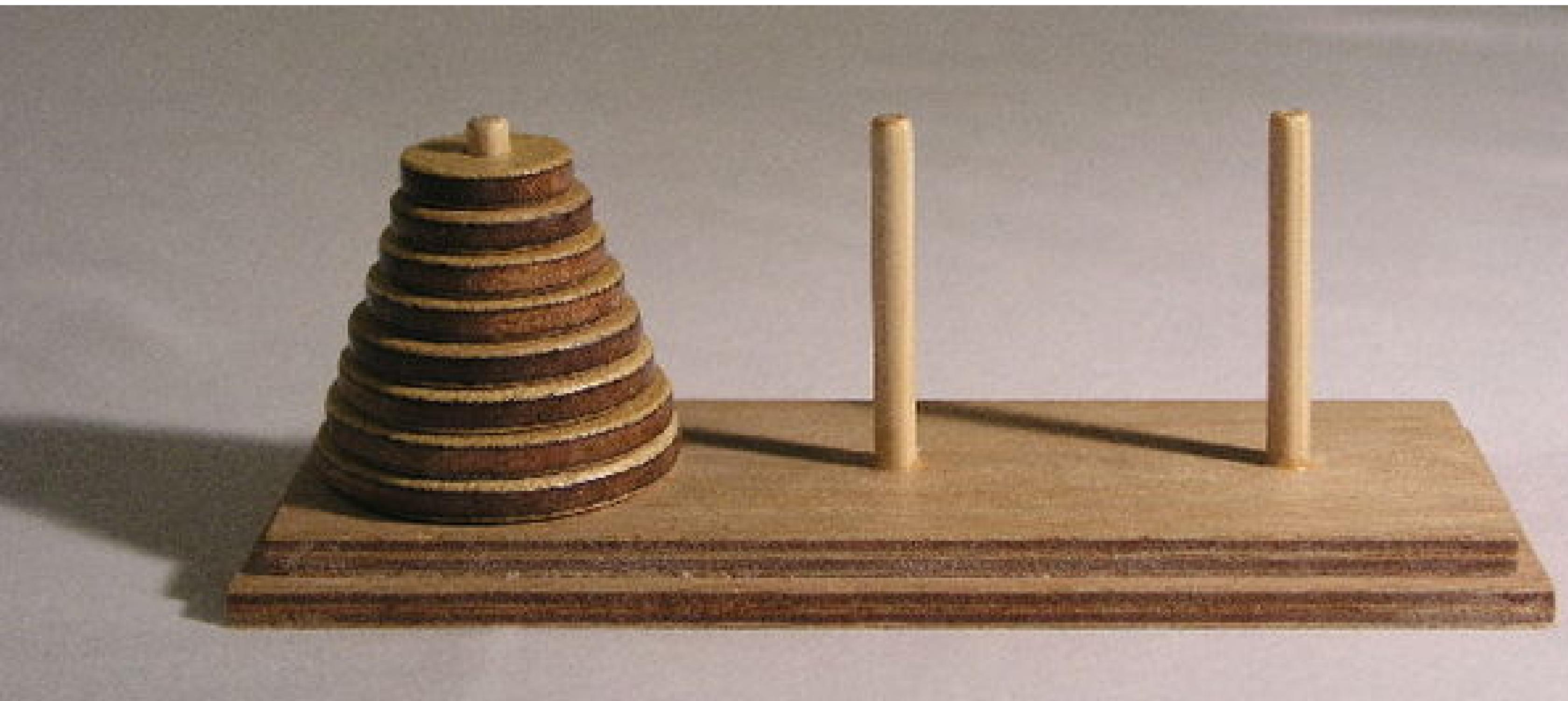
Keivan Alizadeh
Mehrdad Farajtabar

Apple

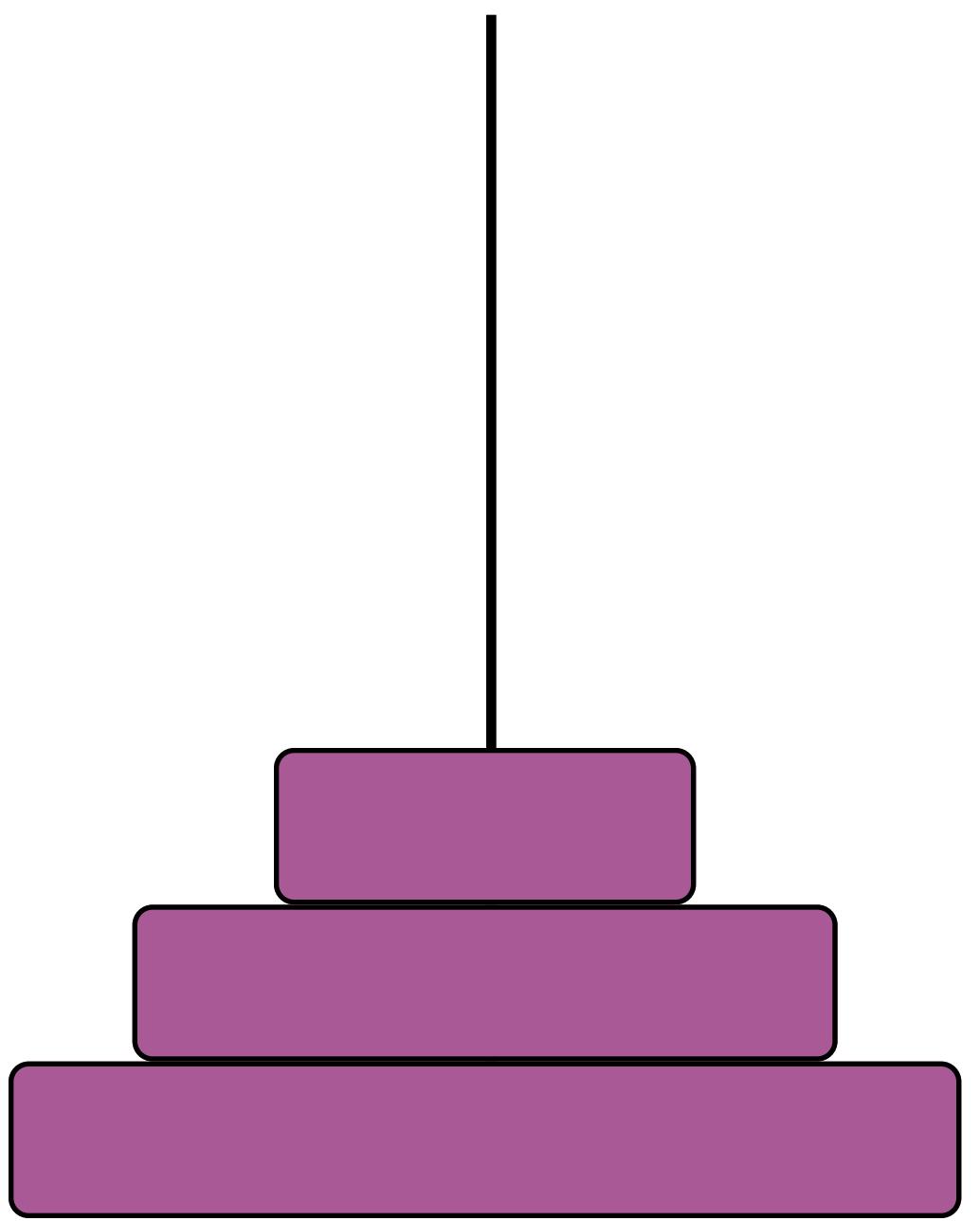
Marcus claims:

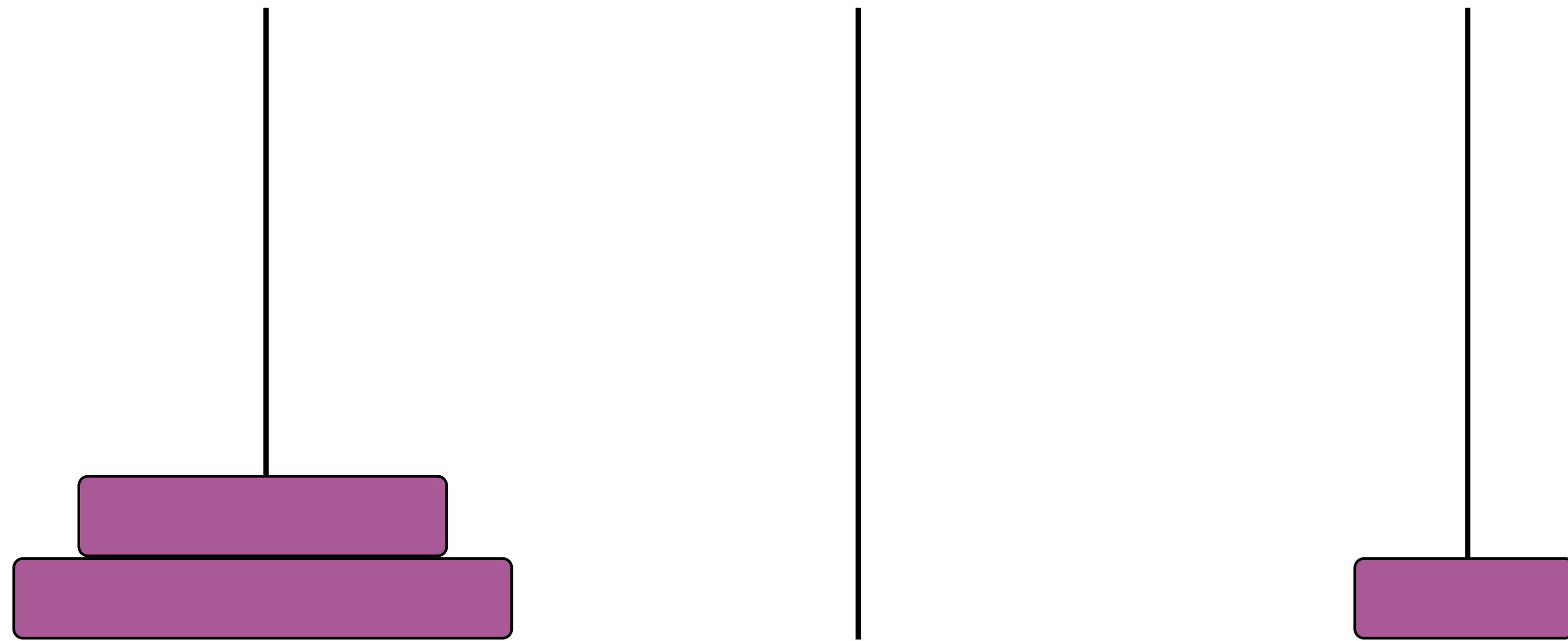
“[The paper shows] that even the latest of these new-fangled *reasoning models* still [...] **fail to reason beyond the distribution** reliably, on a whole bunch of classic problems, like the Tower of Hanoi.”

Tower of Hanoi

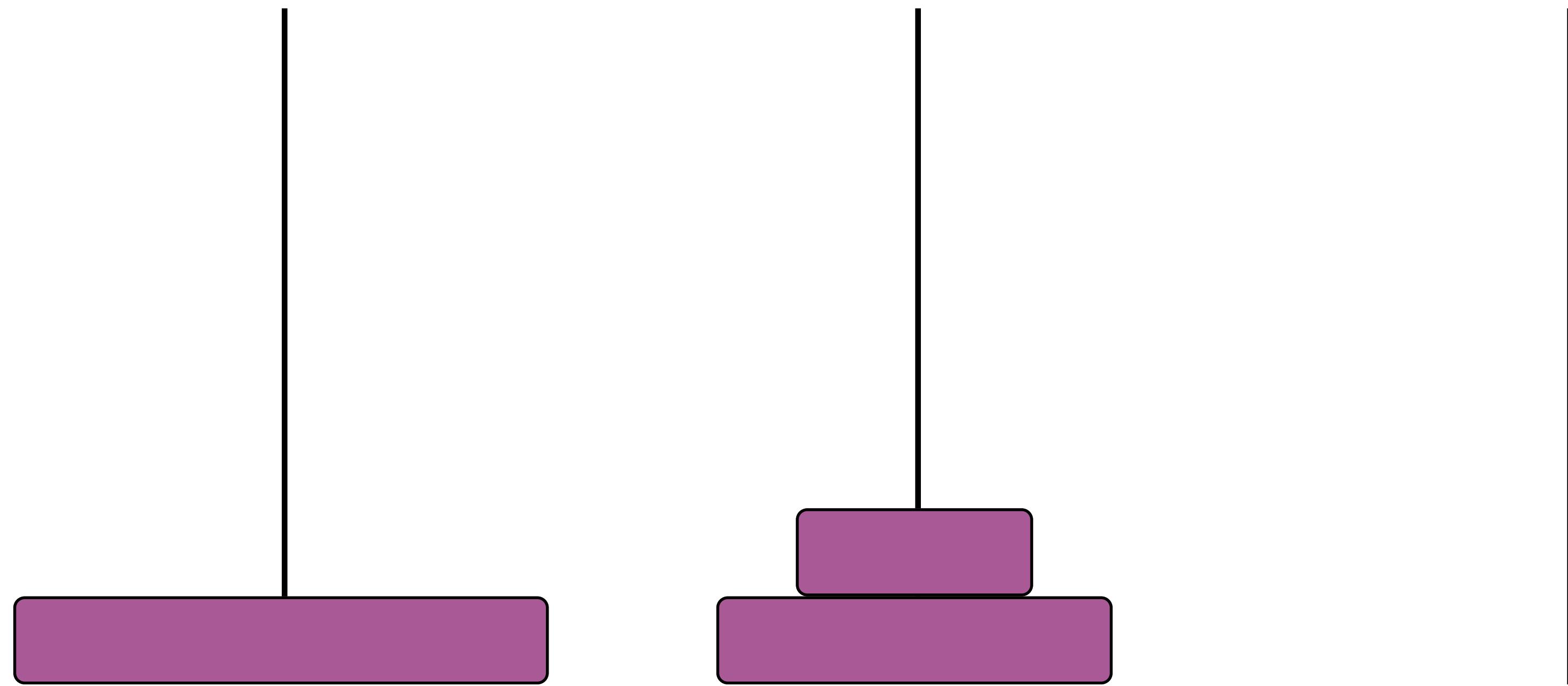


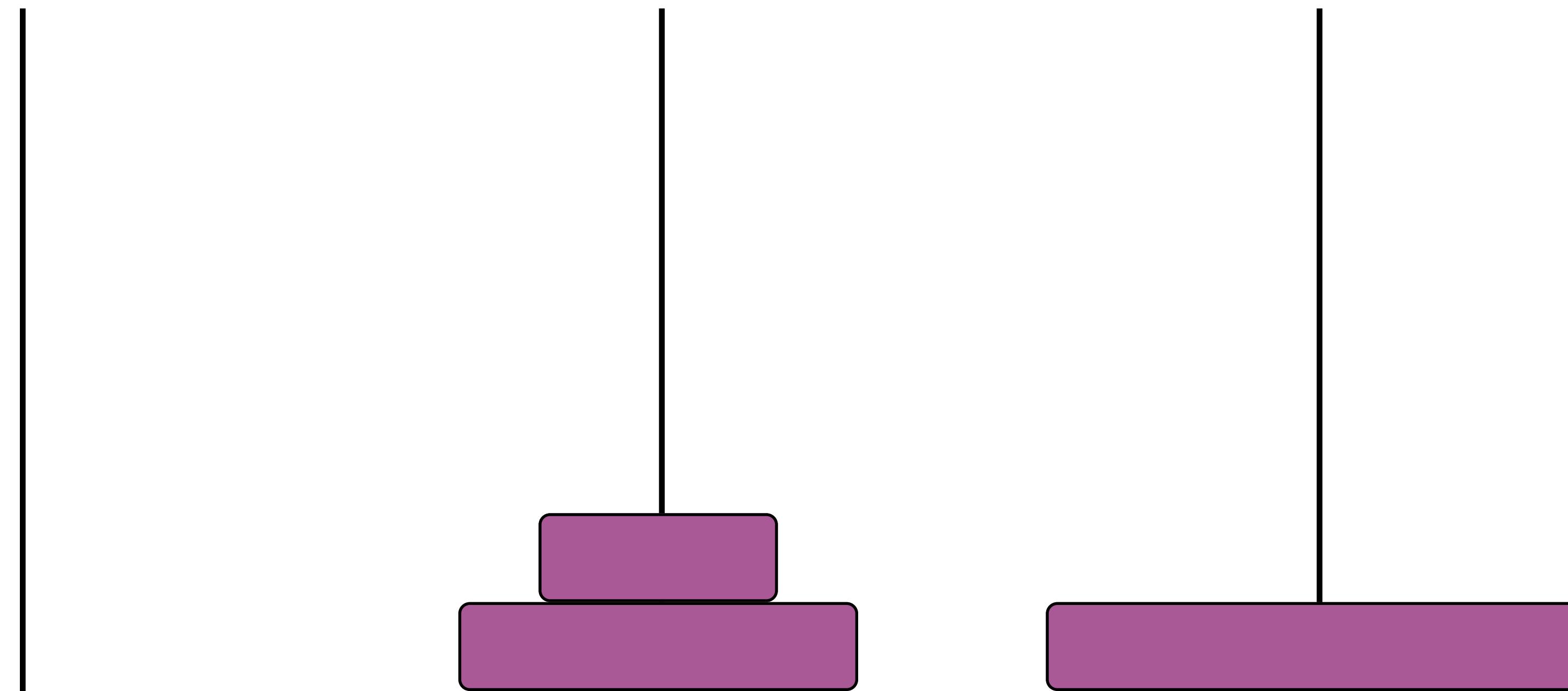
https://en.wikipedia.org/wiki/Tower_of_Hanoi#/media/File:Tower_of_Hanoi.jpeg

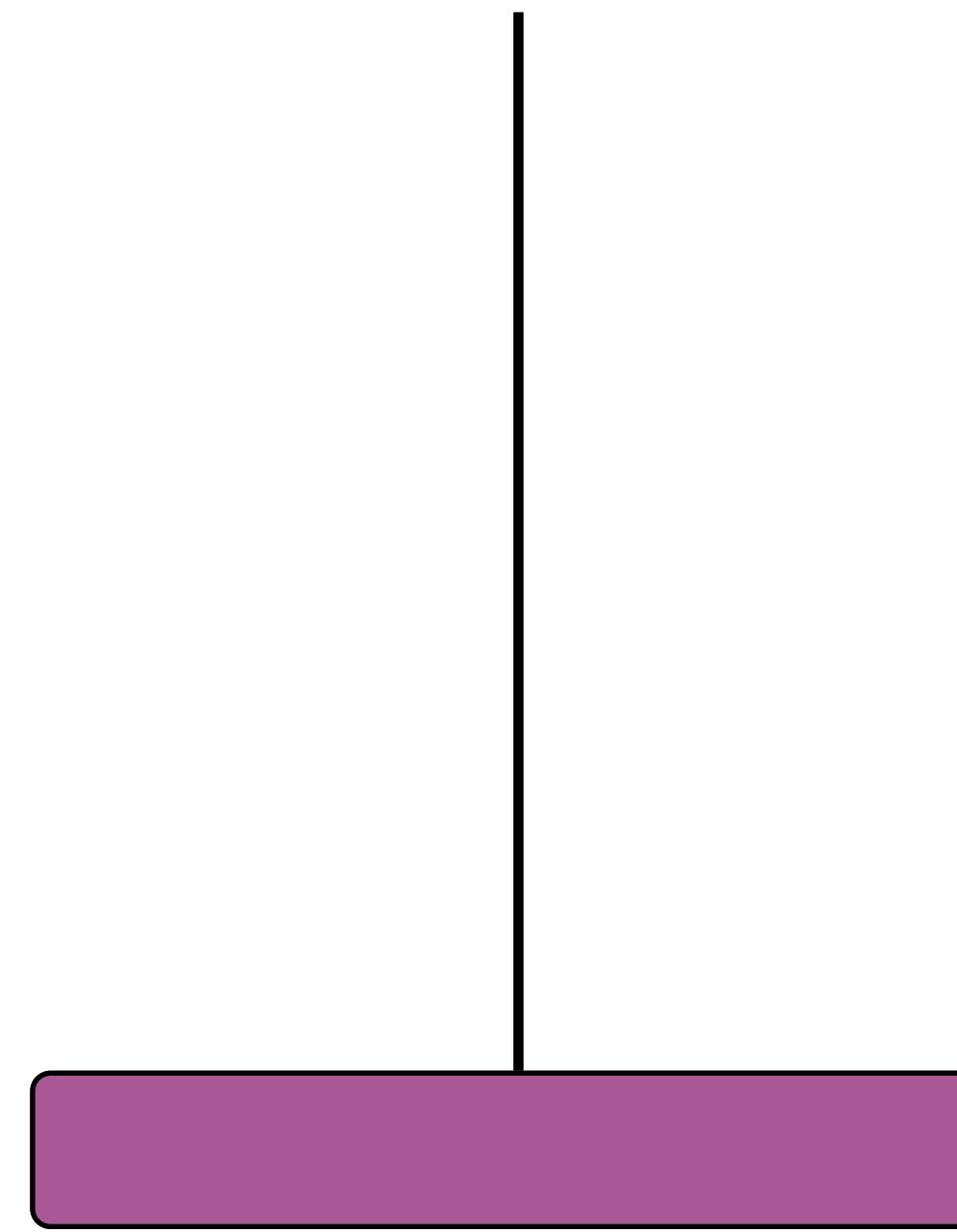
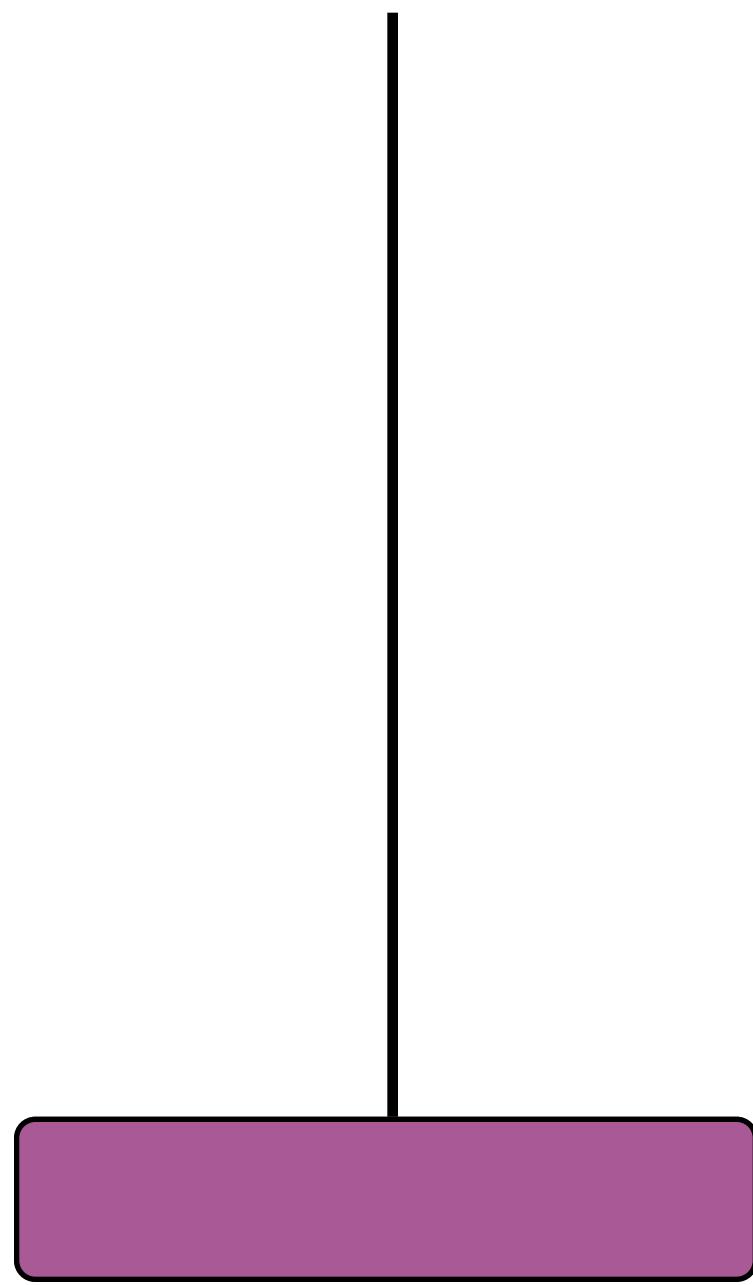


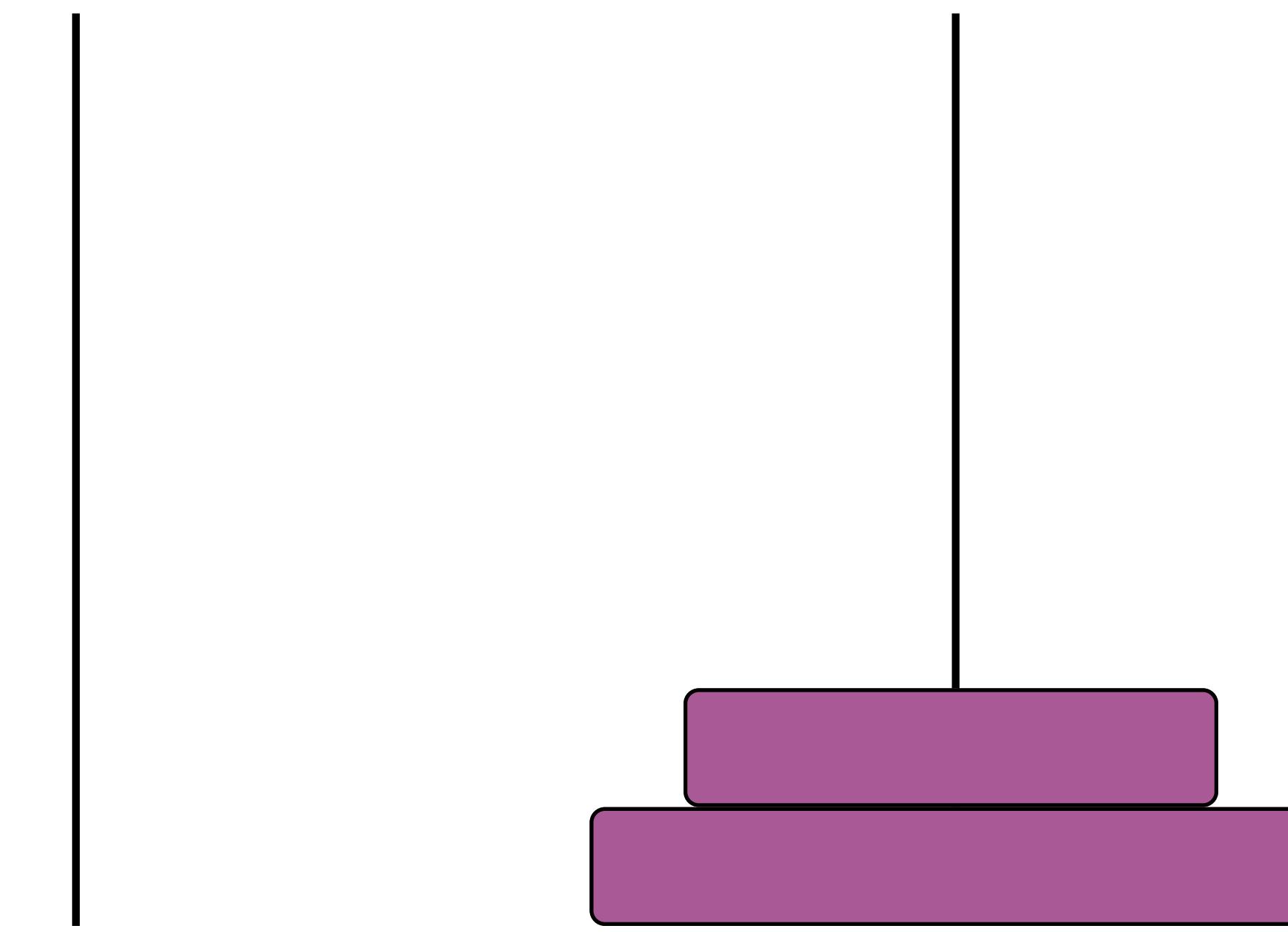
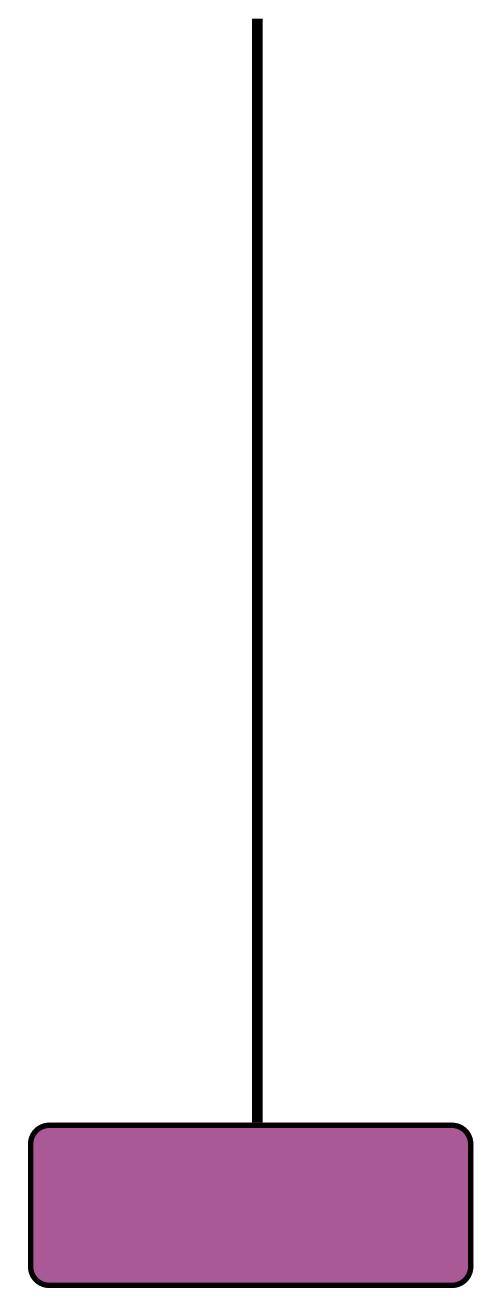


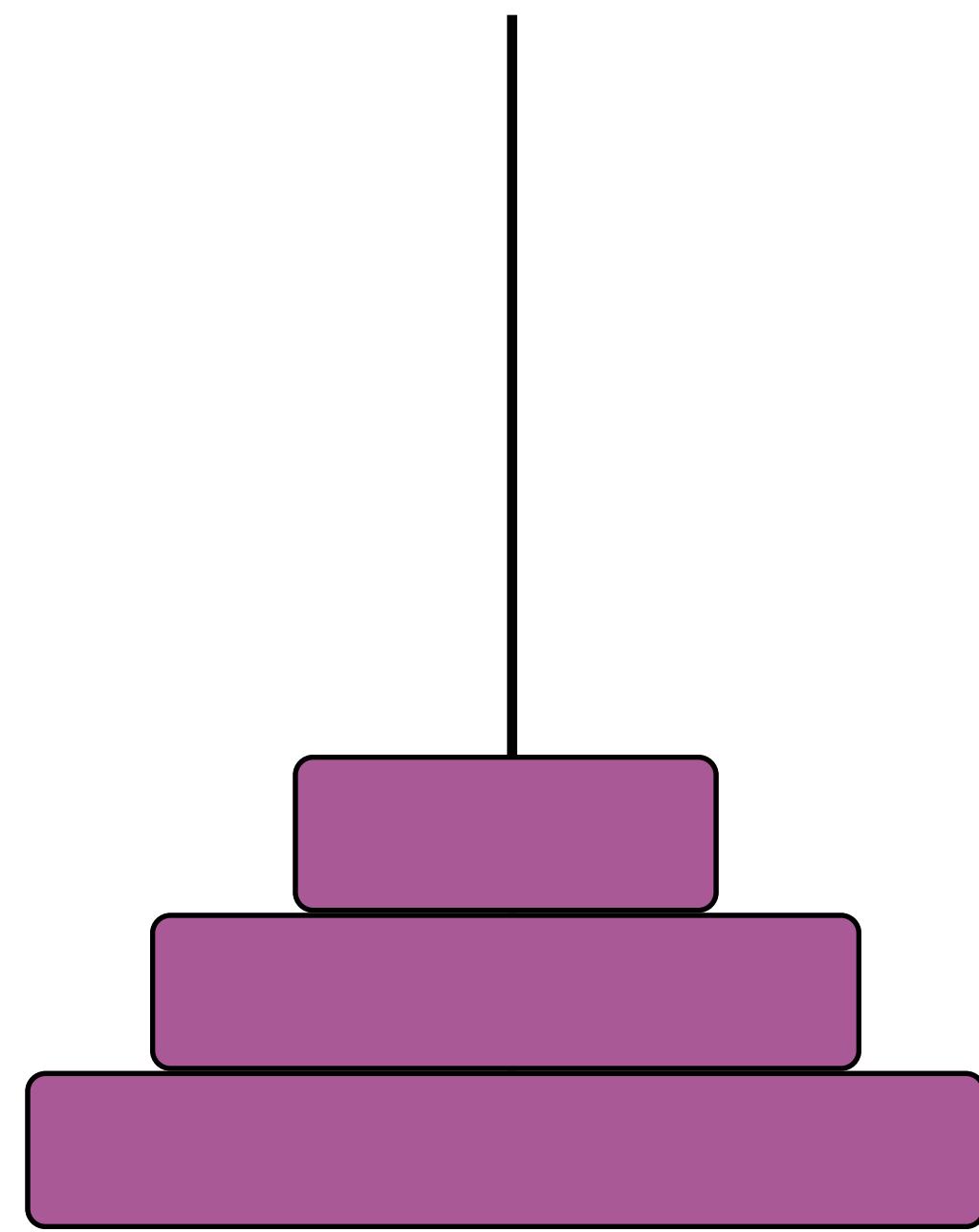








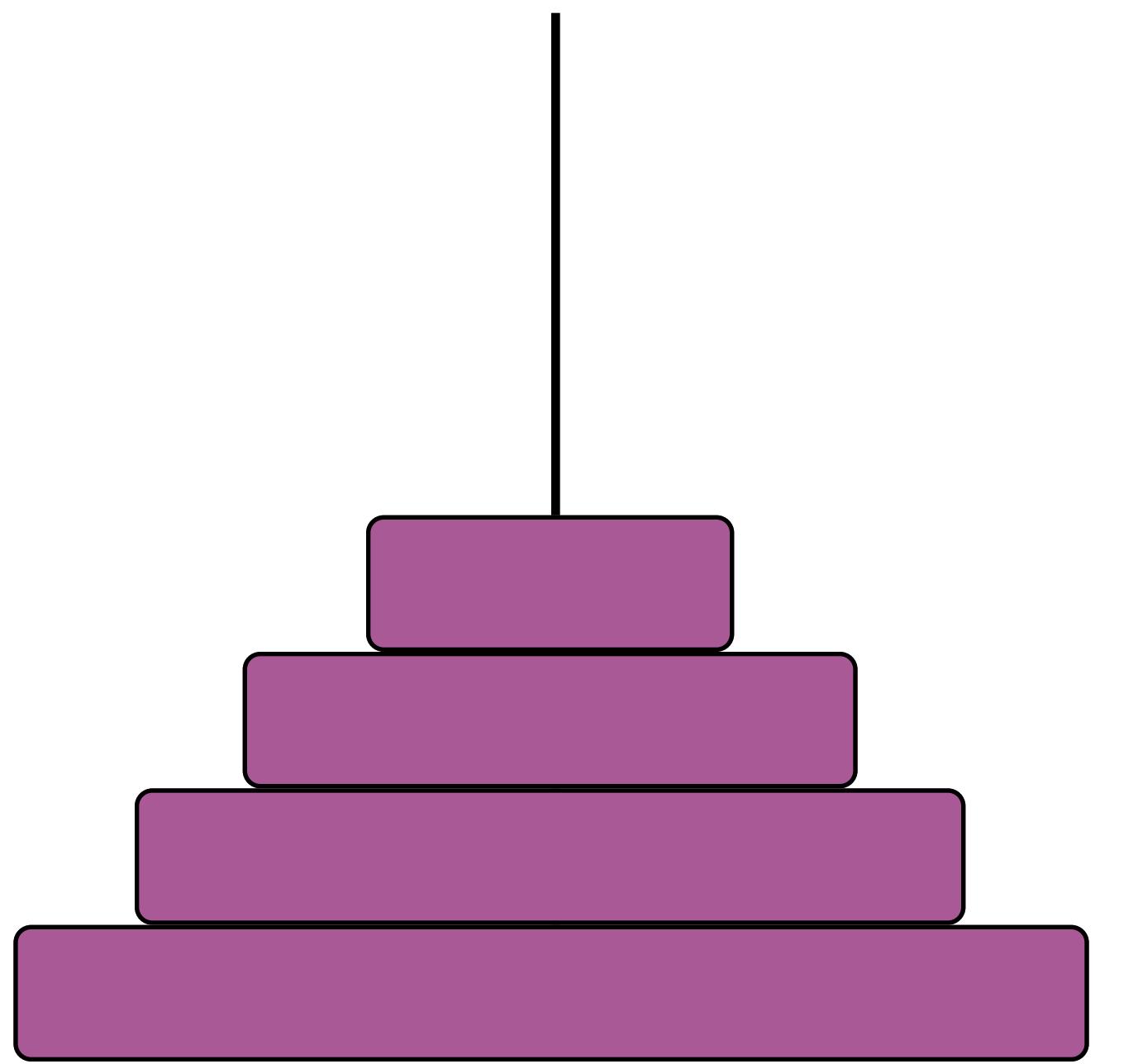


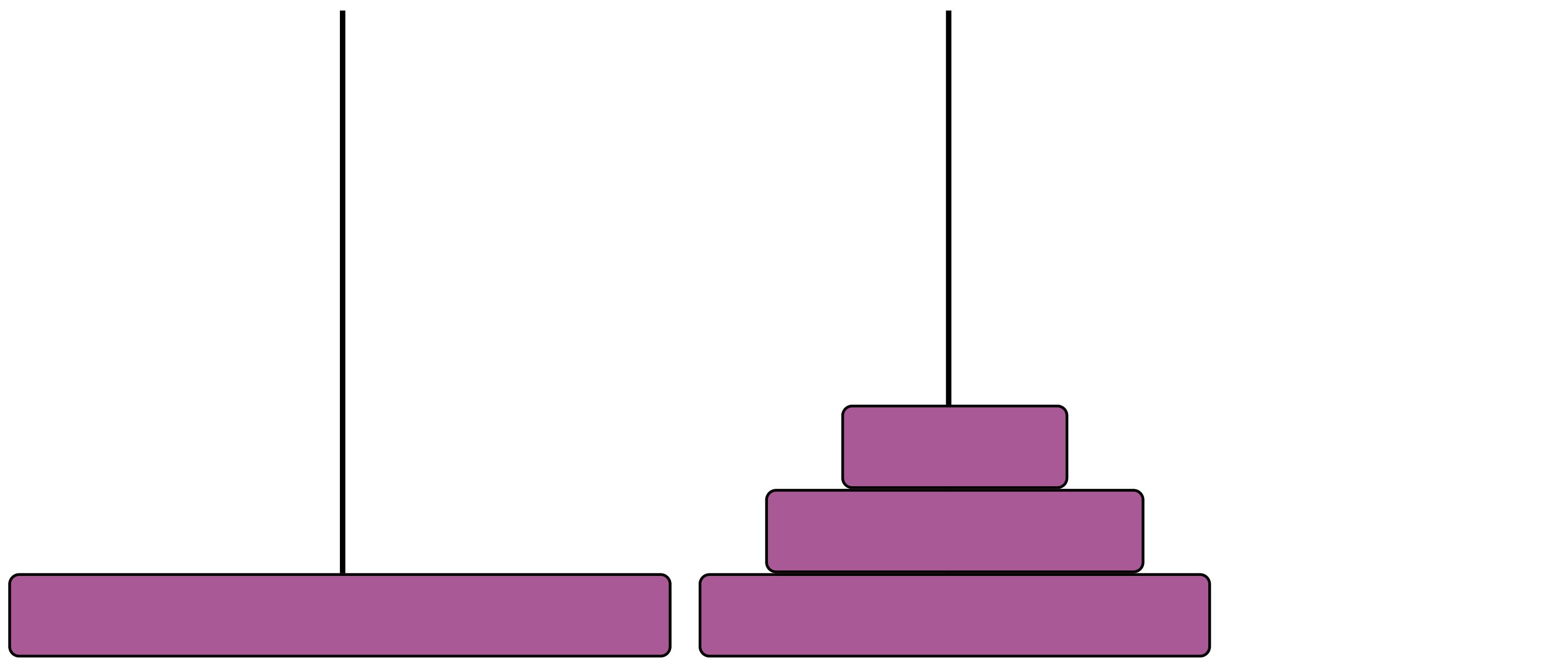


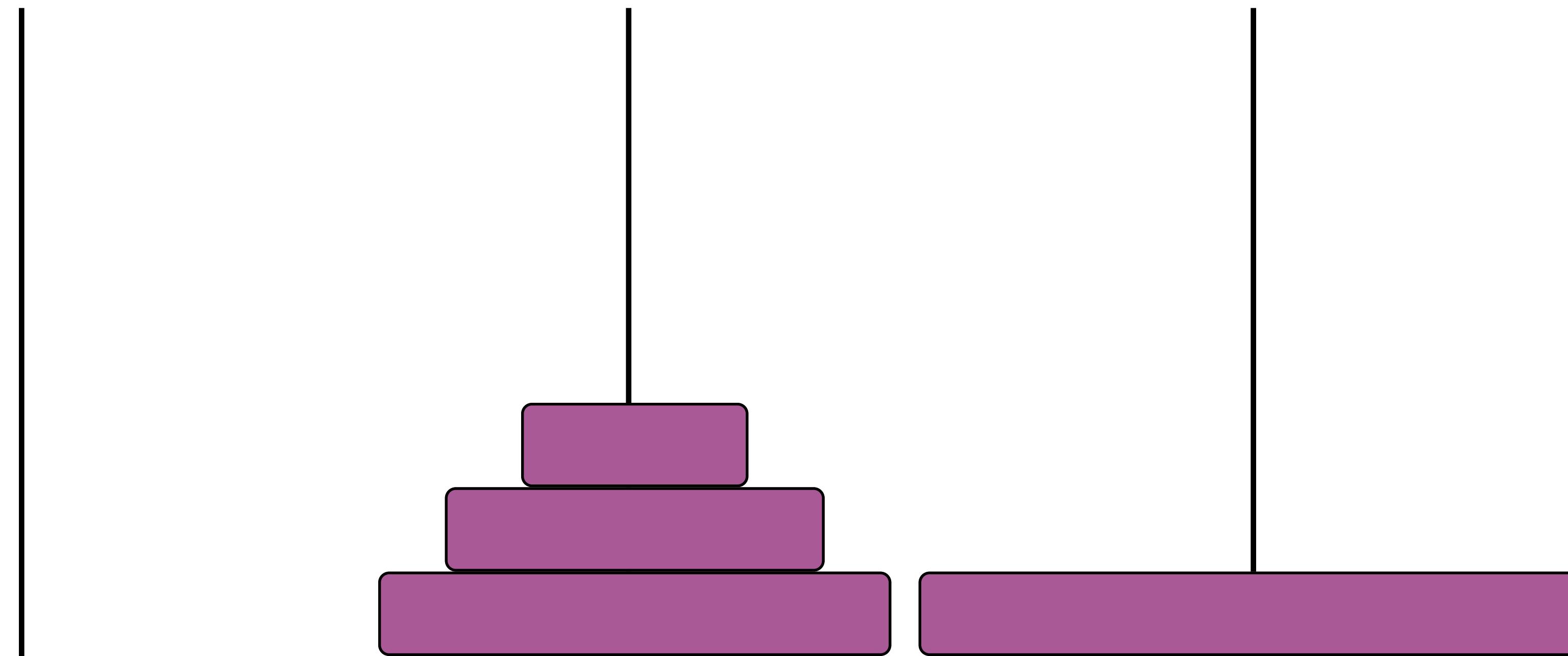
Algorithm 1 Towers of Hanoi

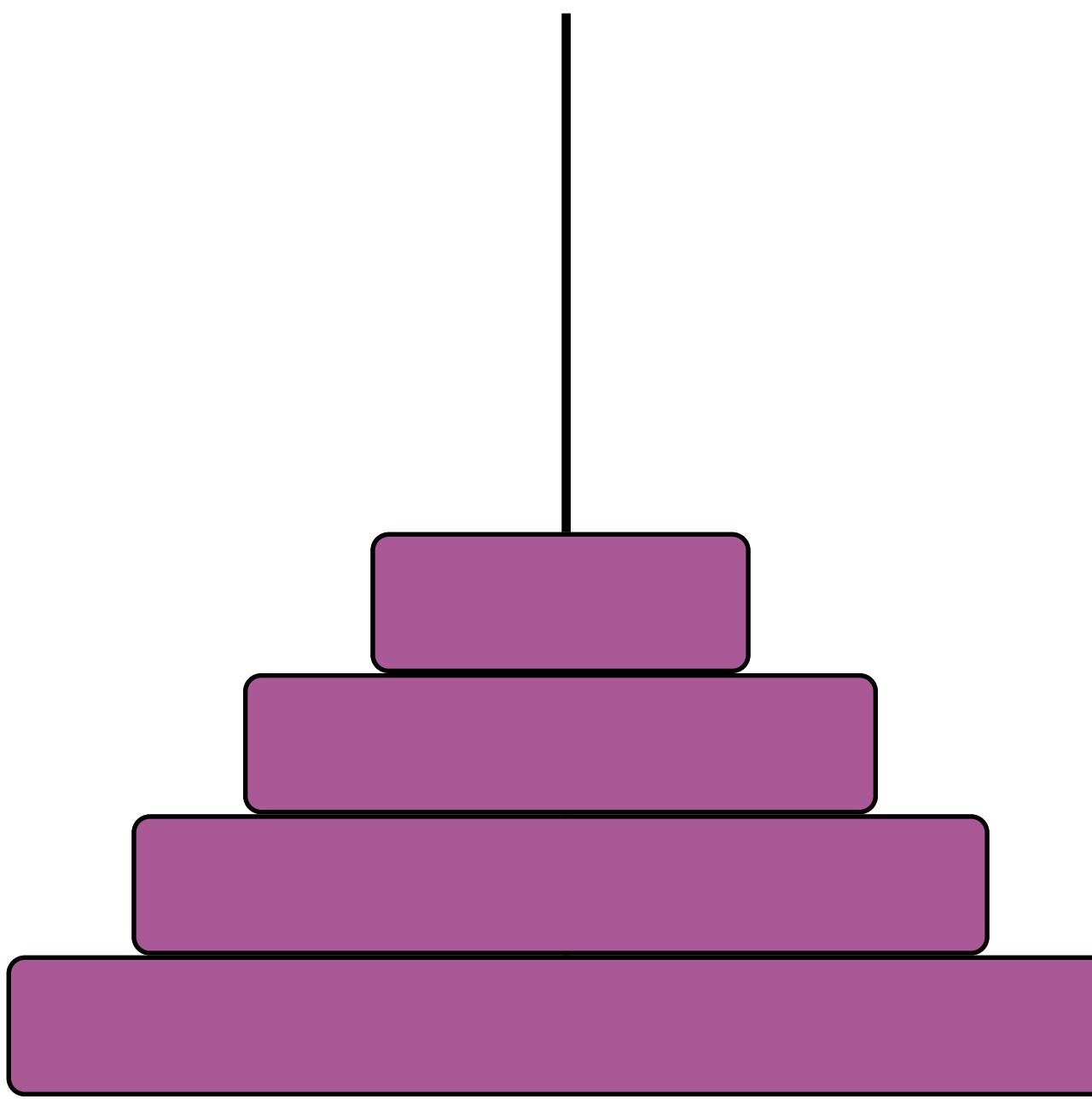
Require: A positive integer n

- 1: Move $n - 1$ from the left to the middle
 - 2: Move the last disk to the right
 - 3: Move $n - 1$ from the middle to the right
-











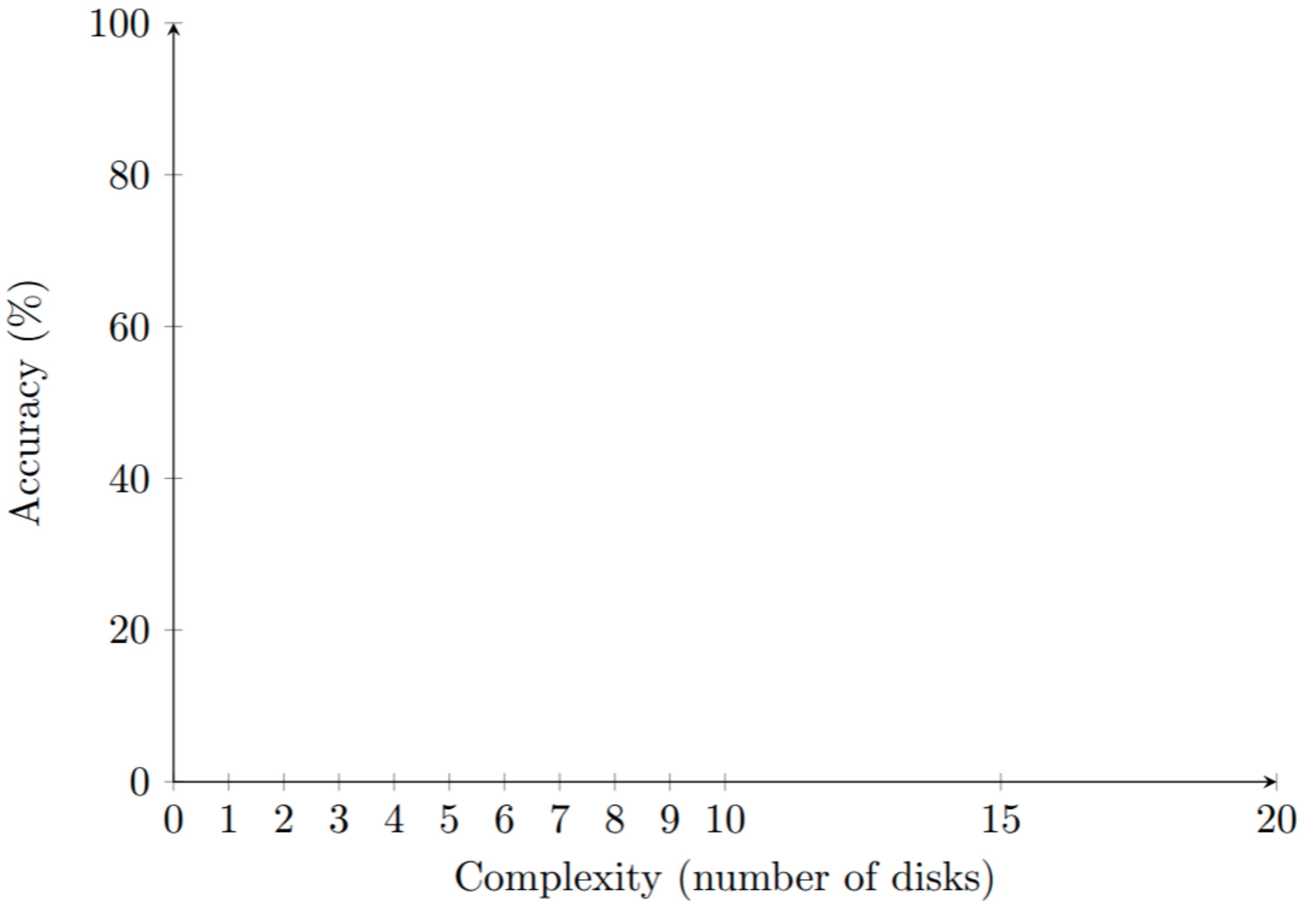
Gemini

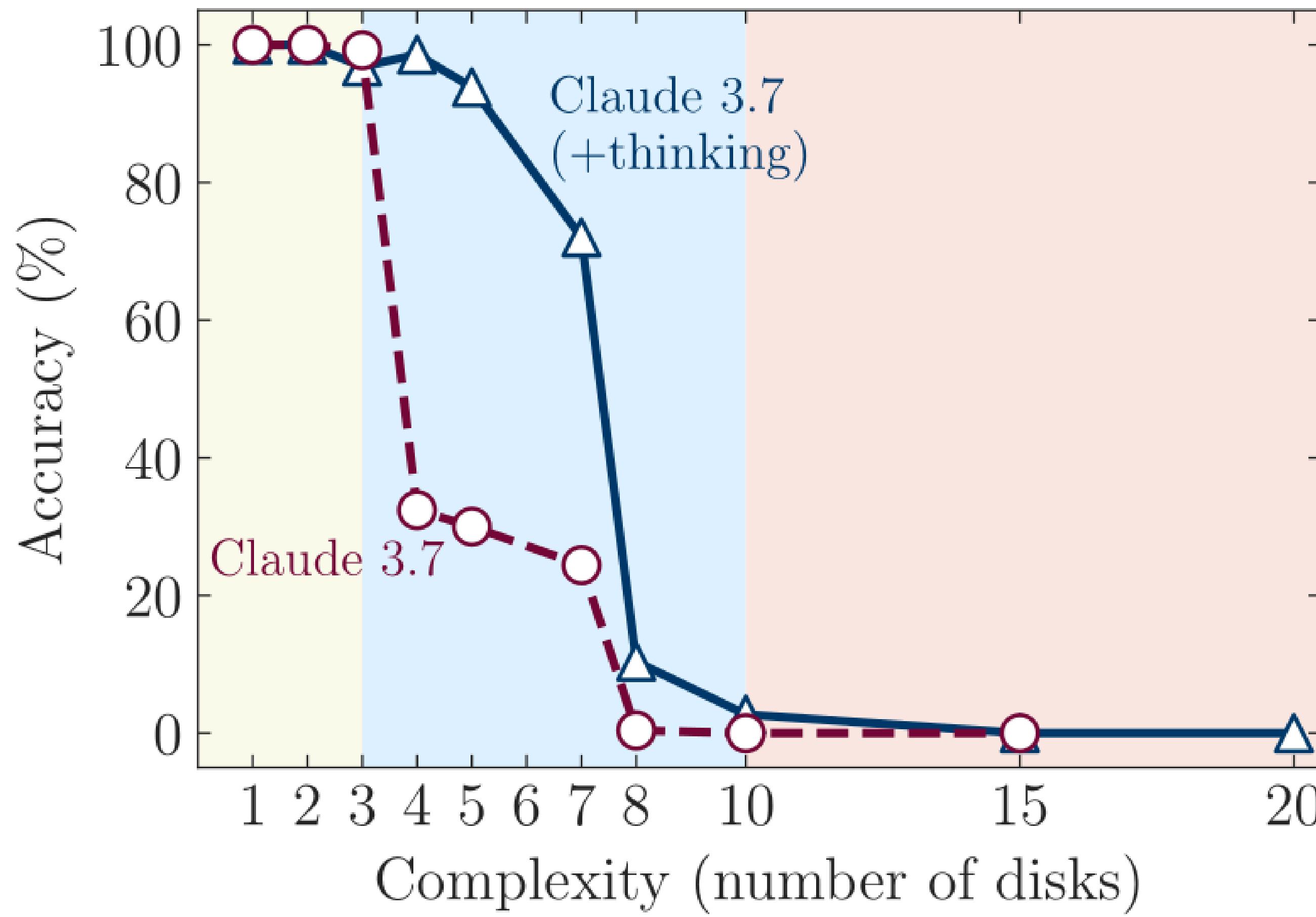
Can they solve it?

Claude

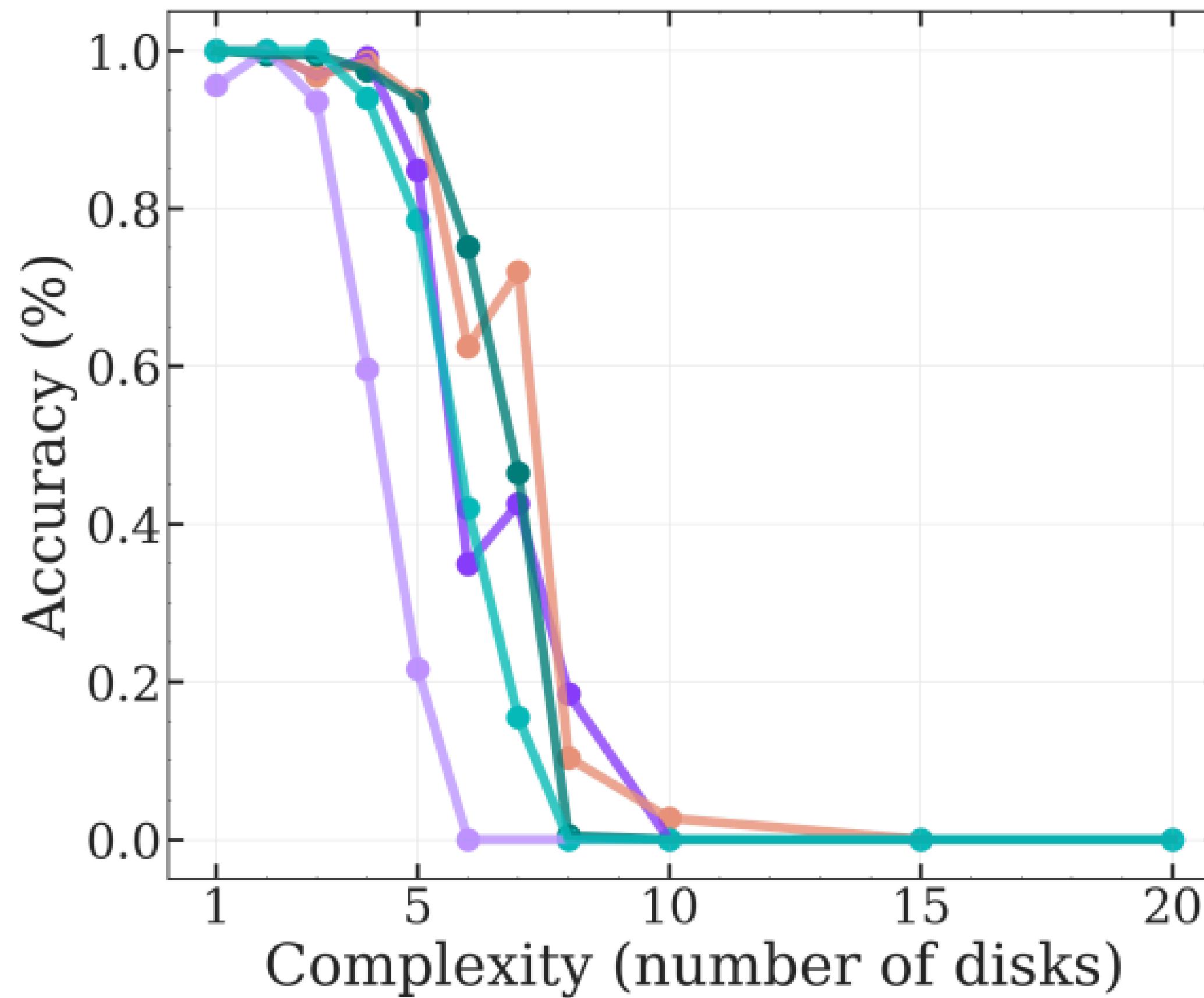


What do you think?





Tower of Hanoi



DeepSeek-R1

Claude-3.7-Sonnet (thinking)

DeepSeek-R1-Distill-Qwen-32B

o3-mini (high)

o3-mini (medium)

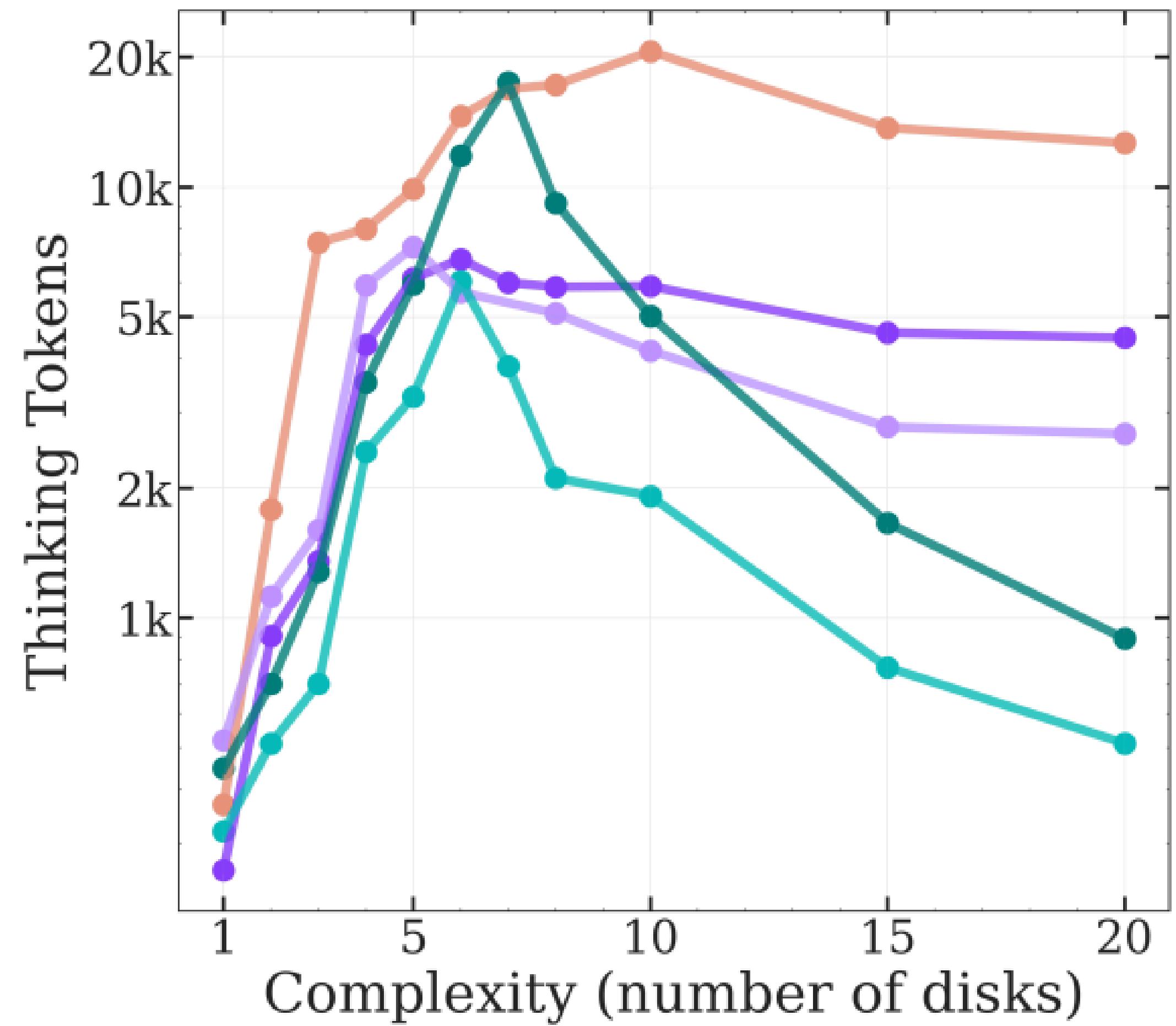
Algorithm 1 Towers of Hanoi

Require: A positive integer n

- 1: Move $n - 1$ from the left to the middle
 - 2: Move the last disk to the right
 - 3: Move $n - 1$ from the middle to the right
-



Maybe the solutions become **too large?**



DeepSeek-R1

Claude-3.7-Sonnet (thinking)

DeepSeek-R1-Distill-Qwen-32B

o3-mini (high)

o3-mini (medium)

Other key findings

“overthinking” on simpler problems

Insufficient symbol manipulation

“[...] we observed their limitations in performing exact computation; for example, when we provided the solution algorithm for the Tower of Hanoi to the models, their performance on this puzzle did not improve.”

Hint for training data dependency

“Moreover, investigating the first failure move of the models revealed surprising behaviors. For instance, they could perform up to 100 correct moves in the Tower of Hanoi but fail to provide more than 5 correct moves in the River Crossing puzzle.”

Marcus concludes:

"But anybody who thinks **LLMs** are a direct route to the sort of **AGI** that could fundamentally transform society for the good is **kidding themselves**.

This does not mean that the field of neural networks is dead, or that deep learning is dead. LLMs are just one form of deep learning, and maybe others — especially **those that play nicer with symbols** – will eventually thrive.

Time will tell. But this particular approach has limits that are clearer by the day."

Outline

1. Marcus' line of argument

2. Critique of his argument

3. Contemporary research

4. Discussion



Gary Marcus ✅
@GaryMarcus

AI is not hitting a wall.

But LLMs probably are (or at least a point of diminishing returns).

We need new approaches, and to diversify the which roads are being actively explored.

Discussion

Are UQOTOMs the right measure?

- MLPs might be better suited for a **different class of** symbol manipulation **problems**.
- MLPs main **purpose** is to approximate functions we don't have a neat **formula** for.

Do recent LLMs support or challenge Marcus claims?

Support:

- '*The Illusions of Thinking*': **LLMs fail to reason** and apply logical steps

Challenge:

- Huge jumps from GPT-2 to GPT-4
- Sam Altman: Bigger models will have better performance (*scaling laws*)

Marcus' argument in a nutshell

- **Premise:** Many algebraic rules are UQOTOM
- **Premise:** Humans can learn/ generalize these rules, so our models should too
- **Conclusion:** Models should be able to **represent and learn UQOTOM** (even with restricted training data)
- **Claim:** MLPs don't generalize UQOTOM well outside the training space (based on observation, argued via training independence)
- **Conclusion:** Vanilla MLPs are **not well suited** for tasks that require **symbol manipulation**
- **Suggestion:** Motivated by computer architecture and supported by case studies, he **suggests** the use of **registers**