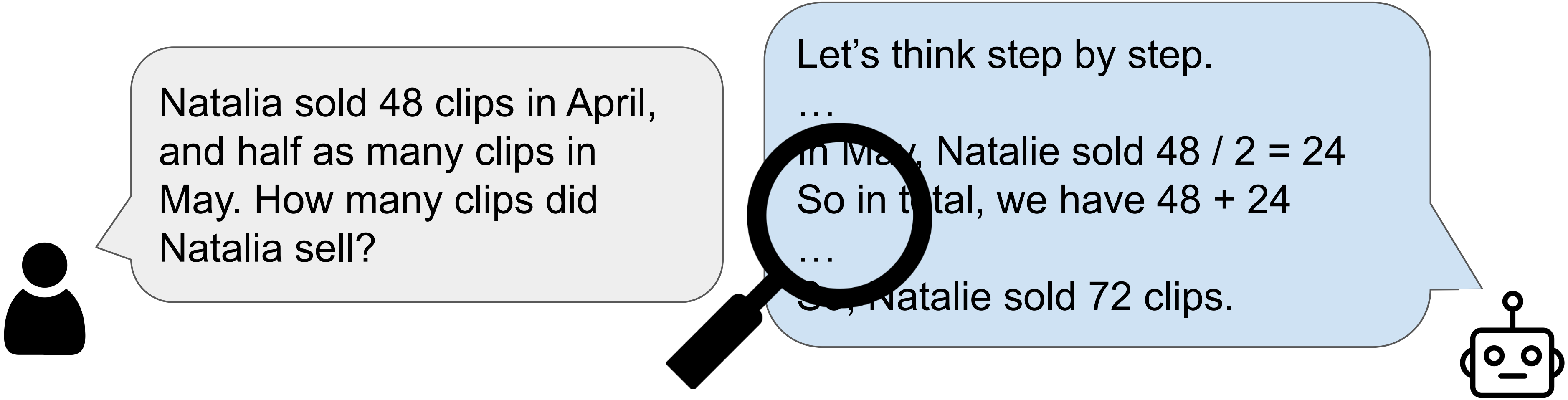
The Geometry of Self-Verification in a Task-Specific Reasoning Model

Andrew Lee, Lihao Sun, Chris Wendler, Fernanda Viegas, Martin Wattenberg

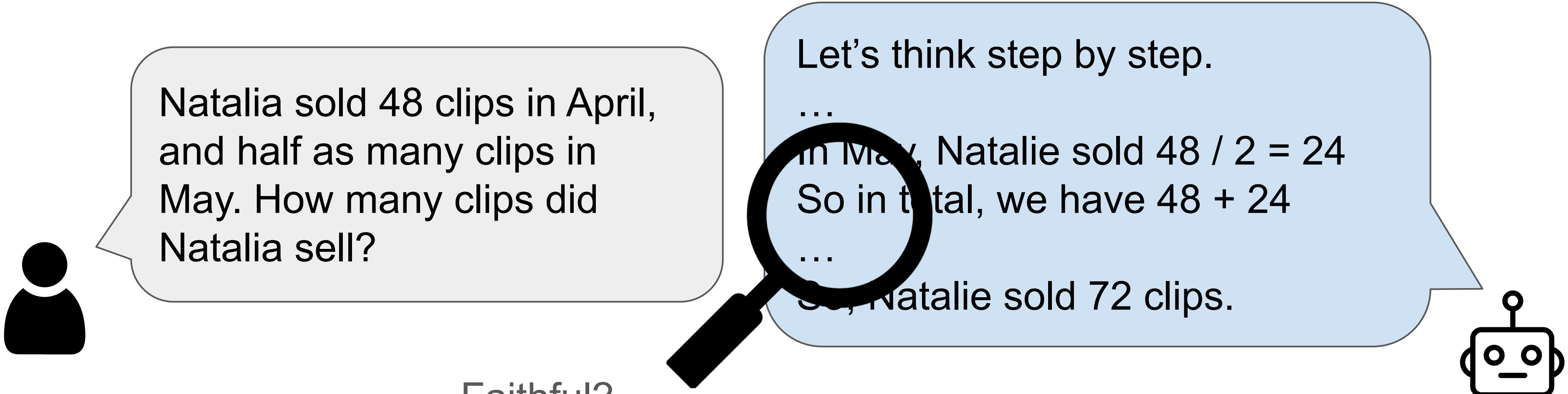
Harvard John A. Paulson School of Engineering and Applied Sciences

insight + interaction lab

Reasoning via CoT is amazing!



But are CoTs “faithful”?



Faithful?

Do these CoT tokens reflect the model’s inner computations?

No! [1, 2, 3, 4]

[1] Lanham et al. “Measuring Faithfulness in Chain-of-Thought Reasoning”. (2024).
[2] Arcuschin et al. “Chain-of-Thought Reasoning In The Wild Is Not Always Faithful. (2025).
[3] Chen et al. Reasoning Models Don’t Always Say What They Think. 2025
[4] Barez et al. Chain-of-Thought is Not Explainability. 2025

Can we monitor & interpret latent state instead?

- Let’s focus on a specific reasoning behavior: Self-verification!
- RQ: How do LMs verify their own answers?
- Let’s train a model on a reasoning task, and study it!

CountDown

- Let's train a model on a specific reasoning task that requires
  - Search
  - Self-verification
- Task: CountDown

Using operands **[20, 14, and 40]**, create an equation that equals **28**.

Solution:
40 / 14 * 20

Training on CountDown

- Train Qwen2.5-3B with DeepSeek R1's recipe
  - Reward when model finds correct solution
  - Reward for correct format (use of <think>, <answer> tags)

Using operands **[20, 14, and 40]**, create an equation that equals **28**.

"**<think>** Let's try different attempts…. **(20 - 14 + 40) = 6 + 40 = 26**. But this does not equal 28, so we need to try another combination. Let's try…

…

Let's try **40 * 14 / 20 = 560 / 20 = 28**. So the answer is 40 * 14 / 20 **</think>**

<answer> 40 * 14 / 20 </answer>

Benefit of Setup: Mode Collapse!

- Preference tuning (RL) leads to mode collapse [1, 2, 3, 4]:
  - Model over-weighs majority preferences
  - Sacrifices diversity
  - (which is why all LMs sound the same?)
- We can leverage this as a feature, not a bug!

[1] Kirk et al. “Understanding the Effects of RLHF on LLM Generalisation and Diversity”. 2023
[2] Murthy et al. “One fish, two fish, but not the whole sea: Alignment reduces language models’ conceptual diversity.” 2024.
[3] Padmakumar et al. “Does Writing with Language Models Reduce Content Diversity?”. 2023.
[4] Slocum et al. “Diverse preference learning for capabilities and alignment”. 2025.

Mode Collapse

Using operands **[20, 14, and 40]**, create an equation that equals **28**.

Mode collapse!
Model **always** generates
highly structured CoT

Makes it easier to parse + study!

~20 g

ttempts…. **(20 - 14 + 40) = 6 + 40 = 26**. But this
ed to try another combination. I could try adding
ich gives me 74, so that's not it either…
…
/ 20 = 28. So the answer is 40 * 14 / 20 </think>
er> 40 * 14 / 20 </answer>

~300 gradient steps

<think>
40 - 14 - 20 = 10 - 20 = -10 **(not 28)**
40 - 14 + 20 = 26 + 20 = 46 **(not 28)**
40 + 14 - 20 = 54 - 20 = 34 **(not 28)**
40 + 14 + 20 = 54 + 20 = 74 **(not 28)**
40 * 14 / 20 = 560 / 20 = 28 (this works)
So, the equation that equals 28 is 40 * 14 / 20.
</think>
<answer>(40 * 14) / 20 </answer>

Mode Collapse

Using operands **[20, 14, and 40]**, create an equation that equals **28**.

~20 gradient steps

“<think> Let’s try different attempts…. **(20 - 14 + 40) = 6 + 40 = 26**. But this does not equal 28, so we need to try another combination. I could try adding 20 to 14 and 40, which gives me 74, so that’s not it either…

…

Le e answer is 40 * 14 / 20 </think>

</answer>

What happens at these timesteps?

~300 gradient steps

<think>

40 - 14 - 20 = 10 - 20 = -10 **(not 28)**

40 - 14 + 20 = 26 + 20 = 46 **(not 28)**

40 + 14 - 20 = 54 - 20 = 34 **(not 28)**

40 + 14 + 20 = 54 + 20 = 74 **(not 28)**

40 * 14 / 20 = 560 / 20 = 28 (this works)

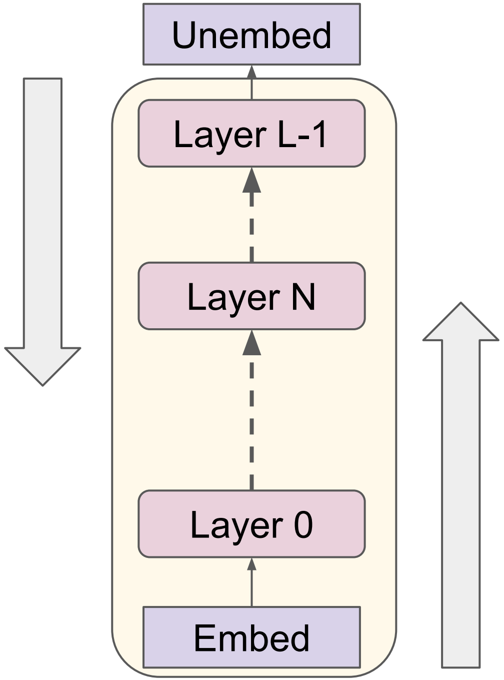
So, the equation that equals 28 is 40 * 14 / 20.

</think>

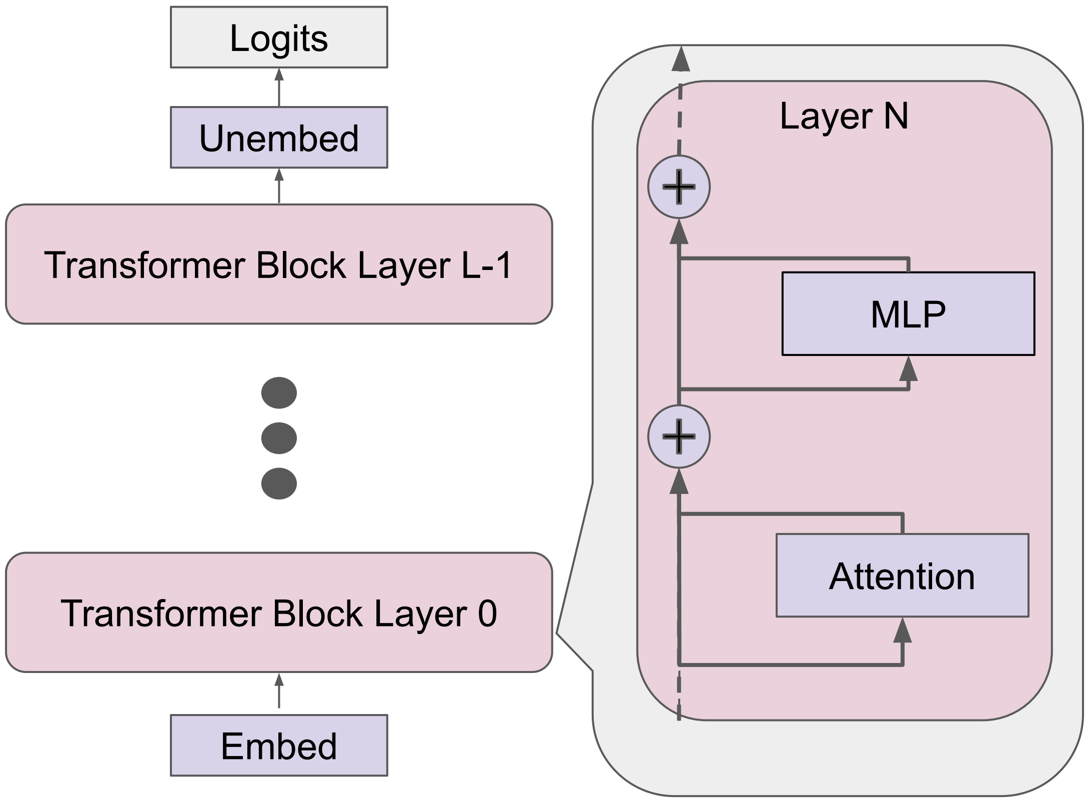
<answer>(40 * 14) / 20 </answer>

How does the model verify its solutions?

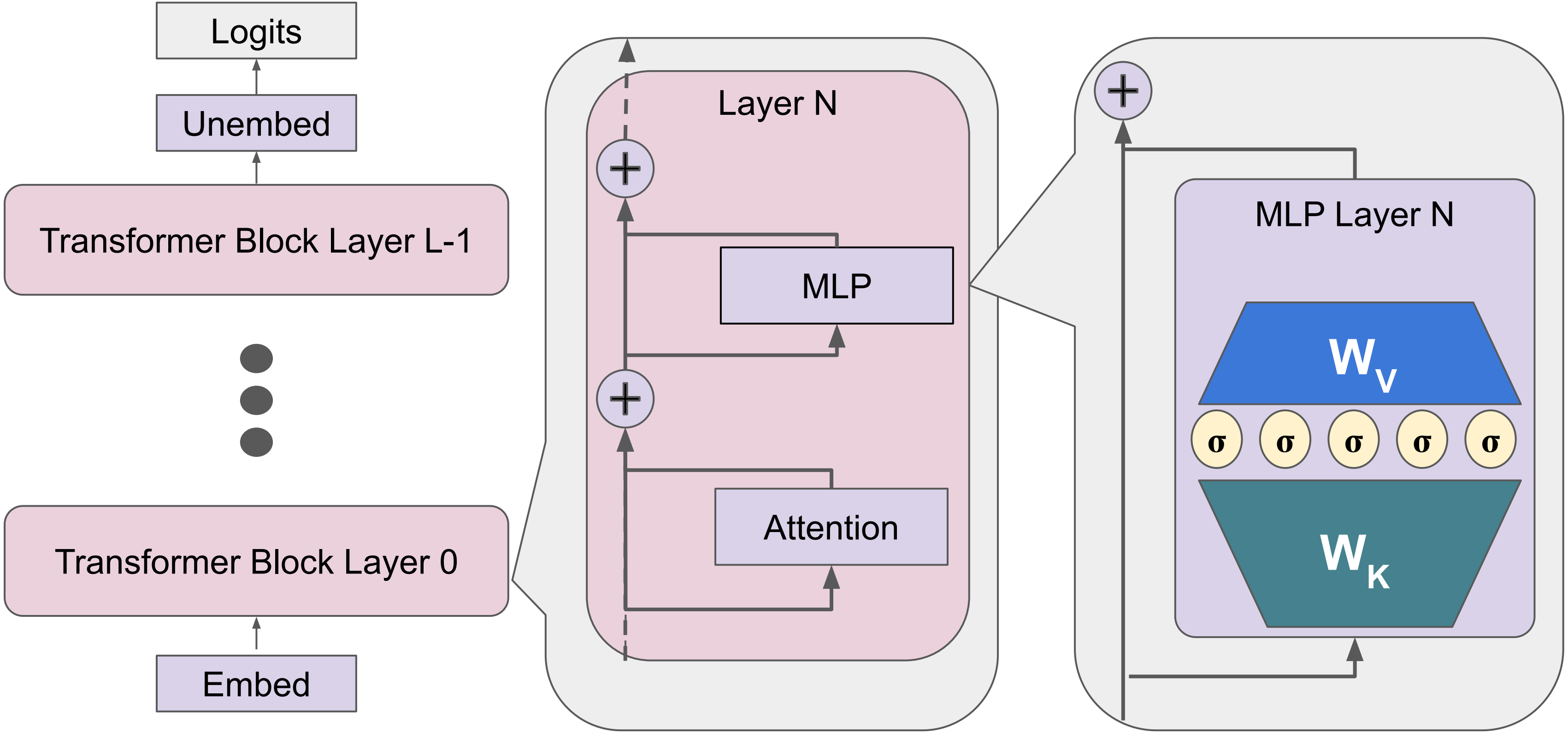
- “Top-down” analysis
- “Bottom-up” analysis
- Analyses meeting in the middle



Revisiting Transformers



Revisiting Transformers

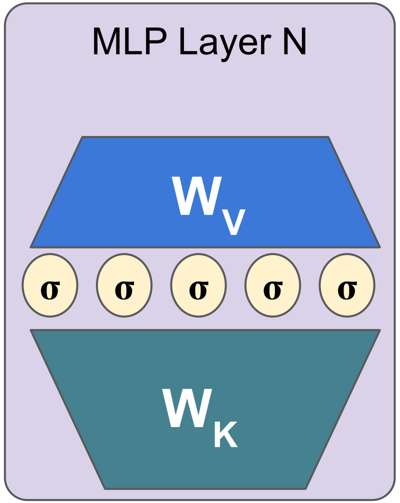


MLPs

- Rather than thinking of MLP as…

$$\mathrm{MLP}^{\ell}(\mathbf{x}^{\ell}) = \sigma(W_K^{\ell}\mathbf{x}^{\ell})W_V^{\ell}$$

$\mathbb{R}^{h\times d}$ $\mathbb{R}^{d}$ $\mathbb{R}^{h\times d}$

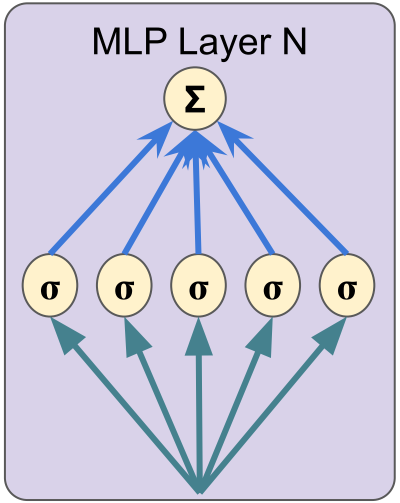


MLPs

- Break it down by rows, columns!

$$\cancel{\mathrm{MLP}^{\ell}(\mathbf{x}^{\ell}) = \sigma(W_K^{\ell}\mathbf{x}^{\ell})W_V^{\ell}}$$

$$\mathrm{MLP}^{\ell}(\mathbf{x}^{\ell}) = \sum^{h} \sigma(\mathbf{x}^{\ell} \cdot \mathbf{k}_i^{\ell})\mathbf{v}_i^{\ell}$$



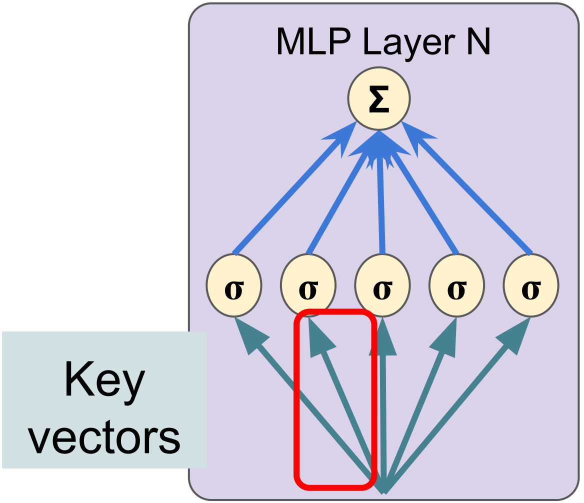
MLPs

- Break it down by rows, columns!

$$\cancel{\mathrm{MLP}^{\ell}(\mathbf{x}^{\ell}) = \sigma(W_K^{\ell}\mathbf{x}^{\ell})W_V^{\ell}}$$

$$\mathrm{MLP}^{\ell}(\mathbf{x}^{\ell}) = \sum^{h} \sigma(\mathbf{x}^{\ell} \cdot \boxed{\mathbf{k}_i^{\ell}})\mathbf{v}_i^{\ell}$$

i’th row of W_K



MLPs

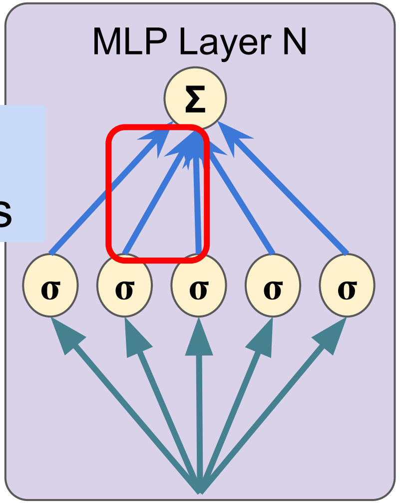
- Break it down by rows, columns!

$$\cancel{\mathrm{MLP}^{\ell}(\mathbf{x}^{\ell}) = \sigma(W_K^{\ell}\mathbf{x}^{\ell})W_V^{\ell}}$$

$$\mathrm{MLP}^{\ell}(\mathbf{x}^{\ell}) = \sum^{h} \sigma(\mathbf{x}^{\ell} \cdot \mathbf{k}_i^{\ell})\boxed{\mathbf{v}_i^{\ell}}$$

i’th column of W_V

Value vectors



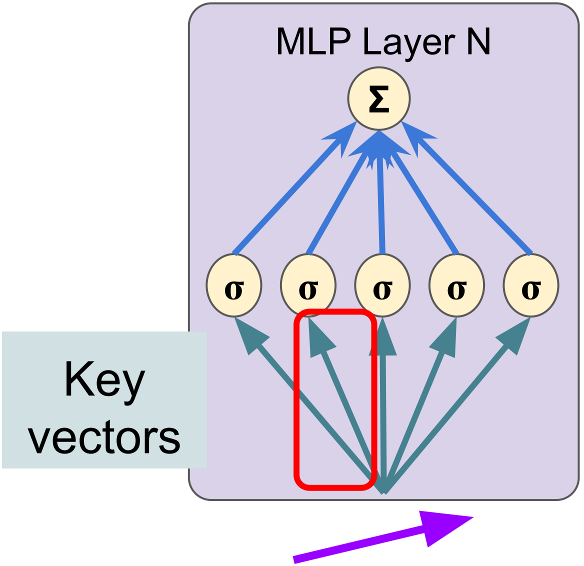
MLPs

- Break it down by rows, columns!

$$\cancel{\mathrm{MLP}^\ell(\mathbf{x}^\ell) = \sigma(W_K^\ell \mathbf{x}^\ell) W_V^\ell}$$

$$\mathrm{MLP}^\ell(\mathbf{x}^\ell) = \sum^{h} \sigma\boxed{(\mathbf{x}^\ell \cdot \mathbf{k}_i^\ell)}\mathbf{v}_i^\ell$$

Scalar



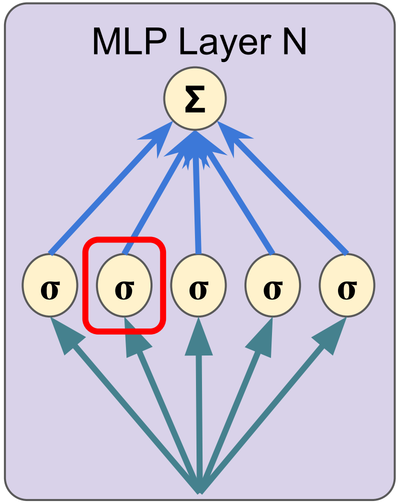
MLPs

- Break it down by rows, columns!

$$\cancel{\mathrm{MLP}^\ell(\mathbf{x}^\ell) = \sigma(W_K^\ell \mathbf{x}^\ell) W_V^\ell}$$

$$\mathrm{MLP}^\ell(\mathbf{x}^\ell) = \sum^{h} \boxed{\sigma(\mathbf{x}^\ell \cdot \mathbf{k}_i^\ell)} \mathbf{v}_i^\ell$$

Scalar



MLPs

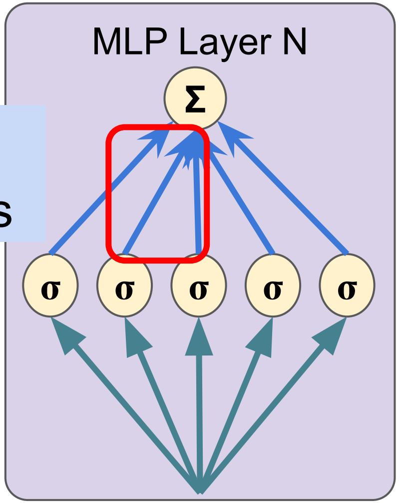
- Break it down by rows, columns!

$$\cancel{\mathrm{MLP}^{\ell}(\mathbf{x}^{\ell}) = \sigma(W_K^{\ell}\mathbf{x}^{\ell})W_V^{\ell}}$$

$$\mathrm{MLP}^{\ell}(\mathbf{x}^{\ell}) = \sum^{h} \boxed{\sigma(\mathbf{x}^{\ell} \cdot \mathbf{k}_i^{\ell})\mathbf{v}_i^{\ell}}$$

Scale a value vector

Value vectors



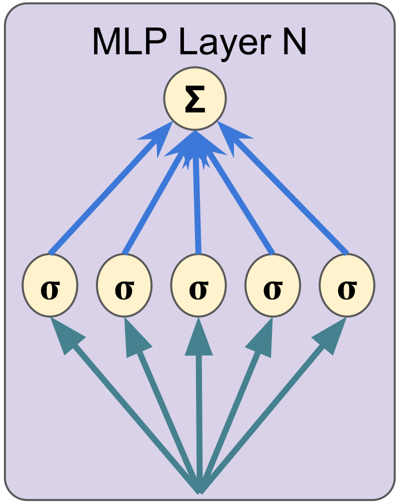
MLPs

- Break it down by rows, columns!

$$\cancel{\mathrm{MLP}^\ell(\mathbf{x}^\ell) = \sigma(W_K^\ell \mathbf{x}^\ell) W_V^\ell}$$

$$\mathrm{MLP}^\ell(\mathbf{x}^\ell) = \sum^h \sigma(\mathbf{x}^\ell \cdot \mathbf{k}_i^\ell)\mathbf{v}_i^\ell$$

$$= \sum^h m_i \mathbf{v}_i^\ell, \;\; m_i = \sigma(\mathbf{x}^\ell \cdot \mathbf{k}_i^\ell)$$



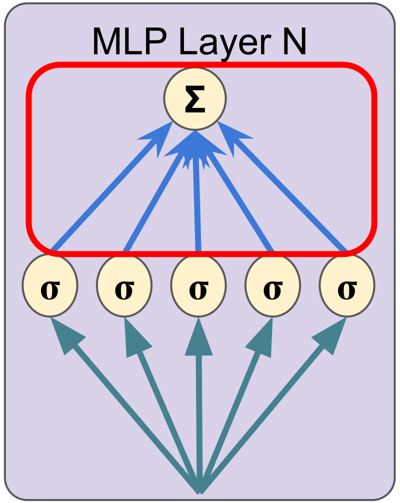
MLPs

- Break it down by rows, columns!

$$\cancel{\mathrm{MLP}^\ell(\mathbf{x}^\ell) = \sigma(W_K^\ell \mathbf{x}^\ell) W_V^\ell}$$

$$\mathrm{MLP}^\ell(\mathbf{x}^\ell) = \sum^h \sigma(\mathbf{x}^\ell \cdot \mathbf{k}_i^\ell) \mathbf{v}_i^\ell$$

$$= \boxed{\sum^h m_i \mathbf{v}_i^\ell}, \quad m_i = \sigma(\mathbf{x}^\ell \cdot \mathbf{k}_i^\ell)$$



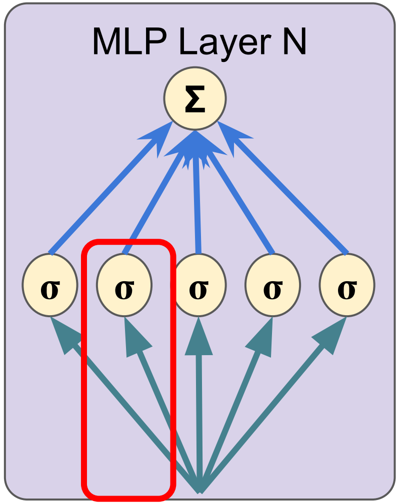
MLPs

- Break it down by rows, columns!

$$\cancel{\mathrm{MLP}^\ell(\mathbf{x}^\ell) = \sigma(W_K^\ell \mathbf{x}^\ell) W_V^\ell}$$

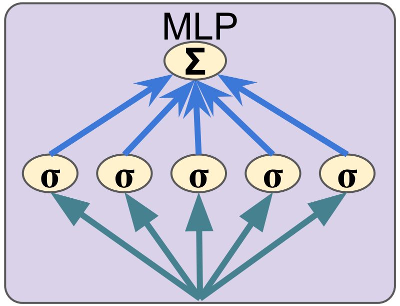
$$\mathrm{MLP}^\ell(\mathbf{x}^\ell) = \sum^h \sigma(\mathbf{x}^\ell \cdot \mathbf{k}_i^\ell)\mathbf{v}_i^\ell$$

$$= \sum^h m_i \mathbf{v}_i^\ell, \quad \boxed{m_i = \sigma(\mathbf{x}^\ell \cdot \mathbf{k}_i^\ell)}$$

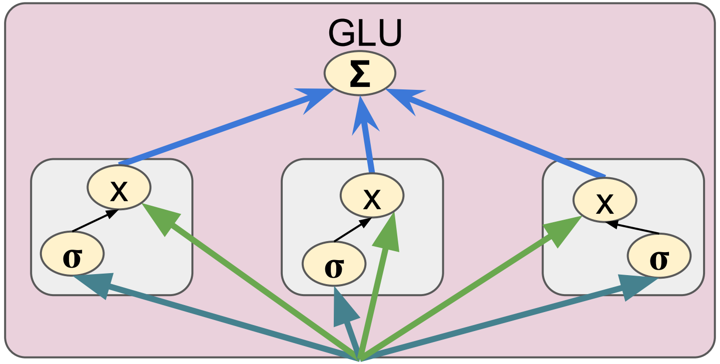


Gated Linear Units (GLUs)

$$\mathrm{MLP}(\mathbf{x}) = \sum_{i}^{d_{mlp}} \boxed{\sigma(\mathbf{x} \cdot \mathbf{k}_i)} \mathbf{v}_i$$



$$\mathrm{GLU}(\mathbf{x}) = \sum_{i}^{d_{glu}} \boxed{\left(\sigma(W_i^{\text{gate}}\mathbf{x}) \cdot W_i^{\text{up}}\mathbf{x}\right)} \mathbf{v}_i$$

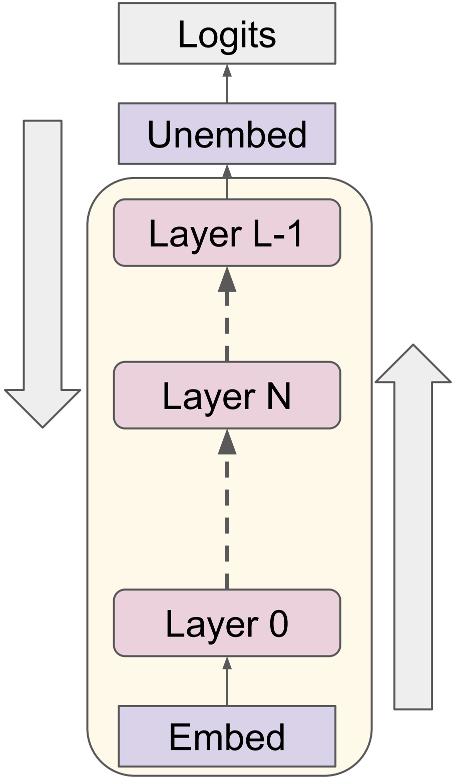


How does R1 self-verify?

Using operands **[20, 14, and 40]**, create an equation that equals **28**.

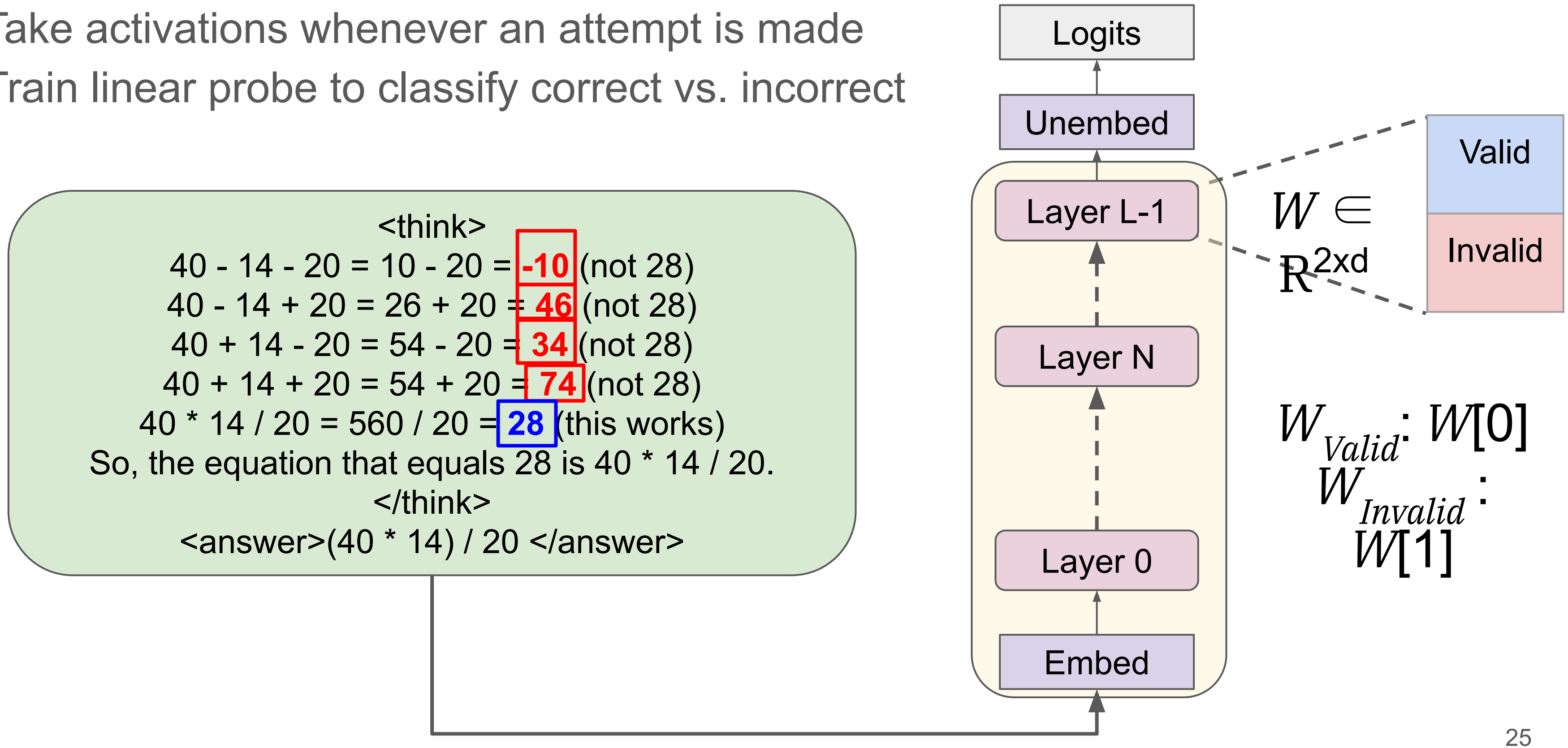
<think>
40 - 14 - 20 = 10 - 20 = -10 **(not 28)**
40 - 14 + 20 = 26 + 20 = 46 **(not 28)**
40 + 14 - 20 = 54 - 20 = 34 **(not 28)**
40 + 14 + 20 = 54 + 20 = 74 **(not 28)**
40 * 14 / 20 = 560 / 20 = 28 **(this works)**
So, the equation that equals 28 is 40 * 14 / 20.
</think>
<answer>(40 * 14) / 20 </answer>

What happens at these timesteps?



Top-Down: Probing for Correctness

- Take activations whenever an attempt is made
- Train linear probe to classify correct vs. incorrect



Steering with Probe

- Near perfect probing accuracy
  - Linear separability between $\mathbf{x}_{Valid}$, $\mathbf{x}_{Invalid}$
- We can use probe (W_{Valid}) to steer model

Original CoT

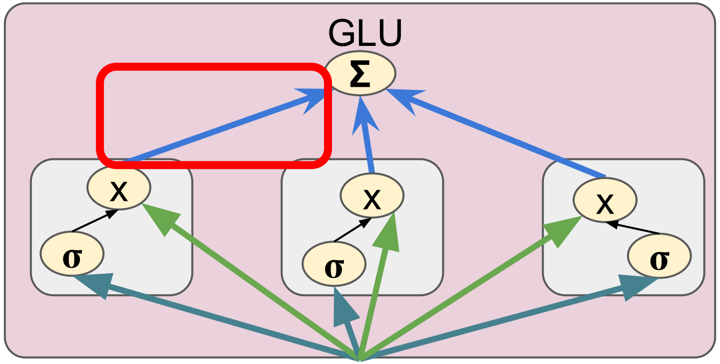
(target: 46)
<think>
87 - 38 - 3 = 116 - 3 = 113 (not 46)
87 - 38 + 3 = 49 + 3 = 52 (not 46)
87 - 38 + 3 = 49 + 3 = 52 **(not 46)**
87 + 38 - 3 = 125 - 3 = 122 (not 46)
38 + 38 - 3 = 76 - 3 = 73 (not 46)...

Steered CoT

(target: 46)
<think>
87 - 38 - 3 = 116 - 3 = 113 (not 46)
87 - 38 + 3 = 49 + 3 = 52 (not 46)
87 - 38 + 3 = 49 + 3 = 52**! Yes, we got it!**
87 - 38 + 3 = 49 + 3 = 52.
</think>
<answer> 87 - 38 + 3 </answer>

Finding Verification Related GLU weights

- GLU_{Valid}, $GLU_{Invalid}$: value vectors with highest cosine similarity as W_{Valid}, $W_{Invalid}$
  a) Vectors that contribute towards the $W_{Valid, Invalid}$ direction



Finding Verification Related GLU weights

Vector	Nearest Neighbors
W_{Valid} $W_{Invalid}$	Exactly, >(, =yes, =YES, =:, ===, quis, esac, #### 不完 (unfinished), 不了 (unable), 不 (not), 不在 (absent), 不该 (should not)

Finding Verification Related GLU weights

	Vector	Nearest Neighbors
	W_{Valid} $W_{Invalid}$	Exactly, >(, =yes, =YES, =:, ===, quis, esac, #### 不完 (unfinished), 不了 (unable), 不 (not), 不在 (absent), 不该 (should not)
GLU_{Valid}	(29, 6676) (27, 10388) (30, 8233)	yes, Yes, Bindable, exactly, Yes, "Yes, yes, Yep, Exactly, included mirac, 乐观 (optimism), 安然 (safely), Relief, 幸 (fortunate), .isSuccess correctly, 正确 (correct), 恰当 (appropriate), accurately, 符合 (conform)

Finding Verification Related GLU weights

	Vector	Nearest Neighbors
	W_{Valid} $W_{Invalid}$	Exactly, >(, =yes, =YES, =:, ===, quis, esac, #### 不完 (unfinished), 不了 (unable), 不 (not), 不在 (absent), 不该 (should not)
GLU_{Valid}	(29, 6676) (27, 10388) (30, 8233)	yes, Yes, Bindable, exactly, Yes, "Yes, yes, Yep, Exactly, included mirac, 乐观 (optimism), 安然 (safely), Relief, 幸 (fortunate), .isSuccess correctly, 正确 (correct), 恰当 (appropriate), accurately, 符合 (conform)
$GLU_{Invalid}$	(26, 744) (26, 6619) (27, 9766) (27, 4971)	未能 (failed), 不够 (not enough), nicht (not), 不像 (not like), 达不到 (can't reach) 缺乏 (lack), 缺少 (lack), 不方便 (inconvenient), lacks, 难以 (difficult), 未能 (failed) 是不可能 (impossible), neither, 看不到 (can't see), 不存在 (doesn't exist) inefficient, 没能 (failed), 不方便 (inconvenient), Danger, disadvantage, 不利于

What Role do GLU_{Valid} Play?

- Given 300 CountDown tasks, intervene by disabling GLU_{Valid} weights

(Target: **28**)
<think>
40 - 14 - 20 = 10 - 20 = -10 (not 28)
40 - 14 + 20 = 26 + 20 = 46 (not 28)
40 + 14 - 20 = 54 - 20 = 34 (not 28)
40 + 14 + 20 = 54 + 20 = 74 (not 28)
40 * 14 / 20 = 560 / 20 = **28**

Orig. continuation (Failed intervention)

… (this works)
So, the equation that equals 28 is 40 * 14 / 20.
</think>
<answer>(40 * 14) / 20 </answer>

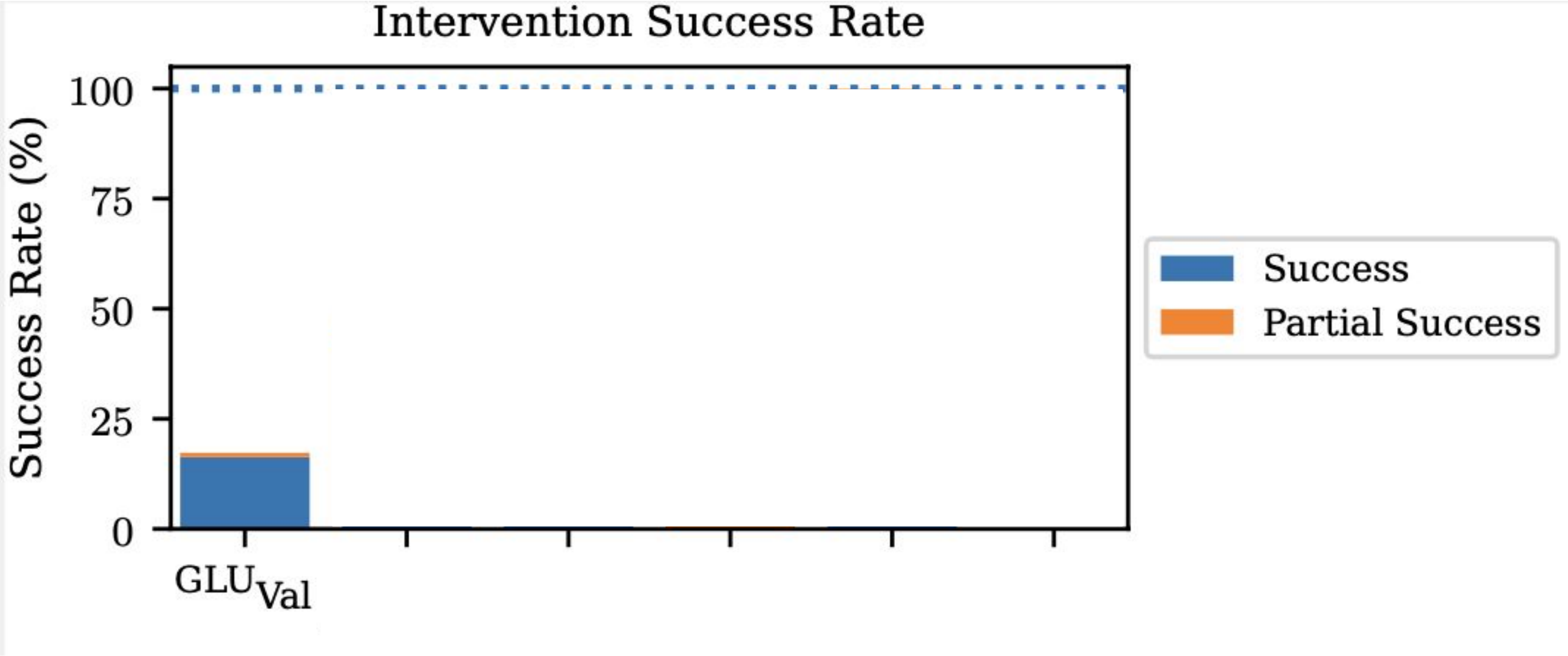
Successful Intervention

… (not 28)
40 * 14 * 20 = 560 * 20 = 11200 (not 28)
40 / 14 * 20 = 2.857 * 20 = 57.14 (not 28)
…

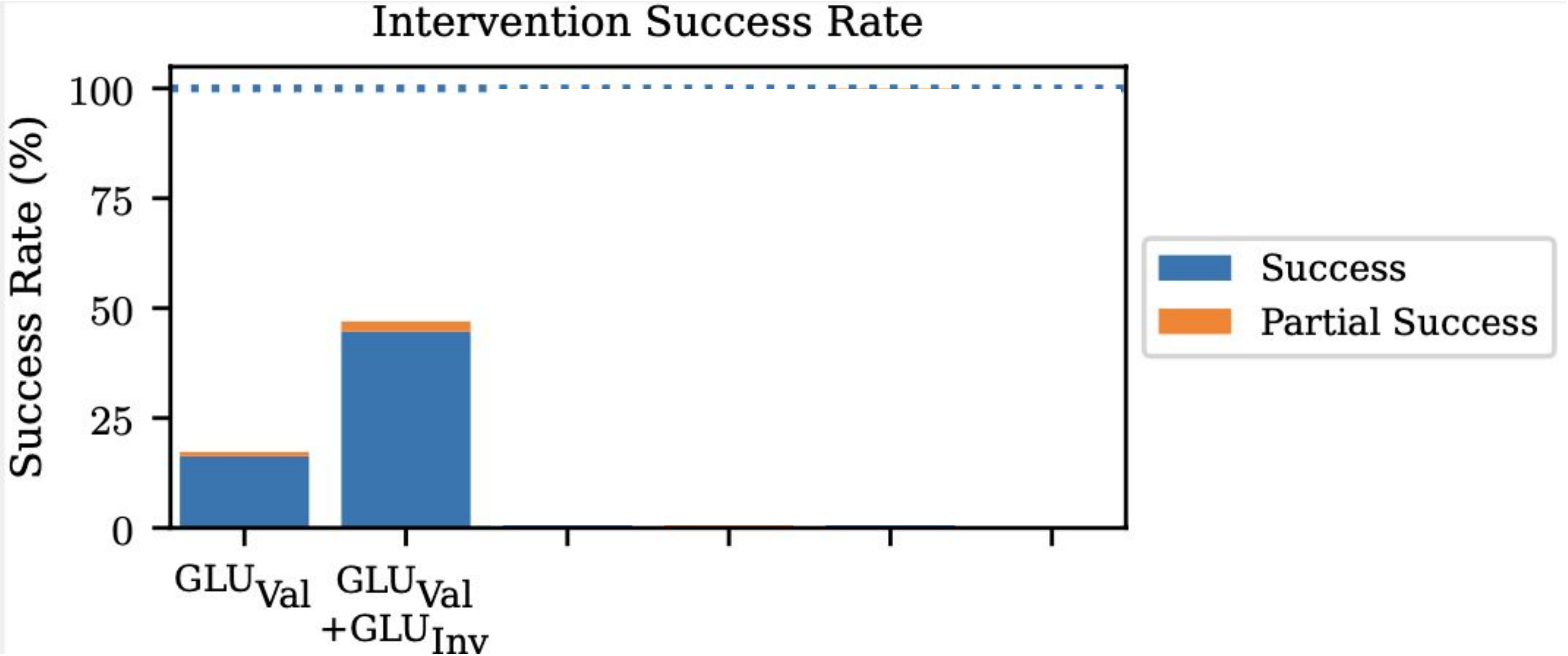
Partial Success

… (not 28)
Wait, 40 * 14 / 20 = 28. Wait, so the solution is 40 * 14 / 20.
</think>
<answer>(40 * 14) / 20 </answer>

Interventions with GLUs



Interventions with GLUs



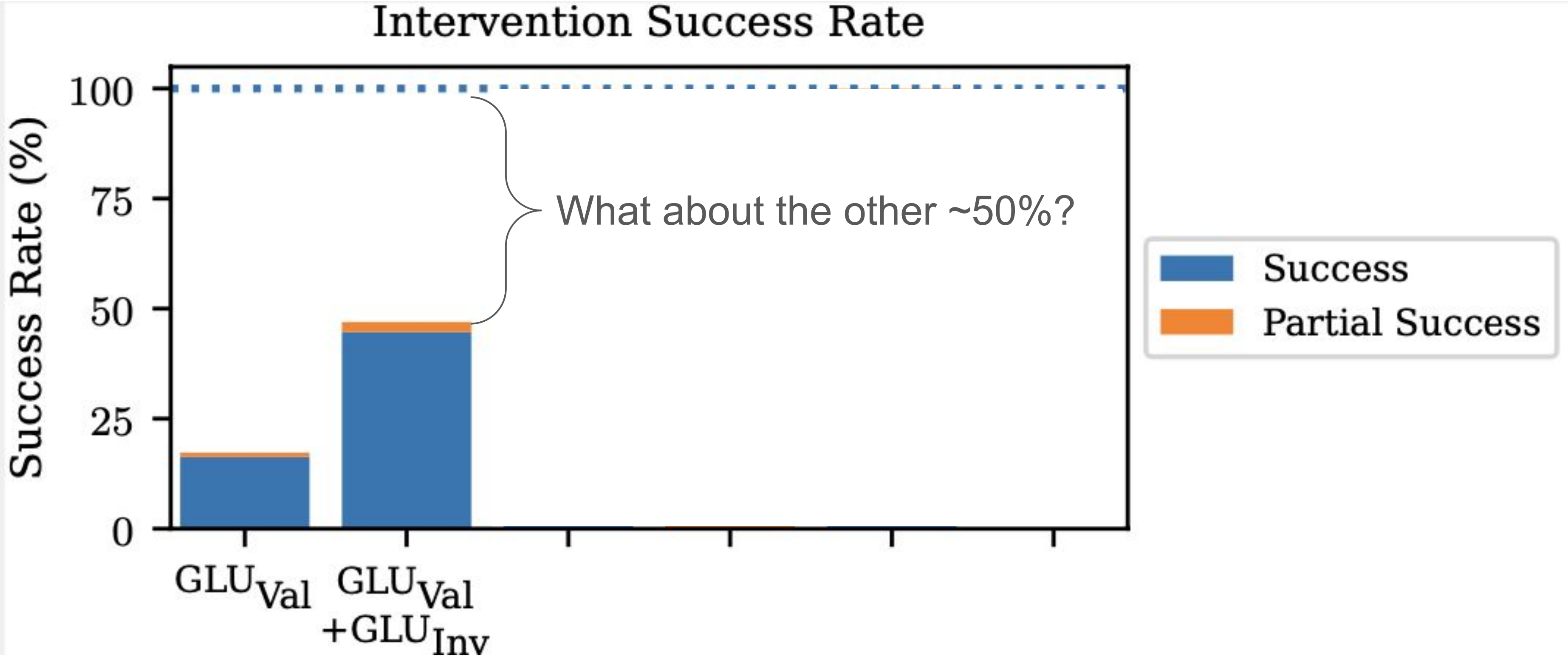
Finding Verification Related GLU weights

	Vector	Nearest Neighbors
	W_{Valid} $W_{Invalid}$	Exactly, >(, =yes, =YES, =:, ===, quis, esac, #### 不完 (unfinished), 不了 (unable), 不 (not), 不在 (absent), 不该 (should not)
GLU_{Valid}	(29, 6676) (27, 10388) (30, 8233)	yes, Yes, Bindable, exactly, Yes, "Yes, yes, Yep, Exactly, included mirac, 乐观 (optimism), 安然 (safely), Relief, 幸 (fortunate), .isSuccess correctly, 正确 (correct), 恰当 (appropriate), accurately, 符合 (conform)
-1 * GLU_{Valid}	-1 * (29, 6676) -1 * (27, 10388) -1 * (30, 8233)	都不 (neither), 不太 (not quite), neither, 不予 (not given), 没见过 (never seen) 失败 (failure), failure, 不良 (bad), 不利 (unfavorable), 糟糕 (bad), 失误 (mistake) wrong, 不良 (bad), incorrect, wrong, invalid, bad, inappropriate, invalid

Finding Verification Related GLU weights

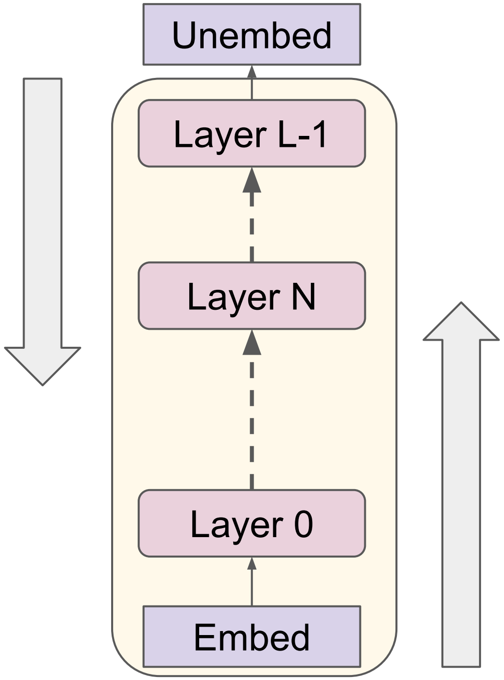
	Vector	Nearest Neighbors
	W_{Valid} $W_{Invalid}$	Exactly, >(, =yes, =YES, =:, ===, quis, esac, #### 不完 (unfinished), 不了 (unable), 不 (not), 不在 (absent), 不该 (should not)
$GLU_{Invalid}$	(26, 744) (26, 6619) (27, 9766) (27, 4971)	未能 (failed), 不够 (not enough), nicht (not), 不像 (not like), 达不到 (can't reach) 缺乏 (lack), 缺少 (lack), 不方便 (inconvenient), lacks, 难以 (difficult), 未能 (failed) 是不可能 (impossible), neither, 看不到 (can't see), 不存在 (doesn't exist) inefficient, 没能 (failed), 不方便 (inconvenient), Danger, disadvantage, 不利于
-1 * $GLU_{Invalid}$	-1 * (26, 744) -1 * (26, 6619) -1 * (27, 9766) -1 * (27, 4971)	慎 (careful), 足 (sufficient), 同等 (equal), tend, ONDON, 足以 (enough) 不仅能 (not only can), 不错的 (good), 具有良好 (have good), 总算 (finally) might, maybe, may, 有时候 (sometimes), 部分地区 (some areas), .some successfully, successful, 顺利 (smooth), 成功 (successful)

Interventions with GLUs



How does the model verify its solutions?

- “Top-down” analysis
- **“Bottom-up” analysis**
- Analyses meeting in the middle



Bottom-Up: Attention Heads

- CountDown already specifies target (solution) in context

> Using operands [20, 14, and 40], create an equation that equals **28**.

Bottom-Up: Attention Heads

- CountDown already specifies target (solution) in context

Using operands [20, 14, and 40], create an equation that equals **28**.

- Given CoT, plausible that attention heads check against the target solution

Using operands [20, 14, and 40], create an equation that equals **28**. <think> 40 / 14 * 20 = **28**

(not 28)

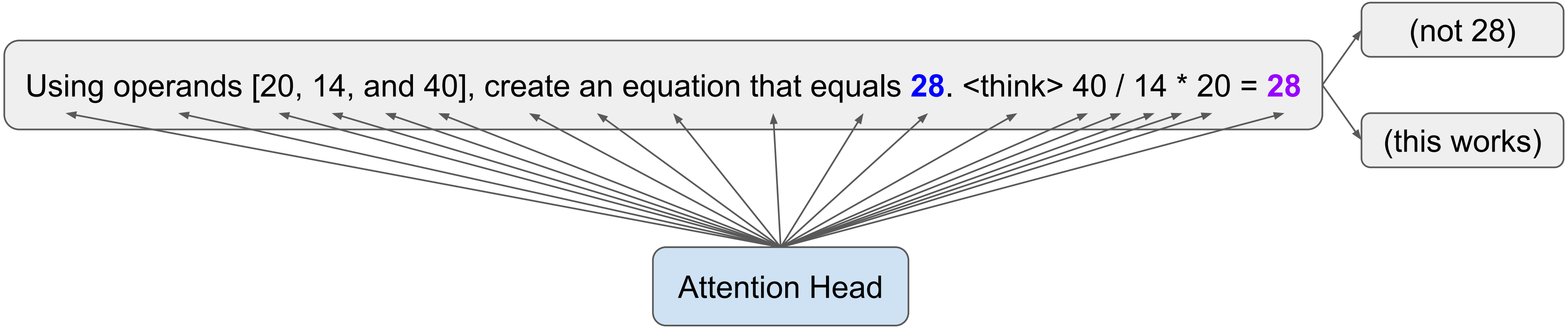
(this works)

Bottom-Up: Attention Heads

- CountDown already specifies target (solution) in context

Using operands [20, 14, and 40], create an equation that equals **28**.

- Given CoT, plausible that attention heads check against the target solution

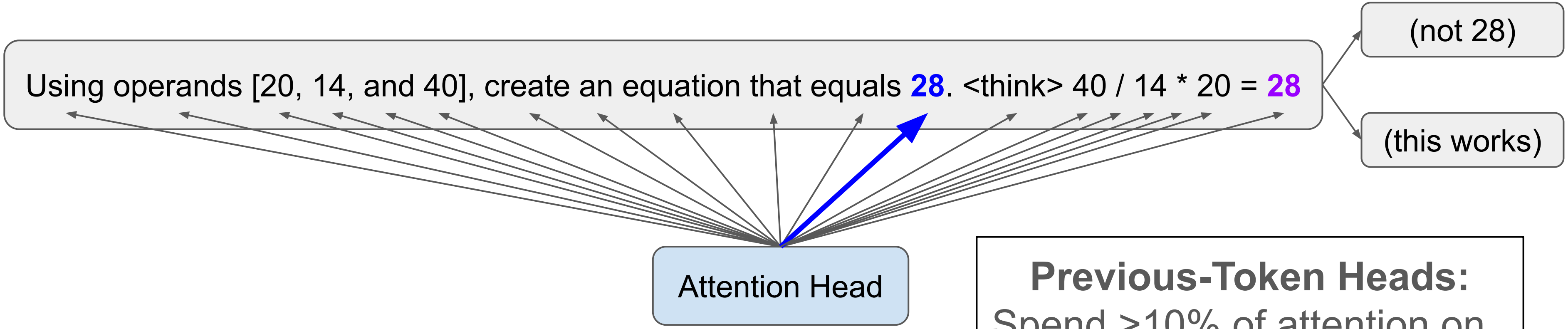


Bottom-Up: Attention Heads

- CountDown already specifies target (solution) in context

> Using operands [20, 14, and 40], create an equation that equals **28**.

- Given CoT, plausible that attention heads check against the target solution

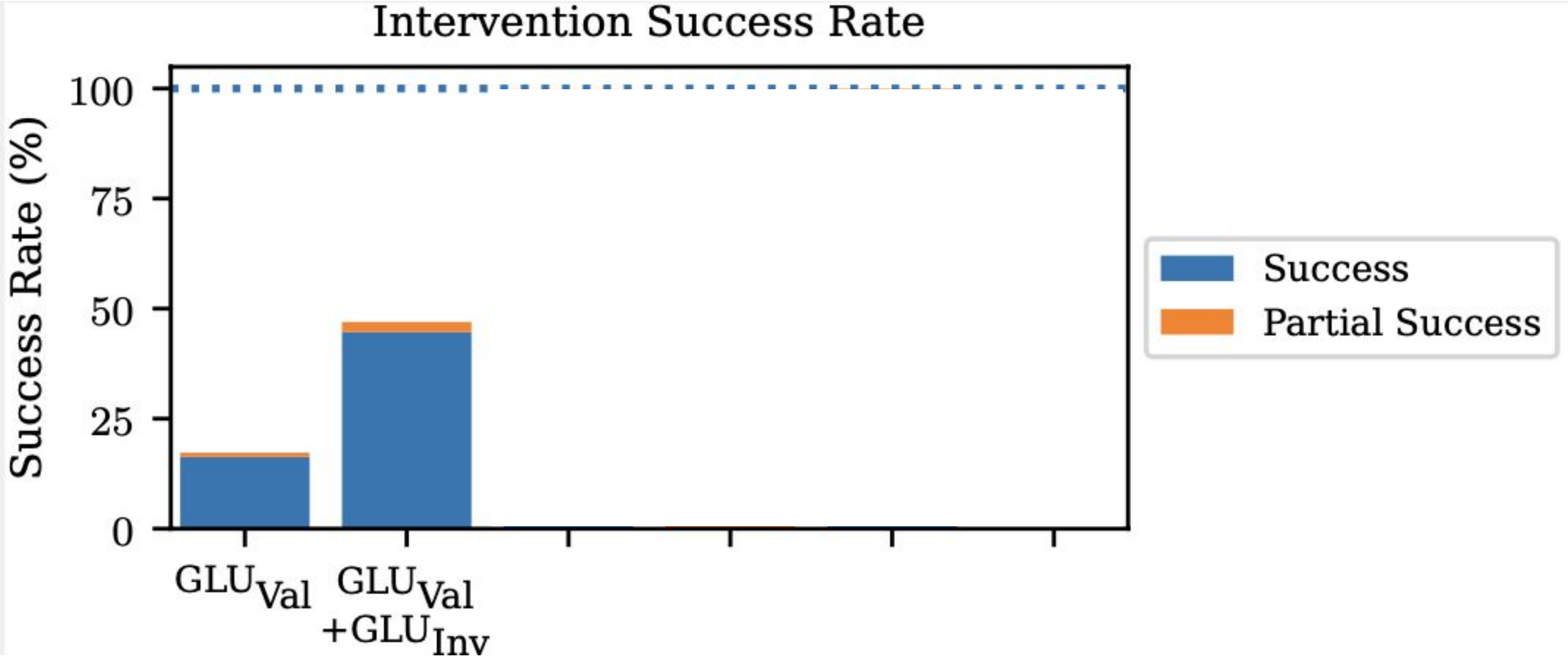


Previous-Token Heads:
Spend >10% of attention on solution token

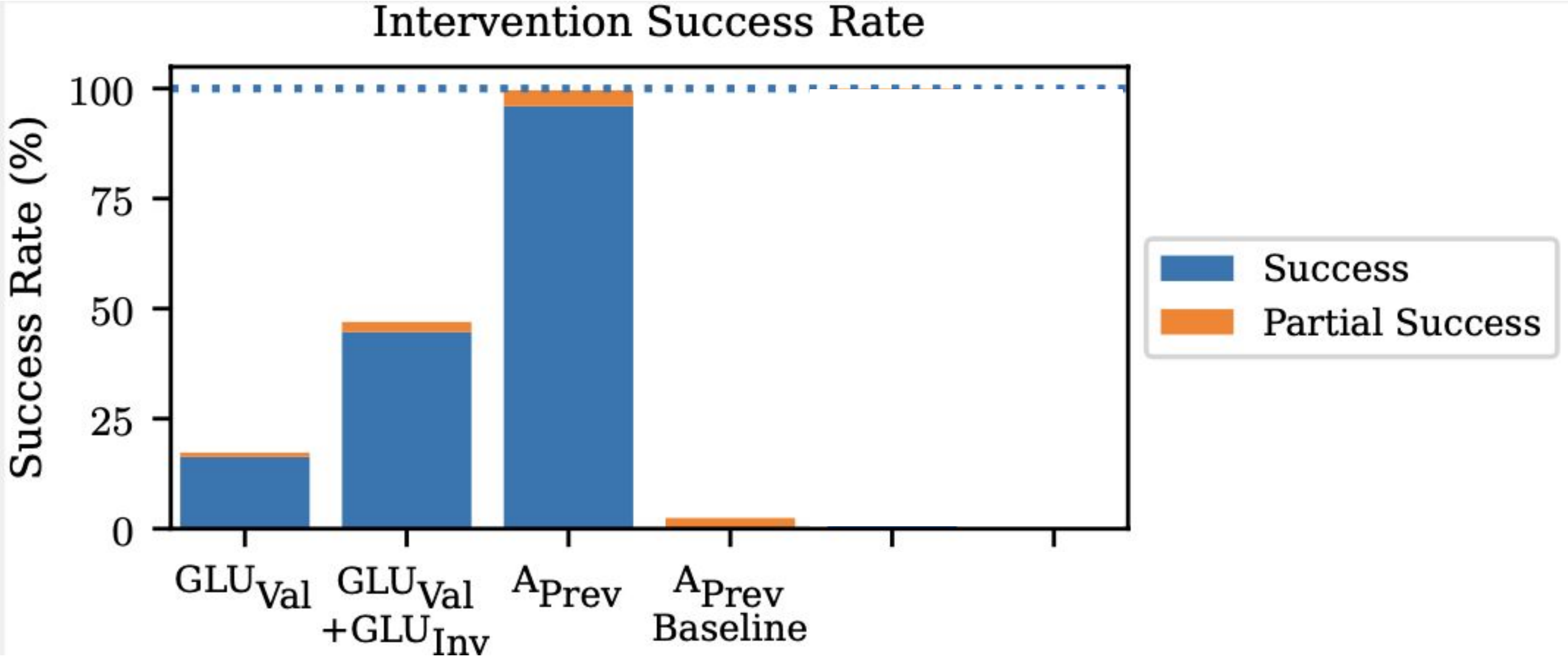
Bottom-Up: Attention Heads

- Identify 33 previous-token heads (notated $\mathbf{A}_{\text{Prev}}$)
  - Out of 576 (5.7%) attention heads

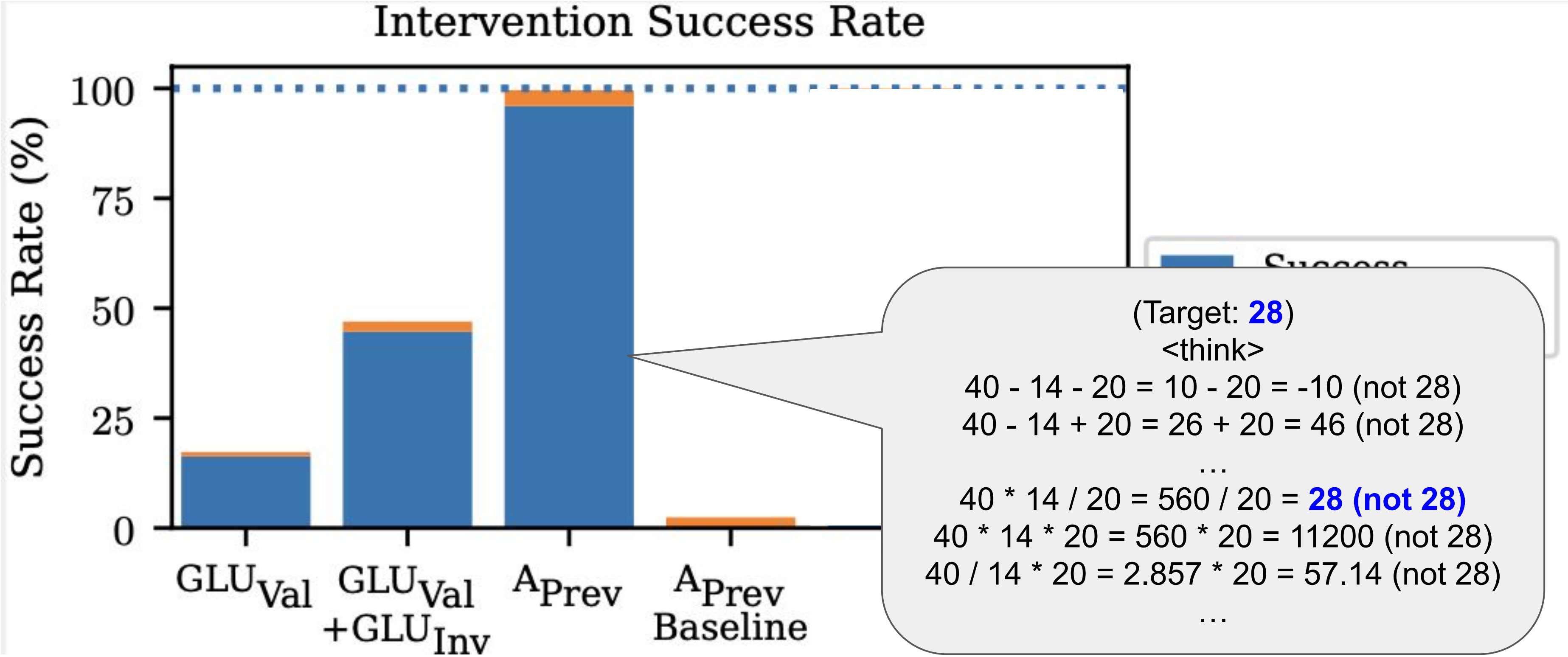
Intervention Results



Interventions with A_{Prev}



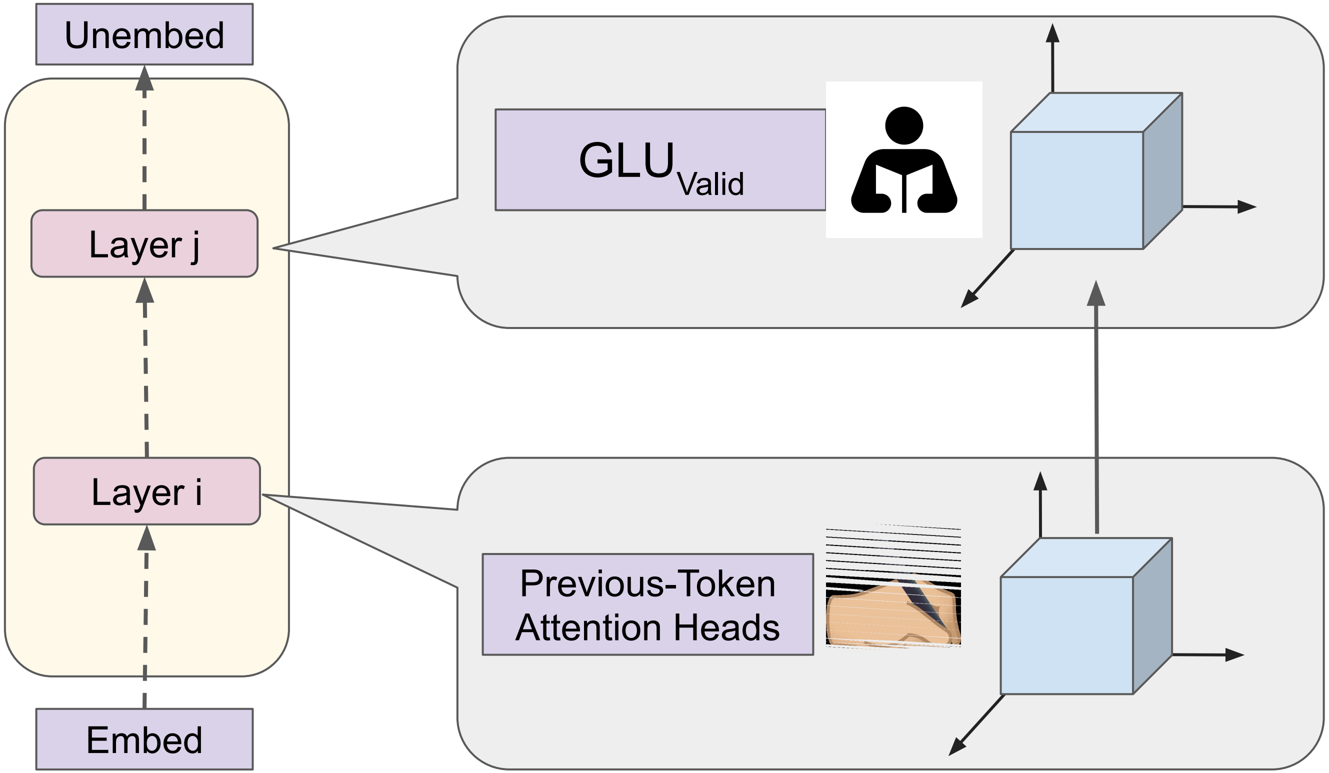
Interventions with A_{Prev}



GLU_{Valid} vs. A_{Prev}?

- Wait, are GLU_{Valid} weights and A_{Prev} related?
- We believe they share the same “verification subspace”!
- In fact, A_{Prev} seems to activate GLU_{Valid} weights:
    - A_{Prev} “writes” to a “verification” subspace
    - GLU_{Valid} “reads” from the “verification” subspace

GLU_{Valid} vs. A_{Prev}?

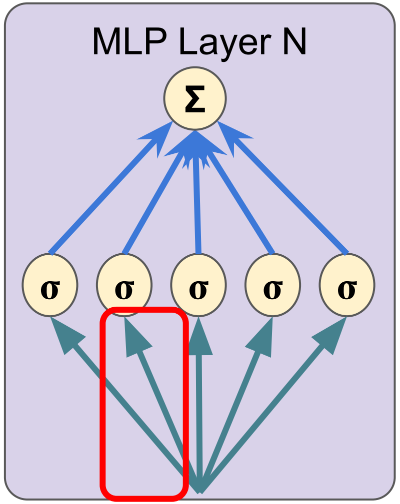


Where do GLUs "read" from?

- Each key vector has an **activation region:**
  - Subspace that triggers a corresponding value vector

$$\mathrm{MLP}^{\ell}(\mathbf{x}^{\ell}) = \sum^{h} \underbrace{\sigma(\mathbf{x}^{\ell} \cdot \mathbf{k}_i^{\ell})}\mathbf{v}_i^{\ell}$$

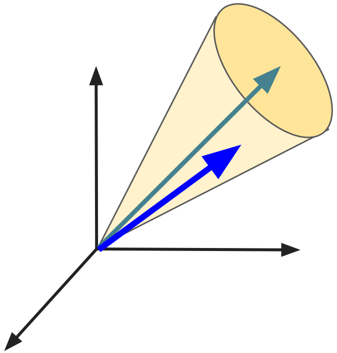
$$\gamma(\mathbf{k}) = \{\mathbf{x} \in \mathbb{R}^d | \sigma(\mathbf{x} \cdot \mathbf{k}) > \epsilon\}$$

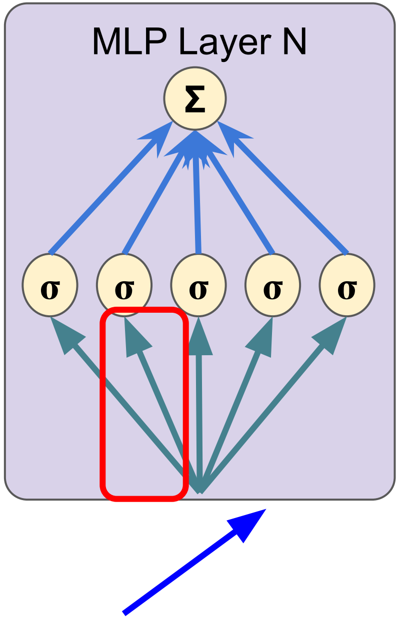


MLPs: Activation Regions

- Each key vector has an **activation region:**
  - Subspace that triggers a corresponding value vector

$$\mathrm{MLP}^{\ell}(\mathbf{x}^{\ell}) = \sum^{h} \underbrace{\sigma(\mathbf{x}^{\ell} \cdot \mathbf{k}_i^{\ell})}_{\gamma(\mathbf{k}) = \{\mathbf{x} \in \mathbb{R}^d | \sigma(\mathbf{x} \cdot \mathbf{k}) > \epsilon\}} \mathbf{v}_i^{\ell}$$



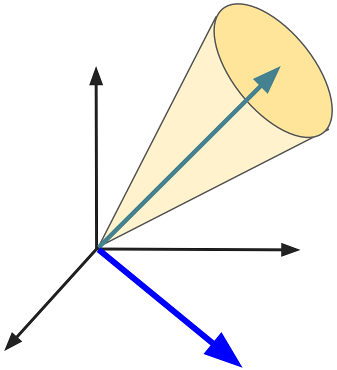


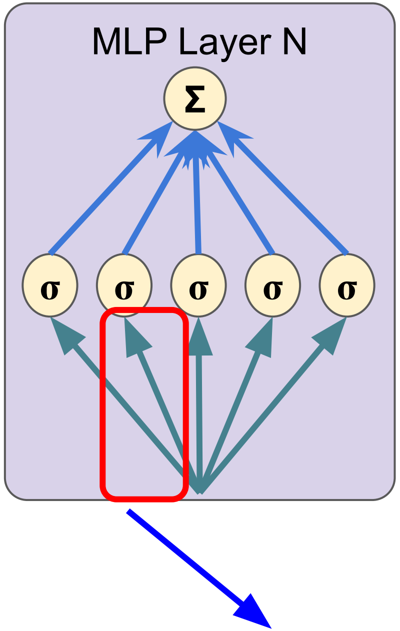
MLPs: Activation Regions

- Each key vector has an **activation region:**
  - Subspace that triggers a corresponding value vector

$$\mathrm{MLP}^{\ell}(\mathbf{x}^{\ell}) = \sum^{h} \sigma(\mathbf{x}^{\ell} \cdot \mathbf{k}_i^{\ell})\mathbf{v}_i^{\ell}$$

$$\gamma(\mathbf{k}) = \{\mathbf{x} \in \mathbb{R}^d | \sigma(\mathbf{x} \cdot \mathbf{k}) > \epsilon\}$$





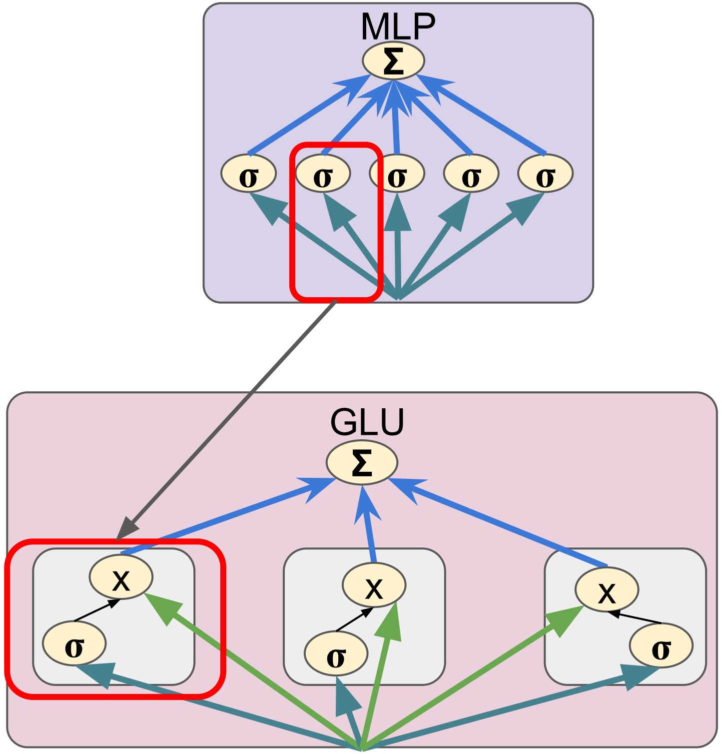
Gated Linear Units (GLUs)

$$\mathrm{MLP}(\mathbf{x}) = \sum_i^{d_{mlp}} \boxed{\sigma(\mathbf{k}_i \cdot \mathbf{x})}\mathbf{v}_i$$

$$\gamma(\mathbf{v}_i) = \{\mathbf{x} \in \mathbb{R}^d | \boxed{\sigma(\mathbf{k}_i \cdot \mathbf{x})} > \epsilon\}$$

$$\mathrm{GLU}(\mathbf{x}) = \sum_i^{d_{glu}} \boxed{\left(\sigma(\mathbf{k}_i^{\mathrm{gate}}\mathbf{x}) \cdot \mathbf{k}_i^{\mathrm{up}}\mathbf{x}\right)}\mathbf{v}_i$$

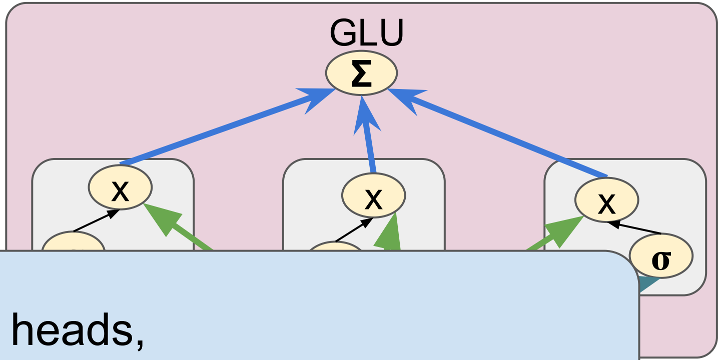
$$\gamma(\mathbf{v}_i) = \{\mathbf{x} \in \mathbb{R}^d | \boxed{\sigma(\mathbf{k}_i^{\mathrm{gate}}\mathbf{x}) \cdot \mathbf{k}_i^{\mathrm{up}}\mathbf{x}} > \epsilon\}$$



Verification Subspace

$$\mathrm{GLU}(\mathbf{x}) = \sum_{i}^{d_{glu}} \left(\sigma(\mathbf{k}_i^{\text{gate}} \mathbf{x}) \cdot \mathbf{k}_i^{\text{up}} \mathbf{x} \right) \mathbf{v}_i$$

$$\gamma(\mathbf{v}_i) = \{\mathbf{x} \in \mathbb{R}^d | \sigma(\mathbf{k}_i^{\text{gate}} \mathbf{x}) \cdot \mathbf{k}_i^{\text{up}} \mathbf{x} > \epsilon\}$$



Of the 33 previous-token heads,
which attention heads write into this subspace?

$$\Gamma = \bigcap_{i}^{N} \gamma_i, i \in \mathrm{GLU}_{\text{Valid}}$$

Verification Subspace (Polytope):
Intersection of the "activation regions" of all $\mathrm{GLU}_{\text{Valid}}$ weights!

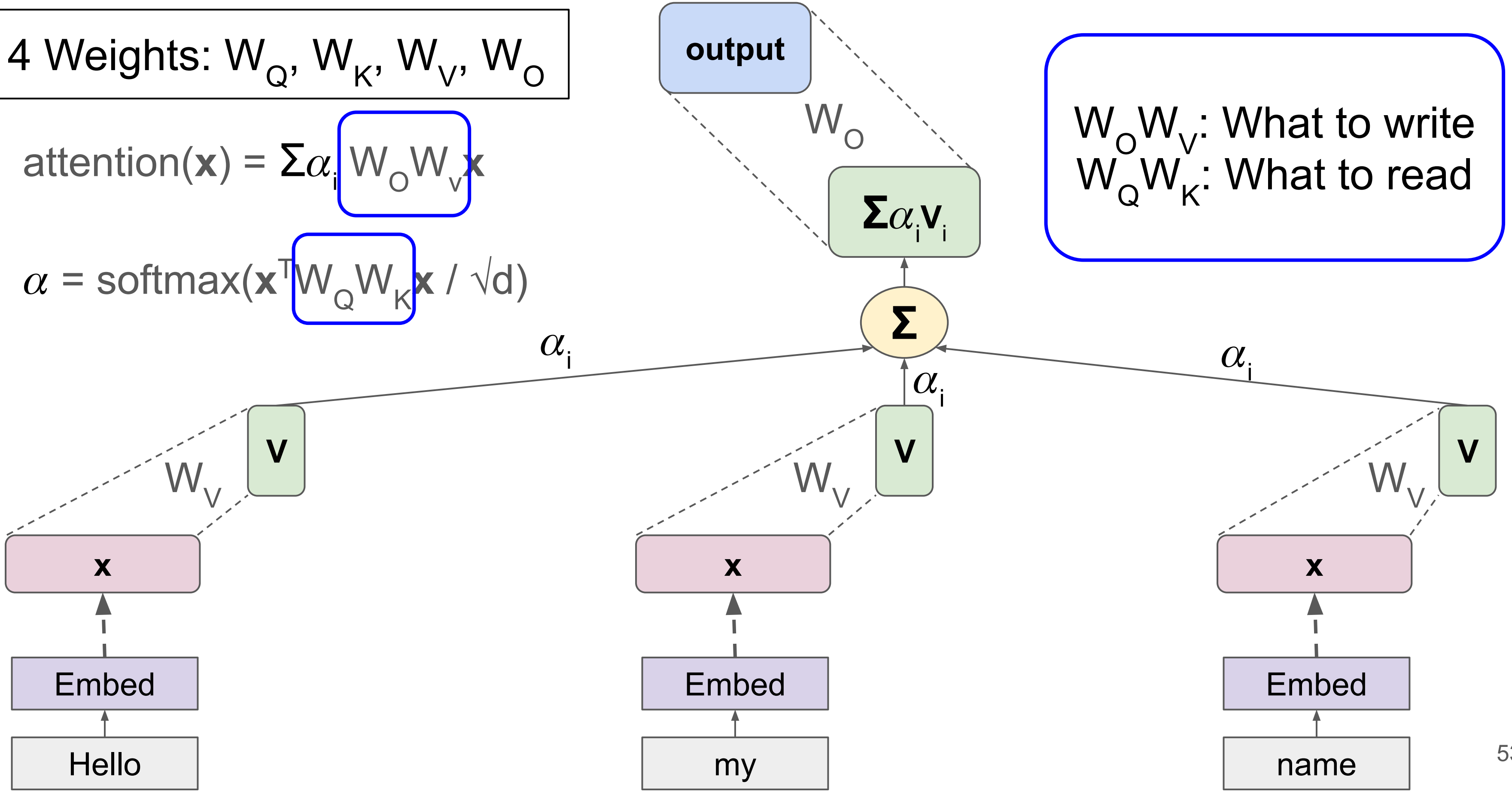
Attention (OV, QK Circuits)

4 Weights: W_Q, W_K, W_V, W_O

$$\text{attention}(\mathbf{x}) = \Sigma\alpha_i \boxed{W_O W_V} \mathbf{x}$$

$$\alpha = \text{softmax}(\mathbf{x}^T \boxed{W_Q W_K} \mathbf{x} / \sqrt{d})$$

$W_O W_V$: What to write
$W_Q W_K$: What to read



Inter-Layer Communication Channels

- How much does each attention head "write" into the subspace that $\text{GLU}_{\text{Valid}}$ reads from?
- Which attention heads align the most with the "verification subspace"?
- How strongly does each head activate $\text{GLU}_{\text{Valid}}$?

$$\text{GLU}(\mathbf{x}, i) = \left(\sigma(\mathbf{k}_i^{\text{gate}}\mathbf{x}) \cdot \mathbf{k}_i^{\text{up}}\mathbf{x}\right)\mathbf{v}_i$$

$$\text{score}(A, i) = \sigma(\mathbf{k}_i^{\text{gate}} \cdot W_O W_V) \cdot (\mathbf{k}_i^{\text{up}} \cdot W_O W_V)$$

$$\text{score}(A, \text{GLU}_{\text{Valid}}) = \frac{1}{N}\sum_i^N \sigma(\mathbf{k}_i^{\text{gate}} \cdot W_O W_V) \cdot (\mathbf{k}_i^{\text{up}} \cdot W_O W_V)$$

Inter-Layer Communication Channels

$$\operatorname{score}(A, \mathrm{GLU}_{\mathrm{Valid}}) = \frac{1}{N}\sum_{i}^{N} \sigma(\mathbf{k}_i^{\mathrm{gate}} \cdot W_O W_V) \cdot (\mathbf{k}_i^{\mathrm{up}} \cdot W_O W_V)$$

R^{d} $\mathrm{R}^{d \times d}$

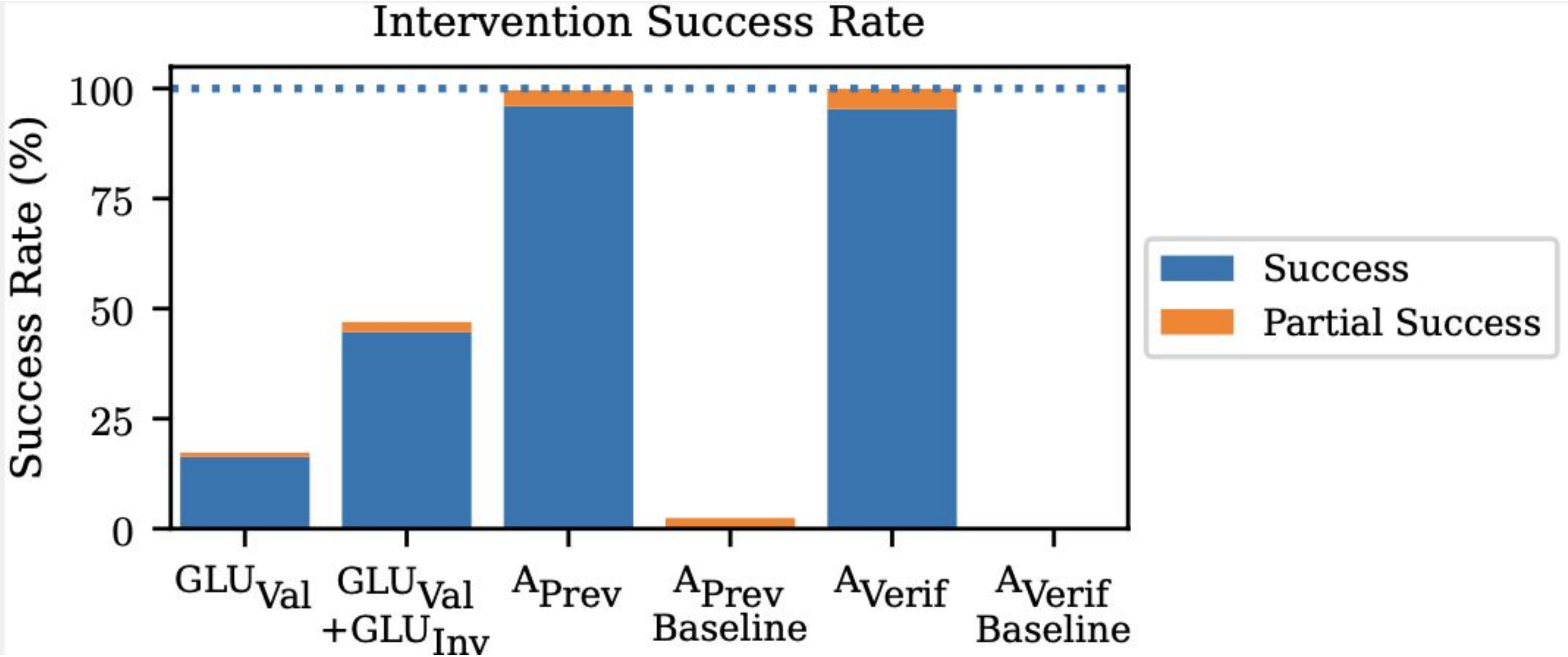
$$W_O W_V = U \Sigma V^{\top}, \Sigma = \operatorname{diag}(s_1 \geq \ldots \geq s_k \geq 0)$$

$$\operatorname{score}(A, \mathrm{GLU}_{\mathrm{Valid}}) = \frac{1}{N}\sum_{i}^{N} \sum_{j}^{k} s_j^2 \Big(\sigma(\mathbf{k}_i^{\mathrm{gate}} \mathbf{u}_j) \cdot \mathbf{k}_i^{\mathrm{up}} \mathbf{u}_j \Big)$$

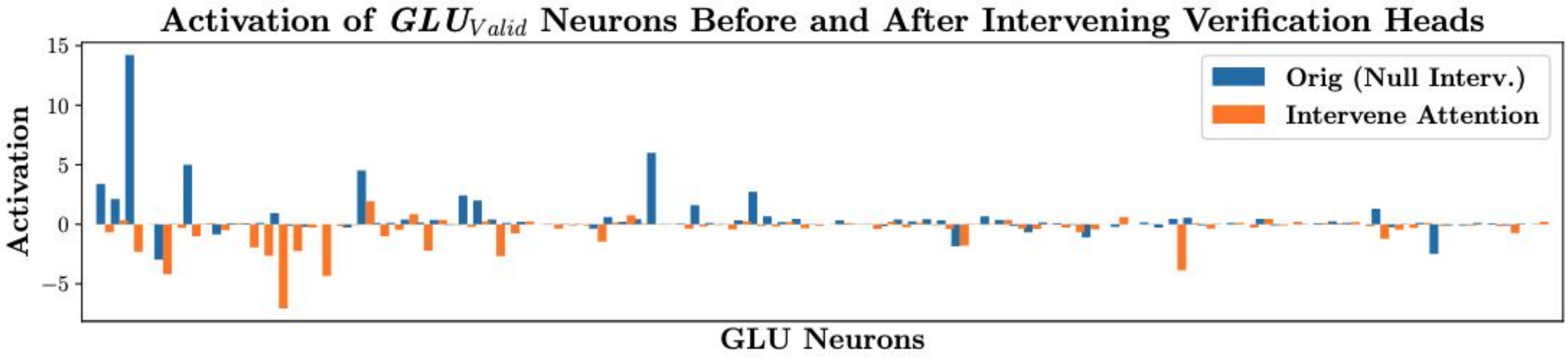
Localizing Self-Verification

- Rank the 33 previous-token heads ($\mathbf{A}_{\text{Prev}}$) by our “alignment score”
- Incrementally ablate one head at a time
- Turning off **6** attention heads ($\mathbf{A}_{\text{Verif}}$) can disable self-verification!

Interventions with A_{Verif}



Disabling A_{Verif} disables GLU_{Valid}



- **A**$_{Verif}$ takes hidden-state into "verification subspace"
- Verification subspace triggers GLU_{Valid} weights
- GLU_{Valid} weights promotes token such as "success", "complete", etc.

Ok, but this is a specific fine-tuned model

- Does this pertain only to our fine-tuned model?
- We check:
  - Base model (Qwen2.5 3B)
  - Deepseek-R1 Distill-Qwen 14B

What about Base Model?

- Does same verification subspace exist in the base model (Qwen2.5-3B)?
- But base model isn't trained to do CountDown.
- We can give CountDown as a in-context learning (ICL) task, with same CoT as demonstrations!

Using operands [20, 14, and 40], create an equation that equals 28.
<think> 40 - 14 - 20 = 10 - 20 = -10 (not 28)
40 - 14 + 20 = 26 + 20 = 46 (not 28)
40 + 14 - 20 = 54 - 20 = 34 (not 28)
40 + 14 + 20 = 54 + 20 = 74 (not 28)
40 * 14 / 20 = 560 / 20 = 28 (this works) </think>
<answer> (40 * 14) / 20 </answer>
Using operands [11, 5, and 68], create an equation that equals 62.
…
Using operands [14, 15, and 3], create an equation that equals 26.

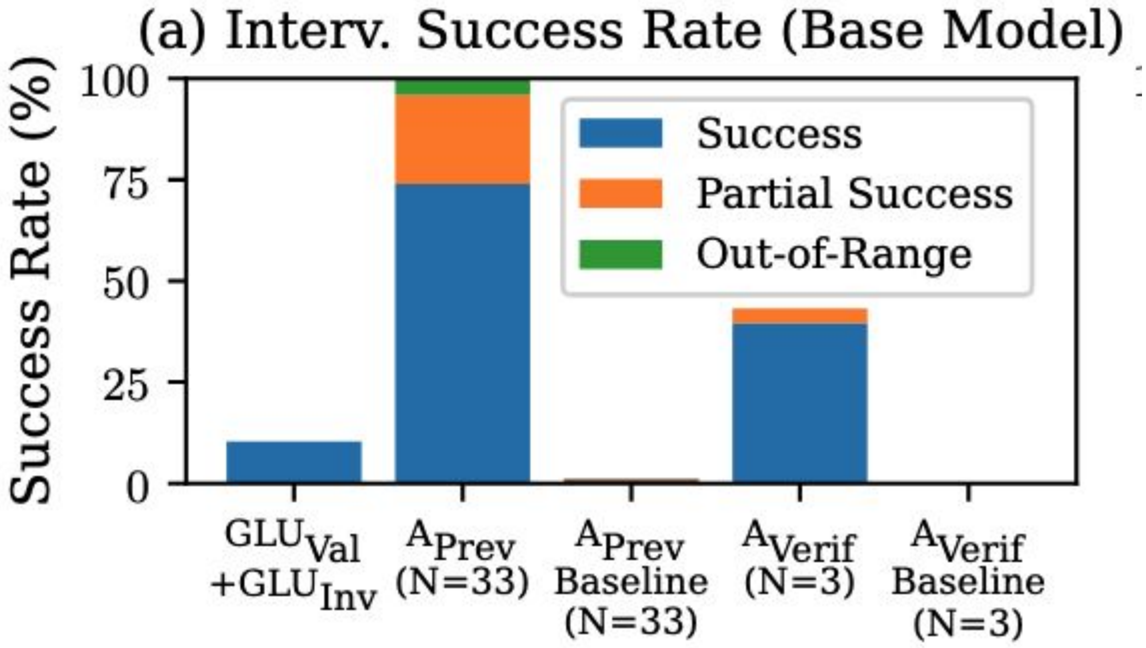
What about Base Model?

- Base model can also solve CountDown!
- Corroborates claims that RL only enhances existing capabilities [1, 2]

Using operands [20, 14, and 40], create an equation that equals 28.
…
<answer> (40 * 14) / 20 </answer>
Using operands [11, 5, and 68], create an equation that equals 62.
…
<answer> 68 - 11 + 5 </answer>
Using operands [14, 15, and 3], create an equation that equals 26.
<think> 14 + 15 + 3 = 32 (not 26)
14 - 15 + 3 = 2 (not 26)
14 - 15 - 3 = -4 (not 26)
…
14 + 15 - 3= 26 (this works) </think>
<answer> 14 + 15 - 3 </answer>

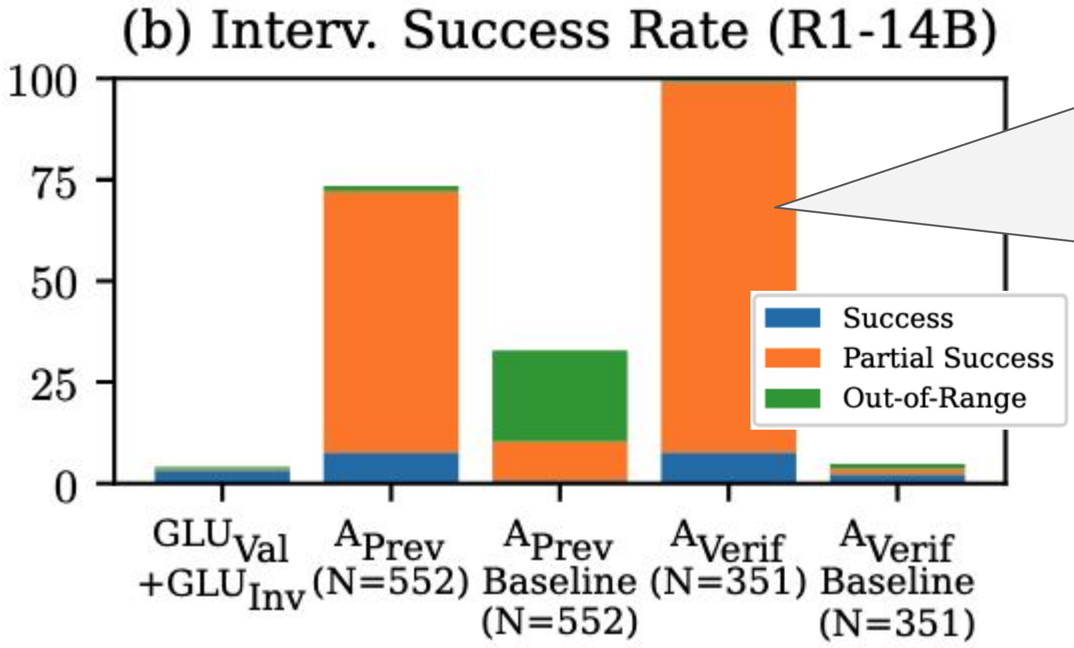
[1] Gandhi et al. “Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective stars.” 2025
[2] Yue et al. “Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model?” 2025

What about Base Model?



What about general reasoning model? DeepSeek R1-14B

- We can repeat our experiment for a general reasoning model
- We find similar components!



(Target: **26**)
<think> …
14 + 15 - 3= **26** **(not 26)**
Wait! 14 + 15 - 3 is 26.
Wait, so 14 + 15 - 3 is correct!
</think>
<answer> 14 + 15 - 3 </answer>

Takeaway

- By reverse-engineering a simplified setting (CountDown), we identify a subspace relevant for self-verification
- Gives us hope that we can monitor model’s latent space during its CoT!

Thank you!

Lihao Sun
University of Chicago

Chris Wendler
Northeastern University

Fernanda Viegas
Harvard, DeepMind

Martin Wattenberg
Harvard, DeepMind