Reasoning by Superposition:
A Theoretical Perspective on Chain of Continuous Thought

Hanlin Zhu [1]

Joint work with Shibo Hao[2], Zhiting Hu[2], Jiantao Jiao [1], Stuart Russell [1], Yuandong Tian [3]

1. UC Berkeley 2. UCSD 3. Meta AI

Contents

1. Background

LLMs on reasoning tasks using CoT

- LLMs are powerful in many reasoning tasks, especially with chain-of-thought (CoT)

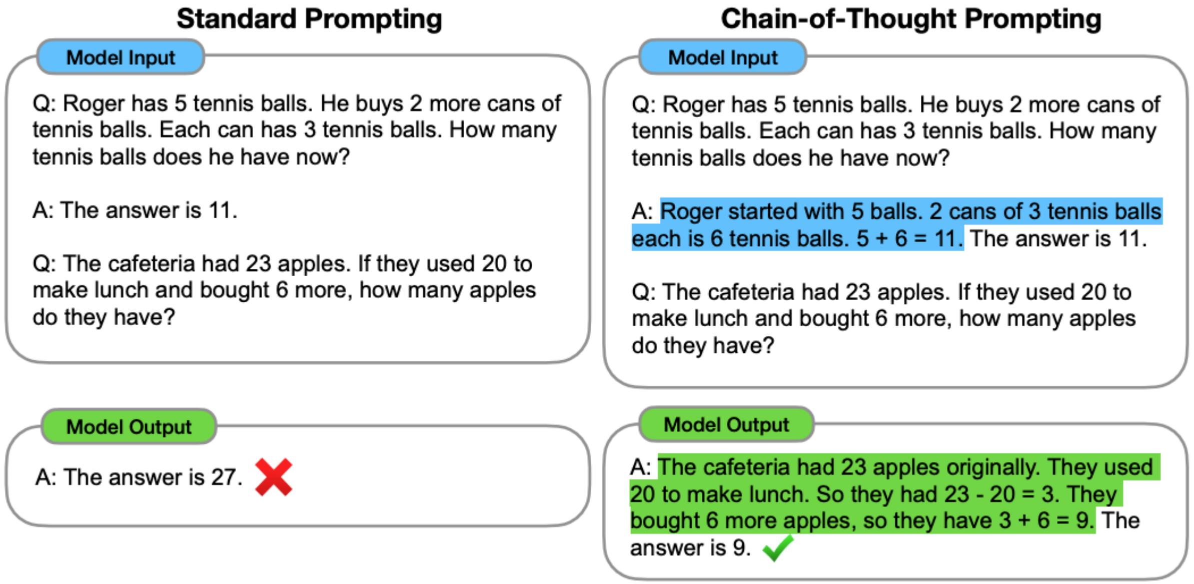


Figure credit to [1]

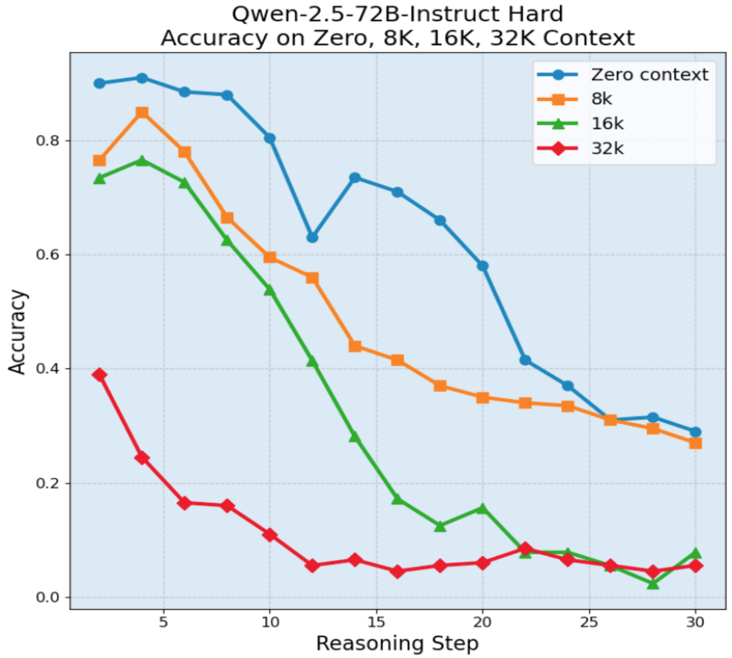


Figure credit to [2]

- LLMs still struggle with more complex reasoning tasks (e.g., longer reasoning steps)
- How to expand existing CoT methods to solve more complex problems?

[1] Wei, Jason, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. "Chain-of-thought prompting elicits reasoning in large language models." *Advances in neural information processing systems* 35 (2022): 24824-24837.
[2] Zhou, Yang, Hongyi Liu, Zhuoming Chen, Yuandong Tian, and Beidi Chen. "GSM-Infinite: How Do Your LLMs Behave over Infinitely Increasing Context Length and Reasoning Complexity?." *arXiv preprint arXiv:2502.05252* (2025).

Existing methods (1)

- Pause tokens[1], filler tokens[2]

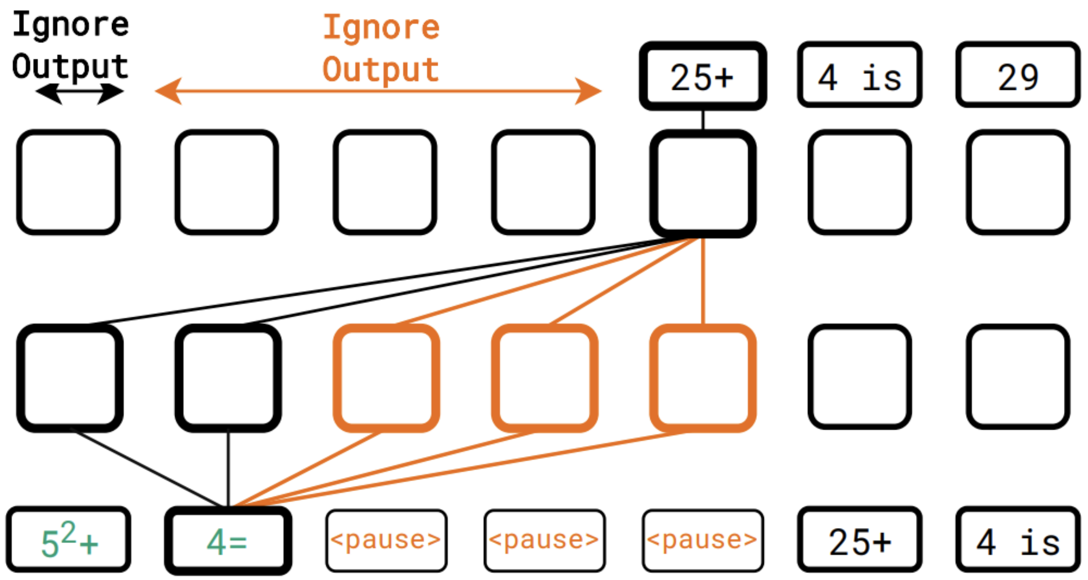


Figure credit to [1]

[1] Goyal, Sachin, Ziwei Ji, Ankit Singh Rawat, Aditya Krishna Menon, Sanjiv Kumar, and Vaishnavh Nagarajan. "Think before you speak: Training language models with pause tokens." *arXiv preprint arXiv:2310.02226* (2023).
[2] Pfau, Jacob, William Merrill, and Samuel R. Bowman. "Let's think dot by dot: Hidden computation in transformer language models." *arXiv preprint arXiv:2404.15758* (2024).

Existing methods (2)

- Implicit CoT[1] (gradually removing intermediate steps)

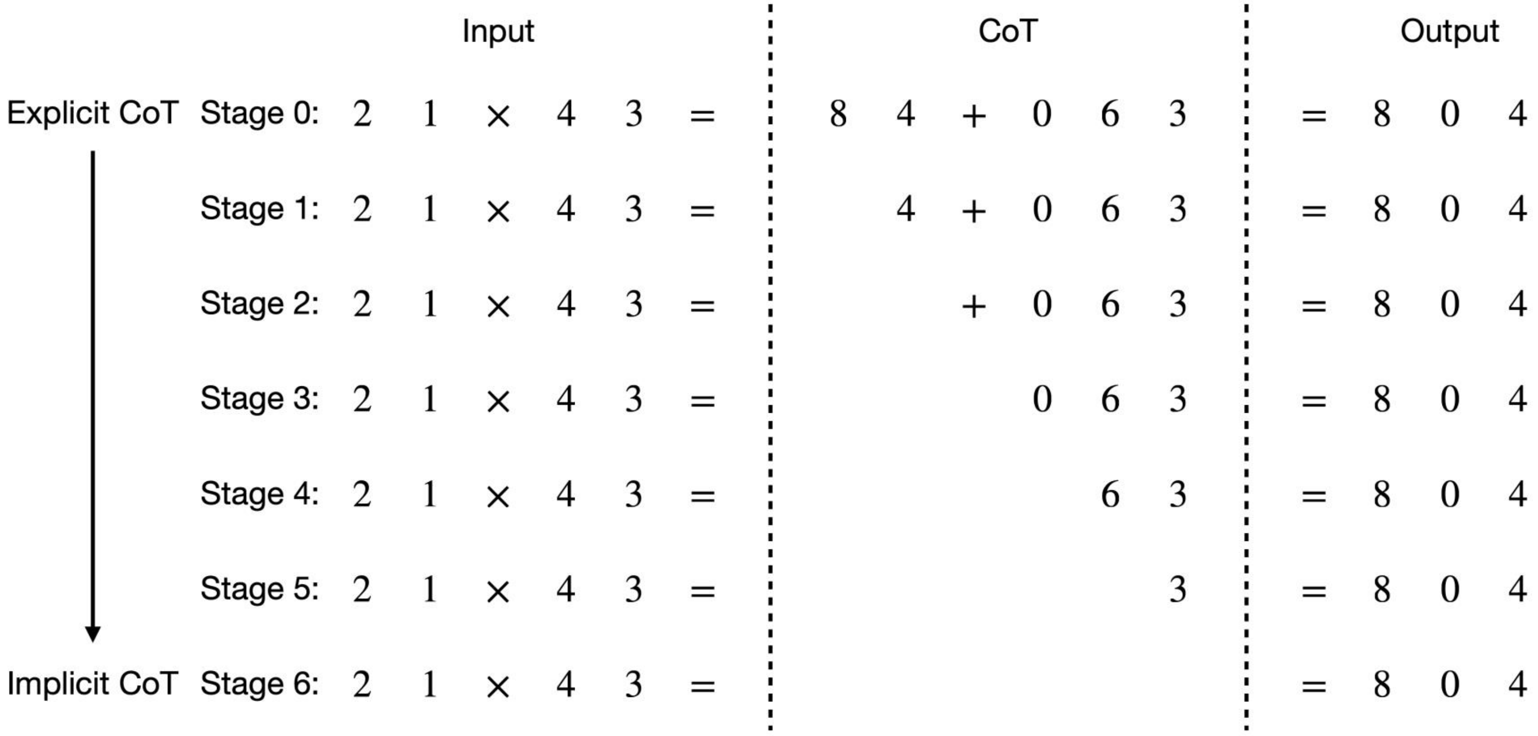


Figure credit to [1]

[1] Deng, Yuntian, Yejin Choi, and Stuart Shieber. "From explicit cot to implicit cot: Learning to internalize cot step by step." *arXiv preprint arXiv:2405.14838* (2024).

Existing methods (3)

- Latent space[1] (use discrete latent tokens as first several steps)

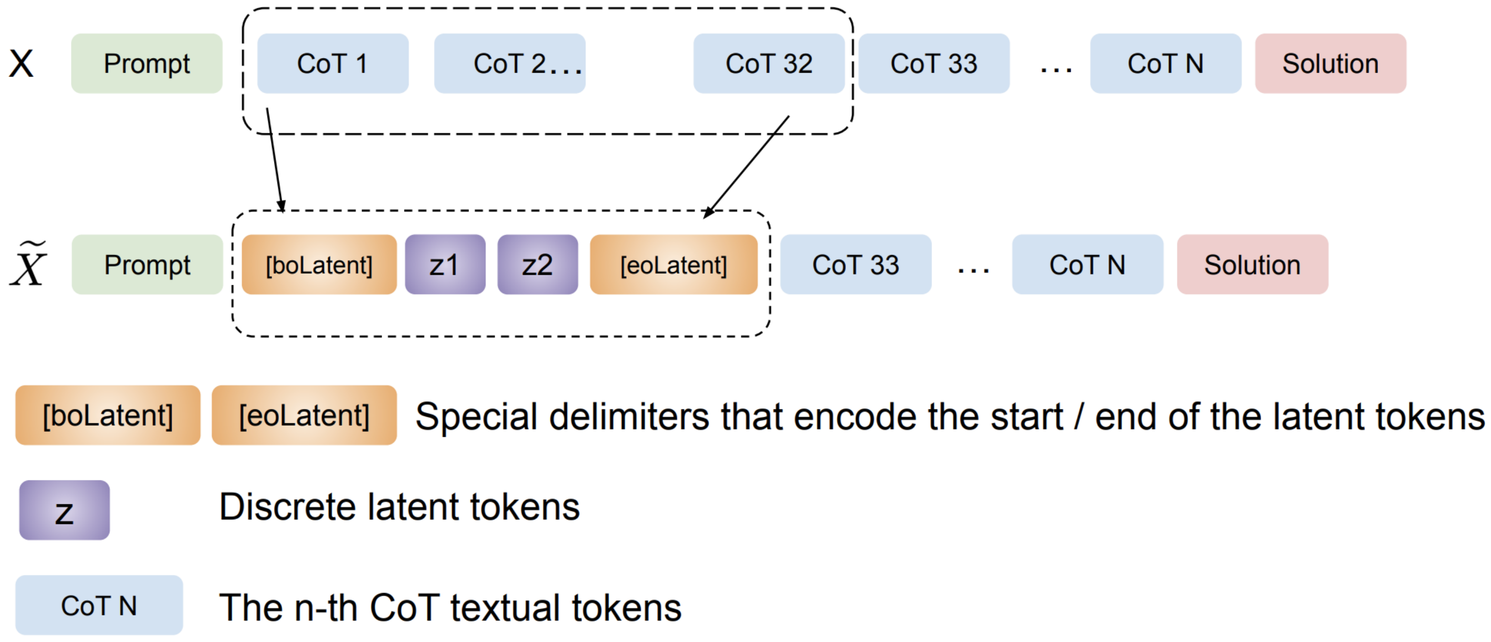


Figure credit to [1]

[1] Su, DiJia, **Hanlin Zhu**, Yingchen Xu, Jiantao Jiao, Yuandong Tian, and Qinqing Zheng. "Token Assorted: Mixing Latent and Text Tokens for Improved Language Model Reasoning." *arXiv preprint arXiv:2502.03275* (2025).

Chain of continuous thought

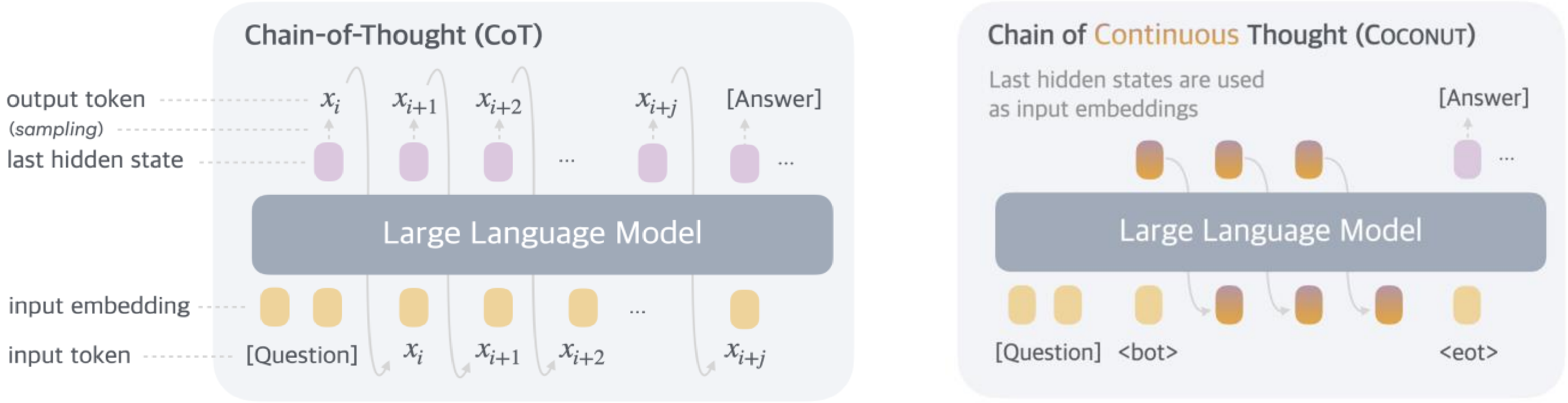


Figure credit to [1]

- Continuous CoT: directly uses the hidden state as the next input
- Outperforms discrete CoTs in various reasoning tasks
  - Especially problems with high branching factors/requires searching
- Lacks theoretical understanding of its power and mechanism

[1] Hao, Shibo, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. "Training large language models to reason in a continuous latent space." *arXiv preprint arXiv:2412.06769* (2024).

Main results

- *Construct* a 2-layer transformer with Continuous CoT that *solves* directed graph reachability using $O(n)$ steps (n: # of vertices)
  - The best known result for constant-depth transformers with discrete CoT requires $O(n^2)$ steps[1]

- **Insights:** Continuous thoughts maintain a "superposition" of explored vertices, performing a parallel BFS

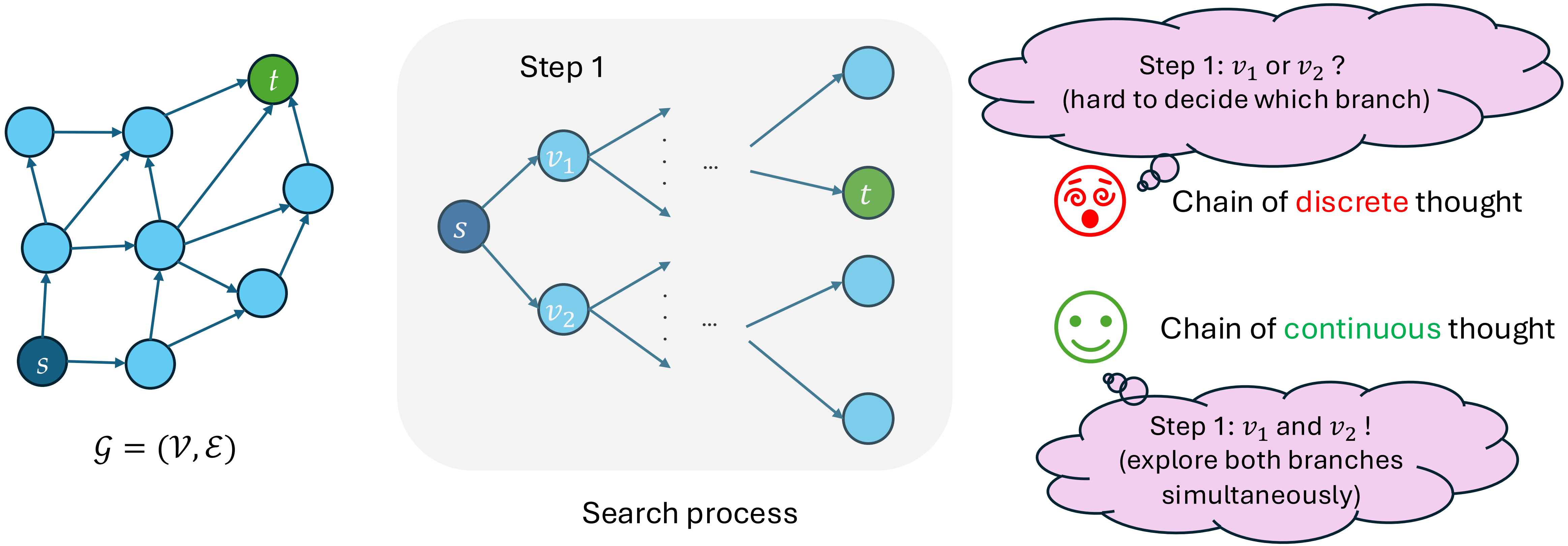
- Empirical study is aligned with theoretical construction
  - Superposition representation **emerges** during training (no supervision)

[1] Merrill, William, and Ashish Sabharwal. "The expressive power of transformers with chain of thought." *arXiv preprint arXiv:2310.07923* (2023).

2. Theoretical Results

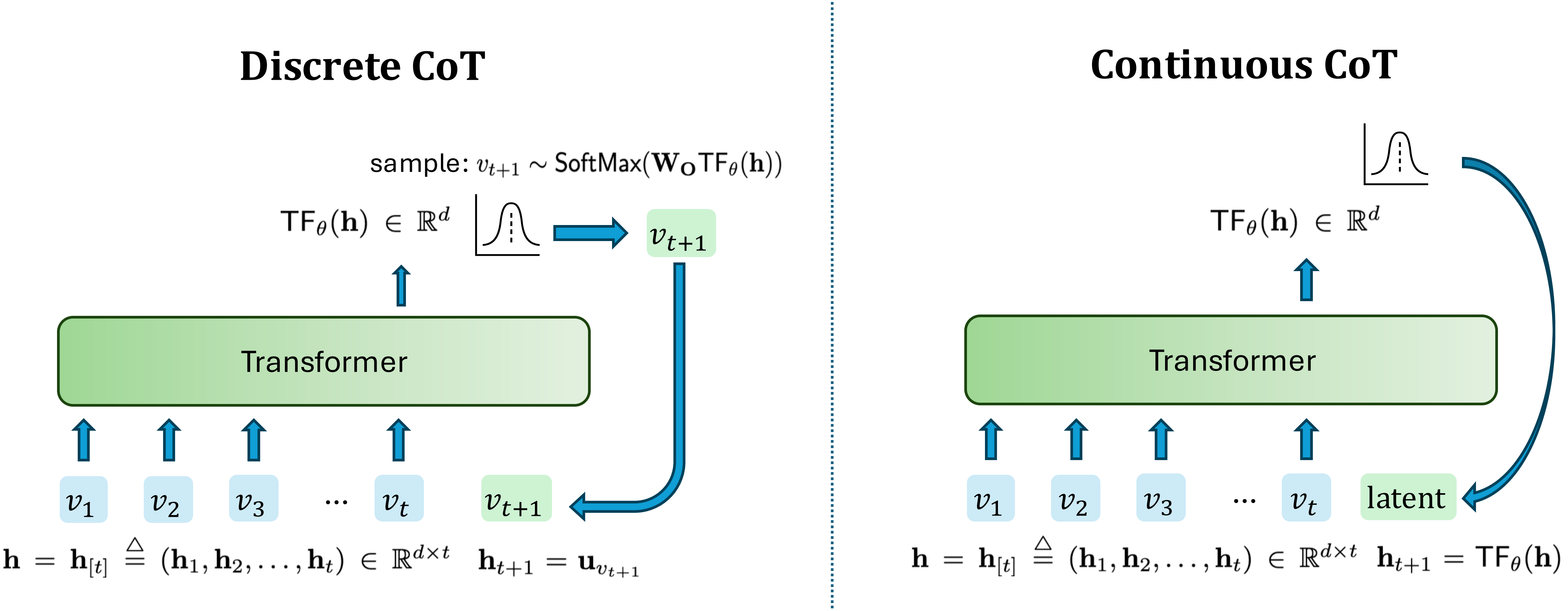
Graph reachability

- Graph reachability: Given a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, decide whether a node s can reach t
  - Many real-world reasoning problem can be abstracted as a graph (e.g., knowledge graph)
  - Many theoretical problems can be reduced to it (e.g., Turing machine halting problem)

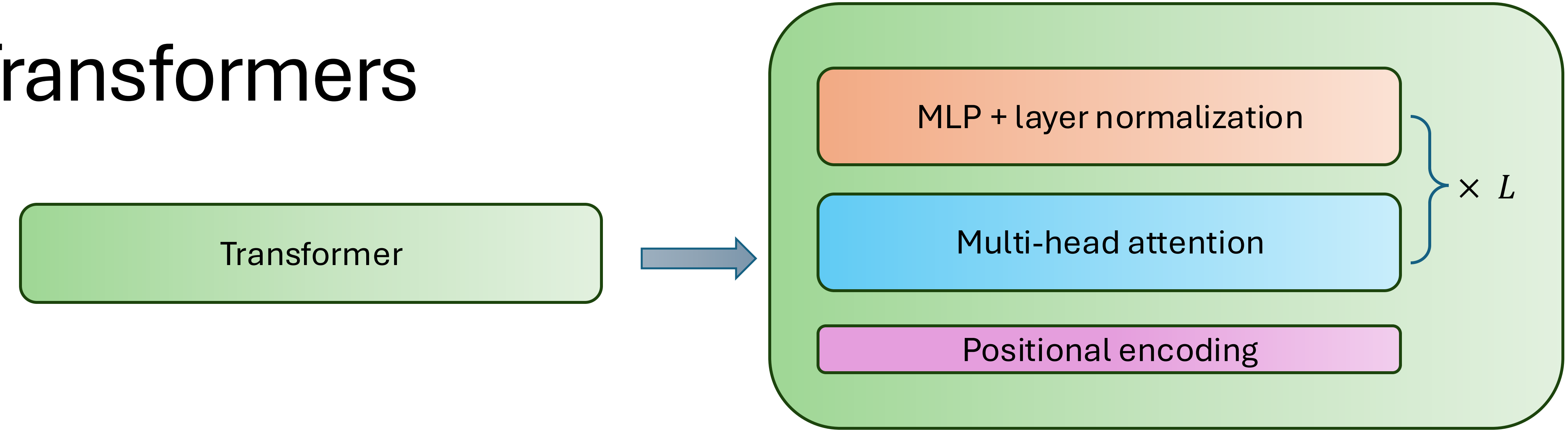


Preliminaries

- Voc = $[V]$: a vocabulary of size V
  - For any token $v \in$ Voc, it has an embedding $\vec{u}_v \in \mathbb{R}^d$



Transformers



$\mathbf{h}_i^{(0)} \leftarrow \mathbf{h}_i + \mathrm{PosEncode}_{\theta_{\mathrm{PE}}}(i), \quad \forall i \in [t]$ → Adding positional encodings

for $l = 0$ to $L - 1$ **do**

$\quad \mathbf{h}^{(l+0.5)} \leftarrow \mathbf{h}^{(l)} + \sum_{h=0}^{H_l - 1} \mathrm{Attn}_{\theta_{\mathrm{Attn}}^{(l,h)}}(\mathbf{h}^{(l)})$ → multi-head attention layers

$\quad \mathbf{h}^{(l+1)} \leftarrow \mathrm{LayerNorm}\left(\mathrm{MLP}_{\theta_{\mathrm{MLP}}^{(l)}}(\mathbf{h}^{(l+0.5)})\right)$ → MLP / FF layers & layer normalization

end for

Attentions and MLPs

Multi-head attention

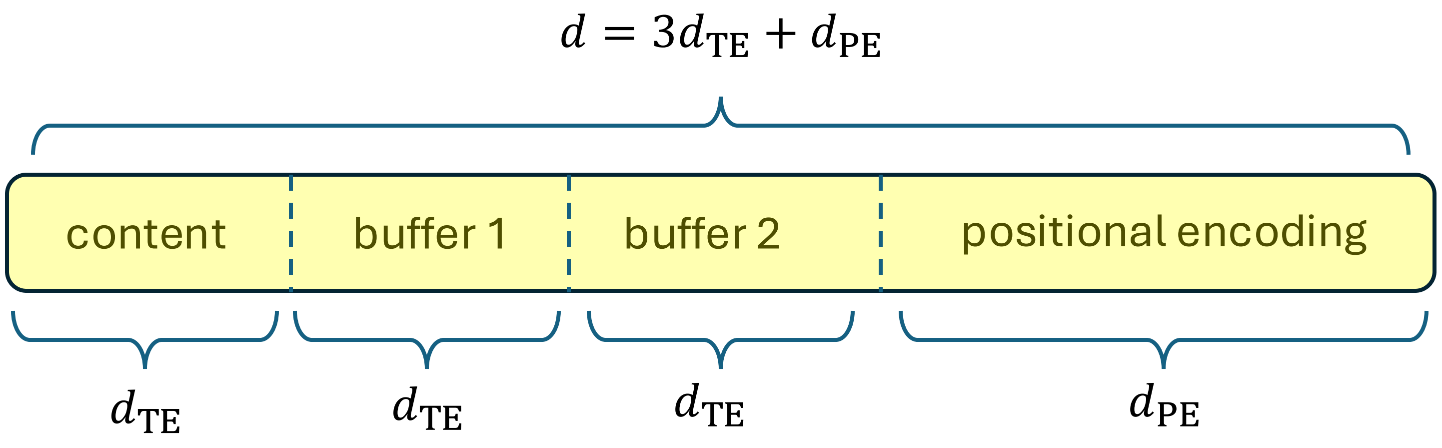
$$\mathbf{q}_i \leftarrow \mathbf{Q}\mathbf{h}_i, \quad \mathbf{k}_i \leftarrow \mathbf{K}\mathbf{h}_i, \quad \mathbf{v}_i \leftarrow \mathbf{V}\mathbf{h}_i, \quad \forall i \in [t]$$

$$s_i \leftarrow \text{SoftMax}(\langle \mathbf{q}_i, \mathbf{k}_1 \rangle, \ldots, \langle \mathbf{q}_i, \mathbf{k}_i \rangle), \quad \mathbf{h}_i^{\text{Attn}} \leftarrow \mathbf{O} \sum_{j=1}^{i} s_{i,j} \mathbf{v}_j$$

MLP

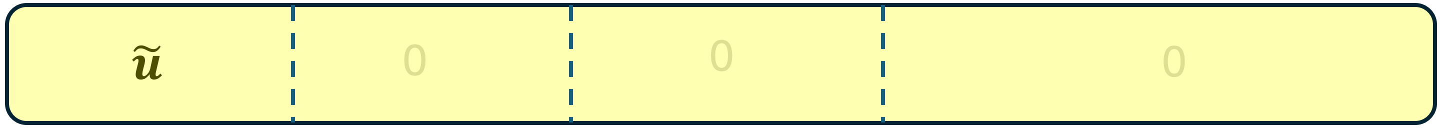
$$\mathbf{h}_i^{\text{MLP}} \leftarrow \mathbf{W}_{L_{\text{MLP}}} \sigma_{L_{\text{MLP}}-1}(\cdots \mathbf{W}_2 \sigma_1(\mathbf{W}_1 \mathbf{h}_i) \cdots)$$

Embedding space

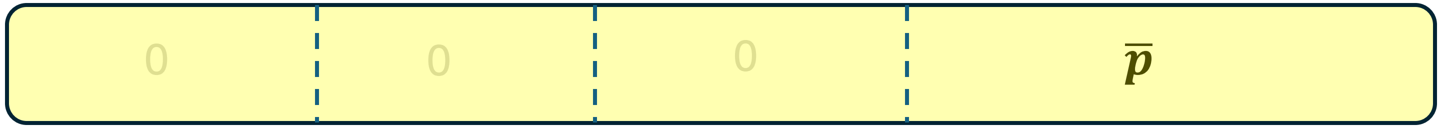


- We use $\text{content}(\vec{u})$ to represent the first d_{TE} entries for a d-dim vector $\vec{u}$
  - Define $\text{buffer}_1(\vec{u})$, $\text{buffer}_2(\vec{u})$, and $\text{pos}(\vec{u})$ similarly
  - Use $\tilde{u} = \text{content}(\vec{u})$ and $\bar{u} = \text{pos}(\vec{u})$ for convenience

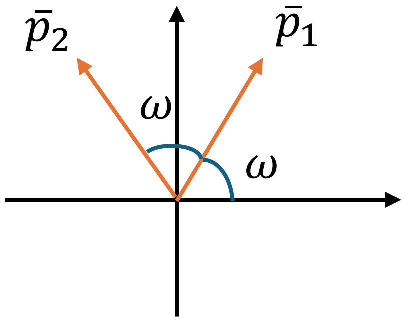
Token embeddings and positional encodings



- For token embedding $\vec{u}_v$, only the content space are non-zero
  - Define the (reduced) embedding matrix $\widetilde{U} = [\tilde{u}_1, \tilde{u}_2, \ldots, \tilde{u}_V] \in \mathbb{R}^{d_{\mathrm{TE}} \times V}$
  - Assume $\widetilde{U}^{\mathrm{T}} \widetilde{U} = \mathrm{I}$ (i.e., token embeddings are orthonormal)

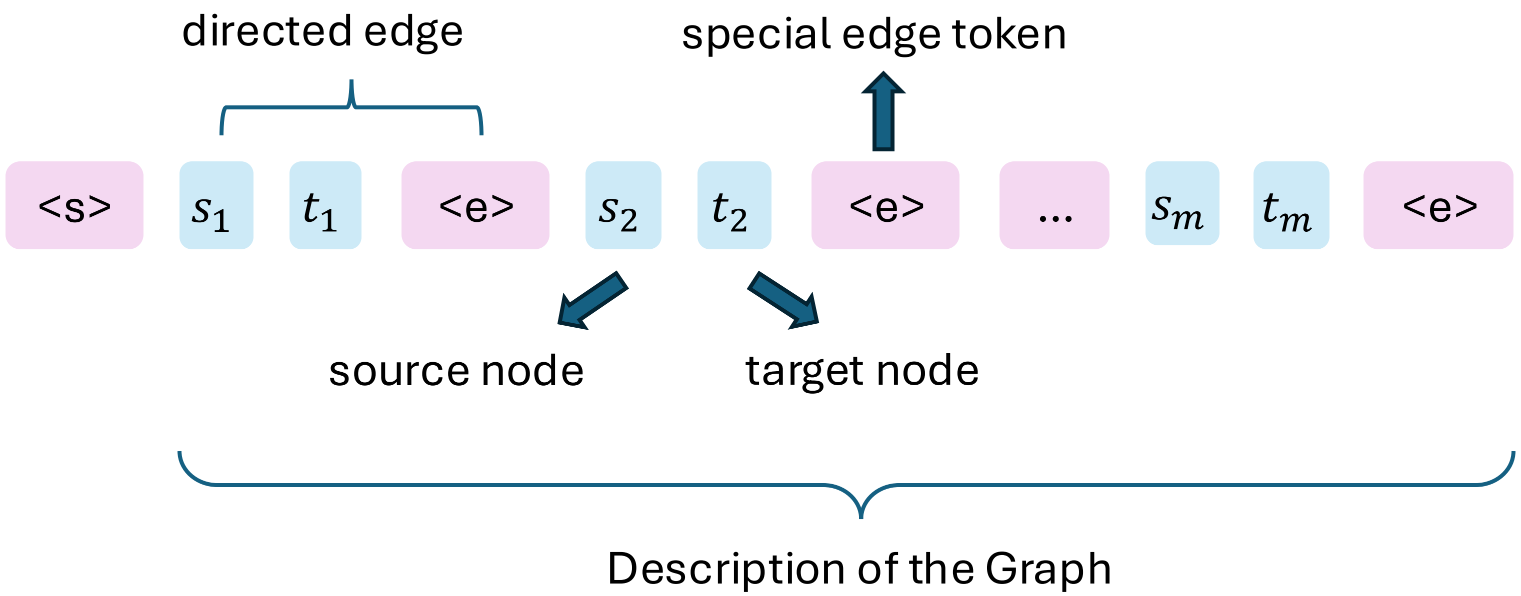


- For positional encoding $\vec{p}_i$, only the position space are non-zero
  - We use sinusoidal positional encodings
  - For any position $i \geq 1$ and $j \in [d_{\mathrm{PE}}/2]$
  - $\bar{p}_{i,2j-1} = \cos(i \cdot \omega^j), \;\; \bar{p}_{i,2j} = \sin(i \cdot \omega^j)$
    - where $\omega = M^{-2/d_{\mathrm{PE}}}$ (in practice, $M = 10^4$ for example)



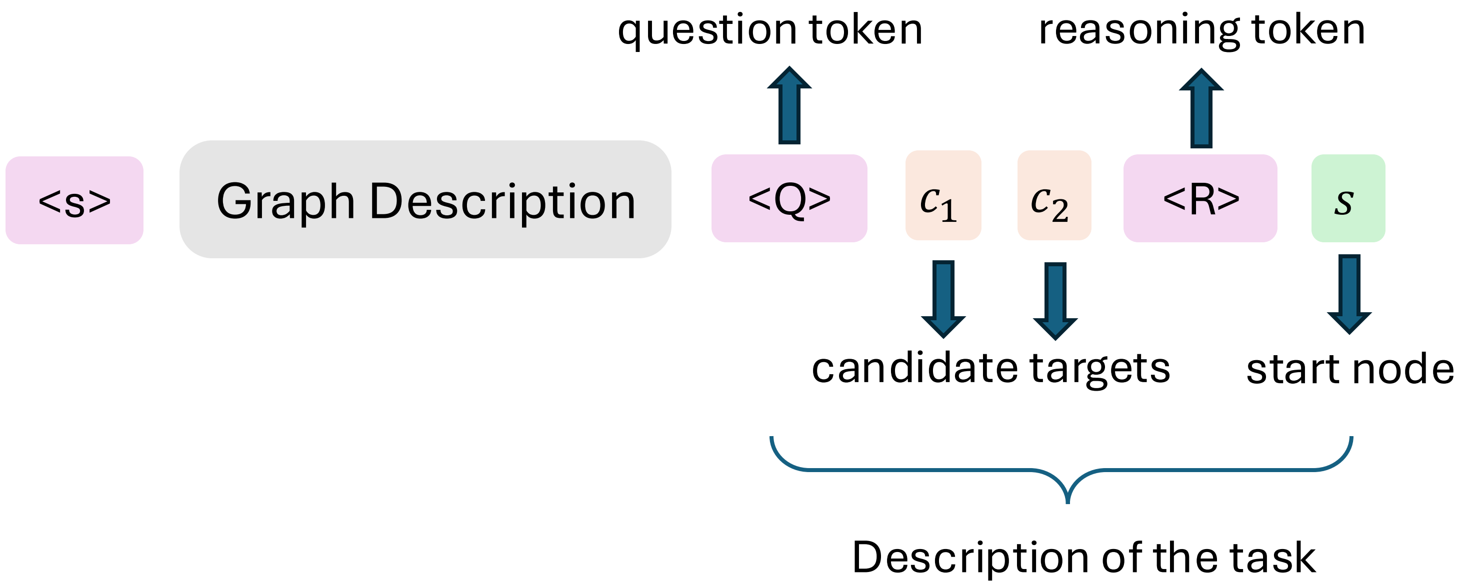
Prompt format

Given two candidate destination nodes, decide which one can be reached



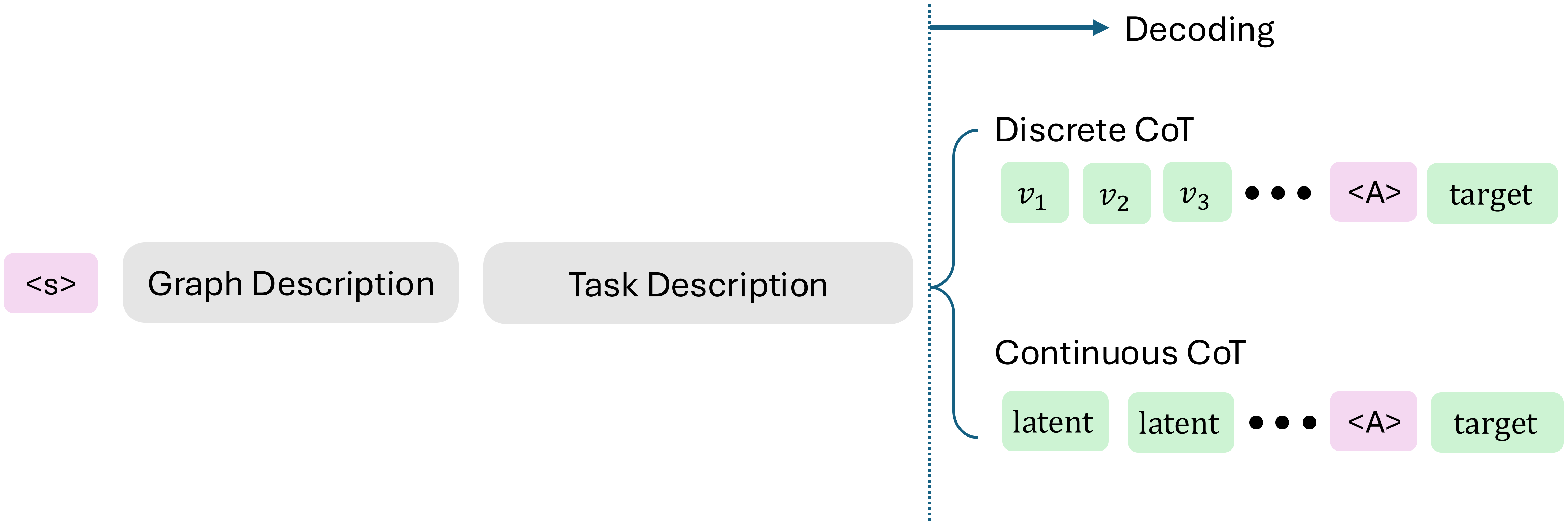
Prompt format

Given two candidate destination nodes, decide which one can be reached



Prompt format

Given two candidate destination nodes, decide which one can be reached



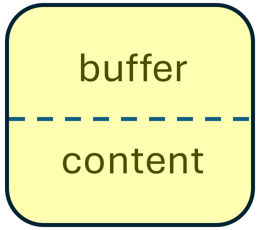
Main theorem

Theorem (informal)

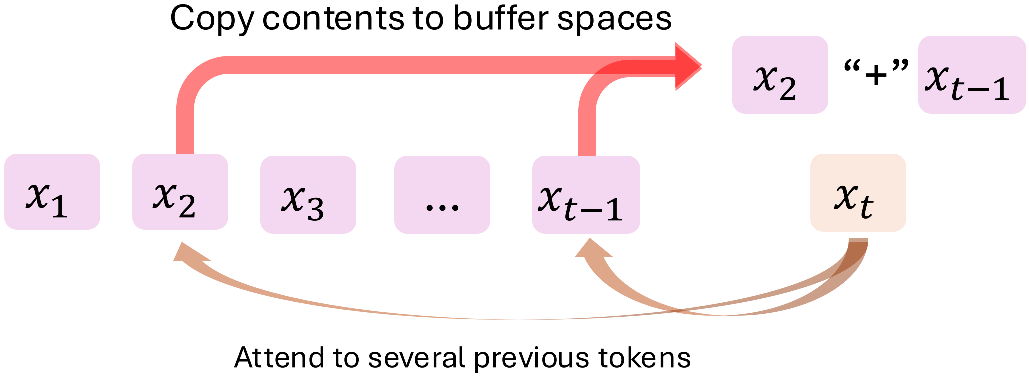
For n-vertex directed graphs, a **2-layer** transformer with continuous CoT can solve reachability using $O(n)$ decoding steps with $O(n)$ embedding dimensions.

Secret Sauce: Superposition of the embeddings!

How does a single attn-MLP block work?



simplified embedding space



Attention as an **aggregator**:

- this is a general component
- can have multiple buffers
- can move contents to different buffers

MLP as a **filter**

$$h = \sum_{v \in \mathrm{Voc}} \lambda_v \vec{u}_v$$

$$h' \propto \sum_{v \in \mathrm{Voc}} \mathbb{I}\{\lambda_v \geq \varepsilon\} \vec{u}_v$$

$$h' = W_2 \sigma(W_1 h) = U \sigma(U^T h)$$

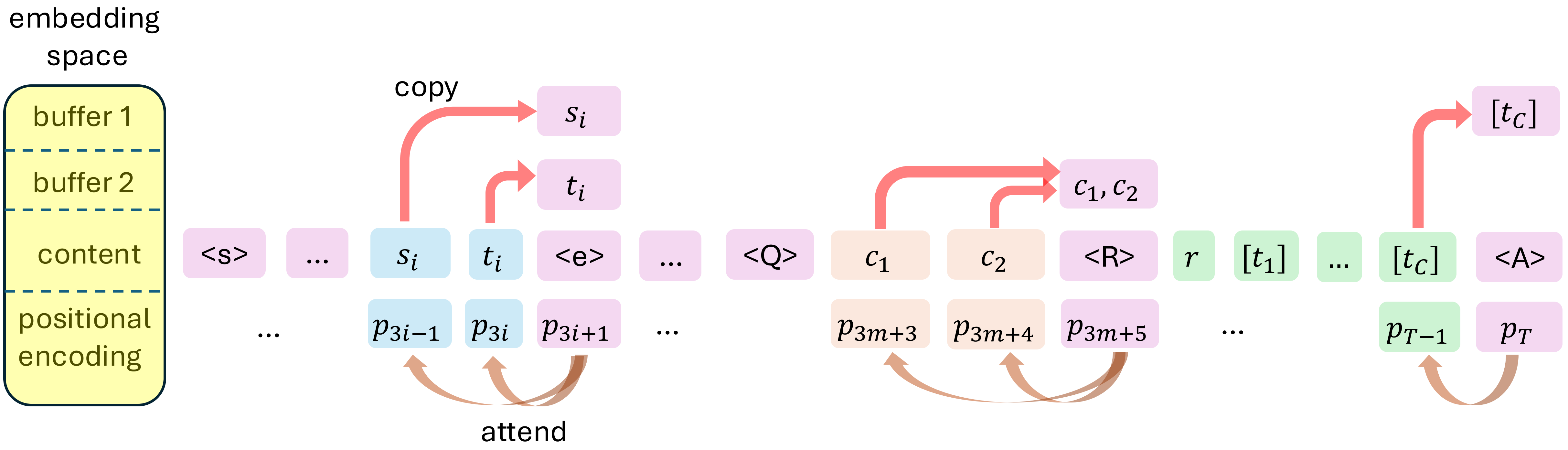
Eliminate noise

$U = [\vec{u}_1, \vec{u}_2, \dots, \vec{u}_V]$: the embedding matrix

The role of each MLP layer:

- $W_1 = U^T$: change to standard basis;
- $\sigma(\cdot) = \mathbb{I}\{\cdot \geq \varepsilon\}$: coordinate-wise filter;
- $W_2 = U$: change the basis back

First-layer attention

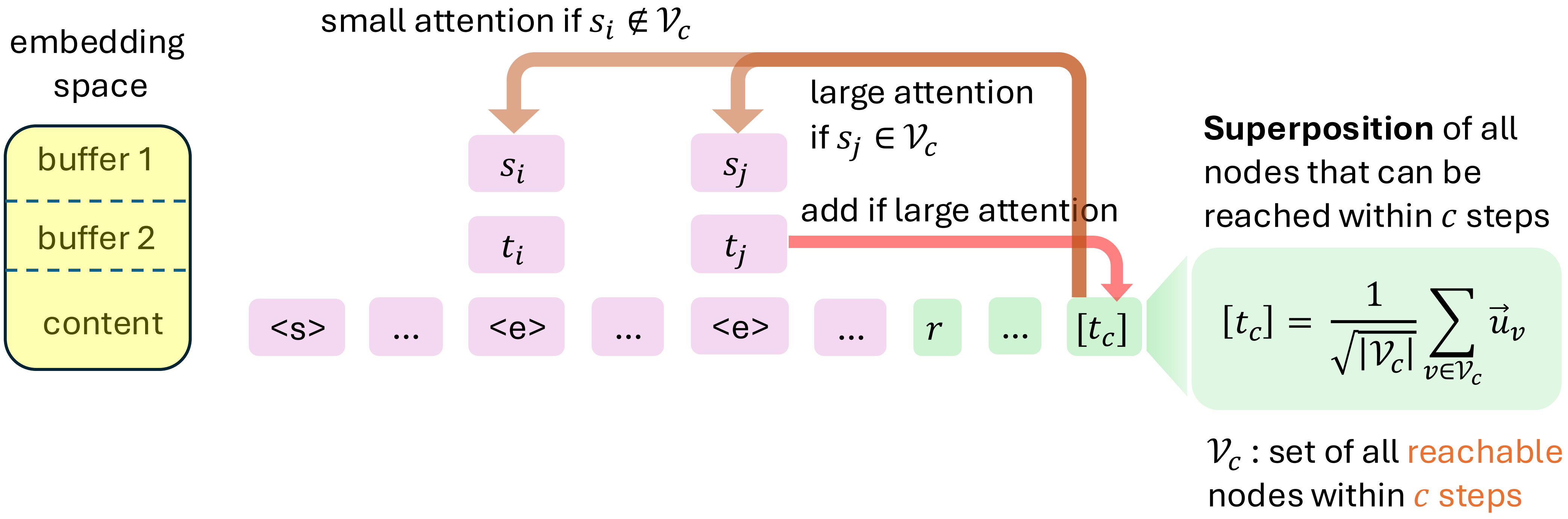


$[t_c]$ Continuous thought at step c

<A> Special answer token

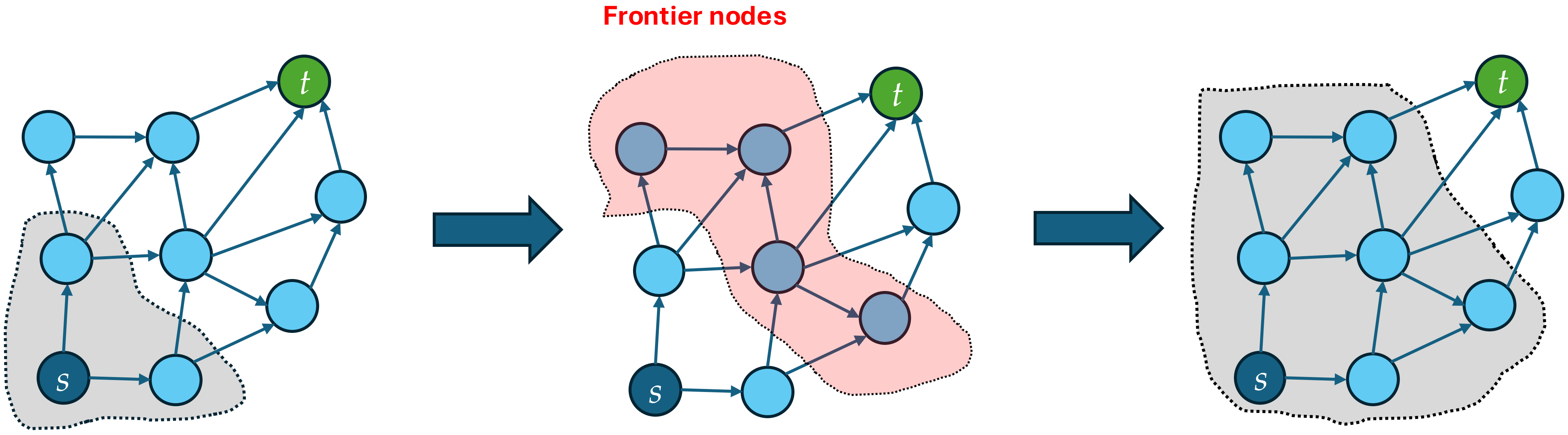
MLP layers: removing low-attended embeddings

Second-layer attention (thought generation)



One-step expansion of $\mathcal{V}_c$

Continuous CoT: Decoding as parallel BFS



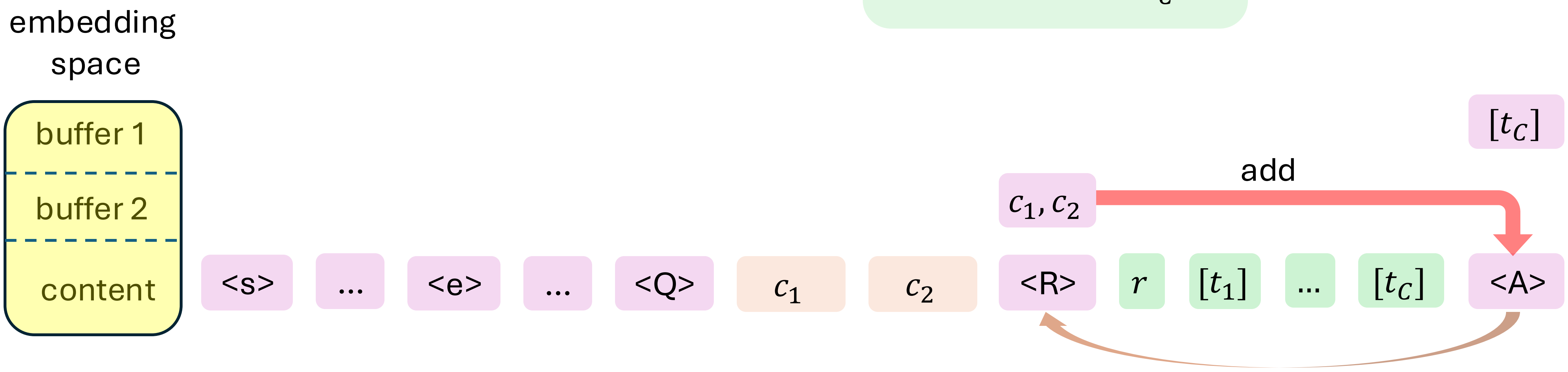
$$[t_1] = \frac{1}{\sqrt{|\mathcal{V}_1(s)|}} \sum_{v \in \mathcal{V}_1} \vec{u}_v$$

$$[t_2] = \frac{1}{\sqrt{|\mathcal{V}_2(s)|}} \sum_{v \in \mathcal{V}_2} \vec{u}_v$$

Second-layer attention (final prediction)

$$[t_C] = \frac{1}{\sqrt{|\mathcal{V}_C|}} \sum_{v \in \mathcal{V}_C} \vec{u}_v$$

Reachable set from the starting point



"Measure" $[t_C]$ using c_1 and c_2

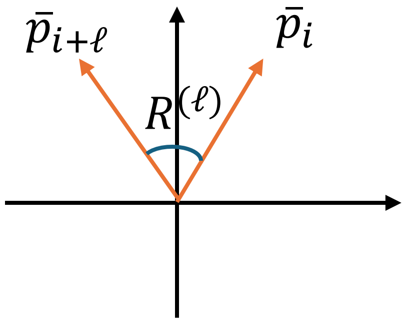
The target $c^\star$ that overlaps with **reachable set** will be picked and returned

Construction of the first-layer attention

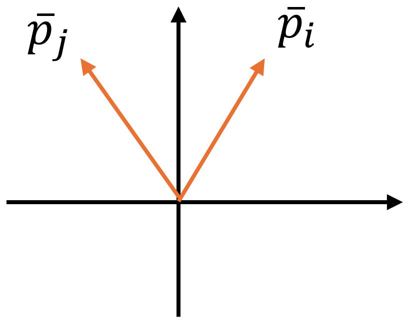
- How do transformers implement copy?
  - Naïve methods: hard-coding many position pairs
    - e.g., pos. 5 attends to pos. 4, pos. 8 attends to pos. 6
    - Drawback: not flexible, vulnerable even to a one-position shift
  - A possible solution: using relative positions
    - E.g., pos. i attends to pos. $(i - \ell)$ for some fixed ℓ
    - Drawback: not every position needs to look ℓ positions back
  - We propose a more flexible building block: attention chooser
    - Fix a special token <x>, and a positive integer ℓ
    - If the token at the current position i is <x>, then attends to position $i - \ell$
    - Otherwise attends to <s> (attention sink)

Properties of sinusoidal positional encodings

- **Proposition 1**: There exists $R^{(\ell)} \in \mathbb{R}^{d_{\text{PE}} \times d_{\text{PE}}}$, s.t., $\bar{p}_{i+\ell} = R^{(\ell)} \bar{p}_i, \ \forall i$
  - $\begin{bmatrix} \cos(\ell \cdot \omega^j) & -\sin(\ell \cdot \omega^j) \\ \sin(\ell \cdot \omega^j) & \cos(\ell \cdot \omega^j) \end{bmatrix} \begin{bmatrix} \cos(i \cdot \omega^j) \\ \sin(i \cdot \omega^j) \end{bmatrix} = \begin{bmatrix} \cos((i+\ell) \cdot \omega^j) \\ \sin((i+\ell) \cdot \omega^j) \end{bmatrix}$



- **Proposition 2**: There exists $\varepsilon > 0$, s.t., $\langle \bar{p}_i, \bar{p}_j \rangle \leq \frac{d_{\text{PE}}}{2} - \varepsilon$ for $i \neq j$
  - $\langle \bar{p}_i, \bar{p}_j \rangle = \sum_{k=1}^{d_{\text{PE}}} p_{i,k} p_{j,k}$
    $= \sum_{k=1}^{d_{\text{PE}}/2} \cos(i \cdot \omega^k) \cos(j \cdot \omega^k) + \cos(i \cdot \omega^k) \cos(j \cdot \omega^k)$
    $= \sum_{k=1}^{d_{\text{PE}}/2} \cos((i-j) \cdot \omega^k)$



Attention chooser

- A single attention head given (<x>, ℓ) that implements:
  - If the token at the current position i is <x>, then attends to position $i - \ell$
  - Otherwise attends to <s>

$$\mathbf{Q} = \begin{bmatrix} \mathbf{0}_{d_{\mathsf{PE}} \times d_{\mathsf{TE}}} & \mathbf{0}_{d_{\mathsf{PE}} \times 2d_{\mathsf{TE}}} & \mathbf{I}_{d_{\mathsf{PE}}} \\ \xi \bar{\mathbf{p}}_1 \otimes \tilde{\mathbf{u}}_{\text{<}\bar{\text{x}}\text{>}} & \mathbf{0}_{d_{\mathsf{PE}} \times 2d_{\mathsf{TE}}} & \mathbf{0}_{d_{\mathsf{PE}} \times d_{\mathsf{PE}}} \end{bmatrix}$$

$$\mathbf{K} = \begin{bmatrix} \mathbf{0}_{d_{\mathsf{PE}} \times 3d_{\mathsf{TE}}} & \eta \mathbf{R}^{(\ell)} \\ \mathbf{0}_{d_{\mathsf{PE}} \times 3d_{\mathsf{TE}}} & \eta \mathbf{I}_{d_{\mathsf{PE}}} \end{bmatrix}$$

$$\tilde{\mathbf{u}}_{\text{<}\bar{\text{x}}\text{>}} = \sum_{v \in \mathsf{Voc} \setminus \{\text{<x>}\}} \tilde{\mathbf{u}}_v \in \mathbb{R}^{d_{\mathsf{TE}}}$$

$$\mathbf{q}_i = \mathbf{Q}(\mathbf{h}_i + \mathbf{p}_i) = \begin{bmatrix} \bar{\mathbf{p}}_i \\ \xi \langle \tilde{\mathbf{u}}_{\text{<}\bar{\text{x}}\text{>}}, \tilde{\mathbf{h}}_i \rangle \bar{\mathbf{p}}_1 \end{bmatrix}$$

$$\mathbf{k}_i = \mathbf{K}(\mathbf{h}_i + \mathbf{p}_i) = \begin{bmatrix} \eta \mathbf{R}^{(\ell)} \bar{\mathbf{p}}_i \\ \eta \bar{\mathbf{p}}_i \end{bmatrix} = \begin{bmatrix} \eta \bar{\mathbf{p}}_{i+\ell} \\ \eta \bar{\mathbf{p}}_i \end{bmatrix}$$

$$\langle \mathbf{q}_i, \mathbf{k}_j \rangle = \eta \left(\langle \bar{\mathbf{p}}_i, \bar{\mathbf{p}}_{j+\ell} \rangle + \xi \langle \tilde{\mathbf{u}}_{\text{<}\bar{\text{x}}\text{>}}, \tilde{\mathbf{h}}_i \rangle \langle \bar{\mathbf{p}}_1, \bar{\mathbf{p}}_j \rangle \right)$$

Attention chooser (continued)

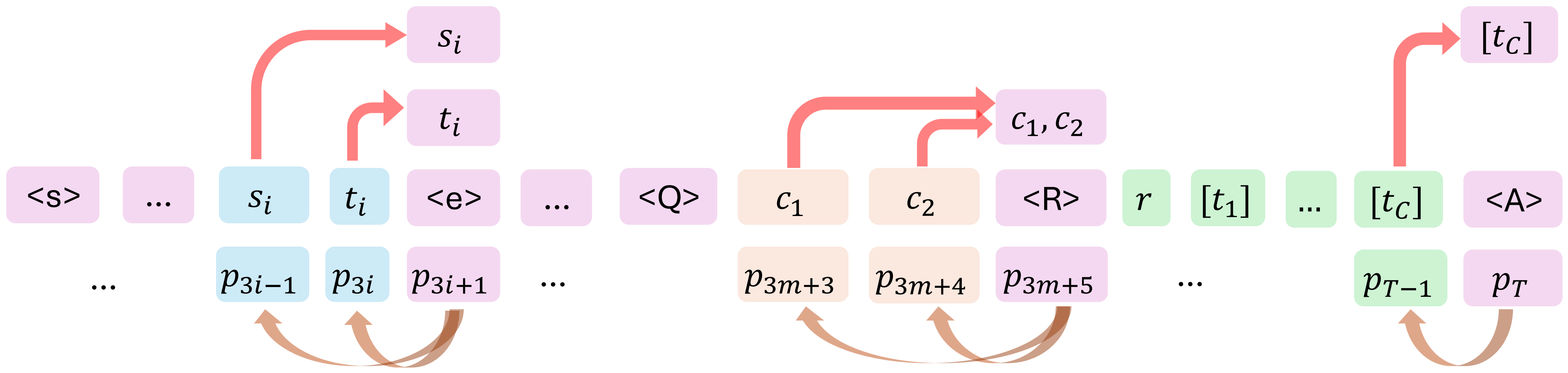
- A single attention head given (<x>, ℓ) that implements:
  - If the token at the current position i is <x>, then attends to position $i - \ell$
  - Otherwise attends to <s>

$$\langle \mathbf{q}_i, \mathbf{k}_j \rangle = \eta \left(\langle \bar{\mathbf{p}}_i, \bar{\mathbf{p}}_{j+\ell} \rangle + \xi \langle \tilde{\mathbf{u}}_{<\bar{\mathbf{x}}>}, \tilde{\mathbf{h}}_i \rangle \langle \bar{\mathbf{p}}_1, \bar{\mathbf{p}}_j \rangle \right)$$

- If $\vec{h}_i = \vec{u}_{<\mathrm{x}>}$, then the second term is zero
  - Determined only by the first term, maximized at $j = i - \ell$
- Otherwise, determined by the second term for a large ξ
  - Maximized at $j = 1$

Implementing the first-layer attention

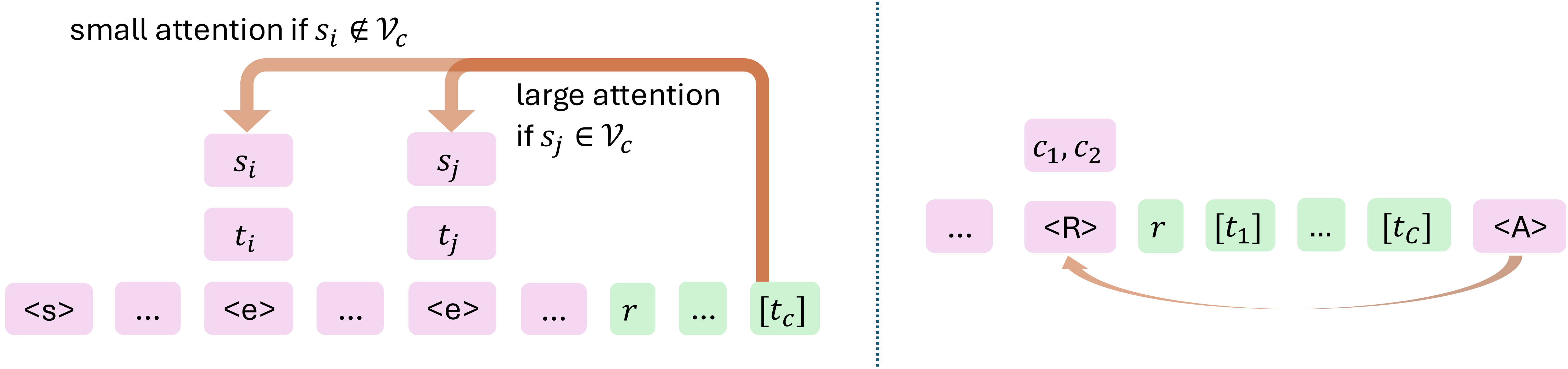
- Attention chooser is a general building block



- Five heads: (<e>, 1), (<e>, 2), (<R>, 1), (<R>, 2), (<A>, 1)
- Value matrix reads, output matrix writes

Implementing the second-layer attention

- Only requires one head



$$\mathbf{Q}^{(1)} = [\mathbf{I}_{d_{\mathsf{TE}}} \quad \mathbf{0}_{d_{\mathsf{TE}} \times d_{\mathsf{TE}}} \quad \mathbf{0}_{d_{\mathsf{TE}} \times d_{\mathsf{TE}}} \quad \mathbf{0}_{d_{\mathsf{TE}} \times d_{\mathsf{PE}}}] \in \mathbb{R}^{d_{\mathsf{TE}} \times d},$$

$$\mathbf{K}^{(1)} = [\tau \tilde{\mathbf{u}}_{\texttt{<A>}} \otimes \tilde{\mathbf{u}}_{\texttt{<R>}} \quad \tau \mathbf{I}_{d_{\mathsf{TE}}} \quad \mathbf{0}_{d_{\mathsf{TE}} \times d_{\mathsf{TE}}} \quad \mathbf{0}_{d_{\mathsf{TE}} \times d_{\mathsf{PE}}}] \in \mathbb{R}^{d_{\mathsf{TE}} \times d}$$

3. Experiments

Dataset: ProsQA

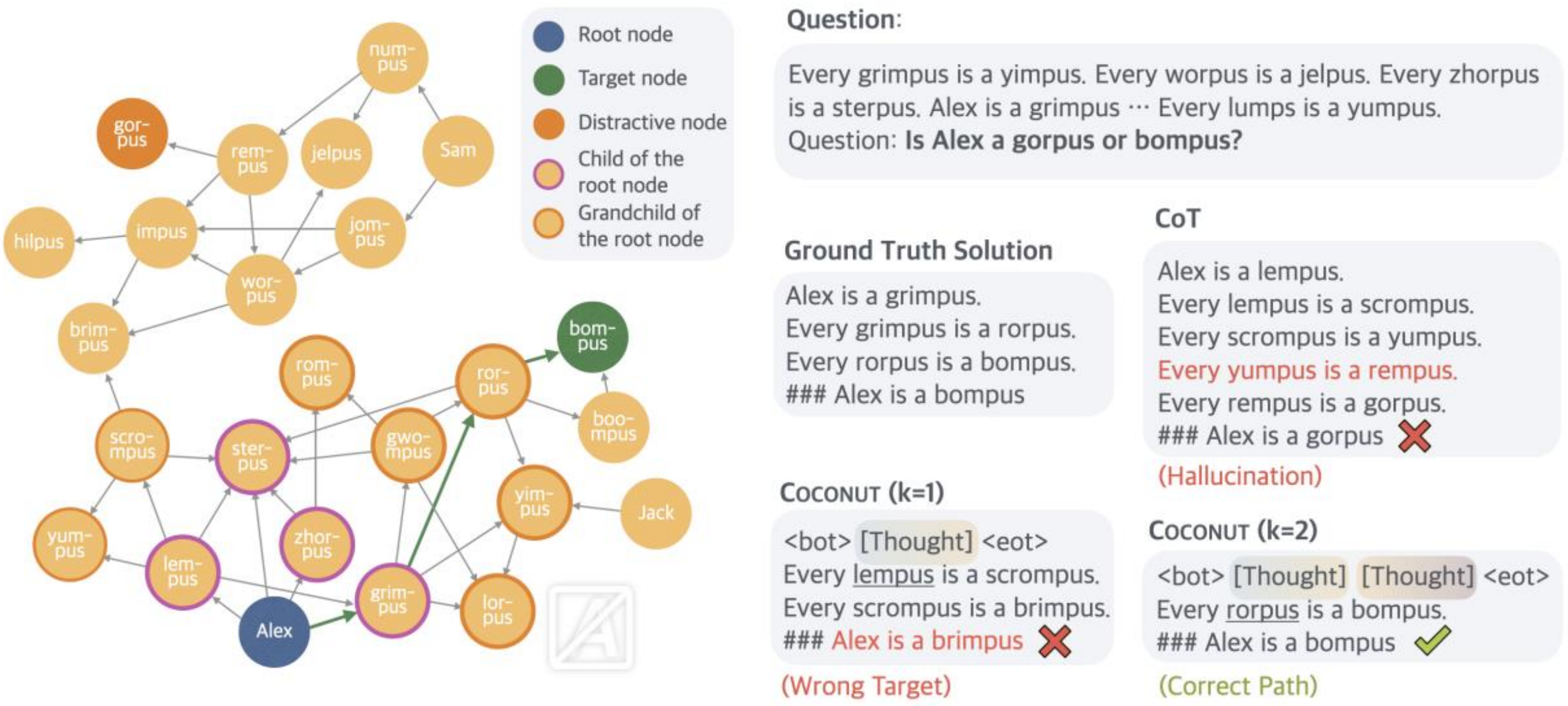


Figure credit to [1]

[1] Hao, Shibo, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. "Training large language models to reason in a continuous latent space." *arXiv preprint arXiv:2412.06769* (2024).

Dataset: ProsQA (symbolic version)

- We use a symbolic version of ProsQA
  - We train models from scratch since we change # of layers
  - Easier to observe and align with our theory

<s> s_1 t_1 <e> s_2 t_2 <e> … s_m t_m <e> <Q> c_1 c_2 <R> r

- Dataset statistics

	#Problems	$\|V\|$	$\|E\|$	Sol. Len.
Train	14785	22.8	36.5	3.5
Val	257	22.7	36.3	3.5
Test	419	22.7	36.0	3.5

Training Methods

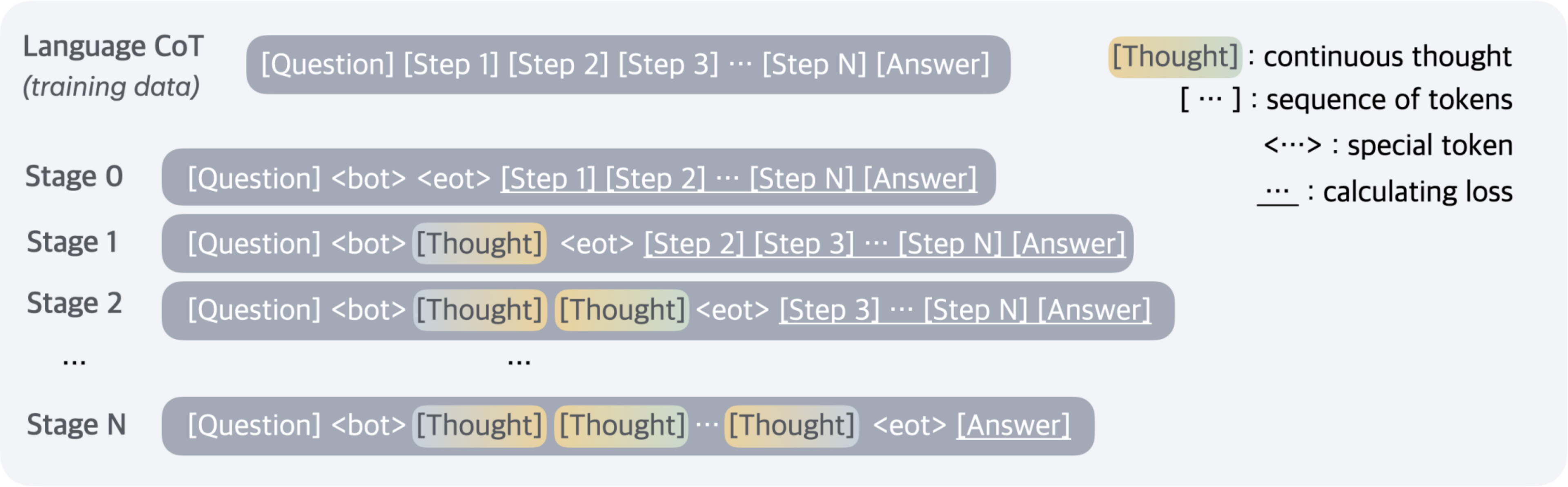


Figure credit to [1]

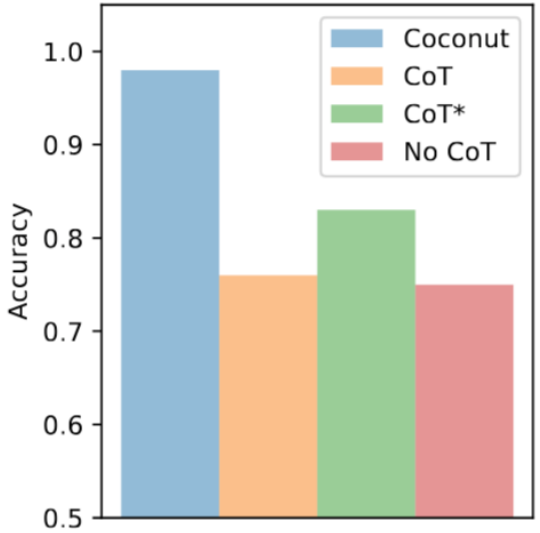
- In our experiments, we only calculate the loss at the position of <eot>

[1] Hao, Shibo, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. "Training large language models to reason in a continuous latent space." *arXiv preprint arXiv:2412.06769* (2024).

Comparison of continuous and discrete CoT

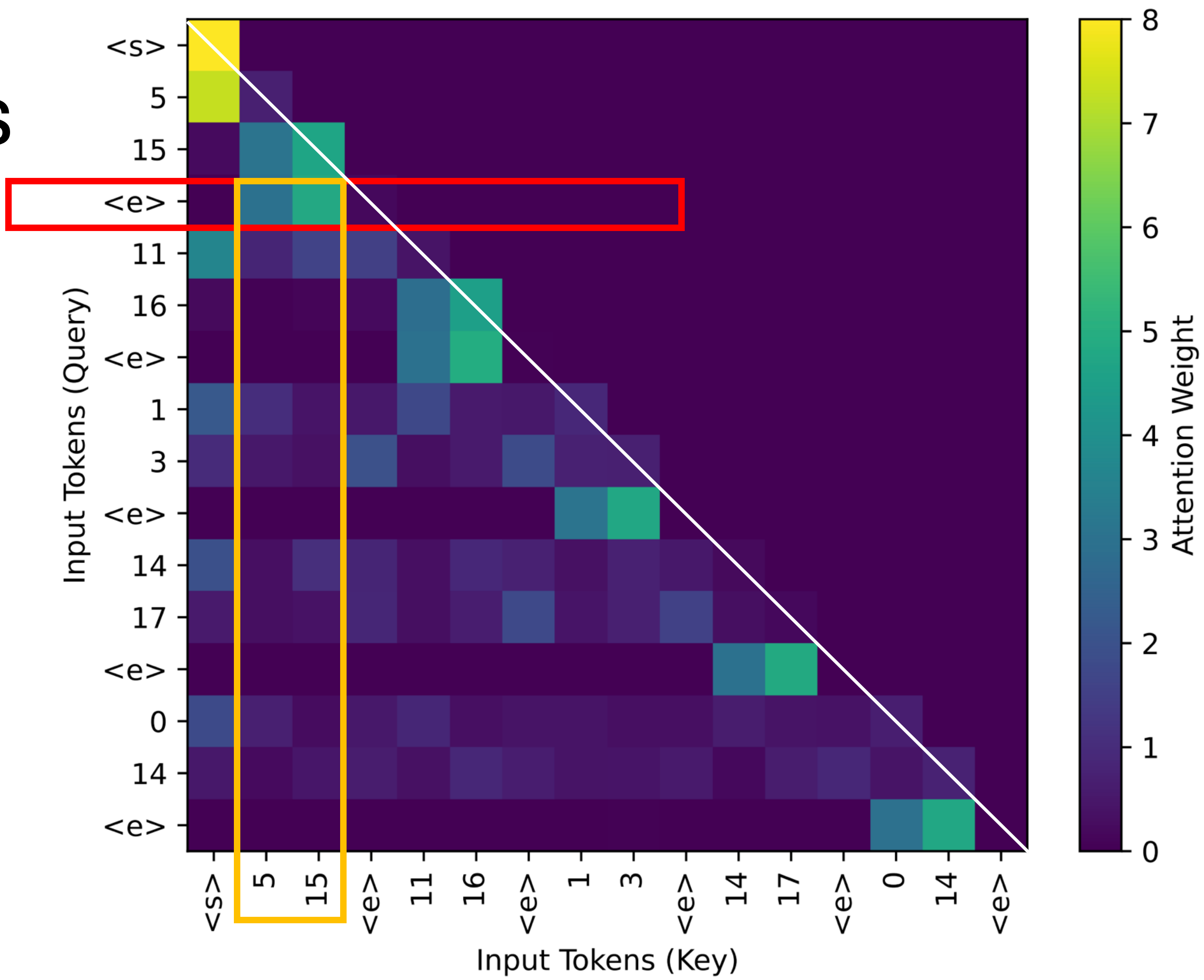
- Dataset: a subset of ProsQA[1], symbolic sequence, 3-4 steps
- Model: GPT2-style decoder
- Training: multi-stage training, stage i predicts i-th node in the optimal path using previous thoughts

- Overall results: 2-layer transformer with continuous CoT (Coconut) beats 12-layer transformer with discrete CoT (CoT*)

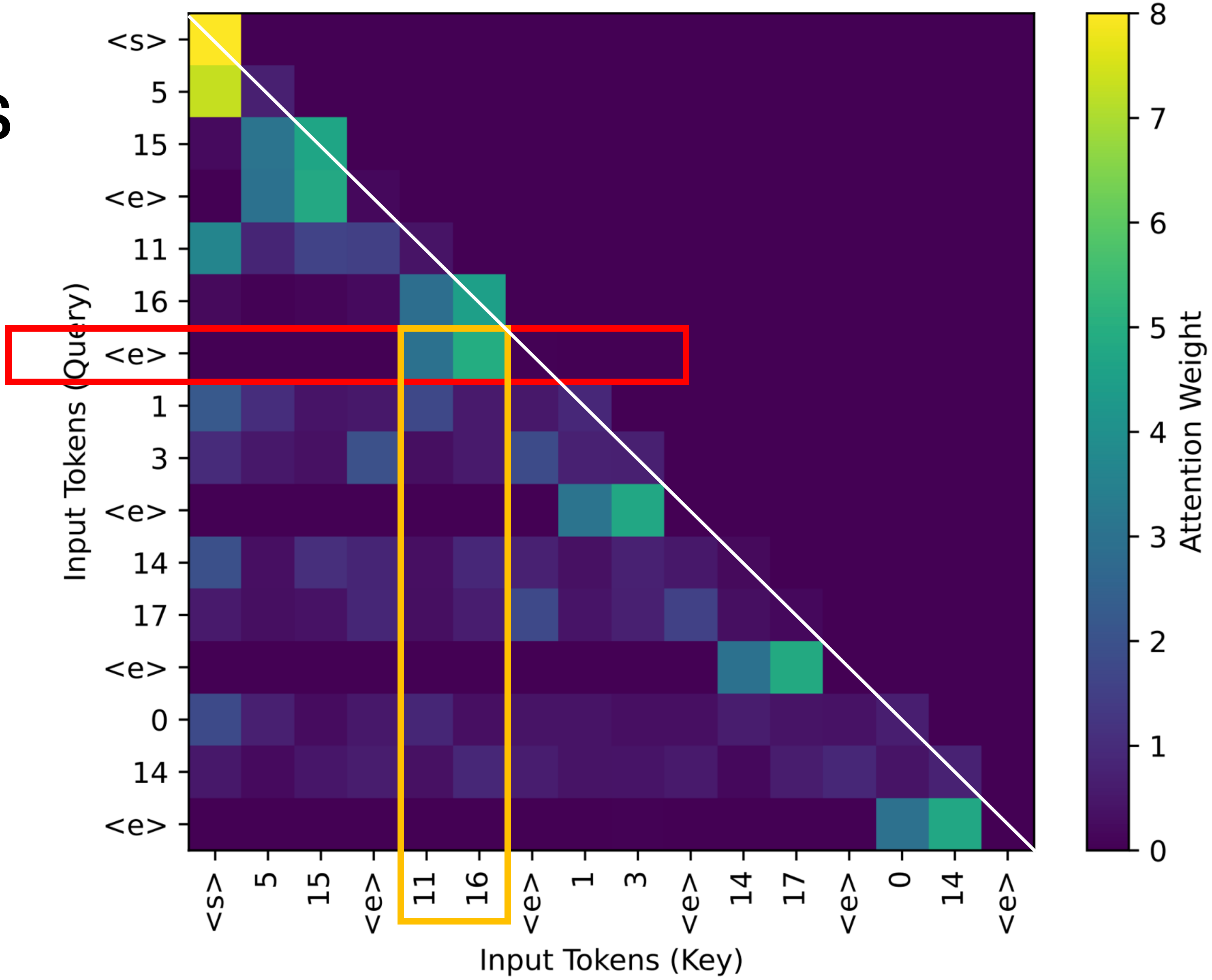


[1] Hao, Shibo, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. "Training large language models to reason in a continuous latent space." *arXiv preprint arXiv:2412.06769* (2024).

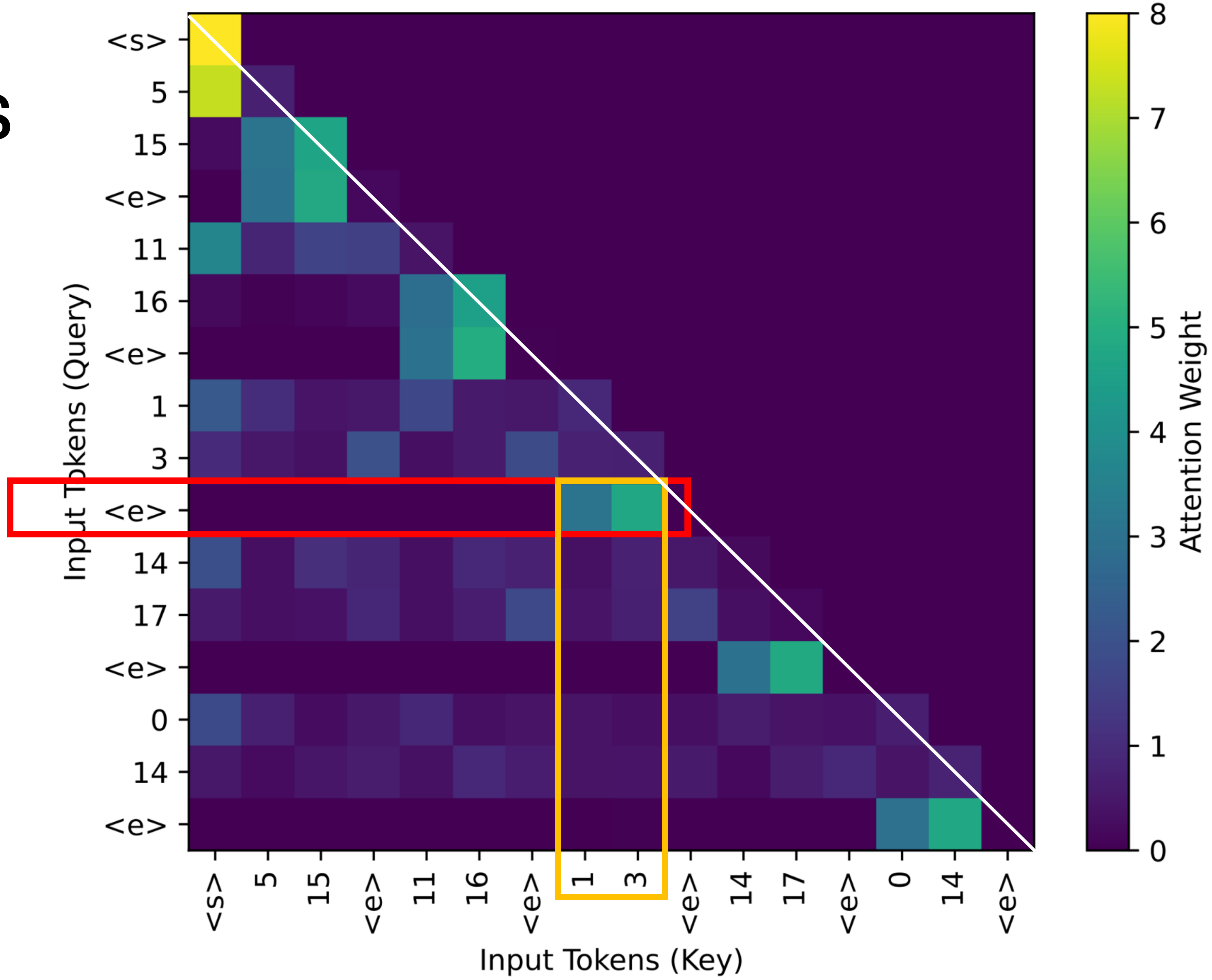
Layer 1
Attention Patterns



Layer 1
Attention Patterns



Layer 1
Attention Patterns



Visualization (Layer 2 attention)

- For step c:
  - **Reachable node** (reachable from start node within c-th steps)
    - *Frontier node* (exactly c-th steps)
    - *Optimal node* (on the shortest path from the start node to the destination node)
  - **Non-reachable node**

$$[t_c] = \frac{1}{\sqrt{|\mathcal{V}_c|}} \sum_{v \in \mathcal{V}_c} \vec{u}_v$$

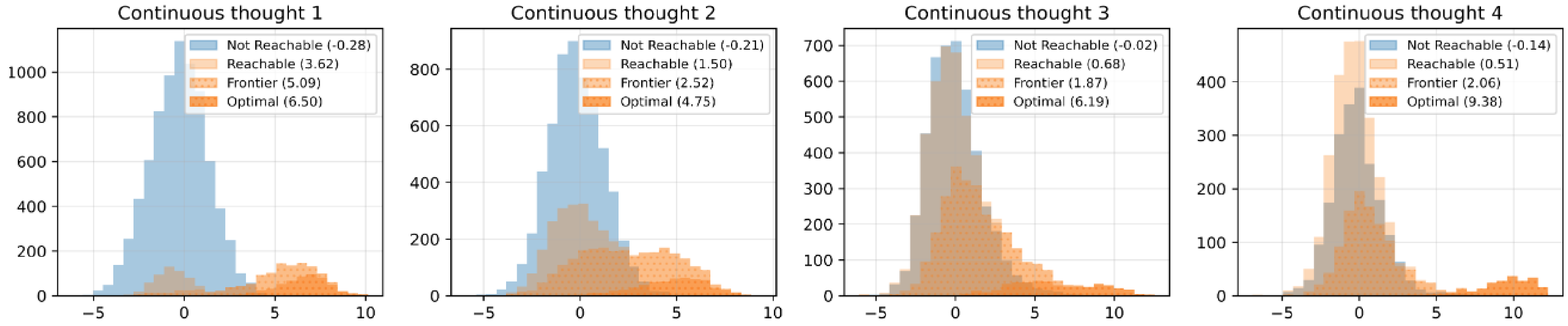
- The attention from the current thought to each edge (group)

	Step 1	Step 2	Step 3	Step 4
Not Reachable	0.04 ±0.07	0.03 ±0.09	0.08 ±0.17	0.12 ±0.20
Reachable	2.12 ±1.07	0.71 ±0.92	0.38 ±0.72	0.29 ±0.66
–Frontier	2.12 ±1.07	1.00 ±0.96	0.67 ±0.87	0.61 ±0.95
–Optimal	2.54 ±1.03	1.72 ±1.13	1.67 ±1.20	2.23 ±1.35

Visualization (superposition)

- Inner products of the current thought and each node embedding

$$[t_c] = \frac{1}{\sqrt{|\mathcal{V}_c|}} \sum_{v \in \mathcal{V}_c} \vec{u}_v$$



- Superposition emerges during training without explicit supervision
  - Note that during training, the target token is always at the optimal path

- Superposition prefers to the optimal nodes
  - Theoretical construction: uniform weights in superposition
  - Experimental results: larger weights for the optimal node
  - Models might have heuristics on which branch is more promising

4. Conclusions

Discussions

- Continuous thoughts can be powerful but hard to control
  - E.g., superposition states can be a subset of tokens (with different weights)
  - It can emerge even if the training data only contain single discrete traces
- Requires a deeper understanding if we want to use it reliably
  - Mechanism for more general tasks
  - How superposition emerges during training and how to control it

Thanks!