Limitations and Comparisons of Attention Mechanisms

Based on intership work at Taylor TCS Lab,
joint with Xiaoyu He, Chao Liao, Chen Wu, and Pinyan Lu

Xiaowei Ye

May 30, 2026

Outline

Motivation

- Transformer core: self-attention, but standard full attention has $O(n^2)$ complexity.

Motivation

- Transformer core: self-attention, but standard full attention has $O(n^2)$ complexity.
- Efficient variants: linear attention, log-linear attention, sparse attention, hybrid architectures.

Motivation

- Transformer core: self-attention, but standard full attention has $O(n^2)$ complexity.
- Efficient variants: linear attention, log-linear attention, sparse attention, hybrid architectures.
- Key question: Is there any loss in expressive power for these variants?

Table of Results

	Eva	**2-Sum**	***L*-SeqComp**
Full Attention	$O(\text{poly} \log n)$[1,4]	$O(\log n)$[3]	$O(\text{poly} \log n)$[2]
Linear + CoT	$\Omega(n \log n)$[1,4]	$\Omega(n \log n)$[5]	
Log-linear + CoT	$\Omega(n)$[5]	$\Omega(n)$[5]	
Sparse (Block)		$\Omega(B \log n)$[5]	
Hybrid			$\Omega(\text{poly } n)$[5]

1. Bhattamishra, S., Hahn, M., Blunsom, P., Kanade, V. (2024). Separations in the representational capabilities of transformers and recurrent architectures. *NeurIPS*.
2. Chen, L., Peng, B., Wu, H. (2025). Theoretical limitations of multi-layer transformer. *FOCS*.
3. Sanford, C., Hsu, D., Telgarsky, M. (2023). Representational strengths and limitations of transformers. *NeurIPS*.
4. Wen, K., Dang, X., Lyu, K. (2025). RNNs are not Transformers (yet): The key bottleneck on in-context retrieval. *ICLR*.
5. Ye, X., He, X., Liao, C., Wu, C., Lu, P. (2026). A Provable Expressiveness Hierarchy in Hybrid Linear-Full Attention. *ICML*.

Full Attention

- Layer ℓ, head h, position i:

$$y_i^{(\ell,h)} = \sum_{j \le i} \alpha_{i,j}^{(\ell,h)} V^{(\ell,h)} x_j^{(\ell-1)}$$

$$\alpha_{i,j}^{(\ell,h)} = \frac{\exp\left((x_i^{(\ell-1)} Q^{(\ell,h)})^\top K^{(\ell,h)} x_j^{(\ell-1)}\right)}{\sum_{j \le i} \exp\left((x_i^{(\ell-1)} Q^{(\ell,h)})^\top K^{(\ell,h)} x_j^{(\ell-1)}\right)}$$

Full Attention

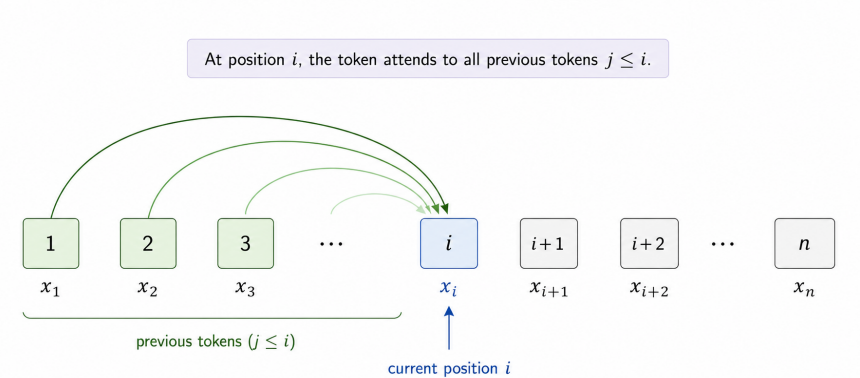
- Layer ℓ, head h, position i:

$$y_i^{(\ell,h)} = \sum_{j \le i} \alpha_{i,j}^{(\ell,h)} V^{(\ell,h)} x_j^{(\ell-1)}$$

$$\alpha_{i,j}^{(\ell,h)} = \frac{\exp\left((x_i^{(\ell-1)} Q^{(\ell,h)})^\top K^{(\ell,h)} x_j^{(\ell-1)}\right)}{\sum_{j \le i} \exp\left((x_i^{(\ell-1)} Q^{(\ell,h)})^\top K^{(\ell,h)} x_j^{(\ell-1)}\right)}$$

- Complexity $O(n^2)$, but strongest expressive power.

Illustration



Linear Attention $\equiv$ RNN (recurrent neural network)

- Linear attention uses a kernel ϕ:

$$\alpha_{i,j}^{(\ell,h)} = \frac{\phi(Qx_i)^\top \phi(Kx_j)}{\sum_{j\le i} \phi(Qx_i)^\top \phi(Kx_j)} = \frac{\phi(Qx_i)^\top S_i}{\phi(Qx_i)^\top Z_i}$$

with $S_i = S_{i-1} + Vx_i \otimes \phi(Kx_i), \quad Z_i = Z_{i-1} + \phi(Kx_i)$.

Linear Attention $\equiv$ RNN (recurrent neural network)

- Linear attention uses a kernel ϕ:

$$\alpha_{i,j}^{(\ell,h)} = \frac{\phi(Qx_i)^\top \phi(Kx_j)}{\sum_{j\le i} \phi(Qx_i)^\top \phi(Kx_j)} = \frac{\phi(Qx_i)^\top S_i}{\phi(Qx_i)^\top Z_i}$$

with $S_i = S_{i-1} + Vx_i \otimes \phi(Kx_i), \quad Z_i = Z_{i-1} + \phi(Kx_i)$.

Definition (RNN)

An RNN maintains a **hidden state** $h_t \in \mathbb{R}^m$, updated as:

$$h_t = g(x_t, h_{t-1}), \qquad y_t = f(h_t),$$

where g (transition) and f (output) are arbitrary functions.

Linear Attention $\equiv$ RNN (recurrent neural network)

- Linear attention uses a kernel ϕ:

$$\alpha_{i,j}^{(\ell,h)} = \frac{\phi(Qx_i)^\top \phi(Kx_j)}{\sum_{j\le i} \phi(Qx_i)^\top \phi(Kx_j)} = \frac{\phi(Qx_i)^\top S_i}{\phi(Qx_i)^\top Z_i}$$

with $S_i = S_{i-1} + Vx_i \otimes \phi(Kx_i), \quad Z_i = Z_{i-1} + \phi(Kx_i)$.

Definition (RNN)

An RNN maintains a **hidden state** $h_t \in \mathbb{R}^m$, updated as:

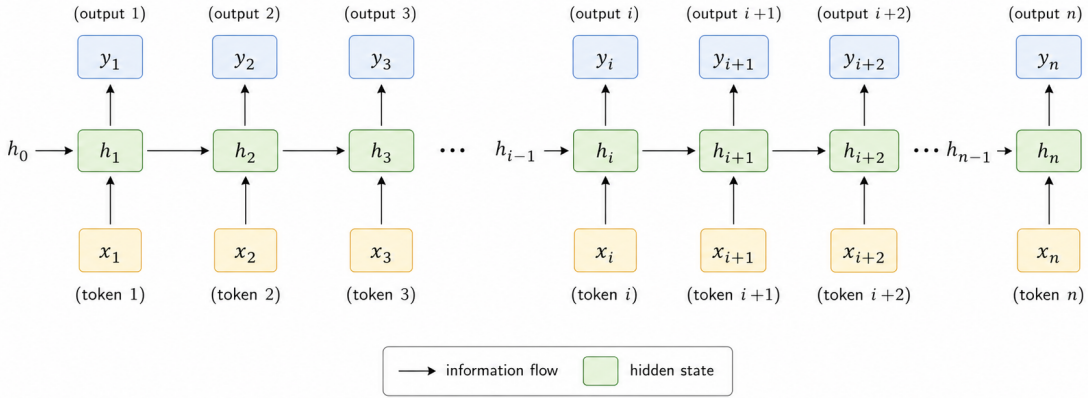
$$h_t = g(x_t, h_{t-1}), \qquad y_t = f(h_t),$$

where g (transition) and f (output) are arbitrary functions.

- Hidden state $h_i = (S_i, Z_i) \in \mathbb{R}^{d\times d} \times \mathbb{R}^d \Rightarrow$ linear attention $=$ RNN

Illustration of RNN

RNN: at position i, the hidden state is computed from the previous hidden state (only $i-1$).



Log-Linear Attention and Sparse Attention

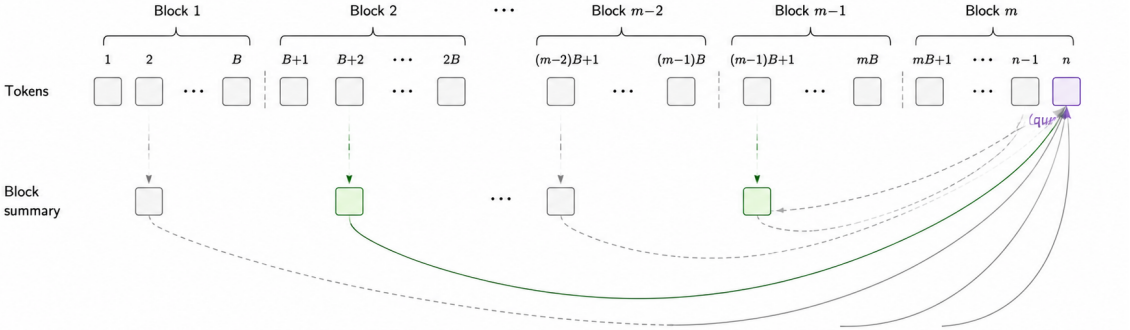
- **Log-linear attention** (Guo et al. 2025): maintain $O(\log t)$ states, achieving logarithmic time/memory.

Log-Linear Attention and Sparse Attention

- **Log-linear attention** (Guo et al. 2025): maintain $O(\log t)$ states, achieving logarithmic time/memory.
- **Sparse attention**: block compression + selection, e.g. (B, k)-sparse attention: $y_i = \lambda y_i^{\text{compression}} + (1-\lambda) y_i^{\text{select}}$ with block size B, selecting k blocks.

Log-Linear Attention and Sparse Attention

- **Log-linear attention** (Guo et al. 2025): maintain $O(\log t)$ states, achieving logarithmic time/memory.
- **Sparse attention**: block compression + selection, e.g. (B, k)-sparse attention: $y_i = \lambda y_i^{\text{compression}} + (1-\lambda) y_i^{\text{select}}$ with block size B, selecting k blocks.



Linear Attention Lower Bound for Function Evaluation

Definition (Function Evaluation (Eva))

Input: A function $f : [n] \to [n]$ represented by n tokens $(f(1), \ldots, f(n))$, and an element $x \in [n]$ as a single token. Output: $f(x)$.

Linear Attention Lower Bound for Function Evaluation

Definition (Function Evaluation (Eva))

Input: A function $f : [n] \to [n]$ represented by n tokens $(f(1), \ldots, f(n))$, and an element $x \in [n]$ as a single token. Output: $f(x)$.

Theorem (Ye et al. 2026, not a new result)

L-layer linear attention solving Eva requires $LHd(d+1)p \geq n \log n$; whereas a single-layer full attention requires only $O(\text{poly} \log n)$.

Linear Attention Lower Bound for Function Evaluation

Definition (Function Evaluation (Eva))

Input: A function $f : [n] \to [n]$ represented by n tokens $(f(1), \ldots, f(n))$, and an element $x \in [n]$ as a single token. Output: $f(x)$.

Theorem (Ye et al. 2026, not a new result)

L-layer linear attention solving Eva requires $LHd(d+1)p \geq n \log n$; whereas a single-layer full attention requires only $O(\text{poly} \log n)$.

Communication Model (2 players, L rounds):

- Alice receives $f(1), \ldots, f(n)$; Bob receives x.

Linear Attention Lower Bound for Function Evaluation

Definition (Function Evaluation (Eva))

Input: A function $f : [n] \to [n]$ represented by n tokens $(f(1), \ldots, f(n))$, and an element $x \in [n]$ as a single token. Output: $f(x)$.

Theorem (Ye et al. 2026, not a new result)

L-layer linear attention solving Eva requires $LHd(d+1)p \geq n \log n$*; whereas a single-layer full attention requires only* $O(\text{poly} \log n)$.

Communication Model (2 players, L rounds):

- Alice receives $f(1), \ldots, f(n)$; Bob receives x.
- In round ℓ, Alice sends a message of $Hd(d+1)p$ bits to Bob (corresponds to hidden state of the ℓ-th RNN layer at position n).

Linear Attention Lower Bound for Function Evaluation

Definition (Function Evaluation (Eva))

Input: A function $f : [n] \to [n]$ represented by n tokens $(f(1), \ldots, f(n))$, and an element $x \in [n]$ as a single token. Output: $f(x)$.

Theorem (Ye et al. 2026, not a new result)

L-layer linear attention solving Eva requires $LHd(d+1)p \geq n \log n$; whereas a single-layer full attention requires only $O(\text{poly} \log n)$.

Communication Model (2 players, L rounds):

- Alice receives $f(1), \ldots, f(n)$; Bob receives x.
- In round ℓ, Alice sends a message of $Hd(d+1)p$ bits to Bob (corresponds to hidden state of the ℓ-th RNN layer at position n).
- L rounds total bits communicated: $LHmp$. Bob outputs the answer.

Linear Attention Lower Bound for Function Evaluation

Definition (Function Evaluation (Eva))

Input: A function $f : [n] \to [n]$ represented by n tokens $(f(1), \ldots, f(n))$, and an element $x \in [n]$ as a single token. Output: $f(x)$.

Theorem (Ye et al. 2026, not a new result)

L-layer linear attention solving Eva requires $LHd(d+1)p \geq n \log n$; whereas a single-layer full attention requires only $O(\text{poly} \log n)$.

Communication Model (2 players, L rounds):

- Alice receives $f(1), \ldots, f(n)$; Bob receives x.
- In round ℓ, Alice sends a message of $Hd(d+1)p$ bits to Bob (corresponds to hidden state of the ℓ-th RNN layer at position n).
- L rounds total bits communicated: $LHmp$. Bob outputs the answer.
- To distinguish all n^n possible functions, need $2^{LHd(d+1)p} \geq n^n \Rightarrow LHd(d+1)p \geq n \log n$.

Full Attention Construction: Retrieval Head

The retrieval head (Chen et al. 2025) enables a single-layer full attention Transformer to solve retrieval tasks with $p = \Theta(\log n)$ precision.

Full Attention Construction: Retrieval Head

The retrieval head (Chen et al. 2025) enables a single-layer full attention Transformer to solve retrieval tasks with $p = \Theta(\log n)$ precision.

Retrieval Task:

- Input: First n tokens: $x_i = (a_i, b_i)$ with $a_i, b_i \in \{0,1\}^D$; Last token x_{n+1}: query vector a.
- Goal: find unique i with $a_i = a$, output b_i.

Full Attention Construction: Retrieval Head

The retrieval head (Chen et al. 2025) enables a single-layer full attention Transformer to solve retrieval tasks with $p = \Theta(\log n)$ precision.

Retrieval Task:

- Input: First n tokens: $x_i = (a_i, b_i)$ with $a_i, b_i \in \{0,1\}^D$; Last token x_{n+1}: query vector a.
- Goal: find unique i with $a_i = a$, output b_i.

Implementation:

- $V_i = b_i, \quad K_i = \log^2(n) \cdot (a_i, \mathbf{1} - a_i), \quad Q_{n+1} = \log^2(n) \cdot (a, \mathbf{1} - a)$.
- $\langle Qx_{n+1}, Kx_i \rangle = \log^2(n) \cdot D$ if $a_i = a$, else $\leq \log^2(n)(D-1)$.

Full Attention Construction: Retrieval Head

The retrieval head (Chen et al. 2025) enables a single-layer full attention Transformer to solve retrieval tasks with $p = \Theta(\log n)$ precision.

Retrieval Task:

- Input: First n tokens: $x_i = (a_i, b_i)$ with $a_i, b_i \in \{0,1\}^D$; Last token x_{n+1}: query vector a.
- Goal: find unique i with $a_i = a$, output b_i.

Implementation:

- $V_i = b_i, \quad K_i = \log^2(n) \cdot (a_i, \mathbf{1} - a_i), \quad Q_{n+1} = \log^2(n) \cdot (a, \mathbf{1} - a)$.
- $\langle Qx_{n+1}, Kx_i \rangle = \log^2(n) \cdot D$ if $a_i = a$, else $\leq \log^2(n)(D-1)$.

Attention Concentration:

$$\alpha_{n+1,i} \geq \frac{\exp(\log^2 n)}{\exp(\log^2 n) + n - 1} \geq 1 - \frac{n}{n \log n}, \quad \alpha_{n+1,j} \leq \frac{1}{n \log n} \ (j \neq i).$$

Full Attention Construction: Retrieval Head

The retrieval head (Chen et al. 2025) enables a single-layer full attention Transformer to solve retrieval tasks with $p = \Theta(\log n)$ precision.

Retrieval Task:

- Input: First n tokens: $x_i = (a_i, b_i)$ with $a_i, b_i \in \{0,1\}^D$; Last token x_{n+1}: query vector a.
- Goal: find unique i with $a_i = a$, output b_i.

Implementation:

- $V_i = b_i, \quad K_i = \log^2(n) \cdot (a_i, \mathbf{1} - a_i), \quad Q_{n+1} = \log^2(n) \cdot (a, \mathbf{1} - a)$.
- $\langle Qx_{n+1}, Kx_i \rangle = \log^2(n) \cdot D$ if $a_i = a$, else $\leq \log^2(n)(D-1)$.

Attention Concentration:

$$\alpha_{n+1,i} \geq \frac{\exp(\log^2 n)}{\exp(\log^2 n) + n - 1} \geq 1 - \frac{n}{n \log n}, \quad \alpha_{n+1,j} \leq \frac{1}{n \log n} \ (j \neq i).$$

$p = \Theta(\log n) \Rightarrow$ attention concentrates on position $i \Rightarrow$ retrieving b_i
$\Rightarrow$ solving Eva by setting $a_i = i$, $b_i = f(i)$.

Previous results

Definition (Index Lookup / Index)

Given a sequence $s_1, \ldots, s_N$ and a position $p \in [N]$, output s_p.

- **Bhattamishra et al. (2024)**:

Previous results

Definition (Index Lookup / Index)

Given a sequence $s_1, \ldots, s_N$ and a position $p \in [N]$, output s_p.

- **Bhattamishra et al. (2024)**:
  - Index lookup: 1-layer Transformer $>$ RNN.
  - Bounded Dyck languages Recognization: RNNs $>$ 1-layer Transformers.
  - String equality / disjointness: 2-layer Transformers $>$ 1-layer Transformers / RNNs.
  - Nearest neighbor algorithm: 2-layer Transformer $>$ RNN.

Previous results

Definition (Index Lookup / Index)

Given a sequence $s_1, \ldots, s_N$ and a position $p \in [N]$, output s_p.

- **Bhattamishra et al. (2024)**:
  - Index lookup: 1-layer Transformer $>$ RNN.
  - Bounded Dyck languages Recognization: RNNs $>$ 1-layer Transformers.
  - String equality / disjointness: 2-layer Transformers $>$ 1-layer Transformers / RNNs.
  - Nearest neighbor algorithm: 2-layer Transformer $>$ RNN.
- **Wen et al. (2025)**:
  - RNN $<$ RNN + CoT $<$ Transformers (associative recall, index, c-gram retrieval, counting, or IsTree).
  - Key bottleneck: RNNs lack *in-context retrieval* capability.
  - Enhancing retrieval (RAG / adding one Transformer layer) closes the gap, allowing RNNs to simulate poly.-time Turing machines with CoT.

Log-Linear Attention Lower Bounds

- Log-linear attention has $O(\log n)$ hiden states, so total hidden state size $O(Hd^2 p \log n)$.

Log-Linear Attention Lower Bounds

- Log-linear attention has $O(\log n)$ hiden states, so total hidden state size $O(Hd^2 p \log n)$.
- For Eva task:

$$LHd^2 p \log n < \Theta(n \log n) \iff LHd^2 p < \Theta(n) \implies \text{cannot solve.}$$

Log-Linear Attention Lower Bounds

- Log-linear attention has $O(\log n)$ hiden states, so total hidden state size $O(Hd^2p \log n)$.
- For Eva task:

$$LHd^2p \log n < \Theta(n \log n) \iff LHd^2p < \Theta(n) \implies \text{cannot solve.}$$

Architecture	Complexity
Full Attention (1-layer)	$Hdp = O(\text{poly} \log n)$
Linear Attention (L layers)	$LHd^2p = \Omega(n \log n)$
Log-Linear Attention (L layers)	$LHd^2p = \Omega(n)$

L-Sequential Function Composition Task

Definition (L-Sequential Function Composition)

Given functions $z_0 \in [m]$, $z_\ell : [N_{\ell-1}] \to [N_{\ell-1}]$ $(\ell = 1, \ldots, L)$ and a query $w = (w_1, \ldots, w_{L-1})$, Compute and output i_L:

$$i_0 = z_0,\ i_1 = z_1(i_0),\ i_2 = z_2(w_1, i_1),\ \ldots,\ i_L = z_L(w_{L-1}, i_{L-1}).$$

L-Sequential Function Composition Task

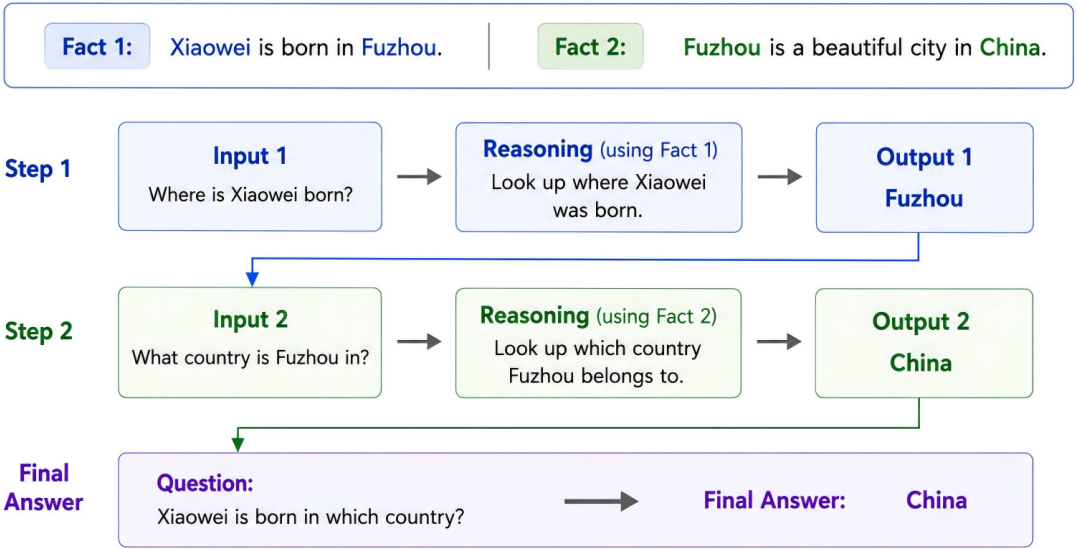
Definition (L-Sequential Function Composition)

Given functions $z_0 \in [m]$, $z_\ell : [N_{\ell-1}] \to [N_{\ell-1}]$ $(\ell = 1, \ldots, L)$ and a query $w = (w_1, \ldots, w_{L-1})$, Compute and output i_L:

$$i_0 = z_0,\ i_1 = z_1(i_0),\ i_2 = z_2(w_1, i_1),\ \ldots,\ i_L = z_L(w_{L-1}, i_{L-1}).$$

This task captures the essence of real tasks such as multi-step reasoning.

Multi-step Reasoning



Full Attention (Chen et al. 2025)

Model	Complexity
L-layer full attention	$\Omega(\text{poly}\, n)$
$(L+1)$-layer full attention	$O(\text{poly}\log n)$

Full Attention (Chen et al. 2025)

Model	Complexity
L-layer full attention	$\Omega(\text{poly}\, n)$
$(L+1)$-layer full attention	$O(\text{poly}\log n)$

Method: Autoregressive Communication Model
Key Concept: indistinguishable decomposition

Autoregressive Communication Model

Setup:

- $L + 2$ players: $-1, 0, 1, \ldots, L$ (arranged on a line).
- Player ℓ receives input z_ℓ ($\ell \geq 0$) or query w (player -1).
- Communication proceeds in L **epochs** (one per layer).

Autoregressive Communication Model

Setup:

- $L+2$ players: $-1, 0, 1, \ldots, L$ (arranged on a line).
- Player ℓ receives input z_ℓ ($\ell \geq 0$) or query w (player -1).
- Communication proceeds in L **epochs** (one per layer).

Epoch ℓ communication:

1. Player i sends its entire information $X_i^{(\ell-1)}$ to all players $j > i$.
2. Player j sends back a message $\Pi_{j,i}^{(\ell)}$ of size $2Hdp \cdot |z_i|$ to player i.
3. Player i updates $X_i^{(\ell)} = X_i^{(\ell-1)} \cup \bigcup_{j>i} \Pi_{j,i}^{(\ell)}$.

Autoregressive Communication Model

Setup:

- $L+2$ players: $-1, 0, 1, \ldots, L$ (arranged on a line).
- Player ℓ receives input z_ℓ ($\ell \geq 0$) or query w (player -1).
- Communication proceeds in L **epochs** (one per layer).

Epoch ℓ communication:

1. Player i sends its entire information $X_i^{(\ell-1)}$ to all players $j > i$.
2. Player j sends back a message $\Pi_{j,i}^{(\ell)}$ of size $2Hdp \cdot |z_i|$ to player i.
3. Player i updates $X_i^{(\ell)} = X_i^{(\ell-1)} \cup \bigcup_{j>i} \Pi_{j,i}^{(\ell)}$.

After L epochs, player -1 outputs answer.

Indistinguishable Decomposition

Definition (Indistinguishable Decomposition)

For $\ell \in [2 : L]$, a pair $(R_{\geq \ell}, Z_{<\ell})$ where: $R_{\geq \ell}$ are inputs to players $\ell, \ldots, L$, and $Z_{<\ell} = Z_{-1} \times \cdots \times Z_{\ell-1}$ are inputs to players $-1, \cdots, \ell - 1$.

Such that for every $z_{<\ell} \in Z_{<\ell}$ and any $\alpha_{\geq \ell}, \beta_{\geq \ell} \in R_{\geq \ell}$, all transcripts $\Pi_{j,i}^{(\ell')}$ (for $j \geq \ell$, $i < \ell$, $\ell' \leq \ell$) are identical.

Indistinguishable Decomposition

Definition (Indistinguishable Decomposition)

For $\ell \in [2 : L]$, a pair $(R_{\geq \ell}, Z_{<\ell})$ where: $R_{\geq \ell}$ are inputs to players $\ell, \ldots, L$, and $Z_{<\ell} = Z_{-1} \times \cdots \times Z_{\ell-1}$ are inputs to players $-1, \cdots, \ell - 1$.

Such that for every $z_{<\ell} \in Z_{<\ell}$ and any $\alpha_{\geq \ell}, \beta_{\geq \ell} \in R_{\geq \ell}$, all transcripts $\Pi_{j,i}^{(\ell')}$ (for $j \geq \ell$, $i < \ell$, $\ell' \leq \ell$) are identical.

When $\ell = L$, player -1 sees the same transcript for all $z_L \in R_{\geq L}$ given any $z_{<L} \in Z_{<L}$. Thus the output must be the same for all $z_L \in R_{\geq L}$.

Indistinguishable Decomposition

Definition (Indistinguishable Decomposition)

For $\ell \in [2 : L]$, a pair $(R_{\geq \ell}, Z_{<\ell})$ where: $R_{\geq \ell}$ are inputs to players $\ell, \ldots, L$, and $Z_{<\ell} = Z_{-1} \times \cdots \times Z_{\ell-1}$ are inputs to players $-1, \cdots, \ell - 1$.
Such that for every $z_{<\ell} \in Z_{<\ell}$ and any $\alpha_{\geq \ell}, \beta_{\geq \ell} \in R_{\geq \ell}$, all transcripts $\Pi_{j,i}^{(\ell')}$ (for $j \geq \ell$, $i < \ell$, $\ell' \leq \ell$) are identical.

When $\ell = L$, player -1 sees the same transcript for all $z_L \in R_{\geq L}$ given any $z_{<L} \in Z_{<L}$. Thus the output must be the same for all $z_L \in R_{\geq L}$.
To prove impossibility: construct $R_{\geq L}, Z_{<L}$ such that:

1. $|R_{\geq L}|$ is large (many distinct z_L).
2. $|\mathcal{I}_{L-1}(Z_{<L})|$ is large (many possible intermediate values i_{L-1}).

Indistinguishable Decomposition

Definition (Indistinguishable Decomposition)

For $\ell \in [2 : L]$, a pair $(R_{\geq \ell}, Z_{<\ell})$ where: $R_{\geq \ell}$ are inputs to players $\ell, \ldots, L$, and $Z_{<\ell} = Z_{-1} \times \cdots \times Z_{\ell-1}$ are inputs to players $-1, \cdots, \ell - 1$.

Such that for every $z_{<\ell} \in Z_{<\ell}$ and any $\alpha_{\geq \ell}, \beta_{\geq \ell} \in R_{\geq \ell}$, all transcripts $\Pi_{j,i}^{(\ell')}$ (for $j \geq \ell$, $i < \ell$, $\ell' \leq \ell$) are identical.

When $\ell = L$, player -1 sees the same transcript for all $z_L \in R_{\geq L}$ given any $z_{<L} \in Z_{<L}$. Thus the output must be the same for all $z_L \in R_{\geq L}$.

To prove impossibility: construct $R_{\geq L}, Z_{<L}$ such that:

1. $|R_{\geq L}|$ is large (many distinct z_L).
2. $|\mathcal{I}_{L-1}(Z_{<L})|$ is large (many possible intermediate values i_{L-1}).

Different z_L and i_{L-1} would require different outputs $\Rightarrow$ contradiction.

Inductive Construction of Indistinguishable Decomposition

Goal: Build $(R_{\geq \ell}, Z_{<\ell})$ for $\ell = 2, \ldots, L$ satisfying size bounds.

Inductive Construction of Indistinguishable Decomposition

Goal: Build $(R_{\geq \ell}, Z_{<\ell})$ for $\ell = 2, \ldots, L$ satisfying size bounds.
Base case $\ell = 2$:

1. Choose $Z_0 = [x_0]$.
2. Pick Z_1 via pigeonhole on first-epoch messages from player 1 to -1.
3. Greedily fix transcripts to players $-1, 0, 1$ for first two epochs, ensuring consistency and large remaining sets.
4. Result: large $R_{\geq 2}$ and $\mathcal{I}_1(Z_{<2})$.

Inductive Construction of Indistinguishable Decomposition

Goal: Build $(R_{\geq \ell}, Z_{<\ell})$ for $\ell = 2, \ldots, L$ satisfying size bounds.
Base case $\ell = 2$:

1. Choose $Z_0 = [x_0]$.
2. Pick Z_1 via pigeonhole on first-epoch messages from player 1 to -1.
3. Greedily fix transcripts to players $-1, 0, 1$ for first two epochs, ensuring consistency and large remaining sets.
4. Result: large $R_{\geq 2}$ and $\mathcal{I}_1(Z_{<2})$.

Inductive step ($\ell \to \ell + 1$):

1. Extract rectangular subset $S_{\geq \ell+1} \times Z_\ell \subseteq R_{\geq \ell}$ with $S_{\geq \ell+1}$, $\mathcal{I}(Z_\ell)$ large.
2. Use that Z_ℓ is indistinguishable to players $[-1 : \ell - 1]$ after ℓ epochs.
3. Fix transcripts to $[-1 : \ell - 1]$ at epoch $\ell + 1$ via pigeonhole (independent of z_ℓ).
4. Fix transcripts to player ℓ via greedy selection.

Key Insight for Inductive Step

Why can we fix transcripts to $[-1 : \ell - 1]$ without knowing z_ℓ?

Key Insight for Inductive Step

Why can we fix transcripts to $[-1 : \ell - 1]$ without knowing z_ℓ?

After ℓ epochs, all $z_\ell \in Z_\ell$ produce the **same information state** $X_i^{(\ell)}$ for every player $i \leq \ell - 1$.

Key Insight for Inductive Step

Why can we fix transcripts to $[-1 : \ell - 1]$ without knowing z_ℓ?

After ℓ epochs, all $z_\ell \in Z_\ell$ produce the **same information state** $X_i^{(\ell)}$ for every player $i \leq \ell - 1$.

- This is because Z_ℓ was chosen from $R_{\geq \ell}$ where all z_ℓ yield identical transcripts to lower players.

Key Insight for Inductive Step

Why can we fix transcripts to $[-1 : \ell - 1]$ without knowing z_ℓ?

After ℓ epochs, all $z_\ell \in Z_\ell$ produce the **same information state** $X_i^{(\ell)}$ for every player $i \leq \ell - 1$.

- This is because Z_ℓ was chosen from $R_{\geq \ell}$ where all z_ℓ yield identical transcripts to lower players.
- Therefore, the $(\ell + 1)$-th epoch messages from players $> \ell$ to players $< \ell$ depend only on $z_{<\ell}$, not on the specific z_ℓ.

Key Insight for Inductive Step

Why can we fix transcripts to $[-1 : \ell - 1]$ without knowing z_ℓ?

After ℓ epochs, all $z_\ell \in Z_\ell$ produce the **same information state** $X_i^{(\ell)}$ for every player $i \leq \ell - 1$.

- This is because Z_ℓ was chosen from $R_{\geq \ell}$ where all z_ℓ yield identical transcripts to lower players.
- Therefore, the $(\ell + 1)$-th epoch messages from players $> \ell$ to players $< \ell$ depend only on $z_{<\ell}$, not on the specific z_ℓ.

Hence we can select transcripts that maximize the remaining entropy in $S_{\geq \ell+1}$ without consulting z_ℓ.

Remark: Implications of Chen et al. 2025

1. **Depth-Size Tradeoff**: There exists a task (L-SeqComp) where an $(L+1)$-layer Transformer needs only $\operatorname{poly}\log n$ parameters, but any L-layer Transformer requires $n^{\Omega(1)}$ parameters.

Remark: Implications of Chen et al. 2025

1. **Depth-Size Tradeoff**: There exists a task (L-SeqComp) where an $(L+1)$-layer Transformer needs only $\operatorname{poly}\log n$ parameters, but any L-layer Transformer requires $n^{\Omega(1)}$ parameters.
2. **Encoder v.s. Decoder**:
   - The same task solved by an $O(\log L)$-layer encoder with $\operatorname{poly}\log n$ parameters, while any L-layer decoder requires polynomial size.
   - First **unconditional** separation between encoder and decoder.

Remark: Implications of Chen et al. 2025

1. **Depth-Size Tradeoff**: There exists a task (L-SeqComp) where an $(L+1)$-layer Transformer needs only $\operatorname{poly}\log n$ parameters, but any L-layer Transformer requires $n^{\Omega(1)}$ parameters.
2. **Encoder v.s. Decoder**:
   - The same task solved by an $O(\log L)$-layer encoder with $\operatorname{poly}\log n$ parameters, while any L-layer decoder requires polynomial size.
   - First **unconditional** separation between encoder and decoder.
3. **Provable Benefits of Chain-of-Thought**:
   - L-SeqComp solved by 1-layer Transformer with L steps CoT, $\operatorname{poly}\log n$ size, but any L-layer Transformer (no CoT) needs polynomial size.
   - First unconditional proof that CoT strictly increases expressive power.

Lower Bound for Hybrid Attention

Theorem

Any hybrid architecture with $L-1$ full attention layers followed by exponentially many linear attention layers cannot solve L-sequential function composition whenever Hdp is sub-polynomial.

Lower Bound for Hybrid Attention

Theorem

Any hybrid architecture with $L-1$ full attention layers followed by exponentially many linear attention layers cannot solve L-sequential function composition whenever Hdp is sub-polynomial.

Even an exponential number of linear attention layers cannot compensate for missing full attention layers.

Lower Bound for Hybrid Attention

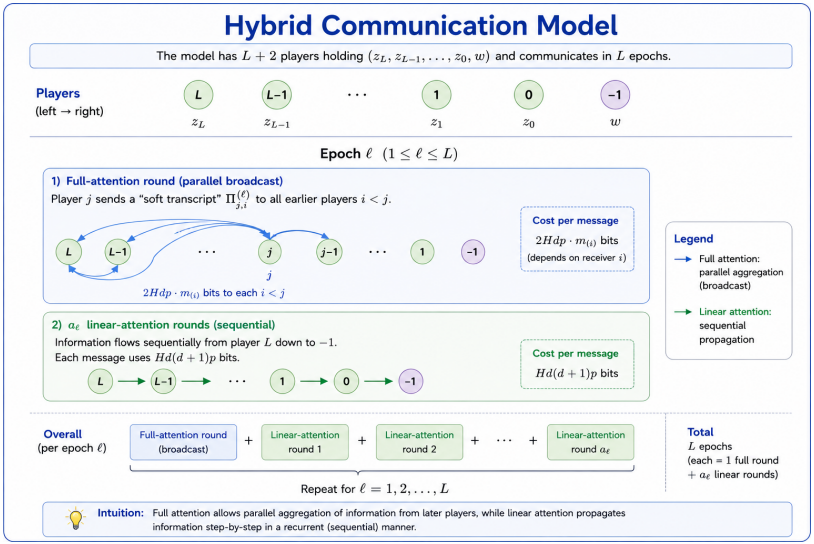
Theorem

Any hybrid architecture with $L-1$ full attention layers followed by exponentially many linear attention layers cannot solve L-sequential function composition whenever Hdp is sub-polynomial.

Even an exponential number of linear attention layers cannot compensate for missing full attention layers.

Model	Complexity
Full, L layers	$\Omega(\text{poly } n)$
Full, $L+1$ layers	$O(\text{poly} \log n)$
Hybrid, $L-1$ full layers	$\Omega(\text{poly } n)$

Modified Communication Model

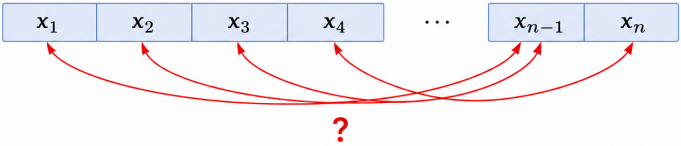


2-Sum Task

Given a sequence of n numbers, determine whether there exist two positions $i \neq j$ such that $x_i + x_j = 0$.

This captures the need for pairwise comparisons across the entire sequence.

Input sequence



Example ($n = 6$)

3	3	0	2	0	5

$0 + 0 = 0 \rightarrow$ **Yes**

Goal: Decide whether any pair (i, j) satisfies $x_i + x_j = 0$.

Lower Bound for Sparse Attention

Theorem (Sanford et al. 2023)

A single-layer full attention ($H = 1, d = 3, p = O(\log n)$) can solve 2-Sum.

Lower Bound for Sparse Attention

Theorem (Sanford et al. 2023)

A single-layer full attention ($H = 1, d = 3, p = O(\log n)$) can solve 2-Sum.

Theorem (Ye et al. 2026)

Any single-layer (B, k)-sparse attention solving 2-Sum must satisfy $Hdp = \Omega(B \log n)$.

Lower Bound for Sparse Attention

Theorem (Sanford et al. 2023)

A single-layer full attention ($H = 1, d = 3, p = O(\log n)$) can solve 2-Sum.

Theorem (Ye et al. 2026)

Any single-layer (B, k)-sparse attention solving 2-Sum must satisfy $Hdp = \Omega(B \log n)$.

Model	**Complexity**
Full	$O(\log n)$
Sparse (Block)	$\Omega(B \log n)$

Full Attention Solves 2-Sum: Trigonometric Construction

Setup: Input $x_1, \ldots, x_{n+1} \in [M]$ with $M = O(n)$. Append a blank token.

Full Attention Solves 2-Sum: Trigonometric Construction

Setup: Input $x_1, \ldots, x_{n+1} \in [M]$ with $M = O(n)$. Append a blank token.
Positional Encoding: Use sinusoidal functions for each coordinate:

$$\phi(x_i) = \left(\cos \frac{2\pi x_i}{M},\ \sin \frac{2\pi x_i}{M},\ 1 \right) \in \mathbb{R}^3.$$

Full Attention Solves 2-Sum: Trigonometric Construction

Setup: Input $x_1, \ldots, x_{n+1} \in [M]$ with $M = O(n)$. Append a blank token.
Positional Encoding: Use sinusoidal functions for each coordinate:

$$\phi(x_i) = \left(\cos \frac{2\pi x_i}{M},\ \sin \frac{2\pi x_i}{M},\ 1\right) \in \mathbb{R}^3.$$

Attention Parameters: $V\phi(x_j) = 1$.
$Q\phi(x_i) = c\left(\cos \frac{2\pi x_i}{M},\ \sin \frac{2\pi x_i}{M},\ 1\right), \quad K\phi(x_j) = \left(\cos \frac{2\pi x_j}{M},\ -\sin \frac{2\pi x_j}{M},\ 0\right).$

Full Attention Solves 2-Sum: Trigonometric Construction

Setup: Input $x_1, \ldots, x_{n+1} \in [M]$ with $M = O(n)$. Append a blank token.
Positional Encoding: Use sinusoidal functions for each coordinate:

$$\phi(x_i) = \left(\cos \frac{2\pi x_i}{M},\ \sin \frac{2\pi x_i}{M},\ 1\right) \in \mathbb{R}^3.$$

Attention Parameters: $V\phi(x_j) = 1$.
$Q\phi(x_i) = c\left(\cos \frac{2\pi x_i}{M},\ \sin \frac{2\pi x_i}{M},\ 1\right), \quad K\phi(x_j) = \left(\cos \frac{2\pi x_j}{M},\ -\sin \frac{2\pi x_j}{M},\ 0\right).$

$$\langle Q\phi(x_i), K\phi(x_j)\rangle = c\cos\left(\frac{2\pi(x_i + x_j)}{M}\right).$$

This equals c iff $x_i + x_j \equiv 0 \pmod{M}$, otherwise $\leq c(1 - 1/M^2)$.

Full Attention Solves 2-Sum: Trigonometric Construction

Setup: Input $x_1, \ldots, x_{n+1} \in [M]$ with $M = O(n)$. Append a blank token.
Positional Encoding: Use sinusoidal functions for each coordinate:

$$\phi(x_i) = \left(\cos \frac{2\pi x_i}{M},\ \sin \frac{2\pi x_i}{M},\ 1\right) \in \mathbb{R}^3.$$

Attention Parameters: $V\phi(x_j) = 1$.
$Q\phi(x_i) = c\left(\cos \frac{2\pi x_i}{M},\ \sin \frac{2\pi x_i}{M},\ 1\right), \quad K\phi(x_j) = \left(\cos \frac{2\pi x_j}{M},\ -\sin \frac{2\pi x_j}{M},\ 0\right).$

$$\langle Q\phi(x_i), K\phi(x_j)\rangle = c\cos\left(\frac{2\pi(x_i + x_j)}{M}\right).$$

This equals c iff $x_i + x_j \equiv 0 \pmod{M}$, otherwise $\leq c(1 - 1/M^2)$.
For $c = \Theta(M^2 \log N)$, the attention concentrates on correct solution.

Other Results of Sanford et al. 2023

- **Sparse Averaging (qSA)**:
  - Input: (y_i, z_i) with $y_i \in \binom{[N]}{q}$, $z_i \in \mathbb{R}^{d'}$, output average of z_j for $j \in y_i$.
  - A single self-attention unit with embedding dimension $m \gtrsim q$ approximates qSA.
  - RNNs require hidden state $\Omega(N)$ bits.

Other Results of Sanford et al. 2023

- **Sparse Averaging (qSA)**:
  - Input: (y_i, z_i) with $y_i \in \binom{[N]}{q}$, $z_i \in \mathbb{R}^{d'}$, output average of z_j for $j \in y_i$.
  - A single self-attention unit with embedding dimension $m \gtrsim q$ approximates qSA.
  - RNNs require hidden state $\Omega(N)$ bits.
- **Triple Detection**: 3-Sum **cannot** be solved by a single multi-head attention layer unless $mpH = \Omega(N/\log\log N)$.
  Conjecture: multi-layer transformers cannot solve 3-Sum efficiently.

Lower Bound: Sparse Attention Cannot Solve 2-Sum

Communication Model ($\frac{n}{B} + 1$ players):

- Each of the first $\frac{n}{B}$ players receives B tokens.
- The last player receives a single token (query).
- Each block player sends Hdp-bit message to the last player.
- The last player then select k players and access their full information.

Lower Bound: Sparse Attention Cannot Solve 2-Sum

Communication Model ($\frac{n}{B} + 1$ players):

- Each of the first $\frac{n}{B}$ players receives B tokens.
- The last player receives a single token (query).
- Each block player sends Hdp-bit message to the last player.
- The last player then select k players and access their full information.

Key Insight: If $2^{Hdp} < \binom{n}{B}$, then two distinct B-tuples $\{a_i\}$ and $\{b_i\}$ yield the same compressed message. Choose a value $v \in \{a_i\} \setminus \{b_i\}$ and set query $x_{n+1} = -v \pmod{n}$.

Lower Bound: Sparse Attention Cannot Solve 2-Sum

Communication Model ($\frac{n}{B} + 1$ players):

- Each of the first $\frac{n}{B}$ players receives B tokens.
- The last player receives a single token (query).
- Each block player sends Hdp-bit message to the last player.
- The last player then select k players and access their full information.

Key Insight: If $2^{Hdp} < \binom{n}{B}$, then two distinct B-tuples $\{a_i\}$ and $\{b_i\}$ yield the same compressed message. Choose a value $v \in \{a_i\} \setminus \{b_i\}$ and set query $x_{n+1} = -v \pmod{n}$. Then:

- For input with $\{a_i\}$: $\exists$ match $\Rightarrow$ output 1.
- For input with $\{b_i\}$: no match $\Rightarrow$ output 0.

Lower Bound: Sparse Attention Cannot Solve 2-Sum

Communication Model ($\frac{n}{B} + 1$ players):

- Each of the first $\frac{n}{B}$ players receives B tokens.
- The last player receives a single token (query).
- Each block player sends Hdp-bit message to the last player.
- The last player then select k players and access their full information.

Key Insight: If $2^{Hdp} < \binom{n}{B}$, then two distinct B-tuples $\{a_i\}$ and $\{b_i\}$ yield the same compressed message. Choose a value $v \in \{a_i\} \setminus \{b_i\}$ and set query $x_{n+1} = -v \pmod{n}$. Then:

- For input with $\{a_i\}$: $\exists$ match $\Rightarrow$ output 1.
- For input with $\{b_i\}$: no match $\Rightarrow$ output 0.

But : identical compressed messages $\implies$ identical selected block

Lower Bound: Sparse Attention Cannot Solve 2-Sum

Communication Model ($\frac{n}{B} + 1$ players):

- Each of the first $\frac{n}{B}$ players receives B tokens.
- The last player receives a single token (query).
- Each block player sends Hdp-bit message to the last player.
- The last player then select k players and access their full information.

Key Insight: If $2^{Hdp} < \binom{n}{B}$, then two distinct B-tuples $\{a_i\}$ and $\{b_i\}$ yield the same compressed message. Choose a value $v \in \{a_i\} \setminus \{b_i\}$ and set query $x_{n+1} = -v \pmod{n}$. Then:

- For input with $\{a_i\}$: $\exists$ match $\Rightarrow$ output 1.
- For input with $\{b_i\}$: no match $\Rightarrow$ output 0.

But : identical compressed messages $\implies$ identical selected block
If selected blocks are $\{b_i\}$, then cannot distinguish remainning blocks $\implies$ contradiction.

Summary and Open Problems

- Expressiveness hierarchy: full attention $>$ hybrid, linear/log-linear, sparse patterns. (Complexity-Expressiveness Trade-offs)

Summary and Open Problems

- Expressiveness hierarchy: full attention $>$ hybrid, linear/log-linear, sparse patterns. (Complexity-Expressiveness Trade-offs)
- Method: constructed tasks + communication complexity.

Summary and Open Problems

- Expressiveness hierarchy: full attention $>$ hybrid, linear/log-linear, sparse patterns. (Complexity-Expressiveness Trade-offs)
- Method: constructed tasks + communication complexity.
- **Limitations**:
  - Analysis is for layer-wise hybridization, not head-wise.
  - Sparse attention lower bound applies to block compression+selection, not learnable sparse patterns (e.g., DSA).
  - Tasks are synthetic; practical implications require empirical verification.

Summary and Open Problems

- Expressiveness hierarchy: full attention $>$ hybrid, linear/log-linear, sparse patterns. (Complexity-Expressiveness Trade-offs)
- Method: constructed tasks + communication complexity.
- **Limitations**:
  - Analysis is for layer-wise hybridization, not head-wise.
  - Sparse attention lower bound applies to block compression+selection, not learnable sparse patterns (e.g., DSA).
  - Tasks are synthetic; practical implications require empirical verification.
- **Open Problem**: Can head-wise hybrid attention overcome the current lower bounds? Can lower bounds be proven for multi-layer sparse attention? $\cdots$

Question?

Question? Thank you!