How Transformers Learn Causal Structure with Gradient Descent

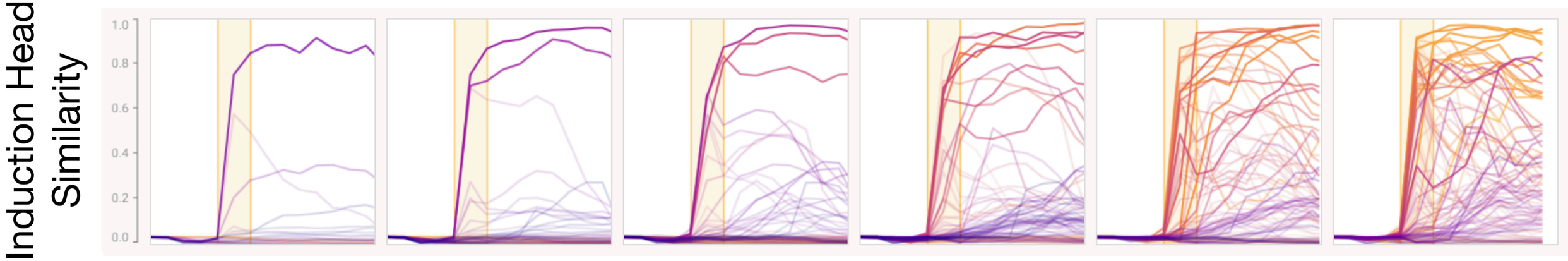
Eshaan Nichani, Alex Damian, Jason D. Lee

Outline

- Background & Motivation
- Problem Setup & Initial Investigation
- Main Theorem
- Proof Sketch
- Extensions

Emergence of In-Context Learning

- In-context learning ability emerges at depth 2
- This ability emerges at consistent times during training regardless of depth
- [Olsson et al. 2022] connected this to the emergence of ***induction heads***



Olsson et al. 2022, In-context Learning and Induction Heads

Induction Heads

Mr and Mrs Dursley, of number four, Privet Drive, were proud to say that they were perfectly normal, thank you very much. They were the last people you'd expect to be involved in anything strange or mysterious, because they just didn't hold with such nonsense. Mr Dursley was the director of a firm called Grunnings, which made drills. He was a big, beefy man with hardly any neck, although he did have a very large moustache. Mrs Durs**ley**

Given a prompt $[\ldots, A, B, \ldots, A, ?]$ the induction head:

1. Scans for previous occurrences of A: $[\ldots, A, B, \ldots, A, ?]$
2. Returns the next token: $[\ldots, A, B, \ldots, A, B]$

In-context Markov structure: use historical patterns of tokens following s_t to predict s_{t+1}

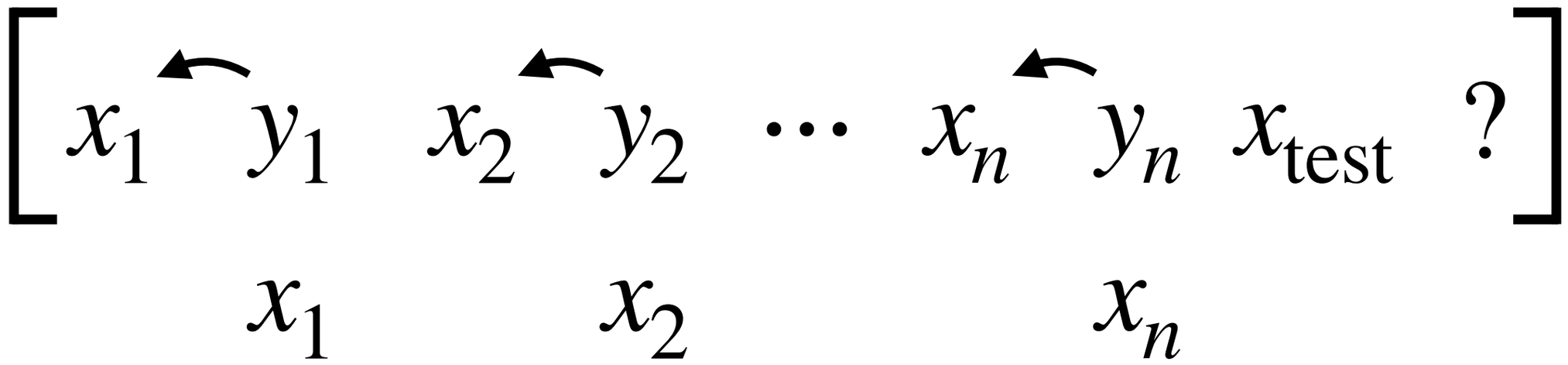
Olsson et al. 2022, In-context Learning and Induction Heads

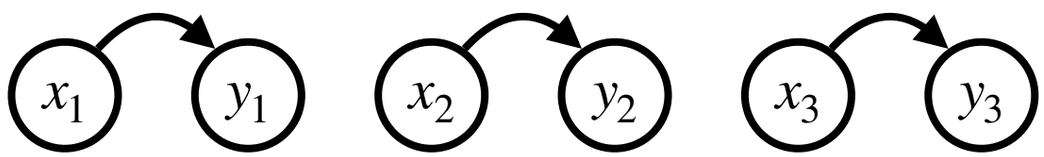
In-Context Learning Function Classes [Garg et al. 2022]

For each sequence:

(1) sample $(x_1, y_1), \ldots, (x_n, y_n), (x_{\text{test}}, y_{\text{test}})$ from a random learning problem

(2) predict y_{test} given $(x_1, y_1), \ldots, (x_n, y_n), x_{\text{test}}$

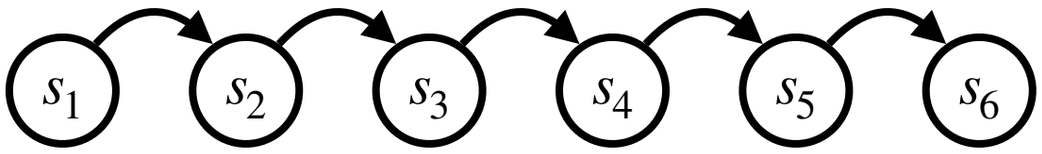




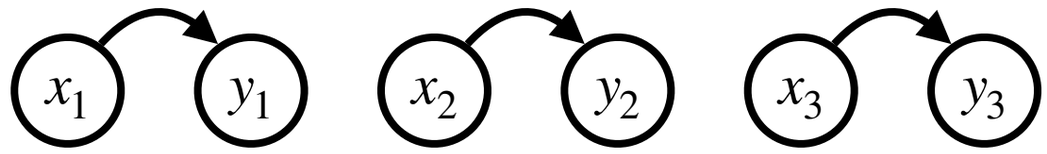
latent causal structure

Garg et al. 2022, What Can Transformers Learn In-Context? A Case Study of Simple Function Classes

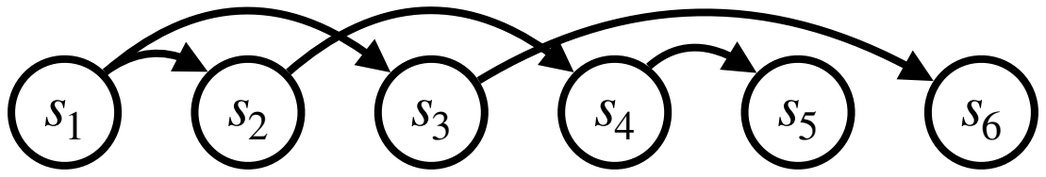
Sequences with Causal Structure



Markovian causal structure



in-context learning a function class



more complex causal structure

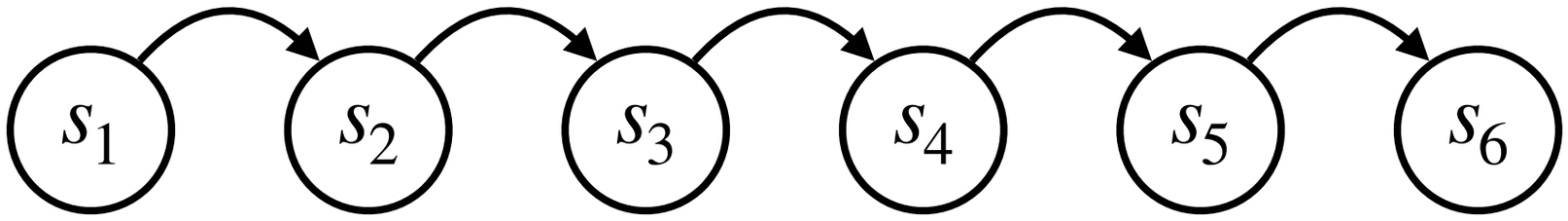
Motivating Question:

How do Transformers learn such causal structure from data?

Our Approach:

1. Construct a family of ICL tasks that require learning causal structure
2. Analyze the dynamics of gradient descent on a Transformer

The Simplest Task: In-Context Markov Chains



in-context Markov chain

To generate each sequence:

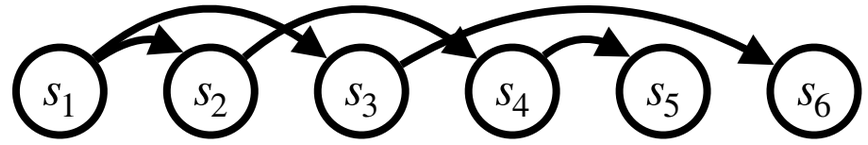
- Sample a transition kernel π from some prior (e.g. Dirichlet)
- Sample s_1 from its stationary measure
- For $i = 1, \ldots, T-1$: sample $s_{i+1} \sim \pi(\cdot \mid s_i)$

Natural Estimator: compute the empirical transition counts in-context

$$\hat{p}(s' \mid s) = \frac{\#s \to s' \text{ transitions in the sequence}}{\#s \text{ in the sequence}}$$

More Complex Causal Structures

Causal Graph:



complex causal structure

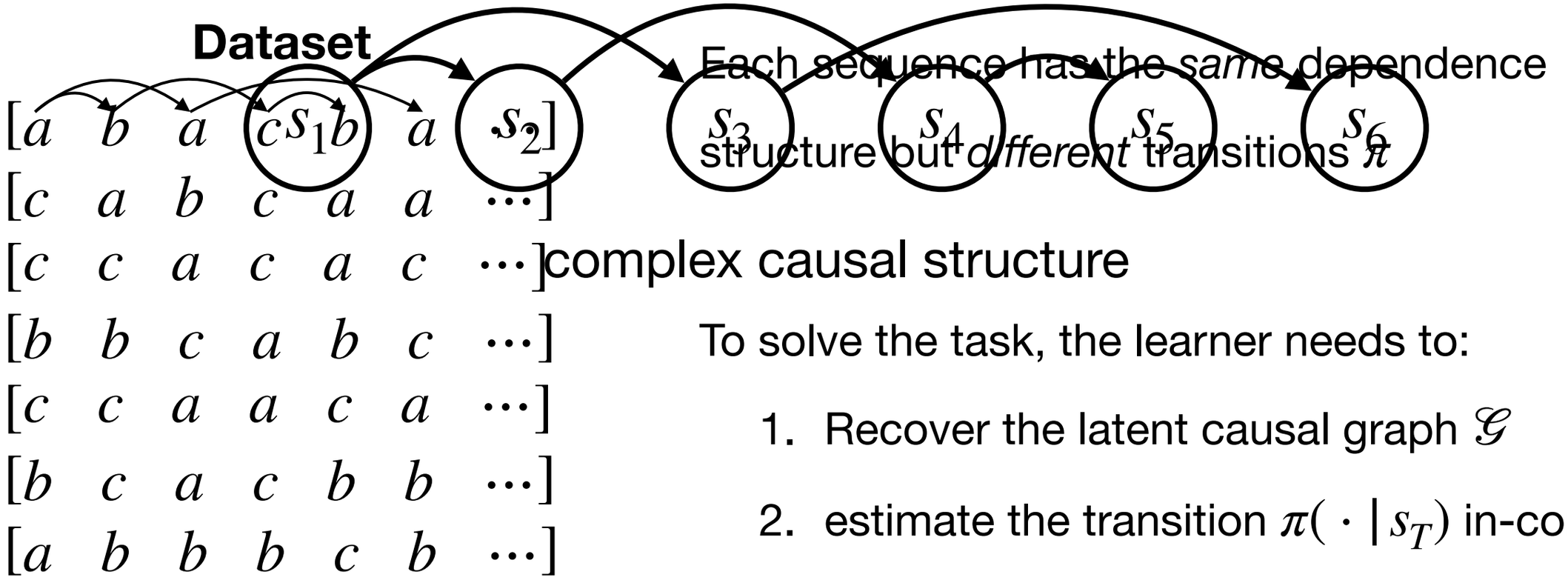
- $\mathcal{G}$ is a directed acyclic graph on $1, \ldots, T$
- Each position i has at most one parent $p(i) < i$

Sequence Generation:

- Draw $\pi \sim P_\pi$, a prior over transition matrices
- For each i, sample $s_i \sim \pi(\,\cdot \mid s_{p(i)})$. If $p(i) = \varnothing$, sample $s_i \sim \mu_\pi$
- Task: predict $s_{t+1} \sim \pi(\,\cdot \mid s_T)$ given $s_1, \ldots, s_T$

Sequences share the same dependence structure, but have different transitions π

More Complex Causal Structures



Each sequence has the *same* dependence structure but *different* transitions π

complex causal structure

To solve the task, the learner needs to:

1. Recover the latent causal graph $\mathcal{G}$
2. estimate the transition $\pi(\cdot \mid s_T)$ in-context

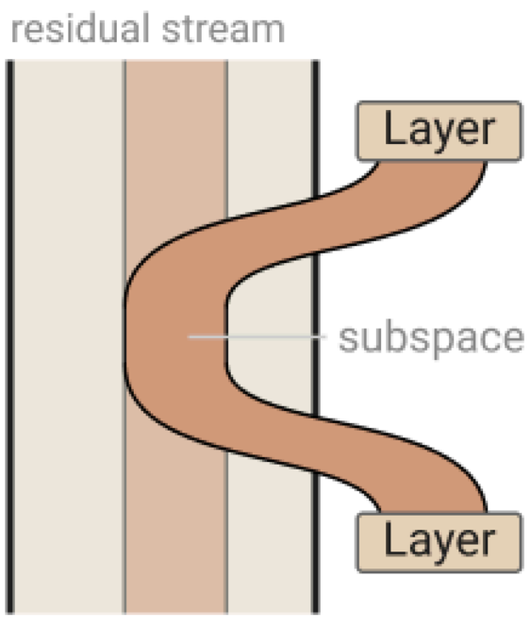
Given $\mathcal{G}$, can compute the empirical transition counts in-context

$$\hat{p}(s' \mid s) = \frac{\#s \to s' \text{ transitions in graph}}{\#\text{ parents equal to } s}$$

How do transformers recover $\mathcal{G}$ from the dataset?

Brief Detour: Residual streams

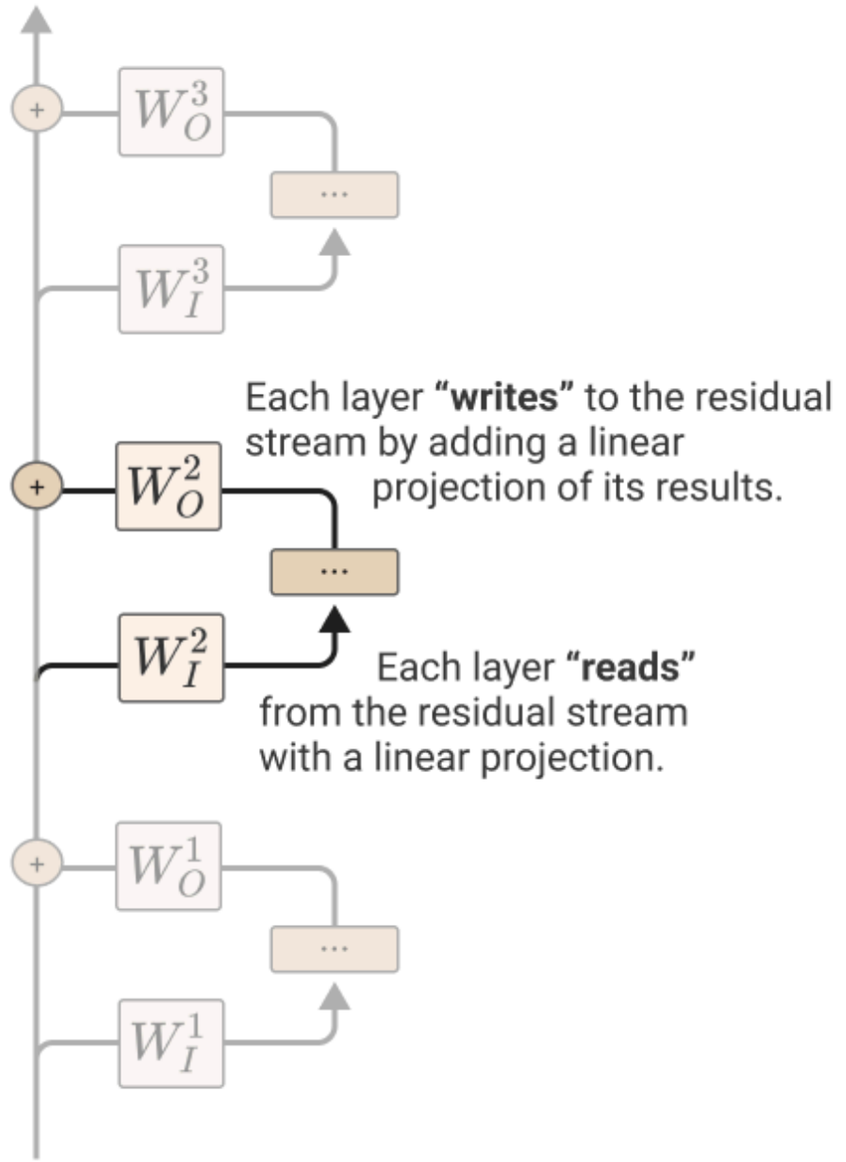
$$X \leftarrow X + \text{attn}(X)V$$
$$X \leftarrow X + \text{mlp}(X)W_O$$



The residual stream is high dimensional and can be divided into different subspaces

Challenge: the residual stream becomes "entangled" when these subspaces overlap

The residual stream is modified by a sequence of MLP and attention layers "reading from" and "writing to" it with linear operations.



$$\text{attn}(X) = PX \text{ where } P = \text{softmax}(\text{mask}(XQK^TX))$$

Olsson et al. 2022, A Mathematical Framework for Transformer Circuits

The Disentangled Transformer

1. Use one-hot token+positional embeddings
2. Replace linear projections with concatenation

$$x_i = [\underbrace{\text{onehot}(s_i)}_{\text{token}} | \underbrace{\text{onehot}(i)}_{\text{position}}]$$

For $i = 1, \ldots, L$:

$$x_i \leftarrow [x_i, \text{attn}(X)_i]$$

$$x_i \leftarrow [x_i, \text{mlp}(X)_i]$$

Return $W_O x_T$

Theorem:

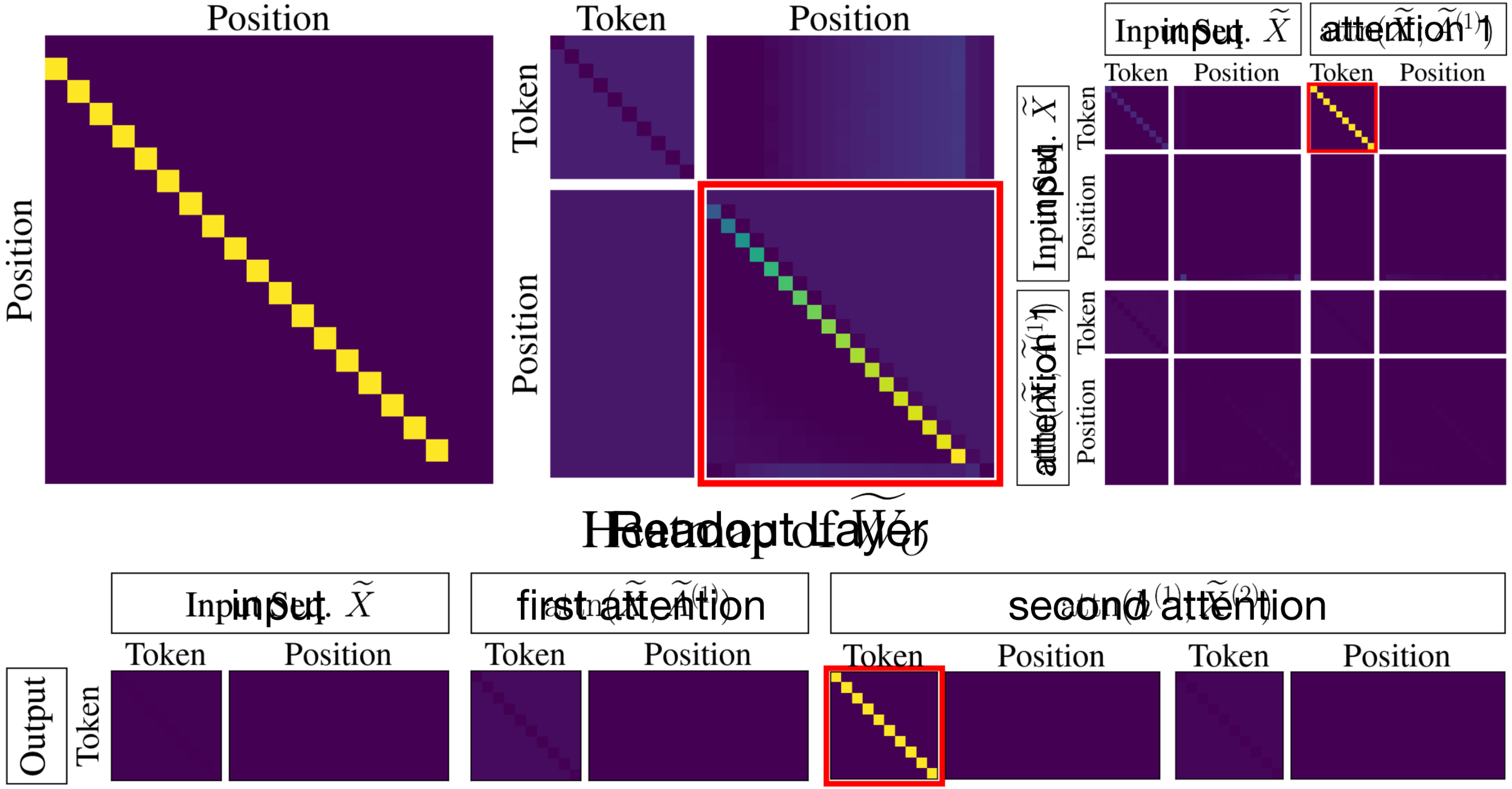
Transformers with H heads and L layers have the same expressive power as disentangled transformers with H heads and L layers.

Completely impractical: the embedding dimension **doubles** at every step

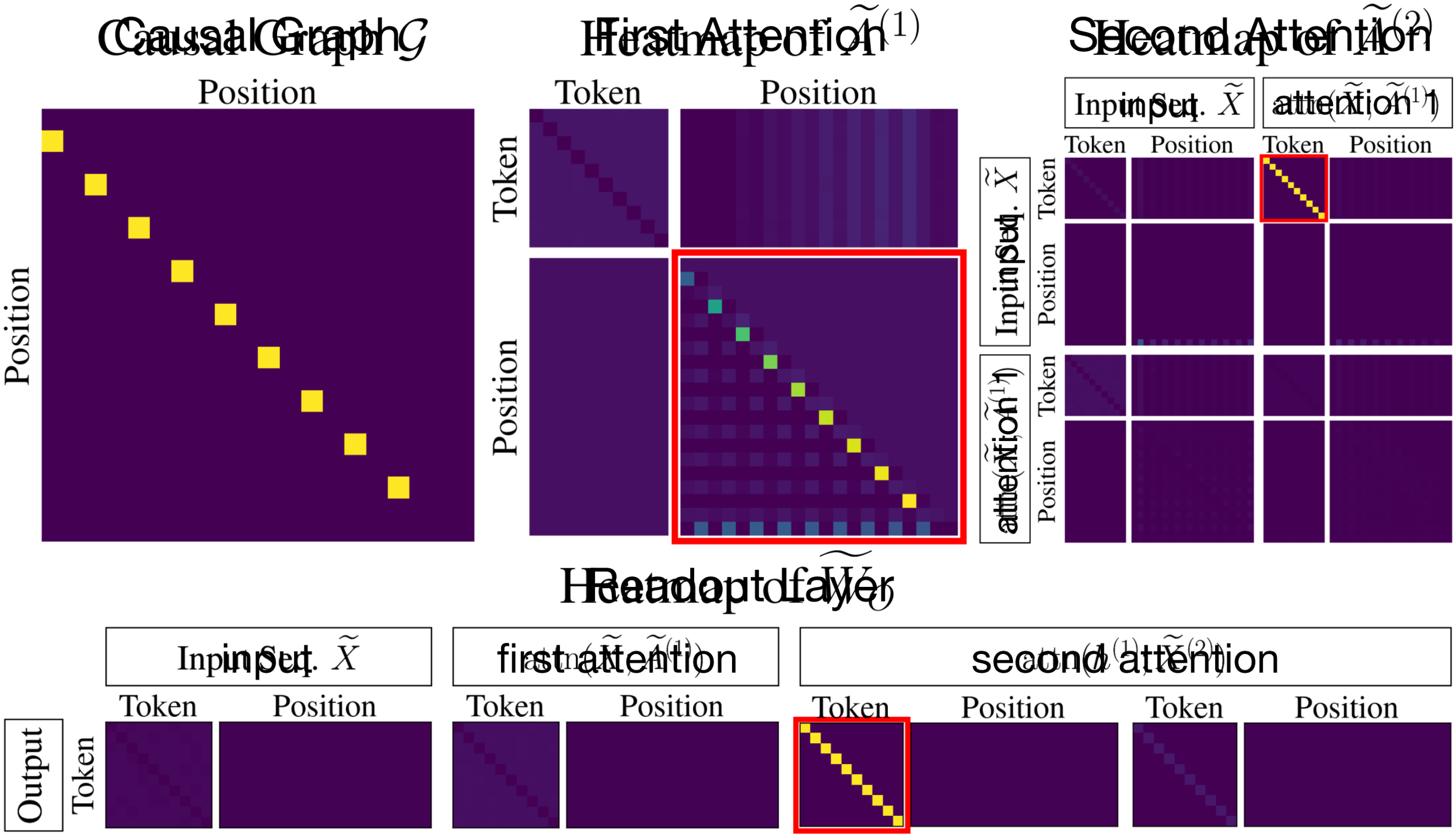
- weights are directly interpretable
- easier to reason about the flow of information through the model
- useful tool for theory and mechanistic interpretability

similar model considered in Friedman et al. 2023, Learning Transformer Programs

How do Transformers solve this task?

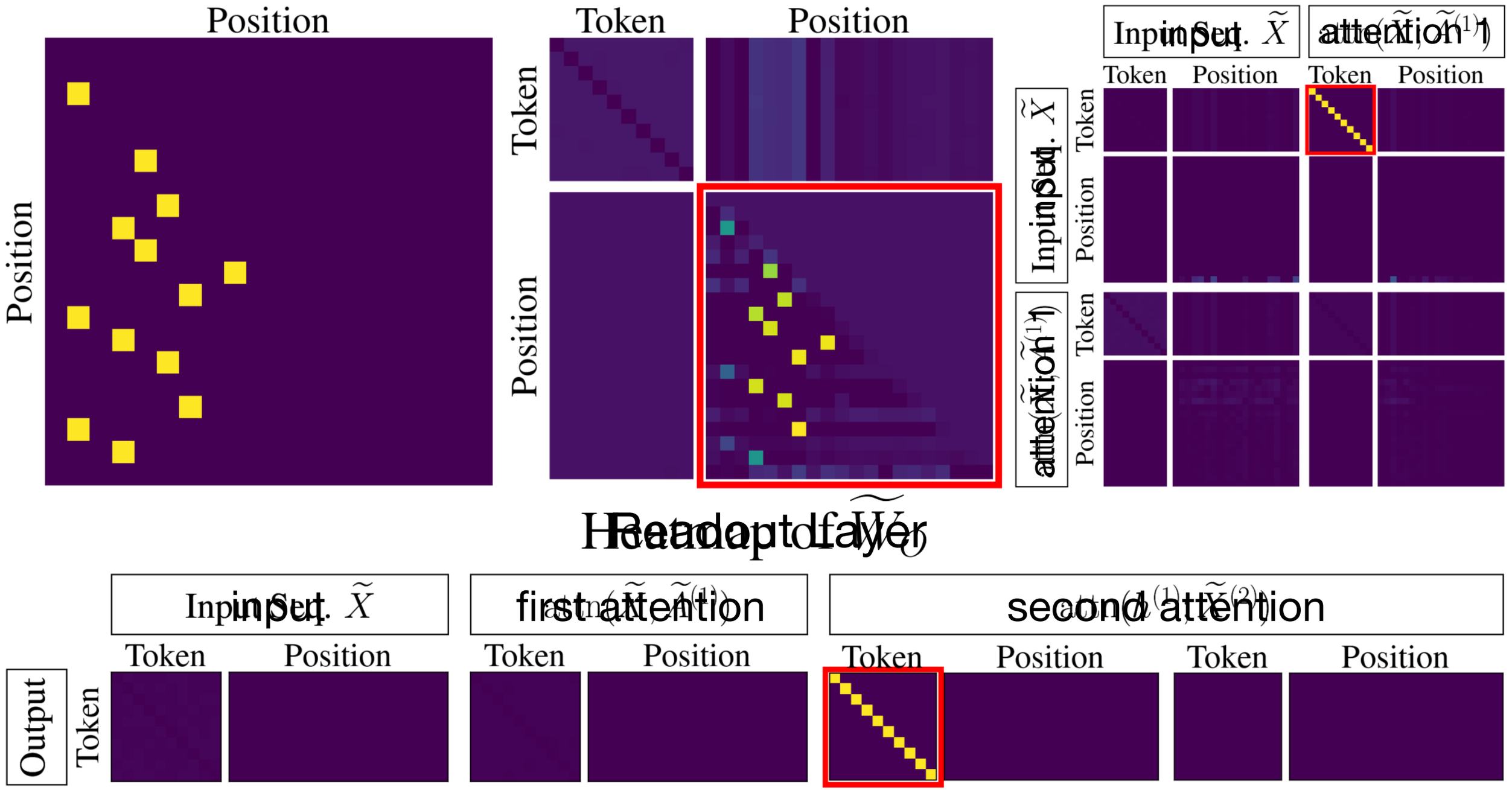


How do Transformers solve this task?



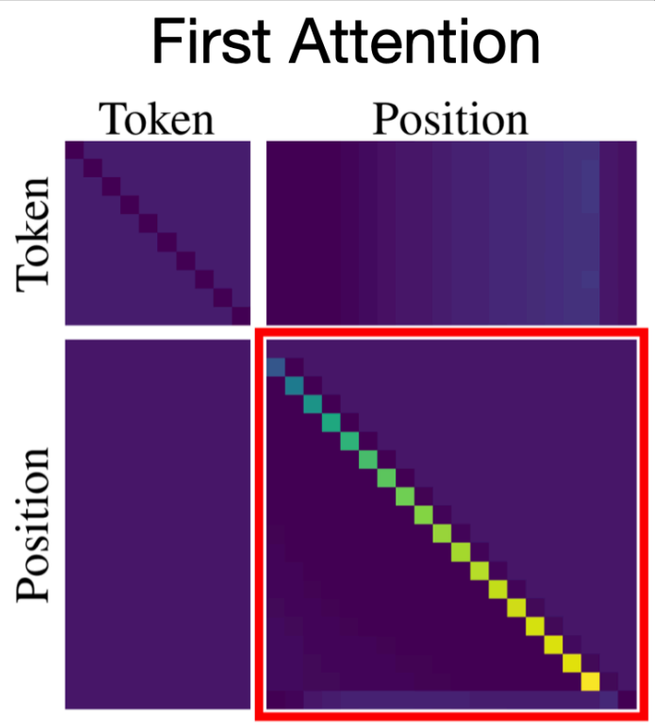
The first attention matrix is the adjacency matrix for the causal graph!

How do Transformers solve this task?



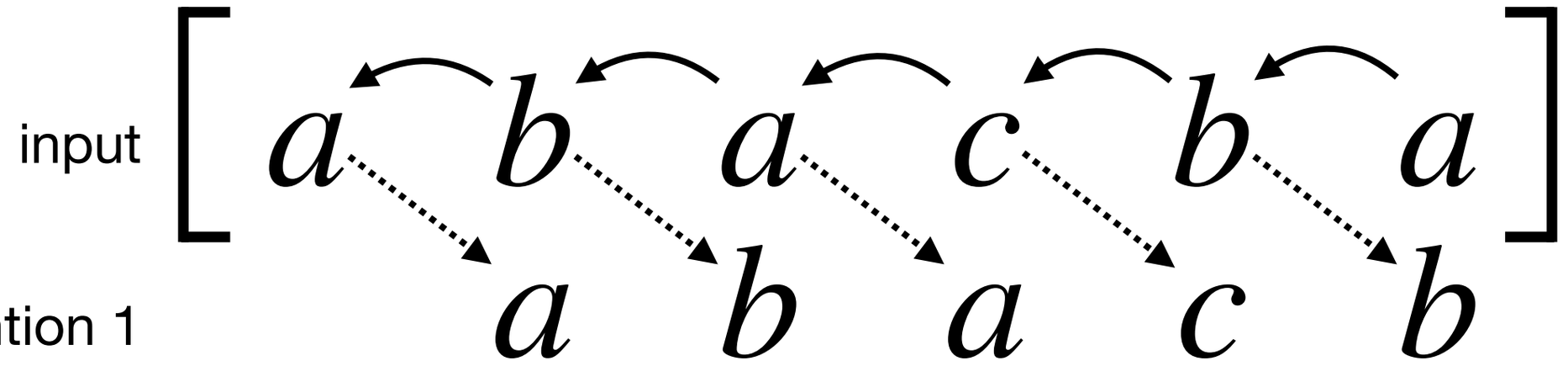
The first attention matrix is the adjacency matrix for the causal graph!

How Transformers Solve This Task

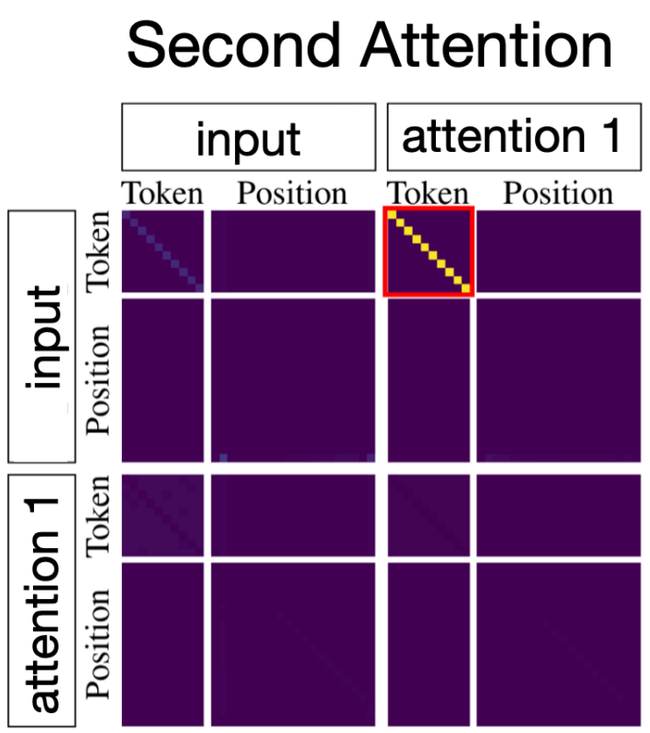


First Attention:

copy each parent

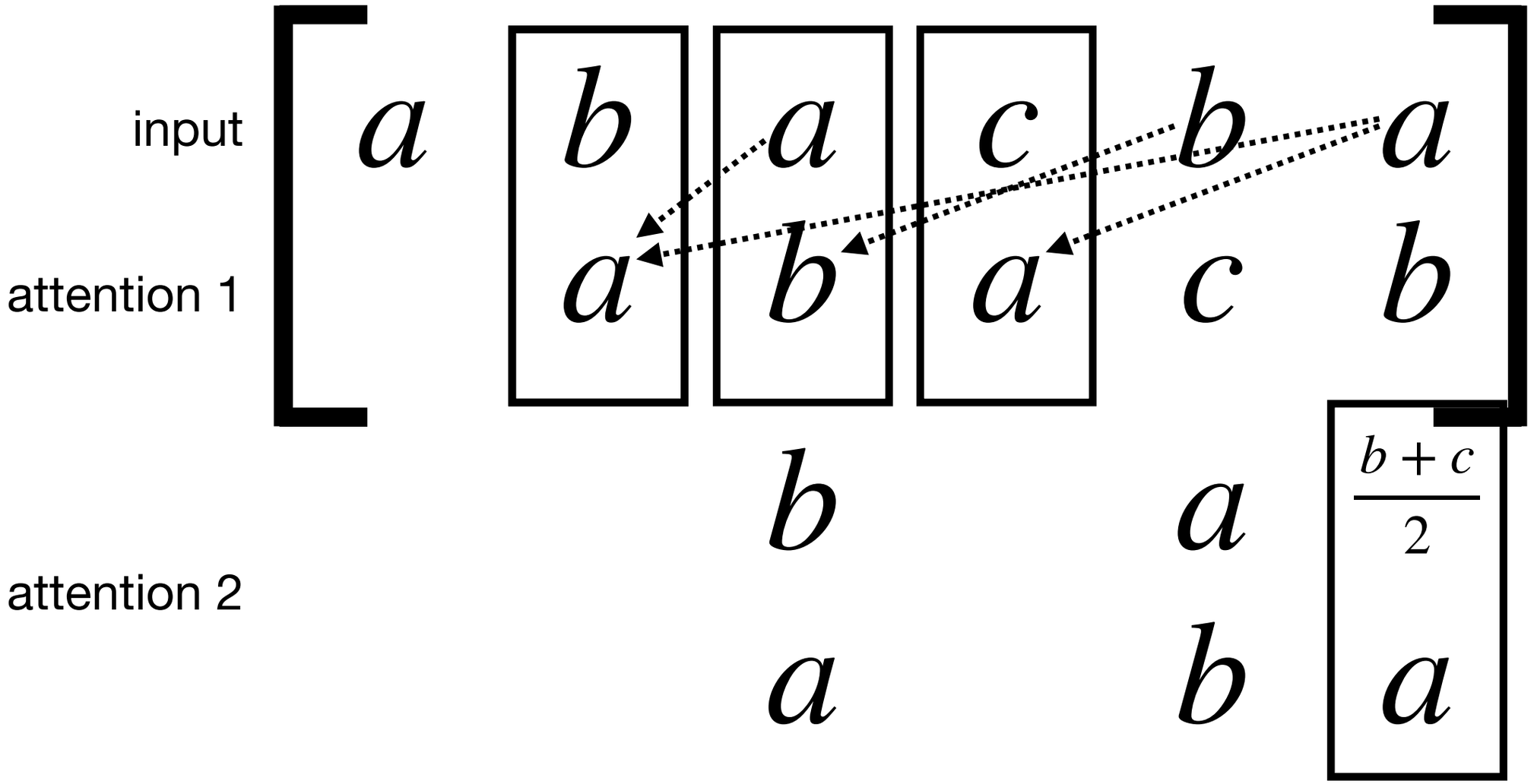


How Transformers Solve This Task

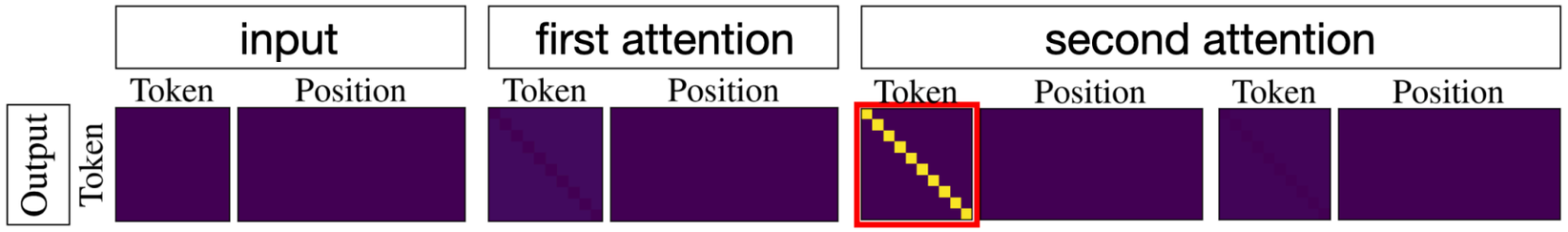


Second Attention:

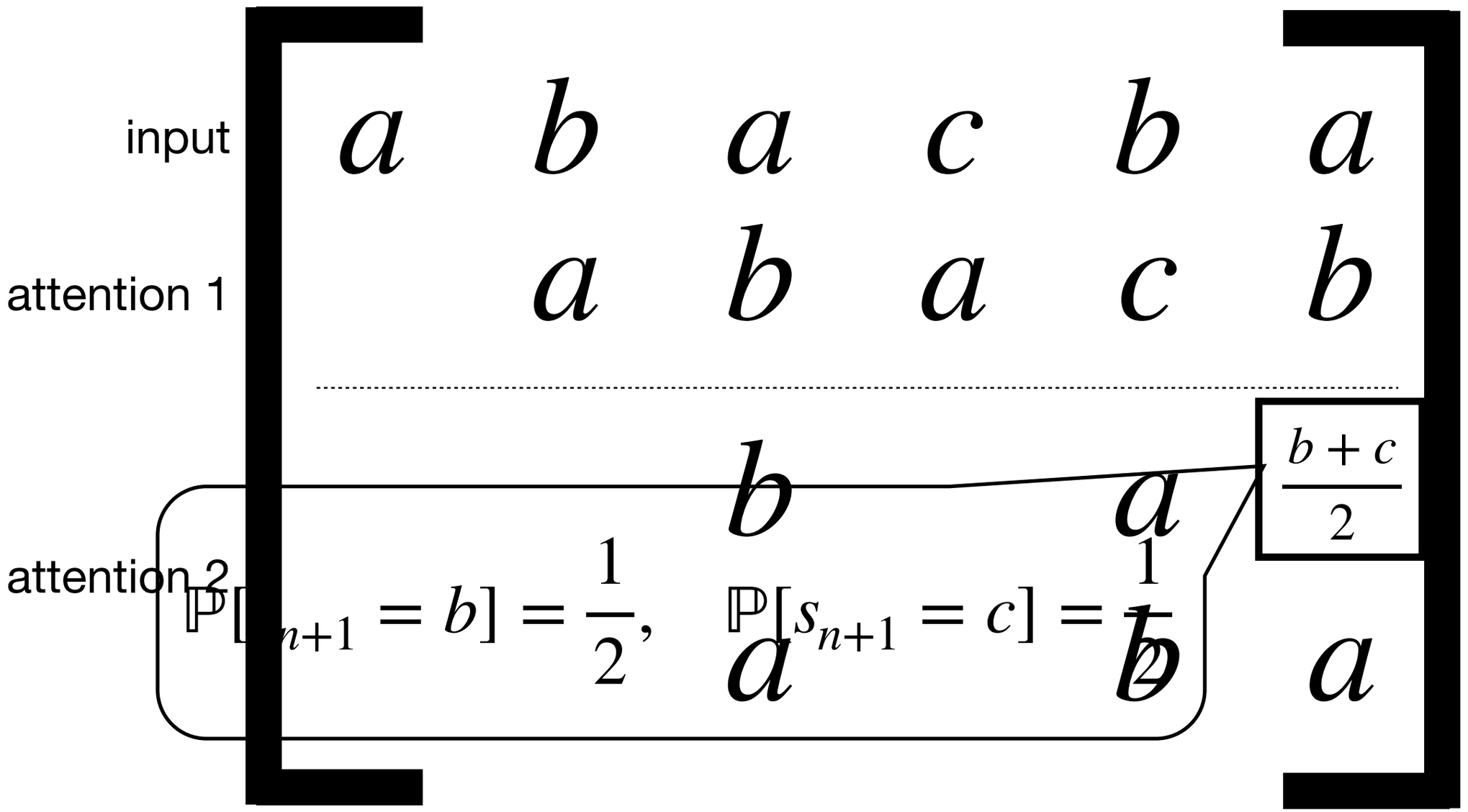
compare to each parent



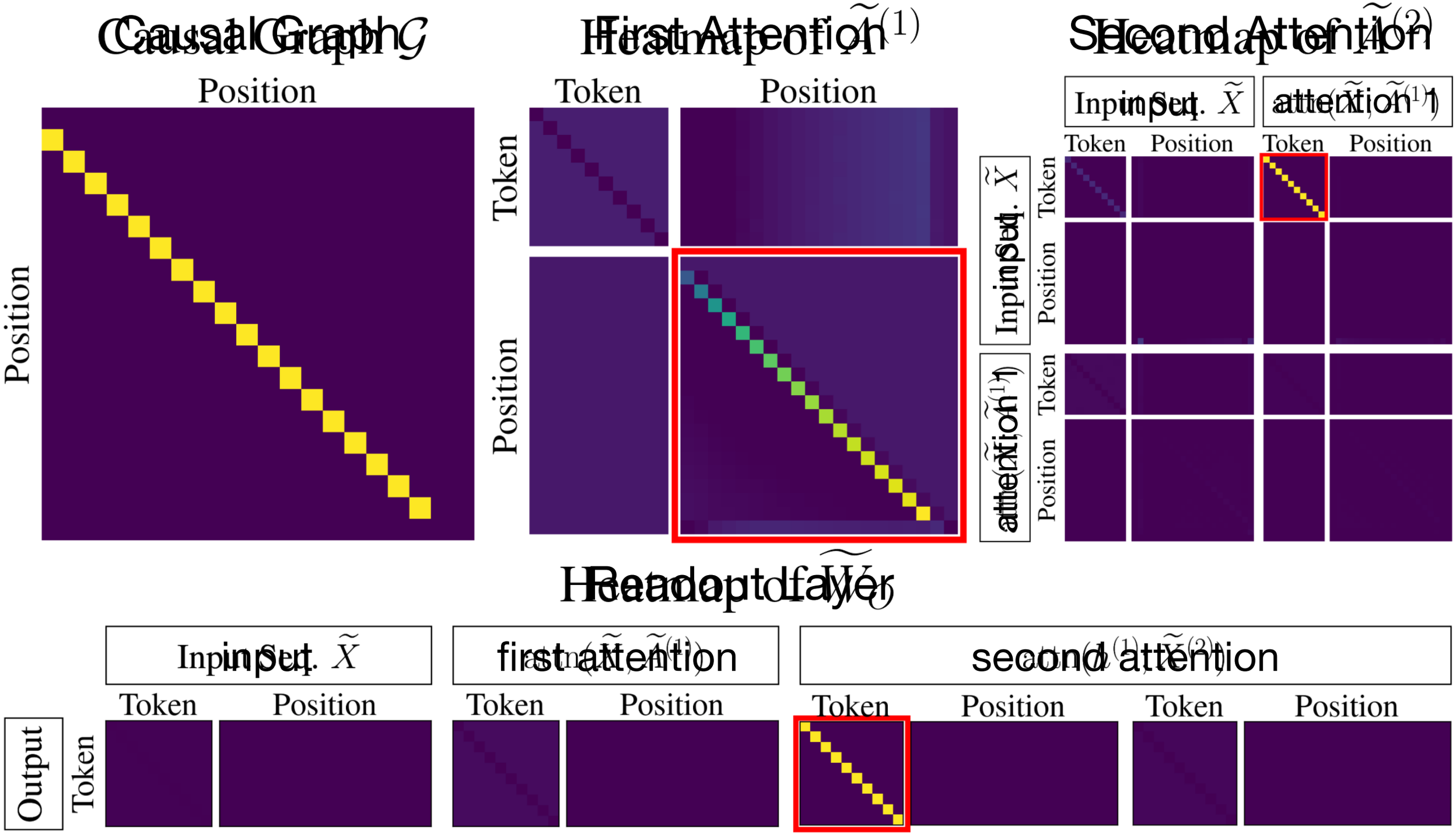
How Transformers Solve This Task



Readout Layer: output empirical counts



Gradient Descent Dynamics



Main Result

Loss: cross-entropy

$$L(\theta) = -\,\mathbb{E}_{\pi, s_{1:T}}\left[\sum_{s' \in [S]} \pi(s' \mid s_T) \log(f_\theta(s_{1:T})_{s'})\right]$$

Theorem (informal): If $\min_{s,s'} \pi(s \mid s') \geq \gamma/S$ almost surely over the prior P_π,

(1) There exists $c > 0$ such that GD returns θ satisfying:

$$L(\theta) - \mathrm{OPT} \lesssim \frac{1}{T^{c\gamma}}$$

(2) For any input sequence, the first attention pattern $A \in \mathbb{R}^{T \times T}$ satisfies:

$$\|A - G\|_\infty \lesssim \frac{1}{T},$$

where G is the adjacency matrix of the causal graph.

Corollary: Transformers trained on in-context Markov chains learn an induction head

OOD Generalization

Mechanistic understanding leads to provable OOD generalization:

Corollary:

Let $\tilde{\pi}$ satisfy $\min_{s,s'} \tilde{\pi}(s' \mid s) \geq \gamma/S$. Then with high probability over draw of $s_{1:T}$:

$$\left\| f_{\hat{\theta}}(s_{1:T}) - \tilde{\pi}(\cdot \mid s_T) \right\|_{\infty} \lesssim \frac{1}{T^{c\gamma}}$$

Note that $\tilde{\pi}$ does not need to be in the support of P_{π}.

Even if you learn an induction head on a very restricted class of sequences, this circuit automatically generalizes out of distribution to arbitrary sequences

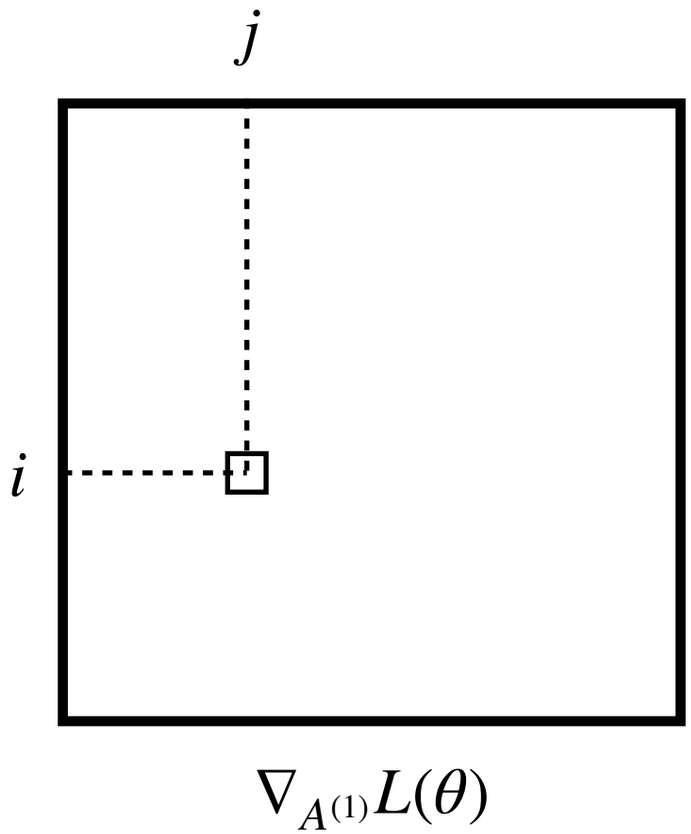
How Transformers Learn Causal Structure

Key Lemma: For $j < i$, the gradient of the first attention layer is approximately

$$\nabla_{A^{(1)}_{ij}} L(\theta) \approx -\, I^2_\chi(s_i; s_j \,|\, \pi) \quad \text{where} \quad I_{\chi^2}(s_i; s_j \,|\, \pi) := \mathbb{E}_\pi \left[\sum_{s_i, s_j} \frac{\mathbb{P}[s_i, s_j]^2}{\mathbb{P}[s_i]\mathbb{P}[s_j]} - 1 \right].$$

how much token i attends to token j

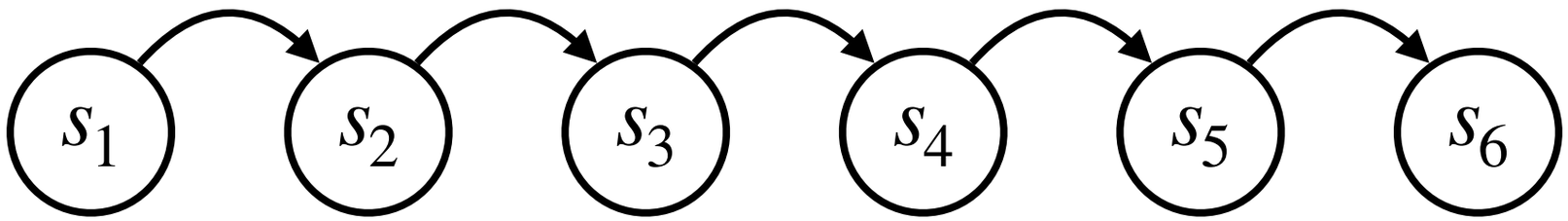
χ^2 mutual information between the token at position i and the token at position j



Corollary: Each position i will eventually attend to the position $j < i$ that maximizes the χ^2 mutual information between s_i and s_j

How Transformers Learn Causal Structure

Corollary: Each position i will eventually attend to the position $j < i$ that maximizes the χ^2 mutual information between s_i and s_j



in-context Markov chain

Data Processing Inequality:

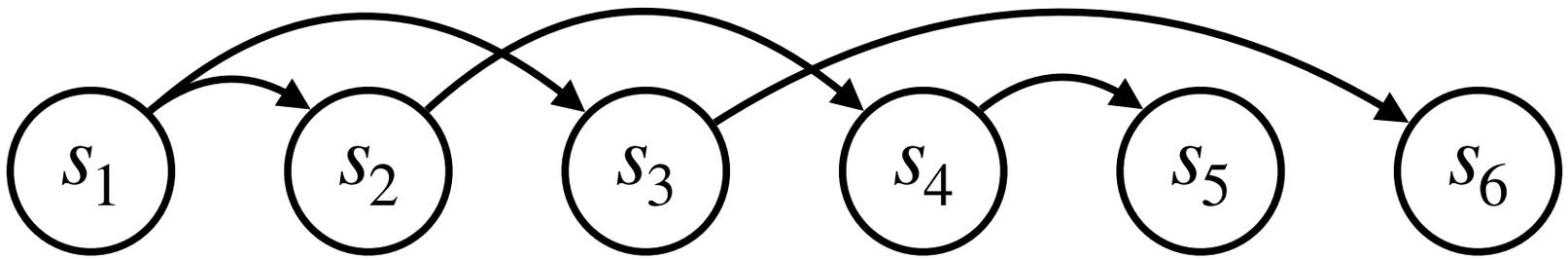
Passing through a channel can only decrease mutual information:

$$\ldots < I_\chi^2(s_6; s_3) < I_\chi^2(s_6; s_4) < I_\chi^2(s_6; s_5)$$

- Each token will attend to the token immediately before it
- The transformer learns an induction head!

How Transformers Learn Causal Structure

Corollary: Each position i will eventually attend to the position $j < i$ that maximizes the χ^2 mutual information between s_i and s_j



more complex causal structure

Data Processing Inequality:

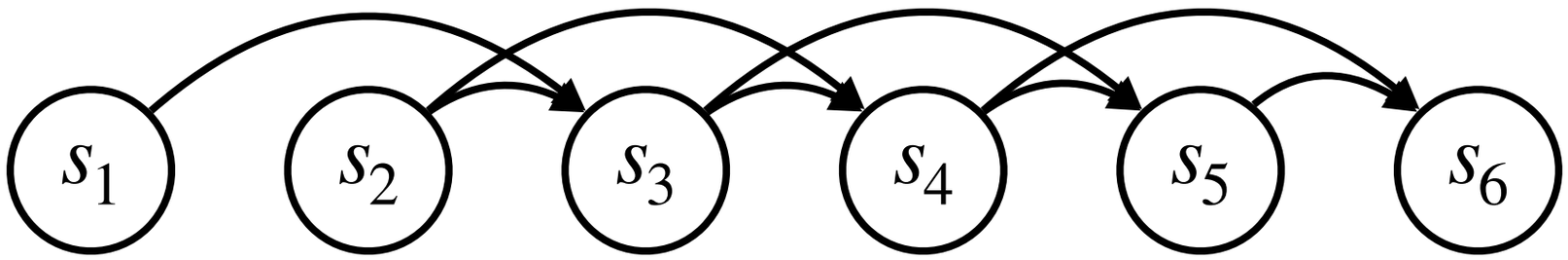
Passing through a channel can only decrease mutual information:

$I_{\chi^2}(s_i, s_j)$ is maximized at $j = p(i)$, the parent of i

- The first attention layer learns the causal graph
- Special case of the well-known Chow-Liu algorithm (Chow & Liu, 1968) for learning tree-structured graphical models!

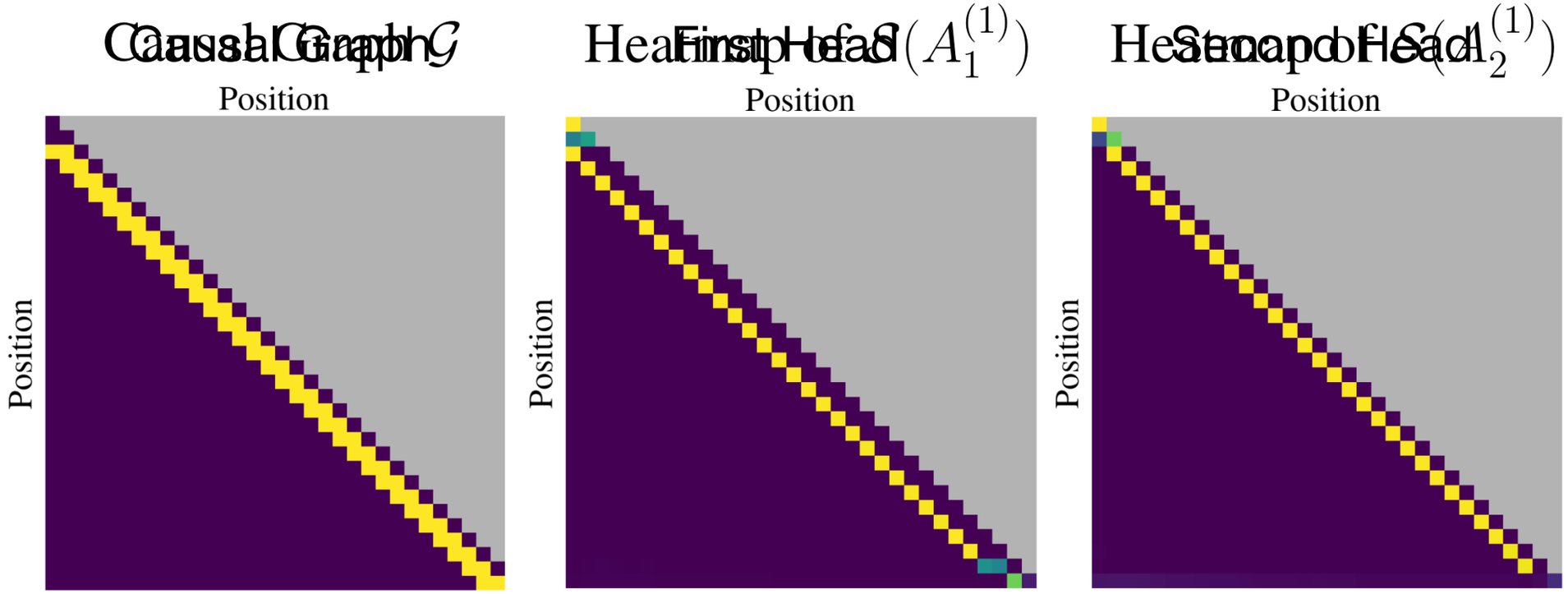
Beyond Tree Graphs — Multiple Parents & n-grams

- Each node can have multiple parents in the causal graph
- Example: n-gram language models



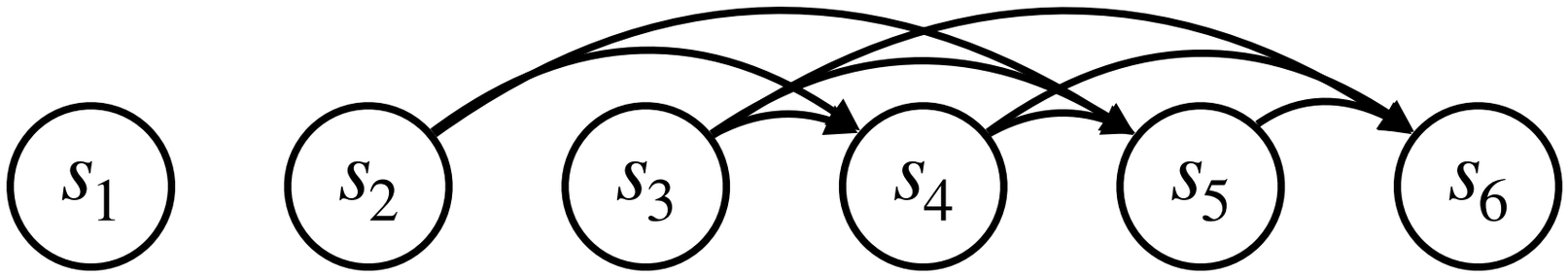
in-context 3-gram

Construction & Experiments: Each head attends to a different parent



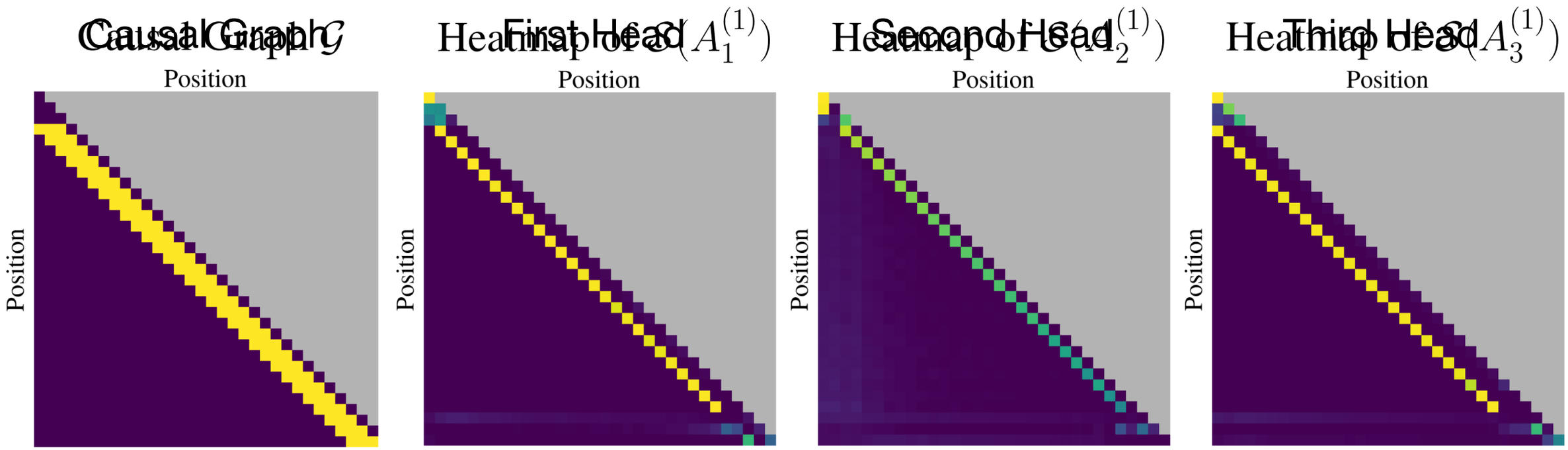
Beyond Tree Graphs — Multiple Parents & n-grams

- Each node can have multiple parents in the causal graph
- Example: n-gram language models



in-context 4-gram

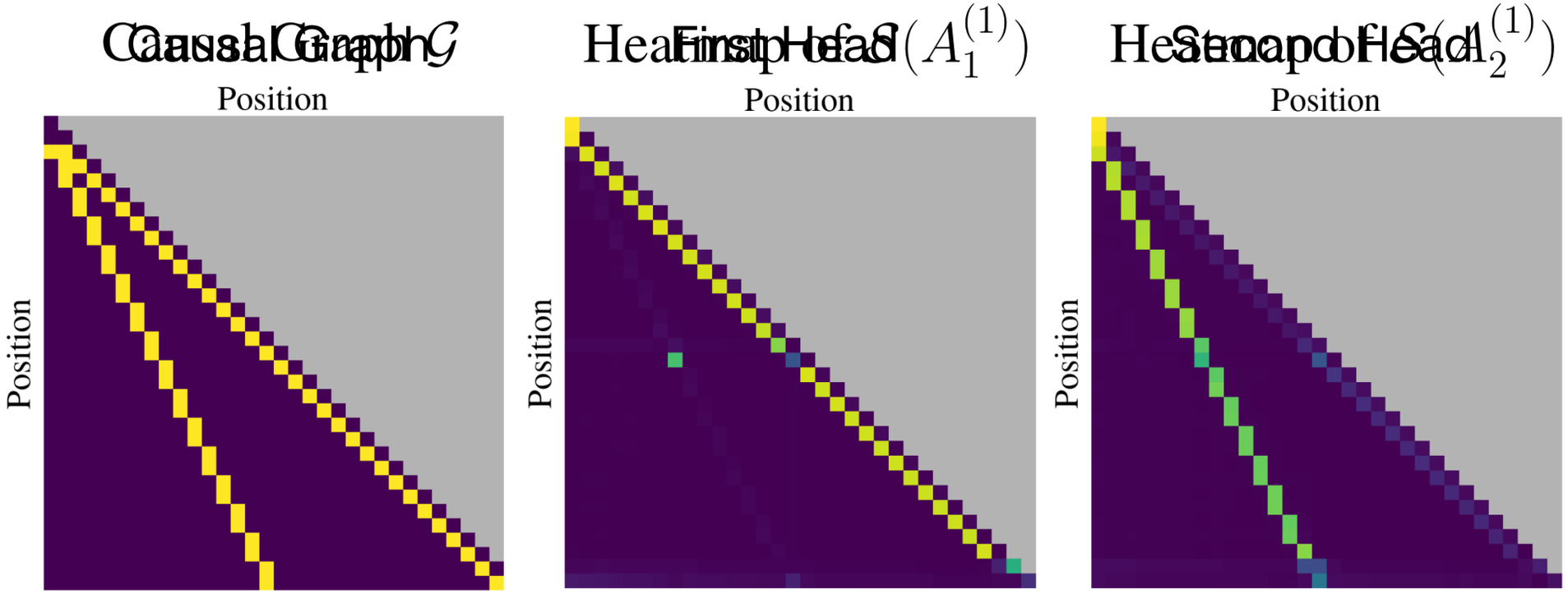
Construction & Experiments: Each head attends to a different parent



Beyond Tree Graphs — Multiple Parents & n-grams

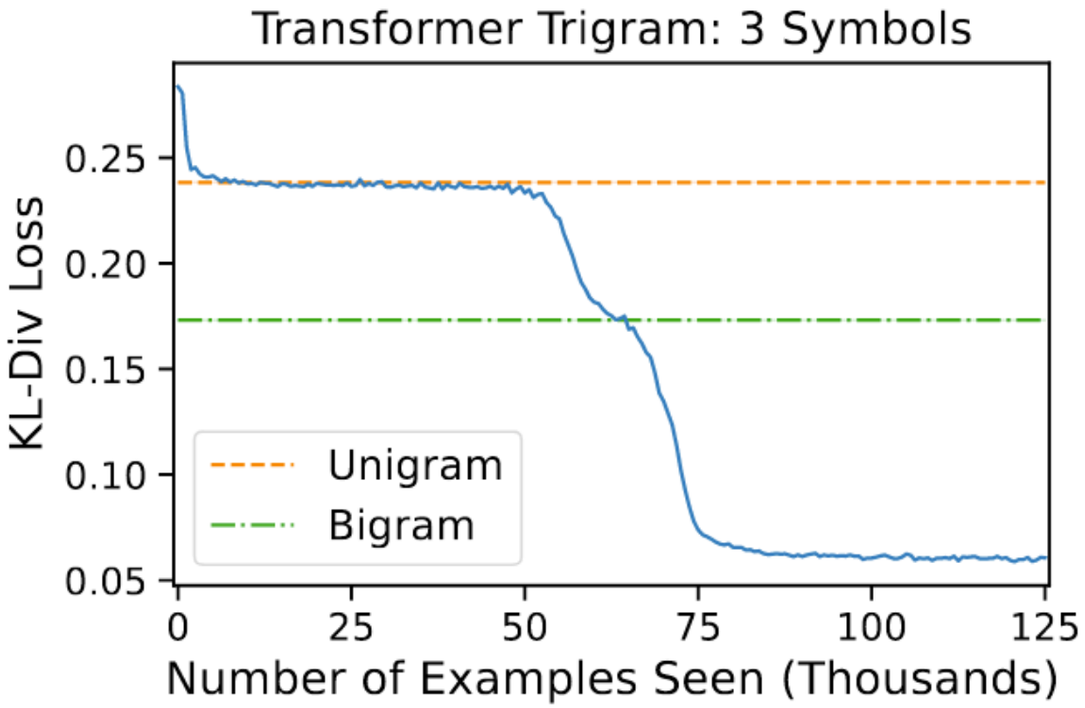
- Each node can have multiple parents in the causal graph
- Example: n-gram language models

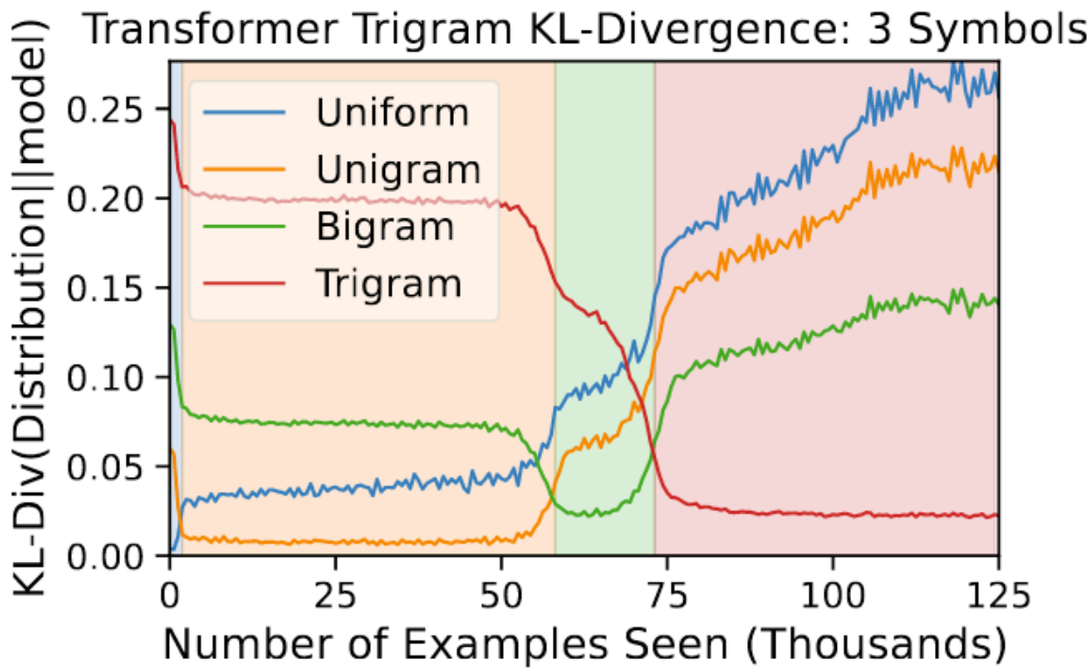
Construction & Experiments: Each head attends to a different parent



Gradient Descent Dynamics?

- Unfortunately, analyzing the GD dynamics is very challenging 🤕
- Dynamics/initialization must somehow break the symmetry between the multiple heads. Similar to learning a teacher net with H neurons
- (Edelman et al., 2024) observe sequential learning behavior. Model first learns best unigram, then bigram, and so on:





Takeaways

- Transformers learn causal structure by estimating and comparing the χ^2 mutual information between between tokens at different positions
- For Markovian sequences, Transformers learn induction heads
- Connection between the GD dynamics and graphical model estimation
- The disentangled transformer may be a useful tool for future theory

Interesting Directions:

- How general is this mechanism? Does it extend to more realistic datasets?
- How do transformers learn causal structure beyond trees?
- More interesting causal structures beyond absolute positional embeddings