A Causal World Model Underlying Next Token Prediction in GPT

Raanan Y. Rohekar*, Yaniv Gurwicz*,
Sungduk Yu, Estelle Aflalo, Vasudev Lal

Intel Labs

Abstract

Are generative pre-trained transformer (GPT) models only trained to predict the next token, or do they implicitly learn a world model from which a sequence is generated one token at a time? We examine this question by deriving a causal interpretation of the attention mechanism in GPT, and suggesting a causal world model that arises from this interpretation. Furthermore, we propose that GPT-models, at inference time, can be utilized for zero-shot causal structure learning for in-distribution sequences. Empirical evaluation is conducted in a controlled synthetic environment using the setup and rules of the Othello board game. A GPT, pre-trained on real-world games played with the intention of winning, is tested on synthetic data that only adheres to the game rules, oblivious to the goal of winning. We find that the GPT model is likely to generate moves that adhere to the game rules for sequences for which a causal structure is encoded in the attention mechanism with high confidence. In general, in cases for which the GPT model generates moves that do not adhere to the game rules, it also fails to capture any causal structure.

1 Introduction

In recent years, the generative pre-trained transformer (GPT) model (Radford et al. 2018) has demonstrated high-quality generative capabilities, as perceived by humans. Although this model is trained to generate one token at a time, it has been demonstrated to perform a range of tasks beyond next-token predictions, such as visual understanding and symbolic reasoning (Liu et al. 2024; Team et al. 2023; Chowdhery et al. 2023). Are these emergent abilities (Li et al. 2023) or merely a 'mirage' resulting from the choice of metric and task (Schaeffer, Miranda, and Koyejo 2024)?

In this paper we suggest that there is no restriction in the GPT architecture for learning conditional independence (CI) relations between tokens in a sequence. Moreover, under certain assumptions, a causal structure is directly entailed from these CI relations. One may ask whether this lack of restriction results in implicitly learning a causal model of the world during the pre-training procedure of GPT. Assuming that, both, a causal world model and a model based on surface statistics are sufficient solutions, one possibility is that a causal world model is a more compact solution and more likely to be learned during pre-training (Occam's razor). For example, if weights are distributed from a uniform distribution in the surface statistics model, then a causal structure limits the range of their distribution. If so, what are the assumptions underlying this causal world model?

Recently, Rohekar, Gurwicz, and Nisimov (2024) derived a causal interpretation for unmasked self-attention in BERT models (Devlin et al. 2019) demonstrating it for explaining movie recommendations (Nisimov et al. 2022). We follow a similar approach, with several differences, and propose a causal interpretation of the masked attention mechanism in GPT. Furthermore, we define a corresponding causal world model. The ABCD method (Rohekar, Gurwicz, and Nisimov 2024) is adapted and used to learn causal structures of which the induced dependency-relations are encoded in the attention matrices in GPT. We then ask whether errors generated by GPT are correlated with the uncertainty in representing the causal structure by the attention matrices. To this end, we define a metric based on the entropy of p-values of CI tests that are used for inferring the causal structures.

Recent work examined the internal process of large language models and examined whether a world model is implicitly learned using a well-defined and constrained setting, such as in the Chess game setting (Toshniwal et al. 2022) and Othello board game setting (Li et al. 2023). For the Othello board game setting, Li et al. (2023) demonstrated that the board state can be inferred from attention matrices in GPT, and Nanda, Lee, and Wattenberg (2023) showed that a linear classifier suffices to reconstruct the state of the board game from the attention matrices. They claim an emergent world model in GPT. Nevertheless, they do not explain how the board game is encoded within the attention matrices and why the attention mechanism can represent the board state. In essence, they do not provide an explanation to the apparent emergence of the world model. In addition, their reconstructed world model (board game state) applies only to the domain for which the GPT model was trained and lacks the generative mechanism underlying the token-sequences.

In this paper, we consider the structural causal model as the world model, which describes the generative process and is generally applicable for a range of applications (not domain specific, such as Othello board state). We explore whether GPT is able to capture properties of this world model, which may explain its apparent emergence. See an example result in Figure 1.

*These authors contributed equally.

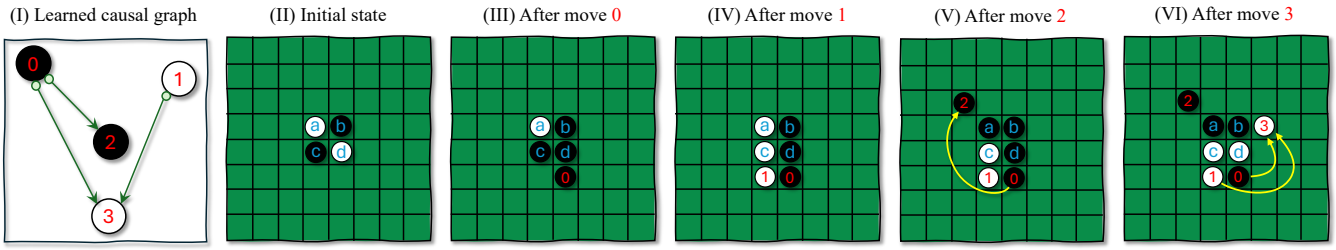

Figure 1: Red numbering {0,1,2,3} on the causal graph nodes and game board discs correspond to the game moves indices. The blueish letters {a,b,c,d} indicate the discs in the initial state of the board game. (I) The causal graph learned by our method given the sequence of moves described hereafter. (II) The initial state of the board game. (III) After move 0: black plays and flips disc 'd' to black. (IV) After move 1: white plays and flips disc 'c' to white. This move does not depend on the previous move 0, and this aligns with the learned causal graph in which node '1' is found independent of node '0'. (V) After move 2: black plays and flips disc 'a' to black. This was made possible since disc 'd' was black after being flipped to black in the earlier move 0 (yellow arrow shows this causal connection). Correspondingly, this causal connection is also revealed in the learned causal graph by node '0' being the sole parent of node '2'. (VI) After move 3: white plays and flips disc 'd' to white. This was made possible because disc '1' was white (due to move 1), and disc 'd' was black (as mentioned before, it was flipped to black earlier in move 0). Therefore we expect both moves 0 and 1 to be the causes of current move 3 (see yellow arrows). This is exactly revealed by the learned causal graph.

2 Preliminaries

In this section we provide notations and descriptions of self attention in the GPT architecture, and structural causal models. Matrices are written in bold, vectors in bold-italic, and models in calligraphic font. A summary of the main symbols used in this paper is given in Table 1 (Appendix A).

Attention in GPT

Attention is a mechanism that estimates network weights with respect to the context in an input sequence of tokens (Schmidhuber 1992). In a GPT model, which is based on the decoder part of the Transformer architecture (Vaswani et al. 2017), an attention layer estimates an $n \times n$ lower-triangular (masked) attention matrix $\mathbf{A}$ given an input sequence of n tokens. The input sequence is in the form of an $n \times d$ matrix $\mathbf{Y}$, where the i-th row vector is $\mathbf{Y}(i,\cdot)$, is an embedding (representation) of the i-th token in d dimensions. The attention matrix is estimated by $\mathbf{A} = softmax(\mathbf{Y}\mathbf{W}_{QK}\mathbf{Y}^\top)$, such that $\mathbf{A}$ is lower triangular and the rows sum to 1[1]. In addition to the attention weights, the attention layer calculates a values matrix, $\mathbf{V} = \mathbf{Y}\mathbf{W}_V$, where row $\mathbf{V}(i,\cdot)$ is the value vector of the i-th token. Then, the output embeddings are

$$\mathbf{Z} = \mathbf{A}\mathbf{V}, \tag{1}$$

where the i-th row, $\boldsymbol{Z}_i$, is the embedding of the i-th output token. In a GPT, several attention layers are stacked, and pretrained such that the i-th output embedding in the last layer predeicts the $(i+1)$-th input token. That is, predicts the next input token.

It is important to note that in the GPT architecture, the embedding of one token is influenced by another token only by the attention matrix, $\mathbf{A}$. In addition, note that an attention matrix $\mathbf{A}$ is estimated *uniquely* for each input sequence of tokens, using weight matrices $\{\mathbf{W}_{QK}, \mathbf{W}_V\}$ learned *commonly* for all in-distribution input sequences.

[1]The weight matrix is $\mathbf{W}_{QK} = \mathbf{W}_Q\mathbf{W}_K^\top/\sqrt{d_K}$, where generally the weight matrices $\mathbf{W}_Q$ and $\mathbf{W}_K$ are learned explicitly and d_K is the number of columns in these matrices (Vaswani et al. 2017).

Structural Causal Model

A structural causal model (SCM) is a model that can encode causal mechanisms in a domain (Pearl 2009; Spirtes, Glymour, and Scheines 2000; Peters, Janzing, and Schölkopf 2017) and explain data samples generated from these causal mechanisms (Pearl and Mackenzie 2018). An SCM is a tuple $\{\boldsymbol{U}, \boldsymbol{X}, \mathcal{F}, P(\boldsymbol{U})\}$, where $\boldsymbol{U} = \{U_1, \ldots, U_m\}$ is a set of latent exogenous random variables, $\boldsymbol{X} = \{X_1, \ldots, X_n\}$ is a set of endogenous random variables, $\mathcal{F} = \{f_1, \ldots, f_n\}$ is a set of deterministic functions describing the values $\boldsymbol{X}$ given their direct causes, and $P(\boldsymbol{U})$ is the distribution over $\boldsymbol{U}$. Moreover, each endogenous variable X_i has exactly one unique exogenous cause U_i $(m = n)$. The value of an endogenous variable $X_i, \forall i \in [1, \ldots, n]$ is determined by

$$X_i \leftarrow f_i(\boldsymbol{Pa}_i, U_i) \tag{2}$$

where $\boldsymbol{Pa}_i$ is the set of direct causes (parents in the causal graph) of X_i, and left-arrow indicates assignment resulting from the cause-effect relation. A graph $\mathcal{G}$ corresponding to an SCM consists of a node per variable, and directed edges for direct cause-and-effect relations that are evident from $\mathcal{F}$.

In this paper we employ a linear-Gaussian SCM (as we later relate it to the linear relations in GPT attention) having directed acyclic graphs (DAG). In these models each variable is determined by a linear combination of its direct causes and an independently distributed additive noise determined by a corresponding normally distributed exogenous variable.

For a linear-Gaussian SCM let $\mathbf{G}$ be a weight matrix, where $\mathbf{G}(i,j)$ is the weight of parent (direct cause) node X_j linearly determining the child (direct effect) node X_i. Node X_k is not a parent of X_i if and only if $\mathbf{G}(i,k) = 0$. In addition, $\boldsymbol{U} \sim \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{U}}, \mathbf{C}_{\boldsymbol{U}})$, where in this paper we assume $\mathbf{C}_{\boldsymbol{U}}$

is a diagonal matrix. The set of functions $\mathcal{F}$ is defined such that $\forall i \in [1, \ldots, n]$,

$$X_i \leftarrow \mathbf{G}(i, \cdot)\boldsymbol{X} + U_i. \tag{3}$$

Assuming a DAG and causally sorted nodes (ancestors precede their descendants), $\mathbf{G}$ is strictly lower triangular (zero diagonal). Given the assignment, we can write in matrix form $\boldsymbol{X} = \mathbf{G}\boldsymbol{X} + \boldsymbol{U}$, and

$$\boldsymbol{X} = (\mathbf{I} - \mathbf{G})^{-1}\,\boldsymbol{U}. \tag{4}$$

Since $\mathbf{G}$ is a *strictly* lower-triangular weight matrix, $(\mathbf{I} - \mathbf{G})^{-1}$ is a lower *uni-triangular* matrix (ones on the diagonal). Note that this is equal to the sum of a geometric series

$$(\mathbf{I} - \mathbf{G})^{-1} = \sum_{k=0}^{n-1} \mathbf{G}^k. \tag{5}$$

It can be seen that element (i, j) represents the cumulative effect of X_j on X_i via all directed paths having length up to $n - 1$. The equivalent weight of a directed path from X_j to X_i is the product of the weights of all edges on that path, and the cumulative effect via all the paths is the sum over equivalent weights of distinct directed paths from X_j to X_i. Note that even if some of the nodes are latent confounders is still $(\mathbf{I} - \mathbf{G})^{-1}$ triangular because, by definition, latent confounders do not have ancestors and are first in a topological ordering. Equation 4 represents a system with input $\boldsymbol{U}$, output $\boldsymbol{X}$ and weights $(\mathbf{I} - \mathbf{G})^{-1}$. The covariance matrix of the output is

$$\begin{aligned}
\mathbf{C}_{\boldsymbol{X}} &= \mathbb{E}[(\boldsymbol{X} - \boldsymbol{\mu}_{\boldsymbol{X}})(\boldsymbol{X} - \boldsymbol{\mu}_{\boldsymbol{X}})^\top] = \\
&= \mathbb{E}[(\mathbf{I} - \mathbf{G})^{-1}\,\hat{\boldsymbol{U}}\hat{\boldsymbol{U}}^\top\,((\mathbf{I} - \mathbf{G})^{-1})^\top] = \\
&= [(\mathbf{I} - \mathbf{G})^{-1}]\,\mathbb{E}[\hat{\boldsymbol{U}}\hat{\boldsymbol{U}}^\top]\,[(\mathbf{I} - \mathbf{G})^{-1}]^\top = \\
&= \left[(\mathbf{I} - \mathbf{G})^{-1}\right]\,\mathbf{C}_{\boldsymbol{U}}\,\left[(\mathbf{I} - \mathbf{G})^{-1}\right]^\top,
\end{aligned} \tag{6}$$

where $\hat{\boldsymbol{U}} = \boldsymbol{U} - \boldsymbol{\mu}_{\boldsymbol{U}}$ and $\boldsymbol{\mu}_{\boldsymbol{X}} = (\mathbf{I} - \mathbf{G})^{-1}\boldsymbol{\mu}_{\boldsymbol{U}}$.

In this paper we employ the constraint-based causal discovery approach (Spirtes, Glymour, and Scheines 2000) that use conditional independence (CI) tests to learn the underlying causal graph. This approach generally requires assuming the causal Markov property and faithfulness.

Definition 1 (Causal Markov) *In a causally Markov graph, a variable is independent of all other variables, except its effects, conditional on all its direct causes.*

Definition 2 (Faithfulness) *A distribution is faithful to a graph if and only if every independence relation true in the distribution is entailed by the graph.*

3 A Causal Interpretation of GPT

We describe the masked attention in GPT as a mechanism that infers correlations between tokens of a given input sequence, where these correlations are induced by a causal structure underlying the output sequence tokens. We then describe a method for learning a causal graph by estimating independence relations between tokens.

A Relation between GPT and SCM World Model

Rohekar, Gurwicz, and Nisimov (2024) derived a causal interpretation of BERT-based models (Devlin et al. 2019). We follow a similar approach, with several important modifications and extensions, to derive a causal interpretation to GPT. First, unlike BERT-based models, which are pre-trained to predict masked tokens within the input sequence using the surrounding tokens (Devlin et al. 2019), GPT is pre-trained to predict the next tokens in the sequence. That is, given an input sequence of tokens, $\{t_0, \ldots, t_{n-1}\}$, GPT predicts tokens $\{\hat{t}_1, \ldots, \hat{t}_n\}$. An attention matrix $\mathbf{A}$ and the corresponding values matrix $\mathbf{V}$ have n rows corresponding to input tokens $\{t_0, \ldots, t_{n-1}\}$ and the output embeddings of of these tokens are the rows of matrix $\mathbf{Z} = \mathbf{AV}$. Thus, Note that $\mathbf{V} = \mathbf{Y}\mathbf{W}_V$, where $\mathbf{W}_V$ is a weight matrix fixed for all input sequences, and $\mathbf{Y}$ is input embedding of a specific sequence tokens. Each column of $\mathbf{W}_V$ can be viewed as an independent vector onto which the input embeddings are projected. That is $\mathbf{V}(i, j)$ is the projection of token t_i input embedding $\mathbf{Y}(i, \cdot)$ on, common to all in-distribution sequences, vector $\mathbf{W}_V(\cdot, j)$. At inference, each attention matrix of the last attention layer, $\mathbf{A}$, is extracted and a lower uni-triangular matrix is calculated, $\mathbf{D}^{-1}\mathbf{A}$, where $\mathbf{D} \equiv \mathrm{diag}(\mathbf{A})$. Then the covariance matrix is estimated

$$\mathbf{C} = \left[\mathbf{D}^{-1}\mathbf{A}\right]\left[\mathbf{D}^{-1}\mathbf{A}\right]^\top. \tag{7}$$

Note that unlike Rohekar, Gurwicz, and Nisimov (2024), which proposed $\mathbf{C} = \mathbf{A}\mathbf{A}^\top$ for *unmasked* self-attention, we utilize the triangular form of the masked attention in GPT to revert the attention normalization performed by the softmax and obtain a uni-triangular form. Thus, this covariance matrix allows us to treat properties calculated from different attention matrices in a similar manner. In this paper (Section 3 and Section 9), properties we calculate are based on p-values when testing conditional independence relations between tokens. Next, following Rohekar, Gurwicz, and Nisimov (2024) we relate each token to an endogenous node in an SCM, and $\mathbf{C}_{\boldsymbol{U}} = \mathbf{I}$ from the central limit theorem (Rohekar, Gurwicz, and Nisimov 2024). Thus, equate covariance $\mathbf{C} = \mathbf{C}_{\boldsymbol{U}}$

$$\left[\mathbf{D}^{-1}\mathbf{A}\right]\left[\mathbf{D}^{-1}\mathbf{A}\right]^\top = \left[(\mathbf{I} - \mathbf{G})^{-1}\right]\,\left[(\mathbf{I} - \mathbf{G})^{-1}\right]^\top, \tag{8}$$

where both $\mathbf{D}^{-1}\mathbf{A}$ and $(\mathbf{I} - \mathbf{G})^{-1}$ are lower uni-triangular matrices, and the (i, j) elements, $\forall i > j$, of these matrices have the same meaning, influence of token/node j on token/node i. Finally, since GPT is pre-trained to predict tokens $\{t_1, \ldots, t_n\}$ given input tokens $\{t_0, \ldots, t_{n-1}\}$, and since the only cross-token influence on embeddings is in the attention layers, the last attention layer captures the causal structure underlying the output tokens. Earlier attention layers transform embeddings of $\{t_0, \ldots, t_{n-1}\}$ to values, $\mathbf{V}$, which are equivalent to instantiations of exogenous variables, $\boldsymbol{U}$ in SCM. This follows from equating Equation 1 and Equation 4, where $\mathbf{D}^{-1}\mathbf{A} = (\mathbf{I} - \mathbf{G})^{-1}$.

In light of the causal interpretation of GPT, one important question is what is the *causal world model* that is supported by the GPT architecture. Often, a single causal structure is assumed to govern a domain. In contrast, the causal

world model that is entailed from the causal interpretation of GPT assumes a distinct structural causal model for each in-distribution sequence. Specifically, in a causal world model supported by a GPT with k-heads in the last attention layer, each in-distribution sequence is assumed to be generated by an ensemble of k distinct SCMs.

In addition, for a given head, the causal structure over a sequence of tokens $\{t_1, \ldots, t_n\}$ is equal to the sub-graph over these tokens for all in-distribution extensions of the sequence. That is, given a sequence of tokens $\{t_1, \ldots, t_n\}$ and a corresponding graph structure $\mathcal{G}_n$, observing any next token, t_{n+1}, such that $\{t_1, \ldots, t_n, t_{n+1}\}$ is in-distribution, should not violate causal relations in $\mathcal{G}_n$ and may only reveal relations between tokens $\{t_1, \ldots, t_n\}$ and token t_{n+1}.

GPT for Zero-Shot Causal Structure Learning

The causal interpretation presented in this paper leads to a view in which each attention module represents associations (correlations) between input tokens that are induced by the underlying causal structure. Although this allows only rung-1 inference in the ladder of causation (Pearl and Mackenzie 2018), under certain assumptions, many of the underlying causal relations can be extracted, even in the presence of latent confounders and selection bias (Spirtes, Glymour, and Scheines 2000). These relations are generally represented in a type of causal structure called partial ancestral graph (PAG) (Richardson and Spirtes 2002). We follow a procedure called ABCD, proposed by Rohekar, Gurwicz, and Nisimov (2024) with several modifications. First, since the causal (topological) order is given (restricted by the masked attention in GPT) we can apply causal discovery recursively to efficiently learn the causal structure. To this end we call the iterative causal discovery (ICD) algorithm (Rohekar et al. 2021), as used in ABCD, to reconstruct a causal structure in each recursive iteration. The procedure is described in Algorithm 2. The input is a sequence of token over which we construct the graph. The output is a PAG structure. In line 2 an exit condition corresponding to the base case (single-node graph) is tested. In line 3, the last token is popped from the sequence and placed in t_n resulting in a shorter sequence $\boldsymbol{S}'$. Then, a recursive call is made in in line 4 to learn the structure over tokens in $\boldsymbol{S}'$. Note that since it is ensured that t_n is not an ancestor of any token in $\boldsymbol{S}'$ the skeleton and v-structure relations of $\mathcal{G}'$ is ensured not to be change when adding back t_n to the graph (Spirtes, Glymour, and Scheines 2000). In lines 5–7 token t_n is connected to every node in $\mathcal{G}'$. Finally, in line 8 edges between t_n and the rest of the graph are learned (removed if conditional independence is found) using the ICD algorithm (Rohekar et al. 2021) and the graph is oriented (Zhang 2008). Although we use ICD, other constrained-based causal discovery algorithms (Colombo et al. 2012; Claassen, Mooij, and Heskes 2013; Yehezkel and Lerner 2009; Spirtes, Glymour, and Scheines 2000; Rohekar et al. 2018; Nisimov et al. 2021), differing in their underlying assumptions, can be used.

Thus, a causal structure for a specific output sequence can be learned in a zero-shot manner directly from the attention matrix in the last layer. In multi-head attention, the last attention layer, having k heads, is the last layer in which tokens may affect one another. Hence, Algorithm 2 is called for each head independently and a set of k structures is returned.

Algorithm 1: Recursive Causal Discovery for GPT

Input: $\boldsymbol{S}$: a sequence of tokens $\{t_1, \ldots, t_n, \}$

Output: $\mathcal{G}$: a partial ancestral graph (PAG)

1 **Function** `LearnStructure(`$\boldsymbol{S}$`)`**:**
2 if $|\boldsymbol{S}| = 1$ then return a graph with the single node in $\boldsymbol{S}$
3 $t_n, \boldsymbol{S}' \leftarrow \mathrm{pop}(\boldsymbol{S})$
4 $\mathcal{G}' \leftarrow$ `LearnStructure(`$\boldsymbol{S}'$`)`
5 $\mathcal{G} \leftarrow \mathcal{G}' + \{t_n\}$
6 set $\boldsymbol{E}$ to the set of edges (circle edge-marks) between t_n and every node in $\mathcal{G}'$
7 connect $\boldsymbol{E}$ in $\mathcal{G}$
8 test CI for edges in $\boldsymbol{E}$ and orient $\mathcal{G}$ using Algorithm 3 (Appendix C)
9 return $\mathcal{G}$

Causal Structure Confidence

In this section we derive a metric that describe how compatible a sequence is with the causal world model implicitly encoded by GPT. Given an output sequence of tokens, $\boldsymbol{S}$, and a causal structure $\mathcal{G}$ recovered from the last attention layer $\mathbf{A}$, can we score the confidence in this causal structure? Recall that in the proposed world model each sequence has its own causal structure, and each causal structure may have latent variables. It is not clear how to calculate likelihood $P(\boldsymbol{S}|\mathcal{G})$. We therefore propose the following approach.

A causal structure-learning algorithm performs multiple statistical tests of conditional independence (CI) using the covariance matrix estimated from the attention matrix. These CI tests calculate p-values and compare them against a predetermined threshold of significance level (α). It is important to note that there is a one-to-one correspondence between the results of these CI test and the entailed causal structure. That is, a causal structure can be represented uniquely by a set of CI tests and their results. Hence, we propose a scoring function based on the distribution of these p-values to evaluate the confidence in a structure learned from a given attention matrix. A complete undirected graph corresponds to lack of knowledge about causal relations. Generally, causal structure-learning algorithms prune edges from this graph based on statistical CI tests between pairs of variables (tokens in our case). The removal of edges between independent variables then may entail causal relations between other variables Zhang (2008). Let $\boldsymbol{p} = \{p_1, \ldots, p_\ell\}$ a set of all p-values computed as part of causal structure learning. The null-hypothesis is independence, where p-values greater than the significance threshold, α, correspond to edges removed from the complete graph. We denote $\boldsymbol{p}_{ind} = \{p : p \in \boldsymbol{p} \text{ and } p \geq \alpha\}$, and $\boldsymbol{p}_{dep} = \{p : p \in \boldsymbol{p} \text{ and } p < \alpha\}$. Since p-values are uni-

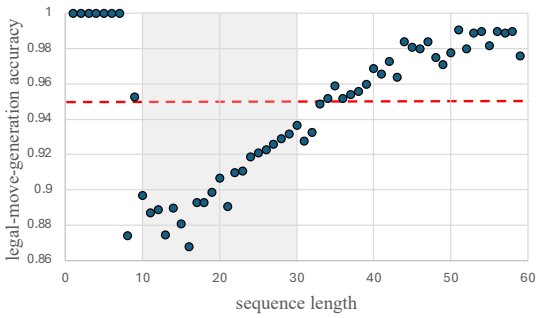

Figure 2: Baseline model accuracy of generating legal Othello game moves. A model trained by Li et al. (2023) on real-world games to predict the next move. Test set consists of randomly generated sequences. Measured accuracy: the percentage of generated moves that are legal according to the Othello game rules. Gray area shows input sequences with sizes in the range $[10, 30]$ where the accuracy is lower than the average of 95% (red dashed line).

formly distributed under the null hypothesis, we expect the entropy of p-values corresponding to independence, redundant relations (spurious correlations), H_{ind} to be higher for matrices that correspond to a structure compared to those that do not. In addition, we expect the distribution of p-values smaller than the significance level to be weighted towards zero. Hence, entropy of p-values corresponding to dependence relations, H_{dep} is expected to be lower for matrices that correspond to a structure compared to ones that do not. We therefore define the following confidence score given an attention matrix $\mathbf{A}$,

$$R(\mathbf{A}) = H_{\text{ind}} - H_{\text{dep}}, \tag{9}$$

where $H_{ind} = -\sum_{p \in \boldsymbol{p}_{ind}} p \log p$ and $H_{\text{dep}} = -\sum_{p \in \boldsymbol{p}_{\text{dep}}} p \log p$, are entropy of p-values corresponding to independence and dependence, respectively.

4 Experiments and Results

We use an experiment setup in which the world layout and rules governing the generation of sequences are well defined and known, but were not utilized during training. We measure how well attention in the trained GPT model represents a causal world model and whether it is correlated with the ability to generate tokens that adhere to the world rules.

Setup

We examine a GPT model trained by Li et al. (2023), for predicting the next move given a sequence of consecutive moves in the Othello strategy board game. They trained the model on approximately 132,000 real-world sequences, where it is assumed the players played with the intention of winning. No information about the game board layout or game rules was used in their training process. For example, positional encoding was not used. In our experiments we use a test set that is not in-distribution with respect to the training set, but in-distribution with respect to the game rules. As a test set we use 1,000 randomly generated sequences of legal moves. That is, each sequence consists of moves that adhere to the Othello game rules but without considering any strategy of winning the game as in the training set. In other words, the support of the test distribution is not a subset of the support of the training distribution,

$$\text{supp}(P_{\text{train}}) \subset \text{supp}(P_{\text{test}}). \tag{10}$$

See Appendix B for more details. This enables evaluating whether the model implicitly encoded the game rules.

In Figure 2 we plot the accuracy of the model in generating a legal next move (vertical axis) as a function of the number of tokens (length) in test input sequences (horizontal axis). Note that length n sequences are test sequences that are trimmed keeping only the first n tokens, such that the same 1,000 sequences are used for all evaluated lengths. Although the average accuracy of the model is 95% (dashed red line), it is not uniformly distributed across different sequence lengths. For example, given a sequence of 15 moves, GPT generates a legal 16-th move in 88% of the times (adheres to the game board state and rules). It is evident that the accuracy is significantly lower for input sequence lengths in the range $[10, 30]$ (lower than the average 95%). By definition of the Othello game rules, at the beginning of a game there are only four legal moves, and as this game unfolds, the number of possible legal moves generally increases before finally decreasing again as the number of vacant spaces on the board decreases. It might be that memorization of surface statistics can take place at the beginning of the game. We therefore report experiment results for input sequences with sizes in the range $[10, 30]$ (gray area) where the accuracy is lower than the average. Throughout the experiments, we employ Algorithm 2 for causal discovery using partial-correlation with $\alpha = 0.01$ as p-value threshold for testing conditional independence (CI tests).

Legal Predictions vs. Structural Confidence

Is there a relation between the accuracy of generating legal tokens (moves) and how well attention matrices represent (implicitly) causal structures? Recall that the model was not trained explicitly to generate legal Othello game moves but rather to predict the next move played by a human with the intention of winning the game. Moreover, information about the game, such as the existence of a board game and rules, were not provided to the model (Li et al. 2023). In this experiment we examine whether the cases in which the model generates illegal tokens, are also cases where the causal structure is less distinctive as measured by the structural confidence score, R (Equation 9). Here, the score for a given sequence is the average of structural confidence scores calculated for the eight attention heads in the last layer. From Figure 3 it is evident that the legal move generation accuracy (vertical axis) monotonically increases with the structural confidence score R (horizontal axis) for sequence lengths in $[15, 30]$. However, for sequence length 10 there is no clear trend. We suspect that for short sequence lengths in the Othello game, memorization of surface statistics, as represented by the attention matrix values without a structural information, enables generating legal tokens with high accuracy.

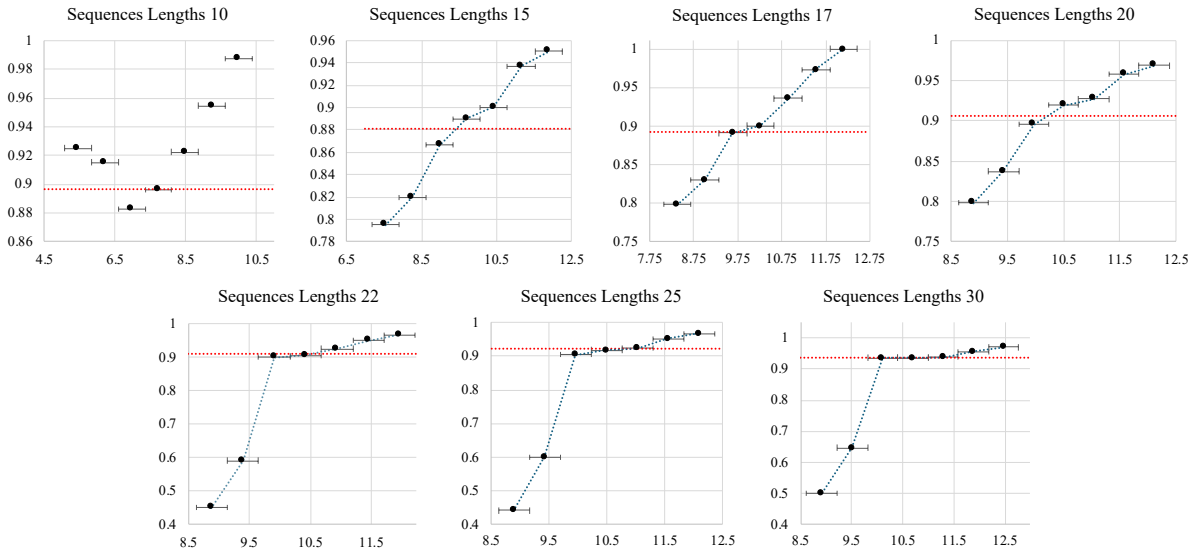

Figure 3: Legal move generation accuracy (vertical axis) as a function of structural confidence score R (horizontal axis). Horizontal limits for each point indicates interval of R in which accuracy was averaged. Horizontal dotted red line indicates average accuracy. It is evident that for sequences having length 15 or longer accuracy increases with the structural confidence score. However, there is no clear trend for sequences having length 10.

Contribution of CI Tests

Next we examine if conditional independence (CI) tests from which the causal structure is entailed provide an advantage over pair-wise correlations directly represented by elements in the attention matrix. To this end we calculate the confidence score (Equation 9) using p-values of a) all pair-wise marginal independence relations (from raw attention-matrix elements)—CI conditioning size 0, b) CI-tests having exactly one node in the conditioning set, c) all CI-tests having empty or exactly one node in the conditioning set, and d) CI-tests used to reconstruct the causal structure without limiting conditioning set sizes. The results are given in Figure 4. Let $\bar{R}_{\text{legal}}$ be the average structural confidence score of sequences for which a legal token was generated, and $\bar{R}_{\text{illegal}}$ be the average structural confidence score of sequences for which an illegal token was generated. The vertical axis represents the difference in structural confidence scores $\bar{R}_{\text{legal}} - \bar{R}_{\text{illegal}}$. Error bars indicate 95% confidence intervals (unpaired t-test). The horizontal axis indicate sequence length. It is evident that relying solely on raw attention values, case a), the difference between legal and illegal generated tokens in not statistically significant, except for sequence length 20. Relying solely on CI-test with exactly one node in the conditioning set, case b), the difference between the structural confidence is positive for all tested sequence lengths but statistically significant only for sequences lengths 17. When employing pair-wise correlations and CI tests with exactly one node in the conditioning tests, the result is statistically significant for both sequence lengths 17 and 20, implying that these two types of tests are complementary. Finally, it is evident that using all CI-tests needed to learn the causal graph, without limiting the conditioning set sizes, case d), provide the best results where sequence lengths in $[15, 22]$ are statistically significant and the difference between legal and illegal is positive, ($\bar{R}_{\text{legal}} > \bar{R}_{\text{illegal}}$) in all tested sequence lengths.

Attention Heads Pruning with Respect to Confidence Score

In this experiment we examine the importance of each attention head (in multi-head attention) for legal-move generation. We evaluate the importance of a head by the degree of confidence with which it represents a causal structure. This is different from the experiments in Section 4 and Section 4 where the average of structural confidence scores of the heads was associated with each test sequence.

Here, a structural confidence score is calculated for each attention head for each sequence in the test set. That is, for 1,000 test sequences and 8 heads in the last attention layer there is a set of 8,000 scores. This set, denoted $\boldsymbol{R}$, is sorted in an ascending order. From this sorted set, nine equally spaced values are selected as thresholds, denoted $\boldsymbol{t} = \{t_1, \ldots, t_9\}$, such that they correspond to 10%, 20%, ..., 90% percentiles. Given a threshold, t_i, for each test sequence the attention heads that have structural confidence scores lower than the threshold are pruned (skipped in the forward pass) and the next token is generated without those heads. Hence, the number of pruned heads may vary from sequence to sequence. We then calculate the legal-move generation accuracy for each threshold, that is, accuracy per pruning percentile. Note that retraining the model after pruning is not required (Voita et al. 2019). In our case, it is expected that pruning the heads with low structural confidence will have limited impact on the accuracy. To examine this, we compare the accuracy to that of a *reverse-order pruning* process. In this process we prune heads having high structural con-

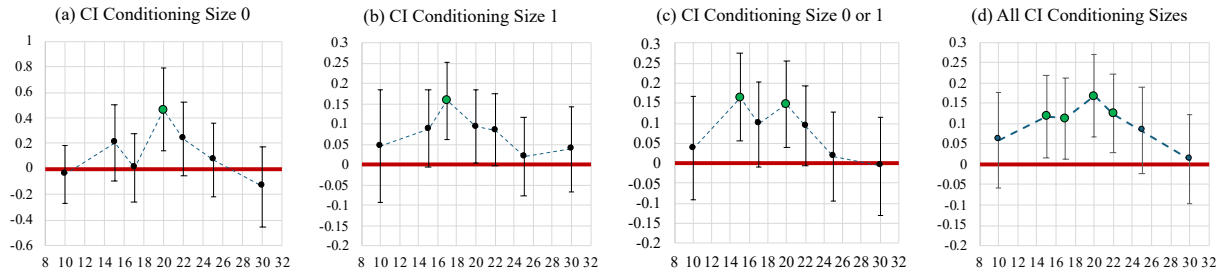

Figure 4: Average difference of structural confidence between legal and illegal move generation (vertical axis) for different input-sequence lengths (horizontal axis). Error bars are 95% confidence interval calculated using t-test. Confidence score are calculated from p-values of: (a) all unconditional (marginal) independence tests, (b) all CI tests having exactly one conditioning node, (c) only tests from both cases a) and b), (d) only CI-tests, without limiting the conditioning set sizes, needed to reconstruct a causal structure.

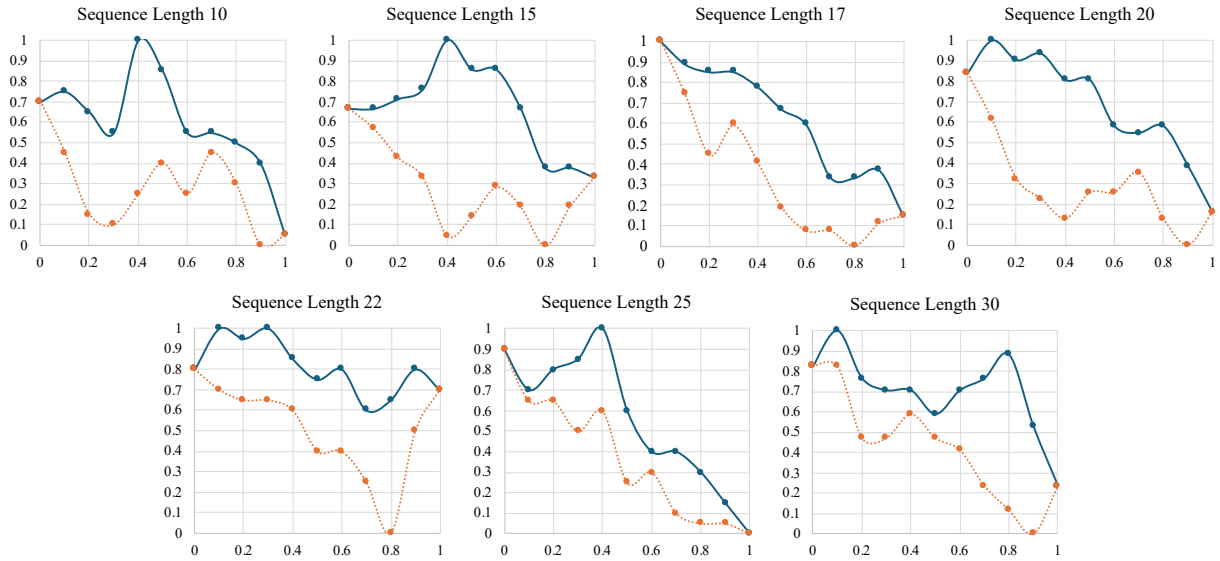

Figure 5: Normalized accuracy of legal-move generation (vertical axis) as a function of the percentage of pruned heads (horizontal axis) with respect to structural confidence. A solid blue curve corresponds to pruning the percentage of heads having the lowest structural confidence, whereas a dotted orange curve corresponds to reverse-order pruning process (pruning the percentage of heads having the highest structural confidence).

fidence scores while keeping those with lower scores. That is, we sort the set of scores, $\boldsymbol{R}$ in a descending order, and for each threshold prune the heads that have higher structural confidence scores. Under the assumption that GPT implicitly uses a causal world model to generate the next tokens, we expect pruning heads having low structural confidence scores will result in higher legal-move accuracy and larger area under curve (accuracy as a function of pruning percentile) than in the reverse-order pruning process.

In Figure 5 it is evident that pruning heads with lower structural confidence scores (solid blue curve) results in higher legal-move generation accuracy and greater area under curve, compared to removing heads with higher structural confidence scores (dotted orange curve). This demonstrates the importance of individual attention heads that encode structural information for generating legal moves.

5 Conclusions

We presented a causal interpretation of GPT that may explain the apparent emergence of world model in recent studies. Following this interpretation, we described a method that utilizes the triangular form of the attention matrices in GPT to efficiently recover the causal structures for input sequences in a zero-shot manner. Finally, using experiments in the controlled environment of the Othello board game we demonstrated that GPT implicitly learns to represent causal structures in attention heads. Specifically, in cases where the confidence in recovering structures from the attention matrices is low, GPT generally fails to generate a token that adheres to the Othello board game rules. In future work, these results may provide insights on the sources of hallucination in GPT-based models and methods for detecting them.

A Main Notations

In Table 1 we provide common symbols and their meaning used in this paper. Table 1 describes two sets of symbols. The first five rows describe symbols used for referring to entities in GPT, whereas the last four rows describe symbols used for referring to entities in SCM.

Table 1: Main notations used for the analogy between GPT and attention in SCM. The first set of symbols describes entities in GPT, and the second set describes entities in SCM.

Symbol	Description
$\boldsymbol{Z}_i$	output embedding of input symbol i, $\boldsymbol{Z}_i \equiv \mathbf{Z}(i,\cdot)$, in attention layer
$\boldsymbol{V}_i$	value vector corresponding to input i, $\boldsymbol{V}_i \equiv \mathbf{V}(i,\cdot)$, in attention layer
$\mathbf{A}$	attention matrix
$\mathcal{T}$	Transformer neural network
$\mathbf{W}_V, \mathbf{W}_{QK}$	learnable weight matrices in GPT
X_i	a random variable representing node i in an SCM
U_i	latent exogenous random variable i in an SCM
$\mathbf{G}$	weighted adjacency matrix of an SCM
$\mathcal{G}$	causal graph structure

B Difference between Test and Training

The data used to train the GPT model was real-world sequences of Othello game moves (Li et al. 2023). These moves were played in a strategic manner with an intention of winning the game. In contrast, the experiments in the paper were conducted using a test data consisting of randomly generated sequences of moves that adhere with the game rules, without considering the outcome of the game.

We measure the difference between distributions of sequences in the training dataset, $\boldsymbol{D}^{\text{train}}$, and test dataset, $\boldsymbol{D}^{\text{test}}$, by estimating n-gram frequencies. For a given sequence, $\{t_0, \ldots, t_{\ell-1}\}$, we extract the last n tokens assuming that the probability of the next generated token t_ℓ depends only on these n tokens,

$$P(t_\ell | t_0, \ldots, t_{\ell-1}) = P(t_\ell | t_{\ell-n}, \ldots, t_{\ell-1}). \tag{11}$$

For the i-th sequence in the test set, trimmed to length ℓ, we count the number of occurrences $N_n^{\text{test}|\text{train}}(i)$ of the n-gram $\{t_{\ell-n}, \ldots, t_{\ell-1}\}$ of the test sequence in the training data sequences, trimmed to length ℓ, and divide by the number of training sequences, $|\boldsymbol{D}^{\text{train}}|$. We estimate the mean μ_n^{test} by averaging over the number of test sequences, $|\boldsymbol{D}^{\text{test}}|$,

$$\mu_n^{\text{test}|\text{train}} = \frac{1}{|\boldsymbol{D}^{\text{test}}|} \sum_i \frac{N_n^{\text{test}}(i)}{|\boldsymbol{D}^{\text{train}}|}. \tag{12}$$

Similarly, using sequences excluded from the training data we estimate $\mu_n^{\text{train}|\text{train}}$, the percentage of occurrences of n-grams of training sequences in the training data sequences. For each sequence length evaluated in the paper, $\ell \in \{15, 17, 20, 22, 25, 30\}$, we calculate percentage of n-gram occurrences for $n \in [2, \ldots, 6]$. We compare the percentage of occurrences $\mu_n^{\text{test}|\text{train}}$ and $\mu_n^{\text{train}|\text{train}}$ in Figure 6. From this evaluation it is evident that the distribution of real-world sequences played with an intention of winning ($\boldsymbol{D}^{\text{train}}$) is different from that of randomly generated sequences ($\boldsymbol{D}^{\text{test}}$) used in the paper to examine the trained GPT model.

C Recursive Causal Discovery from GPT Attention

We described our method in Algorithm 2, where, given an input sequence, a causal structure is learned from an attention matrix in the last layer. In this section, we describe line 8 in more detail where the ICD algorithm (Rohekar et al. 2021), modified to learn only a set of given edges, is called. The operations in line 8 are mostly similar to operations in the ABCD algorithm (Rohekar, Gurwicz, and Nisimov 2024). The main difference is that this step refines a partially learned causal structure, by testing conditional independence between pairs of nodes connected by edges in a given list $\boldsymbol{E}$.

The operations in line 8 of Algorithm 2 are as follows. First, covariance is estimated from an attention matrix $\mathbf{A}$,

$$\mathbf{C} = \left[\mathbf{D}^{-1}\mathbf{A}\right]\left[\mathbf{D}^{-1}\mathbf{A}\right]^\top, \tag{13}$$

where $\mathbf{D} \equiv \text{diag}(\mathbf{A})$ is a diagonal matrix consisting of elements on the diagonal of $\mathbf{A}$ such that $\mathbf{D}^{-1}\mathbf{A}$ is a unitriangular matrix. Then, a correlation matrix is estimated

$$\mathbf{R} = \text{diag}(\mathbf{C})^{-1/2}\ \mathbf{C}\ \text{diag}(\mathbf{C})^{-1/2}. \tag{14}$$

Conditional independence between two variables X and Y conditioned on set $\boldsymbol{Z}$ is estimated by calculating the partial correlation from $\mathbf{R}$. Then, let $\text{Ind}\,(X, Y|\boldsymbol{Z})$ be a CI test based on partial correlation, where p-values are estimated using Fisher z-transform. Finally, call ICD to learn a set of edges using Ind. In Algorithm 3 we provide a simple modification of ICD such that it learns only the edges in $\boldsymbol{E}$ and uses a given initial graph. In red we strike out parts of the ICD and in blue are our additions. The rest of the pseudo code is exactly as given by Rohekar et al. (2021). As input, we add the initial graph $\mathcal{G}$ to be used and further refined, and add the set of edges $\boldsymbol{E}$ to be learned (remove edges connecting conditionally independent nodes). In line 1 we remove the initialization of a complete graph as the initial graph is given as input. In line 3 and line 6, we add the set of edges $\boldsymbol{E}$ to be tested to the ICD iteration function. Lastly, in line 8 only edges in $\boldsymbol{E}$, rather than all edges in $\mathcal{G}$ are tested.

Overall, utilizing the causal order, enforced by the triangular form of the GPT attention matrix, each recursive call assumes that the current graph is the final learned graph except for the edges connecting the newly added node to the rest of the graph nodes (edge list $\boldsymbol{E}$). Note that this does not violate ICD-Sep conditions (Rohekar et al. 2021) that needs to be complied for having a sound and complete causal discovery algorithm. By considering only the edges connecting a node to its predecessors in the given causal order, a significantly lower number of CI tests are required for learning the causal graph compared to the unmodified ICD algorithm.

Algorithm 2: Causal Discovery for GPT

Input: $\boldsymbol{S}$: a sequence of tokens $\{t_1, \ldots, t_n, \}$

Output: $\mathcal{G}$: a partial ancestral graph (PAG)

Function `LearnStructure(`$\boldsymbol{S}$`)`**:**

- if $|\boldsymbol{S}| = 1$ then return a graph with the single node in $\boldsymbol{S}$
- $t_n, \boldsymbol{S}' \leftarrow \text{pop}(\boldsymbol{S})$
- $\mathcal{G}' \leftarrow$ `LearnStructure(`$\boldsymbol{S}'$`)`
- $\mathcal{G} \leftarrow \mathcal{G}' + \{t_n\}$
- set $\boldsymbol{E}$ to the set of edges (circle edge-marks) between t_n and every node in $\mathcal{G}'$
- connect $\boldsymbol{E}$ in $\mathcal{G}$
- test CI for edges in $\boldsymbol{E}$ and orient $\mathcal{G}$ using `ICD` (Rohekar et al. 2021)
- return $\mathcal{G}$

Algorithm 3: Modified ICD (Rohekar et al. 2021) algorithm

Input:
Ind: a conditional independence oracle
$\mathcal{G}$: initial PAG
$\boldsymbol{E}$: set of edges to be learned

Output:
$\mathcal{G}$: a PAG

initialize: $r \leftarrow 0$, ~~$\mathcal{G} \leftarrow$ a complete graph with 'o' edge-marks,~~ and $done \leftarrow$ False

while $(r \leq n)$ & $(done =$ False$)$ **do**
- $(\mathcal{G}, done) \leftarrow$ `Iteration(`$\boldsymbol{E}$, $\mathcal{G}$, r`)` ▷ `refine` $\mathcal{G}$ `using conditioning sets of size` r
- $r \leftarrow r + 1$

return $\mathcal{G}$

Function `Iteration(`$\boldsymbol{E}$, $\mathcal{G}$, r`)`**:**

- $done \leftarrow$ True
- **for** *edge* (X, Y) *in* $\boldsymbol{E}$ ~~edges($\mathcal{G}$)~~ **do**
  - $\{\mathbf{Z}_i\}_{i=1}^{\ell} \leftarrow$ `PDSepRange` $(X, Y, \text{r}, \mathcal{G})$ ▷ $\mathbf{Z}_i$ `complies with ICD-Sep conditions`
  - **if** $\ell > 0$ **then**
    - $done \leftarrow$ False
    - **for** $i \leftarrow 1$ **to** ℓ **do**
      - **if** $\text{Ind}(X, Y | \mathbf{Z}_i)$ **then**
        - remove edge (X, Y) from $\mathcal{G}$
        - record $\mathbf{Z}_i$ as a separating set for (X, Y)
        - **break**
- orient edges in $\mathcal{G}$
- **return** $(\mathcal{G}, done)$

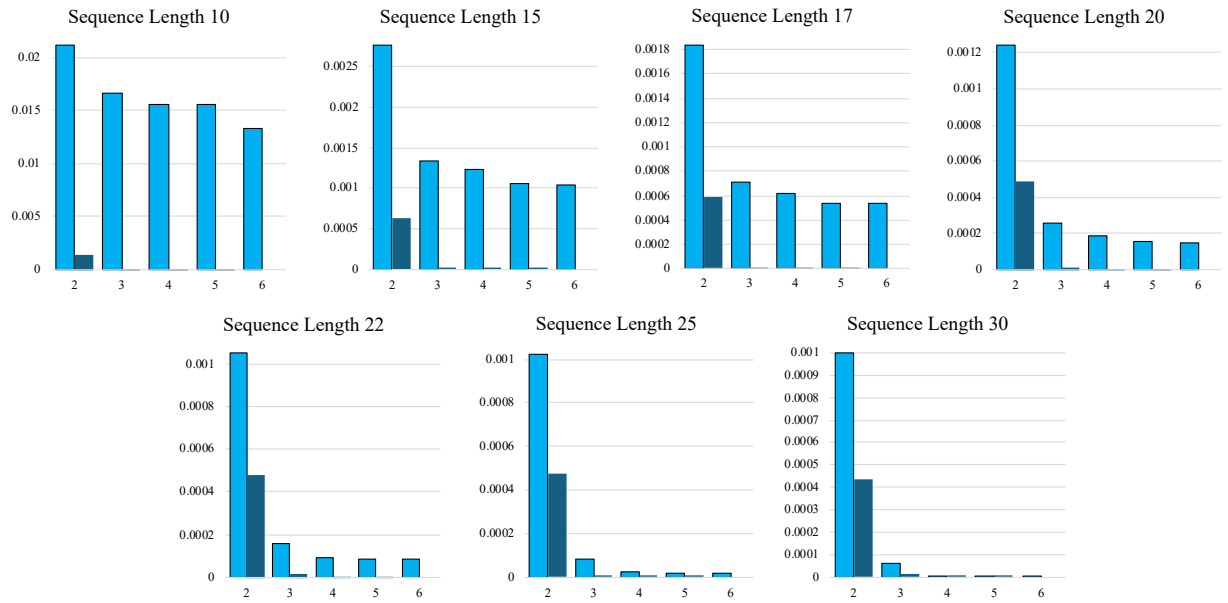

Figure 6: Percentage of occurrences (vertical axis) of n-grams from test and training sequences in the training data for $n \in [2, \ldots, 6]$ (horizontal axis). Light blue columns are $\mu_n^{\text{train}|\text{train}}$, and dark blue are $\mu_n^{\text{test}|\text{train}}$ values. The clear difference between $\mu_n^{\text{test}|\text{train}}$ and $\mu_n^{\text{train}|\text{train}}$ which indicates a clear difference between the distributions of real-world sequences used to train the GPT model and randomly generated sequences used for evaluation.

References

Chowdhery, A.; Narang, S.; Devlin, J.; Bosma, M.; Mishra, G.; Roberts, A.; Barham, P.; Chung, H. W.; Sutton, C.; Gehrmann, S.; et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240): 1–113.

Claassen, T.; Mooij, J. M.; and Heskes, T. 2013. Learning Sparse Causal Models is not NP-hard. In *Uncertainty in Artificial Intelligence*, 172. Citeseer.

Colombo, D.; Maathuis, M. H.; Kalisch, M.; and Richardson, T. S. 2012. Learning high-dimensional directed acyclic graphs with latent and selection variables. *The Annals of Statistics*, 294–321.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL-HLT*, 4171–4186.

Li, K.; Hopkins, A. K.; Bau, D.; Viégas, F.; Pfister, H.; and Wattenberg, M. 2023. Emergent World Representations: Exploring a Sequence Model Trained on a Synthetic Task. In *The Eleventh International Conference on Learning Representations*.

Liu, H.; Li, C.; Wu, Q.; and Lee, Y. J. 2024. Visual instruction tuning. *Advances in neural information processing systems*, 36.

Nanda, N.; Lee, A.; and Wattenberg, M. 2023. Emergent Linear Representations in World Models of Self-Supervised Sequence Models. *EMNLP 2023*, 16.

Nisimov, S.; Gurwicz, Y.; Rohekar, R. Y.; and Novik, G. 2021. Improving Efficiency and Accuracy of Causal Discovery Using a Hierarchical Wrapper. In *Uncertainty in Artificial Intelligence (UAI 2021), the 4th Workshop on Tractable Probabilistic Modeling*.

Nisimov, S.; Rohekar, R. Y.; Gurwicz, Y.; Koren, G.; and Novik, G. 2022. CLEAR: Causal explanations from attention in neural recommenders. *arXiv preprint arXiv:2210.10621*.

Pearl, J. 2009. *Causality: Models, Reasoning, and Inference*. Cambridge university press, second edition.

Pearl, J.; and Mackenzie, D. 2018. *The book of why: the new science of cause and effect*. Basic books.

Peters, J.; Janzing, D.; and Schölkopf, B. 2017. *Elements of Causal Inference: Foundations and Learning Algorithms*. Cambridge, MA, USA: MIT Press.

Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I.; et al. 2018. Improving language understanding by generative pre-training.

Richardson, T.; and Spirtes, P. 2002. Ancestral graph Markov models. *The Annals of Statistics*, 30(4): 962–1030.

Rohekar, R. Y.; Gurwicz, Y.; and Nisimov, S. 2024. Causal Interpretation of Self-Attention in Pre-Trained Transformers. *Advances in Neural Information Processing Systems*, 36.

Rohekar, R. Y.; Gurwicz, Y.; Nisimov, S.; Koren, G.; and Novik, G. 2018. Bayesian structure learning by recursive bootstrap. *Advances in Neural Information Processing Systems*, 31.

Rohekar, R. Y.; Nisimov, S.; Gurwicz, Y.; and Novik, G. 2021. Iterative Causal Discovery in the Possible Presence of Latent Confounders and Selection Bias. *Advances in Neural Information Processing Systems*, 34: 2454–2465.

Schaeffer, R.; Miranda, B.; and Koyejo, S. 2024. Are emergent abilities of large language models a mirage? *Advances in Neural Information Processing Systems*, 36.

Schmidhuber, J. 1992. Learning to control fast-weight memories: An alternative to dynamic recurrent networks. *Neural Computation*, 4(1): 131–139.

Spirtes, P.; Glymour, C.; and Scheines, R. 2000. *Causation, Prediction and Search*. MIT Press, 2nd edition.

Team, G.; Anil, R.; Borgeaud, S.; Wu, Y.; Alayrac, J.-B.; Yu, J.; Soricut, R.; Schalkwyk, J.; Dai, A. M.; Hauth, A.; et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Toshniwal, S.; Wiseman, S.; Livescu, K.; and Gimpel, K. 2022. Chess as a testbed for language model state tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 11385–11393.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Voita, E.; Talbot, D.; Moiseev, F.; Sennrich, R.; and Titov, I. 2019. Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned. *In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, 1–113.

Yehezkel, R.; and Lerner, B. 2009. Bayesian network structure learning by recursive autonomy identification. *Journal of Machine Learning Research (JMLR)*, 10(Jul): 1527–1570.

Zhang, J. 2008. On the completeness of orientation rules for causal discovery in the presence of latent confounders and selection bias. *Artificial Intelligence*, 172(16-17): 1873–1896.