Probing Emergent World Representations in Transformer Networks: Sequential Models Trained to Play Othello

JEFFREY CHIU, DEAN HAZINEH, and ZECHEN ZHANG

At a high level, this project aims to extend previous work on the interpretability of transformers used in sequence generation. Foundation models are shown to be extremely powerful in common-sense reasoning, logical deductions, etc., but do they have an actual understanding of the world or are they simply stochastic parrots? In this project, we consider a toy example to find out. Specifically, we delve deeper into investigations regarding "emergent world representations" in attention-based models trained to play the board game, Othello. The goal is to understand if trained sequence models develop an internal representation of the true generative process underlying the training data or if the next-token predictions rely purely on memorized token statistics. All code is made accessible at https://github.com/DeanHazineh/Emergent-World-Representations-Othello.

Additional Key Words and Phrases: Emergent World, GPT, Learned Representations, Transformer Circuit Theory

1 INTRODUCTION

Auto-regressive language models trained to predict the "next word" in a sequence have demonstrated immense promise in complex tasks ranging from solving logical word puzzles to playing board games. It has consequently been debated whether or not this performance can be solely attributed to memorizing “surface statistics”, i.e., correlations between tokens which do not directly match the causal process used to generate the original sequence. Alternatively, several recent works have investigated if such language models naturally develop interpretable *world models*. This term broadly refers to the encoding of semantic information about the true causal process, referred to throughout as a **world representation**, in the activations of the network. Some examples of world representations include the location and type of game pieces for a language model trained to play chess [8] and classification of object parts for a visual transformer trained for segmentation [1]. Understanding if these models do in fact learn to develop internal world representations in an unsupervised manner remains an important question and can provide clues to the generalizability of deep networks as a whole.

The heart of our project is a set of interpretability studies investigating this emergent world principle in transformer-based models, extending the work recently presented by Kenneth Li et al., [3]. In the original paper, the authors trained an auto-regressive model, structurally similar to the GPT-2 architecture, to play the board game Othello (we provide a summary of the game and its rules in appendix A). The model predicts the next game move given a transcript of previously played moves, passed in as a list of board coordinates, e.g., "c4 c3 e6 ...". While the trained model is unsurprisingly very successful in learning to play the game, the focus and contribution of their work is investigating *how* it forms its predictions. Specifically, they show that the trained model develops an internal understanding that the transcript of played moves corresponds to a particular arrangement on a game board, without being told about such concept. Moreover, they provide initial evidence that the internal world representation is causally connected to the model’s predictions.

We continue investigation of sequential models trained for the task of Othello since it serves as an interesting testbed for interpretability studies. Notably, the rules of play are both few and simple and it is likely that discerning the representation of a board and the rules is the most efficient way to generalize and play legal games without pure memorization. The focus of this report is exploring the extent to which we can determine if the model learns

Authors’ address: Jeffrey Chiu, jeffrey_chiu@college.harvard.edu; Dean Hazineh, dhazineh@g.harvard.edu; Zechen Zhang, zechen_zhang@g.harvard.edu.

to use these principles when making its next-token predictions. We consider two different approaches/theories to interpretability: in Section 2, we leverage a priori principles of a “reasonable” world representation and interrogate the model for evidence that such representations are encoded and/or used. The techniques used in this section closely parallel those used in the original work, although we adapt a different world model and demonstrate new insights. Alternatively, in Section 3, we apply the theory of transformer circuits which does not require a pre-concieved understanding of the task and instead probes the decision logic by directly visualizing the trained weights inside the model. To the best of our knowledge, the application of transformer circuit theory to this problem is new. Circuits are defined as meaningful connections in models thats correspond to real world meaningful algorithms and were first studied in [5] in Vision transformer texts. By looking at Transformer circuits, we are able to better understand the role of the Attention modules in our Transformer network.

Necessary technical background is reviewed in each section. We summarize our findings as follows:

(1) We observe that reasonable performance on a task can emerge without any apparent utilization of an emergent world representation. For models that are deep (referring to more than one layer), we find that world representations may be linearly encoded in the model’s activations to an extent that generally increases with layer depth. Encoding the world representation in a particular layer, however, does not seem to guarantee that it is influential to the model’s predictions. Instead, it appears that semantic understanding in the model is developed and utilized about halfway through the model.

(2) We find regularities in the attention heads, which can be categorized into "my-turn heads" and "your-turn heads" in shallow networks. Furthermore, "first-token head" and "last-token heads" begin to emerge in the last layers of deep networks means and they don’t do much to process information for the prediction. This is consistent with the probe classification findings. The OV circuit seems to be a random matrix, which remains a mystery.

2 PROBING THE EMERGENCE OF A PARTICULAR WORLD REPRESENTATIONS

2.1 Background and Motivation

Given a task such as playing legal moves in Othello, we can define reasonable world representations by considering our own decision process and/or our prior knowledge of the rules used to generate the training dataset. In this case, one natural semantic representation for the task is the state of the game board given a sequence of previously played moves, defined by a classification of each of the 64 tiles as empty, occupied by a white disc, or by a black disc (visualized in the bottom panel of Figure 1). We may then interrogate if our sequential model, pre-trained to play legal moves, has naturally learned this particular world representation by investigating if it is encoded within the activations of the model. This study was conducted by the authors in [3] and although intuitive, they found that this representation is not *linearly* encoded within the model, i.e. it cannot be extracted by a linear transformation of the activation vectors. Instead, they demonstrated that this particular world representation is encoded non-linearly and can be extracted from the activation vectors with up to 98% accuracy using a two layer MLP. Moreover, they showed that replacing an activation vector with a new activation vector corresponding to a board state with a single tile flipped to another color produces causal and predictable changes to the next token predictions of the model. Their intervention process is depicted in Figure 1b.

While we are inspired by this work, we introduce the following set of questions/thoughts which we attempt to answer in subsections 2.2-2.3 by extension.

- Since the models next-move predictions are obtained by a linear transformation of the final layer’s activation, we are inclined to believe that a world representation learned and utilized by a model should also be linearly encoded in the activation. We are also cautious that a two-layer MLP is expressive enough

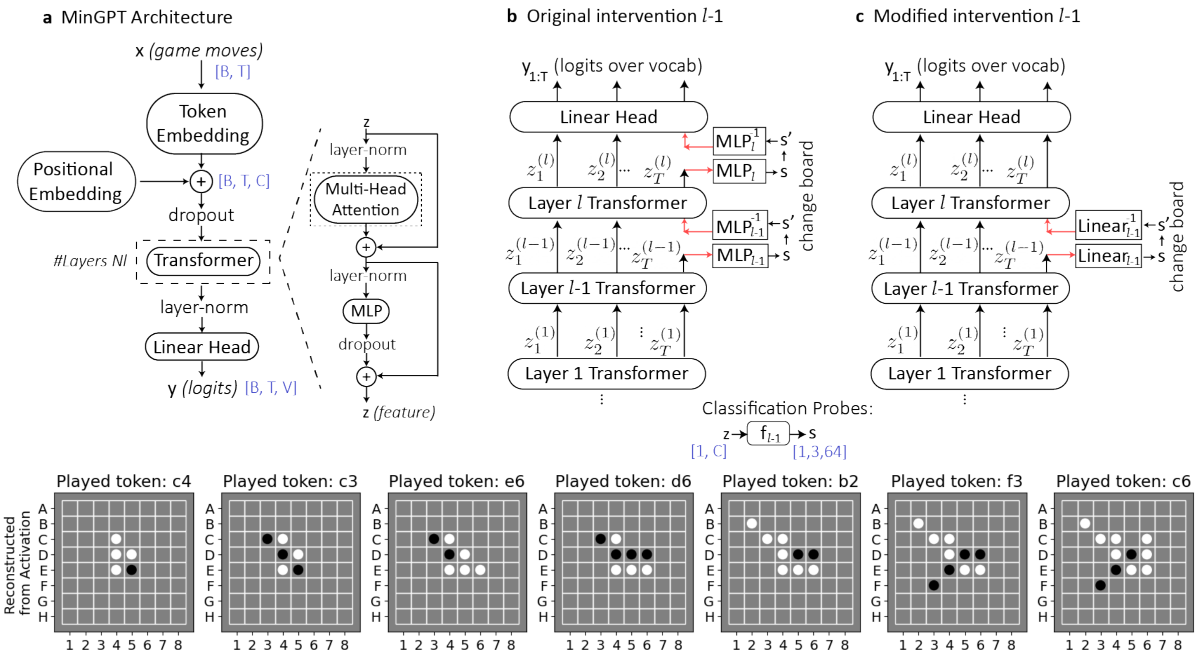


Fig. 1. Overview of the principles in section 2. (a) The neural architecture utilized in this project and the original work, where the number of layers refers to the number of transformer blocks. (b) The original intervention scheme is replaced by an alternate version shown in (c), whereby intervention is applied to a single layer–see text for details. (Bottom Panel) Example of an extracted world representation where the game board state is obtained from the activation vectors.

to hallucinate the principle of an emergent world in contrast to a simple linear transformation. We then raise the question, is there a world representation that is encoded linearly and can we show that it fully defines the model's next-move logits.

- In their intervention method (Figure 1b), they required that all activation vectors starting from a particular layer be replaced to show causality. We hypothesize that this may be related to the particular choice of world representation. By finding a world representation with a linear encoding, can the intervention scheme be simplified to that shown in Figure 1c? In doing so, we may then directly investigate if the influence of the world representation to the model's predictions differs with respect to the model layer.
- Lastly, we raise the new question, does the emergence of a world representation depend on the depth or complexity of the model, specifically the number of sequential attention blocks or the number of heads in the model? If a particular model can be shown to contain a meaningful world representation, does a scaled down version of that model, trained in the same way, lose this possible source of generalizability?

2.2 The Choice of a World Representation with a Linear Encoding

In this section, we first report our findings that there is an alternate but intuitive world representation that is linearly encoded within the activation vectors of the trained sequential model. In addition, we also show that changes to the activation vectors prescribed by tile changes in the world representation produce causal and predictable changes to the next-move predictions when utilizing our simpler intervention scheme (Figure 1c).

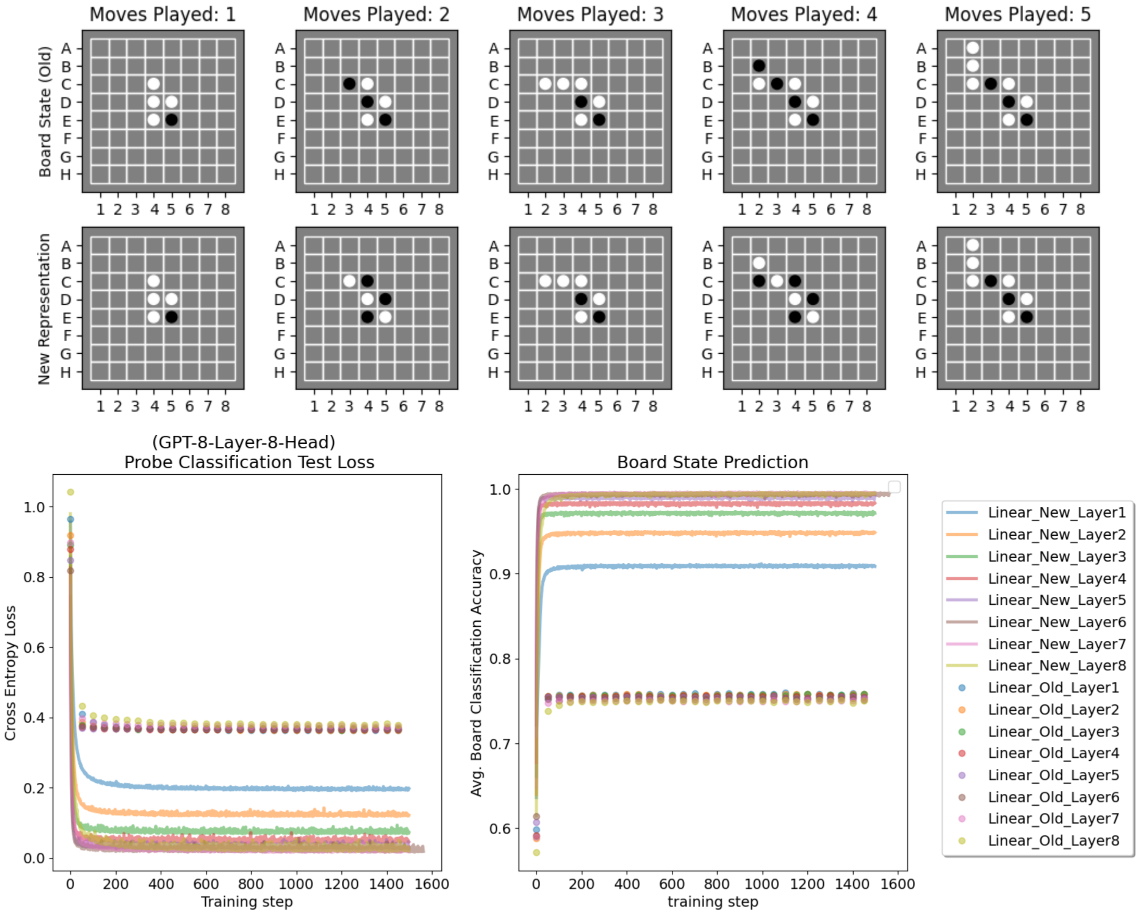


Fig. 2. The old vs new world representation discussed in the text can be visualized as the two sets of board states for a given sequence, shown in the first two rows. In the new representation, we are interested in extracting the information of “my piece” vs “your piece” (displayed as black or white). Linear probes are trained to extract the representations from the activations of the frozen sequential model, with loss and accuracy shown in the bottom.

This is visualized utilizing latent saliency diagrams, discussed later. In subsection 2.3, we present a different form of visualization and show that the next move logits can in fact be completely determined and controlled by the world representation, providing additional support for the principle of a world model. We note that the observation of a linearly encoded world representation for this problem was made first by another researcher via a blog post [4] and directly influenced our course of investigations. We, however, utilize a slightly different implementation although the fundamental representation is the same[1].

[1] In Neel Nanda’s implementation, he learns a different linear mapping between the activation vector and the board state for moves played on even and odd turns. In our case, we learn a single linear mapping but define a the world representation to directly be empty, the tile corresponding to the current players turn, and the tile for the other player. Noting that the first token in the sequence corresponds to white and the remaining tokens in the input alternate between white and black’s move, the representation of Neel and us are equivalent.

The original work postulated a world representation corresponding to the placement of white or black discs on a game board. While this is how most humans would visualize the board when playing the game of Othello (we suspect), it is important to note that the sequential model differs slightly from human play in that the model alternates between acting as the white vs black player depending on the length of the input sequence. As the rules dictating a legal move require only an understanding of which piece is yours and the opponents, a more natural choice for a world representation in this case is instead the classification of a tile as containing "my piece" vs "your piece" (or empty). The two different world representations for a set of game moves are visualized in Figure 2 (in the new representation, one may also interpret it as the model always playing the same color but with tiles already on the board inverting in color depending on the sequence length). We observe that this simple change makes a substantial difference in interpreting emergent world behaviors, suggesting that guessing the right "world representation" plays a crucial role in this form of interpretability analysis.

To quantify this, we first train a multi-headed attention model (diagrammed in Figure 1a) with 8 transformer blocks, referred to as layers, and 8 attention heads on the synthetic Othello dataset released by the original authors. This architecture is chosen as it matches the model discussed in their paper. Once trained to predict legal next-moves, the model's weights are frozen. We then train a separate linear transformation[2] (referred to throughout as a linear probe) to map the activation vector of a particular layer to the world representation, i.e., the mapping $f : \mathbb{R}^{C=512} \rightarrow \mathbb{R}^{64x3}$ where C is the token embedding dimension and the output corresponds to the three-state classification logits for each of the 64 tiles. All probes are trained on a reduced size dataset of over 1 million games (with each game providing 59 activation-board state training pairs) using an Adam optimizer and a fixed learning rate. The accuracy for the linear probe at each layer and for classification to the old vs new world representation is shown in Figure 2 and in Table. 1. Similar to the sequential model, the probe weights are frozen once trained. We observe that the new representation is linearly encoded in the activation vectors, particularly within the deeper layers, and can be recovered from the model's activations by a simple linear transformation with near unity accuracy on a withheld set.

Table 1. Classification Accuracy for Linear Probes Mapping $z \longrightarrow s$

Layer:	1	2	3	4	5	6	7	8
old	75.7%	75.8%	75.7%	75.7%	75.6%	75.4%	74.9%	74.9%
new	90.8%	94.8%	97.1%	98.3%	99.1%	99.5%	99.5%	99.5%

We now show that the model's next move predictions can be understood and interpreted in terms of this linearly encoded world representation. Here, we demonstrate this fact by producing latent saliency plots as was done for the non-linear case in the original paper. We note that while the trained linear probes can extract the representation from the activation vectors in layers 5-8 with similar accuracy, we have empirically observed that the quality of the latent saliency plots differ noticeably depending on the layer choice. We discuss this in more detail in subsection 2.3, but for now we focus only on layer 6 and display the diagrams for three different board states in Figure 3.

The latent saliency plots are obtained for a given game sequence by first computing the model logits for a particular tile t_{inspect}, i.e., the logits for a particular token in the vocabulary. We then consider an intervention where the last activation vector for the chosen layer z is intercepted and passed through the pre-trained (frozen) linear probe to obtain the board state representation s. Given this board state, we then manually edit the vector s to get an alternative game board s' which corresponds to changing the color of one of the previously played

[2] simply torch.nn.Linear(C=512, 64x3=192, bias=True)

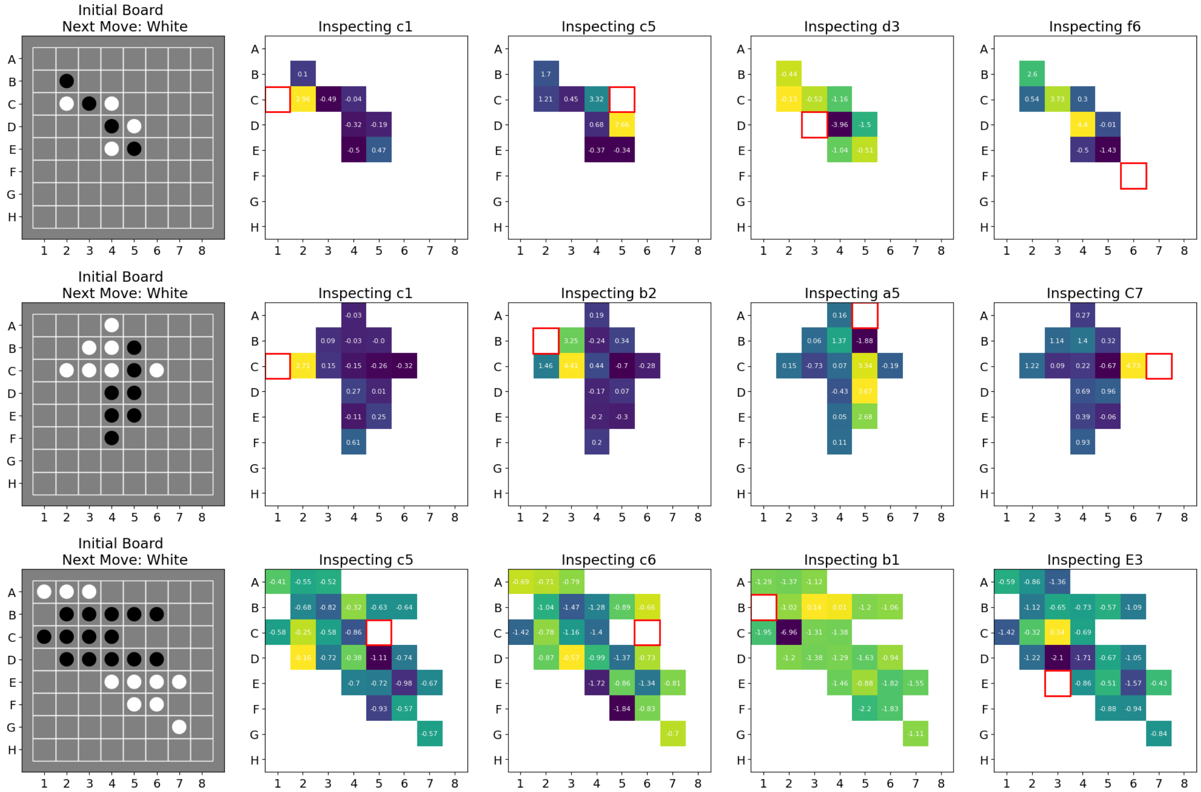


Fig. 3. Latent saliency plots for three game sequences of different lengths (each row). $t_{inspect}$ is highlighted with a red box–see text for calculation details. If a possible next move is currently illegal, we observe that interventions which would make the move legal produce a positive change (bright yellow); alternatively, interventions that make a currently legal move now illegal have a negative change (dark blue), thus supporting the argument that the model utilizes the world representation when making predictions.

discs. To get the inverse mapping from the modified board state to the modified activation vector $s' \longrightarrow z'$, we compute an approximate solution on-the-fly by utilizing gradient descent to solve the minimization problem

$$z' = \arg\min_{z^*} \mathcal{L}\left(\text{Linear}(z^*), s'\right). \tag{1}$$

Once z' is obtained, the original activation vector is manually replaced and then the next-move logit for tile t_{inspect} is recomputed. The change in the logit value is recorded for all possible disc interventions and the latent saliency plot then shows which intervention has the greatest affect to discourage or encourage a particular next-move prediction by the model. By visual inspection of the latent saliency plots, we qualitatively observe that interpretable interventions dictated by the world representation generally produce intuitive changes.

2.3 The Influence of Model Size and Complexity

Given a pre-trained sequential model, we can assert that a world representation is formed if the representation can be extracted from the model's activations by a linear transformation. We also assert that such world representation

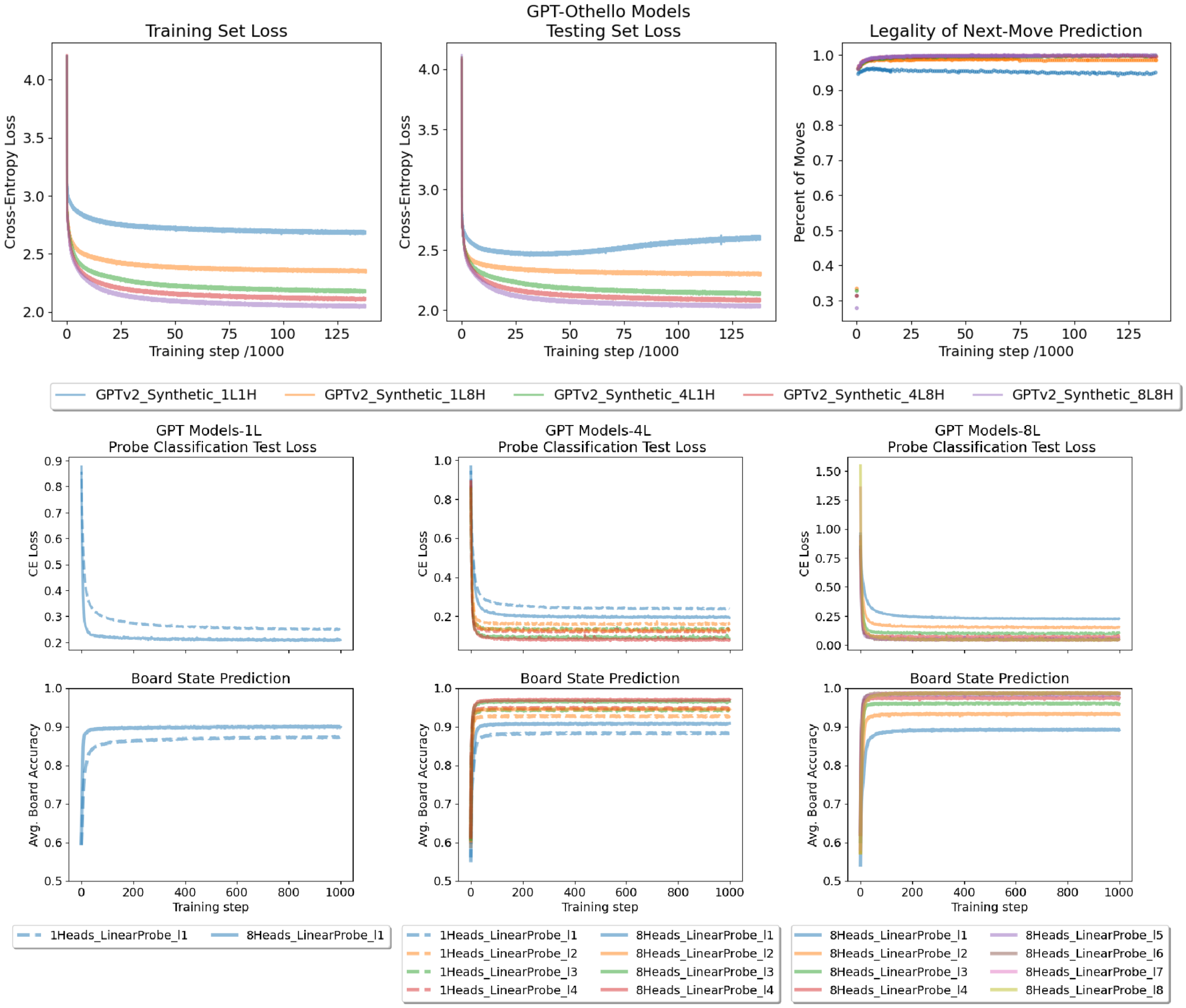


Fig. 4. (First Row) Losses for various sequential models trained to play legal moves in Othello. The average percent of legal next-moves played on a validation set are [94.9%, 98.6%, 99.7%, 99.6%, 99.9%] for the models in the legend. (Bottom rows) Training losses for the linear probes–each GPT-Othello model has a linear probe trained for each of its layers.

is utilized by the model to predict the next token if the next-token logits change predictably with respect to interpretable interventions that are guided by the world representation. Leveraging these ideas, we now attempt to answer if models of different sizes have similar potential to develop an emergent world representation in an unsupervised form and if the utilization of the world representation varies across the model's layers.

To this end, we first train a set of new sequential models to play legal moves in Othello, again based on the GPT-2 architecture (Figure 1a) but with a number of attention blocks (layers) between 1 and 8 and with either

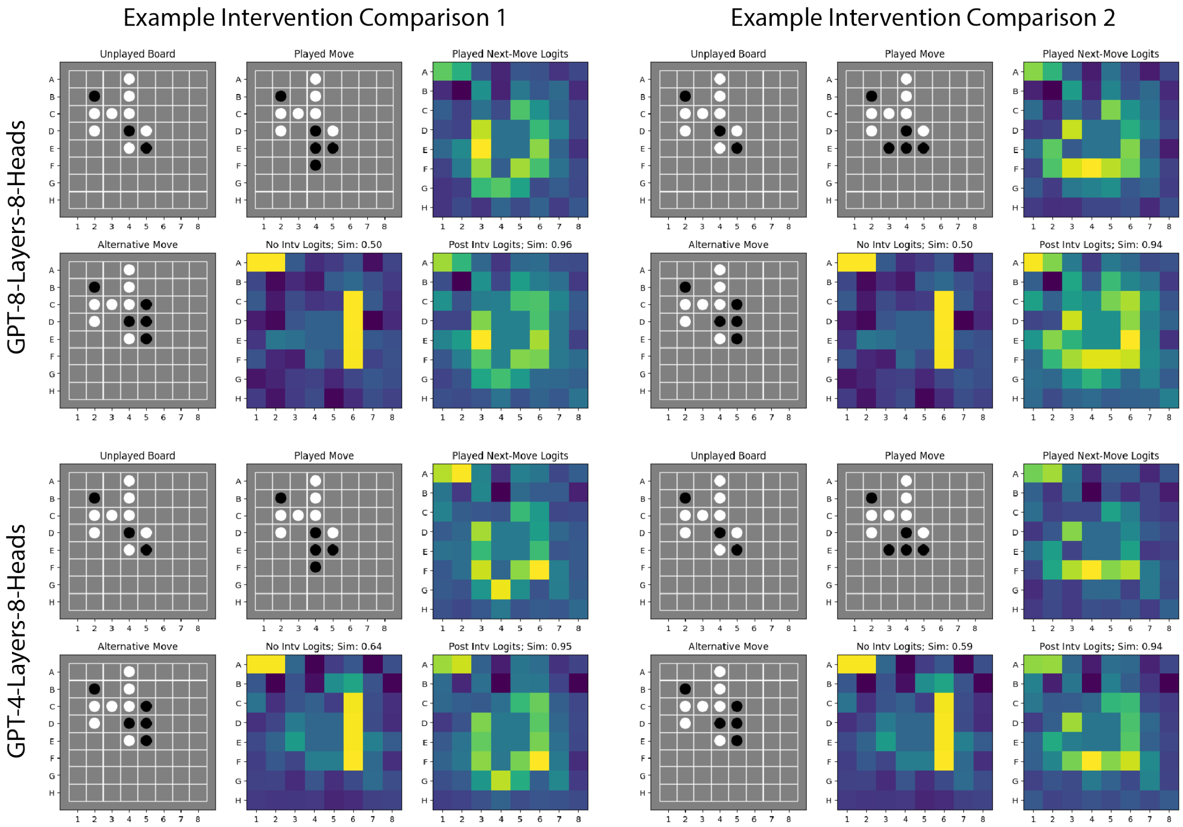


Fig. 5. Examples of new interventions involving more complicated changes to the activation vectors. Starting with an "unplayed board" (displayed in the top left of each set), we obtain two different game sequences, the played-move sequence and the alternate-move sequence. The interventions analysis begins by passing in the alternate-move sequence to the model but then modifying the activation vectors in the model to *trick* the model into believing that the played-move sequence was actually passed in. The logits with and without intervention are shown (bottom right two plots in each set) along with the "ground truth", i.e. the logits if the "played move" was passed in (top right plot of the set).

1 or 8 attention heads[3]. The models are trained on a dataset of over 20 million simulated games (where legal moves are played but without strategy) and the training losses are shown in Figure 4 (top row). During training, we periodically benchmark the models performance by computing the percent of predicted next-moves that are legal for a validation set (500 game sequences/29,500 move predictions). Notably, we find that even the smallest model tested, GPT 1-Layer 1-Head, learns to play legal moves approximately 95% of the time, while the larger models predict legal next-moves over 99% of the time. Next, we then train linear probes that map the activation vector to the new world representation for each layer and each model (see the discussion in subsection 2.2), with the training results shown in Figure 4 (bottom rows). The findings are similar to before in that the mapping is most accurate for deep layers rather than shallow (see for example the layer 1 probe vs the final

[3] Our naming convention is slightly misleading. We refer to GPT models in the sense that our trained neural architectures are similar to the GPT-2 architecture. We are not utilizing "pre-trained transformers" since all models are randomly initialized and trained from scratch.

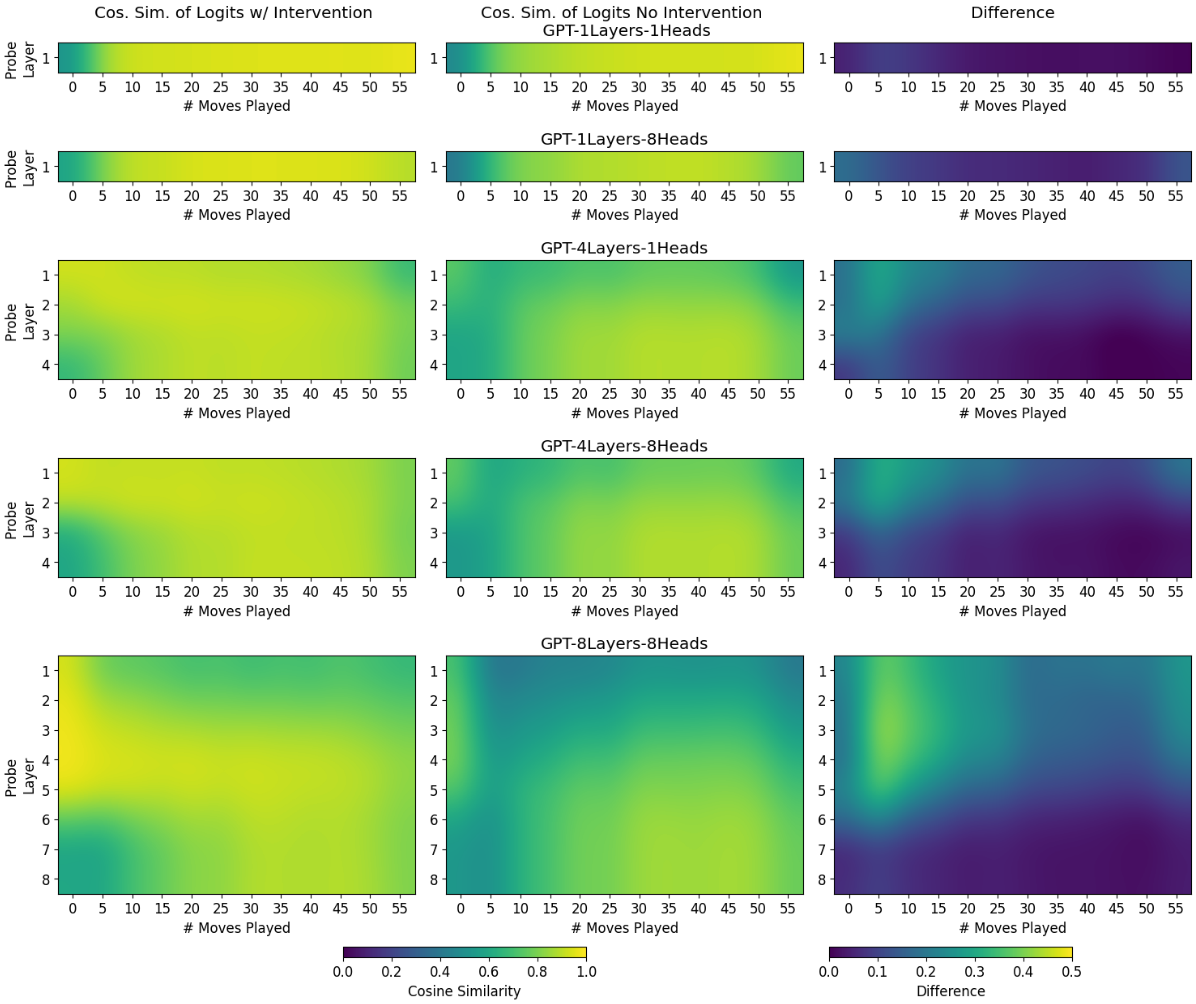


Fig. 6. Sweep of interventions like those in 5 where the data displayed is the cosine similarity of two logit vectors (ground truth vs intervention logit or ground truth vs no intervention logits), averaged over many games. Interventions are conducted for different probe layers and for game sequences of different lengths. The rightmost column displays the difference of the first and second column and reveals where/when the interventions are most impactful to the model's next-move predictions.

layer probe). Interestingly, it appears that all Othello GPT-models tested present a similar ability to encode the world representation for a given layer; for example, the first layer probe accuracy and losses are similar for the GPT-models with 1, 4, and 8 layers.

We now demonstrate that the world representation can fully prescribe and control the model's next-move predictions (beyond the single disc flip of the original work). Moreover, we discuss findings that indicate the world representation is possibly not used in the final layers of the model although it is encoded there to the greatest extent. To do this, we introduce the following intervention analysis: Given a particular game sequence at time $t-1$

(referred to as the unplayed sequence), we consider two new sequences corresponding to two different selections for the move played at time t (referred to as the played-move sequence and the alternate-move sequence). Given either of the two sequences, the GPT-model's next-move logits for time $t+1$ can be computed as normal. We then ask: if the alternate-move sequence is passed in to the model as an input, can we trick the model by intervention to *think* that the played-move game sequence was actually passed in instead? The intervention again refers to replacing the last activation vector, z, in a particular layer with a new one, z', according to equation 1; in general, the world representation for the played-move vs the alternate-move sequence can differ by many tiles.

If the decision logic underlying the model's predictions relies on the world representation, then the logits given the played-move sequence as an input should exactly match the logits post-intervention. In practice, we expect that they will slightly differ since the linear probe transformation is not invertible and the gradient descent method of equation 1 is an approximation. We quantify the extent to which the model relies on the world representation by computing the cosine similarity between the true logits and the post-intervention logits (with a reference value obtained by computing the similarity without intervention). Visualization of the game boards and the logits produced with and without intervention can be seen for two examples in Figure 5. We find the intervention to be very successful in tricking the model.

We then apply this calculation for all trained GPT models and compute the similarity of the logits with respect to the game sequence length and probe intervention layer. The results, averaged over 50 games, are shown in Figure 6. While the next-move logits without intervention have non-negligible cosine-similarity, the difference between the similarity with and without intervention effectively reveals at what layer the world representation is most influential. Notably, for models with both 4 and 8 layers, we find that interventions based on the state of the world representation in the final two layers produced very little changes to the next-move logits. This is consistent with our emperical observations that the latent saliency plots and the causal interventions were not as effective if the probe is applied to the the last layer of the model. This is surprising since the world representation can be extracted by the probes from the final few layers with near peak accuracy according to Table 1.

In general, we also observe from the structure in the difference plots that the world representation is most influential in dictating the model's predictions about halfway through the model, implying that semantic information is most relevant in the middle layers of GPT-like architectures. This fact holds even as the model is reduced from 8 layers to 4. We suspect that the variations with respect to the game length could arise from factors owing more to the similarity of sparse vs dense logit vectors in the analysis rather than reflective of the model itself. Lastly, we highlight that the world representation interventions were not successful for the GPT-1-Layer model, likely owing to the fact that the probe can only be applied to the last layer. We believe this is notable because the GPT-1-Layer model was still able to play legal games of Othello with a valid play rate of almost 95%. This suggests that models may be successful by memorization without relying on the direct encoding of semantic information.

3 MECHANISTIC INTERPRETABILITY WITHOUT AN A PRIORI WORLD REPRESENTATION

Mechanistic interpretability is a general technique to understand how trained networks compute the input-output function by analyzing the components that the function is composed of. The hope is that by zooming into the specific neurons and weight matrices, some human interpretable concepts can be found in the artificial neural networks. For example, by analyzing CNNs, it was demonstrated that the neurons pick up coarse-grained features through the convolutional layers [6]. In the context of transformers, it was shown that by analyzing what attention heads and weight matrices are doing, a "copying" behavior was found in language models, which is attributed to the existence of induction heads [2, 7].

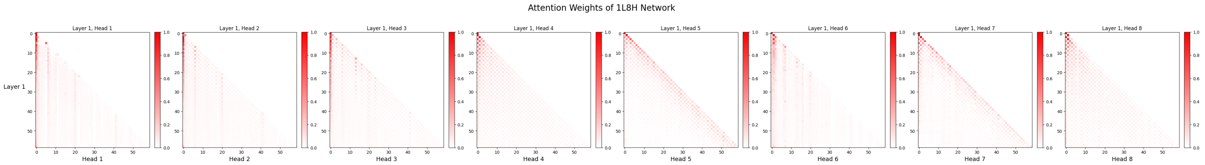


Fig. 7. Attention Matrix of the 1-layer 8-head network for a sample game. Notice that heads all show alternating patterns, which we interpret as processing information from pieces first placed by the same player. For example, Head 3 is keeping track of all my historical moves and Head 4 is keeping track of all their historical moves.

In the context of Othello, we dive into a preliminary examination of whether the trained sequential models directly contain human interpretable concepts. Since the attention mechanism can be written as a tensor product between the attention matrix and so-called OV circuit matrix, we examine the properties of these two components of the attention mechanism. We explain the technical details of interpreting a 1-Layer transformer without MLP layers in Appendix B. In the following subsections, we focus on understanding the 1-layer network with 8 attention heads. Analysis of the 8-layer network can be found in Appendix C.

3.1 Visualizing Attentions

We visualize attention patterns in order to identify meaningful circuits in the attention modules of our network. The attention matrix A for a sample game is calculated as in Eq.5. We find regularities in the attention matrix for the each of the 8 attention heads in the 1 layer network.

In particular, we observe in Figure 7 that there is a conserved alternating patterns in the attention heads, which can be interpreted as keeping track of a single player's historical pieces. For example, head 3 shows alternating weights, and its diagonals are almost empty, which means that the representation of move t_i after the attention mechanism is going to be an average over pieces that used to be "my" historical moves. We deem this head "My-Token Head". (See Appendix C for a detailed discussion.) On the other hand, head 4 has filled diagonals, thus is a "Their-Token Head".

Furthermore, we show in Figure 8 and 9 the attention distribution for heads 3 and 4 respectively, as the game progresses. We see clearly that heads 3 and 4 are each tracking one-half of the game pieces, which corresponds to my and the opponent's historical moves.

3.2 OV Circuit

Since the full operation of the transformer is a tensor product as in Equation 4, we can analyze seperately the OV circuit matrix $W_{OV} = W_O W_V$.

Since we expect that the predicted legal moves should be non-repetitive, we expect that there is no "copying" behavior in the OV circuit, ie. there are not a lot of positive eigenvalues for the circuit matrices W_{OV}^h corresponding to each head h.

Indeed Figure 10 shows that there are roughly equal number of positive eigenvalues as negative eigenvalues. What is suprising is that it seems like the eigenvalues are drawn from a random matrix and there is no structure at all in the circuit. We don't yet know how to interpret this result.

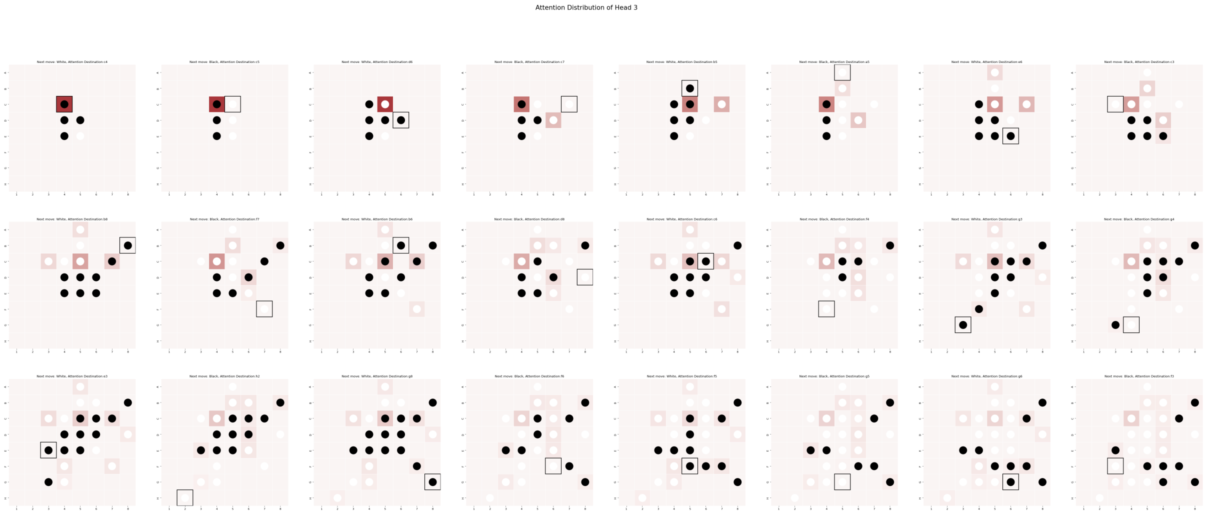


Fig. 8. Attention distribution of head 3 on Othello board for the first 24 moves. For the same game as in Figure 7, we show the board state and attention distribution for the last move on the board, which is labeled by the black square. Dark color corresponds to large weights and light color corresponds to small attention weights.

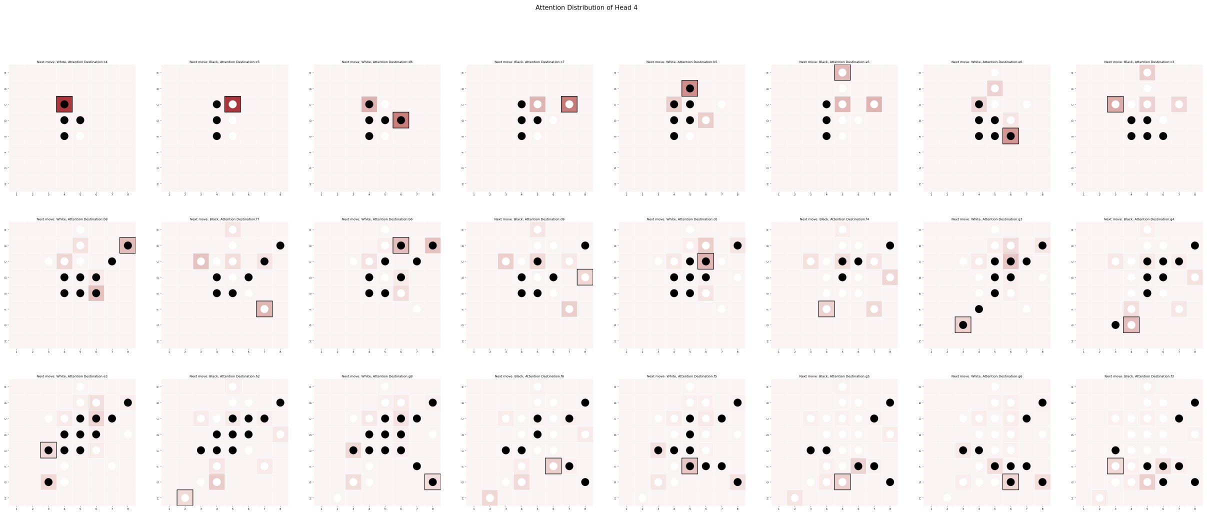
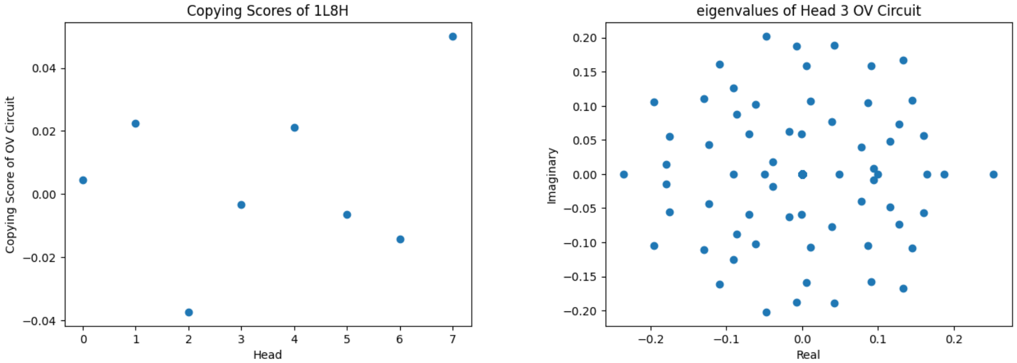


Fig. 9. Attention distribution on Othello board for head 4. Head 4 is tracking a different set of pieces than head 3.

4 CONCLUSION

Through this project, we showed that the Othello-GPT neurons do encode information about the board state linearly, even for a shallow 1-layer network. Furthermore, the board state representation is used to make predictions with deep networks but not shallow networks. Lastly, we showed that the attention heads are indeed keeping track of the turns of the players, which is evident of the GPT's understanding of the game.



(a) Copying score for each head.

(b) Eigenvalue distribution for head 3 OV circuit.

Fig. 10. OV circuit eigenvalues. Here copying score is defined as $\frac{\sum_i \lambda_i}{\sum_i \|\lambda_i\|_2}$, which correlates with how much the head copies the source tokens to predict the next move.

However, how exactly the Othello-GPT makes the next-move prediction and how exactly are board-state representations used remain a mystery. For future directions, we hope to start with the 1L network and analyze the logit attribution by merging information regarding the attention heads and OV circuits. We also want to understand what the causal interventions are really doing from understanding its perturbation of the attention heads and circuit matrices.

A THE RULES OF OTHELLO

Here, we provide a brief summary of the rules for the game. Othello, also known as Reversi, is a classic strategy board game for two players, typically played on an 8x8 grid with two-sided pieces called "discs" that are black on one side and white on the other. The objective is to have more discs of your color on the board than your opponent at the end of the game. All games start with an initial setup of four discs placed at the center of the board, with two white discs forming a diagonal and two black discs forming the opposite diagonal. One player will play black discs while the other will play white. Players then take turns placing one disc of their color on an empty square on the board, with the white player always going first.

- Flipping discs: A move is valid only if it "sandwiches" (or "flanks") one or more of your opponent's discs between the disc you are placing and another of your discs already on the board. This must occur in a straight line (horizontal, vertical, or diagonal). After placing the disc, flip all the sandwiched opponent's discs to your color. If a player cannot make a valid move, they must pass their turn.
- Win Condition: The game ends if neither player can make a valid move or if the board is full. The player with the most discs of their color on the board wins the game. If both players have the same number of discs, the game is a draw.

B TRANSFORMER CIRCUIT OF 1-LAYER ATTENTION NETWORK

Elhage et al. [2] characterized shallow attention networks in a simple mathematical framework deemed transformer circuits. The idea is that due to the presence of large amounts of linear structures in the network, we can simply multiply out the matrices which result in a sum of interpretable end-to-end functions mapping tokens to changes in logits. For example, in a 1L attention network without the MLP layers, the attention input-output function is a rank-2 tensor T, with

$$X_{out} = T \circ X_{in} \tag{2}$$

$$\tag{3}$$

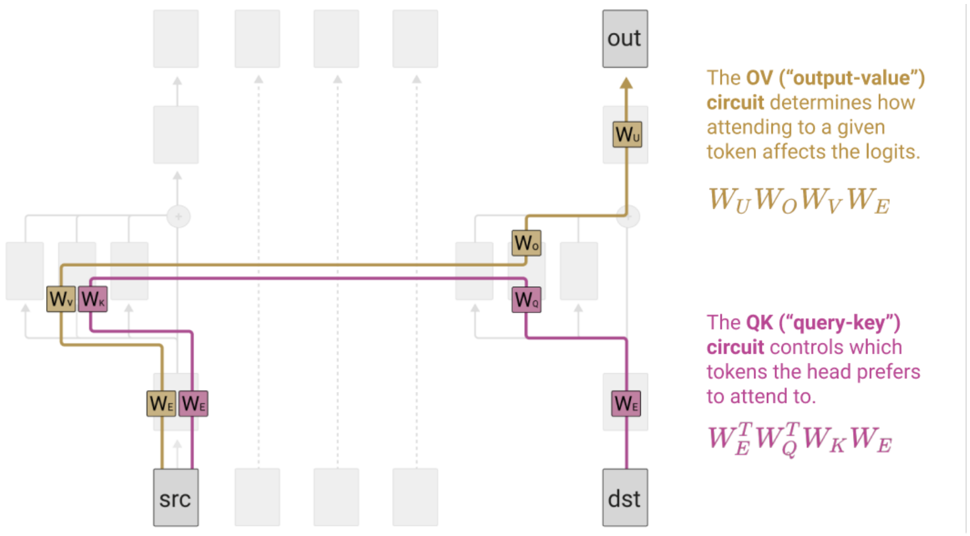
and

$$T = Id \otimes W_U W_E + \sum_h A^h \otimes (W_U W_{OV}^h W_E), \tag{4}$$

with

$$A = \text{softmax}^*(X_{in}^T W_E^T W_{QK}^h W_E X_{in}) \tag{5}$$

where X_{in} is the input sequence matrix, X_{out} is the output logit matrix, and A^h is the attention matrix for each head h, W_E the embedding matrix and W_U the unembedding matrix, $W_{OV} = W_O W_V$, $W_{QK} = W_Q^T W_K$ are products of the output-value and query-key matrices respectively. We can interpret the first part in the sum as the skip connection that bypasses the self-attention, thus resulting in a simple linear transformation in the feature space. The second part is more involved and can be interpreted as a linear transformation in the token space with attention matrix A and independently a linear transformation in the feature space. These functions correspond to "paths" through the model and are linear if one freezes the attention patterns, as illustrated in Figure 11.



The **OV ("output-value") circuit** determines how attending to a given token affects the logits.

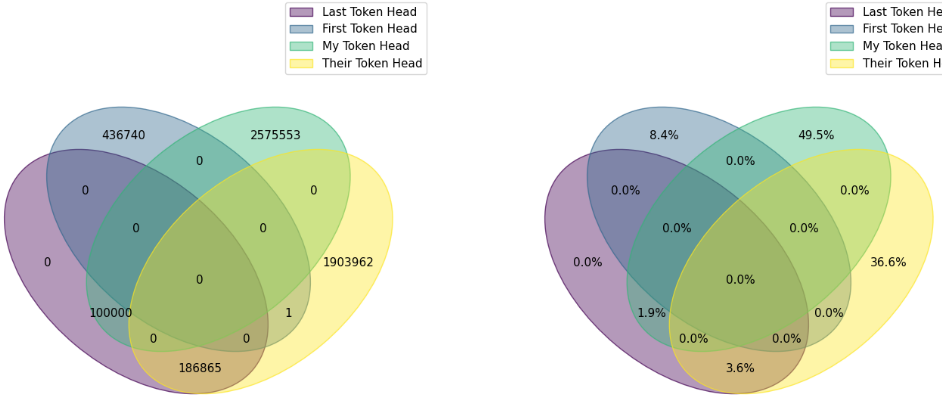
$W_U W_O W_V W_E$

The **QK ("query-key") circuit** controls which tokens the head prefers to attend to.

$W_E^T W_Q^T W_K W_E$

Fig. 11. Circuits representations of the attention mechanism. This figure is borrowed from [2]

$W_U W_{OV}^h W_E$ is thus called the OV circuit and $W_E^T W_{QK}^h W_E$ called the QK circuit. Elhage et al. found that with this circuit formulation of the attention mechanism, we do not need to run the model to get insights about what tokens are attended and what tokens change the logits the most. Concretely, with fixed source token, we can find the destination token that corresponds to the largest attention in the QK circuit, and independently we can find the output token that corresponds to the largest element in the OV circuit. This line of argument results in a



(a) Circuit Count by Type across Entire Synthetic Othello Dataset

(b) Percentages of Circuits by Type across Entire Synthetic Othello Dataset

Fig. 12. There are four type of circuits we find in our Othello model, last token heads (which looks at the last played move), first token heads (which look at the first played move), my token heads (which look at moves I made), and their token heads (which look at moves my opponent made).

[source]...[destination]→ [output] skip-gram. They found that 1L network mainly does copy-and-paste type of skip-grams, with the output token roughly the same as the source token, and with a highly correlated destination token.

C ANALYSIS OF 8-LAYER NETWORK ATTENTION HEADS

We analyze the attention heads of the full 8 layer, 8 head model. We visualize the attention heads for a specific example below Figure 13.

Looking at the visualized attention patterns we see that attention heads fit pretty cleanly into four types of attention circuits

Table 2. Distribution of Circuits across Layers (rows sum to 1)

Layer	Last Token Head	First Token Head	My Token Head	Their Token Head	None
Layer 1	0.00%	0.00%	37.50%	25.00%	37.50%
Layer 2	0.00%	0.00%	24.99%	49.98%	25.03%
Layer 3	0.00%	0.00%	49.98%	49.99%	0.03%
Layer 4	0.00%	0.00%	86.95%	12.50%	0.55%
Layer 5	12.50%	0.00%	62.27%	25.00%	0.23%
Layer 6	0.00%	0.01%	49.90%	37.50%	12.61%
Layer 7	0.00%	28.80%	10.19%	11.54%	49.47%
Layer 8	23.36%	25.79%	0.16%	26.49%	24.20%

- **First Token Heads**: These attention heads focus only on the first token of the sequence; examples include Layer 7 Head 1, Layer 6 Head 1, and Layer 8 Head 2 Figure 13.
- **Last Token Heads**: These attention heads focus on the second-to-last (position of last move you played) or last token (position of last move your opponent played); examples include Layer 5 Head 6, Layer 8 Head 3, and Layer 8 Head 1 Figure 13.
- **My Token Heads**: These attention heads focus on positions corresponding to moves made by the color to play (for example, if white to play, it will pay attention to moves in the sequence that were made by white). Examples include Layer 1 Head 1, Layer 2 Head 3, and Layer 3 Head 2 Figure 13.
- **Their Token Head**: These attention heads focus on positions corresponding to moves made by the color that just played (for example, if white to play, it will pay attention to moves in the sequence that were made by black). Examples include Layer 1 Head 4, Layer 2 Head 1, and Layer 2 Head 5 Figure 13.

We attempt to interpret the purpose of these circuits. Looking at Table 2, we see that the circuits in Layers 1 through Layer 4 only consists of my token heads or their token heads. Additionally, we notice that in Figure 13, the "context size" of the attention heads for the my token heads and their token heads seem to decrease over time despite still following the behavior we enumerated above; we see that the attention heads in earlier layers tend to pay attention to moves across the entire context, whereas attention heads in later layers only pay attention to the moves closest to them.

Another interesting thing that we notice in Table 2 is that Last token heads only appear in Layers 5 and 8 and First Token Heads only in Layers 7 and Layers 8. Taking all this into account, we hypothesize that the necessary representations for the board state (my piece vs. your piece) are largely already finalized by the later layers (Layer 5 and onwards) in which the information is then propagated to the activations in the last positions, to be attended to by the last token heads, which explains why there are last token heads that only attend to the 2nd to last token (my pieces) and last token heads that only attend to the last token (your pieces). We are unsure about how to interpret first token heads, although there is also some evidence that show that First Token Heads may simply be the default nature for unactivated attention heads [2]. Our hypothesis is in-line with the results we see from the linear probes Table 1, where we see that the accuracy of the probes have converged after layer 5.

REFERENCES

[1] Shir Amir, Yossi Gandelsman, Shai Bagon, and Tali Dekel. 2021. Deep ViT Features as Dense Visual Descriptors. *CoRR* abs/2112.05814 (2021). arXiv:2112.05814

[2] Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. 2021. A Mathematical Framework for Transformer Circuits. *Transformer Circuits Thread* (2021). https://transformer-circuits.pub/2021/framework/index.html.

[3] Kenneth Li, Aspen K Hopkins, David Bau, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023. Emergent World Representations: Exploring a Sequence Model Trained on a Synthetic Task. In *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=DeG07_TcZvT

[4] Neel Nanda. 2023. Actually, Othello-GPT Has A Linear Emergent World Model. <https://neelnanda.io/mechanistic-interpretability/othello>

[5] Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. 2020. Zoom in: An introduction to circuits. *Distill* 5, 3 (2020), e00024–001.

[6] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. 2017. Feature Visualization. *Distill* (2017). https://doi.org/10.23915/distill.00007 https://distill.pub/2017/feature-visualization.

[7] Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. 2022.

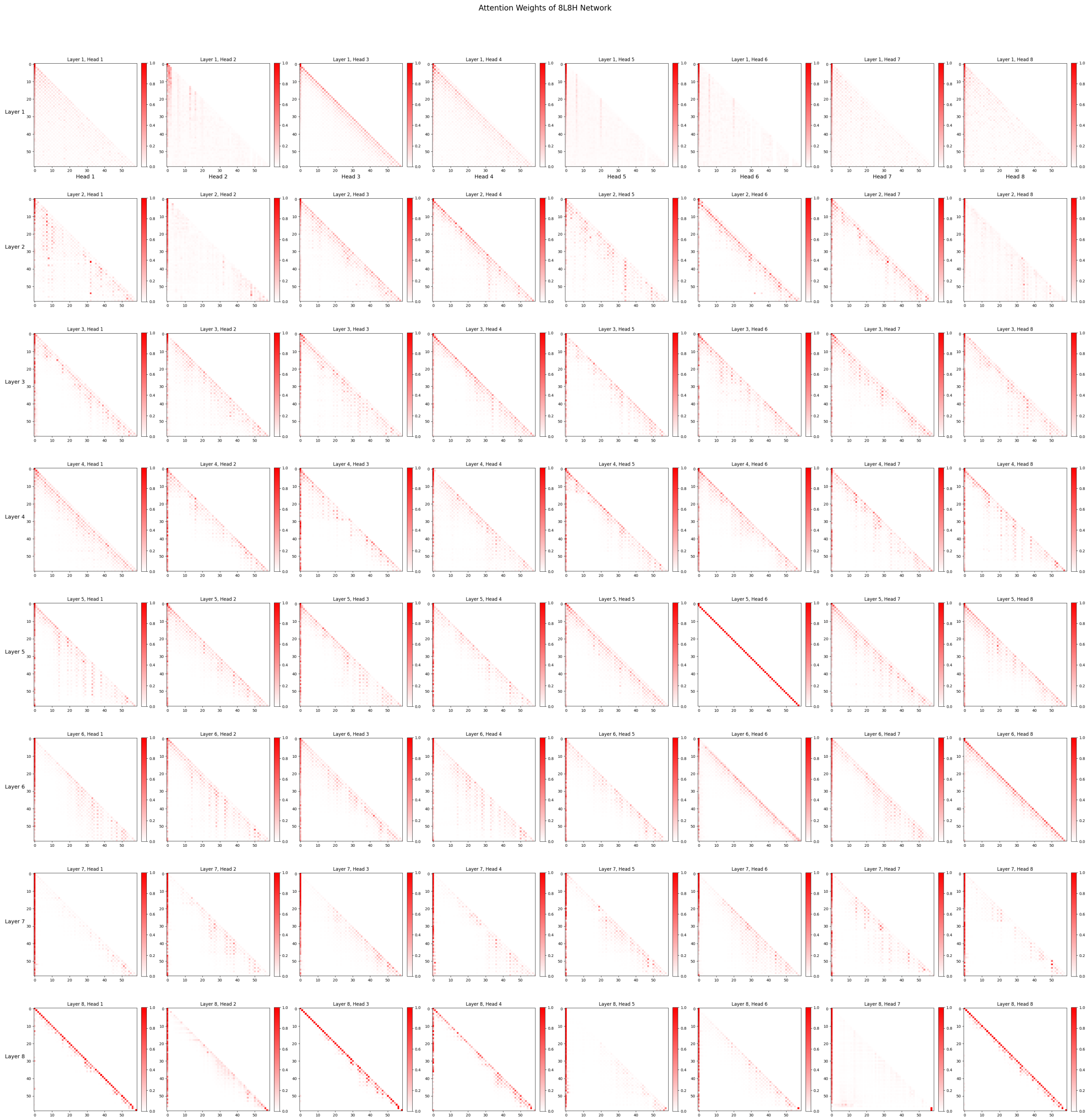


Fig. 13. Example attention matrix for 8L8H Network

In-context Learning and Induction Heads. *Transformer Circuits Thread* (2022). https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html.

[8] Shubham Toshniwal, Sam Wiseman, Karen Livescu, and Kevin Gimpel. 2021. Learning Chess Blindfolded: Evaluating Language Models on State Tracking. *CoRR* abs/2102.13249 (2021). arXiv:2102.13249