EMERGENT WORLD REPRESENTATIONS: EXPLORING A SEQUENCE MODEL TRAINED ON A SYNTHETIC TASK

Anonymous authors
Paper under double-blind review

ABSTRACT

Language models show a surprising range of capabilities, but the source of their apparent competence is unclear. Do these networks just memorize a collection of surface statistics, or do they rely on internal representations of the process that generates the sequences they see? We investigate this question by applying a variant of the GPT model to the task of predicting legal moves in a simple board game, Othello. Although the network has no a priori knowledge of the game or its rules, we uncover evidence of an emergent nonlinear internal representation of the board state. Interventional experiments indicate this representation can be used to control the output of the network and create "latent saliency maps" that can help explain predictions.

1 INTRODUCTION

Recent language models have shown an intriguing range of capabilities. Networks trained on a simple "next-word" prediction task are apparently capable of many other things, such as solving logic puzzles or writing basic code. [1] Yet how this type of performance emerges from sequence predictions remains a subject of current debate.

Some have suggested that training on a sequence modeling task is inherently limiting. The arguments range from philosophical (Bender & Koller, 2020) to mathematical (Merrill et al., 2021). A common theme is that seemingly good performance might result from memorizing "surface statistics," i.e., a long list of correlations that do not reflect a causal model of the process generating the sequence. But relying on spurious correlations may lead to problems on out-of-distribution data (Bender et al., 2021; Floridi & Chiriatti, 2020).

On the other hand, some tantalizing clues suggest language models may do more than collect spurious correlations, building interpretable *world models*—that is, understandable models of the process producing the sequences they are trained on. Recent evidence suggests language models can develop internal representations for very simple concepts, such as color, direction Abdou et al. (2021); Patel & Pavlick (2022), or tracking boolean states during synthetic tasks (Li et al., 2021) (see Related Work (section 6) for more detail).

The question remains, then, of how we might investigate the emergence of world models in more complex domains. One possibility comes from Toshniwal et al. (2021), who explore language models trained on chess move sequences. These models learn to predict legal moves with high accuracy. Furthermore, by analyzing predicted moves, the paper shows that the model appears to track the board state. The authors stop short, however, of investigating the form of any internal representations. Such an investigation will be the focus of this paper.

1.1 THE GAME OF OTHELLO AS TESTBED FOR INTERPRETABILITY

Toshniwal et al. (2021)'s observations suggest a new approach to studying the representations learned by sequence models. If we think of a board as the "world," then games provide us with an appealing experimental testbed to explore world representations of moderate complexity. As our setting, we choose the popular game of Othello (Figure 1), which is simpler than chess. This setting allows us to

[1] See Srivastava et al. (2022) for an encyclopedic list of such examples.

investigate world representations in a highly controlled context, where both the task and sequence being modeled are synthetic and well-understood.

As a first step, we train a language model (a GPT variant we call Othello-GPT) to extend partial game transcripts (a list of moves made by players) with legal moves. **The model has no a priori knowledge of the game or its rules**. All it sees during training is a series of tokens derived from the game transcripts. Each token represents a tile where players place their discs. Note that we do *not* explicitly train the model to make strategically good moves or to win the game. Nonetheless, our model is able to generate legal Othello moves with high accuracy.

Our next step is look for world representations that might be used by the network. In Othello, the "world" consists of the current board position. A natural question is if, within the model, we can identify a representation of the board state involved in producing its next move predictions. To study this question, we train a set of probes, i.e., classifiers which allow us to infer the board state from the internal network activations . This type of probing has become a standard tool for analyzing neural networks (Alain & Bengio, 2016; Tenney et al., 2019; Belinkov, 2016).

Using this probing methodology, we find evidence for an emergent world representation. In particular, we show that a non-linear probe is able to predict the board state with high accuracy (section 3). (Linear probes, however, produce poor results.) This probe defines an internal representation of the board state. We then provide evidence that this representation plays a causal role in the network's predictions. Our main tool is an intervention technique that modifies internal activations so that they correspond to counterfactual board states.

We also discuss how knowledge of the internal world model can be used as an interpretability tool. Using our activation-intervention technique, we create *latent saliency maps*, which provide insight into how the network makes a given prediction. These maps are built by performing attribution at a high-level setting (the board) rather than a low-level one (individual input tokens or moves).

To sum up, we present four contributions: (1) we provide evidence for an emergent world model in a GPT variant trained to produce legal moves in Othello; (2) we compare the performance of linear and non-linear probing approaches, and find that non-linear probes are superior in this context; (3), we present an intervention technique that suggests that, in certain situations, the emergent world model can be used to control the network's behavior; and (4) we show how probes can be used to produce *latent saliency maps* to shed light on the model's predictions.

2 "Language modeling" of Othello game transcripts

Our approach for investigating internal representations of language models is to narrow our focus from natural language to a more controlled synthetic setting. We are partly inspired by the fact that language models show evidence of learning to make valid chess moves simply by observing game transcripts in training data (Srivastava et al., 2022). We choose the game Othello, which is simpler than chess, but maintains a sufficiently large game tree to avoid memorization. Our strategy is to see what, if anything, a GPT variant learns simply by observing game transcripts, without any a priori knowledge of rules or board structure.

The game is played on an 8x8 board where two players alternate placing white or black discs on the board tiles. The object of the game is to have the majority of one's color discs on the board at the end of the game. Othello makes a natural testbed for studying emergent world representations since the game tree is far too large to memorize, but the rules and state are significantly simpler than chess.

The following subsections describe how we train a system with no prior knowledge of Othello to predict legal moves with high accuracy. The system itself is not our end goal; instead, it serves as our object of study.

2.1 Datasets: "Championship" and "Synthetic"

We use two sets of training data for the system, which we call "championship" and "synthetic". Each captures different objectives, namely data quality vs. quantity. While limited in size, championship data reflects strategic moves by expert human players. The synthetic data set is far larger, consisting of legal but otherwise random moves.

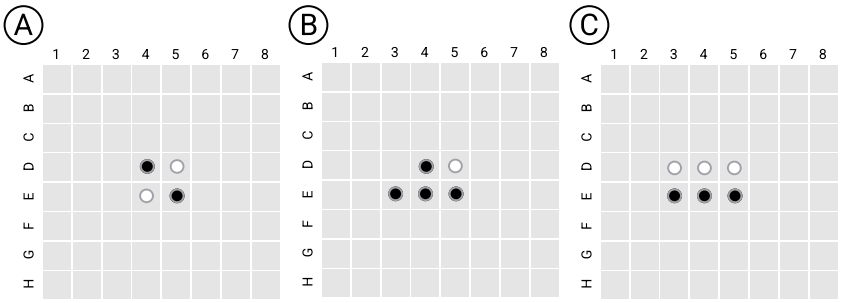

Figure 1: A visual explanation of Othello rules, from left to right: (A) The board is initialized with four discs (two black and two white) placed in the center of the board. (B) Black always moves first. Every move must flip one or more opponent discs by outflanking—or sandwiching—the opponent disc(s). (C) The opponent repeats this process. A game ends when there are no more legal moves.

Our *championship dataset* is produced by collecting Othello championship games from two online sources[2], containing $7,605$ and $132,921$ games, respectively. They are combined and split randomly by $8:2$ into training and validation sets. The games in this dataset were produced by matches where human players presumably made moves with a strategic intent to win. Following this, we generate a *synthetic dataset* with 20 million games for training and $3,796,010$ games for validation. We compute this dataset by uniformly sampling leaves from the Othello game tree. Its data distribution is different from the championship games, reflecting no strategy.

2.2 Model and Training

Our goal is to study how much Othello-GPT can learn from pure sequence information, so we provide as few inductive biases as possible. (Note the contrast with a system like AlphaZero (Silver et al., 2018), where the goal was to win highly competitive chess games.) We therefore use only sequential tile indices as input to our model. For example, A4 and H6 are indexed as the 4rd and the 62st word in our vocabulary, respectively. Each game is treated as a sentence tokenized with a vocabulary of 60 words, where each word corresponds to one of the 60 tiles on which players put discs, excluding the 4 tiles in the center (Figure 1).

We trained an 8-layer GPT model (Radford et al., 2018; 2019; Brown et al., 2020) with an 8-head attention mechanism and a 512-dimensional hidden space. The training was performed in an autoregressive fashion. For each partial game $\{y_t\}_{t=0}^{T-1}$, the computation process starts from indexing a trainable word embedding consisting of the 60 vectors, each for one word, to get $\{x_t^0\}_{t=0}^{T-1}$. They are then sequentially processed by 8 multi-head attention layers. We denote the intermediate feature for the t-th token after the l-th layer as x_t^l. By employing a causal mask, only the features at the immediately preceding layer and earlier time steps $x_{\leqslant t}^{l-1}$ are visible to x_t^l. Finally, x_{T-1}^8 goes through a linear classifier to predict logits for $\hat{y}_T$. We minimize the cross-entropy loss between ground-truth move and predicted logits by gradient descent.

The model starts from randomly initialized weights, including in the word embedding layer. Though there are geometrical relationships between the 60 words (e.g., C4 is below B4), this inductive bias is not explicitly given to the model but rather left to be learned.

2.3 Othello-GPT Usually Predicts Legal Moves

We now evaluate how well the model's predictions adhere to the rules of Othello. For each game in the validation set, which was not seen during training, and for each step in the game, we ask Othello-GPT to predict the next legal move conditioned by the partial game before that move. We then calculate the error rate by checking if the top-1 prediction is legal. The error rate is 0.01% for Othello-GPT trained on the synthetic dataset and 5.17% for Othello-GPT trained on the championship dataset. For comparison, the untrained Othello-GPT has an error rate of 93.29%. The main takeaway is that **Othello-GPT does far better than chance in predicting legal moves** when trained on both datasets.

[2] `www.liveothello.com` and `www.ffothello.org`.

We discuss reasons for the difference between the error rates for the synthetic and championship models later in the paper.

A potential explanation for these results may be that Othello-GPT is simply memorizing all possible transcripts. To test for this possibility, we created a *skewed dataset* of 20 million games to replace the training set of synthetic dataset. At the beginning of every game, there are four possible opening moves: C4, D3, E6 and F5. This means the lowest layer of the game tree (first move) has four nodes (the four possible opening moves). For our skewed dataset, we truncate one of these nodes (C4), which is equivalent to removing a quarter of the whole game tree. Othello-GPT trained on the skewed dataset still yields an error rate of 0.02%. Since Othello-GPT has seen none of these test sequences before, pure sequence memorization cannot explain its performance [3].

If the performance of Othello-GPT is not due to memorization, what is it doing? We now turn to this question by probing for internal representations of the game state.

3 Exploring Internal Representations with Probes

We seek to understand if Othello-GPT computes internal representations of the game state. One standard tool for this task is a "probe" (Alain & Bengio, 2016; Belinkov, 2016; Tenney et al., 2019). A probe is a classifier or regressor whose input consists of internal activations of a network, and which is trained to predict a feature of interest, e.g., part of speech or parse tree depth (Hewitt & Manning, 2019). If we are able to train an accurate probe, it suggests that a representation of the feature is encoded in the network's activations.

In our case, we want to know whether Othello-GPT's internal activations contain a representation of the current board state. To study this question, we train probes that predict the board state from the network's internal activations after a given sequence of moves. Note that the board state—whether each tile is empty or holds a black or white disc—is generally a nonlinear function of the input tokens. On the other hand, it is straightforward to write a program to compute this function, it makes a natural probe target.[4]

We take the autoregressive features x_t^l that summarize the partial sequence $y_{\leqslant t}$ as the input to the probe and study results from different layers l. The output $p_\theta(x_t^l)$ is a 3-way categorical the probability distribution. We randomly split pairs of internal representation and ground-truth tile state by $8:2$ into training and validation set. Error rates on validation set are reported. A best random guess yields 52.95%, if the probe always guess the tile is empty.

3.1 Linear Probes Have High Error Rates

Our first result is that linear classifier probes have poor relative accuracy. Its function can be written as $p_\theta(x_t^l) = \text{softmax}(Wx_t^l)$ where $\theta = \{W \in \mathbb{R}^{F\times 3}\}$. F is the number of dimensions of input x_t^l. As Table 1 shows, error rates never dip below 20%. As a baseline, we have included probes trained on a randomly initialized network[5] We can see that there is only a marginal improvement in accuracy when we move to probing a fully-trained network. This result suggests that if there is an internal representation of the board state, it does not have a simple linear form.

3.2 Nonlinear Probes Have Lower Error Rates

Given the poor performance of linear probes, it is natural to ask whether a nonlinear probes would have higher accuracy. Moving up one notch of complexity, we apply a 2-layer MLP as a probe. This technique has been used successfully in other language model probing work, e.g., Conneau et al. (2018); Cao et al. (2021); Hernandez & Andreas (2021). Its function can be written as

[3] One potential criticism is that certain board states are likely repeated across test sequences, but memorization related to board states rather than move sequences supports our hypothesis of emergent world representations.

[4] Classifying a tile as unoccupied or occupied can be written as a linear function of the input sequence, thus we consider only the 3-way black/white/empty classifiers.

[5] Probes on the randomized network do better than chance; a constant guess of "empty" has a 47% error rate. But that performance is not a surprise, since it makes sense that some information about moves is preserved even by a random network. The key comparison is between the randomized network and the trained network.

	x^1	x^2	x^3	x^4	x^5	x^6	x^7	x^8
Randomized	26.7	27.1	27.6	28.0	28.3	28.5	28.7	28.9
Championship	24.2	23.8	23.7	23.6	23.6	23.7	23.8	24.3
Synthetic	21.9	20.5	20.4	20.6	21.1	21.6	22.2	23.1

Table 1: Error rates (%) of linear probes on Othello-GPTs trained on different datasets across different layers (x^i represents internal representations after the i-th layer).

$p_\theta(x_t^l) = \text{softmax}(W_1\text{ReLU}(W_2 x_t^l))$ where $\theta = \{W_1 \in \mathbb{R}^{H\times 3}, W_2 \in \mathbb{R}^{F\times H}\}$. H is the number of hidden dimensions for the nonlinear probes.

The probe accuracy for trained networks, shown in Table 2, is significantly better than the linear probe in absolute terms. By contrast, the baseline (probing a randomized network with nonlinear probes) shows almost no improvement over the linear case. These results indicate that the probe may be recovering a nontrivial representation of board state in the network's activations. In section 4, we describe intervention experiments validating this hypothesis.

	x^1	x^2	x^3	x^4	x^5	x^6	x^7	x^8
Randomized	25.5	25.4	25.5	25.8	26.0	26.2	26.2	26.4
Championship	12.8	10.3	9.5	9.4	9.8	10.5	11.4	12.4
Synthetic	11.3	7.5	4.8	3.4	2.4	1.8	1.7	4.6

Table 2: Error rates (%) of nonlinear probes on Othello-GPTs trained across layers.

4 Validating Probes with Interventional Experiments

Our nonlinear probe accuracies suggest that Othello-GPT computes information reflecting the board state. It's not obvious, however, whether that information is causal for the model's predictions. To investigate this issue, we evaluate whether the representations uncovered through section 3 play a causal role in Othello-GPT's predictions. In the following section, we adhere to Belinkov (2016)'s recommendation, performing a set of interventional experiments to determine the causal relationship between model predictions and the emergent world representations.

To figure out whether the board state information affects the network's predictions, we influence internal activations *during* Othello-GPT's calculation and measure the resulting effects. At a high level, the interventions are as follows: given a set of activations from the Othello-GPT, a probe predicts a baseline board state B. We save the move predictions associated with B, then modify these activations such that our probe reports an updated board state B'. Through our protocol, only a single tile s distinguishes B' from B's board state (an example of which can be seen in Figure 2. This small modification results in a different set of possible legal moves for B'. If the new predictions match our expectations for B'—and not B—we conclude the representation had a causal effect on the model.

4.1 Intervention Technique

To implement an intervention that changes the predicted state from a board position B to a modified version B' we must decide (a) which layers to modify activations in, and (b) how to modify those activations. The first question is subtle. Given the causal attention mechanism of GPT-2, modifying activations for only one layer is unlikely to be effective as later layer computations incorporate information from prior board representations unaffected by our intervention. Instead, we select an initial layer L_s then modify it and subsequent layers' activations (see Figure 2 (C)). Our modification uses a simple gradient descent method on the probe's class score for the particular tile s whose state is being modified.

Figure 2 illustrates an intervention on a single feature x into x' such that the corresponding board state B is updated to match the desired B'. We observe the effectiveness of these interventions by probing the intervened x' or x at later layers (see Appendix D), as well as the change in next-step prediction in (see subsection 4.2). Consistent with the training process of probes p_θ, we use cross entropy loss between the probe-predicted probability distribution and the desired board state, but

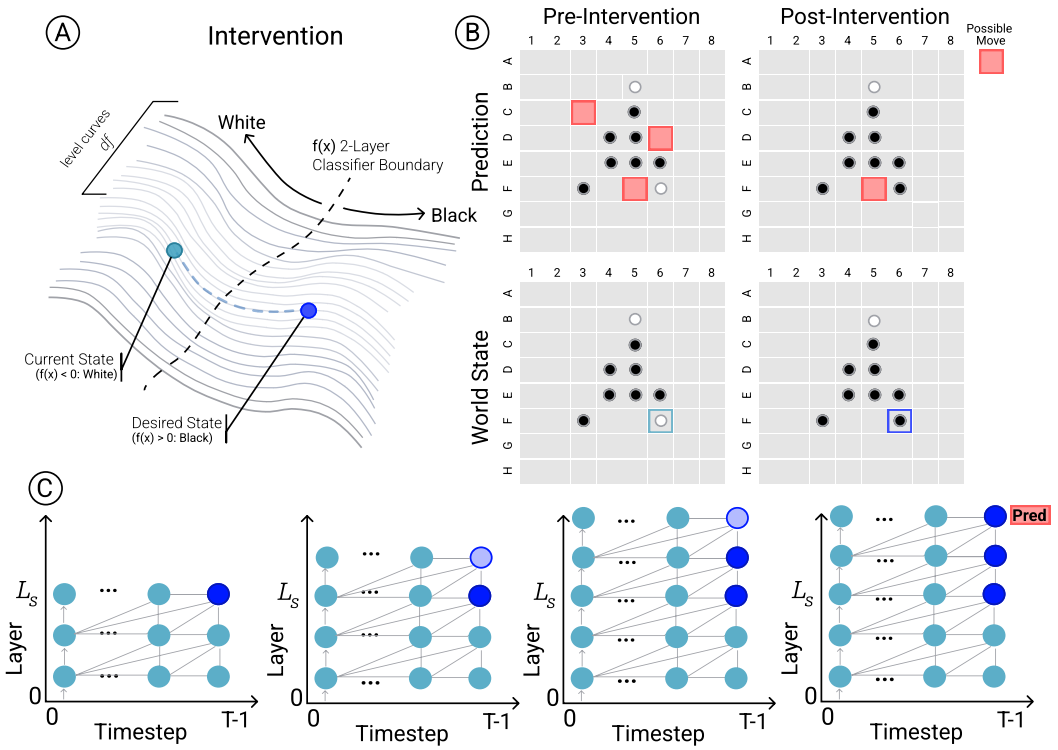

Figure 2: **(A)** explains how we intervene on a board tile. Here, we only want to flip one tile, e.g. F6, from white to black. In **(B)**, four views present an Othello game in progression, which can be reliably probed from an internal representation x. The lower left board represents the model's perceived world state prior to intervention. The upper left board shows the model's predictions for legal moves given this state. Post-intervention, the model's world state is updated—F6's state has been switched from white to black (lower right), leading to a different set of legal move predictions (upper right). Note that two tiles (F6) are highlighted in the world state boards. This is the tile that we "intervene" on, changing from white to black. **(C)** Shows our proposed intervention scheme. Light blue indicates unmodified activations; dark blue represents activations affected by intervention. Starting from a predefined layer, we intervene at the temporally-last token (shown in **(A)**). We replace original internal representations with the post-intervention one and resume computation for the next one layer. Part of the misinformation gets corrected (light blue), but we alternate this intervening and computation process until the last layer, from which the next-step prediction is made.

rather than optimize probe weights θ, we optimize x for intervention [6]:

$$x' \leftarrow x - \alpha \frac{\partial \mathcal{L}_{\text{CE}}(p_\theta(x), B')}{\partial x}.$$

At timestep T, the internal activations of an L-layer Othello-GPT can be viewed as an $L \times T$ grid of activation vectors. Our intervention process will work by running Othello-GPT sequentially, but using gradient descent to modify key activation vectors at the last timestep so that their board state class scores change. Note that if we change activations only at a middle layer, activations at higher layers are directly affected by pre-intervention information. Therefore, we sequentially intervene $\{x^l_{T-1}\}_{l=L_s}^{L}$ at the last timestep, on all activations starting from a preset layer L_s until the final layer, illustrated in Figure 2.

4.2 Evidence for a Causal Role for the Representation

To systematically evaluate if this world representation is causal for model predictions, we create two evaluation benchmarks. Each consists of 1000 intervention cases: one factual, one counterfactual. A test case in these benchmarks consists of a triplet of a partial game, a targeted board tile, and a

[6]Note that α is the learning rate. See more on the intervention hyper-parameters in Appendix E.

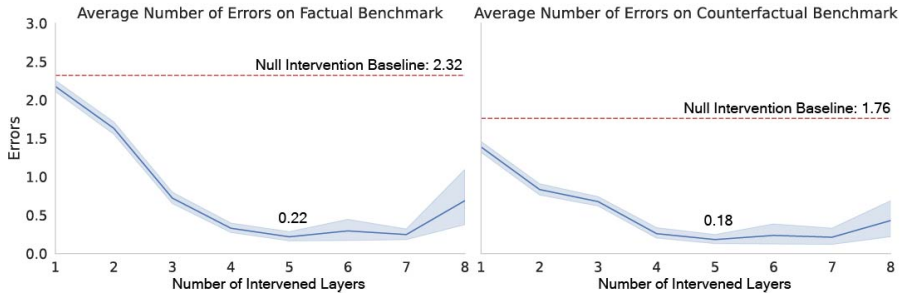

Figure 3: Intervention experiment results. Red dashed lines represents average number of errors by testing pre-intervention predictions on post-intervention ground-truths, representing a null intervention method for contrast. The shaded area represents the 95% confidence interval.

target state. For each case, we will give the partial game to Othello-GPT and perform the intervention described in the previous section. That is, we extract its activations in the middle of the computation process, modify them to change the representation of the targeted board tile into the target state, give back the modified world representation and let it make prediction with this new world state.

In the counterfactual case, we specifically ask whether the model's world representation can represent arrangements of tiles on a board that are unreachable during legal Othello play, i.e., states that do not correspond to any legal sequence of moves. If the model can make correct predictions about such states, it helps rule out the possibility that our probes might have learned to merely project a sequence-oriented internal state to a board-based world model that the probes have hallucinated. If Othello-GPT can make correct predictions about counterfactual states, it provides evidence for an internal representation capable of representing a board rather than just a sequence.

To measure how well the prediction is aligned with ground-truth legal moves, we calculate a prediction set by comparing the prediction probability for each tile with $\frac{1}{2N}$, where N is the number of legal post-intervention moves. Then, we calculate an error per case (a sum of false positives and false negatives, shown in Figure 3)[7]. For both benchmarks, nonlinear probes with $L_S = 5$ give the best result: average errors of $\mathbf{0.22}$ and $\mathbf{0.18}$ respectively. Interventions based on linear probes all give worse than baseline results. Compared to baseline errors ($\mathbf{2.23}$ and $\mathbf{1.76}$), the proposed intervention technique is effective even under counterfactual board states, suggesting the emergent world representations are causal to model predictions.

5 Latent Saliency Maps: Attribution via Intervention

The intervention technique of the previous section provides insight into the predictions of Othello-GPT. We might also use it to create visualizations which contextualize Othello-GPT's predictions in terms of the board state. The basic idea is simple. For any tile on the board, we ask how much the network's prediction would change if we applied the intervention of the previous section to change the state of that tile. This will yield a value per tile, positive or negative, corresponding to its saliency in the prediction (see algorithm 1). We then create a visualization of the board where tiles are colored according to their saliency. Because this map is based on the network's latent space rather than its input, we call it a **latent saliency map**.

Figure 4 shows latent saliency maps for the synthetic and championship versions of Othello-GPT. The two diagrams show a clear pattern. The synthetic Othello-GPT shows high saliency for precisely those tiles that are required to make a move legal. In almost all cases, other tiles have lower saliency values. Even without knowing how synthetic-GPT was trained, an experienced Othello player might be able to guess its goal. The latent saliency maps for the championship version, however, are more complex. Although tiles that relate directly to legality typically have high values, many other tiles show high saliency as well. This pattern makes sense, too. Expert moves rely on complex global features of the board. The difference between the latent saliency maps for the two versions of Othello-GPT suggests that the visualization technique is providing useful information about the two systems.

[7] our qualitative results, along with their interpretation, can be found in Appendix C

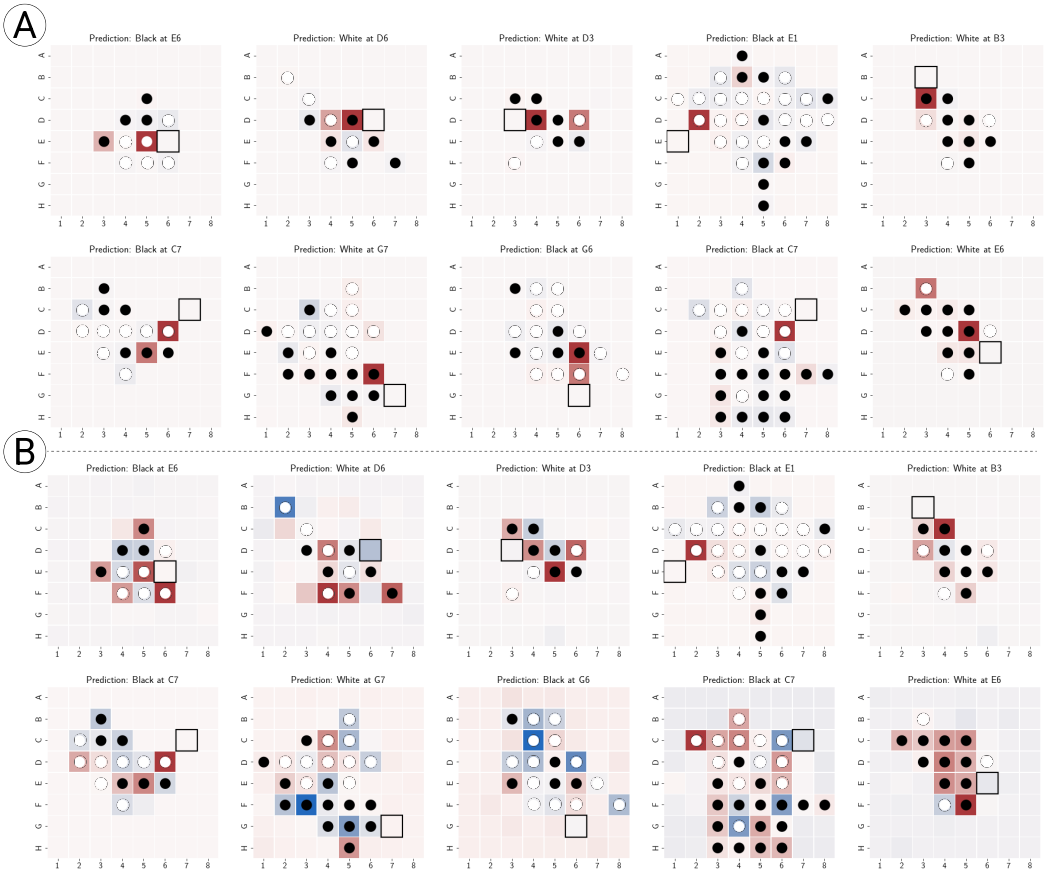

Figure 4: Latent saliency maps: Each subplot shows a different game state, and the top-1 prediction by the model is enclosed in a black box. Colors (red is high, blue is low) indicate the contribution of a square's state to this prediction. The contribution is higher when changing the internal representation of this square makes the prediction less likely. The values are normalized by subtracting the mean of the board. **(A)** Latent saliency maps for Othello-GPT trained on the synthetic dataset, where the model learns *legal* moves. **(B)** Latent saliency maps for Othello-GPT trained on the championship dataset. Rather than learning rules, this Othello-GPT learns to make *strategically good* moves.

Algorithm 1: Attribution via Intervention on Othello-GPT

Inputs :
B the current board state
p a legal next move which we try to attribute
Output :
$\{S_s\}_{s \in B}$ assigned sensitivity values for p
$p_0 \leftarrow f_{\text{p}}(x_{t-1})$
for $s \in B$ **do**
$\tilde{x}_{t-1} \leftarrow \text{Intervention}(x_{t-1}, s)$
$p_s \leftarrow f_{\text{p}}(\tilde{x}_{t-1})$
$S_s \leftarrow p_0 - p_s$

6 Related Work

Our work fits into a general line of investigation into world representations created by sequence models. For example, (Li et al., 2021) fine-tune two language models on synthetic natural language tasks (Long et al., 2016) and find evidence that semantic information about the underlying world state is at least weakly encoded in the activations of the network. More direct evidence of a faithful

representation of 3D color space comes from Abdou et al. (2021), who examine activations in the BERT model and find a geometric connection to a standard 3D color space. Another study by (Patel & Pavlick, 2022) shows that language models can learn to map conceptual domains, e.g., direction and color, onto a grounded world representation via prompting techniques (Brown et al., 2020). These investigations operate in natural language domains, but investigate relatively simple world models.

Another related stream of work concerns neural networks that learn board games. There is a long history of work in AI to learn game moves, but in general, these systems have been given some a priori knowledge of the structure of the game. Even one of the most general-purpose game-playing engines, AlphaZero (Silver et al., 2018), has built-in knowledge of basic board structure and game rules (although, intriguingly, it seems to develop interpretable models of various strategic concepts (McGrath et al., 2021; Forde et al., 2022)).

Closer to the work described here–and a major motivation for our research–is Toshniwal et al. (2021), included in BIG-bench (Srivastava et al., 2022), which trains a language model on chess transcripts. They show strong evidence that transformer networks are building a representation of internal board state, but they stop short at investigating what form that representation takes. Our work can be seen as building on this line of research, with a focus on the geometry of internal representations.

The intervention technique we use in section 4 follows an approach of steering model output while keeping the model frozen. It is related to the ideas behind plug-and-play controllable text generation for autoregressive (Dathathri et al., 2019; Qin et al., 2020; Krause et al., 2020) and diffusion (Li et al., 2022) language models by optimizing the likelihood of the desired attribute and the fluency of generated texts at the same time. These methods naturally involve a trade-off and require several forward and backward passes to generate. Our proposed intervention method stands out by only working on internal representations and requires only one forward pass.

Finally, latent saliency maps can be viewed as a generalization of the TCAV (testing with concept activation vectors) approach Kim et al. (2018); Ghorbani et al. (2019); Koh et al. (2020). In the TCAV setting, attribution is performed via directional derivatives. This is essentially a linearization of the gradient-descent optimization used in our attribution maps.

7 Conclusion

Our experiments provide evidence that Othello-GPT maintains a representation of game board states—that is, the Othello "world"—to produce sequences it was trained on. This representation appears to be nonlinear in an essential way, as supported by the results of our linear probe and nonlinear probe experiments. Further, we find that these representations can be causally linked to how the model makes its predictions. Understanding of the internal representations of a sequence model is interesting in its own right, but may also be helpful in deeper interpretations of the network.

We have also described how interventional experiments may be used to create a "latent saliency map", which gives a picture, in terms of the Othello board, of how the network has made a prediction. Applied to two versions of Othello-GPT that were trained on different data sets, the latent saliency maps highlight the dramatic differences between underlying representations of the Othello-GPT trained on synthetic dataset and its counterpart trained on championship dataset.

There are several potential lines of future work. One natural extension would be to perform the same type of investigations with other, more complex games. It would also be interesting to compare the strategies learned by a sequence model trained on game transcripts with those of a model trained with a priori knowledge of Othello. One way to study this question is to compare latent saliency maps of Othello–GPT with standard saliency maps of an Othello-playing program which has the actual board state as input.

More broadly, it would be interesting to know how our results generalize to models trained on natural language. One stepping stone might be to look at language models whose training data has included game transcript. Will we see similar representation of board state? For more complex natural language tasks, can we find meaningful world representations? The tools described in this paper—nonlinear probes, layerwise interventions, and latent saliency maps—may yet prove useful in natural language settings.

REFERENCES

Mostafa Abdou, Artur Kulmizev, Daniel Hershcovich, Stella Frank, Ellie Pavlick, and Anders Søgaard. Can language models encode perceptual structure without grounding? a case study in color. *arXiv preprint arXiv:2109.06129*, 2021.

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, pp. 1–12, 2016.

Emily M Bender and Alexander Koller. Climbing towards nlu: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 5185–5198, 2020.

Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pp. 610–623, 2021.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

Steven Cao, Victor Sanh, and Alexander M Rush. Low-complexity probing via finding subnetworks. *arXiv preprint arXiv:2104.03514*, 2021.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. What you can cram into a single vector: Probing sentence embeddings for linguistic properties. *arXiv preprint arXiv:1805.01070*, 2018.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. Plug and play language models: A simple approach to controlled text generation. *arXiv preprint arXiv:1912.02164*, 2019.

Luciano Floridi and Massimo Chiriatti. Gpt-3: Its nature, scope, limits, and consequences. *Minds and Machines*, 30(4):681–694, 2020.

Jessica Zosa Forde, Charles Lovering, George Konidaris, Ellie Pavlick, and Michael L Littman. Where, when & which concepts does alphazero learn? lessons from the game of hex. In *AAAI Workshop on Reinforcement Learning in Games*, volume 2, 2022.

Amirata Ghorbani, James Wexler, James Y Zou, and Been Kim. Towards automatic concept-based explanations. *Advances in Neural Information Processing Systems*, 32, 2019.

Evan Hernandez and Jacob Andreas. The low-dimensional linear geometry of contextualized word representations. *arXiv preprint arXiv:2105.07109*, 2021.

John Hewitt and Christopher D Manning. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4129–4138, 2019.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pp. 2668–2677. PMLR, 2018.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *International Conference on Machine Learning*, pp. 5338–5348. PMLR, 2020.

Ben Krause, Akhilesh Deepak Gotmare, Bryan McCann, Nitish Shirish Keskar, Shafiq Joty, Richard Socher, and Nazneen Fatema Rajani. Gedi: Generative discriminator guided sequence generation. *arXiv preprint arXiv:2009.06367*, 2020.

Belinda Z Li, Maxwell Nye, and Jacob Andreas. Implicit representations of meaning in neural language models. *arXiv preprint arXiv:2106.00737*, 2021.

Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B Hashimoto. Diffusion-lm improves controllable text generation. *arXiv preprint arXiv:2205.14217*, 2022.

Reginald Long, Panupong Pasupat, and Percy Liang. Simpler context-dependent logical forms via model projections. *arXiv preprint arXiv:1606.05378*, 2016.

Thomas McGrath, Andrei Kapishnikov, Nenad Tomašev, Adam Pearce, Demis Hassabis, Been Kim, Ulrich Paquet, and Vladimir Kramnik. Acquisition of chess knowledge in alphazero. *arXiv preprint arXiv:2111.09259*, 2021.

William Merrill, Yoav Goldberg, Roy Schwartz, and Noah A Smith. Provable limitations of acquiring meaning from ungrounded form: What will future language models understand? *Transactions of the Association for Computational Linguistics*, 9:1047–1060, 2021.

Roma Patel and Ellie Pavlick. Mapping language models to grounded conceptual spaces. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=gJcEM8sxHK`.

Lianhui Qin, Vered Shwartz, Peter West, Chandra Bhagavatula, Jena Hwang, Ronan Le Bras, Antoine Bosselut, and Yejin Choi. Back to the future: Unsupervised backprop-based decoding for counterfactual and abductive commonsense reasoning. *arXiv preprint arXiv:2010.05906*, 2020.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419): 1140–1144, 2018.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*, 2022.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. Bert rediscovers the classical nlp pipeline. *arXiv preprint arXiv:1905.05950*, 2019.

Shubham Toshniwal, Sam Wiseman, Karen Livescu, and Kevin Gimpel. Learning chess blindfolded: Evaluating language models on state tracking. *arXiv preprint arXiv:2102.13249*, 2021.

A VISUALIZING THE GEOMETRY OF PROBES

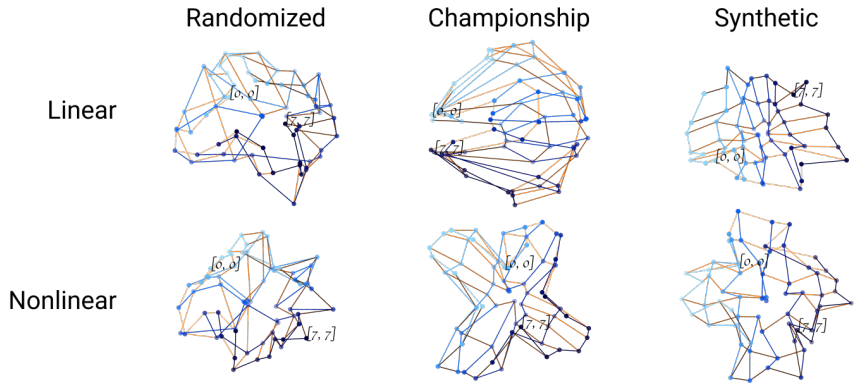

Figure 5: Geometry of probe weights. The first row is for linear probe weights; the second row is for the second-layer weights of the nonlinear probes. Three columns from left to right mean: probes trained on randomly-initialized GPT, on GPTs trained on championship and synthetic datasets respectively. For each subplot, one dot represents one tile on an Othello board, and neighbouring tiles have their two dots connected. Top-left and right-bottom corners are labelled for reference. Togglable 3D plots can be found in Supplementary Material.

Both linear and nonlinear probes can be viewed as geometric objects. In the case of linear probes, we can associate each classifier with the normal vector to the separating hyperplane. In the case of nonlinear probes, we can treat the second layer of the MLP as a linear classifier and take the same view. This perspective associates a vector to each grid tile, corresponding to the classifier for that grid tile.

A natural question is whether these vectors display any geometric patterns. Figure 5 visualizes their configuration using PCA plots. To make the connection with board geometry clear, we have overlaid a grid in which the vector for a given grid tile is connected to the vectors that represent its neighbors. At left, as a baseline, are weights of probes trained on randomized GPTs; the result is a somewhat jumbled version of the board's grid. For classifier vectors, however, we see a somewhat clearer geometric correlation with the grid shape itself. This shape may reflect a set of correlations between neighboring grid cells, and could be an interesting object for future study. One point of interest is that probe geometry for the randomized network does not seem completely random. This may fit with the fact that linear probe baseline performance is better than chance, indicating some information about board state can be gleaned from random projections of game move tokens.

B ABLATION ON NONLINEAR PROBE ACCURACIES

Though high accuracies have been observed on nonlinear probes in section 3, we want to develop a deeper understanding on them. For example, we wish to understand when, during a game, a model has developed world representations of board tiles, where in Othello-GPT that information is stored, how difficult it is to decode that information, and when the model may forget that information.

As shown in Figure 6 (B), we plot probe accuracies of two-layer probes varying to two different experiment settings: (1) Probe Hidden Units: the number of hidden units H in nonlinear probes; (2) Layer: at which layer the representations x^l is taken out. With the increase of hidden units, i.e., probe capacity, probe accuracy is higher as it can capture more knowledge from the hidden space. For layer l, the accuracy peaks at midway, which is aligned with studies on natural language (Hewitt & Manning, 2019), where linguistic properties are found to be best probed in mid-layers. The format of our *what-how-where* plots is similar to the *what-when-where* plots in McGrath et al. (2021), except we ask how many hidden units within our nonlinear probe are necessary to achieve reasonable accuracy given each layer of the GPT instead of looking into number of training epochs.

We are also curious about when (in terms of game progression) these concepts are captured by Othello-GPT. Are these concepts updated immediately after each move? Will they persist or be

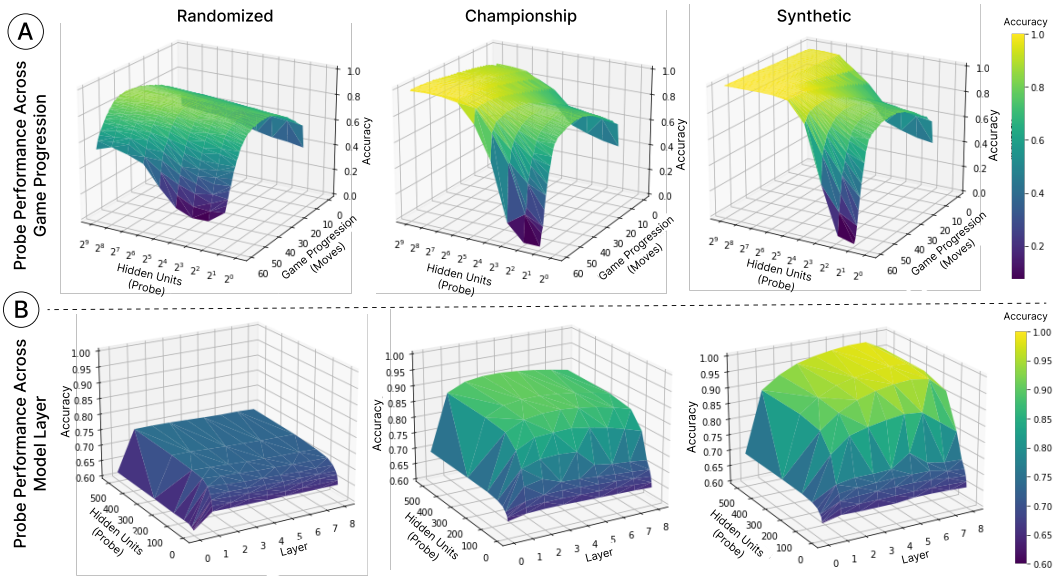

Figure 6: (A) *What-how-where* plots and (B) *What-how-when* plots of nonlinear probes trained on randomly-initialized GPT, on GPTs trained on championship and synthetic datasets respectively. (A) presents probe accuracy across an Othello game progression, while (B) presents accuracy across Othello-GPT layers.

forgotten with newer moves being made? To study this, we divide the data points for probe validation by how many steps the tile has been in its current state and plot *what* concept can be probed by *how* powerful probes *when* in the game progression in Figure 6 (A).

For nonlinear probes with a moderate number of hidden units, a parabolic accuracy curve is shown: concepts are best captured when they have existed for some time but not too long. This tells us: (1) forgetting happens when Othello-GPT changes its world representations; (2) there is a period of uncertainty before changes in board states are updated.

C Prediction heatmaps of counterfactual board states

In Figure 7, we can see one case of how intervention changes model prediction by intervening on the world representation of Othello-GPT. Note that the set of ground-truth legal moves are also changed by the intervention. In this case, both pre-intervention and post-intervention predictions have 0 errors. Figure 3 shows systematic results over $1,000$ cases. A random subset of 100 cases can be found in Supplementary Materials.

D Inter-probe Interaction

In Figure 8 we show the same case as in Appendix C of the world states probed from layers of Othello-GPT starting the 5th, which is the layer we start to intervene, L_S. We can observe that after intervention is successfully done on the 5th layer at C4, the flipped disc is corrected in the immediately following layer, as seen in the pre-intervention probing results at the 6-th layer. However, when the same intervention is done on later layers again, the model is more convinced that C4 should be black and stops to correct it. A total of 100 cases can be found in Supplementary Materials, corresponding to examples in Appendix C.

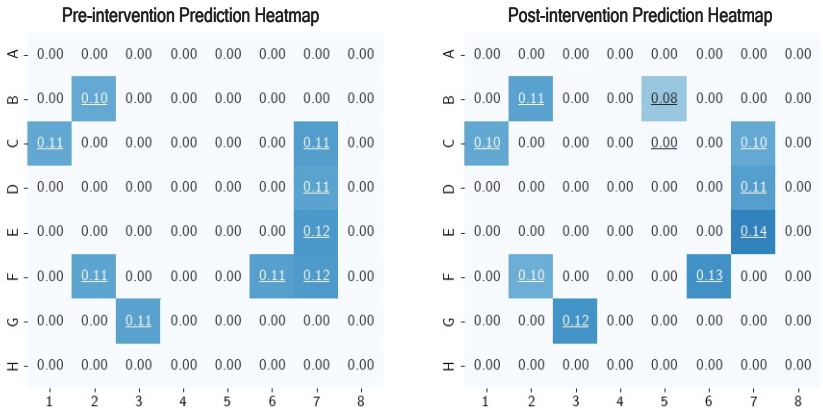

Figure 7: Heatmaps of the probability allocated to each tile on the board with nonlinear intervention done on Othello-GPT trained on synthetic dataset. To the left is the pre-intervention heatmap with pre-intervention legal moves underscored; to the right is post-intervention heatmap, with post-intervention legal moves underscored.

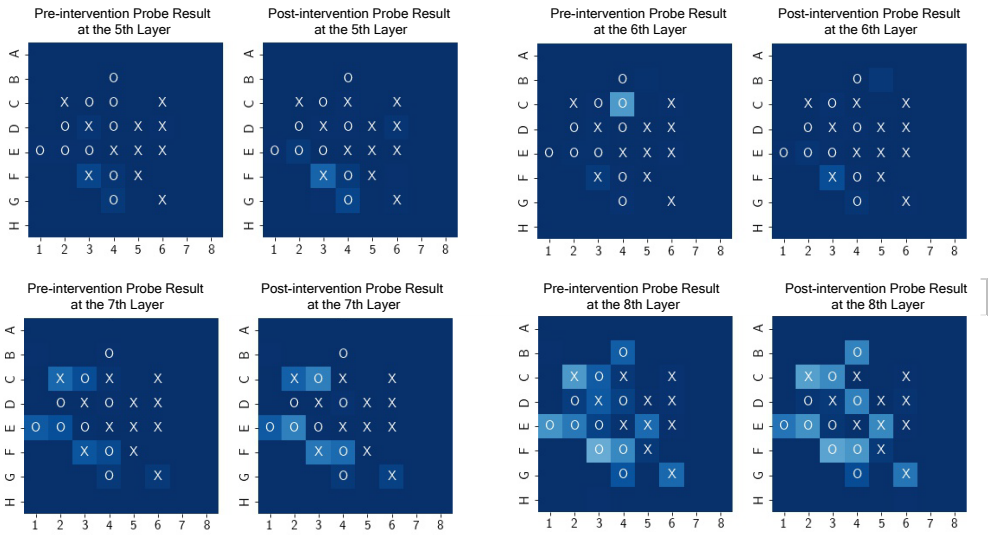

Figure 8: The world states predicted by probes pre-and-post-intervention on internal representations at different layers. Color encodes the confidence of the plotted top-1 prediction for each tile.

E Ablations on Intervention Hyper-parameters

Experiments find this optimization process is robust to different configurations of optimizer, learning rate α, and number of steps.

The complete world state, including states of all the 64 board tiles, is encoded in a single internal representation x, while during intervention experiment, we only wish to change one of them. A natural question is: will the intervention operation flip tiles we are not not intending? It is possible but we can mitigate that by considering the cross entropy losses of other tiles as a regularization term, weighted by a hyper-parameter β. That is to say, the loss in main paper can be expanded as:

$$\mathcal{L}_{\text{CE}}(p_\theta(x), B') = \sum_{s \in B \neq B'} \mathcal{L}_{\text{CE}}(p^s_\theta(x), B'(s)) + \beta \sum_{s \in B = B'} \mathcal{L}_{\text{CE}}(p^s_\theta(x), B(s)).$$

We sweep β at the best $L_S = 5$ on the factual benchmark and report average number of errors in Table 3. We can observe it does not clearly help.

β	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
error	0.22	0.31	0.77	0.19	0.51	0.65	0.42	0.45	0.46	0.55

Table 3: Average number of errors under different β's and $L_S = 5$ on the factual benchmark.

Here we further discuss another hyper-parameter ablated in Figure 3, the starting layer for intervention, L_S. If we intervene with more than 5 layers, shallow layers which have not developed reliable world representations (according to Figure 6) are touched, making intervention hazardous. On the other hand, if we intervene only at the deepest layers, though the world representation can be intervened successfully (see Appendix D), the model does not have enough computation to adapt to the newer world representation and make predictions corresponding to it.