Scaling Effects on Latent Representation Edits in GPT Models (Student Abstract)

Austin L. Davis and Gita Sukthankar

Department of Computer Science, University of Central Florida
4000 Central Florida Blvd, Orlando, FL 32816-2362
austindavis@ucf.edu, gita.sukthankar@ucf.edu

Abstract

Probing classifiers are a technique for understanding and modifying the operation of neural networks in which a smaller classifier is trained to use the model's internal representation to learn a related probing task. Similar to a neural electrode array, training probing classifiers can help researchers both discern and edit the internal representation of a neural network. This paper presents an evaluation of the use of probing classifiers to modify the internal hidden state of a chess-playing transformer. We demonstrate that intervention vector scaling should follow a negative exponential according to the length of the input to ensure model outputs remain semantically valid after editing the residual stream activations.

Introduction

Recent methods enable translation from the latent space vectors from a GPT's residual stream (i.e., the hidden state) into human-comprehensible features (Nanda, Lee, and Wattenberg 2023). One technique that has shown promise trains a linear probe classifier (Belinkov 2022) to predict domain specific knowledge directly from the residual stream activations of a model. Our research explores the causal relationships between weight vectors of a linear probe classifier and the semantic validity of the GPT's output. However, given the complexities of natural language, we constrain our research to GPTs trained in a domain characterized by strict logical rules: the game of chess (Toshniwal et al. 2022).

In this paper, we train a linear probe to predict the board state directly from the residual stream activations of a GPT trained on sequences of chess moves. Once the residual stream activations are decoded, we reverse the process: by adding or subtracting scaled weight vectors from the linear probe classifier to the residual stream, we effectively modify GPT's emergent, internal model of the board, adding or removing pieces at will (Karvonen 2024). Our goal is to investigate how the *scaling* of this intervention vector affects the semantic validity (i.e. move legality) of the GPT's post-intervention output, and it has implications for controlling the emergent representations of GPTs trained on NLP tasks.

Methodology

We trained a 12-layer GPT-2 to perform next-token prediction exclusively on UCI-encoded chess move sequences. Our GPT was given no *a priori* knowledge about the game of chess, but it was nonetheless able to recommend legal moves 99.9% of the time across a hold-out evaluation dataset of games played by human-players. We then trained a linear model, $P_s^\ell : \mathbf{X} \mapsto \mathcal{Z}$, to classify board position from the GPT's intermediate activations at layer ℓ, where $z_s \in \mathcal{Z} = \{P, N, B, R, Q, K, p, n, b, r, q, k, \emptyset\}$ indicates the type of piece positioned on square s using typical chess piece abbreviations or else $\emptyset$ when no piece is present. The probe training data is a set $\mathbf{X} = \{x_i^\ell\}$ of activation vectors at position i and layer ℓ cached from forward passes of the GPT on a set of 120k games. The probe's average classification performance exceeds 0.90 according to the F1-score, accuracy, precision, and recall metrics, and the probe exceeds 0.94 in each metric on layers 7 and 8.

To edit the residual stream, we select an intervention vector $\mathbf{u}_i^\ell$ for each position i and each layer ℓ and modify the GPT's residual stream activation $\mathbf{x}_i^\ell$ as follows:

$$\mathbf{x}_i^\ell \leftarrow \mathbf{x}_i^\ell + \eta \cdot \frac{\mathbf{u}_i^\ell}{\left\|\mathbf{u}_i^\ell\right\|} \cdot \left\|\texttt{LayerNorm}(\mathbf{x}_i^\ell)\right\|$$

where $\eta \in \mathbb{R}$ and $\mathbf{u}_i^\ell$ is simply one of the column vectors from the weights of the linear probe. Since each column vector is associated with a piece type in $\mathcal{Z}$, we can add/remove piece k from square s by setting the intervention vector to the positive/negative of the k-th column vector in the probe weight matrix, i.e., $\mathbf{u}_i^\ell = P_s^\ell[k]$, or when multiple edits are required we simply sum these column vectors.

For instance, Figure 1 shows how probe-based edits to the GPT's activations cause it to make legal moves according to a board which differs from that of the the input sequence. In this instance, we constructed an intervention vector $\mathbf{u}$ from the k^{th} column associated with black pawns from the probe weight matrix as follows:

$$\mathbf{u}^\ell \leftarrow -P_{\texttt{e4}}^\ell[k]$$

This intervention vector precisely erases the white pawn on `e4` from the residual stream representation of the board.

We aim to measure the effect of η on output validity post-intervention. To accomplish this, we first compute a forward

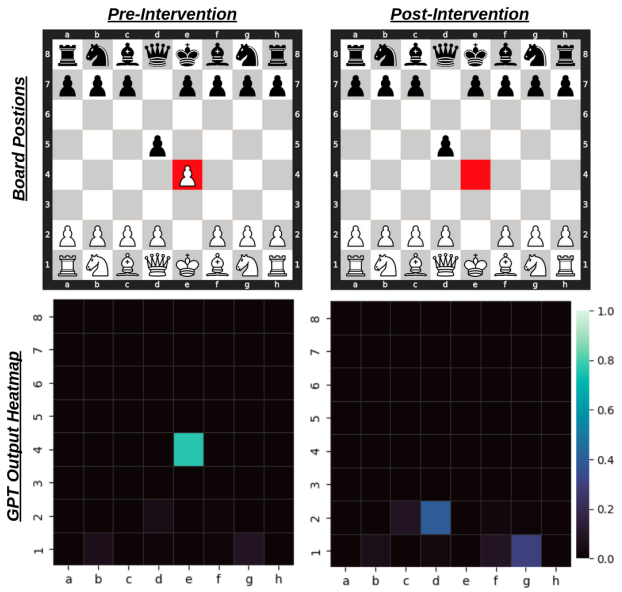

Figure 1: An intervention on the *King's Pawn* opening that removes white's e4 pawn by directly editing latent space activations of the GPT. Pre-intervention, the GPT recommends moving white's `e4` pawn with probability mass 0.764, but this is an illegal move post-intervention. Using $\eta = 1$, the intervention moves 0.39 and 0.30 of the probability mass to the `d2` pawn and `g1` knight, respectively. A heatmap of the pre- and post-intervention outputs is shown at the bottom.

pass for 500 games pre-intervention to determine the square s and piece type k the GPT recommends moving at each token position i. We then construct an intervention vector $\mathbf{u}_i^\ell \leftarrow -P_s^\ell[k]$ to remove that piece from the GPT's latent board representation. This ensures the model's top-1 output *pre*-intervention is invalid as output *post*-intervention.

We performed the interventions while adjusting the intervention vector's scale $\eta \in [0, 2]$ in increments of 0.05. We calculate the post-intervention output's validity according to what the board state *would have been* had the intervention been successful. Specifically, we reconstruct the board state B_i using an external chess engine and compute the set of legal moves from the subgame up to token i . We then compute the legal move probability mass (LMPM) by summing the softmax of the output logits post-intervention across only the tokens which are legal according to B_i. High values of LMPM indicate a successful intervention where the model reoriented its output probability distribution away from the removed piece and toward other tokens that are legal for B_i.

Results

We computed mean LMPM for the first 25 moves (50-ply) of the 500 games in our sample. Mean LMPM varies based on the move and intervention vector scale η (see Figure 2).

The first move is peculiar because it requires a larger η to be effective and over-scaling η has few consequences. For all other moves, when an appropriate η is chosen (e.g., 0.3), the intervention succeeds 92% of the time ($\sigma = 0.02$). Yet, the

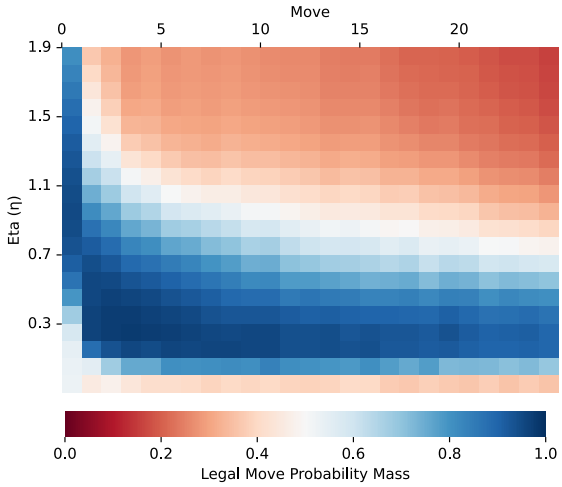

Figure 2: Legal move probability mass post-intervention. Values are averaged over 500 random games of chess.

intervention is highly sensitive to η values outside a narrow Goldilocks region which varies across moves. For a given move x, the center of this region is well approximated by:

$$H(x) = 0.3 + 0.7e^{-x}.$$

We hypothesize the middle of this region decays exponentially over the input sequence because, on average, each subsequent token contributes proportionally less to the residual stream than its predecessor tokens. Furthermore, we conjecture the region simultaneously narrows because the information in the residual stream becomes more nuanced later in an input sequence, and large η values risk overwriting that nuance, corrupting the GPT's internal, emergent world model.

Conclusion and Future Work

Our research demonstrates the effect of scaling on semantic validity during latent space edits. Future work will investigate these effects on GPTs trained on natural language.

Acknowledgments

The conclusions and opinions expressed in this research paper are those of the authors and do not necessarily reflect the official policy or position of the U.S. Government or Department of Defense.

References

Belinkov, Y. 2022. Probing Classifiers: Promises, Shortcomings, and Advances. *Comput'l Linguistics*, 48(1): 207–219.

Karvonen, A. 2024. Emergent World Models and Latent Variable Estimation in Chess-Playing Language Models. In *CoLM*. Philadelphia, PA, United States.

Nanda, N.; Lee, A.; and Wattenberg, M. 2023. Emergent Linear Representations in World Models of Self-Supervised Sequence Models. In *6th BlackboxNLP Workshop*, 16–30.

Toshniwal, S.; Wiseman, S.; Livescu, K.; and Gimpel, K. 2022. Chess as a Testbed for Language Model State Tracking. *AAAI*, 36(10): 11385–11393.