Emergent World Beliefs: Exploring Transformers in Stochastic Games

Anonymous Author(s)
Affiliation
Address
email

Abstract

Transformer-based large language models (LLMs) have demonstrated strong reasoning abilities across diverse fields, from solving programming challenges to competing in strategy-intensive games such as chess. Prior work has shown that LLMs can develop emergent world models in games of perfect information, where internal representations correspond to latent states of the environment. In this paper, we extend this line of investigation to domains of incomplete information, focusing on poker as a canonical partially observable Markov decision process (POMDP). We pretrain a GPT-style model on Poker Hand History (PHH) data and probe its internal activations. Our results demonstrate that the model learns both deterministic structure—such as hand ranks—and stochastic features—such as equity—without explicit instruction. Furthermore, by using primarily nonlinear probes, we demonstrated that these representations are decodeable and correlate with theoretical belief states, suggesting that LLMs are learning their own representation of the stochastic environment of Texas Hold'em Poker.

1 Introduction

Current transformer-based large language models (LLMs) have achieved breakthrough results across various tasks, ranging from answering industry programming questions to solving olympiad-level problems. [Tschisgale et al., 2025, Jain et al., 2024]. The ability of LLMs to complete these tasks lies in their advanced reasoning capabilities, which are extremely evident when playing reasoning-intensive games such as chess [Zhang et al., 2025].

Despite these achievements in LLM reasoning capabilities, the internal execution of their strategies remains a "black-box". Recently, research on LLMs with *internal world representations* has grown to demonstrate higher-level LLM understanding in games as seen in Karvonen [2024] and Li et al. [2024]. The findings of Li et al. [2024] demonstrate the OthelloGPT model's ability to develop its own internal representation of the game states and rules of Othello from move strings in a strictly deterministic, perfect-information setting, with the hope that natural-language models are learning broader semantic "world representations". In this paper, we extend this analysis to world models in games of incomplete information, in particular Poker, to explore how LLMs intrinsically model uncertainty in a Bayesian fashion and provide new insights into their decision-making process.

Our paper's main contributions include: The (1) **extension of LLM internal world representation to games of incomplete information** and (2) **a quantifiable understanding of circuits/features that underlie an LLM's "belief state" for partially observable Markov Decision Processes** (explored through Poker). We defer a theoretical analysis of the second focus to appendix section B, and a further discussion on LLM world models to C.

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

2 Related Works

This paper builds upon the work of Karvonen [2024], Li et al. [2024] and Nanda et al. [2023] who trained language models to play complete information games such as Chess and Othello. Li et al. [2024] demonstrated that OthelloGPT, a LLM trained on sequences of legal moves in Othello spontaneously developed an internal representation of the game board that could be extracted with nonlinear probes. Later it was demonstrated by Nanda et al. [2023] that linear probes could extract this representation as well. Karvonen [2024] extended these findings from Othello to the game of Chess as well, and also demonstrated models that developed these world representations also had the capability to understand and estimate latent variables such as player skill.

3 Poker Model

As a foundation for our studies on LLMs in POMDPs/stochastic games, we pretrain a GPT-style architecture on Poker games, using the Poker Hand History (PHH) format [Kim, 2024] (see Appendix D).

3.1 Dataset

As noted by past papers exploring emergent world representations such as Karvonen [2024], dataset size plays a large role in probe results. Due to the unavailability of large and complete poker hand datasets, we opted to generate our own. We did this by utilizing a large number of game simulations to determine poker equity[Billings et al., 1999] which then drives decision making. Our simulation script generates legal six-player No-Limit Texas Hold'em hands in PHH format. To start each hand, we give each player fresh stacks and randomly initialize their playing style to ensure that there is variation in agent behavior. Agents use a combination of simulation equity estimates and heuristics based on their randomly initialized playing style to make decisions, driving realistic and diverse poker games.

3.2 Training

We fine-tuned a causal transformer language model based on GPT-2 [Radford et al., 2019] using a PHH-formatted dataset comprising over two million synthetically generated poker trajectories, as described above. The model retained the GPT-2 base configuration, with 12 attention heads and a hidden dimensionality of 768. Each hand was tokenized using a custom `PreTrainedTokenizerFast` vocabulary. During preprocessing, for a subset of tokens, we insert a reserved special `<GAP>` token in the input and shift the replaced original token to appear following a special `<ANS>` token later in the sequence. In training, we compute loss exclusively on the tokens that follow `<ANS>` to ensure proper loss calculation.

Optimization used AdamW with $\beta_1 = 0.9$, $\beta_2 = 0.95$, and $\epsilon = 10^{-8}$, with a commonly used learning rate of 5×10^{-5}. We trained for 13 epochs (training was paused after validation loss stopped improving), using an effective batch size of 128 (minibatch size 64 with gradient accumulation of 2). Checkpoints were saved every 5,000 steps and the best-performing checkpoint by validation loss was stored separately. We used a 95-5 train-test split for model training. See Appendix G for more training details.

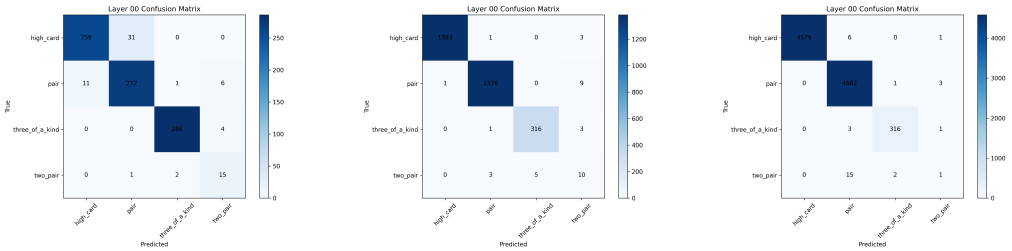
4 Probing Internal Representations

We distinguish between two types of internal representations in our analysis: deterministic representations, which capture absolute aspects like hand rank and actions, and stochastic representations- such as equity- to extract the model's internal belief state of the underlying Poker POMDP (Shai et al. [2024]). To capture these results, we probe internal activations using a linear classifier probe and a two-layer multilayer perceptron (MLP), a technique frequently used ([Li et al., 2024, Hernandez and Andreas, 2021]). The function of a linear probe used for our deterministic model is $p_\theta(x_t^l) = \text{argmax}(Wx_t^l)$, where $\theta = \{W \in \mathbb{R}^{C \times F}\}$, where F is the number of dimensions of the input activation vector x_t^l. The function used for our two-layer MLP probe is $p_\theta(x_t^l) = \text{argmax}(W_1 \,\text{ReLU}(W_2 x_t^l))$, where $\theta = \{W_1 \in \mathbb{R}^{C \times H}, W_2 \in \mathbb{R}^{H \times F}\}$, H is the number

of hidden dimensions for the nonlinear probes, where C denotes the number of classes under consideration for identification (typically $C = 4$ in our context). For our stochastic representation probes, `argmax` was not used as the output was continuous. Overall, the MLP probe achieved higher accuracy than the linear probe, consistent with the findings of Li et al. [2024].

4.1 Deterministic World Model

Expanding on Li et al. [2024], we extract basic deterministic game features such as hand-rank through activation probing. Hand-rank represents the categorical value of a player's hole cards in the context of board cards and is strictly deterministic in our setting. We train a separate probe on each layer of the model to note any possible variations between layers that may signify that layer's responsibility in output generation. To prevent over-representation of frequent hand-ranks (e.g. high_card and pair) and misidentification of internal representational states, we balanced the dataset by capping each class at the 40th percentile of unique hand-rank counts (Appendix H for more details on this). The linear probe achieves 80% accuracy for identifying hand-rank, while the MLP reaches 98% accuracy (1c, similar to results shown in Li et al. [2024]. These results are measured on a dataset excluded from training of the GPT-based model as well as separate from probe training (extended results in Appendix J). As shown in Figure 1c, the prominent diagonal in the confusion matrix indicates high class-wise accuracy, demonstrating that the internal activations of the model reliably encode the rank of the hand, which implies that the model is developing an internal representation of poker hand states, rather than just memorizing statistics.



(a) MLP probe hand-rank identification on Layer 0 – 30th percentile equation.

(b) MLP probe hand-rank identification on Layer 0 – 35th percentile equation.

(c) MLP probe hand-rank identification on Layer 0 – 40th percentile equation.

Figure 1: MLP probe confusion matrices for hand-rank identification on Layer 0 using different percentile equations. Representation of rarer hand-ranks is improved with lower percentiles. See Appendix I for additional deterministic experimental results.

4.2 Stochastic World Model

To demonstrate our hypothesis of the language model having internal representations corresponding to the internal representation of the belief state over the POMDP, we trained a two-layer MLP using simulation based equity estimations [Billings et al., 1999] as our label. For this dataset, we intentionally masked out all hole cards except for those belonging to player one. From our trained probe, we were able to achieve a correlation coefficient of 0.50 on our test dataset predictions (Figure 2). This correlation between model activations and the predicted winning potential of a given hand demonstrates that our GPT model has spontaneously developed some internal representation of the hand strength.

4.3 Activation Maps

As a validation of the LLM's world representation, we observe its ability to discern hands and patterns of different strategic value in poker. We visualized activations using PCA, t-SNE, and UMAP (see Appendix K for extended results). Figure 3b reveals distinct clusters, indicating that the model organizes its representations in accordance with hand rank, pairs, and three-of-a-kind clusters closely, thus demonstrating its ability to learn game-level concepts from unsupervised data. Note that the presence of multiple clusters for hand-ranks such as pair indicates that the model is learning a

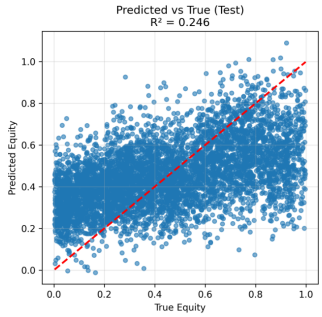
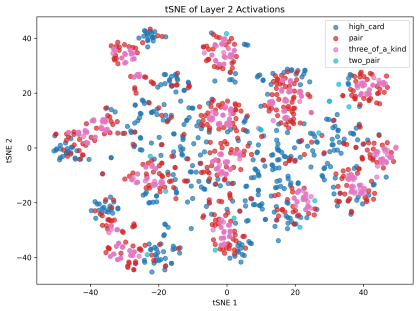
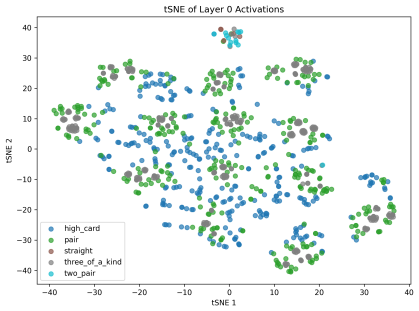


Figure 2: Scatter plot of predicted versus true equity values using the MLP probe on layer 11.

broader representation of pair, where each cluster likely refers to a subset of pairs (the types of cards encompassed in the pair).



(a) t-SNE visualization of Layer 2 activations. Points represent a single hand colored by rank.



(b) t-SNE visualization of Layer 0 activations. Points represent a single hand colored by rank. Note the two_pair cluster near the top.

Figure 3: t-SNE visualized activation plots. Activations are clustered by hand-rank and conceptual similarity.

5 Limitations

Dataset size remains a core limitation of our experiments, as with our generation method it becomes computationally expensive to generate extremely large amounts of data. The size of our dataset impacts the model accuracy on its task as well as results obtained from probes [Karvonen, 2024]. Our deterministic probing analysis is also limited by this factor, as there are insufficient examples of rarer hands such as straights or flushes in our generated datasets for us to accurately probe for them. Additionally, the process for generating data may be overly simplified in relation to the complexity of poker interactions, potentially impacting analysis results. Finally, there are no guarantees that current results regarding LLM beliefs of the Poker POMDP are able to be extended to analysis of new stochastic poker variables.

6 Conclusion and Future Works

We demonstrated that a GPT-2-based model trained on PHH-style data can develop a deterministic understanding of the game state as well as an understanding of stochastic game elements. This brings us closer to extending the emergent world model hypothesis to games characterized by incomplete information. To extend our work, we hope to further scale our base LM and formalize our intuitions of LLM Bayesian behavior—in particular, extracting LLM beliefs of the Poker POMDP (see Appendix B for theory)—and better understand the structure of LLM predictive world representations through SAEs and further probing experiments.

References

Anthropic. A mathematical framework for transformer circuits. Transformer Circuits, 2021. URL https://transformer-circuits.pub/2021/framework/index.html.

Nora Belrose, Igor Ostrovsky, Lev McKinney, Zach Furman, Logan Smith, Danny Halawi, Stella Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the tuned lens. *arXiv:2303.08112*, 2023. URL https://arxiv.org/abs/2303.08112.

Darse Billings, Denis Papp, Lourdes Pena, Jonathan Schaeffer, and Duane Szafron. Using selective-sampling simulations in poker. In *Proceedings of the AAAI Spring Symposium on Search Techniques for Problem Solving under Uncertainty and Incomplete Information*, Stanford, California, 1999. AAAI Press. URL https://poker.cs.ualberta.ca/publications/AAAISS99.pdf.

H. Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. *arXiv:2309.08600*, 2023. URL https://arxiv.org/abs/2309.08600.

Nelson Elhage, Tristan Hume, Catherine Olsson, et al. Toy models of superposition. *arXiv:2209.10652*, 2022. URL https://arxiv.org/abs/2209.10652.

Wes Gurnee and Max Tegmark. Language models represent space and time. In *ICLR*, 2024.

David Ha and Jürgen Schmidhuber. World models. *arXiv:1803.10122*, 2018. URL https://arxiv.org/abs/1803.10122.

Evan Hernandez and Jacob Andreas. The low-dimensional linear geometry of contextualized word representations, 2021. URL https://arxiv.org/abs/2105.07109.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code, 2024. URL https://arxiv.org/abs/2403.07974.

Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *MIT CSAIL*, 101(1–2):99–134, 1998. URL https://people.csail.mit.edu/lpk/papers/aij98-pomdp.pdf.

Adam Karvonen. Emergent world models and latent variable estimation in chess-playing language models. In *COLM*, 2024. URL https://arxiv.org/abs/2403.15498.

Juho Kim. Recording and describing poker hands. In *IEEE Conference on Games*, 2024. doi: https://doi.org/10.1109/CoG60054.2024.10645611. URL https://arxiv.org/abs/2312.11753.

Kenneth Li, Aspen K. Hopkins, David Bau, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Emergent world representations: Exploring a sequence model trained on a synthetic task, 2024. URL https://arxiv.org/abs/2210.13382.

Michael L. Littman, Richard S. Sutton, and Satinder Singh. Predictive representations of state. In *NeurIPS*, 2001. URL https://papers.nips.cc/paper/1983-predictive-representations-of-state.pdf.

Aaron Mei et al. Understanding how chess-playing language models compute linear representations of board state. In *MOSS@ICML*, 2025. URL https://openreview.net/pdf?id=Z9OV9NygER.

Neel Nanda, Andrew Lee, and Martin Wattenberg. Emergent linear representations in world models of self-supervised sequence models. 2023. URL https://arxiv.org/abs/2309.00941.

nostalgebraist. Interpreting gpt: the logit lens. LessWrong, 2020. URL https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners, 2019. URL https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

Adam Shai, Paul M. Riechers, Lucas Teixeira, Alexander Gietelink Oldenziel, and Sarah Marzen. Transformers represent belief state geometry in their residual stream. In *Proceedings of the 38th Conference on Neural Information Processing Systems (NeurIPS)*, December 2024. URL https://openreview.net/forum?id=YIB7REL8UC. OpenReview ID: YIB7REL8UC.

Satinder Singh, Michael James, and Matthew Rudary. Predictive state representations: A new theory for modeling dynamical systems. In *UAI*, 2004.

R. D. Smallwood and E. J. Sondik. The optimal control of partially observable markov processes over a finite horizon. *Operations Research*, 21(5):1071–1088, 1973. URL https://www.jstor.org/stable/168926.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*, 2024. URL https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html.

Paul Tschisgale, Holger Maus, Fabian Kieser, Ben Kroehs, Stefan Petersen, and Peter Wulff. Evaluating gpt- and reasoning-based large language models on physics olympiad problems: Surpassing human performance and implications for educational assessment. *Phys. Rev. Phys. Educ. Res.*, pages –, Jun 2025. doi: 10.1103/6fmx-bsnl. URL https://link.aps.org/doi/10.1103/6fmx-bsnl.

Yinqi Zhang, Xintian Han, Haolong Li, Kedi Chen, and Shaohui Lin. Complete chess games enable llm become a chess master. In *NAACL*, 2025. URL https://arxiv.org/abs/2501.17186.

A Appendix

Our code can be found at:

- https://anonymous.4open.science/r/poker-interp-4653/

B Theoretical Justification of Bayesian World Models

In this section, we motivate LLMs as MLE learners in a POMDP setting, and formalize the connection between these LLM probabilistic "belief states" and concrete residual stream activations that are separated through linear probes. Past work such as Shai et al. [2024] demonstrates that geometry of a Hidden Markov Model's belief states can be recovered in the residual stream of a transformer. In this work, we formalize the meaning of an LLM "belief state" for next-token prediction that represents the trajectory of partially observable Markov processes.

B.1 Belief State and Poker MDP Definitions

Consider Z as some unobserved latent world state in the LLM (i.e. the board state in Othello, hidden cards in Poker, semantic topics of conversation in NLP) along with a history of tokens up to time t as $h = (x_1, \cdots, x_t)$. The true next token distribution is given as

$$p(x_{t+1}|h_t) = \sum_z p(x_{t+1}|z, h_t)p(z|h_t)$$

So the next token depends on our distribution over latent states $p(z|h_t)$, which we encode as our **LLM belief state**. For an ideally trained LLM in deterministic games of chess/Othello, this distribution is just an indicator of the deterministic board state given a sequence of moves, but for games such as poker we uncover a nontrivial distribution over latent space.

In short, *for next-token prediction in a POMDP setting, the LLM must carry information at least as strong as* $p(z|h_t)$. We view Poker as such a partially observable Markov Decision Process (POMDP), defining states S as the full game specification with a partially observable subset O representing

outside cards and behaviors, actions $\mathcal{A}$ as a single player's options (raising, folding, calling, etc), and the transition dynamics $\mathcal{T}$ representing stochasticity over dealt cards as well as other player's actions.

From standard POMDP analysis, we recall that the belief state is a sufficient statistic for decision-making (in our setting, next-token prediction) Kaelbling et al. [1998].

B.2 Linearity of Predictions in the Belief State

For POMDPs over deterministic games as explored in Li et al. [2024], we can justify the use of linear probing through a theoretical analysis as predictions of the future as linear functions on the LLM's belief state.

Linearity Lemma: Given our Poker POMDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{O}, T, \Omega)$ and a policy π (our sequence model trained on poker games), let H_t be the history up to t and $b_t \in \Delta(\mathcal{S})$ the belief $b_t(s) = \Pr(S_t = s \mid H_t)$. For any finite-horizon future event F measurable w.r.t. $(S_{t+1:T}, O_{t+1:T}, A_{t:T-1})$ under π, there exists a vector $v_F \in \mathbb{R}^{|\mathcal{S}|}$ such that

$$\Pr_{\pi}(F \mid H_t) \;=\; \sum_{s \in \mathcal{S}} b_t(s)\, v_F(s) \;=\; \langle b_t, v_F \rangle.$$

In other words, our prediction of future events given our current history under the policy is *linear* in our defined belief state. In particular, then for each observation $o \in \mathcal{O}$,

$$\Pr_{\pi}(O_{t+1} = o \mid H_t) \;=\; \langle b_t, v_o \rangle,$$

where v_o is a per-observation vector of weights (in practice, learned by a linear probe) so the entire next-observation distribution is an affine linear map of b_t.

Proof:

Let $f = \mathbf{1}_F$ be the indicator of F. By the tower rule,

$$\mathbb{E}_{\pi}[f \mid H_t] = \sum_{s \in \mathcal{S}} \Pr_{\pi}(S_t = s \mid H_t)\, \mathbb{E}_{\pi}[f \mid S_t = s, H_t].$$

In a POMDP, the controlled Markov property implies that, given $S_t = s$ (and with policy π fixed), the distribution of $(S_{t+1:T}, O_{t+1:T}, A_{t:T-1})$ does not depend on the specific past H_t (i.e. our token histories); hence $\mathbb{E}_{\pi}[f \mid S_t = s, H_t] = \mathbb{E}_{\pi}[f \mid S_t = s] v_F(s)$ by conditional independence.

Then, this directly gives us the probabilities as $\Pr_{\pi}(F \mid H_t) = \sum_s b_t(s)\, v_F(s) = \langle b_t, v_F \rangle$. Taking $F = \{O_{t+1} = o\}$ yields the desired linearity of observation result.

So the PODMP belief state "automatically" gives us linearity! Intuitively, if the transformer trained on next-token prediction does in fact hold its belief state in the residual stream, then any predictive probe should be *linear* in these activations. Consider the following toy example:

Suppose we have a simple binary hypothesis testing $Z = \{0, 1\}$ to denote which coin is in use among two coins with probabilities p_1, p_2. Conditional on Z, our observations $x_1, \cdots, x_t \in \{H, T\}$ are id with $P(x_i = H | Z = z) = p_z, P(x_i = T | Z = z) = 1 - p_z, z \in \{0, 1\}$. Our log likelihood ratio (LLR) of one hypothesis over the other evolves with ratios $\log(\frac{\theta_1}{\theta_0})$ and $\log(\frac{1-\theta_1}{1-\theta_0})$. Our log likelihood ratio η_t by assumption exists in the residual stream of our sequence prediction model.

How does the residual stream "provide" the belief coordinate? Our connection between linear probes and POMDP belief states lies in the LLM's residual stream, motivated from the theory of transformer circuits Anthropic [2021]. Let $r_t \in \mathbb{R}^d$ denote the residual stream at position t. Assume the model stores η_t along a direction $v \in \mathbb{R}^d$:

$$r_t \;\approx\; r_0 \;+\; \eta_t\, v \;+\; \varepsilon_t \;=\; r_0 \;+\; \Big(\eta_{t-1} + \mathrm{LLR}(x_t)\Big)\, v \;+\; \varepsilon_t, \tag{1}$$

so residual *addition* implements evidence accumulation. With a fixed unembedding $U \in \mathbb{R}^{d \times |\mathcal{V}|}$, the logits satisfy

$$\mathrm{logits}(x \mid x_{1:t}) \;=\; U_x^{\top} r_t + c_x \;\approx\; (U_x^{\top} v)\, \eta_t \;+\; \text{const}, \tag{2}$$

and so we get a linear function of the belief coordinate. Ultimately, we get that a linear probe w can recover η_t from r_t via $\hat{\eta}_t = w^{\top} r_t$.

Related works explore the theoretical justifications of linear probes in partially observable processes. Two complementary works make a precise claim in this direction: *predictions are linear functionals of state*.

POMDPs. Under a fixed policy, the *belief* b_t (posterior over latent state) is a sufficient statistic for prediction/control; for any finite-horizon event F, $\Pr(F \mid H_t) = \langle b_t, v_F \rangle$ for some vector v_F determined by the dynamics/observations [Kaelbling et al., 1998, Smallwood and Sondik, 1973]. Thus the entire next-observation distribution is an affine-linear map of b_t.

PSRs. If the Hankel matrix of future-test probabilities has finite rank k, there exists a k-dimensional *predictive state* $p(h)$ (probabilities of *core tests*) such that the probability of *any* test τ is linear: $\Pr(\tau \mid h) = c_\tau^\top p(h)$ [Littman et al., 2001, Singh et al., 2004]. Hence, *if* a transformer stores an affine transform of b_t or $p(h)$ in its residual stream, there *exists* a linear probe (and the model's own unembedding) that recovers the relevant prediction

B.3 Relation to poker (this paper)

Poker is a canonical partially observable domain where the minimal predictive state is a *belief over hidden hands* (often summarized as a range). Training a next-token model on poker strings (actions and reveals) creates direct pressure to maintain this belief internally, because the Bayes-optimal next-token distribution is the mixture over hypotheses weighted by the current belief. Our empirical program—linear probes for range log-odds, layerwise tuned-lens trajectories, and causal edits along probe directions—follows the Othello/chess playbook while grounding interpretation in POMDP/PSR sufficiency. This explains why (i) range features should be linearly decodable, (ii) updates should approximate additive log-likelihood ratios upon new evidence, and (iii) editing decoded belief directions should steer action logits in predictable ways.

C LLM World Models

In this section we further explore the literature of LLM world models and discuss our contributions in the context of prior work.

In the previous section, we formalize our definition of world models/belief state for POMDPs. In the case of OthelloGPT, this world model takes the form of a representation of the board state/dynamics in the residual stream, but in our Poker case, the relevant latent is instead a belief (range) over hidden information, such as player hands and strategies/deck cards.

C.1 What counts as a world model?

Broader than POMDPs and games, a world model functions as an internal state whose evolution under the model approximates the latent state/dynamics of the data-generating process, such that predictions are a (typically affine-linear) functional of that state. Early neural control work formalized this idea broadly as "world models" [Ha and Schmidhuber, 2018]. In LLMs, the clearest evidence comes from synthetic or structured domains where latent state is objectively defined and recoverable from strings. The primarily goal of LLM world model research in toy settings such as Poker and Othello is to find strong signals that LLMs can learn higher-order structure (translated to the language setting, higher-order emotions/rationalities) from sampled sequences of these unobserved transition dynamics.

C.2 Empirical evidence in trained sequence models

Board games. In *Othello-GPT*, a small transformer trained only to predict legal moves (no board supervision) learns an internal representation of the full board: probes decode square occupancy; causal interventions flip squares and reliably change downstream moves [Li et al., 2024]. Follow-up work in chess reaches similar conclusions: linear decoders recover piece/square features and editing those features predictably shifts move probabilities, indicating persistent board-state coordinates in the residual stream [Mei et al., 2025].

Space & time. When trained on ordinary text corpora, LLMs encode geometric and temporal structure that is linearly recoverable: e.g., countries/cities embed into coherent low-dimensional

coordinate systems and historical entities align along temporal axes [Gurnee and Tegmark, 2024]. These are not merely lexical clusters but approximately *metric* maps, suggesting latent factors aligned with world structure.

C.3 Mechanistic lenses on storage and update

Two families of tools consistently reveal how predictions depend on internal state.

Layerwise decoding: The *logit lens* and the calibrated *tuned lens* linearly decode token distributions from intermediate residual streams, showing a monotone refinement of predictions across depth—consistent with iterative inference/update of a persistent state carried forward by residual addition [nostalgebraist, 2020, Belrose et al., 2023]. In practice, we observe this through the refinement of LLM confidence in poker games as more cards are dealt.

Feature decomposition: Sparse autoencoders (SAEs) and related dictionary-learning methods recover more monosemantic directions in residual space (e.g., individual board squares, entity features), addressing superposition and enabling targeted causal edits [Cunningham et al., 2023, Templeton et al., 2024, Elhage et al., 2022]. These results support a picture in which a small set of task-relevant *state features* are embedded (often nearly linearly) and read out by the fixed unembedding matrix.

D PHH Formatting

In PHH notation cards abbreviated to a rank followed by a suit (King of Hearts -> Kh). Table 1 shows how actions are represented in PHH notation.

Player Actions	
Standard Representation	**PHH Representation**
Hole Cards Dealt	`d dh pN card(s)==s)`
Board Cards Dealt	`d db card(s)`
Fold	`pN f`
Check/Call	`pN cc`
Bet/Raise	`pN cbr amount`
Showdown	`pN sm card(s)`

Table 1: PHH-style representations of player actions

E Dataset Generation Details

Our dataset generation process creates valid six player No Limit Texas Hold'em poker games. The games are generated as a result of six unique and independent game agents playing against each other. Agents use myopic heuristics, driven by simulated win equity estimates. Each agent is randomly initialized for the following impactful values on decision making, using a provided seed for reproducibility.

- Propensity to raise
- Tightness in adhering to equity
- Bluff frequency
- Call willingness
- Initial bet scale
- Raise scale
- Bet continuation
-

This combination of heuristics with equity allows vast amounts of game data to be generated relatively quickly with a competent level of agent play. A considerable limitation of this method is that agents do not adapt over time or learn from others playing styles in a way that humans or more complex game playing agents could.

F Supplemental Figures and Tables

Below we give a diagram of our overall training pipeline for our Poker-GPT model, including our data-masking procedure.

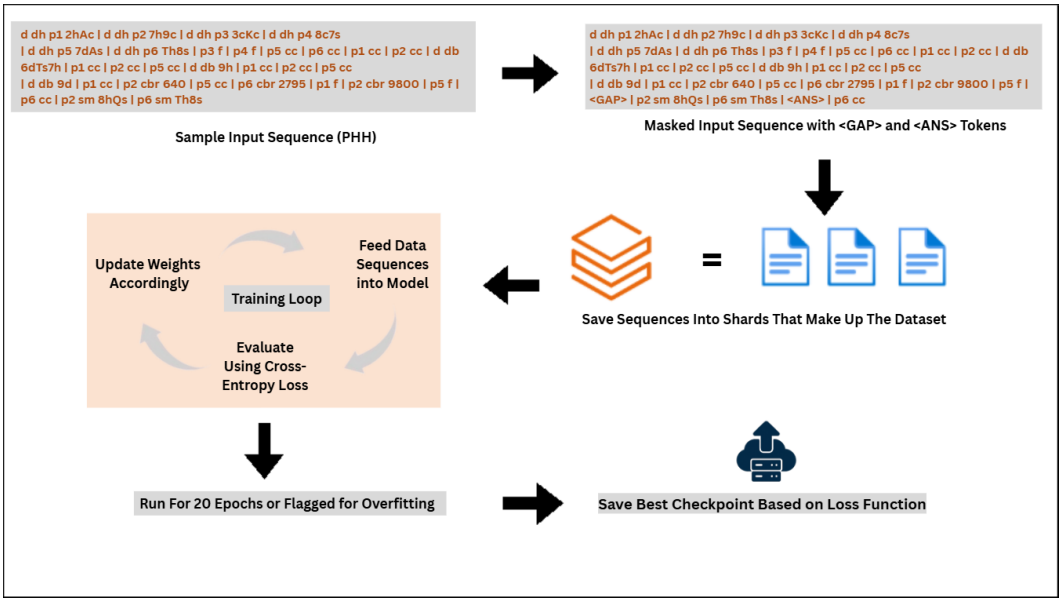


Figure 4: Training Pipeline. We train for up to 20 epochs, with early stopping if validation accuracy declines for three consecutive epochs to prevent overfitting.

G Compute and Memory Resources

We trained our GPT-2-based model, which comprises of 87 million parameters, on an NVIDIA H200 GPU for approximately seven hours. For the training of probes, we leveraged a diverse set of hardware: RTX 5090, NVIDIA H200, A10, and A100 GPUs.

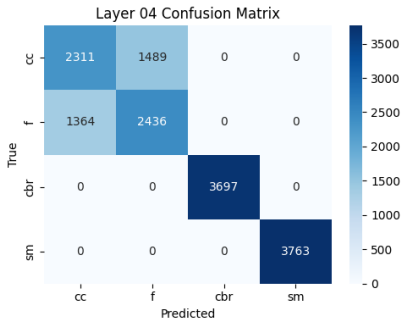
H Percentile Computation

$$
\begin{aligned}
&\text{Let } y = \text{array of hand rank labels,} \\
&\quad \mathcal{L} = \{\ell_1, \ell_2, \ldots, \ell_k\} = \text{unique labels in } y, \\
&\quad c_i = \sum_j \mathbf{1}_{\{y_j = \ell_i\}} \quad \text{for } i = 1, \ldots, k \quad \text{(counts per label)}, \\
&\text{target\_count} = \max\Big(\text{percentile}_{40}(\{c_1, \ldots, c_k\}),\ 10\Big)
\end{aligned}
$$

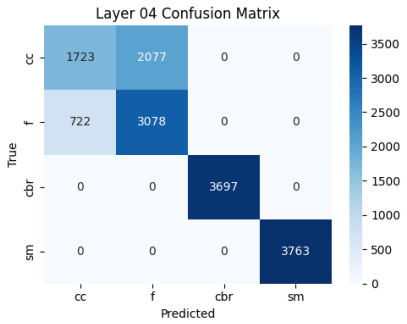
This balance in count helped us mitigate the impact of the excessive abundance of instances of hand ranks such as high_card due to their high-frequency nature by chance. This calculation also helps us validate the probe is not just learning to output one hand rank and, instead, is forced to extract intricate information from the activations of the model.

I Action Identification

To investigate how the model encodes the game state, we analyzed its ability to predict the action taken when the token corresponding to the action (`f`, `cc`, etc.) was masked out. This helped us prevent the model from 'cheating' and seeing the token within the playthrough. With this approach, the model was forced to determine the action taken based on the context of the playthrough. After running both a linear classifier probe and two-layer MLP (only one hidden layer), we noticed that the linear probe (see Figure 5a) and MLP probe (see Figure 5b) both achieved 80% accuracy for action identification. This implies that the model is learning to associate actions to certain contexts such as card reveals (as in the case of `sm`) and possibly learning how they fit in these contexts.



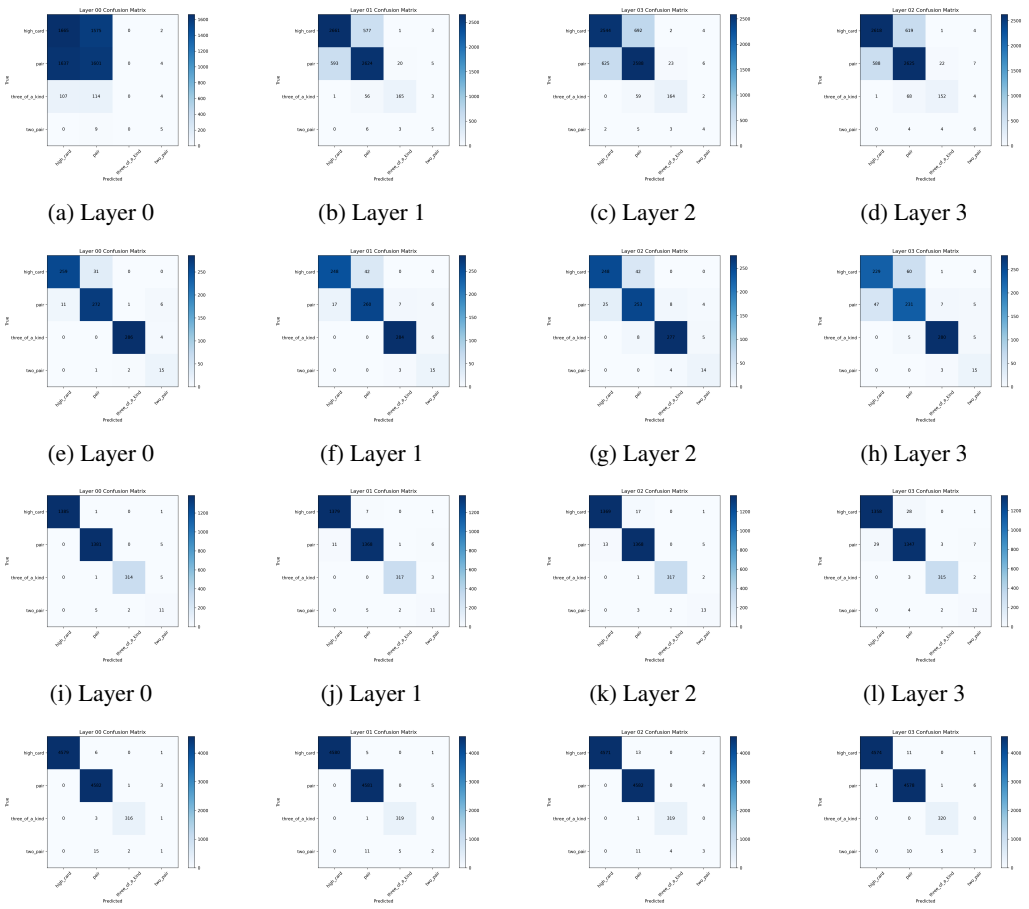
(a) Confusion Matrix for MLP Probe Action Identification on Layer 4.



(b) Confusion Matrix for MLP Probe Action Identification on Layer 4.

Figure 5: Neither model seems to perform considerably better than the other, possibly due to the fact the local context of `cc` and `f` is very similar.

J Hand-Rank Probe Experiments



(a) Layer 0 (b) Layer 1 (c) Layer 2 (d) Layer 3

(e) Layer 0 (f) Layer 1 (g) Layer 2 (h) Layer 3

(i) Layer 0 (j) Layer 1 (k) Layer 2 (l) Layer 3

(m) Layer 0 (n) Layer 1 (o) Layer 2 (p) Layer 3

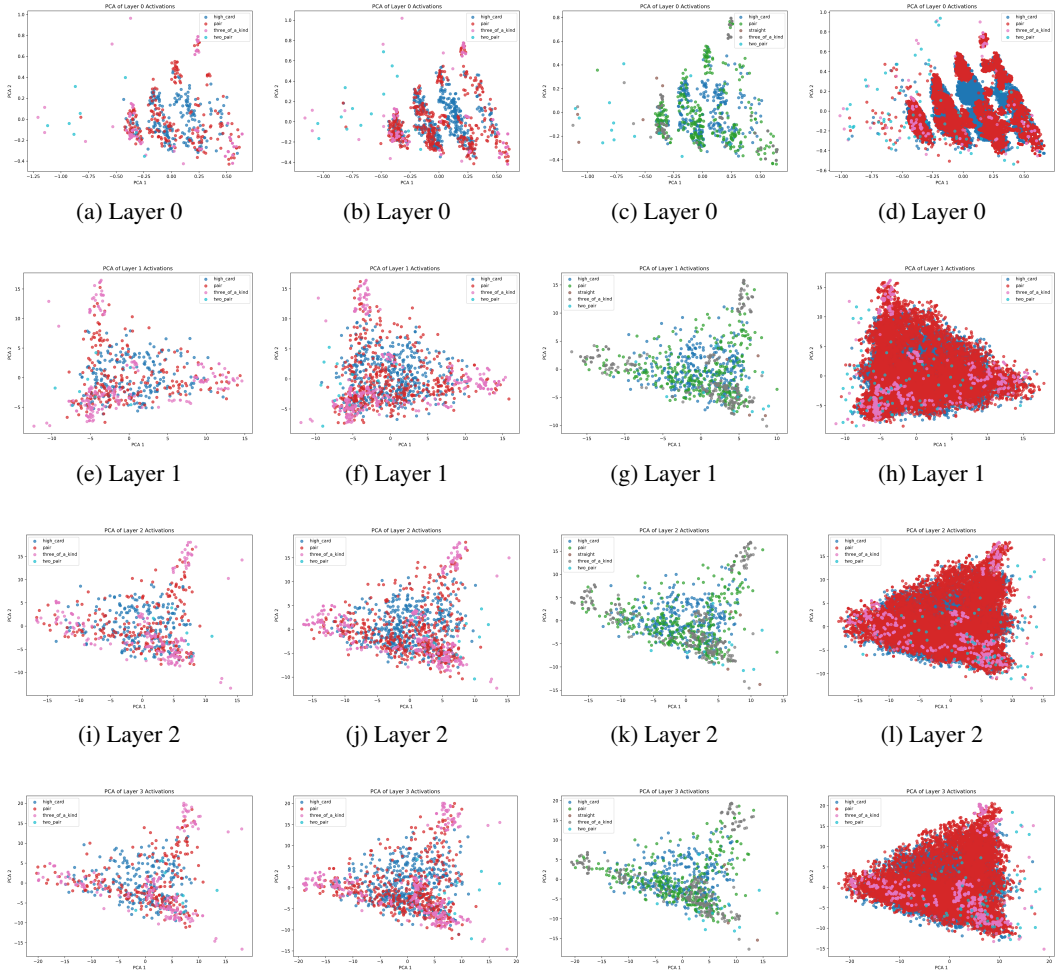
Figure 6: Confusion matrices and additional metrics for probe experiments across Layers 0–3. Row 1: Linear Probe, Row 2: MLP - 30th Percentile Equation, Row 3: MLP - 35th Percentile Equation, Row 4: MLP - 40th Percentile Equation.

K Activation Plots

Below are our activation plots (compressed with PCA, t-SNE, and UMAP, respectively, to 2-D plots) across multiple model layers. We note the per-hand clusters (with each cluster representing a particular "type" of hand, ie. a pair with certain card values, or a three of a kind with a certain card type). We also note that conceptual similarity is also being represented by these clusters. Note: These plots were generated from a test set of 200,000 samples. This set is separate from the training set used for the model and the one used for the probes.

PCA activation analysis revealed triangular activation structures that closely resemble the belief-state geometries described by Shai et al. [2024]. A natural direction for future work is to investigate whether these structures reflect the model's implicit representation of belief states in a POMDP setting. In particular, the vertices of the triangle may correspond to pure beliefs—confident assignments to specific hand ranks—while interior points capture mixtures over multiple possibilities. This interpretation would suggest that the model has learned to encode uncertainty in a manner consistent with POMDP belief representations.

Figure 7: PCA visualizations by layer (rows = 0–3) and training (and test) size (columns = 100k, 200k, 200k (modification with test split), 430k).



(a) Layer 0 (b) Layer 0 (c) Layer 0 (d) Layer 0

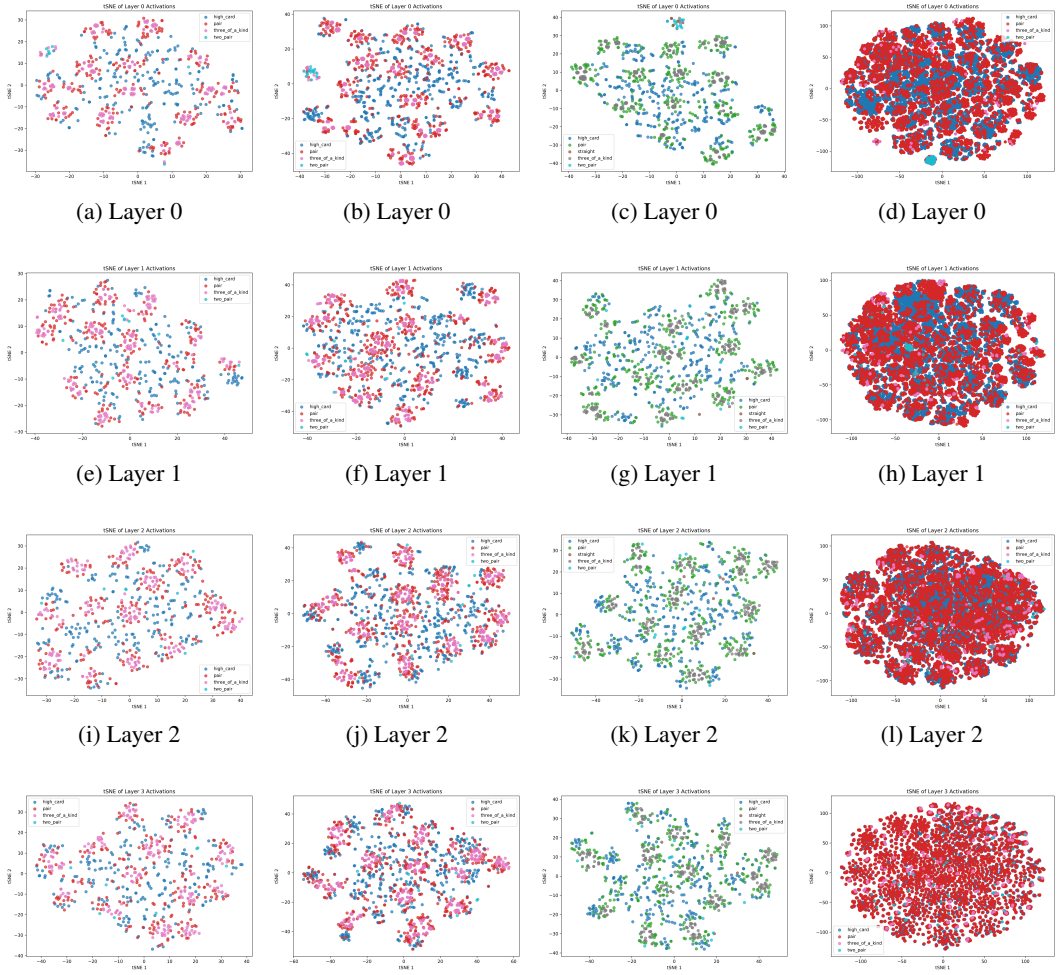
(e) Layer 1 (f) Layer 1 (g) Layer 1 (h) Layer 1

(i) Layer 2 (j) Layer 2 (k) Layer 2 (l) Layer 2

(m) Layer 3 (n) Layer 3 (o) Layer 3 (p) Layer 3

t-SNE offered feature-rich plots for our use cases with visible hand-rank clusters and conceptual similarity clusters being prominent, especially among pair and three_of_a_kind.

Figure 8: t-SNE visualizations by layer (rows = 0–3) and training (and test) size (columns = 100k, 200k, 200k (modification with test split), 430k).



(a) Layer 0 (b) Layer 0 (c) Layer 0 (d) Layer 0

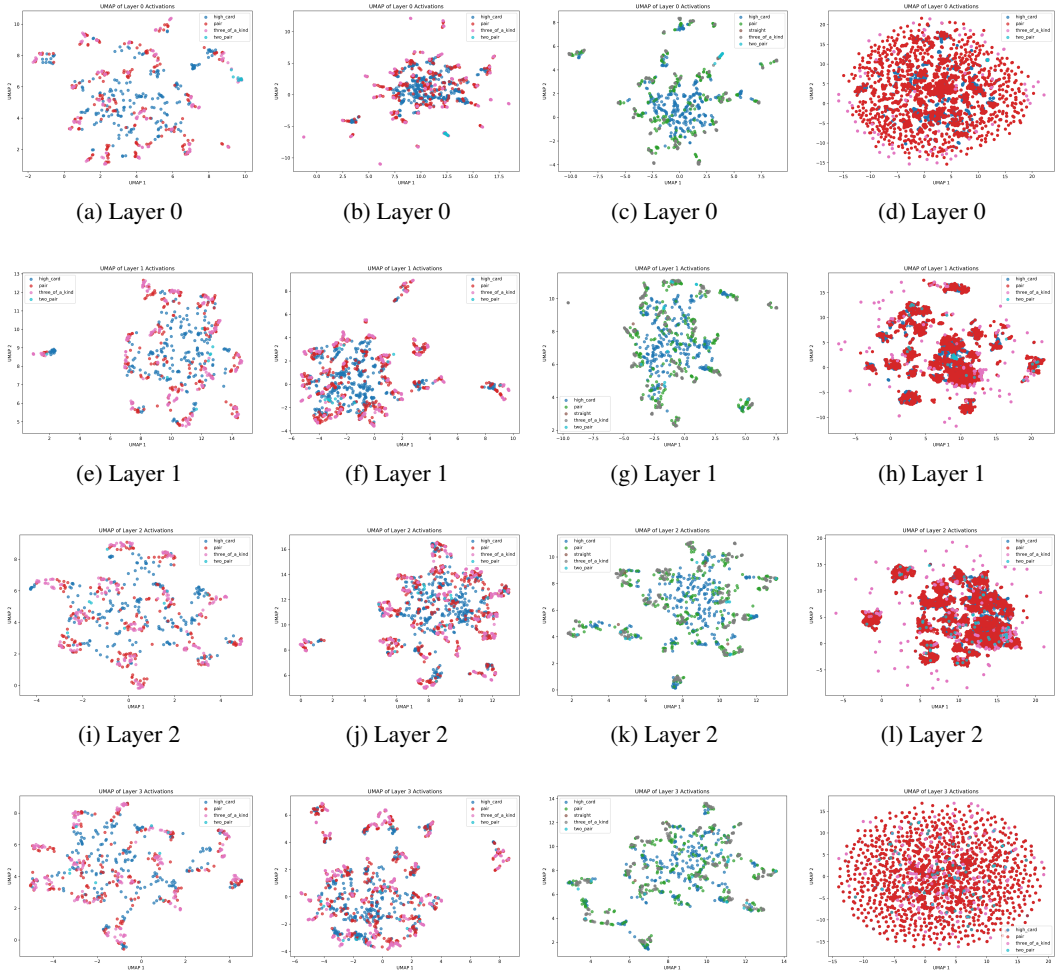
(e) Layer 1 (f) Layer 1 (g) Layer 1 (h) Layer 1

(i) Layer 2 (j) Layer 2 (k) Layer 2 (l) Layer 2

(m) Layer 3 (n) Layer 3 (o) Layer 3 (p) Layer 3

UMAP offered interesting plots that seem to show some clustering but are much less interpretable than t-SNE activation plots.

Figure 9: UMAP visualizations by layer (rows = 0–3) and training (and test) size (columns = 100k, 200k, 200k (modification with test split), 430k).



(a) Layer 0 (b) Layer 0 (c) Layer 0 (d) Layer 0

(e) Layer 1 (f) Layer 1 (g) Layer 1 (h) Layer 1

(i) Layer 2 (j) Layer 2 (k) Layer 2 (l) Layer 2

(m) Layer 3 (n) Layer 3 (o) Layer 3 (p) Layer 3

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims of our language model learning deterministic game structure as well as stochastic features of equity are supported by our results in sections addressing probing deterministic and stochastic world models.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.

- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations section discusses some of the areas where the scope of our paper is not as comprehensive as desired.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Our paper's theoretical results in our appendix are formally derived and fully justified.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Our experiments are completely reproducible, as the process we use for generating our dataset, training our model and conducting our interpretability experiments are fully disclosed in the paper.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Our code for reproducing probing experiments, as well as generating our dataset and training our model is linked in the appendix.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).

- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Data splits, learning rates, and epochs are fully enclosed and clear.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We only report single-run point estimates in the Deterministic and Stochastic World Model sections without confidence intervals, error bars, or tests across seeds/splits.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We list hyperparameters, dataset sizes, GPU details, and data generation times, as well as all the other computer resources needed for data generation and model training.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: The paper contains nothing that violates the code of ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]

Justification: There is no societal impact of our work.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.

- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: There are no risks posed by our paper.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: **[TODO]**

    Justification: The paper does not use existing assets.???

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We introduce new assets—the PHH data generator, probing code, and PokerGPT checkpoints—and provide documentation alongside them: repository READMEs with setup and reproduction commands, environment specs, and data schema; plus in-paper details on PHH formatting and dataset generation (Appendix §D, §G). Links are in the Appendix.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: No crowdfunding or human subjects were involved in the paper.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: No human subjects were involved in the paper.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: LLMs were used to assist in the code implementation of our experiments.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.