Emergent World Beliefs: Exploring Transformers in Stochastic Games

Adam Kamel[*]
University of Waterloo
atkamel@uwaterloo.ca

Tanish Rastogi[*]
Algoverse AI Research
tanishrastogi2020@gmail.com

Michael Ma[*]
Algoverse AI Research
michaelma0311@gmail.com

Kailash Ranganathan[†]
University of California, Berkeley
kranganathan@berkeley.edu

Kevin Zhu[†]
Algoverse AI Research
kevin@algoverse.us

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.

[*]* Equal contribution.
[†]† Senior author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Mechanistic Interpretability.

Our paper's main contributions include:

- We extend LLM internal world representation to games of incomplete information.
- We quantify the 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.

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

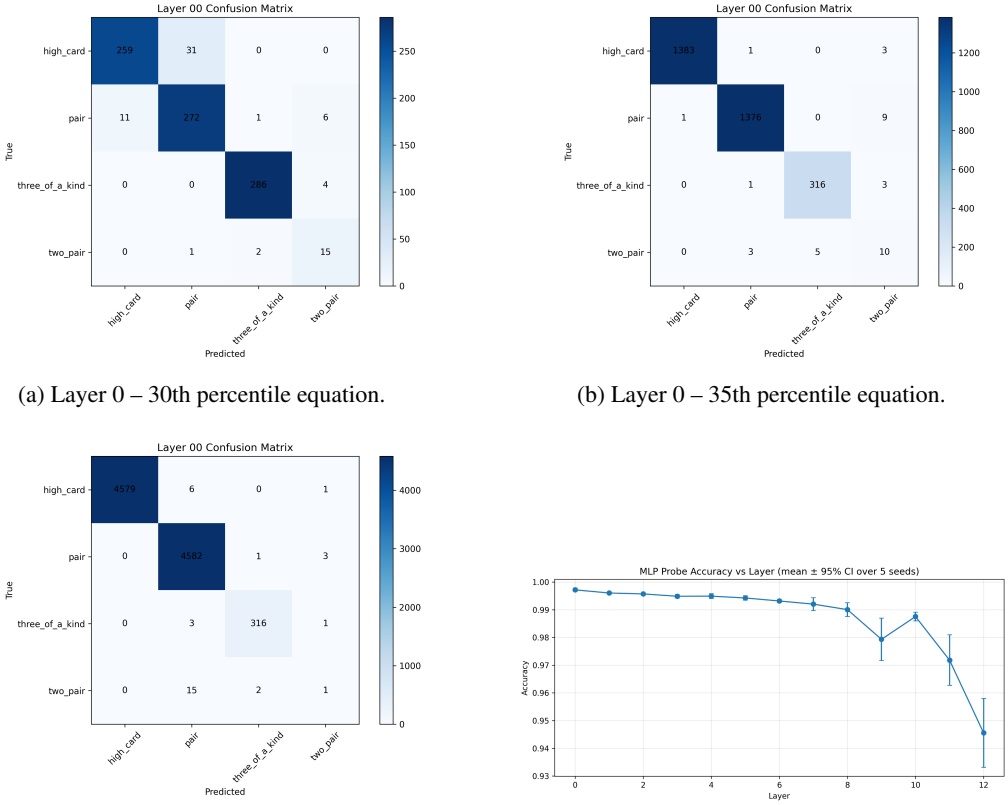
We fine-tuned a causal transformer language model based on GPT-2 [Radford et al., 2019] using a PHH-formatted dataset comprising over two million synthetically generated poker trajectories, as described above. The model retained the GPT-2 base configuration, with 12 attention heads and a hidden dimensionality of 768. The GPT-2 base configuration was chosen due to it being lightweight to work with our limited compute resources while being well understood and robust. 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



(a) Layer 0 – 30th percentile equation.

(b) Layer 0 – 35th percentile equation.

(c) Layer 0 – 40th percentile equation.

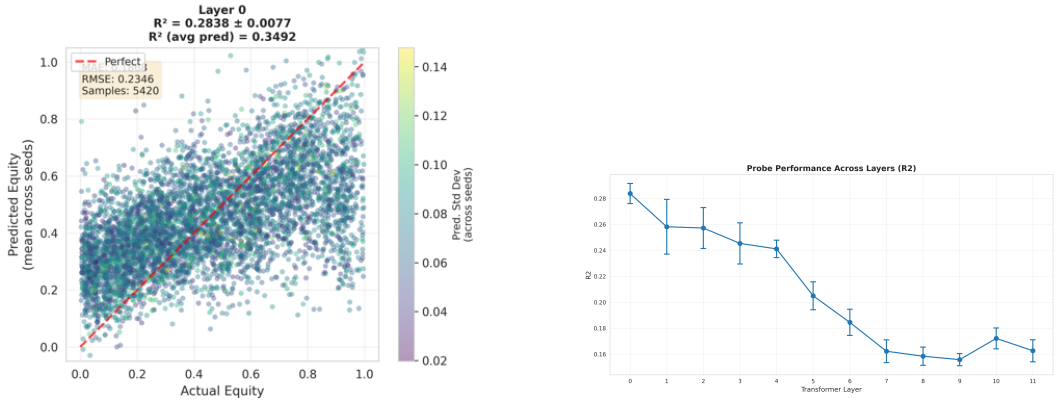
(d) MLP probe accuracy across model layers (40th percentile).

Figure 1: MLP probe performance for hand-rank identification. Panels (a-c) show confusion matricies for Layer 0 using datasets balanced by limiting each hand-rank class to the 30th, 35th, and 40th percentile of its unique sample count. Darker diagonal cells indicate more accurate predictions. Representation of rarer hand ranks, such as two pairs, is improved with lower percentiles. Panel (d) shows probe accuracy across all layers using the 40th percentile dataset, with 95% confidence intervals across five random seeds. Together, results indicate that hand-rank information is encoded strongly and consistently. See Appendix I for additional deterministic experimental results.

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. Furthermore, small error bars reflect low variance across various seeds in Figure 1d in the first several layers, demonstrating that a strong consistency of the model's internal representations that reflects the ~98% accuracy is present.

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 0.59 on our test dataset predictions, averaged across seeds (Figure 2a). 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. Observing the R^2 across layers, the understanding of equity is most strongly encoded in the early layers, becoming diluted after layers 0-4 (Figure 2b). This decrease in R^2 across layers is consistent with information bottleneck style compression [Tishby and Zaslavsky, 2015], with deeper layers retaining information that is more relevant for token prediction, leading to weaker representations of input variables such as hand equity as the representation becomes more focused on the prediction task. This result is similar to what is observed in our deterministic world model evaluations (Figure 1d), with hand recognition experiencing the same trend.



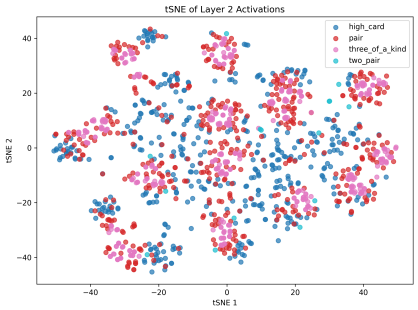
(a) Predicted vs. true equity (layer 0).

(b) Comparison of R^2 across layers.

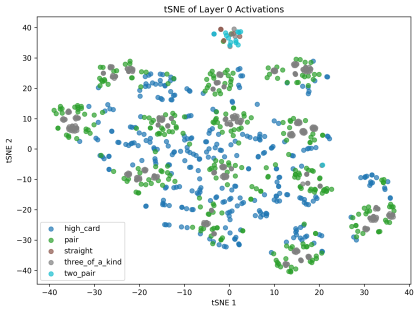
Figure 2: Probe performance on stochastic representations. Panel (a) shows predicted versus true hand equity for Layer 0, demonstrating that model activations contain information about winning probability despite incomplete information. Panel (b) shows the R^2 value of equity prediction across layers, showing that equity information is strongest in earlier layers of (0-5) and progressively diminishes deeper in the network, consistent with information-bottleneck–style compression.

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 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. Distinct clusters indicate that the model internally organizes hands according to rank or equity strength that follows. Multiple clusters for ranks such as "pair" suggest that the model learns more detailed sub-categories (e.g., types of pairs such as J and K) rather than treating each rank as a single class.

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 how the model understands the game and it's simple and complex variables. This is an unfortunate consequence of being forced to use synthetic data, due to a lack of fully documented poker hand datasets. Due to this, we are unable to validate our results with additional datasets. 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, or to novel domains.

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.

7 Acknowledgments

We are extremely grateful to the Algoverse research program for computational resources and extensive mentorship. We also thank the anonymous reviewers of our paper for their helpful feedback.

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.

Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. *2015 IEEE Information Theory Workshop (ITW)*, pages 1–5, 2015. URL https://api.semanticscholar.org/CorpusID:5541663.

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.

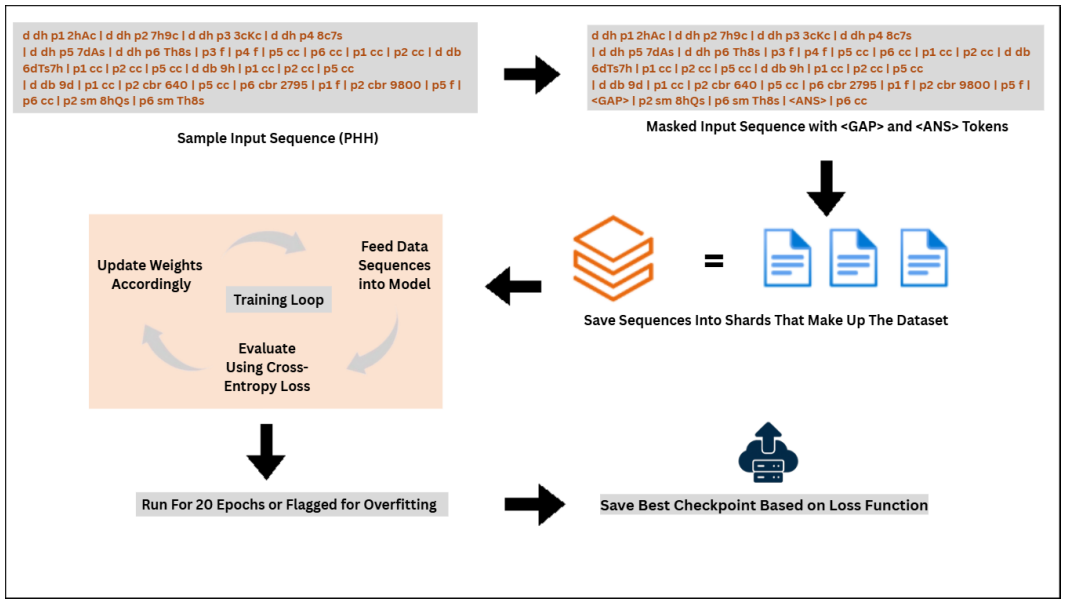


Figure 4: Training Pipeline. We train for up to 20 epochs, with early stopping if validation accuracy declines for three consecutive epochs to prevent overfitting.

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.

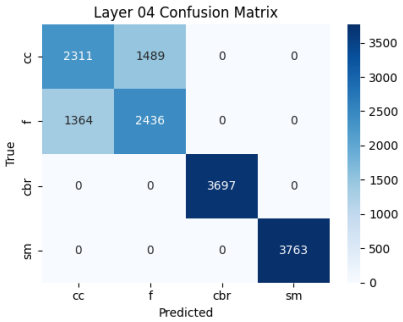
H Percentile Computation

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

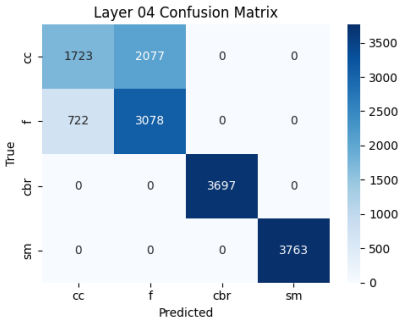
This balance in count helped us mitigate the impact of the excessive abundance of instances of hand ranks such as high_card due to their high-frequency nature by chance. This calculation also helps us validate the probe is not just learning to output one hand rank and, instead, is forced to extract intricate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



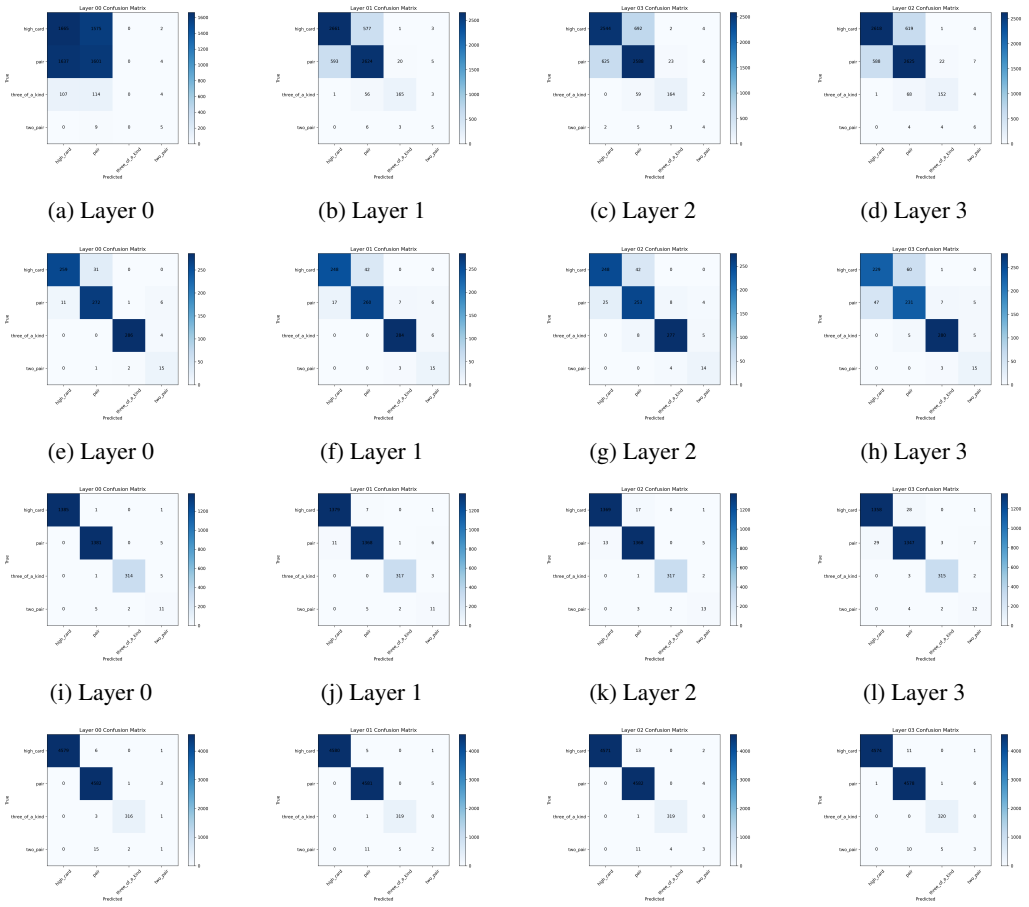
(a) Confusion Matrix for MLP Probe Action Identification on Layer 4.



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

Figure 5: Action identification performance using linear and MLP probes when the action token (e.g., `f`, `cc`, `sm`) is masked out during inference, to prevent models from "cheating". Panels (a) and (b) show confusion matrices for the linear probe and two-layer MLP probe respectively, evaluated on Layer 4 of the transformer. Both probes achieve similar accuracy (∼80%), suggesting that the model's internal activations already encode sufficient information about common actions and their situational context, but also show confusions between actions with similar local structure (e.g., `cc` vs. `f`).

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

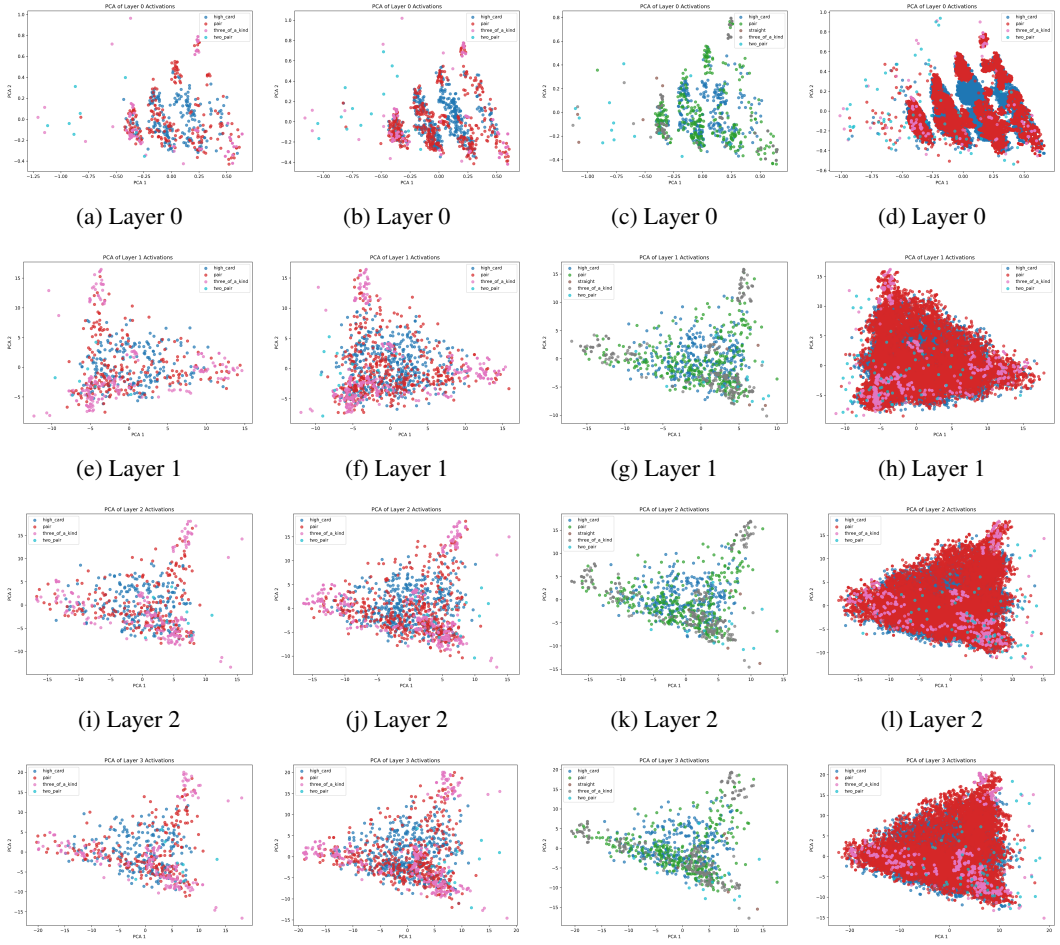
(m) Layer 0 (n) Layer 1 (o) Layer 2 (p) Layer 3

Figure 6: Hand-rank identification results across transformer Layers 0–3 using linear and MLP probes. Each confusion matrix shows probe predictions for the categorical poker hand-rank (e.g., high card, pair, two pair, etc.), evaluated on a held-out test set excluded from both model and probe training. Row 1 shows linear probe performance. Rows 2–4 show MLP probe performance when the training data is balanced at the 30th, 35th, and 40th percentiles of unique hand-rank frequencies, respectively, ensuring that rare hand-ranks are not underrepresented. Across all configurations, strong diagonals indicated by the dark coloring show consistent internal encoding of hand-rank information by early layers, while differences across percentiles displays the effect of dataset balancing for rarer hand-ranks.

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. As seen in figure



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

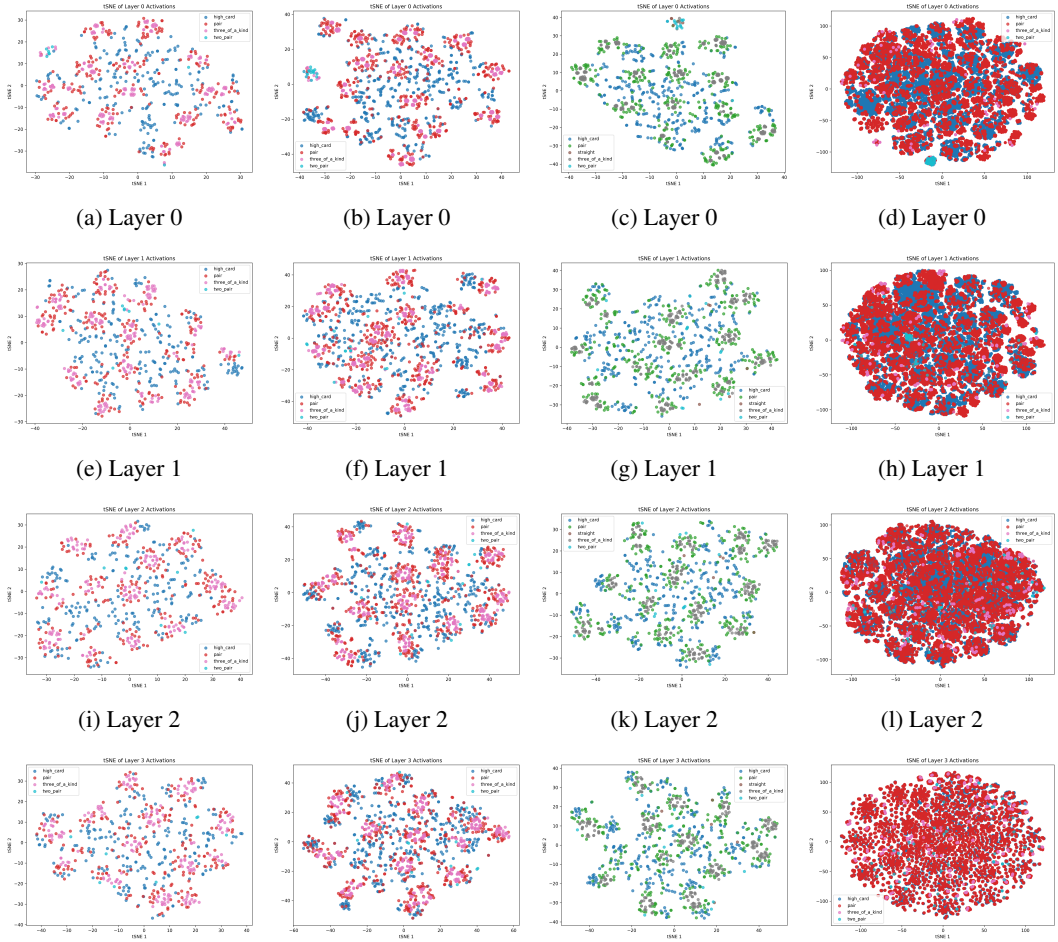
(e) Layer 1 (f) Layer 1 (g) Layer 1 (h) Layer 1

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

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

Figure 7: PCA projections of activation vectors across transformer Layers 0–3 and across four different training set sizes (columns: 100k, 200k, 200k with modified test split, and 430k samples). Each panel shows a 2D PCA embedding of per-token activations colored by hand-rank class. The recurring triangular geometry resembles POMDP belief-state manifolds and becomes increasingly well-separated at larger training sizes, indicating stronger representational structure.

The t-SNE visualizations below further illustrate this structure. Each subplot shows a two-dimensional embedding of activation vectors colored by hand-rank class. Compared to PCA, t-SNE produces sharper and more clearly separated clusters, revealing that the model internally organizes hands by both rank and conceptual similarity. In particular, pair and three-of-a-kind categories form distinct, compact regions, while more ambiguous hands occupy the intermediate space, reflecting graded internal beliefs about hand strength.



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

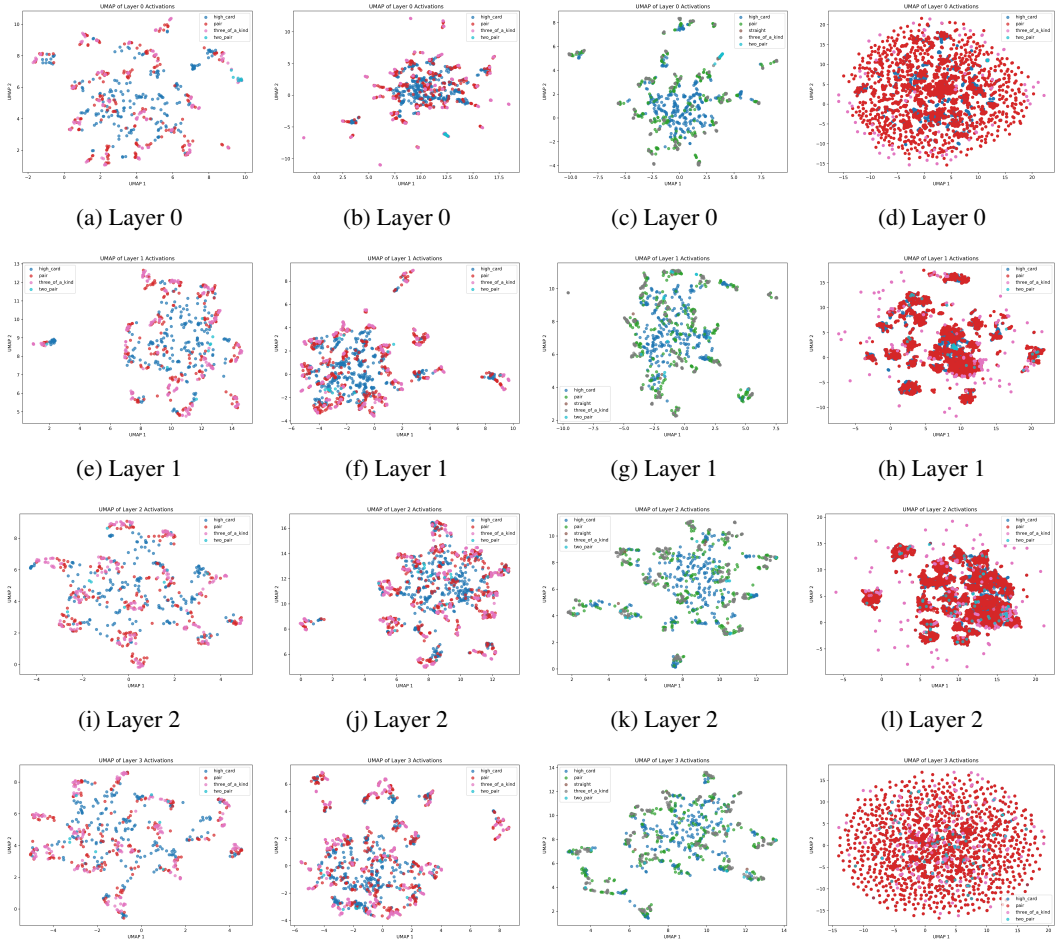
(e) Layer 1 (f) Layer 1 (g) Layer 1 (h) Layer 1

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

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

Figure 8: t-SNE visualizations of activation vectors across transformer Layers 0–3 and four different training set sizes. Each subplot embeds per-token activations into two dimensions, colored by hand-rank class. t-SNE reveals fine-grained cluster structure that is especially pronounced for conceptually similar hands (e.g., pairs and three-of-a-kind). Deeper layers also exhibit tighter, more separated clusters, indicating progressive specialization of internal representations.

Each subplot below shows a UMAP two-dimensional embedding of activation vectors colored by hand-rank class. UMAP reveals partial clustering behavior, though with less separation and interpretability than PCA or t-SNE.



(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

Figure 9: UMAP projections of activation vectors across transformer Layers 0–3 and across four training set sizes. Clusters correspond to semantically related hand-ranks, but the method introduces more distortion in the results.