Revisiting the Othello World Model Hypothesis

Anonymous ACL submission

Abstract

Li et al. (2023) used the Othello board game as a test case for the ability of GPT2 to induce world models, and were followed up by Nanda et al. (2023). We briefly discuss the original experiments, expanding them to include more language models with more detailed probing. Specifically, we analyze sequences of Othello board states and train the model to predict the next move based on previous moves. We evaluate six language models (GPT2, T5, Bart, Flan-T5, Mistral, and Llama-2) on the Othello task and conclude that these models not only learn to play Othello, but also induce the Othello board layout. We find that all models achieve up to 99% accuracy in *unsupervised* grounding and exhibit high similarity in the board features they learned. This provides much stronger evidence for the Othello World Model Hypothesis than previous works.

1 Introduction

Li et al. (2023) used the Othello board game to probe LLMs' ability to induce world models. Their network had a 60-word input vocabulary, corresponding to the 64 tiles of an Othello board, except for the four that are already filled at the start. They trained the network on two datasets: one with about 140,000 real Othello games and another with millions of synthetic games. They then trained 64 independent non-linear probes (two-layer MLP classifiers) to classify each of the 64 tiles into three states: black, blank, and white, using internal representations from Othello-GPT as input. The error rates of these non-linear probes dropped from 26.2% on a randomly-initialized model to only 1.7% on a trained model, while linear probes performed close to random. Li et al. (2023) saw this as evidence that LLMs can induce (non-linear) world models, at least for Othello board games, supporting the Othello World Model Hypothesis.

Nanda et al. (2023) did a follow-up study in which they found that linear probes also work if trained slightly differently. Rather than focus on tile color, they probe the board state relative to the current player at each timestep, using labels such as MINE, YOURS, and EMPTY. This reduces the error rate of the probes to less than 10%. They speculate that world knowledge is often linearly represented in language models, since 'matrix multiplication can easily extract a different subset of linear features for each neuron.'

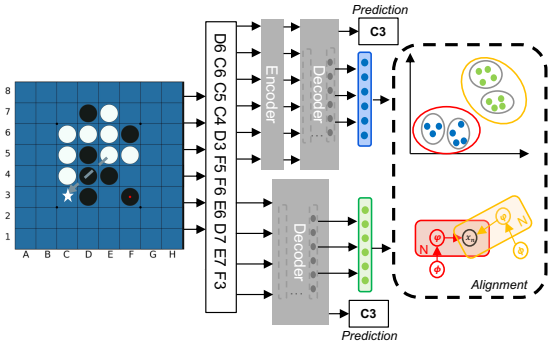


Figure 1: Experimental protocol. We train a transformer-based model to predict the next move in Othello and see whether the board game layout is induced (up to isomorphism).

Now, probing as a research methodology comes with several weaknesses, including: a) Probing classifiers can be prone to spurious correlations (Barrett et al., 2019). b) They do not tell us how information is arranged globally in LLMs.[1] c) They therefore only detect a subset of the interesting properties of world models, e.g., excluding the spatial relations that would enable analogical reasoning (Mikolov et al., 2013).

Contributions We therefore re-evaluate the Othello World Model Hypothesis by other means (see Figure 1), in order to reassess the ability of LLMs

[1] Li et al. (2023) tried to compensate for this by using PCA to plot the probing classifiers in three dimensions. The PCA plots suggest that the induced global structure is meaningful, but the probing paradigm cannot quantify its meaningfulness.

to induce world models. If our results are positive, this significantly stresses the case for the argument that LLMs induce world models; if not, this suggests that the evidence cited in Li et al. (2023) and Nanda et al. (2023) was perhaps a (spurious) effect of the probing paradigm itself. Specifically, we rely on representation alignment tools from the literature on cross-lingual word embeddings (Søgaard et al., 2019) and evaluate six models (GPT2, Bart, T5, Flan-T5, Llama-2, Mistral) across the two datasets presented by Li et al. (2023). Our analysis goes beyond other analyses by considering both pretrained and non-pretrained models, two-hop generation abilities, and learning curves. Our results show that the language models – exhibit solid one-hop performance when trained on large amount of game sequence moves. We find that in some cases, all models can achieve up to 99% accuracy in unsupervised grounding, which means that absent any cross-modal supervision, a model *trained to play* Othello can identify the right positions on a board. More importantly, the alignment similarity score of the board features learned by these models is surprisingly high. This provides a direct counter-example to previous claims that mono-modal models cannot solve visual question answering problems (Bender and Koller, 2020) – or, more generally, symbol grounding problems (Harnad, 1990). These results are significantly stronger than those in Li et al. (2023); Nanda et al. (2023) and, in our view, provide more direct evidence of the Othello World Model Hypothesis.

2 Method

Modeling Following previous works (Liskowski et al., 2018; Li et al., 2023; Nanda et al., 2023), we formulate the problem of playing the board game as a sequence generation problem. Specifically, we fine-tune generative pretrained models in an autoregressive manner to predict the next move given the current board state. Each game is a sequence, with each move represented as a token, and in each round, we predict the next move. Our vocabulary consists of 60 words, each corresponding to one of the 60 tiles, where players place discs, excluding the 4 center tiles, which are already filled when the game begins. See Figure 1 for an example move. Our modeling of Othello, in brief, can be represented as:

$$p_\theta(x_i|X_{<i}) = softmax(f_i(x_1, x_2, ..., x_{i-1})) \quad (1)$$

where $x_1, x_2, ..., x_{i-1}$ represent history moves.

Probing To evaluate the Othello World Model Hypothesis, we depart from previous work and device a new test, more directly evaluating the internal representation of the Othello board in language models. Specifically, during inference, we input the previously generated game moves $X_{<i}$ at step i into the model and prompt it to generate the next step. We then extract the representation from the last hidden layer of Decoder from all steps, denoted as $h_\theta(x_i) \in \mathcal{D}^{s \times l}$ as a pivot comparison target, where s is the number of steps of a game, and l is the size of hidden layer features. We consider the outputs of different models as different source and target spaces. Using the representations from different models with the same input sequence as parallel data, we perform mapping training under both supervised and unsupervised scenarios (details see Section 3.3). For example, the ith step given the input sequence of two models can be seen as a pair, denoted as $h_{\theta 1}(x_i)$ and $h_{\theta 2}(x_i)$, respectively. For supervised training, we use the pairwise data to learn a mapping from the source to the target space using iterative Procrustes alignment (Gower and Dijksterhuis, 2004). For unsupervised training, without any parallel data or anchor points, we learn the mapping through a combination of adversarial training and iterative Procrustes refinement (Lample et al., 2018).

3 Experiments

3.1 Experimental Setup

We use two datasets in our experiments, **CHAMPIONSHIP** and **SYNTHETIC**. Both of them were collected by Li et al. (2023). **CHAMPIONSHIP** comes from real online Othello gaming sources, whereas **SYNTHETIC** is artificially generated according to the rules of Othello game play. Detailed statistics see Appendix B. We use the last 20,000 games from each dataset for testing and validation (10,000 games each).

Following Li et al. (2023), we report the top-1 error rate, including both 1-hop and 2-hop generation. This involves verifying whether the top-1 prediction is legal when the model is prompted to generate 1 and 2 moves at a time. We present the average error rate across all game sequences.

We perform our experiments using several existing baselines, with both Encoder-Decoder or Decoder-only structures. We first adopt some popular PLMs such as GPT2 (Radford et al., 2019), T5 (Raffel et al., 2019), Bart (Lewis et al., 2019).

Method	Type	P	CHAMPIONSHIP			SYNTHETIC				
			2k	20k	full	2k	20k	200k	2M	full
GPT2	D	✗	49.8 | 78.5	17.7 | 34.7	5.6 | 28.1	49.2 | 76.3	26.8 | 70.8	**13.6** | **43.6**	10.4 | 29.0	<0.1 | 5.2
Bart	E-D	✗	25.2 | 54.2	16.6 | 31.1	4.7 | 23.4	73.6 | 86.5	31.7 | 67.2	14.2 | 44.8	16.3 |35.7	<0.1 | 4.2
T5	E-D	✗	**20.9** | **48.8**	15.2 | 28.7	4.3 | 24.4	65.8 | 88.2	28.7 | 67.7	15.7 | 46.9	10.1 | 35.9	<0.1 | 3.4
Flan-T5	E-D	✗	23.4 | 51.8	**4.8** | **20.8**	**3.6** | **21.9**	**35.6** | 79.6	**23.7** | **63.1**	21.2 | 48.6	**7.7** | 26.7	<0.1 | **2.8**
Llama-2	D	✗	27.8 | 60.9	16.5 | 36.3	5.7 | 26.4	57.1 | 87.3	35.4 | 67.8	16.9 | 45.2	10.2 | 36.3	<0.1 | 5.5
Mistral	D	✗	22.1 | 51.4	14.8 | 31.7	4.2 | 22.3	48.2 | **71.2**	34.4 | 77.1	17.7 | 47.9	8.3 | **26.4**	<0.1 | 3.0
GPT2	D	✓	52.6 | 92.2	19.7 | 43.4	13.6 | 37.2	74.4 | 99.6	32.4 | **72.6**	19.9 | 45.5	14.1 | 34.4	<0.1 | 6.2
Bart	E-D	✓	54.0 | 87.0	14.6 | 34.5	13.7 | 27.1	77.2 | 97.8	35.8 | 76.9	24.4 | 64.0	16.6 | 44.5	<0.1 | 5.1
T5	E-D	✓	45.5 | 86.5	19.6 | 36.4	3.8 | 27.0	69.4 | 99.6	36.9 | 78.8	32.6 | 59.9	13.9 | 46.9	<0.1 | 4.6
Flan-T5	E-D	✓	31.7 | 67.9	**4.8** | **31.8**	3.7 | 26.5	70.3 | 98.6	**25.4** | 80.8	45.0 | 79.7	8.7 | 35.3	<0.1 | 3.9
Llama-2	D	✓	43.1 | 66.9	14.7 | 33.4	7.0 | 33.0	74.6 | 94.2	41.5 | 77.6	33.4 | 62.1	**7.6** | **33.2**	<0.1 | 5.2
Mistral	D	✓	**16.8** | **52.0**	15.0 | 40.8	**3.3** | **25.4**	**33.8** | **80.3**	30.6 | 76.0	**18.2** | **42.3**	7.7 | 35.0	<0.1 | **3.8**

Table 1: The error rate of 1-hop and 2-hop game state generation in terms of different size of training data. 'Type' refers to the model type, 'P' denotes if the model is pretrained or not. All the numbers are shown in precentage.

Src.	Trg.	Supervised		Unsupervised	
		CHAM.	SYN.	CHAM.	SYN.
GPT2	Bart	81.4	**93.1**	80.3	91.3
GPT2	T5	83.0	85.0	76.4	80.1
Bart	T5	69.2	84.5	85.2	81.1
GPT2	Mistral	**90.3**	77.2	80.3	82.6
Bart	Mistral	88.0	79.1	**96.1**	**97.2**
Llama-2	Mistral	80.1	74.2	76.2	72.6

Table 2: Representation alignment cosine similarity (%) results. Src. and Trg. represent source and target space.

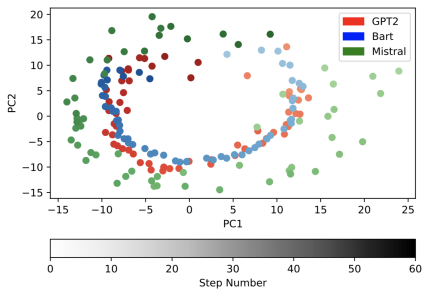


Figure 2: PCA visualization of the 60 steps from various models within one game.

We adopt several LLMs to see the their performance on this task, including Flan-T5 (Chung et al., 2022), Llama-2 (Touvron et al., 2023), and Mistral (Jiang et al., 2023). Details see Appendix C.

3.2 Experimental Results

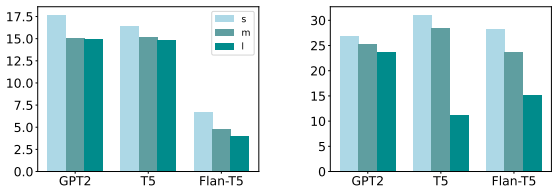
We perform experiments on different methods and report the results in Table 1. We observe that: firstly, there is no clear superiority between models with an Encoder-Decoder structure and those with a Decoder-only structure for this task. However, it is evident that increasing the amount of training data positively impacts overall performance. Compared with language models with a smaller size, LLMs such as Mistral, Flan-T5 show superiority in the task. This suggests that model size and capacity play a crucial role in understanding the Othello game step generation. We also find that pretrained language knowledge sometimes negatively affects the ability to understand the game steps, as the pretrained versions of most models generally perform worse than their non-pretrained counterparts. Additionally, even though using a large amount of data to fine-tune the model results in a reasonable 1-hop performance, it's still challenging for the model to generate more than 1 step at a time.

3.3 Representation Alignment

We probe different models by aligning their representations into one joint vector space. We report the MUSE[2] cosine similarity of the aligned features score under both supervised (Conneau et al., 2018) and unsupervised (Lample et al., 2018) settings in Table 2[3](more details see Appendix D). In order to vividly show such alignment, we also demonstrate the PCA coordinate of the 60 step features $h_\theta(x)$ within one random game in Figure 2. From the results we observe high similarity scores across different language models. For instance, despite having different model structures (Decoder-only v.s. Encoder-Decoder), the SYNTHETIC supervised similarity score between GPT2 and Bart reaches 93.1%. We also observe highly similar step representations across different models in Figure 2. This indicates that the models share common knowledge when modeling the Othello task.

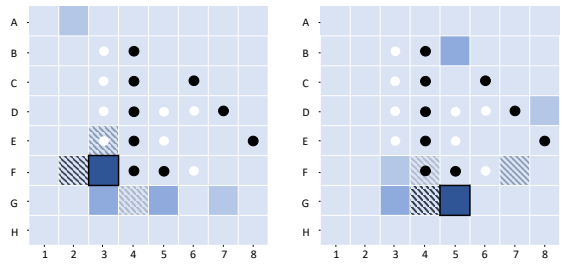
[2]https://github.com/facebookresearch/MUSE

[3]We use the non-pretrained version based on 20k training data for all models.



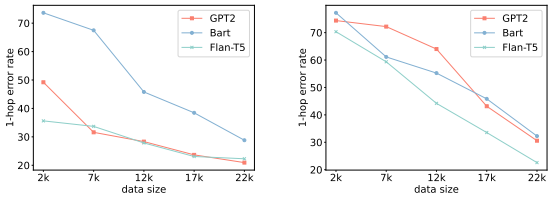
(a) **CHAMPIONSHIP** (b) **SYNTHETIC**

Figure 3: Othello 1-hop generation performance under different model sizes. All models are non-pretrained version fine-tuned with 20k game sequences.



(a) **T5** (b) **Mistral**

Figure 4: Othello visualization result from two best performed models. Colors indicate the likelihood of the position of the next step. Shadows highlight the top three tiles with embeddings closest to the top candidate, with the darkest color in the black box.



(a) **Non-pretrained** (b) **Pretrained**

Figure 5: Analysis of 1-hop error rates on the SYNTHETIC dataset with varying data scales.

4 Extensive Analysis

4.1 Model Size Analysis

To further investigate the impact of model size, we report the performance of each model in different size versions, as shown in Figure 5. For each model, we report the performance under the small, medium, and large sizes. The figure shows that the error rate decreases as the model size increases in both datasets, with this improvement being more pronounced in the SYNTHETIC dataset. This suggests that larger models are more effective at capturing and generalizing from synthetic data. These findings underscore the importance of model scaling in achieving better performance in this task.

4.2 Relevant Position Analysis

We visualize the Othello game steps of two models in Figure 4. It shows that both models successfully predict legal moves given a game sequence. Moreover, other legal moves are also assigned high prediction scores (tiles with lighter blue) by the models. This proves that with a large amount of game sequence data, the model learns the policy of the game. To further investigate whether the models can capture the physical position of each tile, we use shadow marks to highlight the tiles with the closest embedding distance to the tile in the black box. The intensity of the shadow reflects the degree of similarity. We observe that the top-1 tile with the highest similarity (F2 in T5, G4 in Mistral) is the one adjacent to the black box tile in both models. This indicates that the models not only understand the game mechanics but also capture the spatial relationships between tiles.

4.3 Data Scale Analysis

In Table 1, we observe a sharp decrease in model error rates as the dataset size increases from 2k to 20k. To investigate this further, we conduct an analysis by gradually enlarging the SYNTHETIC dataset from 2k to 22k. According to Figure 5, the performance of all models improves gradually as the dataset size increases. Pretrained models exhibit a more consistent decrease in error rate compared to non-pretrained ones. For non-pretrained models, the error rate reduction is more pronounced within the 2k to 12k data size interval. This indicates that while pretrained models benefit steadily from larger datasets, non-pretrained models experience significant initial gains.

5 Conclusion

We conduct a detailed probing of language models' ability to predict legal moves in the Othello board game, based on the settings in (Li et al., 2023). We evaluate six language models, training them to predict the next move based on previous moves. All six models achieve almost 'perfect' one-hop move prediction performance when trained with large amount of data. We then adopt representation alignment tools to align the learned game state features from different models into one joint space. We observe high similarity in the board features they learned. These results, in our view, provide the most solid evidence to date of the Othello World Model Hypothesis presented in previous works.

Limitation

Although this work demonstrates the ability of different language models to understand Othello game rules, certain limitations persist. Firstly, while language models perform reasonably well in 1-hop game state generation, generating sequences of more than one step remains challenging. In our initial experiments, we attempted to train the models to generate entire game sequences, but they achieved nearly zero accuracy, even with a substantial amount of training data. Another limitation is that our experiments show achieving 'perfect' generation ability (i.e., the 1-hop error rate less than 1%) requires a large amount of data for model training. Given the size of LLMs, this also presents significant computational and resource challenges. Therefore, while we provide strong evidence supporting the Othello World Model Hypothesis, further experiments are necessary to demonstrate that language models can serve as a true world model.

References

Maria Barrett, Yova Kementchedjhieva, Yanai Elazar, Desmond Elliott, and Anders Søgaard. 2019. Adversarial removal of demographic attributes revisited. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6330–6335, Hong Kong, China. Association for Computational Linguistics.

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

Naiyuan Chang, Chih-Hung Chen, Shun-Shii Lin, and Surag Nair. 2018. The big win strategy on multi-value network: An improvement over alphazero approach for 6x6 othello. *Proceedings of the 2018 International Conference on Machine Learning and Machine Intelligence*.

Hyung Won Chung, Le Hou, S. Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Wei Yu, Vincent Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed Huai hsin Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *ArXiv*, abs/2210.11416.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. *The Sixth International Conference on Learning Representations*.

John C Gower and Garmt B Dijksterhuis. 2004. *Procrustes problems*, volume 30. OUP Oxford.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. 2023. Reasoning with language model is planning with world model. *ArXiv*, abs/2305.14992.

Stevan Harnad. 1990. The symbol grounding problem. *Physica D*, 42:335–346.

Peter Hase, Mohit Bansal, Been Kim, and Asma Ghandeharioun. 2023. Does localization inform editing? surprising differences in causality-based localization vs. knowledge editing in language models. *ArXiv*, abs/2301.04213.

Dean S. Hazineh, Zechen Zhang, and Jeffery Chiu. 2023. Linear latent world models in simple transformers: A case study on othello-gpt. *ArXiv*, abs/2310.07582.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *North American Chapter of the Association for Computational Linguistics*.

Tianze Hua, Tian Yun, and Ellie Pavlick. 2024. mothello: When do cross-lingual representation alignment and cross-lingual transfer emerge in multilingual models? *ArXiv*, abs/2404.12444.

Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. 2024. The platonic representation hypothesis.

Michael I. Ivanitskiy, Alex F Spies, Tilman Rauker, Guillaume Corlouer, Chris Mathwin, Lucia Quirke, Can Rager, Rusheb Shah, Dan Valentine, Cecilia G. Diniz Behn, Katsumi Inoue, and Samy Wu Fung. 2023. Structured world representations in maze-solving transformers. *ArXiv*, abs/2312.02566.

Albert Qiaochu Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L'elio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *ArXiv*, abs/2310.06825.

Adam Karvonen. 2024. Emergent world models and latent variable estimation in chess-playing language models. *ArXiv*, abs/2403.15498.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Unsupervised machine translation using monolingual corpora only. *The Sixth International Conference on Learning Representations*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdel rahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Annual Meeting of the Association for Computational Linguistics*.

Belinda Z. Li, Maxwell Nye, and Jacob Andreas. 2021. Implicit representations of meaning in neural language models. In *Annual Meeting of the Association for Computational Linguistics*.

Kenneth Li, Aspen K Hopkins, David Bau, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023. Emergent world representations: Exploring a sequence model trained on a synthetic task. In *The Eleventh International Conference on Learning Representations*.

Paweł Liskowski, Wojciech Jaśkowski, and Krzysztof Krawiec. 2018. Learning to play othello with deep neural networks. *IEEE Transactions on Games*, 10(4):354–364.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Neel Nanda, Andrew Lee, and Martin Wattenberg. 2023. Emergent linear representations in world models of self-supervised sequence models. In *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 16–30, Singapore. Association for Computational Linguistics.

Samantha E. Miller Noever and David Noever. 2022. Word play for playing othello (reverses). *Preprint*, arXiv:2207.08766.

Roma Patel and Ellie Pavlick. 2022. Mapping language models to grounded conceptual spaces. In *International Conference on Learning Representations*.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam M. Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Anders Søgaard, Ivan Vulić, Sebastian Ruder, and Manaal Faruqui. 2019. *Cross-Lingual Word Embeddings*, 2 edition. Synthesis Lectures on Human Language Technologies. Morgan Claypool Publishers, United States.

Hiroki Takizawa. 2024. Othello is solved. *Preprint*, arXiv:2310.19387.

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

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony S. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. V. Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288.

Michiel van der Ree and Marco A Wiering. 2013. Reinforcement learning in the game of othello: Learning against a fixed opponent and learning from self-play. *2013 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*, pages 108–115.

Ruoyao Wang, Graham Todd, Ziang Xiao, Xingdi Yuan, Marc-Alexandre Cot'e, Peter Clark, and Peter Jansen. 2024. Can language models serve as text-based world simulators?

Jiannan Xiang, Tianhua Tao, Yi Gu, Tianmin Shu, Zirui Wang, Zichao Yang, and Zhiting Hu. 2023. Language models meet world models: Embodied experiences enhance language models. *ArXiv*, abs/2305.10626.

Tian Yun, Zilai Zeng, Kunal Handa, Ashish V. Thapliyal, Bo Pang, Ellie Pavlick, and Chen Sun. 2023. Emergence of abstract state representations in embodied sequence modeling. In *Conference on Empirical Methods in Natural Language Processing*.

A Related Work

Past work on Othello Most past works on Othello (Chang et al., 2018; van der Ree and Wiering, 2013) adopt reinforcement learning to search for moves. The first attempt to model Othello with deep neural networks dates back to 2018 (Liskowski et al., 2018), focusing on using CNNs to train a strong player. Based on it, Noever and Noever (2022) focus on designing an effective

	CHAMPIONSHIP	SYNTHETIC
Num. of Games	132,588	23,796,010
Avg. length	59.8 ± 1.5	60.0 ± 0.8
Min. length	4	9
Full length portion(%)	95.0	99.1

Table 3: Dataset statistics of the two Othello datasets.

Othello player with LLMs. Motivated by Toshniwal et al. (2021), Li et al. (2023) shift the focus to treating the game as a diagnostic tool for inducing world models from text. Following this, Nanda et al. (2023) provide evidence of a closely related linear representation of the board and propose a simple yet powerful way to interpret the model's internal state. Takizawa (2024) recently presents a provably optimal strategy for playing Othello, exploring the complexity of these strategies and whether LLMs adopt similar ones. Hua et al. (2024) adopt the idea of Othello sequence generation and introduce a Multilingual Othello task to aid in cross-lingual representation alignment.

World models The success of language models in NLP tasks has extended their application to world modeling, where the models simulate, predict, and reason about dynamic environments described by text (Hao et al., 2023; Huh et al., 2024; Patel and Pavlick, 2022; Xiang et al., 2023). For example, Li et al. (2021) fine-tune sequence models on synthetic NLP tasks to find evidence that the world state is weakly encoded in the network's activations. Hase et al. (2023) edit weights in different locations to change how a fact is stored in a model. Hewitt and Manning (2019) develop structural probes to reveal syntactic structures in word embeddings. Wang et al. (2024) evaluate how well LLMs can serve as text-based world simulators with a benchmark. Inspired by Othello-GPT, research have explored more detailed probing (Yun et al., 2023; Hazineh et al., 2023) and more complex scenarios to assess LLMs' ability to understand board states, including games like chess and maze navigation (Karvonen, 2024; Ivanitskiy et al., 2023). Our work aims to revisit the Othello World Hypothesis using a novel probing method that incorporates various LLMs.

B Dataset Statistics

The details of the two datasets are listed in Table 3.

C Experimental Methods

We implement all of the baselines under the Pytorch framework and the HuggingFace model repository. We conduct all of our experiments using 8 A100 GPUs. We use all the default parameters in the repository when fine-tuning. We first fine-tune several PLMs to generate the game moves:

- GPT2. We fine-tune GPT2 to generate the whole game sequence step by step. Specifically, we use the smallest version of GPT-2.
- Bart. We use Bart-base to generate the sequence by feeding the first token into the Encoder and fine-tuning the model to generate the remaining tokens.
- T5. Similar as Bart, we adopt T5-base in our experiment.

We then adopt several LLMs for the task:

- Flan-T5. We adopt Flan-T5-XL, which contains 3B parameters in our experiment.
- Llama-2. We use Llama-2 7B and only fine-tune the LoRA adapter in our experiment.
- Mistral. We use Mistral-7B in our experiments. Similar to Llama-2, we also only fine-tune the LoRA adapter but keep the rest of parameters fixed.

D Alignment Details

We use MUSE, a widely-used multilingual feature alignment tool, to generate alignment features from two models as training pairs. Specifically, the feature of the ith step within the same game from the two models is considered a pair, denoted as $h_{\theta 1}(x_i)$ and $h(x_i)$, respectively. These features are extracted from the last hidden layer of the Decoder. For supervised training, we randomly select 1,000 game sequences from the validation set, resulting in 60,000 training pairs. We report the average cosine similarity of the aligned features on the test set. In the unsupervised setting, we directly report the average cosine similarity score on the test set. For the PCA visualization, we randomly select a game sequence from the test set and map the model-learned features of 60 steps to coordinates for visualization.