

REVISITING THE OTHELLO WORLD MODEL HYPOTHESIS

Anonymous authors

Paper under double-blind review

ABSTRACT

Li et al. (2023) used the Othello board game as a test case for the ability of GPT-2 to induce world models, and were followed up by Nanda et al. (2023b). We briefly discuss the original experiments, expanding them to include more language models with more comprehensive probing. Specifically, we analyze sequences of Othello board states and train the model to predict the next move based on previous moves. We evaluate seven language models (GPT-2, T5, Bart, Flan-T5, Mistral, LLaMA-2, and Qwen2.5) 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 considerably stronger evidence for the Othello World Model Hypothesis than previous works.

1 INTRODUCTION

Li et al. (2023) used the Othello board game to probe the ability of LLMs 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 on about 140,000 real Othello games and another on 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 – the hypothesis that LLMs trained on Othello move sequences can induce a relevant world model, including the Othello board layout.

Nanda et al. (2023b) did a follow-up study in which they found that linear probes also work if trained slightly differently. Instead of focusing on tile color, they probed the board state relative to the current player at each timestep, using labels such as MINE, YOURS, and EMPTY. This reduced the error rate of the probes to less than 10%. They speculated 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.’

Now, training a probe 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 revisit the Othello World Model Hypothesis, reevaluating it using a methodology that does not suffer from weaknesses a)–c) (see Figure 1), in order to reassess the ability of LLMs to induce world models. If our results are positive, they will significantly strengthen the case for the hypothesis that LLMs induce

¹Li 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.

054 world models; if not, they will suggest that the evidence cited in Li et al. (2023) and
 055 Nanda et al. (2023b) was perhaps a (spurious) effect of the probing paradigm itself.
 056 We begin by re-modeling Othello using a range
 057 of model sizes (GPT-2, BART, T5, Flan-T5,
 058 LLaMA-2, Mistral, Qwen2.5), as prior research
 059 has predominantly focused on smaller models
 060 like GPT-2. We retrain these models using
 061 game data of varying scales from the two
 062 datasets presented by Li et al. (2023). Our
 063 analysis extends beyond previous studies by
 064 considering both pretrained and non-pretrained
 065 models (based on upstream language tasks),
 066 evaluating two-hop generation capabilities, and
 067 comparing models of varying sizes. To assess
 068 whether these models capture similar underlying
 069 game strategies, state representations, or other
 070 key aspects, despite differences in architecture
 071 and size, we employ representation alignment
 072 tools inspired by the literature on cross-lingual
 073 word embeddings (Søgaard et al., 2019). Finally,
 074 we visualize these results through latent move
 075 projections, enabling a deeper analysis of the
 076 internal mechanisms of models trained on the
 077 Othello game. Through these probing methods, we
 078 show that the language models – exhibit solid
 079 one-hop performance when trained on large
 080 amount of game sequence moves. We find that
 081 in some cases, all models can achieve up to
 082 99% accuracy in unsupervised grounding, which
 083 means that absent any cross-modal supervision,
 084 a model *trained to play* Othello can identify
 085 the right positions on a board. More importantly,
 086 the alignment similarity score of the board
 087 features learned by these models is surprisingly
 088 high. Additionally, the latent move projection
 089 demonstrates that the models can learn the
 090 spatial structure of the chessboard. This
 091 provides a counter-example to previous claims
 092 that mono-modal models cannot solve visual
 093 question answering problems (Bender & Koller,
 094 2020) – or, more generally, symbol grounding
 095 problems (Harnad, 1990). Beyond that, these
 096 results are significantly stronger than those in
 097 Li et al. (2023); Nanda et al. (2023b) and, in
 098 our view, provide more direct evidence of the
 099 Othello World Model Hypothesis².

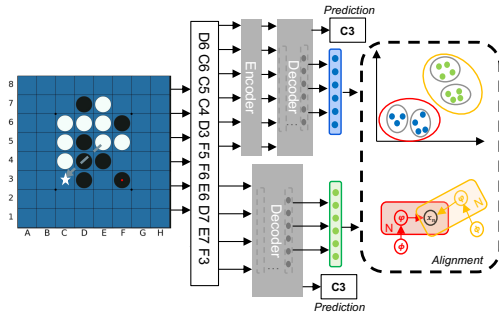


Figure 1: Experimental protocol. We re-train the Transformer-based models to predict the next move in Othello and see whether the board game layout is induced (up to isomorphism).

086 2 RELATED WORK

088 **Past work on Othello** Most past works on Othello (Chang et al., 2018; van der Ree & Wiering,
 089 2013) use reinforcement learning to search for moves. The first attempt to model Othello with deep
 090 neural networks dates back to 2018 (Liskowski et al., 2018), focusing on using CNNs to train a
 091 strong player. Based on it, Noever & Noever (2022) focus on designing an effective Othello player
 092 with LLMs. Motivated by Toshniwal et al. (2021), Li et al. (2023) shift the focus to treating the
 093 game as a diagnostic tool for inducing world models from text. Following this, Nanda et al. (2023b)
 094 provide evidence of a closely related linear representation of the board and propose a simple yet
 095 powerful way to interpret the model’s internal state. Takizawa (2024) recently presents a provably
 096 optimal strategy for playing Othello, delving into the complexity of these strategies and paving the
 097 way for future research to explore whether LLMs adopt similar approaches. Hua et al. (2024) adopt
 098 the idea of Othello sequence generation and introduce a multilingual Othello task to aid in cross-
 099 lingual representation alignment.

100 **World models** The success of language models in NLP tasks, to many, seems to turn on their
 101 ability to simulate, predict, and reason about dynamic environments as portrayed in text (Hao et al.,
 102 2023; Huh et al., 2024; Patel & Pavlick, 2022; Xiang et al., 2023). The seminal work of Li et al.
 103 (2021) presents an example of fine-tuning LLMs on synthetic NLP tasks to find evidence that world
 104 states are weakly encoded in their activations. Wang et al. (2024) evaluate how well LLMs can serve
 105 as text-based world simulators with a benchmark. Inspired by Othello-GPT, research have explored
 106 more detailed probing (Yun et al., 2023; Hazineh et al., 2023) and more complex scenarios to assess

²Detailed definition see Appendix A.

Method	Type	P	CHAMPIONSHIP			SYNTHETIC				
			2k	20k	full	2k	20k	200k	2M	full
GPT-2	D	✗	49.8	17.7	5.6	49.2	26.8	13.6	10.4	<0.1
Bart	E-D	✗	25.2	16.6	4.7	73.6	31.7	14.2	16.3	<0.1
T5	E-D	✗	20.9	15.2	4.3	65.8	28.7	15.7	10.1	<0.1
Flan-T5	E-D	✗	23.4	4.8	3.6	35.6	23.7	21.2	7.7	<0.1
LlaMa-2	D	✗	27.8	16.5	5.7	57.1	35.4	16.9	10.2	<0.1
Mistral	D	✗	22.1	14.8	4.2	48.2	34.4	17.7	8.3	<0.1
Qwen2.5	D	✗	25.2	17.3	5.5	45.9	37.8	20.1	9.2	<0.1
GPT-2	D	✓	52.6	19.7	13.6	74.4	32.4	19.9	14.1	<0.1
Bart	E-D	✓	54.0	14.6	13.7	77.2	35.8	24.4	16.6	<0.1
T5	E-D	✓	45.5	19.6	3.8	69.4	36.9	32.6	13.9	<0.1
Flan-T5	E-D	✓	31.7	4.8	3.7	70.3	25.4	45.0	8.7	<0.1
LlaMa-2	D	✓	43.1	14.7	7.0	74.6	41.5	33.4	7.6	<0.1
Mistral	D	✓	16.8	15.0	3.3	33.8	30.6	18.2	7.7	<0.1
Qwen2.5	D	✓	20.9	18.2	6.0	46.5	39.3	23.4	10.8	<0.1

Table 1: The error rate (%) of 1-hop game move generation in terms of different size of training data. ‘Type’ refers to the model type, ‘P’ denotes if the model is pretrained with upstream language modeling tasks or not. Numbers in bold represent best-performing models.

the ability of LLMs to understand board states, including for games like chess, checker and maze navigation (Karvonen, 2024; Joshi et al., 2024; Ivanitskiy et al., 2023). Our work aims to revisit the Othello World Hypothesis using novel probing methods across a number of different LLMs.

3 MODELING OTHELLO WITH LLMs

Following previous works (Liskowski et al., 2018; Li et al., 2023; Nanda et al., 2023b), 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 Othello board state. We then evaluate whether the predicted move is legal or not. 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 representing one of the 60 playable tiles where players place discs, excluding the 4 center tiles, which are already occupied at the start of the game. See Figure 1 for an example move. Our modeling of Othello, in brief, can be represented as:

$$p_{\theta}(X_{i+k}|X_{<i}) = \prod_{m=0}^k p_{\theta}(X_{i+m}|X_{<i}) = \prod_{m=0}^k \text{softmax}(f_{i+m}(x_1, x_2, \dots, x_{i+m-1})) \quad (1)$$

where x_1, x_2, \dots, x_{i-1} represent history moves, X_{i+k} represents the sequence after k generation steps. During inference, we input the previously generated game moves $X_{<i}$ at step i into the model and prompt it to generate the next steps. Unlike previous works, we not only prompt the model to generate the next move ($k = 1$) but also introduce a new test where the model generates two consecutive moves ($k = 2$), for it prompts models to simulate high-level reasoning, revealing how well LLMs understand game strategy in a zero-shot manner.

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 of the generated move. That if a generated move is not legal, we count it as an error. Specifically, we extend the 1-hop step generation setting in Li et al. (2023) and investigate 2-hop move generation for investigating the model’s capability to anticipate more strategic, long-term planning in Othello. This involves

162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177

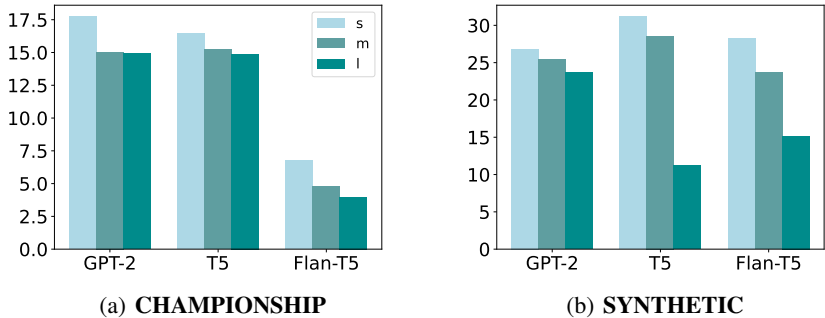
Method	Type	P	CHAMPIONSHIP			SYNTHETIC				
			2k	20k	full	2k	20k	200k	2M	full
GPT-2	D	✗	78.5	34.7	28.1	76.3	70.8	43.6	29.0	5.2
Bart	E-D	✗	54.2	31.1	23.4	86.5	67.2	44.8	35.7	4.2
T5	E-D	✗	48.8	28.7	24.4	88.2	67.7	46.9	35.9	3.4
Flan-T5	E-D	✗	51.8	20.8	21.9	79.6	63.1	48.6	26.7	2.8
LlaMa-2	D	✗	60.9	36.3	26.4	87.3	67.8	45.2	36.3	5.5
Mistral	D	✗	51.4	31.7	22.3	71.2	77.1	47.9	26.4	3.0
Qwen2.5	D	✗	55.9	25.4	22.8	77.6	65.3	44.2	28.7	3.3
GPT-2	D	✓	92.2	43.4	37.2	99.6	72.6	45.5	34.4	6.2
Bart	E-D	✓	87.0	34.5	27.1	97.8	76.9	64.0	44.5	5.1
T5	E-D	✓	86.5	36.4	27.0	99.6	78.8	59.9	46.9	4.6
Flan-T5	E-D	✓	67.9	31.8	26.5	98.6	80.8	79.7	35.3	3.9
LlaMa-2	D	✓	66.9	33.4	33.0	94.2	77.6	62.1	33.2	5.2
Mistral	D	✓	52.0	40.8	25.4	80.3	76.0	42.3	35.0	3.8
Qwen2.5	D	✗	63.1	38.4	25.8	85.0	79.3	45.1	36.0	3.9

178
179

Table 2: The error rate (%) of 2-hop game move generation.

180
181

182
183
184
185
186
187
188
189



190
191

Figure 2: Othello 1-hop generation error rate under different model sizes. All models are non-pretrained versions fine-tuned with 20k game sequences.

192
193

194
195
196
197
198
199
200
201
202

verifying whether the top-1 prediction is legal when the model is prompted to generate one and two moves at a time. We present the average error rate across all game sequences. 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.

203
204
205
206
207

We perform our experiments using several existing baselines, with both Encoder-Decoder or Decoder-only structures. We first adopt some popular PLMs such as GPT-2 (Radford et al., 2019), T5 (Raffel et al., 2019), and Bart (Lewis et al., 2019). We then adopt several LLMs to see the their performance on this task, including Flan-T5 (Chung et al., 2022), LLaMa-2 (Touvron et al., 2023), Mistral (Jiang et al., 2023), and Qwen2.5 (Hui et al., 2024). Details see Appendix C.

208
209

3.2 EVALUATION OF LLM PERFORMANCE IN OTHELLO MOVE GENERATION

210
211
212
213
214
215

We perform experiments using various methods and present the results in Tables 1 and 2. From our observations, several key findings emerge. Firstly, there is **no clear winner** between models with an Encoder-Decoder architecture, such as T5 or Flan-T5, and those with a Decoder-only architecture, such as GPT-2, LLaMA-2, or Qwen2.5 in terms of performance on this task. This indicates that the architectural differences between these models do not significantly impact their ability to generate Othello game steps. However, one consistent trend is the positive correlation between the amount of training data and overall model performance. As we increase the scale of the training data, all mod-

216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269

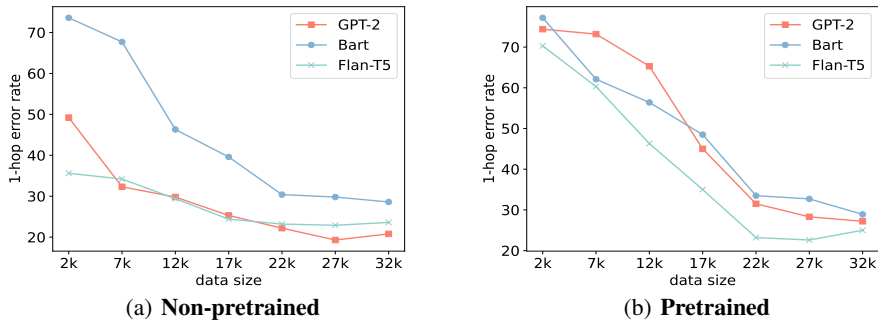


Figure 3: Analysis of 1-hop error rates on the SYNTHETIC dataset with varying data scales.

els tend to improve, underscoring the importance of data availability in mastering complex tasks like Othello move generation. In comparison to smaller language models, LLMs such as Mistral and Flan-T5 demonstrate clear superiority in this task, suggesting that **model size and capacity** are critical factors in understanding Othello game step generation. Larger models are better equipped to capture the intricate patterns and strategies within the game, likely due to their increased representational capacity. Interestingly, we also find that pretrained language knowledge, while generally beneficial for a wide range of natural language tasks, sometimes **negatively impacts** a model’s ability to understand and generate game steps. Specifically, the pretrained versions of many models perform worse than their non-pretrained counterparts in this task, which could indicate that knowledge learned from upstream language tasks introduces biases or distracts from learning the specific structure and rules of Othello. Furthermore, while fine-tuning models on a large amount of data leads to reasonable performance in generating a single step (1-hop), generating more than one step consecutively remains a significant challenge. Even with large-scale data, models struggle to accurately predict two or more consecutive moves. This shows the potential limitation of the 1-hop evaluation since while it mostly focuses on the immediate next move based on the current board state, it inherently overlooks the deeper decision-making process required for gameplay strategies.

3.3 IMPACT OF MODEL SIZE ON OTHELLO MOVE GENERATION

To further explore the impact of model size on the ability to model Othello moves, we analyze the performance of various models across different size configurations, as depicted in Figure 2. For each model, we evaluate performance in small, medium, and large size versions, allowing us to compare how scaling up model capacity affects accuracy in generating game moves. The results show a clear trend: **as model size increases, the error rate consistently decreases across both datasets**. This trend is particularly pronounced in the SYNTHETIC dataset, where larger models achieve significantly lower error rates compared to their smaller counterparts. The stronger improvement in the SYNTHETIC dataset suggests that larger models are better at capturing the structured patterns present in the synthetic data, likely due to their enhanced capacity for learning complex representations and generalizing across more varied scenarios. These findings highlight the importance of model scaling, showing that increasing the model size can lead to substantial performance gains in Othello move generation, especially in environments where the data is highly structured or synthetic in nature. Furthermore, the results emphasize that larger models are not just marginally better, but often significantly outperform smaller models, reinforcing the need to consider model capacity as a critical factor when tackling tasks that require a deep understanding of game strategies and sequential decision-making processes.

3.4 IMPACT OF DATA SIZE ON OTHELLO MOVE GENERATION

In Table 2, 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 32k. According to Figure 3, the performance of all models improves gradually as the dataset size increases. Interestingly, non-pretrained models exhibit a faster reduction in er-

ror rates within the 2k to 12k data size range, with diminishing improvements beyond that point compared to pretrained models. In contrast, pretrained models take longer to achieve comparable performance, highlighting their slower adaptation to the task. This suggests that non-pretrained models are better suited for quickly learning game rules and adapting to fundamental patterns in the data. Furthermore, it indicates that the prior natural language knowledge embedded in pretrained models does not significantly enhance their understanding of the game. This observation aligns with our findings discussed in Section 3.2, where we also observed the limited impact of pretrained knowledge in tasks requiring specialized domain adaptation.

4 OTHELLO REPRESENTATION ALIGNMENT ACROSS LANGUAGE MODELS

Drawing inspiration from the literature on cross-lingual word embeddings, we perform Othello representation alignment across different models to compare how each model, despite differences in architecture and size, internalizes and represents game strategies and states. This helps us assess whether the learned representations in Section 3 are consistent across models and whether they capture similar underlying patterns essential for accurate Othello move generation.

4.1 ALIGNMENT METHOD

To validate the Othello World Model Hypothesis, we directly evaluate the internal representation of the Othello board in language models. Using the representations from different models, denoted as F_1, F_2 from the same input sequence $X_{<i}$, we perform mapping training under both *supervised* and *unsupervised* scenarios³. A linear mapping W is learned to map F_1 and F_2 into the same space:

$$W^* = \arg \min_{W \in M_i(\mathbb{R})} \|WF_1 - F_2\| \quad (2)$$

where $F_1, F_2 \in \mathbb{R}^{i \times h}$ are representations from the final hidden Decoder layer in different language models trained for Othello generation. $M_i(\mathbb{R})$ is the space of $i \times i$ matrices of real numbers.

Supervised training. We consider the internal representations of different models within different source and target spaces. For supervised training (see Algorithm 1)⁴, we use the pairwise data to learn a mapping from the source to the target space using iterative Procrustes alignment (Gower & Dijkstra, 2004). We use representations from two models as training pairs. Specifically, the representations of the i th step within the same game from the two models are considered a pair, denoted as $h_{\theta_1}(X_{<i})$ and $h_{\theta_2}(X_{<i})$, respectively. In our experiment, we randomly select 1,000 game sequences from the validation set as training pairs.

Algorithm 1: Supervised Training for Othello Representation Alignment

Inputs :

$h_{\theta_1}(\cdot), h_{\theta_2}(\cdot)$ representations from the final hidden layer of Decoder in two models: Θ_1, Θ_2
 $X_{<i} = \{x_1, \dots, x_{i-1}\}$ input game sequence at time step i
 r number of refinement iterations

Output:

s Similarity score of the aligned feature learned from the two models
 $F_1 \leftarrow h_{\theta_1}(X_{<i}), F_2 \leftarrow h_{\theta_2}(X_{<i})$
for $i = 1$ **to** r **do**
 if $i! = 1$ **then**
 $F_1 \leftarrow \text{BuildDic}(F_1), F_2 \leftarrow \text{BuildDic}(F_2)$ // build a dictionary from aligned
 embeddings containing best aligned pairs
 $W \leftarrow \text{Procrustes}(F_1, F_2)$
 $F_1 \leftarrow WF_1$
 $s \leftarrow \text{CosSim}(F_1, F_2)$

Unsupervised training. For unsupervised training, without any parallel data or anchor points, following Conneau et al. (2018), we learn the mapping through a combination of adversarial training

³Both of the algorithms are implemented using MUSE, a library designed for multilingual embedding alignment (<https://github.com/facebookresearch/MUSE>).

⁴More details (e.g. BuildDict() of Algorithms 1, 2) see Conneau et al. (2018).

and iterative Procrustes refinement (Lample et al., 2018) (see Algorithm 2). The process involves first learning an initial proxy of the mapping W using an adversarial criterion. Where an additional Discriminator model is trained to identify the origin of an embedding, yet the target mapping W aims at preventing the discriminator from doing so. Then, the mapping W is further refined via Procrustes using the same strategy in supervised training. We then report the average cosine similarity of the aligned features on the test set.

Algorithm 2: Unsupervised Training for Othello Representation Alignment

Inputs :

$h_{\theta_1}(\cdot), h_{\theta_2}(\cdot)$ representations from the final hidden layer of Decoder in two models: Θ_1, Θ_2
 $X_{<i} = \{x_1, \dots, x_{i-1}\}$ input game sequence at time step i
 k, r number of adversarial training iterations, number of refinement iterations

Output:

s Similarity score of the aligned feature learned from the two models
 $F_1 \leftarrow h_{\theta_1}(X_{<i}), F_2 \leftarrow h_{\theta_2}(X_{<i})$
 RandomInitialize(W)
for $i = 1$ **to** k **do**
 $\mathcal{D} \leftarrow \text{TrainDiscriminator}(\mathcal{W}, \mathcal{D}, F_1, F_2)$ // train the discriminator \mathcal{D}
 $W \leftarrow \text{FoolDiscriminator}(\mathcal{W}, \mathcal{D}, F_1, F_2)$ // train W to fool the discriminator
 $F_1 \leftarrow WF_1$
for $i = 1$ **to** r // refine W
do
 $F_1 \leftarrow \text{BuildDic}(F_1), F_2 \leftarrow \text{BuildDic}(F_2)$
 $W \leftarrow \text{Procrustes}(F_1, F_2)$
 $F_1 \leftarrow WF_1$
 $s \leftarrow \text{CosSim}(F_1, F_2)$

4.2 MAPPING RESULT

We probe different models by aligning their representations into one joint vector space. We report the cosine similarity of the aligned features score under both supervised (Conneau et al., 2018) and unsupervised (Lample et al., 2018) settings in Table 3⁵.

Src.	Trg.	Supervised		Unsupervised	
		CHAM.	SYN.	CHAM.	SYN.
GPT-2	Bart	81.4	93.1	80.3	91.3
GPT-2	T5	83.0	85.0	76.4	80.1
Bart	T5	69.2	84.5	85.2	81.1
GPT-2	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
Qwen2.5	LlaMa-2	84.2	80.1	81.3	84.9

Table 3: Representation alignment cosine similarity (%) results. Src. and Trg. represent source and target space. CHAM., SYN represent CHAMPIONSHIP and SYNTHETIC dataset.

From the results, we observe consistently high similarity scores across different language models, indicating that despite architectural differences, these models capture similar underlying representations when tasked with the Othello game. For instance, the SYNTHETIC supervised similarity score between GPT-2 (a Decoder-only model) and Bart (an Encoder-Decoder model) reaches an impressive 93.1%. This suggests that, although these models process information differently due to their structural variances, they still converge on shared knowledge and representations when learning to model the Othello task. Such a high similarity score points to the possibility that both model types learn similar strategic patterns and rules intrinsic to the game, reinforcing the idea that fundamental aspects of the Othello task are captured across architectures.

⁵We use the non-pretrained version based on 20k training data for all models.

378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431

4.3 PCA VISUALIZATION

In order to vividly show such alignment, we also demonstrate the dimension-reduced PCA coordinate of 60 step features $h_\theta(X)$ within one entire random game in Figure 4. We also observe highly similar step representations across different models. This suggests that these models are learning comparable internal representations of the game states, indicating that the models are aligned in how they interpret the sequential nature of Othello. Even though they may be built differently (e.g., Decoder-only versus Encoder-Decoder), the core representations they learn about the game states converge to a similar space. This result highlights a level of consistency and robustness in the way generative models process game-related information. Despite differences in architecture or training objectives, the models seem to internalize and represent Othello game states in a similar manner. This convergence suggests that these models, when trained on the Othello task, are not only learning task-specific patterns but are also aligning on a shared understanding of the underlying problem space. To sum up, such alignment enhances the interpretability of these models, as their internal representations become more comparable.

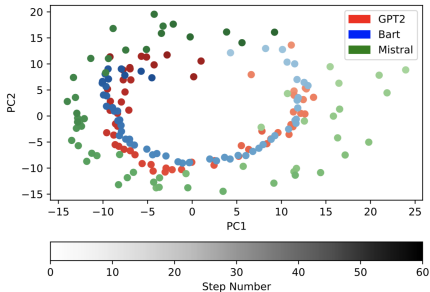


Figure 4: PCA visualization of the 60 steps from various models within one game.

4.4 MAPPING ACROSS DIFFERENT LAYERS

We compare the mapping similarity across different Decoder hidden layers in GPT-2 and Flan-T5⁶ to understand how each model progressively learns to represent the Othello game, evolving from simple board states to more complex strategies. As shown in Figure 5, despite their structural differences, GPT-2 and Flan-T5 exhibit similar learned representations at corresponding layers. Both models, when trained on Othello game sequences, seem to converge toward learning comparable internal representations, as highlighted by the heatmap. This conclusion is supported by the following observations:

- (1) **High Similarity in the Upper Right Diagonal.** The heatmap reveals a prominent diagonal pattern where corresponding layers from GPT-2 and Flan-T5 show high similarity, especially in the upper half of the heatmap. This suggests that, despite their differing architectures (GPT-2 being autoregressive and Flan-T5 following an Encoder-Decoder structure), models eventually learn something in common (particularly at layer 11, where high similarity is observed) despite the difference from the beginning. This alignment indicates that their layer-wise learning processes evolve in comparable ways as they both adapt to the Othello game environment.
- (2) **Layer-Specific Correspondences.** We notice that specific layers in GPT-2 show high similarity with certain layers in Flan-T5, even though they may not follow a strict diagonal pattern, this suggests that both models are learning certain shared features or patterns in game sequences at particular stages of their processing pipelines.

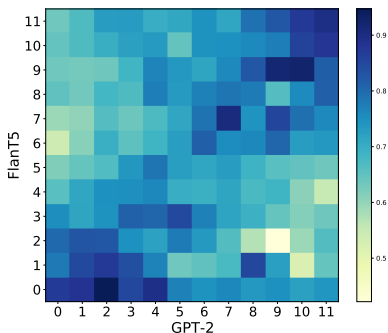


Figure 5: Decoder feature similarity heatmap across different layers.

⁶We use GPT-2-small and Flan-T5-Base trained on 20k SYNTHETIC dataset, as both have 12 decoder hidden layers.

432

433

434

435

436

437

438

439

440

441

442

443

444

445

446

447

448

449

450

451

452

453

454

455

456

457

458

459

460

461

462

463

464

465

466

467

468

469

470

471

472

473

474

475

476

477

478

479

480

481

482

483

484

485

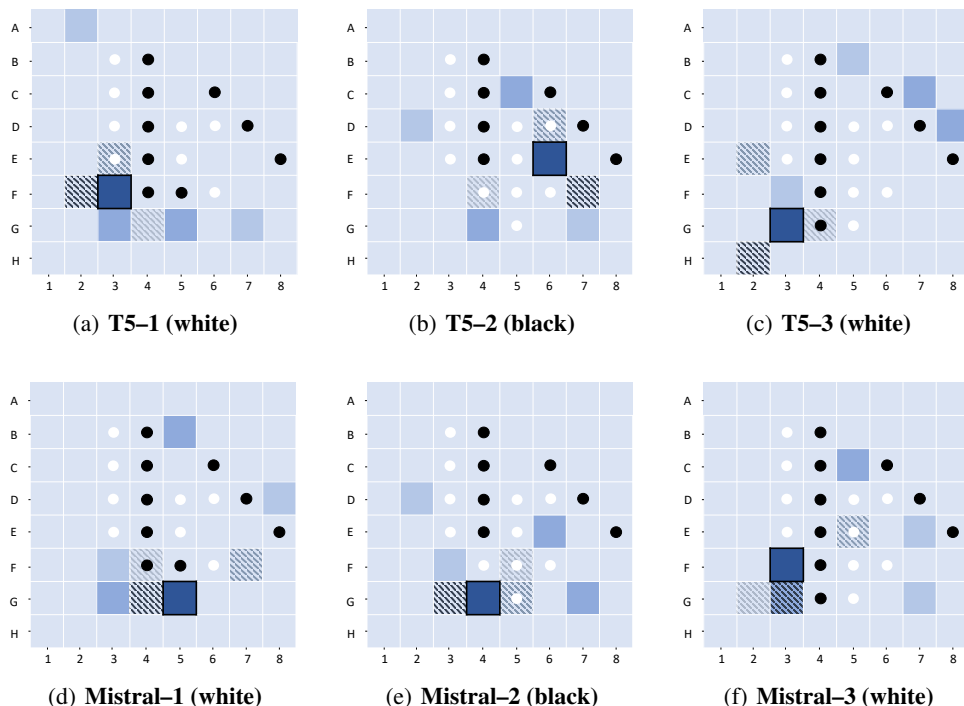


Figure 6: Othello latent move projection from two best performed models. Colors indicate the likelihood of the position of the next step. Shadows highlight the top three tiles with embeddings closest to the top candidate, with the darkest color in the black box.

5 LATENT MOVE PROJECTION: WHAT ELSE DOES LLMs LEARN?

To gain deeper insights into how models learn strategies and predict future moves, we project latent features onto a visual space. For a given game sequence $X_{<i}$, we highlight the top-5 candidate tile positions with the highest predicted probabilities for the next move. Additionally, we compare the embeddings of the top candidate tile with those of the other tiles. We mark the top three tiles whose embeddings are closest to the top candidate to examine their spatial relationships on the board.

We perform latent move projection on the Othello game steps of two models in Figure 6 (results for other models in Appendix G). 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 rule 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 Figure 10(a), G4 in Figure 10(g)) 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.

6 LIMITATIONS

Although this work demonstrates the ability of different language models to understand Othello game rules, several limitations persist that require further investigation:

Challenges in Multi-step Move Generation. While language models can predict the next move with reasonable accuracy, they struggle to predict entire game sequences. The key question is whether strong multi-step performance is a reasonable expectation. Othello is a dynamic game where optimal play often involves sacrificing short-term gains for long-term advantages. The com-

486 plexity arises from the interplay of distinct player strategies and the rotational invariance of the
487 board, leading to many game states where the subjectively or objectively best move is inherently
488 underdetermined. As a result, the ability to accurately predict entire sequences may remain elusive,
489 given the complexity and variability of decision-making in the game.

490 **Limitations in Data Requirements.** Our experiments show that reducing the 1-hop error rate to
491 less than 0.1% demands a large volume of training data. This reliance on vast datasets presents
492 a scalability issue, as access to Othello game data is limited. Moreover, training on such large
493 datasets is computationally expensive and time-consuming, which can be a prohibitive factor for
494 many researchers or organizations without access to substantial computational resources.

496 7 POTENTIAL IMPACT

499 Training language models on Othello game sequences can imply that LLMs function as a world
500 model because it showcases their ability to learn and internalize the structured dynamics and rules
501 of a complex system, rather than merely memorizing patterns. Investigating the parallels between
502 how language models learn structured representations and how humans internalize similar concepts
503 can shed light on the cognitive processes underlying reasoning, strategy, and language. This could
504 deepen our understanding of human cognition and inform theories of learning and representation.
505 The observation that language models, regardless of architecture or scale, learn similar patterns from
506 Othello game sequences suggests that these models converge on universal representations when
507 trained on structured data. This implies that the underlying mechanisms of representation learning
508 in LLMs are robust and consistent, highlighting their ability to capture the rules and dynamics
509 of structured systems. The ability of language models to learn patterns from Othello sequences
510 provides more hints on the idea that LLMs can act as world models, capable of internalizing rules,
511 strategies, and dynamics. This has far-reaching implications for tasks requiring reasoning about
512 complex environments, such as planning, simulation, and autonomous decision-making.

513 8 FUTURE DIRECTIONS

515 We list several possible future directions to study how our results could generalize to other broader
516 scenarios.

518 **More Complicated Games.** Since this work is primarily limited to the Othello game, an intriguing
519 question arises: could similar findings be observed in other games such as chess, checkers, or
520 Go? These games, like Othello, involve strategic planning, dynamic state transitions, and trade-offs
521 between short-term gains and long-term advantages. Exploring how large language models (LLMs)
522 learn and represent strategies in these contexts could be highly valuable.

523 **Multimodal Support.** Leveraging Multimodal LLMs (MLLMs) to train models and investigate feature
524 alignment across different modalities is also a highly relevant and promising research direction.
525 In the context of Othello, this approach could involve aligning visual representations of the game
526 board with text-based sequential moves. Such alignment can help bridge the gap between symbolic
527 reasoning and natural language understanding, enabling models to not only predict optimal moves
528 but also provide hints if the world model theory could also be applied in other modalities.

530 9 CONCLUSION

532 We conduct a detailed probing of language models' ability to predict legal moves in the Othello
533 board game, based on the settings in Li et al. (2023). We evaluate seven language models, training
534 them to predict the next move based on previous moves. All seven models achieve almost 'perfect'
535 one-hop move prediction performance when trained with large amount of data. We then adopt
536 representation alignment tools to align the learned game state features from different models into
537 one joint space. We observe high similarity in the board features they learned. In addition, latent
538 move projection is performed to show the models not only understand the game mechanics but
539 also capture the spatial relationships between tiles. These results, in our view, provide more solid
evidence to date of the Othello World Model Hypothesis presented in previous works.

REFERENCES

- 540
541
542 Mostafa Abdou, Artur Kulmizev, Daniel Hershcovich, Stella Frank, Ellie Pavlick, and Anders
543 Sogaard. Can language models encode perceptual structure without grounding? a case study
544 in color. In *Conference on Computational Natural Language Learning*, 2021. URL <https://api.semanticscholar.org/CorpusID:237491991>.
545
- 546 Maria Barrett, Yova Kementchedjhiya, Yanai Elazar, Desmond Elliott, and Anders Sogaard. Adver-
547 sarial removal of demographic attributes revisited. In Kentaro Inui, Jing Jiang, Vincent Ng, and
548 Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Lan-
549 guage Processing and the 9th International Joint Conference on Natural Language Processing
(EMNLP-IJCNLP)*, pp. 6330–6335, Hong Kong, China, November 2019. Association for Compu-
550 tational Linguistics. doi: 10.18653/v1/D19-1662. URL [https://aclanthology.org/
551 D19-1662](https://aclanthology.org/D19-1662).
552
- 553 Emily M. Bender and Alexander Koller. Climbing towards NLU: On meaning, form, and under-
554 standing in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for
555 Computational Linguistics*, pp. 5185–5198, Online, July 2020. Association for Computational
556 Linguistics. doi: 10.18653/v1/2020.acl-main.463. URL [https://aclanthology.org/
557 2020.acl-main.463](https://aclanthology.org/2020.acl-main.463).
558
- 559 Naiyuan Chang, Chih-Hung Chen, Shun-Shii Lin, and Surag Nair. The big win strategy on multi-
560 value network: An improvement over alphazero approach for 6x6 othello. *Proceedings of the
561 2018 International Conference on Machine Learning and Machine Intelligence*, 2018. URL
562 <https://api.semanticscholar.org/CorpusID:54461634>.
- 563 Hyung Won Chung, Le Hou, S. Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi
564 Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun
565 Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Dasha Valter, Sharan Narang, Gau-
566 rav Mishra, Adams Wei Yu, Vincent Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav
567 Petrov, Ed Huai hsin Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and
568 Jason Wei. Scaling instruction-finetuned language models. *ArXiv*, abs/2210.11416, 2022. URL
569 <https://api.semanticscholar.org/CorpusID:253018554>.
- 570 Alexis Conneau, Guillaume Lample, Marc’Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou.
571 Word translation without parallel data. *The Sixth International Conference on Learning Repre-
572 sentations*, 2018.
573
- 574 John C Gower and Garnt B Dijkstraerhuis. *Procrustes problems*, volume 30. OUP Oxford, 2004.
- 575 Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu.
576 Reasoning with language model is planning with world model. *ArXiv*, abs/2305.14992, 2023.
577 URL <https://api.semanticscholar.org/CorpusID:258865812>.
578
- 579 Stevan Harnad. The symbol grounding problem. *Physica D*, 42:335–346, 1990.
- 580 Dean S. Hazineh, Zechen Zhang, and Jeffery Chiu. Linear latent world models in simple trans-
581 formers: A case study on othello-gpt. *ArXiv*, abs/2310.07582, 2023. URL [https://api.
582 semanticscholar.org/CorpusID:263834692](https://api.semanticscholar.org/CorpusID:263834692).
583
- 584 Tianze Hua, Tian Yun, and Ellie Pavlick. mothello: When do cross-lingual representation alignment
585 and cross-lingual transfer emerge in multilingual models? *ArXiv*, abs/2404.12444, 2024. URL
586 <https://api.semanticscholar.org/CorpusID:269282665>.
587
- 588 Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. The platonic representation hy-
589 pothesis. 2024. URL <https://api.semanticscholar.org/CorpusID:269757765>.
- 590 Binyuan Hui, Jian Yang, Zeyu Cui, Jiayi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Ji-
591 ajun Zhang, Bowen Yu, Kai Dang, An Yang, Rui Men, Fei Huang, Shanghaoran Quan,
592 Xingzhang Ren, Xuancheng Ren, Jingren Zhou, and Junyang Lin. Qwen2.5-coder technical
593 report. *ArXiv*, abs/2409.12186, 2024. URL [https://api.semanticscholar.org/
CorpusID:272707390](https://api.semanticscholar.org/CorpusID:272707390).

- 594 Michael I. Ivanitskiy, Alex F Spies, Tilman Rauker, Guillaume Corlouer, Chris Mathwin, Lu-
595 cia Quirke, Can Rager, Rusheb Shah, Dan Valentine, Cecilia G. Diniz Behn, Katsumi In-
596 oue, and Samy Wu Fung. Structured world representations in maze-solving transform-
597 ers. *ArXiv*, abs/2312.02566, 2023. URL [https://api.semanticscholar.org/
598 CorpusID:265659365](https://api.semanticscholar.org/CorpusID:265659365).
- 599 Albert Qiaochu Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh
600 Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lu-
601 cile Saulnier, L’elio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao,
602 Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b. *ArXiv*,
603 abs/2310.06825, 2023. URL [https://api.semanticscholar.org/CorpusID:
604 263830494](https://api.semanticscholar.org/CorpusID:263830494).
- 605 Abhinav Joshi, Vaibhav Sharma, and Ashutosh Modi. Checkersgpt: Learning world models through
606 language modeling. In *Annual Meeting of the Association for Computational Linguistics*, 2024.
607 URL <https://api.semanticscholar.org/CorpusID:272779453>.
- 608 Adam Karvonen. Emergent world models and latent variable estimation in chess-playing language
609 models. *ArXiv*, abs/2403.15498, 2024. URL [https://api.semanticscholar.org/
610 CorpusID:268681535](https://api.semanticscholar.org/CorpusID:268681535).
- 611 Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc’Aurelio Ranzato. Unsupervised
612 machine translation using monolingual corpora only. *The Sixth International Conference on
613 Learning Representations*, 2018.
- 614 Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdel rahman Mohamed,
615 Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence
616 pre-training for natural language generation, translation, and comprehension. In *Annual
617 Meeting of the Association for Computational Linguistics*, 2019. URL [https://api.
618 semanticscholar.org/CorpusID:204960716](https://api.semanticscholar.org/CorpusID:204960716).
- 619 Belinda Z. Li, Maxwell Nye, and Jacob Andreas. Implicit representations of meaning in neural
620 language models. In *Annual Meeting of the Association for Computational Linguistics*, 2021.
621 URL <https://api.semanticscholar.org/CorpusID:235294296>.
- 622 Kenneth Li, Aspen K Hopkins, David Bau, Fernanda Viégas, Hanspeter Pfister, and Martin Wat-
623 tenberg. Emergent world representations: Exploring a sequence model trained on a synthetic
624 task. In *The Eleventh International Conference on Learning Representations*, 2023. URL
625 https://openreview.net/forum?id=DeG07_TcZvT.
- 626 Paweł Liskowski, Wojciech Jaśkowski, and Krzysztof Krawiec. Learning to play othello with deep
627 neural networks. *IEEE Transactions on Games*, 10(4):354–364, 2018. doi: 10.1109/TG.2018.
628 2799997.
- 629 Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Dis-
630 tributed representations of words and phrases and their compositionality. In C.J.
631 Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger (eds.), *Ad-
632 vances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.,
633 2013. URL [https://proceedings.neurips.cc/paper_files/paper/2013/
634 file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf).
- 635 Neel Nanda, Andrew Lee, and Martin Wattenberg. Emergent linear representations in world models
636 of self-supervised sequence models. *ArXiv*, abs/2309.00941, 2023a. URL [https://api.
637 semanticscholar.org/CorpusID:261530966](https://api.semanticscholar.org/CorpusID:261530966).
- 638 Neel Nanda, Andrew Lee, and Martin Wattenberg. Emergent linear representations in world mod-
639 els of self-supervised sequence models. In Yonatan Belinkov, Sophie Hao, Jaap Jumelet, Na-
640 joun Kim, Arya McCarthy, and Hosein Mohebbi (eds.), *Proceedings of the 6th BlackboxNLP
641 Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 16–30, Singapore, Decem-
642 ber 2023b. Association for Computational Linguistics. doi: 10.18653/v1/2023.blackboxnlp-1.2.
643 URL <https://aclanthology.org/2023.blackboxnlp-1.2>.
- 644 Samantha E. Miller Noever and David Noever. Word play for playing othello (reverses), 2022.
645
646
647

- 648 Roma Patel and Ellie Pavlick. Mapping language models to grounded conceptual spaces.
649 In *International Conference on Learning Representations, 2022*. URL <https://api.semanticscholar.org/CorpusID:251647156>.
650
651
- 652 Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever.
653 Language models are unsupervised multitask learners. 2019. URL <https://api.semanticscholar.org/CorpusID:160025533>.
654
- 655 Colin Raffel, Noam M. Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena,
656 Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified
657 text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67, 2019. URL <https://api.semanticscholar.org/CorpusID:204838007>.
658
- 659 Anders Søgaard, Ivan Vulić, Sebastian Ruder, and Manaal Faruqui. *Cross-Lingual Word Embed-*
660 *dings*. Synthesis Lectures on Human Language Technologies. Morgan Claypool Publishers,
661 United States, 2 edition, 2019. doi: 10.2200/S00920ED2V01Y201904HLT042.
662
- 663 Hiroki Takizawa. Othello is solved, 2024.
- 664 Shubham Toshniwal, Sam Wiseman, Karen Livescu, and Kevin Gimpel. Learning chess blindfolded:
665 Evaluating language models on state tracking. *ArXiv*, abs/2102.13249, 2021. URL <https://api.semanticscholar.org/CorpusID:232069003>.
666
667
- 668 Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei,
669 Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas
670 Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernan-
671 des, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, An-
672 thony S. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Ma-
673 dian Khabsa, Isabel M. Kloumann, A. V. Korenev, Punit Singh Koura, Marie-Anne Lachaux,
674 Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mi-
675 haylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi
676 Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia
677 Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan
678 Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez,
679 Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned
680 chat models. *ArXiv*, abs/2307.09288, 2023. URL <https://api.semanticscholar.org/CorpusID:259950998>.
- 681 Michiel van der Ree and Marco A Wiering. Reinforcement learning in the game of othello: Learning
682 against a fixed opponent and learning from self-play. *2013 IEEE Symposium on Adaptive Dynamic*
683 *Programming and Reinforcement Learning (ADPRL)*, pp. 108–115, 2013. URL <https://api.semanticscholar.org/CorpusID:1695896>.
684
- 685 Ruoyao Wang, Graham Todd, Ziang Xiao, Xingdi Yuan, Marc-Alexandre Cot’e, Peter Clark, and
686 Peter Jansen. Can language models serve as text-based world simulators? 2024. URL <https://api.semanticscholar.org/CorpusID:270371867>.
687
688
- 689 Jiannan Xiang, Tianhua Tao, Yi Gu, Tianmin Shu, Zirui Wang, Zichao Yang, and Zhiting
690 Hu. Language models meet world models: Embodied experiences enhance language mod-
691 els. *ArXiv*, abs/2305.10626, 2023. URL <https://api.semanticscholar.org/CorpusID:258762577>.
692
- 693 Tian Yun, Zilai Zeng, Kunal Handa, Ashish V. Thapliyal, Bo Pang, Ellie Pavlick, and Chen
694 Sun. Emergence of abstract state representations in embodied sequence modeling. In *Confer-*
695 *ence on Empirical Methods in Natural Language Processing, 2023*. URL <https://api.semanticscholar.org/CorpusID:265034007>.
696
697

698 A OTHELLO WORLD MODEL HYPOTHESIS

700 According to previous works Li et al. (2023); Nanda et al. (2023a), a world model refers to a rep-
701 resentation or a mapping of a world, ideally a homomorphism. Language models have been shown

	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 4: Dataset statistics of the two Othello datasets.

to develop internal representations for simple concepts, such as color and direction Patel & Pavlick (2022); Abdou et al. (2021). Training language models on Othello game sequences further supports the idea that LLMs can function as a world model. This is demonstrated by their ability to learn and internalize the structured dynamics and rules of a complex system, rather than merely memorizing patterns. This capacity highlights their potential for understanding and representing intricate environments through abstract, systematic reasoning.

B DATASET STATISTICS

The details of the two datasets are listed in Table 4.

C COMPARED METHODS

We perform our experiments using several existing baselines, with both Encoder-Decoder or Decoder-only structures. We first adopt some popular language models such as

GPT-2. We fine-tune GPT-2 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 LLaMA2-7B 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.

Qwen2.5. We use Qwen2.5-7B in our experiments, one of the most state-of-the-art LLMs for sequence generation.

D MODEL SIZE ANALYSIS ON TWO-HOP GENERATION

We present the 2-hop performance across various model sizes in Figure 7. As we scale up the model, the error rate decreases, suggesting that a larger model size positively affects game understanding. However, the impact of model size diminishes when compared to the 1-hop performance, indicating a diminishing return on performance gains with increased model size.

E DATA SIZE ANALYSIS ON CHAMPIONSHIP DATASET

We also present the data size analysis on the CHAMPIONSHIP dataset (see Figure 8). We see similar conclusions as in Figure 3. The prediction accuracy gets better when we increase the data size. Also, the error rate demonstrates a more steady drop in models pretrained with upstream language modeling tasks.

756

757

758

759

760

761

762

763

764

765

766

767

768

769

770

771

772

773

774

775

776

777

778

779

780

781

782

783

784

785

786

787

788

789

790

791

792

793

794

795

796

797

798

799

800

801

802

803

804

805

806

807

808

809

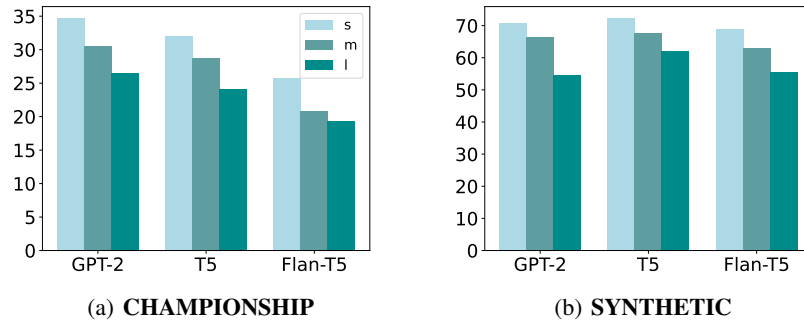


Figure 7: Othello 2-hop generation performance under different model sizes. All models are non-pretrained versions fine-tuned with 20k game sequences.

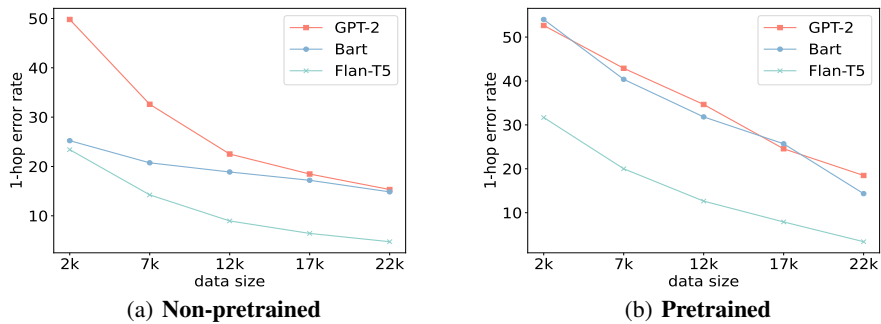


Figure 8: Analysis of 1-hop error rates on the CHAMPIONSHIP dataset with varying data scales.

F SUPERVISED MAPPING HEATMAP

We also present the supervised mapping results for the same setting in Section 4.4. The mapping in Figure 9 reveals a more pronounced diagonal pattern of similarity, with particularly high similarity observed in the upper-right corner. This provides further evidence that the models converge and acquire shared knowledge when trained on Othello data, indicating a strong alignment in their learned representations.

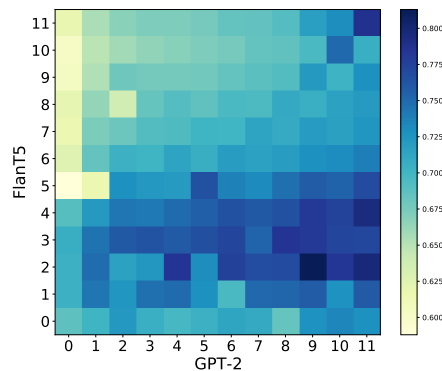


Figure 9: Decoder feature similarity (supervised) heatmap across different layers.

810

811

812

813

814

815

816

817

818

819

820

821

822

823

824

825

826

827

828

829

830

831

832

833

834

835

836

837

838

839

840

841

842

843

844

845

846

847

848

849

850

851

852

853

854

855

856

857

858

859

860

861

862

863

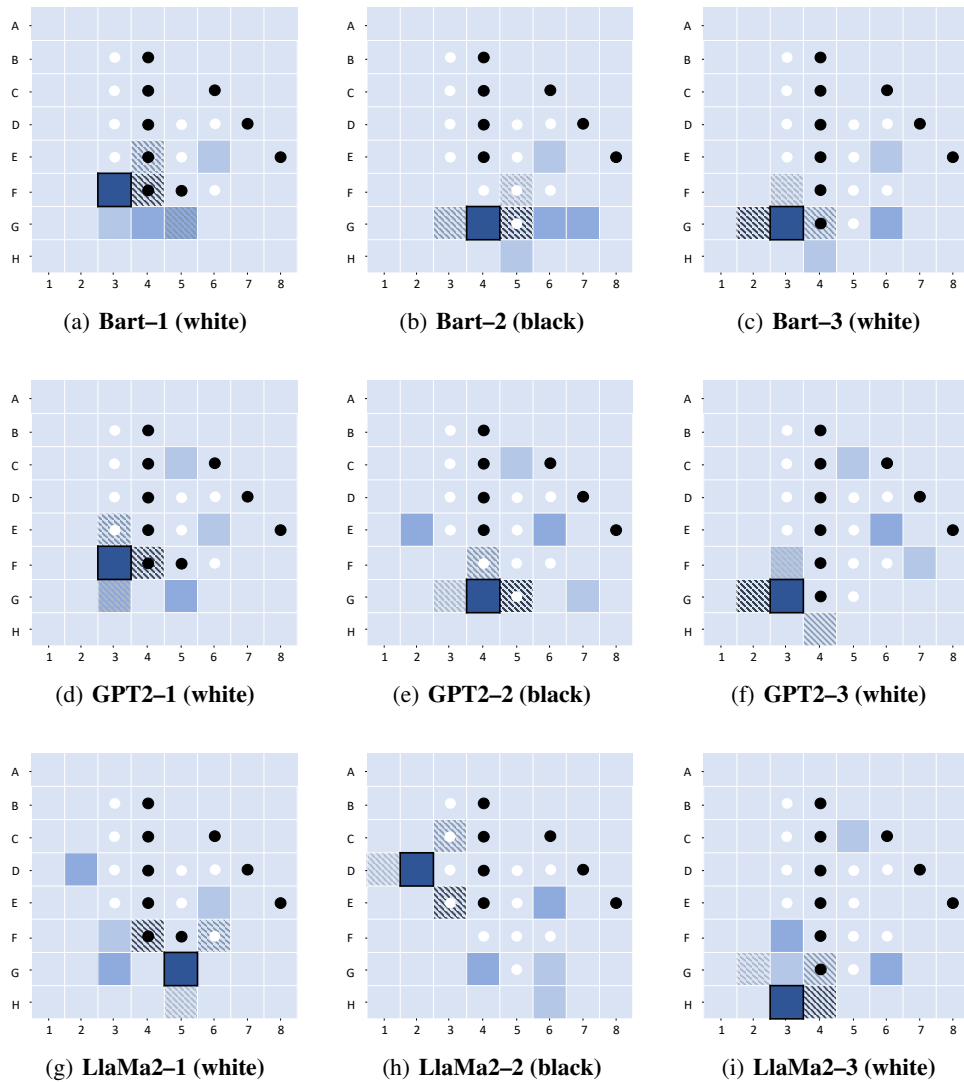


Figure 10: Othello latent move projection from two best performed models.

G LATENT MOVE PROJECTION (FULL VERSION)

We attach the prediction from different models of the same game state in Figure 10. By comparing the performance of different models on the task, we find that overall, Mistral shows the best performance. It consistently demonstrates the best performance across different scenarios, effectively generating legal moves and showing a nuanced understanding of game rules. The Bart model frequently predicts adjacent tiles, leading to numerous failure cases, particularly when trained with smaller datasets. Llama-2 exhibits inconsistent performance, with a tendency to favor certain tile positions or exhibit a bias in move selection. While its predictions are often reasonable, the model appears to lack the robust policy understanding seen in Mistral, especially under constrained training conditions.