What if Othello-Playing Language Models Could See?

Xinyi Chen[*♠] **, Yifei Yuan**[*♡♣]**, Jiaang Li**[♣]**, Serge Belongie**[♣]**,**
Maarten de Rijke[♠]**, Anders Søgaard**[♣]
[♠]University of Amsterdam, [♡]ETH Zürich, [♣]University of Copenhagen
{x.chen2, M.deRijke}@uva.nl, yuanyif@ethz.ch,
{jili, s.belongie, soegaard}@di.ku.dk

Abstract

Language models are often said to face a symbol grounding problem. While some have argued the problem can be solved without resort to other modalities, many have speculated that grounded learning is more efficient. We explore this question in Othello, a simplified, rule-based world that offers a controlled and interpretable testbed for studying world understanding. Building on prior work, we introduce VIS-OTHELLO, a multi-modal model trained jointly on move sequences and board images. Using the Othello rule understanding task, we examine whether multi-modal learning provides advantages over text-only approaches. We further evaluate robustness under semantically irrelevant perturbations and analyze the consistency of cross-modal alignment. Our results suggest that multi-modal training not only improves performance and robustness but also promotes convergence toward shared internal representations across different model architectures.

1 Introduction

Does a language model truly understand what *cat* refers to? Of course, none of us fully know what a cat is in an absolute sense, but human language users know enough to use the word appropriately. We can identify cats in images, infer that the furry, mouse-loving pet someone just described is likely a cat, and use the term in context with ease. Whether mono-modal language models can achieve this level of grounding remains an open question.

This paper does not aim to engage with the discussion over whether symbol grounding is *in principle* impossible for mono-modal language models (Mitchell and Krakauer, 2023; Mollo and Millière, 2023). Instead, we focus on the hypothesis that the inclusion of multiple modalities can facilitate more *efficient* learning. The question is orthogonal, but entirely consistent with the idea that mono-modal language models can induce (a form of) referential semantics (Søgaard, 2023; Huh et al., 2024).

To test this hypothesis, i.e., to what extent multi-modal language models are more sample-efficient, we turn to the task of learning to play Othello with language models, a domain that offers a well-defined, symbolic environment with clear rules and a compact action space, making it an ideal testbed (Li et al., 2023; Hua et al., 2024). Prior work has used this setup to investigate emergent world representations, training models ranging from small-scale language models (Li et al., 2023) to large language models (LLMs) (Yuan and Søgaard, 2025) to predict the next move from prior moves, with performance evaluated by next legal move accuracy to assess rule learning. A probing classifier is trained to investigate the representations learned for intermediate game states (e.g., my move vs. your move) (Nanda et al., 2023). Evidence suggests that language models can learn to track the board state, which potentially forms a rudimentary world model, when trained on large amounts of sequential data.

While Li et al. (2023) showed that text-only models can develop emergent world representations in Othello, our work extends this framework into the multi-modal regime, by introducing **VIS-OTHELLO**, an Othello model trained on sequences of move histories and their corresponding board images (see Figure 1). For each sequence of moves, we generate a corresponding sequence of board state images, with each image depicting the board at a specific time step. We then apply masking strategies to selected move tokens and train the model to predict the missing steps, using both the move history and associated visual context.

Our main goal is to investigate whether access to visual state information enhances sample efficiency and accelerates learning. We break down the main research questions into several related aspects:

[*]Equal contribution.

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 12598–12609
November 4-9, 2025

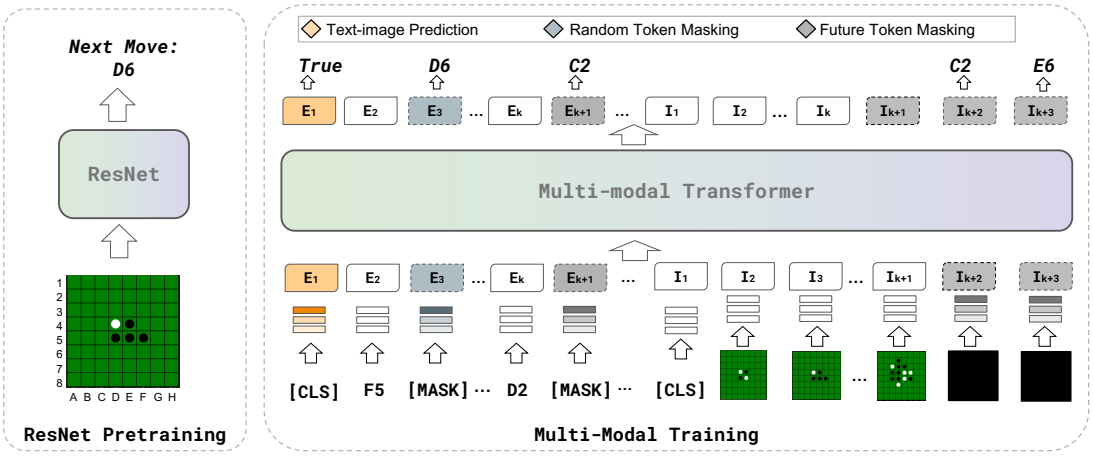


Figure 1: Architecture of VISOTHELLO. The model integrates visual and textual inputs by encoding board images and corresponding move sequences using a Transformer. During pretraining, (i) a ResNet is trained to predict the next move from the current board image; (ii) the multi-modal Transformer is pretrained with three objectives: text-image prediction, random token masking, and future token masking.

Main question	Is multi-modal (Othello) learning faster?
Sub$_1$	Is multi-modal learning better?
Sub$_2$	Is multi-modal grounding better?
Sub$_3$	Do multi- and mono-modal models learn aligned representations?

To address **Sub**$_1$, we compare VISOTHELLO against several baselines on the task of *next move prediction*, where the model predicts the next token given a partial game sequence. We evaluate the performance across varying data scales to assess learning efficiency. For **Sub**$_2$, we perform a semantically irrelevant perturbation analysis by rotating the board image during inference, assessing whether the models trained on original images remain robust and continue to predict legal moves accurately. Regarding **Sub**$_3$, inspired by Lample et al. (2017); Li et al. (2024b), we apply two feature alignment techniques to project intermediate representations from different models into a shared vector space and evaluate their similarity.

We show that multi-modal training improves performance and sample efficiency over text-only training. We also observe that multi-modal models exhibit greater robustness to board rotations. Furthermore, through a feature alignment analysis, we find that representational similarity between models increases with more training data—suggesting that, despite differences in architecture and modality, models can converge on shared internal representations.

Contributions. We are the first to compare the learning curves of mono-modal and multi-modal language models on the task of Othello move prediction and their internal representation learning. We evaluate model robustness by testing invariance to semantically irrelevant perturbations (i.e., board rotations). Additionally, we further study grounding by aligning the internal representations of different models and modalities using supervised and unsupervised methods. Our code is available at https://github.com/shin-ee-chen/multimodal-othello.

2 Related Work

2.1 LLMs for Game Sequence Modeling

Using AI models to play games is not a new concept. Early models, such as AlphaGo, were designed to master gameplay by using predefined game rules and structured environments (Silver et al., 2016, 2017; Feng et al., 2023). Recently, modeling games with LLMs and examining their understanding of game dynamics has become a popular research direction in LLM cognitive probing. Li et al. (2023) train GPT-2 on synthetically generated Othello games, then use probing techniques to determine whether the model develops internal representations of the game state—effectively inferring a world model. Building on this work, Nanda et al. (2023) demonstrate that game-related knowledge is linearly encoded within the model. Following this line, research has expanded the scope of world knowledge acquisition in other scenarios with more advanced probing methods (Hao et al., 2023; Yun et al., 2023; Vafa et al., 2024). For instance, works train similar models with other game datasets, such as chess, maze and checkers, finding that the same encoding patterns hold

in these more complex games (Karvonen, 2024; Spies et al., 2024; Joshi et al., 2024; Karvonen et al., 2024). More relevant to our work, Yuan and Søgaard (2025) extend the study beyond GPT-2, evaluating state-of-the-art LLMs (e.g., LLaMA-2 (Touvron et al., 2023), Qwen (Bai et al., 2023)) to assess their capacity for structured game knowledge representation. Hua et al. (2024) explore this phenomenon in multilingual settings, examining how language models encode and transfer game-related knowledge across different languages. Our work is the first to incorporate visual information in Othello game understanding, providing deeper insights into board state representations.

2.2 Multi-modal Alignment

A growing body of research explores cross-modal alignment as a lens to understand the extent to which language models can internalize and generalize knowledge from text-only inputs (Pereira et al., 2018; Caucheteux et al., 2022; Li et al., 2024a; Ngo and Kim, 2024). Notably, Merullo et al. (2023) demonstrate that visual representations can be effectively projected into the linguistic embedding space using simple linear transformations, revealing a surprising degree of structural compatibility between visual and textual modalities. Building on the theme, Li et al. (2024b) and Huh et al. (2024) argue that as model capacity increases, representations across modalities tend to converge toward a shared, modality-agnostic statistical structure of the world. Unlike prior work focused on aligning visual and linguistic representations of concrete objects, we extend this to abstract game mechanics, enabling deeper insight into how models understand structured environments from text alone.

3 Multi-modal Othello Training

3.1 Training Paradigm

Different from prior works that train Othello models in an autoregressive manner by predicting moves step-by-step, we adopt a **BERT-style masked language modeling** approach for training VISOTHELLO. This avoids the computational overhead and complexity of autoregressive generation, enabling efficient bidirectional reasoning over static visual-text inputs without framing the task as video modeling (for a detailed explanation, see Appendix B). Specifically, we train VISOTHELLO based on VisualBERT (Li et al., 2019).

3.2 Input Representation

Textual input. Following prior works (Li et al., 2023; Karvonen et al., 2024), we represent each game as a sequence of moves, where each move at time step t is treated as a token, denoted as m_t. Our vocabulary consists of 64 unique tokens, corresponding to the 64 tiles on the board. For example, C4 and E6 correspond to the 27th and 45th token in the vocabulary, respectively.

Image input. In addition to the textual input, we provide the model with a sequence of corresponding board images. As demonstrated in Figure 1, each image b_t represents the board state after moves $m_1, m_2, \ldots, m_{t-1}$, and serves as visual context for predicting the next move m_t. To extract visual features, considering the differences between Othello board images and object images in ImageNet (Russakovsky et al., 2015), we pretrain an Othello-specific image encoder using a ResNet-18 backbone (He et al., 2016). We then extract visual features using this encoder, resulting in a visual embedding:

$$\mathbf{v}_t = \phi(\mathbf{b}_t) \in \mathbb{R}^{d_v},$$

where ϕ denotes the image encoder, and d_v is the dimensionality of the visual representation. The image embeddings v_t are treated as image tokens to the input of models and are separated from the text tokens by a special token *[SEP]*.

3.3 VISOTHELLO Training

We train the VISOTHELLO model using two types of masked language modeling (MLM) strategies to enhance its ability to learn meaningful representations of both textual and visual game sequences. MLM enables the model to develop a deeper understanding of game dynamics.

Random token masking. Following the training setup of BERT and VisualBERT, we apply random masking to the move sequence with an 80% probability, masking 15% of the move tokens at random, while keeping the image tokens fully visible. With the random masking task, the model learns to infer missing information using both modalities, reinforcing cross-modal alignment.

Future token masking. To align with the next-move prediction setup used in Othello-GPT (Li et al., 2023), we additionally apply future token masking to the game sequence with a 20% probability. Given a textual move sequence of $m_1, m_2, \ldots,$

m_s, we randomly select a step t ($1 \leq t \leq s$) as the prediction target, and then mask all future tokens from m_t, which are $m_t, m_{t+1}, \ldots, m_s$. To prevent information leakage, we also mask all the image tokens that contains the future move information, which is $v_{t+1}, v_{t+2}, \ldots, v_s$. This task encourages the model to rely on previous move sequence rather than future information when predicting the next move. By masking all future tokens and images beyond a randomly selected time step, the model is trained to make predictions in a uni-directional manner, similar to autoregressive models. This setup reduces dependence on bidirectional context and fits better with the next-move prediction setup in the Othello game.

Text-image prediction. We also adapt the sentence-image prediction task in original VisualBERT training for the Othello task. For a given sequence of image tokens, we replace the corresponding move sequence to a random sequence at a chance of 50%. The model is trained to distinguish whether the text and image sequences are from the same game via binary classification. This task helps to train the model better learn implicit alignments between language and vision.

Training objective. The overall training objective $\mathcal{L}_{\text{total}}$ is defined as the sum of the masked modeling loss and the text-image prediction loss:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{mask}} + \mathcal{L}_{\text{ti}},$$

where the masked modeling loss $\mathcal{L}_{\text{mask}}$ combines two components:

$$\mathcal{L}_{\text{mask}} = \alpha \cdot \mathcal{L}_{\text{random}} + (1 - \alpha) \cdot \mathcal{L}_{\text{future}}.$$

Here, $\mathcal{L}_{\text{random}}$ is the random token masking loss, $\mathcal{L}_{\text{future}}$ is the future token masking loss, $\mathcal{L}_{\text{ti}}$ is the text-image prediction loss, and α is set to 0.8.

4 Experiments

In this section, we evaluate whether incorporating visual information improves learning efficiency and grounding in the Othello game setting.

4.1 Experimental Setups

Compared models. To better assess the impact of multi-modal learning, we include text-only and vision-only baselines for direct comparison.

Split	Games	Images	Avg. per Game
Train	20,525	1,247,852	60.8
Validation	1,282	78,141	60.9
Test	3,850	233,975	60.8
Total	25,657	1,559,968	60.8

Table 1: Dataset statistics. The number of games and images per split. Each game comprises a sequence of steps, with one image per step.

Text-only models. We evaluate two text-only models with different architectures. (i) Othello-GPT, introduced by Li et al. (2023), is based on GPT-2 and trained autoregressively on Othello move sequences to predict the next move in a purely textual setting. (ii) BERT (Devlin et al., 2019) is trained using the same language learning objectives as VISOTHELLO, including both random token masking and future token masking. As BERT serves as the language backbone of our multi-modal model, it provides a strong baseline for isolating the contribution of visual information in learning Othello strategies.

Vision-only models. As a vision-only baseline, we train a ResNet-18 model (He et al., 2016) on board images. Unlike VISOTHELLO, which processes a sequence of board images and move tokens, the ResNet model is trained to predict the next move based solely on a single board image representing the current game state. It does not observe any move history or future states.

Datasets. We collect a total of 25,657 real game records from the EOTHELLO website,[1] which serve as the textual sequence inputs for our dataset. For each recorded step in a game, we generate a corresponding image to capture its visual state. As summarized in Table 1, this process yields approximately 1.56 million images, averaging around 60.8 images per game across all splits. We split the dataset into training (80%), validation (5%), and test (15%) sets, while maintaining a consistent number of images per game across all splits.

Evaluation metrics. When evaluating VISOTHELLO, we adopt the same setup as the future token masking objective following Li et al. (2023): to predict the legal move at step m_t, we mask all future tokens starting from m_t, and all image embeddings from v_{t+1} onward, to prevent information leakage from future states. This setup assesses whether a model can learn the underlying rules of

[1] https://www.eothello.com/

Train Size	0	1k	3k	5k	10k	20k
Othello-GPT	7.3±0.0	20.8±2.8	62.1±15.1	66.8±18.5	70.1±19.8	79.7±2.6
BERT-S	17.1±2.7	89.5±0.9	90.5±0.5	90.8±0.4	91.8±0.1	92.9±0.1
ResNET-18-S	21.1±12.5	62.4±2.3	70.9±3.3	74.0±1.7	82.4±1.4	87.2±2.7
VISOTHELLO-S	14.4±11.3	**91.3±0.5**	**92.9±0.9**	**93.6±0.3**	**93.8±0.3**	93.8±0.3
BERT-P	16.7±2.7	88.9±1.0	90.6±0.7	91.4±0.6	91.7±0.3	92.4±0.5
ResNET-18-P	**26.3±21.8**	73.1±1.6	84.9±0.3	89.2±1.6	91.7±0.9	92.7±0.5
VISOTHELLO-P	25.0±16.0	91.2±0.5	92.3±0.2	93.4±0.5	93.7±0.3	**93.9±0.2**

Table 2: Legal move accuracy (%) for next move prediction in different models across different data sizes. We report mean ± std over 3 runs, and highlight the best performing model of each training set size in bold. $-P$ indicates the model is pretrained, while $-S$ indicates it is trained from scratch.

Train Size	0	1k	3k	5k	10k	20k
Othello-GPT	0.4±0.0	1.9±0.2	16.1±18.3	16.7±19.6	16.8±17.2	27.6±16.6
BERT-S	1.3±0.2	20.5±0.2	26.2±1.1	29.1±1.1	33.8±0.3	**39.3±0.3**
ResNet-18-S	**1.6±0.0**	10.9±0.3	14.0±0.3	15.4±0.4	18.9±0.5	21.3±0.2
VISOTHELLO-S	1.0±0.7	**27.2±0.3**	**29.7±0.6**	**31.6±1.1**	33.5±0.7	36.7±0.6
BERT-P	1.3±0.4	26.1±0.2	27.7±0.2	31.3±1.2	**34.3±0.0**	37.4±2.1
ResNet-18-P	1.5±0.2	12.9±0.5	19.5±0.8	22.1±1.6	24.8±1.7	26.7±0.1
VISOTHELLO-P	1.0±0.4	26.1±0.2	29.3±0.2	30.6±1.2	33.4±0.3	35.8±1.3

Table 3: Exact match accuracy (%) for next move prediction in different models across different data sizes. For Othello-GPT, BERT, ResNet, and VISOTHELLO we report mean ± std over 3 runs.

Othello game from sequential move data, in contrast to AlphaZero (Silver et al., 2017), which focus on the winning strategy. Accordingly, we use **legal move accuracy** as our evaluation metrics. Specifically, we evaluate whether the predicted move m_t, given the move history $m_1, m_2, \ldots, m_{t-1}$, is valid under Othello's rules, respectively. We also report **exact match accuracy**, which measures whether the predicted move m_t exactly matches the ground-truth move, reflecting the model's ability to replicate expert gameplay.

Training details. To assess the learning efficiency of mono-modal and multi-modal models, we train all models on the full dataset (20k samples) as well as on randomly sampled subsets of 1k, 3k, 5k, and 10k examples. Training and evaluation are conducted with three random seeds (5, 12, and 42). For VISOTHELLO, we use the best performing image encoder, ResNet-18 pretrained and fine-tuned on the full 20k dataset, for feature extraction. We also investigate the impact of pretraining by training each model either from scratch or from publicly available pretrained weights.[2] For training details, see Appendix C.

[2]For BERT, we use `google-bert/bert-large-uncased`; the pretrained ResNet-18 is from the HuggingFace transformers library; the pretrained VisualBERT weights are from the Volta framework.

4.2 Experimental Results

Table 2 and 3 report the exact match and next legal move prediction accuracy of various models across different dataset sizes. Several key observations emerge from these results.

Multi-modal learning is more sample efficient. VISOTHELLO achieves high accuracy (over 91% in next legal move prediction) with as few as 1k training examples, while uni-modal models either require more data or fail to reach the same performance ceiling. This suggests that multi-modal learning is more sample-efficient—VISOTHELLO shows stronger performance at smaller scales than uni-modal baselines. This observation aligns with previous work from Zhuang et al. (2024).

Pretraining information is not consistently helpful. While pretraining improves performance in some cases, especially in low-data regimes, its effect is not consistent across modalities. With ResNet-18, pretraining is very helpful at small sizes, but its effect decreases as the dataset gets larger. In contrast, for both BERT and VISOTHELLO, pretraining does not consistently lead to significant gains across training sizes. This observation is consistent with prior findings by Yuan and Søgaard (2025), which suggest that linguistic pretraining may offer limited benefit for structured, rule-based environments such as Othello.

Method	Legal Move Accuracy
VISOTHELLO	94.03
Pooling	92.43
Area	91.80
W/O FT ResNet	92.04
W/O FTM	62.03

Table 4: Ablation results for VISOTHELLO, using different image encoders (Pooling, Area, ResNet without fine-tuning) and without future token masking (W/O FTM). All results are reported from the best validation checkpoints with training seed 42.

4.3 Ablation Study

To evaluate the contribution of the modified components in VISOTHELLO relative to the original VisualBERT, we conduct ablation studies focusing on the image encoder and the future token masking strategy.

Image encoder. We test whether fine-tuning a ResNet model is necessary for extracting image features. For comparison, we consider three alternatives that do not involve task-specific adaptation: (a) a simple pooling projection that downsamples the raw $600 \times 600 \times 3$ image into a 1200-dimensional embedding, (b) an *Area* projection that partitions the image into patches, averages pixel values within each patch, and flattens the resized image into a 1200-dimensional vector, and (c) a ResNet-18 encoder without fine-tuning on Othello images.

Future token masking. To test the necessity of future token masking (FTM), we train VISOTHELLO without this component (denoted as *W/O FTM*) and compare performance with the full model.

Results. As shown in Table 4, both the choice of image encoder and the use of FTM substantially affect performance. Replacing the fine-tuned ResNet with simple Pooling or Area projections reduces legal move accuracy from 94.03% to 92.43% and 91.80%, respectively, while using an unfine-tuned ResNet yields 92.04%. These results highlight the benefit of domain-specific adaptation for the visual encoder. More strikingly, removing FTM causes performance to collapse to 62.03%, underscoring its critical role in aligning the training objective with the causal structure of the game.

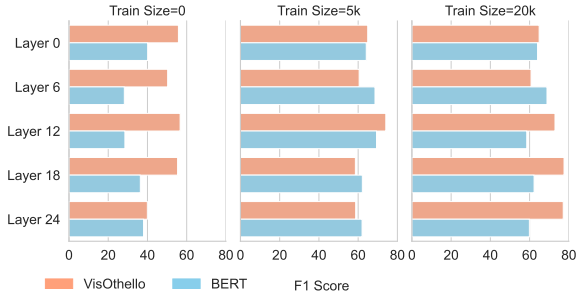


Figure 2: Illustration of probing results for BERT and VISOTHELLO trained with different dataset set sizes.

4.4 Probing Internal Representations

To assess whether VISOTHELLO learns meaningful internal representations of the board state, we train a linear probe to predict the state of each tile—i.e., whether it is empty, contains the player's disc, or the opponent's disc—based on hidden activations after processing a move sequence, following the approach of Nanda et al. (2023).

Probe training. The linear probe is trained on 5,000 samples from the training set and evaluated on the same test set used in previous experiments. Due to architectural differences, we restrict this analysis to VISOTHELLO and the BERT baseline. We evaluate both models after training on 0, 5k, and 20k examples, while keeping the probing setup fixed across all conditions.

Results. Figure 2 shows F1 scores from linear probes trained to predict tile-level board states from selected layers of BERT and VISOTHELLO. When models are randomly initialized (0 examples), VISOTHELLO already encodes more board-relevant structure than BERT, achieving substantially higher probe performance in early layers. This may be attributed to the use of a ResNet encoder pretrained on Othello board images, which already encodes useful spatial structure. As training dataset increases, both models improve, but VISOTHELLO consistently achieves higher scores—especially in deeper layers. After training on 20k examples, VISOTHELLO reaches 77.55 F1 at Layer 18, compared to 62.28 for BERT. This suggests that VISOTHELLO learns more accurate internal representations of the board state, benefiting from both multimodal input and architectural modifications.

5 Semantically Irrelevant Perturbation

To evaluate model robustness and generalization, we test performance under semantically irrelevant perturbations—input transformations that alter sur-

face form without changing game state.

5.1 Board Rotation

We focus on board rotation as a concrete instance, which each test board is rotated 180 degrees. As illustrated in Figure 3, this transformation corresponds to a spatial inversion: for image-based models, this involves rotating the game board in the input image; for language-based models, it requires flipping the row and column indices of the move representation (e.g., D3 becomes E6, resulting in a different move token ID).

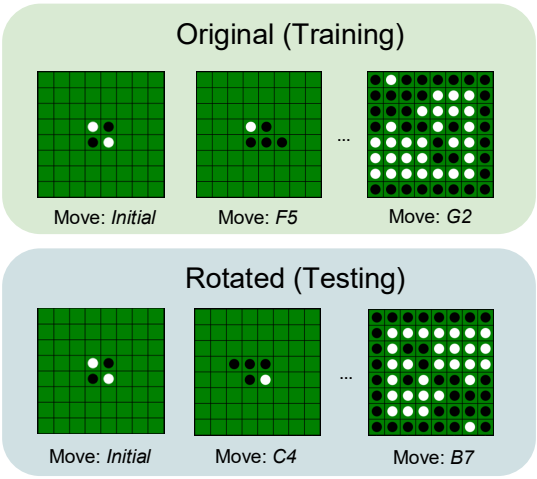


Figure 3: Illustration of Rotation 180°. A 180° rotation preserves game dynamics due to the board's inherent symmetry and the uniformity of move rules, making such transformations invariant under play.

We apply the rotation **only at test time**, evaluating models that were trained on the original (unrotated) training data. All models are assessed on their ability to predict both the next legal move and the exact next move, as described in Section 4. This setup allows us to examine whether models rely on absolute visual or positional cues, or whether they have learned more abstract, generalizable representations of the board state.

As shown in Figure 4, BERT remains relatively robust under board rotation, with accuracy of 90–93% across all settings. This stability is expected: since BERT operates on symbolic move sequences, board rotation can be handled through a deterministic remapping of move tokens (e.g., D3 becomes E6). In contrast, ResNet-18 suffers a substantial drop in accuracy under rotation, falling to 28–35% depending on training size and pretraining. This suggests the model fails to learn rotation-invariant representations and relies heavily on absolute spatial patterns. Lacking access to move history or turn information, ResNet depends on ambiguous visual cues that can be easily disrupted—highlighting a key limitation of purely visual models in game playing tasks like Othello.

VISOTHELLO, which combines ResNet's image features with BERT's move sequence encoding, maintains high accuracy after rotation (91–93%). Compared with ResNet, VISOTHELLO receives explicit sequence information, including the player turn and previous moves, which helps disambiguate the rotated board. The language modality further guides the interpretation of visual features, enabling the model to maintain stable predictions under spatial transformations. This result illustrates the strength of multi-modal grounding: *by aligning perceptual input with symbolic context,* VISOTHELLO *overcomes the spatial brittleness seen in purely visual models.*

6 Feature Alignment

We perform representation alignment across models trained on Othello game sequences to assess whether models trained on different modalities (i.e., image and text) learn similar representations. Through this, we investigate whether modality-specific models encode analogous patterns that are fundamental to rule-following gameplay.

6.1 Alignment Method

We extract intermediate representations, denoted as H_i, from different models for alignment, using the same input sequence, and corresponding board images for multi-modal models. Specifically, we use the features extracted from final hidden layer of both encoder-only models (e.g., BERT, VISOTHELLO) and decoder-only models (e.g., Othello-GPT). Given the learned representations H_1 and H_2 of dimensions d_1 and d_2, respectively, extracted from models M_1 and M_2 based on the same game sequence input, we first apply PCA to project them into a shared-space of dimension $d = \min(d_1, d_2)$:

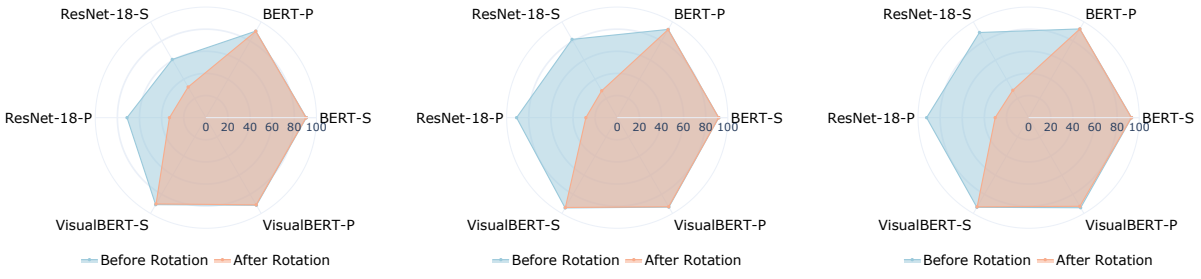
$$H_1' = P_d(H_1), H_2' = P_d(H_2), \quad (1)$$

where $H_1', H_2' \in \mathbb{R}^d$ are projected vectors.

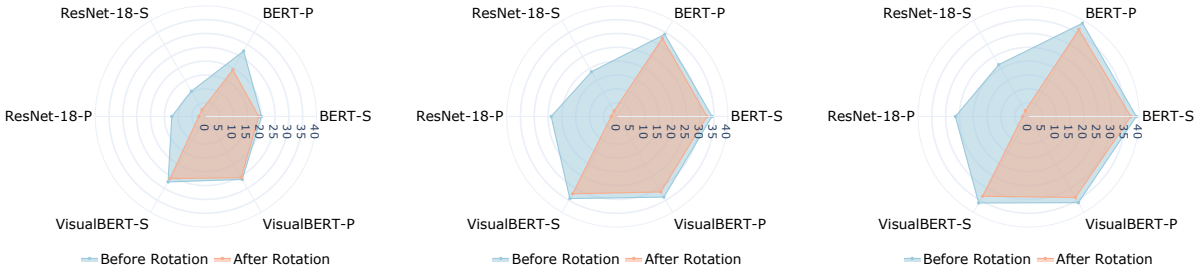
Next, we align these representations into a common vector space using the MUSE package,[3] originally developed for mapping multilingual word embeddings into a shared space. The aim is to learn a linear mapping matrix W, for each projected representation H_1' and H_2'

$$W^* = \underset{W \in \mathcal{M}_i(\mathbb{R})}{\arg\min} \|H_i' W - H_j'\|, \quad (2)$$

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



(a) Legal move acc. with 1K samples (b) Legal move acc. with 10K samples (c) Legal move acc. with 20K samples



(d) Exact match acc. with 1K samples (e) Exact match acc. with 10K samples (f) Exact match acc. with 20K samples

Figure 4: Comparison of models' performance with and without board rotation across different training dataset sizes. The results demonstrate that multi-modal models maintain better performance under rotation compared to purely visual models. $-P$ indicates the model is pretrained, while $-S$ indicates it is trained from scratch.

where $i, j \in \{1, 2\}$ and $i \neq j$. This denotes learning the optimal linear mapping matrix W^* that aligns representation H'_i to H'_j.

6.2 Alignment Training

To obtain the optimal mapping matrix, we use both supervised and unsupervised training methods.

Supervised training. We treat representations from different models (e.g., Othello-GPT and VISOTHELLO) corresponding to the same game sequence as paired training data. For example, given the Othello move sequence input "*F5 F6 E6 F4 C3 D7*", the pairwise training input H'_1 and H'_2 correspond to the representations extracted from Othello-GPT and VISOTHELLO models, respectively, for this exact sequence and the associated images (when applicable). The mapping matrix W is learned and optimized with iterative Procrustes alignment (Gower and Dijksterhuis, 2004), which alternates between solving for the optimal orthogonal transformation and refining the mapping. This process minimizes the distance between the transformed source representations and the target representations, resulting in better alignment across the two vector spaces.

Unsupervised training. We also adopt the unsupervised training approach (Conneau et al., 2018; Lample et al., 2017) with the absence of paired data or predefined anchors to learn the alignment. Given a set of game features H' from both the source and target space, the process begins with adversarial training, where a discriminator is trained to distinguish whether the feature comes from the source or target representation space. Simultaneously, the mapping matrix W is optimized to make this distinction harder, effectively aligning the distributions. Once an initial mapping is obtained, we apply iterative Procrustes refinement, similar to the supervised setting, to improve the alignment. Alignment quality is evaluated and improved using the average cosine similarity between mapped source and target features on the test set.

6.3 Alignment Training Setups

To construct the alignment training set, we randomly sample one subsequence from each complete game, resulting in 3,849 input sequences, each paired with the corresponding board state images. We then divide the data into training and testing sets with an 80%/20% split, resulting in 3,079 and 770 instances. We adopt cosine similarity to measure the alignment quality between representations from different models. After projecting the representations into a shared space, we compute the average pairwise cosine similarity between aligned feature vectors. A higher similarity score indicates better alignment, suggesting that the models, despite being trained on different modalities, capture similar underlying patterns. We train the alignment

Source	Target	0	1k	3k	5k	10k	20k
BERT	ResNet	25.37	27.39	29.48	32.07	33.34	34.25
BERT	Othello-GPT	**86.01**	62.02	56.59	57.14	61.60	63.30
BERT	VISOTHELLO	83.92	61.76	54.83	53.96	55.59	57.94
Othello-GPT	ResNet	32.16	31.79	29.95	34.33	33.74	36.41
Othello-GPT	VISOTHELLO	83.09	**77.68**	**81.62**	**76.56**	**82.07**	**82.35**
VISOTHELLO	ResNet	11.62	26.03	29.04	33.83	32.49	38.99

Table 5: Supervised alignment similarity between target and source models. Highest in bold.

Source	Target	0	1k	3k	5k	10k	20k
BERT	ResNet	31.53	37.46	36.96	36.98	38.70	40.25
BERT	Othello-GPT	**90.38**	65.61	62.26	57.68	61.01	63.29
BERT	VISOTHELLO	90.52	67.52	62.23	58.97	63.60	63.77
Othello-GPT	ResNet	33.94	46.43	44.55	50.15	46.96	47.89
Othello-GPT	VISOTHELLO	87.20	**80.50**	**80.53**	**79.27**	**85.81**	**82.46**
VISOTHELLO	ResNet	23.04	43.44	44.39	45.38	52.64	57.79

Table 6: Unsupervised alignment similarity between target and source models. Highest in bold.

model using a single NVIDIA A100 GPU. All hyperparameters follow the default settings provided by the original MUSE implementation, with no additional tuning. All models in this experiment are trained with a fixed random seed 42.

6.4 Mapping Result

Table 5 and 6 demonstrate the mapping results under supervised and unsupervised training. We find that the alignment similarity generally improves as the size of the training data increases. This trend suggests that with more data, the models learn richer and shared representations that are easier to align across modalities. Also, despite the difference in training strategy (i.e., autoregressive training and mask language modeling), Othello-GPT and BERT exhibit strong alignment, reflected in their high similarity scores. Surprisingly, Othello-GPT exhibits a strong alignment score with VISOTHELLO, indicating that despite differences in architecture and training modalities, the two models learn remarkably similar representations. This suggests that the underlying patterns essential for Othello gameplay are captured consistently across both language-based and multi-modal models. Such alignment highlights the potential for cross-modal knowledge transfer and opens avenues for further exploration of unified representations in complex tasks.

7 Conclusion

We studied the task of learning to play Othello and extended it to a multi-modal setting by introducing VISOTHELLO. Our experiments examined whether access to visual state information improves sample efficiency and accelerates learning, comparing VISOTHELLO against text-only and vision-only baselines. To further assess the benefits of multi-modal grounding, we introduce a board rotation perturbation and conduct feature alignment analysis to evaluate whether the models learn more robust and aligned representations. Our findings suggest that grounding language models with visual input leads to more efficient and stable learning. Beyond Othello, our framework provides a controlled testbed for the analysis of grounded representations and has the potential to extend to other model architectures, tasks, and modalities (see Appendix A for further discussion).

Limitations

A notable limitation of this work is that we are not able to compare VISOTHELLO with autoregressive multi-modal large language models (MLLMs) due to fundamental differences in training paradigms. Autoregressive MLLMs treat images as part of a sequential token stream, effectively converting static visual-text inputs into video modeling tasks, which significantly increases computational complexity and alters the problem structure. In contrast, our model uses masked language modeling (MLM) to enable efficient bidirectional reasoning over static data, making direct comparison with autoregressive MLLMs infeasible without substantial task reformulation.

Moreover, we do not include comparisons with large-scale text-only language models, as these have been thoroughly investigated in prior work (Yuan and Søgaard, 2025). Given that pretraining on language alone does not necessarily enhance understanding of the structured reasoning inherent in Othello, scaling up to such models and benchmarking against them is not currently a priority. Instead, our use of lightweight language

models offers a practical and efficient probe into how much language pretraining contributes to this domain.

Ethics Statement

We ensure that all datasets used in this work are publicly available and released under appropriate open-source licenses. No personal information about players or tournaments is included or revealed. Additionally, all corresponding images used in our experiments are synthetically generated, and do not depict real individuals or contain sensitive content.

Acknowledgments

We thank the reviewers and area chair for their suggestions to incorporate the world model, which helped clarify our contributions better. We are also grateful to the members of the CoAStaL group at the University of Copenhagen and the IRLab at the University of Amsterdam for their valuable feedback on the experiments. Xinyi Chen is funded by the project LESSEN (NWA.1389.20.183) of the research program NWA-ORC 2020/21, which is (partly) financed by the Dutch Research Council (NWO), as well as by travel support from ELIAS (GA No.101120237) through the ELLIS program. Serge Belongie and Jiaang Li are supported by the Pioneer Centre for AI, DNRF grant number P1. Maarten de Rijke was supported by the Dutch Research Council (NWO), under project numbers 024.004.022, NWA.1389.20.183, and KICH3.LTP.20.006, and the European Union under grant agreements No. 101070212 (FINDHR) and No. 101201510 (UNITE). Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of their respective employers, funders and/or granting authorities.

References

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, and 1 others. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Charlotte Caucheteux, Alexandre Gramfort, and Jean-Rémi King. 2022. Deep language algorithms predict semantic comprehension from brain activity. *Scientific Reports*, 12(1):16327.

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*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (long and short papers)*, pages 4171–4186.

Xidong Feng, Yicheng Luo, Ziyan Wang, Hongrui Tang, Mengyue Yang, Kun Shao, David Mguni, Yali Du, and Jun Wang. 2023. Chessgpt: Bridging policy learning and language modeling. *Advances in Neural Information Processing Systems*, 36:7216–7262.

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 preprint arXiv:2305.14992*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778.

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

Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. 2024. Position: The Platonic representation hypothesis. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 20617–20642. PMLR.

Abhinav Joshi, Vaibhav Sharma, and Ashutosh Modi. 2024. CheckersGPT: Learning world models through language modeling. In *Annual Meeting of the Association for Computational Linguistics*.

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

Adam Karvonen, Benjamin Wright, Can Rager, Rico Angell, Jannik Brinkmann, Logan Smith, Claudio Mayrink Verdun, David Bau, and Samuel Marks. 2024. Measuring progress in dictionary learning for language model interpretability with board game models. *Advances in Neural Information Processing Systems*, 37:83091–83118.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2017. Unsupervised machine translation using monolingual corpora only. *arXiv preprint arXiv:1711.00043*.

Jiaang Li, Antonia Karamolegkou, Yova Kementchedjhieva, Mostafa Abdou, and Anders Søgaard. 2024a. Structural similarities between language models and neural response measurements. In *Proceedings of the 2nd NeurIPS Workshop on Symmetry and Geometry in Neural Representations*, volume 228 of *Proceedings of Machine Learning Research*, pages 346–365. PMLR.

Jiaang Li, Yova Kementchedjhieva, Constanza Fierro, and Anders Søgaard. 2024b. Do vision and language models share concepts? A vector space alignment study. *Transactions of the Association for Computational Linguistics*, 12:1232–1249.

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. *ICLR*.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2019. VisualBERT: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*.

Jack Merullo, Louis Castricato, Carsten Eickhoff, and Ellie Pavlick. 2023. Linearly mapping from image to text space. In *ICLR*.

Melanie Mitchell and David C. Krakauer. 2023. The debate over understanding in AI's large language models. *Proceedings of the National Academy of Sciences*, 120(13):e2215907120.

Dimitri Coelho Mollo and Raphaël Millière. 2023. The vector grounding problem. *arXiv preprint arXiv:2304.01481*.

Neel Nanda, Andrew Lee, and Martin Wattenberg. 2023. Emergent linear representations in world models of self-supervised sequence models. *arXiv preprint arXiv:2309.00941*.

Jerry Ngo and Yoon Kim. 2024. What do language models hear? Probing for auditory representations in language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5435–5448, Bangkok, Thailand. Association for Computational Linguistics.

Francisco Pereira, Bin Lou, Brianna Pritchett, Samuel Ritter, Samuel J. Gershman, Nancy Kanwisher, Matthew Botvinick, and Evelina Fedorenko. 2018. Toward a universal decoder of linguistic meaning from brain activation. *Nature Communications*, 9(1):963.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252.

David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, and 1 others. 2016. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, and 1 others. 2017. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*.

Anders Søgaard. 2023. Grounding the vector space of an octopus: Word meaning from raw text. *Minds and Machines*, 33(1):33–54.

Alex F. Spies, William Edwards, Michael I. Ivanitskiy, Adrians Skapars, Tilman Rauker, Katsumi Inoue, Alessandra Russo, and Murray Shanahan. 2024. Transformers use causal world models in maze-solving tasks. *arXiv preprint arXiv:2412.11867*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Keyon Vafa, Justin Chen, Ashesh Rambachan, Jon Kleinberg, and Sendhil Mullainathan. 2024. Evaluating the world model implicit in a generative model. *Advances in Neural Information Processing Systems*, 37:26941–26975.

Yifei Yuan and Anders Søgaard. 2025. Revisiting the Othello world model hypothesis. *arXiv preprint arXiv:2503.04421*.

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. *arXiv preprint arXiv:2311.02171*.

Chengxu Zhuang, Evelina Fedorenko, and Jacob Andreas. 2024. Visual grounding helps learn word meanings in low-data regimes. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1311–1329.

A Potential Impacts

Our multi-modal Othello framework demonstrates how integrating visual and textual modalities can enhance structured reasoning in environments with strict rule-based dynamics. Beyond board games, this approach offers insights into multi-modal learning for tasks requiring spatial-temporal understanding, such as strategy modeling, robotics, and educational AI systems. By disentangling perceptual and symbolic reasoning, it also serves as a testbed for evaluating how models learn abstract rules from multi-modal input, potentially informing the design of more robust, interpretable, and generalizable multi-modal AI systems. Future work may generalize these insights to more complex domains and explore the role of other modalities, such as spatial or tactile input, in supporting the emergence of grounded representations.

While Othello is a harmless testbed, our methods for aligning multimodal features could, in principle, be adapted to sensitive domains. Our findings on data efficiency may also lower the barrier to training multimodal agents in low-resource settings. We emphasize that our work is intended solely for research on interpretability and grounding.

B Model Design Motivation

We use a masked language model (MLM) rather than an autoregressive multi-modal large language model (MLLM) for two main reasons detailed below.

Autoregressive training paradigm is not well-suited for our task setup. Othello involves dynamic visual changes, as discs flip after each move (Figure 1). Understanding the current board state requires access to the complete move history, as it cannot be inferred from a single image alone—even for human players. Thus, the input must consist of a sequence of move tokens paired with the corresponding sequence of board states. This token-aligned multimodal sequence deviates significantly from standard MLLM training paradigms, which are typically designed for single image–text pairs or interleaved inputs without sequential dependencies. A more suitable framing is to model the game as a video sequence, with each board as a frame. However, feeding full sequences into current MLLMs introduces the risk of information leakage from future states and would require specialized causal multimodal masking, implying non-trivial architectural and training modifications.

Autoregressive training incurs extremely high resource costs. Even if the technical challenges above were addressed, autoregressive training would remain computationally demanding: modeling an n-step game requires n forward passes with progressively longer input sequences, whereas our MLM objective learns from the entire game in a single pass with partial masking. Our experiments were conducted on a single A100 GPU (40GB) with a dataset of approximately 20k Othello games, a scale that makes MLLM training infeasible.

Since the goal of this paper is to investigate the role of images in model understanding, we adopt an MLM-based approach with VisualBERT rather than MLLMs. This choice provides a lightweight framework for probing and analyzing the representations learned in multimodal Othello training.

C Model Training Details

All models are trained for up to 1000 epochs, with validation performed every 10 epochs. We apply early stopping with a patience of 5 validation steps, and retain the checkpoint with the highest validation accuracy for final evaluation. Training is conducted on a single NVIDIA A100-40GB GPU. BERT and VISOTHELLO are trained with a batch size of 128 and a learning rate of 1e-4, while ResNet is trained with a batch size of 512 using the same learning rate.

D Model Sizes And Compute Resources

We report the parameter sizes and compute resources for all models used in our experiments. VISOTHELLO, based on VisualBERT-base, has about 112M parameters. BERT (25 layers, hidden size 768) has about 177M parameters. Othello-GPT, based on GPT-2 Medium, contains 345M parameters. ResNet-18 has about 11.7M parameters. All models were trained on a single NVIDIA A100-40GB GPU. Training on 20k tasks required approximately 10 hours for VISOTHELLO, 0.5 hours for ResNet-18, and 3.5 hours for BERT