Predicting Weak-to-Strong Generalization from Latent Representations

Anonymous Author(s)
Affiliation
Address
email

Abstract

AI alignment seeks to align models with human values, yet humans may be unable to evaluate whether a model is aligned on tasks exceeding human capabilities. Weak-to-strong generalization (WSG) has been proposed as a proxy for studying this problem, where a weaker model stands in for human evaluation of a stronger model. While prior work provides evidence of WSG success, it suffers from train-test contamination or relies on oversimplified linear models. We introduce a clean testbed where transformer model pairs are pretrained on different variants of Othello and Tic-Tac-Toe, then the stronger model is finetuned on data from the weaker model. Using mechanistic interpretability techniques, we demonstrate that the stronger model outperforms the weaker model if and only if it has better board representations. Across 111 WSG pairs and 6 game rules, we find a 0.844 Spearman correlation between WSG success and superior board representations in the strong model as measured by linear probes. By open-sourcing our code, models and probes, we hope to accelerate research on interpreting WSG.

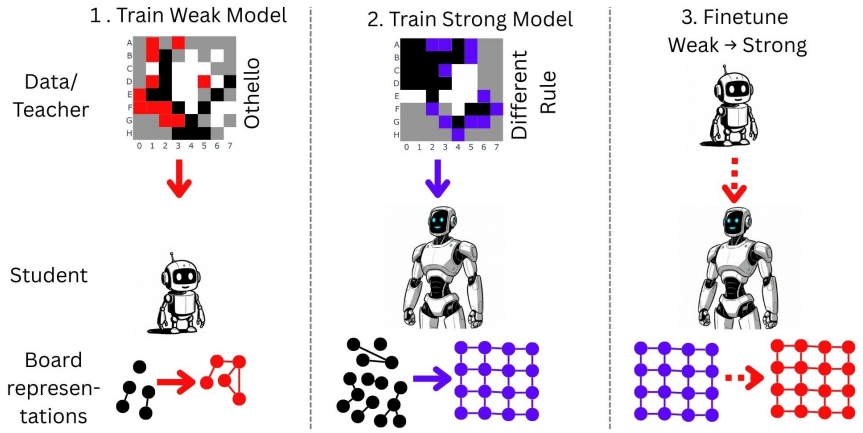


Figure 1: **Overview of our method:** Each column is one step in our pipeline. The top row is the training signal, the middle row the model that gets trained and the bottom row displays how features change during training (structure represents high-quality features, and color the goal). 1. First we train a weak transformer to play standard Othello (left, red). It learns basic representations around the board state. 2. Secondly, we train a stronger transformer to play under a different ruleset (middle, blue) leading to a better world model of the Othello board. 3. Lastly, we use the weak model to finetune the strong model (right) and can observe that the strong model performs better in standard Othello than the weak model originally did. We show that this occurs if and only if the strong model's board representations are better than those of the weak model. An interpretation is that the strong model learns from the weak model how to utilize its superior features to play standard Othello.

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

1 Introduction

Current alignment techniques such as RLHF (Christiano et al. 2023) rely on the ability of humans to evaluate AI, i.e. we have a strong teacher (human) that supervises a weaker student (AI). If AI performance surpasses human performance on a task, the direction of supervision stays the same, but the dynamic inverts: A weak teacher (human) has to convey the intended objective to a stronger student (AI). To understand these dynamics, Burns et al. (2023) proposed to study an analogous setup where we train a *weak model* and a *strong model* on two different tasks (called *weak-* and *strong-task*). Then the weak model finetunes the strong model. It was shown on a variety of tasks that **the strong model can surpass the performance of its weaker teacher on the weak task. This phenomenon is called Weak-to-Strong Generalization (WSG)** (Burns et al. 2023; Y. Guo and Yi Yang 2024; J. Guo et al. 2024; Somerstep et al. 2024). The common hypothesis is that the strong model learns from the weak model how to utilize its superior capabilities (Burns et al. 2023; Shin et al. 2024). If this hypothesis is true, it should be possible to predict whether the strong model can surpass its weak-teacher by comparing "useful" features between both models.

WSG has been studied both empirically and theoretically. Burns et al. (2023) finetuned GPT4-base using a human-finetuned GPT2-sized model and recovered roughly GPT-3.5 level performance on binary NLP classification tasks. One limitation of this class of experiments is that both the task-evaluation and pretraining of the strong model were conducted on human data while future AI won't have examples of superhuman aligned behavior that it can elicit. On specific learning tasks, such as linear models over Gaussian features, it has been theoretically shown that WSG works if the strong model has features that are useful for the weak task (Wu and Sahai 2024; Shin et al. 2024), but do not help with learning errors of the weak model (Xue et al. 2025; Hunter Lang et al. 2024). There has been no prior analysis of what these features actually are. Previous mechanistic interpretability work (Elhage, Nanda, et al. 2021) has analyzed the internal features of models, but not around WSG. E.g. it has been shown that a transformer, which plays Othello through next-token-prediction, internally represents and uses the full board state (Nanda, Lee, et al. 2023; K. Li et al. 2024). We use these results to shed new light on this.

We first show that in Othello and Tic-Tac-Toe by training models on different rules WSG can be achieved. We then investigate WSG with mechanistic interpretability tools in the context of Othello. We train a weak transformer to play legal moves in Othello and a strong transformer to play under a modified ruleset. After finetuning the strong model using the output of the weak model, the strong model plays legal moves more reliably than the weak model. We demonstrate this for multiple strong rules and additionally apply the same idea to Tic-Tac-Toe, resulting in the smallest transformer setup with shown WSG. These environments have no pre-train leakage, and we can build on (Nanda, Lee, et al. 2023; K. Li et al. 2024) which showed that in Othello transformers represent the board state as linear directions in the residual stream. We take the accuracy of linear probes on the weak and strong models as a proxy for the strength of board representations. Empirically, the weak-supervision succeeds almost if and only if the strong model has better board representations than the weak model.

For 6 different strong rules, we finetune a total of 111 pairs of weak and strong models. We then show that the prediction rule

"WSG occurs if and only if the strong model has better board representations than the weak model."

has a 93% accuracy. We initially established this result using two rules and subsequently confirmed it with an additional four. For each pair we compute the Performance-Gap-Recovered (PGR) (Burns et al. 2023) which is a score for how well the strong model generalized. We show that linear probes trained to predict whether a square is empty, ours, or the opponent's are the most correlated with WSG, as measured by accuracy (93%), Spearman rank-correlation (0.884), and R^2 (0.214). As an ablation we compare the predictiveness with the difference in the model size, performance before finetuning and two different board feature bases (empty/filled), (empty/black/white) and a non-linear transform of the board. Note that (Nanda, Lee, et al. 2023; K. Li et al. 2024) showed that transformers use the empty/ours/opponents stone bases instead of empty/black/white.

As this is the first investigation of WSG using tools from mechanistic interpretability, our work serves as a proof of concept. Othello and Tic-Tac-Toe serve as new toy-testbeds for this, because they have no pretrain leakage and their rules can be easily modified to test the conditions under which WSG is effective. Our results strengthen the hypothesis, supported by prior theoretical work, that the superior features of the strong model enable WSG.

2 Background

2.1 Weak-to-Strong Generalization

Set-up. Weak-to-Strong Generalization refers both to the phenomenon of weak supervision working, and a setup to experimentally investigate it. First proposed in Burns et al. (2023), a weak model M_w gets trained on a weak task D_w and a strong model M_s on a strong task D_s. In the analogy, M_w stands for the human and M_s for the superhuman AI with a different objective. To test if humans can align superhuman AI, we finetune M_s on the output of M_w to get $M_{s\mapsto w}$. A large enough model M_s converges towards M_w, but it was observed that early in the finetuning the strong model $M_{s\mapsto w}$ can surpass its weak teacher M_w (Xu et al. 2025; Burns et al. 2023). Experiments usually early-stop the finetuning on the ground truth labels D_w, although this is unrealistic since we won't have superhuman ground truth labels.

Performance-Gap-Recovered (PGR). In our experiments, we measure performance through the Cross-Entropy loss $CE(M, D)$ for a model M and task D. We say WSG occurs when the strong model surpasses its weak teacher, i.e. $CE(M_{s\mapsto w}, D_w) < CE(M_w, D_w)$. We want to know how close the weakly finetuned model $M_{s\mapsto w}$ comes to a strong baseline M_{sb}, which is M_s pretrained on ground-truth data (D_w instead of D_s). Burns et al. (2023) proposed the Performance-Gap-Recovered (PGR) metric that we adapt for CE-loss. It is 0 if $M_{s\mapsto w}$ matches M_w, positive if it surpasses M_w and 1 if it matches M_{sb}:

$$PGR(M_w, M_{s\mapsto w}, M_{sb}) = \frac{CE(M_w, D_w) - CE(M_{s\mapsto w}, D_w)}{CE(M_w, D_w) - CE(M_{sb}, D_w)}.$$

2.2 Mechanistic Interpretability of Othello.

Linear probe. The linear representation hypothesis states that models represent a significant fraction of their features as linear directions in their activations (Elhage, Hume, et al. 2022; B. Z. Li et al. 2021; Gurnee, Nanda, et al. 2023; Geva et al. 2021). A linear probe is a linear model trained on intermediate activations of a transformer. If it can consistently predict the value of a feature correctly, it suggests that the feature is linearly represented.

Othello. We want to analyze whether a superior world understanding of the strong model helps to learn from a weak supervisor. We use the board game Othello, because it was previously shown that a transformer trained autoregressively on Othello moves internally represents the board state and uses it for its output (Nanda, Lee, et al. 2023; K. Li et al. 2024). This is an example of a world model, where to solve a task a model has to gain understanding of the task and not just superficial correlations (Gurnee and Tegmark 2024; Lovering et al. 2022).

Othello is a 2-player game that is played on a 8x8 chessboard. Players take turns placing a stone on an empty field that flanks opponent pieces in one of the 8 directions between the new stone and an existing stone of the same color. After placing, the color of all flanked stones is flipped. Making a legal move therefore depends on the stones in other areas of the board. First, (K. Li et al. 2024) showed that there are no linear features that represent whether a field on a board is empty/white/black. Afterwards, (Nanda, Lee, et al. 2023) showed that the board state is represented as linear directions empty/mine/yours for each of the 64 fields. Using these directions to modify the activations to represent a different board state changed the predicted legal moves accordingly (Nanda, Lee, et al. 2023; Belinkov 2021).

3 Related Work

Tasks with shown WSG. The most used task is NLP classification, either binary (Burns et al. 2023; Ye et al. 2024; Yao et al. 2025; W. Yang et al. 2024; Hao Lang et al. 2025) or multiclass (Y. Guo and Yi Yang 2024). Because these tasks are more similar to knowledge elicitation and use complex LLMs, evaluating the role that world models play in WSG is difficult.

Generative tasks have been used for tasks such as answering math questions (Y. Guo and Yi Yang 2024; Yuqing Yang et al. 2024), answering in a specific style (Somerstep et al. 2024) and solving chess puzzles (Burns et al. 2023). These setups have pre-training leakage (Burns et al. 2023), since

the strong models dataset D_s contains examples of the weak task D_w. But superhuman AI won't have examples of superhuman aligned behaviour that it can elicit. For example, Burns et al. (2023) used as a weak task D_w chess puzzles where a model learns to predict the best move in a chess puzzle. But the pretraining dataset D_s of the strong model GPT4 contained not only text data but also chess games of players ranked above 1800 Elo. J. Guo et al. (2024) use a task without pretrain leakage in which they finetune a vision-autoencoder for image classification. But the used models are not transformers and it is not a generative task. Our Othello and Tic-Tac-Toe environments are the first generative tasks, where the strong data D_s does not contain examples of the weak task D_w.

Theoretical analysis of WSG has been mostly focused on linear models over fixed features. These findings often got empirically validated on Gaussian distributions (Wu and Sahai 2024; Ildiz et al. 2025; Shin et al. 2024; Charikar et al. 2025) which do not exhibit characteristics such as superposition (Elhage, Hume, et al. 2022). Xue et al. (2025) empirically tests the role of the strong models features by finetuning a full transformer, while other work has trained only a linear model as the last layer.

Games in mech-int. Board games have been successfully used in mechanistic interpretability before. For chess (Toshniwal et al. 2022), Othello (Nanda, Lee, et al. 2023; K. Li et al. 2024) and Tic-Tac-Toe (Ayyub 2025) parts of the algorithm were reconstructed. Chess has also been used as a benchmark for interpretability techniques (Toshniwal et al. 2022). We continue this line of work by showing that WSG occurs in Othello and Tic-Tac-Toe and can be investigated using these environments.

Theoretical analysis of WSG. Prior theoretical results for simplified scenarios have shown that the features of the weak and strong model influence whether WSG is possible. The strong model ideally has features that are helpful for the weak task and unrelated to the errors of the weak model (Hunter Lang et al. 2024; Xue et al. 2025; Hunter Lang et al. 2024; Wu and Sahai 2024). Furthermore, data points are needed on which the strong model can learn from the weak model how to utilize its superior features (Shin et al. 2024). Dong et al. (2025) and Xu et al. (2025) showed theoretically, that with enough finetuning the strong model converges to the weak model, but if the strong features are good enough it first surpasses the weak model's performance. Through mechanistic interpretability, we add a novel class of evidence for the importance that the features of the strong model play - without theoretical assumptions or Gaussian features.

4 Game Environment to study WSG

4.1 Method

Othello Data. We base our Othello environment on the code of (K. Li et al. 2024) and expand it by adding new game rules and the WSG-pipeline. Othello has 60 playable fields (the center 4 stones are pre-placed) and each field can get placed on exactly once. We define 60 tokens, one for each field. We then sample games under 7 different rule sets 1. For example, in standard Othello we choose the next move autoregressively with uniform probability over all legal moves. We remove the $\approx 1\%$ of games that end early because no player can move. We split the data into four sets. To prevent data leakage from a model memorizing sequences, we split over the 12 possible combinations of first two moves in Othello and start training on the third move. This split is the same for all rules we train on, that is, for non-Othello rules we still sample the first 2 moves from this split. We use 26M games for pretraining the weak and strong model and for linear probes (`train`), 13M for the weak model's finetuning phase (`finetune`), 4M for early-stopping (`val`) and 9M for evaluation (`test`).

Othello Training. We then train transformers autoregressively using cross-entropy loss. For example, if we train on the weak rule of Standard Othello, our model predicts the third move of [F4, F5, F6, G4, ...] given the preceding moves [F4, F5]. Since we sampled the data uniformly over all legal moves, a perfect model should assign for the 4 legal moves $\{$F6, D2, C3, E5$\}$ a uniform probability of $1/4$ each. We do not explicitly teach the model any rules of Othello, instead it learns purely from next-token prediction on randomly sampled games.

We adopted hyperparameters A.2 from (K. Li et al. 2024) where possible. We train with this procedure the weak-rule `standard` and the strong rules `bias_clock`, `next_to_opponent` and `no_flipping` as defined in 1. We create two additional strong models that stay untrained: one with random

parameters `untrained` and one with constant parameters `constant_parameters`. Lastly, we take the Chess-transformer from Toshniwal et al. (2022) and train new embedding and unembedding matrices for Othello by keeping all other parameters fixed and training on the `bias_clock` data to obtain `chess`. This idea was previously applied to vision models in LLaVA Liu et al. (2023). We use 7 different sizes `nano`, `micro`, `mini`, `small`, `medium`, `large`, `huge` A.1 where `huge` is the same architecture as the Othello (K. Li et al. 2024) and Chess (Toshniwal et al. 2022) transformers.

As a result, we have for all 7 rules (1 weak + 6 strong rules) each 7 transformers pretrained and one for `chess`. Then we finetune for every pair (M_w, M_s), where M_w is smaller than M_s, on the softlabels of the weak model M_w through CE-loss the strong model M_s. This results in $5 \cdot (1 + 2 + 3 + 4 + 5 + 6) + 1 \cdot (6) = 111$ datapoints. Finetuning is early-stopped based on ground-truth labels, which is the standard way in the literature. However, this is a missing piece in the analogy of aligning superhuman models since we might not be able to evaluate the performance or alignment of superhuman models.

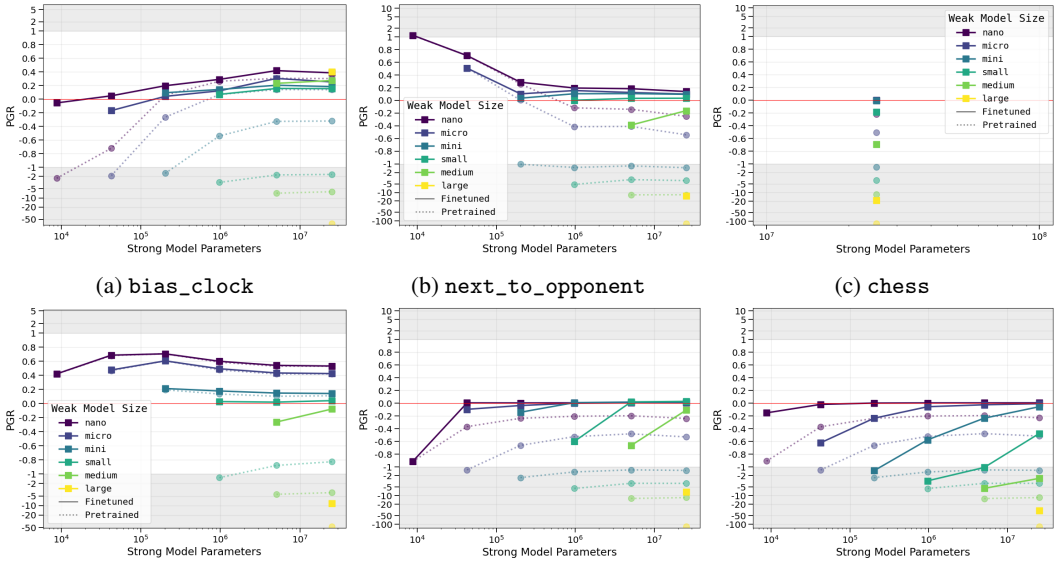
Tic-Tac-Toe. Our Tic-Tac-Toe environment and training work in the same way as in Othello. It builds on the Tic-Tac-Toe implementation of Ayyub (2025). Instead of focusing on legal moves, the model is now required to learn uniform probabilities over the optimal moves (which we determine using a min-max algorithm). The Tic-Tac-Toe strong rule `no_diagonals` is similar to Tic-Tac-Toe, but a player that completes a diagonal automatically looses (instead of winning) 1. The train test split is again over the first two moves. Since Tic-Tac-Toe is small, it happens that two games from train and test that started differently end up at the same state. To minimize this, we modify the reward to -1 for a loss, 0 for a draw, 1 for a win and 2 if one of the winning conditions includes a player's first stone (we split over these.). 10% of the training board states are still also part of the test set. The reason is that a board states with enclosed first placed stones can also occur in the same way in the test set. We run a sweep of $n = 10$ by independently generating the data, splits and model trainings.

Table 1: **Rule Definitions for Othello and Tic-Tac-Toe**

Rule	**Definition**
Othello	
`standard` (weak rule)	Only legal Othello moves. Uniform probability.
`bias_clock`	Only legal Othello moves. 80% chance of field closest to corner `move-index % 4`. 20% uniform $\implies$ strong bias.
`next_to_opponent`	Uniform over fields next/diagonal to an opponent piece. Only flips all direct neighbors $\implies$ no long-range dependencies.
`no_flipping`	70% chance of uniform over fields next/diagonal to an opponent piece. 30% chance random field. No stones get flipped.
`chess`	Chess model from Toshniwal et al. (2022) adapted to Othello vocabulary with LLaVA (Liu et al. 2023).
`untrained`	Strong model is randomly initialized.
`constant_parameters`	Strong model starts with all weights and biases set to their mean (except embeddings and first attention module).
Tic-Tac-Toe	
`standard` (weak rule)	Uniform over min-max optimal moves in Tic-Tac-Toe.
`no_diagonals`	Uniform over min-max optimal moves for Tic-Tac-Toe if completing a diagonal instantly looses.

4.2 Results

Othello. In 2 we can see that for rules more similar to standard Othello, the PGR is positive, i.e. strong student surpassed its weak teacher. In `bias_clock` the model is able to unlearn a bias, and in `next_to_opponent` and `no_flipping` it learned the basics of playing on the board from



(a) `bias_clock` (b) `next_to_opponent` (c) `chess`

(d) `no_flipping` (e) `untrained` (f) `constant_parameters`

Figure 2: **Othello: Performance-Gap-Recovered for different model sizes and strong rules**. Each plot shows the WSG result of finetuning a large model pretrained on a strong rule through a weak model pretrained on standard Othello. E.g. in 2a a biased Othello model has to learn to play unbiased. The x-axis is the strong model size, while the color is the weak model size (as defined in 3). Each square is one pair (M_w, M_s) where M_w is the weak model trained on Othello and M_s is a bigger model trained on the strong rule stated in the caption. The y-axis is the Performance-Gap-Recovered metric (PGR) 2.1. It is 0 if, after finetuning, M_s exactly matches M_w's performance on the weak rule. It is positive if M_s surpasses it, which indicates that WSG worked. The circular dots are the same metric but computed before the finetuning as an ablation to check if the strong model was already performing well before finetuning. We can roughly see that for rules that are closer to Othello (`bias_clock` 2a, `next_to_opponent` 2b, `no_flipping` 2d) the strong model surpasses the weak model in many cases. But if the strong model was not pretrained on a useful task (`chess` 2c, `untrained` 2e, `constant_parameters` 2f) the strong model usually at most matches the weak supervisor's performance.

pre-training, but the core dynamics of long-range dependencies and the stone flipping had to be learned during finetuning. Note that for the model sizes `nano` and `micro` some strong models were already better than the weak model before finetuning (dotted lines are also positive in 2). For `chess`, `untrained` and `constant_parameters`, the strong model does not or only slightly (PGR=4%) surpass its weak teacher. These pairs differ in how well the strong model is able to match the performance of its weak teacher. We can see that an intuitive ordering of how similar the rules are to Othello also matches roughly how well the generalization happens. Chess did not work, however this might be because the model from Toshniwal et al. (2022) was trained on portable game notation, where a single token does not necessarily correspond to a single move.

Tic-Tac-Toe. In 3 its visible that the smaller models `nano`, `small`, `mini` all get surpassed by their strong students. The biggest model recovers roughly 60% of the performance compared to getting trained from ground-truth data. But the better the supervisor, the more difficult it is for the student to surpass them. Except for some outliers, the resulting PGR score is stable across the sweeps.

5 Predicting WSG through mechanistic interpretability

5.1 Method

Training Linear Probe. Nanda, Lee, et al. (2023) and K. Li et al. (2024) showed a transformer trained on Othello (linearly) represents the board state and makes use of it. Therefore, we measure

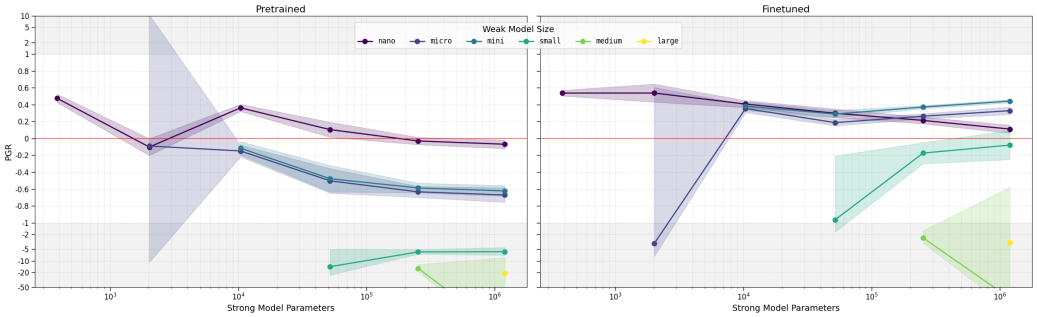


Figure 3: **Tic-Tac-Toe: Performance-Gap-Recovered for different model sizes**. The plot is the same as 2. We plot for a sweep of $n = 10$ the mean and its standard errors, i.e. $\mu \pm \sigma/\sqrt{n}$. For smaller weak models PGR is positive, i.e. they are surpassed by their stronger students. If the weak model is larger, PGR is negative, because they are already playing close to perfect.

the board understanding of a model as the accuracy of linear probes that predict the board state. Our goal is to test the hypothesis that the difference in board understanding between the strong and weak model is related to the strength of weak-to-strong generalization. We use a similar setup to (Nanda, Lee, et al. 2023) and predict with a linear probe the board state in the basis empty/mine/yours for each model. Concretely: For a subsequence $x_1, \ldots, x_k$ where the Transformer has to predict x_{k+1} we take the activations after layer $l = \text{round\_down}(4/3 \cdot \text{n\_layer})$, which is a vector $a_l \in \mathbb{R}^{\text{d\_model}}$. We define as the board state after the first k moves for each of the 64 board fields one target $y_1, \ldots, y_i, \ldots y_{64} \in \{(1,0,0),(0,1,0),(0,0,1)\}$ where y_i is $(1,0,0)$ if the i-th field is empty, $(0,1,0)$ for having a stone of "my" color, i.e. of the currently moving player and $(0,0,1)$ if it is the color of the opponent ("yours"). The strong rules `no_flipping` and `next_to_opponent` have different game dynamics, but we still train probes on standard Othello games. Then, we train 64 linear models of the form $\text{softmax}(W_i \cdot a_l) \approx y_i$ where $W_i \in \mathbb{R}^{3 \times \text{d\_model}}$ using the hyperparameters A.2. The linear probe accuracy is defined as the proportion of board fields averaged over a sample of (sub-)games that gets correctly predicted if we take the maximum of the predicted soft-probabilities. An untrained model should have 33% accuracy and since fields are more often empty, each probe should get at least 46% after training. We calculate this accuracy for every pretrained weak/strong model and every finetuned model and obtain a score LP-acc(M) for each.

Measure of success. To test the hypothesis that better board representations mean that strong models can generalize the weak supervision, we define $X_i = \text{LP-acc}(M_s) - \text{LP-acc}(M_w)$, where (M_w, M_s) is the i-th (weak, strong) model pair. Note that the weak model is smaller than the stronger one. We want to measure the relation between $Y_i = PGR(M_w, M_{s \mapsto w}, M_{sb})$ and X_i. Across all i, we evaluate a) the sign accuracy $\mathbb{E}[\mathbf{1}_{\text{sign}(X_i)=\text{sign}(Y_i)}]$ where $\mathbf{1}$ denotes the indicator function, b) Spearman's rank correlation coefficient $\rho_s(X,Y) = \rho(\text{Rank}(X), \text{Rank}(Y))$ and c) the coefficient of determination $R^2(X,Y)$. Note that since PGR is a non-linear transformation of model performances, Spearman's rank correlation is more informative than Pearson's, as the former captures non-linear monotonic relationships between X and Y (de Winter et al. 2016). We are using the sign accuracy of X and Y as the simplest possible classification rule that tests the hypothesis that better representations allow the strong student to surpass its weak teacher.

Ablation. For our ablation study we test this for 3 different definitions of board state y_i. We use the basis empty/white/black which was shown not to be used by a large Transformer in K. Li et al. (2024). The empty/filled basis gets used (Nanda, Lee, et al. 2023), but since it only tracks whether a token already occured it is a feature that could also occur in non-Othello models. Further, we define a highly non-linear board state feature by creating a vector in $\{0,1,2\}^{64}$ through the empty/mine/yours basis. Then we multiple it with a random matrix in $\{-1,0,1\}^{64 \times 64}$ and apply a modulo 3 operation to the resulting vector to obtain 3 classes again. We expect no model to internally represent this feature well, since its non-linearity makes it expensive to compute and it is unrelated to the task. Lastly, we also check if in general bigger models generalize over smaller models, i.e. we set $X_i = \log(\text{n\_params}(M_s)) - \log(\text{n\_params}(M_w))$. We also investigate if a strong model that is already proficient on the weak task generalizes better (this is trivially true if the strong model is better than the weak model even before finetuning) by setting $X_i = \text{CE}(M_w, D_w)) - \text{CE}(M_s, D_w))$.

5.2 Results

Relationship between Features and WSG. Table 2 shows that the difference in strength of linear representations of the empty/mine/yours basis is strongly correlated with the strength of WSG. It has in 93% of the cases the same sign and a Spearman correlation of 0.844. Its correlation metrics are higher than those of all other baselines. We can further see, that the other features also correlate with WSG - altough less. In 4a we can see how if WSG occured or not is linearly separable by the board representation features. In 4b we plot X vs. Y and see a strong monotonic relationship which is reflected in the high Spearman correlation. The cross-entropy loss before finetuning 4d is only monotonically related to WSG for the pairs where the strong model is already better before finetuning (i.e. right side with $X > 0$). Neither the model size 4e nor the highly non-linear feature 4c has a clear relation with the success of WSG.

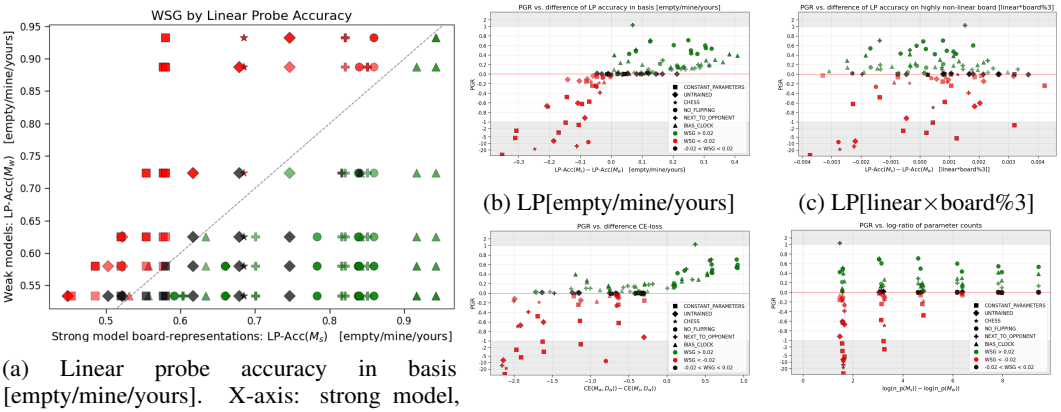
Name	**Definition** X_i	**Same Sign**	**Spearman** ρ **(p-val)**	R^2
Linear Probes				
Empty/Mine/Yours	LP-Acc(M_s) − LP-Acc(M_w)	**0.937%**	**0.844 (3.4e-31)**	**0.214**
Empty/Filled	LP-Acc(M_s) − LP-Acc(M_w)	0.838%	0.771 (4.7e-23)	0.107
Empty/Black/White	LP-Acc(M_s) − LP-Acc(M_w)	0.829%	0.781 (5.5e-24)	0.096
Linear×board%3	LP-Acc(M_s) − LP-Acc(M_w)	0.532%	0.009 (9.2e-01)	0.136
Non interpretability based methods				
Cross-Entropy	CE$(M_w, D_w))$ − CE$(M_s, D_w))$	0.604%	0.729 (1.1e-19)	0.167
N_params (n_p)	log(n_p(M_s)) − log(n_p(M_w))	0.640%	-0.041 (6.7e-01)	0.036

Table 2: **Metrics to predict PGR vs. actual PGR 5.1 5.1**. N=111, 71 positive, 40 negative PGR. Each row defines for each pair (M_w, M_s) a value X_i, e.g. the first row is for the difference in accuracy for linear probes trained on M_w and M_s. The correlation metrics on the right are computed between X_i and $Y_i = PGR$ 2.1 over all 111 pairs. The difference in accuracy of a linear probe, which predicts if a board field is empty or has the color of the current player or the other players color, is the most correlated with the success of WSG. The p-values for having at least as strong Spearman correlation as observed are for everything except the non-linear board state and number of parameters significant/very low - but we have to account for hierarchical dependence which inflates the number of independent datapoints (Bogdan 2025). The full datapoints are visualized in 4.

Training dynamics. In 5a we see that if the strong model surpasses its weak teacher, it happens early during the first 1000 finetuning steps. While if the strong model only matched or did not reach the weak model's performance, the strong model reached its best validation score late in training. This suggests that if WSG occurs, it occurs through small changes to the model's parameters, while if it does not surpass, it converges towards the weak model's output. Plot 5b supports this. Here we can see that in examples where WSG occured the strong model after finetuning has roughly the same level of board representations as before finetuning. But in the examples of no WSG, its Othello board representations improved. One loose interpretation is that WSG works if the strong model only has to learn small rule changes, while if it also has to learn a world model it starts to copy the weak model and converge against it instead of surpassing it.

6 Limitations

We base our analysis on a toy language derived from the game of Othello, which may not transfer to frontier LLMs. Our 111 Othello-finetuning runs differ in model size and rule pairs, but since they are based on seven different rules played on an Othello board they are hierarchically dependent (Bogdan 2025). Since we only use linear probes, we show that high-quality board representations correlate highly with WSG. However, we do not show that the strong model actually uses these features to fit the weak-finetuning signal. But, prior work (Nanda, Lee, et al. 2023; K. Li et al. 2024) has shown that board representations are used to play Othello. Our work provides insights into when WSG works, but it does not offer practical future-proof techniques, since probing for useful features in superhuman models might be equally difficult as evaluations.



(a) Linear probe accuracy in basis [empty/mine/yours]. X-axis: strong model, y-axis: weak model. Dotted line is y=x.

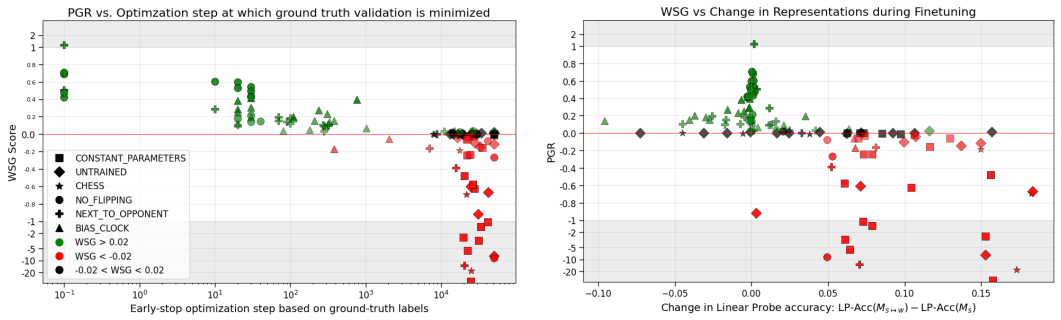
(b) LP[empty/mine/yours]

(c) LP[linear×board%3]

(d) CE-loss

(e) log(n_params)

Figure 4: **Relation between PGR and different metrics.** Our goal is to find a metric $X_i(M_w, M_s)$ that given a pair of weak and strong model can predict if WSG works, as measured by Performance-Gap-Recovered $Y_i = PGR$ 2.1. In all plots, the color represents the PGR: Green means the strong model surpassed the weak model's performance on the weak task, i.e. it plays legal moves in Othello more reliably. Black signifies that the strong model matched the performance, and red that it performed worse. The shapes represent the strong rule of the model. The x- and y-axis are different metrics based on M_w and M_s. On the left 4a we have on the x-axis the strong models and on the y-axis the weak models board representation. The decision boundary that splits green vs. red points is naturally on the line LP-Acc(M_s) > LP-Acc(M_w). On the right, we plot X_i on the x-axis vs. $Y_i = PGR$ on the y-axis (same as in 2). In 4b, we can see how the difference in board representations in the basis [empty/mine/yours] monotonically relates to the PGR metric. The other metrics 4c, 4d, 4e are less related because there is no point on the x-axis that splits the green and red points and the relation between X_i and Y_i is weaker.



(a) PGR vs. optimization step at which the ground truth validation loss is minimized.

(b) PGR vs. the change in accuracy of board representations during finetuning. Positive means it got better.

Figure 5: **Finetuning dynamics of PGR and board representations.** The plot is similar to 4. We want to understand the board representations during finetuning. The left plot 5a has green points on the left and red points on the right. It indicates that if the strong model surpasses its weak teacher, this happens early in the finetuning. On the right 5b the green points are around 0 and the red ones are positive. If WSG occurred, the board representations remain mostly unchanged. But if the strong model had worse representations and had to learn them during finetuning, WSG does not work.

7 Conclusion

While previous environments often either didn't use transformers, or leaked examples of the weak model task into the pretraining of the strong model, board games with different rules provide a clean environment. We show an example of interpretable features that are related to the success of WSG. A further idea that we did not yet finish was to use a Crosscoder (Minder et al. 2025; Lindsey et al. 2024) to determine if the similarity between a model and a very strong Othello model serves as a better measure since it also includes less interpretable features.

Works Cited

Ayyub, Omar (July 2025). *Discovering Player Tracking in a Minimal Tic-Tac-Toe Transformer*. https://omar.bet/2025/07/17/Discovering-Player-Tracking-in-a-Minimal-Tic-Tac-Toe-Transformer/.

Belinkov, Yonatan (2021). *Probing Classifiers: Promises, Shortcomings, and Advances*. arXiv: 2102.12452 [cs.CL]. URL: https://arxiv.org/abs/2102.12452.

Bogdan, Paul (Aug. 2025). *Statistical suggestions for mech interp research and beyond*. LessWrong. URL: https://www.lesswrong.com/posts/GxhtzqMwdTHo6326y/statistical-suggestions-for-mech-interp-research-and-beyond (visited on 08/12/2025).

Burns, Collin et al. (2023). *Weak-to-Strong Generalization: Eliciting Strong Capabilities With Weak Supervision*. arXiv: 2312.09390 [cs.CL]. URL: https://arxiv.org/abs/2312.09390.

Charikar, Moses, Chirag Pabbaraju, and Kirankumar Shiragur (2025). "Quantifying the gain in weak-to-strong generalization". In: *Proceedings of the 38th International Conference on Neural Information Processing Systems*. NIPS '24. Vancouver, BC, Canada: Curran Associates Inc.

Christiano, Paul et al. (2023). *Deep reinforcement learning from human preferences*. arXiv: 1706.03741 [stat.ML]. URL: https://arxiv.org/abs/1706.03741.

De Winter, Joost C. F., Samuel D. Gosling, and Jeff Potter (Sept. 2016). "Comparing the Pearson and Spearman correlation coefficients across distributions and sample sizes: A tutorial using simulations and empirical data." In: *Psychological Methods* 21.3, pp. 273–290. URL: http://dx.doi.org/10.1037/met0000079.

Dong, Yijun et al. (2025). *Discrepancies are Virtue: Weak-to-Strong Generalization through Lens of Intrinsic Dimension*. arXiv: 2502.05075 [cs.LG]. URL: https://arxiv.org/abs/2502.05075.

Elhage, Nelson, Tristan Hume, et al. (2022). "Toy Models of Superposition". In: *Transformer Circuits Thread*. https://transformer-circuits.pub/2022/toy$_model/index.html$.

Elhage, Nelson, Neel Nanda, et al. (2021). "A Mathematical Framework for Transformer Circuits". In: *Transformer Circuits Thread*. https://transformer-circuits.pub/2021/framework/index.html.

Geva, Mor et al. (2021). *Transformer Feed-Forward Layers Are Key-Value Memories*. arXiv: 2012.14913 [cs.CL]. URL: https://arxiv.org/abs/2012.14913.

Guo, Jianyuan et al. (2024). *Vision Superalignment: Weak-to-Strong Generalization for Vision Foundation Models*. arXiv: 2402.03749 [cs.CV]. URL: https://arxiv.org/abs/2402.03749.

Guo, Yue and Yi Yang (2024). "Improving weak-to-strong generalization with reliability-aware alignment". In: *arXiv preprint arXiv:2406.19032*.

Gurnee, Wes, Neel Nanda, et al. (2023). *Finding Neurons in a Haystack: Case Studies with Sparse Probing*. arXiv: 2305.01610 [cs.LG]. URL: https://arxiv.org/abs/2305.01610.

Gurnee, Wes and Max Tegmark (2024). *Language Models Represent Space and Time*. arXiv: 2310.02207 [cs.LG]. URL: https://arxiv.org/abs/2310.02207.

Ildiz, M. Emrullah et al. (2025). *High-dimensional Analysis of Knowledge Distillation: Weak-to-Strong Generalization and Scaling Laws*. arXiv: 2410.18837 [stat.ML]. URL: https://arxiv.org/abs/2410.18837.

Karpathy, Andrej (2020). *minGPT: A minimal PyTorch re-implementation of the OpenAI GPT training*. https://github.com/karpathy/mingpt.

Lang, Hao, Fei Huang, and Yongbin Li (2025). "Debate helps weak-to-strong generalization". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 39. 26, pp. 27410–27418.

Lang, Hunter, David Sontag, and Aravindan Vijayaraghavan (2024). "Theoretical analysis of weak-to-strong generalization". In: *Advances in neural information processing systems* 37, pp. 46837–46880.

Li, Belinda Z., Maxwell Nye, and Jacob Andreas (2021). *Implicit Representations of Meaning in Neural Language Models*. arXiv: 2106.00737 [cs.CL]. URL: https://arxiv.org/abs/2106.00737.

Li, Kenneth et al. (2024). *Emergent World Representations: Exploring a Sequence Model Trained on a Synthetic Task*. arXiv: 2210.13382 [cs.LG]. URL: https://arxiv.org/abs/2210.13382.

Lindsey, Jack et al. (2024). *Sparse crosscoders for cross-layer features and model diffing*. Transformer Circuits Thread. Accessed: 2025-08-10. URL: https://transformer-circuits.pub/2024/crosscoders/index.html.

Liu, Haotian et al. (2023). *Visual Instruction Tuning*. arXiv: 2304.08485 [cs.CV]. URL: https://arxiv.org/abs/2304.08485.

Lovering, Charles et al. (2022). *Evaluation Beyond Task Performance: Analyzing Concepts in AlphaZero in Hex*. arXiv: 2211.14673 [cs.AI]. URL: https://arxiv.org/abs/2211.14673.

Minder, Julian et al. (2025). *Overcoming Sparsity Artifacts in Crosscoders to Interpret Chat-Tuning*. arXiv: 2504.02922 [cs.LG]. URL: https://arxiv.org/abs/2504.02922.

Nanda, Neel and Joseph Bloom (2022). *TransformerLens*. https://github.com/TransformerLensOrg/TransformerLens.

Nanda, Neel, Andrew Lee, and Martin Wattenberg (2023). *Emergent Linear Representations in World Models of Self-Supervised Sequence Models*. arXiv: 2309.00941 [cs.LG]. URL: https://arxiv.org/abs/2309.00941.

Radford, Alec et al. (2019). "Language Models are Unsupervised Multitask Learners". In: URL: https://api.semanticscholar.org/CorpusID:160025533.

Shin, Changho, John Cooper, and Frederic Sala (2024). "Weak-to-strong generalization through the data-centric lens". In: *arXiv preprint arXiv:2412.03881*.

Somerstep, Seamus et al. (2024). "A transfer learning framework for weak-to-strong generalization". In: *arXiv preprint arXiv:2405.16236*.

Toshniwal, Shubham et al. (2022). *Chess as a Testbed for Language Model State Tracking*. arXiv: 2102.13249 [cs.CL]. URL: https://arxiv.org/abs/2102.13249.

Wu, David X and Anant Sahai (2024). "Provable weak-to-strong generalization via benign overfitting". In: *arXiv preprint arXiv:2410.04638*.

Xu, Gengze et al. (2025). *On the Emergence of Weak-to-Strong Generalization: A Bias-Variance Perspective*. arXiv: 2505.24313 [cs.LG]. URL: https://arxiv.org/abs/2505.24313.

Xue, Yihao, Jiping Li, and Baharan Mirzasoleiman (2025). "Representations shape weak-to-strong generalization: Theoretical insights and empirical predictions". In: *arXiv preprint arXiv:2502.00620*.

Yang, Wenkai et al. (2024). "Super (ficial)-alignment: Strong models may deceive weak models in weak-to-strong generalization". In: *arXiv preprint arXiv:2406.11431*.

Yang, Yuqing, Yan Ma, and Pengfei Liu (2024). *Weak-to-Strong Reasoning*. arXiv: 2407.13647 [cs.CL]. URL: https://arxiv.org/abs/2407.13647.

Yao, Wei et al. (2025). *Revisiting Weak-to-Strong Generalization in Theory and Practice: Reverse KL vs. Forward KL*. arXiv: 2502.11107 [cs.LG]. URL: https://arxiv.org/abs/2502.11107.

Ye, Ruimeng, Yang Xiao, and Bo Hui (2024). "Weak-to-strong generalization beyond accuracy: a pilot study in safety, toxicity, and legal reasoning". In: *arXiv preprint arXiv:2410.12621*.

A Technical Appendices and Supplementary Material

A.1 Model Hyperparameters

Table 3: **Model hyperparameters.** We use GPT-2 style transformers (Radford et al. 2019) with $d_{\text{mlp}} = 4 \times d_{\text{model}}$ and $d_{\text{model}} = n_{\text{head}} \times d_{\text{head}}$ through the TransformerLens library (Nanda and Bloom 2022) and MinGPT (Karpathy 2020) as used in (K. Li et al. 2024). The `huge` transformer hyperparameters are identical to those of the model investigated in (Nanda, Lee, et al. 2023; K. Li et al. 2024).

	Othello				**Tic-Tac-Toe**			
Model Size	n_{layer}	n_{head}	d_{model}	$n_{\text{parameters}}$	n_{layer}	n_{head}	d_{model}	$n_{\text{parameters}}$
`nano`	1	1	7	≈ 2.0K	1	1	1	68
`micro`	1	2	20	≈ 8.7K	1	2	4	390
`mini`	2	2	38	≈ 43K	2	4	8	≈ 2K
`small`	3	3	72	≈ 200K	3	4	16	≈ 10K
`medium`	4	5	140	≈ 970K	4	8	32	≈ 52K
`large`	6	6	264	≈ 5.1M	5	8	64	≈ 250K
`huge`	8	8	512	≈ 25M	6	16	512	≈ 1.2M

A.2 Training Hyperparameters

We do shorter pretraining (roughly 12h A100 vs. estimated 900h) than (K. Li et al. 2024). As a worst-case comparison, on the largest standard Othello model, our approach results in an illegal move probability of 2.97% vs. 0.07% and a out of sample linear probe accuracy of 95.99% vs. 95.93%. However, all models we used in the paper appear fully converged since we only use smaller models for standard Othello and the other rules are simpler with even smaller models playing close to perfect on them.

Table 4: **Othello: Hyperparameters for Pretraining, Finetuning, and Linear Probing.**

Hyperparameter	**Pretrain**	**Finetune**	**Linear Probe**
Max epochs	2	2	1
Early stop patience	—	100	2
Early stop val every n steps	—	100	100
Batch size	512	512	32
Weight decay	0.1	0.1	0.01
Learning rate	5×10^{-4}	1×10^{-5}	1×10^{-4}
Adam betas (β_1, β_2)	(0.9, 0.95)	(0.9, 0.95)	(0.9, 0.95)
Grad norm clip	1.0	1.0	—
LR decay schedule	Cosine	—	—
LR warmup	First 5% (linear)	First 5% (linear)	First 5% (linear)

Table 5: **Tic-Tac-Toe: Hyperparameters for Pretraining, Finetuning.**

Hyperparameter	**Pretrain**	**Finetune**
Learning Rate	1×10^{-3}	1×10^{-5}
Weight Decay	1×10^{-4}	1×10^{-2}
Max Epochs	1000	1000
Batch Size	64	64
Early Stopping Patience over epochs	3	—
Early Stopping Patience over optimization steps	—	100