TIAOQI: A SOLVED GAME AS EXACT GROUND TRUTH FOR CONCEPT DISCOVERY AND ITS VALIDATORS

Anonymous authors
Paper under double-blind review

ABSTRACT

Concept-discovery methods return features with names, and the field validates a name against the model, its activations or human labels, never against whether the concept is true. A strongly solved game supplies that ground truth. Every position has an exact value, so a claimed concept can be checked against it, and so can the test that validated it. We introduce TiaoQi, a benchmark on Chinese checkers that, to our knowledge, is the first to check every concept claim about a network against exact truth, and the first to use the same truth to grade its own tests: how often they pass a random direction, and how often a passing feature passes again. On it, a standard sparse-autoencoder pipeline returns few but real causal handles on the network's moves, 39–62 per agent where noise would earn 0–3.8. The handles sit on the game's own folk concepts, the features found for a fact are needed for play as a set, and the count replicates across seeds, layers, positions, dictionaries and rule variants. The same truth also grades four of the field's validation practices. A shuffled-label control certifies every dictionary where a matched random slice does not, a value-readout pass rule admits 1.8% to 5.5% of noise, a single feature's label is one draw, and decodable and used come apart in both directions.

1 INTRODUCTION

A concept-discovery method takes a trained network and returns features, directions in its activations, each with a name. The name is earned in one of three ways. A person, or a language model that then predicts the feature's own activations, inspects the positions that most excite it (Templeton et al., 2024; Paulo et al., 2025); a probe is fitted to a labelled concept (Hewitt & Liang, 2019); or the feature is pushed and the change in the output is scored (Turner et al., 2024). Each compares the feature with the model, with its own activations or with labels people wrote. None compares it with whether the concept is true. The authors of the largest such study record that whether a feature fires wherever its concept is present could not be measured rigorously (Templeton et al., 2024); an exact table answers that question, and its converse, by construction.

That comparison is not available where discovery is practised, because nobody knows the truth there. A chess feature named "a sacrifice is right here" can be checked against an engine or a grandmaster, and both are opinions. The field says so itself. The survey of concept-description evaluation sorts its metrics into five families, simulation, input-based, output-based, semantic similarity and human judgment, and none compares the description with whether the concept is true of the domain (Feldhus & Kopf, 2025). Every benchmark with known truth constructs its system or scores a proxy, and the work closest to ours in spirit builds simulated systems with known explanations precisely because, in its words, naturally occurring systems paired with objective ground truth are missing (Méloux et al., 2026) (Section 2).

A strongly solved game supplies one. Chinese checkers on the 6×6 board with six pieces a side has 2,313,100,389,600 positions and a table that gives every one its exact value under best play (Sturtevant, 2019). From that table and the board, any yes-or-no property of a position that we care to define can be computed exactly for every position. Whether the mover must wait, whether a retreat is the only good move and whether a greedy advance loses the game are three such properties. Agents trained by self-play never see the table, and their internals can be graded against it. A competent agent must carry an accurate model of the game's dynamics (Richens et al., 2025); whether it carries true strategic concepts is the open question. The substrate is one game family on boards up to 6×6,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

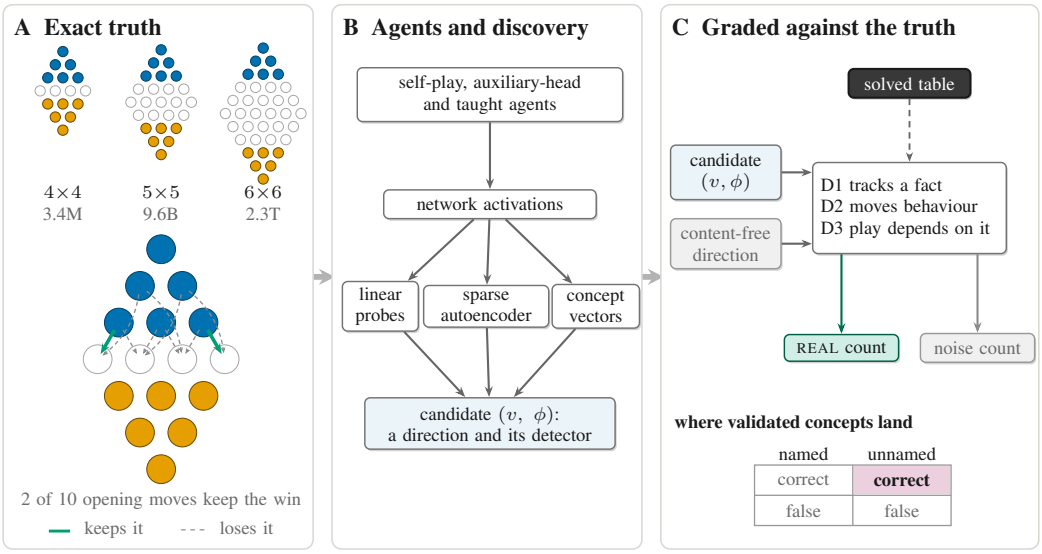


Figure 1: The benchmark in one picture. (A) The testbed: strongly solved boards, every position with an exact value (the 4×4 board is drawn, its opening moves labelled by the table as keeping or losing the win; the grading is on 6×6). (B) Three kinds of agent and three discovery methods propose candidates. (C) The grading protocol grades each candidate against the solved table beside directions with no fact content, which earn the noise count printed beside every count on the report card; a validated concept lands in one of four cells.

and a concept is graded only if it can be written as a fact; what that narrowness buys is decidability. The field made the same trade once before, when board-state recovery became the ground truth for evaluating dictionaries because open domains offered none (Karvonen et al., 2024).

We build a *benchmark* on this substrate in two parts (Figure 1). The *testbed* is the material, four exact tables, a family of exact facts drawn from the game's lore, position pools, trained agents, and two rule-variant worlds with tables of their own (Section 3). The *grading protocol* is the scoring, with representation, causal, dependence and set-level tests under matched controls, and a null program that sends random directions through the same tests to measure the protocol's own false-pass and re-pass rates (Section 4). Its output is a *report card* per method and agent, on which a sparse-autoencoder pipeline returns few but real *causal handles* on the network's moves, for the game's own strategic concepts, and four validation practices are measured against the same truth (Section 5).

Our contributions are four. (1) The testbed is the first where every concept claim about a trained network can be checked against exact truth, released with a small version that runs end to end (Section 3). (2) The grading protocol is the first whose own false-pass rate on random directions and re-pass rate are measured on the network under study (Sections 4 and 5.4). (3) The findings are the first discovered concepts graded against exact truth: features that track exact facts, sets of features that play needs, and few but real handles, named and unnamed, that travel across pools, dictionaries and rules (Sections 5.1 to 5.3). (4) The lessons for validation practice are that the shuffled-label control certifies everything, a value-readout pass rule admits 1.8% to 5.5% of noise, a single feature's label is one draw, and decodable and used come apart in both directions (Sections 5.4, 5.2 and 6).

2 Related Work

Validating discovered features. A discovered feature is validated, in the literature that discovers them, against the model. Top-activation naming checks that the named concept is present where the feature fires and that pushing the feature moves the output (Templeton et al., 2024); the same authors record that the reverse question, whether the feature fires wherever the concept is present, could not be measured rigorously. Automated interpretation, steering, probing under a control task and concept activation vectors score a feature against its own activations, the outputs or a supplied direction (Paulo et al., 2025; Turner et al., 2024; Hewitt & Liang, 2019; Belinkov, 2022; Kim et al.,

2018). In game agents, where discovery has advanced furthest, the validators are agreement with expert or engine labels, puzzles solved correctly "according to AZ's solution", teachability, Elo lost on removal, and probe accuracy, which does not certify a concept's identity (McGrath et al., 2022; Lovering et al., 2022; Pálsson & Björnsson, 2023; Hammersborg & Strümke, 2023; Schut et al., 2025; Pálsson & Björnsson, 2024; Lysnæs-Larsen et al., 2025). None compares the concept with the truth of the domain. The field has begun to measure how much these validators let through: random directions pass them, features differ across seeds, proxy metrics miss practical quality, and a patching result can be an illusion of the subspace or the metric (Korznikov et al., 2026; Paulo & Belrose, 2025; Karvonen et al., 2025; Makelov et al., 2024; Zhang & Nanda, 2024; Reblitz-Richardson, 2026). Concurrent work shows that probe-decodable and behaviourally used directions barely overlap, each against the model's own behaviour (Tiwari et al., 2026; Koushik et al., 2026; Fomin et al., 2026). No work we know measures the false-pass rate of its own causal rule. Two recent assessments find that on detecting or steering a known concept, judged against labels a language model wrote, simpler methods beat sparse autoencoders, and place the autoencoders' use at finding concepts nobody named in advance (Peng et al., 2026; Wu et al., 2025); the largest such study reports that many of its own features are not adequately monosemantic (Gao et al., 2025). We grade causal handles against a solved table, where the same pipeline returns few but real ones. Our protocol measures the same failure modes against exact truth, prints their rates beside every count, grades causal handles rather than detection, and runs the decodability dissociation in the direction the language-model setting cannot.

Benchmarks with known truth. Where interpretability benchmarks have ground truth, they obtain it by construction, from a compiled or specified system, a simulated explanation, or labels that a language model wrote or approximates (Lindner et al., 2023; Gupta et al., 2024; Méloux et al., 2026; Venhoff et al., 2024; Wu et al., 2025). Where the system is real, the metrics are proxies; SAEBench names the availability of ground-truth data as its ceiling (Karvonen et al., 2025), and the closest work in spirit grades a specified explanation, not a discovered one (Méloux et al., 2026). Even where truth is exact it has served faithfulness to the model rather than truth about a domain (Lange et al., 2026; Jacovi & Goldberg, 2020). None of these has a real, unmodified system with exact truth over its whole state space; ours obtains that truth from the domain, for networks trained the ordinary way, and measures its grader's error rate, which none of the above reports.

Games as ground truth. Games supply one. The line that founded the study of learned world models defines its truth as the board state and trains no model toward winning play (Li et al., 2023). Its extensions to chess, to poker, to seven language models and to edits of the represented state keep that truth (Karvonen, 2024; Karvonen et al., 2024; Yuan & Søgaard, 2025; Kamel et al., 2025; Harang et al., 2025; Davis et al., 2026; Lee et al., 2026). Exact game truth already grades decisions, human chess moves against endgame tablebases and the play of trained agents against solved small games (Anderson et al., 2016; Terauchi et al., 2024; Marasco & Givigi, 2025; Kong et al., 2026), and superhuman strength coexists with false belief (Lin et al., 2024; Wang et al., 2023; Karki, 2024). Sparse autoencoders on a game agent's internals have been matched to named strategies with no causal test (Tan et al., 2026), and a controlled suite of Othello rule variants studies how one transformer organises several world models (Chawla et al., 2026). The content of a network's representations has not been graded against exact game truth; we grade discovered concepts against each world's own solved table, and we use rule variants to measure what a rule induces.

3 THE TESTBED

Solved tables and facts. A *solved table* assigns every position of a game its exact value under best play, which is a win, a draw or a loss for the player to move. We grade against the table of 6×6 Chinese checkers with six pieces per side, which holds 2,313,100,389,600 positions. A *fact* is a yes-or-no property of a position that is computed exactly from the table and the board, and never from a network; the *fact family* is the fixed list of facts we grade against (Appendix G). A *move fact* is about which moves keep the solved value, and it is graded on the move the network's policy ranks first, without search. An example is `only_move`, which holds when the mover has a win or a draw and exactly one legal move keeps it. A *state fact* is about the position itself and is graded on the network's value head. Because every fact is exact, a claim that a learned feature detects one can be checked position by position.

Positions and the split. Every number is read on positions from games. An *agent-free pool* holds the positions of complete games played by three generators that involve no trained agent (optimal play after a short random prefix, random play and distance-greedy play), in equal shares and with duplicates removed. Games, not positions, are split at random into a training half and a held-out half. The training half fits every threshold, probe and erasure; the held-out half is the only side any number is read on, and within it we score the *scoring positions*, where the mover has a win or a draw to keep. Matched controls draw from a position's *distance cell*, the pair of both sides' distances to goal (Appendix C).

Agents. The subject is a self-play agent, a residual network with 3 blocks of width 512 trained by self-play with tree search for a fixed budget, with the graded snapshot chosen by a playoff that never consults the solved table. Two references stand beside it. One is a second self-play arm with an auxiliary head trained on two exact facts. The other is a *taught agent* of the same architecture trained by supervision on the solved table itself; its value head was trained on the table, so every value-side test is trivial for it, and it is a ceiling and a control, not the study's subject. Each arm has up to three seeds, and we report values per seed and never a pooled mean (Appendix E).

Rule-variant worlds. To measure what a rule induces, we change it. In the *delay world* on the 5×5 board a defender delays an attacker's arrival, and in the *cooperative world* both sides work to swap camps; each has its own exact table. The readout is what the agent plays: how often its move is an *on-schedule move*, one that realizes the table's value, on the positions where a fact holds. We read this for the world's own agent beside a base control, a self-play agent of the unmodified game on the same positions, so that the difference isolates the rule. A taught agent and a random mover give the ceiling and the floor (Appendix P).

4 The grading protocol

Candidates. The candidates we grade are the latents of a sparse autoencoder (Huben et al., 2024; Bricken et al., 2023; Gao et al., 2025), a dictionary of directions of which a few are active on any position, the form that superposition motivates (Elhage et al., 2022). One is trained on each layer's activations, with 8 latents per input dimension (4,096 at width 512), of which 16 are active on a position. Two further methods provide comparison rows. Concept vectors are directions built from the network's own value and policy signals and never see the table, and a linear probe (Gurnee et al., 2023) is fitted to each fact directly, which makes it a reference and never a discovery. Each candidate is assigned the fact it detects best on the held-out half, with its threshold and polarity fitted on the training half (Appendix F). Appendix A contains the terms.

D1: does the candidate track a fact? A sparse feature cannot reach a high balanced accuracy on a fact that is more common than the feature's firing, so a single accuracy would rank features by how often they fire. D1 therefore reports the *purity margin* for a sparse candidate, the share of the held-out positions where it fires on which its fact holds, minus the share that the distance profile of those positions alone would give. For a feature assigned to the rear-piece rule (every move that keeps the value moves the mover's rearmost piece), the margin is how much more often that holds where the feature fires than at other positions with the same distances to goal. Each margin is read beside two bands computed on the same row. One is the margin of the shuffled-label control, which keeps the feature's firing density and distance profile and nothing else; the other is the margins of 3 *matched random slices*, random directions of the same space thresholded to the same firing rate. D1 supports a comparison between latents and such slices; it sets no pass threshold (Appendix H).

D2: is the candidate a handle? A *candidate* is a pair $c = (v, \phi)$ of a unit direction v in one layer's activations and a detector ϕ that says where it fires. D2 tests whether the network's behaviour depends on the candidate in the way its fact predicts. We take held-out positions s where the candidate fires and the fact holds, and for each a twin position s', matched on distance to goal, where the fact is false. We replace the candidate's coordinate at s with its value at s', so that $a(s)$ becomes $a(s) + \langle a(s') - a(s), v\rangle v$ (Figure 2). For a move fact the score is the resulting drop in how often the network's chosen move keeps the solved value; for a state fact it is the share of pairs whose value estimate moves toward the twin's true value. A candidate passes only if it beats 40 random pushes of equal size by three of their standard deviations, shifts of the activation at s along random directions

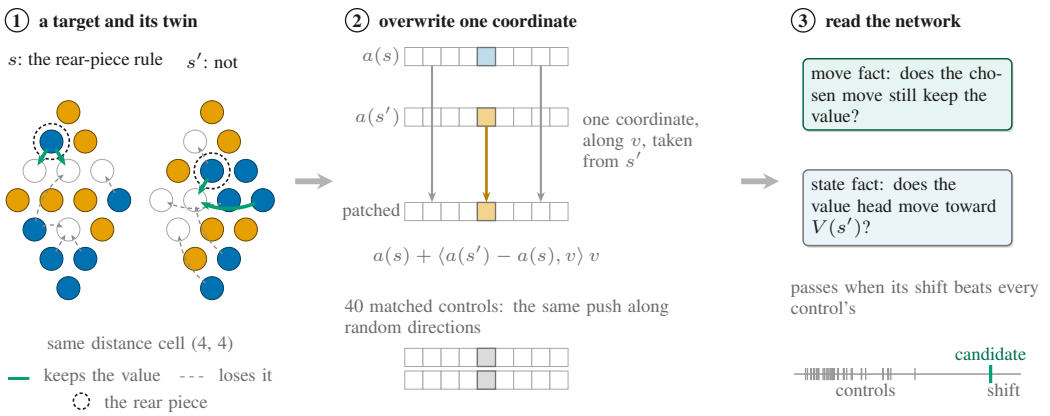


Figure 2: The twin patch of D2, on two real positions. (1) A target s where the fact holds, here the rear-piece rule: the mover is winning, every move that keeps the win (green) moves the ringed piece, the one farthest from its goal, and every move of another piece loses it; and a twin s' from the same distance cell where the rule is false, since a move of another piece keeps the win; both on the 4×4 board. (2) The candidate's coordinate at s is overwritten with its value at s', beside 40 random pushes of equal size. (3) The network is recomputed and its readout compared with the controls'; the ticks of the bar are schematic (Appendix I).

by as much as the candidate's coordinate moved, a value-blind fake that knows the fact's density and nothing else, and, on a state fact, the value head's own gradient (Appendix I). On a move fact the pushes are sized by their effect on the value head (Appendix I says what protects that readout from pushes that are too small).

D3 and the REAL label. D3 is the dependence test, run on the pairs of D2. It measures whether the agent's play depends on the candidate's fact content more than on a random direction. The candidate's coordinate is replaced twice, once from a twin where the fact is false and once from a twin where it is true, the second chosen as far from the target in activation space as the first. The effect is the difference between the two drops in value-keeping play. For a rear-piece feature, the first swap comes from a position where a value-keeping move of another piece exists and the second from one where every value-keeping move moves the rear piece. Play that relies on the feature then suffers more from the first swap than from the second. The same swap with 10 random directions gives a band, and the D3 condition requires that the effect exceed the band's mean by more than its standard deviation. A candidate is REAL when it passes the per-row rule of D2 and meets the D3 condition while firing on at least 6,000 held-out scoring positions (Appendix J).

The noise count. A count of REAL candidates means little without the count that noise would earn. For a cell (one agent, layer, fact and polarity) we send *content-free directions*, random directions fired at the density of the agent's own REAL latents, through the entire chain with fresh twins and controls. We record the rate at which they come out REAL. That rate times the number of eligible latents is the *noise count*. It is printed beside every REAL count together with its status, measured on that cell or borrowed from a neighbouring cell by a rule fixed in advance (Appendix K).

The set-level test. Necessity cannot be established one feature at a time by removal, because the other members of a redundant set stand in for the one removed. The *set-level test* therefore removes all directions a method assigned to a fact at once, every forced-wait feature of the dictionary together, for example. It compares the drop in value-keeping play with 10 random sets and 3 *decoy sets* of the same size (Luo, 2026), at sizes $\{1, 2, 4, 8\}$ and at the full set. A set that fills half the layer is read at the small sizes only, since there a random set removes as much as any other (Appendix J).

Named and unnamed. Each REAL latent is sorted once. It belongs to the race class if its fact is a value class (the race) or the negation of a fact. Otherwise it is named if its purity margin beats all of its own matched random slices, trivial if its margin is no better than the shuffled-label controls', and *unnamed* if neither. Both bars admit known error rates in opposite directions, so the unnamed count is reported as a count with those rates and is neither an upper nor a lower bound (Appendix L).

The cross-play exam. The agent-free pool contains no agent's play. For the *cross-play exam* the nine graded networks play each other in a round-robin, and the positions they visit form the *cross-play pool*. Each arm is then graded a second time on that pool by the same grader. A difference between the two gradings is reported only for a fact whose base rate stays within a factor of 2 between the pools, and beside a reference re-scored at the first pool's base rate (Appendix M).

The greedy-rule pass. The *greedy-rule pass* measures whether the network itself holds a false rule of thumb. On held-out positions where every distance-greedy move gives away the solved value, it measures how often the network plays a greedy move and how its value head prices those positions. Both are read against *look-alike positions* of the same value class and distance cell where greedy play is not wrong. A rank-one linear erasure of the fact then tests whether that discrimination sits in a single direction (Appendix N).

Depth. The solved table says who wins and is silent on how soon. We compute the *conversion depth*, the exact number of plies to the decision under best play, for every position within 7 plies, and derive eleven depth facts from it. For each latent we test whether a depth fact describes it better than its previous best fact, its value-blind fake and distance alone do (Appendix O).

5 RESULTS

This section reports what a standard pipeline returns on the report card, which concepts the handles belong to, whether the result travels, and what the protocol itself lets through and misses. Every claim is made at the level of counts beside a measured noise count, because a single feature's label survives a fresh draw about half the time (Section 5.4).

5.1 WHAT DOES A STANDARD PIPELINE RETURN?

The pipeline's features track facts. Over both self-play agents and every seed and layer, 0.38 to 0.65 of graded features beat every one of their own matched random slices, where a direction with no content would do so one time in four, and 0.21 to 0.48 exceed the mark that a tenth of slices reach (Table 9 in the appendix gives every layer and the taught agents).

Table 1, the report card of the self-play agent, shows few handles, and real ones. The rear-piece rule shows the path of one fact through the tests. At the first layer of the first seed, 34 features were assigned to it and could be tested. Of these, 14 passed the handle test, and 3 of those also passed the dependence test and are REAL. On the move readout the self-play agent's sparse autoencoder returns 52, 62, 39 REAL features per agent (seeds listed in order), and the auxiliary-head agent 54, 43, 48, against noise counts of 0–3.8, 0–3.6 (Table 1 prints each layer; Figure 3; Table 10 in the appendix gives every agent and layer). Every seed and every layer of both arms sits far above its noise count. On the value readout the counts sit under their noise counts at the first two layers and above them at the last, and under them on the cross-play pool (every value-readout count is an upper bound; Table 11 in the appendix). We make single-feature claims on the move readout only and print the value readout beside its noise count, which says what the value score itself lets through.

Three methods were graded on the same report card, and the two discovery methods return different things. The concept-vector method found the network's value estimate. On the win class, the fact that the mover is winning, its directions come out REAL on 2, 3, 10 of the 64, 61, 49 tested on the self-play agent, and on 3, 0, 1 of 94, 67, 80 on the auxiliary-head agent (upper bounds, as on every value-readout count). The sparse autoencoder features on the same cell come out REAL on 2, 5, 0 of 1,023, 757, 506. The method returned no handle for any move concept on any run, so every move-level handle in this paper comes from the sparse autoencoder pipeline. As a set, its win-class directions cost value-keeping play 3.7, 5.4 random-set standard deviations on the self-play agent (two seeds), and on the taught attacker of the delay world its only-move set stands 129 out (one seed). These counts are those of one layer and one recipe (Appendix F).

Necessity belongs to the set (Appendix J). Removing the features a method assigned to a fact costs value-keeping play far beyond the removal of as many random directions. On every agent most feature sets small enough to be read stand at least 3 random-set standard deviations out, while decoy sets of the same size rarely do (Table 13 in the appendix). The rear-piece rule's set at the first layer

Table 1: Three methods on one report card. The sparse autoencoder features return few but real handles on move choice at every layer, the concept vector method finds the value estimate and no move fact, and the linear probe is a reference. Rows are the self-play agent's layers per method; cells list three seeds in order. Columns: the candidates graded; the median purity margin, with the matched random slices' median in parentheses; the REAL count on each readout, with its noise count (an expected count, so a decimal) in parentheses, the first two layers' borrowed from the last layer's measured cells (Table 10). Every value-readout count is an upper bound.

method	layer	graded	purity margin (random slices')	move: REAL (noise count)	value: REAL (noise count)
SAE features	L0	3,043, 2,988, 2,292	0.124, 0.108, 0.116 (0.082, 0.071, 0.059)	17, 19, 12 (0, 0, 1.2)	0, 2, 10 (23.9, 24.6, 18.2)
	L1	3,023, 2,875, 1,900	0.121, 0.108, 0.113 (0.081, 0.069, 0.052)	17, 26, 12 (0, 0, 1.3)	6, 7, 1 (22.7, 24.5, 15.2)
	L2	2,648, 2,904, 2,401	0.116, 0.097, 0.110 (0.089, 0.075, 0.104)	18, 17, 15 (0, 0, 1.3)	22, 78, 81 (18.8, 24.2, 15.1)
concept vectors	L1	73, 63, 71	0.082, 0.073, 0.088 (dense)	none assigned	2, 3, 10 (not measured)
linear probes (reference)	L1	21, 21, 21	0.058, 0.057, 0.083 (dense)	0, 0, 0 on forced wait	reference

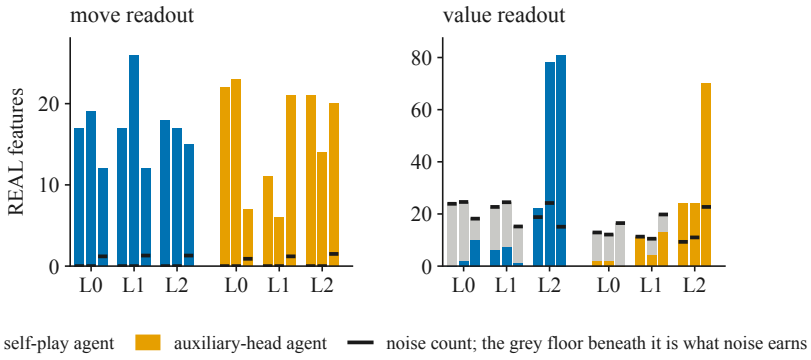


Figure 3: Few handles, and real ones. On the move readout every count stands far above its noise count in both self-play agents, every layer and seed; on the value readout the counts sit inside it at the first two layers and above it at the last. Three seeds in order per layer; a floor of no height marks a measured cell where none passed (Appendix K); Table 10 prints every value.

stands 91, 61, 62 out by seed and the sacrifice set 40, 45, 52. The forced-wait set fills more than half of the first layer, so it is read at the largest size of the ladder. There its features stand 33.3, 63.7, 12.0 random-set standard deviations out, by seed.

5.2 WHICH CONCEPTS, AND ARE THEY WHAT A PROBE WOULD SAY?

The move-side handles sit on the positions of the game's folk concepts, in a network that never saw a human game. Summed over layers and both polarities, the self-play agent holds 22, 23, 11 REAL features on forced-wait positions (most of them stalls, where no advancing move exists at all; Appendix G), 9, 30, 16 on sacrifice positions, 8, 2, 5 on the rear-piece rule, 3, 0, 2 on the entry trap, 3, 6, 2 on greedy-wrong positions and 7, 1, 3 on a move fact that no human source names (Figure 6 in the appendix). Of these, 7, 17, 15 match none of the facts in the family, a handle with no fact rather than a fact with no human name. Their purity margin clears the shuffled-label band, and no fact names them better than an arbitrary matched random slice of the same space does (the value readout's unnamed features are printed beside their noise count in Table 18 and no claim is made on them).

What passes strictly is the value itself and the move facts. The strict bar of D2 passes candidates on the win class, on forced wait, on the unnamed move fact and on sacrifice, and none of the resolved candidates on the three state-level facts, which a linear probe reads as well as it reads the move facts (Table 19 in the appendix; one seed). The linear probe's own direction is no handle either (Appendix L). Decodable and used come apart, in the direction in which a probe overstates.

They come apart in the other direction too, on a fact the network uses in play but decodes from its layers no better than from the board, and the greedy-rule pass shows it. On positions where every distance-greedy move gives away the solved value, the four networks play a greedy move on 11.3% to 17.0% of rows, against 47.1% to 59.7% on look-alike positions, those of the same value class and distance cell where greedy play is not wrong (Table 22 in the appendix; one seed). So the networks' play tells the trap from its look-alikes, and none of them holds the greedy rule of thumb. Their value heads hold a different false belief. All four price these positions, 94,159 of which 91,384 are wins, -0.24 to -0.34 against their look-alikes on the value scale and further below the table, where a value-blind fake label of the same density shows no comparable gap. The discrimination the play shows does not sit in one linear direction (Appendix N).

5.3 Does it travel?

The cross-play pool holds the positions the graded networks reach against each other, and a dictionary is extracted afresh on those activations. There the self-play agent keeps 19, 20 and 17 move-side REAL features per layer against a noise count of at most 3.5, where the agent-free pool gave 17, 17 and 18 on the same seed, and the auxiliary-head agent 17, 9 and 15 against 0 (Figure 4a in the appendix; one seed). The count survives a change of positions and of dictionary.

The folk concepts are rarer on the positions the agents actually play than on the agent-free pool. Forced wait falls from 8.8% of positions to 4.1%, and sacrifice, the mutual stall and the pseudo-dead position fall further, the four to between 0.46 and 0.039 of their agent-free rate (Table 20 in the appendix). An evaluation on played positions alone would hold too few of these positions to grade the concepts (one seed; Appendix M gives the probe's shift on the comparable facts).

On the two rule-variant worlds the same rule returns 26 to 196 handles per layer in both attackers, the taught defender and the cooperative network, above their noise counts (Figure 4b and Table 12 in the appendix). A network that knows a world carries handles on that world's own concepts; the taught defender's include the wall-building fact, which exists only under the delay rule. Self-play did not teach the worlds. The self-play attacker plays an on-schedule move on 54.3% of its positions, a base-rules agent placed on the same positions plays one on 50.5%, the taught network on 81.0% and a random mover on 9.0% (one seed; Table 26 gives every fact and the defender). The taught defender, which knows its world, holds 111 handles across its layers; the self-play defender, whose play is near random, holds 16; the two attackers hold 221 and 221 (one seed per world agent).

5.4 What does the protocol let through, and what does it miss?

Take the forced-wait features, the one move fact with measured null cells on this pool. To measure what the rule lets through there, we draw random directions in the same layer, make each fire as often as a typical REAL feature, and send it through the same tests with fresh twins and controls. On the move readout such a direction passes the whole causal rule on 0% of 490 trials in the self-play agent's forced-wait cell, on at most 0.5% in any cell of the two self-play agents and on at most 1.5% in any measured cell. On the value readout the same directions pass on 1.8% to 4.3% of trials across seeds, arms and pools, and a sparser family of directions on 5.5% (Table 14; Appendix K).

On the representation test the control decides what passes. A matched random slice is built per feature. For a forced-wait feature that fires on a given share of positions, it is a random direction of the same layer, thresholded to fire on the same share and given its own best fact by the same rule. On the self-play agents a feature beats all of its own slices far more often than chance (Section 5.1). On both taught agents' last layers it does so less often than chance, and on the table-taught agent the slices' median is twice the features' (Table 9 and Figure 5 in the appendix). A shuffled-label control certifies the dictionary in both cases; the slice separates them.

The depth facts show what a read without bars lets through. Of the self-play agent's last-layer features, 37% score better on a depth fact than on any base fact, by some margin; once that score

must also beat the feature's shuffled-label control and what distance alone reads on the depth fact, 1 of 2,648 is a depth detector, and none of the REAL features is (Appendix O).

The second question is the re-pass rate. The self-play agent's own REAL features, re-scored on a fresh draw of twins and controls, pass again on 4 of 7, 3 of 6 and 1 of 3 move-side rows and on 10 of 20, 14 of 40 and 17 of 40 value-side rows (the last layer, seeds in order). Of the 9 forced-wait features that were REAL at the first layer of the first seed, 5 passed again. A count per method replicates while a single feature's label is a label of one draw, which is why every claim above is made at the level of counts.

6 Discussion and limitations

What the findings mean. A standard sparse-autoencoder pipeline, run on a network that learned the game from self-play alone, returns a few dozen causal handles per agent out of thousands of features, and those handles are real. A dictionary is a source of candidates, not of findings. The handles sit on the game's folk strategy and on positions that no fact of ours names, and the features returned for a fact matter for play as a set. A single feature keeps its causal label on a fresh draw about half the time while the count per method holds, so a claim about what a network carries belongs at the level of counts, beside the count that noise would earn. For a validator this means two things. The control to report beside a dictionary is a matched random slice, because a shuffled-label control certifies every dictionary we graded (Section 5.4). A pass rule on the value readout admits 1.8% to 5.5% of directions with no content, so a count there needs its noise rate beside it and supports no single-feature claim. Decodable and used come apart in both directions (Section 5.2).

Limitations. The study covers one game family on boards up to 6×6, one architecture and one dictionary recipe. The two self-play agents have three seeds each, the taught agents one or two, and each world agent and cross-play run one. The fact family is a choice, so a concept outside it is graded on its nearest member or left unnamed. Twins remove distance and not every correlate of the fact inside a distance cell, so a handle is never called the network's representation of the fact. Necessity is a property of the returned set, and the set test does not show that the loss is specific to the fact whose set was removed; the matched other-fact control was not run. Our content-free directions price a direction with no content, not one that tracks a correlated real property (Appendix K). On the worlds the facts are largely predictable from the distance to the goal, and on the 5×5 pools random directions score as well on a fact as real features do, so no single feature's fact label is quotable. Whether a rule taught the agent a concept is read from its play against a base-rules agent (Section 5.3). Each appendix section states what its instrument cannot distinguish.

What comes next. On a solved world, concept discovery can be automated end to end and every step of it checked. The fact family, the one human choice left in the pipeline, can be generated from a grammar of board and table terms and filtered by the solved table. The same grammar can name the causal handles that no fact names (Peng et al., 2026). Any method that returns a direction and a detector fits the report card, and so does any validator that scores one. The benchmark carries over to any domain with an exact oracle, such as chess endgames under tablebases, other solved games and settings where a checker decides every step. Appendix Q lists the directions in full.

7 Conclusion

Exact truth makes validation claims decidable. A network that learned Chinese checkers from self-play alone carries causal handles for the concepts that players of the game name. Of the 743, 852, 736 sparse-autoencoder features per agent that fire often enough to be tested, on at least 6,000 positions, 52, 62, 39 are handles where noise would earn 0, 0, 3.8 (seeds in order). Patching one costs value-keeping play far beyond matched controls, and more so from a position where its fact is false. A feature beats all three of its own matched random directions on 0.38 to 0.65 of cases, where chance is one in four. Removing the features returned for a fact costs play far more than removing as many random directions. Four of the field's validation practices were graded on the same truth, and every one of these statements was checked against the solved table. The benchmark is released as a testbed and a protocol so that other methods can be graded on the same report card.

REPRODUCIBILITY STATEMENT

The supplementary material contains the code that produced the results and trained the agents and the dictionaries, and the result files behind the paper's numbers and tables. A runnable 4×4 example, with its solved table and one trained network, grades a network into a report card end to end. The 6×6 and 5×5 solved tables and the trained networks are too large to include.

REFERENCES

Ashton Anderson, Jon Kleinberg, and Sendhil Mullainathan. Assessing human error against a benchmark of perfection. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016. doi: 10.1145/2939672.2939803.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 2022.

Nora Belrose, David Schneider-Joseph, Shauli Ravfogel, Ryan Cotterell, Edward Raff, and Stella Biderman. LEACE: Perfect linear concept erasure in closed form. In *Advances in Neural Information Processing Systems*, 2023.

Trenton Bricken, Adly Templeton, Joshua Batson, et al. Towards monosemanticity: Decomposing language models with dictionary learning. Transformer Circuits Thread, 2023.

Aviral Chawla, Galen Hall, and Juniper Lovato. MetaOthello: A controlled study of multiple world models in transformers. In *International Conference on Machine Learning*, 2026.

Austin L. Davis, Robinson Vasquez Ferrer, and Gita Sukthankar. Exploring the limits of probes for latent representation edits in GPT models. *AI*, 7(3), 2026. doi: 10.3390/ai7030092.

Nelson Elhage, Tristan Hume, Catherine Olsson, et al. Toy models of superposition. *arXiv preprint arXiv:2209.10652*, 2022.

Nils Feldhus and Laura Kopf. Interpreting language models through concept descriptions: A survey. In *Proceedings of the 8th BlackboxNLP Workshop*, pp. 149–162. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.blackboxnlp-1.8.

Max Fomin, Elad David, and Amit LeVi. Internal-state probes read the situation, not the action: Three negative results for pre-action misalignment monitoring. *arXiv preprint arXiv:2606.30449*, 2026.

Leo Gao, Tom Dupré la Tour, Henk Tillman, et al. Scaling and evaluating sparse autoencoders. In *International Conference on Learning Representations*, 2025.

Rohan Gupta, Iván Arcuschin, Thomas Kwa, and Adrià Garriga-Alonso. InterpBench: Semi-synthetic transformers for evaluating mechanistic interpretability techniques. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks*, 2024.

Wes Gurnee, Neel Nanda, Matthew Pauly, Katherine Harvey, Dmitrii Troitskii, and Dimitris Bertsimas. Finding neurons in a haystack: Case studies with sparse probing. *Transactions on Machine Learning Research*, 2023.

Patrik Hammersborg and Inga Strümke. Reinforcement learning in an adaptable chess environment for detecting human-understandable concepts. *IFAC-PapersOnLine*, 56(2):9050–9055, 2023. doi: 10.1016/j.ifacol.2023.10.135.

Romain Harang, Jason Naradowsky, Yaswitha Gujju, and Yusuke Miyao. Tracking world states with language models: State-based evaluation using chess. *arXiv preprint arXiv:2508.19851*, 2025.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *Proceedings of EMNLP-IJCNLP*, 2019.

Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *International Conference on Learning Representations*, 2024.

Alon Jacovi and Yoav Goldberg. Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? In *Proceedings of ACL*, 2020.

Adam Kamel, Tanish Rastogi, Michael Ma, Kailash Ranganathan, and Kevin Zhu. Emergent world beliefs: Exploring transformers in stochastic games. *arXiv preprint arXiv:2512.23722*, 2025.

Bigyan Karki. Evaluating AlphaZero in a strongly solved game. Master's thesis, University of Alberta, 2024.

Adam Karvonen. Emergent world models and latent variable estimation in chess-playing language models. In *Conference on Language Modeling*, 2024.

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

Adam Karvonen, Can Rager, Johnny Lin, Curt Tigges, Joseph Bloom, David Chanin, Yeu-Tong Lau, Eoin Farrell, et al. SAEBench: A comprehensive benchmark for sparse autoencoders in language model interpretability. In *International Conference on Machine Learning*, 2025.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viégas, and Rory Sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). In *International Conference on Machine Learning*, 2018.

Brent Kong, Tejas Ram, and Tony Yue Yu. AlphaZero in sparsely rewarded games: Limits and auxiliary supervision. *arXiv preprint arXiv:2607.08984*, 2026.

Anton Korznikov, Andrey Galichin, Alexey Dontsov, Oleg Rogov, Ivan Oseledets, and Elena Tutubalina. Sanity checks for sparse autoencoders: Do SAEs beat random baselines? *arXiv preprint arXiv:2602.14111*, 2026.

Girish A. Koushik, Diptesh Kanojia, and Helen Treharne. Decodable but misrouted: Sparse features uncover a readout gap in vision-language models for harmful meme detection. *arXiv preprint arXiv:2609.18860*, 2026.

Georg Lange, Rick Goldstein, Kamal Maher, and Katherine Dearstyne. Cross-layer transcoders are incentivized to learn unfaithful circuits. In *ICML 2026 Workshop on Mechanistic Interpretability*, 2026.

Andrew Lee, Fernanda Viégas, and Martin Wattenberg. Tensor product representation probes reveal shared structure across linear directions. *arXiv preprint arXiv:2605.09967*, 2026.

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

Daisy Lin, Brenden Lake, and Wei Ji Ma. Limitations in planning ability in AlphaZero. In *Conference on Cognitive Computational Neuroscience*, 2024.

David Lindner, János Kramár, Sebastian Farquhar, Matthew Rahtz, Thomas McGrath, and Vladimir Mikulik. Tracr: Compiled transformers as a laboratory for interpretability. In *Advances in Neural Information Processing Systems*, 2023.

Semyon Lomasov, Judah Goldfeder, Mehmet Hamza Erol, Matthew So, Yao Yan, Addison Howard, Nathan Kutz, and Ravid Shwartz Ziv. Exploring human-AI conceptual alignment through the prism of chess. In *Advances in Neural Information Processing Systems, Creative AI Track*, 2025.

Charles Lovering, Jessica Zosa Forde, George Konidaris, Ellie Pavlick, and Michael L. Littman. Evaluation beyond task performance: Analyzing concepts in AlphaZero in Hex. In *Advances in Neural Information Processing Systems*, 2022.

Daming Luo. When are sparse feature interventions actually localized? matched evaluation for SAE-based safety control. *arXiv preprint arXiv:2607.10226*, 2026.

Jacob Lysnæs-Larsen, Marte Eggen, and Inga Strümke. Probing the probes: Methods and metrics for concept alignment. *arXiv preprint arXiv:2511.04312*, 2025.

Aleksandar Makelov, Georg Lange, Atticus Geiger, and Neel Nanda. Is this the subspace you are looking for? An interpretability illusion for subspace activation patching. In *International Conference on Learning Representations*, 2024.

Anthony Marasco and Sidney Givigi. Can AlphaZero master human concepts in tic-tac-toe? In *IEEE International Conference on Systems, Man, and Cybernetics*, 2025. doi:10.1109/SMC58881.2025.11343571.

Thomas McGrath, Andrei Kapishnikov, Nenad Tomašev, Adam Pearce, Martin Wattenberg, Demis Hassabis, Been Kim, Ulrich Paquet, and Vladimir Kramnik. Acquisition of chess knowledge in AlphaZero. *Proceedings of the National Academy of Sciences*, 119(47):e2206625119, 2022. doi: 10.1073/pnas.2206625119.

Maxime Méloux, Tiago Pimentel, François Portet, and Maxime Peyrard. Validating causal abstraction metrics on simulated complex systems. *arXiv preprint arXiv:2607.00267*, 2026.

Aðalsteinn Pálsson and Yngvi Björnsson. Unveiling concepts learned by a world-class chess-playing agent. In *Proceedings of IJCAI*, pp. 4864–4872, 2023.

Aðalsteinn Pálsson and Yngvi Björnsson. Empirical evaluation of concept probing for game-playing agents. In *ECAI*, pp. 874–881, 2024. doi:10.3233/FAIA240574.

Gonçalo Paulo and Nora Belrose. Sparse autoencoders trained on the same data learn different features. *arXiv preprint arXiv:2501.16615*, 2025.

Gonçalo Paulo, Alex Mallen, Caden Juang, and Nora Belrose. Automatically interpreting millions of features in large language models. In *International Conference on Machine Learning*, 2025.

Kenny Peng, Rajiv Movva, Jon Kleinberg, Emma Pierson, and Nikhil Garg. Position: Use sparse autoencoders to discover unknowns. In *International Conference on Machine Learning*, 2026.

Orion Reblitz-Richardson. Calibrating interpretability instruments before trusting their verdicts. *arXiv preprint arXiv:2609.14754*, 2026.

Jonathan Richens, Tom Everitt, and David Abel. General agents need world models. In *International Conference on Machine Learning*, 2025.

Lisa Schut, Nenad Tomašev, Thomas McGrath, Demis Hassabis, Ulrich Paquet, and Been Kim. Bridging the human–AI knowledge gap through concept discovery and transfer in AlphaZero. *Proceedings of the National Academy of Sciences*, 2025. doi:10.1073/pnas.2406675122.

Yang Sheng and Jie Fu. Circuit claims depend on what is extracted and how it is compared. *arXiv preprint arXiv:2607.18921*, 2026.

Nathan R. Sturtevant. On strongly solving Chinese checkers. In *Advances in Computer Games*, 2019. doi:10.1007/978-3-030-65883-0_13.

Shiwei Tan, Yusong Zhao, Weiyi Qin, Wentian Wang, Jacob Feldman, Lazaros K. Gallos, Paul B. Kantor, Vladimir Menkov, and Hao Wang. Interpretable GOHR agents via sparse autoencoders. *arXiv preprint arXiv:2607.25132*, 2026.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, et al. Scaling monosemanticity: Extracting interpretable features from Claude 3 Sonnet. Transformer Circuits Thread, 2024.

Shunsuke Terauchi, Takaharu Kubota, and Kiminori Matsuzaki. Using strongly solved Mini2048 to analyze players with N-tuple networks. In *Technologies and Applications of Artificial Intelligence*, volume 2075 of *Communications in Computer and Information Science*, pp. 165–178, 2024. doi: 10.1007/978-981-97-1714-9_14.

Devesh Tiwari, Camille Davis, Shivank Sinha, and Talia Weaver. Decodability is not causality: Dissociating probe readouts from behavioral drivers via SAE decomposition. *arXiv preprint arXiv:2609.18080*, 2026.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*, 2024.

Constantin Venhoff, Anisoara Calinescu, Philip Torr, and Christian Schroeder de Witt. SAGE: Scalable ground truth evaluations for large sparse autoencoders. *arXiv preprint arXiv:2410.07456*, 2024.

Tony T. Wang, Adam Gleave, Tom Tseng, Kellin Pelrine, Nora Belrose, Joseph Miller, Michael D. Dennis, Yawen Duan, et al. Adversarial policies beat superhuman Go AIs. In *International Conference on Machine Learning*, 2023.

Zhengxuan Wu, Aryaman Arora, Atticus Geiger, Zheng Wang, Jing Huang, Dan Jurafsky, Christopher D. Manning, and Christopher Potts. AxBench: Steering LLMs? even simple baselines outperform sparse autoencoders. In *International Conference on Machine Learning*, 2025.

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

Fred Zhang and Neel Nanda. Towards best practices of activation patching in language models: Metrics and methods. In *International Conference on Learning Representations*, 2024.

A TERMS

Each term below is defined where the paper first uses it; the table restates each in plain words.

Table 2: The paper's terms, each with the section that defines it and a plain reading.

term	defined in	plain reading
fact	Section 3	a yes-or-no property of a position, computed exactly from the solved table and the board
fact family	Section 3	the fixed list of facts the paper grades against (Table 8)
move fact	Section 3	a fact about which moves keep the solved value; graded on the move the network chooses
state fact	Section 3	a fact about the position itself; graded on the network's value head
agent-free pool	Section 3	the positions of games played by three generators with no trained agent
cross-play pool	Section 4	the positions the graded networks visit when they play each other
scoring positions	Section 3	held-out positions where the mover has a win or a draw to keep
distance cell	Section 3	the pair of both sides' distances to goal; the stratum every matched control draws from
taught agent	Section 3	a network trained by supervision on the solved table; a ceiling and a control
candidate	Section 4	a unit direction in one layer's activations with a detector that says where it fires
purity margin	Section 4, D1	among the positions where a feature fires, the share on which its fact holds, minus what distance alone would give
shuffled-label control	Section 4, D1	the feature's scores permuted within each distance cell: its firing density and distance profile, nothing else
matched random slice	Section 4, D1	a random direction of the same layer, thresholded to the feature's firing rate and given its own best fact
twin	Section 4, D2	a position from the same distance cell as the target where the fact is false (or, for D3, true)
matched controls	Section 4, D2	40 random pushes of the same size as the candidate's

Table 2: The paper's terms, each with the section that defines it and a plain reading.

term	defined in	plain reading
value-blind fake	Section 4, D2	a direction fitted to the fact's labels permuted within each distance cell: the fact's density and none of its content
strict bar	Section 4, D2	the candidate's shift beats every control's and the controls' mean by three of their standard deviations
per-row rule	Section 4, D2	the strict bar together with the value-blind fake rung and, on a state fact, the value-gradient direction rung
REAL	Section 4, D3	passes the per-row rule, meets the D3 condition and fires on at least 6,000 held-out scoring positions
random-direction rung	Appendix J	the D3 swap repeated with 10 random directions on the same pairs; its mean and spread are the band of Section 4 that the effect must clear
content-free direction	Section 4	a random direction fired at the density of a REAL feature and sent through the whole chain
noise count	Section 4	the rate at which content-free directions come out REAL, times the eligible latents of the cell
set-level test	Section 4	every direction a method assigned to a fact removed at once, against random sets and decoy sets
decoy set	Section 4	a set of value-blind fake directions of the same size as the feature set
race class	Section 4	the REAL latents whose fact is a value class (who is winning) or the negation of a fact; sorted out before named and unnamed
unnamed	Section 4	a REAL latent that is not in the race class, names no fact better than its own matched random slices, and clears the shuffled-label controls' band
cross-play exam	Section 4	each arm graded a second time, by the same grader, on the cross-play pool
greedy-rule pass	Section 4	how often the network plays a greedy move where every greedy move gives away the value, against look-alike positions
look-alike positions	Section 4	positions of the same value class and distance cell where greedy play is not wrong
conversion depth	Section 4	the exact number of plies to the decision under best play
delay world, cooperative world	Section 3	the two rule variants of the 5×5 board, each with its own solved table
on-schedule move	Section 3	a move that realizes the solved table's value
testbed	Section 1	the material: four exact tables, the fact family, the pools, the agents, the two worlds
grading protocol	Section 1	the scoring: the representation, causal, dependence and set-level tests with their controls and the null program
report card	Section 1	the grading protocol's output, one per method and agent
causal handle	Section 1	a REAL candidate: a direction that moves the network's behaviour as its fact predicts

B USE OF LARGE LANGUAGE MODELS

We used generative AI tools as general-purpose tools to assist with implementing experiment code and analysis scripts, analysing results, generating plots, and drafting and polishing sections of the paper, and we used the conference's paper-feedback tool once. All AI-assisted work was verified by the authors. The research idea, the experimental design and the conclusions are the authors'. The authors take full responsibility for all content.

C POSITION POOLS AND THE TRAIN/HELD-OUT SPLIT

What it is. Every instrument in this paper scores positions, and this section says where they come from and which of them any number is read on. An *agent-free pool* holds the positions of complete games played from the start position by three generators that involve no trained agent, with duplicates removed. We build one per seed on the 6×6 board and one per world on the 5×5 board, and

every agent of a seed is graded on that seed's pool. The *cross-play pool* holds every position visited in a round-robin among the graded agents and is described with the cross-play exam (Appendix M).

The pool. The optimal generator picks uniformly among the moves that keep the solved value, and in a lost position uniformly among the legal moves, after a random legal prefix of at most 6 plies that gives the games variety. The random generator picks uniformly among legal moves, and the greedy generator takes the move that most reduces the mover's own distance to goal. Each generator plays until it has contributed its share of a target of 6,600,000 unique positions on the 6×6 board (500,000 on a 5×5 world), with games capped at 3,000 plies (200 on the worlds). The three sets are merged, the generator that first emitted a position is recorded, which lets a number be read per generator, and the shares that result are measured, not taken from the targets; on the first seed's pool they are 32% optimal, 38% random and 30% greedy (one seed). Every fact is then evaluated on every position. Each position also receives its *distance cell*, the pair of the mover's and the opponent's single-agent distances to goal, which is the stratum every matched control in the paper uses.

The split. Games, not positions, are split. One seeded coin sends each game to the training half with probability 0.5, and a position inherits the side of its game. The training half fits thresholds, probes and erasures, and the held-out half is the only side on which a number is read. Within it, the *scoring positions* are the playable positions, where the mover has a legal move and, on the 6×6 board, a solved value that is not a loss. Keeping the value is defined only where there is a win or a draw to keep. On a world the condition is that the world's value is defined and some move keeps it. D1 reads every held-out position, the lost ones included; the causal tests read the scoring positions only, so a candidate assigned a loss fact is never tested by D2 or D3.

Safeguards. The split by game rules out the same position sitting in both halves. Reporting the realized composition rules out an assumed mix, since random games yield more unique positions per game than optimal ones. On the cross-play pool, test-side positions that lie in an agent-free pool's training half are dropped before anything is fitted or scored, because a taught agent trained on that half. On a world, the base control, a self-play agent of the unmodified game, started some of its training games from positions of the pool; those positions are dropped from every agent's scoring positions.

What it cannot distinguish. The pools hold positions that games visit and are no sample of the state space, so every number is about positions that arise in play, and positions no generator reaches are absent. A finished game's last position, where the side to move has already lost, is in the pool as a held-out position of value loss and is never among the scoring positions. Every base rate, and with it every matched control, depends on the mix of generators, and the equal-thirds target is a convention: another mix is another pool. The three seeds on the 6×6 board are graded on three different pools, so a difference between seeds includes a difference between pools. A position reached by several games belongs to the first game that reached it, so a later game's positions can straddle the two halves, and standard errors clustered by game are for that reason lower bounds on the uncertainty. The split guarantees that whole games sit on one side and that no position is in both halves. It cannot keep consecutive positions apart, because in any board game consecutive positions differ by one move, so a held-out position can have a training position next door. We measure how many such neighbouring positions the split leaves and what they are worth to a probe.

Neighbouring positions. The split by game leaves a held-out position one move from a training position on 18% of the held-out rows (one seed). Only the taught agents trained on the pool's training half, so only they can gain from a neighbour; the self-play agents never saw the pool. The size, measured on the table-taught agent against the self-play agent, is small. On the rows of the distance-greedy generator, which hold nearly all such positions, a linear probe reads a fact at most 0.02 better with a neighbour than without, and 0.04 on one fact over all rows; three of the four differences sit outside the chance range of label shuffles, so they are small and real (Table 3).

D SOLVED TABLES

What it is. A solved table gives every position of a game its exact value under best play, and every claim in this paper about what a network knows is graded against one. We use four. The base tables

Table 3: What a neighbouring position is worth (one seed). For each fact: a linear probe on the table-taught agent scores held-out positions that have a training position one move away against those that do not; the same gap for the self-play agent, which never trained on the pool, is subtracted, so the number is the help the taught agent gets from the neighbour. First on the distance-greedy generator's rows, which hold nearly all such positions, with the range the number takes when the fact's labels are shuffled 200 times and whether it sits inside that range; then over all rows.

fact	a neighbour's help to the taught agent, greedy-generator rows	chance range from 200 label shuffles, low	high	inside the chance range	the same help, all rows
`forced_wait`	−0.0023	−0.0017	0.0014	no	−0.0090
`mover_wins`	0.0036	−0.0012	0.0009	no	−0.0049
`only_move`	0.0189	−0.0027	0.0025	no	0.0138
`sacrifice`	0.0043	−0.0045	0.0043	yes	0.0406

of the 6×6 and 5×5 boards with six pieces per side label each position a win, a draw or a loss for the player to move, or illegal. The table of the delay world gives the number of plies until the attacker's goal is full when the attacker minimises and the defender maximises, and marks the positions where the defender can prevent it forever. The table of the cooperative world gives the length of the shortest joint plan by which both sides swap camps.

Procedure. The base rules follow Sturtevant (2019). Pieces step to adjacent cells or make chained jumps over any piece, there is no passing and no capture, and a player wins when the opponent's start area is full and holds at least one of the player's pieces. Two kinds of position are removed from the game graph, count as illegal, and may not be entered by a move. The first is a position in which the mover's goal is already full with one of the mover's pieces in it. A player wins only by making the move that fills the goal, so such a position can arise only when the opponent filled the last cell, and that move is not allowed. The second is a position in which a goal can no longer be filled: the cell at the tip of the goal triangle is empty and the four cells along its two outer edges are held by the player whose start area it is, so no piece can reach the tip. Repetition is a draw in play and is not a value of the table. The base solve is retrograde analysis to a fixpoint. A state is a win if some successor is a loss for the opponent, a loss if every successor is a win for the opponent, and a draw if it is still unlabelled when nothing changes. The packed table of the large board stores no draw code, and a state left unlabelled at the fixpoint is read as a draw. On the 6×6 board the solver stores one state per symmetry class under the swap of players and the mirror, which leaves 578,944,875,600 stored states for 2,313,100,389,600 positions. The delay world is solved by relaxation until a labelling pass changes nothing, the attacker taking one plus the minimum over successors and the defender one plus the maximum; the cooperative world takes one plus the minimum for either mover. The agents of a world are not trained on the raw value d but on a target built from a hidden deadline B, drawn uniformly from a range that is read from the pool before any training, so that the network trains on a bounded target like the base game's win or loss rather than on a count of plies. The target is $2\,P(B \geq d) - 1$ for the attacker and its negative for the defender.

Safeguards. The rules were fixed on one small board and were then required to reproduce the published counts of wins, draws and illegal states on the three 7×7 boards with one to three pieces without further change, which they do. The 6×6 table is too large for a check of every state, so three partial checks stand in. The count of illegal states equals a closed-form enumeration, which needs no solver, and the illegal predicate is thereby checked independently of the solve. A second, independently written move generator agrees with the table on 500,000 uniformly sampled states and on as many pool states, which rules out a transcription error in either. The labels within 7 plies of a terminal state pass a least-fixpoint check. The labels near the end of a game are therefore right. The world tables are checked cell by cell against a brute-force search on the 4×4 board, and the 5×5 tables are the same code at the larger size.

What it cannot distinguish. A table is exact for its rules, and a different reading of a rule is a different game; the agents, trained under our rules, are graded against the table of the game they play. The 6×6 table is checked by counts and on samples, so a wrong label deeper than the horizon

of the fixpoint check, or a second fixpoint confined to the draw region, would pass every check we ran. An error shared by both move generators would be invisible. The world tables at 5×5 rest on the gates at 4×4 and on shared code, because no independent solution exists to compare with. A table says what the value is and says nothing about how hard it is to compute or how a network could represent it.

The check of the large board's table. Table 4 gives the table's counts and the outcome of every check that ran, and Table 5 sets the 6×6 row beside the published one (Sturtevant, 2019). The two differ, ours minus the published row, by 80,757,600 illegal states, $-31{,}173{,}059$ wins and $-18{,}411{,}482$ draws, the same total; the 5×5 row with six pieces differs in the same way. The difference is one detail of the second removal rule, the sealed goal. Our table files a position under illegal as soon as the tip of the goal triangle is empty and its two outer edges are held by the player whose start area it is; the published counts apply a narrower test of the same idea, whose form we do not know, so a position our table files under illegal the published row files under a win or a draw. Two published rows of smaller boards, the 4×4 board with six pieces and the 7×7 board with four, also differ from our counts; our two independently written solvers agree with each other on both, and the three smallest 7×7 boards match the published rows exactly. The table we grade against rests on none of those rows. The table's author has since confirmed that his own solvers reproduce our 4×4 counts and that the published row appears to contain a copy-and-paste error (personal communication; the note on his publications page, `https://www.movingai.com/pubs.html`, says the same and that his 6×6 row was verified with a second solver).

Table 4: Checks of the large board's solved table (each sample was drawn with one seed): its counts, where a win is a position won by a given player, not by the mover; the positions of a uniform sample and of a pool sample whose stored value disagrees with a re-derivation one ply deep; and the violations of the depth witness.

side	positions	wins	draws	illegal	mismatches, uniform sample	mismatches, pool sample	witness violations
6	2,313,100,389,600	1,152,969,765,114	5,181,409,122	1,979,450,250	0	0	0

Table 5: The 6×6 row beside the published one. Per count: our table (one seed: the table has none), the published row, and ours minus the published row. Wins are positions won by a given player.

count	our table	the published row	ours minus published
draws	5,181,409,122	5,199,820,604	$-18{,}411{,}482$
illegal states	1,979,450,250	1,898,692,650	80,757,600
wins	1,152,969,765,114	1,153,000,938,173	$-31{,}173{,}059$

E AGENTS AND TRAINING

What it is. This section describes the networks whose internals we examine. A self-play agent is trained by self-play with tree search and never sees the solved table; it is the subject of the study. A second self-play arm adds an auxiliary head trained on sacrifice, only_move, two exact facts labelled from the table. It is a reference that has seen the table, intended as the known positive for the state readout, and it is never treated as a subject of discovery. A *taught agent* has the same architecture and is trained by supervision on the solved table, with the solved value as its value target and the uniform distribution over value-keeping moves as its policy target. A variant lets a teacher re-weight each round toward the positions the student gets most wrong. Taught agents are ceilings that know the table, and they are controls, never subjects. On the 5×5 worlds we train a self-play agent under each world's rules, a taught agent on each world's table, and a base control, a self-play agent of the unmodified game that is graded on the worlds' positions.

Procedure. The network reads the board as planes in the mover's frame, passes through a linear stem of width 512 and 3 residual blocks, and ends in a policy head over moves and a value head. The outputs of the blocks are the layers L0, L1 and L2 from which candidates are extracted. Each

iteration of self-play runs 256 games with 128 search simulations per move and then takes 400 optimiser steps on a buffer of the last 200,000 positions. A game that ends by repetition gives both sides the value target -0.5, which keeps the agent from settling into repetition draws, and the cap on game length rises with the median length of finished games, so games are not cut off as play lengthens. Every run has a fixed budget of 1,000 iterations with a snapshot every 50. The graded network is chosen afterwards by a playoff that never consults the table. Snapshots are walked in order, each plays 128 games against the standing champion, and it replaces the champion if its share of the score exceeds 0.55. The auxiliary head is trained jointly at a loss weight of 0.25. A taught agent trains only on the training half of the agent-free pool and stops when its knowledge of the value and its accuracy on a fixed probe set are both flat. On the worlds, a share 0.5 of the training games start from positions of the pool's training half, so that self-play visits the kind of positions the pool holds, and the reward is whether the attacker arrives within a hidden deadline.

Safeguards. A fixed budget and a playoff that is blind to the table rule out a choice of stopping time or snapshot that is informed by the truth. The playoff compares weights and refuses to play the same weights twice. Because a taught agent draws its lessons from the training half alone, it is never trained on a position it is graded on. The base control on the worlds rules out crediting a world's rules with behaviour that the unmodified game already induces.

What it cannot distinguish. One network is one draw of its recipe. Three seeds bound the spread without establishing the typical case, and the seeds do differ on the counts our instruments produce, which is why we report ranges over seeds. The champion is the winner of a playoff and need not be the strongest network of its run by any measure taken from the table, so the graded network is a competent one, not the best one. The auxiliary head is trained on two facts and says nothing about the others. A taught agent knows the solved value and the value-keeping moves and is never told how deep a win lies, so a depth fact it detects was not taught. The world agents and the base control have one seed each, and the self-play agent of the delay world plays little better than the base control, so its results are read beside the taught agent's.

How well the agents play. Table 6 gives, for every graded network, the share of held-out scoring positions on which its chosen move keeps the solved value, beside the share a random legal move keeps on the same positions.

Table 6: Every graded network's value-keeping rate on the held-out scoring positions, beside the random-move floor of the same positions. One row per graded run, one seed each; the rate is the ladder's first rung (Appendix J).

agent	seed	value-keeping rate	random-move floor
cooperative, taught	0	0.779	0.216
delay, self-play attacker	0	0.543	0.090
delay, self-play defender	0	0.358	0.264
delay, taught attacker	0	0.810	0.090
delay, taught defender	0	0.751	0.264
self-play	0	0.896	0.675
self-play	1	0.892	0.680
self-play	2	0.893	0.678
auxiliary-head	0	0.883	0.675
auxiliary-head	1	0.897	0.680
auxiliary-head	2	0.892	0.678
curriculum-taught	0	0.951	0.675
curriculum-taught	1	0.949	0.680
table-taught	0	0.960	0.675
self-play	0	0.906	0.629
auxiliary-head	0	0.893	0.629
curriculum-taught	0	0.914	0.629
table-taught	0	0.922	0.629

The auxiliary head, checked. Table 7 compares the trained auxiliary head with an untrained head and with a ridge linear probe on the same layer. The trained head reads its two facts far above the

untrained head and no better than the ridge linear probe, so the head learned its facts but added nothing a probe could not already read from the layer. The head changed neither of the two counts we can compare across the arms: the auxiliary-head agent returns 54, 43, 48 handles on move choice where the self-play agent returns 52, 62, 39, and its features beat all of their own matched random slices on 0.43 to 0.64 of cases against 0.38 to 0.65 (seeds in order; every layer and seed). At three seeds there is no measurable difference between the two arms.

Table 7: The auxiliary head of the auxiliary-head agent on its two facts (one seed): the fact's base rate, and the balanced accuracy on held-out positions of the trained head, of an untrained head of the same shape read the same way, and of a ridge linear probe on the same layer.

fact	base rate	the trained head	an untrained head	a ridge probe on the same layer
`only_move`	3.37%	0.724	0.541	0.722
`sacrifice`	0.702%	0.720	0.522	0.823

F CANDIDATE DIRECTIONS

What it is. A candidate is a unit direction in one layer's activations together with a score per position, from which a threshold and a polarity are fitted. We obtain candidates in three ways. The latents of a sparse autoencoder (Gao et al., 2025), trained per layer on the network's activations over the whole pool, are the candidates the paper grades. Concept vectors (Kim et al., 2018) are directions that separate sets of positions built from the network's own value and policy signals, so they never see the table; they are extracted at layer L1. A linear probe is fitted per fact by ridge regression, and because it is fitted to the facts themselves it is a reference row in every table and is never counted as a discovery.

Procedure. The sparse autoencoder normalises the activations, encodes them linearly with a rectifier, keeps the 16 largest latents per position and decodes linearly with unit-norm columns. It has 8 latents per input dimension and is trained on the reconstruction error for 4 epochs at batch size 4,096 and learning rate 1e-3. A latent's direction is its decoder column mapped back to the raw activation space, and its score is its sparse code. A latent enters causal grading when its code is nonzero on at least 100 positions of the pool. It enters D1 when it fires at least 1,000 times on the held-out scoring positions (500 on the worlds), which is the count the scorer needs for a proportion to be estimable. The position sets a concept vector separates are built from the network's own signals in three ways. They are the top tenth of the value head's outputs against the bottom tenth, the top tenth against a random sample, and the top tenth of the policy's confidence against its bottom tenth. Concept vectors are ℓ_1-regularised logistic separations fitted over a grid of sparsity levels and 8 subsamples each, and near-duplicates are merged when the absolute cosine exceeds 0.95. A linear probe's direction is the difference of the two class weight vectors of a closed-form ridge fit with penalty 1e-2. Every candidate is then assigned the fact it detects best. Thresholds and both polarities are swept on the training half, and the best balanced accuracy on the held-out half names the fact. That choice reads the held-out half. Had it been made on the training half instead, 2 of 73 REAL features of the self-play agent's first seed would carry a different fact, and 2% to 10% over its three seeds and the auxiliary-head agent's first, each a near-tie between two facts in balanced accuracy. The purity margin chooses its fact on the training half and is unaffected.

Safeguards. A hash of the network is stored with its dictionary and checked at every reload, so a network is never graded through another network's dictionary. The self-play agent's nine dictionaries explain 0.83 to 0.89 of the variance of their layer's activations with 0.12 to 0.29 of their latents dead, and over all graded dictionaries the ranges are 0.67 to 0.98 and 0 to 0.71, so the few handles do not come from a dictionary that failed to fit its layer. The network's loss under reconstruction through the dictionary was not measured. Every number from the dictionary is read against the references of D1 (Appendix H), and the content-free directions of Appendix K measure what a direction without content earns under the full causal rule.

What it cannot distinguish. The dictionary is one decomposition at one fixed setting, and a feature the sparse autoencoder did not learn, or split across several latents, is invisible to every instrument downstream. Keeping a fixed number of latents per position forces that count on positions that hold fewer or more concepts. The sparse autoencoder and the concept vectors are fitted on the activations of every position, the held-out half included, but no fact label reaches them; a latent is shaped by the activations it is later scored on and by nothing it is graded against; only the linear probes, which are fitted to the facts, are confined to the training half. The rules that admit a latent remove rare latents before any grading, so a rare true feature is unresolvable and is not shown to be absent. The concept vector method was run at one layer with one recipe, and the null program priced no densely firing direction, so its REAL counts on the value readout are upper bounds like every other count there. The dictionaries of different agents differ in how many latents are usable, and a raw count compared across agents mixes the agent with its dictionary; we report both denominators.

G THE FACT FAMILY

What it is. A fact is an exact yes-or-no property of a position, computed from the solved table and the board and never from a network. It is the target a candidate is scored on in D1, the label from which twins are built in D2 and D3, and the population on which behaviour is read on the worlds. The fact family has three blocks: the base facts of the 6×6 board, which comprise the value classes, a set of strategic facts and eight parametric families; the facts native to each 5×5 world; and the depth facts of Appendix O. Table 8 defines each one in words.

Procedure. Every fact is a boolean rule over statistics of the position, such as its solved value, the numbers of legal and of value-keeping moves, the best one-move advance of each side, whether the greedy moves keep the value, and similar quantities. A parametric family, such as "the best one-move advance is at least j", enters at the smallest parameter whose base rate on the training half is at most 0.15. The choice is made once per pool from that pool's training half, and the cross-play pool reuses the choices of the agent-free pool. Each fact has a readout class that fixes the form of the causal tests. The move facts (only_move, forced_wait, greedy_wrong, straggler_critical, premature_entry_trap, nd_blunder, sacrifice, sacrifice_deep, sacrifice_soon, dual_threat) are graded on the chosen move, and every other fact is a state fact graded on the value head. For the facts mutual_stall, pseudo_dead, draw, no twin construction exists, so no candidate is assigned to them and they are scored by D1 only. Each fact also carries a naming tag that records whether people had a name for the concept before this work. The tag is folk for the lore of the game, ML for the literature on games and game-playing programs, value for the value classes, and unnamed where we found no prior source.

Three terms used in the table. Each cell of the board has one of four types, given by the parities of its two coordinates, and the goal holds a fixed number of cells of each type. The *slide debt* of the mover is the sum over the four types of the excess of its pieces on that type over the goal's cells of that type. It is a lower bound on the slides its pieces still owe, and `on_type` holds when the debt is zero. The *rear piece* is the mover's piece with the largest slide-only distance to the goal, a breadth-first distance that ignores jumps; when two pieces tie, the first in cell order is taken. A *pseudo-legal* move is one the move generator produces on the current occupancy, with no check of the resulting position against the solved table.

Controls. The distance null of a fact is the best detector that can be built from the distance cell alone, and it rules out reading as a concept what the race to the goal already predicts. The value-blind fake rules out a candidate that reads a fact's density and distance profile and nothing else. Fixing parameters by a rate rule on the training half rules out a threshold chosen after the numbers were seen. Exclusions are stated with their reasons and are never silent. Three forms of a free-runner fact, a piece that can advance with nothing in its way (chess's passed pawn), are left out because distance alone predicts them almost perfectly.

What it cannot distinguish. The fact family is a choice. A concept outside it is graded on its nearest member or is left unnamed. The facts are exact and are nevertheless written in the vocabulary of the race (advances, distances, moves that keep the value), so many are strongly predictable from distance. The distance null and the value-blind fake measure how much of a fact distance explains,

and on a fact that distance predicts well a feature has less left to detect. The meaning of a parametric family shifts with the pool on which its parameter was chosen, so its threshold on one pool is not its threshold on another. The naming tags are a taxonomy with stated ambiguities and are not a measurement. Forced wait holds on every stall, a position in which the mover has no advancing legal move at all, and stalls are between 82% and 83% of the held-out forced-wait positions on the three agent-free pools and 52% on the cross-play pool; a feature graded on forced wait is therefore graded mostly on stalls (Appendix L gives the handles' side).

Table 8: The fact family. Every fact the paper grades against: the readout its causal tests use (move, value, or a dash where the fact is scored by D1 only), its definition and its naming tag.

fact	readout	definition	tag
`att_forced_wait`	move	In the delay world with the attacker to move: no advancing move is on schedule and some non-advancing move is (forced wait under the delay table).	folk
`att_greedy_inf`	move	In the delay world with the attacker to move: an on-schedule move exists and some distance-greedy move leads to a position the defender can hold for ever, so the greedy move walks into a permanent block. The distance-greedy moves are the legal moves whose own distance change is the smallest, ties included, permanent-block moves among them.	unnamed
`att_greedy_wrong`	move	In the delay world with the attacker to move: an on-schedule move exists and no distance-greedy move is on schedule (greedy wrong carried to the delay table). The distance-greedy moves are the legal moves whose own distance change is the smallest, ties included, permanent-block moves among them.	folk
`att_only_move`	move	In the delay world with the attacker to move: exactly one move is on schedule.	ml
`att_sacrifice`	move	In the delay world with the attacker to move: every on-schedule move retreats.	folk
`att_stalled`	value	In the delay world with the defender to move: the attacker's best one-move advance is at most zero on this occupancy, a one-sided mutual stall read from the board alone. Defined only where the defender has an on-schedule move.	folk
`boost_k2`	value	The opponent's best one-move advance exceeds its advance on the board with the mover's pieces removed by at least k cells (k = 2): the mover's pieces extend the opponent's reach. On that second board a jump chain may pass through the mover's cells but may not end on one.	folk
`borrowed_d2`	value	The mover's best one-move advance exceeds its advance on the board with the opponent's pieces removed by at least d cells (d = 2): reach borrowed from the opponent's pieces. On that second board a jump chain may pass through the opponent's cells but may not end on one.	unnamed
`breakable_d2`	value	Some single pseudo-legal move of the opponent reduces the mover's best one-move advance by at least d cells (d = 2): the mover's reach can be broken.	unnamed
`coop_forced_wait`	move	In the cooperative world: no advancing move is on schedule and some non-advancing move is.	folk
`coop_only_move`	move	In the cooperative world: exactly one move is on schedule.	ml
`coop_sacrifice`	move	In the cooperative world: every on-schedule move retreats, raising the mover's own single-agent distance to goal.	folk

fact	readout	definition	tag
def_only_move	move	In the delay world with the defender to move: exactly one move realises the maximum delay.	ml
denial	move	In the delay world with the defender to move: some on-schedule defender move reduces the attacker's best one-move advance by at least one (the base denial fact with the defender as the denier).	folk
denial_d1	value	Some value-keeping move of the mover reduces the opponent's best one-move advance by at least d cells (d = 1).	folk
draw	–	The solved value for the player to move is a draw.	value
dual_threat	move	The mover wins and at least two distinct moves lead to a win within 2 further plies, a fork. After the mover's move the opponent is to move, and an odd depth there would be the opponent's win, so the successors counted are those at depth 0 or 2.	ml
forced_wait	move	The mover is winning or drawing, no advancing move keeps the value, and some non-advancing move does.	folk
forfeitable_draw	value	The position is a draw and the mover has a legal move whose successor is a win for the opponent: a draw the mover can throw away.	value
greedy_wrong	move	The mover is winning or drawing, a value-keeping move exists, and no distance-greedy move (a legal move with the best one-move advance, ties included) keeps the value.	folk
jump_asym_a2	value	The two sides' single-agent distances to goal differ by at most one, and the mover's best one-move advance exceeds the opponent's by at least a cells (a = 2).	unnamed
long_reach_j3	value	The mover's best one-move advance is at least j cells (j = 3 on the 6x6 board).	ml
lose_deeper_7	value	The mover loses and the loss lies deeper than K = 7 plies.	value
lose_in_1	value	The mover loses and the loss comes within 1 ply of best play, which holds only on a finished game, where the side to move has already lost.	ml
lose_in_3	value	The mover loses and the loss comes within 3 plies of best play.	ml
lose_in_5	value	The mover loses and the loss comes within 5 plies of best play.	ml
mover_loses	–	The solved value for the player to move is a loss.	value
mover_wins	value	The solved value for the player to move is a win.	value
mutual_stall	–	The best one-move advance is at most zero for both sides, a fact of the board alone in which no table value enters.	folk
nd_blunder	move	The mover is winning or drawing and at least two moves tie for the best one-move advance, of which at least one keeps the value and at least one forfeits it.	unnamed
off_type_2plus	value	The mover's slide debt is at least two (the inverted form of on_type).	folk
on_type	value	The mover's slide debt, a lower bound on the slides its pieces still owe, is zero.	folk
only_move	move	The mover is winning or drawing, has at least two legal moves, and exactly one of them keeps the value.	ml
partner_reach_boost	move	In the cooperative world: some on-schedule move raises the partner's best one-move advance by at least one, the helping bridge.	unnamed
premature_entry_trap	move	The mover is winning or drawing, a move into the goal exists, every entering move forfeits the value, and some non-entering move keeps it.	folk
pseudo_dead	–	A mutual stall in which the mover wins.	folk

fact	readout	definition	tag
rule_forces_base_blunder	move	On a position that is legal under the base rules and whose base value for the mover is a win or a draw: every on-schedule move of the world forfeits the base value, so the world's rule forces what the base game calls a blunder. A world move whose successor is illegal under the base rules counts as a forfeit. Evaluated where at least one on-schedule move exists: on both turns of the cooperative world, and in the delay world only with the attacker to move and the defender holding as many pieces as the attacker, so that the position is a base position.	unnamed
sacrifice	move	The mover is winning or drawing, a value-keeping move exists, and every value-keeping move retreats, raising the mover's own single-agent distance to goal, so that retreat is optimal.	folk
sacrifice_deep	move	A sacrifice in which the mover wins and the win lies deeper than K = 7 plies.	folk
sacrifice_soon	move	A sacrifice in which the mover wins within K = 7 plies.	folk
sharpness_m70	value	The mover is winning or drawing and at least m percent of its legal moves forfeit the value (m = 70); a mover with no legal move never fires.	folk
some_schedule_base_blunder	move	The companion of rule_forces_base_blunder: some on-schedule move of the world forfeits the base value, on the same positions and under the same convention.	unnamed
straggler_critical	move	The mover is winning or drawing, has at least two legal moves, a value-keeping move exists, and every value-keeping move moves the rear piece.	folk
straggler_s5	value	The rear piece's slide distance exceeds the mean of the other pieces' by at least s cells (s = 5): a structural cousin of straggler_critical that uses no table value.	folk
wall_move	move	In the delay world with the defender to move: some delay-maximising defender move raises the attacker's single-agent distance, computed with the defender's pieces as fixed obstacles, by at least one, so the defender builds a wall.	unnamed
win_deeper_7	value	The mover wins and the win lies deeper than the K = 7 plies the depth pass labels.	value
win_in_1	value	The mover wins within 1 ply of best play.	ml
win_in_3	value	The mover wins within 3 plies of best play.	ml
win_in_5	value	The mover wins within 5 plies of best play.	ml

H D1: THE TWO-NUMBER REPRESENTATION SCORE

What it measures. D1 tests whether a candidate's activity tracks a fact on positions that were not used to fit it. It reports two numbers, because a single accuracy ends up comparing how often features fire and says little about what they know. The *detector score* E is the balanced accuracy, on the held-out half, of the detector "the candidate is above its threshold"; E^* is the best E over the fact family and names the candidate's fact. It is quoted for dense candidates only. A sparse feature that fires on a share f of positions cannot exceed $\frac{1}{2}(f/b + 1)$ on a fact of base rate $b > f$, so for latents of a sparse autoencoder the score is recorded and never quoted. The *purity margin* is the number for sparse candidates. It is, among the held-out positions where the feature fires, the share on which the fact holds, minus the share a feature with the same distance profile and no knowledge of value would obtain.

Procedure. On the training half an exact sweep over thresholds and both polarities maximises balanced accuracy, and the fitted cut is scored on the held-out half. For a latent only the active side is graded, since the inactive side covers nearly every position. For the purity margin, the fact

is chosen on the training half by the largest margin, and the two shares are then measured on the held-out firing set. The distance-null share is the training-half rate of the fact in each distance cell, averaged over the firing positions. A candidate whose held-out firing set is below the floor of 1,000 positions is unresolvable for that fact and is counted without being scored.

Controls. Two bands are computed for every row in the same run. The value-blind fake permutes the candidate's scores within each distance cell, which keeps its firing density and its distance profile exactly and destroys everything else, and it rules out a margin earned by density or distance alone. The 3 *matched random slices* per latent are random unit directions thresholded to the latent's own firing rate and given their own best fact by the same rule as the latent. They are a competing readout and not a null, since they show what an arbitrary slice of the same activation space earns at that density. A detector built from the distance cell alone bounds what distance explains, and 500 random readouts of the activations and of the raw board bound what any direction obtains for free. Planted detectors, synthetic detectors made to fire where a fact holds, exactly or nine times in ten, check that a feature which really is the fact lands on it. In every D1 file of the large board, 44 of 46 planted detectors are assigned their own fact (the same two miss in every file), with a detector score of at least 0.87.

What it cannot distinguish. Beating the distance null means that a feature carries information relevant to value beyond distance. It does not mean the feature is a unit of the fact, because arbitrary matched random slices of the same space also earn positive margins. The claim D1 supports is therefore comparative, a latent against the matched random slices and both against true sparse units, and we derive no pass threshold from the margin. Where facts are dense or the agent is a taught agent, the matched random slices can exceed the latents and the margin does not discriminate. The score averages over the positions of the pool and not over the agent's own play, so it says what a feature tracks on positions that arise, not on the agent's games. The concept vectors come from the same agent's value head and are no independent evidence. The value-blind fake permutes within distance cells, so it is blind to the mix of generators, and information about playing style that a generator leaves in its positions is absorbed by the matched random slices and not removed.

How far the comparison with the slices runs. Table 9 gives, per agent and layer, three shares of the graded features of the sparse autoencoder. They are the features whose purity margin beats every one of their own matched random slices, those above the mark that a tenth of the slices reach, and those above the slices' median. A direction with no content would reach the first one time in four, the second one time in ten and the third one time in two. On the table-taught agent's last layer a feature beats all of its own matched random slices on 0.15 of features (one seed), below the one in four that a direction with no content would reach. There the slices' median of 0.246 is twice the features' 0.124.

I D2: THE TWIN READOUT

What it measures. D2 tests whether a candidate direction is a handle on the network's behaviour. We take a position where the candidate fires and its fact holds, and we overwrite the candidate's coordinate with the value it takes in a matched position where the fact is false; this is activation patching (Zhang & Nanda, 2024) restricted to one direction. If the network reads the fact through this direction, its output should shift the way the fact's falsity predicts, and it should shift more than it does under any comparable push. D2 is the handle half of the causal test and D3 (Appendix J) is the dependence half; a candidate counts as REAL only when it passes both.

Procedure. A candidate is a pair $c = (v, \phi)$ of a unit direction v in the activation space of one layer and a detector ϕ that says on which positions it fires. Each candidate is graded against the fact it detects best on the held-out half of the pool, with its threshold and polarity fitted on the training half, and that fact fixes which of two readouts is scored. Targets are held-out scoring positions s where the candidate fires and the fact holds. For each target we draw a fact-false *twin* s' from the same distance cell, and then a fact-true twin from that distance cell whose distance to the target in activation space matches the fact-false twin's, searching up to 400 positions. A set of pairs counts as matched when the two mean distances agree within a relative tolerance of 0.05. We draw 500

Table 9: The pipeline's features track facts. On both self-play agents a feature beats every one of its own matched random slices far more often than the one time in four that chance gives; on the table-taught agent's last layer it does so less often than chance. The share of graded features, by seed, whose purity margin beats all of their own slices, exceeds the slices' top-tenth mark, and exceeds the slices' median. The two self-play arms hold three seeds; the taught agents' rows rest on one seed or two (one seed for the table-taught agent).

agent	layer	beats all own slices	above the top-tenth mark	above the median
self-play	L0	0.57, 0.49, 0.58	0.41, 0.31, 0.42	0.79, 0.77, 0.82
	L1	0.55, 0.50, 0.65	0.41, 0.32, 0.46	0.78, 0.78, 0.87
	L2	0.43, 0.39, 0.38	0.28, 0.21, 0.27	0.69, 0.66, 0.53
auxiliary-head	L0	0.54, 0.51, 0.64	0.37, 0.35, 0.48	0.76, 0.75, 0.85
	L1	0.55, 0.53, 0.47	0.40, 0.37, 0.34	0.77, 0.76, 0.69
	L2	0.53, 0.43, 0.44	0.36, 0.23, 0.29	0.78, 0.74, 0.69
table-taught	L0	0.53	0.34	0.80
	L1	0.42	0.22	0.71
	L2	0.15	0.09	0.23
curriculum-taught	L0	0.59	0.36	0.88
	L1	0.22, 0.20	0.12, 0.09	0.41, 0.40
	L2	0.19, 0.17	0.10, 0.08	0.32, 0.31

pairs per candidate on the 6×6 board and 100 on the 5×5 worlds, and a candidate with fewer pairs is recorded as unresolvable and never graded. The patch sets the activation $a(s)$ at that layer to

$$a(s) + \langle a(s') - a(s),\, v\rangle\, v,$$

and the rest of the network is recomputed from there. The same operation, a projection of the dense difference onto one direction, patches the candidate, every control, the value-blind fake and every direction of the random-direction rung. For concept vectors, linear probes and all the controls there is no latent activation that could be substituted instead, which is why the patch takes the projection and not the latent's own activation.

The move readout applies to a move fact, a fact about which moves keep the solved value, such as the fact that exactly one move does. Let $p(s)$ indicate that the network's chosen move at s preserves the solved value, and $\tilde{p}(s)$ the same after the patch. The move score is the mean drop over the pairs, $S_{\mathrm{move}}(c) = \overline{p(s) - \tilde{p}(s)}$. *The state readout* applies to a state fact, a fact about the position itself. Among the pairs on which the value head $\hat{V}$ moves and the solved values $V(s)$ and $V(s')$ differ, the state score $S_{\mathrm{state}}(c)$ is the share on which $\hat{V}$ moves toward $V(s')$.

Controls. Each candidate is compared with 40 *matched controls*, random unit directions u_j in the same activation space. Each is rescaled by bisection until it displaces the value head as far as the candidate does, and each is applied with a random sign per pair. A control is therefore a push of the same size with no relation to the fact. Size is matched on the value head's displacement and not on the coordinate's shift, because an equal coordinate shift was found to have unequal effects downstream across layers; the same matching serves both readouts. With S the score of the readout in force,

$$z(c) = \frac{S(c) - \mathrm{mean}_j\, S(u_j)}{\mathrm{sd}_j\, S(u_j)},$$

and the *strict bar* is passed when $z(c) \geq 3.0$ and $S(c)$ exceeds every $S(u_j)$. The *per-row rule* is the strict bar together with two further rungs, which apply to each row. The *value-blind fake* is a direction fitted to the fact's labels after the labels are permuted within each distance cell. It keeps the fact's density in every cell and none of its content, and the candidate must beat it by 2 game-clustered standard errors. On a state fact the candidate must also beat the *value-gradient direction*, the value head's own gradient at that layer, matched in size and signed toward the twin, on at least 50 scored pairs. The value-blind fake can clear the strict bar on a state fact, so strict counts alone are upper bounds and the per-row rule is the one we report. For a taught agent, whose value head was trained on the solved table itself, state rows never count as passes and are reported in a separate two-number form. For the self-play agent's REAL features on the move readout, the patch from the

fact-false twin costs a median 0.21, 0.18, 0.13 of value-keeping play, where the matched controls' mean costs a median 0.027, 0.015, 0.023 and their largest 0.12, 0.07, 0.06 (seeds in order), so a push of the same size on the value head moves play, on average, by about a tenth of what the handle moves it. No direction known in advance to be a handle was sent through D2 and D3; the re-pass rate of Section 5.4 is therefore the only bound on the rule's sensitivity, and the counts are more likely too low than too high.

What it cannot distinguish. D2 moves one direction in isolation, so it tests whether a single direction is a usable handle. A concept spread over several directions that matters only in combination can fail D2, and if one of its directions passes, the pass says nothing about the combination. The matched controls absorb the generic damage of leaving the activation manifold, but they are no test of such combinations. The state score is a share of pairs capped at one and cannot rank effect sizes. Twins matched on distance remove distance as the driver and leave other correlates of the fact inside a distance cell, which is why a handle is never called the network's representation of the fact. A candidate's fact is the member of our fact family that it detects best, so a feature tracking something outside the family is graded on its nearest member. Passing D2 shows that the handle moves the readout as the fact predicts; it does not show that the network computes the fact as such. On a move fact the pushes are sized by their effect on the value head; a control sized by its shift of the policy was not run. Two things protect the move readout from pushes that are too small. The whole rule's false-pass rate there is measured with these same controls (Section 5.4), and a candidate must also pass D3, which a direction with a merely large effect on the policy does not. One linear direction may be the wrong unit for a state fact, as tensor-product probes show in an Othello model (Lee et al., 2026).

J D3 AND THE SET LEVEL

What it measures. D3 tests whether the network's play on a fact's positions depends on the candidate. With the pairs of D2, the candidate's coordinate is replaced once from the fact-false twin and once from the fact-true twin. The effect is the mean difference between the two drops in value-keeping play,

$$\Delta(c) = \overline{\big(p(s) - \tilde{p}_{\text{false}}(s)\big) - \big(p(s) - \tilde{p}_{\text{true}}(s)\big)}.$$

D3 is read on play for every fact, because dependence is a question about play. D2 reads a state fact on the value head because a state fact names no move and the play readout showed no signal on such facts. The *set-level test* puts the same question to a whole set. It measures whether play on a fact's positions degrades more when every direction a method assigned to that fact is projected out of the layer at once than when as many random directions are removed.

Procedure. The *random-direction rung* repeats the swap with 10 random unit directions in place of the candidate, on the same pairs, each direction u receiving its own projection $a(s) + \langle a(s') - a(s), u\rangle u$, so the candidate's scalar is never reused. A candidate meets the D3 condition when $\Delta(c)$ minus the rung's mean is positive and larger than the rung's standard deviation, its twins are matched on distance, and it fires on at least 6,000 held-out scoring positions. A candidate is REAL when, all at once, it passes the strict bar of D2, beats the value-blind fake rung and, on a state fact, the value-gradient direction rung, meets the D3 condition with its twins matched on distance, and fires on at least 6,000 held-out scoring positions. Its D3 effect is small beside the D2 score. For the self-play agent's REAL move-side features the effect has a median of 0.016, 0.016, 0.014 against a game-clustered standard error of 0.013, 0.014, 0.013, exceeds that error on 37, 43, 25 of 52, 62, 39 features and twice it on 18, 22, 8, and is a median 0.11, 0.15, 0.11 of the fact-false drop (seeds in order); the REAL rule reads the effect against the rung, not against this error. Over the rows of one method at one layer we also report a sign test, the share of positive values of $\Delta(c)$ minus the rung's mean with its exact binomial p, which cannot reach significance below 6 nonzero rows. For the set-level test, the firing positions of a set's members are pooled and restricted to held-out positions where the fact holds, subsampled to 20,000 when there are more. At each k in $\{1, 2, 4, 8\}$ and at the full size of the set, the span of the first k members by ascending latent index (neither purity nor the D2 score orders them) is projected out. The drop in value-keeping play is compared with 10 random sets and 3 *decoy sets* of the same size, a decoy set being made of value-blind fake directions. The set-level z is the set's drop minus the mean drop of the random sets, in units of their standard

deviation. The participation ratio of a set, $(\sum_i \lambda_i)^2 / \sum_i \lambda_i^2$ over the eigenvalues of its Gram matrix, counts its effectively distinct directions.

Controls. The random-direction rung rules out an effect that any swap of a single coordinate of this size produces. The fact-true twin rules out the swap itself as the cause, since both arms replace the coordinate from a position in the same distance cell at a matched distance and differ only in whether the fact holds. Matching on distance limits, and does not remove, a sign driven by unequal sizes of perturbation, and a row whose twins are unmatched has its D3 withheld. A row whose play before any patch already sits at the level of a random mover is flagged, because a drop has no room to show there. Random sets rule out a drop earned by removing any k directions, and decoy sets rule out a drop earned by the fact's distance profile.

What it cannot distinguish. Necessity cannot be identified feature by feature through removal in this representation, because the effect of removal scales with the rank removed and the other members of a redundant set stand in for the one removed. The set is therefore the unit of our necessity claim. An intervention can succeed through a pathway the network never uses, and set-level summaries stay stable where single-component overlap falls to chance (Makelov et al., 2024; Sheng & Fu, 2026). The D3 condition admits noise by construction, because the spread of a rung of 10 draws is narrower than a row's own standard error, so a direction without content clears it on 7% to 20% of trials over the measured cells. The noise rate of REAL is therefore measured on the conjunction (Appendix K). The swap acts at one layer; a fact carried redundantly elsewhere, or nonlinearly, survives it. The band of the set-level test rests on 10 draws, so a z of 3 is no three-sigma event, and decoy sets clear that bar on a minority of sets, which is why the decoy rate is printed beside every set count. Once a set holds half the layer's width, a random set removes as much as the candidate set and the band collapses; such sets are read at small k. The sign test counts directions and is silent on sizes.

REAL counts beside the noise count, for every graded agent. Tables 10, 11 and 12 give the number of REAL sparse autoencoder features beside the noise count for every graded agent, per layer and readout, on the agent-free pool, on the cross-play pool and on the two rule-variant worlds. Seeds are listed in order and are never pooled. The noise count is the count that directions with no content would earn under the same rule on the same rows (Appendix K). Figure 4 draws the two self-play agents' counts on the two pools and the five world agents' counts beside their noise counts. In the two taught agents the handles sit early. The table-taught agent returns 11, 8 and 1 handles at the first, second and last layer, against noise counts of 0.5, 0.7 and 3.5 (one seed); the curriculum-taught agent returns 5, 8 and 1 on one seed and 3, 10 and 2 on the other. At the last layer no taught agent returns more than 2, where noise alone would earn at least 2.2 and both self-play agents hold 14 to 21 (Table 10). On the cross-play pool the table-taught agent again holds its handles early and none at the last layer, and the curriculum-taught agent does not (Table 11), and the last layer is also where both taught agents' features beat their own matched random slices less often than chance (Table 9).

The numbers behind the main text. Removing the self-play agent's rear-piece set at the first layer costs 0.48, 0.27, 0.21 of value-keeping play by seed, where as many random directions cost 0.003 to 0.005, and removing its sacrifice set costs 0.28, 0.28, 0.18 against -0.001 to 0.022. Three floors stand between a latent and a count, and each table's denominator is one of them. D1 grades a latent that fires at least 1,000 times on the held-out scoring positions; the causal tests run on a candidate that holds 500 twin pairs; and a feature can be REAL only if it fires on at least 6,000 held-out scoring positions. At the first layer of the first seed these leave 3,043 graded latents, 677 resolved candidates on the move facts and 241 move-side rows above the REAL floor. The concept vector method's sets are read at the set level too. As a set, its win-class directions cost value-keeping play 3.7, 5.4 random-set standard deviations on the self-play agent (two seeds), and on the taught attacker of the delay world its only-move set stands 129 out (one seed). The set-level result travels too. On the cross-play pool 46 of 59 and 42 of 54 of the two self-play agents' readable feature sets clear the bar where decoy sets clear it on 2 of 104 and 0 of 98, and on the worlds every readable set of the three taught agents does (Table 13; one seed per run).

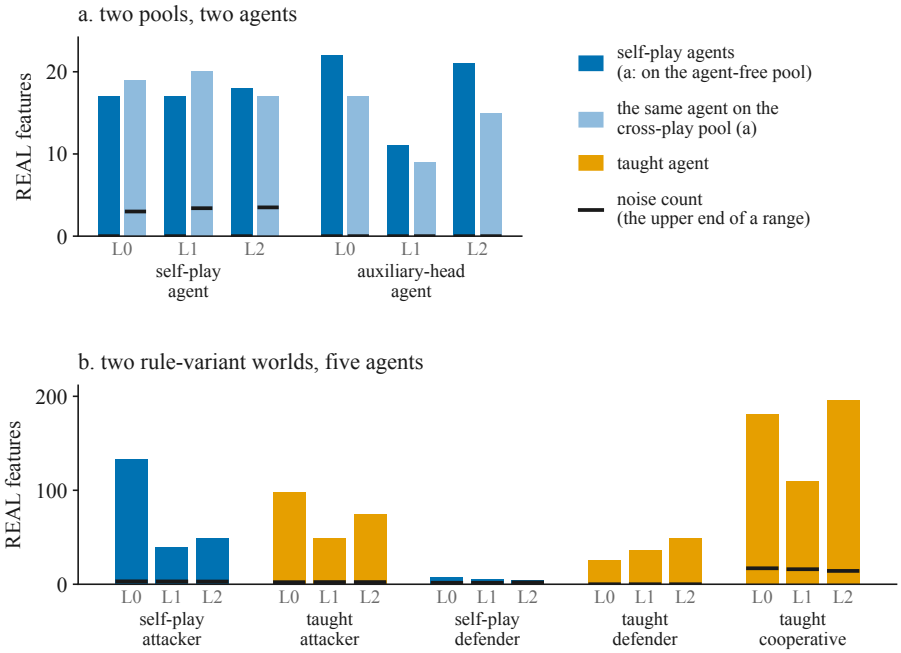


Figure 4: It travels. (a) The two self-play agents' move-side REAL features per layer on the agent-free pool and on the cross-play pool, with a dictionary extracted afresh there, each above its noise count. (b) The same rule on the two rule-variant worlds, per layer, for the five world agents. In both panels the ink cap is the cell's noise count, measured on that cell or borrowed by the rule of Appendix K, and the upper end where the count is a range (Tables 11 and 12 give both ends and each count's source); a cap at zero is a noise count of zero; one seed throughout.

Table 10: Every agent and layer, the count beside its noise count. Both self-play agents stand far above it on the move readout in every layer and seed; on the value readout the counts sit inside it at the first two layers and above it at the last, severalfold in five of the six agent-seeds. Seeds in order, on the agent-free pool. The superscript on a noise count gives its source: m, measured on a cell of that layer; l, borrowed from a measured cell of another layer of the same agent; r, borrowed from another seed of the agent or from the range of the measured cells of that board. Every value-readout count of the two self-play agents is an upper bound, and the rows of the taught agents rest on one seed or two. The value rows of a taught agent are not counted, because a value head trained on the table passes the value test trivially, and they have no noise count.

agent	layer	move: REAL	move: noise count	value: REAL	value: noise count
self-play	L0	17, 19, 12	0^m, 0^l, 1.2^l	0, 2, 10	23.9^l, 24.6^l, 18.2^l
	L1	17, 26, 12	0^l, 0^l, 1.3^l	6, 7, 1	22.7^l, 24.5^l, 15.2^l
	L2	18, 17, 15	0^m, 0^m, 1.3^m	22, 78, 81	18.8^m, 24.2^m, 15.1^m
auxiliary-head	L0	22, 23, 7	0^l, 0^l, 0.9^l	2, 2, 0	12.9^l, 12.1^l, 16.5^l
	L1	11, 6, 21	0^l, 0^l, 1.2^l	11, 4, 13	11.3^l, 10.5^l, 19.8^l
	L2	21, 14, 20	0^m, 0^m, 1.5^m	24, 24, 70	9.3^m, 11.0^m, 22.7^m
table-taught	L0	11	0.5^m	not a count	–
	L1	8	0.7^l	not a count	–
	L2	1	3.5^l	not a count	–
curriculum-taught	L0	5, 3	0.8^l, 0.2–1.3^r	not a count	–
	L1	8, 10	4.2^m, 2.0–4.1^r	not a count	–
	L2	1, 2	3.5^l, 2.2–4.0^r	not a count	–

Table 11: The count survives a change of positions and of dictionary. The same counts on the cross-play pool, with a dictionary extracted afresh on that pool's activations; every row rests on one seed, and the superscripts are those of Table 10. The value-readout counts of the two self-play agents are an upper bound.

agent	layer	move: REAL	move: noise count	value: REAL	value: noise count
self-play	L0	19	0–3.0^{r}	0	13.7–70.3^{r}
	L1	20	0–3.4^{r}	0	13.2–67.9^{r}
	L2	17	0–3.5^{r}	7	10.6–54.4^{r}
auxiliary-head	L0	17	0^{m}	1	27.0^{l}
	L1	9	0^{l}	5	25.7^{l}
	L2	15	0^{l}	20	23.7^{m}
table-taught	L0	10	2.2^{m}	not a count	–
	L1	5	1.5^{l}	not a count	–
	L2	0	4.8^{l}	not a count	–
curriculum-taught	L0	5	0.4^{l}	not a count	–
	L1	11	1.1^{m}	not a count	–
	L2	3	0.8^{l}	not a count	–

Table 12: On the two rule-variant worlds the same rule returns 26 to 196 handles per layer in both attackers, the taught defender and the cooperative network, above their noise counts, and a network that knows a world carries handles on that world's own concepts. The same counts on the two rule-variant worlds; every row rests on one seed, and the superscripts are those of Table 10. Only the two defenders of the delay world have value-readout rows, and the taught defender's are not counted, as above.

world, agent	layer	move: REAL	move: noise count	value: REAL	value: noise count
delay, self-play attacker	L0	133	2.5–3.1^{m}	–	–
	L1	39	2.4–3.0^{l}	–	–
	L2	49	2.3–2.9^{l}	–	–
delay, self-play defender	L0	7	1.6^{l}	6	3.7^{l}
	L1	5	1.6^{l}	12	3.4^{m}
	L2	4	2.0^{m}	3	3.3^{l}
delay, taught attacker	L0	98	1.5–2.2^{m}	–	–
	L1	49	1.4–2.3^{l}	–	–
	L2	74	1.2–2.3^{l}	–	–
delay, taught defender	L0	26	0^{l}	not a count	–
	L1	36	0^{l}	not a count	–
	L2	49	0^{m}	not a count	–
cooperative, taught	L0	181	17.0^{l}	–	–
	L1	110	16.0^{l}	–	–
	L2	196	14.2^{m}	–	–

K THE CONTENT-FREE NULL PROGRAM

What it measures. The null program answers one question per cell, a cell being one agent's graded results at one layer for one fact and one polarity. The question is how often, on this agent's own network, pool, twins and controls, a direction with no content about the fact passes each bar of the causal chain, up to REAL itself. That rate, multiplied by the number of the agent's latents of that fact and polarity that fire often enough to be eligible, is the *noise count*. It is printed beside every REAL count in the paper.

Procedure. A *content-free direction* is a random direction in the layer's activation space. We draw several families that differ in how many coordinates they spread over (dense Gaussian directions,

Table 13: Necessity belongs to the set. On every agent, pool and world most feature sets small enough to be read stand at least 3 random-set standard deviations out, 40, 46, 52 of 49, 53, 58 on the self-play agent, while decoy sets of the same size do so on 0, 1, 5 of its 90, 94, 99 sets; the most on any run is 20 of 115, on the table-taught agent's cross-play run. The set-level test per agent and pool; seeds in order, and a row with a single number rests on one seed. Sets: the sets of every method on that agent. Feature sets above the bar: of the sparse autoencoder sets small enough to be read (fewer members than half the layer's width), those whose removal costs value-keeping play at least 3 random-set standard deviations more than random sets of the same size do. The two decoy set columns count, over all sets, the decoy sets that clear the same bar and the decoy sets whose removal costs less than the random sets' mean.

agent	pool	sets	feature sets above the bar	decoy sets above the bar	decoy sets below random
self-play	agent-free	90, 94, 99	40 of 49, 46 of 53, 52 of 58	0 of 90, 1 of 94, 5 of 99	74 of 90, 70 of 94, 69 of 99
auxiliary-head	agent-free	86, 77, 94	38 of 47, 29 of 38, 39 of 50	0 of 86, 0 of 77, 6 of 94	69 of 86, 68 of 77, 70 of 94
table-taught	agent-free	116	54 of 73	7 of 116	75 of 116
curriculum-taught	agent-free	97, 106	39 of 55, 55 of 65	1 of 97, 8 of 106	86 of 97, 83 of 106
self-play	cross-play	104	46 of 59	2 of 104	84 of 104
auxiliary-head	cross-play	98	42 of 54	0 of 98	83 of 98
table-taught	cross-play	115	54 of 73	20 of 115	69 of 115
curriculum-taught	cross-play	103	40 of 63	3 of 103	92 of 103
delay, self-play attacker	its world's	48	25 of 33	0 of 48	38 of 48
delay, self-play defender	its world's	34	13 of 17	0 of 34	28 of 34
delay, taught attacker	its world's	51	28 of 28	2 of 51	41 of 51
delay, taught defender	its world's	40	16 of 16	1 of 40	34 of 40
cooperative, taught	its world's	50	16 of 16	2 of 50	41 of 50

sparse directions on a few coordinates, and single coordinates), with 200 directions per family in most cells (Table 14 lists the trials per cell). Each direction fires on the top share q of its projection over the held-out scoring positions, where q is the median firing share of the agent's own REAL latents. On a world, q is raised until every fact in question yields enough targets to be scored. Each direction then goes through the whole chain with fresh random streams (its own twins, matched controls, value-blind fake, value-gradient direction and random-direction rung) and is judged by the same code that judges a latent. The rate is the number of REAL passes over the eligible trials, which are the resolved directions above the firing floor, with a Wilson interval at $z = 1.96$. On the positive side, the agent's own REAL latents for the fact are re-scored through the same path on a fresh draw, on their own firing sets and at the density q. A replication draw re-scores a cell a second time, so that we also know how often a direction is REAL on two independent draws. In the one cell where it was run, the self-play agent's win class at the last layer, 0.6% of 504 directions with no content passed both draws, beside 3.4% on one (one seed). A cell that was measured supplies its own noise count. Every other row borrows a rate by a fixed order of preference. The first choice is the same agent and fact at another layer, then the same agent and another fact of the same readout, then the same arm at another seed, and last the range over the board's measured cells. A rate is never borrowed across boards, and the rule that picks the source cell was fixed before the numbers were seen.

Controls. Sending a content-free direction through the full rule rules out reading a REAL count as evidence without knowing what the rule lets through for noise. Several families rule out one shape of random direction standing for all, because the rates differ with how spread out a direction is. The positive side rules out a null that is small only because nothing can pass at that density. If the agent's own REAL latents pass again at a high rate while noise stays near zero, the null is informative. The replication draw rules out a per-latent label that is the luck of one draw. Every pass-or-fail bar in the pipeline has both sides measured in the same way, what noise earns and what a known-real candidate

earns, so no bar is used without its two error rates known; the inventory of those bars labels each one from its two rates and gates nothing.

What it cannot distinguish. The null measures what a direction with no content earns. It does not measure a direction that carries a correlated fact outside our fact family; a covariance-matched null, the field's structured alternative, is the stronger test and can saturate until every direction looks typical (Reblitz-Richardson, 2026). The rate depends on how spread out the directions of a family are, and the real latents sit between our families, so a borrowed rate can be off in either direction. The rate also depends on the density through the number of pairs, which makes it a property of a cell and no constant of noise. Most rows borrow their noise count, and the status of every expectation (measured, or the tier it was borrowed at) travels with it, so every printed noise count says whether it was measured or borrowed. A REAL label on a single latent is a label of one draw, which is why the paper's claims are counts per method and carry their replication rate.

The measured cells. Tables 14 and 15 list every cell the null program measured, with the rate at which directions with no content passed the full causal rule there, the upper end of that rate's Wilson interval, and the share of the agent's own REAL features that passed again on a fresh draw. The upper end of the interval does not change the headline. In the self-play agent's last-layer forced-wait cell, where no direction passed, the upper end is 0.78% of trials and the noise count's upper end 1.0 features against 18 REAL ones. On the value readout at the first two layers the features pass less often than random directions of the same density, so there the noise count overstates what a feature without content earns; the paper makes no claim on that readout, and why the features pass less often there was not measured. Figure 5 draws those two rates beside the three controls of D1 on the same features. On the move readout the highest rates belong to taught agent cells that fire at higher densities. On the value readout a sparser family of directions passes more often than the dense family (5.5% against the cells of Table 14; one seed), so those passes are a property of the value score and not of dense directions. On the worlds a REAL feature passes again on a fresh draw on 47 of 58 cases in the cooperative world. On the self-play attacker's fact that the delay rule itself defines, a greedy move into a permanent block, it does so on 17 of 36 (one seed; Table 15).

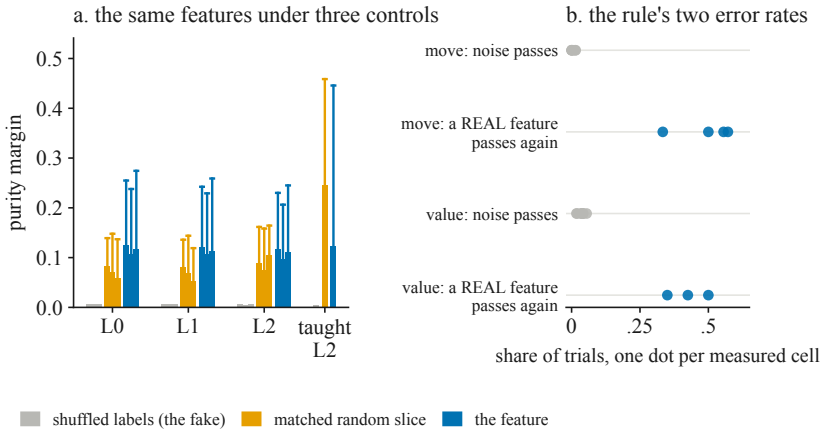


Figure 5: What each control lets through. Against the shuffled-label control every dictionary passes; against a matched random slice of the same space the same features are better by less (Table 1 prints both medians), and on the taught agent's last layer the slices win. (a) The purity margin of the self-play agent's features, the median as a bar and the top tenth as a whisker, three seeds, beside its matched random slices and its value-blind fake, per layer, and the taught agent's last layer (one seed). (b) The rule's two measured error rates by readout, one dot per measured cell: the share of content-free directions that pass the full causal rule (Table 14), and the share of the agent's own REAL features that pass again on a fresh draw, on 3 to 40 features per cell.

Table 14: Directions with no content pass the full causal rule on at most 4.3% of trials in any cell of the table, and an agent's own REAL feature passes again on a fresh draw about half the time. The null program on the two pools of the large board. For each measured cell (an agent, a seed, a layer and a fact): the directions with no content that were tried, the share of them that passed the full causal rule, the upper end of that share's Wilson interval at $z = 1.96$, and how many of the agent's own REAL features of that cell passed again on a fresh draw of twins and controls. The win class is a value-readout cell; every other fact is a move-readout cell. Each cell was measured once, on one seed of its agent.

agent	pool	seed	layer	cell's fact	trials	pass the full rule	upper end	own REAL features passing again
self-play	agent-free	0	L0	`forced_wait`	366	0%	1.04%	5 of 9
self-play	agent-free	0	L2	`forced_wait`	490	0%	0.78%	4 of 7
self-play	agent-free	0	L2	`mover_wins`	504	3.37%	5.34%	10 of 20
self-play	agent-free	1	L2	`forced_wait`	385	0%	0.99%	3 of 6
self-play	agent-free	1	L2	`mover_wins`	398	4.27%	6.73%	14 of 40
self-play	agent-free	2	L2	`forced_wait`	395	0.51%	1.83%	1 of 3
self-play	agent-free	2	L2	`mover_wins`	400	3.75%	6.09%	17 of 40
auxiliary-head	agent-free	0	L2	`forced_wait`	583	0%	0.65%	5 of 14
auxiliary-head	agent-free	0	L2	`mover_wins`	599	2.00%	3.47%	9 of 24
auxiliary-head	agent-free	1	L2	`forced_wait`	392	0%	0.97%	2 of 7
auxiliary-head	agent-free	1	L2	`mover_wins`	399	1.75%	3.58%	7 of 24
auxiliary-head	agent-free	2	L2	`forced_wait`	389	0.51%	1.86%	7 of 9
auxiliary-head	agent-free	2	L2	`mover_wins`	400	4.00%	6.40%	19 of 40
curriculum-taught	agent-free	0	L1	`forced_wait`	399	1.50%	3.24%	0 of 1
table-taught	agent-free	0	L0	`forced_wait`	399	0.50%	1.81%	–
auxiliary-head	cross-play	0	L0	`nd_blunder`	330	0%	1.15%	1 of 7
auxiliary-head	cross-play	0	L2	`mover_wins`	397	4.03%	6.45%	7 of 19
curriculum-taught	cross-play	0	L1	`nd_blunder`	383	0.52%	1.88%	–
table-taught	cross-play	0	L0	`nd_blunder`	379	1.32%	3.05%	1 of 3

Table 15: The same on the two rule-variant worlds (one seed per agent). A fact whose name begins with `not_` is the inverted form of that fact.

world, agent	role	layer	cell's fact	trials	pass the full rule	upper end	own REAL features passing again
cooperative, taught		L2	`coop_only_move`	800	0.62%	1.45%	47 of 58
delay, self-play	attacker	L0	`att_greedy_inf`	287	0%	1.32%	17 of 36
delay, self-play	attacker	L0	`att_sacrifice`	621	0.48%	1.41%	3 of 11
delay, self-play	attacker	L0	`not_att_greedy_inf`	710	0.42%	1.23%	25 of 45
delay, self-play	attacker	L0	`not_att_sacrifice`	797	0.25%	0.91%	17 of 30
delay, self-play	defender	L1	`att_stalled`	575	0.52%	1.52%	1 of 7
delay, self-play	defender	L2	`wall_move`	793	0.38%	1.11%	0 of 4
delay, taught	attacker	L0	`att_greedy_inf`	707	0%	0.54%	8 of 13
delay, taught	attacker	L0	`att_sacrifice`	641	0.16%	0.88%	13 of 32
delay, taught	attacker	L0	`not_att_greedy_inf`	792	0%	0.48%	2 of 13
delay, taught	attacker	L0	`not_att_sacrifice`	796	0.38%	1.10%	13 of 34
delay, taught	defender	L2	`wall_move`	798	0%	0.48%	8 of 24

L THE UNRESOLVED SHARE AND THE CENSUS

What it measures. The unresolved-share read sorts the REAL latents of one agent at one layer into those that carry the race itself, those that carry a fact of the fact family, and those that carry nothing the fact family names. The census counts, per agent and per fact, how many candidates of any method were assigned to the fact and how many of them pass each causal test.

Procedure. Each REAL latent is sorted in a fixed order, and the first class that applies ends the sort. It is in the race class if the fact of its purity margin is a value class, a depth fact about value, or the inverted form of any fact. It is named if its purity margin exceeds the largest margin among its own 3 matched random slices. It is trivial if its margin is at or below the 0.9 quantile of the value-blind fakes' margins for the same fact and band of firing density. Otherwise it is *unnamed*, and a sample of 20 such latents per layer is kept for inspection. The census reads counts the graded results already hold. They are the candidates assigned to each fact, those with a D3 result, those whose D3 effect exceeds the rung's mean, and those that pass the strict bar of D2. At the set level they are the sets built for the fact and the sets that stand at least 3 standard deviations of the random sets above them.

When the census is summed by naming tag, the inverted forms are left out, because each would count its parent's concept a second time.

Controls. The matched random slices rule out calling a space that is dense with facts a concept, since a latent is named only if it names its fact better than arbitrary directions of the same density do. The quantile of the value-blind fakes rules out an unnamed count made of latents whose margin is at the level a label-blind null reaches. Sorting the race class first rules out counting the network's knowledge of value as a strategic concept or as something unnamed. Because only REAL latents are sorted, the split inherits every control of the causal chain.

What it cannot distinguish. Each latent is labelled by one fact and one threshold per bar. A latent without content beats three exchangeable matched random slices one time in four by construction, so truly unnamed latents are labelled named at that rate. In the other direction, a fragment of value or distance that the purity margin cannot tell from a unit is labelled unnamed. The unnamed count is therefore neither an upper nor a lower bound. The split says nothing about what an unnamed latent carries; the inspection sample is the only evidence. The census counts by the fact assigned in causal grading, which need not be the fact of the purity margin, so the two reads can file one latent under different facts.

The classes by readout, and the census for all four agents. Tables 16, 17 and 18 give the classes of the REAL features by readout, and Table 19 gives the census, per fact, for all four agents. The linear probe's own direction is no handle either. Graded by the same D2 on the type census (`on_type`) and on the lagging rear piece, the dense direction that decodes the fact passes at no layer and in no seed. The count is 0 over the strict, per-row and REAL entries of both facts (upper bound). Figure 6 draws the self-play agent's move-side REAL features per fact. A handle is a handle on the fact's positions and never a detector of the fact. The self-play agent's move-side REAL features fire on positions where their assigned fact holds with a median precision of 0.23, 0.05, 0.03 and cover a median 0.014, 0.022, 0.023 of the fact's positions (seeds in order, from the D1 rows). Every move-side handle that beats all three of its own matched random slices names a folk-tagged fact by its purity margin. Forced wait is named by 22, 23, 5 of them, sharpness by 9, 3, 3 and the type census by 4, 5, 5, seeds in order, and the mutual stall by 4, 2 on the second and third seeds (Table 8 carries the tags). The named handles fall on the facts that are easiest to read: the type census and the mutual stall are properties of the board alone, computable with no table, and none of the named handles names sacrifice, the rear-piece rule or the entry trap, the strategic facts the handles of Section 5.2 sit on. On those facts the network has handles that no representation test names. On forced wait, the fact with the most handles, most positions of the agent-free pool are stalls in which the mover has no advancing legal move (between 82% and 83% of the held-out forced-wait positions on the three pools; 52% on the cross-play pool); the handles' target positions are stalls at a median share of 0.93, 0.94, 0.88 by seed (0.61 on the cross-play pool), and outside the stalls their purity margin on forced wait falls from a median 0.21, 0.19, 0.14 to 0.01, 0.02, 0.01, so a forced-wait handle is best read as a handle on positions where the mover cannot advance. Because forced wait holds on every such position, the counts stand as counts of forced-wait positions.

M THE CROSS-PLAY EXAM

What it measures. The *cross-play exam* tests whether an agent's grading changes when the positions are ones that the graded agents actually play. Each arm at its first seed is graded twice with the same network, the same recipe for the dictionary and the same grader, once on its agent-free pool and once on the cross-play pool. The positions of the arm's own games are read beside. Its number is a difference between the two gradings on the same fact, read next to a control that separates "the fact became rarer" from "the reads moved".

Procedure. The cross-play pool is built once. The nine graded networks play every unordered pair (36 pairs) from shared random openings, twice with colours swapped, under the stochastic play of the training recipe. Every position of every game is kept, duplicates are removed, and the pool is sized to the agent-free pool. Each position records which networks visited it, the split is again by game, and the parametric facts keep the parameters chosen on the agent-free pool, so a fact means the same thing on both pools and only its base rate moves. The grader then runs unchanged, with a

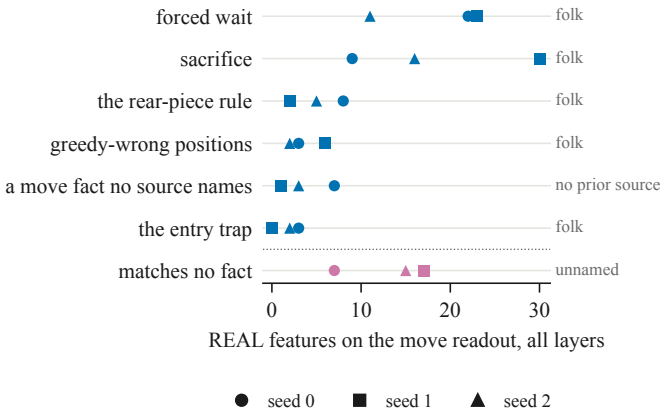


Figure 6: Which concepts. The move-side handles sit on the game's folk concepts. Per fact, the self-play agent's REAL features on the move readout, summed over layers and both polarities, one marker per seed; the last row re-counts, from the rows above, the handles that track no fact better than their own random slices (a handle carries two labels: the fact assigned in causal grading, which places it in a row above, and the representation test's named or unnamed sort, which places it in the last row). Each fact carries its naming tag, folk lore or no prior source.

Table 16: On the agent-free pool 7, 17, 15 of the self-play agent's move-side REAL features and 9, 4, 10 of the auxiliary-head agent's match none of the facts in the family, a handle with no fact rather than a fact with no human name. The classes of the REAL sparse autoencoder features on the move readout on the two pools of the large board, per agent, pool and layer; seeds in order, and a row with a single number rests on one seed. The last column counts REAL features with no purity margin on file, which the sort cannot place.

agent	pool	layer	REAL	race	named	trivial	unnamed	no purity report
self-play	agent-free	L0	17, 19, 12	1, 1, 4	13, 15, 3	0, 0, 0	3, 3, 5	0, 0, 0
	agent-free	L1	17, 26, 12	6, 6, 1	11, 16, 7	0, 0, 0	0, 4, 4	0, 0, 0
	agent-free	L2	18, 17, 15	3, 3, 4	11, 4, 5	0, 0, 0	4, 10, 6	0, 0, 0
auxiliary-head	agent-free	L0	22, 23, 7	6, 3, 4	12, 18, 2	0, 0, 0	4, 2, 1	0, 0, 0
	agent-free	L1	11, 6, 21	2, 0, 6	7, 5, 13	0, 0, 0	2, 1, 2	0, 0, 0
	agent-free	L2	21, 14, 20	7, 7, 3	11, 6, 10	0, 0, 0	3, 1, 7	0, 0, 0
table-taught	agent-free	L0	11	3	8	0	0	0
	agent-free	L1	8	1	3	0	4	0
	agent-free	L2	1	0	0	0	1	0
curriculum-taught	agent-free	L0	5	4	1	0	0	0
	agent-free	L1	8, 10	1, 1	5, 4	0, 0	2, 5	0, 0
	agent-free	L2	1, 2	0, 0	0, 0	0, 0	1, 2	0, 0
table-taught	cross-play	L0	10	0	8	0	2	0
	cross-play	L1	5	2	2	0	1	0
	cross-play	L2	0	0	0	0	0	0
curriculum-taught	cross-play	L0	5	0	4	0	1	0
	cross-play	L1	11	4	3	0	4	0
	cross-play	L2	3	0	1	0	2	0

Table 17: The same classes on the move readout for the agents of the two rule-variant worlds; every row rests on one seed.

world, agent	layer	REAL	race	named	trivial	unnamed	no purity report
delay, self-play attacker	L0	133	0	62	35	19	17
	L1	39	0	14	7	9	9
	L2	49	0	11	18	10	10
delay, self-play defender	L0	7	0	5	0	2	0
	L1	5	0	1	0	3	1
	L2	4	0	0	0	2	2
delay, taught attacker	L0	98	0	24	16	41	17
	L1	49	0	14	8	21	6
	L2	74	0	16	13	33	12
delay, taught defender	L0	26	0	15	1	5	5
	L1	36	0	16	0	9	11
	L2	49	0	8	0	22	19
cooperative, taught	L0	181	0	93	36	52	0
	L1	110	0	29	13	68	0
	L2	196	0	12	20	164	0

Table 18: The same classes on the value readout, for the three agents that have a REAL feature there; every other agent has none at any layer. The defender's rows rest on one seed.

agent	layer	REAL	race	named	trivial	unnamed	no purity report
self-play	L0	0, 2, 10	0, 1, 5	0, 1, 3	0, 0, 0	0, 0, 2	0, 0, 0
	L1	6, 7, 1	5, 6, 1	1, 1, 0	0, 0, 0	0, 0, 0	0, 0, 0
	L2	22, 78, 81	14, 37, 51	4, 17, 5	0, 0, 0	4, 24, 25	0, 0, 0
auxiliary-head	L0	2, 2, 0	1, 1, 0	0, 0, 0	0, 0, 0	1, 1, 0	0, 0, 0
	L1	11, 4, 13	9, 3, 7	2, 1, 1	0, 0, 0	0, 0, 5	0, 0, 0
	L2	24, 24, 70	16, 17, 38	6, 4, 19	0, 0, 0	2, 3, 13	0, 0, 0
delay, self-play defender	L0	6	0	0	1	3	2
	L1	12	0	2	1	6	3
	L2	3	0	2	0	1	0

dictionary extracted per agent on the cross-play pool's own activations. Three quantities are added to every row without touching its numbers. The first is the fact's base rate on both pools, with the label NOT COMPARABLE when the two differ by more than a factor of 2; a pair under that label is printed and never differenced. The second is a rate-matched reference, in which the row is scored again on a subsample of whole games thinned until the fact's rate matches the agent-free pool's, provided at least 1,000 firings remain. The third is the same detection numbers on the arm's own positions.

Controls. Grading the same network with the same grader on a different pool rules out a change of instrument as the source of a difference. The rate-matched reference rules out rarity as the whole story. The label rules out reading a difference on a fact whose base rate moved severalfold. Dropping test-side positions that lie in an agent-free pool's training half rules out scoring a position that a taught agent was trained on. For a linear probe the two gradings are compared on the probe's own fact, because the fact assigned in causal grading can differ between pools.

What it cannot distinguish. Blind spots shared by all the networks are invisible on the cross-play pool by construction; the exam cannot find a concept that every network lacks. An arm's own positions are its play against these particular opponents and are no sample of its play in general. The rate-matched reference removes rarity and leaves in place any drift in the mix of value classes;

Table 19: What passes strictly is the value itself and the move facts, while no state-level fact yields more than 1 strict pass on either self-play agent, and the three with the most candidates, which a linear probe reads as well as the move facts, yield none on the self-play agent: decodable and used come apart, in the direction in which a probe overstates. Per fact and agent (the first seed of each agent, so every column rests on one seed). Under each agent: the candidates of every method that pass the strict bar of D2 out of those resolved, then a linear probe's detector score on the fact where one was fitted. The inverted forms of the facts are left out. For the two taught agents, the counts on the facts that are graded on the value head (the state facts and the value classes) are not read as pass rates, because a value head trained on the table passes the value test trivially.

fact	self-play	probe	auxiliary-head	probe	table-taught	probe	curriculum-taught	probe
boost_k2	0 of 15	0.662	0 of 13	0.668	0 of 72	0.734	0 of 32	0.714
borrowed_d2	0 of 0	0.684	0 of 1	0.682	0 of 0	0.722	0 of 0	0.707
breakable_d2	0 of 0	0.658	0 of 0	0.657	0 of 1	0.698	0 of 0	0.686
denial_d1	0 of 50	0.746	0 of 40	0.741	0 of 29	0.761	0 of 9	0.758
forced_wait	256 of 1,202	0.832	146 of 1,147	0.835	7 of 303	0.857	4 of 250	0.852
forfeitable_draw	1 of 5	0.753	1 of 4	0.736	2 of 13	0.853	0 of 3	0.809
greedy_wrong	16 of 43	0.756	9 of 44	0.740	5 of 36	0.852	4 of 26	0.838
jump_asym_a2	0 of 45	0.713	0 of 33	0.702	0 of 50	0.762	0 of 28	0.737
long_reach_j3	0 of 5	0.695	0 of 9	0.694	1 of 62	0.733	0 of 17	0.718
mover_loses	0 of 0	0.919	0 of 0	0.910	0 of 0	0.962	0 of 0	0.957
mover_wins	262 of 2,769	0.920	352 of 2,202	0.910	32 of 394	0.962	34 of 722	0.957
nd_blunder	61 of 228	0.760	49 of 190	0.745	4 of 91	0.837	2 of 33	0.824
off_type_2plus	1 of 65	0.822	1 of 114	0.826	0 of 35	0.883	0 of 6	0.877
on_type	0 of 278	0.827	1 of 251	0.824	0 of 53	0.871	0 of 27	0.868
only_move	0 of 8	0.745	2 of 6	0.729	2 of 59	0.867	0 of 34	0.848
premature_entry_trap	34 of 135	0.754	16 of 144	0.739	4 of 59	0.849	3 of 44	0.836
sacrifice	21 of 162	0.834	21 of 159	0.837	10 of 141	0.867	5 of 140	0.860
sharpness_m70	0 of 68	0.778	0 of 71	0.758	0 of 162	0.901	0 of 142	0.885
straggler_critical	28 of 94	0.738	12 of 128	0.723	10 of 246	0.860	3 of 165	0.842
straggler_s5	0 of 257	0.842	1 of 258	0.843	1 of 54	0.889	0 of 42	0.871

a reference matched on value class and a control using a sibling arm's positions were designed and not built. Concept vectors form a list that depends on the pool and are compared as distributions, never one by one. On the arm's own positions too few of the pool's pairs remain for D2 and D3, so that column reports detection only. The difference between pools carries no error term, because the standard error was stored on one side only; Table 21's differences are therefore read as sizes without a test. The positions a self-play agent meets on the cross-play pool overlap its own training distribution by an amount we did not size, so for the self-play arms the cross-play pool is less fresh a test than it is for the taught agents.

Base rates and the probe's shift. Table 20 gives every fact's base rate on the two pools and Table 21 the shift of a linear probe's score between them. The self-play agent's move-side handles change their facts between the pools while their total holds (Section 5.3). On the cross-play pool, where stalls are rare, its forced-wait handles number 6 against 22, 23, 11 by seed on the agent-free pool, while the rear-piece rule holds 13, greedy-wrong positions 13, the unnamed move fact 13, the entry trap 8 and sacrifice 2 (one seed). On the comparable facts a probe's score typically moves by 0.014 between the pools, and at the extremes by -0.061 and 0.150 (one seed). In chess, concept recognition falls by a tenth to a fifth when randomised starts remove opening theory (Lomasov et al., 2025); that axis is memorisation, ours is which concepts a pool holds enough positions of to grade.

N THE SHORTCUT PASS AND THE ERASURE

What it measures. D2 and D3 test whether a feature for a fact moves play. The *greedy-rule pass* measures the network itself. On positions where every distance-greedy move gives away the solved value (the fact greedy_wrong), it records whether the network plays a greedy move and whether

Table 20: The folk concepts are rarer on the positions the agents actually play than on the agent-free pool. Base rates of the facts on the held-out halves of the two pools (one seed), their ratio, and whether the fact is compared across the pools. A fact whose ratio leaves the band of Section 4 is labelled not comparable, and its reads are never differenced.

fact	agent-free pool	cross-play pool	ratio	compared across the pools
boost_k2	3.13%	1.87%	0.60	yes
borrowed_d2	3.03%	1.57%	0.52	yes
breakable_d2	2.58%	1.39%	0.54	yes
denial_d1	13.8%	9.12%	0.66	yes
draw	0.175%	0.238%	1.36	yes
forced_wait	8.84%	4.06%	0.46	not comparable
forfeitable_draw	0.175%	0.238%	1.36	yes
greedy_wrong	2.55%	2.37%	0.93	yes
jump_asym_a2	0.940%	0.415%	0.44	not comparable
long_reach_j3	2.77%	1.24%	0.45	not comparable
mover_loses	43.8%	48.4%	1.10	yes
mover_wins	56.0%	51.4%	0.92	yes
mutual_stall	7.80%	0.850%	0.11	not comparable
nd_blunder	11.1%	10.9%	0.98	yes
off_type_2plus	60.6%	61.2%	1.01	yes
on_type	5.22%	4.51%	0.86	yes
only_move	3.37%	4.20%	1.24	yes
premature_entry_trap	4.12%	4.00%	0.97	yes
pseudo_dead	4.32%	0.168%	0.04	not comparable
sacrifice	0.702%	0.186%	0.26	not comparable
sharpness_m70	13.1%	15.8%	1.20	yes
straggler_critical	2.33%	2.49%	1.07	yes
straggler_s5	9.22%	5.19%	0.56	yes

Table 21: A linear probe's detector score on the cross-play pool minus its score on the agent-free pool, per fact and layer, for the two taught agents (one seed each; no such difference exists for a self-play agent). The differences carry no error term; n.c. marks a fact that is not comparable across the pools, whose difference is never taken.

fact	table-taught L0	L1	L2	curriculum-taught L0	L1	L2
boost_k2	−0.017	−0.029	−0.028	−0.029	−0.029	−0.024
borrowed_d2	−0.012	−0.022	−0.025	−0.015	−0.026	−0.018
breakable_d2	0.013	0.004	0.004	0.010	0.001	0.007
denial_d1	0.024	0.023	0.024	0.022	0.022	0.021
forced_wait	n.c.	n.c.	n.c.	n.c.	n.c.	n.c.
forfeitable_draw	−0.015	−0.011	−0.011	0.018	0.150	0.149
greedy_wrong	0.013	0.001	−0.002	0.009	0.003	0.005
jump_asym_a2	n.c.	n.c.	n.c.	n.c.	n.c.	n.c.
long_reach_j3	n.c.	n.c.	n.c.	n.c.	n.c.	n.c.
mover_loses	−0.010	−0.012	−0.014	−0.010	−0.012	−0.013
mover_wins	−0.010	−0.012	−0.013	−0.010	−0.012	−0.013
nd_blunder	0.025	0.020	0.016	0.023	0.022	0.026
off_type_2plus	0.030	0.033	0.019	0.034	0.029	0.018
on_type	0.033	0.033	0.023	0.040	0.033	0.027
only_move	−0.012	−0.010	−0.012	−0.013	−0.005	0.004
premature_entry_trap	−0.006	−0.012	−0.013	−0.008	−0.005	0.005
sacrifice	n.c.	n.c.	n.c.	n.c.	n.c.	n.c.
sharpness_m70	−0.002	−0.007	−0.006	0.003	−0.002	0.005
straggler_critical	−0.013	−0.019	−0.021	−0.017	−0.012	−0.006
straggler_s5	−0.037	−0.061	−0.038	−0.038	−0.046	−0.049

its value head prices the position as if greedy play were fine. Two probe-side reads accompany it. The erasure tests whether what the pass measures sits in one linear direction at a layer, and the probe gap tests whether a nonlinear probe reads a fact better than a linear one.

Procedure. The fall-for share is the share of held-out scoring positions with `greedy_wrong` on which the network's chosen move is a greedy move. It is read beside the same share on *look-alike positions*: within every stratum of solved value class and distance cell, as many positions as there are `greedy_wrong` ones, drawn from positions where greedy play is not wrong. The discrimination is the matched difference, each position's indicator minus the reference share of its stratum. The value difference is the value head at a `greedy_wrong` position minus the mean value head of its stratum's other positions. The same statistic on look-alike positions is zero by construction, and its scatter over 200 draws is the band in whose standard deviations the real statistic is quoted. For the erasure, the held-out scoring positions are split by game. On one half a rank-one LEACE erasure (Belrose et al., 2023) of `greedy_wrong` is fitted on a population balanced within every stratum, together with the same fit on a look-alike label and 10 random directions. On the other half the fall-for share and the value difference are read after each erasure, along with the same reads on matched positions without the fact. A linear probe and a one-hidden-layer probe are also read there before and after each erasure. A nonlinear residue is claimed only if the one-hidden-layer probe on the erased activations beats the same probe on the raw board by 3.0 standard errors. For the probe gap, the width of the one-hidden-layer probe is chosen from $\{16, 64, 256, 1024\}$ by selectivity. A gap counts only if, after 2 bootstrap standard errors over 200 resamples by game are subtracted, it still exceeds both the gap on a label permuted within distance cells and the gap on the raw board.

Controls. Look-alike positions rule out the network's plain rate of greedy play as the explanation of the fall-for share, and the band rules out reading one draw's scatter as the reference. Balancing the fit within strata rules out an erased direction that carries distance. The look-alike and random erasures rule out an effect of erasing any rank-one direction, and the matched positions rule out an erasure that moves every position alike. The raw-board probes rule out claiming the board's own nonlinear structure as the network's.

What it cannot distinguish. The pass measures one shortcut on one fact, and a network that avoids greedy play there for another reason reads the same. The band measures the scatter of the reference and cannot measure a systematic difference between the fact's positions and their look-alikes beyond value class and distance cell. An erased direction is no located circuit, and a residue that survives says only that the discrimination is not linear of rank one at that layer. Whether a residue is claimed depends on the raw-board bar, whose own noise rate we did not measure. The probe gap was run on one taught agent at two layers, is exploratory, and has a synthetic positive control only.

The pass and the erasure on the four networks. Table 22 gives the pass for the four networks and Table 23 the outcome of the erasure. On a population balanced on value class and distance cell, a linear probe reads the trap fact from the self-play agent's layers at 0.698 to 0.690 (standard error 0.0016). The raw board reads 0.700 to a linear reader and 0.746 to a small non-linear one. The erasure tests whether that discrimination sits in one linear direction, and it does not. At the first layer it takes a linear reader from 0.698 to 0.522, while a small non-linear reader still reads 0.695 (one seed). The value belief behaves differently. At the first layer the same rank-one erasure takes the taught agents' value gap to look-alike positions from -0.239 and -0.263 to -0.077 and -0.074, two thirds of it, at a cost of 0.042 and 0.025 in value-keeping play, and moves the two self-play agents' gaps less. The same erasure fitted to a value-blind fake label leaves the taught gaps at -0.238 and -0.263 (Table 23; one seed each). So the play-side discrimination does not sit in one linear direction, while most of the taught agents' under-pricing is removed by one linear cut. The cut is not the belief's alone. It changes the chosen move on 45.7% and 34.6% of ordinary matched positions, and the value of ordinary positions after it was not read.

O DEPTH FACTS

What it measures. The solved table says who wins from a position and is silent on how soon. The *conversion depth* of a position is the exact number of plies of best play until the game is decided.

Table 22: No network holds the greedy rule of thumb. Every value head, the taught agents' included, under-prices the positions where greedy play fails. The greedy-rule pass on the four networks (one seed each), on the held-out scoring positions where every distance-greedy move gives away the solved value. Then, in order: how often the network plays a greedy move there, and on look-alike positions; the matched difference; the value head's estimate there minus its estimate on look-alike positions, and minus the table's value; and the same value difference under a value-blind fake label of the same density. The matched columns rest on the 94,159 positions whose stratum holds a look-alike.

agent	positions	plays a greedy move	on look-alike positions	matched difference	value against look-alike positions	value against the table	value, value-blind fake label
self-play	94,185	16.3%	50.2%	−0.339	−0.344	−0.778	0.0004
auxiliary-head	94,185	17.0%	47.1%	−0.300	−0.315	−0.869	0.0012
table-taught	94,185	11.3%	54.1%	−0.427	−0.238	−0.364	−0.0005
curriculum-taught	94,185	14.0%	59.7%	−0.457	−0.259	−0.401	−0.0014

Table 23: The discrimination of the trap fact does not sit in one linear direction. The rank-one erasure, per network and layer (one seed each): a linear reader's balanced accuracy before and after the erasure; a small non-linear reader's after it; the standard error of a read; the value gap to look-alike positions before and after; the share of matched ordinary positions on which the erasure changes the chosen move; and its cost in value-keeping play.

agent	layer	linear read before	after the erasure	small network after	SE of a read	value gap before	value gap after	move changed on matched rows	cost in value-keeping play
self-play	L0	0.698	0.522	0.695	0.0016	−0.345	−0.236	13.2%	0.040
	L1	0.697	0.520	0.693	0.0016	−0.345	−0.227	20.0%	0.049
	L2	0.690	0.511	0.688	0.0016	−0.345	0.014	55.3%	0.207
auxiliary-head	L0	0.700	0.509	0.688	0.0016	−0.315	−0.241	15.6%	0.052
	L1	0.699	0.512	0.683	0.0016	−0.315	−0.247	16.3%	0.054
	L2	0.694	0.512	0.680	0.0016	−0.315	0.009	59.3%	0.209
table-taught	L0	0.740	0.507	0.750	0.0016	−0.239	−0.077	45.7%	0.042
	L1	0.746	0.511	0.753	0.0016	−0.239	−0.148	45.1%	0.049
	L2	0.753	0.523	0.751	0.0016	−0.239	−0.057	70.7%	0.274
curriculum-taught	L0	0.732	0.507	0.747	0.0016	−0.263	−0.074	34.6%	0.025
	L1	0.742	0.526	0.742	0.0016	−0.263	−0.119	34.0%	0.004
	L2	0.740	0.524	0.744	0.0016	−0.263	−0.059	76.8%	0.368

We compute it for every position whose depth is at most 7 and give all deeper positions, the whole draw region included, a single label. Eleven depth facts follow. They are a win or a loss within k plies for $k \in \{1, 3, 5\}$, a win or a loss that lies deeper than the horizon, a fork, and the split of one strategic fact by whether its win is near or far.

Procedure. Depth comes from a retrograde solve in synchronised levels, in which each sweep reads only the labels of the sweep before, so a state labelled at sweep p has depth exactly p. The usual solver, which chains labels within a pass, does not guarantee this. Predecessors are enumerated exactly, a candidate predecessor being accepted only if the forward move generator produces the move. Whenever depths are attached to a pool, every labelled position's parity is checked against the solved table, where an odd depth must be a win for the mover and an even depth a loss. Depth counts plies to the end of the game under best play, so a lost position has an even depth, and a loss within one ply holds exactly on finished games, where the side to move has already lost. The depth facts then join the fact family and are scored by D1 like any other fact. Our read of them tests

whether any latent is better described by a depth fact than by the fact it was assigned before. A latent scores best on a depth fact, nominally, when its best detector score on a depth fact exceeds its best score on the base facts; it counts as a depth detector when that score also exceeds the best score of its value-blind fake and the distance-only reference for that depth fact, the two bars that do the work. Detectors are counted over all latents of a layer and again over the REAL latents.

Controls. A dense reference solve agrees with the sweeps at every state of three small boards. The parity check and a least-fixpoint check on sampled states guard the depths used at full size. The three bars rule out a switch that is trivial, since the latent must beat its own previous best, what its density and distance profile earn, and what distance alone predicts.

What it cannot distinguish. Depth is exact only inside the horizon, so the depth facts separate "soon" from "not soon" and cannot tell a depth of nine from a depth of thirty. They are highly predictable from distance alone, so a latent can read them well without reading depth, which is why the distance bar binds. No causal test was run on the depth facts; a depth detector is a claim about detection and no more. The read covers the first seed of two agents. The fork is one definition among several.

The depth read in numbers. Table 24 gives the read and Table 25 shows how far the two distances alone predict each depth fact.

Table 24: With the bars in force at most 2 of a layer's 778 to 3,043 features are depth detectors, and none of the REAL features is. The depth read on the first seed of two agents (one seed each): the sparse autoencoder features graded; those that score best on a depth fact by any margin, with their share and median margin; those that remain depth detectors under the three bars; the REAL features; and the depth detectors among them.

agent	layer	features	best on a depth fact	share	median margin	detectors	REAL	detectors among REAL
self-play	L0	3,043	1,076	35.4%	0.0011	1	17	0
	L1	3,023	1,055	34.9%	0.0010	1	23	0
	L2	2,648	970	36.6%	0.0013	1	40	0
table-taught	L0	778	291	37.4%	0.0020	0	11	0
	L1	908	336	37.0%	0.0026	1	8	0
	L2	2,572	886	34.4%	0.0024	2	1	0

Table 25: Balanced accuracy of the two distances to goal alone on each depth fact (the pool of one seed).

depth fact	balanced accuracy from the two distances alone
`dual_threat`	0.952
`lose_deeper_7`	0.862
`lose_in_1`	0.957
`lose_in_3`	0.978
`lose_in_5`	0.986
`sacrifice_deep`	0.780
`sacrifice_soon`	0.957
`win_deeper_7`	0.857
`win_in_1`	0.870
`win_in_3`	0.903
`win_in_5`	0.928

P THE RULE-VARIANT WORLDS

What it measures. A world is the 5×5 board under a changed rule, with its own exact table. In the *delay world* an attacker races to fill its goal while a defender with 6 pieces, barred from the goal,

delays it; the table gives the number of plies to arrival under best play by both. In the *cooperative world* both sides cooperate to swap camps, and the table gives the length of the shortest joint plan. On a world we read the agent's play first: whether its move is an *on-schedule move*, one whose successor's value is exactly one less than the position's, so that the position stays on the shortest schedule the table gives; for each fact we report how often an agent's chosen move is on schedule on the positions where the fact holds.

Procedure. That share is reported for four agents on identical positions. They are the world's own self-play agent and the base control, a self-play agent of the unmodified game whose choice is restricted to the moves legal in the world. A taught agent trained on the world's table serves as a ceiling, and a random legal mover as a floor. The instruments of the 6×6 board are carried over unchanged with the world's objects substituted, so that the on-schedule moves take the place of the value-keeping moves, and the hidden-deadline target of Appendix D takes the place of the value order. Because a 5×5 pool is smaller, a candidate needs 100 pairs to be graded, and the firing floor is a share 0.0029 of the held-out scoring positions. One network plays both roles of the delay world and receives a role plane and a plane for elapsed plies as extra input. Its results are therefore graded once per role. One dictionary is trained on all positions, and everything is scored on one role's positions, where the role plane is constant. A band of 300 random directions per layer measures what a fact label earns by chance on a pool of this size; a sparse latent cannot reach that band, so the purity margin carries the numbers for sparse rows.

Controls. The base control rules out crediting the world's rule with behaviour that the unmodified game already induces, so the claim for a world is the difference between the world's agent and the base control on the same positions. Grading per role rules out the role plane as a free detector of the facts of one role. Positions from which the base control started its training games are dropped from every agent's scoring positions.

What it cannot distinguish. Every position is graded with the elapsed-ply input at zero, a single clock value, so behaviour later in a game is not measured. The unmodified game has no role like the defender's; on the defender's rows the base control plays a role it never met in training, and it is undefined on positions that are illegal in the unmodified game. No positive control exists for the move readout on the worlds; a pass count near zero there is a reading and no failure. The world facts are predictable from the race to a fair degree, so a feature can read them without reading the rule. The two role gradings share one dictionary and are not independent samples. Each world agent has one seed.

Behaviour per fact, and the worlds in numbers. Tables 26 and 27 give the behavioural readout per fact, and Table 28 describes the two worlds. On ten of the eleven fact rows of the two roles the world's own agent plays on schedule more often than the base-rules agent on the same positions; the exception is the attacker's fact that some on-schedule move forfeits the base value, 53.6% against 55.9%. The attacker's largest shift is on a fact the rule itself defines, 43.6% against 32.7% where a greedy move walks into a permanent block. On the defender the wall, which the rule also defines, gains 33.8% against 26.3%, and denial, a fact inherited from the base game, gains more, 32.1% against 23.8%. The self-play defender sits at 35.8%, between a random mover's 26.4% and the taught network's 75.1% (one seed). Table 28 also holds one board of another shape. Before the four-by-eight board was solved we fixed a threshold of 0.825, from the trend of the square boards, under which its shape would be read as mattering; the two distances alone read the win class there at 0.797, so by the rule fixed in advance board shape matters to how far the race predicts the game. Set beside the base game, the worlds return more handles, 26 to 196 per layer on the four world networks of Section 5.3, where the self-play agent of the large board holds 12 to 26, and they include facts that exist only under the changed rule, in a world where a perfect defender adds 15.1 plies to the attacker's arrival (Table 28). The two ranges do not measure what a rule adds, because they differ in board, dictionary and fact family, and a world's noise count belongs to its firing density (Appendix K).

Table 26: Self-play did not teach the worlds. How often the move played is an on-schedule move, on the positions where each fact holds, in the delay world: for a random mover, for a base-rules agent placed on the same positions, for the world's own self-play agent and for the taught network (one seed per agent). The first row of each block counts every scored position.

world, agent	positions where	positions	random mover	base-rules agent	the world's own agent	taught network
delay, self-play attacker	all positions	94,567	8.97%	50.5%	54.3%	81.0%
	`att_forced_wait`	36,072	7.99%	43.4%	47.0%	78.7%
	`att_greedy_inf`	10,211	9.32%	32.7%	43.6%	72.8%
	`att_greedy_wrong`	28,399	7.25%	36.4%	39.6%	73.0%
	`att_only_move`	53,676	4.82%	41.8%	45.0%	76.5%
	`att_sacrifice`	12,059	6.82%	40.6%	43.0%	81.3%
	`rule_forces_base_blunder`	1,930	5.65%	24.3%	28.3%	61.2%
	`some_schedule_base_blunder`	5,516	11.8%	55.9%	53.6%	78.3%
delay, self-play defender	all positions	81,590	26.4%	29.8%	35.8%	75.1%
	`att_stalled`	20,747	22.3%	26.3%	30.3%	81.2%
	`def_only_move`	29,637	6.04%	8.29%	12.3%	60.3%
	`denial`	15,336	16.5%	23.8%	32.1%	77.7%
	`wall_move`	36,960	20.1%	26.3%	33.8%	76.3%

Table 27: The same for the taught network of the cooperative world (one seed). The rows of the two taught networks of the delay world are the last column of Table 26.

world, agent	positions where	positions	random mover	base-rules agent	taught network
cooperative, taught	all positions	234,327	21.6%	41.3%	77.9%
	`coop_forced_wait`	71,113	16.9%	28.6%	71.2%
	`coop_only_move`	83,765	4.36%	18.8%	70.2%
	`coop_sacrifice`	13,515	12.0%	17.4%	65.1%
	`partner_reach_boost`	71,645	26.5%	45.2%	82.5%
	`rule_forces_base_blunder`	30,810	7.76%	10.0%	70.9%
	`some_schedule_base_blunder`	82,438	26.3%	42.9%	80.0%

Q WHAT COMES NEXT, IN FULL

Interpretability methods are now run on production language models and their features read as evidence about what those models represent (Templeton et al., 2024; Gao et al., 2025), where no concept claim can be checked against the truth of the domain. A solved domain is where a method and its validators can be given measured error rates first, before the method is trusted where no truth is available. The recipe needs an exact oracle and a trained model, so its reach is set by where oracles exist and not by the game; what a checked benchmark says about the methods it grades is a matter of measurement, not of the domain's size.

On a solved world, concept discovery can be automated end to end and every step of it checked. The fact family is the one human choice left in the pipeline. It can be generated from a stated grammar of board and table terms, or proposed by a language model, and filtered by the solved table through the gates the hand-written family passed. In the other direction, the grammar can be searched for

Table 28: Whole-state-space numbers of the solved rule-variant worlds and of one board of another shape. They are exact and have no seed.

quantity	value
cooperative world on the four-by-four board: plies of the optimal joint plan from the start	19
four-by-eight board: the threshold under which board shape matters, fixed before the solve	0.825
four-by-eight board: balanced accuracy of the distances alone on the win class	0.797
cooperative world: plies of the optimal joint plan from the start	20
delay world: balanced accuracy of the distances alone on 'the defender can delay'	0.831
delay world: base rate of 'the defender can delay'	0.963
delay world: plies a perfect defender adds, mean over positions	15.1
delay world: share of positions the defender holds for ever	0.0780

the term whose firing set best matches a causal handle that no fact names, so that the solved table names what the agent found (Peng et al., 2026).

Any method that returns a direction and a detector fits the report card. Among them are dictionaries of other forms, attribution-based methods, sets of units proposed with no fact labels, and the concept vector method at every layer and under more than one recipe. The same truth can grade validators we did not grade. A language model names a feature from its top-firing positions (Paulo et al., 2025), and the solved table says whether that name holds where the feature fires; steering scores (Turner et al., 2024) and aggregate dictionary metrics (Karvonen et al., 2025) can be graded the same way. The named-but-false cell of Figure 1 can be filled from the human side by grading the game's own lore, luring potential and the folk rules among it, against the solved table.

Agents trained on the unsolved 9×9 game can then be checked for the concepts validated here, as evidence about the agents and never a validation of truth. The benchmark carries over to any domain with an exact oracle, chess endgames under tablebases, other solved games, and settings where a checker decides every step, formal proof and program execution among them. Removing as many features of other facts on the same positions would show whether the cost of removal belongs to the fact, and would sharpen the set-level result.