ONE SEARCH, TWO VALUES: TEST-TIME SEARCH EFFICIENCY IN THREE-PLAYER GENERAL-SUM GAMES

Anonymous authors
Paper under double-blind review

ABSTRACT

With three or more players, a search should back up the value aligned with the players' objective. Objectives such as win probability or rank credit make that aligned value coarse: a player's actions are mostly exactly tied or far apart, so a few-simulation search over the aligned value is often left undecided. The cardinal score is dense and decisive but misaligned, since the score-optimal action can be strictly worse in win probability. We carry both values through one search. A tree search guided by the objective-aligned value head also backs up the cardinal head's vector through the same nodes, so both action-value vectors reach the root with no additional search simulation. A gated readout then plays the aligned decision where the aligned margin clears a calibrated gate threshold and the cardinal ordering where it does not. The readout leaves the tree unchanged, so its effect is confined to the states where it fires, and its expected gain is the firing probability times the accuracy difference of the two readouts on those states, which we measure. On an exactly solved eight-token ranking game and on a reduced three-player Chinese Checkers, the gated readout at 64 simulations per move beats the aligned readout at 64 and 128 simulations and the cardinal search at both budgets. The aligned readout needs about seven times the simulation budget to match it on the ranking game under the rank objective, and on Chinese Checkers it does not match it within 1024 simulations.

1 INTRODUCTION

Consider the first player of a three-player game choosing between two actions. One scores 6 for certain and never wins, because an opponent always finishes with 7; the other averages a score of 5 and wins half the time. A search that backs up expected score prefers the first action, and more simulations make it more certain. The value a search should back up in a game with more than two players is therefore the one aligned with the objective the players are judged by, and two-player board-game systems already back up the win itself (Silver et al., 2018; Schrittwieser et al., 2020). At three players, however, the aligned value has a cost that does not appear at two players. Objectives defined on final ranks, such as win probability or rank credit, are coarse. On the Tally probe states, the exact aligned values of a player's actions are either exactly tied or at least half the payoff range apart, and a few-simulation search over them is left undecided on a large share of positions. What the search then plays on those positions is set by convention rather than by the values. At three players the convention can matter, since actions tied for the mover need not be tied for the opponents (Proposition 2). On the same game and recipe at two players, the aligned readout's regret on the win network is less than a third of its three-player value (Appendix D.1). The cardinal score has the opposite properties: it is dense and decisive on almost every position, and wrong where score and rank disagree.

This paper asks whether a single few-simulation search can use both values, each where it is informative, at the cost of one search. We propose a search that carries two value vectors through one tree. The tree is grown by the objective-aligned value head. The same nodes also back up the cardinal head's vector under the same $\max^n$ rule, so when the search ends the root holds an aligned and a cardinal action-value vector. This costs no additional search simulation, because both heads share one network body and one forward pass per node. A gated readout then selects the vector

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.

whose action is played. If the aligned head separates its best action from the runner-up by at least a calibrated gate threshold, the aligned decision is played; otherwise the action that the cardinal vector ranks first, the cardinal ordering, is played.

Three findings support the design. The aligned and the cardinal readout fail on complementary states. As the aligned margin grows, the aligned readout's regret falls while the cardinal readout's regret changes little, and the two cross at two to four times the calibrated error scale (§3). Because the readout leaves the tree unchanged, its effect is confined to the states where it fires. Its expected gain equals the firing probability times the accuracy difference of the two readouts on those states, which the first finding measures directly (§4). Finally, the cardinal readout is not the exact score-optimal action. On the win network that action is rank-dominated often enough to be a worse decision than the aligned readout. The cardinal readout, restricted to the part of the tree the aligned head explored, is a better decision than that action there (§5.4).

We evaluate on Tally, an eight-token ranking game whose exact vector-max^{n} values we compute (48 token tables, each a different game, with three network seeds per table), and on a reduced three-player Chinese Checkers whose endgame band is exactly solvable. In full games against two copies of a fixed AlphaZero-style opponent, the gated readout at 64 simulations per move beats the aligned readout at 64 and at 128 simulations on both games and both training objectives. It also beats the cardinal search at 64 and 128 simulations and a re-search gate, which runs a second search with the cardinal head whenever the aligned search is undecided and therefore spends 109 to 128 simulations per move. In test-time search efficiency, the aligned readout needs about 440 simulations to match the gated readout at 64 simulations on Tally under the rank objective. Under the win objective its credit at 1024 simulations is slightly below, and the interval of the matching budget starts at 323 simulations. On Chinese Checkers, the gated readout at 8 simulations exceeds the aligned readout at 1024 simulations. The effect depends on the player count. On Tally at two players, with the same recipe, the gated readout's difference from the aligned readout at the same budget is not resolved, and it loses to the aligned readout at twice the budget. At four players the gain remains, with larger point estimates than at three (Appendix D.1). The contributions are (i) the finding that the aligned and cardinal readouts of a three-player search fail on complementary states ordered by the aligned margin, measured against an exact solver on two games; (ii) the dual-backup search and the gated readout, with an aligned-margin certification and an exact decomposition of its gain; (iii) an evaluation of test-time search efficiency along the simulation budget, in full games and on exact probe states, with every cost measured at the test seat, and ablations that separate the tree-guidance effect, the selection effect and the backup rule.

2 SETTING

Games and objectives. We study three-player, perfect-information, general-sum games in which every terminal state assigns each player a cardinal score. *Tally*: players alternately claim one of eight tokens, each token has a value for each player drawn once per instance from $\mathcal{N}(0, 5^2)$, and a player's score is the sum of the values, to that player, of the tokens it claimed. An instance is a token table; its $8!$-leaf tree is solved exactly, and we use the first 48 tables, in a fixed random order, whose win-optimal and score-optimal first moves differ. *Chinese Checkers*: a reduced three-player board (side-2 hexagram, two pegs per player, single jumps, 40-ply cap; Appendix B) with an exactly solvable endgame band; full games are played to the end. An *objective* maps final ranks to payoffs: the win objective $[1, 0, 0]$ with ties shared, and the rank objective $[1, \frac{1}{2}, 0]$ (a third, $[1, 1, 0]$, is reported in Appendix H). We call the payoff under an objective its *credit*. Because an objective is a function of ranks, the value aligned with it takes finitely many values. An objective equal to the cardinal score itself would make the cardinal view the aligned one, and the question of this paper would not arise.

Value views and heads. A *value view* is a transform of the terminal scores that a search backs up and a value head is trained on: the aligned views (win, rank) equal the objectives, and the *cardinal* view is the raw score. Views share dynamics and legal moves, so an action recommended under one is legal under every other. Each network has one body and three heads: a policy head, a primary value head predicting the aligned view of its training objective (the aligned head), and an auxiliary value head predicting the cardinal view (the cardinal head). We train one network per instance, seed

and objective (the *win network* and the *rank network*); the cardinal head is fitted to the same self-play games' final scores and adds no simulation to self-play (Appendix C).

Vector max^{n} search and exact values. All searches are per-player-vector variants of PUCT (Silver et al., 2018) and Gumbel MuZero (Danihelka et al., 2022). A node stores one value per player, and selection uses the mover's own component. The max^{n} backup (Luckhardt & Irani, 1986) replaces a node's vector by that of the child maximizing the mover's component among evaluated children, with the network's prediction standing in for unevaluated ones; ties go to the first child in move order. A search's decision is its most-visited root child. The budget is B simulations per move, $B{=}64$ unless stated. Every quantity we call exact comes from full max^{n} solving with the same convention. On Tally we score decisions on *probe states* by the exact regret of the recommended action. A probe state is a reachable decision-relevant state, one at which the mover has two or more legal actions of different exact value under the objective, and we draw up to twelve per table and objective. Full games are scored by *tournament credit*, the test seat's payoff under the objective against two copies of a fixed opponent (Appendix D).

Calibrated error scale and statistics. Per game and view we fix a calibrated error scale $\hat{\Delta}$. It is the mean absolute deviation between the exact action values and the root action values of a 64-simulation search whose value function returns zero everywhere, measured on states whose best action is unique. On Tally $\hat{\Delta}_{\mathrm{win}}{=}0.066$ and $\hat{\Delta}_{\mathrm{rank}}{=}0.090$, and on Chinese Checkers the two values are 0.187 and 0.311 (Appendix C). The gate uses only the product $c\,\hat{\Delta}$, the *gate threshold*, with $c{=}2$. The statistical unit is the token table on Tally ($n{=}48$) and the opening on Chinese Checkers ($n{=}32$). Comparisons are paired by game slot (the same table or opening, network seed, seat and repeat), averaged within the unit over the three network seeds, and reported with a 95% percentile bootstrap interval over units. We call a paired difference *resolved* when its interval excludes zero. Every number in the text and every difference in the tables is computed from the unrounded values, so it can differ by one unit in the last digit from the same arithmetic on the rounded entries of another table.

3 Why One Value Head Is Not Enough at Three Players

The dense value is misaligned. Backing up expected score is the default wherever the environment reports a score, and under an objective defined on final ranks it can be wrong at any number of players.

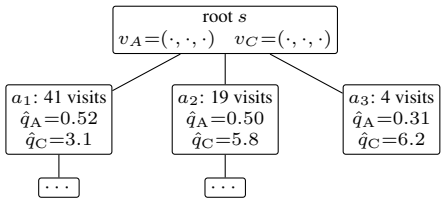
Proposition 1 (Rank domination)**.** *For every number of players $N \geq 2$ there are states in which the action maximizing expected score is strictly dominated in win probability: some other action wins strictly more often.*

The introduction's example is the proof (Appendix A.1). On the Tally rank network a search that backs up the cardinal head loses to a search that backs up the aligned head under the aligned objective, with a paired interval below zero. On Chinese Checkers the former scores above the latter (Table 1). On the probe states, even the *exact* score-optimal action is a worse decision than the aligned readout on the win network (Table 3).

The aligned view is coarse under objectives defined on final ranks. A win view assigns a player's action the value 0, $\frac{1}{3}$, $\frac{1}{2}$ or 1. On the Tally probe states, the exact aligned values of a mover's actions are either exactly tied or at least $\frac{1}{2}$ apart, and 49 to 55 percent of action pairs are exact ties (Appendix G.1). At a tie the value the search backs up is under-determined:

Proposition 2 (Kingmaker indeterminacy)**.** *(i) In two-player constant-sum views, children tied in the mover's component carry identical value vectors, so the exact value is convention-independent. (ii) At three or more players there are states whose exact max^{n} value differs across tie-break conventions.*

One node proves (ii): a mover tied between children valued $(\frac{1}{2}, \frac{1}{2}, 0)$ and $(\frac{1}{2}, 0, \frac{1}{2})$ is indifferent while the opponents are not (Appendix A.2). Beyond the exact ties, the few-simulation estimates of a coarse value are unreliable on a larger set. On the Tally probe states, the trained aligned readout at 64 simulations has an exact regret of 0.107 on the rank network and 0.161 on the win network. Under the win objective it therefore picks a losing action in about one probe state in six. On the



One tree, two vectors. Every node holds the aligned vector v_A, which grows the tree, and the cardinal vector v_C, backed up through the same nodes by the same max^n rule; both come from one forward pass of the shared network body.

Root readout. $a_A = a_1$ (most visited). Aligned margin $\hat{m}_A = 0.52 - 0.50 = 0.02$; gate threshold $c\,\hat{\Delta}_A = 0.18$.

Gate. $\hat{m}_A < c\,\hat{\Delta}_A$, so the aligned head has not separated its top two actions: play $a_C = \arg\max_a \hat{q}_C(a) = a_3$. Had the margin cleared the threshold, a_1 would be played and the cardinal vector ignored.

Figure 1: How the gated readout decides at one root. The search costs one budget of simulations; the readout adds no search and changes the tree in no way.

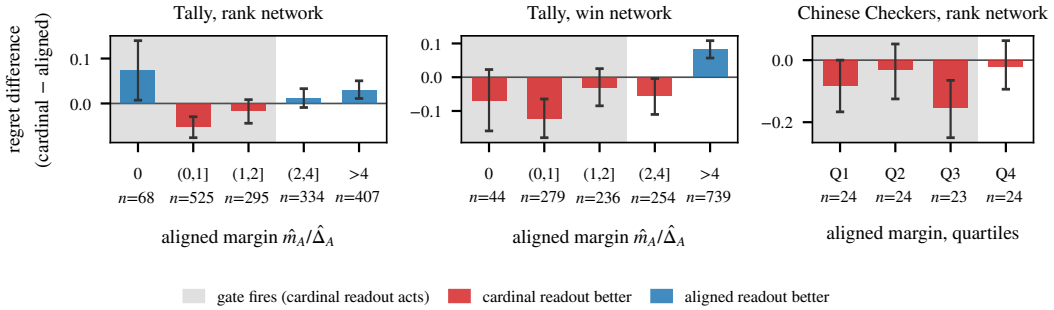


Figure 2: Why the rule helps: the two readouts of the same tree fail on complementary states. Bars: exact regret of the cardinal readout minus that of the aligned readout, both read from one 64-simulation dual-backup search, by band of the aligned margin (Tally: 48 tables, three seeds, n state-seed pairs per band, 95% bootstrap intervals over state-seed pairs; Chinese Checkers: 32 exactly solved, decision-relevant endgame states under the rank objective, margin quartiles). Negative bars mean the cardinal readout is the better decision. The gate plays the cardinal readout on every pair in the shaded bands and the aligned readout on every pair in the unshaded Tally bands. In the fourth Chinese Checkers quartile, unshaded, it plays the cardinal readout on 18 of 24 pairs. Data in Appendix F.

same game and recipe at two players, where part (i) applies, the win network's readout has a regret of 0.047. On the win network the share of exactly tied action pairs is about the same at two, three and four players (0.53, 0.55 and 0.54; Appendix D.1), so coarseness alone does not explain the failure. Our operating signal is the search's own aligned margin $\hat{m}_A$, the gap between the top two aligned action values over the visited root children, compared with the gate threshold. In the calibration model of Appendix A.3, a search whose margin clears $c\,\hat{\Delta}_A$ recovers the exact aligned action in most draws, and in more of them as the margin grows. The criterion needs only to separate the states where the aligned readout is reliable from those where it is not.

The two readouts fail on complementary states. Figure 2 compares the exact regret of the two readouts by band of the aligned margin. For every Tally probe state and seed we run one 64-simulation search guided by the aligned head that also backs up the cardinal head, the search of §4. We compare the exact regret of two readouts of the same tree: the most-visited action and the action that the cardinal vector ranks first among the visited root children. Where the aligned margin is positive but at most one $\hat{\Delta}_A$, the cardinal readout is the better decision on 16 to 19 percent of state-seed pairs and the worse one on 6 to 7 percent. Above four $\hat{\Delta}_A$ the numbers reverse, 2 to 4 percent against 10 to 11 percent, and the crossing lies between two and four $\hat{\Delta}_A$. From the first of these bands to the second, the aligned readout's regret falls from 0.19 to 0.05 on the rank network and from 0.37 to 0.06 on the win network. Across all bands, the cardinal readout's regret stays within 0.08 to 0.14 and 0.14 to 0.25 on the two networks (Appendix F). On the Chinese Checkers probe states the gate fires on 90 of the 96 state-seed pairs, including every pair in the first three margin quartiles. The cardinal readout has the lower mean regret in every quartile there, resolvably only in the third. The method of the next section is the rule that reads the cardinal head below the crossing and the aligned head above it.

Algorithm 1: gated readout at state s with budget B.

1. Run B simulations of PUCT with the max^{n} backup on the aligned head; back up the cardinal head's vector through the same nodes with the same rule.
2. Let a_{A} be the most-visited root child and $\hat{m}_A$ the gap between the largest and the second-largest $\hat{q}_{\text{A}}$ over visited root children (undefined if fewer than two were visited).
3. If $\hat{m}_A$ is defined and $\hat{m}_A \geq c\,\hat{\Delta}_A$: play a_{A}.
4. Otherwise (the gate *fires*): play $a_{\text{C}} = \arg\max_a \hat{q}_{\text{C}}(a)$ over visited root children, first in move order on ties.

Cost: B simulations on every move. Constants: c=2 and the calibrated error scale $\hat{\Delta}_A$ of the aligned view (§2), fixed before any experiment of this paper.

4 Method: Dual-Backup Search with a Margin-Gated Readout

Dual-backup search. The search is PUCT with the max^{n} backup on the aligned head, unchanged, plus one addition. Every node stores a second value vector, initialized from the cardinal head's prediction at the node or from the exact cardinal payoff at a terminal. After every simulation this vector is updated by the same max^{n} rule as the aligned vector, applied to the cardinal components, over the same evaluated children. Selection, expansion, visit counts and the aligned vector are untouched; the added vector is computed from quantities the network has produced. At the root, for every visited child a, the search exposes the aligned action value $\hat{q}_{\text{A}}(a)$, the cardinal action value $\hat{q}_{\text{C}}(a)$ (the mover's own components of the two backed-up vectors) and the visit count. Figure 1 shows one root; Algorithm 1 states the readout.

The gate and its cost. The readout has no fitted parameter; c and $\hat{\Delta}_A$ come from the offline calibration of §2, which uses no tournament data, and both are examined in §5.3. The gated readout spends exactly B simulations on every move. The alternative that runs a second search, a baseline throughout, is the *re-search gate*: run the aligned search and, if it is undecided, that is, if the gate of Algorithm 1 would fire, a second search of B simulations on the cardinal head, playing that search's most-visited action. Its cost is $B(1+f)$ with f the firing rate, 109 simulations per move on Tally and 127 to 128 simulations on Chinese Checkers at B=64. Every arm is compared at its measured cost at the test seat.

What the margin certifies. Because the readout leaves the tree unchanged, its effect is confined to the states where it fires. Under a bound on the estimation error, the margin certifies the aligned decision, and the root estimates bound the exact aligned action value that a switch forgoes.

Proposition 3 (Aligned-margin certification)**.** *Fix a state and let the root estimates satisfy $|\hat{q}_{\text{A}}(a) - q_{\text{A}}(a)| \leq \varepsilon$ for every visited child a, with q_{A} the exact aligned action values. Let $\tilde{a}_{\text{A}}$ be the child maximizing $\hat{q}_{\text{A}}$ and $\hat{m}_A$ the gap to the runner-up. (i) If $\hat{m}_A \geq 2\varepsilon$, then $\tilde{a}_{\text{A}}$ maximizes q_{A} among the visited children. (ii) If $c\,\hat{\Delta}_A \leq 2\varepsilon$, the readout that plays $\tilde{a}_{\text{A}}$ where $\hat{m}_A \geq c\,\hat{\Delta}_A$ and a_{C} elsewhere plays $\tilde{a}_{\text{A}}$ on every state that (i) certifies. (iii) The exact aligned action value forgone by playing a visited child a instead of $\tilde{a}_{\text{A}}$ satisfies $q_{\text{A}}(\tilde{a}_{\text{A}}) - q_{\text{A}}(a) \leq \hat{q}_{\text{A}}(\tilde{a}_{\text{A}}) - \hat{q}_{\text{A}}(a) + 2\varepsilon$; a switch to the runner-up on a fired state forgoes at most $\hat{m}_A + 2\varepsilon < c\,\hat{\Delta}_A + 2\varepsilon$.*

The proof and the origin of the constants are in Appendix A.3. The certification applies to this variant of the gated readout, which plays the maximizer $\tilde{a}_{\text{A}}$ of $\hat{q}_{\text{A}}$. The readout of Algorithm 1 plays the most-visited child a_{A} instead, and the certification does not extend to it; Appendix F measures this implementation gap.

What a switch gains. The gate fires when step 4 of Algorithm 1 applies, and it switches when the fired readout a_{C} differs from a_{A}; off the fired states the two readouts coincide, and so do they on fired states that do not switch. The gain depends on which readout is more often right on the fired states.

Proposition 4 (Decomposition of the gain)**.** *Let F be the event that the gate fires at a state drawn from a population, R_A and R_C the exact regrets of the aligned and the cardinal readout, and R_G*

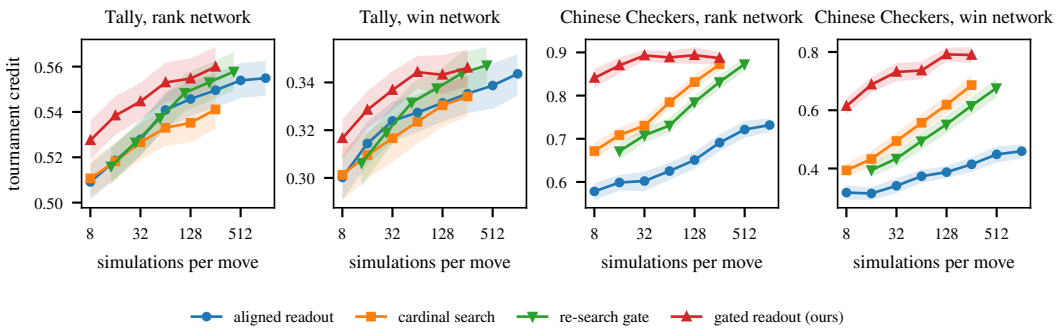


Figure 3: Tournament credit of the test seat against two copies of the win-trained opponent versus simulations per move, measured at the test seat, on 48 Tally tables and on Chinese Checkers, for the aligned readout, the cardinal search, the re-search gate (at its measured cost) and the gated readout; bands are 95% bootstrap intervals over tables or openings. The gate keeps its 64-simulation calibration at every budget. The same sweep on the exactly solved probe states is in Appendix E.

that of the gated readout. Then $\mathbb{E}[R_G] - \mathbb{E}[R_A] = \mathbb{P}(F)\,\mathbb{E}[R_C - R_A \mid F]$. *Under the win objective with a unique winner,* $R \in \{0, 1\}$ *and the conditional term equals* $\mathbb{P}(\textit{aligned correct} \mid F) - \mathbb{P}(\textit{cardinal correct} \mid F)$*: the gated readout improves on the aligned readout exactly when the cardinal readout is more often correct on the fired states.*

The identity is immediate, since the two readouts coincide off F (Appendix A.4). Its right-hand side is the quantity that Figure 2 measures. The gate fixes the trade-off between $\mathbb{P}(F)$ and the conditional accuracy, and the crossing of the two readouts is where the conditional term changes sign.

5 Experiments

Protocol. Every arm, a readout or search configuration under test, uses the same trained networks, the *test networks*: two dual-head networks per Tally table and per Chinese Checkers configuration, one per training objective, three seeds each, trained once and never re-tuned (Appendix C). The opponent in every tournament is the *win-trained opponent*: a single-head network of the same instance and seed, trained under the win objective, with a Gumbel max^n search at 64 simulations, an AlphaZero-style configuration. The test player is seated in each of the three seats in turn. Openings are fixed random-walk states, twelve two-ply openings per Tally table and 32 four-ply openings on Chinese Checkers, drawn from a seed used for no other purpose. Tally plays one game per opening and seat, and Chinese Checkers three (Appendix D). Credit is the test seat's payoff under the network's training objective. The arms are the *aligned readout* (PUCT on the aligned head, most-visited action) at B and $2B$; the *cardinal search* (PUCT on the cardinal head) at B and $2B$; the *re-search gate*; and the *gated readout* at B. The ablations add, all at B: the *always-cardinal readout*, the dual-backup search with the cardinal readout played on every move; the *learned selector*, a logistic rule on the two root gaps fitted on the probe states of 32 of the 48 tables (Appendix G); three alternative backups; fixed blends of the two heads in the KataGo style (Wu, 2019); and other values of c and $\hat{\Delta}$.

5.1 Credit versus simulation budget

Figure 3 gives the main result. On both Tally networks the gated readout is above the aligned readout at every budget from 8 to 256 simulations per move, with every interval above zero. The paired differences are $+0.009$ to $+0.020$ under the rank objective and $+0.011$ to $+0.017$ under the win objective. At every one of these budgets the gated readout is also above the aligned readout at twice its budget. Those intervals lie above zero at four of the six budgets on the rank network and three on the win network (Appendix E). Doubling the aligned readout's own budget from 64 to 128 simulations adds 0.005 and 0.004 credit on Tally and 0.025 and 0.014 on Chinese Checkers. On the Tally rank network, the aligned readout reaches the gated readout's 64-simulation credit at 439 simulations (interval 156 to beyond 1024). On the Tally win network, the aligned readout's credit at

Table 1: Tournament credit at 64 simulations (the aligned readout and the cardinal search also at 128 simulations) against two copies of the win-trained opponent, with the test seat's measured simulations per move; mean and 95% bootstrap interval over 48 tables (Tally, 5184 games per arm) or 32 openings (Chinese Checkers, CC; 864 games per arm), under each network's training objective. Bold marks the highest credit in each column among the baselines and the gated readout; the two ablation arms below the horizontal line are compared in Table 2.

		Tally		CC	
Arm	sims/move	rank network	win network	rank network	win network
Aligned readout, B	64	0.541 [.533, .549]	0.327 [.319, .335]	0.625 [.605, .645]	0.373 [.351, .396]
Aligned readout, $2B$	128	0.546 [.538, .553]	0.331 [.324, .339]	0.651 [.631, .670]	0.388 [.367, .408]
Cardinal search, B	64	0.533 [.525, .541]	0.324 [.314, .332]	0.785 [.770, .799]	0.557 [.525, .588]
Cardinal search, $2B$	128	0.535 [.527, .544]	0.330 [.321, .339]	0.831 [.814, .848]	0.619 [.588, .648]
Re-search gate	109–128	0.548 [.540, .557]	0.338 [.330, .344]	0.785 [.770, .799]	0.551 [.520, .581]
Gated readout (ours)	64	**0.553** [.544, .562]	**0.344** [.337, .351]	**0.888** [.874, .902]	**0.736** [.706, .767]
Always-cardinal readout	64	0.542 [.533, .550]	0.337 [.329, .345]	0.889 [.876, .903]	0.740 [.712, .769]
Learned selector	64	0.560 [.551, .568]	0.352 [.344, .359]	0.887 [.874, .899]	0.739 [.712, .766]

1024 simulations (0.3436) is below the gated readout's at 64 simulations (0.3443), and the interval's lower end is 323. On Chinese Checkers the gated readout at 8 simulations, 0.841 and 0.614 under the two objectives, exceeds the aligned readout at 1024 simulations, 0.732 and 0.460. Its credit there changes little beyond 32 simulations on the rank network and beyond 128 on the win network, while the aligned readout and the cardinal search keep rising to the end of the measured range. On the exactly solved probe states, scored by regret, the gated readout is below the aligned readout at every budget up to 128. There the aligned readout needs 179 simulations (interval 103 to 337) on the rank network and 239 (110 to 461) on the win network to match the gated readout at 64 simulations (Appendix E).

5.2 Full games at one budget

Table 1 reports every arm's credit at 64 simulations and Table 2 the paired differences. The gated readout beats the aligned readout at the same budget by $+0.012$ and $+0.017$ on Tally and by $+0.263$ and $+0.363$ on Chinese Checkers. Against the aligned readout at twice the budget, the differences are $+0.007$ and $+0.013$ on Tally and $+0.238$ and $+0.349$ on Chinese Checkers, and every interval is above zero. It beats the cardinal search at the same and at twice the budget. It also beats the re-search gate, which spends 109 simulations per move on Tally and 127 to 128 simulations on Chinese Checkers, by $+0.004$ and $+0.007$ on Tally and $+0.104$ and $+0.185$ on Chinese Checkers. The re-search gate's differences from the aligned readout at 128 simulations on Tally, $+0.003$ and $+0.006$, are not resolved.

5.3 Ablations

The lower blocks of Table 2 separate the three effects of the method. *Selection.* On Tally the always-cardinal readout is 0.011 and 0.007 below the gated readout. On the rank network its difference from the aligned readout is not resolved and it loses almost all of the gain; on the win network it retains part of the gain. On Chinese Checkers the gate fires on 99 percent of moves, so the two nearly coincide. The learned selector, fitted on the probe states of 32 tables, is 0.007 better than the gate over all 48 tables. On the 16 tables it never saw it is not resolvably better ($+0.004$, interval -0.001 to $+0.009$, and $+0.005$, -0.001 to $+0.011$), and on Chinese Checkers it matches the gate. The paper's claims use the gate, which has no fitted parameter; the learned selector is reported as an ablation. *Tree guidance.* The always-cardinal readout beats the cardinal search at the same budget on both games, by $+0.009$ and $+0.014$ on Tally and $+0.105$ and $+0.183$ on Chinese Checkers. Growing the tree with the aligned head and reading the cardinal vector is therefore better than growing the tree with the cardinal head, before any selection. *Backup rule.* Carrying the cardinal vector along the aligned rule's choice at every node costs 0.006 and 0.008 on Tally. A lexicographic interior that breaks exact aligned ties by the cardinal component costs 0.007 on the win network and 0.004, not resolved, on the rank network, and a banded lexicographic interior is similar (Table 13). On Chinese Checkers none of the three differs resolvably from the backup of §4. No variant is resolvably better than it. *Fixed blends and constants.* The best fixed blend of the two heads in search and readout,

Table 2: Paired differences, gated readout minus arm, on the same games (positive favors the gated readout), with 95% bootstrap intervals over units; columns as in Table 1; every arm below the first block spends 64 simulations per move. Bold marks the baseline comparisons whose interval lies entirely above zero. The remaining blends and constants are in Table 13.

	Tally		CC	
Gated readout minus arm	rank network	win network	rank network	win network
Baselines				
Aligned readout at B	**+.012** [.006,.018]	**+.017** [.010,.024]	**+.263** [.241,.285]	**+.363** [.325,.400]
Aligned readout at $2B$	**+.007** [.001,.014]	**+.013** [.005,.021]	**+.238** [.216,.259]	**+.349** [.315,.382]
Cardinal search at B	**+.020** [.014,.026]	**+.021** [.012,.030]	**+.104** [.087,.121]	**+.179** [.152,.206]
Cardinal search at $2B$	**+.018** [.012,.024]	**+.014** [.006,.022]	**+.057** [.038,.079]	**+.117** [.083,.152]
Re-search gate	**+.004** [.000,.009]	**+.007** [.002,.012]	**+.104** [.087,.121]	**+.185** [.158,.213]
Selection				
Always-cardinal readout	+.011 [.007,.016]	+.007 [.001,.013]	-.001 [-.003,.000]	-.004 [-.008,.000]
Learned selector	-.007 [-.010,-.003]	-.007 [-.011,-.004]	+.002 [-.006,.009]	-.003 [-.023,.017]
Backup variants				
carried along the aligned choice	+.006 [.002,.009]	+.008 [.003,.013]	+.011 [-.007,.028]	-.003 [-.036,.032]
lexicographic interior, exact ties	+.004 [-.000,.008]	+.007 [.000,.013]	+.011 [-.007,.028]	-.003 [-.036,.032]
Fixed blends				
Fixed blend λ=0.75	+.007 [.001,.012]	+.005 [-.001,.012]	+.131 [.115,.147]	+.218 [.183,.253]
Robustness				
Gate constant c=1	+.004 [.001,.006]	+.004 [.001,.007]	+.001 [-.001,.004]	+.009 [.002,.017]

λ=0.75, adds 0.005 and 0.012 over the aligned readout on Tally. The gated readout is above it by 0.007 (interval 0.001 to 0.012) and 0.005 (-0.001 to 0.012). On Chinese Checkers every blend is near or below the cardinal search, and the gated readout beats the best one by 0.10 and 0.19. The gate constant c=1 fires less often and loses 0.004 on Tally and 0.009 on the Chinese Checkers win network, and c=4 is within 0.004 of c=2 and above it on the Tally win network. Replacing $\hat{\Delta}_A$ by the other game's value loses up to 0.016 but stays at or above the aligned readout in every column (Table 13).

5.4 Where the gain comes from and where it stops

Table 3 scores five decisions on the Tally probe states with the exact solver. An oracle that picked, on every state, the better of the two readouts would remove 46 and 49 percent of the aligned readout's regret; the gated readout removes 15 and 17 percent. The gate fires on 56 and 38 percent of probe state-seed pairs, on 70 percent of Tally tournament moves and on 99 percent of Chinese Checkers moves. The exact score-optimal action, computed by the solver, is a worse decision than the aligned readout on the win network, consistent with Proposition 1. On the win network the cardinal readout, which ranks the visited root children by the cardinal max^n values of the subtree the aligned head explored, is a better decision than the exact score-optimal action. The gated readout is better still. A control experiment in Appendix G.1 shows that the gain is not a near-tie effect. With isotropic noise injected into the exact values, the gate does not help at any point of the grid, because the exact aligned values are discrete and the gate then fires only at exact ties. The gain of the trained system therefore comes from the heads' structured errors: the margin identifies states where the aligned readout is wrong more often, and on those states the cardinal readout is more often right. For this reason Proposition 4 is stated as a decomposition to be measured rather than as a property of a noise model.

The gain is above zero against each of six opponents trained on 300 to 10,000 self-play games and searching at 64 or 256 simulations per move, and at two players on Tally it is not resolved (Appendices D.1 and D.2). With longer training it shrinks on both games: at 10,000 self-play games it is below its 300-game value against every opponent, at zero or below on Tally and at 24 and 15 percent of its 300-game value on Chinese Checkers. Both readouts change with training. The aligned readout's probe-state regret falls, and where the gate switches the cardinal readout, the better decision at 300 games, has lost its advantage by 10,000 on all four networks; the gate reads only the aligned margin and keeps switching, so on Tally switching no longer improves the decision. On Chinese Checkers the probe-state decomposition of Proposition 4 no longer favors the gated readout at 10,000 games while the gain in games stays above zero; the two measure different populations (Appendix D.2, Table 8).

Table 3: Five decisions on the Tally probe states, scored by exact regret under the network's training objective (48 tables, three seeds; mean and 95% bootstrap interval over tables). The exact score-optimal action is computed by the solver, not by a search. Bold marks the lowest regret in each column among the decisions above the horizontal line; the oracle below the line is not implementable.

Decision (readouts at B=64; exact regret)	Rank network	Win network
Aligned readout	0.107 [0.091, 0.123]	0.161 [0.133, 0.190]
Cardinal readout of the same tree	0.101 [0.086, 0.115]	0.182 [0.138, 0.232]
Gated readout (ours)	**0.090** [0.077, 0.103]	**0.133** [0.105, 0.161]
Exact score-optimal action (solver)	0.118 [0.098, 0.138]	0.229 [0.177, 0.286]
Oracle choice between the two readouts	0.058 [0.047, 0.070]	0.082 [0.061, 0.104]

6 Related Work

max^n search is classical (Luckhardt & Irani, 1986; Sturtevant & Korf, 2000), and its tie-break convention is known to determine the backed-up vector (Luckhardt & Irani, 1986; Sturtevant, 2003); multi-player tree search, Monte Carlo (Sturtevant, 2008; Nijssen & Winands, 2011) and minimax-style (Schadd & Winands, 2011), has been tuned without learned value networks. Per-player value vectors backed up per player inside an AlphaZero loop appear in Petosa & Balch (2019), which reports no curve of performance against its own simulation budget. The few-simulation construction of Gumbel MuZero (Danihelka et al., 2022) and the toolkits that reproduce it (Niu et al., 2023) include no board game with more than two players. Equal-share learning of symmetric constant-sum multiplayer games (Ge et al., 2025) and multiagent Gumbel planning in cooperative games (Hao et al., 2024) address other axes of the setting. KataGo adds a score utility to the win utility with a fixed, state-independent weight (Wu, 2019), and the choice of score against win rate as a reward has been studied in score-based games (Pasqualini et al., 2022). Our fixed-blend baseline follows the KataGo design, and the gated readout is a calibrated per-state alternative to it. Metareasoning about which computation to run (Hay et al., 2012) and sequential halving, which splits a fixed simulation budget across actions by successive elimination (Pepels et al., 2014; Ugadiarov et al., 2026), allocate simulations; our readout allocates nothing and selects between two values one search already computed. The misalignment between a dense proxy and the objective is an instance of optimizing an objective that only imperfectly captures the intent (Amodei et al., 2016), studied empirically as reward misspecification (Pan et al., 2022).

7 Limitations and Conclusion

The results come from two games with small networks and one Chinese Checkers configuration with a single network set. Only Tally is also played at two and four players, since Chinese Checkers fixes three by its board. The Tally tables were selected for a disagreement between the win-optimal and the score-optimal first move, and the gain persists on 32 random tables drawn without that filter (Appendix D). The method changes only test-time search: training is unchanged, and we make no claim about learning-time efficiency. The gain comes from structured errors of the trained heads, which we measure but do not model. The gate reads only the aligned margin and leaves about two thirds of the regret that the oracle of Table 3 removes. Its two constants come from an offline calibration under an isotropic noise model, which is not the mechanism of the gain (Appendix G.1).

Under an objective defined on final ranks, at three players, a search that backs up the objective-aligned value is often undecided, and the cardinal value is decisive but misaligned. One tree can carry both values, and one search can read the aligned decision where the aligned margin clears the gate threshold and the cardinal ordering where it does not. The always-cardinal readout beats the cardinal search on both games, and on Tally the gated readout beats it in turn. To match this readout at 64 simulations, the aligned readout needs about seven times the simulations on Tally under the rank objective and more than sixteen times on Chinese Checkers.

AI USE STATEMENT

In this work, we used generative AI tools (Claude Opus and Claude Fable, accessed through Claude Code) for implementing methods, namely the experiment code that runs the searches, the training loops and the tournaments and the scripts that generate the tables and figures; for the design of and feedback on the experiments, including the pre-registered predictions and the ablation plan; for formulating the propositions and drafting their proofs; for interpreting the results from the committed result files; for drafting parts of the manuscript; and for assistance with translation between the authors' working language and English. We have not used generative AI tools to generate synthetic data sets, and dataset cleaning or reformatting and qualitative or thematic data analysis are not applicable to this work. Additionally, we used generative AI tools to edit the manuscript for readability, to create or edit software code, to suggest the structure of the paper and to search for and check references. We have reviewed all AI-assisted work: the authors approved the design of every experiment, every experiment was run on the authors' hardware from the committed code under the authors' direction, the authors checked every reported number against the result files, reviewed every proposition and proof line by line, verified the AI-assisted code with the test suite of Appendix I, which checks the searches against the exact solver, checked every cited reference against its published record, and reviewed and edited the final manuscript in full. All final content and its logical verification are the authors' own. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Appendix I describes how every experiment is a deterministic function of recorded seeds and names the test suite that checks the searches against the exact solver and every added option against its default. Appendix B and Appendix C give the games and configurations, the network architecture, the training recipe and the calibration of the two gate constants; Appendix D gives the tournament protocol, the populations of tables and openings and the bootstrap procedure; Appendix A gives the proofs of the four propositions with their assumptions. Every table and figure in the paper is generated by a committed script from the committed result files, and the anonymized repository at `https://anonymous.4open.science/r/Test-Time-Search-Efficiency-in-Three-Player-General-Sum-Games-DC77/` contains the code, the trained networks, the result files and those scripts, with a README that maps each part of the paper to them.

REFERENCES

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in AI safety. *arXiv:1606.06565*, 2016.

Ivo Danihelka, Arthur Guez, Julian Schrittwieser, and David Silver. Policy improvement by planning with Gumbel. In *Proc. Int. Conf. Learning Representations (ICLR)*, 2022.

Jiawei Ge, Yuanhao Wang, Wenzhe Li, and Chi Jin. Securing equal share: A principled approach for learning multiplayer symmetric games. In *Proc. Int. Conf. Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 18989–19010. PMLR, 2025. arXiv:2406.04201.

Xiaotian Hao, Jianye Hao, Chenjun Xiao, Kai Li, Dong Li, and Yan Zheng. Multiagent Gumbel MuZero: Efficient planning in combinatorial action spaces. *Proc. AAAI Conf. Artificial Intelligence*, 38(11):12304–12312, 2024.

Nicholas Hay, Stuart Russell, David Tolpin, and Solomon Eyal Shimony. Selecting computations: Theory and applications. In *Proc. Conf. Uncertainty in Artificial Intelligence (UAI)*, pp. 346–355. AUAI Press, 2012.

Carol A. Luckhardt and Keki B. Irani. An algorithmic solution of n-person games. In *Proc. AAAI Conf. Artificial Intelligence (AAAI)*, pp. 158–162, 1986.

J. A. M. Nijssen and Mark H. M. Winands. Enhancements for multi-player Monte-Carlo tree search. In *Computers and Games (CG 2010)*, volume 6515 of *Lecture Notes in Computer Science*, pp. 238–249. Springer, 2011.

Yazhe Niu, Yuan Pu, Zhenjie Yang, Xueyan Li, Tong Zhou, Jiyuan Ren, Shuai Hu, Hongsheng Li, and Yu Liu. LightZero: A unified benchmark for Monte Carlo tree search in general sequential decision scenarios. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023), Datasets and Benchmarks Track*, 2023. arXiv:2310.08348.

Alexander Pan, Kush Bhatia, and Jacob Steinhardt. The effects of reward misspecification: Mapping and mitigating misaligned models. In *Proc. Int. Conf. Learning Representations (ICLR)*, 2022.

Luca Pasqualini, Maurizio Parton, Francesco Morandin, Gianluca Amato, Rosa Gini, Carlo Metta, Marco Fantozzi, and Alessandro Marchetti. Score vs. winrate in score-based games: which reward for reinforcement learning? In *21st IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 573–578, 2022. doi: 10.1109/ICMLA55696.2022.00099. arXiv:2201.13176.

Tom Pepels, Tristan Cazenave, Mark H. M. Winands, and Marc Lanctot. Minimizing simple and cumulative regret in Monte-Carlo tree search. In *Computer Games Workshop (CGW 2014), Revised Selected Papers*, volume 504 of *Communications in Computer and Information Science*, pp. 1–15. Springer, 2014.

Nick Petosa and Tucker Balch. Multiplayer AlphaZero. Deep RL Workshop, NeurIPS; arXiv:1910.13012, 2019.

Maarten P. D. Schadd and Mark H. M. Winands. Best reply search for multiplayer games. *IEEE Trans. Computational Intelligence and AI in Games*, 3(1):57–66, 2011. doi: 10.1109/TCIAIG.2011.2107323.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, Timothy Lillicrap, and David Silver. Mastering Atari, Go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020. doi: 10.1038/s41586-020-03051-4.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science*, 362(6419):1140–1144, 2018. doi: 10.1126/science.aar6404.

Nathan R. Sturtevant. A comparison of algorithms for multi-player games. In *Computers and Games (CG 2002)*, volume 2883 of *Lecture Notes in Computer Science*, pp. 108–122. Springer, 2003.

Nathan R. Sturtevant. An analysis of UCT in multi-player games. In *Computers and Games (CG 2008)*, volume 5131 of *Lecture Notes in Computer Science*, pp. 37–49. Springer, 2008.

Nathan R. Sturtevant and Richard E. Korf. On pruning techniques for multi-player games. In *Proc. AAAI Conf. Artificial Intelligence (AAAI)*, pp. 201–207, 2000.

Leonid Ugadiarov, Yuri Kuratov, Aleksandr Panov, and Alexey Skrynnik. Revisiting tree search for LLMs: Gumbel and sequential halving for budget-scalable reasoning. In *Proc. Int. Conf. Automated Planning and Scheduling (ICAPS)*, volume 36, pp. 713–717, 2026. arXiv:2603.21162.

David J. Wu. Accelerating self-play learning in Go, 2019. arXiv:1902.10565.

A Proofs

A.1 Rank domination (Proposition 1)

Proof (counterexample). Give player 0 two actions whose outcomes are lotteries over terminal payoff vectors: $\texttt{risky} = \{(10,3,4)$ w.p. $\frac{1}{2},\ (0,7,8)$ w.p. $\frac{1}{2}\}$ and $\texttt{greedy} = \{(6,7,2)$ w.p. $1\}$. Then $\mathbb{E}[u_0(\texttt{risky})] = 5$ with $\mathbb{P}(\text{win}_0 \mid \texttt{risky}) = \frac{1}{2}$, while $\mathbb{E}[u_0(\texttt{greedy})] = 6$ with $\mathbb{P}(\text{win}_0 \mid \texttt{greedy}) = 0$, since player 1 always leads with 7. The action maximizing expected score is `greedy`; its win probability, 0, is strictly dominated. The lotteries stand for subtrees whose

leaves realize the listed vectors at the listed frequencies; a deterministic perfect-information witness lets an indifferent opponent choose between the two leaves, with the consequence that the win probability then depends on how that opponent breaks its tie, which is why the proposition claims nothing about how indifference is resolved. The third player plays no part: with $(10, 3)$ against $(0, 7)$ and $(6, 7)$ the same numbers prove the two-player case, and any number of further players with fixed scores below 7 proves every $N \geq 2$. □

Two remarks. Win probability is not a function of per-player expected scores: two states with identical score means and different variances have different win probabilities, so no value head regressing expected score can represent it, whereas $\mathbb{P}(\text{win}_i)$ is itself a per-player scalar and is regressed directly by our aligned heads. Second, the win signal admits no per-move reward decomposition: the contribution of one transition depends on how the others finish, so it is irreducibly terminal.

A.2 KINGMAKER INDETERMINACY (PROPOSITION 2)

In two-player constant-sum views ($V_1 + V_2 = k$ at terminals; win and rank views have k=1), the constant sum propagates to every interior node by induction: the mover takes $V_i = \max_a V_i(\text{child}_a)$ and the opponent holds $V_j = k - V_i$. If the mover ties between children a, a', then $V_j(\text{child}_a) = k - V_i(\text{child}_a) = V_j(\text{child}_{a'})$: tied children carry identical value vectors, every tie-break convention backs up the same vector, and the value is convention-independent, which is part (i). At three or more players the induction fails because the mover's own component no longer determines the rest of the vector: let the mover tie between children valued $(\frac{1}{2}, \frac{1}{2}, 0)$ and $(\frac{1}{2}, 0, \frac{1}{2})$, as when a shared win is split with either opponent. The mover's component is $\frac{1}{2}$ either way while players 2 and 3 exchange payoffs, so the first and last conventions back up different vectors, which is part (ii). □

A.3 CERTIFICATION OF THE ALIGNED READOUT (PROPOSITION 3)

Let V be the set of visited root children, $\tilde{a}_\text{A} = \arg\max_{a \in V} \hat{q}_\text{A}(a)$, and $\hat{m}_A = \hat{q}_\text{A}(\tilde{a}_\text{A}) - \max_{a \in V \setminus \{\tilde{a}_\text{A}\}} \hat{q}_\text{A}(a)$. (i) For any $a \in V$, $a \neq \tilde{a}_\text{A}$: $q_\text{A}(a) \leq \hat{q}_\text{A}(a) + \varepsilon \leq \hat{q}_\text{A}(\tilde{a}_\text{A}) - \hat{m}_A + \varepsilon \leq \hat{q}_\text{A}(\tilde{a}_\text{A}) - \varepsilon \leq q_\text{A}(\tilde{a}_\text{A})$ whenever $\hat{m}_A \geq 2\varepsilon$. (ii) The variant plays $\tilde{a}_\text{A}$ whenever $\hat{m}_A \geq c\,\hat{\Delta}_A$; if $c\,\hat{\Delta}_A \leq 2\varepsilon$, every state with $\hat{m}_A \geq 2\varepsilon$ satisfies the condition, so no state certified by (i) is changed. (iii) For any $a \in V$, $q_\text{A}(\tilde{a}_\text{A}) - q_\text{A}(a) \leq (\hat{q}_\text{A}(\tilde{a}_\text{A}) + \varepsilon) - (\hat{q}_\text{A}(a) - \varepsilon)$; for the runner-up the estimated gap is $\hat{m}_A$, and on a fired state $\hat{m}_A < c\,\hat{\Delta}_A$. □

Calibration of $\hat{\Delta}$ and c. The error scale $\hat{\Delta}$ and the constant c=2 were calibrated offline, before any experiment of this paper, in the following model. If the root estimates are $\hat{q}_\text{A}(a) = q_\text{A}(a) + \varepsilon(a)$ with $\varepsilon(a)$ independent $\mathcal{N}(0, \sigma^2)$, the probability that the estimated argmax is the exact one is an orthant probability of an equicorrelated Gaussian vector in the standardized gaps; it is bounded above by $\Phi(m/(\sqrt{2}\sigma))$ in the minimum margin m and below by a union bound, so the recovery rate is a sigmoid in m/σ whose effective constant c is at least $\sqrt{2}$ and is measured near 2 on the gap structures of these games. The error scale $\hat{\Delta} = \mathbb{E}|\hat{q}_\text{A} - q_\text{A}| = \sigma\sqrt{2/\pi}$ under the model is estimated with a search whose value function returns zero everywhere, at 64 simulations, on states whose best action is unique, three search draws per state, and c is that effective constant, fitted under injected noise to the exact action values at the roots of 600 random three-player matrix games, where the fit gives $c = 2.1$ with the recovery rate at zero margin fixed at one half. The model supplies the shape of the criterion and the values of the two constants; Appendix G.1 shows that it does not supply the mechanism of the gain.

A.4 DECOMPOSITION OF THE GAIN (PROPOSITION 4)

On the complement of F the gated readout plays a_A, so $R_G = R_A$ there; on F it plays a_C, so $R_G = R_C$. Hence $\mathbb{E}[R_G - R_A] = \mathbb{E}[(R_C - R_A)\mathbf{1}_F] = \mathbb{P}(F)\,\mathbb{E}[R_C - R_A \mid F]$. Under a win objective with a unique winner every action is either exactly optimal (R=0) or loses the win (R=1), so $\mathbb{E}[R_C - R_A \mid F] = \mathbb{P}(R_C{=}1 \mid F) - \mathbb{P}(R_A{=}1 \mid F)$, the difference of the two readouts' error rates on the fired states. □

B GAMES AND CONFIGURATIONS

Tally. Eight tokens; three players move in turn, the player count being a parameter of the game that Appendix D.1 sets to two and four; a move claims one remaining token; the game ends when all tokens are claimed. Token values are drawn per instance from $\mathcal{N}(0, 5^2)$ independently per token and player, and a player's cardinal score is the sum of the values, to that player, of its claimed tokens. The win view assigns 1 to the highest score, shared equally on ties; the rank view assigns $[1, \frac{1}{2}, 0]$ by rank with ties sharing the tied ranks' credit. Instances are the first 48 seeds, in a fixed order, whose win-optimal and score-optimal first moves differ; the full tree of an instance is solved exactly.

Chinese Checkers. A side-2 hexagram with two pegs per player on alternating corners, forward moves and single jumps, and a 40-ply cap, a configuration selected by a solvability check: for states k plies before the terminal of random playouts we record the solve rate under a 5×10^5-node budget, the tie-break sensitivity of the exact value vector, and the mover's decision relevance. Table 4 reproduces the check; the adopted configuration has an exactly solvable endgame band ($k \leq 8$) with kingmaker states, whose exact value vector depends on the tie-break convention (Proposition 2). Full games are played from four-ply openings to the end; the profile of Figure 2 uses 32 exactly solved, decision-relevant endgame states under the rank objective, sampled from random playouts.

Table 4: Solvability check of the Chinese Checkers configurations (20 trajectories per configuration; k = plies before the terminal). The kingmaker fraction is the share of solved states whose exact value vector depends on the tie-break convention, and the decision-relevance fraction the share of solved states at which the mover has two or more legal actions of different exact value.

config (side,pegs,cap)	k	n	solve rate	kingmaker	decision-rel.
(2,2,40)	2	20	1.00	0.10	0.35
(2,2,40)	4	20	1.00	0.20	0.40
(2,2,40)	6	20	1.00	0.45	0.40
(2,2,40)	8	20	0.95	0.53	0.32
(2,2,30)	2	20	1.00	0.10	0.35
(2,2,30)	4	20	1.00	0.00	0.60
(2,2,30)	6	20	1.00	0.45	0.45
(2,2,30)	8	20	0.70	0.43	0.43
(2,3,60)	2	20	1.00	0.00	0.35
(2,3,60)	4	20	1.00	0.25	0.40
(2,3,60)	6	20	1.00	0.40	0.15
(2,3,60)	8	20	0.15	0.67	0.33
(1,1,20)	2	20	1.00	0.00	0.25
(1,1,20)	4	20	1.00	0.00	0.50
(1,1,20)	6	20	1.00	0.00	0.70
(1,1,20)	8	18	1.00	0.00	0.50

C TRAINING AND CALIBRATION

Networks. A multilayer perceptron with two hidden layers of 128 ReLU units on a one-hot board encoding, a policy head over the action space, a primary value head with one output per player, and an auxiliary value head with one output per player. The primary head regresses the aligned view of the training objective (win or rank), the auxiliary head the cardinal view divided by a per-game constant s, the standard deviation of the cardinal outcomes of random playouts (at least 1). The opponent network is the same architecture with the primary head only, trained on the win view.

Self-play. 300 self-play games per network; moves chosen by a Gumbel max^n search with 32 simulations and 16 considered actions at temperature 1; a replay buffer of 50,000 positions; after every game, four gradient steps on batches of 128 with Adam at learning rate 10^{-3} once 128 positions are stored; the auxiliary loss has coefficient 1. Seeds 0, 1, 2 per instance and objective. The Tally networks are per token table (48 tables, two objectives, three seeds); Chinese Checkers has one network per objective and seed.

Calibrated error scale. $\hat{\Delta}$ per game and view, from a 64-simulation Gumbel max^n search whose value function returns zero everywhere, on reachable states whose exact best action is unique, drawn from the first six Tally tables (up to four per table) and from the solvable Chinese Checkers endgame band, three search draws per state: Tally win 0.0658, rank 0.0898, cardinal 5.20; Chinese Checkers win 0.187, rank 0.311, cardinal 5.16. The gate uses the aligned view's value with $c{=}2$; the learned selector also uses the cardinal error scale, and the fixed blends divide the cardinal head by the per-game constant s.

D Tournament Protocol and Populations

Opponents and seating. The opponent is the single-head win-trained network of the same instance and seed with a Gumbel max^n search at 64 simulations and 16 considered actions. The test player occupies each of the three seats in turn against two copies of that opponent; the terminal is scored under every objective view for the test seat.

Openings. Tally: twelve two-ply openings per table, drawn by uniform random walks from a fixed seed used for no other purpose (Tally has 56 two-ply openings). Chinese Checkers: 32 four-ply openings drawn the same way. Tally plays one game per opening and seat per network seed (108 games per table and arm; 5184 per arm), Chinese Checkers three (864 games per arm). Random seeds are keyed by opening, seat and repeat, so every arm faces the same opponent draws.

Units and intervals. Paired differences are computed per game slot, averaged within the table (Tally) or opening (Chinese Checkers), and bootstrapped over units with 10,000 resamples; intervals are the 2.5th and 97.5th percentiles. Equivalent budgets interpolate the aligned readout's credit linearly in $\log_2 B$ between measured budgets and are bootstrapped over units with both curves recomputed per resample; values beyond the largest measured budget are reported as such.

No tuning on the evaluation populations. The gated readout has no fitted parameter; c and $\hat{\Delta}$ come from the offline calibration of Appendix A.3 and were fixed before any tournament was played. The learned selector of Appendix G is the only fitted rule; it is fitted on the probe states of the first 32 tables in the fixed order and reported separately on the remaining 16 tables, which it never saw. The populations, arms and primary comparisons of Table 2's first block were recorded before the first game, and every arm that was run is reported.

A stress test on unfiltered tables. The 48 Tally tables of the main text were selected, before any network was trained, for a disagreement between the win-optimal and the score-optimal first move, so that the two value views differ on every table. That selection could favor the gated readout. We therefore drew 32 further tables from fresh token seeds without any filter, trained the same three networks per table and seed (both dual-head networks and the win-trained opponent, same recipe), and ran the same tournament protocol (twelve openings per table, three seats, three seeds; 3456 games per arm). Table 5 sets the two populations side by side. The unfiltered tables have a root disagreement between the win and the cardinal view on 25 of the 32 tables (and between the rank and the cardinal view on 21), so the filter removes only the minority of tables on which the two views agree from the first move. On the unfiltered tables the gated readout beats the aligned readout at the same budget by $+0.017$ (interval $+0.009$ to $+0.025$, rank network) and $+0.013$ ($+0.005$ to $+0.021$, win network), against $+0.012$ and $+0.017$ on the filtered tables; against the aligned readout at twice the budget the difference is $+0.008$ ($+0.000$ to $+0.016$) on the rank network and $+0.003$ (-0.006 to $+0.011$) on the win network, where the filtered tables gave $+0.007$ and $+0.013$. The gated readout beats the cardinal search and the re-search gate on both populations, and the gate fires on 70 to 71 percent of moves on both. The selection enriches disagreement but does not create the effect.

Cost. Every arm's simulations per move are measured at the test seat and reported in Table 1 and Figure 3; the opponents' cost is the same for every arm.

Table 5: The filtered tables of the main text against 32 unfiltered random tables (training objective; unit = table; 95% bootstrap intervals). Credit of every arm at 64 simulations (the aligned readout also at 128 simulations) and the paired difference of the gated readout from it; the gated readout's firing rate in parentheses.

Tables	Arm	Rank network		Win network	
		credit	gated − arm	credit	gated − arm
Filtered (48 tables)	aligned readout, B	0.541	+.012 [.006,.018]	0.327	+.017 [.010,.024]
	aligned readout, $2B$	0.546	+.007 [.001,.014]	0.331	+.013 [.005,.021]
	cardinal search, B	0.533	+.020 [.014,.026]	0.324	+.021 [.012,.030]
	re-search gate	0.548	+.004 [.000,.009]	0.338	+.007 [.002,.012]
	gated readout (ours)	0.553 (fires 0.70)		0.344 (fires 0.70)	
Unfiltered (32 tables)	aligned readout, B	0.529	+.017 [.009,.025]	0.328	+.013 [.005,.021]
	aligned readout, $2B$	0.538	+.008 [.000,.016]	0.338	+.003 [-.006,.011]
	cardinal search, B	0.524	+.022 [.015,.031]	0.326	+.015 [.006,.023]
	re-search gate	0.535	+.011 [.005,.017]	0.335	+.006 [.001,.012]
	gated readout (ours)	0.546 (fires 0.71)		0.341 (fires 0.70)	

D.1 THE PLAYER COUNT: THE SAME GAME AT TWO, THREE AND FOUR PLAYERS

The main text attributes the cost of the aligned value to the player count. This subsection measures that on Tally with the player count N as the only variable. For $N \in \{2, 3, 4\}$ we draw 48 tables by the filter of Appendix B at that N (the first 48 seeds, in the fixed order, whose win-optimal and score-optimal first moves differ; token values are drawn per token and player, so the tables differ across N), train the same networks by the recipe of Appendix C (the two dual-head networks and the win-trained opponent, three seeds; at N=2 only the win network, since the rank credit $[1, 0]$ is the win credit), recalibrate $\hat{\Delta}$ at that N by the protocol of Appendix C ($\hat{\Delta}_{\mathrm{win}} = 0.128$, 0.066 and 0.088 for $N = 2, 3, 4$; $\hat{\Delta}_{\mathrm{rank}} = 0.090$ and 0.213 for $N = 3, 4$), draw up to twelve decision-relevant probe states per table and objective (every three- and four-player table has at least one; eight of the two-player tables have none, so the two-player probe rows cover 40 tables), and play the tournament of this appendix with the test seat rotated over all N seats on a fresh draw of twelve two-ply openings per table, one game per opening, seat and seed. At N=4 the rank objective is the linear rank credit $[1, \frac{2}{3}, \frac{1}{3}, 0]$, of which the main text's $[1, \frac{1}{2}, 0]$ is the three-player case. The arms are the aligned readout at B and $2B$, the cardinal search at B and $2B$, and the gated and the always-cardinal readouts at B, with c=2 throughout. Run through the same pipeline, N=3 reproduces the results of the main text: the 48 tables are those of the main text, the recalibrated error scales equal those of Appendix C to six decimals, the 3294 probe rows behind Figure 2 are reproduced exactly, and the tournament on the fresh openings gives $+0.017$ (interval $+0.011$ to $+0.023$) and $+0.013$ ($+0.007$ to $+0.020$) for the gated readout against the aligned readout at B, against $+0.012$ and $+0.017$ on the openings of Table 2.

Table 6 gives the result. At two players the gated readout's difference from the aligned readout at the same budget is not resolved ($+0.004$, interval -0.000 to $+0.008$; on the probe states -0.001, -0.011 to $+0.009$), and it loses to the aligned readout at twice the budget (-0.005, -0.009 to -0.001): it matches the aligned readout at 86 simulations (interval 64 to 117), where at three players the matching budget is above 128 simulations on both networks and, in the sweep of Appendix E, about 440 and more than 1024. The gate fires on more moves at two players than at three (0.85 against 0.70), because the recalibrated error scale is larger there ($\hat{\Delta}_{\mathrm{win}} = 0.128$ against 0.066), and the firing changes little because on the fired states the cardinal readout is not the better decision. At four players each of the gated readout's three contrasts with the aligned readout has a larger point estimate than at three ($+0.020$ and $+0.021$ at the same budget, $+0.012$ and $+0.017$ against twice the budget, -0.039 and -0.053 on the probe states), with intervals that overlap those of the three-player run except on the rank network's probe states (-0.052 to -0.027 against -0.026 to -0.007), and the comparison across N is unpaired because the tables differ; its margin over the always-cardinal readout is smaller at four, and zero on the rank network, where the gate fires on 93 percent of moves. What changes with the player count is not the coarseness of the aligned value: the share of exactly tied action pairs on the probe states of the win network is 0.53, 0.55 and 0.54 at $N = 2, 3, 4$ (on the rank network 0.49 at three and 0.42 at four). It is the reliability of the aligned readout, whose exact regret on the win network rises from 0.047 to 0.161 to 0.203, and with it the region in which the

cardinal readout is the better decision: in the margin-band profile of Appendix F recomputed at each N on the win network, the cardinal readout is resolvably better on no band at two players, below one $\hat{\Delta}_A$ at three and below four $\hat{\Delta}_A$ at four, and resolvably worse above four $\hat{\Delta}_A$ at every N. This is the direction Proposition 2 predicts, since at two players tied children carry identical vectors and at three or more the backed-up value at a tie is under-determined; the sweep is consistent with that mechanism but does not isolate it from what else changes with N: the number of opponents, the variance of the win outcome, the per-player depth and the tables themselves.

The misalignment of the cardinal value, by contrast, is present at every player count: on the win network the exact score-optimal action has a regret of 0.199, 0.229 and 0.184 at $N = 2, 3, 4$, and relative to the aligned readout, which is most accurate at two players, it is largest there, a worse decision by 0.152 of win probability (0.088 to 0.225), where the cardinal search also loses 0.032 credit to the aligned readout (0.019 to 0.045); at three players the corresponding numbers are 0.069 and a cardinal-search difference that is not resolved (-0.003, -0.011 to $+0.005$). The two failures of §3 therefore separate (Proposition 1 at every N, Proposition 2 at $N\geq3$): the cardinal value is misaligned at any N, and the aligned readout becomes unreliable at $N\geq3$, which is where a readout that uses the cardinal value where the aligned readout is unreliable can improve on it. At four players under the rank objective the gate fires on 93 percent of moves, the gated and the always-cardinal readouts coincide, and the cardinal search itself beats the aligned readout by 0.004 (0.001 to 0.007), the regime of Chinese Checkers in §5.3; the four-player rank credit is also finer than the three-player one, which is the mechanism of §3 working in the other direction, and that row should be read with this in mind. The limits of the comparison are one game and one network size; a per-player depth that changes with N (four, three or two, and two decisions per player), in the direction opposite to the effect; 40 of the 48 two-player tables having decision-relevant probe states (the tournament uses all 48); and an equivalent budget read from the B and $2B$ points only. The script, the networks for N=2 and N=4 and the result files are in the repository (Appendix I).

Table 6: Tally at two, three and four players (training objective; unit = table; n=48 tables for the full games and for the three- and four-player probe states, 40 for the two-player probe states; 95% bootstrap intervals). Probe states: exact regret of the aligned readout and the paired differences of the gated readout and of the exact score-optimal action from it. Full games: the gated readout's firing rate and its paired differences from the aligned readout at B and $2B$, the cardinal search's difference from the aligned readout, and the aligned budget whose credit matches the gated readout at B (from the B and $2B$ points; ">128" where the gated readout exceeds the aligned readout at $2B$).

N	Network	Probe states (exact regret)			Full games (credit, B=64)				Aligned budget
		aligned	gated − aligned	exact score-optimal − aligned	fire	gated − aligned, B	gated − aligned, $2B$	cardinal search − aligned	matching gated at B
2	win	0.047	-.001 [-.011,.009]	+.152 [.088,.225]	0.85	+.004 [.000,.008]	-.005 [-.009,-.001]	-.032 [-.045,-.019]	86 [64, 117]
3	rank	0.107	-.016 [-.026,-.007]	+.011 [-.010,.033]	0.71	+.017 [.011,.023]	+.007 [.000,.014]	-.006 [-.011,-.001]	>128
3	win	0.161	-.028 [-.041,-.015]	+.069 [.016,.129]	0.70	+.013 [.007,.020]	+.008 [.002,.013]	-.003 [-.011,.005]	>128
4	rank	0.122	-.039 [-.052,-.027]	-.034 [-.055,-.011]	0.93	+.020 [.016,.025]	+.012 [.008,.018]	+.004 [.001,.007]	>128
4	win	0.203	-.053 [-.072,-.035]	-.019 [-.054,.018]	0.79	+.021 [.015,.027]	+.017 [.011,.023]	+.003 [-.002,.009]	>128

D.2 Training level of the networks and strength of the opponent

The gain comes from the structured errors of two trained heads (§5.4). This subsection asks whether it persists when the test networks are trained longer, so that their aligned readout improves, and against opponents that are trained longer or search more. Two variables change and nothing else: the self-play games of both dual-head networks, $T \in \{100, 300, 1000, 3000, 10{,}000\}$, and the opponent, a win-trained network with 300, 3000 or 10,000 self-play games searching at 64 or 256 simulations per move, in a full factorial of 30 cells per game. Every network is trained once to 10,000 games by the recipe of Appendix C and saved at each level; the gated readout keeps c=2 and the calibrated error scale of Appendix C, which is measured without a network; the tables, openings, seats, repeats and random seeds are those of this appendix; and the arms are the aligned readout at B and $2B$, the cardinal search at B, and the gated and the always-cardinal readouts at B. The experiments of the main text are the cell of 300-game test networks against the 300-game opponent at 64 simulations, and the sweep reproduces them exactly: every 300-game network is bit-identical to the network of the main text (432 on Tally, 9 on Chinese Checkers), that cell reproduces the games of the five arms in Table 1 one by one (51,840 Tally and 8640 Chinese Checkers games), and the probe rows at 300 games reproduce those behind Figure 2 (3294 on Tally, 96 on Chinese Checkers, where the

win network is scored on 32 further endgame states solved under the win view). The design and the rules for reading the results were committed before the first game.

On Chinese Checkers, longer training makes the aligned readout more accurate, as the question supposes (Figure 5): against the opponent of the main text the aligned readout's credit rises from 0.625 to 0.836 on the rank network and from 0.373 to 0.662 on the win network between 300 and 10,000 games, and its exact regret on the probe states falls from 0.206 to 0.107 and from 0.281 to 0.177. On Tally it does so up to 3000 games, where the probe-state regret falls from 0.107 to 0.085 and from 0.161 to 0.131, and not beyond (0.092 and 0.139 at 10,000 games); at that saturated level the policy head keeps improving while the decisions of both value heads, read without search, worsen. The cardinal readout of the same tree has its lowest probe-state regret at 3000 games on Tally and at 1000 on Chinese Checkers, and at 10,000 games its regret is above its 300-game value on all four networks (Table 9). The opponent axis changes the opponent but does not always order it by strength: at 300 games, a longer-trained opponent lowers the aligned readout's credit on Tally by at most 0.015 while 256 simulations change it by less than 0.005, and on Chinese Checkers a longer-trained opponent is weaker, the aligned readout's credit against it rising from 0.625 to 0.806, while 256 simulations make the opponent slightly stronger in five of the six comparisons (0.598 against 0.625 for the 300-game opponent).

Table 7 and Figure 4 give the gain, the gated readout minus the aligned readout at the same 64 simulations. Across the six opponents, Tally keeps the gain above zero in all twelve cells at 300 and at 1000 games, in five at 3000 and in none at 10,000, where four rank-network cells are below zero (-0.008 to -0.011); Chinese Checkers keeps it above zero in all 60 cells, and in 59 of them the gated readout also beats the aligned readout at twice the budget. Paired by table or opening, the gain at 10,000 games is below its value at 300 games in all 24 opponent cells of the two games, by 0.011 to 0.026 on Tally and by 0.15 to 0.33 on Chinese Checkers, where against the opponent of the main text it falls to 24 and 15 percent of its 300-game value. Against every opponent, the gain at the main text's training level is above zero on both games (24 of 24 cells), including, for each network, the opponent that lowers its aligned readout's credit most ($+0.016$ and $+0.014$ on Tally, $+0.276$ and $+0.394$ on Chinese Checkers). On Tally the opponent barely changes its size: 44 of the 50 paired comparisons with the opponent of the main text are not resolved. On Chinese Checkers the size varies with the opponent while the sign does not; against the weaker 10,000-game opponent at 64 simulations the rank network's gain is $+0.175$ instead of $+0.263$, as the aligned readout has a credit of 0.806 there.

Table 9 separates what changes, and Table 8 shows its 300- and 10,000-game rows with the two networks side by side. On the probe states, Proposition 4 writes the gated readout's regret minus the aligned readout's as the share of states where the gate switches times the regret difference on those states, cardinal minus aligned, since the two readouts coincide elsewhere. The share stays between 0.15 and 0.28 on Tally at every level and falls from 0.55 and 0.53 at 300 games to 0.42 and 0.39 at 10,000 on Chinese Checkers. The difference goes from -0.066 and -0.159 on Tally and -0.142 and -0.186 on Chinese Checkers at 300 games to $+0.025$, $+0.035$, $+0.081$ and $+0.032$ at 10,000. The gate keeps switching, and the cardinal readout, the better decision on the switched states at 300 games, has lost its advantage there by 10,000. In games the gate fires on 70 percent of Tally moves at 300 games and on 79 and 78 percent at 10,000, and on 98 to 100 percent of Chinese Checkers moves at every level. The gain splits exactly into the always-cardinal readout's difference from the aligned readout, the credit difference due to playing the cardinal ordering on every move, and the gated readout's difference from the always-cardinal readout, the credit difference due to the selection. From 300 to 10,000 games the first falls from $+0.001$ and $+0.010$ to -0.020 and -0.015 on Tally and from $+0.264$ and $+0.367$ to $+0.063$ and $+0.049$ on Chinese Checkers. On Tally the second stays between $+0.007$ and $+0.019$ at every level, with every interval above zero. On Chinese Checkers the always-cardinal readout stays within 0.008 of the gated readout at every level and against every opponent. On Tally, table by table, the change in the gain from 300 to 10,000 games tracks the change in the always-cardinal readout's difference from the aligned readout (correlation $+0.74$ to $+0.85$ for the opponent of the main text and for the 10,000-game opponent at 256 simulations). The gate cannot register this change. Its threshold $c\,\hat{\Delta}_A$ comes from the calibration of Appendix C, which uses no trained network, and it reads the aligned margin only. On Tally at 10,000 games it therefore keeps switching to a cardinal readout that no longer improves the decision, and it still loses less than the always-cardinal readout, which falls 0.010 to 0.028 below the aligned readout, with 11 of the 12 intervals below zero. Read without search, the cardinal head shows the same pattern: on

the Chinese Checkers probe states its own decisions are best near 1000 games (regret 0.065 and 0.134) and worse at 10,000 (0.115 and 0.188).

Proposition 4 is an identity on any population of states, and the probe states and the tournament are different populations. On Tally the probe states come from random playouts over the whole game, and at every training level the two measures favor the same readout: the gated readout has the lower probe-state regret exactly where its gain in games is positive. On Chinese Checkers the probe states are decision-relevant endgame states from random playouts, scored by exact regret, which assumes exact play after the decision. The tournament plays whole games from four-ply openings, with the test search and the two copies of the win-trained opponent choosing every later move. There the two disagree at 3000 games on the rank network and at 10,000 games on both networks. At 10,000 games the products of Table 9 are $0.42 \times 0.081 = +0.034$ and $0.39 \times 0.032 = +0.012$ on the probe states, with intervals -0.016 to $+0.094$ and -0.035 to $+0.059$ over the 32 states. The gain in games at the same level is $+0.063$ and $+0.055$, with intervals above zero.

The limits of the sweep are that its opponent axis does not order opponents by strength on Chinese Checkers, that the 10,000-game level of Tally is saturated rather than more accurate, and that part of the decline on Chinese Checkers is a ceiling (the rank network's aligned readout reaches a credit of 0.98 against the 10,000-game opponent). The Chinese Checkers probe states also cover only the exactly solvable endgame band, and the training recipe, replay buffer and learning rate included, is that of Appendix C at every level. The script and the result files are in the repository (Appendix I); the networks of the sweep are retrained by the script, whose 300-game level is checked against the committed networks.

Table 7: Gain of the gated over the aligned readout at B=64, the paired difference of the first row of Table 2, by the test networks' self-play games against the opponent of the main text (95% bootstrap interval over tables or openings); in parentheses, the number of the six opponents (300, 3000 or 10,000 self-play games at 64 or 256 simulations per move) against which the interval lies above zero and, if any, below zero.

	Tally		CC	
Self-play games	rank network	win network	rank network	win network
100	+.005 [-.001,.010] (4/6)	+.009 [.002,.016] (5/6)	+.242 [.223,.262] (6/6)	+.253 [.218,.290] (6/6)
300	+.012 [.006,.018] (6/6)	+.017 [.010,.024] (6/6)	+.263 [.241,.285] (6/6)	+.363 [.325,.400] (6/6)
1,000	+.013 [.007,.019] (6/6)	+.018 [.012,.024] (6/6)	+.227 [.205,.249] (6/6)	+.393 [.369,.417] (6/6)
3,000	+.005 [-.001,.011] (2/6)	+.002 [-.006,.010] (3/6)	+.052 [.027,.076] (6/6)	+.207 [.182,.231] (6/6)
10,000	-.009 [-.016,-.002] (0/6, 4 below)	-.001 [-.009,.008] (0/6)	+.063 [.044,.082] (6/6)	+.055 [.030,.080] (6/6)

Table 8: The gain declines with training because the cardinal readout loses its advantage on the states where the gate switches, while the gate keeps switching. Probe states, at 300 and 10,000 self-play games of the test networks against the win-trained opponent of Table 1 (CC is Chinese Checkers): exact regret of the aligned (R_A) and the cardinal (R_C) readout of one 64-simulation dual-backup search, and $R_C - R_A$ on the state-seed pairs where the gate switches, their share in parentheses; the product of the two is the gated minus aligned regret (Proposition 4), pooled over state-seed pairs (Tally 48 tables, CC 32 endgame states, three seeds). Full games: the gate's firing rate per move and the paired differences of credit always-cardinal minus aligned (AC−A) and gated minus aligned, the gain; bold marks 95% bootstrap intervals above or below zero. Table 9 gives every training level.

	rank network						win network					
Game, self-play games	R_A	R_C	$R_C - R_A$ (share)	fire	AC−A	gain	R_A	R_C	$R_C - R_A$ (share)	fire	AC−A	gain
Tally, 300	0.107	0.101	-0.066 (0.25)	0.70	+0.001	**+0.012**	0.161	0.182	-0.159 (0.17)	0.70	**+0.010**	**+0.017**
Tally, 10,000	0.092	0.116	+0.025 (0.27)	0.79	**-0.020**	**-0.009**	0.139	0.198	+0.035 (0.18)	0.78	**-0.015**	-0.001
CC, 300	0.206	0.135	-0.142 (0.55)	0.99	**+0.264**	**+0.263**	0.281	0.198	-0.186 (0.53)	0.99	**+0.367**	**+0.363**
CC, 10,000	0.107	0.141	+0.081 (0.42)	1.00	**+0.063**	**+0.063**	0.177	0.214	+0.032 (0.39)	0.99	**+0.049**	**+0.055**

E BUDGET CURVES IN NUMBERS, AND THE FULL-GAME TABLES

Tables 10 and 11 list the tournament credits of the aligned readout, the cardinal search and the gated readout behind Figure 3; the re-search gate's credit and measured cost at each budget are in

Table 9: What changes with the networks' self-play games (CC is Chinese Checkers). Probe states: exact regret of the aligned and of the cardinal readout (means over tables or states, as in Figure 5); the share of state-seed pairs on which the gate switches; the cardinal minus aligned regret on those pairs; and their product, the gated minus aligned regret over all pairs (Proposition 4). The last three are pooled over state-seed pairs (Tally 48 tables, CC 32 endgame states, three seeds). Full games against the opponent of the main text: the gated readout's firing rate per move and the paired differences always-cardinal minus aligned, gated minus always-cardinal and gated minus aligned, the last the sum of the two before it and the gain of Table 7; bold marks the differences whose 95% bootstrap interval lies above or below zero. Table 8 shows the 300- and 10,000-game rows with the two networks side by side.

	Probe states, exact regret					Full games, credit			
Self-play games	aligned	cardinal	switched share	cardinal − aligned on switched	product	firing rate	always-cardinal − aligned	gated − always-cardinal	gated − aligned
Tally, rank network									
100	0.121	0.124	0.28	-0.051	-0.014	0.68	**-0.014**	**+0.019**	+0.005
300	0.107	0.101	0.25	-0.066	-0.017	0.70	+0.001	**+0.011**	**+0.012**
1,000	0.099	0.106	0.23	-0.013	-0.003	0.73	+0.002	**+0.011**	**+0.013**
3,000	0.085	0.094	0.23	-0.035	-0.008	0.74	-0.005	**+0.010**	+0.005
10,000	0.092	0.116	0.27	+0.025	+0.007	0.79	**-0.020**	**+0.011**	**-0.009**
Tally, win network									
100	0.172	0.227	0.18	-0.102	-0.019	0.69	-0.004	**+0.013**	**+0.009**
300	0.161	0.182	0.17	-0.159	-0.028	0.70	**+0.010**	**+0.007**	**+0.017**
1,000	0.145	0.184	0.15	-0.112	-0.017	0.72	**+0.009**	**+0.009**	**+0.018**
3,000	0.131	0.174	0.16	-0.100	-0.016	0.73	-0.009	**+0.011**	+0.002
10,000	0.139	0.198	0.18	+0.035	+0.006	0.78	**-0.015**	**+0.014**	-0.001
CC, rank network									
100	0.206	0.195	0.66	-0.016	-0.010	0.99	**+0.241**	+0.001	**+0.242**
300	0.206	0.135	0.55	-0.142	-0.078	0.99	**+0.264**	-0.001	**+0.263**
1,000	0.164	0.104	0.55	-0.108	-0.060	1.00	**+0.227**	+0.001	**+0.227**
3,000	0.120	0.146	0.46	+0.023	+0.010	1.00	**+0.052**	+0.000	**+0.052**
10,000	0.107	0.141	0.42	+0.081	+0.034	1.00	**+0.063**	+0.000	**+0.063**
CC, win network									
100	0.286	0.222	0.69	-0.109	-0.075	0.98	**+0.253**	+0.000	**+0.253**
300	0.281	0.198	0.53	-0.186	-0.099	0.99	**+0.367**	-0.004	**+0.363**
1,000	0.260	0.172	0.55	-0.151	-0.083	0.99	**+0.392**	+0.001	**+0.393**
3,000	0.248	0.231	0.41	-0.081	-0.033	0.99	**+0.205**	+0.002	**+0.207**
10,000	0.177	0.214	0.39	+0.032	+0.012	0.99	**+0.049**	**+0.006**	**+0.055**

the result files (Appendix I). Table 12 repeats Table 1 with full arm names. On Tally the gated readout minus the aligned readout at the same budget is $+0.018$, $+0.020$, $+0.017$, $+0.012$, $+0.009$, $+0.010$ (rank network) and $+0.016$, $+0.014$, $+0.013$, $+0.017$, $+0.012$, $+0.011$ (win network) for $B = 8, 16, 32, 64, 128, 256$, every interval above zero; against the aligned readout at twice the budget the differences are $+0.009$, $+0.011$, $+0.004$, $+0.007$, $+0.005$, $+0.006$ and $+0.002$, $+0.005$, $+0.009$, $+0.013$, $+0.008$, $+0.007$, with intervals above zero at $B \in \{8, 16, 64, 256\}$ and $\{32, 64, 128\}$ respectively. On the Tally probe states (Figure 6) the gated readout minus the aligned readout at the same budget is -0.029, -0.029, -0.019, -0.016, -0.012, -0.004 (rank network) and -0.031, -0.032, -0.032, -0.028, -0.018, -0.012 (win network), resolved up to B=128.

F The Margin-Band Profile

Table 14 gives the numbers behind Figure 2. For Tally the bands are in units of $\hat{\Delta}_A$; the gate fires on every pair in the first three bands and on none in the last two. For Chinese Checkers the rows are quartiles of the aligned margin over the 95 state-seed pairs where it is defined; the gate fires on every pair of the first three quartiles and on 18 of the 24 of the fourth.

The implementation gap of Proposition 3. The proposition certifies the readout that plays the maximizer of $\hat{q}_{\mathrm{A}}$; the deployed readout plays the most-visited child. On the Tally probe states the two pick the same action on 74 and 75 percent of state-seed pairs, and the $\hat{q}_{\mathrm{A}}$-maximizer's regret differs from the most-visited child's by -0.007 (interval -0.019 to $+0.004$) on the rank network

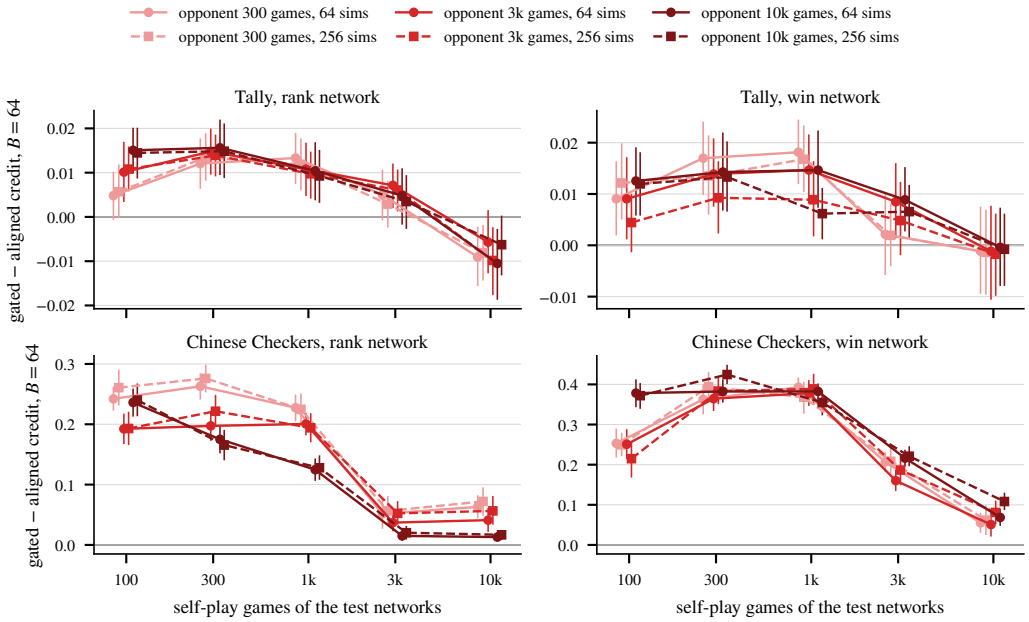


Figure 4: The gain of the gated readout over the aligned readout at 64 simulations against the test networks' self-play games, one line per opponent: color for the opponent's self-play games, solid for 64 and dashed for 256 simulations; 95% bootstrap intervals over tables or openings.

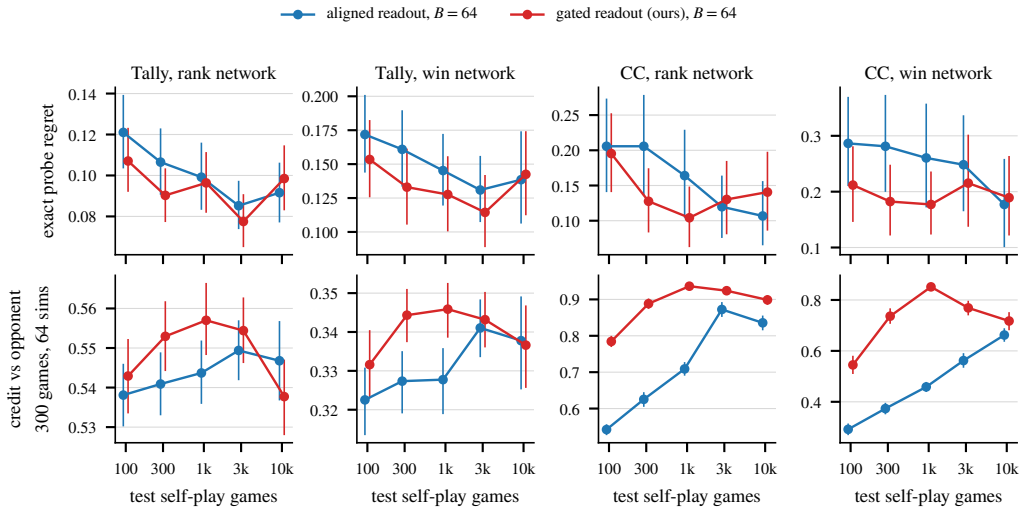


Figure 5: The aligned and the gated readout against the test networks' self-play games: exact regret on the probe states (top; Tally 48 tables, CC 32 endgame states per network) and credit against the opponent of the main text (bottom), with 95% bootstrap intervals.

and -0.006 (-0.021 to $+0.009$) on the win network. This is a measurement, not a certification: the certified and the deployed readouts are close on these states, and nothing in Proposition 3 says they must be.

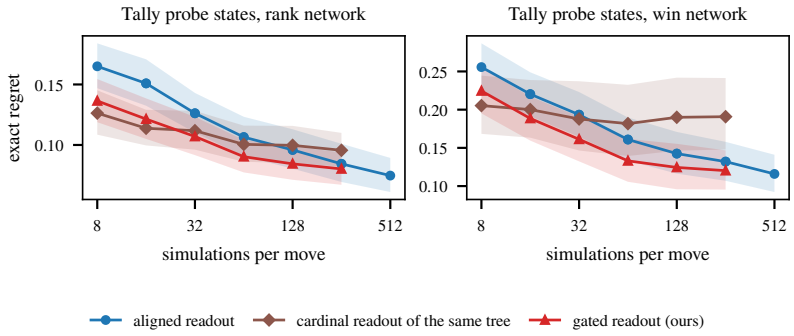


Figure 6: Exact regret versus simulation budget on the Tally probe states (48 tables, three seeds; 95% bootstrap intervals over tables) for the aligned readout, the cardinal readout of the same tree and the gated readout, the gate keeping its 64-simulation calibration.

Table 10: Tournament credit versus simulations per move on Tally (48 tables; training objective). The last row is the aligned readout's budget equivalent to the gated readout at 64 simulations.

	B	Tally, rank network			Tally, win network		
		aligned	cardinal search	gated	aligned	cardinal search	gated
	8	0.509	0.511	0.527	0.300	0.301	0.317
	16	0.519	0.518	0.538	0.314	0.310	0.329
	32	0.528	0.527	0.545	0.324	0.317	0.337
	64	0.541	0.533	0.553	0.327	0.324	0.344
	128	0.546	0.535	0.555	0.331	0.330	0.343
	256	0.550	0.541	0.560	0.335	0.334	0.346
	512	0.554	–	–	0.339	–	–
	1024	0.555	–	–	0.344	–	–
aligned budget matching the gated readout at 64		439 [156, > 1024]			> 1024 [323, > 1024]		

G Backup Variants and the Injected-Noise Model

The learned selector. The selector replaces the gate by a logistic function of two standardized features read from the same root: the cardinal gap $\hat{q}_{\mathrm{C}}(a_{\mathrm{C}}) - \hat{q}_{\mathrm{C}}(a_{\mathrm{A}})$ divided by the cardinal error scale $\hat{\Delta}_C$, and the aligned gap $\hat{q}_{\mathrm{A}}(a_{\mathrm{A}}) - \hat{q}_{\mathrm{A}}(a_{\mathrm{C}})$ divided by $\hat{\Delta}_A$. It is consulted only when $a_{\mathrm{A}} \neq a_{\mathrm{C}}$ and switches to a_{C} when its output exceeds one half. Its coefficients were fitted, once per training objective, by weighted logistic regression on the probe states of 32 Tally tables, the weight of a state-seed pair being the absolute exact benefit of the switch, and were then used unchanged on all 48 tables and on Chinese Checkers.

Backup variants on the probe states. Table 15 scores the gated readout under four backup rules on the same states and seeds. The rule of Algorithm 1 backs up each vector by its own max^{n} choice. *Carried* takes, at every node, the cardinal vector of the child the aligned rule selected. *Lexicographic, exact ties* selects, among children tied in the aligned mover component, the one with the largest cardinal component and backs up both of its vectors. *Lexicographic, band* widens the tie to $c\,\hat{\Delta}_A$. The lexicographic interiors also change the aligned readout itself: its regret rises by $+0.002$ and $+0.006$ (rank network) and $+0.010$ and $+0.014$ (win network) for the exact-tie and banded rules.

G.1 A control: isotropic noise on exact values does not produce the gain

A gate of this form could benefit from near-ties by breaking them with a second signal. On exact values it does not. We took the exact aligned and cardinal action values of every probe state, on Tally and on 400 random three-player $3\times3\times3$ matrix games, added independent Gaussian noise of scale σ_A to the aligned values and σ_C to the cardinal values, and applied the gate with $\hat{\Delta}_A$ set to the noise's mean absolute deviation, $\sigma_A\sqrt{2/\pi}$. At every point of a grid over $\sigma_A \in \{0.01, 0.03, 0.1, 0.3\}$,

Table 11: Tournament credit versus simulations per move on Chinese Checkers (32 openings; training objective).

	B	CC, rank network			CC, win network		
		aligned	cardinal search	gated	aligned	cardinal search	gated
	8	0.578	0.672	0.841	0.317	0.394	0.614
	16	0.599	0.709	0.870	0.314	0.433	0.688
	32	0.602	0.730	0.893	0.341	0.494	0.732
	64	0.625	0.785	0.888	0.373	0.557	0.736
	128	0.651	0.831	0.893	0.388	0.619	0.792
	256	0.691	0.872	0.887	0.414	0.686	0.789
	512	0.721	–	–	0.449	–	–
	1024	0.732	–	–	0.460	–	–
aligned budget matching the gated readout at 64		>1024 [>1024, >1024]			>1024 [>1024, >1024]		

Table 12: Table 1 with full arm names; bold as in Table 1.

		Tally		CC	
Arm	sims/move	rank network	win network	rank network	win network
Aligned readout, B=64	64	0.541 [0.533, 0.549]	0.327 [0.319, 0.335]	0.625 [0.605, 0.645]	0.373 [0.351, 0.396]
Aligned readout, $2B$=128	128	0.546 [0.538, 0.553]	0.331 [0.324, 0.339]	0.651 [0.631, 0.670]	0.388 [0.367, 0.408]
Cardinal search, B=64	64	0.533 [0.525, 0.541]	0.324 [0.314, 0.332]	0.785 [0.770, 0.799]	0.557 [0.525, 0.588]
Cardinal search, $2B$=128	128	0.535 [0.527, 0.544]	0.330 [0.321, 0.339]	0.831 [0.814, 0.848]	0.619 [0.588, 0.648]
Re-search gate (aligned B, then cardinal B on fire)	109–128	0.548 [0.540, 0.557]	0.338 [0.330, 0.344]	0.785 [0.770, 0.799]	0.551 [0.520, 0.581]
Gated readout (ours), B=64	64	**0.553** [0.544, 0.562]	**0.344** [0.337, 0.351]	**0.888** [0.874, 0.902]	**0.736** [0.706, 0.767]
Always-cardinal readout	64	0.542 [0.533, 0.550]	0.337 [0.329, 0.345]	0.889 [0.876, 0.903]	0.740 [0.712, 0.769]
Learned selector	64	0.560 [0.551, 0.568]	0.352 [0.344, 0.359]	0.887 [0.874, 0.899]	0.739 [0.712, 0.766]

σ_C from 0 to the cardinal spread, and $c \in \{1, 2, 4, \infty\}$, the gated readout is worse than the noisy aligned readout. Table 16 shows the c=2, σ_C=0 slice. Among a mover's action pairs, 33 percent (matrix, win), 31 percent (matrix, rank), 55 percent (Tally, win) and 49 percent (Tally, rank) are exact aligned ties, and every other pair differs by at least $\frac{1}{2}$; within any band below $\frac{1}{2}$ no non-tied pair exists. The noisy aligned readout is exact to within a regret of 0.0002 up to σ_A=0.1, against 0.107 for the trained head; the gate fires only at exact ties, and the switch lowers the aligned value whenever the cardinal readout lies outside the tied set, which happens on 12 to 25 percent of fired draws at small noise. The gain of the trained system is therefore not a near-tie effect; it comes from the heads' structured errors, which the margin identifies and on which the cardinal readout is more often right.

H Other Objectives

Every terminal is scored under the three objective views. Table 17 gives the gated readout's paired difference from the aligned readout at 64 simulations under each; the main text reports each network under its training objective.

I Reproducibility

Every experiment is a deterministic function of recorded seeds: token tables, openings, self-play, search draws and tournament repeats are all seeded by position in fixed orders, and every arm reuses the same networks, openings and random seeds. Searches are pure-Python vector-max^n implementations validated by a 406-test suite that checks exact-solver agreement, byte-identical defaults for every added option (the dual-backup search reproduces the visit counts and root values of the plain aligned search under the same seed), the reductions of the gate to the aligned readout at c=0 and to the always-cardinal readout at c=∞, and the backup variants on synthetic nodes. Every table and figure in this paper is generated by a committed script directly from the result files: the per-configuration tournament files and their summaries, the probe-state files, the budget sweeps, the synthetic-noise grid, the player-count sweep of Appendix D.1 and the training and opponent sweep of Appendix D.2. The anonymized repository is at `https://anonymous.4open.science/r/Test-Time-Search-Efficiency-in-Three-Player-General-Sum-Games-DC77/`; its README maps each section, table and

Table 13: Every paired contrast, gated readout minus arm, including the blends and constants omitted from Table 2; bold as in Table 2.

	Tally		CC	
Gated readout minus arm	rank network	win network	rank network	win network
Baselines				
Aligned readout at B	**+0.012** [+0.006, +0.018]	**+0.017** [+0.010, +0.024]	**+0.263** [+0.241, +0.285]	**+0.363** [+0.325, +0.400]
Aligned readout at $2B$	**+0.007** [+0.001, +0.014]	**+0.013** [+0.005, +0.021]	**+0.238** [+0.216, +0.259]	**+0.349** [+0.315, +0.382]
Cardinal search at B	**+0.020** [+0.014, +0.026]	**+0.021** [+0.012, +0.030]	**+0.104** [+0.087, +0.121]	**+0.179** [+0.152, +0.206]
Cardinal search at $2B$	**+0.018** [+0.012, +0.024]	**+0.014** [+0.006, +0.022]	**+0.057** [+0.038, +0.079]	**+0.117** [+0.083, +0.152]
Re-search gate (cost $B(1+f)$)	**+0.004** [+0.000, +0.009]	**+0.007** [+0.002, +0.012]	**+0.104** [+0.087, +0.121]	**+0.185** [+0.158, +0.213]
Selection				
Always-cardinal readout	+0.011 [+0.007, +0.016]	+0.007 [+0.001, +0.013]	-0.001 [-0.003, -0.000]	-0.004 [-0.008, +0.000]
Learned selector	-0.007 [-0.010, -0.003]	-0.007 [-0.011, -0.004]	+0.002 [-0.006, +0.009]	-0.003 [-0.023, +0.017]
Backup variants				
Backup: cardinal carried along the aligned choice	+0.006 [+0.002, +0.009]	+0.008 [+0.003, +0.013]	+0.011 [-0.007, +0.028]	-0.003 [-0.036, +0.032]
Backup: lexicographic interior (exact ties)	+0.004 [-0.000, +0.008]	+0.007 [+0.000, +0.013]	+0.011 [-0.007, +0.028]	-0.003 [-0.036, +0.032]
Backup: lexicographic interior (band $c\hat{\Delta}$)	+0.004 [-0.001, +0.010]	+0.008 [+0.001, +0.015]	-0.004 [-0.020, +0.014]	+0.007 [-0.029, +0.043]
Fixed blends				
Fixed blend λ=0.25	+0.015 [+0.009, +0.021]	+0.014 [+0.006, +0.023]	+0.102 [+0.085, +0.117]	+0.189 [+0.156, +0.221]
Fixed blend λ=0.5	+0.010 [+0.004, +0.015]	+0.006 [-0.001, +0.013]	+0.107 [+0.088, +0.125]	+0.205 [+0.179, +0.232]
Fixed blend λ=0.75	+0.007 [+0.001, +0.012]	+0.005 [-0.001, +0.012]	+0.131 [+0.115, +0.147]	+0.218 [+0.183, +0.253]
Robustness				
Gate constant c=1	+0.004 [+0.001, +0.006]	+0.004 [+0.001, +0.007]	+0.001 [-0.001, +0.004]	+0.009 [+0.002, +0.017]
Gate constant c=4	+0.002 [-0.000, +0.004]	-0.004 [-0.007, -0.000]	-0.001 [-0.003, -0.000]	-0.004 [-0.008, +0.000]
Error scale $\hat{\Delta}$ of the other game	+0.010 [+0.006, +0.014]	-0.002 [-0.006, +0.002]	+0.002 [-0.003, +0.007]	+0.016 [+0.003, +0.029]

Table 14: Exact regret of the aligned readout and of the cardinal readout of the same tree, and the shares of state-seed pairs on which each is the strictly better decision, by band of the aligned margin (Tally: 48 tables, three seeds; Chinese Checkers: 32 exactly solved endgame states, three seeds, rank network).

Population	$\hat{m}_A/\hat{\Delta}_A$ band	n	aligned regret	cardinal regret	P(cardinal better)	P(aligned better)
Tally, rank network	$= 0$	68	0.059	0.132	0.06	0.19
Tally, rank network	$(0, 1]$	525	0.190	0.136	0.16	0.07
Tally, rank network	$(1, 2]$	295	0.120	0.103	0.08	0.06
Tally, rank network	$(2, 4]$	334	0.075	0.087	0.05	0.07
Tally, rank network	> 4	407	0.047	0.077	0.04	0.10
Tally, win network	$= 0$	44	0.205	0.136	0.09	0.02
Tally, win network	$(0, 1]$	279	0.369	0.247	0.19	0.06
Tally, win network	$(1, 2]$	236	0.250	0.220	0.11	0.08
Tally, win network	$(2, 4]$	254	0.205	0.150	0.12	0.06
Tally, win network	> 4	739	0.057	0.139	0.02	0.11
CC, rank network	quartile 1 ($\hat{m}_A \leq 0.007$)	24	0.229	0.146	0.25	0.08
CC, rank network	quartile 2 ($\hat{m}_A \leq 0.035$)	24	0.219	0.188	0.17	0.12
CC, rank network	quartile 3 ($\hat{m}_A \leq 0.202$)	23	0.293	0.141	0.35	0.00
CC, rank network	quartile 4 ($\hat{m}_A > 0.202$)	24	0.083	0.062	0.17	0.04

figure of the paper to the script and the result files behind it and gives the commands that verify the implementation, regenerate the tables and figures, rerun the experiments from the shipped networks and retrain them.

Table 15: Backup variants on the Tally probe states (48 tables, three seeds): regret of the gated readout under each rule and its paired difference from the rule of Algorithm 1.

Backup	Rank network		Win network	
	gated readout regret	minus Algorithm 1	gated readout regret	minus Algorithm 1
Independent backups (Algorithm 1)	0.0903	–	0.1330	–
Cardinal carried along the aligned choice	0.0965	+0.0062 [+0.0000, +0.0125]	0.1469	+0.0138 [+0.0005, +0.0274]
Lexicographic interior, exact ties	0.0975	+0.0072 [+0.0005, +0.0139]	0.1566	+0.0236 [+0.0077, +0.0393]
Lexicographic interior, band $c\hat{\Delta}_A$	0.0911	+0.0009 [-0.0065, +0.0083]	0.1553	+0.0223 [+0.0075, +0.0377]

Table 16: The gate on exact values with injected noise (c=2, exact cardinal values). At every point of this slice it is worse than the noisy aligned readout.

Population	σ_A	aligned regret	gated readout regret	fire	P(aligned better | fire)
Matrix, win	0.01	0.0000	0.0134	0.05	0.25
Matrix, win	0.03	0.0000	0.0128	0.05	0.25
Matrix, win	0.1	0.0000	0.0129	0.05	0.25
Matrix, win	0.3	0.0161	0.1166	0.22	0.49
Matrix, rank	0.01	0.0000	0.0055	0.07	0.12
Matrix, rank	0.03	0.0000	0.0057	0.07	0.13
Matrix, rank	0.1	0.0002	0.0081	0.08	0.17
Matrix, rank	0.3	0.0695	0.1417	0.54	0.34
Tally, win	0.01	0.0000	0.0487	0.41	0.12
Tally, win	0.03	0.0000	0.0490	0.41	0.12
Tally, win	0.1	0.0000	0.0486	0.41	0.12
Tally, win	0.3	0.0138	0.0919	0.53	0.16
Tally, rank	0.01	0.0000	0.0278	0.43	0.12
Tally, rank	0.03	0.0000	0.0275	0.43	0.12
Tally, rank	0.1	0.0001	0.0299	0.44	0.12
Tally, rank	0.3	0.0624	0.0882	0.74	0.16

Table 17: Gated readout minus aligned readout at 64 simulations under each objective view.

Gated readout minus aligned readout at B	win $[1, 0, 0]$	rank $[1, \frac{1}{2}, 0]$	$[1, 1, 0]$
Tally, rank network	+0.011 [+0.003, +0.018]	+0.012 [+0.006, +0.018]	+0.013 [+0.007, +0.020]
Tally, win network	+0.017 [+0.010, +0.024]	+0.034 [+0.027, +0.042]	+0.051 [+0.042, +0.062]
CC, rank network	+0.368 [+0.339, +0.395]	+0.263 [+0.241, +0.285]	+0.158 [+0.132, +0.186]
CC, win network	+0.363 [+0.325, +0.400]	+0.359 [+0.328, +0.388]	+0.355 [+0.320, +0.388]