HUMAN DIFFICULTY IS NOT TASK DIFFICULTY: WHAT POLICY REPRESENTATIONS KNOW BEYOND AN OPTIMAL SOLVER

Anonymous authors
Paper under double-blind review

ABSTRACT

Human difficulty is not task difficulty. A policy network is trained to choose actions, not to judge how hard a problem is for people, yet part of it is linearly readable from its activations. Using chess as a model system, where difficulty is calibrated on large human populations, we probe eight networks. In Maia-2 a probe adds $\Delta R^2 = 0.0436$ over gradient-boosted engine-search controls plus the model's own policy entropy, and 0.0350 with its own move predictions also controlled. Deeper engine search, at the depths tested, predicts human difficulty worse. The signal is shared across seven models after adjusting for the target, controls and a raw board encoding. We distinguish two measured aspects of difficulty: the population-calibrated rating and, separately, how long the original player deliberated, beyond that rating, the clock and engine features (the two readouts correlate at $r = 0.59$). Under matched ablations, neither direction has a distinctive effect on the solution move (the pre-specified primary outcome); the thinking-time direction shifts the move distribution (pre-specified secondary outcomes, small in absolute terms) without favouring or disfavouring the solution relative to matched controls. A person's error may also depend on how the situation arose: given the game so far, a human-imitation policy raises the probability of the wrong move a player made (0.111–0.136 nats, natural-log units, on months before Maia-3-79M's training window); an engine-trained network does not.

1 INTRODUCTION

A system that finds the right action need not know which problems people find hard. Take two problems equally clear to a strong solver. One follows a familiar route, and the other invites a tempting mistake. The solver's confidence need not show which is harder for people. Even how often people fail misses part of the picture, because problems solved equally often may demand different amounts of thinking. So we ask whether networks trained to choose actions, none with a difficulty target, already encode information about human difficulty and thinking time, and whether they use it. We answer in one domain, chess, used as a model system (McIlroy-Young et al., 2020).

Chess supplies at scale four things that rarely coincide and that any such domain needs: item difficulty calibrated on a large human population and external to every model; a strong engine, the task-optimal solver of our title, as a control for the task's own complexity; recorded human decisions with their times; and a policy whose own uncertainty is measurable on the same items.

A policy network maps a problem to probabilities over actions, and its internal activations are its representation. We keep each network frozen and fit a linear probe (a weighted sum of activations) to predict a human measurement on held-out problems. The probe gets credit only for what it adds beyond controls: an engine's search features, the network's policy entropy (how spread its move probabilities are) and the player's rating. Without them, a probe could merely re-read the engine or the network's own output. We ask three questions. Can the probe predict how hard a problem is for people beyond these controls? Does the original player's thinking time add information beyond that difficulty, along a separate direction? And if we set each direction's component to its average and rerun the network (an ablation), do its choices change more than for matched random directions?

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.

We make three contributions, each with a limit. (i) Human difficulty is readable beyond these controls, even beyond Maia-2's probability of the correct move, so it is not the network's uncertainty renamed. It appears in a network trained only on engine targets, is shared across seven networks, and is predicted worse by deeper search at the depths tested, the first sense in which human difficulty is not task difficulty (Section 4). (ii) We distinguish two measured aspects of human difficulty, the population-calibrated rating and the original player's thinking time, in line with, though not a test of, the idea that people allocate thinking time by its expected benefit, as in resource-rational accounts (Griffiths et al., 2015; Russek et al., 2025). The representation predicts thinking time beyond the puzzle's rating, the clock and the player's own rating, along a different direction from difficulty; a solver run to a fixed depth never decides how long to look, a second sense (Section 6). (iii) Finding a signal does not show that the network uses it (Section 7). Under this ablation the difficulty direction has no distinctive effect on the solution move, also on a second architecture and after fine-tuning the policy's downstream layers on human solve rates in the tested configuration. The thinking-time direction measurably shifts the move distribution (pre-specified secondary outcomes, small in absolute terms), without favouring or disfavouring the solution relative to matched controls. A test at the input adds a third sense (Section 8): recent history improves a human-imitation policy's prediction of a player's wrong move, not an engine-trained one's. For interpreting learned representations, what a network encodes about people is measured against the people who faced its problems.

Figure 1 maps the study. Section 3 describes the data and readouts. Sections 4–6 show that human difficulty is predictable beyond the controls and shared across networks, and that thinking time adds a separate signal. Section 7 tests whether the network uses the two directions, and Section 8 whether recent history at the input predicts a person's actual error. Sections 9 and 10 discuss and conclude.

2 Related work

What is new is the test, on policy networks: increments over engine-search features and the policy's own outputs, then a matched-null intervention with a positive control.

Decodability versus causal use. That a property can be decoded from a representation without being used by the computation reading it is a standing result, established by amnesic probing (Elazar et al., 2021) and a critical literature on what probes can license (Belinkov, 2022). The pattern has since been reported for selective units in vision networks (Hayne et al., 2022), hierarchical state in bracket-sequence transformers (Sharma et al., 2026), and logical validity in language models, where probe-derived directions barely move behaviour against random controls (Sudheendra & Srivastava, 2026). One account may explain it: Liao & Cao (2026) show the same direction in the same representation has no causal effect until training rewards it, so use depends on the objective.

Predicting difficulty directly from items. In chess, a sub-literature predicts Lichess's Glicko-2 puzzle rating, supervised: by fine-tuning Maia-2 with Leela and Stockfish features (Björkqvist, 2025) or a pretrained Transformer with Maia-2 solve-rate inputs (close to our solution-aware control; Miłosz, 2025), from move-prediction error or raw moves (Schütt et al., 2024; Ruta et al., 2024), and earlier, for Chess Tempo difficulties, with a search tree modelling human problem-solving (Bratko et al., 2021); Carow & Witzig (2025) relate time pressure to move risk. We ask whether difficulty is already linearly present in a representation never trained to predict it, so our probes measure rather than compete with such predictors. Probes of chess networks recover concepts of the position or of optimal play (McGrath et al., 2022; Schut et al., 2025), and in a chess language model a steerable player-skill estimate (Karvonen, 2024); Anderson et al. (2016) predict where players err.

Difficulty representations in other model families. Linear probes on many LLMs decode math-problem difficulty IRT-calibrated from human success rates (Lugoloobi & Russell, 2025); a follow-up finds a model-derived signal distinct from human difficulty that diverges from it under extended reasoning (Lugoloobi et al., 2026). The first also steers along its difficulty direction, changing model behaviour. Closest to our design, Lee et al. (2025) probe, localise attention heads, ablate, and relate the result to entropy on LLM math-problem difficulty: the same steps for another label and intervention; we claim no priority. Across over twenty models, Li et al. (2025) find that scaling does not reliably improve alignment with human difficulty and that high task performance often *impedes* difficulty estimation — independent evidence that human difficulty and task skill come apart.

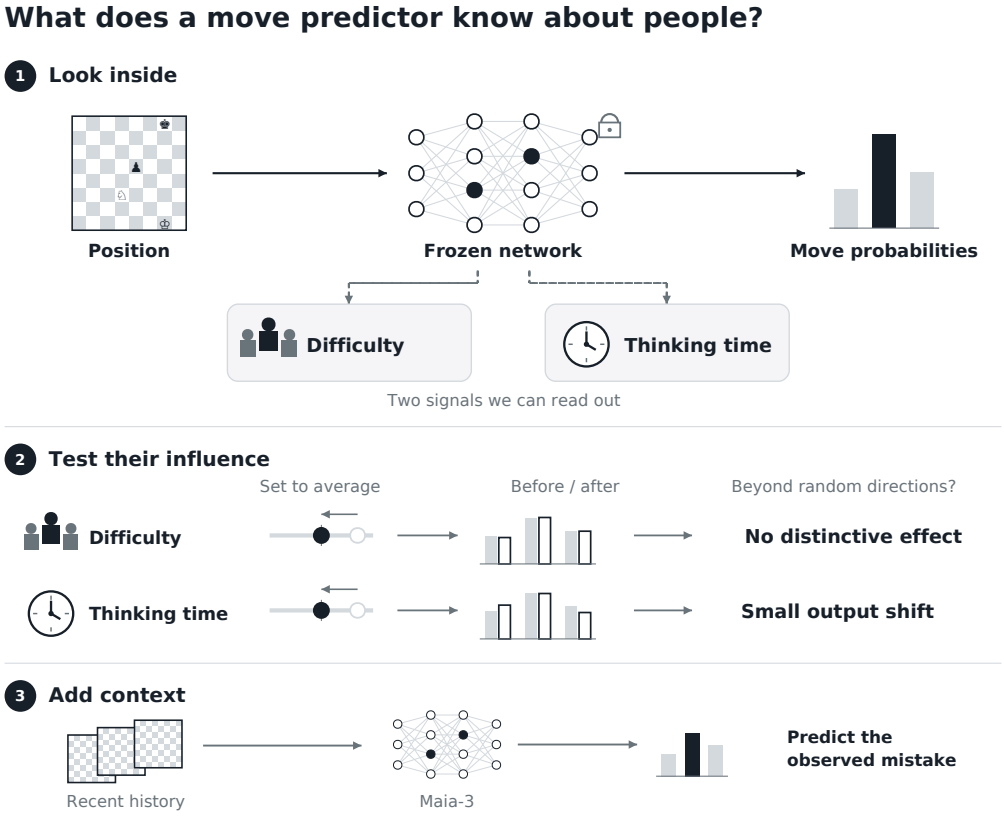


Figure 1: Reading human information from a policy and testing its use. (1) Two linear readouts of a frozen policy's internal features predict population-calibrated difficulty and the original player's log thinking time, two measurements from different people. (2) Setting a readout direction's component to its mean tests the policy's response against matched random directions and a positive control: the difficulty direction shows no distinctive effect, the thinking-time direction a small shift of the move distribution. (3) Varying recent history with the position fixed scores the person's actual wrong move. Bar heights are schematic. Reading a signal out does not show that the policy uses it.

Representational convergence. That independently trained networks converge is argued, mostly for perceptual and semantic content (Huh et al., 2024), with similarity metrics confounded by shared inputs and scale (Gröger et al., 2026); our latent is a human behavioural quantity no network was trained on, so agreement cannot come from a shared training label, and we measure it after removing the target, controls and a raw board encoding. Convergence need not imply use: in language models, information decodable across models at 66% transfer accuracy moves ablated predictions by only 1.5–5.5% (Usama & Chang, 2026), the conjunction we report for another target and domain.

3 Setup and method

Chess supplies Section 1's four needs; we describe its data, models and readouts (Figure 1, panel 1).

One row of data. One row is one Lichess puzzle, a forced tactical sequence from a position in a real online game. Its difficulty rating comes from later players who attempted it as a puzzle. Each puzzle names its source game and move. Replaying that game's record, which has a centisecond clock, recovers the move, thinking time and earlier boards of the player who faced it (the history of Section 8); the puzzle archive alone gives only the rating. The two human measurements therefore come from different people in different settings, which is one reason their agreement is informative. The rating is calibrated from those later players' outcomes and ratings by a Glicko-style system,

so it describes that population; no policy here is trained on it, except in the fine-tuning test of Appendix A.10. The analysis set is $n = 26{,}352$ puzzle positions from games of January 2019 (2019-01, the headline month), with $75{,}596{,}070$ recorded solve attempts (median 427.5 per puzzle) after publication. That month is also a development tier in a concurrent submission to this conference (Appendix A.15); no data split, trained model, estimate, figure, table or text is shared.

Models. We study eight chess-playing networks from three architecture families. The first is Maia-2 (`maia2`) (Tang et al., 2024), a convolutional board encoder with two Transformer blocks. The second is the Chessformer family (Monroe et al., 2026), the Maia-3 capacity ladder (`maia3_5m/23m/79m`), trained to imitate human moves, and Leela-CF (`leela_cf`), distilled at larger scale from engine self-play and never shown a human game. With them we group ChessBench-BC-9M (Ruoss et al., 2024), a bidirectional Transformer trained by behavioural cloning on engine (Stockfish) labels. The third is Allie and Allie-Qwen (`allie`, `allie_qwen`) (Zhang et al., 2025), Transformer decoders that generate each move in turn from the game's log of earlier moves and metadata; Allie is also trained to predict move time. Maia-2, the Maia-3 ladder and the Allie models are also *Elo-conditioned*: they take the original player's rating as an input. A probe could partly read that input back. We therefore control for that rating at every tier, and the effect is present in Leela-CF, which never receives it. Seven of the eight enter the analyses needing a model's own entropy; ChessBench-BC-9M is excluded on grounds fixed before its score was known (its checkpoint's parameter names mismatch its code). Leela-CF lacks entropy on three rows ($n = 26{,}349$). Section 5 also uses nine Maia-1 networks (McIlroy-Young et al., 2020), identical in architecture and size. The control features come from a Stockfish search at depths 12 and 15. This engine is the task-optimal solver of our title, and "task difficulty" means what its features capture.

Probe and estimator. For every model, layer and target we average a layer's activations over its board squares or tokens (mean pooling), standardize them within each training fold, and fit a linear regression with a fixed penalty on large weights (a ridge probe, α=10, set a priori). Predictions are out of fold: five folds, each source game in one fold (`GroupKFold` by `GameId`), each fold predicted from the other four. Every headline number is a nested increment, $\Delta R^2 = R^2(\text{controls} + \text{probe}) - R^2(\text{controls})$ on identical folds. It is an absolute gain in the share of variance explained. Rating rows are weighted by $1/\mathrm{RD}^2$, normalised to mean one over the headline rows, where RD is the rating's uncertainty (itself a target in Appendix A.3), so uncertain ratings count less. Controls enter at three tiers: a weighted linear fit on 8 features, a gradient-boosted tree model (GBM) on the same 8, and a maximal tier, a GBM on 23 features that include them (engine search, position features, solution length, the player's Elo; Appendix A.2). The maximal tier credits the probe only with what the engine's own description of the position cannot explain. Each control family targets one alternative, that the probe reads the task's structure (engine features), the network's uncertainty (policy entropy), the player's skill (rating), the answer itself (Section 4's solution-aware features) or the time available (Section 6's clock controls). Intervals are percentile bootstraps over source games (one puzzle per game, so equivalent to a row bootstrap) in three conventions whose widths are not comparable. The linear tier refits its controls per resample, *fixed-OOF* intervals hold the out-of-fold control predictions fixed, and *full-OOF-refit* intervals redo the whole cross-fit (Appendix A.1).

4 Do activations predict human difficulty beyond baselines?

We first ask whether internal features predict human difficulty beyond the engine and the policy's outputs, since only that increment is new information about the representation; all seven do.

Stronger controls halve the increment. For Maia-2 (Rating, best layer `cnn_block_4`, $n = 26{,}352$), the probe adds $\Delta R^2 = 0.0908$ to the linear controls plus the model's own policy entropy (95% CI $[0.0849, 0.0963]$, n_{boot}=1000, controls refit per resample) and 0.0436 to the gradient-boosted controls plus entropy, which contain the linear ones (full-OOF-refit CI $[0.0406, 0.0504]$, n_{boot}=100), on the same rows. Both tiers and the rule for reading them were fixed in advance; the claim that survives is about half the first number, and no tier between them was run for Maia-2.

The model's own outputs. To check that the probe does not re-read the model's own predictions, we add three of its output features to the maximal tier plus entropy ("C1"), giving "C2": its log-probability of the solution move, the gap between its two most probable moves and whether its top

move is the solution's first move. The first and third need the answer key, which the probe never sees, so C2 is stricter than anything the model produces unaided. The increment is 0.0436 under C1 (the headline) and 0.0350 under the solution-aware C2 (CI $[0.0310, 0.0388]$, n_{boot}=100). The paired shrinkage is 0.0085 (CI $[0.0057, 0.0141]$); 0.805 of the headline survives C2. A probe on the raw 737-column board adds 0.0051, and the same probe on an untrained Maia-2 adds 0.0070–0.0081, so most of the increment depends on training, not on an untrained featurisation of the board.

Across models. Table 1 gives the maximal-GBM increment for all seven models, best layer per model, fixed-OOF 95% CIs, n_{boot}=1000, so the column is like-for-like. Maia-2 has the smallest increment; Leela-CF, never shown a human game, the largest; we ran no paired test across models.

Model	Best layer	n	ΔR^2 [95% fixed-OOF CI]
maia2	`cnn_block_4`	26,352	0.0436 [0.0393, 0.0479]
allie_qwen	`block_25`†	26,352	0.0439 [0.0395, 0.0482]
maia3_5m	`pre_policy`	26,352	0.0499 [0.0456, 0.0547]
allie	`block_22`	26,352	0.0501 [0.0457, 0.0549]
maia3_23m	`transformer_block_7`	26,352	0.0776 [0.0729, 0.0826]
maia3_79m	`transformer_block_7`	26,352	0.1097 [0.1039, 0.1162]
leela_cf	`encoder_14`	26,349	0.1258 [0.1200, 0.1312]
raw board	737 columns	26,352	0.0051 [0.0033, 0.0072]
untrained maia2	`cnn_block_4`, seed 0	26,352	0.0070 [0.0049, 0.0091]
untrained maia2	`cnn_block_4`, seed 1	26,352	0.0081 [0.0059, 0.0103]

Table 1: †allie_qwen: `block_25` is the raw-R^2 sweep maximum, not a verified best layer for the increment (no controlled sweep). Maia-2's full-OOF-refit interval in the text is $[0.0406, 0.0504]$ (n_{boot}=100). The last three rows are baselines (raw board; untrained Maia-2), each under a fifth of Maia-2's increment. All intervals exclude zero.

Deeper search predicts human difficulty *worse*. On a fixed subsample of $n = 8{,}000$ rows, with identical features at three depths, the engine-only model explains $R^2 = 0.3313$ of Rating's variance at depth 12, 0.3194 at depth 15 and 0.3011 at depth 20. The probe's increment over that engine-only model moves the other way for every model in that analysis: `leela_cf` $0.1966 \to 0.2327$, Maia-2 $0.0955 \to 0.1081$ from depth 12 to 20. Within these depths, features and rows, human difficulty is not task difficulty, and the engine-only result needs no probe.

Robustness, briefly. A piece count as a 24th control moves the headline only from 0.0436 to 0.0444; holding a tactical motif out of training keeps a median 0.844 of the motif-seen increment, and 7 of 9 testable motifs clear zero; and it replicates in sign and presence, not magnitude, on 2020-01 and 2022-01 for Maia-2 and Maia-3-79M, months before Maia-3-79M's training window (a temporal separation for Maia-3-79M only; an identical position could recur in another game). Transfer is not shown at one boundary: held out by tag, endgames give -0.0336 (full-OOF-refit CI $[-0.105, +0.0401]$), though each partition carries signal fitted within itself (Appendix A.14).

5 Are difficulty-related signals shared across networks?

If the signal of Section 4 were one network's quirk, others would not share it, so we ask whether it is (convergence, Section 2); it is, which rules out a quirk but not shared board structure, and agreement is lowest between networks trained on different objectives, which also differ in data and scale.

Cross-model transfer. Agreement could come merely from the target, the controls or the board, so we regress all three out of each model's readout in two steps: the shared target and the maximal control block (the 23-feature tier plus the model's own entropy, fitted by GBM), then a linear fit of a raw 737-column board occupancy. The residual readouts agree for all $21/21$ pairs of the seven-model roster, every fixed-OOF cluster-bootstrap CI excluding zero ($n = 26{,}352$): mean pairwise correlation $0.410 \to 0.390$ from before to after the board removal, as for each group below. Agreement is ordered by relatedness: same lineage (4 pairs, $0.551 \to 0.528$) > Maia-2 with Maia-3, one project across architectures (3, $0.463 \to 0.438$) > human-imitation models across projects (8,

$0.422 \rightarrow 0.404$) > Leela-CF with each of them (6, $0.274 \rightarrow 0.257$). The Leela-CF–Maia-3 pairs share an architecture yet fall below every human-imitation pair.

Transfer is asymmetric. Scored as the share of one model's residual readout that another's activations recover, relative to what its own activations recover (normalized transfer R^2), the engine-trained Leela-CF's space predicts the human-imitation models at $0.350 \rightarrow 0.288$ over its six pairings, but theirs predict it at only $0.146 \rightarrow 0.055$ (group means; pairs vary widely, some below zero).

What organizes it. Each model's first-move accuracy on the same rows predicts its maximal-GBM-tier difficulty increment (Section 4's ΔR^2, *not* a raw or linear-tier decodability score): Pearson $r = 0.8696$, exact-permutation $p = 0.0030$ ($n = 7$). Dropping Leela-CF (accuracy 0.961, well above the next model) leaves $r = 0.7714$, $p = 0.082$ at $n = 6$: Leela-CF carries the significance but not the sign. Scale does not: capacity's coefficient in the cross-model meta-regression changes sign and size across reasonable respecifications, so we report no capacity finding.

We report an association and decline the mechanism. Competence co-varies with architecture, capacity and objective breadth across the seven models, so the correlation above cannot attribute the ordering to any one of them. Inside a family where those are held fixed (the nine Maia-1 networks of Section 3), first-move accuracy does *not* predict difficulty-encoding: $r = -0.3117$ at each network's best layer (exact 9!-permutation $p = 0.414$), and $r = 0.076$ ($p = 0.84$) with the readout layer held fixed for all nine. The manipulation is itself weak (accuracy spans only 0.566–0.601 against the seven-model roster's 0.675–0.961); the spread still exceeds sampling error (Appendix A.12).

6 Does thinking time add information beyond puzzle difficulty?

Sections 4–5 concern a rating calibrated from the later players' outcomes and ratings. It reflects whether people succeed, not how long they think, so we next ask whether the representation also carries the original player's thinking time beyond that rating. It does, along a separate direction. Below, *effort* labels the thinking-time signal; the label claims no mechanism. Thinking time in chess has been modelled as resource-rational (Russek et al., 2025), and planning depth has been tied to expertise in four-in-a-row (van Opheusden et al., 2023); we claim only that thinking time is readable beyond the rating. A solver run to a fixed depth never decides how long to look, and the tested engine features do not exhaust what predicts move time.

On the headline month at centisecond resolution ($n = 26{,}064$ of the $26{,}352$ puzzles) we predict the log time spent, with a control block T_0 carrying the maximal complexity features, the model's policy entropy *and* the time left, the time control, the ply (half-move count) and the player's own rating (without them, a time result would measure time pressure). T_1 adds Rating itself.

Beyond the rating. The increment over T_1 is 0.0093 for Maia-2 (fixed-OOF $[0.0065, 0.0121]$; full-OOF-refit $[0.0031, 0.0114]$) and 0.0082 for `leela_cf` (fixed-OOF $[0.0050, 0.0111]$; refit $[0.0058, 0.0138]$), against Rating's own $+0.0138$ and $+0.0117$ over T_0. It replicates on both held-out months for Maia-2 at the reduced 18-feature tier ($+0.0089$ $[0.0051, 0.0127]$ on 2020-01 and $+0.0089$ $[0.0078, 0.0101]$ on 2022-01, $n = 129{,}388$), near the primary magnitude.

The two signals are separable. Out-of-fold predictions of a Rating probe and a thinking-time probe correlate at $r = 0.59$ ($r^2 \approx 0.35$): largely shared but not collinear. The thinking-time probe still adds 0.0077 $[0.0051, 0.0101]$ for Maia-2 and 0.0068 $[0.0038, 0.0095]$ for `leela_cf` over a block containing *both* the true Rating *and* the Rating probe's prediction, so thinking time is not reducible to the rating under these controls. About half the raw effect is shared with the model's policy entropy (a plausible "uncertain position, deliberate longer" mediator, since omitting entropy from T_0 roughly doubles the increment, to 0.0190 and 0.0198), and about half is not.

One place, two directions. Separability could mean two directions in one layer or in two; it is the first. Fitting both probes at every cached layer of three models, both peak at the *same* layer (Maia-2's `cnn_block_4`, Maia-3-79M's final two layers) with a similar shape. The hypothesis that effort would peak at a characteristically different depth is not supported. What holds layer-wide is the separation: the two probes' out-of-fold predictions correlate between 0.45 and 0.59 at all 21 layers examined, lowest where both signals are weakest, never approaching collinearity.

Applying Section 5's cross-model analysis to effort (same roster, folds and removals of target, control block, the model's own entropy and raw board occupancy), the effort component also converges across all seven models, and converges less than difficulty does: mean residual agreement 0.282 against 0.389 for difficulty on identical rows, with all $21/21$ pairwise intervals excluding zero in both cases (0.389 is the paired-comparison reproduction of Section 5's 0.390, differing by seed and by the 288 excluded rows). Effort is thus shared across models but more model-specific than difficulty.

Three limits. The timed player is the original game's mover and Rating comes from a later puzzle-solving crowd, so the two are different people, as verified (Appendix A.14). The 288 puzzles without usable centisecond clocks are systematically easier (mean Rating 1250.95 against 1452.86 overall). And the effect is modest: $\Delta R^2 \approx 0.008$–0.009 of a total R^2 near 0.37–0.43, stable across interval conventions, models, held-out months and, for Maia-2, clock resolutions (Appendix A.4).

7 Do the identified directions affect action selection?

So far, internal features predict human difficulty beyond the engine and the policy's outputs (Section 4), the signal is shared across networks (Section 5), and thinking time adds a separate signal (Section 6). We now ask whether the network uses what it carries about people when it chooses an action. The difficulty direction has no distinctive effect, and the thinking-time direction slightly shifts the move distribution. We test use with a pre-specified ablation of one direction (rank 1) by mean replacement: in Maia-2's `cnn_block_4`, each row's component along the difficulty direction (the raw Rating probe's, whose explained variance the engine controls largely share) is set to its average, and the rest of the network is rerun (Figure 1, panel 2), 5-fold cross-fitted over all $n = 26{,}352$ rows. After the written primary, a five-direction paired contrast, was computed (when is not recorded), the primary became the 50-direction percentile below; the written contrast is also null ($+0.00004$ nats, CI $[-0.00013, +0.00022]$). Effects are in nats and accuracy, not comparable to any ΔR^2. After ablation, a refitted probe recovers raw $R^2 = 0.2773$ of 0.3011 (App. A.5).

Ablating the difficulty direction places its effect on the solution move's negative log-likelihood (NLL) at the 24th percentile of an empirical null of 50 control directions, random directions matched to it in variance, with dose (the size of the displacement) recorded and compared. As a positive control we ablate the policy-Jacobian direction, the one the output is locally most sensitive to, and the same intervention detects it ($+0.00748$ nats), while the difficulty direction's effect sits inside the matched-control range: an effect above the controls' 97.5th percentile, 16–18% of the positive control's, would have stood out. The policy head is not blind to this direction. It lies nearly orthogonal to the policy-Jacobian direction (smallest angle 82.59°), yet their squared cosine is $4.25\times$ that of two random directions in this 256-dimensional space.

What mean replacement can miss. Mean replacement moves each row by $s_i = u^\top(h_i - \mu)$, the distance of its activation h_i along the direction u from the global mean μ (not the mean of the evaluation rows). A row with $s_i = 0$ is left untouched, so the response passes through the origin, as we confirm empirically. Fitting each row's change in NLL as $y_i = \beta_1 s_i + \beta_2 s_i^2$, the fitted response is *dominantly linear*. At one standard deviation of s the linear term is about five times the quadratic. Yet the linear term supplies only 8% of the contrast, the average change over rows, because s averages to nearly zero while s^2 does not: rows pushed one way cancel rows pushed the other. In this fit across rows, a mean-replacement contrast therefore credits little of a response that is mostly slope, a property of the intervention that sharpens standard guidance on interpreting such experiments (Heimersheim & Nanda, 2024; Vaidyanathan et al., 2026).

So we measure the slope directly, by regressing each row's ΔNLL on its own s_i. The difficulty direction's slope is at the 10th percentile of the 50 matched controls, where the policy-Jacobian direction is at the 0th. Fitting both terms leaves it ordinary on each (slope 10th, curvature 12th). Nor does the slope vary distinctively across item-difficulty quintiles: a heterogeneity test places it at the 86th percentile of the same controls. The null holds on every NLL statistic this intervention gives, and the test detects the positive control on two, the average change and the slope.

And it holds on a second architecture. Repeating the whole design on Maia-3-79M, a transformer rather than a CNN, at its own best layer with the same rows, probe, folds, control families

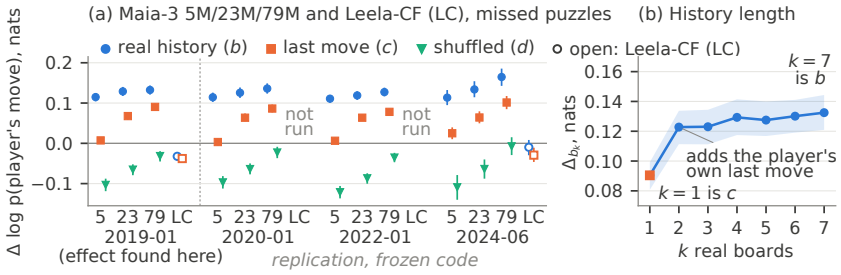


Figure 2: Δ: change in log-probability of the player's wrong move. (a) History: real (b), last-move (c) or shuffled (d). (b) Maia-3-79M, 2019-01. 95% intervals. Real history raises Δ for Maia-3 only.

and rank, puts the difficulty direction at the 64th percentile of its matched controls, while the policy-Jacobian positive control's signed policy entropy is 159 times the null's own standard deviation.

Ablating effort shifts the move distribution, not the solution move. The same intervention on the effort direction (same layer, rank, folds, metrics, 50 matched controls; the effort probe is unweighted and its row set is the 26,064 think-time rows, while the difficulty probe is rating-weighted) gives different answers by metric. On the solution move's NLL, its pre-specified primary outcome, it is as null as the difficulty direction (14th percentile; a consistent null by its written rule). On two of the three listed secondary outcomes, which no decision rule covered, it is not: the KL divergence between the move distributions before and after is $+0.00090$ nats against a null mean of $+0.000063$, and the top-1 flip rate, how often the most likely move changes, is 1.04% against 0.275%, both above all 50 controls, though small in absolute terms. Exploratory, not dose-checked within rating quintiles, both effects grow with puzzle rating faster than under any control; the difficulty direction's do not. The effort direction's dose is only mildly above the controls', and, assuming additive effects, the KL excess survives removing a policy-entropy direction's effect (Appendix A.11).

Decoded-but-unused directions are a standing result (Section 2). We add an unusual target, human difficulty, absent from every training objective, and a matched-control test under which its use is not shown. One account (Liao & Cao, 2026) may explain the null; our data cannot test it. They find invariants decodable at $R^2 \approx 1$ yet without causal effect, and show that the same direction in the same representation becomes load-bearing once the training objective rewards it. Human item difficulty is the case that account predicts: it is available in the input, useful enough to be encoded, and rewarded by no difficulty term in these networks' losses (only Allie's includes a timing term). On that reading the null is expected, and the account makes a prediction we can partly test post hoc. Fine-tuning only the downstream layers on human solve rates raises the correlation of the policy's solution probability with the expected human solve rate from 0.37 to 0.50 (0.42 under shuffled ratings), yet leaves this direction ordinary (slope at the 14th percentile of matched controls; Appendix A.10). In this form the prediction is not borne out. Whatever difficulty the policy learned to use did not route through this rank-1 direction at this layer. The direction can still be read out.

8 Does recent history improve prediction of human errors?

So far the representation carries human difficulty and thinking time, and only the thinking-time direction shifts the move distribution beyond matched random ones, all but the Allie models' from the current position. A solver's answer depends only on the problem; a person's error may also depend on how they got there. We therefore ask whether recent history helps predict a person's actual wrong move, as a user model would need. For a human-imitation policy it does; for an engine-trained one it does not. Tested at the input, it complements Section 7 without bearing on the difficulty direction. Maia-3 reads up to seven earlier boards of the game. On items the player failed (missed puzzles, 7,053 of the 26,165 with a recorded outcome on 2019-01), we score their wrong action given the current state only (a), the real history (b), only the opponent's last action (c), or another episode's history (d, "shuffled"). We measure the change Δ in its log-probability relative to a, in nats (0.132 nats multiplies a probability by about 1.14).

On failed items real history raises that probability, beating c and d at every model size (paired intervals excluding zero), while shuffled history never raises it (Figure 2). Found post hoc on 2019-01, it was rerun on three later months with frozen code and criteria fixed in advance: all four criteria hold at every size, 36 of 36 checks (48 of 48 with 2019-01), with Δ_b between 0.111 and 0.165 nats. The engine-trained Leela-CF, given the same history, moves the other way on 2019-01 (-0.032 $[-0.043, -0.022]$) and is indistinguishable from zero on 2024-06 (-0.010 $[-0.030, 0.008]$, $n = 2{,}515$ failed items). The history-aware readout also predicts whether this player misses beyond Rating, their own rating and the clock: replacing the current-state readout with it raises out-of-fold AUC by $+0.027$ $[0.024, 0.029]$ on 2019-01. Beyond c, most of the gain comes from the board showing the player's own previous move (83.3% and 77.0% of it for 23M and 79M on 2019-01), and it is largest for the fastest third of errors (0.176–0.204, against 0.077–0.096 for the slowest): context predicts the least deliberated errors. History improves human move prediction on average (Monroe et al., 2026; Tarun Kumar S, 2026), and fine-tuning on a player's past games raises their moves' probability, errors included (McIlroy-Young et al., 2022); ours comes from the episode in progress (App. B.8), so it concerns the trajectory rather than the player's identity. One history-capable human-imitation family was tested, effects are modest, and the engine contrast, also differing in objective, scale and input encoding, isolates no cause. The gain may be more probability on wrong moves generally rather than on this one; it was found post hoc, then replicated without pre-registration; and 2024-06, the largest, is in 79M's training window (0.111–0.136 nats without it; App. B.7).

9 Discussion

What the findings change. Linear readouts of policies trained to act, none on difficulty, recover a human population's difficulty and the original player's thinking time beyond the tested engine features and the policies' outputs, shared across networks and along separate directions in one layer. The difficulty direction has no distinctive effect under the intervention, so reading a signal out does not show the network acts on it; a signal this intervention does not move can still be read out and used, to adapt systems to users (Section 8) or to trace mechanism. Mean replacement can hide a mostly linear response unless the slope is measured (Section 7).

What they do not establish. They identify no cognitive mechanism: the networks are not models of people, the resource-rational reading of thinking time is untested, and what the residual signal encodes is unidentified. The raw-board and untrained baselines (Table 1) show only that most of the increment depends on training. No claim concerns an individual's cognition, the two human measurements come from different people (Section 6), and no application is evaluated. The null holds for the tested rank-1 intervention, direction, layer and metrics only, not for the beyond-engine residual. The prediction of Liao & Cao (2026) was not borne out in the one form tested (representation frozen). Effects are modest, transfer to held-out endgames is not shown, and no primary table is corrected for multiplicity (App. A.14).

Why a second domain is needed. All this comes from one domain, so its general claims are hypotheses for a second domain that supplies the four things of Section 1. In language models, decodable, convergent information also barely moves ablated predictions (Usama & Chang, 2026); whether these controlled findings hold elsewhere is open.

10 Conclusion

A network trained to choose actions can carry information about how hard its problems are for people. In the policies studied, frozen features predict population-calibrated difficulty beyond an engine and the policies' outputs, a signal shared across networks, and the original player's thinking time along a separate direction in one layer. Use of the difficulty direction is not shown: under the tested intervention it has no distinctive effect on the solution move, while the thinking-time direction measurably shifts the move distribution (pre-specified secondary outcomes, small in absolute terms) without favouring or disfavouring the solution relative to matched controls. Both signals can still be read out and could be used. Recent history improves a human-imitation policy's prediction of a person's actual error, unlike an engine-trained network's. These answers come from one domain, chess.

REPRODUCIBILITY STATEMENT

All data are public. Puzzles, their human-calibrated ratings and rating deviations come from the Lichess puzzle database, and the positions they are drawn from come from the Lichess monthly game dumps (Lichess, 2026); no non-public data holding is required to reproduce any number in this paper. The analysis set is the $n = 26{,}352$ puzzle rows described in Section 3, and the per-model checkpoints are the public releases cited there. The engine features come from Stockfish (The Stockfish developers, 2026), and Leela-CF is a Leela Chess Zero (lc0) network (The Leela Chess Zero developers, 2026). Section 8 also uses the 2020-01, 2022-01 and 2024-06 puzzle months (Appendix B.1).

Appendix A.1 states the estimator step by step; its hyperparameters were fixed a priori, and the layer is chosen on the evaluation rows. The probe is a ridge regression on mean-pooled layer activations, standardised within each training fold, predicting the target out of fold under 5-fold `GroupKFold` clustered by `GameId`; the ridge penalty is $\alpha = 10$, fixed a priori. Control tiers are fitted on the same folds: weighted least squares at the linear tier, and `HistGradientBoostingRegressor` (Pedregosa et al., 2011) at the two gradient-boosted tiers (hyperparameters in Appendix A.1). Weighting is per target, not global: on the Rating target observations are weighted by $1/\mathrm{RD}^2$, normalised to mean one over the headline rows (`rating_weight`), while the rating-uncertainty target of Appendix A.3 is fitted *unweighted*. The global seed is 42 except in Section 8, whose seeds are stated in Appendix B.

Interval types are not interchangeable and we label each one wherever it appears, as Appendix A.1 sets out: the linear tier refits its controls inside every resample ($n_{\text{boot}} = 1000$); the canonical gradient-boosted tier conditions on fixed out-of-fold predictions ($n_{\text{boot}} = 1000$); and the intervals labelled full-OOF-refit redo the entire inner cross-fit inside every resample ($n_{\text{boot}} = 100$). All are percentile bootstraps over `GameId` clusters. These intervals resample source games only. They condition on the layer chosen for each model (the argmax of out-of-fold decodability, on the same rows), on the stated seeds, and on one-way clustering by game, so games that share a player are not grouped. The results are also conditional on the fixed choices of the estimator (the ridge penalty, mean pooling, the control features and their search depths, rank-1 ablations); where one is varied, the text says so (ablation rank, Appendix A.6; search depth, Section 4).

The derived tables available with the code map each reported number to its source cell. The analysis code (the control-ladder battery, the output-control ladder, the held-out-motif generalization run, the ablation study, the readout fine-tuning test of Appendix A.10 (training-set construction, fine-tuning, the ablation rerun, and its human-likeness and percentile tables) and the Section 8 history analyses) is included in the supplementary material (`code/`), together with the derived tables behind the reported analyses (the code README lists the exceptions), and is organised so that each script writes the table it is cited for.

Both ablations of Section 7, of the difficulty direction and of the effort direction, were each specified in writing before its own first forward pass, and those specifications (design, decision rule, primary and secondary outcomes, and the interpretation attached to each possible outcome) are included in the supplementary material; the supplied copies carry the editorial edits listed in the supplement's README, and the unaltered originals are available from the authors, so that the null we report can be checked against what was committed to in advance. One amendment, listed there too: the difficulty ablation's primary statistic changed from the written five-direction paired contrast to the 50-direction percentile after that contrast had been computed (when is not recorded), with more rows and controls; by the original rule the result is also a null ($+0.00004$ nats, CI $[-0.00013, +0.00022]$). The effort run's decision rule covered its primary outcome only, not its three listed secondary outcomes, two of which we report. We describe them as pre-specified rather than pre-registered: they are dated internal documents, not entries in a public registry. The readout fine-tuning test (Appendix A.10) is exploratory: it was designed after these results, though its prediction was written down before it was run.

The supplementary material contains the pre-specification documents of Sections 7 and 8 and a note on the status of the concurrent manuscripts; the analysis code is included in the supplementary material (`code/`). A concurrent submission to this conference shares the public puzzle archive and the January 2019 archive; Appendix A.15 states what is and is not shared.

ETHICS STATEMENT

This work uses public online chess game and puzzle data from the Lichess open database, which the platform publishes in bulk for research use; players appear there only under pseudonymous usernames, which we do not report, publish or attempt to de-anonymize. The Lichess puzzle and game databases are released under the CC0 licence (Lichess, 2026). This is a secondary analysis of publicly released, pseudonymous game records; no participants were recruited, no new data were collected, and no username is reported. Puzzle ratings, rating deviations and solve counts are aggregates the platform had already computed. Individual players' records from the public game data are also used: the original mover's rating, a control throughout, and in Sections 6 and 8 the move played, the time spent on it, the clock and whether they found the puzzle's solution. We report them only as statistics over many positions, never for a named or identifiable player. Any future controlled human study would require its own IRB review and informed-consent process, which we would seek before running it. The measurements describe populations and model representations, not individuals; readouts of difficulty or thinking time could in principle profile players, so we report only aggregate results and release no per-player estimates. Beyond this profiling risk, we see no dual-use or safety risk specific to this work: it studies interpretability of small, openly available chess-playing policy networks rather than frontier or safety-relevant systems.

AI USE STATEMENT

Generative AI tools (Claude, Anthropic, operated through an agentic coding tool) were used under the authors' direction, working from the authors' own literature review, hypotheses, analyses and specifications, for the following tasks with required disclosure: refining the authors' hypotheses, in part (the history analysis of Section 8 was refined from the authors' hypothesis in the light of the results, and the readout fine-tuning test of Appendix A.10 was suggested during that process); giving feedback on the authors' research design and experiments, in part, within the authors' decision rules; implementing methods (the analysis scripts, written by AI assistants under the authors' direction); cleaning and reformatting datasets (probe-table construction and row filtering); and interpreting results, in part, with the authors verifying every reported number against its source file and deciding every claim. We have not used generative AI tools to develop theoretical models or conceptual frameworks, to formulate mathematical claims, to provide ingredients for or write proofs, or to generate synthetic data; translation and qualitative or thematic data analysis are not applicable to this work. Additionally, we used generative AI tools for drafting and editing text from the authors' inputs, analyses and specifications, for producing figures from the authors' results, and for searching and summarising the literature in support of the authors' own review. The authors formulated the research questions and hypotheses, designed the study, directed and verified the analyses, and take full ownership of and responsibility for every process involved in the preparation of this document, including its text, claims, code, and artefacts.

REFERENCES

Ashton Anderson, Jon Kleinberg, and Sendhil Mullainathan. Assessing human error against a benchmark of perfection. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 705–714, 2016. doi: 10.1145/2939672.2939803.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 2022. doi: 10.1162/coli_a_00422.

Sebastian Björkqvist. Estimating the difficulty of chess puzzles by combining fine-tuned Maia-2 with hand-crafted and engine features. In *2025 20th Conference on Computer Science and Intelligence Systems (FedCSIS)*, volume 43 of *Annals of Computer Science and Information Systems*, pp. 801–806, 2025. doi: 10.15439/2025f6497.

Ivan Bratko, Dayana Hristova, and Matej Guid. Predicting problem difficulty in chess. In *Human-Like Machine Intelligence*, pp. 487–504. Oxford University Press, 2021. doi: 10.1093/oso/9780198862536.003.0024.

Johannes Carow and Niklas M. Witzig. Time pressure and strategic risk-taking in professional chess. *Journal of Economic Behavior & Organization*, 238:107218, 2025. doi: 10.1016/j.jebo.2025.107218.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. Amnesic probing: Behavioral explanation with amnesic counterfactuals. *Transactions of the Association for Computational Linguistics*, 9:160–175, 2021. doi: 10.1162/tacl_a_00359.

Thomas L. Griffiths, Falk Lieder, and Noah D. Goodman. Rational use of cognitive resources: Levels of analysis between the computational and the algorithmic. *Topics in Cognitive Science*, 7 (2):217–229, 2015. doi: 10.1111/tops.12142.

Fabian Gröger, Shuo Wen, and Maria Brbić. Revisiting the platonic representation hypothesis: An Aristotelian view. *arXiv preprint arXiv:2602.14486*, 2026. doi: 10.48550/arXiv.2602.14486.

Lucas Hayne, Abhijit Suresh, Hunar Jain, Rahul Kumar, and R. McKell Carter. Much easier said than done: Falsifying the causal relevance of linear decoding methods. *arXiv preprint arXiv:2211.04367*, 2022. doi: 10.48550/arXiv.2211.04367.

Stefan Heimersheim and Neel Nanda. How to use and interpret activation patching. *arXiv preprint arXiv:2404.15255*, 2024. doi: 10.48550/arXiv.2404.15255.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2733–2743, 2019. doi: 10.18653/v1/D19-1275.

Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. The platonic representation hypothesis. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, 2024.

Thomas Johnson. ChessMimic: Per-rating transformer models for human move, clock, and outcome prediction in online blitz chess. *arXiv preprint arXiv:2606.04473*, 2026. URL https://arxiv.org/abs/2606.04473.

Adam Karvonen. Emergent world models and latent variable estimation in chess-playing language models. In *Conference on Language Modeling (COLM 2024)*, 2024. URL https://arxiv.org/abs/2403.15498.

Sunbowen Lee, Qingyu Yin, Chak Tou Leong, Jialiang Zhang, Yicheng Gong, Shiwen Ni, Min Yang, and Xiaoyu Shen. Probing the difficulty perception mechanism of large language models. *arXiv preprint arXiv:2510.05969*, 2025. doi: 10.48550/arXiv.2510.05969.

Ming Li, Han Chen, Yunze Xiao, Jian Chen, Hong Jiao, and Tianyi Zhou. Can LLMs estimate student struggles? Human-AI difficulty alignment with proficiency simulation for item difficulty prediction. *arXiv preprint arXiv:2512.18880*, 2025. doi: 10.48550/arXiv.2512.18880.

Quanhao Li and Wei Jiang. Tracking vs. deciding: The dual-capability bottleneck in searchless chess transformers. *arXiv preprint arXiv:2603.29761*, 2026. URL https://arxiv.org/abs/2603.29761.

Chih-Ting Liao and Xin Cao. The objective decides: When a learned dynamics model uses a conserved quantity. *arXiv preprint arXiv:2607.03728*, 2026. doi: 10.48550/arXiv.2607.03728.

Lichess. lichess.org open database. https://database.lichess.org/, 2026. Accessed 26 September 2026.

William Lugoloobi and Chris Russell. LLMs encode how difficult problems are. *arXiv preprint arXiv:2510.18147*, 2025. doi: 10.48550/arXiv.2510.18147.

William Lugoloobi, Thomas Foster, William Bankes, and Chris Russell. LLMs encode their failures: Predicting success from pre-generation activations. *arXiv preprint arXiv:2602.09924*, 2026. doi: 10.48550/arXiv.2602.09924.

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

Reid McIlroy-Young, Siddhartha Sen, Jon Kleinberg, and Ashton Anderson. Aligning superhuman AI with human behavior: Chess as a model system. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2020)*, pp. 1677–1687, 2020. doi: 10.1145/3394486.3403219.

Reid McIlroy-Young, Russell Wang, Siddhartha Sen, Jon Kleinberg, and Ashton Anderson. Learning models of individual behavior in chess. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2022)*, pp. 1253–1263, 2022. doi: 10.1145/3534678.3539367.

Szymon Miłosz. Pretraining transformers for chess puzzle difficulty prediction. In *2025 20th Conference on Computer Science and Intelligence Systems (FedCSIS)*, volume 43 of *Annals of Computer Science and Information Systems*, pp. 831–835, 2025. doi: 10.15439/2025f7603.

Daniel Monroe, George Eilender, Philip Chalmers, Zhenwei Tang, and Ashton Anderson. Chessformer: A unified architecture for chess modeling. In *International Conference on Learning Representations (ICLR 2026)*, 2026. URL https://arxiv.org/abs/2605.19091.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Anian Ruoss, Grégoire Delétang, Sourabh Medapati, Jordi Grau-Moya, Li Kevin Wenliang, Elliot Catt, John Reid, Cannada A. Lewis, Joel Veness, and Tim Genewein. Amortized planning with large-scale transformers: A case study on chess. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024)*, pp. 65765–65790, 2024. doi: 10.52202/079017-2102.

Evan M. Russek, Daniel Acosta-Kane, Bas van Opheusden, Marcelo G. Mattar, and Thomas L. Griffiths. Time spent thinking in online chess reflects the value of computation. *Cognitive Science*, 49(10):e70119, 2025. doi: 10.1111/cogs.70119.

Dymitr Ruta, Ming Liu, and Ling Cen. Moves based prediction of chess puzzle difficulty with convolutional neural networks. In *2024 IEEE International Conference on Big Data (BigData)*, pp. 8390–8395, 2024. doi: 10.1109/BigData62323.2024.10825595.

Lisa Schut, Nenad Tomašev, Thomas McGrath, Demis Hassabis, Ulrich Paquet, and Been Kim. Bridging the human–AI knowledge gap through concept discovery and transfer in AlphaZero. *Proceedings of the National Academy of Sciences*, 122(13):e2406675122, 2025. doi: 10.1073/pnas.2406675122.

Anan Schütt, Tobias Huber, and Elisabeth André. Estimating chess puzzle difficulty without past game records using a human problem-solving inspired neural network architecture. In *2024 IEEE International Conference on Big Data (BigData)*, pp. 8396–8402, 2024. doi: 10.1109/bigdata62323.2024.10826087.

Aryan Sharma, Cutter Dawes, and Shivam Raval. Dissociating decodability and causal use in bracket-sequence transformers. *arXiv preprint arXiv:2604.22128*, 2026. doi: 10.48550/arXiv.2604.22128.

Smitha Muthya Sudheendra and Jaideep Srivastava. When decodability is not enough: Logical validity representations, behavioral dissociation, and causal tests in language models. *arXiv preprint arXiv:2609.02438*, 2026. doi: 10.48550/arXiv.2609.02438.

Zhenwei Tang, Difan Jiao, Reid McIlroy-Young, Jon Kleinberg, Siddhartha Sen, and Ashton Anderson. Maia-2: A unified model for human-AI alignment in chess. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024)*, pp. 20919–20944, 2024. doi: 10.52202/079017-0659. URL https://arxiv.org/abs/2409.20553.

Tarun Kumar S. Otter: A time-aware, history-conditioned human chess AI. *arXiv preprint arXiv:2608.05206*, 2026. URL https://arxiv.org/abs/2608.05206.

The Leela Chess Zero developers. Leela Chess Zero. https://lczero.org/, 2026. Accessed 26 September 2026.

The Stockfish developers. Stockfish open-source chess engine. https://stockfishchess.org/, 2026. Accessed 26 September 2026.

Muhammad Usama and Dong Eui Chang. Convergence without understanding: When language models agree on representations but disagree on reasoning. *arXiv preprint arXiv:2605.23315*, 2026. doi: 10.48550/arXiv.2605.23315.

Sankaran Vaidyanathan, David Arbour, Aaron Mueller, Scott Niekum, and David Jensen. The curse of multiple mediators: Hidden interaction effects in activation patching. *arXiv preprint arXiv:2606.27510*, 2026. doi: 10.48550/arXiv.2606.27510.

Bas van Opheusden, Ionatan Kuperwajs, Gianni Galbiati, Zahy Bnaya, Yunqi Li, and Wei Ji Ma. Expertise increases planning depth in human gameplay. *Nature*, 618(7967):1000–1005, 2023. doi: 10.1038/s41586-023-06124-2.

Yiming Zhang, Athul Paul Jacob, Vivian Lai, Daniel Fried, and Daphne Ippolito. Human-aligned chess with a bit of search. In *International Conference on Learning Representations (ICLR 2025)*, 2025. URL https://arxiv.org/abs/2410.03893.

A Supplementary methods and results

This appendix gives the interval conventions and the inert grouping of Section 3, the control features behind Sections 3–6, rating uncertainty as a second target, the effort signal at a second clock resolution, how the ablation's matched null in Section 7 is built, and the results that the main text summarises: the higher-rank ablations, the centred-ablation decomposition, the dose gradient, the second architecture, the readout fine-tuning test, the effort direction, competence inside an architecture-matched family, cross-model agreement by pair, the limitations in full, and the data shared with a concurrent submission. Appendix B covers Section 8.

A.1 Interval conventions and the inert grouping

Two properties of the design. First, **grouping by source game is inert on this corpus.** `GameId` and `PuzzleId` are in exact 1:1 correspondence on all 26,352 puzzle rows, so `GroupKFold`-by-`GameId` and the cluster bootstrap built on it reduce to ordinary K-fold and an i.i.d. row bootstrap; the estimates and intervals are unaffected, but no claim that they are robust to within-game correlation follows, because no such correlation exists in this design. We keep the grouping because it costs nothing. Second, three interval conventions appear in this paper and are never compared for width: the linear tier refits its controls on every resample (n_{boot}=1000); the maximal-GBM tier fixes out-of-fold predictions with no refit (n_{boot}=1000, fixed-OOF); and our headline row's interval redoes the entire cross-fit inside every resample (n_{boot}=100, full-OOF-refit). The refit/fixed width ratio is 1.13 at $n = 26{,}352$ and 1.68 at an $n = 3{,}320$ robustness stratum — not a constant, so we name n each time an interval appears.

The estimator step by step. (1) Rows are split by `GameId` into five folds (`GroupKFold`). (2) Within each training fold the activations are standardized and the ridge probe (α=10) is fitted to the target, with Rating rows weighted by $1/\mathrm{RD}^2$, normalised to mean one over the headline rows; its predictions on the held-out fold are the probe's out-of-fold predictions. (3) The control block is fitted on the same folds, once on the controls alone and once on the controls plus the probe's out-of-fold prediction: weighted least squares at the linear tier, and at the gradient-boosted tiers a `HistGradientBoostingRegressor` (`max_iter` 200, `learning_rate` 0.05, `l2_regularization` 1.0, unbounded depth, `random_state` 42). (4) $\Delta R^2 = R^2(\text{controls} + \text{probe out-of-fold prediction}) - R^2(\text{controls})$, both evaluated out of fold on the identical fold structure. (5) The layer is the argmax of out-of-fold decodability on the same rows and is held fixed in the bootstrap, so the intervals do not propagate the uncertainty of that selection (Allie-Qwen uses the raw-R^2 sweep maximum; no controlled sweep was run). Whether the probe predictions used for a control fit's training rows came from probes that had seen that fit's test fold is not recorded. The full-OOF-refit intervals are mildly upward-biased and not centred on their point estimates. The baselines of Table 1 use the same folds, ridge probe and C1 controls, with only the probe's input changed: the raw board encoding (12 piece types $\times$ 64 squares plus side to move, the

737 columns non-constant on these rows), or the Maia-2 architecture with random weights (seeds 0 and 1), read at `cnn_block_4` with the same mean pooling. C1's entropy control is trained Maia-2's on every row.

A.2 CONTROL FEATURES

The control ladder. Table 2 reports the engine-alone R^2 (no entropy, no probe) for the three tiers (rows 1, 3: Maia-2 / Rating / `cnn_block_4`, $n = 26{,}352$; row 2: no Maia-2 row — see caption). The linear and flexible tiers control for the *same* 8 features (`n_legal`, `gap_best_second_cp`, `eval_top5_var`, `ply`, `material_diff`, `material_total`, `solution_len`, `solver_elo` (the original mover's rating)), with the two Stockfish-derived features at depth 12; the maximal tier's 23 features strictly contain those 8, adding eval-level and near-optimal-move-count features at depth 12, the analogous block at depth 15, phase indicators and check/forced flags. In full, the additions are the depth-12 search features `eval_best_cp`, `eval_second_cp`, `n_moves_within_50cp` and `n_lines_returned`; the depth-15 block `eval_best_cp_d15`, `eval_second_cp_d15`, `gap_best_second_cp_d15`, `eval_top5_var_d15` and `n_moves_within_50cp_d15`; and the board features `phase_score`, `phase_opening`, `phase_middlegame`, `phase_endgame`, `in_check` and `is_forced`.

Tier	Estimator	#features	R^2(controls)
Linear	WLS	8	0.291246
Flexible	GBM	8 (same)	0.390360
Maximal	GBM	23	0.402231

Table 2: Engine-alone R^2 (no entropy, no probe), n=26,352. Row 2 is shared bit-for-bit by all five flexible-tier rows at this n; no Maia-2 row exists for that comparison. With the same features, the flexible (GBM) tier explains more than the linear one: a linear control is too weak.

The nonlinearity point. Holding the 8 features and the n=26,352 rows fixed and swapping only the estimator raises engine-alone R^2 from 0.291246 (WLS) to 0.390360 (GBM): board complexity relates to Rating nonlinearly, and a linear control is too weak. The 0.390360 figure is a property of the control block alone, not of any probed model — it recurs bit-for-bit on every flexible-tier row at $n = 26{,}352$ (maia3_79m, maia3_23m, maia3_5m, allie, allie_qwen); Maia-2 has no row in this comparison. The maximal tier's 0.402231 must *not* be read as further evidence of nonlinearity against 0.291246: it changes the feature set from 8 to 23 as well as the estimator, so that comparison is confounded.

A.3 RATING UNCERTAINTY AS A SECOND TARGET

The same pipeline applied to $y = \log \mathrm{RatingDeviation}$ — the uncertainty of the item's rating estimate, not how hard it is, restricted to well-played puzzles (`NbPlays` ≥ 100), unweighted, $n =$ 19,331 (Leela-CF $n = 19{,}328$) — adds $\Delta R^2 = 0.0434$ at Maia-2's own best-decodable layer on this target, `cnn_block_4`, over a 24-feature maximal tier (the 23 features of Section 3 plus the log number of recorded attempts, which this target carries at every tier) plus entropy; 95% CI $[0.0367, 0.0502]$, fixed-OOF, n_{boot}=1000 — a different row set and a different interval convention from the Rating headline, so the two are named separately rather than compared for width.

A.4 THE EFFORT SIGNAL AT WHOLE-SECOND CLOCK RESOLUTION

Section 6 measures thinking time at centisecond resolution. The second resolution is the whole-second clock of the standard game record: the increment over T_1 is then 0.0079 $[0.0048, 0.0108]$ for Maia-2 (centisecond 0.0093) and 0.0112 $[0.0081, 0.0144]$ for `leela_cf` (centisecond 0.0082), fixed-OOF, overlapping the centisecond intervals. The result is therefore stable across clock resolutions for Maia-2 and larger for `leela_cf` at whole seconds.

A.5 THE ABLATION'S MATCHED NULL

Ablating the difficulty direction (rank 1, mean-replacement) places its effect on the solution move's NLL at the **24th percentile of an empirical null built from 50 variance-matched control directions (dose recorded, not matched)** (null mean -0.00007 nats, SD 0.00075 nats, $n = 26{,}352$ puzzle rows) — on this average, indistinguishable from a generic matched-variance direction, while the identical intervention on the policy-Jacobian direction is detected ($+0.00748$ nats, far outside this same null).[1] The null is not empty: the same code, rank and rows detect the rank-1 policy-Jacobian direction, a positive control for the machinery, at ΔNLL $+0.00748$ nats, KL 0.00599 and top-1 flip 2.87%. The difficulty direction's effect lies inside the matched-control cloud (the 24th, 66th and 64th percentiles on ΔNLL, KL and top-1 flip). An effect above the controls' 97.5th percentile, 0.0012 nats above the null's own mean and 16.4% of the policy-Jacobian direction's effect on the same rows with the same code, would have shown as an outlier, as the policy-Jacobian direction did. The null is centred slightly negative, a systematic property of mean replacement toward a global mean, shared by every condition. Two cross-checks give similar values: a one-sided 95% normal cross-check (1.645 SD) gives 0.00123 nats and the central-95% half-width gives 0.00134, so the range is 16–18%; with 50 controls the tails are coarsely resolved.

Across an empirical null of 50 variance-matched directions at rank 1, the difficulty direction sits at the 24th percentile on ΔNLL, 66th on KL, 64th on top-1 flip rate, and 34th on Δtop-1 accuracy — squarely inside the control cloud on every full-sample metric. The KL and flip-rate readings stay inside it when split by puzzle rating as well (slopes at the 32nd and 62nd percentile of the control slopes, $|z| < 0.5$); the ΔNLL reading does not, and Appendix A.8 shows why. Each control direction starts as a random direction and is tilted until it carries the same share of activation variance as the tested direction, then ablated the same way (the tilt is along the eigenvectors of the activation covariance, set by bisection on the variance share averaged over the five folds). How far the ablation moves each row (its dose, the mean size of the displacement relative to $\|h_i - \mu\|$) was recorded, not fixed in advance; the difficulty direction's dose lies within the controls' range. The ablated direction is the raw Rating probe's (fitted to Rating itself, not to its residual after the engine controls), and most of the variance that probe explains is shared with the engine controls. A probe refitted on the ablated activations (each fold ablated along the direction fitted on its own training rows) still recovers $R^2 = 0.2773$ $[0.2690, 0.2857]$ of the 0.3011 before ablation on these $n = 26{,}352$ rows, and 0.2755 $[0.2669, 0.2833]$ of the 0.2995 on the $n = 26{,}064$ think-time rows (effort there: 0.1370 $[0.1292, 0.1450]$ of 0.1427; fixed-OOF row-bootstrap 95% intervals, n_{boot}=1000): the rank-1 edit removes the fitted direction but most of the decodable information survives along correlated directions. The null therefore concerns that direction at that layer, neither the beyond-engine residual specifically nor the information the direction carries. How the pooled-space edit is applied to each square's activation, the normalization and coordinate system of u, and the construction of the policy-Jacobian direction are not stated here; the pre-specification in the supplementary material describes the intended construction.

The written rule's other control families. The written specification also named five random directions (`rand`), five directions fitted by the same estimator to permuted Rating labels (`shuffle`; a control task in the sense of Hewitt & Liang (2019)) and one fitted to the model's own policy entropy, and its rule for a positive result also required the difficulty direction to exceed `rand` and `shuffle`. All were run. The primary contrast was already null, so that condition was never reached; for the record, on the solution move's NLL at rank 1 the difficulty direction minus the random family is $+0.00056$ nats (95% row-bootstrap CI $[+0.00038, +0.00074]$; the random directions also move each row further), minus the label-shuffled family -0.00051 ($[-0.00067, -0.00035]$, the wrong sign) and minus the entropy direction $+0.00212$ ($[+0.00180, +0.00243]$). These are five-direction (or single-direction) contrasts, subject to the caveat of the footnote above. The effort run's written rule (Appendix A.11) is stated against the variance-matched null only; its random, label-shuffled and entropy families were run as well and are in the derived tables.

[1] A fixed *five*-direction control family gives a numerically similar-looking point estimate ($+0.00004$ nats, CI $[-0.00013, +0.00022]$), but that interval is not a stable estimate of anything: the between-direction sampling error of a five-draw mean, estimated from the 50-direction null itself, is ≈ 0.0003 nats — comparable to the interval's own width — and a fresh five-direction draw on a closely related row set gives -0.00068. Both are unremarkable draws from the same noisy process. We retain the number because it shows *why* a 50-direction null was necessary, but the percentile above is the trustworthy test.

A.6 HIGHER RANKS

Higher ranks do not rescue the difficulty direction: the ΔNLL contrast stays negative from rank 5 (-0.00195) through rank 50 (-0.01623), and the KL contrast turns positive and grows only from rank 25 onward (-0.00005 at rank 1, $+0.00251$ at rank 25, $+0.00641$ at rank 50) — exactly the ranks at which the minimum angle to the policy-Jacobian subspace collapses ($82.6°$ at rank 1, $38.1°$ at rank 25, $23.6°$ at rank 50) — the subspace has folded onto the policy read-out, so that KL rise is not evidence about difficulty.

A.7 THE CENTRED-ABLATION DECOMPOSITION

What a centred ablation can see, and a null that survives the question. The contrast above is the mean of a per-row response to a displacement $s_i = u^\top(h_i - \mu)$ that is *centred* on a global mean rather than on the evaluation rows. The intervention leaves a row with $s_i = 0$ untouched, so the response passes through the origin — we confirm this empirically, the 100 rows of smallest $|s_i|$ showing changes at numerical-noise scale and none at all in top-1 move or accuracy — and we fit $y_i = \beta_1 s_i + \beta_2 s_i^2$ through it. The response is then *dominantly linear*: at one standard deviation of s the linear term is about five times the quadratic one. Yet because $\mathbb{E}[s]$ is near zero while $\mathbb{E}[s^2]$ is not, that linear term supplies only 8% of the contrast in this common-coefficient fit across rows, against 131% from the quadratic term and a -39% residual no quadratic can represent (each term's share is its mean contribution over rows divided by the mean change). In this fit, a mean-replacement contrast credits little of a response that is mostly slope, a property of the intervention. It sharpens rather than contradicts the standard guidance on interpreting these experiments (Heimersheim & Nanda, 2024), and the nearest published result we are aware of concerns how interaction effects scale with the distance between clean and patched activations (Vaidyanathan et al., 2026): a different decomposition of the same underlying fact, that what such an experiment reports depends on *how far* it moves the input. We therefore measure the slope directly rather than inferring it. Regressing each row's ΔNLL on its own s_i: the difficulty direction's slope is -0.0448, the **10th percentile** of the 50 matched controls ($z = -1.39$), while the policy-Jacobian direction is at the 0th ($z = -16.85$). Decomposing into both terms leaves the difficulty direction ordinary on each (β_1: 10th percentile; β_2: 12th) and the policy-Jacobian direction extreme on the slope. Nor is the slope's variation across difficulty quintiles itself distinctive: a heterogeneity χ^2 against the direction's own pooled slope puts the difficulty direction at the 86th percentile of the same 50 controls, so the roughly threefold spread in its per-quintile slope is what matched random directions also show. The null holds on every NLL statistic we report from this intervention, and the positive control is detected on two of them.

A.8 THE DOSE GRADIENT

The apparent difficulty gradient is the dose gradient. Splitting the same rows into quintiles of puzzle rating makes the difficulty direction's effect on the solution move's NLL look strongly graded — $+0.00268$ nats on the easiest quintile against -0.00260 on the hardest, the 100th and 0th percentile of each quintile's own null, with the controls flat throughout. This is not evidence of a difficulty-dependent response, and we report it because the pattern is easy to misread. The displacement s_i is itself graded by rating: the probe predicts rating, so $\mathrm{corr}(s_i, \mathrm{Rating}) = 0.579$ for the difficulty direction against 0.014–0.176 in absolute value for the controls, and it changes sign near the middle quintile — which is where the effect changes sign. In this fit, a centred ablation along a direction correlated with the splitting variable can produce this pattern, and the per-unit-dose response that would distinguish a real difficulty dependence from it is ordinary (10th percentile, Appendix A.7). The split was not pre-specified; the written specification fixed the full-sample contrast and the control families only.

A.9 THE SECOND ARCHITECTURE

The null holds on a second architecture. We repeated the entire design on **Maia-3 (79M)**, a transformer rather than a CNN, at its own best layer (`transformer_block_7`, selected, as throughout, for decodability, not for where the policy reads it), on the same $n = 26{,}352$ rows with the same probe, folds, control families and rank. The difficulty direction is again indistinguishable from its 50 matched controls on the solution move's NLL — the 64th percentile, against the 24th on Maia-2 — while the policy-Jacobian positive control is detected at **159 times the null's**

own standard deviation on signed policy entropy on this network. The positive control moves; this direction does not. The intervention's effect on the difficulty direction is roughly thirty times smaller here than on Maia-2 (-1.6×10^{-5} against -5.2×10^{-4} nats), so the network is less sensitive to a rank-1 intervention overall, but the separation between the positive control and the control cloud is wider, not narrower, than on Maia-2. The null holds on a CNN and on a transformer, with the positive control detected on both. This run also carried two one-sided tests written down before it was read: that the difficulty direction's effect on the solution move's NLL would fall with puzzle rating more steeply than every control's, and that its effect on policy entropy would exceed every control's. Both failed; the quantities above are that run's other outputs, reported descriptively.

A.10 Fine-tuning the readout on human difficulty

Fine-tuning the readout on human difficulty leaves the direction ordinary. We ran this test after the fact, as an exploratory check of the prediction stated at the end of Section 7, not a pre-specified one. From the same Maia-2 checkpoint we froze the whole convolutional trunk, through `cnn_block_4` and the convolution after it, so the probed representation and the difficulty direction are unchanged (500 checked rows bit-identical to the base model's), and trained only the transformer blocks and output heads. The fine-tuning data are 300,000 Lichess puzzles (290,000 for training and 10,000 for validation) with at least 100 recorded attempts each, disjoint by puzzle and by source game from the 26,352 evaluation rows. For each puzzle a player rating is sampled, and the policy's probability of the solution move is trained toward the Glicko expected solve rate of a player of that rating on that puzzle, plus a KL term to the base policy that keeps the tuned policy close to it. Arm B is identical except that Rating is shuffled across puzzles; both arms are evaluated after the same four epochs, with the same row order and sampled ratings. On the evaluation rows, where the target uses each row's own player rating, the correlation of the policy's solution probability with it (the policy's difficulty correlation) rises from 0.37 (base) to 0.50 (arm A) and 0.42 (arm B); arm B rises too because the target also depends on the player's rating, which is learnable without Rating. The rank-1 ablation of Section 7, with the same direction, rows and 50 matched controls, leaves the difficulty direction ordinary in every arm: on the solution move's NLL at the 24th, 16th and 20th percentile (base, A, B), and on the per-row slope at the 10th, 14th and 34th, none beyond the 5th or 95th. **Three caveats bound this.** First, the tuned policy has drifted: $\mathrm{KL}(\text{base} \,\|\, \text{A}) = 0.090$ nats on the evaluation rows, about 15 times the largest rank-1 ablation KL on this model (the policy-Jacobian positive control's, 0.00599 nats), so the test concerns a model some distance from the Maia-2 studied elsewhere. Second, the positive control is the base model's policy-Jacobian direction, reused rather than recomputed for the tuned arms; it still registers there, but the tuned arms' own policy-Jacobian directions were not computed, so power on the tuned arms is assumed, not shown. Third, the trunk is frozen, as in the design of Liao & Cao (2026), whose readout is a small head trained on frozen features; so this tests whether a readout trained on human difficulty recruits the existing direction, not whether training would change the representation itself. Unlike their small head, the module trained here (the transformer blocks and output heads) is large and may use other features.

A.11 The effort direction in full

The effort direction's effect differs by metric. The effort ablation was run on Maia-2 only. Applying the identical intervention to the *effort* direction of Section 6 — same layer, rank, folds, metrics and 50-direction control family, on the rows of the thinking-time analysis — separates the metrics sharply. That run's written specification made the solution move's NLL its primary outcome, null by its own rule; KL and top-1 flip rate were two of the three secondary outcomes it listed (the third, Δtop-1 accuracy, did not clear the specification's percentile threshold and is not reported), which no decision rule covered, and the rating grading is exploratory (the specification counted it as a finding only if the primary cleared its null, and it was not dose-checked by quintile). On the solution move's NLL it behaves like the difficulty direction: 14th percentile of its own matched null on the full sample (-0.00079 nats), and the same sign-flipping rating gradient underneath (-0.00659 per 1000 Elo, 0th percentile, $z = -10.8$). But on those two, KL and top-1 flip rate, it lies above all 50 of its controls — KL $+0.00090$ against a null mean of $+0.000063$, top-1 flip rate 1.04% against 0.275% — and, unlike the difficulty direction, *these are graded too* in that exploratory grading: effort's KL and flip-rate slopes on rating both lie above every control slope, where the difficulty direction's are ordinary draws (32nd and 62nd, $|z| < 0.5$). Per quintile the flip-rate excess over the matched null rises monotonically from $2.42\times$ on the easiest puzzles to $4.50\times$ on the hardest. Ablating the

effort direction reshuffles where the model puts its probability mass, more so the harder the puzzle in the exploratory, not dose-checked, grading, without the solution move itself being favoured or disfavoured relative to its matched controls.

Two confounds were checked before reporting this rather than after. The effort ablation's full-sample dose is mildly elevated — relative dose 0.0313 against a control mean of 0.0296 (SD 0.00089), the 98th percentile, about 1.9 SD — a small elevation; the output effects lie 9–19 SD above their own null, though the two SDs describe different quantities. And the effort direction overlaps the model's own policy-entropy direction (mean rank-1 cosine 0.517 over five folds, range 0.477–0.546), but, under the assumption of additive effects, subtracting that direction's own intervention leaves effort's KL excess intact ($+0.00061$, CI $[+0.00057, +0.00065]$), and a linear overlap estimate attributes at most $\approx 9\%$ of the magnitude to it.

The two directions are not equivalent, and we do not present them as such. Effort's probe explains roughly half as much variance ($R^2 = 0.1427$ against 0.2995), so the direction it identifies is noisier, and its controls' 97.5th percentile is 13–15% of its own run's policy-Jacobian effect (difficulty's: 16–18%; the denominators differ). The geometry differs: effort sits at $89.87°$ to the rank-1 policy-Jacobian direction — essentially orthogonal — where the difficulty direction sits at $82.90°$ on the same rows. (Both angles are from this run, on the $26{,}064$ rows with usable centisecond clocks, where the difficulty direction is refitted; the $82.59°$ of Section 7 is the original run's, on all $26{,}352$ rows.) *A direction orthogonal to the top policy-Jacobian direction nonetheless moves the output distribution 19 null standard deviations above the matched controls' mean, beyond every one of them*, which is the clearest evidence in this paper that rank-1 Jacobian alignment is a weak proxy for whether a representation is read.

A.12 Competence inside an architecture-matched family

Competence is confounded with architecture, and the one design that separates them is null. In the seven-model roster, competence co-varies with architecture family, capacity and training-objective breadth; the seven-model correlation of Section 5 cannot attribute the ordering to any one of them. We therefore tested competence inside a family where the others are held fixed: the nine Maia-1 networks (nominal rating bands 1100–1900), verified identical in architecture, size, input format and policy vocabulary, and matched the native engine's top-1 move on $180/180$ test positions. **Within that family, first-move accuracy (how often a model's top move is the puzzle's first solution move) does not predict difficulty-encoding**: Pearson $r = -0.3117$ at the argmax layer (exact 9! permutation $p = 0.414$), and $r = 0.076$ ($p = 0.84$) with the readout layer held fixed for all nine. The manipulation is itself weak — accuracy spans only 0.566–0.601 against the roster's 0.675–0.961, and is flat-to-inverted in the training band ($r = -0.479$), because these networks are trained to imitate the move a player of a given rating would make rather than to play well — so power is low. But the variation that exists is real rather than noise: across the nine networks the spread exceeds sampling error for both accuracy (SD $= 0.0108$ vs binomial SE $= 0.0030$) and ΔR^2 (SD $= 0.0033$ vs mean CI half-width 0.0030). **We therefore report the seven-model association as an association, and do not claim competence is the operative variable: an architecture-matched test of that reading returns a null.** We do not pool the nine networks with the seven: their readout width is 64 against the roster's 256–2048, making them simultaneously the narrowest-readout and lowest-ΔR^2 points in any pooled set, so a pooled correlation would be confounded with probe width and would in any case reflect the location of one added cluster rather than nine independent observations.

A.13 Cross-model agreement by pair group

Cross-model transfer. Removing the shared target and the maximal control block — the 23 engine and puzzle features of Section 3 plus the model's own policy entropy — via a nonlinear (GBM) fit (tier T3), then further removing a linear fit of raw 737-column piece-square board occupancy (tier T4), leaves all $21/21$ pairs across the seven-model roster in positive residual agreement with every fixed-OOF cluster-bootstrap CI excluding zero, $n = 26{,}352$: mean pairwise correlation 0.410 (T3) falling to 0.390 (T4). For the convergence analysis only, Leela-CF's three missing policy-entropy values were replaced by their mean, keeping all $26{,}352$ rows; the difficulty analysis drops those rows ($26{,}349$). Agreement is ordered by relatedness, not flat: models of the same lineage

(4 pairs, $0.551 \rightarrow 0.528$) > Maia-2 with Maia-3, one project across two architectures (3 pairs, $0.463 \rightarrow 0.438$) > human-imitation models from different projects (8 pairs, $0.422 \rightarrow 0.404$) > the engine-trained `leela_cf` with each human-imitation model (6 pairs, $0.274 \rightarrow 0.257$; before→after the board removal throughout). The three `leela_cf`–Maia-3 pairs share the Chessformer architecture (Section 3) yet fall in the lowest group, each below every pair of human-imitation models at both tiers. We call the result residual readout agreement after the two-step adjustment (a GBM fit, then a linear fit of board occupancy); neither step guarantees that all target, control or board information is gone. An asymmetry sits inside this: averaged over its six pairings, `leela_cf` as the source activation space reaches normalized transfer R^2 of 0.350 (before the board removal) / 0.288 (after), against 0.146 / 0.055 as the target being predicted from a human-imitation space — ratios of group means 2.40 (before) and 5.24 (after). This is a ratio of means, not a per-pair statement: individual per-pair ratios range from about 1.8 to over 280 before the board removal (the `allie_qwen` pairing's reverse-direction transfer is 0.001), and at T4 two of the six reverse-direction values are themselves negative (`allie` -0.040, `allie_qwen` -0.162), so no single per-pair ratio should be quoted in its place. The direction is consistent throughout: transfer from the engine-trained representation to the human-imitation ones works better than the reverse. The per-pair correlations and intervals behind these group means, for all 21 pairs at both tiers, are supplied in `code/tables/5-convergence/xm2_HEADLINE_TABLE.csv` in the supplementary material (see the code README); we do not tabulate them here.

A.14 Limitations in full

Section 9 names these limits; they are stated here in full.

- **Single domain, and what portability would require.** All results come from one domain: chess policies evaluated on puzzle positions. Another domain would need: a population-calibrated item difficulty independent of any model under study; a strong engine as a complexity control; recorded human decisions with their times; and a policy whose own output is measurable on the same items. Chess satisfies all four at once, and we do not know how common that conjunction is elsewhere. One part of the pattern already has independent support outside chess: information that is decodable and convergent yet not shown to be used under ablation is also reported in language models (Usama & Chang, 2026, Section 2) — evidence for that pattern, on a target unrelated to human difficulty. That separates two questions this paper cannot settle: whether decodable-but-not-shown-to-be-used is an artefact of our design (the outside evidence makes this less likely) from whether these controlled findings hold elsewhere, which remains open.
- **A population asymmetry, verified rather than assumed.** The timed player is the original game's mover — `solver_elo` (the original mover's rating) matches that player's own rating exactly in 20 of 20 sampled puzzles — while Rating comes from a later, anonymous puzzle-solving crowd under different incentives.
- **Transfer to the endgame partition is not shown.** Held out by Lichess tag, its increment is -0.0336 (full-scale, full-OOF-refit CI $[-0.105, +0.0401]$, removing all $n = 11{,}858$ endgame-tagged puzzles from training) — yet both partitions carry signal fitted within themselves: $\Delta R^2 = 0.0483$ (fixed-OOF $[0.0419, 0.0541]$, $n = 14{,}494$ non-endgame) and 0.0461 ($[0.0389, 0.0530]$, $n = 11{,}858$ endgame), against the full-set headline of 0.0436. Thirty matched-n random subsamples place the non-endgame value 1.7 SD above a null mean of 0.0431; transfer is not demonstrated at the phase boundary, though each partition carries signal, and the interval does not exclude a modest effect.
- **Training objectives.** Six of the seven imitate rated human players and none has a difficulty target; Allie's objective includes a time-prediction loss (Zhang et al., 2025), so its timing readout is not evidence of emergence without supervision; Maia-2 and Leela-CF carry that evidence. The untrained baseline of Table 1 uses one random initialisation per seed (two seeds) and no training, probed at the trained model's best layer only; no paired trained-minus-baseline test was run.
- **What the ablation removes.** Mean replacement along one direction need not remove the information, which may stay decodable along correlated directions; a refitted probe recovers most of the signal (Appendix A.5). Nor can a rank-1 linear edit detect use that is non-linear in the activations.

- **The depth comparison.** Its roster, the models run with engine controls at depths 12, 15 and 20, is not Table 1's: it lacks `allie_qwen` and includes ChessBench-BC-9M (engine-only controls, no entropy needed). The engine-only decline across depths is reported without an interval. From depth 12 to 20, `leela_cf`'s lead over Maia-3-79M in probe R^2 on the engine-only residual also grows, by $+0.0118$ $[+0.0059, +0.0182]$.
- **Replications.** The held-out months recompute the probe and the controls on each month's own activations and engine block, at an 18-feature control tier (those months were searched at depth 12 only), and are read against an 18-feature bridge on 2019-01 (Maia-2 0.0457, Maia-3-79M 0.1112): Maia-2 gives 0.0624 and 0.0324 and Maia-3-79M 0.1260 and 0.0794 on 2020-01 and 2022-01, so the presence and sign replicate and the magnitude does not.
- **The effect is modest.** For the thinking-time increment over the rating, $\Delta R^2 \approx 0.008$–0.009 against a total explained variance near 0.37–0.43, stable across interval conventions, models and held-out months, and across clock resolutions for Maia-2; it is larger for `leela_cf` at whole seconds (Section 6; Appendix A.4).
- **Multiplicity.** Forty leave-one-theme-out tests ran and no correction column exists in any primary table; a Holm threshold over the family lies below both bootstrap resolution floors and rejects nothing regardless of the truth. Under Benjamini–Hochberg at $q = 0.05$ (fixed-OOF convention), 21 of 40 survive, 19 positive and 2 negative, and the family mean over the nine powered tactical motifs is 0.0867, paired full-OOF-refit CI $[0.0708, 0.1011]$, excluding zero; two motifs, `discoveredAttack` and `defensiveMove`, do not clear zero, and we name them. No family-wise correction spans the choices of layer, control tier, interval convention and secondary Section 7 statistic.

A.15 Concurrent submissions and shared data

One data tier is shared with a concurrent submission to this conference that asks a different question: it estimates a per-person representation of speed and accuracy (a person vector over a frozen human-like reference policy) from calibrated puzzle decisions, tests its reliability over time, its held-out predictive value and its amortized estimation, and instruments the effect of think time on accuracy. Both papers draw on the public Lichess puzzle archive, its source games and the Maia-2 policies, and both use the January 2019 archive: it is that submission's development tier and this paper's headline month. No data split, trained model, estimate, figure, table or passage of text is shared, and no quantity here is taken from that submission. Two further manuscripts by the authors are under review at other venues on related behavioural questions; neither is supplied, and nothing here depends on either.

B Recent history and human errors: methods and full tables

This appendix specifies the analysis of Section 8 and gives every per-month value behind it. All quantities are in nats (changes in log-probability) or AUC; none is comparable in scale to a ΔR^2.

B.1 Data

Each puzzle is a position from a real Lichess game; the *player* is that game's mover at the position, and the puzzle is *missed* when the move they actually played in the game is not the puzzle's first solution move (*solved* otherwise: first-move agreement, not completion of the sequence). Puzzles without a recorded outcome are dropped. The headline month, 2019-01, is the puzzle set of Section 3; there the player's move was reconstructed from the game record, and it agrees with the recorded outcome on all $26{,}165$ puzzles that have one. The three later months carry the player's move directly: 2020-01 ($n = 21{,}508$), 2022-01 (a $25{,}000$-puzzle random subsample, seed 20260924, of the $129{,}388$ used in Sections 4 and 6) and 2024-06 ($n = 9{,}909$). For them the game history is rebuilt by replaying each game from the start to the puzzle ply; the replay reaches the puzzle position on every row of every month.

B.2 CONDITIONS AND TARGET

Maia-3 takes the current board and the seven boards before it, oldest first; when fewer real boards are given it pads the front by repeating the oldest one, which is its own convention. With f the puzzle position, f^- the position before the opponent's last move and h the up-to-six real boards before f^-: (a) $[f]$, Maia-3's input everywhere else in this paper; (b) $[h, f^-, f]$, the real history; (c) $[f^-, f]$, which shows the opponent's last move and nothing earlier; (d) $[h', f'^-, f]$, the history of a different game, matched on side to move and on history length, never from the same game (fixed seed 1234). Matching side to move keeps the alternation of board orientations that real games produce, so b and d differ in content, not in geometry. The clock inputs are zero in all four conditions; the released Maia-3 models do not read clock time. Condition (a) reproduces the stored Maia-3 outputs used elsewhere in this paper to within 10^{-3} on a 300-row check. For Maia-3 and Leela-CF, condition (a) is also the matched input for every Rating analysis in this paper, because a puzzle attempter sees the position, not the game that led to it. Allie is not used here: it reads the position only through the move sequence, so condition (a) does not exist for it. The target is $\log p$ of the move the player actually made; Δ_x is its mean change from (a) over missed (or solved) puzzles.

B.3 INTERVALS, AND HOW THE EFFECT WAS FOUND

Every interval is a 95% percentile bootstrap over puzzles with 1,000 draws (seed 20260924). The contrasts $\Delta_b - \Delta_c$ and $\Delta_b - \Delta_d$ are row-paired: the common term $\log p_a$ cancels, so each draw resamples the same puzzles for both conditions. Missed and solved puzzles are disjoint, so the missed-minus-solved difference resamples each stratum independently (the written criterion said paired). The effect was found on 2019-01 while testing a different hypothesis — that history makes puzzles easier for the network, which the data reversed — by measuring the probability of the player's own move alongside that of the solution. Before any later month was scored, the analysis code was frozen and four pass criteria were written into a design document: $\Delta_b > 0$; $\Delta_b - \Delta_d > 0$; $\Delta_b - \Delta_c > 0$ in at least two of three sizes; and Δ_b larger on missed than on solved puzzles, each with its interval excluding zero. This is a replication against pre-specified criteria, not a pre-registration: the document is dated and internal, not a registry entry. Every criterion passed at every size in every month: 36 of 36 checks on the three later months (the third criterion held at every size, so it is counted per size), and 48 of 48 counting 2019-01, where the effect was found (Tables 3 and 4). The replication script imports the bootstrap of the original 2019-01 analysis rather than reimplementing it and reproduces that analysis's 2019-01 point estimates exactly.

B.4 THE LEELA-CF INPUT

Leela-CF (lc0 network BT4-1024x15x32h-swa-6147500) is scored through lc0's own encoder, so its history enters as lc0's stacked input planes for the current and seven previous positions rather than as a token sequence. (a) The position alone, with lc0's convention of repeating it into the history planes — exactly the input behind Leela-CF's results elsewhere in this paper; (b) the game replayed from the start position to the puzzle ply, so the encoder fills the history planes with the real previous positions; (c) the position before the opponent's last move, plus that move. All three reach the identical current position and legal moves, and the auxiliary planes (castling rights, en passant, the fifty-move counter) were identical between (a) and (b) on a 50-row check, so the conditions differ only in history. Condition (a) reproduces the stored Leela-CF outputs used elsewhere in this paper to within 10^{-3} (native lc0, 40 rows). The shuffled condition (d) was not run: lc0 builds its history from a legal move sequence, and another game's moves do not lead to this position. Leela-CF was scored on 2019-01 and 2024-06 (Table 5).

B.5 FAILURE PREDICTION

The outcome is whether the player's game move matched the puzzle's first solution move. The base features are the puzzle's Rating, the player's rating, the log remaining clock, the base time, the increment and the ply; +A adds Maia-3's log-probability of the solution and its policy entropy under condition (a); +B replaces them with the same two under condition (b). Predictions are out of fold over 5 folds (shuffled, seed 20260924; one puzzle per game, so this is also a split by game), from a logistic regression standardised within each training fold and from a histogram gradient-boosted classifier (200 iterations, no early stopping). ΔAUC is a paired bootstrap over the out-

of-fold predictions, both feature sets on the same resampled puzzles, $1{,}000$ draws. Because the position is fixed between +A and +B, ΔAUC(+B$-$+A) isolates what the game so far adds; +A over base mixes that instance-level signal with difficulty the policy reads from the board and Rating misses (Tables 6 and 7).

B.6 Error types, deliberation and history length

On missed puzzles each wrong move receives the first matching label in this order: *recapture*, landing on the square the opponent's last move landed on (whether or not that move captured); *continuation*, moving the piece the player's own previous move landed with; *check*; *capture* (en passant included); *other*. The same classification of the correct moves on solved puzzles gives a base rate. Deliberation is the player's own centisecond thinking time at the position, as in Section 6; tertiles are cut within each month's missed puzzles. The regression of Table 9 is least squares of each missed puzzle's gain on log thinking time, Rating, the player's rating and the log remaining clock, with and without indicators for the error type (reference: other); its intervals come from $1{,}000$ row-resample refits. In the history-length analysis, condition b_k gives Maia-3 the k most recent real boards before the position; b_1 and b_7 are, board for board, conditions (c) and (b), and their outputs match those conditions exactly. The share of the gap, $(\Delta_{b_k} - \Delta_{b_1})/(\Delta_{b_7} - \Delta_{b_1})$, has a paired bootstrap interval in which numerator and denominator are computed on the same resample. The history-length analysis was run on 2019-01 only, for the 23M and 79M models (Table 10).

B.7 Reading the tables

Four points qualify Section 8. First, 2024-06 gives the largest gains (Maia-3-79M 0.165 $[0.143, 0.186]$ against 0.132 $[0.121, 0.144]$ on 2019-01) on the smallest sample, and it is the only month inside Maia-3-79M's reported training window (January 2023 – July 2025), so its games may overlap training data; this applies equally to its failure-prediction and error-type values, and every finding holds on the three months that predate that window. Second, the error-type account is partial: the residual *other* type is the largest (Table 8) and its gain is statistically indistinguishable from recapture's in every month; the capture gain's interval includes zero in every month; and the check gain's interval, which includes zero on 2019-01 and 2020-01, excludes it on 2022-01 and 2024-06, so "no gain on checks" is not a finding we report. Third, in the type-controlled regression the continuation-versus-other coefficient excludes zero in three months but not on 2024-06, the smallest sample; the thinking-time coefficient excludes zero in all four. Fourth, the gain could reflect more mass on non-solution moves in general rather than on this move in particular (Table 5 shows the solution move losing probability as the player's move gains); a ranking test among non-solution moves would separate the two.

B.8 Relation to prior work

Conditioning on a game's recent moves improves human move prediction on average: Maia-1 feeds the previous twelve plies (McIlroy-Young et al., 2020), the Chessformer paper, whose human-imitation model is the Maia-3 used here, ablates board history in a 5M-parameter model and finds it raises move-matching accuracy (Monroe et al., 2026), and Otter, a history- and time-conditioned human model, reports a within-model history gain of the same kind while noting that it systematically under-predicts blunders (Tarun Kumar S, 2026); Maia-2, by contrast, deliberately removed that history (Tang et al., 2024). Other recent work looks at human errors in aggregate: ChessMimic models human moves, clocks and outcomes per rating band and reports how blunder rates fall with remaining clock across the population (Johnson, 2026), and a study of searchless chess transformers measures how often a model blunders where humans do, and finds that stripping history flips some of its decisions (Li & Jiang, 2026). Two results are closer still. Maia-1 also predicts, for a given player and position, whether the next move will be a blunder — from the current position, without move history — and its authors report that reusing their history-aware move-prediction model for that task was far exceeded by specialised models (McIlroy-Young et al., 2020). And fine-tuning Maia on a player's own past games yields a model that gives that player's moves more probability than the population model does, on optimal moves, errors and blunders alike (McIlroy-Young et al., 2022): individual conditioning raising the probability of an individual's own moves, but through long-run fine-tuning rather than the game in progress, not specifically on errors, and without puz-

zles, shuffled or last-move controls, or a difficulty covariate. We have not found a report of in-game history raising the probability of the specific mistake a player makes, under those controls.

B.9 TABLES

Month	Maia-3	n missed	Δ_b real history	Δ_c last move only	Δ_d shuffled
2019-01	5M	7,053	0.115 [0.104, 0.125]	0.007 [−0.001, 0.015]	−0.103 [−0.119, −0.088]
	23M	7,053	0.129 [0.118, 0.140]	0.068 [0.058, 0.076]	−0.065 [−0.079, −0.052]
	79M	7,053	0.132 [0.121, 0.144]	0.090 [0.080, 0.099]	−0.032 [−0.045, −0.019]
2020-01	5M	6,514	0.114 [0.103, 0.126]	0.003 [−0.006, 0.012]	−0.097 [−0.112, −0.081]
	23M	6,514	0.126 [0.113, 0.139]	0.064 [0.055, 0.074]	−0.064 [−0.077, −0.050]
	79M	6,514	0.136 [0.123, 0.149]	0.086 [0.077, 0.096]	−0.023 [−0.037, −0.009]
2022-01	5M	8,574	0.111 [0.101, 0.120]	0.006 [−0.001, 0.014]	−0.121 [−0.137, −0.106]
	23M	8,574	0.119 [0.108, 0.129]	0.064 [0.055, 0.072]	−0.087 [−0.100, −0.074]
	79M	8,574	0.127 [0.117, 0.138]	0.078 [0.070, 0.086]	−0.035 [−0.047, −0.024]
2024-06	5M	2,515	0.113 [0.095, 0.132]	0.025 [0.010, 0.040]	−0.109 [−0.140, −0.079]
	23M	2,515	0.134 [0.114, 0.154]	0.064 [0.050, 0.079]	−0.064 [−0.088, −0.040]
	79M	2,515	0.165 [0.143, 0.186]	0.101 [0.085, 0.117]	−0.008 [−0.029, 0.015]

Table 3: History gains on missed puzzles: mean change in log-probability (nats) of the move the player actually made, relative to the current board only (condition a), with 95% percentile bootstrap intervals (1,000 row resamples). 2019-01 is the month on which the effect was found; the other three are the frozen-code replication. Real history raises the probability of the player's wrong move at every size in every month.

Month	Maia-3	$\Delta_b - \Delta_c$ (paired)	$\Delta_b - \Delta_d$ (paired)	Δ_b on solved	Δ_b missed − solved
2019-01	5M	0.108 [0.097, 0.119]	0.218 [0.201, 0.235]	0.004 [0.000, 0.008]	0.111 [0.100, 0.122]
	23M	0.061 [0.053, 0.070]	0.194 [0.177, 0.212]	0.006 [0.003, 0.010]	0.123 [0.111, 0.134]
	79M	0.042 [0.034, 0.051]	0.164 [0.148, 0.180]	0.006 [0.002, 0.009]	0.127 [0.116, 0.139]
2020-01	5M	0.111 [0.099, 0.122]	0.211 [0.192, 0.229]	−0.001 [−0.006, 0.003]	0.116 [0.103, 0.128]
	23M	0.062 [0.053, 0.071]	0.190 [0.172, 0.205]	0.002 [−0.002, 0.007]	0.123 [0.109, 0.136]
	79M	0.050 [0.041, 0.058]	0.159 [0.142, 0.176]	0.001 [−0.003, 0.005]	0.135 [0.123, 0.148]
2022-01	5M	0.104 [0.094, 0.114]	0.232 [0.216, 0.249]	0.007 [0.003, 0.011]	0.104 [0.094, 0.115]
	23M	0.055 [0.047, 0.063]	0.206 [0.189, 0.221]	0.006 [0.002, 0.009]	0.113 [0.101, 0.124]
	79M	0.049 [0.041, 0.057]	0.163 [0.148, 0.178]	0.003 [0.000, 0.007]	0.124 [0.113, 0.136]
2024-06	5M	0.088 [0.069, 0.108]	0.223 [0.192, 0.258]	0.010 [0.005, 0.016]	0.103 [0.084, 0.122]
	23M	0.070 [0.053, 0.087]	0.198 [0.168, 0.228]	0.010 [0.005, 0.015]	0.124 [0.102, 0.144]
	79M	0.063 [0.049, 0.079]	0.172 [0.145, 0.203]	0.003 [−0.002, 0.008]	0.162 [0.140, 0.184]

Table 4: The four replication criteria, per month and size (nats, 95% intervals). $\Delta_b - \Delta_c$ and $\Delta_b - \Delta_d$ are row-paired on the missed puzzles; the missed-minus-solved difference resamples the two disjoint strata independently. Every interval in the first two and the last column excludes zero; so does Δ_b on missed puzzles (Table 3). Every criterion passes at every size in every month.

	real history (b)		last move only (c)	
Model	player's move	solution	player's move	solution
2019-01, missed puzzles, $n = 7{,}053$				
Maia-3 5M	0.115 $[0.104, 0.125]$	-0.130 $[-0.141, -0.120]$	0.007 $[-0.001, 0.015]$	-0.040 $[-0.048, -0.031]$
Maia-3 23M	0.129 $[0.118, 0.140]$	-0.142 $[-0.153, -0.131]$	0.068 $[0.058, 0.076]$	-0.062 $[-0.070, -0.054]$
Maia-3 79M	0.132 $[0.121, 0.144]$	-0.158 $[-0.168, -0.147]$	0.090 $[0.080, 0.099]$	-0.090 $[-0.097, -0.083]$
Leela-CF	-0.032 $[-0.043, -0.022]$	0.015 $[0.010, 0.019]$	-0.038 $[-0.048, -0.027]$	0.014 $[0.009, 0.018]$
2024-06, missed puzzles, $n = 2{,}515$				
Maia-3 5M	0.113 $[0.095, 0.132]$	-0.120 $[-0.139, -0.102]$	0.025 $[0.010, 0.040]$	-0.023 $[-0.039, -0.009]$
Maia-3 23M	0.134 $[0.114, 0.154]$	-0.148 $[-0.167, -0.131]$	0.064 $[0.050, 0.079]$	-0.073 $[-0.088, -0.060]$
Maia-3 79M	0.165 $[0.143, 0.186]$	-0.151 $[-0.170, -0.133]$	0.101 $[0.085, 0.117]$	-0.089 $[-0.103, -0.077]$
Leela-CF	-0.010 $[-0.030, 0.008]$	0.009 $[0.002, 0.016]$	-0.029 $[-0.046, -0.011]$	0.011 $[0.004, 0.017]$

Table 5: Human-imitation versus engine-trained policy on the same missed puzzles and the same real history (nats, 95% intervals). "Player's move" is the change in log-probability of the wrong move the player made, relative to the current board alone (condition a); "solution" is the same change for the puzzle's solution move. Leela-CF was scored on 2019-01 and 2024-06 only, and not under the shuffled condition d (Appendix B.4). Given the same history, Leela-CF moves the other way on 2019-01 and is indistinguishable from zero on 2024-06, while the human-imitation policy's gain is positive in both months.

Month	Classifier	n	base	+A	+B	ΔAUC(+A$-$base)	ΔAUC(+B$-$+A)
2019-01	logistic	26,165	0.683	0.798	0.824	0.115 $[0.109, 0.121]$	0.027 $[0.024, 0.029]$
	GBM	26,165	0.672	0.801	0.826	0.128 $[0.122, 0.135]$	0.025 $[0.021, 0.028]$
2020-01	logistic	21,508	0.664	0.795	0.822	0.131 $[0.124, 0.139]$	0.027 $[0.025, 0.030]$
	GBM	21,508	0.650	0.790	0.819	0.141 $[0.133, 0.148]$	0.029 $[0.025, 0.032]$
2022-01	logistic	25,000	0.678	0.809	0.832	0.132 $[0.125, 0.137]$	0.023 $[0.021, 0.025]$
	GBM	25,000	0.663	0.809	0.833	0.146 $[0.140, 0.153]$	0.023 $[0.020, 0.026]$
2024-06	logistic	9,909	0.693	0.813	0.842	0.121 $[0.110, 0.130]$	0.029 $[0.025, 0.033]$
	GBM	9,909	0.661	0.807	0.836	0.146 $[0.135, 0.157]$	0.029 $[0.023, 0.035]$

Table 6: Predicting whether the player's game move matches the puzzle's first solution move: out-of-fold AUC (5-fold) and paired bootstrap ΔAUC (1,000 draws, 95% intervals). base: puzzle Rating, the player's rating, remaining clock, base time, increment and ply. +A adds Maia-3-79M's log-probability of the solution and its policy entropy computed from the current board only; +B the same two quantities computed with the real history. GBM is a histogram gradient-boosted classifier. 2022-01 is a seeded $25{,}000$-puzzle subsample of that month. History (+B over +A) raises the AUC in every month with both classifiers.

	Maia-3-5M		Maia-3-23M	
Month	logistic	GBM	logistic	GBM
2019-01	0.022 $[0.020, 0.024]$	0.027 $[0.024, 0.030]$	0.026 $[0.023, 0.028]$	0.026 $[0.022, 0.029]$
2020-01	0.021 $[0.018, 0.023]$	0.023 $[0.019, 0.026]$	0.026 $[0.023, 0.029]$	0.028 $[0.024, 0.031]$
2022-01	0.020 $[0.018, 0.023]$	0.023 $[0.020, 0.026]$	0.023 $[0.021, 0.025]$	0.024 $[0.021, 0.027]$
2024-06	0.021 $[0.018, 0.024]$	0.021 $[0.015, 0.027]$	0.026 $[0.022, 0.030]$	0.029 $[0.024, 0.035]$

Table 7: ΔAUC(+B$-$+A) at the two smaller Maia-3 sizes (same design as Table 6). Every interval excludes zero: history helps at the smaller sizes too.

Month	Type of wrong move	n	share of misses	share of solutions	Δ_b (79M)
2019-01	recapture	891	12.6%	16.4%	0.144 [0.116, 0.172]
	continuation	1,802	25.5%	27.0%	0.196 [0.173, 0.222]
	check	554	7.9%	38.8%	0.023 [−0.009, 0.055]
	capture	514	7.3%	5.8%	−0.021 [−0.060, 0.018]
	other	3,292	46.7%	12.0%	0.137 [0.119, 0.156]
2020-01	recapture	793	12.2%	17.8%	0.125 [0.097, 0.154]
	continuation	1,605	24.6%	26.9%	0.196 [0.169, 0.222]
	check	453	7.0%	33.1%	0.035 [−0.001, 0.072]
	capture	464	7.1%	6.9%	−0.034 [−0.073, 0.000]
	other	3,199	49.1%	15.3%	0.148 [0.129, 0.168]
2022-01	recapture	1,005	11.7%	18.1%	0.117 [0.093, 0.142]
	continuation	2,236	26.1%	27.0%	0.178 [0.156, 0.202]
	check	569	6.6%	32.5%	0.044 [0.012, 0.076]
	capture	620	7.2%	6.9%	0.007 [−0.028, 0.045]
	other	4,144	48.3%	15.5%	0.131 [0.116, 0.148]
2024-06	recapture	293	11.7%	14.0%	0.145 [0.086, 0.204]
	continuation	738	29.3%	29.8%	0.205 [0.164, 0.247]
	check	270	10.7%	44.7%	0.082 [0.031, 0.134]
	capture	136	5.4%	3.1%	0.041 [−0.045, 0.128]
	other	1,078	42.9%	8.4%	0.178 [0.144, 0.212]

Table 8: Wrong moves on missed puzzles by type (exclusive labels, first match in the order listed) and the Maia-3-79M history gain within each type (nats, 95% intervals). "Share of solutions" is the same classification applied to the correct moves on solved puzzles, as a base rate. No move was unclassifiable in any month. The residual *other* type is the largest, so the error-type account is partial.

Month	tertile bounds (s)	Δ_b fast	Δ_b middle	Δ_b slow
2019-01	0–5 / 6–15 / 16–393	0.192 [0.171, 0.212]	0.123 [0.104, 0.143]	0.079 [0.060, 0.100]
2020-01	0–6 / 7–16 / 17–473	0.183 [0.161, 0.205]	0.133 [0.111, 0.153]	0.084 [0.064, 0.105]
2022-01	0–3 / 4–9 / 10–594	0.176 [0.157, 0.196]	0.121 [0.103, 0.139]	0.077 [0.059, 0.095]
2024-06	0–3 / 4–8 / 9–219	0.204 [0.172, 0.238]	0.178 [0.135, 0.222]	0.096 [0.059, 0.135]

Month	n missed	$\beta_{\log t}$, no type controls	$\beta_{\log t}$, type-controlled	R^2 (no type / type)
2019-01	7,053	−0.041 [−0.053, −0.028]	−0.040 [−0.053, −0.028]	0.025 / 0.037
2020-01	6,514	−0.038 [−0.051, −0.024]	−0.039 [−0.052, −0.026]	0.019 / 0.031
2022-01	8,574	−0.041 [−0.054, −0.029]	−0.041 [−0.054, −0.029]	0.013 / 0.020
2024-06	2,515	−0.032 [−0.057, −0.006]	−0.034 [−0.058, −0.010]	0.023 / 0.029

Table 9: Deliberation. Top: Maia-3-79M history gain on missed puzzles by tertile of the player's own thinking time at the position (nats, 95% intervals). Bottom: row-level least squares of the per-puzzle gain on log thinking time, controlling for Rating, the player's rating and remaining clock, without and with indicators for the error type of Table 8; coefficient in nats per unit log second, 95% intervals from 1,000 row-resample refits. The gain is largest for the fastest third of errors.

Maia-3	k	Δ_{b_k}	step from $k-1$ (paired)	share of the $c \to b$ gap
23M	1 ($= c$)	0.068 $[0.059, 0.077]$		
	2	0.119 $[0.108, 0.129]$	0.0510 $[0.0418, 0.0592]$	83.3% $[77.1, 89.5]$
	3	0.116 $[0.104, 0.127]$	-0.0031 $[-0.0068, 0.0009]$	78.2% $[71.8, 84.4]$
	4	0.124 $[0.113, 0.135]$	0.0082 $[0.0040, 0.0122]$	91.6% $[88.0, 95.2]$
	5	0.123 $[0.111, 0.135]$	-0.0003 $[-0.0029, 0.0022]$	91.0% $[87.1, 94.8]$
	6	0.124 $[0.113, 0.135]$	0.0007 $[-0.0018, 0.0032]$	92.2% $[89.9, 94.2]$
	7 ($= b$)	0.129 $[0.117, 0.140]$	0.0048 $[0.0035, 0.0059]$	
79M	1 ($= c$)	0.090 $[0.081, 0.099]$		
	2	0.123 $[0.111, 0.134]$	0.0324 $[0.0255, 0.0398]$	77.0% $[67.7, 86.5]$
	3	0.123 $[0.111, 0.134]$	0.0002 $[-0.0031, 0.0036]$	77.5% $[68.5, 87.1]$
	4	0.129 $[0.117, 0.141]$	0.0063 $[0.0025, 0.0100]$	92.5% $[86.8, 97.8]$
	5	0.127 $[0.117, 0.140]$	-0.0018 $[-0.0039, 0.0003]$	88.1% $[82.0, 93.6]$
	6	0.130 $[0.119, 0.141]$	0.0026 $[0.0003, 0.0051]$	94.4% $[90.9, 97.6]$
	7 ($= b$)	0.132 $[0.121, 0.144]$	0.0024 $[0.0010, 0.0037]$	

Table 10: History length, 2019-01, missed puzzles ($n = 7{,}053$): the k most recent real boards before the current one. $k = 1$ reproduces condition c and $k = 7$ condition b exactly (point estimates; intervals from a separate bootstrap); $k = 2$ is the first to show the player's own previous move. Share of the gap is $(\Delta_{b_k} - \Delta_{b_1})/(\Delta_{b_7} - \Delta_{b_1})$ with a paired bootstrap interval (numerator and denominator on the same resample). Most of the gap closes at $k = 2$, the first board showing the player's own previous move.