CHECKMATE CP: BIDIRECTIONAL CONFORMAL SCREENING OF BEHAVIORAL SKILL DISCREPANCIES IN ONLINE CHESS

Anonymous authors
Paper under double-blind review

ABSTRACT

Online platforms increasingly rely on behavioral models to estimate user ability and identify unusual patterns of decision-making. A central challenge is determining whether an individual's observed behavior is consistent with their expected skill. We study this problem in online chess, where recorded ratings guide matchmaking and human-policy models provide a way to infer skill from observed decisions. We introduce CheckMate CP, a calibrated framework for measuring behavioral skill discrepancies. It estimates Behavioral Elo from move likelihoods under a frozen human-policy model and compares it with the player's recorded rating. The resulting gap indicates whether an observation window exhibits stronger- or weaker-than-rated play. Conformal calibration converts these gaps into directional screening alerts with finite-sample marginal false-alert control under exchangeability with a reference population. We evaluate CheckMate CP across natural games, controlled behavioral interventions, human-player substitutions, and games with logged assistance. The results distinguish rating-estimation accuracy, behavioral response, and alert detection. In our window-length analysis, longer observation windows improve rating accuracy but can reduce alert sensitivity. Logged assistance can also raise Behavioral Elo without necessarily triggering an alert. On ChessFraud, the neural estimator flags 54.5% of 44 assisted windows, compared with 27.3% for Stockfish top-move matching, at a 5% upward alert level. Both methods are calibrated using public games with the same time control and produce comparable empirical alert rates on held-out public windows. CheckMate CP provides an interpretable framework for prioritizing discrepancies between observed play and expected skill for human review, without inferring intent or establishing rule violations.

1 INTRODUCTION

AlphaGo's match victories (Silver et al., 2016) and AlphaZero's self-play success (Silver et al., 2018) made one goal of game-playing AI unmistakable: win. These systems search for moves that improve the chance of victory. Human-policy models start from a different objective: to explain how people actually play. In the same chess position, a club player and a master may make different choices. Maia and Maia-2 learn how those choices vary with playing strength (McIlroy-Young et al., 2020; Tang et al., 2024). Once a model captures these patterns, we can use a player's moves to estimate the level of human play they express (Kuboki et al., 2025).

This matters in online chess because an account's rating helps determine its opponents. The rating is updated from past results, and matchmaking uses it to pair players of similar expected strength (Herbrich et al., 2006). When recent play is much stronger or weaker than that rating, the match may no longer be balanced. Engine assistance can produce stronger-than-rated play. Deliberate underperformance can lower a rating and lead to easier opponents later, a practice known as *sandbagging* (Lichess, 2026b). Improvement, fatigue, and ordinary variation also change how someone plays. The screening question is whether recent play expresses a level of skill unusually far from the rating used to match that account.

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.

Existing chess-integrity analyses compare moves with engine recommendations, measure engine-assessed move quality, estimate engine-based performance ratings, or classify games using examples of assistance (Regan & Haworth, 2011; Regan et al., 2012; Barnes & Hernandez-Castro, 2015; Linich et al., 2026). Our target is the mismatch between the level of human play expressed in recent moves and the account's recorded rating. This comparison examines both stronger- and weaker-than-rated play without relying on a particular chess engine's signature.

We introduce CheckMate CP, a framework for checking whether play matches an account's rating. Its name plays on *checkmate*, the chess term for a king in check with no legal escape; CP stands for conformal prediction. For each ten-game account window, a fixed Maia-3 model (Monroe et al., 2026) assigns likelihoods to observed moves at different skill levels. Three alternative estimators turn this policy evidence into *Behavioral Elo*, an estimate of expressed skill: Direct likelihood, Rating-calibrated likelihood, and Neural residual. Comparing Behavioral Elo with the account's recorded rating yields a signed discrepancy: positive for stronger-than-rated play and negative for weaker-than-rated play. The remaining question is how unusual that discrepancy is among comparable windows. Standard conformal p-values (Vovk et al., 2005; Bates et al., 2023) determine upward and downward alerts relative to a chosen reference population. For a fixed scorer and exchangeable calibration and new reference windows, the rule controls marginal per-window alert probability for that population. An alert identifies a window for review, not the cause of its discrepancy.

The evaluation covers rating accuracy, behavioral response, and calibrated screening. Natural games measure rating accuracy. Controlled interventions provide known changes in play, while matched controls measure false-alert rates. Human-player substitutions use moves from other real players, and ChessFraud provides games with logged assistance (Linich et al., 2026). In the paired ChessFraud analysis of 13 participants, four assisted games raise Behavioral Elo on average, yet most of these windows do not alert.

Our contributions are:

- A rating-relative screening framework that uses human-policy models to assess the skill expressed in recent play against the rating used for matchmaking.
- Three Behavioral Elo estimators whose signed rating discrepancies support directional, reference-relative alerts through standard conformal calibration.
- An empirical comparison across natural games, controlled interventions, human substitutions, and logged assistance, showing when changes in Behavioral Elo do and do not produce alerts.

2 Related Work

Human Policy Modeling and Skill Estimation. Human-policy models predict the moves people choose at different skill levels. Maia models rating-specific play, and Maia-2 and Maia-3 extend this line of human move modeling (McIlroy-Young et al., 2020; Tang et al., 2024; Monroe et al., 2026). Such models have also been used to study what moves reveal about a player: Kuboki et al. (2025) combine likelihoods from rating-specific Maia policies with other features to estimate playing strength, while Karvonen (2024) finds skill-related representations in a chess language model trained on game sequences. McIlroy-Young et al. (2021) use move patterns to identify individual players. CheckMate CP uses skill-conditioned move probabilities to estimate the skill expressed in a window of play, then compares it with the account's recorded rating.

Chess Integrity Analysis and Behavioral Anomaly Detection. Engine-based analyses estimate playing strength from move quality (Regan & Haworth, 2011; Regan et al., 2012), and work on cheating detection has examined the limits of engine-based evidence (Barnes & Hernandez-Castro, 2015). ChessFraud provides games with logged assistance and evaluates detection signals from engines and human-aligned models (Linich et al., 2026). These studies address both strength estimation and assistance detection. Our score compares human-policy expressed skill with recorded rating, so an upward or downward discrepancy has a direct interpretation relative to the account's expected level. It does not depend on matching play to a particular engine.

Conformal Calibration for Distribution-Free Screening. Conformal inference calibrates a score against exchangeable reference examples (Vovk et al., 2005; Lei et al., 2018); conformal p-values can then be used for outlier testing (Bates et al., 2023). We apply this established calibration to the upward and downward rating discrepancies of fixed ten-game windows. The chosen reference population determines what counts as unusual. Under exchangeability, the alert level applies to a new window from that population. The framework links that behavioral rating gap to a reference-calibrated review decision.

3 METHOD

CheckMate CP asks whether the skill expressed in an account's play agrees with its recorded rating. Each screening decision uses a ten-game *window*. For window i, let E_i be the recorded rating used as the reference and $\widehat{E}_i^{\,b}$ the skill estimated from its moves, which we call *Behavioral Elo*. Here Elo denotes rating points, not the rating-update algorithm; Lichess uses Glicko-2 (Lichess, n.d.). In the main natural analysis, E_i is the median rating recorded within the window; generated windows retain their source window's reference. Figure 1 shows the three steps: estimate Behavioral Elo with a human-policy model, compare it with recorded rating, and calibrate the discrepancy against reference windows. The output is an upward alert, a downward alert, or no alert.

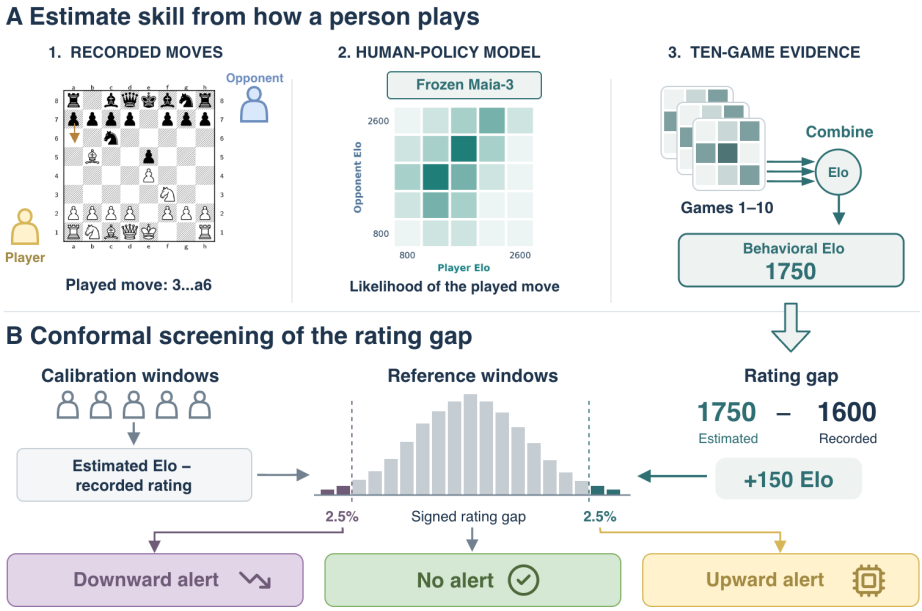

Figure 1: CheckMate CP overview. (A) Estimate Behavioral Elo from ten games using a frozen human-policy model. (B) Calibrate its gap from recorded rating against reference windows to produce directional alerts. The board and values are schematic; scoring excludes the first eight plies.

3.1 ESTIMATING BEHAVIORAL SKILL FROM HUMAN POLICY MODELS

Maia-3 (Monroe et al., 2026) serves as a frozen foundation model for our skill estimators. Trained on human games across skill levels, it assigns probabilities to observed moves given board history and the two players' candidate ratings. Let $g \in \{1, \ldots, 10\}$ index games and $\mathcal{G} = \{800, 900, \ldots, 2600\}$ be the candidate-rating grid. For candidate focal-player rating e and opponent rating o, both in $\mathcal{G}$, let $P_{ig}(e, o)$ be the product of probabilities assigned to the focal player's retained moves after the first eight plies in game g. We also compute likelihoods from the opponent's moves. Neither player's recorded rating nor the game outcome enters these prediction features. The pretrained likelihood profiles provide evidence about which skill levels explain the observed choices. All three estimators reuse these profiles to produce Behavioral Elo.

Direct likelihood. We first infer each opponent's rating weights $q_{ig}(o)$ from that opponent's moves, using uniform candidate priors. These weights sum to one. For each focal-player rating, we average the move likelihood over these opponent weights and combine the evidence across games. Normalizing across e gives weights $w_i^{\mathrm{D}}(e)$; their mean is the Direct estimate:

$$w_i^{\mathrm{D}}(e) \propto \prod_{g=1}^{10} \sum_{o \in \mathcal{G}} q_{ig}(o) P_{ig}(e, o), \qquad \widehat{E}_i^{\mathrm{direct}} = \sum_{e \in \mathcal{G}} e\, w_i^{\mathrm{D}}(e).$$

Each game can have a different inferred opponent strength. This estimator fits no project-specific parameters.

Rating-calibrated likelihood. This estimator combines both players' move likelihoods within each game, with a prior favoring nearby candidate ratings. Aggregating across games and applying a fixed temperature gives normalized weights $w_i^{\mathrm{C}}(e)$. An affine mapping with fitted offset a and slope β produces $\widehat{E}_i^{\mathrm{cal}} = a + \beta \sum_{e \in \mathcal{G}} e\, w_i^{\mathrm{C}}(e)$. Thus, both the evidence aggregation and its mapping to recorded rating differ from Direct.

Neural residual. A shared multilayer perceptron encodes each game's likelihood features, including rating profiles, opponent weights, move counts and color. The encodings' feature-wise means and standard deviations summarize play across the window, independently of game order. A prediction head combines these pooled features with window summaries. Let H_i denote these game and window inputs, and θ the fitted network parameters. The head learns an Elo correction $\delta_\theta(H_i)$ to Direct:

$$\widehat{E}_i^{\mathrm{neural}} = \widehat{E}_i^{\mathrm{direct}} + \delta_\theta(H_i). \tag{1}$$

The affine mapping minimizes squared rating error; the neural model minimizes Gaussian negative log-likelihood against recorded ratings. Both give equal total weight to each fitting account. Maia-3 remains frozen. Appendix B gives the complete likelihood calculations, feature definitions and training details.

3.2 Behavioral Skill Discrepancy

Write $\widehat{E}_i^b$ for the chosen estimator's output and define the raw rating gap $D_i = \widehat{E}_i^b - E_i$. Positive gaps indicate stronger-than-rated play; negative gaps indicate weaker-than-rated play. Typical gaps can vary with rating and time control. Let μ_i and $\sigma_i > 0$ be the account-balanced mean and standard deviation of development gaps for the chosen estimator and window i's recorded-rating band and time control. The standardized gap is

$$Z_i = (D_i - \mu_i)/\sigma_i. \tag{2}$$

Windows in the same group use the same mean and scale, fixed before calibration. Generated experiments retain D_i in Elo; human-donor and main ChessFraud analyses use Z_i to adjust for typical rating- and time-control differences. The screening score S_i is either D_i (in Elo) or Z_i (dimensionless), as specified for each analysis. Appendix C specifies the groups and handling of small groups.

3.3 Distribution-Free Calibration of Directional Alerts

Calibration measures how unusual a window's score is among reference windows processed by the same fixed estimator and standardization rule. Let S_j^{cal} be the score of calibration window $j \in \{1, \ldots, n\}$, where n is the calibration sample size, and S_i the new window's score. The indicator $\mathbf{1}\{\cdot\}$ equals one when its condition holds and zero otherwise. The upward and downward conformal p-values are

$$p_i^{\uparrow} = \frac{1 + \sum_{j=1}^{n} \mathbf{1}\{S_j^{\mathrm{cal}} \geq S_i\}}{n+1}, \qquad p_i^{\downarrow} = \frac{1 + \sum_{j=1}^{n} \mathbf{1}\{S_j^{\mathrm{cal}} \leq S_i\}}{n+1}. \tag{3}$$

Ties are counted conservatively. For each direction $d \in \{\uparrow, \downarrow\}$, choose a level $\alpha_d \geq 0$ independently of calibration and test scores. The total nominal level is $\alpha = \alpha_\uparrow + \alpha_\downarrow \leq 1$. An alert in direction d occurs when $p_i^d \leq \alpha_d$. Our primary rule assigns 2.5% to each direction, for total 5%. External assistance screening uses $\alpha_\uparrow = 0.05$ and $\alpha_\downarrow = 0$.

Proposition 1 (Reference-relative alert control)**.** *Fix the estimator and standardization. If the n calibration scores and one new reference score are jointly exchangeable, each directional alert has probability at most α_d, and either alert has probability at most α. These probabilities are marginal over the calibration sample and new window.*

This follows from standard conformal rank validity and a union bound (Vovk et al., 2005; Bates et al., 2023); the proof is in Appendix C.6. The selected reference population defines the operational null. An alert marks an unusual discrepancy for review; it does not identify its cause.

3.4 Data and Evaluation Design

The evaluation tests whether Behavioral Elo estimates recorded rating, responds to changes in play, and converts those changes into calibrated alerts. Natural games, generated interventions, human-donor substitutions and logged assistance address these questions under different conditions.

Natural games: fitting and rating estimation. The April–July 2026 Lichess corpus (Lichess, 2026a) supplies 3,075 accounts in disjoint roles: 880 for fitting, 918 for development, 1,000 for calibration and 277 for testing. The fitting accounts contribute 20,845 ten-game windows from 208,398 distinct games, with 6.22 million eligible focal-player moves. Within development, 308 accounts select model settings; the full development set estimates gap-standardization parameters. The main rating comparison uses the 197 test windows with scores available for all nine estimators. These windows contain 1,970 games and 54,460 eligible focal-player moves.

Maia-3 was pretrained on January 2023–July 2025 Lichess games (Monroe et al., 2026), before our project data, and its weights remain frozen. Reusing the foundation model's move probabilities lets the two trainable estimators learn the mapping to Behavioral Elo while keeping the chess policy fixed. The input audit found no natural calibration or test games in project fitting or development. The primary analysis retains public BOT accounts and games against them: BOT status alone does not establish a discrepancy between behavior and rating. A separate sensitivity analysis excludes affected windows and refits the project-trained components. Data provenance, split definitions and the BOT audit are in Appendices A and H.

Observation length. Ten games is the primary decision point, balancing evidence across games against the observation length needed for review. A separate analysis compares the first 5, 10 and 20 games of the same 209 test accounts. Rating-calibrated and Neural residual are fitted and calibrated separately at each length. In this separate analysis, accuracy uses the within-window median rating; screening uses the rating before the first game, fixed across lengths. This comparison measures the trade-off between estimation accuracy and detection at later decision points; it does not select an optimal length (Appendix I).

Model comparisons. Alongside the three Maia-3 estimators, two Maia-2 estimators use focal-player moves or both players' moves (Tang et al., 2024). Three further baselines fit Ridge regression to frozen representations from Karvonen's chess language model (Karvonen, 2024) and two models pretrained for move prediction: a convolutional neural network (CNN) and a Transformer. A CNN trained from scratch on board–move pairs (Scratch CNN) completes the nine rating estimators. Project-fitted comparators use the same 880 fitting accounts and 308 model-selection accounts. Mean opponent rating provides a separate matchmaking-based comparator for rating accuracy. For screening, Stockfish 18 (The Stockfish Team, 2026) supplies top-match (agreement with the engine's preferred move). Each score uses its own conformal reference; the available windows can differ across score families. Architectures, fitting procedures and calibration details are in Appendix C.

Generated changes: controlled sensitivity. Generated games test the full screening procedure under changes whose direction, size and frequency can be controlled. Maia-3 continues matched historical openings while the focal player's rating reference and opponent policy stay fixed. Policy-conditioning offsets range from ± 200 to ± 800 Elo; engine assistance is assigned at different move frequencies in 1, 2, 5 or 10 games. These settings specify interventions, not realized skill gains. The common comparison uses 498 unchanged calibration windows and one window per condition from the same 249 test accounts across 25 conditions, including a matched control. The controls assess

false alerts; the interventions assess sensitivity in both directions. Because Maia-3 also supplies the scoring backbone, these generated changes are not used to rank estimators.

Human-donor substitutions: changes using real play. A simple complementary test evaluates a donor's complete ten-game window against a different recipient's recorded rating. Controls use donors with similar ratings; alternatives use stronger or weaker donors, matched by month, time-control category and color balance. Substitution changes both the focal player's moves and opponent context, testing discrepancies across complete human-game windows. Shared games make the matched windows dependent.

ChessFraud: response and detection under logged assistance. ChessFraud, introduced at KDD 2026 (Linich et al., 2026), records engine use during controlled human tournaments. After data checks, 498 of its 505 games from 49 participants remain. Paired assisted and unaided windows from 13 participants measure Behavioral Elo response; 44 windows, one per participant and each containing 2–8 assisted games, measure detection against participants' reported ratings. The cohorts partly overlap. The main Maia-3 screening reference uses 890 calibration and 897 test windows from August public Lichess games at the same five-minute time control. The generated, donor and ChessFraud constructions are detailed in Appendices E, F and G.

Evaluation measures. Rating accuracy is measured by mean absolute error (MAE) in Elo. Changes in Behavioral Elo measure response to an intervention, whether or not an alert follows. The *false-alert rate* is the fraction of unchanged control windows flagged in either direction, estimating Type I error. Comparing this rate with the nominal level checks conformal calibration. The *detection rate* is the fraction of intervention windows flagged in the intended direction, estimating directional power. On public accounts without verified assistance labels, the alert fraction is *review burden*. Operating curves vary the nominal level from 1% to 20%, comparing detection with false alerts or review burden on the corresponding reference. The primary rule allocates 2.5% to each direction; assistance-only screening allocates 5% upward. Test-account and calibration resampling quantify separate sources of uncertainty, conditional on the fitted models (Appendix C.8).

4 Results

4.1 Rating estimation from moves

Neural residual has the lowest observed mean absolute error (MAE) among the nine estimators in Table 1, evaluated on the same 197 held-out windows. Its MAE is 30.5% lower than Maia-2 (both players). Rating-calibrated also improves on this comparator. Mean opponent rating achieves a lower MAE of 39.0 Elo, but with opponents held fixed it cannot respond to changes in the focal player's moves. Additional estimators, error metrics and paired comparisons are in Appendix D.

Table 1: **Rating estimation on 197 test windows, one per account.** Confidence intervals (CI) use test-account bootstrap resampling with fitted models fixed. Bold marks the lowest MAE.

Estimator	MAE (Elo) ↓	95% CI
Comparison methods		
Maia-2 (both players)	133.6	[115.0, 153.2]
Maia-2 (focal moves)	169.8	[147.2, 193.2]
Karvonen + Ridge	145.4	[126.0, 166.5]
Pretrained Transformer + Ridge	194.3	[171.6, 217.7]
Pretrained CNN + Ridge	202.2	[178.8, 227.1]
Scratch CNN	256.5	[229.6, 284.3]
CheckMate CP estimators		
Direct likelihood	165.7	[143.2, 188.5]
Rating-calibrated	112.2	[94.9, 132.0]
Neural residual	**92.8**	[78.5, 109.0]

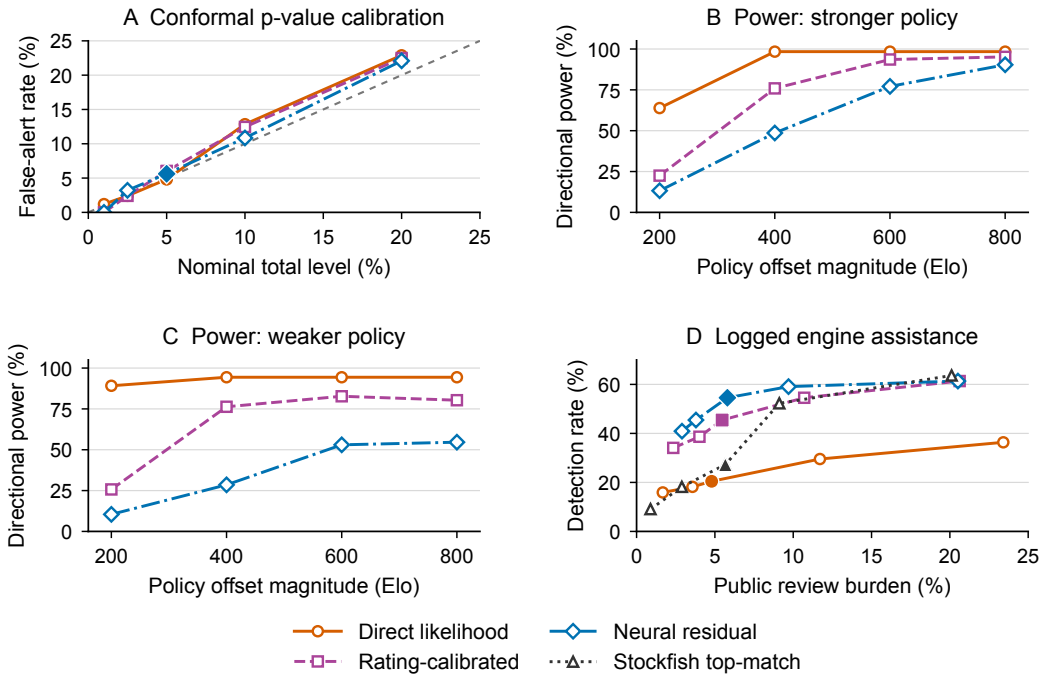

Figure 2: **Conformal calibration and power.** (A) Observed false-alert rates; the diagonal denotes equality with the nominal level. (B,C) Directional power at total 5% for stronger/weaker generated policies. Each condition uses 249 test windows and 498 calibration windows. (D) ChessFraud detection versus public review burden as the upward p-value cutoff varies (Table 2). Levels in A,D are 1, 2.5, 5, 10 and 20%; filled markers denote 5%. Uncertainty appears in Tables E.3 and G.4.

4.2 Conformal calibration and detection power

Conformal screening produces false-alert rates near the nominal 5% on unchanged generated windows. With 498 separate calibration windows and 249 test windows, the observed rates are 4.8% for Direct likelihood, 6.0% for Rating-calibrated, and 5.6% for Neural residual. A false alert occurs when either directional p-value is at most 0.025. Figure 2A checks rejection rates across nominal levels; all three test-bootstrap intervals at the primary level include 5% (Table E.3).

Larger policy offsets generally produce higher directional power (Figure 2B–C): the p-value in the intervention direction crosses 0.025 more often. Engine assistance also shows a dose effect. At 10% move assignment in one game, detection rates are 2.8–4.8%, close to the upward-control rates of 2.0–3.6%. At fixed 80% assignment, extending assistance from one to five to all ten games raises detection from 4.0–5.2% to 26.5–59.4% to 86.7–98.4% across the three estimators. Each condition uses the same 249 test windows and total 5% rule. Assignment probabilities do not measure the fraction of moves actually changed. These experiments test sensitivity to controlled changes; they do not rank estimators for human play because Maia-3 generates the games and supplies Direct's likelihoods. Appendix E reports the full intervention grid.

4.3 Screening changes in human play

Screening detects rating discrepancies when the substituted play comes from real players (Figure 3A). At total 5%, Neural residual detects 80.8% of the 26 stronger-donor windows and 83.9% of the 31 weaker-donor windows. All three estimators produce zero matched-control alerts. The comparison uses donors rated about 400 Elo above or below the recipient; shared donors and games make these descriptive rates dependent (Appendix F).

Logged assistance raises Behavioral Elo without necessarily triggering an alert. In the 13 paired ChessFraud participants (Linich et al., 2026), four assisted games raise mean Behavioral Elo by

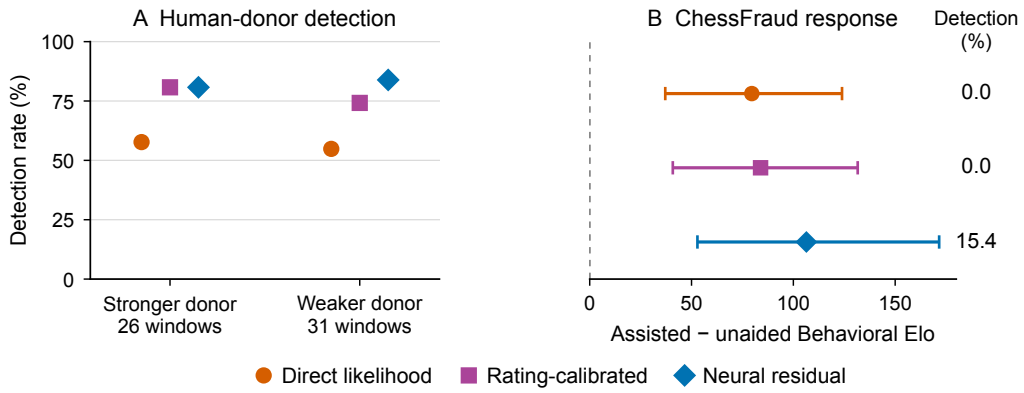

Figure 3: **Human behavior and logged assistance.** (A) Directional detection after human-donor substitution, at total 5%; matched controls have zero alerts. (B) Assisted-minus-unaided Behavioral Elo for 13 paired ChessFraud participants, with four assisted games per assisted window. Bars are paired-bootstrap 95% intervals; the right column gives assisted-window detection at upward 5% with August calibration.

Table 2: **ChessFraud detection at upward 5% with August calibration.** Rates are percentages; detection uses 44 assisted windows. The 95% CI resamples participants with calibration fixed. Behavioral estimators use standardized gaps, 890 calibration windows and 897 public test windows; Stockfish uses its native score and 898/900 windows, respectively. Bold marks the highest detection rate.

Screening method	Detection ↑	95% CI	Review burden
Stockfish top-match	27.3	[13.6, 40.9]	5.7
Direct likelihood	20.5	[9.1, 31.8]	4.8
Rating-calibrated	45.5	[31.8, 59.1]	5.5
Neural residual	**54.5**	[38.6, 68.2]	5.8

79.6–106.4 points across estimators, with paired intervals excluding zero (Figure 3B). Yet only 0.0–15.4% of these assisted windows have upward p-values at most 0.05. A measurable change in expressed skill need not be extreme relative to the calibration population.

The positive mean response also persists when game selection is repeated. Averaging each participant's response over the original and 20 additional selections gives mean increases of 57.1, 69.7 and 94.2 Elo for Direct likelihood, Rating-calibrated and Neural residual, respectively, with all three 95% intervals above zero (Appendix G).

With August calibration, Neural residual detects 54.5% of the 44 assisted windows, compared with 27.3% for Stockfish top-match, at similar public review burdens (Table 2). Under natural calibration, Stockfish detects more windows, with higher public review burden (Appendix G).

Increasing the upward p-value cutoff from 0.05 to 0.10 raises Neural residual's detection to 59.1%, while public review burden rises from 5.8% to 9.7% (Figure 2D). Public windows lack verified assistance labels, so review burden measures workload rather than false-positive rate. The paired and larger cohorts overlap.

4.4 Sensitivity to window length

Both fitted estimators improve rating accuracy with longer windows, while Neural residual's detection decreases. This separate 209-account comparison refits models and calibration at each length and uses a different rating reference from Figure 2B–C. Neural residual's MAE falls from 137.0 to 99.3 to 81.5 Elo at 5, 10 and 20 games. Its detection of generated +600 policy shifts falls from

35.4% to 30.1% to 9.6% at total 5%. From ten to twenty games, Neural residual shows a smaller mean Elo increase over matched controls and a higher calibrated threshold for upward alerts (Table I.2). Accuracy targets the within-window median rating; alerts use the fixed pregame rating. The comparison does not establish an optimal length (Appendix I).

5 Discussion and Conclusion

This work studies a simple but important question: can observed decision behavior reveal when an individual's expressed level of competence differs from their expected profile? Using online chess as a testbed, we develop a behavioral skill discrepancy measure by combining human-policy models with distribution-free calibration. Our results highlight several distinctions that are important for behavioral screening. First, accurately predicting a player's recorded rating and identifying unusual behavioral changes are different objectives. Estimators that incorporate flexible rating calibration can substantially reduce rating prediction error, while Direct likelihood is more responsive to the Maia-3-generated policy shifts. The shared generator and scorer limit this comparison. On donor substitutions and the 44 assisted ChessFraud windows, the fitted estimators detect more cases than Direct. Thus, model selection for screening should consider the downstream objective rather than prediction accuracy alone.

Second, a measurable behavioral response does not necessarily imply a statistically significant alert. Logged assistance and controlled interventions can increase inferred Behavioral Elo without always producing extreme discrepancies relative to the reference population. This distinction is important because screening systems operate under a trade-off between sensitivity and review burden. A calibrated alert should therefore be interpreted as evidence of an unusual behavioral pattern rather than a direct attribution of the underlying cause.

Third, the validity of conformal screening depends on the choice of reference population. Our guarantees concern the probability of alerts for new windows that are exchangeable with the calibration population. They apply to individual windows, without batch false discovery rate (FDR) (Benjamini & Hochberg, 1995; Bates et al., 2023) or repeated-monitoring control. Deployment populations may differ in rating dynamics, time controls or playing styles. Adapting conformal prediction beyond exchangeability (Barber et al., 2023) to these rating-window scores remains a direction for future work.

Several limitations should be considered. The comparison of 5-, 10- and 20-game windows shows that waiting for more games does not consistently improve detection, even when rating estimation improves. Window selection in other populations requires further study. Downward screening is evaluated through generated and donor substitutions; ChessFraud provides assistance labels only. Small human cohorts and shared games limit independent replication. Finally, a behavioral discrepancy does not uniquely identify its cause: genuine improvement, rating lag, unusual play styles, or other factors may produce similar signals.

CheckMate CP also suggests applications beyond chess wherever observed behavior can be compared with a recorded assessment of skill. Examples include online games that use skill ratings for matchmaking (Herbrich et al., 2006) and education, where knowledge-tracing models estimate student knowledge from response sequences (Piech et al., 2015). Future work could examine whether discrepancies between recent behavior and recorded assessments help identify cases for reassessment. Extending the framework would require behavioral models for the target domain, skill estimates on a comparable scale, and a suitable calibration population.

AI USE STATEMENT

In this work, generative AI tools provided feedback on the authors' study design and assisted with literature search, interpretation of experimental results, manuscript drafting and editing, translation, figure preparation, and limited code editing and command-line debugging. AI assistance with the author-developed proposition and proof was limited to language editing. The authors checked numerical claims against saved experimental outputs and manually tested AI-assisted code used in the analyses. They take responsibility for the final text, code, analyses, and claims.

ETHICS STATEMENT

CheckMate CP studies existing Lichess game records, the published ChessFraud dataset, and controlled chess simulations to support fair online competition. We report aggregate findings and do not identify individual accounts as cheaters. Account histories may contain identifying information, so any sharing of derived data should minimize personal information and follow source permissions. An alert flags an unusual discrepancy between behavior and recorded rating for review, not proof of cheating or intent; consequential decisions should include contextual assessment, human review, and a way for players to contest them.

REPRODUCIBILITY STATEMENT

Section 3 defines the Behavioral Elo estimators, screening scores, and conformal calibration. The Appendix documents cohort construction, model readouts, comparators, uncertainty estimation, and the calibration proof (Appendices A–C). It also specifies the controlled interventions and ChessFraud evaluation (Appendices E and G), with analysis denominators and results across operating levels in the accompanying tables.

REFERENCES

Rina Foygel Barber, Emmanuel J. Candès, Aaditya Ramdas, and Ryan J. Tibshirani. Conformal prediction beyond exchangeability. *The Annals of Statistics*, 51(2):816–845, 2023. doi: 10.1214/23-AOS2276.

David J. Barnes and Julio C. Hernandez-Castro. On the limits of engine analysis for cheating detection in chess. *Computers & Security*, 48:58–73, 2015. doi: 10.1016/j.cose.2014.10.002.

Stephen Bates, Emmanuel Candès, Lihua Lei, Yaniv Romano, and Matteo Sesia. Testing for outliers with conformal p-values. *The Annals of Statistics*, 51(1):149–178, 2023. doi: 10.1214/22-AOS2244.

Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(1):289–300, 1995. doi: 10.1111/j.2517-6161.1995.tb02031.x.

Ralf Herbrich, Tom Minka, and Thore Graepel. TrueSkill: A Bayesian skill rating system. In *Advances in Neural Information Processing Systems*, volume 19, pp. 569–576, 2006.

Adam Karvonen. Emergent world models and latent variable estimation in chess-playing language models. In *Proceedings of the First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=PPTrmvEnpW.

Kyota Kuboki, Tatsuyoshi Ogawa, Chu-Hsuan Hsueh, Shi-Jim Yen, and Kokolo Ikeda. Policies of multiple skill levels for better strength estimation in games. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 21, pp. 81–90, 2025. doi: 10.1609/aiide.v21i1.36812.

Jing Lei, Max G'Sell, Alessandro Rinaldo, Ryan J. Tibshirani, and Larry Wasserman. Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 113 (523):1094–1111, 2018. doi: 10.1080/01621459.2017.1307116.

Lichess. Lichess open database, 2026a. URL https://database.lichess.org/. Monthly game exports; accessed September 23, 2026.

Lichess. Terms of service, 2026b. URL https://lichess.org/terms-of-service. Fair Play Violations section; last modified August 1, 2026; accessed September 26, 2026.

Lichess. Chess rating systems, n.d. URL https://lichess.org/page/rating-systems. Accessed September 26, 2026.

Anastasiia Linich, Artem Lepin, Andrey Sakhovskiy, Anita Toleutaeva, Georgii Lepa, Andrei Neznamov, and Semen Budennyy. ChessFraud: Exploring the capabilities of human-aligned models for cheating detection in online chess. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2026. doi: 10.1145/3770855.3817587.

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*, pp. 1677–1687, 2020. doi: 10.1145/3394486.3403219.

Reid McIlroy-Young, Yu Wang, Siddhartha Sen, Jon M. Kleinberg, and Ashton Anderson. Detecting individual decision-making style: Exploring behavioral stylometry in chess. In *Advances in Neural Information Processing Systems*, volume 34, pp. 24482–24497, 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/hash/ccf8111910291ba472b385e9c5f59099-Abstract.html.

Daniel Monroe, George Eilender, Philip Chalmers, Zhenwei Tang, and Ashton Anderson. Chessformer: A unified architecture for chess modeling. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/3d167db04a90885ad5208fe8b273668b-Abstract-Conference.html.

Chris Piech, Jonathan Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas J. Guibas, and Jascha Sohl-Dickstein. Deep knowledge tracing. In *Advances in Neural Information Processing Systems*, volume 28, 2015. URL https://proceedings.neurips.cc/paper/2015/hash/bac9162b47c56fc8a4d2a519803d51b3-Abstract.html.

Kenneth W. Regan and Guy McC. Haworth. Intrinsic chess ratings. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 25, pp. 834–839, 2011. doi: 10.1609/aaai.v25i1.7951.

Kenneth W. Regan, Bartlomiej Macieja, and Guy McC. Haworth. Understanding distributions of chess performances. In *Advances in Computer Games*, volume 7168 of *Lecture Notes in Computer Science*, pp. 230–243. Springer, 2012. doi: 10.1007/978-3-642-31866-5_20.

David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529:484–489, 2016. doi: 10.1038/nature16961.

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

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*, volume 37, pp. 20919–20944, 2024. doi: 10.52202/079017-0659.

The Stockfish Team. Stockfish 18, 2026. URL https://stockfishchess.org/blog/2026/stockfish-18/.

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Springer, 2005. doi: 10.1007/b106715.

APPENDIX CONTENTS

A Data and Evaluation Cohorts **14**

A.1 Natural-game splits . . . 14

A.2 From selected accounts to analysis cohorts . . . 14

A.3 ChessFraud cohorts and labels . . . 15

B Estimating Behavioral Elo **16**

B.1 Move probabilities at candidate ratings . . . 16

B.2 Direct likelihood: infer opponents, then combine games . . . 16

B.3 Rating-calibrated likelihood: combine both sides and fit the scale . . . 17

B.4 Neural residual: learn a correction from ten games . . . 17

C Comparators and Conformal Calibration **18**

C.1 Additional rating estimators . . . 18

C.2 Comparator architectures . . . 19

C.3 Engine score . . . 20

C.4 Standardizing rating gaps . . . 21

C.5 Screening protocols and metrics . . . 21

C.6 Proof of reference-relative alert control . . . 21

C.7 Calibration size and attainable thresholds . . . 21

C.8 Bootstrap uncertainty . . . 22

D Rating Prediction on Natural Games **22**

E Controlled Changes in Playing Behavior **24**

E.1 Constructing matched interventions . . . 24

E.2 Detection across intervention sizes . . . 24

F Engine-Score Checks and Human-Donor Substitution **27**

F.1 Stockfish scoring of generated play . . . 27

F.2 Substituting real human play . . . 27

G Logged Assistance in ChessFraud **28**

G.1 Behavioral response and assisted-window detection . . . 28

G.2 Choice of calibration reference . . . 28

G.3 Sensitivity to game selection . . . 29

G.4 Detection and review burden at different thresholds . . . 29

G.5 Additional estimators and comparison scope . . . 31

H Public BOT Accounts and Exclusion Sensitivity **31**

I Sensitivity to Observation Window Length **34**

I.1 Accuracy targets and alert boundaries . . . 35

A Data and Evaluation Cohorts

We use natural games to fit and evaluate rating estimators, generated games to test controlled changes in play, and ChessFraud to study logged assistance. Table A.2 summarizes which accounts enter each analysis. A ten-game window is the screening unit. A physical game contributes two player-game perspectives, one for each player; these are not independent games.

Natural windows contain consecutive games at one time control, with the median recorded player rating as their rating reference. Simulated windows retain their source window's reference; ChessFraud uses the participant's reported rating, with the conversion described below where needed. Within-window rating updates may attenuate discrepancies relative to a pre-window reference. External windows are assembled for the analysis and are not verified chronological sequences. Ten games is the primary observation length; Appendix I evaluates separate five-, ten- and twenty-game fits.

A.1 Natural-game splits

Project fitting, development and natural evaluation use April–July 2026 games from the Lichess Open Database (Lichess, 2026a). The estimators that require project-specific training use 880 accounts, contributing 20,845 windows. This fitting is separate from Maia-3 pretraining, which uses January 2023–July 2025 Lichess games (Monroe et al., 2026). The project corpus is later than this training period; the pretrained weights remain fixed. The accuracy comparison uses the same 197 held-out accounts for every method. Development accounts are used to select model settings and estimate rating-gap standardization parameters. Calibration accounts then determine alert thresholds. The input audit found no account overlap between fitting, development, natural calibration and natural test sets. It also checked that natural calibration and test games were separate from fitting and development games. This separation protects against fitting leakage. Joint exchangeability is an additional assumption about the eligible calibration and new scores; neither account separation nor group standardization establishes it.

The 197-account accuracy comparison is a subset of the 277-account natural test cohort, with scores available for all nine estimators. Historical evaluation accounts reused for model development are assigned to development, not to this held-out comparison. The expanded calibration and test cohorts are disjoint from the current 880 fitting accounts; they are not used to evaluate earlier models fitted on overlapping accounts.

The current fitting, development, calibration and test roles contain 3,075 distinct accounts in total. Table A.1 distinguishes accounts, windows, physical games and eligible moves. The fitting windows contain 208,450 player-game perspectives from 208,398 distinct physical games: 52 games contribute both players' perspectives. Those physical games contain 14,107,521 plies, where a ply is one player's move. Eligible-move counts below refer only to the focal player's moves retained for model scoring; they are not counts of independent training examples.

Table A.1: Natural-game data volume. Games count distinct physical games within each row. Eligible moves count the focal-player moves retained in the saved scoring inputs. Standardization uses one window per development account; model selection uses additional windows from its 308-account subset. The common rating-test row is a subset of the full natural test.

Role	Accounts	Windows	Games	Eligible moves
Project fitting	880	20,845	208,398	6,222,229
Standardization development	918	918	9,180	262,459
Natural calibration	1,000	1,000	10,000	285,356
Natural test	277	277	2,770	78,729
Common rating test	197	197	1,970	54,460

A.2 From selected accounts to analysis cohorts

Selected counts describe the input cohort; analyzed counts describe accounts with valid scores for a particular comparison. They count actual accounts, not weighted sample sizes. A complete ten-game

Table A.2: Datasets, sample sizes and analysis roles. Counts refer to accounts unless stated otherwise. Calibration/test counts are listed in that order. The usable sample depends on which estimators have valid scores. Project fitting is separate from foundation-model pretraining.

Source / role	Size	Use and overlap
Estimator fitting	880 accounts; 20,845 windows	Account-balanced fit; disjoint from development, calibration and test.
Development	308 validation; 918 natural development	The 918 comprise 308 validation plus 610 historical evaluation accounts. Model settings use 308; Maia-3 gap standardization uses 918.
Natural calibration/test	1,000 /277; test core 197	One window per account. The core has 200 calibration accounts; the extension adds 800 calibration and 80 test accounts. Main accuracy uses the 197 core test accounts, the subset scored by all nine estimators.
Generated calibration/test	Planned 500/250; common 498/249	Subsets of corresponding natural roles; same test accounts in 25 conditions. 6,750 planned windows, with accounts reused across conditions.
August public reference	Selected 1,000/1,000	One ten-game window per account. Method-specific analysis counts appear in Table A.3.
ChessFraud source/retained	505/498 physical games; 49 players	996 retained perspectives: 399 assisted, 597 unassisted.
ChessFraud assisted cohort	44 assisted ten-game windows	One assisted window per participant; 100 physical games are shared between windows.
ChessFraud paired cohort	13 players; doses 0/1/2/4	Partly overlaps the assisted cohort. Same participants at each dose. Original selection plus 20 resamplings; no extra participants.
Human-donor substitution	88 calibration; 26 up /31 down	Donors and recipients are matched across distinct accounts and games. Shared-game links connect observations into one dependent group.

history does not guarantee a usable score: estimators can require eligible moves from the tracked player, the opponent, or both. Comparisons on common accounts retain only accounts scored by every method in that comparison.

In the simulation, Direct likelihood, Rating-calibrated and Neural residual each score 499 calibration accounts and all 250 test accounts in all 25 conditions. One calibration window has no eligible tracked-player moves. Maia-2 (both players) additionally lacks eligible opponent moves for one calibration account and one test account in 24 conditions. The shared panel therefore uses 498 calibration and 249 test accounts in every condition. In the natural-reference comparison, the eight calibration accounts absent from the five-system group also lack eligible opponent moves for Maia-2 (both players).

For the August reference, the selected cohort remains 1,000 calibration and 1,000 test accounts. Maia-3 requires eligible Elo inputs across all ten games; 213 of the 2,000 selected windows do not meet this requirement. Scoring availability differs between Maia-3, Stockfish and the five-system backbone comparison. Table A.3 reports the resulting counts without replacing unavailable scores or refilling the selected cohort.

A.3 ChessFraud cohorts and labels

ChessFraud labels record engine use during controlled tournaments (Linich et al., 2026). Participants used Stockfish 15 at assigned depths of 12, 14, 16 or 18, with up to five suggested lines displayed. The labels therefore identify assistance directly, rather than inferring it from similarity to engine moves.

The analysis retains 498 of 505 physical games from 49 participants. Four excluded games continue after an automatic terminal position; three have inconsistent player-game and move-level assistance labels. The retained games yield 996 player-game perspectives: 399 assisted and 597 unassisted. Games use 300+0 time control: five minutes without increment.

Table A.3: Selected accounts and final analysis counts. Paired counts list calibration/test accounts; natural-reference rows list calibration only. "Not analyzed" is the difference between selected and analyzed counts for that scoring group, not an additional exclusion stage. The five-system group contains Maia-2 (focal moves) and Maia-2 (both players), pretrained CNN and Transformer, and Karvonen.

Source	Scoring group	Selected	Not analyzed	Analyzed
Simulation	Shared estimator panel	500/250	2/1	498/249
Natural	Maia-3 standardized gaps	1,000	0	1,000
	Stockfish top-match	1,000	1	999
	Five-system raw-gap comparison	1,000	8	992
August	Maia-3 standardized gaps	1,000/1,000	110/103	890/897
	Stockfish top-match	1,000/1,000	102/100	898/900
	Five-system raw-gap comparison	1,000/1,000	112/104	888/896

The two ten-game cohorts answer different questions. The 44-player cohort measures detection among assisted windows, with 2–8 assisted games in each window. The 13-player paired cohort compares the same people at four assistance doses, including unaided play. The cohorts partly overlap. Shared physical games also link some participants' windows, so their counts cannot be added as independent observations.

Four source ratings are available only from Chess.com and are converted using an approximate +300 offset. The available identity mapping does not permit a separate analysis excluding these four players. Assistance labels describe the games; the fraction of assisted moves among each scorer's retained positions is not measured.

B Estimating Behavioral Elo

All three estimators begin with the same frozen Maia-3 move probabilities. They differ in how they combine this evidence and whether they learn a mapping to recorded ratings. This section gives the full calculations behind the main-text formulas. We describe one ten-game window and omit its window index i throughout this section.

B.1 Move probabilities at candidate ratings

Let $g \in \{1, \ldots, 10\}$ index the games, and let $\mathcal{G} = \{800, 900, \ldots, 2600\}$ be the candidate-rating grid. The variables $e, o \in \mathcal{G}$ denote candidate ratings for the focal player and opponent. For game g, define $P_g(e, o)$ as the product of Maia-3 probabilities assigned to the focal player's retained moves. Define $O_g(o, e)$ in the same way for the opponent's moves, reversing which rating conditions the mover.

Each eligible move is evaluated at all 361 candidate matchups. The model receives eight board histories, excludes clock input, and skips the first eight plies. Neither player's declared rating nor the game result enters the predictor. Calculations use summed log-probabilities and log-sum-exp to avoid numerical underflow. These products summarize move evidence; statistical calibration is performed across ten-game windows, not across individual moves.

B.2 Direct likelihood: infer opponents, then combine games

The first step estimates each opponent's strength from their own moves. Let $q_g(o)$ be the resulting opponent-rating weights under uniform candidate priors. The second step averages the focal player's likelihood over these weights, giving game-level evidence $A_g(e)$. Multiplying this evidence across ten games and normalizing gives the rating weights $w^{\mathrm{D}}(e)$. Their mean is the Direct likelihood

estimate, $\widehat{E}^{\mathrm{direct}}$. Here o' and e' are summation indices over $\mathcal{G}$:

$$q_g(o) = \frac{\sum_{e\in\mathcal{G}} O_g(o,e)}{\sum_{o'\in\mathcal{G}}\sum_{e\in\mathcal{G}} O_g(o',e)},$$
$$A_g(e) = \sum_{o\in\mathcal{G}} q_g(o) P_g(e,o),$$
$$w^{\mathrm{D}}(e) = \frac{\prod_{g=1}^{10} A_g(e)}{\sum_{e'\in\mathcal{G}}\prod_{g=1}^{10} A_g(e')}, \qquad \widehat{E}^{\mathrm{direct}} = \sum_{e\in\mathcal{G}} e\, w^{\mathrm{D}}(e).$$

This procedure allows a different inferred opponent rating in each game while combining evidence about one focal player across the window. The output is a weighted mean over candidate ratings, not the single highest-weight rating. No project-specific parameters are fitted.

B.3 Rating-calibrated likelihood: combine both sides and fit the scale

This estimator combines both players' evidence within each game. It uses a prior that gives greater weight to opponents near the focal player's candidate rating. Let $\sigma_{\mathrm{prior}} = 100$ Elo be the prior width and $\pi(o \mid e)$ be proportional to $\exp(-(o-e)^2/(2\sigma^2_{\mathrm{prior}}))$, normalized over o.

Let $L(e)$ denote the resulting ten-game log-evidence. We convert it to normalized weights $w^{\mathrm{C}}(e)$ at fixed temperature $T = 8$, which spreads weight more evenly than temperature one. An affine transformation then maps their mean to recorded ratings. With offset a, slope β, and prediction $\widehat{E}^{\mathrm{cal}}$,

$$L(e) = \sum_{g=1}^{10} \log\left[\sum_{o\in\mathcal{G}} P_g(e,o) O_g(o,e) \pi(o \mid e)\right],$$
$$w^{\mathrm{C}}(e) = \frac{\exp(L(e)/T)}{\sum_{e'\in\mathcal{G}}\exp(L(e')/T)}, \qquad \widehat{E}^{\mathrm{cal}} = a + \beta\sum_{e\in\mathcal{G}} e\, w^{\mathrm{C}}(e).$$

We fit a and β by account-balanced least squares on the 880 fitting accounts, assigning each window weight inversely proportional to the number of fitting windows from its account. The prior width and temperature are held fixed during this fit; only a and β are estimated on these accounts. Maia-3 remains frozen. Compared with Direct likelihood, this estimator changes both the way move evidence is combined and the mapping to the rating scale.

B.4 Neural residual: learn a correction from ten games

The neural estimator learns which features of a ten-game window predict errors in Direct likelihood. For game g, let x_g be a 102-element vector containing direct and joint rating profiles, probability vectors, opponent weights, move counts, uncertainty summaries and color. Let v be an 11-element vector of window summaries, and let θ denote the fitted network parameters. Together, these inputs form $H = (x_1, \ldots, x_{10}, v)$, the main text's H_i with the window index omitted.

Table B.1 specifies these inputs. The joint features use uniform opponent priors, without the rating-calibrated estimator's prior, temperature or affine mapping. Game and window features are standardized using fitting accounts only. All three estimators use opponent-move evidence as well as focal-player moves. The human-donor experiment changes both sources together (Appendix F).

Here entropy is the negative sum of probability times its natural logarithm; endpoint mass is the total probability at the lowest and highest grid ratings. Joint evidence for a game sums $P_g(e,o)O_g(o,e)$ over uniformly weighted opponent candidates. Multiplying the game evidence and normalizing gives the joint window-rating distribution used in v.

A shared encoder f_θ maps each game's features through layers of width $102 \to 64 \to 32$, with a rectified linear unit (ReLU) after the first layer. We pool valid game encodings by their feature-wise means and population standard deviations (sd). Concatenating these with v produces the 75-element vector z:

$$z = [\mathrm{mean}_g\, f_\theta(x_g), \mathrm{sd}_g\, f_\theta(x_g), v].$$

Table B.1: Neural residual inputs; sd denotes standard deviation. Each rating profile has 19 entries in ascending candidate-rating order. Log-evidence profiles are centered by subtracting their maximum; probability profiles are normalized to sum to one.

Feature group	Dimensions
Per-game vector x_g	
Direct and joint centered log-evidence profiles	$19 + 19$
Direct and joint normalized rating profiles	$19 + 19$
Inferred opponent-rating probabilities q_g	19
Retained focal-player and opponent move counts	2
Opponent-rating sd, entropy, maximum mass and endpoint mass	4
Focal player is White (binary indicator)	1
Per-game total	**102**
Window vector v	
Direct and joint window-rating mean, sd, entropy, maximum mass and endpoint mass	$5 + 5$
Direct minus joint window-rating mean	1
Window total	**11**

Pooling summarizes both typical play and variation across games, without depending on game order. A prediction head of width $75 \to 32 \to 16 \to 2$, with hidden ReLU activations, returns two scalars, u_1 and u_2. Let s be the training-target standard deviation, floored at 100 Elo, and define $\mathrm{softplus}(u) = \log(1 + \exp u)$ for a scalar u. The predicted rating $\widehat{E}^{\mathrm{neural}}$ and standard deviation $\widehat{\sigma}^{\mathrm{neural}}$ are

$$\widehat{E}^{\mathrm{neural}} = \widehat{E}^{\mathrm{direct}} + su_1, \qquad \widehat{\sigma}^{\mathrm{neural}} = s[\mathrm{softplus}(u_2) + 0.1].$$

The correction $\delta_\theta(H) = su_1$ is the term added to Direct likelihood in equation 1. Training minimizes Gaussian negative log-likelihood against recorded ratings, with equal total weight per fitting account. The fixed training recipe uses AdamW with learning rate 5×10^{-4}, weight decay 10^{-4}, gradient-norm clipping at 5, and 300 full-data optimization steps, with random seed 20260721. Each step accumulates the account-weighted loss over the fitting windows before updating parameters. Physical batches of 128 windows bound memory use; they do not cause separate optimizer updates. Masked games are excluded from pooling. The final model uses the fixed training length. The predicted spread is used in training, but it does not normalize the conformal score. Screening uses either the raw rating gap or its development-based standardization.

C Comparators and Conformal Calibration

We compare rating estimates with engine-based measurements of play. These inputs measure different aspects of behavior, so we describe their construction before explaining how each becomes a screening score.

C.1 Additional rating estimators

Behavior-only estimators use moves and board context while excluding declared ratings and game results from their predictor inputs. For estimators fitted in this project, recorded ratings provide supervised training targets. Table C.1 summarizes their inputs and fitting roles.

Maia-2 likelihood estimators. Two estimators use the official pretrained Maia-2 (Tang et al., 2024) blitz or rapid checkpoint according to time control, without further training. Both evaluate an 11-point rating grid from 1050 to 2050 with uniform priors over player and opponent ratings, and report the posterior mean. Their estimates are confined to this range, narrower than Maia-3's 800–2600 grid. Maia-2 (focal moves) uses the focal player's move likelihoods, averaging over candidate opponent ratings. Maia-2 (both players) also includes the opponent's move likelihoods. These names describe estimators constructed for this comparison, not separate pretrained Maia-2 models.

Pretrained CNN and Transformer. These two baselines reuse policy networks from earlier experiments by the authors, with 10.0 million and 8.9 million parameters, respectively. They were

pretrained on mixed-rating Lichess positions to predict human moves with cross-entropy over legal moves, using a 1,968-action vocabulary. The recorded pretraining budget is one million board–move examples per model, with random seed one. Validation top-1 move accuracy selects the retained checkpoints at epochs ten and five, respectively. The retained records do not establish an identical pretraining corpus for the two models. For rating estimation, both networks remain frozen. Their hidden features are averaged over focal-player positions within each game. The original move-prediction heads do not supply the rating estimates.

Ridge fitting on frozen features. Karvonen's chess language model (Karvonen, 2024) uses the public eight-layer Lichess checkpoint.[1] The game representations have 768 dimensions for the pretrained CNN, 384 for the pretrained Transformer and 512 for Karvonen. For each of these three backbones, the window input averages ten game vectors with equal weight. Feature standardization and Ridge regression are fitted on the 880 fitting accounts, predicting the window's recorded rating. Each account receives equal total weight across its windows in both steps. The Ridge penalty is selected from $0.1, 1, 10, 100, 1000$ by window-level MAE on the 308 development accounts. No backbone weights are updated during this fit. The 880-account split therefore describes rating-regression fitting, separately from policy pretraining. Because these systems differ in pretraining and inputs, the comparison evaluates complete estimators rather than architecture alone.

Table C.1: Inputs and fitting roles for the rating comparators. The 880 fitting accounts and 308 model-selection accounts are separate from the common 197-account accuracy test.

Estimator	Behavioral input	Project fitting
Maia-2 (both players)	Both players' move likelihoods	None
Maia-2 (focal moves)	Focal moves; opponent candidates averaged	None
Pretrained CNN / Transformer	Frozen representations	Ridge regression; development penalty selection
Karvonen + Ridge	Frozen eight-layer chess-language-model representations	Ridge regression; development penalty selection
Scratch CNN	Board–move pairs	Network trained on 880 accounts; development epoch selection

C.2 Comparator architectures

Figures C.1–C.3 show how the three architectures map ten-game windows to rating estimates. The pretrained CNN and Transformer encode each board as 20 planes of size 8×8: 12 piece planes and eight planes for side to move, castling rights, en passant and move counters. Neither policy is conditioned on a player's rating. The currently played move is not appended to these inputs; the observed move history determines the positions being encoded.

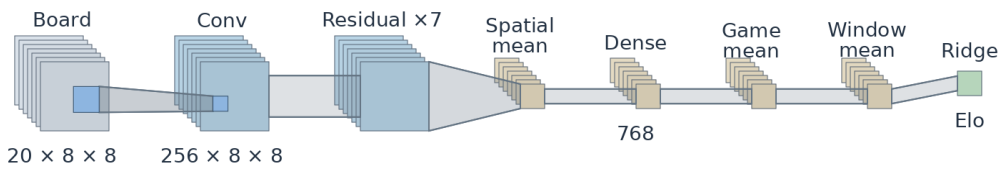

Figure C.1: **Pretrained CNN + Ridge.** Spatial mean pools within one board; game mean averages focal-position features, and window mean averages ten games equally. The encoder stays frozen; feature standardization and Ridge are fitted for rating prediction. Stacks are schematic.

Pretrained CNN + Ridge. A 3×3 convolution maps the 20 input planes to 256 feature maps, followed by batch normalization and ReLU. Seven residual blocks follow, each containing two 3×3 convolutions with batch normalization and a skip connection. ReLU follows the first convolution and the residual addition. Spatial average pooling reduces the board to a 256-element vector. A fully

[1] Checkpoint: `lichess_8layers_ckpt_no_optimizer.pt`.

connected layer and ReLU produce the 768-element representation extracted before the original 1,968-move output layer, with dropout disabled at evaluation. Game and window averaging produce the input to Ridge, a linear rating predictor with an ℓ_2 penalty on its coefficients.

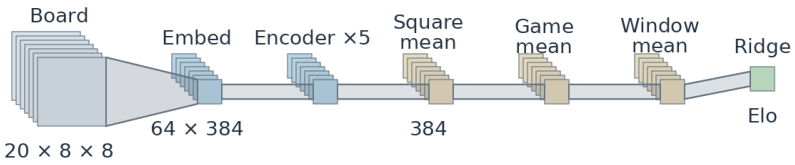

Figure C.2: **Pretrained Transformer + Ridge.** Attention operates among the 64 squares of one board. Square mean follows final layer normalization. The frozen features use the same game and window means and fitted Ridge regression as the pretrained CNN.

Pretrained Transformer + Ridge. The board encoding becomes 64 square tokens. A linear projection maps each token's 20 features to 384 dimensions; learned square-position embeddings preserve location. Five Transformer encoder layers use eight attention heads and a feed-forward width of 1,536 with GELU activations. After final layer normalization, the 64 square states are averaged into a 384-element representation. The pretrained policy combines from-square, to-square and promotion scores with a learned action bias. Rating estimation instead uses the pooled square states, with the same game averaging, window averaging and fitted Ridge regression as the pretrained CNN.

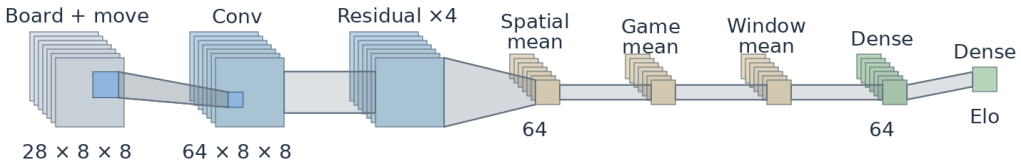

Figure C.3: **Scratch CNN.** Board–move features are averaged within each game and equally across ten games. The encoder and dense rating head are trained jointly from random initialization.

Scratch CNN. This network is trained from random initialization on the 880 fitting accounts, with its training epoch selected on the 308 development accounts. The input encodes the board and observed move in 28 planes: 12 piece planes, side to move, four castling rights, en passant, move origin and destination, five promotion categories, halfmove count and two repetition indicators. At most 32 evenly spaced focal-player positions are taken from each game. A 3×3 convolution maps 28 channels to 64, followed by ReLU and four residual blocks. Each block has two 3×3 convolutions and eight-group normalization. Spatial means are averaged within each game and then across the ten games. A $64 \to 64 \to 1$ head with a hidden ReLU predicts rating. The encoder and rating head are trained jointly using account-weighted absolute error, AdamW at learning rate 10^{-3} and weight decay 10^{-4}, and random seed 20260916. Each update uses 32 windows, accumulated in batches of eight. Development MAE selects the epoch, with patience five and at most 30 epochs; the primary model selects epoch eight.

C.3 Engine score

Stockfish 18 (The Stockfish Team, 2026) evaluates at most 20 focal-player positions per game after the first eight plies, using 100,000 search nodes. Top-match measures agreement with Stockfish's preferred move. Its window score is the top-match fraction, with larger values indicating more engine agreement.

Our Stockfish analysis is separate from the tournament assistance plugin. A logged assisted move need not match the top choice of this analysis.

C.4 STANDARDIZING RATING GAPS

Rating errors can vary with playing level and time control. For Maia-3, we group development accounts using rating-band boundaries of 1200, 1600, 2000 and 2400, separately by time control. Within each group, we estimate an account-balanced mean and population standard deviation of the rating gap. For a window i in that group, these are μ_i and σ_i in Equation 2.

If a group has fewer than 20 accounts, we use the time-control group; if that is also too small, we use the full development set. The standard deviation is floored at one Elo. Maia-3 uses 918 development accounts for this calculation. The reported external backbone and Maia-2 comparison uses raw rating gaps.

C.5 SCREENING PROTOCOLS AND METRICS

Table C.2 separates the score, reference and decision rule for each primary screening experiment. Detection rate (directional power) is the fraction of intervention windows alerting in the intended direction. False-alert rate is the fraction of unchanged generated test windows alerting in either direction. Review burden is the fraction of public reference windows flagged by the specified rule; their assistance status is unverified.

Table C.2: Primary screening protocols. Calibration and test counts refer to windows. For ChessFraud, 44 assisted windows are scored alongside the listed public test windows; the paired 13-participant analysis is separate.

Experiment	Score / reference	Cal.	Test	Tail levels
Generated	Raw gap / generated controls	498	249	2.5% each
Human donor	Standardized gap / near-rating donors	88	26 up; 31 down	2.5% each
ChessFraud	Standardized gap / August	890	897 public	5% upward
ChessFraud engine	Top-match / August	898	900 public	5% upward

C.6 PROOF OF REFERENCE-RELATIVE ALERT CONTROL

Proof of Proposition 1. Fix a direction $d \in \{\uparrow, \downarrow\}$ and condition on the scorer. Exchangeability makes the rank of the new score among the $n+1$ scores uniform after independent uniform tie-breaking. Let R_d be the new score's rank, ordered from largest to smallest for upward tests and from smallest to largest for downward tests. The tie-conservative p-value in Equation 3 obeys $p_i^d \geq R_d/(n+1)$. For any $u \in [0,1]$, where u is a candidate one-tail level,

$$\Pr(p_i^d \leq u) \leq \frac{\lfloor (n+1)u \rfloor}{n+1} \leq u.$$

Applying this bound at $u = \alpha_d$ gives the directional claim. The union bound then gives probability at most $\alpha_\uparrow + \alpha_\downarrow = \alpha$ for either alert. The two directional p-values need not be independent. All probabilities average over the calibration sample and new score, conditional on the fixed scorer, with directional levels held fixed independently of calibration and test scores. Random tie-breaking is used only in the proof; the implemented p-values count ties conservatively. □

This marginal result does not bound alert probability conditional on a realized calibration sample, for a particular account, or within every rating or time-control subgroup. Group standardization changes the score; it does not provide group-conditional validity.

C.7 CALIBRATION SIZE AND ATTAINABLE THRESHOLDS

Conformal p-values lie on a discrete grid determined by calibration size. Let n be the number of usable calibration windows and α_{tail} the requested one-tail level. Define $k = \lfloor \alpha_{\text{tail}}(n+1) \rfloor$, the largest permitted rank. The attainable threshold is $k/(n+1)$. When $k = 0$, even the smallest p-value exceeds the requested level; ties can reduce the number of alerts further.

With $n = 498$ generated calibration windows, total bidirectional levels of 1%, 2.5%, 5%, 10% and 20% correspond to per-tail ranks $k = 2, 6, 12, 24, 49$. The primary rule therefore uses the attainable

threshold $12/499$ for each tail, just below 2.5%. For Maia-3 with August calibration, $n = 890$: upward levels of 5% and 10% use $44/891$ and $89/891$. For raw Stockfish top-match, $n = 898$ gives $44/899$ and $89/899$. For donor screening at 2.5% per tail, $n = 88$ gives $2/89$; after publicBOT exclusion, $n = 86$ gives $2/87$. These thresholds determine which p-values trigger alerts. Observed control and reference-alert fractions can exceed the nominal level in a finite evaluation sample without contradicting the marginal bound.

This finite resolution also explains our choice of external references for ChessFraud. A reference built only from ChessFraud must exclude the query player and any window sharing a physical game, without replacement. Only 8–13 calibration windows remain. Even at $n = 13$, the smallest p-value is $1/14$, too large for either upward 5% or bidirectional total 5% screening. We therefore use the natural and August reference populations.

C.8 Bootstrap uncertainty

We separate uncertainty in the test sample from uncertainty in the calibration sample. Test-account bootstraps keep the fitted estimators and calibration reference fixed. Calibration bootstraps instead keep test predictions fixed and recompute alerts after resampling the reference. Each simulation and external analysis uses 2,000 draws. Paired comparisons resample the same participants together.

These intervals describe sampling variation in the observed accounts. They do not include retraining, new simulation seeds or shared-opponent dependence. Only accounts with valid calibration scores enter the ranks; missing scores are not imputed.

D Rating Prediction on Natural Games

Table D.1 compares the nine reported move-based estimators on the same 197 accounts. Direct likelihood and Maia-2 require no project-specific fit; the remaining estimators use 880 fitting accounts. We report mean absolute error (MAE), root mean squared error (RMSE), signed bias, and the 95th percentile of absolute error (p95), all in Elo. Bias is predicted minus recorded rating.

Table D.1: Rating prediction on 197 held-out accounts. Brackets show 95% intervals from 2,000 test-account bootstrap draws, with fitted estimators fixed. RMSE and p95 are point estimates. This table evaluates rating accuracy before applying an anomaly threshold.

Method	MAE [95% interval]	Bias [95% interval]	RMSE	p95
Direct likelihood	165.7 [143.2,188.5]	45.3 [15.6,76.3]	230.7	447.2
Rating-calibrated	112.2 [94.9,132.0]	0.5 [-24.8,24.3]	175.1	286.6
Neural residual	92.8 [78.5,109.0]	-10.2 [-31.2,8.4]	140.1	252.4
Maia-2 (both players)	133.6 [115.0,153.2]	-2.4 [-29.0,23.4]	191.9	363.1
Maia-2 (focal moves)	169.8 [147.2,193.2]	-8.3 [-42.6,24.3]	236.0	480.9
Karvonen + Ridge	145.4 [126.0,166.5]	-9.3 [-39.4,18.4]	204.5	387.4
Pretrained Transformer + Ridge	194.3 [171.6,217.7]	-9.1 [-44.8,26.0]	251.9	465.7
Pretrained CNN + Ridge	202.2 [178.8,227.1]	-19.0 [-57.0,19.5]	265.0	517.6
Scratch CNN	256.5 [229.6,284.3]	-35.2 [-80.2,11.2]	327.2	672.7

Neural residual has lower MAE than Direct likelihood and Rating-calibrated. The paired MAE differences, defined as comparator minus Neural residual, are 72.9 [56.7,90.0] and 19.4 [10.4,28.7] Elo, respectively.

The main text focuses on three estimation approaches selected after exploratory comparisons; Table D.2 reports four additional Maia-3 estimators evaluated on the same windows. Gradient boosting combines compact likelihood features with hidden-state features. Hidden linear fits Ridge regression to hidden-state features. The compact-feature network uses likelihood summaries, while Evidence-hidden fusion combines hidden states with likelihood evidence. Paired MAE differences from Neural residual are 4.3 [-4.6,12.5] Elo for Gradient boosting and 2.9 [-5.9,11.0] for Evidence-hidden fusion; both intervals include zero.

Table D.3 gives the constant and metadata comparators on the same test windows. Mean opponent rating has MAE 39.0 Elo, consistent with matchmaking among similarly rated accounts. With op-

Table D.2: Additional Maia-3 estimators from exploratory comparisons. All use 880 fitting accounts and the same 197 test windows as Table D.1. Intervals use test-account bootstrap resampling with fitted models fixed. MAE is in Elo.

Estimator	MAE	95% CI
Gradient boosting	97.1	[83.3,112.3]
Hidden linear	108.1	[94.1,125.4]
Compact-feature network	111.4	[96.6,128.7]
Evidence-hidden fusion	95.7	[84.6,108.5]

ponents held fixed, it cannot respond to changes in the focal player's moves. Bootstrap intervals and opponent-rating bias are not reported for these comparators.

Table D.3: Constant and metadata comparators on the same 197 test windows. MAE and bias are in Elo; bias is predicted minus recorded rating. Fit counts project fitting accounts. NR means not reported.

Method / input	Fit	MAE	Bias
Training-mean constant	880	344.5	-12.5
Training-median constant	880	344.7	-34.6
Mean opponent rating (privileged)	0	39.0	NR

E CONTROLLED CHANGES IN PLAYING BEHAVIOR

E.1 CONSTRUCTING MATCHED INTERVENTIONS

The simulation changes the focal player's policy while keeping the reference rating and opponent policy fixed. Both players continue legally from the same historical opening. Matching controls and interventions by source account and game lets us compare alerts under a known change in the generation policy.

Generation retains the first eight plies, then rolls out both sides with Maia-3 for at most 160 plies. A ply is one player's move; games truncated at this limit remain in the analysis. The grid contains a control, policy-conditioning offsets of $\pm 200, \pm 400, \pm 600, \pm 800$ Elo, and engine-assistance assignment probabilities of 10%, 30%, 50% or 80% in 1, 2, 5 or 10 games. Offsets are generator inputs, not measured human skill changes. The 500 selected calibration accounts each supply one control window; 250 selected test accounts each supply 25 conditions, giving 6,750 planned windows. The common scored panel contains 498 calibration and 249 test accounts (Appendix A.2).

Engine assistance uses Stockfish 18 with 100,000 search nodes per recommendation, the same engine and node budget used for top-match scoring. We first generate complete assisted games at each assignment probability. We then combine them with their matched control games to form windows containing 1, 2, 5 or 10 assistance-enabled games. Histories are not spliced after the two games diverge. An assigned engine move can match the human-policy move. An assistance-enabled game may therefore contain no changed move and is still included in its assigned condition. All conditions use one generation seed; bootstrap resampling measures a separate source of variation. Maia-3 supplies both the continuation policy and Direct's likelihoods. The generated comparison measures sensitivity within this construction; the human-data experiments assess behavior from other sources.

E.2 DETECTION ACROSS INTERVENTION SIZES

Table E.2 gives the rating shifts and full assistance grid. Each cell separates alerts in the intended direction from alerts in the opposite direction. This distinction is useful for small interventions, where an alert can reflect ordinary variation rather than the assigned change.

For Direct likelihood, Rating-calibrated and Neural residual, upward control counts are 5/249, 8/249 and 9/249. These are the relevant comparisons for upward assistance detection. Any-direction control counts are larger—12/249, 15/249 and 14/249—because they also include downward alerts. Table E.1 reports the control counts underlying the nominal-level check in Figure 2A.

Table E.1: Any-direction alerts in 249 unchanged generated test windows. Each nominal total level is divided equally between the two tails; 498 separate generated windows calibrate each estimator.

Estimator	1%	2.5%	5%	10%	20%
Direct likelihood	3	6	12	32	57
Rating-calibrated	0	6	15	31	56
Neural residual	0	8	14	27	55

Table E.3 gives uncertainty for the main simulation conditions. Its two columns use the separate test and calibration bootstraps described in Appendix C.8.

Table E.2: Rating shifts and engine-assistance conditions. Entries count alerts among the same 249 test windows. Intended and opposite refer to the intervention's direction; all assistance conditions are upward. Scores are raw rating gaps, calibrated on 498 generated controls at total 5% (2.5% per tail). No window triggers both tails. Assistance percentages are assignment probabilities; game counts indicate assistance-enabled games, not games with a confirmed changed move.

	Direct likelihood		Rating-calibrated		Neural residual	
Condition	Intended	Opposite	Intended	Opposite	Intended	Opposite
+200 / ten games	159	0	56	2	33	2
−200 / ten games	222	0	64	0	26	3
+400 / ten games	245	0	189	0	121	1
−400 / ten games	235	0	190	1	71	4
+600 / ten games	245	0	233	0	192	2
−600 / ten games	235	0	206	7	132	9
+800 / ten games	245	0	237	0	225	3
−800 / ten games	235	0	200	8	136	12
10% / 1 game	7	7	8	6	12	4
10% / 2 games	8	6	11	6	10	4
10% / 5 games	9	2	10	7	10	4
10% / 10 games	23	1	16	6	12	3
30% / 1 game	7	6	12	8	9	4
30% / 2 games	10	5	12	7	6	3
30% / 5 games	35	2	25	7	12	3
30% / 10 games	137	0	52	3	46	2
50% / 1 game	10	3	13	7	9	5
50% / 2 games	17	1	14	6	11	3
50% / 5 games	84	0	41	6	31	2
50% / 10 games	227	0	121	1	112	2
80% / 1 game	10	4	13	6	10	2
80% / 2 games	32	2	23	6	22	3
80% / 5 games	148	0	66	4	76	4
80% / 10 games	245	0	216	1	220	3

Table E.3: Counts and uncertainty for the primary simulation results. Intervals are percentages. Controls use any-direction alerts; interventions use intended-direction detections. Raw rating gaps are evaluated at total 5%, with 498 calibration and 249 test accounts. Each column summarizes 2,000 resamples of the indicated population. The ranges do not account for additional generation seeds or opponent dependence.

Method	Condition	Alerts /249	Test 95% CI	Cal. 95% range
Direct likelihood	Control	12	[2.4,8.0]	[3.6,9.6]
	+400	245	[96.8,99.6]	[96.8,99.2]
	−400	235	[91.6,97.2]	[92.8,96.0]
	+600	245	[96.8,99.6]	[97.6,99.2]
	−600	235	[91.6,97.2]	[92.8,96.0]
	+800	245	[96.8,99.6]	[97.6,99.2]
	−800	235	[91.6,97.2]	[92.8,96.0]
	50% / 10 games	227	[87.6,94.4]	[88.8,96.4]
	80% / 10 games	245	[96.8,99.6]	[97.6,99.2]
Rating-calibrated	Control	15	[3.2,9.2]	[4.0,8.8]
	+400	189	[70.7,81.1]	[59.8,84.7]
	−400	190	[70.7,81.5]	[70.3,80.7]
	+600	233	[90.4,96.4]	[90.8,94.4]
	−600	206	[77.5,87.1]	[80.7,84.3]
	+800	237	[92.4,98.0]	[92.8,96.0]
	−800	200	[75.1,85.1]	[78.7,81.1]
	50% / 10 games	121	[42.2,55.0]	[36.9,53.4]
	80% / 10 games	216	[82.3,90.8]	[81.9,88.0]
Neural residual	Control	14	[2.8,8.8]	[4.0,8.4]
	+400	121	[42.6,54.6]	[37.3,61.0]
	−400	71	[22.9,34.1]	[23.7,45.0]
	+600	192	[71.9,82.3]	[69.5,81.1]
	−600	132	[46.6,59.0]	[49.0,63.1]
	+800	225	[86.7,93.6]	[85.5,91.2]
	−800	136	[48.2,60.6]	[51.4,59.8]
	50% / 10 games	112	[39.0,51.4]	[38.6,49.8]
	80% / 10 games	220	[84.3,92.0]	[84.3,90.0]

F ENGINE-SCORE CHECKS AND HUMAN-DONOR SUBSTITUTION

Stockfish measures move agreement without using Maia-3 likelihoods. Human-donor substitution tests rating–behavior mismatches using recorded human games. These provide two complementary checks of the generated results.

F.1 STOCKFISH SCORING OF GENERATED PLAY

With 499 calibration controls and 250 test windows, Stockfish top-match flags 7 controls at total 5%. Intended-direction detections are 70 for $+800$, 152 for -800, and 250 for 80% assistance across ten games. The policy-shift results show changes visible to an engine-based measure. The assistance condition is a same-engine positive control; it does not provide independent evidence of real-world assistance detection.

F.2 SUBSTITUTING REAL HUMAN PLAY

This experiment assigns a complete ten-game human donor window to a recipient's recorded-rating reference. Calibration uses near-zero-gap substitutions; alternatives use higher- or lower-rated donors.

Matching and selection. Matching uses exact agreement on recorded month, speed category and number of games played as White. A control donor's recorded window rating must be within 50 Elo of the recipient reference. The alternative's donor-minus-recipient gap must be within 50 Elo of the requested signed offset. Matching uses no game outcomes, predicted Elo, screening scores or opponent ratings. Substitution changes both focal behavior and opponent context, both of which enter the estimators (Appendix B).

Fitting-exposed accounts and games are excluded, and calibration/test roles remain separate. Recipient, control donor and alternative donor have distinct accounts and games. Donors are used without replacement within each offset, but can recur across offsets.

Of 200 eligible calibration and 197 eligible test recipients, selection with seed 20260916 retains 128 and 96 before matching, without refilling unmatched cases. Near-zero controls are found for 88 calibration recipients. Table F.1 summarizes matching and detection.

Screening and dependence. Standardized gaps use 88 calibration accounts and the conformal rank rule of Section 3, with total 5% screening. Shared-account and shared-game links connect the test windows into one dependency component, so detection and control counts are descriptive.

Table F.1: Human-donor matching and screening. Panel A starts from 96 preselected recipients per offset. Gaps are donor minus recipient recorded rating, summarized before scoring exclusions; brackets give the observed range. Panel B uses standardized gaps, 88 calibration windows and total 5%. Intervention counts are in the intended direction; controls count either direction. Shared donors and games make the test windows dependent.

A. Matching and usable test windows

Nominal offset	Matched / unmatched	Analyzed	Mean gap [range], Elo
$+400$	27 / 69	26	$386.5\ [352.5, 426.5]$
-400	32 / 64	31	$-392.4\ [-450.0, -354.5]$

B. Detection and matched controls

Method	$+400$/26	Control/26	-400/31	Control/31
Direct likelihood	15	0	17	0
Rating-calibrated	21	0	23	0
Neural residual	21	0	26	0

G LOGGED ASSISTANCE IN CHESSFRAUD

G.1 BEHAVIORAL RESPONSE AND ASSISTED-WINDOW DETECTION

Logged assistance increases mean Behavioral Elo in the paired cohort, while most of these windows remain below the upward alert threshold (Table G.1). The same 13 participants contribute windows with 0, 1, 2 and 4 assisted games. Each window contains ten games. The primary response compares four assisted games with unaided play. The separate 44-window cohort contains 2–8 assisted games per window and measures detection across more participants. These cohorts overlap. Assistance labels determine window construction and evaluation, and do not enter the predictors.

Table G.1: Response and alerts for the same 13 paired participants. Response is four-assisted-game minus unaided Behavioral Elo, with a paired 95% bootstrap interval. Alert counts use standardized gaps, August calibration and the upward-5% rule. Each alert column has 13 windows.

Method	Mean (Elo)	95% interval	Unaided	Four assisted
Direct likelihood	+79.6	[37.1,123.9]	0	0
Rating-calibrated	+83.9	[40.8,131.6]	1	0
Neural residual	+106.4	[52.9,171.6]	1	2

G.2 CHOICE OF CALIBRATION REFERENCE

We compare the original natural calibration with an August 2026 public Lichess reference, holding ChessFraud windows and predictions fixed. The August reference initially selects 1,000 calibration and 1,000 test accounts, each with a ten-game window at the same five-minute time control as ChessFraud. Tables A.2 and A.3 report selected and usable cohorts. Public alerts measure review burden. The validity guarantee applies to exchangeable reference windows; transfer to ChessFraud is assessed empirically. The external upward-only and level-sensitivity analyses are retrospective; ChessFraud outcomes do not select the thresholds.

Table G.2 separates detection, unaided-window alerts and public review burden under each reference. Figure G.1 shows how reference choice affects the fixed 44-window predictions, including additional estimators.

Table G.2: Upward-5% alerts under natural and August calibration, with predictions and test windows fixed. The 44-window and paired 13-participant cohorts overlap. Public columns report alerts / reference windows. Calibration sizes are 1,000/890 (natural/August) for behavioral scores and 999/898 for Stockfish.

Method	Reference	Assisted /44	Unaided /13	Dose 4 /13	Public
Standardized rating gaps					
Direct likelihood	Natural	9	0	0	39/897
	August	9	0	0	43/897
Rating-calibrated	Natural	16	0	0	24/897
	August	20	1	0	49/897
Neural residual	Natural	18	0	1	29/897
	August	24	1	2	52/897
Native engine score					
Stockfish top-match	Natural	26	3	8	96/900
	August	12	1	5	51/900

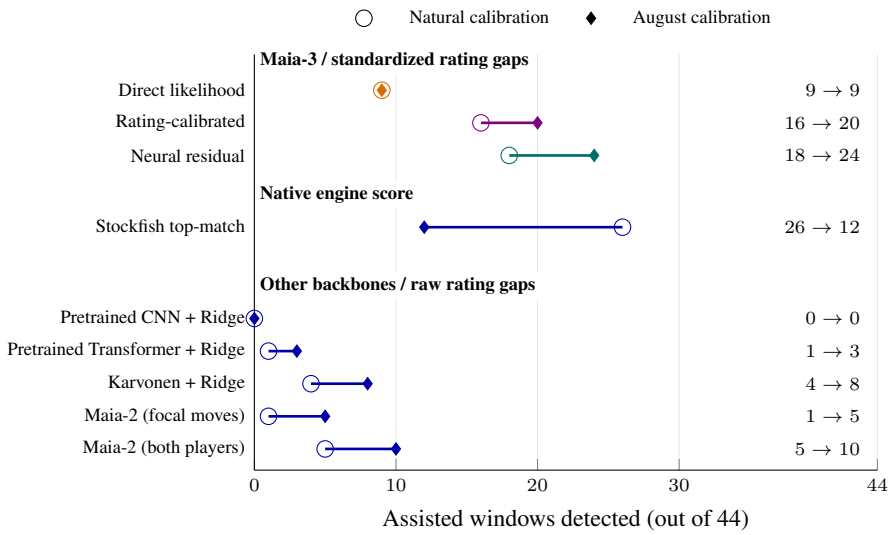

Figure G.1: ChessFraud detections in the same 44 assisted windows at upward one-sided 5%. Open circles and filled diamonds use natural and August calibration, respectively; lines connect these counts, not uncertainty bounds. Predictions remain fixed. Groups have different scores and calibration sizes (Section 3.4). Tables G.2 and G.5 give detailed results; Figure 2 shows reference-alert burden.

G.3 Sensitivity to game selection

Positive mean response persists across repeated choices of games. We repeat the original paired-window construction twenty times to examine sensitivity to which games are selected.

For each participant, we average the response across the original and twenty additional selections, then summarize these participant averages. The resulting mean responses and 95% intervals are +57.1 [30.6,88.9], +69.7 [29.3,112.4] and +94.2 [49.5,145.5] Elo. The numbers of participants with a positive average response are 11, 12 and 12 out of 13. These results show a positive mean response after averaging over game selection. They complement the original fixed-set analysis, which uses one specified selection per participant. Repeated selections do not add independent participants.

G.4 Detection and review burden at different thresholds

Table G.3 reports upward alerts over the full nominal level grid. The assisted cohort, paired participants and public reference accounts are held fixed as the threshold changes. This makes the increase in assisted-window detections directly comparable with the increase in review burden within each method. At nominal 20%, Stockfish detects 28/44 assisted windows and Neural residual 27/44, with public review burdens of 181/900 and 184/897, respectively. The detection advantage for Neural residual at 5% does not persist at this higher operating level.

Table G.4 reports sampling uncertainty at upward 5%.

Table G.3: Upward alerts with August calibration. Columns give one-sided nominal levels; each cohort is unchanged across columns. The first three estimators use standardized gaps with 890 calibration and 897 public test accounts. Stockfish uses raw top-match with 898 calibration and 900 public test accounts. Dose 4 denotes four assisted games in a paired ten-game window.

Method	Cohort	1%	2.5%	5%	10%	20%
Direct likelihood	Assisted /44	7	8	9	13	16
	Unaided /13	0	0	0	0	1
	Dose 4 /13	0	0	0	2	3
	Public /897	15	32	43	105	210
Rating-calibrated	Assisted /44	15	17	20	24	27
	Unaided /13	0	0	1	1	1
	Dose 4 /13	0	0	0	2	5
	Public /897	21	36	49	96	185
Neural residual	Assisted /44	18	20	24	26	27
	Unaided /13	0	0	1	1	2
	Dose 4 /13	1	1	2	4	5
	Public /897	26	34	52	87	184
Stockfish top-match	Assisted /44	4	8	12	23	28
	Unaided /13	0	0	1	2	7
	Dose 4 /13	2	4	5	8	10
	Public /900	8	26	51	82	181

Table G.4: Upward-5% detection uncertainty in the same 44 assisted windows, reported as percentages. Test intervals resample participants with calibration fixed; calibration ranges resample references with test windows fixed, each using 2,000 draws. For natural/August calibration, sample sizes are 1,000/890 for standardized Maia-3 gaps and 999/898 for raw Stockfish top-match. Shared-game dependence is not adjusted.

Method	Calibration	Alerts /44	Test 95% CI	Cal. 95% range
Direct likelihood	Natural	9	[9.1,31.8]	[18.2,20.5]
	August	9	[9.1,31.8]	[20.5,22.7]
Rating-calibrated	Natural	16	[22.7,50.0]	[34.1,38.6]
	August	20	[31.8,59.1]	[43.2,50.0]
Neural residual	Natural	18	[25.0,54.6]	[36.4,45.5]
	August	24	[38.6,68.2]	[45.5,54.5]
Stockfish top-match	Natural	26	[43.2,72.7]	[47.7,59.1]
	August	12	[13.6,40.9]	[22.7,38.6]

G.5 Additional estimators and comparison scope

Table G.5 compares five additional estimators on the same 44 assisted windows using raw rating gaps. It reports both total-5% bidirectional screening and upward-only 5% screening. The upward-only rule assigns the full nominal level to one tail, so it can flag windows that do not cross the bidirectional threshold.

In the 13-player paired cohort, none of these five estimators produces an upward alert at any assisted dose under the bidirectional rule (2.5% per tail) with August calibration. A downward alert on an assisted window indicates weaker-than-rated play under that score, not evidence of intentional underperformance.

Table G.5: Additional estimators on 44 assisted windows. All use raw rating gaps, with 992 natural and 888 August calibration accounts. Bidirectional columns use 2.5% per tail. The upward-5% column uses the separate one-sided rule. No window triggers both tails. Maia-3 standardized-gap results are reported separately in Table G.2.

		Bidirectional 5%		Upward 5%
System	Calibration	Up /44	Down /44	Up /44
Pretrained CNN + Ridge	Natural	0	10	0
	August	0	2	0
Pretrained Transformer + Ridge	Natural	0	9	1
	August	2	4	3
Karvonen + Ridge	Natural	2	3	4
	August	6	0	8
Maia-2 (focal moves)	Natural	1	3	1
	August	1	2	5
Maia-2 (both players)	Natural	3	6	5
	August	8	4	10

The raw-gap comparison and Maia-3 standardized-gap comparison use different score constructions. Their detection counts therefore describe the reported systems, rather than isolating backbone quality. The ChessFraud authors' move-level classifiers are outside this window-level comparison because they lack a matched aggregation and reference-calibration protocol.

H Public BOT Accounts and Exclusion Sensitivity

Public-BOT exclusion preserves the nine-estimator rating-MAE ordering and the positive mean ChessFraud response. Some calibration thresholds and individual alert decisions change, as detailed below.

Account identity and the screening target. CheckMate CP measures discrepancies between the playing strength expressed in observed moves and an account's recorded rating. This target is relevant to rating-based competitive matching. Public BOT status identifies an account type and does not by itself establish such a discrepancy. We retain the original platform-account cohorts as the primary analysis and examine sensitivity to public BOT exposure by excluding affected windows. The analysis refits project estimators and updates calibration while keeping pretrained backbones fixed.

Audit and window exclusion. We checked the original Lichess PGN headers, marking a game when either `WhiteTitle` or `BlackTitle` was `BOT`. Every natural evaluation game was recovered from its original monthly archive: 21,950 distinct games for April–July and 19,960 for the August reference. The latter supply 20,000 account-game views because 40 games appear from both players' perspectives; no physical game crosses August calibration and test. We also recovered all 2,592 physical source games in the donor analysis.

We excluded an entire ten-game window if any constituent game involved a public BOT, without replacement or selection by model score. Matched simulation controls and interventions were removed together according to their historical source window; the retained interventions remain

model-generated games. For donors, recipient, control and alternative windows were removed as a group, without rematching. Table H.1 gives the affected cohorts.

Table H.1: Public-BOT exposure by data role. Counts are ten-game windows except for donor calibration, which counts recipient pairs. The August rows describe selected windows before method-specific scoring exclusions.

Role	Original	Excluded	BOT identity in affected games
Project fitting	20,845	35	Opponent only
Natural development	918	1	Opponent only
Natural calibration	1,000	6	Focal account or opponent
Natural test	277	1	Focal account
August calibration	1,000	3	Focal account or opponent
August test	1,000	5	Focal account
Donor calibration pairs	88	2	BOT opponent or BOT recipient

In project fitting, 150 BOT-opponent games affected 35 windows across nine accounts; no focal fitting account was marked BOT. Exclusion leaves 878 accounts and 20,810 windows. Natural calibration and test each contained one focal BOT account, in contrast to the fitting-set exposure through opponents. The common rating-prediction test retains 196 of 197 accounts. Common simulation calibration/test support changes from 498/249 to 493/248. August Maia-3 support changes from 890/897 to 887/892, while Stockfish support changes from 898/900 to 895/895. Donor test support remains 26 and 31 recipients.

Refitting and evaluation. We first evaluated the original fitted models on retained windows. We then refitted Rating-calibrated, Neural residual, the three Ridge regressions and Scratch CNN using their original training and development-selection recipes. Direct and the Maia-2 likelihood estimators have no project-fitted parameters. The refitted analysis updates applicable predictions, development standardization and conformal calibration on the retained roles.

Table H.2: Natural rating-prediction MAE in Elo under public-BOT exclusion. The final two columns use the same 196 retained test accounts. Refitting updates project-trained components only; all pretrained backbones stay fixed.

Method	Original 197 accounts	Exclusion only 196 accounts	Exclusion + refit 196 accounts
Neural residual	92.8	91.1	92.7
Rating-calibrated	112.2	109.8	109.7
Maia-2 (both players)	133.6	133.7	133.7
Karvonen + Ridge	145.4	145.4	145.5
Direct likelihood	165.7	166.0	166.0
Maia-2 (focal moves)	169.8	170.1	170.1
Pretrained Transformer + Ridge	194.3	194.4	193.2
Pretrained CNN + Ridge	202.2	202.5	202.7
Scratch CNN	256.5	256.7	223.2

Rating prediction. The nine-method MAE ordering is unchanged (Table H.2). On the retained sample, Rating-calibrated minus Neural MAE is 17.0 Elo, with paired 95% bootstrap interval $[8.2, 26.2]$; Direct minus Neural is 73.3 Elo, with interval $[58.4, 90.6]$. The small total change for Neural combines opposing effects: removing the test BOT lowers MAE, while refitting increases retained-sample MAE by 1.6 Elo. Scratch's selected epoch changes from 8 to 30 under the original patience-five, maximum-30-epoch recipe. Its improvement includes training and selection variation and cannot be attributed solely to BOT exclusion. The reported 2,000-resample intervals describe test-sample uncertainty conditional on fitted models; they do not include repeated-training variability.

Screening and assistance response. Table H.3 compares the original and refitted screening results. With August calibration, Direct, Rating-calibrated and Neural detect 9, 20 and 23 of 44

assisted windows, compared with 9, 20 and 24 originally. The same 13 participants retain a positive mean Behavioral Elo response to four assisted games: Rating-calibrated changes from 83.9 to 84.0 Elo, and Neural from 106.4 to 106.9 Elo. Across the original selection and 20 resamplings, first averaged within each participant, Neural's mean response changes from 94.2 to 94.9 Elo, with 12 of 13 participants positive. These selections do not add independent participants.

Table H.3: Screening before and after public-BOT exclusion and refitting. Entries are alerts / windows. Generated detection counts the intended direction at total 5%; control alerts count either direction. ChessFraud uses upward 5% with August calibration; its public alerts measure review burden. "After" includes exclusion and refitting.

	Detection		Control / public alerts	
Method	Before	After	Before	After
Generated $+400$; raw rating gaps				
Direct likelihood	245/249	244/248	12/249	12/248
Rating-calibrated	189/249	190/248	15/249	16/248
Neural residual	121/249	119/248	14/249	13/248
Generated -400; raw rating gaps				
Direct likelihood	235/249	234/248	12/249	12/248
Rating-calibrated	190/249	192/248	15/249	16/248
Neural residual	71/249	84/248	14/249	13/248
ChessFraud; standardized rating gaps				
Direct likelihood	9/44	9/44	43/897	42/892
Rating-calibrated	20/44	20/44	49/897	48/892
Neural residual	24/44	23/44	52/897	51/892
ChessFraud; native engine score				
Stockfish top-match	12/44	12/44	51/900	46/895

The principal findings persist, but individual operating points can change. For example, Rating-calibrated detections for generated $+400$ at total 2.5% increase from 124/249 to 148/248 (49.8% to 59.7%). For Neural at total 5%, deleting affected test windows changes $+400$ detections from 121/249 to 120/248; excluding calibration windows increases them to 129/248; updating development standardization leaves that count unchanged; refitting gives 119/248. The upper threshold changes from 122.4 to 116.9 Elo after calibration exclusion and to 132.2 Elo after refitting. A small affected fraction can therefore change tail thresholds and several alert decisions.

Table H.4 gives donor results under the same conformal rank rule as Table F.1. Exclusion reduces calibration windows from 88 to 86 without changing test recipients. Neural's $+400$ controls include two alerts after exclusion and refitting.

Table H.4: Human-donor screening after public-BOT exclusion and refitting. The 86 retained calibration windows use standardized gaps and total 5%. Intervention counts are in the intended direction; control counts include either direction. Test windows are unchanged from Table F.1.

Method	$+400$/26	Control/26	-400/31	Control/31
Direct likelihood	15	0	17	0
Rating-calibrated	21	0	24	0
Neural residual	20	2	27	0

Scope. Public-title exclusion does not verify the absence of unmarked automation or assistance. It also does not reconstruct the training composition of the released Maia-3 checkpoint. This analysis measures sensitivity to observed BOT exposure in project data under fixed pretrained backbones. The reference-relative guarantee remains the one stated in Section 3.

I Sensitivity to Observation Window Length

We compare nested first-5, first-10 and first-20 windows on a common complete-20 cohort: 696 fitting, 717 development, 757 natural calibration and 209 test accounts. Rating-calibrated and Neural residual are fitted separately at each length. The natural alert panel uses 500 natural calibration accounts; the generated panel uses 500 separate generated controls. Every rule assigns 2.5% to each tail. This appendix's ten-game refit differs from the primary 880-account fit and its 249-account generated evaluation. The comparison covers these two Elo estimators and Stockfish top-match.

Accuracy uses the median recorded rating within each observed window. Elo alert scores instead subtract the rating before game 1, fixed across lengths. Table I.1 separates accuracy, natural alerts, and generated detection. All endpoints use the same 209 test accounts. The generated arms apply sustained policy offsets or nominal 80% engine assignment throughout the eligible continuation turns. Stockfish top-match uses the same engine and 100,000-node budget as assistance generation; its Engine80 result is a same-engine positive control.

Table I.1: Nested window-length comparison. MAE is in Elo; all alert counts are out of 209. Natural and generated controls count either-direction alerts; interventions count only the intended direction. Models and calibration are length-specific. Stockfish's direction refers to top-match fraction, not an Elo estimate.

Method	Quantity	5 games	10 games	20 games
A. MAE against two rating targets				
Rating-calibrated	Window median	120.89	99.20	91.81
Neural residual	Window median	137.03	99.32	81.51
Rating-calibrated	Fixed pregame	144.46	131.72	133.01
Neural residual	Fixed pregame	156.67	129.71	120.72
B. Natural reference alerts				
Rating-calibrated	Any	12	11	9
Neural residual	Any	11	9	8
Stockfish top-match	Any	24	10	18
C. Generated controls and interventions				
Rating-calibrated	Control: any	8	13	12
	+600: upward	131	129	97
	−600: downward	142	164	157
	Engine80: upward	122	111	84
Neural residual	Control: any	14	16	10
	+600: upward	74	63	20
	−600: downward	40	20	14
	Engine80: upward	189	169	156
Stockfish top-match	Control: any	16	9	22
	+600: upward	41	70	94
	−600: downward	77	89	109
	Engine80: upward	209	209	209

Reference and control alert rates vary across window lengths. At ten games, natural review burden is 5.3% (11/209) for Rating-calibrated and 4.3% (9/209) for Neural residual. Stockfish's natural review burden reaches 11.5% (24/209) at five games, and its generated-control false-alert rate is 10.5% (22/209) at twenty games. Neural residual's generated-control false-alert rate is 7.7% (16/209) at ten games. These are observed rates under the separate length-specific calibrations; they are not uniformly close to the nominal 5% level.

I.1 ACCURACY TARGETS AND ALERT BOUNDARIES

From ten to twenty games, MAE against the observed-window median falls by 7.39 Elo for Rating-calibrated (paired 95% interval: 1.74–13.24) and 17.81 for Neural residual (8.84–26.96). Against the fixed pregame rating, however, Rating-calibrated MAE changes from 131.72 to 133.01 Elo. Its paired difference is +1.30 [-4.22, 6.36]. Better prediction of the later observed rating therefore does not imply better agreement with the fixed alert reference.

Table I.2 shows how saved scores and their calibration boundaries change. The upper strict boundary is the score that a window must exceed to trigger the upward alert. Rating-calibrated's mean response changes little, while its boundary rises. Neural residual's mean response falls and its boundary rises. These aggregate changes accompany lower detection; they do not isolate the cause of each window's alert transition.

Table I.2: Score and boundary diagnostics for sustained $+600$ policy shifts on the common 209 test accounts. All entries are in Elo. Response is the mean paired intervention-minus-control score. Boundaries use separate length-specific calibration at 2.5% per tail.

Estimator	Quantity	10 games	20 games
Rating-calibrated	Mean response	268.79	272.32
	Mean intervention score	220.76	225.95
	Upper strict boundary	215.90	248.38
Neural residual	Mean response	198.45	176.50
	Upper strict boundary	247.34	318.70

Uncertainty uses 2,000 paired account-bootstrap draws, conditional on the fitted models. The alert contrast below holds calibration fixed; calibration-resampling uncertainty is assessed separately. From ten to twenty games, Rating-calibrated's $+600$ any-alert count falls by 32/209 (15.3 percentage points; paired 95% interval: 9.1–21.5). This interval concerns any-direction alerts, not correct-direction power. The resampling does not adjust for shared-opponent dependence. The cohort is restricted to complete-20 eligibility, and the three endpoints do not supply a sequential-monitoring guarantee.