PERA

Performance Evaluation Rating Algorithm

A Performance-Based Rating Framework for Competitive Sports

Abstract

PERA is a rating framework that measures competitive performance using point-level data instead of binary win/loss outcomes. Each match update is driven by the gap between a player's actual point share and the share expected from current ratings. The framework adds three components on top of this core: a recursive uncertainty estimator that tracks recent prediction error, a bounded adaptive K-factor that increases responsiveness when uncertainty is high without permitting runaway volatility, and a status ladder that exposes rating reliability to users.

This document specifies the system in full and provides justification for every design choice in a set of appendices. PERA is presented as a defensible, implementable framework with an explicit validation roadmap, not as an empirically calibrated final product. Sections 2–7 define the system. Appendices A–L explain why each choice was made, what alternatives were rejected, and which parameters require empirical calibration before deployment at scale.

1. Introduction and Philosophy

Most rating systems answer one question: did the player win? PERA replaces that question with a different one: how did the player perform compared to what was expected of them? The distinction is consequential. Under a binary outcome model, a 21–19 win and a 21–5 win are identical events. Under a point-based performance model, they are not. The same logic applies in reverse: a narrow loss is informationally distinct from a one-sided loss, and a rating system that treats them identically discards signal that exists in every match.

PERA is built on a single premise:

Performance relative to expectation is the central signal of competitive value.

Everything that follows — the score function, the expectation curve, the update rule, the uncertainty tracker, the adaptive K-factor, the doubles model, and the ecosystem-level recommendations — is a consequence of taking that premise seriously.

This document is organized so that the system itself is readable end-to-end without interruption. Each design choice is marked with a reference to the appendix that justifies it. Readers who want only the rating mechanics can stop at Section 7. Readers who want to challenge the design should read the appendices, which are where the alternatives, trade-offs, and known limitations are discussed.

2. The Rating System

2.1 Actual Performance Score

A player's actual performance in a match is defined as their share of total points played:

$$S = \frac{\text{points won}}{\text{total points played}}$$

Worked examples:

Match outcome	Score	Performance *S*
Narrow win	21–19, 19–21, 21–19	0.504
Dominant win	21–10, 21–12	0.656
Narrow loss	19–21, 21–23	0.476

S is bounded in [0, 1] and continuous. It captures information that is invisible to a win/loss flag: the closeness of the match, the dominance of the result, and the player's contribution to total points played.

Justification: see Appendix A.

2.2 Expected Performance

Expected point share is modeled with a logistic function on the rating gap:

$$E = \frac{1}{1 + 10^{\,\Delta/c}}, \quad \text{where } \Delta = R_{\text{opp}} - R$$

PERA uses *c* = 500 as a defensible prior. Interpretation:

- 0 rating difference → *E* = 0.500
- +100 rating → *E* ≈ 0.589
- +200 rating → *E* ≈ 0.671
- +400 rating → *E* ≈ 0.802

The choice of *c* = 500 (versus the chess Elo value of 400; Elo, 1978) produces a flatter expectation curve, reflecting the higher variability of badminton points relative to chess moves. This is a starting prior that should be replaced with an empirically calibrated value once sufficient match data exists.

Justification: see Appendix B.

2.3 Rating Update Rule

After each match, a player's rating is updated by:

$$R_{\text{new}} = R_{\text{old}} + K_n \cdot (S - E)$$

Three properties of this rule deserve emphasis:

- If $S = E$, rating does not change. The match contained no surprise.
- If $S > E$, rating rises. The player out-performed expectation, regardless of who won.
- If $S < E$, rating falls. The player under-performed expectation, regardless of who won.

A direct consequence: a player can win a match and still lose rating, or lose a match and still gain rating. This is intentional. It is also the design choice most likely to be questioned by users, and it must be communicated clearly in any deployment.

Justification: see Appendix C.

2.4 Uncertainty Tracking

Each player carries a recursive uncertainty estimate that measures how well their current rating is predicting their actual performance:

$$U_n = \alpha \cdot U_{n-1} + (1 - \alpha) \cdot (S - E)^2$$

With $\alpha = 0.85$, the recursion has the following properties:

- Each match contributes 15% of new information; previous estimate retains 85% weight.
- The effective memory length is approximately $1 / (1 - \alpha) \approx 6.7$ matches.
- Squared deviation amplifies large prediction errors and dampens small ones.

Initialization: U_0 is set to a moderate value (typically 0.04–0.06) for new players to ensure they begin in a responsive state. Bounds are imposed on U_n (see Section 2.5) to prevent the estimate from collapsing to zero or diverging due to long runs of mismatched opponents.

Justification: see Appendix D.

2.5 Bounded Dynamic K-Factor

The K-factor that scales each rating update is itself a function of uncertainty:

$$K_n = K_{\text{base}} \cdot \left(1 + \gamma \cdot \left(1 - e^{-a \cdot U_n}\right)\right)$$

Recommended parameter values:

Parameter	Value	Role
K_{base}	16	Base responsiveness for a stable, well-calibrated player
γ (gamma)	1	Hard cap on amplification: K can at most double
a	8	Sensitivity of K to uncertainty (steepness of the curve)

Behavior:

- Low uncertainty ($U_n \approx 0$): $K_n \approx K_{base} = 16$. Stable players move slowly.
- High uncertainty ($U_n \to \infty$): $K_n \to K_{base} \cdot (1 + \gamma) = 32$. K is capped at 2× the base regardless of how chaotic recent results have been.
- The $1 - e^{-a \cdot U_n}$ term is monotonic and bounded in [0, 1]. There is no value of U_n that produces unbounded K.

The boundedness is the key design point. Earlier formulations of adaptive K-factor in the literature use unbounded exponential forms (e.g., $K = K_0 \cdot e^{c \cdot U}$) which can spike under runs of mismatched opponents and produce explosive volatility. The PERA form preserves adaptive behavior while making instability mathematically impossible.

Bounds are also imposed on U_n itself: U_n is clamped to a working range (e.g., [0.005, 0.50]) to prevent floor effects in highly stable players and to prevent ceiling effects after pathological match sequences.

Justification: see Appendix E.

2.6 Starting Rating and Scale

All new players begin at $R_0 = 1500$ unless a deployment has data justifying a different prior. The absolute number is not mathematically meaningful; only rating differences enter any formula. 1500 is chosen for familiarity and for symmetry around a neutral midpoint, not for any computational property.

Optional descriptive bands may be displayed alongside the numeric rating:

Rating range	Descriptive label
Below 1500	Beginner / Developing
1500–1799	Intermediate
1800–2199	Advanced
2200+	Elite

The rule is unidirectional: bands are derived from rating, never the other way around. Performance defines position, not labels.

Justification: see Appendix F.

2.7 Status System

A rating number alone does not communicate how trustworthy it is. A 1700 rating after 5 matches is not the same as a 1700 rating after 100 matches. PERA exposes this through a status ladder derived from match volume M and uncertainty U_n:

Status	Criteria	Meaning
Provisional	$M < 12$	Insufficient evidence
Developing	$12 \leq M < 30$	Building evidence
Established	$M \geq 30$ and $U_n < 0.07$	Reliable rating
Stable	$M \geq 50$ and $U_n < 0.04$	High-confidence rating
Dynamic	$M \geq 30$ and $U_n \geq 0.07$	Evolving or mis-calibrated

Match count is the volume of evidence; uncertainty is the quality of evidence. Status combines both. A confidence percentage may be derived internally from M and U_n, but the user-facing primary signal is the discrete status label.

Justification: see Appendix G.

3. Format Handling

3.1 Multiple Rating Tracks

Singles and doubles require different skill profiles. PERA supports independent rating tracks per format. The number of tracks a deployment maintains depends on data volume:

- A deployment with limited data may run a single combined track.
- A typical deployment will run separate singles and doubles tracks.
- Mature deployments may add mixed-doubles or other format-specific tracks.

Rating-track decisions are an implementation choice, not a property of the framework. All tracks use the same underlying mechanics.

3.2 Doubles Rating

Team rating is the average of partner ratings:

$$R_{\text{team}} = \frac{R_1 + R_2}{2}$$

Expected team performance uses the same logistic form as singles:

$$E = \frac{1}{1 + 10^{\,\Delta/500}}, \quad \text{where } \Delta = R_{\text{opp\_team}} - R_{\text{team}}$$

Actual team performance is team point share:

$$S = \frac{\text{team points won}}{\text{total points played}}$$

The rating update applies a bounded rating-skewed split rather than an equal split. For partner *i*:

$$\Delta R_i = m_i \cdot K_n \cdot (S - E)$$

where the multiplier m_i is:

$$m_i = 1 + \text{clip}\left(\frac{R_i - R_{\text{partner}}}{2000},\ [-0.2,\ +0.2]\right)$$

with clip*(x, [a, b])* defined as the value of *x* clamped to the interval *[a, b]*. Each partner's individual K_n and U_n are tracked separately from their own uncertainty state.

The multiplier m_i is bounded in [0.8, 1.2]. Three cases illustrate the behavior:

- Equal partners ($R_1 = R_2$): $m_1 = m_2 = 1$. The update reduces to the team-level update applied equally to both partners.
- Moderate gap ($R_1 - R_2 = 200$): $m_1 = 1.05$, $m_2 = 0.95$. The higher-rated partner moves about 10% more than the lower-rated partner.
- Large gap ($R_1 - R_2 \geq 400$): the skew saturates at the cap. $m_1 = 1.2$, $m_2 = 0.8$ (60/40 split). The split does not widen further regardless of how large the partner gap becomes.

The split applies symmetrically to gains and losses: the higher-rated partner absorbs a larger share of team surprises in either direction. This reflects the intuition that the stronger player carries more responsibility for team outcomes, while the bound prevents that intuition from becoming a runaway feedback loop. When $S = E$ (no surprise), no rating change occurs regardless of the partner gap.

Partner-switching: in environments with frequent partner rotation (social leagues, ladder play), individual ratings will exhibit additional noise from partner variance. This noise is real, not a

system failure, and is captured automatically by the uncertainty term U_n. Deployments with very high partner-rotation rates may want to use a higher α (longer memory) on the doubles track.

Justification: see Appendix H.

4. System Behavior

The interaction between S, E, U_n, and K_n produces three distinct steady-state behaviors that the system enters automatically without manual intervention.

Player profile	S vs E pattern	U_n behavior	K_n behavior
Stable	$S \approx E$	Low, decaying	Near 16
Improving	$S > E$ consistently	Rising	Toward 32
Declining	$S < E$ consistently	Rising	Toward 32

No explicit inactivity penalty is required: a returning player whose form has shifted will produce $S \neq E$ in their first matches back, U_n will rise, and K_n will scale up automatically until the rating re-converges. The system is self-correcting in the small.

5. Path Dependence and Historical Recalculation

PERA is path-dependent: a player's rating after n matches depends on the order in which those matches were processed. This is mathematically unavoidable in any incremental rating system, and it has two practical consequences.

5.1 Recalculation on backfill

When historical match data is added after-the-fact (for example, a club uploads a year of records that pre-date system adoption), ratings must be recomputed from the earliest available match forward. The procedure is:

- Reset all affected players to initial state (R_0, U_0).
- Sort all matches by canonical timestamp.
- Reprocess matches in chronological order.
- Recompute R, U_n, K_n, and status for each player.
- Publish a versioned rating snapshot with the recomputation date.

Full recomputation is $O(n)$ in the number of matches. For a single club this is trivial. For a federation-scale deployment with millions of matches, full recomputation on every backfill is operationally expensive. PERA addresses this with two mechanisms:

- Checkpointing: rating snapshots are stored at the end of each calendar period (week or month). Backfilled matches before checkpoint T_k require recomputation only from T_k forward, not from the system epoch.
- Player-scoped recomputation: when only a small set of players is affected by a backfill, recomputation can be limited to that set if the backfilled matches do not connect to the broader player graph during the recomputation window.

5.2 Canonical match ordering

When two matches occur at the same timestamp at different venues, the rating outcome can depend on which is processed first. PERA resolves this with a canonical ordering rule, applied in priority order:

- Primary key: match start timestamp (UTC, second precision).
- Tiebreaker 1: match identifier hash (deterministic, content-derived).
- Tiebreaker 2: lexicographic ordering of venue identifier.

This rule makes the rating output reproducible from the underlying match log regardless of upload sequence.

Justification: see Appendix I.

6. Network Connectivity

A rating system is a graph problem: each match is an edge between two players, and rating values become globally comparable only when the player graph is sufficiently connected. In a fragmented ecosystem (Raleigh players never play California players), each cluster develops a locally valid rating scale, but cross-cluster comparison is unreliable.

PERA does not solve this with a formula — no rating algorithm can. It is a structural problem about the match network, and it requires structural responses. The framework specifies three:

6.1 Connectivity diagnostics

Each deployment computes and publishes connectivity statistics for its player graph: the largest connected component size, the diameter of that component, the number of bridge matches between identifiable clusters, and the ratio of intra-cluster to inter-cluster matches. These statistics are exposed alongside ratings so users can interpret cross-cluster comparisons appropriately.

6.2 Cluster-aware confidence

When a player's match history is contained entirely within a small or weakly connected cluster, their status is capped at Established (not Stable) regardless of M and U_n. This prevents a 100-

match rating built entirely from a 12-player club from being treated as equivalent to a 100-match rating built from cross-cluster play.

6.3 Structured cross-cluster interaction

Beyond diagnostics, PERA recommends operational mechanisms that increase the edge-density of the player graph. These are non-elimination, multi-match formats specifically designed to maximize cross-cluster pairings rather than determine a single winner. The HmBr mela construct is one implementation of this idea, with the following properties:

- Predefined matchups across clusters.
- Multiple matches per participant.
- Cross-level pairings (mixing rating bands within bounded ranges).
- Non-elimination structure.

Mela events are calibration mechanisms, not ranking events. They feed the rating system but do not produce a separate ranking output. Match results from mela events are tagged in the match log for analytics, but the rating formula does not apply event-specific multipliers — opponent strength and actual performance already capture what matters.

Justification: see Appendix J.

7. Limitations and Validation Roadmap

PERA is presented as a defensible framework with explicit unknowns, not as a validated final product. The following limitations are acknowledged and define the work required for empirical maturity.

7.1 Parameters that require empirical calibration

Parameter	Default	Calibration method
c (expectation scale)	500	MLE fit of logistic to historical match data; binned point-share by rating gap
α (uncertainty memory)	0.85	Sensitivity analysis on out-of-sample log-loss across player activity profiles
K_{base}	16	Joint optimization with c against held-out predictive accuracy
γ, a (K curve)	1, 8	Stress-tested against pathological match sequences
D, cap (doubles split)	2000, ±0.2	Compared against equal-split and full-proportional baselines on doubles match data

Parameter	Default	Calibration method
Status thresholds	various	Calibrated against rating drift on simulated data

7.2 Required validation work before scaled deployment

- Hold-out predictive comparison against binary Elo (Elo, 1978), Glicko-2 (Glickman, 2013), and TrueSkill (Herbrich et al., 2007) on a single corpus of badminton match data, scored by log-loss and Brier score (Brier, 1950) on the next-match S value.
- Convergence-rate analysis: how many matches does a new player require before their rating stabilizes within a target band? How does this compare to the RD convergence in Glicko-2 (Glickman, 2013)?
- Sensitivity analysis on α and the K-factor parameters across different player activity regimes (recreational vs. tournament vs. junior).
- Network connectivity simulation: at what bridge-match density does cross-cluster rating drift fall below a target threshold?
- User-comprehension study on the “win-but-lose-rating” outcome and the status ladder.

7.3 Out-of-scope concerns

PERA does not currently model: in-game momentum effects, retirement or injury during a match (which corrupt S), opponent-specific style matchups, or doubles team chemistry as an entity distinct from the partners' individual ratings. Each is a defensible extension once the core has been validated.

Justification: see Appendix L.

8. Conclusion

PERA replaces outcome-based rating with performance-based rating by treating point share as the actual signal and the rating gap as the expected signal. The core update rule is a single line. The framework's real complexity lives in three places: the recursive uncertainty estimator, the bounded adaptive K-factor, and the structural treatment of network connectivity. Each is a deliberate response to a known failure mode of simpler systems.

The framework is intentionally conservative. Where a parameter is not yet empirically calibrated, it is presented as a defensible prior with a calibration path. Where a problem is structural (network connectivity, partner contribution), the framework names it explicitly and provides a recommended response rather than claiming a formula-only solution.

Outcome is an event. Performance is the signal.

Appendices

The following appendices provide the full justification for each design choice in the system. Each appendix follows the same structure: what the choice is, why it was made, what alternatives were considered, what assumptions it rests on, and what would invalidate it.

Appendix A. Justification for the Performance Score *S*

A.1 The choice

S = points won / total points played, computed across all games of a match.

A.2 Why this was chosen

Win/loss is a one-bit summary of a match. Across a season of badminton, a player who wins 30 matches at 21–19 and a player who wins 30 matches at 21–5 are indistinguishable under win/loss. Point share is the lowest-cost continuous signal that distinguishes them, and it is available from any complete scoreline.

A.3 Alternatives considered

- Win/loss only (*S* = 1 or 0). Rejected: discards within-match information.
- Game share (*S* = games won / games played). Rejected: too coarse in a best-of-3 format. A 21–19, 21–19 win and a 21–5, 21–5 win produce the same *S*.
- Margin-of-victory weighted scoring (e.g., point differential normalized). Considered. Equivalent to point share for two-player games up to a monotonic transform; point share is preferred for interpretability.
- Rally-level performance models (serve win rate, return win rate, etc.). Rejected at this stage: requires data that most deployments do not have.

A.4 Acknowledged limitations

Three limitations of *S*, all real but bounded in their impact, must be acknowledged. Each is addressed by other components of the framework rather than by modifying *S*.

(i) Points are not exchangeable.

A point at 20–20 in game 3 is not informationally equivalent to a point at 20–5 in game 1. A purist treatment would weight points by leverage. PERA does not, for two reasons. First, the data required (point-by-point sequence with context) is not available in most deployments. Second, over a sufficiently large sample of matches, the leverage-weighting differences average out: a player who consistently wins close points will also consistently win blowouts, and the ranking they produce under unweighted *S* will agree with the ranking they would produce under

leverage-weighted S to within the noise floor of either. The leverage-weighted variant is identified as a future extension (Section 7.3).

(ii) S conflates context.

A 21–5, 21–5 result could mean genuine dominance, an injured opponent, or a no-show substitute. PERA does not attempt to detect these cases at the score level. The uncertainty estimator U_n is the mechanism that handles them: a context-corrupted match produces an outsized ($S - E$) value, which inflates U_n, which raises K_n on subsequent matches, which causes the rating to re-correct as soon as normal-context matches resume. Operationally, deployments are encouraged to flag retirements and walkovers in match metadata so they can be excluded from rating updates entirely.

(iii) S is bounded and format-dependent.

In best-of-3 to 21, S is mathematically constrained: a player cannot post S above 0.840 (21–0, 21–0) or below 0.160 (0–21, 0–21). The expectation curve E maps to the same bounded interval, so the difference ($S - E$) is consistent within a single format. Cross-format mixing (rally-21 vs. rally-15 vs. best-of-5) does change the S distribution, and PERA addresses this by maintaining separate rating tracks per format (Section 3.1) rather than by modifying S.

A.5 What would invalidate this choice

Empirical evidence that point share has lower predictive validity than an alternative low-data signal (e.g., game share with score-margin weighting) on next-match outcomes. The validation roadmap (Section 7.2) tests this explicitly.

Appendix B. Justification for Logistic Expectation and c = 500

B.1 The choice

Expected point share is modeled as a logistic function of the rating gap, with scale constant c = 500.

B.2 Why logistic

The logistic curve has the right shape: it is monotonic in the rating gap, saturates at 0 and 1 (which matches the bounds of S), is symmetric around E = 0.5 at zero rating gap, and has been used in rating systems for over half a century without empirical pathology. Linear and lookup-table alternatives were rejected for reasons that follow.

B.3 Alternatives considered

- Linear expectation, $E = 0.5 + a \cdot (R - R_{opp})$. Rejected: produces values outside [0, 1] for large rating gaps and has no theoretical motivation.

- Normal-CDF expectation (Thurstone, 1927), used in Glicko (Glickman, 1999). Mathematically defensible and empirically very close to logistic; differences are small in practice. Logistic is preferred for closed-form simplicity and direct compatibility with the broad Elo literature.
- Empirical lookup table by rating-gap bucket. Rejected as a primary form: brittle, requires retraining as data accumulates, and obscures the model. An empirical table is, however, the right tool for validation — it is what c is calibrated against.

B.4 Why $c = 500$ (not 400)

The chess Elo value $c = 400$ (Elo, 1978) is calibrated for chess, where outcomes are deterministic given player ability and small differences in skill consistently produce wins. Badminton points are noisier: rallies have stochastic outcomes even between equal players, fatigue accumulates within matches, and a single bad serve can lose a game. A flatter curve (larger c) reflects this: rating differences imply a probabilistic edge, not near-certainty.

Numerical comparison at +200 rating gap:

c value	E at +200	Implied edge
400 (chess Elo)	0.760	Strong
500 (PERA)	0.671	Clear but soft
600	0.612	Weak

B.5 Acknowledged limitation: $c = 500$ is a prior, not a calibration

PERA does not currently provide an empirical fit of c against badminton match data. This is the single largest open methodological item in the framework. The calibration procedure is straightforward and is specified in Section 7.2: bin historical matches by rating gap, compute the empirical mean S in each bin, and fit c by maximum likelihood (or equivalently by minimizing log-loss). Until that fit is performed, $c = 500$ should be treated as a defensible starting value, not as the answer.

B.6 What would invalidate this choice

An empirical fit returning $c \ll 500$ would indicate badminton point outcomes are more deterministic than assumed. A fit returning $c \gg 500$ would indicate they are noisier. Either result is a reason to update the prior, not to abandon the logistic form.

Appendix C. Justification for the Update Rule

C.1 The choice

$R_{new} = R_{old} + K_n \cdot (S - E)$. The update is linear in the prediction error $(S - E)$.

C.2 Why this form

This is the Elo-class update rule (Elo, 1978), with two modifications: S is continuous in [0, 1] rather than binary in {0, 0.5, 1}, and K is a function of recent uncertainty rather than a fixed constant. The structural form (additive update proportional to prediction error) is the simplest update that satisfies three natural requirements:

- Zero surprise ($S = E$) produces no rating change.
- Positive surprise ($S > E$) raises the player's rating.
- The magnitude of the change scales with the magnitude of the surprise.

These properties together imply a form proportional to $(S - E)$. The constant of proportionality is K_n. Higher-order forms (quadratic in error, Bayesian posterior updates) were considered and rejected at this stage; their treatment is deferred to Appendix K.

C.3 The “win but lose rating” consequence

The most-questioned property of the rule is that it permits a player to win a match and lose rating, or vice versa. Worked example:

Player A rated 1800; opponent rated 1700; gap = +100.

E for A = $1 / (1 + 10^{-0.20}) \approx 0.613$.

If A wins 21–19, 19–21, 21–19, then A's points won are 61, total points are 119, and $S = 0.513$.

$(S - E) = -0.100$. With $K_n = 18$, $\Delta R = -1.8$. A wins the match and loses 1.8 rating points.

This is the correct behavior for a performance-based rating system. A player rated 100 points above their opponent is expected to dominate; a near-tie is an underperformance, and the system records that. The alternative — raising A's rating because A won — would mean rewarding a player for performing below their stated skill level, which is exactly what binary outcome systems do and what PERA is designed to avoid.

C.4 Adoption risk and mitigation

This property is the single largest social adoption risk in the framework. The system is mathematically correct from the user's perspective; it can be incomprehensible. Recommended mitigations for any deployment:

- Always show (S, E, ΔR) together in the post-match summary, not just ΔR.
- Provide a one-sentence explanation in plain language: “You won, but you were expected to win more decisively.”
- Show a small history of $(S - E)$ over the last several matches so users can see the trajectory of their performance vs. expectation, not just isolated snapshots.

- Reserve the descriptive bands (Section 2.6) for ratings of Established status or above, so provisional players are not labeled at the same time their rating is still bouncing.

C.5 What would invalidate this rule

Empirical evidence that a different update form (e.g., margin-weighted, or Bayesian-posterior) produces lower out-of-sample log-loss on next-match S predictions. This is the central comparison in the validation roadmap.

Appendix D. Justification for Uncertainty Tracking

D.1 The choice

$U_n = \alpha \cdot U_{n-1} + (1 - \alpha) \cdot (S - E)^2$, with $\alpha = 0.85$.

D.2 Why squared deviation

Squared deviation amplifies the contribution of large prediction errors and dampens the contribution of small ones. A small mismatch ($S - E = 0.05$) contributes 0.0025 to the recursion; a large mismatch ($S - E = 0.30$) contributes 0.09 — thirty-six times as much, not six times. This is the behavior we want: many small fluctuations should not destabilize a rating, but a single large surprise should change how confident we are in it.

Absolute deviation $|S - E|$ was considered and rejected for the same reason: it is too lenient on large mismatches relative to small ones.

D.3 Why $\alpha = 0.85$

The recursion is an exponentially-weighted moving average (Hunter, 1986). Its effective memory length is $1 / (1 - \alpha)$. For $\alpha = 0.85$, this gives ≈ 6.7 matches — long enough to suppress single-match noise, short enough to respond to a real shift in form within roughly two months of weekly play.

This is a defensible default, but it is a single value applied to all players, and player activity profiles vary. A recreational player who plays twice a month has a 6.7-match memory of approximately three months; a tournament player who plays 20 matches a week has a memory of approximately a third of a week. These are very different exposures to the same parameter.

D.4 Acknowledged limitation: α should be activity-aware

A more principled implementation would make α a function of expected match frequency or of calendar time rather than match index. Two implementation paths are available:

- Calendar-weighted recursion: replace α with $\alpha(\Delta t) = \exp(-\Delta t / \tau)$ where Δt is the time since the previous match and τ is a target memory horizon (e.g., 60 days). This preserves the EWMA structure but anchors memory in calendar time.

- Activity-tier α: classify players into activity tiers (low / medium / high match frequency) and use a per-tier α. Simpler operationally but coarser.

PERA recommends the calendar-weighted form for any deployment where match frequency varies by more than an order of magnitude across users. The fixed $\alpha = 0.85$ form is acceptable in tournament-scale or league-scale deployments where activity is roughly homogeneous.

D.5 Bounds on U_n

U_n is clamped to $[U_{min}, U_{max}]$ where $U_{min} = 0.005$ and $U_{max} = 0.50$. The lower bound prevents the K-factor from collapsing toward the base value during long stretches of well-predicted matches (a stable player who suddenly shifts form should still receive a meaningfully responsive K). The upper bound prevents pathological match sequences from inflating U_n indefinitely. Combined with the bounded K-factor (Appendix E), this gives the system two layers of stability protection.

D.6 What would invalidate this choice

A sensitivity analysis showing that out-of-sample predictive accuracy is significantly higher at $\alpha \neq 0.85$, or showing systematic bias in U_n across activity tiers under the fixed-α form. Both tests are part of the validation roadmap.

Appendix E. Justification for the Bounded Dynamic K-Factor

E.1 The choice

$K_n = K_{base} \cdot (1 + \gamma \cdot (1 - e^{-a \cdot U_n}))$, with $K_{base} = 16$, $\gamma = 1$, $a = 8$.

E.2 Why a bounded form

Adaptive K-factor systems in the rating literature (e.g., Glickman, 1999, 2013) have a well-documented failure mode under unbounded forms. Consider $K_n = K_{base} \cdot e^{c \cdot U_n}$: a player who experiences a run of unusual matches (e.g., several no-shows, or a sequence of severely mismatched opponents) accumulates a large U_n, which under the exponential form produces a large K, which then inflates the rating swing on the next match, which can in turn produce another large $(S - E)$ and inflate U_n further. The system is then in a positive-feedback regime where one anomaly begets another.

The PERA form prevents this by construction. The factor $(1 - e^{-a \cdot U_n})$ is bounded above by 1 for all $U_n \geq 0$, so K_n is bounded above by $K_{base} \cdot (1 + \gamma)$ regardless of how large U_n becomes. With $\gamma = 1$, $K_n \leq 32$. No match can move a player's rating by more than approximately 32 points, even in the most pathological scenarios.

E.3 Why these specific parameters

K_{base} = 16 is the chess Elo standard for established players, scaled appropriately for the PERA scale. With ($S - E$) typically in the range [−0.30, +0.30], this produces single-match swings of at most 4.8 points for stable players — meaningful but not volatile.

γ = 1 caps amplification at 2×. Higher values (e.g., γ = 2 giving a 3× cap) were considered and rejected as too aggressive: a 96-point swing on a single match would override the system's noise rejection.

a = 8 controls how quickly K rises with U_n. The half-amplification point ($K_n = K_{base} \cdot (1 + \gamma/2)$) occurs at $U_n = \ln(2) / a \approx 0.087$, which corresponds roughly to consistently mispredicted point shares of around ±0.30. This is calibrated so that a clearly mis-rated player gets noticeable K-amplification within five to ten matches, while a stable player whose U_n stays below 0.04 gets nearly no amplification.

E.4 Numerical sanity check

With the chosen parameters:

U_n	K_n	Player profile
0.005	16.6	Highly stable
0.04	20.6	Established
0.10	28.8	Dynamic / evolving
0.30	31.5	Highly mispredicted
0.50 (cap)	31.7	At U_n ceiling

K_n approaches its upper bound of 32 asymptotically, never crossing it. This is the key property the unbounded exponential form lacks.

E.5 Cold-start behavior

New players are initialized with U_0 = 0.04, placing K_n at ≈20.6 from the first match. This is intentional: provisional players need responsive ratings, but not so responsive that a single anomalous match can move them by 60+ points. After the first dozen matches, U_n reflects actual play and K_n settles into the appropriate range.

E.6 What would invalidate this choice

Stress testing on long match sequences showing systematic rating drift in either direction, or showing that the 16–32 K range is empirically too narrow (under-responsive) or too wide (over-responsive). Both are part of the validation roadmap.

Appendix F. Justification for Starting Rating and Scale

F.1 The choice

R_0 = 1500 for all new players.

F.2 Why this is mathematically arbitrary

Every formula in PERA depends on rating differences, never on absolute values. The expectation E depends on ($R_{opp} - R$), the update ΔR adds to R without referencing its magnitude, the uncertainty U_n is a function of ($S - E$), and the K-factor depends on U_n. None reference the absolute level of R.

Consequently, R_0 = 1500, R_0 = 0, R_0 = 5000, and R_0 = −1500 all produce identical match-by-match dynamics. The only thing R_0 changes is what number is printed on a player's profile.

F.3 Why 1500 specifically

1500 is chosen for three non-mathematical reasons: it matches the historical convention of chess Elo, it positions the rating bands around recognizable thresholds (1500, 1800, 2200 are familiar in both chess and table tennis communities), and it leaves room above and below for natural separation without any player ever needing to display a negative rating.

F.4 Cold-start prior with data

Deployments with prior knowledge about a player (e.g., a known tournament player joining a club system) may set R_0 above or below 1500. This is reflected by two operational rules: any non-default R_0 must be paired with a higher initial U_0 (typically 0.06–0.08) so the system is responsive enough to correct an inaccurate prior; and the prior should be drawn from a comparable rating system (e.g., national federation rating) and noted in match metadata.

F.5 What would invalidate this choice

Nothing about R_0 = 1500 is mathematically invalidate-able — it is a convention. The convention itself is invalidate-able only by user research showing a different starting number is significantly more legible to the target community.

Appendix G. Justification for the Status Ladder

G.1 The choice

Status ∈ {Provisional, Developing, Established, Stable, Dynamic}, derived from match count M and uncertainty U_n.

G.2 Why a discrete ladder

A continuous confidence percentage (e.g., “73% confident”) is mathematically defensible but operationally hostile. Users do not have intuitions for the difference between 71% and 73% confidence. They do have intuitions for the difference between “still being calibrated” and

“stable.” The ladder is a projection of the underlying continuous quantity onto a small set of human-readable categories.

G.3 Why both M and U_n

Match count and uncertainty answer different questions. M answers: how much evidence do we have? U_n answers: how consistent is the evidence? A 100-match rating with $U_n = 0.20$ is built on a lot of contradictory data; a 30-match rating with $U_n = 0.02$ is built on less data but very consistent data. Either alone is an incomplete signal. The ladder uses both:

- Provisional / Developing thresholds are M-only (volume gating).
- Established / Stable thresholds combine M and U_n (volume + consistency).
- Dynamic is the case where M is sufficient but U_n indicates instability.

G.4 Why these specific thresholds

The thresholds (M = 12, 30, 50; U_n = 0.07, 0.04) are calibrated so that under typical play a player progresses Provisional → Developing within roughly two months, Developing → Established within four to six months, and Established → Stable within a year of consistent play. They are starting values, intended to be tuned per deployment based on observed progression rates.

G.5 Acknowledged limitation: cluster-blindness

The ladder as defined treats all matches equally, regardless of who they were played against. A player whose 50 matches are all against the same five opponents in a small club has objectively less reliable rating evidence than a player whose 50 matches span thirty opponents across multiple clubs. Section 6.2 addresses this by capping status at Established for cluster-bounded players, regardless of M and U_n. This rule is part of the deployment configuration, not the formula.

G.6 What would invalidate this choice

Observation that the categorical distinctions do not predict any user-visible difference (e.g., that Stable and Established players have indistinguishable next-match outcome distributions). If so, the ladder collapses to fewer, broader categories.

Appendix H. Justification for the Doubles Model

H.1 The choice

Team rating is the average of partner ratings; team expectation E uses the same logistic form as singles; team performance S is team point share; and the rating update is a bounded rating-skewed split:

$$\Delta R_i = m_i \cdot K_n \cdot (S - E), \quad m_i \in [0.8, 1.2]$$

where $m_i = 1 + \text{clip}((R_i - R_{partner}) / 2000, [-0.2, +0.2])$. Equal partners produce equal updates ($m = 1$ for both). A 200-point partner gap produces a 55/45 split; gaps of 400 or larger saturate at the 60/40 cap. The bound applies symmetrically to gains and losses.

H.2 Why averaging team rating

Average team rating preserves the singles–doubles scale alignment: a 1700/1700 pair plays a 1700/1700 pair as an even match ($E = 0.5$), and a 1900/1500 pair plays the same 1700/1700 pair as an even match (also $E = 0.5$). The latter is approximately correct as a first-order model: if partner skills are independent and contribute additively to team performance, their average is the right summary statistic.

Alternatives considered:

- Sum of ratings. Rejected: places doubles on a different scale from singles, complicating cross-format interpretation.
- Min, max, or harmonic mean. Considered. The harmonic mean would penalize lopsided pairings more strongly, which has some intuitive appeal but no clear empirical justification at this stage. Average is preferred for simplicity until data exists to favor a different summary.

H.3 The split rule: equal, proportional, or bounded mild-skew

How a team-level rating change should be distributed between two partners is the single largest design decision in the doubles model. PERA selects bounded mild-skew after considering equal split and full proportional split. Each is discussed below.

(i) Equal split.

Equal split assigns the same ΔR to both partners regardless of their ratings. It is the maximum-entropy update under the assumption that no contribution data is available: the scoreboard alone does not say which partner played at, above, or below their rating, so the rule attributes the team-level surprise equally. Equal split is robust against attribution error — in particular, it never uses prior ratings to attribute current performance, which avoids the feedback loops described below. Its weakness is that it ignores a real intuition about doubles: the higher-rated partner typically takes more responsibility for team outcomes (anchoring formation, taking pressure shots, deciding strategy), and their rating arguably should be more sensitive to team results.

(ii) Full proportional split.

Full proportional assigns weights $w_i = R_i / (R_1 + R_2)$ and updates $\Delta R_i = 2 \cdot w_i \cdot K_n \cdot (S - E)$. It implements the intuition cleanly: a 1900/1500 pair has $w_1 = 0.559$ and $w_2 = 0.441$, so the 1900-

rated partner gains and loses about 27% more rating per match than the 1500-rated partner. Higher-rated players are more reactive in both directions.

Full proportional split has a structural problem: it uses prior ratings to attribute current performance. The match in front of the system is the only new evidence available; weighting that evidence by rating means the update partly reproduces what the system already believes, rather than purely incorporating what it just observed. In particular, full proportional has an asymmetric failure mode that is most pronounced in early-deployment regimes (see H.5).

(iii) Bounded mild-skew (the chosen rule).

Bounded mild-skew is the compromise. The multiplier m_i takes a value in [0.8, 1.2], with the magnitude of the skew proportional to the rating gap between partners and capped before the asymmetry can become large. At small partner gaps the rule is nearly equal split (m close to 1 for both); at large partner gaps the rule is bounded at 60/40 rather than allowing the skew to grow without limit. The form preserves the directional intuition of full proportional while preventing the worst behaviors of either extreme.

H.4 Why these specific parameters

Two parameters define the bounded mild-skew form: the gap-normalization constant D (set to 2000) and the cap on δ (set to ±0.2).

D = 2000 is chosen so that a partner gap of 200 rating points produces a modest 55/45 skew (m = 1.05 vs. 0.95) and a gap of 400 reaches the cap. This matches the practical shape of doubles partnerships: most paired players are within 100–300 rating points of each other, where the skew is modest; extreme pairings (500+ rating gaps) are uncommon but exist in social play, and for those the cap protects the lower-rated partner from being absorbed into their partner's rating noise.

The cap of ±0.2 (60/40 split) is calibrated to be perceptible but not extreme. A 60/40 split means the higher-rated partner moves 50% more rating per match than the lower-rated partner. Higher caps (e.g., 70/30 at ±0.4) were considered and rejected as too aggressive: at that ratio, the lower-rated player is effectively a passenger, which is empirically not how doubles works even in lopsided pairings.

Numerical illustration:

Partner gap	Skew δ	m_{higher} / m_{lower}	Per-match swing ratio
0	0.000	1.00 / 1.00	Equal
100	0.025	1.025 / 0.975	≈1.05×
200	0.050	1.05 / 0.95	≈1.11×

Partner gap	Skew δ	m_{higher} / m_{lower}	Per-match swing ratio
300	0.075	1.075 / 0.925	≈1.16×
400	0.100	1.10 / 0.90	≈1.22×
≥ 400 (cap)	0.200	1.20 / 0.80	1.50×

Note that the cap engages immediately at $\delta = 0.1$ (which corresponds to a 400-point gap given $D = 2000$) and rises no further. The table's last row shows the cap value for any gap of 400 or larger.

H.5 The asymmetric-correction problem (and why the bound is essential)

The deepest reason for capping the skew is that full proportional split fails asymmetrically when partners are mis-rated. Two scenarios illustrate the failure mode.

Scenario 1: high-rated partner is over-rated (correctable case).

Suppose a 1900-rated player has true skill 1600, paired with a correctly-rated 1500 player. The team will systematically underperform expectation: $S < E$, negative team surprises. Under full proportional split, the 1900 partner absorbs the larger share of the loss in rating, accelerating their convergence toward 1600. This case is handled well by full proportional.

Scenario 2: low-rated partner is under-rated (failure case).

Suppose a 1500-rated player has true skill 1800, paired with a correctly-rated 1900 player. The team will systematically over-perform expectation: $S > E$, positive team surprises. Under full proportional split, the 1900 partner gains more rating per match than the under-rated 1500 partner does. The 1900 drifts away from their already-correct rating, while the 1500 corrects toward 1800 more slowly than they would under equal split. The system actively works against the correction it should be making.

These two scenarios are roughly equally likely in a calibrated mature deployment, where the failures cancel on average. But during bootstrapping — the period in which any deployment is most fragile and most needs accurate ratings — the second case predominates, because every player begins at the default R_0 and most are mis-rated downward (their true skill is somewhere above 1500). Full proportional split therefore systematically degrades early-deployment rating accuracy.

Bounded mild-skew preserves the rating-skew intuition while limiting the magnitude of the asymmetric failure. At the chosen cap, the worst-case attribution error is 60/40 rather than the unbounded ratio that full proportional permits. Equal split would eliminate this failure mode entirely but at the cost of ignoring the rating-skew intuition altogether. The bound is the design point that makes the rule defensible in early-deployment conditions.

H.6 Future extension: contribution-weighted split

When per-player contribution data becomes available, the rating-skewed weights m_i should be replaced with empirical contribution weights w_i derived from the data. Sources of contribution data, in increasing order of richness:

- Manually-tracked per-rally point credit (kept by a scorer). Sufficient for a weighted split, but operationally costly.
- Video-based rally analysis (computer vision identifying the last-touch player on winning/losing rallies). Sufficient and increasingly feasible.
- Full rally-level analysis (positioning, reaction time, shot quality). The most informative; also the highest-data-cost.

The contribution-weighted form has the same mathematical structure as the rating-skewed form: $\Delta R_i = w_i \cdot K_n \cdot (S - E)$, where $w_1 + w_2 = 2$ (so the team-level update magnitude is preserved) and w_i is computed from observed contribution rather than from prior rating. This change removes the circularity entirely: the update no longer references prior ratings for attribution. A principled probabilistic version of this idea is implemented in TrueSkill (Herbrich et al., 2007). Until contribution data exists at the deployment scale, the bounded rating-skewed form serves as a defensible substitute that captures the same directional intuition.

H.7 Partner switching

In environments where players rotate partners across weeks, the variance of ΔR increases for any individual player. This is real signal noise, not a system failure. The uncertainty estimator U_n captures it: a player who rotates partners frequently will have higher U_n and a more responsive K_n, which is the correct system response. Deployments with very high partner-rotation rates (e.g., social round-robins where partners change every game) may want a higher α on the doubles track to give U_n a longer memory window.

H.8 What would invalidate this choice

Three empirical results would prompt revision. First, head-to-head comparison showing equal split produces lower out-of-sample log-loss on next-match doubles performance than bounded mild-skew — this would mean the rating-skew intuition is wrong empirically and the simpler equal split is preferred. Second, comparison showing full proportional outperforms bounded mild-skew on a corpus of mature, well-calibrated ratings (where the asymmetric-correction problem is less of a concern) — this would justify a higher cap or removal of the cap in mature deployments. Third, sensitivity analysis showing the chosen parameter values (D = 2000, cap at ±0.2) are far from optimal on real data — this would prompt re-tuning rather than abandonment of the form.

Appendix I. Justification for Path Dependence and Recalculation

I.1 The choice

Ratings are recomputed from the earliest available match when historical data is added; chronological order is the canonical processing order; ties are broken by match-id hash.

I.2 Why path dependence is unavoidable

Any incremental rating system that updates per match is path-dependent: the rating after match n depends on the rating before match n, which depends on match $n-1$, recursively. The only path-independent rating systems are batch-fitted models (logistic regression on all matches, Bradley-Terry (Bradley & Terry, 1952), etc.), and these have other trade-offs: they require global re-fitting on every new match, they don't naturally express uncertainty, and they don't handle a player's evolving form.

I.3 Why recompute on backfill

If a club uploads a year's worth of pre-existing matches after their players have already been rated for several months in the new system, two options exist: (a) ignore the historical matches, accepting that rating reliability will be lower than it could be; (b) recompute from the earliest available match, producing a self-consistent rating that reflects all available evidence.

PERA chooses (b) because the first option degrades systematically: the longer the system runs, the less of each player's actual history is reflected in their current rating. Recomputation is the only way to keep the system internally consistent as data accumulates.

I.4 The operational cost and how it is managed

Full recomputation is $O(n)$ in match count. For a small deployment this is instantaneous. For federation-scale deployments (millions of matches across thousands of clubs), nightly full recomputation is operationally expensive. The framework specifies two mitigations:

- Period checkpointing. Rating snapshots are stored at end-of-month boundaries (R, U_n, M, status for every player). When backfilled matches arrive, recomputation begins from the most recent checkpoint that pre-dates the earliest backfilled match. In a system with monthly checkpoints and backfills concentrated in the last six months, this reduces typical recomputation to $O(n/12)$ or better.
- Player-graph-scoped recomputation. When backfilled matches involve a small set of players who have not interacted with the broader graph during the window, recomputation can be limited to that subgraph. This is a valid optimization only when graph-isolation can be verified.

I.5 Canonical match ordering

When two matches share a timestamp, the rating outcome can depend on which is processed first. The framework specifies a deterministic tiebreaker:

- Primary: match start time, UTC, second precision.
- Tiebreaker 1: SHA-256 hash of the canonical match record (player IDs sorted, score string, venue ID), lexicographic order.
- Tiebreaker 2: venue identifier, lexicographic order.

This makes the rating output a deterministic function of the match log, regardless of upload order. Two operators recomputing the same match log will produce identical ratings.

I.6 What would invalidate this choice

Empirical observation that full-history recomputation produces meaningfully different rankings from incremental updates after long backfills, indicating the system has accumulated significant path drift. This is itself a calibration signal: it would prompt a reduction in K_{base} or a tightening of U_n bounds.

Appendix J. Network Connectivity: Open Problems and Approach

J.1 The problem

PERA produces locally valid ratings within any connected match graph. When the graph fragments into disconnected clusters (Raleigh club players never play California players), each cluster develops its own internally consistent rating scale, but the absolute rating values across clusters are not directly comparable. A 1700 in Raleigh is not necessarily equivalent to a 1700 in California; the two clusters may have drifted relative to each other over time.

J.2 Why no formula solves this

The connectivity problem is not a defect of the formula; it is a property of the match graph. No update rule, however clever, can extract information from matches that did not occur. The Bradley-Terry model (Bradley & Terry, 1952) fails in the same way; Glicko-2 (Glickman, 2013) fails in the same way; TrueSkill (Herbrich et al., 2007) fails in the same way. The problem is graph-theoretic and the response must be operational.

J.3 The PERA response: diagnose, cap, connect

Three mechanisms are specified in Section 6:

(i) Diagnose. Each deployment publishes connectivity statistics for its match graph: largest connected component size, diameter, bridge-match count, intra/inter-cluster ratio. These are published alongside ratings so users can interpret cross-cluster comparisons appropriately. A

cross-cluster comparison between two players sharing fewer than *k* bridge matches in their path can be flagged with reduced confidence.

(ii) Cap. Players whose match histories are confined to a small or weakly connected subgraph are capped at Established status, regardless of *M* and U_n. This prevents "100-match Stable rating" from being claimed by a player who has only played twelve different opponents in one club.

(iii) Connect. Operationally, deployments are encouraged to host non-elimination, multi-match formats specifically designed to maximize cross-cluster pairings. The HmBr mela construct is one implementation. Mela events are calibration mechanisms, not ranking events; they feed the rating system without producing a separate ranking. Match results from mela events are tagged in the log for analytics, but the rating formula does not apply event-specific multipliers.

J.4 Open problems acknowledged

- Minimum bridge-match density for global comparability is not analytically characterized. Simulation work (part of the validation roadmap) is required to estimate this as a function of cluster size and rating variance.
- Cross-cluster offset detection (estimating the relative bias between two clusters that have insufficient direct bridge matches) is not yet implemented. A graph-theoretic formulation in terms of effective resistance distance (Klein & Randić, 1993) is available in the network rating literature and is a candidate extension.
- The framework does not currently engage with the network-rating literature (e.g., HodgeRank; Jiang et al., 2011; spectral methods on tournament graphs). This is a gap in scholarly positioning that should be closed before journal submission.

J.5 What would invalidate this approach

Simulation results showing that even with aggressive structured cross-cluster interaction, ratings cannot achieve cross-cluster comparability within a reasonable time horizon. This would indicate the operational mechanisms are insufficient and the framework needs an explicit cluster-offset correction step.

Appendix K. Comparison with Prior Art

PERA shares conceptual ancestry with three prominent rating systems. A side-by-side comparison clarifies what is new in PERA and what is borrowed.

K.1 Elo (Elo, 1978)

- Update form: $R_{new} = R_{old} + K \cdot (S - E)$, where $S \in \{0, 0.5, 1\}$.
- Expectation: Logistic in rating gap with $c = 400$.

- K-factor: Fixed (typically tiered by rating).
- Uncertainty: None.
- Relation to PERA: PERA generalizes Elo by replacing binary S with continuous point share, replacing fixed K with bounded adaptive K, and adding an uncertainty estimator. PERA reduces to Elo when $S \in \{0, 1\}$, K is held constant, and U_n is ignored.

K.2 Glicko-2 (Glickman, 1999, 2013)

- Update form: Bayesian posterior update on a Gaussian prior over true skill.
- Expectation: Logistic on rating gap, with adjustment for rating deviation (RD).
- Uncertainty: Rating Deviation (RD) and Volatility (σ), updated using closed-form Bayesian formulas.
- Relation to PERA: Glicko-2's RD plays a role analogous to PERA's U_n, and its volatility plays a role analogous to PERA's K-factor adaptation. The differences:
    - Glicko-2 uses binary S; PERA uses continuous S. This is the central distinction.
    - Glicko-2's RD update is closed-form Bayesian; PERA's U_n is an EWMA. The EWMA is a deliberately simpler approximation: it sacrifices some statistical rigor for transparency and computational triviality. For sports communities where the rating system must be explainable to non-statisticians, this is a feature.
    - Glicko-2 batches matches into rating periods (typically 10–15 matches); PERA updates per match. PERA's per-match approach is friendlier to live rating displays but makes path dependence more pronounced.
- Honest assessment: Glicko-2 is mathematically more rigorous than PERA's uncertainty mechanism. PERA's claim is not that U_n is statistically superior to RD — it isn't — but that the combination of continuous S with a transparent uncertainty estimator produces a system that is both more performance-sensitive and more explainable than either Elo or Glicko-2 in isolation.

K.3 TrueSkill (Herbrich et al., 2007)

- Update form: Full Bayesian posterior update via factor graph message passing.
- Expectation: Gaussian-CDF on skill difference.
- Uncertainty: Per-player skill mean (μ) and skill standard deviation (σ); both updated jointly through Bayesian inference.
- Teams: Native support for arbitrary team sizes and partial orderings (e.g., a 4-team free-for-all where all four teams finish in different positions).
- Relation to PERA: TrueSkill's team handling is more principled than PERA's doubles model. PERA's bounded rating-skewed split (Appendix H) is a heuristic compromise designed to operate without per-player contribution data; TrueSkill computes the actual

posterior contribution of each team member to the team's skill estimate from observed match outcomes. For deployments with reliable per-player performance data, a TrueSkill-style approach is the right comparison point and the natural future evolution of PERA's contribution-weighted extension.

- Trade-off: TrueSkill is computationally heavier and harder to explain. For sports ecosystems where users need to understand why their rating moved by a specific amount, a TrueSkill update ("the message-passing algorithm converged to a new posterior") is opaque in a way the PERA update is not.

K.4 Margin-of-victory models in sports analytics

Continuous-margin scoring has been studied extensively in baseball (Pythagorean expectation; James, 1980–1988), American football (FPI, Sagarin, Massey, 1997), and tennis (point-share ratings). PERA's continuous-*S* choice belongs to this lineage. The specific design contribution of PERA over generic margin-of-victory rating is the integration of margin-based *S* with the bounded adaptive K-factor and the explainability-first status ladder.

K.5 Summary positioning

PERA is best understood as: continuous-*S* Elo with a transparent uncertainty tracker and a bounded adaptive *K*. It is mathematically less rigorous than Glicko-2 or TrueSkill on its uncertainty mechanism, but more performance-sensitive than Elo on its outcome signal. It is positioned for sports communities where explainability and operational simplicity matter as much as statistical optimality — conditions under which Glicko-2 and TrueSkill have historically faced adoption friction.

Appendix L. Validation Roadmap

PERA's status as a defensible framework rather than a validated system rests on the following empirical work, which should be completed before any large-scale deployment treats the parameter values as final.

L.1 Predictive validation

Required:

- A corpus of badminton match data with full point-level scoring (not just win/loss). Federation tournament archives are the most likely source.
- Compute PERA, binary Elo, Glicko-2, and (where feasible) TrueSkill ratings on the same corpus, using the same chronological ordering.
- On a held-out tail of matches, compare each system's predicted next-match *S* (or win probability) against the realized outcome. Score with log-loss and Brier score.
- Report results separately by player activity tier and by status level.

L.2 Parameter calibration

- c (expectation scale): MLE fit of logistic E to binned empirical S vs. rating gap.
- α (uncertainty memory): grid search on out-of-sample log-loss; check stability across player-activity tiers.
- K_{base}, γ, a: joint optimization with c on held-out predictive accuracy, with bounds constraints to preserve the design properties (boundedness, responsiveness).
- D and cap (doubles split): three-way comparison of equal split, full proportional split, and bounded mild-skew on a corpus of doubles matches with varying partner-rating gaps. Score by predictive log-loss on next-match individual performance, separately for under-rated and over-rated partners.
- Status thresholds: tuned against simulated rating drift to produce target progression rates.

L.3 Stress testing

- Pathological match sequences (long runs of mismatched opponents, simulated retirements, alternating-extreme outcomes) tested against the bounded K to verify no rating-runaway behavior occurs.
- Cold-start behavior: how many matches does a player rated truly at level X require to converge from R_0 = 1500 to within ±50 of X under each K-factor variant? Compare to Glicko-2's RD-based convergence.

L.4 Ecosystem validation

- Network connectivity simulation. Generate synthetic player ecosystems with varying cluster sizes and bridge-match densities. Measure rating drift between clusters as a function of bridge density. Identify the critical density at which cross-cluster comparability becomes acceptable.
- Mela-style intervention simulation. In a synthetic ecosystem with two large clusters and zero bridge matches, simulate the introduction of a structured cross-cluster event. Measure how quickly cross-cluster rating drift collapses as a function of event frequency and event size.

L.5 User comprehension

- Survey or interview users after exposure to a “won but lost rating” outcome. Measure understanding, perceived legitimacy, and behavioral response (do they continue to use the system?).
- A/B test the status ladder against a continuous confidence percentage. Measure comprehension and trust separately.

L.6 Required before publication

The minimum bar for academic publication of PERA is: predictive validation on a real corpus, head-to-head comparison against Glicko-2 and binary Elo, calibrated values for c and α on at least one match dataset, and explicit engagement with the network-rating and computer-sports-analytics literature. Until that work is done, PERA stands as a framework specification with a roadmap, not as a validated rating system.

Appendix M. References

References are provided for the foundational works that PERA builds on, compares against, or borrows technical machinery from. The list is selective: general statistical and machine-learning textbook concepts (logistic regression, maximum likelihood estimation, exponential moving averages as a class) are not individually cited.

Bradley, R. A., & Terry, M. E. (1952). Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika*, 39(3/4), 324–345.

Brier, G. W. (1950). Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1), 1–3.

Elo, A. E. (1978). *The Rating of Chess Players, Past and Present*. Arco Publishing.

Glickman, M. E. (1999). Parameter estimation in large dynamic paired comparison experiments. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 48(3), 377–394.

Glickman, M. E. (2013). Example of the Glicko-2 system. Working paper, Boston University. Available at *http://www.glicko.net/glicko/glicko2.pdf*.

Herbrich, R., Minka, T., & Graepel, T. (2007). TrueSkill™: A Bayesian skill rating system. In *Advances in Neural Information Processing Systems 19* (NIPS 2006), 569–576. MIT Press.

Hunter, J. S. (1986). The exponentially weighted moving average. *Journal of Quality Technology*, 18(4), 203–210.

James, B. (1980–1988). *Baseball Abstract* (annual editions). Self-published. [Origin of the Pythagorean expectation in baseball.]

Jiang, X., Lim, L. H., Yao, Y., & Ye, Y. (2011). Statistical ranking and combinatorial Hodge theory. *Mathematical Programming*, 127(1), 203–244.

Klein, D. J., & Randić, M. (1993). Resistance distance. *Journal of Mathematical Chemistry*, 12(1), 81–95.

Massey, K. (1997). *Statistical models applied to the rating of sports teams*. Bachelor's thesis, Bluefield College.

Thurstone, L. L. (1927). A law of comparative judgment. *Psychological Review*, 34(4), 273–286.