PickleRating: An ELO based Pickleball Rating System

Sam Fox

Abstract

Recreational pickleball presents a unique rating challenge which makes it difficult to quantify players' skill. The inconsistent and small player pool along with volatile team combinations renders the standard ELO system ineffective at accurately presenting how skillful a player truly is. This paper devises a rating system based on concepts presented in Mark Glickman's Glicko system [2], Microsoft's TrueSkill 2 [3], and the standard ELO formula [1]. These concepts are used to create a modified ELO formula which can quickly and accurately assess a player's rating.

1 Introduction

While many pickleball rating systems exist, few can encompass the difficult task of rating recreational players within a small group. The most well-known pickleball rating system, DUPR [4], is used to rank players globally and exists more so as a professional rating system; comparable to the better-known ATP system in tennis. The problem with DUPR is that to even get a rank you either need to participate in a sanctioned DUPR event/tournament or otherwise get rated by a DUPR coach. The average recreational pickleball player is likely not doing either of these things and that's where this rating system comes in. The goal is to create a rating system which can accurately quantitatively determine players' skill with a high convergence speed based only on the score of a recreational pickleball match.

The real skill (RS) of a player is the hypothetical abstract quantitative measure of how good a player really is. PickleRating (PR) is the real value associated with a player and is used to measure their skill and predict the outcome of matches.

Before working on a formula, it's important that the priorities and assumptions for the system are clear and relevant. A rating system is only as good as the underlying assumptions and after much deliberation I feel the following are the most relevant:

1.1 Priorities

1. The system can support doubles and singles ratings independently but allow for statistical correlations between them when establishing a baseline
2. Changeable skill ratings. It must be possible for a player's skill rating to meaningfully change no matter how many matches they have played in the past. This ensures players

receive meaningful feedback on their performance.
3. Scoring incentives. The over or under performance of a win/loss should factor into the rating difference
4. Good players should not be disincentivized from playing with weaker players.
5. Minimal training data requirements and a high convergence speed
6. Winning a match should never result in a loss of rating and losing a match should never result in a gain of rating.
7. The system should be zero-sum or near zero-sum to prevent rank inflation

1.2 Assumptions

1. An incredibly low ranked player will have a more significant impact on a team than an incredibly high ranked player of an equal PR degree.
2. A player's skill in singles is assumed to be statistically correlated to their skill in doubles and vice versa
3. The score differential of a game is a useful indicator to a player's skill. And as such, a projected score can be calculated based on players skill.
4. The longer it's been since a player has played a match the less certain you can be about their PR
5. Most matches played will not have balanced teams and often matches will be played between players of significantly different RS
6. PR gaps do not linearly scale, especially for players of lower skill levels.

2 Classic ELO

The standard ELO formula, used most commonly in chess [1], is defined as follows

$$E_A = \frac{1}{1+10^{(R_B - R_A)/c}} \tag{1}$$

Where E_A is the expected score of player A, R_A is the rating of player A and R_B is the rating of player B. The value c is tuned to determine how sensitive the expected score is to rating gaps. For every difference c in rating between player A and B, player A's odds of winning increase by a factor of 10 (an expected score of 10/11 ≈ 90.9%). This can be conversely expressed as follows for the expected score of player B.

$$E_B = \frac{1}{1+10^{(R_A - R_B)/c}} \tag{2}$$

This expected score is then used to calculate the player's rating change by using a "K-factor" which determines the maximum possible adjustment per game.

$$R'_A = R_A + K(S_A - E_A) \tag{3}$$

The rating adjustment uses values 0, 0.5, and 1 to represent a loss, win, and draw respectively and represents that as S_A. For our purposes we treat S_A as binary assuming a match cannot end in a draw in pickleball.

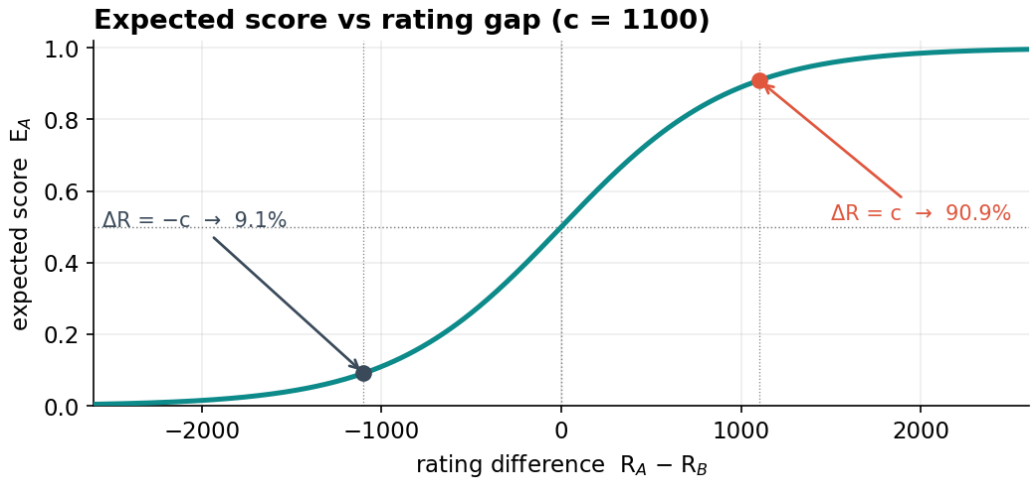


Figure 2.1 Expected score as a function of the rating gap between two players (c = 1100). A gap of one full c raises the stronger player to a 90.9% expected score, and the curve is symmetric about an even matchup.

3 Score Driven Adjustments

Accounting for scores in matches allows for a much faster convergence speed. Most ELO based systems avoid using game metrics in order to determine player skill for a variety of reasons. Whether it be misalignment, random variation across games, or in game statistics not correlating to players RS it is generally rare to see them used. However, Microsoft's TrueSkill 2 system [3] was built specifically to address TrueSkill's lack of reliance on game metrics. While TrueSkill 2 is a significantly more complicated Bayesian based prediction system, the concepts presented for using game metrics to determine player skill are relevant.

They found that tracking in game metrics (specifically kill counts in video games) was actually predictive of skill and the system was able to rate individuals with a team more accurately and converge faster. For the purpose of pickleball, the in-game stat being used will be the score differential of the match. We must first determine the probability that a team will win a rally using a near identical equation to the ELO one (1) with a new arbitrary variable c'.

$$p = \frac{1}{1+10^{\frac{R_B - R_A}{c'}}} \tag{4}$$

Using this probability, we can then determine what the projected score margin of the match will be. The following equation is capped at -11 and 11 as by standard pickleball rules you cannot win or lose by more than 11 points.

$$M_p = 11 * sign(2p-1) * |2p-1|^y \tag{5}$$

The y variable can be tuned to set the degree of sensitivity of the p factor when determining the projected score margin. We can then compare the actual margin to the real score margin in order to get an "f" factor which will multiply the final PR change.

$$f = f_{min} + (f_{max} - f_{min}) \cdot \sigma(s\lambda\Delta) \tag{6}$$

$$s = sign(S_A - E_A) \tag{7}$$

$$\Delta = M_A - M_p \tag{8}$$

$$\sigma(x) = 1/\left(1 + e^{(-x)}\right) \tag{9}$$

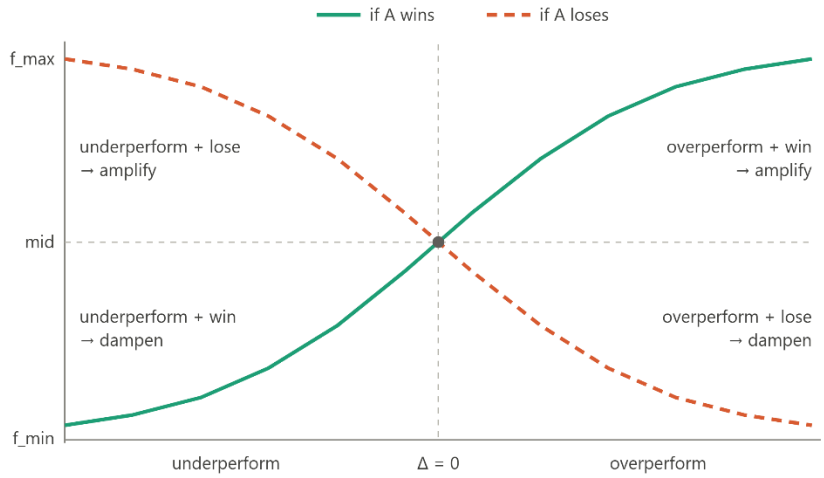


Figure 3.1 The impact of the f factor based on how the actual score margin differs from the expected score margin

This equation (6) allows for tuning of f_{min} and f_{max} to determine the bounds to which the f factor, or in essence, the score differential will impact PR changes. λ is an arbitrary variable which is used to determine the sensitivity of the Δ which is just the actual score differential vs the predicted.

The f factor doesn't just reward or punish players for winning by a lot of points or getting beaten by a lot of points. Rather, the *f* factor rewards and punishes players for performing better or worse than what's predicted. This is an important distinction because rewarding dominant players regardless of projections is just creating a larger PR gap rather than actually accounting for score margin.

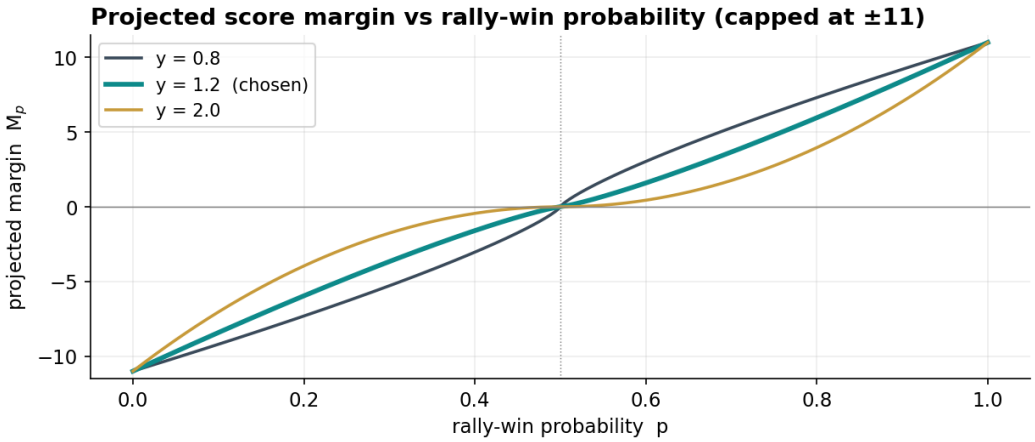


Figure 3.2 The projected score margin as a function of a player or team's rally-win probability p, for several values of the exponent y (capped at ±11). The chosen value y = 1.2 yields a near-linear mapping that steepens slightly as a rally advantage grows.

4 Rating Deviation and Glicko

In order to account for player inactivity and allow for a faster convergence on a player's RS we adopt the rating deviation (RD) from Glickman's Glicko [2]. This factor acts as a measure of rating confidence that decays during inactivity and tightens with play. The role that RD will serve for PR though is much narrower. Glicko is a Bayesian based rating system which reads RD as a posterior standard deviation. For our purposes we adopt RD's operational behavior directly using it to modulate the K factor without carrying the entirety of Glicko's full inference-based system. When determining RD after a time interval use the following equation.

$$RD = \min\left(\sqrt{RD_{\text{old}}^2 + c^2},\ RD_0\right) \tag{10}$$

In this equation "c is a constant that governs the increase in uncertainty between rating periods." [2] This means we can arbitrarily determine a c value based on a time interval passing. RD_0 is the initial RD value chosen for new players. In order to prevent confusion between this c value and the one used for expected score function the time sensitivity constant will be replaced with ψ.

The following is the exact equation that Glickman outlines to calculate RD after a rating interval occurs:

$$RD' = \sqrt{\left(\frac{1}{RD^2} + \frac{1}{d^2}\right)^{-1}} \tag{11}$$

$$q = \frac{\ln 10}{400} = 0.0057565 \tag{12}$$

$$g(\text{RD}) = \frac{1}{\sqrt{1+3q^2(\text{RD}^2)/\pi^2}} \tag{13}$$

$$E\left(s \mid r, r_j, \text{RD}_j\right) = \frac{1}{1+10^{-g\left(\text{RD}_j\right)\left(r-r_j\right)/400}} \tag{14}$$

$$d^2 = \left(q^2 \sum_{j=1}^{m} \left(g\left(\text{RD}_j\right)\right)^2 E\left(s \mid r, r_j, \text{RD}_j\right)\left(1 - E\left(s \mid r, r_j, \text{RD}_j\right)\right)\right)^{-1} \tag{15}$$

For the purpose of PR RD serves three very useful purposes. It allows new players to more quickly converge on an accurate PR threshold without needing a large sample size of games. RD also allows players to have less significant shifts once they've reached a PR range that is accurate to their RS. And it allows players who have not played many games to get quick substantial feedback on their performance.

The only necessary adjustment to the RD formula for singles is to change the "400" value in (12) for q to be the same as the chosen c value for the ELO formula.

5 Calculating Rating for Singles

Singles rating is significantly easier to calculate than for teams. There is a lot less entanglement when attempting to calculate the skill of an individual player. We can essentially just use the standard ELO (1) and RD (11) equations with our own choice of variables, and along with the *f* factor (6) can get a good gauge on a player's RS. This paired with the score driven adjustments mentioned in §3 should allow for the most accurate representation of a player PR. The final truncated equation for a singles match appears as follows:

$$R'_A = R_A + f\frac{\mathrm{RD}}{RD_0}\mathrm{K}(S_A - E_A) \tag{16}$$

Where f, RD′, and E_A are as defined in (6), (11), and (1):

$$f = f_{min} + (f_{max} - f_{min}) \cdot \sigma(s\lambda\Delta)$$

$$RD' = \sqrt{\left(\frac{1}{RD^2} + \frac{1}{d^2}\right)^{-1}}$$

$$E_A = \frac{1}{1 + 10^{(R_B - R_A)\,/\,\mathrm{c}}}$$

This leaves us with tunable parameters λ, y, c, c', and K. Higher λ's will increase the sensitivity of overperforming or underperforming your expected scores. The c value can be tuned to set the PR gap at which a player's odds of winning a matchup increase by a factor of 10 (≈90.9% win probability), and the same for c' but applied to rally win probability. *y* is used to determine how sensitive projected scores are to projected rallies. And K determines how sensitive rating adjustments are in general and determines how spread-out players should be.

With the singles rating formula established its worth noting some shortcomings of the formula as it stands. The biggest flaw lies in the *f* factor. The reason for this is because without significant data collection it's difficult to get a gauge on the empirical data of a player winning a rally. The outcome of a match can be easily measured and logged, and the c parameter can be tuned to fit the real win percentage of players, but rallies are often inconsistent and based on a multitude of factors and matchups. Weather conditions, playstyles, fatigue, and a multitude of other factors can contribute to a lower or higher rally percentage than expected. The caveat to this is that since the expected rally chance is tied to PR, we can assume that a higher PR player is also more likely to win their rally. So, while the calculation for *p* is not perfect it still works as the best way to calculate the expected score of a match.

The concepts of entanglement mentioned above only become more difficult to account for when playing doubles.

6 Addressing Teams in the ELO Formula

A typical recreational match of pickleball is often unbalanced. This system does not assume the luxury of matches being played between players of relatively similar RS, rather it must be

assumed that matches will be played between players of wildly different skill levels. This incorporates a difficult challenge as a new and unskilled player will reduce the team skill (TS) more significantly than a seasoned highly skilled player will raise the TS. This is largely because in pickleball players have isolated moments from their teammates, more specifically, serving and returning the ball. In these moments no matter how high a player's RS, if their teammate is unable to return, or serve the ball the team loses 50% of their scoring opportunities. This handicap is too significant to be accounted for by the higher RS player, assuming TS is calculated by averaging the player's PR.

To address this, a new factor representing the TS will be used. The players on a team can represented as x_{min} and x_{max} corresponding to each teammate's individual PR.

$$TS = x_{min} + \beta(G)G \tag{17}$$

$$G = x_{max} - x_{min} \tag{18}$$

$$\beta(G) = \beta_{min} + (\frac{1}{2} - \beta_{min})e^{-G/\tau} \tag{19}$$

τ is an arbitrary factor which can be tuned in order to create the soft threshold where the difference between partners starts to scale nonlinearly. When partners gap (G) is negligible TS essentially represents an average of both partners' PR. β_{min} makes the curve monotonic so that a stronger player offers diminishing returns on TS as G increases. The TS factor will be used in place of the player rating in the ELO formula (1) giving:

$$E_A = \frac{1}{1+10^{(TS_B - TS_A)\,/\,c}} \tag{20}$$

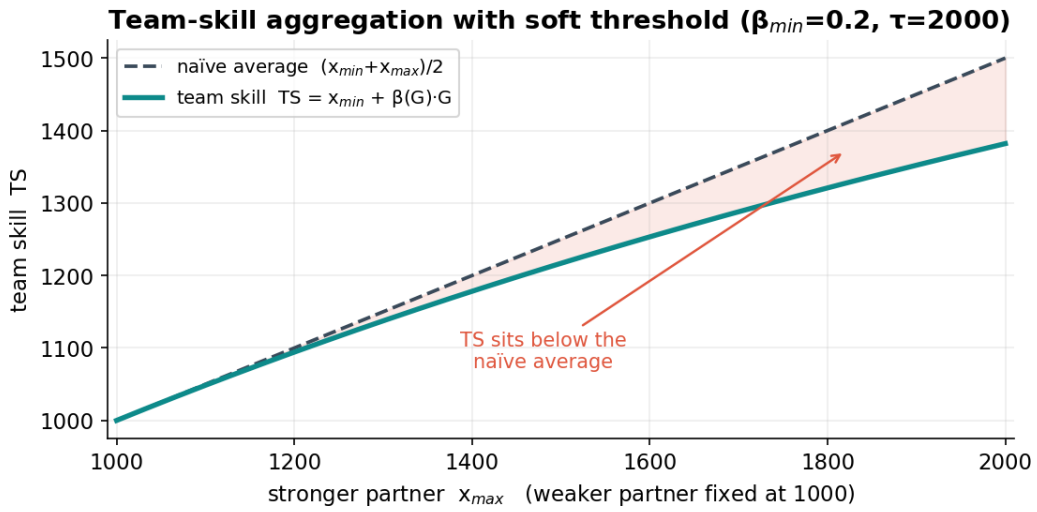


Figure 6.1 Team skill as the stronger partner improves while the weaker partner is held at 1000. The soft-threshold weighting β(G) pulls the team skill below the naïve average as the partner gap G grows, reflecting the larger handicap an unbalanced pairing carries.

7 Modifying Rating Deviation for Teams

For doubles we will be making a few modifications to the Glicko formula. We will of course use our modified expected-score function (20) from §6 which accounts for TS. We will also change Glicko's 400 value in the q equation to the chosen c value in the aforementioned expected-score function. The difficult factor to account for is RD_j. Capturing the uncertainty about an individual is easy but for a team you have to capture the uncertainty of the *pairing itself.* There is a lot of uncertainty for a pairing of players whether it be their chemistry, coverage, forehand dominance, etc. So, an equation is needed which can assess the certainty of each individual player along with the certainty of that pairing of people. The following attempts to do just that:

$$RD_{team} = \sqrt{\left(RD_{skill}^2 + RD_{pair}^2\right)} \tag{21}$$

$$RD_{skill}^2 = \left(\frac{\partial TS}{\partial x_{min}}\right)^2 RD_{min}^2 + \left(\frac{\partial TS}{\partial x_{max}}\right)^2 RD_{max}^2 \tag{22}$$

Here we separately assess the influence of each player on the rating deviation by multiplying their impact on the TS by their individual RD. Equations for TS (17) and $\beta(G)$ (19) can be found in §6. Each derivative can be expressed as:

$$\frac{\partial TS}{\partial x_{max}} = \beta(G) + G\beta'(G) \tag{23}$$

$$\frac{\partial TS}{\partial x_{min}} = 1 - \beta(G) - G\beta'(G) \tag{24}$$

$$\beta'(G) = -\frac{\frac{1}{2} - \beta_{min}}{\tau} e^{-G/\tau} \tag{25}$$

With the individual impact of each player's RD assessed we now need to account for the RD of that pairing of players. This is essentially just the RD equation used for Glicko (11) but accounting specifically for a player pair.

$$RD_{pair} = \min\left(\sqrt{RD_{pair,old}^2 + \psi^2 \Delta t_{pair}}, RD_{pair0}\right) \tag{26}$$

$$\frac{1}{RD_{pair}{}'^2} = \frac{1}{RD_{pair}^2} + \frac{1}{\kappa_{pair}^2} \tag{27}$$

These two equations adjust the RD for that pairing of players for time spent between matches periods and updates after each match respectively. With all of this in mind we can create our modified RD system. It's worth noting that RD is representative of an individual's rating deviation and RD_{team} is a separate variable representative of the team's rating deviation.

$$RD = \min\left(\sqrt{RD_{\text{old}}^2 + \psi^2 \Delta t}, RD_0\right) \tag{28}$$

$$RD' = \sqrt{\left(\frac{1}{RD^2} + \frac{1}{d^2}\right)^{-1}} \tag{29}$$

$$q = \frac{\ln 10}{c} \tag{30}$$

$$g\left(\mathrm{RD}_{team,j}\right) = \frac{1}{\sqrt{1+3q^2\left(\mathrm{RD}_{team,j}{}^2\right)/\pi^2}} \tag{31}$$

$$E\left(s \mid TS, TS_j, \mathrm{RD}_{team,j}\right) = \frac{1}{1+10^{-g\left(\mathrm{RD}_{team,j}\right)\left(TS-TS_j\right)/c}} \tag{32}$$

$$d^2 = \left(q^2 \sum_{j=1}^{m}\left(g\left(\mathrm{RD}_{team,j}\right)\right)^2 E\left(s \mid TS, TS_j, \mathrm{RD}_{team,j}\right)\left(1 - E\left(s \mid TS, TS_j, \mathrm{RD}_{team,j}\right)\right)\right)^{-1} \tag{33}$$

8 Calculating Rating for Doubles

With team score and its effects on rating deviation established we can now create the framework for calculating the changes to a players doubles PR. One thing worth noting is the change to the p factor (4) which now includes TS rather than an individual players' rating.

$$p = \frac{1}{1+10^{\frac{TS_B - TS_A}{c\prime}}} \tag{34}$$

It should also be noted that d^2 will be calculated as explained by (33) in §7 using RD_{team} rather than using a player's individual RD.

$$R_A' = R_A + f\frac{\mathrm{RD}}{RD_0}\mathrm{K}(S_A - E_A) \tag{35}$$

Where f, RD′, E_A, and TS are as defined in (6), (11), (20), and (17):

$$f = f_{min} + (f_{max} - f_{min}) \cdot \sigma(s\lambda\Delta)$$

$$RD' = \sqrt{\left(\frac{1}{RD^2} + \frac{1}{d^2}\right)^{-1}}$$

$$E_A = \frac{1}{1 + 10^{(TS_B - TS_A)\,/\,\mathrm{c}}}$$

$$TS = x_{min} + \beta(G)G$$

9 Correlating Doubles and Singles Rating

While doubles and singles will have distinct PR rankings for each player as each game type tests different skills it is more likely than not that highly skilled doubles player will also be skilled in singles. With this in mind, it's important that if a player has an established PR in doubles that their initial singles PR is reflective of their RS or vice versa. In order to do that we will once again reference Microsoft's TrueSkill 2 system [3] as they address this exact point.

The way that Microsoft handles it is by constructing a single skill rating which takes a weighted average of the player's rating in each mode where the weight is determined by the probability of playing that mode. Microsoft uses a complex Gaussian process to determine that probability, but our approach will be much simpler. The weighted average will simply be determined by the number of games in doubles/singles over the total games played.

$$w_d = \frac{G_d}{G} \tag{36}$$

$$w_s = \frac{G_s}{G} \tag{37}$$

This makes it incredibly easy to calculate a player's overall PR.

$$PR_O = w_d PR_d + w_s PR_s \tag{38}$$

What becomes more difficult is updating a player's singles or doubles rating based on their rating from the opposite game mode. The reason this is necessary is because if a player plays exclusively doubles and climbs to a rating of say 2000 and then plays a singles match at a base rating of 1000 that probability is nowhere near what it should be. The following equations are run when a match is first played in the opposite game mode:

First match in doubles if singles has established PR:

$$\mathrm{PR}_d' = \mathrm{PR}_d + C(\mathrm{PR}_s - \mathrm{PR}_d) \tag{39}$$

First match in singles if doubles has established PR:

$$PR_s' = PR_s + C(PR_d - PR_s) \tag{40}$$

Here *C* represents the correlation factor allowing for tuning of how correlated doubles and singles rating should be with a *C* of 1 being a direct correlation.

10 Variable Selection

Before running any calculations, justifications for the chosen variable values are provided. These values are ultimately a matter of convention rather than ground truth. The values chosen were selected to accomplish the following five goals:

1. Create a meaningful spread of ~1500 points between the best and worst players. Players want to immediately see a notable gap between themselves and a player who is much better or worse than them. A gap of 750 points between an average player and the best

player is interpreted as more significant than a gap of, say, 100.

2. Allow players' ranks to meaningfully change even after they have played hundreds of matches. Players want to keep getting feedback on their games even once they are at a rank where they belong. While this adds noise to a player's PR, it is worth the tradeoff for a better player experience.

3. Do not allow players to have a negative amount of PR, both so that none of the equations break, and so that the scale feels natural.

4. Allow a quick convergence speed of around 5 matches. Players who are first joining want to get a rank, and get it quickly, even if it is not perfectly accurate.

5. Account for players who will only win 10–15% of the time. In a recreational group it is likely that a new player will have a long streak of losing; these players must be handled in a way where good players must not be discouraged from playing them.

The remainder of this section first describes the methodology used to choose and validate the parameters, then presents the chosen values grouped by function: scale, dynamics, score-margin shape, and boundaries. For each group simulation evidence is provided.

10.1 Methodology

The parameters do not act independently. The f-factor multiplies the effective gain K, the rating deviation (RD) damps every update, and the score-margin terms couple the rally model to the match model. Because of these interactions and entanglement there is no "correct answer" as to how these variables should be set to accomplish the aforementioned goals. In order to determine what values should be used though comprehensive simulations were run to assign values.

Each simulated player is assigned a rally-winning skill θ. A simulated match is then played out rally by rally to 11 (win by 2), so that both the match outcome and the final score margin come from the same skill factor. This matters because margins are generated rather than assumed, the simulation can test whether PR's score-margin machinery genuinely improves the system or merely adds noise or inflation.

Leagues of 15–30 players were simulated with random pairings, updating every rating with the full PR update after each match. Four quantities were tracked:

1. **Spread:** the gap between the strongest and weakest player ratings.

2. **Convergence speed:** the games per player required for the spread to reach 90% of its ceiling value.

3. **Noise:** the standard deviation of an established player's rating once converged.

4. **Calibration (accuracy)**: the log-loss of PR's pre-match win probability E_a against actual outcomes, measured against the optimal achievable log-loss. The optimal value is the entropy of the true match-win probabilities (estimated by Monte Carlo from the latent

skills); the gap between PR's log-loss and this floor, the excess log-loss, measures how much predictive accuracy is sacrificed to mis-scaling or noise. A system whose excess log-loss is near zero predicts as accurately as the underlying randomness allows, which is the central check that the formulas are sound.

A structural property of the system makes selection tractable. If the parameters c, c′, K, RD_0, ψ, the starting rating, and the floor are all multiplied by a common factor α, every player's rating is multiplied by α while convergence speed, noise, and calibration remain identical (verified in simulation to within 0.01 points). The overall scale of the system is therefore a free cosmetic choice; it sets the units of the rating, not its accuracy. The consequence is decisive: the spread target (Goal 1) is met purely by choosing the scale, after which only the ratios among the scale-group parameters and the four dimensionless shape parameters (y, λ, f_{min}, f_{max}) affect behavior. The parameters below are presented in functional groups on that basis.

10.2 Scale parameters (Goal 1)

Variable	Value	What it controls	Why this value
c	1100	Rating scale / point spread	A genuine 15%-vs-88% skill gap maps to ~1500 points; the best player lands in Champion.
c′	2750	Rally-model scale (margin reference)	c′/c = 2.5 so that projected-margin reference matches c.

Because of scale invariance, c alone sets the spread. Empirically the spread of a converged league is approximately the skill range times c. For a standard recreational population where the weakest player winning ~15% of matches, strongest ~88% c = 1100 produces a best-to-worst spread of roughly 1500, with the median player in Gold (matching the start = 1000 convention), the strongest in Champion (~1750), and the weakest in Bronze (~300). The average-to-best gap is roughly 750, satisfying the "meaningful gap" requirement of Goal 1.

Spread cannot, however, be guaranteed by c alone, because it is the product of c and the actual skill range. A narrow friend group in which the weakest player still wins ~43% and the strongest ~62% will spread only ~450–850 even at large c, because the skill differences needed to justify a wider spread do not exist; forcing 1500 points onto such a group would manufacture gaps that are not real. Conversely, a league containing genuine outliers spreads beyond 2000, with those players correctly landing in Legend. c = 1100 is therefore calibrated to a standard league, and each league's spread is allowed to reflect its true competitive range.

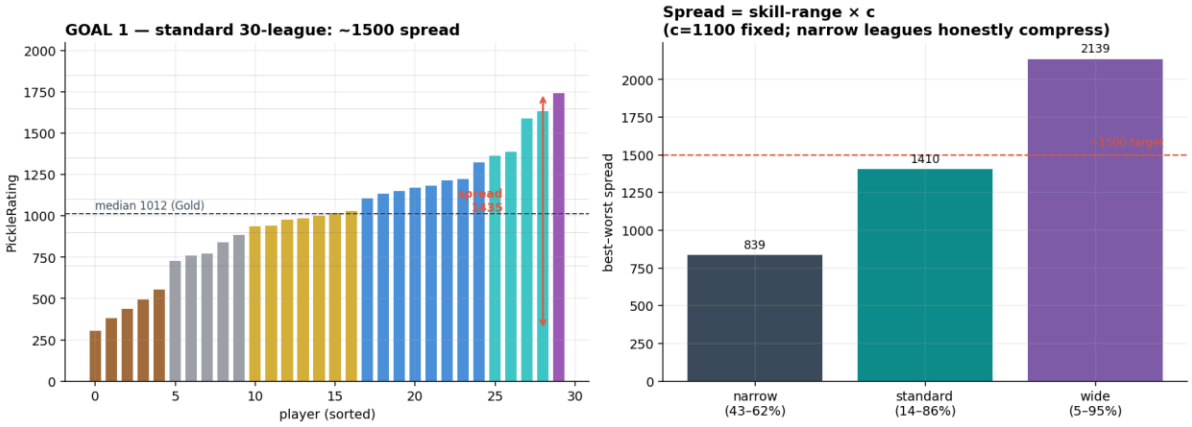


Figure 10.1

Left: converged ratings of a standard 30-player league, colored by tier, showing a ~1500 best-to-worst spread with the median in Gold.

Right: best-to-worst spread versus league skill range at fixed c = 1100; narrow leagues compress and wide leagues extend, by design.

10.3 Dynamics: convergence, responsiveness and noise (Goals 2 and 4)

Variable	Value	What it controls	Why this value
K	100	Base ranking scale factor	Converges in ~58 games/player; established swings ~±30–45, max ~150. Larger K is faster but noisier; smaller K is more accurate but slow.
RD_0	350	Initial / maximum RD	Suggested base RD value from Glicko
ψ	30	RD inflation per period	Stops RD decaying to zero; holds RD* ≈ 175 so established players keep getting meaningful adjustments forever (Goal 2).

K is the factor which affects speed/noise control. At K = 100 a league converges in roughly 58 games per player, and an established rating carries a standard deviation of about 87 points. Halving K to 50 improves calibration (excess log-loss 0.004 versus 0.007) but nearly doubles convergence time to ~106 games per player, while K above 200 degrades both noise and accuracy sharply. K = 100 sits at the fast end of the accurate range, prioritizing the quick feedback recreational players expect. With RD_0 = 350 the maximum single-match swing is $f_{max} \cdot K$ = 150 points and typical established swings are ±30–45, consistent with the existing system.

The rating deviation determines how large an established player's steps remain. Under the Glicko update RD shrinks with every game; without an inflation term it would decay toward zero and a settled rating would freeze, meaning no result could move it. The inflation parameter

ψ counteracts this: at ψ = 30 the RD stabilizes around 175 rather than decaying, so a player who has settled at their correct rank keeps receiving meaningful adjustments and will track a genuine change in skill. With ψ = 0 the step size collapses and a real improvement is tracked slowly and incompletely. This responsiveness is exactly the noise Goal 2 accepts; its calibration cost is small; excess log-loss rises only from +0.011 (ψ = 10) to +0.016 (ψ = 30).

RD_0 is the cap on the rating deviation and the value at which new players enter. Because a new player's step is scaled by $RD/RD_0 = 1$ while an established player's is scaled by $RD^*/RD_0 \approx 0.5$, newcomers move roughly twice as fast as settled players even before any explicit placement boost. $RD_0 = 350$ sets this ratio at about 2; raising it strengthens the provisional effect at the cost of noisier early ratings.

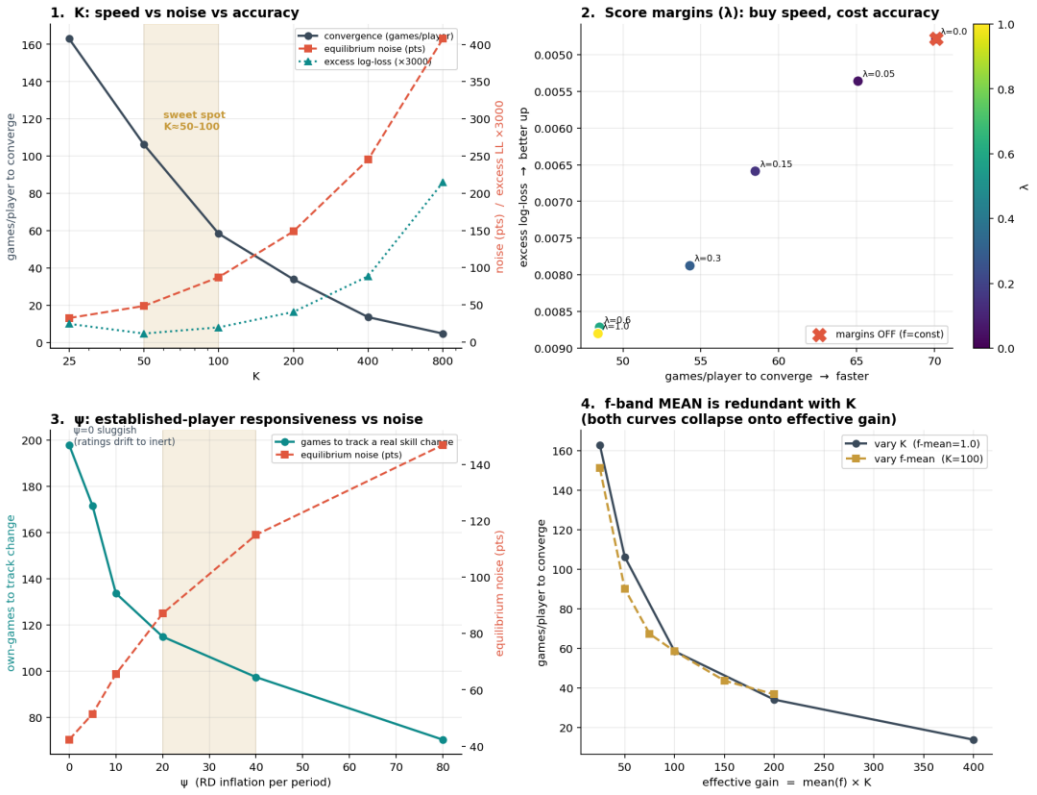


Figure 10.2 Per-parameter sweeps.

(1) K trades convergence speed against noise and accuracy with an optimal K value between 50–100.

(2) The margin sensitivity λ buys convergence speed at a small accuracy cost.

(3) ψ trades established-player responsiveness against noise.

(4) The mean of the f-band is interchangeable with K, both collapse onto a single effective-gain curve.

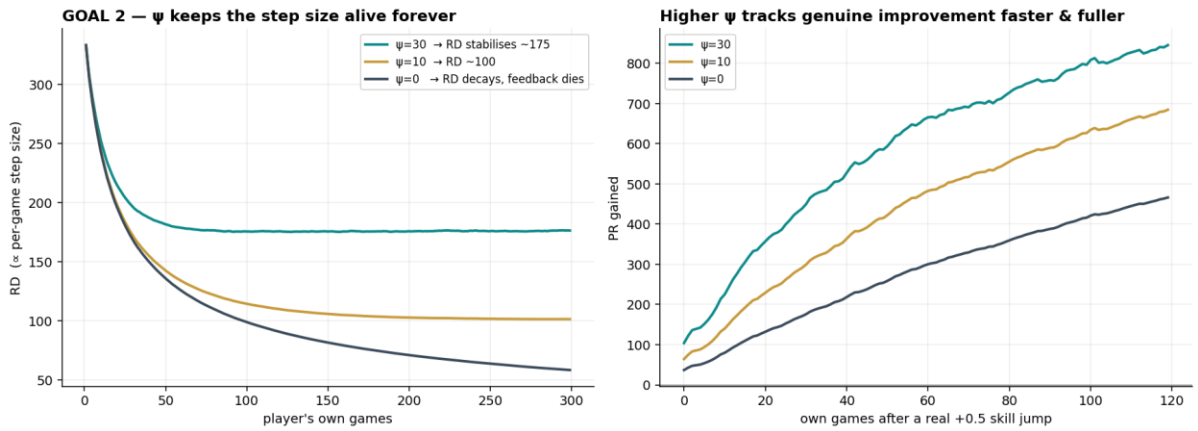


Figure 10.3 Goal 2.

Left: with ψ = 30 the rating deviation stabilizes rather than decaying to zero, so feedback never dies.

Right: higher ψ tracks a genuine mid-career skill improvement faster and more completely.

10.4 Score-margin shape: the f-factor

Variable	Value	What it controls	Why this value
y	1.2	Margin-projection exponent	Shapes projected margin vs rally-win probability; values ≥ 1 calibrate slightly better.
λ	0.10	Margin-surprise sensitivity	How strongly f responds to over/under-performance vs the projected margin. Modest value speeds convergence and rewards decisive play; large values add noise.
f_{min}	0.5	Lower f bound	With f_{max}, gives a band mean of 1.0 (so K stays interpretable) and a width of 1.0 making the margin relevant.
f_{max}	1.5	Upper f bound	See f_{min}.

The f-factor multiplies each update by a value between f_{min} and f_{max} according to how the actual score margin compares with the margin projected from the rating gap. Three facts from the sweeps in Figure 10.2 govern its tuning. First, the mean of the band acts as a global multiplier on K: a band centered on 1.0 (f_{min} = 0.5, f_{max} = 1.5) leaves K meaning exactly what the dynamics analysis assumed, which is why the band is centered there. Second, the sensitivity λ increases convergence speed, using score margins cuts convergence from ~70 to ~58 games per player at a small calibration cost (excess log-loss rises ~0.002), because score margins are a noisier signal than the binary result. λ = 0.10 takes most of the speed benefit while keeping the cost negligible and realizes Priority 3's intent that decisive scorelines be rewarded. Third, the exponent y has only a second-order effect, with values at or above 1 slightly preferable; y = 1.2 is chosen on that basis. A natural refinement, not adopted here, is to decay λ as a player's RD falls taking the

convergence benefit early and the calibration benefit later.

10.5 Boundaries and initial values (Goals 3, 4 and 5)

Variable	Value	What it controls	Why this value
start	1000	Initial rating	Places new players in Gold, the average tier.
floor	0	Minimum rating	Guarantees non-negative PR (Goal 3) and gives chronic losers a stable resting point (Goal 5).
placement multiplier	3×	K multiplier during placement	Lets a newcomer reach approximately the right tier within ~5 matches (Goal 4).
placement games	5	Length of placement phase	Number of initial games over which the multiplier applies.

Without a floor, a player who only wins rarely has no stable equilibrium: their rating drifts downward without bound, breaks the cosmetic scale, and, by removing points from a near-zero-sum system faster than they can be returned, distorts everyone else's ratings. A hard floor at zero solves all three problems. None of the PR equations contains a logarithm or other undefined operation at zero, so pinning a rating at the floor is safe; the expected-score and RD updates simply continue to operate on the floored value. Figure 10.4 shows a chronic loser pinned just above zero in contrast to the unbounded divergence that occurs without the floor; across 40 simulated leagues the minimum rating is always non-negative. The floor has one expected side effect: because losses that would push a rating below zero are not fully removed from the pool, the league mean inflates slightly over time, settling around 1011–1070 which is consistent with the 1072–1121 drift observed in the live system.

A player who wins only 10–15% of matches settles low in Bronze (around 150–300 points) and stays there stably without going negative (Figure 10.5, left). The concern that strong players might be discouraged from playing them is addressed, in singles, by the RD damping: an established strong player has a low RD, so although beating a much weaker opponent yields only a few points, an upset loss is also bounded on the order of −30 to −55 rather than catastrophic (Figure 10.5, right). This asymmetry is important to any ELO-type system and is intentional, but its magnitude is capped. Discouragement from partnering with a weak player is handled by the team-skill aggregation of §6 rather than by these singles parameters.

Because a wide rating scale makes individual points "cheap," a newcomer moving from the start value to their true rating by ordinary updates would take many matches, too slow for Goal 4. During the first five matches the update gain is therefore multiplied by three. Figure 10.6 shows that with this boost a weak or average newcomer is placed in the correct tier within five games, and a strong newcomer reaches within one tier of their eventual rank and keeps climbing, whereas without the boost the same strong player is still far short at game five. The boost applies only to the placement window, so it accelerates initial placement without adding lasting noise.

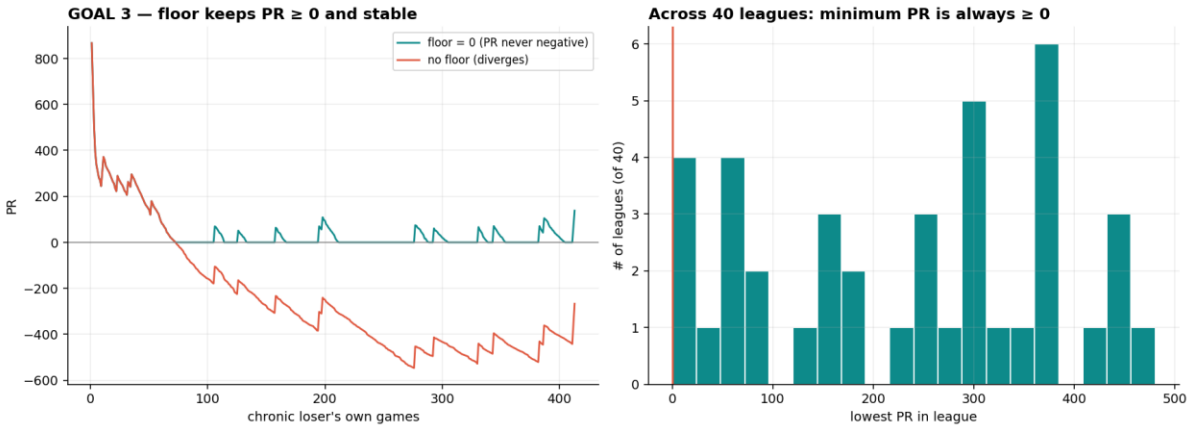


Figure 10.4 Goal 3.

Left: a chronic loser with the floor (stable just above zero) versus without it (unbounded negative PR).

Right: across 40 leagues the minimum rating is always ≥ 0.

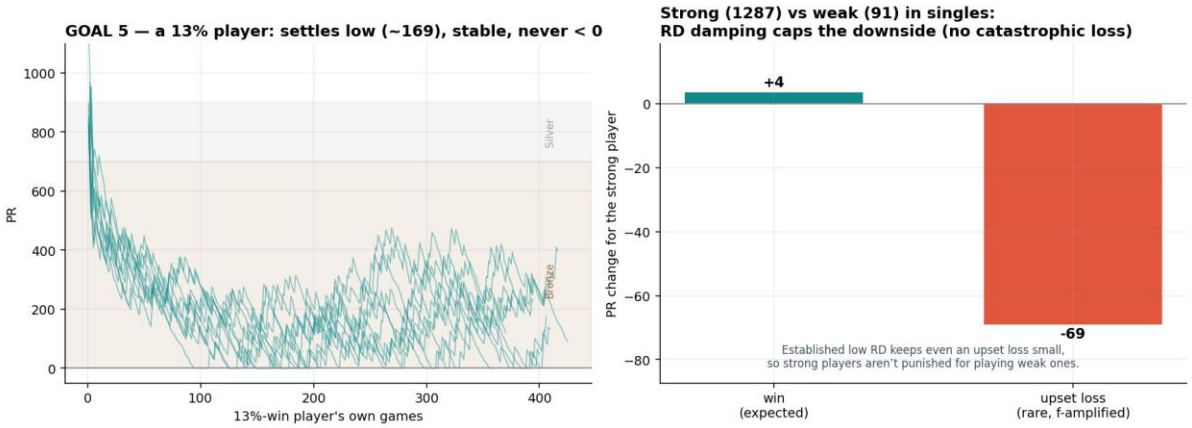


Figure 10.5 Goal 5.

Left: a 13%-win player settles low in Bronze, stably and non-negatively.

Right: RD damping caps a strong player's downside against a weak opponent, so playing weaker players is not heavily penalized.

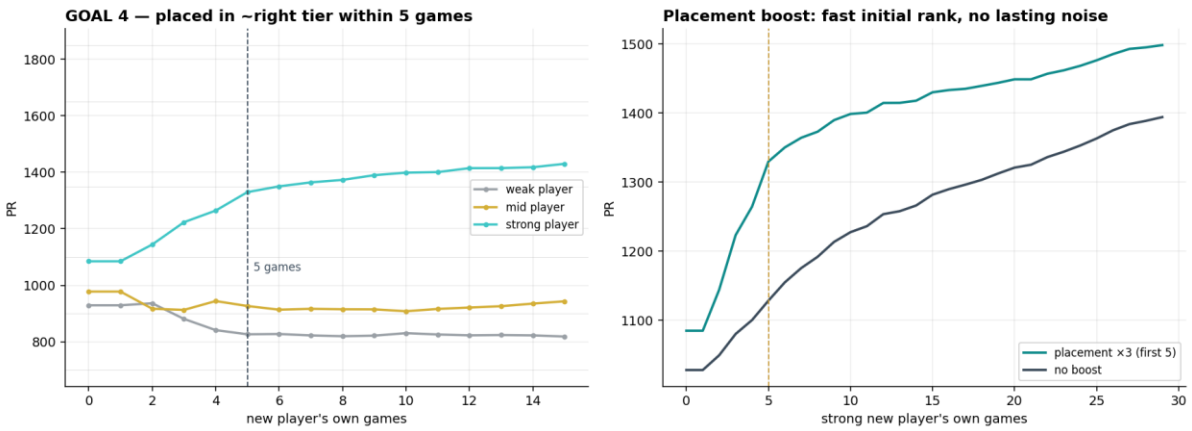


Figure 10.6 Goal 4.

Left: weak, average and strong newcomers are placed in approximately the correct tier within five games.

Right: the ×3 placement boost reaches a usable rank quickly without lasting noise.

10.6 Rank thresholds

The rank bands are fixed PR ranges rather than percentiles, so a player's tier is purely a function of their rating, and the population distribution is whatever the ratings produce. The bands below were validated against the simulated distributions: the median player sits in Gold (matching start = 1000), the mass of a standard league clusters from Bronze through Platinum, and Diamond, Champion and Legend form a thinning upper tail. Legend (1850+) is reached only by genuine standouts or in large, wide leagues, consistent with its intended rarity.

Tier	PR range	Width	Notes
Bronze	0 – 699	700	Below the starting PR; weak and chronic-loss players.
Silver	700 – 899	200	Slightly below average
Gold	900 – 1099	200	Average player (start = 1000 sits here).
Platinum	1100 – 1349	250	Slightly above average
Diamond	1350 – 1649	300	Wider bands deliberately slow upper-tier progress.
Champion	1650 – 1849	200	Very strong players and likely the highest rank within the average league
Legend	1850 +	∞	Rare large, active leagues only. Best of the best players

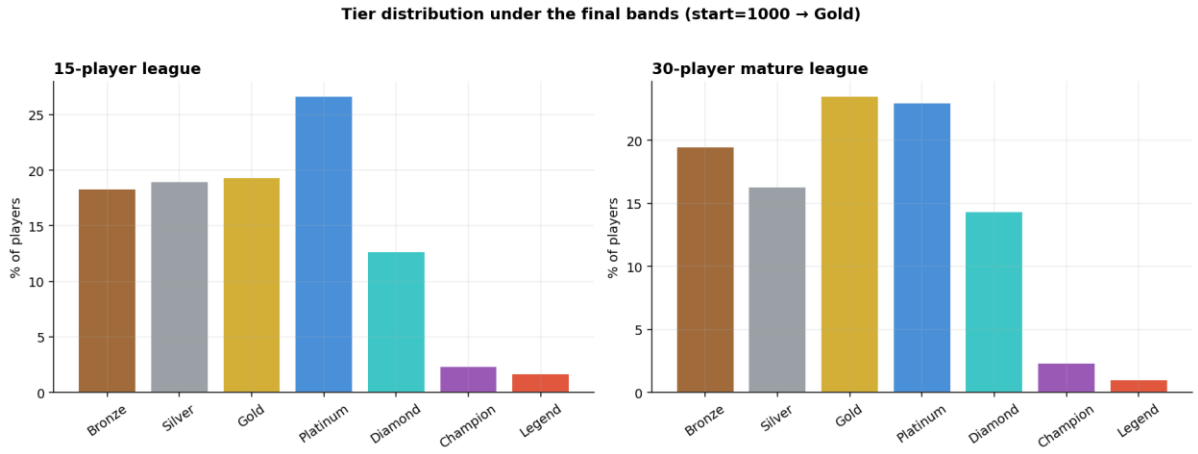


Figure 10.7 Simulated tier distribution under the final bands, for a 15-player and a 30-player league. Mass clusters Bronze–Platinum; Champion (~2%) and Legend (~1%) remain rare.

In simulation each tier from Bronze through Platinum holds roughly 16–26% of players, Diamond about 13–14%, Champion about 2% and Legend about 1%. One optional adjustment: because the floor-induced inflation pushes the mean toward the Gold/Platinum boundary, Platinum is co-modal with Gold; nudging the Gold/Platinum boundary up by ~50 points would re-center the distribution on Gold, but the bands are otherwise well-matched to the scale.

10.7 Soundness and calibration

The final parameter set was checked against the calibration metric of §10.1. Across simulated standard leagues PR's pre-match win probabilities achieve a log-loss of 0.605–0.610 against an optimal (irreducible) value of 0.594 — an excess of only 0.011–0.016. The reliability curve in Figure 10.8 lies almost exactly on the diagonal: when PR predicts a 70% chance for the favorite, the favorite wins close to 70% of the time. The system is therefore extracting nearly all of the predictive information available from match results; the formulas are sound, and the small residual gap is the deliberate noise introduced for responsiveness (Goal 2). The right panel confirms the bounded mean inflation discussed in §10.5.

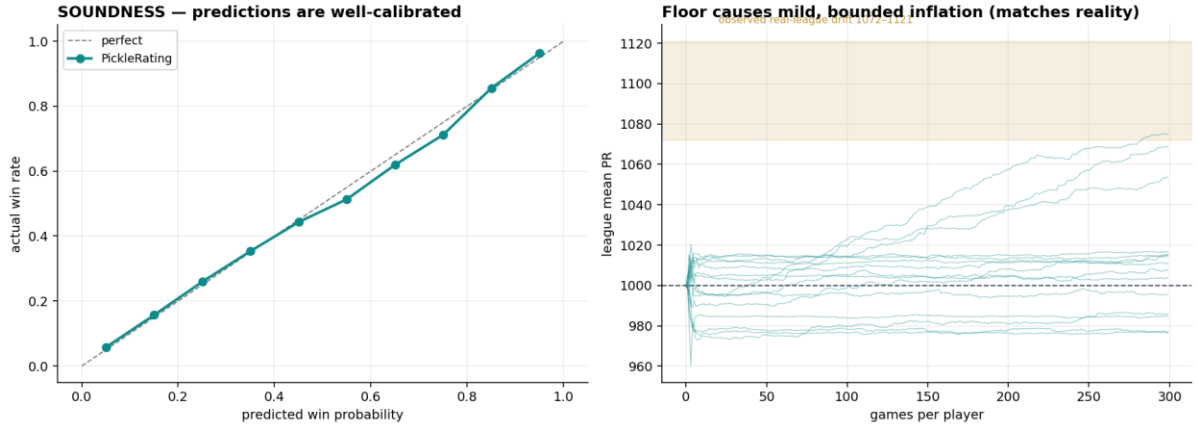


Figure 10.8

Left: predicted versus actual win rates lie on the diagonal (well-calibrated).

Right: league mean rating drifts modestly upward and settles within the range observed in the live system.

10.8 Summary of chosen values

Parameter	Group	Value	Role
c	Scale	1100	Rating scale; sets the ~1500 spread
c′	Scale	2750	Rally-model scale (margin reference), c′/c = 2.5
K	Dynamics	100	Per-match step; speed vs noise
RD_0	Dynamics	350	Max / initial RD; provisional strength
ψ	Dynamics	30	RD inflation; keeps ratings responsive
y	Shape	1.2	Margin-projection exponent
λ	Shape	0.10	Margin-surprise sensitivity
f_{min}	Shape	0.5	Lower f bound
f_{max}	Shape	1.5	Upper f bound (band mean 1.0)
start	Boundary	1000	Initial rating (Gold)
floor	Boundary	0	Minimum rating (no negative PR)
placement multiplier	Onboarding	3×	Gain boost during placement
placement games	Onboarding	5	Placement window length

11 Sample Calculations

For both singles and doubles three sample calculations will be run for 3 different scenarios: favored player/team wins, even matchup where the slightly less favored player/team wins, and significant upset where the heavy favorite loses. This allows us to see the PR system in action for 3 of the most important scenarios.

Each example uses the parameters of §10 (c = 1100, c′ = 2750, y = 1.2, K = 100, f_min = 0.5, f_max = 1.5, λ = 0.10). Both participants are treated as established players with RD = 175, so the damping factor RD/RD_0 in each update is approximately 0.5; the doubles examples additionally assume τ = 2000 and β_min = 0.2.

11.1 Singles

As a worked example, take player A (1300) against player B (1000), with A winning 11–6. The update proceeds as follows:

Expected score (1): $E_A = 1 / (1 + 10^{((1000 - 1300)/1100)}) \approx 0.652$

Rally probability (4): $p = 1 / (1 + 10^{((1000 - 1300)/2750)}) \approx 0.562$

Projected margin (5): $M_p = 11 \cdot (2p - 1)^{1.2} \approx +0.91$

Margin surprise (7, 8): $\Delta = M_A - M_p = (+5) - (+0.91) = +4.09$, with $s = \text{sign}(S_A - E_A) = +1$

f-factor (6): $f = 0.5 + (1.5 - 0.5) \cdot \sigma(0.10 \cdot 4.09) \approx 1.10$

Rating update (16): $\Delta PR_A = f \cdot (RD/RD_0) \cdot K \cdot (S_A - E_A) = 1.10 \cdot 0.5 \cdot 100 \cdot (1 - 0.652) \approx +19$

Player A therefore rises to 1319 and player B falls to 981. The remaining two scenarios follow the same steps:

Scenario	Ratings (A, B)	Score (A–B)	E_A	f	ΔPR (A / B)
Favored A wins	1300, 1000	11–6	0.65	1.10	+19 / −19
Even, B wins	1050, 1000	9–11	0.53	1.05	−28 / +28
Heavy upset, B wins	1500, 1000	8–11	0.74	1.11	−41 / +41

The favored win produces only a small gain, the even matchup a moderate swing, and the upset the largest correction.

11.2 Doubles

Doubles adds a single step: each team's skill is aggregated (§6) before the same update is applied. Take team A (1100, 1350) against team B (1000, 1050), with team A winning 11–7.

Team A skill (17, 19): $G = 250$, $\beta(G) = 0.2 + 0.3 \cdot e^{(-250/2000)} \approx 0.465$, so $TS_A = 1100 + 0.465 \cdot 250 \approx 1216$

Team B skill: $G = 50$, $\beta(G) \approx 0.493$, so $TS_B \approx 1025$

Expected score (20): $E_A = 1 / (1 + 10^{((1025 - 1216)/1100)}) \approx 0.599$

f-factor (6): $f \approx 1.09$

Rating update (35): $\Delta PR = 1.09 \cdot 0.5 \cdot 100 \cdot (1 - 0.599) \approx +22$

Each member of team A gains about +22 and each member of team B loses the same amount. The three doubles scenarios:

Scenario	Team A → TS_A	Team B → TS_B	Score	E_A	ΔPR each (A / B)
Favored A wins	(1100, 1350) → 1216	(1000, 1050) → 1025	11–7	0.60	+22 / −22
Even, B wins	(1050, 1150) → 1099	(1000, 1200) → 1094	9–11	0.50	−26 / +26
Heavy upset, B wins	(1300, 1500) → 1394	(950, 1050) → 999	9–11	0.70	−38 / +38

The aggregation pulls each team below the simple average of its two players, most visibly for the wider-gap pairings; aside from substituting TS for the individual rating, the doubles update is identical to the singles one.

Use of Generative AI

Some figures and numerical results in this paper were produced with the assistance of a generative AI model, Anthropic's Claude Opus 4.8 [5] including: the Monte Carlo simulations of §10, the parameter sweeps behind them, the sample calculations of §11, and all figures seen in the paper. All outputs were reviewed and validated.

References

[1] Elo, A. E. (1978). The Rating of Chessplayers, Past and Present. Arco Publishing.
[2] Glickman, M. E. (1999). “Parameter estimation in large dynamic paired comparison experiments.” Applied Statistics, 48(3), 377–394. Available: http://www.glicko.net/glicko/glicko.pdf
[3] Minka, T., Cleven, R., & Zaykov, Y. (2018). “TrueSkill 2: An improved Bayesian skill rating system.” Microsoft Research Technical Report MSR-TR-2018-8. Available: https://www.microsoft.com/en-us/research/wp-content/uploads/2018/03/trueskill2.pdf
[4] DUPR Inc. (n.d.). Dynamic Universal Pickleball Rating [Rating platform]. Retrieved June 25, 2026, from https://www.dupr.com
[5] Anthropic. (2026). Claude (Opus 4.8) [Large language model]. https://www.anthropic.com