How Bad Am I, Actually?

Building a Pickleball Rating System That Doesn't Lie

Split (Implementation) Dane Sabo (System Design)

February 2026

TL;DR: Four Ways We Made Your Rating More Honest

1. **Per-point scoring:** Instead of arbitrary bonuses for blowouts, we now calculate how many individual points you'd be expected to win against your opponent, then compare to reality.
2. **Fixed RD distribution:** New/returning players now update faster; established players update slower. It was backwards before.
3. **Partner credit:** In doubles, your rating change now accounts for how much your teammate carried you (or didn't). Strong partner? Lower effective opponent. Weak partner? Higher effective opponent.
4. **One unified rating:** Instead of separate singles/doubles ratings, you now have one rating that moves based on all matches.

Bottom line: Your rating will be more honest about how good you actually are.

1 Introduction

Welcome to the documentation of the Pickleball ELO System v2—an overhaul of how we measure skill in our recreational league.

If you've ever wondered why you seem amazing in doubles but mediocre in singles (or vice versa), or felt like your rating wasn't moving even though you're clearly getting better, you've hit on the very problems we're solving here.

Why We Built This

Recreational sports rating systems are hard. You can't just use win/loss records because match quality varies: beating a 1300-rated player is not the same as beating a 1600-rated player. Enter *Glicko-2*, a probabilistic rating system that adjusts expectations based on opponent strength and your own uncertainty.

We implemented Glicko-2 for our pickleball league, but over a year of matches, we discovered four systematic problems:

1. **Arbitrary margin bonuses:** The old system gave bigger rating boosts for lopsided wins. This worked okay, but it wasn't grounded in probability.
2. **Backwards RD distribution:** Confidence intervals (RD) were getting smaller updates, certainty larger updates—the opposite of what should happen.
3. **Team rating blindness:** In doubles, we averaged both players' ratings. A 1600-rated player paired with a 1400-rated player was treated identically to two 1500-rated players, which is nonsense.
4. **Rating bifurcation:** We maintained separate singles and doubles ratings, which felt artificial and made leaderboards confusing.

This document walks through the old system, why it failed, and how v2 fixes each problem.

2 The Old System (v1)

2.1 Glicko-2 Fundamentals

Glicko-2 is a probabilistic rating system. Instead of a single number, each player has three parameters:

Definition 1 (Glicko-2 Parameters)**.**

- **Rating (μ in Glicko-2 scale, R in display scale):** Your estimated skill level. In display scale, typical range is 1400–1600 for recreational players.
- **RD (Rating Deviation, ϕ in Glicko-2 scale):** Your uncertainty. Lower RD = more confident in your rating. New players start with RD ≈ 350. After many matches, RD converges to around 50–100.
- **Volatility (σ):** The long-term instability of your skill. Ranges 0.02–0.10. Higher if your skill fluctuates; lower if you're consistent.

When you play a match, Glicko-2 updates all three parameters:

$$\text{Expected Probability} = \frac{1}{1 + e^{-g(\phi_j)\cdot(\mu - \mu_j)}} \tag{1}$$

$$\text{Rating Change} \propto g(\phi_j) \cdot (\text{Actual Outcome} - \text{Expected}) \tag{2}$$

$$\text{New RD} \propto \sqrt{\phi_*^2 + \sigma'^2} \tag{3}$$

The key idea: if you beat someone you were *supposed* to beat (expected outcome ≈ 1), your rating barely moves. But if you upset a much stronger player (expected outcome $\ll 1$, actual outcome $= 1$), your rating jumps.

2.2 The Arbitrary Margin Bonus (v1)

The old system used a heuristic formula to convert match results into a *weighted score* fed into Glicko-2:

$$\text{Weighted Score} = \frac{1}{1 + e^{-\lambda \cdot m}} \tag{4}$$

where m is the margin of victory (points won minus points allowed) and λ is some tuning constant. In our case, we used a tanh approximation, which gave:

$$\text{Weighted Score} \approx 0.5 + 0.3 \cdot \tanh(m/5) \tag{5}$$

The problem: This formula was *arbitrary*. Why tanh? Why divide by 5? It worked okay in practice, but it had no theoretical foundation. It just... looked reasonable.

Example: A player rated 1500 beats a 1500-rated opponent 11–2.

$$\text{Margin} = 11 - 2 = 9$$
$$\text{Weighted Score} = 0.5 + 0.3 \cdot \tanh(9/5) \approx 0.79$$

But *why* is 0.79 the right number? The system didn't say.

2.3 Team Rating: Simple Average

In doubles, we computed the team rating as:

$$\text{Team Rating} = \frac{R_{\text{partner}} + R_{\text{self}}}{2} \tag{6}$$

And team RD as:

$$\text{Team RD} = \sqrt{\frac{\text{RD}_{\text{partner}}^2 + \text{RD}_{\text{self}}^2}{2}} \tag{7}$$

Then the team played Glicko-2 against the opposing team's aggregated rating.

The problem: A 1600-rated player paired with a 1400-rated player produces a team rating of 1500. But so does two 1500-rated players. These are *not* equivalent pairings:

- Scenario A: 1600 + 1400 → Team 1500. The 1600-rated player is carrying. If the team wins, the 1600-rated player overperformed and should get rewarded.
- Scenario B: 1500 + 1500 → Team 1500. Both players played at skill level. If the team wins, each should get normal credit.

The system couldn't distinguish between these cases.

2.4 The Backwards RD Distribution

When rating changes were distributed among doubles partners, the old code was:

$$\text{Weight}_{\text{partner}} = \frac{1}{\text{RD}_{\text{partner}}^2} \tag{8}$$

This means:

- Low RD (e.g., 100) → Weight = 1/10000 = 0.0001 (tiny fraction of rating change)
- High RD (e.g., 200) → Weight = 1/40000 = 0.000025 (even tinier!)

The logic was backwards. In Glicko-2, ratings with high uncertainty should converge *faster* to their true skill. A new player (RD 350) should see big rating swings; an established player (RD 50) should see tiny ones.

Instead, the old system did the opposite: established players got big changes, new players got small ones.

2.5 Separate Singles/Doubles Ratings

The database maintained six rating columns per player:

- singles_rating, singles_rd, singles_volatility
- doubles_rating, doubles_rd, doubles_volatility

This created:

1. **Psychological confusion:** Which rating matters? You're probably better at one format.
2. **Leaderboard ambiguity:** Do we show singles or doubles rank?
3. **Sample size issues:** Good players might have played 50 doubles matches but only 5 singles. Their doubles rating is more reliable, but they look worse at singles.

3 Why It Needed to Change

Over a year of matches, we observed several systematic issues:

1. **Blowout bonuses were too arbitrary:** A player could beat a much weaker opponent 11–2 and get a huge rating boost. But mathematically, what's the *expected* advantage? The system had no answer.
2. **New players weren't updating fast enough:** A new player (RD 350) who plays a match would get tiny rating changes. But they should be updating aggressively until their true skill is revealed!
3. **Strong partners were invisible:** A 1300-rated player paired with a 1600-rated player was rated as 1450. If they won, the 1300-rated player got normal credit for an easy win. If they lost against a 1400+1400 team, they got punished despite a weaker team.
4. **Rating bifurcation was confusing:** Players would complain: "My doubles is 1520 but my singles is 1480. Which one am I?"

4 The New System (v2)

4.1 Per-Point Expected Value

Instead of an arbitrary margin formula, we now compute the probability of winning each individual point and compare to reality.

Definition 2 (Point Win Probability)**.** Given two players with ratings R_{self} and R_{opp}, the probability that self wins a single point is:

$$P(\text{win point}) = \frac{1}{1 + 10^{(R_{\text{opp}} - R_{\text{self}})/400}} \tag{9}$$

This is the standard Elo formula, applied at the point level instead of the match level.

Definition 3 (Performance Ratio)**.** Over a match with p_{scored} points won and p_{allowed} points conceded:

$$\text{Performance} = \frac{p_{\text{scored}}}{p_{\text{scored}} + p_{\text{allowed}}} \tag{10}$$

This is the *actual* fraction of points won.

The weighted score fed into Glicko-2 is now:

$$\boxed{\text{Weighted Score} = \frac{p_{\text{scored}}}{p_{\text{scored}} + p_{\text{allowed}}}} \tag{11}$$

Why this works:

- **Probabilistically sound:** Each point is an independent trial. If you're expected to win 64% of points and you win 55%, you underperformed.
- **Scale-invariant:** An 11–2 match and an 11–9 match are both graded on *how many individual points you won* relative to expectation, not the margin.
- **Fair to upsets:** A 1400-rated player upsetting a 1500-rated player 11–9 (55% of points) is *expected* to win $\approx 40\%$ of points. They won 55%, a 15-point overperformance. Big rating boost—correctly earned!

Example calculation:

$$\begin{aligned} R_{\text{self}} &= 1500 \\ R_{\text{opp}} &= 1500 \\ P(\text{point}) &= \frac{1}{1 + 10^{0/400}} = 0.5 \\ \text{Actual points won} &= 11 \\ \text{Total points played} &= 20 \\ \text{Performance} &= 11/20 = 0.55 \end{aligned}$$

Glicko-2 sees outcome = 0.55, expected outcome = 0.50, and adjusts the rating accordingly. Clean, principled, done.

4.2 Fixed RD Distribution

The new distribution formula is:

$$\boxed{\text{Weight}_{\text{partner}} = \text{RD}^2_{\text{partner}}} \tag{12}$$

If the team gets a rating change of ΔR:

$$\Delta R_1 = \Delta R \cdot \frac{\text{RD}_1^2}{\text{RD}_1^2 + \text{RD}_2^2} \tag{13}$$

$$\Delta R_2 = \Delta R \cdot \frac{\text{RD}_2^2}{\text{RD}_1^2 + \text{RD}_2^2} \tag{14}$$

Why this is correct:

- **Higher RD = more uncertain = update faster:** A new player (RD 350) paired with an established player (RD 75) will get 95% of the rating change. Their rating should move aggressively until we know what they really are.
- **Follows Glicko-2 principle:** Glicko-2 adjusts uncertain ratings more because uncertainty is bad. The system converges faster when you provide larger updates to uncertain ratings.

Numerical example:
Suppose a doubles pair wins a match and the team rating goes up by 20 points:

$$\begin{aligned}
\text{Partner 1 RD} &= 100 \text{ (experienced)} \\
\text{Partner 2 RD} &= 200 \text{ (new)} \\
\text{Weight}_1 &= 100^2 = 10,000 \\
\text{Weight}_2 &= 200^2 = 40,000 \\
\text{Total Weight} &= 50,000 \\
\Delta R_1 &= 20 \cdot \frac{10,000}{50,000} = 4 \text{ points} \\
\Delta R_2 &= 20 \cdot \frac{40,000}{50,000} = 16 \text{ points}
\end{aligned}$$

The experienced player gets +4, the new player gets +16. Much more sensible!

4.3 Effective Opponent Calculation

In doubles, each player now faces a personalized effective opponent rating:

$$\boxed{R_{\text{eff}} = R_{\text{opp1}} + R_{\text{opp2}} - R_{\text{teammate}}} \tag{15}$$

Intuition:

- Strong opponents make it *harder* → higher effective opponent rating
- Strong teammate makes it *easier* → lower effective opponent rating (they helped you)

- Weak teammate makes it *harder* → higher effective opponent rating (you did all the work)

Examples:

R_{opp1}	R_{opp2}	R_{teammate}	R_{eff}
1500	1500	1500	1500
1500	1500	1600	1400
1500	1500	1400	1600
1600	1400	1500	1500

In the second row, you have a strong teammate (1600) against average opponents (1500 each). Your effective opponent is rated 1400—you're expected to win more points because your partner is helping.

In the third row, you have a weak teammate (1400) against the same opponents. Your effective opponent is now 1600—you're expected to win fewer points because you're carrying.

Why this matters:

The Glicko-2 algorithm uses the effective opponent rating to compute $P(\text{expected outcome})$.

With the old system, a 1600-rated player paired with a 1400-rated teammate would face an effective opponent of 1500 (simple average). If they beat a pair of 1500-rated players, the algorithm thought the team was evenly matched.

With the new system, the 1600-rated player sees the effective opponent as 1500 - 100 = 1400. If they win against a 1500 + 1500 team, they've beaten a slightly harder team than their rating suggests. Their rating increases slightly less than if they faced a true 1400-rated pair.

This is subtle but important: it rewards you for winning despite a weak partner, and penalizes (slightly) your rating gains when winning with a strong partner.

5 A Worked Example

Let's walk through a concrete match using both v1 and v2 to see the differences.

Match Setup

Singles match:

- **Player A:** Rating 1500, RD 150, Volatility 0.06
- **Player B:** Rating 1550, RD 150, Volatility 0.06
- **Result:** A wins 11–9

v1 Calculation

Step 1: Compute weighted score using margin bonus:

$$\begin{aligned} m &= 11 - 9 = 2 \\ \text{Weighted Score} &\approx 0.5 + 0.3 \cdot \tanh(2/5) \\ &\approx 0.5 + 0.3 \cdot 0.37 \\ &\approx 0.611 \end{aligned}$$

Step 2: Feed into Glicko-2 as outcome = 0.611.
Step 3: Glicko-2 computes:

$$\begin{aligned} \text{Expected outcome} &\approx 0.47 \text{ (player A is rated lower)} \\ \text{Actual outcome} &= 0.611 \\ \text{Overperformance} &= 0.141 \\ \text{Rating change} &\approx +8 \text{ to } +10 \text{ points} \end{aligned}$$

v2 Calculation

Step 1: Compute performance-based score:

$$\text{Performance} = \frac{11}{11+9} = 0.55$$

Step 2: Feed into Glicko-2 as outcome = 0.55.
Step 3: Glicko-2 computes:

$$\begin{aligned} \text{Expected outcome} &\approx 0.47 \text{ (same as before)} \\ \text{Actual outcome} &= 0.55 \\ \text{Overperformance} &= 0.08 \\ \text{Rating change} &\approx +5 \text{ to } +7 \text{ points} \end{aligned}$$

Comparison

Metric	v1	v2
Weighted Outcome	0.611	0.55
Overperformance	+14.1%	+8.0%
Rating Gain	+10 pts	+6 pts

Why the difference?

v1's margin bonus (0.611) inflated the outcome because the match was somewhat lopsided (11–9). v2's performance ratio (0.55) is more conservative: Player A won 55% of points when expected to win 47%.

In this case, v2 is *fairer*. A 2-point win over a slightly stronger opponent should yield modest rating gains, not aggressive ones. If Player A is actually better, they'll demonstrate it over many matches. One 11–9 win isn't definitive.

Doubles Example

Now consider a doubles match:

- **Team A:** Players A (1500) + B (1300)
- **Team B:** Players C (1550) + D (1450)
- **Result:** Team A wins 11–9

v1 Doubles Rating

$$\text{Team A rating} = (1500 + 1300)/2 = 1400$$
$$\text{Team B rating} = (1550 + 1450)/2 = 1500$$

Team A (rated 1400) beats Team B (rated 1500) 11–9. Expected outcome for A ≈ 0.40. Actual outcome = 0.611 (using margin bonus). Huge upset! Both players on Team A get large rating gains.

v2 Doubles Rating

Performance outcome: 0.55 (as before).

But now each player sees a different effective opponent:

For Player A (rated 1500):

$$\begin{aligned} R_{\text{eff}} &= R_{\text{C}} + R_{\text{D}} - R_{\text{B}} \\ &= 1550 + 1450 - 1300 \\ &= 1700 \end{aligned}$$

Expected outcome vs. 1700-rated opponent: ≈ 0.23. Actual: 0.55. Massive upset! Player A gets large rating gains.

For Player B (rated 1300):

$$\begin{aligned} R_{\text{eff}} &= 1550 + 1450 - 1500 \\ &= 1500 \end{aligned}$$

Expected outcome vs. 1500-rated opponent: ≈ 0.31. Actual: 0.55. Decent upset. Player B gets moderate rating gains.

Key difference: v2 recognizes that Player A (the 1500-rated strong player) did the carrying work. They face a harder effective opponent and get rewarded more for the win. Player B gets credited fairly for their contribution.

6 Discussion: Tradeoffs and Future Work

6.1 Why v2 Is Better

1. **Principled:** Every formula is grounded in probability theory, not heuristics.
2. **Fair to uncertainty:** New and returning players update faster, as they should.

3. **Personalized doubles:** Partner strength now matters; you're not rewarded for winning with a carry.

4. **Simpler to explain:** "Your rating updates based on the fraction of points you actually won vs. expected."

6.2 Tradeoffs and Concerns

1. **Smaller rating swings:** v2 tends to award smaller updates per match. This is intentional and correct, but might *feel* slower. Rest assured: over a season, your rating will converge to your true skill level faster.

2. **Blowout wins are less rewarding:** An 11–2 match gives the same outcome (0.846) regardless of opponent strength. Is this fair? Yes—you won 84.6% of points. The magnitude of your overperformance is what matters, not the opponent's feelings.

3. **Doubles partner dependency:** Your rating now depends slightly on who you play with. Pairing with stronger players gives you lower effective opponents, slightly smaller gains. This is correct: you should be rewarded less for beating weaker teams.

4. **RD still converges slowly:** Even with correct distribution, RD converges gradually. A new player might take 30–50 matches to stabilize. This is by design (Glicko-2 is conservative), but it means new players are volatile.

6.3 What v2 Still Doesn't Address

1. **Player improvement over time:** Glicko-2 assumes your skill is stationary. If you've been training and are getting better, your volatility increases—which is correct, but it delays rating convergence.

2. **Format differences:** Your unified rating is now used for singles and doubles. If you're much better at one format, the rating will be a compromise. Future work: weight by match type or maintain separate histories.

3. **Population drift:** All ratings are calibrated to a population mean of 1500. If the player base gets stronger or weaker, old ratings become less meaningful. (This is true of all Elo-based systems.)

4. **Match quality:** Glicko-2 doesn't account for match importance, time of day, or other external factors. Two 11–9 matches are scored identically, even if one was high-pressure and one casual.

6.4 Possible Future Improvements

1. **Time-based rating decay:** If a player hasn't played in 6 months, increase their RD to reflect the uncertainty.

2. **Match quality weighting:** Tournament matches could carry higher weight than casual league play.

3. **Format-specific ratings (optional):** Maintain separate ratings but with shared history. A strong singles player gets a boost in doubles for free, but can specialize later.

4. **Skill ratings by court:** Rating adjustments could account for court quality, wind, etc. (This is probably overkill for recreational pickleball.)

5. **Win streak bonuses:** In traditional sports, momentum is real. A streak of wins might deserve an extra boost. (Again, this adds complexity for marginal gains.)

7 Conclusion

The Pickleball ELO System v2 addresses four major flaws in v1:

1. **Per-point expected value** replaces arbitrary margin bonuses with probabilistic reasoning.

2. **Correct RD distribution** ensures new players improve their ratings quickly.

3. **Effective opponent calculations** personalize doubles ratings by partner strength.

4. **Unified ratings** simplify the system while still tracking match type for future analysis.

The math is cleaner. The results are fairer. Your rating now reflects not just wins and losses, but *how well you actually played relative to expectation.*

Is it perfect? No. Is it a massive step forward? Absolutely.

So go out there, play some pickleball, and find out exactly how bad you actually are. (The data doesn't lie—not anymore!)

For technical details, see the Rust implementation in `src/glicko/` *and the test cases in each module.*