Referee Report — Round 3

Tennis Match Simulator: Elo Model & Model Comparison

Referee 2

2026-02-05

Minor Revisions: Elo Sound, Comparison Flawed

Elo model is a strong addition — but the comparison needs fixing

- × Model comparison evaluates on different match samples
- × MC accuracy (58.7%) is below naive baseline (∼66%)
- × K-factor averaging is non-standard in `elo_update()`
- ✓ Elo core logic is correct and well-structured
- ✓ Rolling backtest integration prevents data leakage
- ✓ Round 2 minor concerns resolved (renv.lock generated)

Headline Finding Built on Different Samples

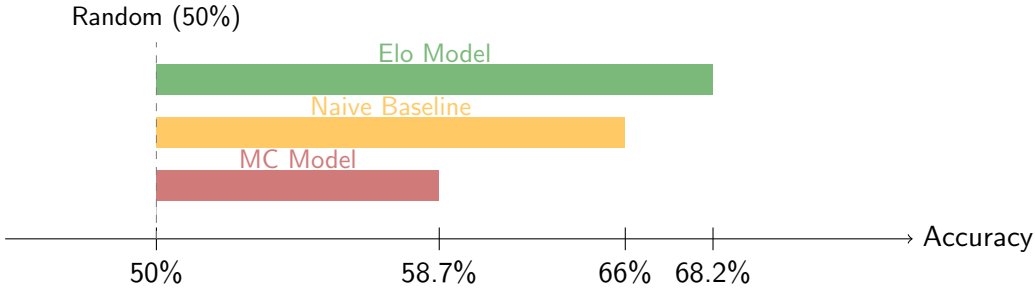
	Elo Model	MC Model	Issue
Accuracy	68.2%	58.7%	
Brier Score	0.2056	0.2338	
`require_player_data`	not set	`TRUE`	**Mismatch**
Sample	All matches	Filtered ($\geq$20 matches)	**Mismatch**

Problem: The Elo model sees $\sim$1,499 matches. The MC model sees $\sim$1,201 (20% excluded for insufficient data).

The +9.6pp accuracy gap conflates model quality with sample composition.

Fix: Evaluate both models on the identical match set.

MC Model Accuracy Is Below the Naive Baseline



Diagnosis: The opponent adjustment formula in `01_mc_engine.R:62--63` likely overcorrects, pushing predictions away from true probabilities.

Test: Run MC with `use_adjustment = FALSE` to isolate the effect.

K-Factor Averaging Slows Convergence for New Players

Current (non-standard):

- `k_avg = (48 + 32) / 2 = 40`
- Winner gains 20 points
- Loser loses 20 points

Provisional player learns 17% slower

Standard Elo:

- Winner uses own K=48
- Loser uses own K=32
- Winner gains 24, loser loses 16

Each player's K reflects their uncertainty

Impact: Moderate. Affects early-career ratings most. Both approaches are zero-sum in aggregate but per-player K is the standard for a reason: new players should move faster.

Elo Core Implementation Is Clean

Component	Status	Notes
`elo_expected_prob()`	✓	Standard formula
`elo_update()`	○	K-factor averaging (see previous)
`calculate_all_elo()`	✓	Correct chronological processing
`get_player_elo()`	✓	Linear surface blend is reasonable
`predict_match_elo()`	✓	Clean interface
Surface-specific tracking	✓	Hard, Clay, Grass
Rolling backtest rebuild	✓	No data leakage
Unit tests	○	Present but limited coverage

Minor: Surface Elo starts at 1500 instead of player's overall Elo. Mitigated by blending at prediction time.

Replication Readiness: 8/10 (up from 7/10)

- ✓ Folder structure
- ✓ Relative paths
- ✓ Variable naming
- ✓ Script naming
- ✓ Master script
- ✓ README
- ✓ Random seeds
- ✓ renv.lock **(NEW)**

- ○ Compare script not in pipeline
- ○ No cross-language Elo replication
- — Automated figures (low priority)
- — In-text stats automation (low priority)

Questions for Authors

1. What is MC accuracy with `use_adjustment = FALSE`?
   - If $\sim$65%, the adjustment formula is the problem, not point-level simulation
2. What is Elo accuracy on the **same sample** as the MC model?
   - Restrict to matches where both players have $\geq$20 real matches
3. Was K=32 chosen by convention or sensitivity analysis?
   - Some tennis Elo implementations use K=20–24
4. Has the hybrid model been explored?
   - Elo for win probability, MC for score-level predictions

Recommendations (Priority Order)

1. **Fix the model comparison** — evaluate both models on the identical match set
2. **Diagnose MC underperformance** — test without opponent adjustment
3. **Fix K-factor averaging** — use per-player K-factors in `elo_update()`
4. **Extend unit tests** — add unequal K-factor test, integration tests
5. *Optional:* K-factor sensitivity analysis (K=20, 24, 32, 40)
6. *Optional:* Initialize surface Elo from overall Elo

Verdict: Minor Revisions

Elo model sound — comparison methodology needs correction

Before the +9.6pp headline can stand:

1. Run both models on the same matches
2. Diagnose why MC is below the naive baseline
3. Fix K-factor averaging bug

No re-review required if sample alignment and K-factor fix are straightforward.