PGV: A TWO-CHECK VERIFICATION HARNESS FOR SELF-PLAY PROMOTION GATES, VALIDATED ON FOUR REAL INCIDENTS IN DEPLOYED SYSTEMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Competitive self-play reinforcement learning relies on a promotion gate to decide which candidate policy is actually better. When the gate's own measurement is wrong in a way that is not obviously noise, the loop can silently stall, silently promote a policy that has not improved, or silently reuse data it presents as independent. We introduce **PGV**, a lightweight verification harness of two offline checks for self-play promotion gates, each a one-line invariant over bookkeeping the gate already keeps: the *Disjoint-Reseed Check* (a candidate's certifying data, and any data offered as an independent replication, must occupy regions of the seed space disjoint from the selection sweep and from each other) and the *Train/Eval Symmetry Check* (every structural degree of freedom the evaluation condition exposes – seat, side, deck – must be sampled the way training samples it). Neither check was run in either of two independently built, deployed self-play systems we studied. We validate PGV on four real incidents rather than a synthetic benchmark. The Disjoint-Reseed Check flags two real certification failures – a candidate certified as best of roughly twenty with a $+0.0054$ placement-point edge that falls to -0.0005 (interval spanning zero) under selection-disjoint fresh evaluation, consistent with a winner's curse, and a seed-stepping bug that made a nominal "independent replication" silently reuse 58.33% of the original run's blocks – and, applied to a different, later improvement claim in the same system, *passes*: five mutually disjoint evaluations all clear the calibrated tie point, so the check does not drive every claim to null (of two independently trained replicas, one clears on its own; their pooled estimate clears). The Train/Eval Symmetry Check would have flagged the imbalanced gate condition behind a 124-generation freeze of an independently built card-game ladder, diagnosed from a seat-conditional win-rate split (0.14 vs. 0.66); the mahjong gate avoids the identical mismatch by explicit design, and a pre-registered synthetic reproduction shows the mechanism's direction only at larger biases while falsifying its own pre-registered magnitude threshold, reported as falsified. Every number traces to a committed file or a pinned upstream commit, and an evidence plan frozen before the analyses fixed how each claim is narrowed when its own check comes back negative, which happened twice and is reported both times.

1 INTRODUCTION

Competitive self-play systems train a policy against evolving versions of itself and use a promotion gate to decide which candidate survives into the next generation of training data. The gate's only job is to measure whether a candidate is actually better; if its measurement is wrong in a way that is not obviously noise, the loop can silently stall, silently promote a policy that has not improved, or silently reuse data it presents as fresh. These are not training-algorithm failures. They are properties of the harness that measures training's output, and they are correspondingly easy to miss, because training curves and reported win rates both look ordinary from the outside.

Two distinct sources of harness error motivate this paper, and each admits a check simple enough to run automatically alongside a gate that logs its seeds and evaluation conditions. The first is selection bias: choosing the best of many noisy estimates inflates the chosen estimate even when every

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

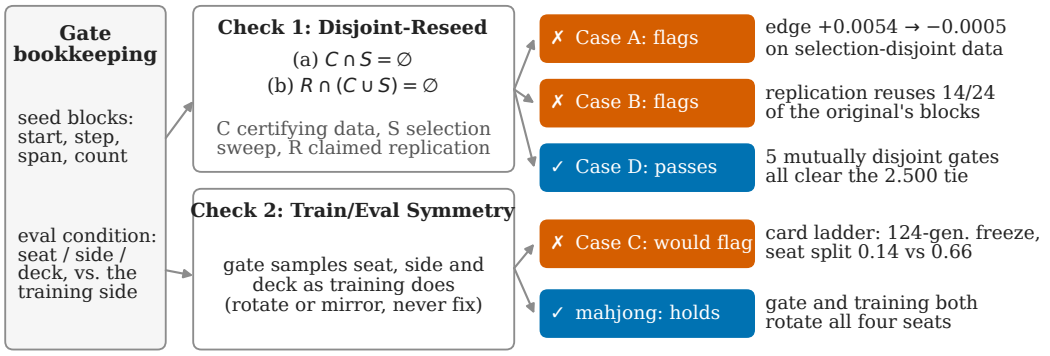

Figure 1: PGV at a glance. Both checks read bookkeeping a promotion gate already keeps (left) and test one invariant each (middle). Right: the four real incidents. The Disjoint-Reseed Check flags two certification failures in a mahjong self-play system (Cases A, B) and passes a later claim in the same system (Case D); the Train/Eval Symmetry Check would have flagged a card-game ladder's freeze (Case C) and holds by design in the mahjong gate. Schematic; the printed values are those of Tables 2–3 and Section 5.1.

individual estimate is unbiased, a mechanism named in decision analysis (Smith & Winkler, 2006) and, in 2026, for adaptive benchmarking (Xu et al., 2026). The correcting check is a disjointness invariant over the gate's own seed bookkeeping: certifying data, and any data offered as an independent replication, must occupy seed regions that selection and every other claimed-independent stage never touched. The second is protocol-specific: a gate's evaluation condition can carry a structural asymmetry (which seat, side, or deck a candidate is assigned) that training does not share, silently capping every candidate's measured performance regardless of skill. The correcting check is a symmetry invariant. Both are offline audits of artifacts the gate already produces.

We call the pair **PGV** (Promotion-Gate Verifier) and validate it on four real incidents from two independently built, deployed self-play systems – a Chinese Standard Mahjong agent built for the IJCAI/Botzone competition platform, and an unrelated competitive card-game ladder we refer to as "Boom" – plus one synthetic check of a mechanism no real system can isolate on its own. Cases A–C were written into a committed evidence plan, with success and failure criteria, before their analyses were run, and each claim's pre-written failure action governs how it is reported if its check comes back negative. The plan does not make those cases prospective: their data already existed when it was frozen, so it fixes the analysis and its reporting, not the outcome, and Case D was added after the freeze. Only the mahjong symmetry inspection and the synthetic reproduction had unknown outcomes when their criteria were frozen; "pre-registered" here means frozen in a committed file, not a public registry. Figure 1 summarizes our contributions: (i) PGV's two invariants over a gate's seed and condition bookkeeping (Section 3); (ii) two real incidents the Disjoint-Reseed Check catches, a winner's-curse decay of a certified $+0.0054$ edge to a non-significant -0.0005 (Section 4.1) and a seed-stepping bug that made $14/24 = 58.33\%$ of a nominal replication's blocks reuse the original run's (Section 4.2); (iii) one real incident in which the same check passes a different claim (Section 4.3); (iv) one real incident the Train/Eval Symmetry Check would have flagged, a 124-generation ladder freeze absent by explicit design from the mahjong gate (Section 5.1), with a pre-registered synthetic reproduction whose magnitude threshold was falsified (Section 5.2); and (v) one pre-specified ablation that remains blocked by unavailable model weights (Section 6).

2 Related Work

Selecting the best of many noisy estimates. The mechanism behind Case A is old. Capen et al. (1971) named the winner's curse in competitive bidding; Smith and Winkler's (2006) "optimizer's curse" proves that choosing the alternative with the highest *estimated* value biases that estimate upward even when every estimate is individually unbiased – exactly the shape of a best-of-N gate. Gelman and Loken (2013) state the same point for data-contingent analysis choices. In supervised learning the fix is nested cross-validation: Varma and Simon (2006) and Cawley and Talbot (2010)

show that reusing the selection data to report performance yields optimistic estimates, and Nie et al. (2018) show the analogous bias in adaptively collected bandit data. Closest in time and mechanism, SIREN (Xu et al., 2026) proposes, for adaptive language-model benchmarking, the fix PGV's first check generalizes: separate the data used to select a winner from the data used to report it. None gives an invariant a self-play gate can check against its own seed bookkeeping, or measures the mechanism inside a deployed promotion pipeline beside a case where the same check passes; that two-sided validation (Sections 4.1–4.3), not a new statistical claim, is our contribution.

Empirical rigor in reinforcement-learning evaluation. A line of methodology work documents how seeds, variance, and reporting choices distort single-agent RL comparisons: Henderson et al. (2018) on seed sensitivity, Colas et al. (2018) on how many seeds a comparison needs, Jordan et al. (2020) on evaluation protocols, and Agarwal et al. (2021) on interval estimates for few-run benchmarks. Patterson et al.'s (2024) survey names "maximization bias" in hyperparameter selection and states plainly that RL lacks a nested-cross-validation analogue, but stops at single-agent hyperparameter selection and never treats promotion gates. Eimer et al. (2023) show that tuned RL configurations depend strongly on the tuning seed and recommend separate tuning and test seeds; condition (a) of the Disjoint-Reseed Check carries that practice to a promotion gate's block-seed bookkeeping, and condition (b) extends it to runs offered as replications, which their recommendation does not cover. Kapoor and Narayanan (2023) catalogue leakage between training and evaluation data across many scientific fields; Case B is a leakage of a different kind – between two evaluation runs presented as independent – that their train/test taxonomy does not cover and that no source we found describes.

Self-play, league training, and gating. AlphaGo Zero (Silver et al., 2017) promotes a challenger only once it clears a fixed win-rate threshold against the incumbent; AlphaZero (Silver et al., 2018) drops the evaluator and updates a single network continually; KataGo (Wu, 2019; lightvector, 2024) keeps candidate-network gating in its public pipeline, and its issue tracker reports no gate freeze via asymmetry. All three play symmetric two-player games, where the Train/Eval Symmetry Check is trivially satisfied. AlphaStar's league (Vinyals et al., 2019), population-based training in Capture the Flag (Jaderberg et al., 2019), and OpenAI Five (Berner et al., 2019) replace a binary gate with rating-based population evaluation; the AlphaStar full text mentions only "frozen" opponent parameters, and treats StarCraft's race asymmetry as a property of the game, not as an evaluator artifact. Czarnecki et al. (2020) (full text searched) do not discuss an evaluation condition diverging from training's. The empirical game-theoretic line – PSRO (Lanctot et al., 2017), its sample-complexity analysis (Tuyls et al., 2018), Nash averaging for agent evaluation (Balduzzi et al., 2018), and the recent survey (Bighashdel et al., 2024) – concerns which solution concept scores a meta-game; its closest passage, Wang et al.'s (2021) warning against scoring exploration with the solver that generated the strategies, concerns the scoring concept, not a threshold made unwinnable by a mismatched seat or deck condition. None of these sources discusses ladder freezing, seat-conditional win-rate diagnostics, or a checkable symmetry invariant.

Mahjong and duplicate evaluation. Suphx (Li et al., 2020), the closest domain-matched system, was searched full text for seat, position, and compensation language and does not discuss promotion-gate symmetry or freezing. AIVAT (Burch et al., 2018), from poker duplicate evaluation, reduces variance within one match via mirrored deals and a control variate; it does not address a gate whose asymmetric condition keeps every candidate from passing.

Positioning. The Disjoint-Reseed Check packages a known mechanism as a checkable invariant; the Train/Eval Symmetry Check is, as far as the search above can establish, not previously documented as one (Table 1). We state what we searched rather than assert an uncheckable negative.

3 PGV: A Two-Check Verification Harness

The system studied. The primary system is a Chinese Standard Mahjong self-play agent whose promotion pipeline gates a candidate through a duplicate-wall evaluation: the same shuffled wall is replayed once with the candidate in each of the four seats against three copies of a reference policy, and a placement score (higher is better) is compared against a calibrated three-way-tie point of 2.500. Evaluation runs are organized into blocks of duplicate walls, each block identified by

Table 1: What each line of prior work addresses, as read in the sources of Section 2 (a summary of that reading, not a rating). ✓: addressed; ✗: not addressed in the full text or sources we searched; –: not applicable. Last column: stated as an invariant a gate can check on its own logs.

Line of work	Setting	Selection bias	Replication independence	Train/eval symmetry	Checks gate logs
Selection and leakage					
Winner's / optimizer's curse	decision analysis	✓	✗	✗	✗
Nested cross-validation	supervised selection	✓	✗	✗	✗
SIREN	LLM benchmarking	✓	✗	✗	✗
RL evaluation rigor[a]	single-agent RL	✓	✗	✗	✗
Leakage taxonomy	many fields	✗	✗	✗	✗
Self-play and game evaluation					
AlphaGo Zero / KataGo gating	symmetric games	✗	✗	–[b]	✗
League, PSRO, Nash avg.	population rating	✗	✗	✗	✗
Suphx; AIVAT[c]	mahjong; poker	✗	✗	✗	✗
PGV (ours)	self-play gates	✓	✓	✓	✓

[a] For hyperparameter tuning only: Patterson et al. name maximization bias; Eimer et al. separate tuning and test seeds. [b] No seat or side asymmetry exists to mismatch. [c] AIVAT mirrors deals within one match to reduce variance; it is not a check on a gate's condition.

Table 2: Decay of a certified best-of-$\sim$20 candidate's placement edge over the calibrated 2.500 tie. Cells: mean $\pm$ 95% t-interval half-width over 24 block means (2,000 games per block). Check 1: which Disjoint-Reseed condition the row's data violate (✗) or satisfy (✓). The edge shrinks on same-region fresh data and falls below zero once condition (a) holds; the two fresh rows are not distinguishable from each other (Section 4.1). Rows are copied from the system's gate audit.

Stage	Check 1	Placement	Edge over tie
Selection-contaminated data			
Certified (original gate)	✗ (a)	2.5054 ± 0.0042	$+0.0054$
Nominal replication (Case B)	✗ (b)	2.5057 ± 0.0039	$+0.0057$
Same-region fresh re-gate	✗ (a)	2.5017 ± 0.0026	$+0.0017$
Selection-disjoint data			
Fresh region (48,000 games)	✓	2.4995 ± 0.0024	-0.0005

a deterministic starting seed and a fixed span of walls; a gate run concatenates a fixed number of blocks. This block-seed structure is what the Disjoint-Reseed Check audits. The metric, success threshold, and failure action of each claim below were frozen before its analysis was run (Section 1 states what this does and does not constrain); source versions are pinned and claim-relevant source excerpts are hashed at retrieval (Reproducibility Statement). PGV audits the gate's existing seed and condition logs after the fact, with no change to training.

Check 1: the Disjoint-Reseed Check. A gate certifies a candidate C as beating a reference using evaluation data D_{cert} drawn from a seed space S. If C was chosen as the best of N candidates swept over a selection region $S_{\mathrm{sel}} \subseteq S$, and a later run offers D_{repl} as an independent replication, the check requires (a) $D_{\mathrm{cert}} \cap S_{\mathrm{sel}} = \emptyset$ if the certified edge is to be read as an unconditional estimate of C's strength rather than a selection-conditional one, and (b) $D_{\mathrm{repl}} \cap D_{\mathrm{cert}} = \emptyset$ and $D_{\mathrm{repl}} \cap S_{\mathrm{sel}} = \emptyset$ if D_{repl} is to count as independent confirmation. Both are computable from the gate's block-seed generator (start, step, span, count) by pairwise interval intersection, with no evaluation re-run. A violation of (a) predicts winner's-curse shrinkage under fresh, disjoint re-measurement (Section 4.1); a violation of (b) means a "replication" is not independent evidence (Section 4.2); Section 4.3 reports the case where both hold and the certified edge survives.

Check 2: the Train/Eval Symmetry Check. Let $\mathcal{A} = \{a_1, \ldots, a_k\}$ be the structural degrees of freedom a game exposes that are not skill-relevant on their own (seat, side, deck, board orientation). The check requires that the evaluation condition sample or rotate across $\mathcal{A}$ the same way training does. If training samples $\mathcal{A}$ uniformly (or mirrors across all of it, as a duplicate design does) while evaluation fixes some a_i asymmetrically, every candidate's measured performance is capped by

that asymmetry's effect size regardless of true skill, and a fixed promotion threshold can become structurally unwinnable – a freeze the gate's own win rates do not distinguish from "no candidate has improved." Section 5.1 reports a real system where this check would have flagged an active 124-generation freeze; Section 5.2 reports a pre-registered synthetic reproduction of the mechanism.

Both checks are deliberately narrow: they audit the two relationships a gate's bookkeeping can misrepresent (data independence and condition symmetry) and make no claim about optimization, exploration, or reward design.

4 Validating the Disjoint-Reseed Check

4.1 Case A (catch): Winner's Curse in a Certified Best-of-Approximately-20 Gate

A candidate certified by the mahjong system's knowledge-distillation (KD) ensemble gate as the best of roughly twenty gated candidates carries a placement-point edge of $+0.0054$ over the 2.500 tie point[1], on data that violate condition (a): the candidate was chosen from the sweep that produced D_{cert}. Table 2 reports this edge, its nominal replication, and two re-evaluations under progressively less selection-contaminated conditions, each read from the system's stored gate results.

Same-region fresh re-evaluation – new wall seeds from the region the selection process itself drew from, still violating (a) – shrinks the edge to $+0.0017$, with an interval that reaches below the tie; it is not statistically distinguishable from the certified edge ($z \approx 1.5$). A region disjoint from the selection process, satisfying (a), gives -0.0005 with 95% interval $[-0.0029, +0.0019]$, spanning zero, and this *is* distinguishable from the certified edge ($z \approx 2.5$, two-sided $p \approx 0.01$; Holm-adjusted over the three pairwise comparisons, $p \approx 0.04$), so by its frozen failure criterion Case A is not narrowed to a null. The two fresh estimates are ordered as the winner's-curse account predicts but are not distinguishable from each other (upstream audit: $z = 1.05$; recomputed from Table 2's intervals: $z \approx 1.3$). All z values we recompute are unpaired normal approximations from the four table rows. The data show that the certified edge does not survive selection-disjoint evaluation, not that same-region fresh data retain the bias. We present Case A as an instrumented demonstration of the mechanism Smith and Winkler (2006) and SIREN (Xu et al., 2026) describe, at the size of a deployed system's headline result, not as a novel finding.

4.2 Case B (catch): A Seed-Stepping Bug That Silently Reused 58.33% of a Replication

The nominal replication in Table 2 reused most of the original run's evaluation data because of a seed-stepping bug in the block-seed generator, a violation of condition (b). The original gate's 24 blocks start at wall-seed $350{,}000$ and step by $1{,}000$, each spanning 500 walls; the replication's start at $360{,}000$ with the same step and span (Figure 2a). Its start is only ten block-steps above the original's, while each run spans $24{,}000$, so the ranges $[350{,}000, 374{,}000)$ and $[360{,}000, 384{,}000)$ overlap on a span of $14{,}000$: exactly 14 of the replication's 24 blocks, $14/24 = 58.33\%$, have seed ranges identical to an original block. The two passing runs the pipeline treated as independent confirmations are not independent.

We re-derived the overlap from the two launcher scripts' own seed loops by brute-force interval intersection over all 24 blocks per run; it agrees with the closed form above and with the original audit exactly. The cause is deterministic arithmetic, caught by exactly the pairwise test condition (b) specifies, which the pipeline had never automated.

The reuse is visible in the scores. Seed arithmetic says which blocks share walls; the stored per-block scores show what that did (Figure 2b). Both launchers store their block scores in lexicographic rather than numeric order of block index (0, 1, 10, ...), which we invert to seed order. Each of the 14 replication blocks on a shared range then reproduces its original block's placement score to all four stored decimals; pairing the stored lists naively matches none of the 14. The replication's 24-block mean, 2.5057 (95% interval $[2.5018, 2.5097]$), therefore contains 14 repeated measurements. Its 10

[1]The stored gate result records a mean of 2.5054 (Table 2); the system's own evaluation summary quotes this edge as $+0.0055$ from unrounded block data. We print the stored value.

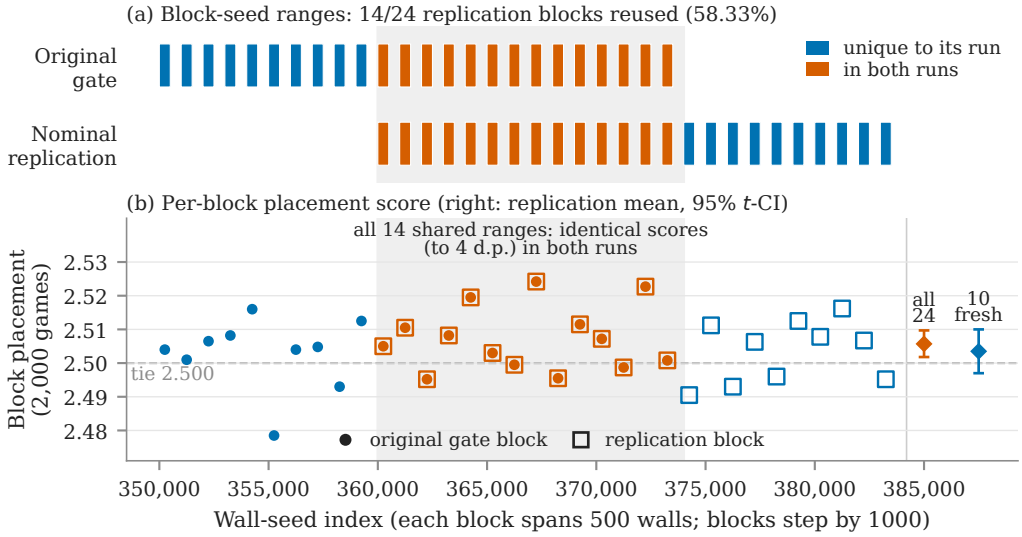

Figure 2: Case B on the real blocks. (a) Block-seed ranges of the original gate and its nominal replication; shaded: the overlap of the two 24-block ranges. (b) Every block's placement score, placed at its seed range (circles: original; squares: replication; vermilion: a seed range both runs evaluated). All 14 shared ranges give the same score in both runs, so the replication re-reads 14 of the original's measurements. Right: the replication's mean with 95% t-interval, over all 24 blocks and over its 10 fresh blocks only (wider, as $n = 10$). All blocks are shown; none is selected.

fresh blocks alone give 2.5035, $[2.4970, 2.5100]$, not clearing the tie; with $n = 10$ this interval is wide, so it shows that the confirmation rested on reused blocks, not that the fresh edge is zero.

4.3 Case D (pass): A Different Improvement Claim That Passes the Same Check

Cases A and B leave a question open: does the check discriminate, or does fresh, disjoint re-measurement revert every claim toward the tie point? We report a later, methodologically distinct improvement claim in the same system, evaluated against the same tie point and block-seed bookkeeping, where the check holds and the edge survives.

The claim is a self-play actor-critic fine-tune of the same imitation-learned baseline used as the calibration anchor in Cases A and B: reverse-KL-anchored REINFORCE with a learned value baseline, trained as a chain of resumed segments. An initial variant without a learned baseline is confirmed *harmful* (95% interval entirely below the tie point); the actor-critic fix removes the harm and, after further training, yields a candidate certified as beating the baseline. Rather than trust that single edge, we apply the check by construction: four checkpoints from the lineage are each gated on a block-seed range drawn fresh for that checkpoint alone and verified pairwise disjoint from every other stage's range and from any selection sweep, per (a)–(b). Table 3 reports all five gates.

No two lineage rows share a block-seed range, and no gate overlaps a selection sweep, since each checkpoint is gated once rather than chosen as the best of several. Under the check that dissolved Case A's edge, this edge does not dissolve: all five intervals clear the tie point, with lower bounds $+0.0002$ to $+0.0018$ above 2.500. The upstream version history also records a later checkpoint of the same lineage clearing the tie point (placement CI $[2.5031, 2.5156]$); it is not tabulated, and omitting it does not favor the claim. Two supporting upstream checks (a mechanism decomposition and a common weak reference) are summarized in Appendix A.

All five rows share one training lineage, so the check might validate disjoint *evaluation* data without ruling out one lucky training trajectory. We therefore trained two fully independent replicas (seeds different from each other and from the lineage) and gated each identically (Table 3, bottom). They do not agree: the second clears the tie point (raw game score, zero-sum and so 0 at equal strength: $+0.2248$/game, CI $[+0.1005, +0.3491]$), the first does not ($+0.1275$/game, CI

Table 3: Disjoint-Reseed Check on a self-play actor-critic fine-tune. Each row is gated on its own fresh block-seed range, disjoint from every other row's and from any selection sweep. Cells: placement mean and 95% t-interval over block means; ✓: interval above the 2.500 tie. Bottom: two replicas trained from scratch and their pooled blocks (post-hoc pooling; neither run was selected on its outcome). Per-gate files are in the upstream training repository; the replica and pooled intervals also appear in its version history.

Gate	Blocks (games)	Placement mean	95% CI	Clears tie
One lineage (cumulative training iteration)				
Iter. 134	18 (36,000)	2.5066	$[2.5014, 2.5119]$	✓
Iter. 134, independent reseed	18 (36,000)	2.5062	$[2.5018, 2.5107]$	✓
Iter. 211	18 (36,000)	2.506	$[2.5003, 2.5118]$	✓
Iter. 311	18 (36,000)	2.508	$[2.5013, 2.5146]$	✓
Iter. 371	18 (36,000)	2.5064	$[2.5002, 2.5126]$	✓
Independent replicas (fresh initialization, no resumption)				
Replica 1	18 (36,000)	2.5036	$[2.4986, 2.5087]$	✗
Replica 2	18 (36,000)	2.5070	$[2.5021, 2.5119]$	✓
Replicas pooled	36 (72,000)	2.5053	$[2.5019, 2.5087]$	✓

$[-0.0316, +0.2866]$). Neither reverts to a negative or clearly null result as Case A's edge did, but an effect this modest will not clear significance in every replica at one gate's power even when real. The pooled estimate clears both thresholds (raw score $+0.1762$/game, CI $[0.0786, 0.2738]$), and we take it, not either replica, as the number to carry forward from Case D. The check tells a reader whether a measurement is selection-contaminated, not what sample size a modest effect needs.

So the invariant that flags Case A's edge leaves this one intact across five applications plus a pooled sixth: the check does not drive every claim to null. Ground truth is unobserved and one replica does not clear alone, so Case D shows only that the check can pass a surviving edge; one pass and two catches cannot estimate its error rates.

5 Validating the Train/Eval Symmetry Check

5.1 Case C (catch): An Asymmetric-Evaluation Gate That Froze a Ladder for 124 Generations

A real freeze, and its diagnosis. An independently built self-play ladder for an unrelated card game ("Boom") stopped promoting any candidate for 124 consecutive generations, from generation 192 to 316 – a violation of the Train/Eval Symmetry Check. During the freeze the gate evaluated candidates under an imbalanced default deck condition (one archetype favoured over the other) while training used a mirrored deck condition. The diagnostic signature was a seat-conditional win-rate split: the candidate won as one player position at approximately 0.14 and as the other at approximately 0.66 – an artifact of which deck the evaluated seat held, not of skill. Adding a mirrored evaluation mode that pools both single-deck assignments, restoring the symmetry the check requires, gives a fair self-match rate of approximately 0.53 (95% lower bound 0.41): correctly not promotable, and close to the 0.5 of matched skill. The frozen candidates' true skill is unobserved, so this is indirect evidence that the freeze was a gate artifact; it does not rule a skill plateau out.

Mechanism. If the gate does not mirror a structural, non-skill asymmetry the way training does, the asymmetry alone can cap every candidate's measured win rate below the threshold. The mechanism predicts a freeze when (a) the evaluation condition carries such an asymmetry, (b) its effect is large relative to skill-driven movement between generations, and (c) the threshold is a fixed win-rate bound it can push below; it predicts *no* freeze when the check is satisfied, consistent with symmetric self-play systems not reporting this failure (Section 2).

Absence from a second, independently built system. A natural hypothesis is that the identical mismatch recurs in the mahjong system's own gate. Direct inspection of a pinned source version does not support this: every gate script in the system's history (nine or more) replays each evaluated

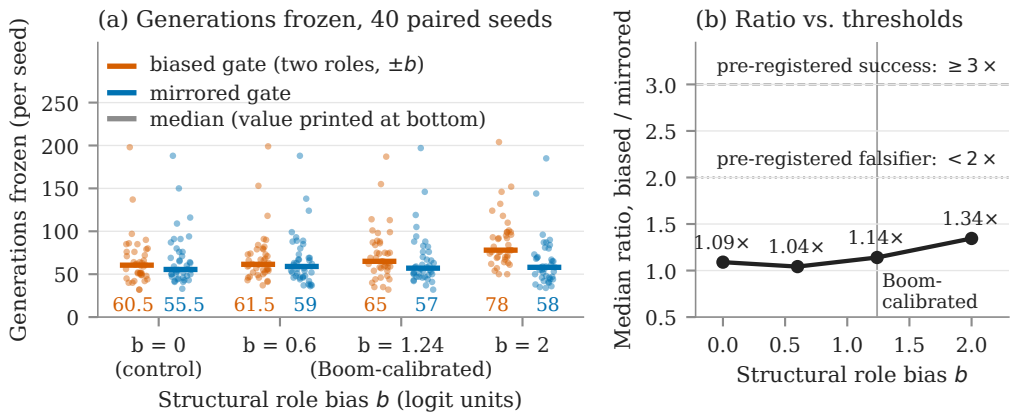

Figure 3: Pre-registered synthetic reproduction of the freeze mechanism, 40 paired seeds per bias level, no censored cells. (a) Generations frozen per seed under the biased and mirrored gates, with medians. (b) Median ratio against the pre-registered success threshold and falsifier: the ratio is not monotonic in b, and the magnitude threshold at the Boom-calibrated $b = 1.24$ is falsified. Thresholds are those of the preregistration. Per-level medians are in Appendix B.

wall once with the candidate in each of the four seats, and the training-side code matches this by explicit design – both RL training scripts carry the verbatim comment "Learner seat rotates over {0,1,2,3} to match the gate rotation," and one defines its objective as numerically identical to the gate's calibrated duplicate metric. No earlier, unfixed version was found in the repository's history or postmortem writeups. This is the negative case the mechanism predicts, so we narrow the finding per its pre-written failure action: found once, in a real deployed system (Boom); absent from the literature searched; and absent from a second system where the check was satisfied by a load-bearing design choice rather than never having been at risk.

5.2 A Pre-Registered Synthetic Reproduction of the Mechanism

To test the mechanism with a known ground-truth skill trajectory replayed through several gate conditions, we built a minimal synthetic ladder and pre-registered its prediction before running it. The bias magnitude and starting skill offset were fixed by back-solving Boom's own numbers (win rates 0.14 and 0.66 in the two player positions, giving a logit-space bias $b = 1.24$), not searched post hoc. The pre-registered prediction was that, across 40 paired seeds sharing one ground-truth trajectory, the biased gate's median generations-frozen would be at least $3\times$ the mirrored gate's at $b = 1.24$, with a falsifier triggered if the ratio fell below $2\times$.

Figure 3 reports all four bias levels. The freeze-duration ratios for $b = 0$, 0.6, 1.24 and 2.0 are $1.09\times$, $1.04\times$, $1.14\times$ and $1.34\times$: the ratio exceeds the control's ($b = 0.0$, two-role split kept) at $b = 1.24$ and 2.0, but $b = 0.6$ falls below it, so the sweep is not monotonic. The magnitude threshold is not met ($1.14\times$ at $b = 1.24$); the falsifier is triggered, and no parameter was retuned. Per paired seed, the biased gate freezes longer in 29 of 40 seeds at $b = 1.24$ (sign test $p = 0.006$; ratio 95% bootstrap interval $[1.05, 1.34]$, below the $2\times$ falsifier) and 39 of 40 at $b = 2.0$, against 20 of 40 in the control (both Holm-significant over the four levels).

The reason is mechanical (Figure 4): an additive logit bias is a translation, and pooling the two roles gives back almost the mirrored rate. At a skill difference of 0.3, the mirrored win rate is $\sigma(0.3) = 0.574$ while the biased pooled rate is $(\sigma(0.3 - 1.24) + \sigma(0.3 + 1.24))/2 = 0.552$, too small a gap to hold a Wilson-gated promotion back for anywhere near 124 generations under steady skill growth. Boom's much longer freeze is therefore not fully explained by this toy's assumptions (symmetric additive logit bias, steady linear drift); long skill plateaus, a hard per-seat win-probability ceiling, or a different sample size and gate cadence are untested alternatives, named rather than chased into a retuned run. The synthetic result shows the mechanism's direction outside the two real systems at a smaller effect size; it is not a third real-world instance and is not counted as one.

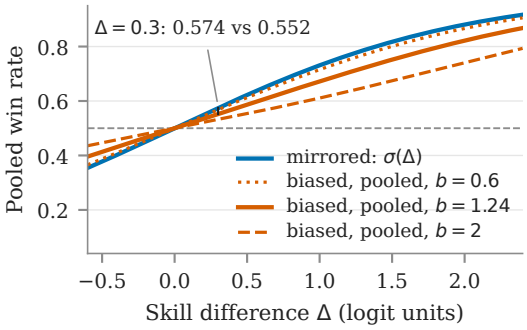

Figure 4: Why the toy's freeze stays short. Pooled win rate of a candidate with skill difference Δ under the mirrored gate, $\sigma(\Delta)$, and under the biased gate, $(\sigma(\Delta-b)+\sigma(\Delta+b))/2$, for the non-zero bias levels of Figure 3 (analytical curves, not simulation output). Just above $\Delta = 0$ the pooled curves stay close to the mirrored one, so an additive bias delays a Wilson-gated promotion only briefly under steady skill growth.

6 Discussion and Limitations

None of the four incidents (Figure 1) is visible from a gate's reported win rate or placement score alone; each needed the checks run against internal seed and condition bookkeeping. The mahjong gate, which never violated the symmetry check, cannot show that satisfying it prevents a freeze.

Limitations. One pre-specified ablation remains unresolved: the Disjoint-Reseed Check's removal ablation – replaying Case A's three-stage measurement on a candidate never selected as best-of-N – was not run: the pre-selection candidates' weights were not retained with the pinned source version and could not be recovered, and the one deployed candidate whose weights exist cannot stand in for a *set* of unselected candidates. This missing-artifact gap (not a compute gap) is reported as unresolved rather than approximated. The evidence base is deliberately narrow (two systems in depth, one synthetic check). Four incidents cannot estimate either check's false-positive or false-negative rate, and we compare PGV with no alternative safeguard (a standing held-out seed range, or a multiplicity-adjusted promotion threshold), and whether the checks generalize to other self-play deployments is open. Case C's numbers come from a postmortem record of the Boom system, not from stored gate logs, and Case D's per-gate files are in the upstream training repository. The mahjong gates, and Case D's training, run in an in-house simulator that departs from the official rules, for example by never scoring a win on another player's discard when the winner already holds a copy of that tile; each Disjoint-Reseed verdict is a comparison within this one simulator and so stands, but the placement, raw-score and Appendix A values are measured under this variant, not under the official rules.

7 Conclusion

We introduced PGV, a two-check verification harness for self-play promotion gates, and validated it on four real incidents in two deployed systems: the Disjoint-Reseed Check flags two certification failures and passes a later claim; the Train/Eval Symmetry Check would have flagged a 124-generation freeze and holds by design in the second system, and its synthetic reproduction falsified its own magnitude threshold. We report each narrowing in the same voice as the confirmations; a frozen plan that reports only favorable checks is not worth freezing.

Reproducibility Statement

Every number in this paper traces to a stored result or a pinned source version, and the code, result logs, preregistrations, and evidence plan are included in the supplementary material. Case A: the four rows of Table 2 are copied byte-for-byte from the upstream gate audit with a provenance note; the cryptographic hashes of the results behind the same-region and selection-disjoint rows match those frozen in the project's evidence plan before any analysis was run. The $z = 1.05$ comparison comes from the upstream audit and is not regenerable from our code; the other Case A comparisons are recomputed from the four table rows, with a Holm adjustment; paired-seed intervals for the synthetic reproduction are computed from its stored per-seed output. Case B is independently rederivable: one script recomputes the reused-block fraction from the two launcher scripts' own seed loops (the launcher excerpts are included and hashed), and a second matches the two runs' stored per-block scores; Figure 2 is generated from both outputs, and every figure is generated from stored

results by the same plotting code. Case D's per-gate results, mechanism decomposition, and weak-reference check are in the upstream self-play training repository (pinned versions), an anonymous repository to be released with the camera-ready; the part of its version history corroborating the quoted intervals is included. Case C's numbers come from the Boom project's postmortem record, transcribed in the evidence plan. The source excerpts behind the mahjong symmetry check are included and hashed at the pinned version. The synthetic reproduction is a CPU-only, standard-library script whose preregistration was frozen before it was run and is included unmodified alongside its output; Figure 3 is generated from that output. The unresolved ablation is documented, with what was checked and why it remains blocked, in the ablation plan.

AI USE STATEMENT

We used a generative AI assistant (Claude, Anthropic), under the authors' direction, to give feedback on experimental design, to implement analysis and plotting code, to help interpret results, to draft and edit the text, to create and edit figures and tables, to identify and summarise related literature, and to format references. We did *not* use generative AI tools to generate synthetic data reported as measurements, or for translation. We have reviewed all AI-assisted work: every reported number was checked by the authors against the logged experimental outputs, and every reference was checked against its published record. We take responsibility for the final content of this work, including all text, claims and artifacts produced with the aid of generative AI.

REFERENCES

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron Courville, and Marc G. Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

David Balduzzi, Karl Tuyls, Julien Pérolat, and Thore Graepel. Re-evaluating evaluation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019.

Ariyan Bighashdel, Yongzhao Wang, Stephen McAleer, Rahul Savani, and Frans A. Oliehoek. Policy space response oracles: A survey. In *Proceedings of the 33rd International Joint Conference on Artificial Intelligence (IJCAI), Survey Track*, pp. 7951–7961, 2024.

Neil Burch, Martin Schmid, Matej Moravcik, Dustin Morrill, and Michael Bowling. AIVAT: A new variance reduction technique for agent evaluation in imperfect information games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

E. C. Capen, R. V. Clapp, and W. M. Campbell. Competitive bidding in high-risk situations. *Journal of Petroleum Technology*, 23(6):641–653, 1971.

Gavin C. Cawley and Nicola L. C. Talbot. On over-fitting in model selection and subsequent selection bias in performance evaluation. *Journal of Machine Learning Research*, 11:2079–2107, 2010.

Cédric Colas, Olivier Sigaud, and Pierre-Yves Oudeyer. How many random seeds? Statistical power analysis in deep reinforcement learning experiments. *arXiv preprint arXiv:1806.08295*, 2018.

Wojciech M. Czarnecki, Gauthier Gidel, Brendan Tracey, Karl Tuyls, Shayegan Omidshafiei, David Balduzzi, and Max Jaderberg. Real world games look like spinning tops. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Theresa Eimer, Marius Lindauer, and Roberta Raileanu. Hyperparameters in reinforcement learning and how to tune them. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 9104–9149, 2023.

Andrew Gelman and Eric Loken. The garden of forking paths: Why multiple comparisons can be a problem, even when there is no "fishing expedition" or "p-hacking" and the research hypothesis was posited ahead of time, 2013. Unpublished manuscript, Department of Statistics, Columbia University.

Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

Max Jaderberg, Wojciech M. Czarnecki, Iain Dunning, et al. Human-level performance in 3D multiplayer games with population-based reinforcement learning. *Science*, 364(6443):859–865, 2019.

Scott M. Jordan, Yash Chandak, Daniel Cohen, Mengxue Zhang, and Philip S. Thomas. Evaluating the performance of reinforcement learning algorithms. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020.

Sayash Kapoor and Arvind Narayanan. Leakage and the reproducibility crisis in machine-learning-based science. *Patterns*, 4(9):100804, 2023.

Marc Lanctot, Vinicius Zambaldi, Audrunas Gruslys, Angeliki Lazaridou, Karl Tuyls, Julien Pérolat, David Silver, and Thore Graepel. A unified game-theoretic approach to multiagent reinforcement learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

Junjie Li, Sotetsu Koyamada, Qiwei Ye, Guoqing Liu, Chao Wang, Ruihan Yang, Li Zhao, Tao Qin, Tie-Yan Liu, and Hsiao-Wuen Hon. Suphx: Mastering mahjong with deep reinforcement learning. *arXiv preprint arXiv:2003.13590*, 2020.

lightvector. KataGo: An open-source Go self-play engine with network gating, 2024. `https://github.com/lightvector/KataGo`.

Xinkun Nie, Xiaoying Tian, Jonathan Taylor, and James Zou. Why adaptively collected data have negative bias and how to correct for it. In *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2018.

Andrew Patterson, Samuel Neumann, Martha White, and Adam White. Empirical design in reinforcement learning. *Journal of Machine Learning Research*, 25(318):1–63, 2024.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. Mastering the game of Go without human knowledge. *Nature*, 550:354–359, 2017.

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

James E. Smith and Robert L. Winkler. The optimizer's curse: Skepticism and postdecision surprise in decision analysis. *Management Science*, 52(3):311–322, 2006.

Karl Tuyls, Julien Pérolat, Marc Lanctot, Joel Z. Leibo, and Thore Graepel. A generalised method for empirical game theoretic analysis. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, 2018.

Sudhir Varma and Richard Simon. Bias in error estimation when using cross-validation for model selection. *BMC Bioinformatics*, 7:91, 2006.

Oriol Vinyals, Igor Babuschkin, Wojciech M. Czarnecki, et al. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575:350–354, 2019.

Yongzhao Wang, Qiurui Ma, and Michael P. Wellman. Evaluating strategy exploration in empirical game-theoretic analysis. *arXiv preprint arXiv:2105.10423*, 2021.

David J. Wu. Accelerating self-play learning in Go. *arXiv preprint arXiv:1902.10565*, 2019.

Yang Xu, Jiefu Zhang, Haixiang Sun, Zihan Zhou, Tianyu Cao, and Vaneet Aggarwal. Towards reliable LLM evaluation: Correcting the winner's curse in adaptive benchmarking. *arXiv preprint arXiv:2605.05973*, 2026.

A Case D: Supporting Upstream Checks

A companion mechanism analysis of paired games against the same baseline finds win rate and deal-in rate statistically flat while mean fan per win is higher and the self-draw share of wins is *lower*, consistent with a specific mechanism (building higher-value hands). All four quantities are as scored by the in-house simulator, which never scores a win on a discard of a tile the winner already holds (Section 6), so they describe play under that rule variant. A further check gates both the baseline and the Case D claim against a common weak reference (a behavior-cloned imitation of a much stronger competition entry); both clear it by a large margin, and the claim's margin exceeds the baseline's ($+0.021$, two-sample $t \approx 3.6$) – a structurally independent confirmation with a different opponent and seed range. Both analyses are in the upstream training repository (Reproducibility Statement).

B Synthetic Reproduction: Per-Level Medians

Table 4 tabulates the medians plotted in Figure 3, read from the reproduction's stored output (40 paired seeds per level; zero censored cells out of 320).

Table 4: Median generations frozen under the biased and mirrored gates, and their ratio, per bias level. Pre-registered success threshold: ratio $\geq 3\times$ at $b = 1.24$; falsifier: ratio $< 2\times$.

Bias b	Median frozen (biased)	Median frozen (mirrored)	Ratio
0.0 (control)	60.5	55.5	$1.09\times$
0.6	61.5	59.0	$1.04\times$
1.24 (Boom-calibrated)	65.0	57.0	$1.14\times$
2.0	78.0	58.0	$1.34\times$