

Does small-blind limping survive contact with counterfactual regret minimization? A heads-up No-Limit Hold'em experiment

By <https://Balancedrange.com>: Written with AI assistance (Claude); methodology, code, and results below are real and reproducible from the artifacts listed at the end, although our results are not peer reviewed — more in caveats section (see below).

Summary

We ran a CFR-driven poker agent ("Vegas Boss") through 6,000+ hands of self-play warmup against a fixed, non-learning heuristic opponent ("recommendation_bot"), then measured 1,000 further matches (~17,300 hands) of pure best-response play. Across every effective-stack bucket we measured (from under 10 big blinds to 80+), the CFR agent open-limped the small blind 30-52% of the time, versus 7-16% for the static heuristic control — a large, statistically overwhelming gap (illustrative 95% CIs at the best-sampled bucket do not approach overlapping: $42.1\% \pm 1.5\%$ vs. $14.4\% \pm 1.1\%$, $n > 3,600$ each side). This holds even at stack depths where small-blind limping is conventionally treated as close to strictly dominated by a shove-or-fold approach.

We think this is a useful, if narrow, contribution to a long-standing debate in no limit hold'em theory: can limping strategies contribute to mixed-strategy equilibria in heads-up when playing from the SB, or is it strictly dominated by a raise or fold strategy? While our results do not definitively settle that question, they show that a best response computed against a specific, theory-grounded opponent — not a straw-man — puts real, repeated weight on limping rather than treating it as a rounding error.

Background

Heads-up No-Limit Hold'em strategy discussions have long included a specific disagreement: should the small blind's opening range ever include limping (calling the big blind rather than folding or raising), or is the raise-or-fold approach always at least as good? The received wisdom in most modern strategy material leans toward raise-or-fold, especially at shallower stacks, but the question of whether limping survives as part of a genuine mixed equilibrium — appearing with some nonzero frequency in a solved or near-solved strategy — has stayed open

in less formal poker discussion, distinct from the rigorously solved (but much simpler) heads-up *limit* game.

Heads-up Limit Hold'em itself was essentially weakly solved by the University of Alberta's Computer Poker Research Group in 2015 (Bowling, Burch, Johanson & Tammelin, *Science*), using the CFR+ algorithm to compute a strategy within a provably tiny margin of Nash equilibrium. No-Limit Hold'em — with its continuous bet-sizing action space — has not been solved in that same formal sense; practical systems instead approximate strong play through abstraction, self-play, and regret minimization. Counterfactual regret minimization (CFR) itself was introduced by Zinkevich, Johanson, Bowling & Piccione (NeurIPS 2007), with sample-based variants (Monte Carlo CFR) developed by Lanctot, Waugh, Zinkevich & Bowling (NeurIPS 2009) to make the approach tractable without enumerating an entire game tree on every iteration.

Our engine's heuristic opponent (`recommendation_bot`, in this experiment) is not a straw-man built to lose — its pre-flop range tables and much of its post-flop decision logic originate from Michael Acevedo's *Modern Poker Theory* (D&B Poker, 2019), a widely-used published reference for GTO-informed tournament and cash-game strategy. Over many months of separate development, those heuristics were further refined by an iterative process: run the heuristic strategy, let a CFR-trained agent play many hands against it and find profitable counter-adjustments, revise the heuristic to close the exploit, repeat. This is loosely in the spirit of fictitious-play-style convergence (each revision is a best response to the previous one), though we want to be precise that it is not a formal proof of convergence to equilibrium — see caveats.

It's worth being precise about what "originate from" means here, since the book itself leaves a fair amount undetermined. Acevedo's charts communicate mixed-strategies (e.g. "raise/fold/call") through shaded regions rather than printed percentages — a reader has to visually estimate a frequency from the shaded area. More importantly, even a correctly-estimated frequency only specifies *how often* a hand class should take an action, not *which specific card combinations* should fall on which side of that split — the precise individual-combinations that compose those frequencies are left up to the reader. Converting that qualitative guidance into an exact, combo-by-combo implementation was a separate, non-trivial design task on our part: an internally-developed procedure that converts an estimated mixing frequency into a concrete selection of individual combos, with an aim toward keeping card-removal effects (which specific suits get excluded, and how that affects blocker balance against the rest of the range) relevant rather than arbitrary. Although the precise methodology used in the combination selection procedure is outside the scope of this paper, we believe it produces ranges that in practice approximate the canonical range charts produced by commercial solver tools.

The CFR agent's own approach — computing a strategy via self-play regret minimization, then *additionally* adapting to exploit a specific opponent rather than stopping at a pure equilibrium strategy — mirrors the framing in Yi & Yi's recent work, "Beyond Game Theory Optimal: Profit-Maximizing Poker Agents for No-Limit Hold'em" (arXiv:2509.23747), which explicitly targets profit-maximizing deviation from GTO once a strong baseline is established.

Method

Agents. `recommendation_bot` acts by querying our engine's human-facing hint system at every decision and resolving its output into a concrete action. Many of that system's outputs are literal mixed-frequency recommendations (e.g. "Check 72%, Raise half-pot 28%"); rather than always taking the modal action, we drew a weighted random action from the stated percentages at every single decision, so the bot's realized strategy matches the hint engine's *intended* mixed strategy rather than collapsing it to a pure one. `recommendation_bot`'s strategy is entirely fixed for the duration of the experiment — it does not learn or adapt.

`Vegas Boss` is driven by our engine's `DeepStackAI` module — named in homage to the landmark DeepStack system (Moravčík et al., 2017), which combined recursive reasoning, game-tree decomposition, and deep-learning-based value estimation to defeat professional players at heads-up No-Limit Hold'em. Our module is considerably simpler: a tabular CFR implementation (per-information-set regret tables, no neural value function or continual resolving) that (a) applies a set of hand-coded heuristic overrides for the first 6,000 hands of its training history ("warmup"), gated off entirely once that threshold is crossed, and (b) thereafter selects actions via CFR's average-strategy convergence rather than the noisier current-iterate strategy. All results reported below are from *after* warmup — pure CFR best response, no heuristic overrides active.

Runs. Session 1 played matches in a deterministic mode (always the modal recommended action, for the heuristic side) until the CFR agent's persisted hand count crossed 6,000 — 550 matches, won 367–183 (66.7%) by `recommendation_bot`. Session 2 then played 1,000 further matches in the frequency-respecting mode described above, entirely with the CFR agent past warmup, won 620–380 (62.0%) by `recommendation_bot`; this is the sample all limp/raise/fold numbers below come from.

Stakes. Heads-up, flat blinds throughout (100/200, 50 ante — no escalation), 16,000-chip starting stacks (80 big blinds). We deliberately disabled our engine's normal blind-escalation feature (present to keep matches short for human players) for this run, since escalating blinds would have confounded effective-stack bucket with how far into a match a given hand fell.

Measurement. For every hand, we identified the small blind's first pre-flop decision (fold, call/limp, or raise/all-in) directly from the match's action log, and recorded the effective stack (the shorter of the two stacks, in big blinds) at that decision. This gives one classified event per hand for each of 1,000 matches (~17,300 hands, 19,333 usable SB-open decisions after excluding 12 edge cases where a short-stacked small blind's forced blind post consumed their entire stack, leaving no actual decision to classify).

Results

Effective stack	recommendation_bot limp% (n)	Vegas Boss limp% (n)
\< 10bb	7.2% (335)	29.6% (362)
10–20bb	7.7% (639)	40.2% (669)
20–40bb	8.1% (1,787)	35.9% (1,856)
40–60bb	12.2% (2,399)	40.1% (2,357)
60–80bb	14.4% (3,692)	42.1% (4,152)
80bb+	16.0% (1,054)	51.6% (31)

The static, theory-derived control shows a small-blind limp rate that rises roughly monotonically with stack depth — consistent with the standard implied-odds argument that limping becomes relatively more attractive as effective stacks deepen. The CFR best-response agent limps far more often at every depth, including the shallowest buckets where raise-or-fold is conventionally treated as close to strictly correct. The gap does not shrink as stacks get shallow; if anything it's largest in the sub-20bb range, which is the opposite of what a "limping is a deep-stack luxury" hypothesis would predict.

We caution against over-reading the 80+bb row for Vegas Boss specifically — a genuine effective stack of exactly 80bb or more only occurs in the very first hands of a match before variance moves the stacks apart, so that bucket's sample (n=31) is thin relative to the others.

As an internal consistency check: an earlier, informal internal test run (weeks prior, different engine build, escalating rather than flat blinds) never measured limping directly, but its VPIP-PFR gap — the difference between how often Vegas Boss voluntarily entered a pot and how often she raised when doing so, a standard indirect proxy for limping in poker analytics — showed the identical pattern: roughly a 7-point gap pre-warmup, widening to roughly 33-36 points post-warmup. This run's direct measurement and that run's indirect proxy agree closely despite small differences in engine version between the two.

What we think this does and does not show

What it shows: a CFR agent computing a best response against a specific non-adaptive opponent converges to a strategy that limps the small blind frequently and consistently across stack depths — not as noise, not as a rare exploratory action, but as a stable, large-magnitude part of its post-warmup strategy. That is a real, reproducible-in-shape (if not in exact numbers — see caveats) empirical fact about this specific matchup.

What it does not show: that limping is part of the Nash equilibrium strategy for heads-up No-Limit Hold'em in general. A best response to one fixed opponent is not the same claim as an equilibrium strategy against all possible opponents, and our CFR agent trained against exactly one static counterparty rather than against a population of adversaries or an approximation of a fully unrestricted range of opponents. It's entirely possible — likely, even — that some of what we are measuring is Vegas Boss finding and exploiting a specific, real weakness in `recommendation_bot`'s implementation of a raise/limp/fold strategy, rather than rediscovering some universal game-theoretic truth about limping. In order to derive more general conclusions regarding the relevance of limping as part of small-blind mixed-strategy equilibria, replicating these results against multiple and differentiated fixed opponents, as well as against a self-play-only baseline with no fixed opponent at all, is required.

Caveats

- **Single, unseeded run.** Our codebase has no RNG seeding anywhere (the resolver's action draws and the engine's Monte Carlo equity estimates both use unseeded randomness). We have not repeated this experiment with different seeds to check how much the exact percentages above would vary run-to-run — though the sample sizes involved (hundreds to low thousands of events per bucket) make it very unlikely that the qualitative gap disappears, we cannot yet quantify run-to-run variance directly.
- **Non-equilibrium control.** `recommendation_bot` is a fixed heuristic, not a solved-game baseline.
- **Non-equilibrium treatment.** Our CFR implementation's self-play convergence against a single fixed opponent is not a proof of Nash-equilibrium convergence for the full game; it's a best response to one specific counterparty.
- **Thin high-stack sample.** The 80+bb bucket (Vegas Boss side, $n=31$) should be treated as suggestive, not confirmatory. The lopsided split between the two bots here (`recommendation_bot` $n=1,054$ vs. Vegas Boss $n=31$) is not a skill or win-rate effect —

it's mechanical: our engine assigns the small blind to seat 0 (recommendation_bot) by default at the start of every fresh match, so essentially all of hand 1 across all 1,000 matches has recommendation_bot as the small blind at exactly 80bb. Vegas Boss's 31 events here come only from the rarer case of a hand later in a match happening to end in an exact chop that returns both stacks to precisely their starting value.

- **Blind-escalation-free is a deliberate departure from normal play** for this app (which escalates blinds to keep human games short) — a design choice made specifically to make the stack-depth bucketing meaningful, not a claim about what "real" play in this app looks like.

Reproducibility

Included with this paper is the full methodology described above, the raw per-hand event log (19,333 records, [sb open events test12.json](#)), the aggregate results ([sb limp stackdepth report test12.json](#)), and the persisted CFR regret state at the end of the run ([cfr learned state vegas boss test12 sb limp stackdepth.json](#)). This run is not reproducible bit-for-bit in any case (see "single, unseeded run" above); we would welcome anyone running an analogous experiment using a specific fixed opponent of their own.

References

- Acevedo, M. (2019). *Modern Poker Theory*. D&B Poker.
- Bowling, M., Burch, N., Johanson, M., & Tammelin, O. (2015). Heads-up limit hold'em poker is solved. *Science*, 347(6218), 145–149. <https://doi.org/10.1126/science.1259433>
- Zinkevich, M., Johanson, M., Bowling, M., & Piccione, C. (2007). Regret minimization in games with incomplete information. *Advances in Neural Information Processing Systems (NeurIPS)* 20.
- Lanctot, M., Waugh, K., Zinkevich, M., & Bowling, M. (2009). Monte Carlo sampling for regret minimization in extensive games. *Advances in Neural Information Processing Systems (NeurIPS)* 22.
- Moravčík, M., Schmid, M., Burch, N., Lisý, V., Morrill, D., Bard, N., Davis, T., Waugh, K., Johanson, M., & Bowling, M. (2017). DeepStack: Expert-level artificial intelligence in heads-up no-limit poker. *Science*, 356(6337), 508–513. <https://doi.org/10.1126/science.aam6960>

- Yi, S., & Yi, S. (2025). Beyond Game Theory Optimal: Profit-Maximizing Poker Agents for No-Limit Hold'em. arXiv:2509.23747.