

When More Self-Play Makes an Agent Worse: Empirical Instability in MuZero-Style Training

Clark Whitehead
Independent Researcher
clarkwhitehead.com

October 2026

Working manuscript — preliminary, not peer reviewed

Abstract

Self-play reinforcement learning improves policies by training on trajectories generated by their own evolving behavior. We study a recurrent failure mode in a MuZero-style agent trained in a custom competitive combat environment: additional self-play can reduce head-to-head performance even while the supervised training objective continues to improve. Across multiple training runs and model widths, competitive performance commonly peaks after approximately 24,000–32,000 self-play games and subsequently declines. Lowering the learning rate, increasing model capacity, and extending training do not consistently prevent degradation. The instability depends on training configuration: in one comparison, two parameter updates per game with three replay blocks yielded 21% wins against a fixed reference at 32,000 games, whereas larger replay configurations produced competitive agents. We also observe a preliminary stabilization result from continuing one strong checkpoint against a pool of historical opponents. These findings motivate the hypothesis that changing self-play data distributions and insufficient retention of useful prior behaviors contribute to performance collapse. The available evidence does not establish a causal mechanism or demonstrate catastrophic forgetting in the formal multi-task sense. We describe controlled experiments needed to distinguish distribution drift, optimization effects, evaluation artifacts, and representational degradation.

1 Introduction

MuZero-style agents combine a learned latent dynamics model, policy and value estimates, and tree-search planning [1]. In self-play, the training distribution changes as the agent changes: unlike supervised learning on a fixed dataset, the agent’s latest policy helps determine what it will learn next. This feedback permits rapid improvement but can also make training difficult to interpret.

We report experiments with a MuZero-style implementation in a custom competitive combat environment. The central observation is simple: *training longer can make the agent less competitive*. The effect is not adequately summarized by a single unsuccessful seed. Multiple runs peak at intermediate training checkpoints, then lose to fixed reference opponents despite continued decreases in training loss. We study this phenomenon as a stability problem rather than claim a new training algorithm.

Our contributions are (i) an empirical characterization of late-training performance decline, (ii) evidence that training-loss improvements, increased width, and learning-rate reduction do not by themselves predict stable competitive performance, and (iii) preliminary tests suggesting that historical-opponent mixtures deserve further controlled investigation. All observations are from the author’s experiment notes; raw logs, implementation details, and independent replications are required before the claims can be considered established.

2 Background and Research Question

Let π_{θ_t} denote an agent’s policy at training stage t , and let $\mathcal{D}_t$ denote the replay distribution induced by recent self-play. Training approximately optimizes an objective $\mathcal{L}(\theta; \mathcal{D}_t)$ consisting of model, value, and policy prediction terms. Because $\mathcal{D}_t$ depends on evolving behavior, a decline in $\mathcal{L}$ does not necessarily imply improved competitive performance against a fixed distribution of opponents.

We therefore distinguish *training fit*, evaluated on the agent’s replay data, from *external strength*, evaluated against frozen checkpoints and opponent mixtures. Our primary question is: *what causes external strength to degrade while fit to self-generated experience improves?* The working hypothesis is that self-play distribution shift, interacting with finite replay and optimization, can favor locally self-consistent behaviors that are weak against earlier or alternative strategies. This remains an interpretation to test, not a demonstrated explanation.

3 Experimental Setup

Environment. Experiments use a custom competitive combat game with episode outcomes and an action representation combining movement with discretized aiming. Early experiments revealed a significant representation problem: fire directions specified in global coordinates were hard to learn. Re-expressing the 16 aiming bins relative to the nearest living opponent substantially improved play. Results below therefore focus on the later opponent-relative-aim setup. This is *not* standard chess or an Atari benchmark; a separate chess pilot was too preliminary to support conclusions about training collapse.

Agent and training. The strongest reported configuration uses a 448-unit hidden representation, 64-step observation history, and 32 search simulations at evaluation. Prominent runs use two updates per self-play game and six replay blocks. Experiments vary hidden width (448, 640, 896, and 1280), updates per game, replay blocks, learning-rate schedules, opponent mixtures, and training duration. The exact network modules, replay sampling algorithm, loss weights, optimizer hyperparameters beyond reported learning rates, and complete environment specification must be documented from source code in a submission-ready version.

Evaluation. Checkpoints are compared in head-to-head bouts against frozen reference agents, with separate development and held-out seeds where available. Results can be non-transitive: beating one strong opponent does not imply superiority to another. Consequently, a robust comparison requires several anchor opponents, paired seeds, and uncertainty estimates. The current summary contains selected outcomes and pooled bout counts, but not a complete table of all seeds and confidence intervals.

4 Results

4.1 Late-training decline despite decreasing loss

Many 448- and 896-width training runs attained their strongest reported play near 24,000–32,000 games and weakened with additional training. During a reported collapse, training loss fell from approximately 2.65 to 2.17 while mean self-play game length shortened from about 37 to 30 steps. These are correlations; game length alone does not diagnose strategic weakness. Reducing the Adam learning rate after peak performance typically delayed collapse by at most one checkpoint or failed to prevent it.

The strongest selected run, designated v12, was trained from scratch with a stepped learning-rate schedule (initially 3×10^{-4} , then 2×10^{-4} , then 10^{-4}). At 32,000 games it won 437 of 632 decided games against v11 across reported bouts (69%), and 288 of 499 against v10 (58%). However, further continuation to 40,000–48,000 games at lower rates reduced its performance against v10. Importantly, this checkpoint was a

selected high-performing seed: four other seeds under the same schedule did not reproduce the gain. The schedule cannot be credited as a general improvement.

Table 1: Selected observations from experiment notes. Comparisons are not a single controlled factorial study. Percentages are against fixed reference opponents unless otherwise stated.

Condition	Reported observation
448-width, strong runs	Peak strength near 24K–32K games; later decline in several chains
896-width extensions	No reported seed surpassed top 448-width models; all reported 40K continuations deteriorated
2 updates/game, 3 replay blocks	21% vs v10 at 32K (one seed)
2 updates/game, 6 replay blocks	Multiple seeds competitive with leading checkpoints
4 updates/game, 448-width	23% vs v10 at 32K (one seed)
Historical-opponent pool from v12 at 32K	52% vs v12 over 503 games at 40K (one continuation)

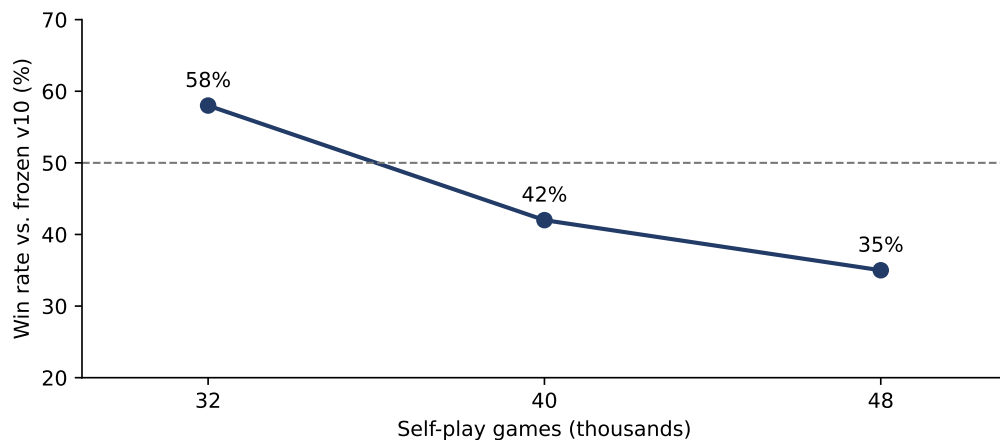


Figure 1: Selected v12 checkpoint results against frozen v10, as reported in the experiment notes. Continued training corresponds to lower win rates in these bouts, but the points involve different evaluations and do not constitute a controlled learning curve. The v12 seed was selected for high performance; these results should not be generalized across seeds.

4.2 Replay and model scale are not simple remedies

At two parameter updates per game, reducing replay from six to three blocks coincided with severe deterioration: a three-block run ended at approximately 21% wins against v10 after 32,000 games, whereas several six-block runs reached the leading group. Increasing the update count to four per game without a corresponding increase in replay also performed poorly in the tested 448-width run (23% at 32,000 games). These comparisons suggest an interaction between update intensity and replay diversity, but seed replication and matched data budgets are required for causal claims.

Increasing network width did not reliably help. Multiple 896-width seeds matched leading performance at intermediate checkpoints, but none surpassed the top 448-width reference in the reported comparisons, and reported 40K extensions deteriorated. The 640- and 1280-width results also failed to establish scale as a remedy. These observations are specific to the current environment and training budgets.

4.3 Historical opponents: an encouraging but preliminary intervention

One 32,000-game v12 checkpoint was continued with a pool of frozen partners at a learning rate of 5×10^{-5} . The resulting 40,000-game checkpoint went 263–240 against its 32,000-game parent (52% across 503 decided games) and scored 67% against v10. A corresponding ordinary 40,000-game continuation had substantially weaker reported performance against v10. Earlier use of opponent pools did *not* yield a consistent benefit across seeds. Thus the result motivates a timing-dependent hypothesis, not a claim that opponent pools solve the problem.

4.4 Evaluation can conceal instability

Results show strong seed effects and non-transitive matchups. A model with 58% wins against v10 and 54.5% against v11 nevertheless won only 44% against another strong averaged checkpoint. The practice of selecting one winning seed or testing against only the current champion can therefore exaggerate progress. Future results should report all seeds and a common matrix of opponents, not just selected promotions.

5 Mechanistic Hypotheses and Decisive Tests

We propose four non-exclusive explanations.

H1: Self-play distribution drift. The agent may increasingly encounter a narrow range of states and actions, eroding robustness against historical behaviors. Test by logging state visitation statistics, policy divergence on a fixed probe set, and performance against a frozen opponent league at each checkpoint.

H2: Replay-induced forgetting. New trajectories may displace strategically important earlier experiences. Test fixed-capacity FIFO replay against reservoir sampling, age-stratified replay, and stratified mixtures of early and recent games, with matched training updates and game budgets.

H3: Model-head degradation. Planning may fail because policy priors, value estimates, or learned dynamics degrade on strategically important states. Test policy-only, value-only, and dynamics-related diagnostics on the same fixed held-out trajectories across checkpoints. Existing world-model checks found value prediction AUC about 0.71–0.79 and reward AUC as high as 0.93–0.97, but their ranking correlations with Elo were at most approximately 0.2; better head prediction metrics are therefore not sufficient evidence of stronger play.

H4: Opponent mismatch rather than global forgetting. A later checkpoint might adapt to its present opponent distribution without losing general competence. Test against a diverse frozen league, newly trained agents, exploiter policies, and earlier checkpoints. Measure worst-case as well as mean performance. Genuine catastrophic forgetting across tasks cannot be concluded from current single-environment bouts.

6 Limitations and Reproducibility

This draft rests on an experiment summary rather than audited source code and raw match logs. It includes multiple exploratory comparisons, selected checkpoints, and heterogeneous evaluation sample sizes. Potential sources of bias include checkpoint selection, repeated testing against familiar opponents, correlated game seeds, and unreported implementation choices. The notes explicitly record earlier replay and evaluation bugs, emphasizing the need for validation. Confidence intervals, per-seed trajectories, exact baselines, a public

environment specification, compute costs, and replication in a second environment are outstanding. We do not assert that the observed instability is unique to MuZero or that it proves catastrophic forgetting.

7 Conclusion

In a custom MuZero-style self-play system, more training frequently coincides with weaker head-to-head play, even as the training objective improves. Model scaling and learning-rate decay do not reliably reverse the effect; replay configuration and opponent diversity appear important but remain incompletely controlled. The strongest immediate research direction is a mechanistic, multi-seed investigation of how an evolving self-play distribution interacts with experience replay and frozen-opponent competence. Establishing such a mechanism and a reproducible mitigation would provide a more substantial contribution than reporting a single successful checkpoint.

Acknowledgment of status. This is an author-facing working manuscript prepared from experiment notes dated October 2026. All findings should be checked against training code, checkpoints, and original logs before public submission.

References

- [1] J. Schrittwieser et al. Mastering Atari, Go, chess and shogi by planning with a learned model. *Nature*, 588:604–609, 2020.
- [2] Z. Liu, G. Fu, C. Du, W. S. Lee, and M. Lin. Continual reinforcement learning by planning with online world models. *Proceedings of ICML*, PMLR 267:38397–38423, 2025.
- [3] R. Zhang et al. A survey on self-play methods in reinforcement learning. arXiv:2408.01072, 2024.