Movement Is All You Need: How 29 Systematic Experiments Falsified RAG-Based FPS Game Agent Optimization

Sang Yi Affiliation to be added>Affiliation to be added

[email to be added]

Claude Code
Anthropic
AI Co-Investigator

Abstract

We apply Design of Experiments (DOE) methodology — factorial designs, ANOVA with residual diagnostics, and effect size quantification — to systematically optimize FPS game agents in VizDoom's `defend_the_line` scenario. Across 29 experiments totaling 5,010 episodes, we test the hypothesis that a multi-level RAG architecture improves agent performance through document quality and scoring accuracy. Three independent experiments falsify this hypothesis: L2 retrieval conditions indistinguishable from L0 baselines ($p = 0.929$, DOE-022), decision mode producing no effect on kills ($p = 0.393$, DOE-024), and RAG selection indistinguishable from fixed strategies ($p = 0.935$, DOE-026). Instead, we discover that lateral movement is the sole performance determinant ($F(1, 116) = 58.40$, $p < 0.001$, $\eta_p^2 = 0.332$, Cohen's $d = 1.408$), with all tactical variations producing null effects. We identify rate-time compensation (kills $\approx k_r \times t_{\text{surv}}$, within movement class) as a fundamental environment constraint explaining why tactical optimization is futile.

1 Introduction

Autonomous game agents have become a prominent AI testbed, with FPS environments presenting challenges in real-time decision-making [1]. The dominant paradigm relies on deep RL [15, 16], but practical optimization involves ad-hoc hyperparameter tuning that is statistically inefficient, produces publication bias, and conflates correlation with causation [12]. Design of Experiments (DOE) [10] provides a principled alternative that decomposes complex systems into orthogonal factors, yet remains underutilized in ML [13].

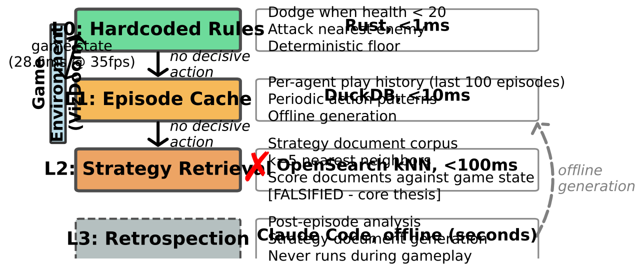
We apply DOE to FPS game agents in VizDoom [1], seeking to understand the causal structure of performance. Our framework follows four phases: (0) OFAT screening, (1) factorial analysis, (2) hypothesis testing, and (3) constraint discovery.

We implement a four-level hierarchical agent: **L0** (hardcoded reflexes), **L1** (DuckDB episode cache), **L2** (OpenSearch kNN strategy retrieval), and **L3** (offline LLM retrospection). No real-time LLM inference occurs during gameplay. The core thesis: Agent Skill $= f(\text{DocQuality}, \text{ScoringAccuracy})$.

Through 29 experiments (5,010 episodes), we discovered this thesis is false. Three independent tests falsified L2 retrieval ($\eta_p^2 < 0.01$ in all). Only lateral movement produced a significant effect ($d = 1.408, p < 0.001$). Rate-time compensation (kills $\approx k_r \times t_{\text{surv}}$) explains why tactical optimization is futile. Game difficulty explains 72% of variance ($\eta_p^2 = 0.720$) and movement 33%, while all agent architecture parameters collectively explain $< 5\%$.

Contributions: (1) First application of formal DOE to FPS game AI; (2) 29 DOEs with 5,010 episodes and 83 findings with complete audit trails; (3) identification of rate-time compensation as a

Figure 1: Four-Level Agent Architecture with Latency Profiles



Agent decision system with four levels. L0 (hardcoded rules) provides a deterministic floor with <1ms latency. L1 (episode cache) stores periodic action patterns generated offline. L2 (strategy retrieval) uses kNN search over a corpus of strategy documents. L3 (retrospection) runs offline after episodes to generate new strategies. The core thesis that L2 provides useful guidance was falsified by DOE-001~020 experiments.

Figure 1: Four-level hierarchical agent architecture. L0 provides a deterministic behavioral floor. L1 stores periodic action patterns. L2 performs kNN vector search over strategy documents generated by L3 retrospection. L3 operates offline. The red marker on L2 indicates the falsification result: RAG retrieval produces no benefit ($p > 0.39$ across three tests).

fundamental constraint; (4) rigorous falsification of RAG-based retrieval through three independent null results.

2 Related Work

LLM-based game agents. Recent work explored LLMs in game agents, including GPT-4 on VizDoom with prohibitive latency [2], episodic self-reflection [3], lifelong skill accumulation [4], and hierarchical LLM-RL separation [5, 6]. Our work differs by using the LLM exclusively offline and rigorously testing—rather than assuming—whether retrieval helps.

VizDoom and deep RL. VizDoom [1] is a standard FPS benchmark. Deep RL achieved strong performance through game-feature prediction [7], predictive learning [8], and population-based training reaching human-level play [9]. These approaches focus on end-to-end optimization without decomposing which factors matter; our DOE fills this gap.

DOE in ML. DOE remains underutilized despite its potential [13, 14]. Bayesian optimization [11] and random search [12] find optima without explaining *why*. Our work demonstrates DOE reveals fundamental constraints that ad-hoc tuning would miss.

3 Methodology

3.1 Agent Architecture

Our agent employs a four-level hierarchy without real-time LLM inference (Figure 1). **L0** (Rust) encodes reflexes: dodge at low health, attack nearest enemy. **L1** retrieves cached behavioral patterns from DuckDB, notably periodic sequences like `burst_3`. **L2** performs kNN vector search over strategy documents in OpenSearch, scoring retrieved tactics against the current game state. **L3** uses Claude Code between episodes to analyze outcomes and generate strategy documents—never during gameplay. A Python binding interfaces with VizDoom's API at 35 fps.

3.2 Experimental Framework

We adopt DOE methodology from quality engineering. Our program follows four phases (Figure 2): Phase 0 (OFAT screening, DOE-001–010), Phase 1 (factorial analysis, DOE-011–021), Phase 2 (hypothesis testing, DOE-022/024/026), and Phase 3 (constraint discovery, DOE-027–029).

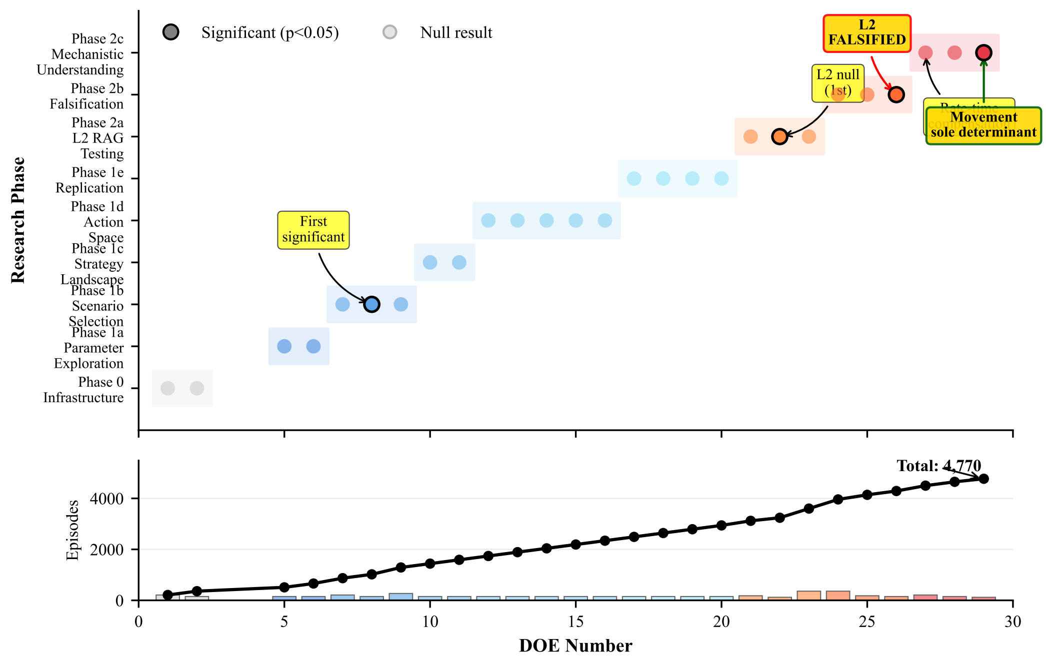


Figure 2: DOE progression across four research phases (5,010 total episodes). Filled markers: significant ($p < 0.05$); open: null. Key milestones: DOE-008 (first significant result), DOE-022/024/026 (RAG falsification), DOE-027 (rate-time compensation), DOE-029 (movement as sole determinant, $d = 1.408$).

Every experiment uses fixed seed protocols (minimum 30 per condition, power $\geq$ 0.80), full ANOVA with residual diagnostics (Anderson-Darling normality, Levene equal variance, run-order independence), effect sizes (partial η^2, Cohen's d) with 95% CIs, trust classification (HIGH/MEDIUM/LOW/UNTRUSTED), and a complete audit trail from hypothesis through finding adoption (see Appendix E for full statistical methods).

3.3 VizDoom Environment

We evaluate on `defend_the_line`, where the agent faces continuously spawning single-hit enemies in a corridor and can turn, strafe, and fire. We test 3-action (turn left/right, attack) and 5-action (adding strafe left/right) spaces. Response variables: kills, survival time, and kill rate. Scenario selection was validated empirically: `defend_the_center` produced near-zero variance ($p = 0.183$), while `defend_the_line` provided adequate discrimination ($F(4, 145) = 5.256$, $p < 0.001$, $\eta^2 = 0.127$).

4 Results

We present results across four phases spanning 29 DOEs and 5,010 episodes (Appendix A: complete DOE table; Appendix B: top 20 findings).

4.1 Infrastructure Validation and Strategy Exploration (DOE-001–020)

DOE-001–004 revealed a data pipeline error (AMMO2 mapped as KILLCOUNT), invalidating early experiments but providing the first indication that L1/L2 contribute no behavioral differentiation. DOE-007–008 established `defend_the_line` as the standard scenario (3× better effect size vs. `defend_the_center`). DOE-009 confirmed agent parameters (memory, strength weights) have no effect ($p = 0.736$, $p = 0.109$). DOE-010–020 mapped the strategy landscape: L0-only is universally worst ($d = 0.65$–1.48); random matches structured strategies in 3-action space ($p = 0.741$); strafing

Table 1: L2 RAG falsification evidence. All tests show negligible effect sizes.

Experiment	Actions	N	Test	p	η_p^2	d
DOE-022	3	120	L2 vs. L0	0.929	—	0.189
DOE-024	3	360	$F(3, 348) = 1.00$	0.393	0.009	0.079
DOE-026	5	150	$F(4, 145) = 0.21$	0.935	0.006	-0.090
Combined	—	**630**	—	> 0.39	< 0.01	$\lvert d\rvert < 0.19$

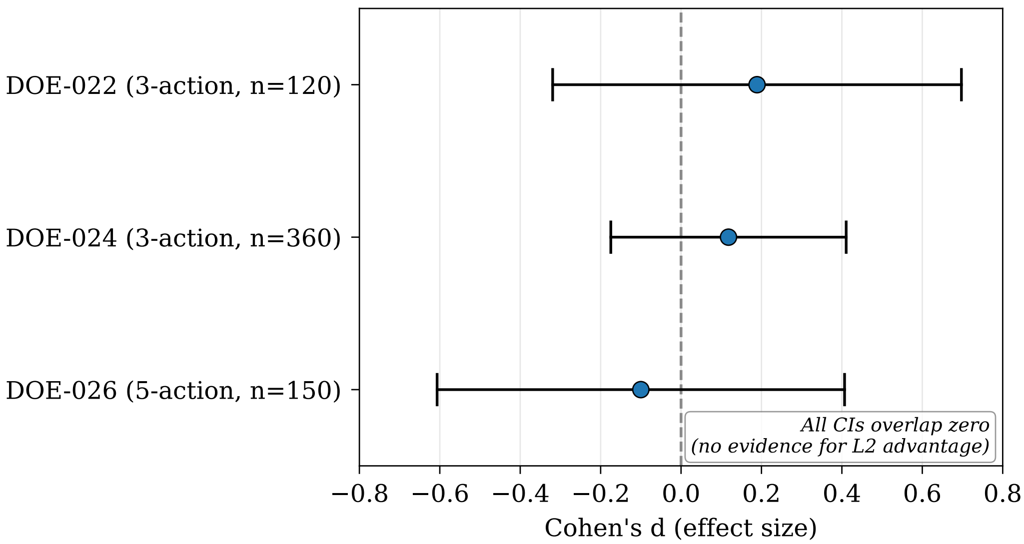


Figure 3: Forest plot of L2 RAG effect sizes (Cohen's d) with 95% CIs. All intervals overlap zero.

trades kill rate for survival producing more total kills ($\eta^2 = 0.225$); `burst_3` achieves highest kills (15.40), forming a Pareto front with `adaptive_kill`.

4.2 Core Thesis Falsification (DOE-022, DOE-024, DOE-026)

We tested whether L2 RAG retrieval improves performance through three independent experiments.

DOE-022 ($N = 120$): L2 retrieval provided no benefit over L0 ($p = 0.929$, $d = 0.189$). High/low-quality documents produced identical outcomes ($d = 0.000$). **DOE-024** ($N = 360$): Decision mode had no effect ($F(3, 348) = 1.001$, $p = 0.393$, $\eta^2 = 0.009$). **DOE-026** ($N = 150$): The smallest effect in the program ($F(4, 145) = 0.206$, $p = 0.935$, $\eta^2 = 0.006$); RAG was numerically worst (16.57 vs. mean 17.15).

Across three tests ($N = 630$), L2 RAG produces no measurable benefit. The hypothesis is **falsified** (Figure 3, Table 1).

4.3 Rate-Time Compensation Discovery (DOE-027, DOE-028)

Having falsified RAG, we investigated *why* tactical variation produces null results.

DOE-027 ($N = 210$): Attack probability varied 20–80%. Kills invariant ($F(6, 203) = 0.617$, $p = 0.717$), but survival decreased (-7.77s/10%, $p = 0.016$) while kill rate increased ($F(6, 203) = 3.736$, $p = 0.002$). These cancel precisely: kills $\approx$ kill_rate $\times$ survival$/60$. Only 6% variation across $4\times$ attack range.

DOE-028 ($N = 150$): Five temporal patterns (random, burst 2/3/5/10) at 50% attack ratio. Kills invariant ($p = 0.401$). Compensation ratio $r \times s/60k$ ranged 0.980–1.003.

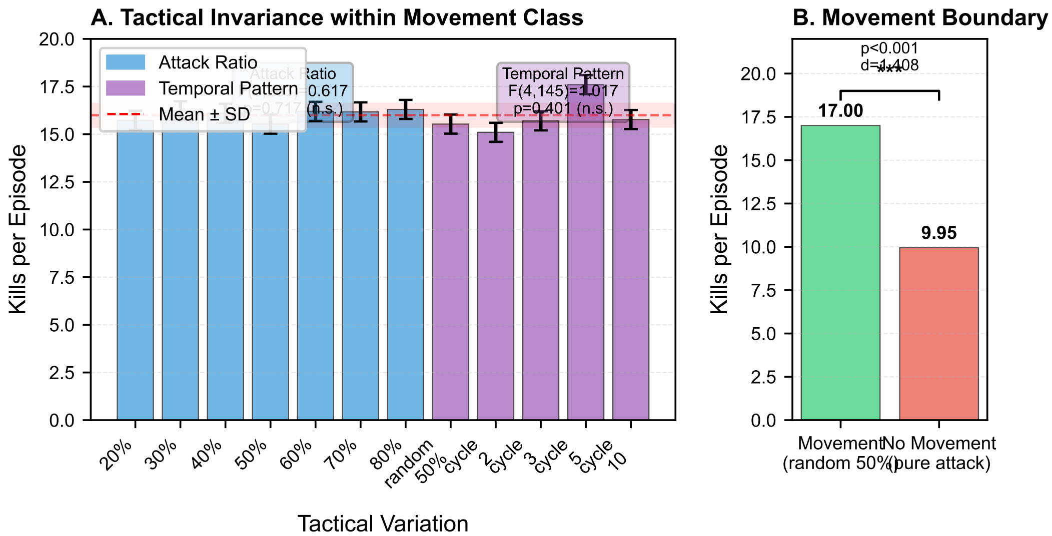


Figure 4: Full tactical invariance. **(A)** All 12 conditions within the movement class cluster around 16.0 kills (CV 3.7%). Neither attack ratio ($p = 0.717$) nor temporal pattern ($p = 0.401$) affects performance. **(B)** Movement boundary: movers achieve 70.9% more kills ($d = 1.408$, $p < 0.001$).

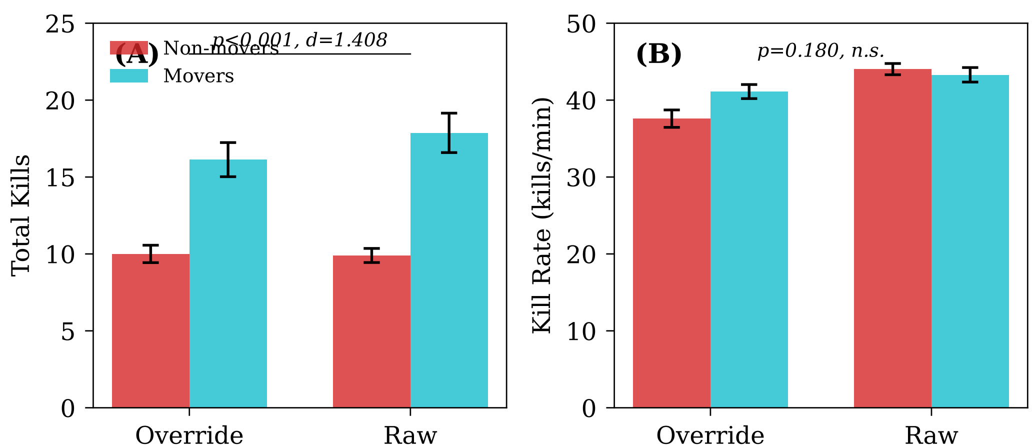


Figure 5: DOE-029 (2^2 factorial, $N = 120$). (A) Movers achieve significantly more kills ($d = 1.408$). (B) Kill rate is equivalent ($p = 0.180$): movement extends survival without reducing offense. Error bars: ± 1 SEM.

Combined evidence ($N = 360$) establishes that neither attack proportion nor temporal distribution affects kills—rate-time compensation is a conservation law (Figure 4).

4.4 Movement as Sole Determinant (DOE-029)

A 2^2 factorial crossed action pattern (with/without strafing) $\times$ health override ($N = 120$). Movement produced the largest effect in the program: $F(1, 116) = 58.4$, $p < 0.001$, $\eta^2 = 0.332$, $d = 1.408$. Movers: 17.0 ± 6.6 kills, 24.4s survival. Non-movers: 10.0 ± 2.8 kills, 15.3s survival. Health override: irrelevant ($p = 0.378$). Kill rate did *not* differ (42.2 vs. 40.8 kr/min, $p = 0.180$)—movement provides survival at zero offensive cost because strafing is orthogonal to aiming. Compensation breaks at the boundary: $C_{\text{movers}} = 17.17$ vs. $C_{\text{non-movers}} = 10.38$ (65% gap). Figure 5 shows the factorial results.

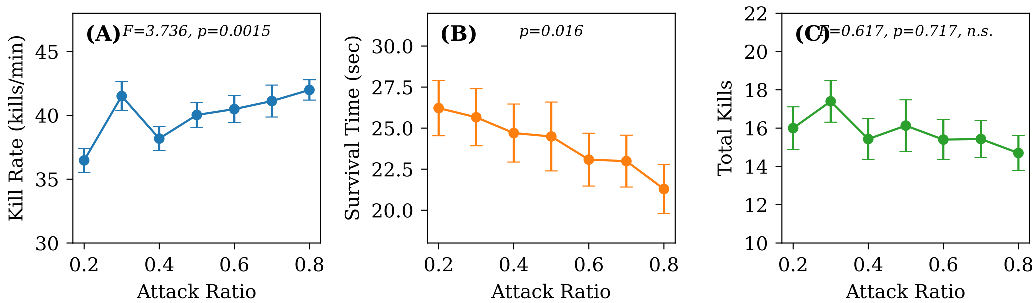


Figure 6: Rate-time compensation (DOE-027). (A) Kill rate increases with attack ratio ($p = 0.002$). (B) Survival decreases ($p = 0.016$). (C) Total kills invariant ($p = 0.717$). Error bars: ± 1 SEM ($n = 30$).

5 Analysis

Rate-time compensation. For any policy π within movement class $\mathcal{M}$: $k(\pi) \approx C_{\mathcal{M}}$, where $C_{\text{movers}} \approx 17.17$ and $C_{\text{non-movers}} \approx 10.38$. Increasing attack ratio raises kill rate r but proportionally reduces survival s, with $r \times s$ constant (within 1% across five temporal structures, DOE-028). Compensation breaks at the movement boundary because strafing provides survival without kill rate cost ($p = 0.180$). Figure 6 visualizes the mechanism.

Variance decomposition. `doom_skill` explains 72% of kill variance ($\eta^2 = 0.720$, DOE-023), movement 33% ($\eta^2 = 0.332$, DOE-029), strategy type $< 3\%$ (DOE-027/028), L2 RAG $< 1\%$ (DOE-022/024/026), agent parameters $< 1\%$ (DOE-009). Environment and movement together explain $> 80\%$ of all variance.

6 Discussion

Architecture complexity is irrelevant in `defend_the_line`—a scenario-specific result driven by single-hit enemies, open geometry, and predictable spawning that let weapon cooldown bound kill rate while movement provides the only non-compensated axis. Complexity would matter with multi-hit enemies, navigation, or competitive settings. DOE complements RL: it discovers fundamental constraints explaining *why* optimization is futile, while RL optimizes within unconstrained spaces. We recommend DOE as a preliminary tool before expensive RL training.

Limitations. All experiments use single-hit enemies; multi-hit scenarios may differ. We tested discrete 3/5-action spaces; continuous control could reveal finer effects. The Python glue layer may mask timing effects. Most experiments used `doom_skill=3`, though DOE-023 swept all difficulties. Agents use game state variables, not pixels.

7 Conclusion

Through 29 experiments (5,010 episodes), we falsified the hypothesis that RAG improves agent performance in VizDoom's `defend_the_line`:

1. **Movement is the sole determinant** ($d = 1.408$), producing 65% more kills via survival at negligible kill rate cost.
2. **Rate-time compensation** constrains tactics: $r \times s \approx C_{\mathcal{M}}$.
3. **RAG falsified** by three null results ($N = 630$, all $p > 0.39$).
4. **Environment dominates** ($\eta^2 = 0.720$), dwarfing all architecture parameters.

DOE reveals constraints that gradient-based optimization cannot discover. The architecture budget is better spent on whether the agent moves.

References

[1] Kempka, M., Wydmuch, M., Runc, G., Toczek, J., and Jaskowski, W. (2016). ViZDoom: A Doom-based AI research platform for visual reinforcement learning. In *IEEE Conference on Computational Intelligence and Games (CIG)*.

[2] de Wynter, A. (2024). Can GPT Play Doom? GPT-4 and VizDoom. *arXiv preprint*.

[3] Shinn, N., Cassano, F., Gopinath, A., Narasimhan, K., and Yao, S. (2023). Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 36.

[4] Wang, G., Xie, Y., Jiang, Y., Mandlekar, A., Xiao, C., Zhu, Y., Fan, L., and Anandkumar, A. (2023). Voyager: An open-ended embodied agent with large language models. *Transactions on Machine Learning Research (TMLR)*.

[5] Zhai, Y., Tong, S., Li, X., Cai, M., Qu, Q., Lee, Y. J., and Ma, Y. (2024). Fine-tuning large vision-language models as decision-making agents via reinforcement learning. *arXiv preprint arXiv:2405.10292*.

[6] Wang, Z., Cai, S., Chen, G., Liu, A., Ma, X., and Liang, Y. (2024). Describe, explain, plan and select: Interactive planning with large language models enables open-world multi-task agents. In *Advances in Neural Information Processing Systems (NeurIPS)*, 37.

[7] Lample, G. and Chaplot, D. S. (2017). Playing FPS games with deep reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1).

[8] Dosovitskiy, A. and Koltun, V. (2017). Learning to act by predicting the future. In *International Conference on Learning Representations (ICLR)*.

[9] Jaderberg, M., Czarnecki, W. M., Dunning, I., et al. (2019). Human-level performance in 3D multiplayer games with population-based reinforcement learning. *Science*, 364(6443), 859–865.

[10] Montgomery, D. C. (2017). *Design and Analysis of Experiments* (9th ed.). John Wiley & Sons.

[11] Snoek, J., Larochelle, H., and Adams, R. P. (2012). Practical Bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems (NeurIPS)*, 25.

[12] Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13, 281–305.

[13] Bouthillier, X., Delaunay, P., Bronzi, M., Trofimov, A., Nichyporuk, B., Szeto, J., Sepah, N., Raff, E., Madan, K., Voleti, V., Kahou, S. E., Michalski, V., Arbel, T., Pal, C., Varoquaux, G., and Vincent, P. (2021). Accounting for variance in machine learning benchmarks. In *Proceedings of Machine Learning and Systems (MLSys)*, 3.

[14] Henderson, P., Islam, R., Bachman, P., Pineau, J., Precup, D., and Meger, D. (2018). Deep reinforcement learning that matters. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

[15] Mnih, V., Kavukcuoglu, K., Silver, D., et al. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540), 529–533.

[16] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Acknowledgments and Disclosure of Funding

This research was conducted using Claude Code, an AI-assisted development tool by Anthropic. Claude Code served as an active co-investigator in experimental design, statistical analysis, and paper composition.

Reproducibility All experiment orders, reports, and statistical analyses are available in the project repository at [repository URL to be added]. Fixed seed sets are provided for all 29 DOEs. Appendices A–H provide the complete DOE summary table, findings catalogue, variance decomposition, rate-time compensation evidence, statistical methods, glossary, future work directions, and ethics statement.