Journal of Integrated Knowledge
Vol. 2 No. 7 (2026)

Forecasting Later Procgen Return Streams from Recorded PPO Diagnostics: An Adverse Temporal Comparison

Gregory Harrison[1] · Trevor Carter[2] · Jason Reed[3]

[1] United Kingdom · [2] France · [3] Australia

Published: July 08, 2026 | Article 862

Abstract

Do recorded update diagnostics improve prediction of subsequent reinforcement-learning return streams beyond their observed history? We compare two fixed ridge scores using public Procgen PPO logs for sixteen games and three source run labels. Fitting uses 624 earlier windows; evaluation uses 672 later windows from the same streams. Adding recent ratio-exceedance and squared log-probability-change summaries increases mean squared error from 0.005571 to 0.006520, a paired difference of 0.000949. Mean absolute error also increases. All thirty-three single-group loss deletions retain the adverse aggregate direction, although fourteen individual curves favor augmentation. These are finite temporal prediction results for evolving rolling-return records. They identify neither causal update effects nor the efficacy of rollback or adaptive clipping algorithms.

Forecasting Later Procgen Return Streams from Recorded PPO Diagnostics: An Adverse Temporal Comparison

Abstract

Do recorded update diagnostics improve prediction of subsequent reinforcement-learning return streams beyond their observed history? We compare two fixed ridge scores using public Procgen PPO logs for sixteen games and three source run labels. Fitting uses 624 earlier windows; evaluation uses 672 later windows from the same streams. Adding recent ratio-exceedance and squared log-probability-change summaries increases mean squared error from 0.005571 to 0.006520, a paired difference of 0.000949. Mean absolute error also increases. All thirty-three single-group loss deletions retain the adverse aggregate direction, although fourteen individual curves favor augmentation. These are finite temporal prediction results for evolving rolling-return records. They identify neither causal update effects nor the efficacy of rollback or adaptive clipping algorithms.

1. An internal diagnostic requires its own measurement argument

Policy optimization generates more information than a final return score. Training logs can include likelihood-ratio exceedances and summaries named approximate divergence. Such quantities appear to describe the update process directly, yet their predictive content depends on how they are calculated, aggregated and aligned with recorded outcomes. A variable name alone does not establish which parameter state or completed episodes it describes. This study asks whether two available diagnostics add predictive information under one declared temporal comparison, after accounting for the return stream already observed.

The clipped surrogate in proximal policy optimization changes incentives beyond a likelihood-ratio interval; it does not impose a hard bound on every ratio after optimization (Schulman et al., 2017). Procgen supplies procedurally generated games for examining reinforcement-learning behavior across training and test levels (Cobbe et al., 2019). Neither fact determines the value of a logged diagnostic as a predictor. Implementation choices can substantially change policy-gradient behavior (Engstrom et al., 2020), while reproducibility work documents variability in reported reinforcement-learning results (Henderson et al., 2018). We therefore inspect the source calculation before interpreting the numbers.

The research question is deliberately predictive: does adding recent training-worker diagnostics reduce subsequent squared forecast error relative to a score using current test return, recent test-return change, progress and game identity? Shmueli (2010) distinguishes predictive assessment from explanatory modeling. We do not estimate the effect of changing clipping or applying rollback. We compare two fully specified mappings from indexed recorded history to a later indexed change, using the same observations and loss definition for both mappings.

2. The source separates worker roles but preserves a shared evolving policy

The numerical source is OpenAI's released easy-mode, two-hundred-training-level Procgen logs, pinned to train-procgen revision 1a2ae2194a61f76a733a39339530401c024c3ad8 (OpenAI, n.d.-a). Sixteen games, three source run labels and paired training/test files produce ninety-six original CSV files. Each contains 3,051 update records. The retained source comprises 146,448 training-worker rows and the same number of test-worker rows. All ninety-six current files match their retained SHA256 hashes and Git blob identifiers. We analyze these existing files locally, without running the released training program or starting new training.

The training specification assigns zero gradient weight to test workers and exposes those workers to the test-level stream. The optimizer still communicates the shared policy update to them. Consequently, zero gradient contribution does not make each test row an independent evaluation of a frozen policy. The documented Baselines revision is 9ee399f5b20cd70ac0a871927a6cf043b478193f (OpenAI, n.d.-b). This explicit installation pin supplies a readable code contract; it does not attest to the historical executable or library build behind every released record.

At that revision, a rollout precedes optimization and logging. Episode summaries enter a return buffer with maximum length one hundred, whose mean is then reported. Buffer occupancy and episode identity are absent from the CSV. Adjacent records can reuse completed episodes, and an episode can span changing policy states. The matched training/test update grids establish an index correspondence only. They do not establish matched episode times, identical policies throughout an episode, or simultaneous record availability. This missing measurement information belongs in the source documentation, following the purpose of dataset documentation articulated by Gebru et al. (2021).

The exact implementation defines approxkl as one half the sample mean of squared differences between current and old negative log probabilities. It is a nonnegative sampled squared-change statistic, rather than a signed estimate or an exact distributional KL divergence. The likelihood ratio is the exponential of old minus current negative log probability. clipfrac is the sample fraction whose absolute ratio deviation from one exceeds the clipping parameter. With the released specification, that parameter is 0.2, and the update contains three epochs of eight minibatches. The log averages twenty-four minibatch statistics before communicator-level aggregation. Statistics are fetched with the optimization operation; they are not a single final-policy measurement, a gradient norm or a rollback count.

All selected source fields are finite, and both diagnostics vary within every training curve. Across the complete retained training table, approxkl ranges from about 0.000050 to 6.054288 and clipfrac from zero to 0.287740. Seventy-eight training records have zero clipfrac; these are valid measurements, not missing values. The declared history average and logarithm are applied without winsorization, outlier deletion or a diagnostic-dependent cohort rule. Every planned window is reconstructable, so no game, source run label or anchor is removed for missingness. This completeness concerns the released scalar fields, while episode identities and final-policy diagnostics remain unobserved.

3. Two update proposals clarify what this archive cannot establish

Zhao et al. (2023) propose Quadratic Exponential Decrease Roll-Back as a smooth modification of the PPO surrogate, discussing gradient behavior and rollback through the ratio-dependent update mechanism. The readable author-uploaded abstract and method discussion motivate distinguishing a mechanism from its telemetry. The accessible text has substantial font-encoding corruption, so this study does not reproduce its formulas or audit its proofs. More fundamentally, the released Procgen archive contains ordinary PPO training, without a QED treatment, its surrogate values or its gradient measurements. A forecast association involving approxkl cannot validate QED's gradient claims.

Xu et al. (2023) present PPO-Exp for fixed-wing formation and adapt the clipping constraint to balance exploration and surrogate-estimation considerations. Their mechanism makes the threshold itself relevant. In the present source, the threshold is fixed, and no adaptive-epsilon trajectory or exploratory-advantage estimate is available. Ratio exceedance under a fixed threshold cannot reveal the outcome of the adaptive intervention. The two proposals thus have specific roles in defining the measurement boundary: both concern the update mechanism, whereas this archive supports an observational prediction comparison from ordinary-PPO summaries. No aircraft, formation or control outcome is measured here.

4. A frozen indexed design compares the same future response

For each game, we transform test-worker return into z using the released easy-mode score anchors: subtract the lower anchor and divide by the upper-minus-lower anchor. Anchors are fixed source constants, parsed without executing their file. Values are not clipped to zero or one. This coding is a benchmark scale, without a probability interpretation, utility interpretation or success threshold. Cross-series accuracy measures require explicit attention to scale (Hyndman and Koehler, 2006); our anchor scaling is not their mean absolute scaled error procedure.

At anchor update u, the response is z at u plus one hundred minus z at u. Baseline predictors are current z, its preceding one-hundred-update change and u divided by 3,051, together with an intercept and fifteen game indicators. The alphabetically first game supplies the reference category. Source run labels identify streams but do not enter the score. The augmented score adds the mean training-worker clipfrac over updates u minus ninety-nine through u and the natural logarithm of one plus the mean approxkl over that same history. The two added predictors enter jointly; their comparison does not isolate the contribution of either diagnostic.

Fitting anchors are 200 through 1,400 in increments of one hundred, giving thirteen windows per curve and 624 windows overall. Their response endpoints extend through update 1,500. Evaluation anchors are 1,600 through 2,900, giving fourteen windows per curve and 672 windows overall. Anchor 1,500 is omitted from the scoring cohorts. This temporal boundary prevents fitting on later response endpoints, but cannot make the episode buffers independent. Structured validation must match its intended generalization unit (Roberts et al., 2017). Here evaluation concerns a later period of the same games and source streams, rather than unseen games or held-out random seeds.

Tashman (2000) identifies split rules, forecasting origins and coefficient recalibration as consequential parts of an out-of-sample assessment. Our coefficients are fitted once and remain unchanged throughout the later period. All continuous predictors use fitting-only means and population standard deviations; every recorded scale is positive. Later responses are excluded from estimation, standardization and feature definition. This explicit information boundary addresses the leakage concern described by Kapoor and Narayanan (2023). The shared source clocks still preclude a claim that the same information would have arrived simultaneously in a deployed monitoring system.

Both scores minimize fitting mean squared error plus 0.001 times the sum of squared standardized continuous slopes. Intercepts and game indicators are unpenalized. Quadratic shrinkage follows the ridge principle of trading fit against coefficient magnitude (Hoerl and Kennard, 1970); no guaranteed prediction improvement follows. The baseline has nineteen columns and the augmented score twenty-one. We use stable augmented least squares, without tuning the penalty, searching histories or replacing the declared baseline after viewing later errors. Arlot and Celisse (2010) distinguish evaluation from model selection, and Cawley and Talbot (2010) show why optimizing an assessment criterion can bias its reported performance.

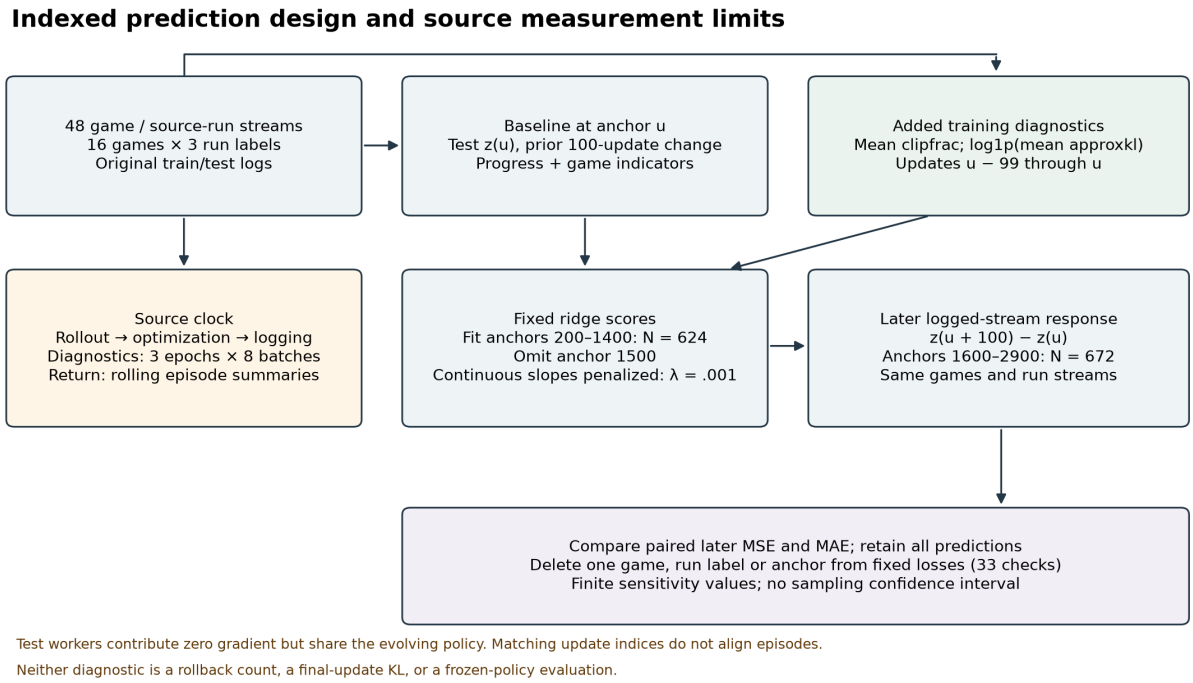

Figure 1. *Original analysis model. Arrows describe information flow in the scoring design, without representing causal effects. The train/test clock match supplies indexed windows, not independent frozen-policy evaluations.*

5. Paired loss, weighting and finite sensitivity are separate quantities

The primary contrast is augmented minus baseline mean squared error on the 672 later windows. Negative values would favor adding diagnostics. We also report mean absolute error, prediction-minus-response mean error and prediction ranges. A point forecast should be evaluated with a score suited to its task (Gneiting, 2011). Squared error defines our primary fitting and assessment target; absolute error describes a different error magnitude. Its transparent magnitude interpretation motivates reporting it alongside squared loss (Willmott and Matsuura, 2005), without claiming that either equals an operational cost.

Each of the forty-eight curves contributes fourteen later windows. Equal weighting of curves within games and equal weighting of games therefore equals the ordinary mean over this rectangular evaluation table. Weighting specifies the descriptive target, rather than automatically recovering a larger population (Solon et al., 2015). Our estimand concerns these sixteen games and three source run labels. It does not represent all procedurally generated tasks or a distribution of independently repeated training jobs.

We retain the fitted scores and delete each game, each source run label and each anchor position from the later loss table. These sixteen, three and fourteen deletions yield thirty-three finite sensitivity values. No coefficients are re-estimated. Cook (1977) studies observation influence on fitted regression estimates; our calculation instead measures loss-composition sensitivity and is not Cook's distance. The resulting minimum and maximum are neither confidence limits nor standard errors. Dependence requires a defensible resampling model (Lahiri, 2003), which an independent-window bootstrap would not supply here. The concern about few-run aggregate uncertainty in reinforcement learning remains relevant (Agarwal et al., 2021); these deletions do not resolve it.

6. Diagnostics improved earlier fit but worsened the later comparison

The augmented score has lower fitting MSE, 0.003989 versus 0.004074. Both raw designs have full column rank, and the augmented least-squares condition numbers are 26.002 and 45.305. These numerical properties establish that the declared systems were estimable. They do not establish stability under a changing return-generating process. Table 1 shows the reversal between earlier fit and later evaluation without selecting a different penalty or model in response.

Table 1. *Actual fitted-score performance. Both scores use 624 fitting windows and 672 later windows from forty-eight curves. Errors concern normalized logged-return change. Mean error is prediction minus observed change; prediction ranges are unrestricted.*

Fixed score	Fitting MSE	Later MSE	Later MAE	Later mean error	Later prediction range
History, progress and game	0.004074	0.005571	0.058864	0.031222	[-0.255399, 0.199189]
Baseline plus both diagnostics	0.003989	0.006520	0.062239	0.030809	[-0.490098, 0.197675]

Adding diagnostics increases later MSE from 0.005571 to 0.006520. The paired difference is 0.000948625, or 17.028 percent of baseline MSE. Later MAE increases by 0.003374311. The slightly smaller positive mean error of the augmented score does not offset its larger squared or absolute errors. Observed later changes include 312 decreases, sixteen exact zeros and 344 increases, spanning minus 0.300000 to plus 0.440000. All are retained. Predictions extending below the observed later minimum also remain in the score, including the augmented minimum of minus 0.490098.

Table 2. *Primary paired loss difference and every deletion family's range. Differences are augmented minus baseline. Each deletion removes losses from the fixed predictions; the ranges describe only the enumerated retained subsets and supply no sampling coverage guarantee.*

Loss composition	Values enumerated	Windows retained per value	ΔMSE or range	ΔMAE or range
Full later table	1	672	0.000948625	0.003374311
Delete one game	16	630	[0.000317328, 0.001202242]	[0.000968637, 0.005369443]
Delete one run label	3	448	[0.000641243, 0.001389529]	[0.002246846, 0.005526731]
Delete one anchor	14	624	[0.000676517, 0.001051811]	[0.002618598, 0.003905159]

Every deletion leaves both loss differences positive. The smallest game-deleted MSE difference is 0.000317328, so no single game removal reverses the aggregate direction. However, fourteen of forty-eight individual curve means have negative paired MSE differences and thirty-four have positive differences. Figure 2 retains all of these values. The aggregate penalty of augmentation is compatible with local benefit in some streams; it does not establish that every game or trajectory is harmed by the extra predictors.

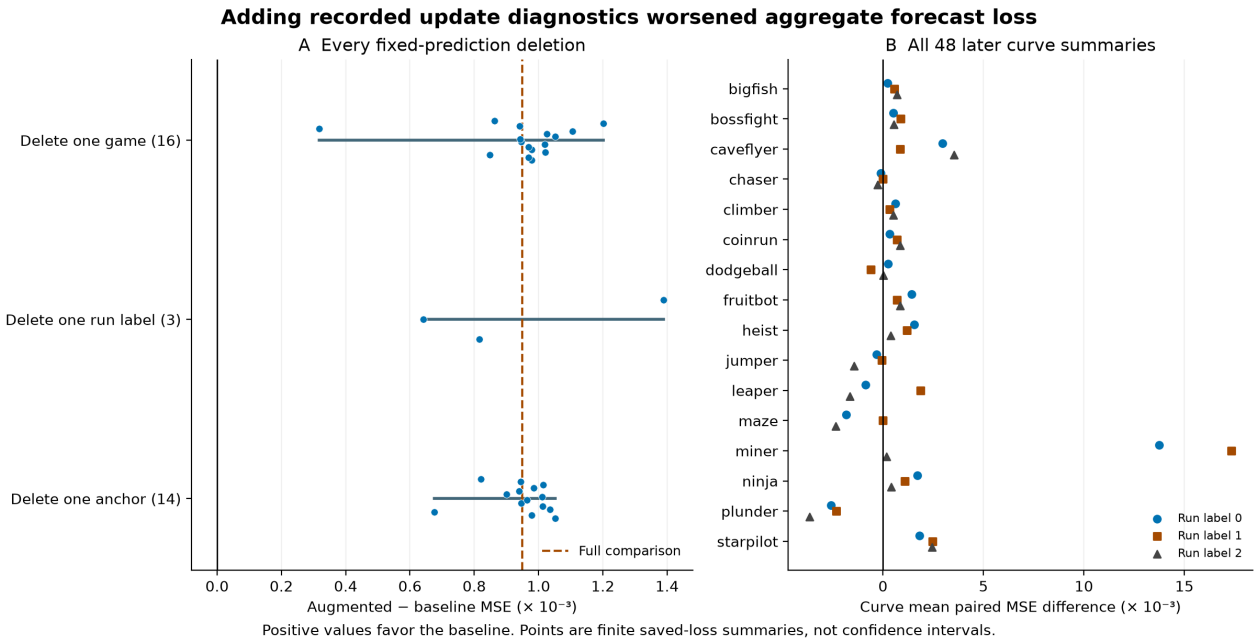

Figure 2. *Observed loss heterogeneity. Positive differences favor the baseline. The dashed reference is the full-table difference. Range lines join finite deletion extrema and are not confidence intervals. Curve means each contain fourteen later anchor windows.*

An independent calculation re-parses all ninety-six original CSV files using a separate reader and reconstructs all 1,296 analysis windows with explicit update indices and scalar summation. Its largest feature discrepancy is 3.33×10^{-16}. An alternate rank-revealing QR least-squares solution agrees with the primary singular-value solution within 5.00×10^{-15} across 2,592 predictions. Direct factorization methods provide complementary numerical checks (Golub and Van Loan, 2013). All model metrics, thirty-three deletion rows and forty-eight curve summaries are independently recalculated. Agreement demonstrates computational consistency, without recovering episode identities or supplying statistical replication.

7. What the adverse result says about telemetry

The declared diagnostic augmentation does not improve this later-period prediction task. A smaller earlier residual loss is therefore insufficient evidence that these internal summaries transfer usefully to later logged changes. Nonstationarity, nonlinear relationships, aggregation and redundancy with the baseline history are possible explanations. This design does not distinguish them. It also cannot conclude that internal diagnostics contain no useful information under another score, horizon or measurement system. Changing those specifications after seeing this comparison would require a fresh evaluation rather than reinterpretation of the present loss.

The clock limits matter even for a predictive claim. The response is a difference between rolling means with unknown episode overlap and variable episode durations. Normalization equalizes a declared scale, not the amount of new experience represented by each record. Progress extrapolates beyond the fitting anchors, while the game categories and source streams remain shared. Diagnostics reflect multiple minibatch states rather than a final update displacement. The analysis consequently describes conditional indexed prediction in an evolving archive. It supplies no verified real-time warning rule, causal policy-update effect, rollback efficacy estimate or evaluation of adaptive clipping.

This is a secondary analysis of an archive also used for distinct return-gap, selection and threshold-persistence questions. The present unit is the paired forecast loss with internal update features and an explicit logger-phase audit. Reuse of the same archive is disclosed; this result is not an independent replication of those earlier analyses. An alternative official Stable-Baselines3 model archive was

inspected before design freeze, but contained evaluation arrays and monitor records without internal update diagnostics. That failed source path is preserved, and its returns do not enter the analysis.

NumPy performs the primary numerical solution (Harris et al., 2020), while SciPy supplies the independent QR solver (Virtanen et al., 2020). Pandas assembles the tables and grouped loss exports (McKinney, 2010), and Matplotlib renders the original model and result figures (Hunter, 2007). The local environment records NumPy 2.5.4, Pandas 3.0.6 and Matplotlib 3.11.2; remaining runtime versions accompany the evidence bundle. Source hashes, protocol, transformations and prediction-level outputs bind the calculation to its inputs. Computational reproducibility requires this inspectable chain (Peng, 2011), including intermediate artifacts and program versions (Sandve et al., 2013).

The source code and released log snapshot retain the OpenAI MIT notice. The analysis makes no blanket license claim about external papers or the rejected model archive. New analysis code, minimum numerical projections, complete loss tables and source locators are retained locally. No human participants or new physical experiments were involved in this numerical reuse. Accountable authorship, funding and conflict declarations, source-claim review, rights review and final scientific approval remain pending. Automated and AI-assisted checks do not represent human verification or submission authorization.

References

1. OpenAI. (n.d.-a). train-procgen: easy-mode,200-training-level PPO logs and training specification (revision 1a2ae2194a61f76a733a39339530401c024c3ad8) [Data and source code]. https://github.com/openai/train-procgen/tree/1a2ae2194a61f76a733a39339530401c024c3ad8. Accessed 17 November 2026.
2. Golub, Gene H.; Van Loan, Charles F. (2013). Matrix Computations. Johns Hopkins University Press, fourth edition. https://doi.org/10.56021/9781421407944.
3. Cobbe, Karl; Hesse, Christopher; Hilton, Jacob; Schulman, John. (2019). Leveraging Procedural Generation to Benchmark Reinforcement Learning. arXiv preprint. https://arxiv.org/abs/1912.01588.
4. Roberts, David R.; Bahn, Volker; Ciuti, Simone; Boyce, Mark S.; Elith, Jane; Guillera-Arroita, Gurutzeta; Hauenstein, Severin; Lahoz-Monfort, José J.; Schröder, Boris; Thuiller, Wilfried; Warton, David I.; Wintle, Brendan A.; Hartig, Florian; Dormann, Carsten F. (2017). Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure. Ecography 40(8), 913-929. https://doi.org/10.1111/ecog.02881.
5. Lahiri, S. N. (2003). Resampling Methods for Dependent Data. Springer Series in Statistics. https://doi.org/10.1007/978-1-4757-3803-2.
6. Cawley, Gavin C.; Talbot, Nicola L. C. (2010). On Over-fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation. Journal of Machine Learning Research. https://www.jmlr.org/papers/v11/cawley10a.html.
7. Gebru, Timnit; Morgenstern, Jamie; Vecchione, Briana; Vaughan, Jennifer Wortman; Wallach, Hanna; Daumé, Hal; Crawford, Kate. (2021). Datasheets for datasets. Communications of the ACM 64(12), 86-92. https://doi.org/10.1145/3458723.
8. Hyndman, Rob J.; Koehler, Anne B. (2006). Another look at measures of forecast accuracy. International Journal of Forecasting 22(4), 679-688. https://doi.org/10.1016/j.ijforecast.2006.03.001.
9. Henderson, Peter; Islam, Riashat; Bachman, Philip; Pineau, Joelle; Precup, Doina; Meger, David. (2018). Deep Reinforcement Learning That Matters. Proceedings of the AAAI Conference on Artificial Intelligence 32(1). https://doi.org/10.1609/aaai.v32i1.11694.
10. Zhao, Runhao; Xu, Dan; Jian, Shuoqu; Tan, Tianle; Sun, Xiaohui; Zhang, Wanchao. (2023). Quadratic Exponential Decrease Roll-Back: An Efficient Gradient Update Mechanism in Proximal Policy Optimization. 2023 2nd International Conference on Machine Learning, Cloud Computing and Intelligent Mining (MLCCIM), 65-70. https://doi.org/10.1109/mlccim60412.2023.00015.
11. Gneiting, Tilmann. (2011). Making and Evaluating Point Forecasts. Journal of the American Statistical Association 106(494), 746-762. https://doi.org/10.1198/jasa.2011.r10138.
12. Sandve, Geir Kjetil; Nekrutenko, Anton; Taylor, James; Hovig, Eivind. (2013). Ten Simple Rules for Reproducible Computational Research. PLoS Computational Biology 9(10), e1003285. https://doi.org/10.1371/journal.pcbi.1003285.

13. Hunter, John D. (2007). Matplotlib: A 2D Graphics Environment. Computing in Science & Engineering 9(3), 90-95. https://doi.org/10.1109/mcse.2007.55.

14. Peng, Roger D. (2011). Reproducible Research in Computational Science. Science 334(6060), 1226-1227. https://doi.org/10.1126/science.1213847.

15. Arlot, Sylvain; Celisse, Alain. (2010). A survey of cross-validation procedures for model selection. Statistics Surveys 4(none), 40-79. https://doi.org/10.1214/09-ss054.

16. Xu, Dan; Guo, Yunxiao; Yu, Zhongyi; Wang, Zhenfeng; Lan, Rongze; Zhao, Runhao; Xie, Xinjia; Long, Han. (2023). PPO-Exp: Keeping Fixed-Wing UAV Formation with Deep Reinforcement Learning. Drones 7(1), 28. https://doi.org/10.3390/drones7010028.

17. McKinney, Wes. (2010). Data Structures for Statistical Computing in Python. Proceedings of the Python in Science Conference, 56-61. https://doi.org/10.25080/majora-92bf1922-00a.

18. Virtanen, Pauli; Gommers, Ralf; Oliphant, Travis E.; Haberland, Matt; Reddy, Tyler; Cournapeau, David; Burovski, Evgeni; Peterson, Pearu; Weckesser, Warren; Bright, Jonathan; van der Walt, Stéfan J.; Brett, Matthew; Wilson, Joshua; Millman, K. Jarrod; Mayorov, Nikolay; Nelson, Andrew R. J.; Jones, Eric; Kern, Robert; Larson, Eric; Carey, C J; Polat, İlhan; Feng, Yu; Moore, Eric W.; VanderPlas, Jake; Laxalde, Denis; Perktold, Josef; Cimrman, Robert; Henriksen, Ian; Quintero, E. A.; Harris, Charles R.; Archibald, Anne M.; Ribeiro, Antônio H.; Pedregosa, Fabian; van Mulbregt, Paul; SciPy 1.0 Contributors; Vijaykumar, Aditya; Bardelli, Alessandro Pietro; Rothberg, Alex; Hilboll, Andreas; Kloeckner, Andreas; Scopatz, Anthony; Lee, Antony; Rokem, Ariel; Woods, C. Nathan; Fulton, Chad; Masson, Charles; Häggström, Christian; Fitzgerald, Clark; Nicholson, David A.; Hagen, David R.; Pasechnik, Dmitrii V.; Olivetti, Emanuele; Martin, Eric; Wieser, Eric; Silva, Fabrice; Lenders, Felix; Wilhelm, Florian; Young, G.; Price, Gavin A.; Ingold, Gert-Ludwig; Allen, Gregory E.; Lee, Gregory R.; Audren, Hervé; Probst, Irvin; Dietrich, Jörg P.; Silterra, Jacob; Webber, James T; Slavič, Janko; Nothman, Joel; Buchner, Johannes; Kulick, Johannes; Schönberger, Johannes L.; de Miranda Cardoso, José Vinícius; Reimer, Joscha; Harrington, Joseph; Rodríguez, Juan Luis Cano; Nunez-Iglesias, Juan; Kuczynski, Justin; Tritz, Kevin; Thoma, Martin; Newville, Matthew; Kümmerer, Matthias; Bolingbroke, Maximilian; Tartre, Michael; Pak, Mikhail; Smith, Nathaniel J.; Nowaczyk, Nikolai; Shebanov, Nikolay; Pavlyk, Oleksandr; Brodtkorb, Per A.; Lee, Perry; McGibbon, Robert T.; Feldbauer, Roman; Lewis, Sam; Tygier, Sam; Sievert, Scott; Vigna, Sebastiano; Peterson, Stefan; More, Surhud; Pudlik, Tadeusz; Oshima, Takuya; Pingel, Thomas J.; Robitaille, Thomas P.; Spura, Thomas; Jones, Thouis R.; Cera, Tim; Leslie, Tim; Zito, Tiziano; Krauss, Tom; Upadhyay, Utkarsh; Halchenko, Yaroslav O.; Vázquez-Baeza, Yoshiki. (2020). SciPy 1.0: fundamental algorithms for scientific computing in Python. Nature Methods 17(3), 261-272. https://doi.org/10.1038/s41592-019-0686-2.

19. Harris, Charles R.; Millman, K. Jarrod; van der Walt, Stéfan J.; Gommers, Ralf; Virtanen, Pauli; Cournapeau, David; Wieser, Eric; Taylor, Julian; Berg, Sebastian; Smith, Nathaniel J.; Kern, Robert; Picus, Matti; Hoyer, Stephan; van Kerkwijk, Marten H.; Brett, Matthew; Haldane, Allan; del Río, Jaime Fernández; Wiebe, Mark; Peterson, Pearu; Gérard-Marchant, Pierre; Sheppard, Kevin; Reddy, Tyler; Weckesser, Warren; Abbasi, Hameer; Gohlke, Christoph; Oliphant, Travis E. (2020). Array programming with NumPy. Nature 585(7825), 357-362. https://doi.org/10.1038/s41586-020-2649-2.

20. Engstrom, Logan; Ilyas, Andrew; Santurkar, Shibani; Tsipras, Dimitris; Janoos, Firdaus; Rudolph, Larry; Madry, Aleksander. (2020). Implementation Matters in Deep Policy Gradients: A Case Study on PPO and TRPO. arXiv preprint. https://arxiv.org/abs/2005.12729.

21. Cook, R. Dennis. (1977). Detection of Influential Observation in Linear Regression. Technometrics 19(1), 15-18. https://doi.org/10.1080/00401706.1977.10489493.

22. Solon, Gary; Haider, Steven J.; Wooldridge, Jeffrey M. (2015). What Are We Weighting For? Journal of Human Resources 50(2), 301-316. https://doi.org/10.3368/jhr.50.2.301.

23. Shmueli, Galit. (2010). To Explain or to Predict? Statistical Science 25(3), 289-310. https://doi.org/10.1214/10-sts330.

24. Hoerl, Arthur E.; Kennard, Robert W. (1970). Ridge Regression: Biased Estimation for Nonorthogonal Problems. Technometrics 12(1), 55-67. https://doi.org/10.1080/00401706.1970.10488634.

25. Schulman, John; Wolski, Filip; Dhariwal, Prafulla; Radford, Alec; Klimov, Oleg. (2017). Proximal Policy Optimization Algorithms. arXiv preprint. https://arxiv.org/pdf/1707.06347.

26. Agarwal, Rishabh; Schwarzer, Max; Castro, Pablo Samuel; Courville, Aaron; Bellemare, Marc G. (2021). Deep Reinforcement Learning at the Edge of the Statistical Precipice. arXiv preprint. https://arxiv.org/abs/2108.13264.

27. Tashman, Leonard J. (2000). Out-of-sample tests of forecasting accuracy: an analysis and review. International Journal of Forecasting 16(4), 437-450. https://doi.org/10.1016/s0169-2070(00)00065-0.

28. Willmott, Cort J.; Matsuura, Kenji. (2005). Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. Climate Research 30, 79-82. https://doi.org/10.3354/cr030079.

29. OpenAI. (n.d.-b). Baselines: PPO2 model, rollout, logging and MPI optimizer implementation (revision 9ee399f5b20cd70ac0a871927a6cf043b478193f) [Source code]. https://github.com/openai/baselines/tree/9ee399f5b20cd70ac0a871927a6cf043b478193f. Accessed 17 November 2026.

30. Kapoor, Sayash; Narayanan, Arvind. (2023). Leakage and the reproducibility crisis in machine-learning-based science. Patterns 4(9), 100804, Article 100804. https://doi.org/10.1016/j.patter.2023.100804.