Journal of Insights in Basic and Applied Sciences
مجلة رؤى في العلوم الأساسية والتطبيقية
Volume 2, Issue 3, 2026, Pages: 644-649
Journal homepage: https://jibas.com.ly/index.php/jibas/en

Warmup Hurts: A Two-Stage Schedule Ablation and Physical-Perturbation Robustness Analysis of Lion in PPO

Osama Meftah Wanis [1*], Hamza Alzarok[2]
[1] College of Electronic Technology, Bani-Walid, Libya
[2] Department of Electrical and Electronic Engineering, Faculty of Engineering, Bani Waleed University, Libya

الإحماء يضرّ: دراسة استئصال جدول زمني من مرحلتين وتحليل متانة الاضطراب الفيزيائي لمحسِّن Lion في PPO

أسامة مفتاح ونيس[1*]، حمزة الزروق[2]
[1] كلية التقنية الإلكترونية، بني وليد، ليبيا
[2] قسم الهندسة الكهربائية والإلكترونية، كلية الهندسة، جامعة بني وليد، ليبيا

*Corresponding author: osamayagaa@gmail.com

Received: June 26, 2026	Accepted: August 25, 2026	Published: September 07, 2026

Abstract:
Despite AdamW's continued status as the default optimizer for PPO, its two-buffer moment state (2P memory) costs twice what the sign-based optimizer Lion requires (1P). Lion's update magnitude, however, tracks the instantaneous learning rate directly and exclusively, without anything that plays the role of AdamW's internal self-normalization when a schedule decays; this structurally predisposes it to chattering — the sliding-mode-control oscillation documented in discontinuous, sign-based control laws — and should leave its competitiveness against AdamW schedule-sensitive in a way AdamW's is not. We isolate this sensitivity with a two-stage ablation on PPO in four MuJoCo continuous-control environments (HalfCheetah-v4, Walker2d-v4, Ant-v4, Hopper-v4; n = 15 seeds/arm, Welch's t-test with Holm–Bonferroni correction), structurally decomposing three Lion arms (lion_raw: no schedule; lion_cosine: cosine decay from step 0; lion_warmup_cosine: linear warmup and plateau, then the same cosine decay) into two independent, additive comparisons. In the three passively stable environments (HalfCheetah-v4, Walker2d-v4, Ant-v4), decay is beneficial and warmup is significantly detrimental, reducing mean return by 39–55% and increasing chattering by 12–18% relative to decay alone, with the same directional pattern — inverted for warmup, the opposite of its usual supervised-learning justification — confirmed in post-training policy entropy. In the minimally stable Hopper-v4, no single-step ablation reaches significance, though the combined lion_raw-vs-lion_warmup_cosine contrast does. We further report an exploratory, post-hoc physical-perturbation (added mass, sensor noise) robustness analysis: mean post-fine-tune return recovery ranges from 54.7% (Walker2d-v4, worst) to 113.7% (Ant-v4, best, an outright improvement), and appears governed by a property distinct from clean-condition passive stability.

Keywords: Lion optimizer; AdamW; learning-rate warmup; ablation study; chattering; robustness; Proximal Policy Optimization

الملخص
رغم استمرار AdamW كمُحسِّن افتراضي لخوارزمية PPO، فإن حالته اللحظية ذات المخزنين (ذاكرة 2P) تُكلّف ضعف ما يتطلبه المُحسِّن القائم على الإشارة Lion (1P). غير أن مقدار تحديث Lion يتتبّع معدل التعلم اللحظي بشكل مباشر وحصري، دون أي آلية تقوم مقام التطبيع الذاتي الداخلي لـ AdamW عند تراجع الجدول الزمني؛ وهو ما يجعله عرضة بنيويًا لظاهرة الرفرفة (chattering)،

وينبغي أن يجعل تنافسيته مع AdamW حساسة للجدول الزمني بخلاف AdamW. نعزل هذه الحساسية عبر دراسة استئصال من مرحلتين على PPO في أربع بيئات تحكم مستمر من MuJoCo (HalfCheetah-v4، Walker2d-v4، Ant-v4، Hopper-v4؛ 15 بذرة عشوائية لكل تهيئة، باستخدام اختبار Welch's t مع تصحيح (Holm-Bonferroni)، حيث نُحلّل ثلاث تهيئات لـ Lion (lion_raw: بلا جدول زمني؛ lion_cosine: تراجع جيبي منذ البداية؛ lion_warmup_cosine: إحماء خطي وهضبة ثم التراجع الجيبي نفسه) إلى مقارنتين مستقلتين ومتراكمتين. في البيئات الثلاث المستقرة سلبيًا (Ant-v4، Walker2d-v4، HalfCheetah-v4)، يكون التراجع مفيدًا بينما الإحماء ضارًا بشكل معنوي، إذ يُخفّض متوسط العائد بنسبة 39-55% ويزيد الرفرفة بنسبة 12-18% مقارنة بالتراجع وحده، ويتأكد هذا النمط الاتجاهي - المعكوس بالنسبة للإحماء، عكس مبرره المعتاد في التعلم الخاضع للإشراف - في إنتروبيا السياسة بعد التدريب. أما في بيئة Hopper-v4 الأقل استقرارًا، فلا يصل أي استئصال منفرد إلى الدلالة الإحصائية، رغم أن المقارنة المجمّعة بين lion_raw وlion_warmup_cosine تصل إليها. كما نُقدّم تحليلًا استكشافيًا بعديًا لمتانة الاضطراب الفيزيائي (زيادة الكتلة، ضوضاء الاستشعار): يتراوح متوسط استعادة العائد بعد إعادة الضبط الدقيق بين 54.7% (الأسوأ، Walker2d-v4) و113.7% (الأفضل، Ant-v4، وهو تحسّن فعلي)، ويبدو أنه محكوم بخاصية مختلفة عن الاستقرار السلبي في الحالة النظيفة.

الكلمات المفتاحية: مُحسِّن Lion؛ AdamW؛ إحماء معدل التعلم؛ دراسة استئصال؛ الرفرفة (Chattering)؛ المتانة؛ PPO

Introduction

Discovered via symbolic program search, Lion (Evolved Sign Momentum; Chen et al., 2023) cuts AdamW's optimizer-state memory in half: rather than storing two moment buffers, it keeps a single momentum buffer and updates parameters using only the sign of a momentum-blended direction signal c_t, θ_t = θ_t−1 − η_t(sign(c_t) + λθ_t−1). Because sign(c_t) ∈ {−1, 0, +1}, every coordinate update is fixed at exactly the current learning rate η_t whenever c_t ≠ 0; unlike AdamW, whose f(m̂_t,v̂_t) term partially absorbs the effect of a decaying schedule, Lion has no such cushion against decay. The practical consequence is asymmetric: cutting η_t while a policy is still exploring removes Lion's entire remaining update capacity, whereas the same cut applied only once exploration has largely settled costs comparatively little. This same fixed-magnitude, sign-only mechanism is also why Lion should be prone to chattering — the persistent, high-frequency oscillation that classical sliding-mode and variable-structure control theory attributes to discontinuous, sign-based control laws (Utkin, 1977); its update behaves, structurally, like a relay or bang-bang controller acting in parameter space, and we track chattering as a control-smoothness outcome alongside return throughout this paper.

We isolate this schedule-sensitivity mechanism directly with PPO (Schulman et al., 2017) across four MuJoCo continuous-control benchmarks, under a seed-controlled design (n = 15/arm) with Holm–Bonferroni correction. Three Lion arms anchor the comparison — lion_raw (no schedule), lion_cosine (cosine decay from step 0), and lion_warmup_cosine (the same cosine decay preceded by linear warmup and a plateau) — constructed so each differs from its neighbor in exactly one structural respect. This lets the pairwise contrast lion_raw-vs-lion_cosine be read as an isolated test of decay, and lion_cosine-vs-lion_warmup_cosine as an isolated test of warmup, rather than folding both manipulations into a single three-way comparison. The Results section reports the full ablation, the accompanying post-training policy-entropy pattern, the realized learning-rate curves that rule out a schedule-implementation confound, and an exploratory, post-hoc physical-perturbation robustness analysis added after the confirmatory design was locked in. This paper is a companion to a larger seed-controlled statistical comparison of the same optimizer family across the same four MuJoCo benchmarks; that companion study reports the main-effect comparisons against AdamW, while the present paper focuses specifically on decomposing the Lion schedule itself into its decay and warmup components.

Material and methods

Optimizer Update Rules and Step-Size Coupling

Decoupling weight decay from the gradient step is what distinguishes AdamW (Loshchilov & Hutter, 2019) from the Adam optimizer (Kingma & Ba, 2015) it builds on:

$$\theta_t = \theta_{t-1} - \eta_t \left(\hat{m}_t / (\sqrt{\hat{v}_t} + \delta) + \lambda\theta_{t-1} \right), \quad (1)$$

with bias-corrected moments m̂_t = m_t/(1−β1^t), v̂_t = v_t/(1−β2^t). Lion (Chen et al., 2023) takes a different route: it blends a momentum direction signal c_t and separately maintains a smoothed buffer m_t,

$$c_t = \beta_1 m_{t-1} + (1-\beta_1) g_t, \quad m_t = \beta_2 m_{t-1} + (1-\beta_2) g_t, \quad (2)$$

then updates using only the sign of c_t: θ_t = θ_t−1 − η_t(sign(c_t) + λθ_t−1), keeping a single buffer m_t in memory rather than AdamW's two. Because sign(c_t) discards all magnitude information, every nonzero Lion coordinate update is exactly η_t in size:

$$|\Delta\theta_t| = \eta_t \text{ whenever } c_t \neq 0, \quad (3)$$

whereas AdamW's step, $|\Delta\theta_t| = \eta_t \cdot f(\hat{m}_t, \hat{v}_t)$ with $f = |\hat{m}_t|/(\sqrt{\hat{v}_t} + \delta)$, is scaled by a self-normalizing multiplier that stays roughly stable regardless of η_t. This produces an asymmetry under a decaying schedule: AdamW's effective step is cushioned whenever the gradient direction is consistent, while Lion's shrinks in lockstep with η_t, one-to-one. If escaping a poor early region depends on sustained update magnitude, decaying η_t too soon removes Lion's entire capacity to make that escape, not merely a compensated fraction of it — the structural hypothesis this paper's ablation is designed to test.

Chattering, Environments, and Statistical Framework

Since Equation (3) carries no counterpart to AdamW's magnitude-damping term $f(\hat{m}_t, \hat{v}_t)$, we monitor chattering, $C_{chatter}$, defined as the mean absolute first difference of the evaluation action trace ($|a_t - a_{t-1}|$), first averaged within an episode and then across episodes and seeds for each arm. Four MuJoCo (Todorov et al., 2012) environments accessed through Gymnasium (Towers et al., 2024) give a graduated test of kinematic sensitivity: HalfCheetah-v4 and Ant-v4 rarely or never trigger fall termination and are passively stable; Walker2d-v4 terminates on falls and demands moderate balance; Hopper-v4 terminates on falls and is highly sensitive to equilibrium, balancing on a single ground-contact point. Every statistical comparison uses Welch's t-test (which does not assume equal variances) with Holm–Bonferroni correction (Holm, 1979) applied across six pairwise comparisons per environment among the four matched-condition arms (adamw, lion_raw, lion_cosine, lion_warmup_cosine); the fifth arm, adamw_tuned, uses RL Zoo hyperparameters (Raffin et al., 2021) under an unmatched configuration and is reported descriptively rather than folded into this corrected family. Each arm draws on $n = 15$ seeds, sized following the power-analysis guidance for seed-controlled deep RL comparisons in Colas et al. (2018). We report Cohen's d (Cohen, 1988) as the effect-size measure throughout, with a negative value indicating that the first-named arm in a comparison performed worse.

Arms Under Ablation

The two-stage ablation uses three of the four matched-condition arms, sharing network architecture, GAE parameters (Schulman et al., 2016), clipping threshold (Schulman et al., 2017), entropy coefficient, and rollout length $T = 2048$ across all three: lion_raw — no schedule at all, η_t is held constant for the full run; lion_cosine — identical starting rate to lion_raw, but annealed via a cosine curve beginning at training step 0, no warmup, no plateau; lion_warmup_cosine — the same cosine decay as lion_cosine, only preceded by a linear warmup ramp and a plateau held at the full rate. Each environment's version of this three-phase schedule is timed to its own training length (2048 steps/iteration): on the 1000-iteration horizon (HalfCheetah-v4, Hopper-v4), warmup covers iterations 0–80, the plateau 80–600, and decay 600–1000; on the 1500-iteration horizon (Walker2d-v4, Ant-v4), the corresponding windows are 0–120, 120–900, and 900–1500. Because lion_warmup_cosine adds exactly one thing to lion_cosine (the warmup-plus-plateau phase) and lion_cosine adds exactly one thing to lion_raw (decay), the corresponding pairwise rows in the ablation table isolate warmup and decay as independent, additive effects rather than conflating them; both rows belong to the same six-comparison, Holm-corrected family described above rather than a test corrected on its own.

Post-Training Policy Entropy

As a third outcome measure alongside return and the chattering metric $C_{chatter}$, we report post-training policy entropy at test time, averaged across seeds within each arm, to test whether the ablation's effect on return and control smoothness is accompanied by a measurable shift in exploratory behavior.

Physical-Perturbation Robustness Protocol

After the confirmatory comparisons were finalized, we ran an exploratory, post-hoc perturb-then-fine-tune adaptation protocol, applied identically to all four environments and all five arms (including adamw and the descriptive adamw_tuned reference): each saved checkpoint was exposed to a physical perturbation (added mass, +50%, or sensor noise) and then given a short re-adaptation (fine-tuning) window, after which return and control metrics were re-evaluated. This protocol was not part of the pre-specified confirmatory design, and none of the confirmatory ablation claims depend on it; we report only the headline, descriptive findings the Discussion draws on.

Results and discussion

Two-Stage Ablation: Isolating Decay from Warmup

In the three passively stable environments, both ablation steps point the same direction: decay is beneficial (lion_cosine outperforms lion_raw), reaching significance in Walker2d-v4 and Ant-v4, with a non-significant trend in the same direction on HalfCheetah-v4 ($d = -0.81$, $p_{holm} = 0.106$); warmup is significantly detrimental

to return (lion_cosine > lion_warmup_cosine), significant in Walker2d-v4 (d = 1.56) and Ant-v4 (d = 2.07), with the same trend on HalfCheetah-v4 (d = 0.80). Concretely, adding warmup to the identical cosine schedule reduces mean return by 39% on Walker2d-v4 (3541.8 → 2153.9) and 55% on Ant-v4 (3038.1 → 1373.0), while increasing chattering by 11.9% (0.260 → 0.291) and 18% (0.340 → 0.401) respectively — the schedule change and its consequences move together, consistent with withholding the full learning rate during early training delaying discovery of an efficient, low-chattering gait.

Table 1 The two ablation-relevant pairwise comparisons, restricted to the three Lion arms, from the six-comparison Holm-corrected family per environment. Bold rows are Holm-significant (p_holm < 0.05).

Environment	Comparison (ablates)	t	p_holm	Cohen's d	95% CI	Sig.
HalfCheetah-v4	lion_raw vs lion_cosine (decay)	-2.22	0.106	-0.81	[-1.56, -0.07]	no
	lion_cosine vs lion_warmup_cosine (warmup)	2.18	0.106	0.80	[+0.05, +1.54]	no
Walker2d-v4	**lion_raw vs lion_cosine (decay)**	**-12.04**	**<0.0001**	**-4.40**	**[-5.75, -3.04]**	**yes**
	lion_cosine vs lion_warmup_cosine (warmup)	**4.26**	**0.0004**	**1.56**	**[+0.73, +2.38]**	**yes**
Hopper-v4	lion_raw vs lion_cosine (decay)	-2.09	0.185	-0.76	[-1.51, -0.02]	no
	lion_cosine vs lion_warmup_cosine (warmup)	-0.94	1.000	-0.34	[-1.06, +0.38]	no
Ant-v4	**lion_raw vs lion_cosine (decay)**	**-9.49**	**<0.0001**	**-3.47**	**[-4.62, -2.31]**	**yes**
	lion_cosine vs lion_warmup_cosine (warmup)	**5.66**	**0.0001**	**2.07**	**[+1.17, +2.97]**	**yes**

Overall lion_raw-vs-lion_warmup_cosine (combined, non-additive contrast): HalfCheetah-v4 d = 0.00 (ns); Walker2d-v4 d = −1.97, p_holm = 0.0002 (Holm-significant); Hopper-v4 d = −1.11, p_holm = 0.032 (Holm-significant, the only significant contrast on this environment); Ant-v4 d = −2.48, p_holm < 0.0001 (Holm-significant).

The overall lion_raw-vs-lion_warmup_cosine contrast is also Holm-significant on Walker2d-v4 (d = −1.97), and adamw remains significantly behind lion_warmup_cosine there too (d = −1.79); stacking warmup on top of decay does not recover AdamW-competitive performance — lion_warmup_cosine still clears both lion_raw and adamw by a large margin even after warmup erodes part of the cosine-decay advantage. On Ant-v4, complete removal of decay (lion_raw, mean return 307.3) is the worst-performing Lion configuration across all four environments in this study, confirming that lion_cosine's advantage depends on an unimpeded decay curve rather than a constant rate.

On Hopper-v4, neither single-step ablation reaches significance (lion_raw-vs-lion_cosine d = −0.76; lion_cosine-vs-lion_warmup_cosine d = −0.34); only the combined, non-additive lion_raw-vs-lion_warmup_cosine contrast survives Holm correction (d = −1.11), an interaction between the two schedule manipulations rather than an effect of either alone. Schedule structure is thus a first-order driver of performance in the three passively stable environments, but decay and warmup, considered separately, are not significant in the one minimally-stable morphology tested.

Entropy, Schedule Verification, and Robustness

The ablation's effect on return and chattering is mirrored in post-training policy entropy across all four environments. Adding decay (lion_raw → lion_cosine) raises test-time entropy by 5.92 on HalfCheetah-v4 (from −7.72 to −1.80), 4.21 on Hopper-v4 (from −4.63 to −0.42), 7.09 on Walker2d-v4 (from −11.80 to −4.71), and 4.21 on Ant-v4 (from −19.08 to −14.87); adding warmup on top of decay (lion_cosine → lion_warmup_cosine) reverses part of this gain, reducing entropy by 4.41, 2.57, 6.11, and 3.28 respectively, with lion_warmup_cosine's entropy lying between lion_raw's and lion_cosine's in every environment, trending toward lion_raw. Decay is thus the dominant driver of the entropy difference between the three arms, and warmup only partially reverses it — the same schedule change that costs return and increases chattering also measurably pushes the policy toward more deterministic, lower-entropy test-time behavior, extending the ablation's effect to a third, independent measure.

All four environments reproduce the same three realized learning-rate shapes (constant for lion_raw; decay from step 0 for lion_cosine; warmup, plateau, then the same decay curve for lion_warmup_cosine), confirming the schedule logic is environment-independent and ruling out a schedule-implementation confound: the lion_cosine-

vs-lion_warmup_cosine performance gap reflects the intended structural difference (presence or absence of early-training warmup), not an unintended difference in the decay portion of the schedule, on any environment tested.

Under the exploratory perturb-then-fine-tune protocol (added mass, +50%, and sensor noise, followed by a short re-adaptation window, all five arms), Ant-v4 showed the best post-fine-tune return recovery of the four environments (mean 113.7% of pre-perturbation clean return, averaged across all five arms and both perturbation conditions), HalfCheetah-v4 next (88.4%), Hopper-v4 intermediate (79.0%), and Walker2d-v4 worst (54.7%). Passive mechanical stability and robustness to disturbance therefore appear to be related but distinct properties of these environments; the full arm-level and chattering-level breakdown of this protocol, together with the joint-level control diagnostics (contact force, actuator saturation, torque chattering) that motivate a sliding-mode-control reading of it, is reserved for a dedicated control-theoretic companion study, which summarizes here only the headline recovery figures needed to support the Discussion's practical implications.

Cross-Environment Diagnostics

Beyond return, chattering, and entropy, we additionally examined a battery of exploratory, post-hoc joint-level diagnostics (survivorship confounds in the episode-length/chattering relationship, adamw_tuned's policy entropy, actuator saturation fraction, fall rate and episode length, and steady-state locomotion velocity) across the four primary environments. No single arm's behavior on these diagnostics is environment-independent, reinforcing that morphology, not a fixed optimizer property, governs which configuration best matches a task's actuation and stability demands; a full joint-level, control-theoretic treatment of these diagnostics is beyond the present paper's decay/warmup focus.

Discussion

The Cost of Warmup in Passively Stable Systems

Warmup's cost is systematic across the passively stable environments: significant on Walker2d-v4 ($t = 4.26$, $p_{holm} = 0.0004$, $d = 1.56$) and Ant-v4 ($t = 5.66$, $p_{holm} = 0.0001$, $d = 2.07$), with a matching non-significant trend on HalfCheetah-v4 ($d = 0.80$). A plausible account is that, unlike Hopper-v4, these environments' passive stability gives the policy room to recover from early mistakes, so withholding the full learning rate through warmup delays discovery of an efficient gait rather than preventing early collapse — the opposite of warmup's usual supervised-learning justification, and consistent with the step-size-coupling asymmetry (Equation (3)): a schedule that erodes Lion's already-uncompensated update magnitude before exploration is complete costs it disproportionately. Warmup's cost is relative, not absolute: lion_warmup_cosine still beats lion_raw on every environment tested and is nominally the highest-return arm on Hopper-v4 (2185.9, not statistically distinguishable) — never the worst matched-condition arm, just beaten by cosine decay alone on stable morphologies.

Practical Implications

Practitioners deploying Lion should not import a warmup schedule from supervised-learning practice by default: on passively stable morphologies it costs 39–55% mean return and 12–18% additional chattering relative to decay alone, confirmed independently in post-training entropy. On the one minimally stable morphology tested (Hopper-v4), warmup's cost was not detectable in isolation, and the general recommendation to validate the specific morphology rather than assume transfer extends to the choice of warmup versus decay-only scheduling specifically. Robustness to physical perturbation should be treated as a property to validate on the target system rather than inferred from clean-condition passive-stability class: Ant-v4 combined strong clean-condition performance with the best perturbation recovery in this study, but Walker2d-v4 combined strong clean-condition performance with the worst recovery, and Hopper-v4's return recovery under mass perturbation moved in the opposite direction from Ant-v4's under the identical condition.

Conclusion

This paper isolated the schedule-sensitivity mechanism structurally predicted by Lion's step-size coupling (Equation (3)): because Lion's update magnitude tracks the learning rate exactly, with nothing playing the role of AdamW's self-normalizing compensation, a schedule that withholds full update capacity early should cost more on passively stable morphologies with room to recover than on minimally stable ones. A two-stage ablation confirms this sensitivity is structural: on Walker2d-v4 and Ant-v4, decay helps and warmup hurts substantially (39–55% return, 12–18% chattering, both Holm-significant); on HalfCheetah-v4 the same directional trend appears for return but does not reach significance ($p_{holm} = 0.106$); on Hopper-v4, neither single-step ablation moves the needle, though the combined lion_raw-vs-lion_warmup_cosine contrast does. The effect replicates in post-training policy entropy across all four environments, and realized learning-rate traces rule out a schedule-

implementation confound. An exploratory, post-hoc physical-perturbation analysis further shows that robustness to disturbance is governed by a property distinct from clean-condition passive stability, and should be validated per-system rather than inferred from morphology class. The overall recommendation is narrow and specific: lion_cosine, without warmup, is a strong, often superior default for morphologically stable continuous-control PPO under a matched tuning budget, and importing a supervised-learning warmup schedule by default actively costs performance on exactly the systems where Lion is otherwise most competitive.

Compliance with ethical standards

Disclosure of conflict of interest

The authors declare that they have no conflict of interest.

References

[1] Chen, X., Liang, C., Huang, D., Real, E., Wang, K., Liu, Y., Pham, H., Dong, X., Luong, T., Hsieh, C.-J., Lu, Y., & Le, Q. V. (2023). Symbolic discovery of optimization algorithms. In Advances in Neural Information Processing Systems 36 (NeurIPS).

[2] Cohen, J. (1988). Statistical power analysis for the behavioral sciences (2nd ed.). Lawrence Erlbaum Associates.

[3] Colas, C., Sigaud, O., & Oudeyer, P.-Y. (2018). How many random seeds? Statistical power analysis in deep reinforcement learning experiments. arXiv preprint arXiv:1806.08295.

[4] Holm, S. (1979). A simple sequentially rejective multiple test procedure. Scandinavian Journal of Statistics, 6(2), 65–70.

[5] Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In International Conference on Learning Representations (ICLR).

[6] Loshchilov, I., & Hutter, F. (2019). Decoupled weight decay regularization. In International Conference on Learning Representations (ICLR).

[7] Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., & Dormann, N. (2021). Stable-Baselines3: Reliable reinforcement learning implementations. Journal of Machine Learning Research, 22(268), 1–8.

[8] Schulman, J., Moritz, P., Levine, S., Jordan, M., & Abbeel, P. (2016). High-dimensional continuous control using generalized advantage estimation. In International Conference on Learning Representations (ICLR).

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

[10] Todorov, E., Erez, T., & Tassa, Y. (2012). MuJoCo: A physics engine for model-based control. In 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 5026–5033).

[11] Towers, M., Kwiatkowski, A., Terry, J., Balis, J. U., De Cola, G., Deleu, T., Goulão, M., Kallinteris, A., Krimmel, M., KG, A., Perez-Vicente, R., Pierré, A., Schulhoff, S., Tai, J. J., Tan, H., & Younis, O. G. (2024). Gymnasium: A standard interface for reinforcement learning environments. arXiv preprint arXiv:2407.17032.

[12] Utkin, V. I. (1977). Variable structure systems with sliding modes. IEEE Transactions on Automatic Control, 22(2), 212–222.