ASYMMETRIC MEMBRANE CREDIT FOR SPIKING ACTOR–CRITIC LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

A stateful spiking core produces both binary events and continuous membrane states, but an event-only Critic does not directly read the latter. We therefore introduce Asymmetric Membrane Credit (AMC), a training-time interface that gives the Critic a zero-initialized readout of the shared pre-threshold membrane without adding an Actor inference branch. A matched detached control retains the trainable readout while blocking only its additional encoder-gradient route, separating membrane-informed Critic learning from direct encoder credit. A reusable stateful spiking Actor–Critic substrate implements AMC across PPO, SAC and TD3, with common policy computation and rule-specific state reconstruction and encoder updates. Across six MuJoCo-v5 tasks and three paired training seeds per setting, the complete AMC pathway achieves higher mean area under the learning curve (AUC) than event-only controls in 17 of 18 settings. Fixed final policies achieve higher mean return than the same controls in 16 settings when evaluated on 20 common reset seeds. Direct credit improves both AUC and final-policy return on SAC HalfCheetah but reduces both on SAC Walker in all three training pairs. In a separate Ant-v4 system comparison at the same interaction budget, Membrane-Credit PPO (MCrPPO) improves mean AUC by 64.8% over a SpikeGym reproduction retaining its native training schedule. Paired GPU and neuromorphic evaluation verifies Actor-only execution of selected MCrPPO policies after Critic removal. Together, AMC and the substrate combine improved training performance with explicit control over state access and encoder updates, while retaining an Actor-only deployment path.

1 INTRODUCTION

Directly trained spiking neural networks already support continuous control. Hybrid systems combine spiking Actors with ANN Critics (Tang et al., 2021; Xu et al., 2026), and fully spiking Actor–Critics have been trained with PPO (Zanatta et al., 2024; Zhang et al., 2025). These advances motivate a common computational substrate for stateful spiking Actor–Critic learning: a reusable policy-side architecture on which internal-state access and representation learning can be studied across learning rules.

Persistent neural state makes this more than a network substitution. State carried during interaction must be distinguished from state supplied to a training update. Our on-policy learner uses recorded recurrent states, whereas replay-based learners reconstruct states under the online and target parameters. A shared encoder also couples policy and Critic learning through its updates. We therefore separate common spiking forward computation from rule-specific state construction and encoder-update ownership, while retaining each learner's objectives.

Existing work studies Actor–Critic representation sharing and gradient coupling (Garcin et al., 2025), and spiking controllers use training-time proxy targets or privileged guides to support learning (Xu et al., 2025; Van den Berghe et al., 2025). Our question concerns the shared spiking core itself: which internal state the Critic reads, and whether its readout supplies direct encoder credit, hereafter direct credit.

The same spiking-core update produces a continuous membrane potential and a binary event. With other inputs and states fixed, distinct membrane values can produce the same event vector. A membrane readout can distinguish these values without changing the Actor's event-mediated access.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Reading the membrane and allowing gradients through that readout are separate choices. A surrogate specifies differentiation through the event map (Neftci et al., 2019); it does not expose membrane amplitude to the Critic's forward prediction.

We introduce *Asymmetric Membrane Credit* (AMC), a Critic-only readout of the shared pre-threshold membrane. Unlike SpikingNav's symmetric post-reset membrane access (Zhang et al., 2026), AMC retains an event-mediated Actor path. Appendix I provides extended comparisons. Membrane-Credit PPO (MCrPPO) is the PPO implementation; SAC and TD3 extend the study to stochastic and deterministic off-policy learning. Here, *full-SNN* denotes spiking representations and membrane readouts without a separate conventional ANN Critic, not event-only arithmetic.

Our contributions are threefold:

- **AMC: membrane-informed Critic learning without an additional Actor inference branch.** The interface reuses continuous state already computed by the shared spiking core, without privileged observations. Zero initialization preserves the initial Critic prediction and introduces no direct encoder credit before the readout learns. A matched detached control retains the trainable readout while blocking only its additional encoder-gradient route. Together with a local gradient identity, this control separates learning with membrane access from enabling direct credit at the same parameter count and forward function class.
- **A reusable stateful spiking Actor–Critic substrate.** The substrate separates common policy computation from the state and update procedures required by PPO, SAC and TD3. Each independently trained instance shares a stateful encoder between Actor and Critic. PPO jointly updates it from policy and value objectives using recorded states; SAC and TD3 reconstruct online and target states separately and assign encoder updates to the Q objective. This organization supports matched AMC interventions while retaining each learner's objectives and an Actor that executes without the Critic.
- **Cross-rule training gains and Actor-only neuromorphic evaluation.** Across 162 E/D/A training runs in 54 matched triplets, the complete AMC pathway achieves higher mean learning-curve AUC than event-only controls in 17 of 18 task–algorithm settings, and higher mean final-policy return in 16 settings under the common 20-reset evaluation. Detached comparisons identify task-dependent benefits and costs of direct credit. Separate system comparisons include a 64.8% mean AUC gain over a SpikeGym reproduction on Ant-v4 at the same interaction budget. Paired GPU and Lynxi HP201 evaluation tests Actor-only execution after Critic removal.

2 Stateful Spiking Substrate and AMC

The substrate separates the spiking computation used to act from the procedures used to train it. Within each learner, Actor and Critic share an encoder but can read different neural quantities. Across learners, the architecture is reused with independently trained parameters. Figure 1 shows this organization and the AMC interface.

2.1 Shared stateful spiking computation

Across PPO, SAC and TD3, the policy uses an affine input projection and a shared layer of 512 leaky integrate-and-fire (LIF) neurons. The membrane and event updates are

$$h_t = 0.2h_t^- \odot (1 - z_t^-) + W_x\hat{x}_t + b_x, \qquad z_t = \mathbf{1}[h_t > 0.5].$$

Here, $\hat{x}_t$ is the normalized observation, (h_t^-, z_t^-) is the episode-masked incoming state, and $\eta = (W_x, b_x)$ denotes the shared encoder parameters. The Actor accesses the shared representation through z_t and updates a continuous action membrane,

$$o_t = 0.2o_t^- + W_o z_t + b_o.$$

A rule-specific bounded action transform reads o_t; MCrPPO uses an affine–tanh Gaussian policy.

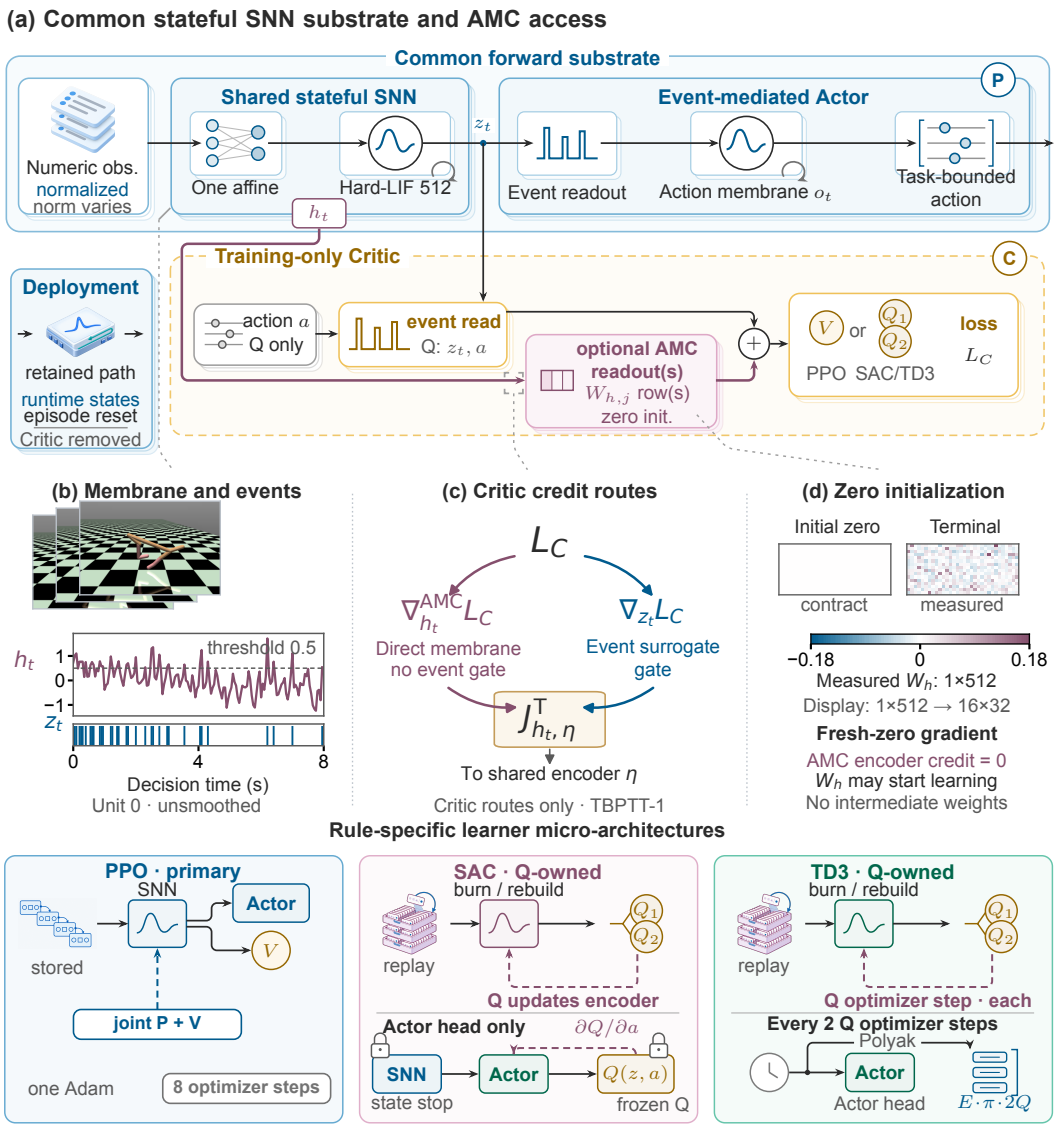

Figure 1: Shared stateful spiking computation and the AMC interface. (a) Event-mediated Actor access and a training-only Critic. (b) Recorded MCrPPO membrane/event traces and simulator frames; images are not policy inputs. (c) Local Critic-to-encoder gradient routes. (d) Specified zero initialization and measured final MCrPPO readout weights. E omits the membrane readout; D trains it while blocking only its added encoder-gradient route; A enables that route. The lower panels show PPO joint policy/value learning and SAC/TD3 Q-owned encoder updates. Diagnostic details are in Appendix D.6.

Encoder and Actor states persist across decisions and collection boundaries, and reset at episode boundaries. This execution state is distinct from the state supplied to a training update: PPO uses recorded states, whereas off-policy learners reconstruct states from raw-observation prefixes under online or target parameters. Memory and temporal credit assignment are distinct (Ni et al., 2023); here, persistent execution is trained with truncated gradients. The recurrence and reset boundary are shared, while normalization, Critic state and encoder-update ownership remain rule-specific. Membranes, normalization and bounded action transforms remain continuous. Appendix B specifies the dynamics and task bounds.

Table 1: State-access controls within each learning rule. AMC (A) is the complete attached condition; D is its matched detached control. A and D share the forward function class and differ only in the additional encoder-gradient route.

Condition	Membrane read	Direct credit	Interpretation
Event-only (E)	No	No	A–E: complete added pathway
Detached (D)	Yes	No	D–E: membrane-informed Critic learning
AMC (A)	Yes	Yes	A–D: enable the added encoder gradient

2.2 Asymmetric membrane access

Critic access to neural state is distinct from the choice of surrogate. At fixed other inputs and states, distinct membrane vectors h, h' can produce the same event vector. An event-only Critic then receives the same drive; a membrane readout can distinguish them when $W_{h,j}(h - h') \neq 0$. AMC adds this access only to Critic head j:

$$c_{j,t}^{\mathrm{AMC}} = c_{j,t}^{\mathrm{event}}(z_t, \chi_t) + W_{h,j} h_t. \tag{1}$$

PPO adds this drive to its recurrent value-membrane carry and decodes V_t; off-policy heads use it directly as Q_j, with χ_t including the queried action. The readout reuses shared neural state without privileged observations or another Actor inference branch.

In MCrPPO, the value membrane is $u_t = 0.2u_t^- + W_V z_t + b_V + W_h h_t$, decoded as $V_t = \sigma_p^V u_t + \mu_p^V$ using collection-time value statistics (μ_p^V, σ_p^V) for update cycle p (Appendix B.1.1). The readout adds 512 parameters in PPO or 1024 across twin-Q heads. Zero initialization changes neither the Critic prediction nor the encoder gradient. Readout weights can learn on the first Critic update and supply direct encoder credit on subsequent updates.

With Actor parameters, normalization, initial states, task bounds and input/reset/noise sequences fixed, removing the Critic leaves Actor states and actions unchanged in exact arithmetic (Proposition 1). MCrPPO deployment retains (h, z, o) and removes u, W_h and the remaining Critic and training state. The Critic can shape the shared representation during training without becoming a dependency of Actor execution.

2.3 Separating membrane access from direct credit

Table 1 defines AMC (A) as the complete attached condition with $W_h h_t$, D as its matched detached control with $W_h \,\mathrm{sg}(h_t)$, and E as the event-only control without the added readout. Here sg denotes stop-gradient. A–E tests the complete pathway; D–E adds membrane access and readout capacity with the direct route disabled; A–D enables that route at the same parameter count and forward function class. D retains ordinary event-mediated Critic gradients. Separately trained conditions collect different data, so their performance differences do not form an additive causal mediation decomposition.

For a fixed MCrPPO training sample, let $d_{o,t} = \partial L_\pi / \partial o_t$, $d_{u,t} = \partial L_V / \partial u_t$, and $D_t = \mathrm{diag}(s(h_t))$. We use $\widetilde{\nabla}$ for the implemented surrogate gradient and $J_{h_t,\eta}$ for the current-step encoder Jacobian with incoming state held fixed. The encoder gradient is

$$\widetilde{\nabla}_\eta (L_\pi + L_V) = J_{h_t,\eta}^{\mathsf{T}} \left[D_t (W_o^{\mathsf{T}} d_{o,t} + W_V^{\mathsf{T}} d_{u,t}) + W_h^{\mathsf{T}} d_{u,t} \right]. \tag{2}$$

The final term bypasses the event-surrogate derivative. For the rectangular surrogate $s(h) = \mathbf{1}[0 < h < 1]$, this term can be nonzero where that derivative is zero. It remains an additional route for noncompact surrogates. The encoder Jacobian or cancellation across samples can nevertheless make the resulting parameter gradient zero.

At identical parameters, input, recorded recurrent state, target and value-loss branch, A and D yield identical value predictions and W_h gradients. Their encoder-gradient difference is

$$\widetilde{\nabla}_\eta L_V^A - \widetilde{\nabla}_\eta L_V^D = J_{h_t,\eta}^{\mathsf{T}} W_h^{\mathsf{T}} d_{u,t}. \tag{3}$$

D retains the event-mediated Critic term in Eq. 2; separately trained A and D policies need not follow identical trajectories. Twin-Q direct terms sum across heads (Eq. 40).

These are current-step identities under one-step truncated backpropagation through time (TBPTT-1), with recorded states or finite replay burn-in, not identities for gradients through the full recurrent trajectory. Appendix A distinguishes the isolated gradient contribution from the actual optimizer update.

2.4 PPO, SAC AND TD3 IMPLEMENTATIONS

The common recurrence is trained using the rule-specific procedures in Table 2. MCrPPO jointly updates the encoder from the PPO policy and recurrent normalized-value objectives (Schulman et al., 2017), holding recorded recurrent states fixed under TBPTT-1 (Algorithm 1).

Table 2: State handling, Critic architecture and update ownership.

Rule	Critic	Training state	Encoder updates	Target updates
PPO	Recurrent normalized V	Recorded state; TBPTT-1	Policy and value	None
SAC	Twin Q	Separate online/target burn-in	Q loss	Encoder and Q heads each Q step; no target Actor
TD3	Twin Q	Separate online/target burn-in	Q loss	Full target every second Q step

SAC and TD3 retain their objectives (Haarnoja et al., 2018a; Fujimoto et al., 2018). Finite replay burn-in reconstructs online and target states under each network's own parameters. Each Q head processes shared events and the queried action through 256 private, nonrecurrent threshold units, then adds its membrane readout. Only Q losses update the encoder. The Actor phase rebuilds states with that updated encoder, then freezes encoder and Q parameters while retaining the surrogate Q-to-action gradient. Thus, the Actor head learns from the updated representation without another encoder update. At fixed state, the membrane term has zero query-action derivative. Delayed TD3 Actor-head updates do not freeze the policy: intervening Q updates can change its encoder. Algorithms 2 and 3 in Appendix B.2 give the complete updates.

3 EXPERIMENTAL DESIGN

Matched E/D/A comparisons assess the complete membrane pathway and its additional encoder-gradient route. Separate system and deployment studies evaluate control performance and Actor-only execution (Table 3).

Table 3: Experimental questions and evidence. E/D/A contrasts are paired within each learning rule; system and deployment comparisons address different properties.

Question	Comparison	Property assessed
Complete-pathway utility	A–E	Learning-curve AUC and final-policy return
Direct-credit benefit	A–D, alongside D–E	Added encoder-gradient effect
System performance	MuJoCo-v5 references; Ant-v4 SpikeGym	Control under the specified protocols
Final-policy sensitivity	Monitoring; final policies	Dependence on evaluation conditions
Actor-only execution	GPU and Lynxi HP201	Control preservation and execution cost

The controlled study uses six MuJoCo-v5 tasks (Todorov et al., 2012; Towers et al., 2024), with 54 matched E/D/A triplets (three training seeds per task and rule) at $10\,\mathrm{Mi}$ PPO or $1\,\mathrm{Mi}$ SAC/TD3 steps, where $1\,\mathrm{Mi} = 2^{20}$ steps. Each rule keeps one hyperparameter configuration across tasks, with task-specific dimensions and bounds and independently trained models. E is our AMC-free control; SNN-PPO (adapted) is a separate system reference.

The $20\,\mathrm{Mi}$ system comparison includes MCrPPO, ANN-PPO (Raffin et al., 2021) with matched rollout size and optimizer-update count, and SNN-PPO (adapted) (Zanatta et al., 2024), using five common training seeds per task/method. Acquisition AUC covers the first $10\,\mathrm{Mi}$; final policies are evaluated after $20\,\mathrm{Mi}$. Deployment compares 18 selected Actors on GPU and Lynxi HP201. Run

inclusion, configurations, surrogate sensitivity and deployment protocols are in Appendices C, F, H.1 and G.1.

Five fixed monitoring reset seeds supply curves and AUC normalized by its integration-window length. Reevaluating 276 fixed final models on 20 new common reset seeds yields 5520 episodes. Returns are averaged within models before comparing paired training runs; episodes do not add training replicates (Agarwal et al., 2021). Algorithm-specific inference adapters define the common evaluation; the separate PPO reset-only study retains its original evaluator (Appendix C).

4 RESULTS

4.1 TRAINING GAINS FROM THE COMPLETE AMC PATHWAY

The complete AMC pathway achieves higher mean learning-curve AUC than E on all six PPO tasks and 11 of 12 SAC/TD3 settings (Tables D3 and E2). Figure 2 shows the corresponding trajectories. These results indicate broad average gains under both value and Q objectives.

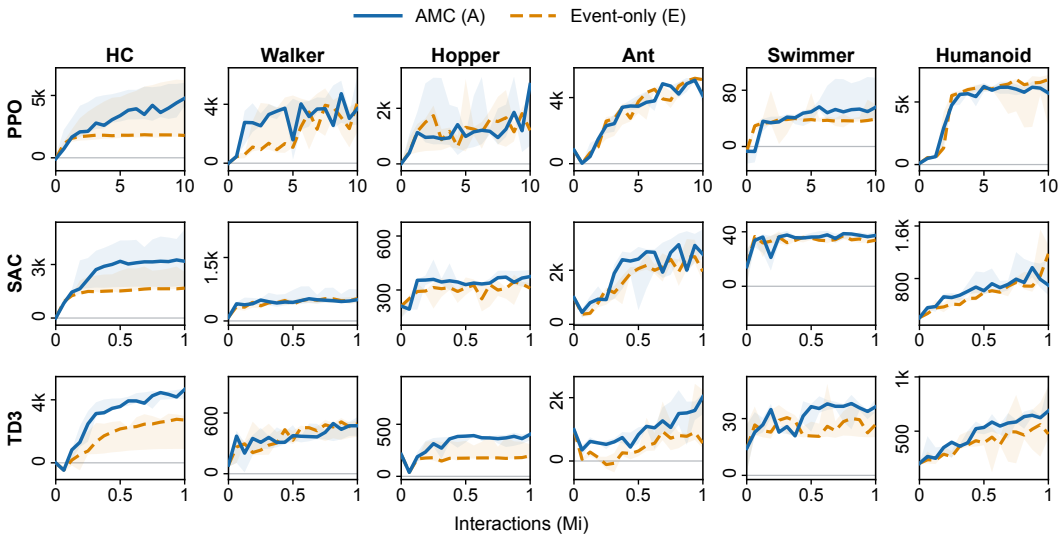

Figure 2: Learning curves with and without the complete AMC pathway. Lines and shading show medians and ranges across three training seeds on six MuJoCo-v5 tasks. The 17 displayed checkpoints use every tenth PPO checkpoint through 10 Mi and all SAC/TD3 checkpoints through 1 Mi. Mean AUCs average per-run integrals on complete grids, not displayed medians. Figure 3 reports final-policy D–E/A–D contrasts; Figure 16 provides complete E/D/A curves.

On the common 20-reset evaluation, mean final-policy A–E differences are positive in 16 of 18 settings, with all three training-pair differences positive in six settings. The two negative means occur on Humanoid under PPO and SAC. TD3 HalfCheetah improves from 2194.7 to 4394.6 in mean return (+100.2%); both SAC and TD3 improve on Hopper in every training pair. These comparisons establish the observed utility of the complete pathway, not the benefit of its direct encoder-gradient route in isolation.

4.2 MEMBRANE ACCESS AND DIRECT CREDIT HAVE DISTINCT EFFECTS

The detached control distinguishes adding a trainable membrane readout without direct credit (D–E) from enabling its additional encoder-gradient route (A–D). Figure 3 reports their final-policy effects.

On SAC Walker, mean final-policy differences are $+1296.6$ for D–E and -1242.8 for A–D, with each sign consistent across all three pairs; A–E is only $+53.8$. On SAC HalfCheetah, A–D improves AUC and final-policy return in every pair, with a mean return difference of $+1226.3$. Thus, a trainable membrane readout can help even when its direct encoder gradient does not.

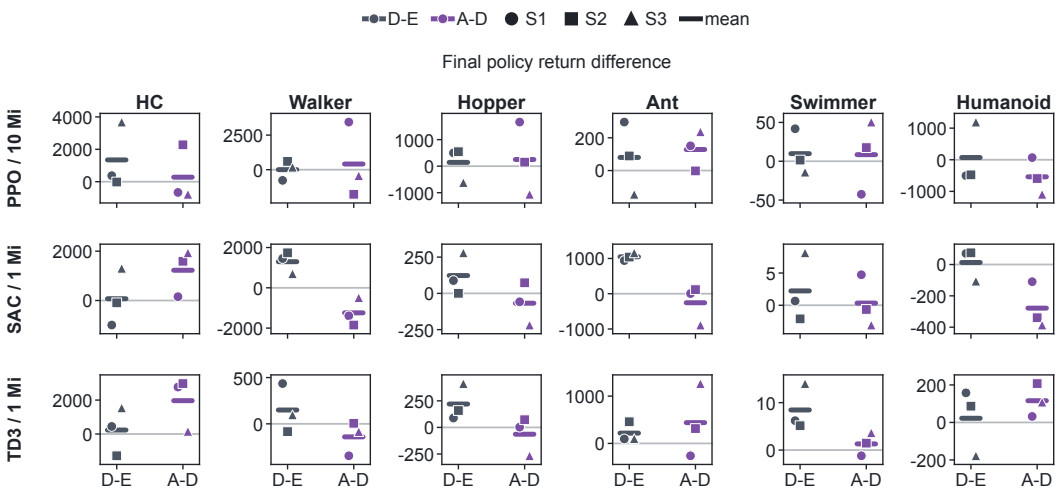

Figure 3: Effects on final-policy return of membrane-informed Critic learning (D–E) and direct encoder credit (A–D). Each point is a paired training-run difference averaged over 20 shared evaluation seeds; bars show the mean across three pairs. Training budgets: 10 Mi PPO, 1 Mi SAC/TD3.

Mean final-policy D–E and A–D differences are positive in 18 and 11 settings, respectively (monitoring comparison: Appendix C.4). D is therefore both an informative control and a useful alternative in observed settings where direct credit is detrimental.

Matched A/D shadow branches from six final checkpoints pass forward-equality and Eq. 3 checks in all six cases (Appendix D.5). In the single-seed surrogate study, A has higher HalfCheetah AUC than D with a fast-sigmoid surrogate, while ATan reverses this ordering. The positive example extends beyond compactly supported surrogates but does not establish consistent effects across families (Appendices D and H).

4.3 System-level control performance

The system comparison uses five common training seeds per task, with runs paired by seed across methods. Architectures and learner protocols differ across systems; Appendix F documents the configurations and included runs.

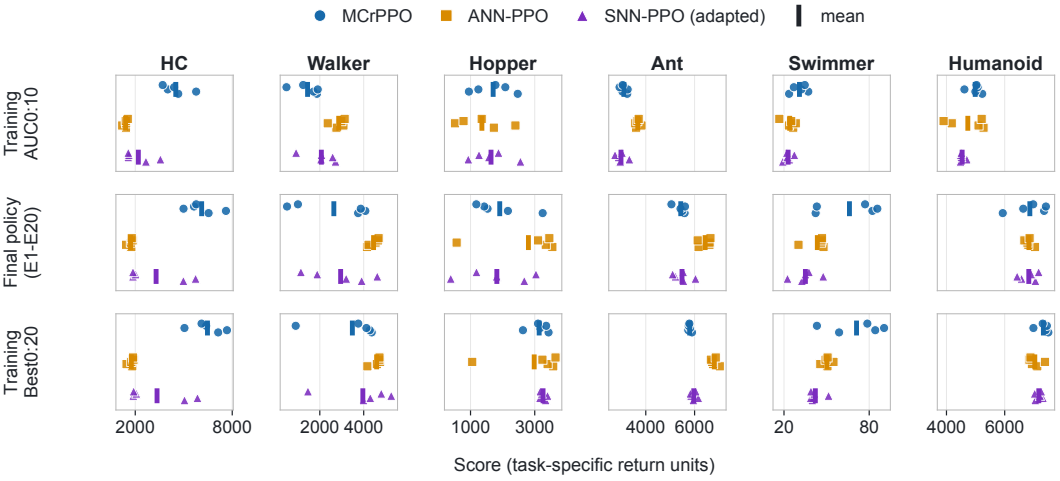

Figure 4: System performance across five common training seeds per task: points are individual training replicates and bars indicate means. Final-policy returns use all 20 shared evaluation seeds; $\mathrm{AUC}_{0:10}$ and $\mathrm{Best}_{0:20}$ use evaluations during the 20 Mi training runs. Complete results: Appendix F.

MCrPPO achieves higher mean acquisition AUC than SNN-PPO (adapted) on five of six tasks and than ANN-PPO on four (Figure 4). In the common 20-reset final-policy evaluation, its mean return is higher on four tasks than SNN-PPO (adapted) and on three than ANN-PPO. Two prominent contrasts are positive in every training pair: HalfCheetah AUC improves by 105.7%, and Swimmer final-policy return by 88.4%, relative to SNN-PPO (adapted). Performance varies across tasks; Appendix F reports all outcomes and configuration differences.

Comparison with SpikeGym. We reproduce SpikeGym with its reported one-layer architecture and native training schedule. At the same 32.768-million-step budget, MCrPPO raises mean AUC from 3118.4 to 5138.0 (+64.8%; Figure 5). Common 20-reset final-policy return rises from 5457.5 to 5934.8 (+8.7%). Both differences are positive in all three training pairs. The larger AUC gain places the main advantage in performance accumulated over training. This is a complete-system comparison, not an isolated AMC effect; Appendix F.5 details schedule differences and remaining metrics.

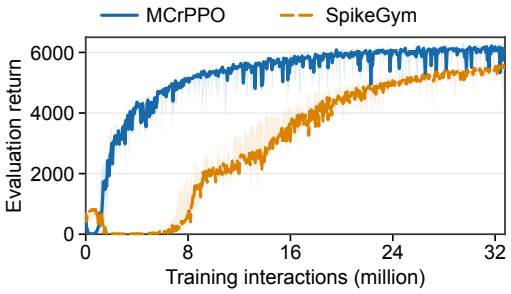

Figure 5: Ant-v4 system comparison over 32.768 million environment steps: medians and ranges across three paired training seeds. Native schedules are retained; environment, interaction budget, monitoring grid and reset seeds are matched (Appendix F.5).

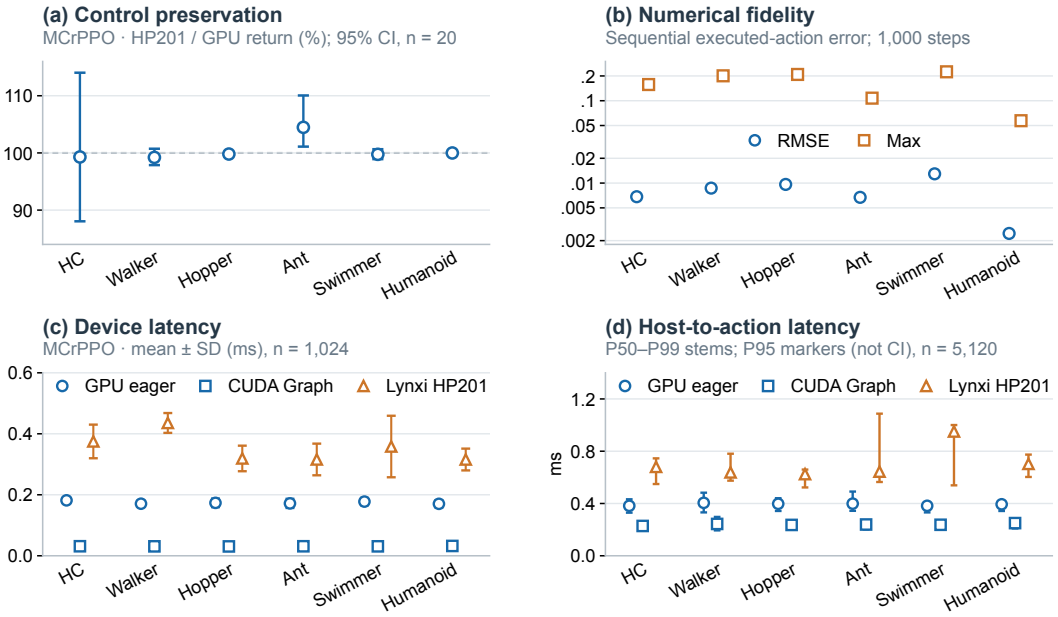

Figure 6: GPU/Lynxi HP201 evaluation of selected MCrPPO Actors. (a) HP201/GPU mean-return ratios and 95% paired-reset bootstrap intervals over 20 resets, conditional on each selected Actor. (b) Executed-action RMSE and maximum error over 1000 shared observations with independent state carry. (c) Device latency, mean ± SD over 1024 calls: GPU eager, CUDA Graph and HP201 SDK reset plus execute. (d) Host-to-action P50/P95/P99 over 5120 calls per task/mode; percentile stems are not confidence intervals. Both timing scopes exclude environment simulation (Appendix G).

4.4 ACTOR-ONLY NEUROMORPHIC EXECUTION

AMC enriches Critic access during training without adding an Actor inference branch. Closed-loop evaluation in the simulated tasks compares 18 fixed Actors on GPU and a Lynxi HP201 neuromorphic accelerator card: six each for MCrPPO, SNN-PPO (adapted) and converted ANN–SNN. For each method and task, we evaluate the five final models on monitoring resets M1–M5 and select the highest mean return; converted models are evaluated after conversion. The selected models are then tested on 20 paired new reset seeds (Appendix G.1).

Figure 7 depicts Actor-only execution with the learned shared weights retained. For the six selected MCrPPO Actors, HP201/GPU mean-return ratios span 99.2%–104.5%, with uncertainty conditional on each model (Figure 6). On 1000 shared observations with independent state carry, executed-action RMSE is 0.00245–0.01295 in task action coordinates.
HP201 has higher latency in this pipeline: mean host-to-action time is 0.534–0.608 ms versus 0.200–0.215 ms with CUDA Graph. Timing includes normalization, transfers and action mapping, but excludes environment stepping. These results demonstrate Actor-only execution on real accelerator hardware, not a latency advantage. Detailed timing protocols and energy measurements are in Appendix G.

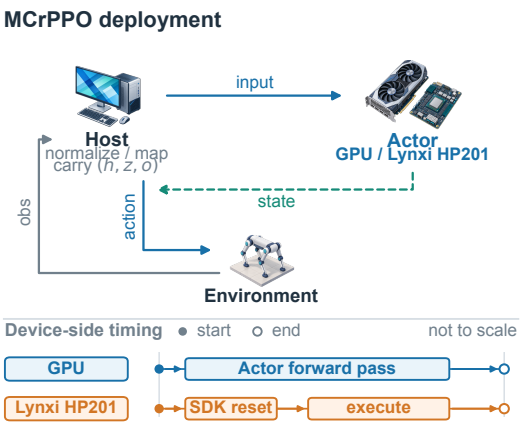

Figure 7: Critic-free MCrPPO execution. The host carries (h, z, o), normalizes observations and maps actions; GPU or Lynxi HP201 advances the Actor state. Timed operations include HP201 SDK reset, distinct from episode reset: logical recurrent state passes explicitly between calls. Not to scale; protocol in Appendix G.1.

5 DISCUSSION AND CONCLUSION

A useful membrane readout need not imply a useful direct encoder gradient. The detached comparisons make this distinction actionable when evaluating new Critic branches. Measuring the added route's alignment with existing encoder gradients during training may help explain its task-dependent effects.

Conclusion. We introduced AMC and a reusable stateful spiking Actor–Critic substrate spanning PPO, SAC and TD3. Relative to event-only controls, the complete AMC pathway achieves higher mean learning-curve AUC in 17 of 18 task–algorithm settings and higher mean final-policy return in 16 settings under the common 20-reset evaluation. The substrate separates common spiking computation from rule-specific state handling and encoder-update ownership, retaining each learner's objectives and enabling controlled comparisons. System comparisons demonstrate competitive MCrPPO control under the evaluated protocols, and GPU/Lynxi HP201 evaluation verifies Actor-only execution after Critic removal.

More broadly, shared spiking computation need not impose identical state access for action selection and Critic learning. AMC uses this freedom to enrich training while retaining an event-mediated Actor path; the detached controls show why membrane access and its direct encoder-gradient route should be specified separately. Together, the effective training interface and reusable substrate provide a concrete basis for developing stateful full-SNN control methods with explicit learning interfaces and Actor-only deployment.

Reproducibility Statement

Appendix A gives assumptions and derivations, and Appendix B specifies the three learning-rule implementations. Appendices C–F document evaluation protocols, configurations, training-run inclusion and seed-level outcomes. Appendix G specifies Actor export and hardware evaluation. Anonymized code and configurations for the AMC E/D/A experiments under PPO, SAC and TD3 are provided as supplementary material.

AI Use Disclosure

We used AI tools to assist with language polishing, manuscript formatting, literature search and review, the preparation of figures and decorative visual elements, the preparation and organization of the anonymized supplementary code package, and checking mathematical derivations. The authors reviewed the AI-assisted materials and take responsibility for the final manuscript and supplementary code.

References

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron Courville, and Marc Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In *Advances in Neural Information Processing Systems*, volume 34, pp. 29304–29320. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/f514cec81cb148559cf475e7426eed5e-Paper.pdf.

Marcin Andrychowicz, Anton Raichuk, Piotr Stańczyk, Manu Orsini, Sertan Girgin, Raphael Marinier, Léonard Hussenot, Matthieu Geist, Olivier Pietquin, Marcin Michalski, Sylvain Gelly, and Olivier Bachem. What matters in on-policy reinforcement learning? a large-scale empirical study. *arXiv preprint arXiv:2006.05990*, 2020. URL https://arxiv.org/abs/2006.05990.

Andrea Baisero and Christopher Amato. Unbiased asymmetric reinforcement learning under partial observability. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, pp. 44–52, 2022. URL https://www.ifaamas.org/Proceedings/aamas2022/pdfs/p44.pdf.

Ding Chen, Peixi Peng, Tiejun Huang, and Yonghong Tian. Deep reinforcement learning with spiking q-learning. *arXiv preprint arXiv:2201.09754*, 2022. doi: 10.48550/arXiv.2201.09754. URL https://arxiv.org/abs/2201.09754.

Ding Chen, Peixi Peng, Tiejun Huang, and Yonghong Tian. Fully spiking actor network with intra-layer connections for reinforcement learning. *arXiv preprint arXiv:2401.05444*, 2024. doi: 10.48550/arXiv.2401.05444. URL https://arxiv.org/abs/2401.05444.

Karl W Cobbe, Jacob Hilton, Oleg Klimov, and John Schulman. Phasic policy gradient. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 2020–2027. PMLR, 2021. URL https://proceedings.mlr.press/v139/cobbe21a.html.

Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Firdaus Janoos, Larry Rudolph, and Aleksander Madry. Implementation matters in deep policy gradients: A case study on PPO and TRPO. *arXiv preprint arXiv:2005.12729*, 2020. URL https://arxiv.org/abs/2005.12729.

Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pp. 1587–1596, 2018. URL https://proceedings.mlr.press/v80/fujimoto18a.html.

Samuel Garcin, Trevor McInroe, Pablo Samuel Castro, Christopher G. Lucas, David Abel, Prakash Panangaden, and Stefano V. Albrecht. Studying the interplay between the actor and critic representations in reinforcement learning. In *International Conference on Learning Representations*, volume 2025, pp. 35590–35616, 2025.

URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/57da66da25d0ce77e0129b246f358851-Abstract-Conference.html.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pp. 1861–1870, 2018a. URL https://proceedings.mlr.press/v80/haarnoja18b.html.

Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, and Sergey Levine. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018b. doi: 10.48550/arXiv.1812.05905. URL https://arxiv.org/abs/1812.05905.

Justus Huebotter, Pablo Lanillos, Marcel van Gerven, and Serge Thill. Spiking neural networks for continuous control via end-to-end model-based learning, 2025. URL https://arxiv.org/abs/2509.05356.

Jessica Hunter, Md Maruf Hossain Shuvo, and Krishna Roy. Spiking neural networks for continuous control: Neuromorphic reinforcement learning in conventional computing. *arXiv preprint arXiv:2608.22729*, 2026. doi: 10.48550/arXiv.2608.22729. URL https://arxiv.org/abs/2608.22729.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2015. URL https://arxiv.org/abs/1412.6980.

Skander Moalla, Andrea Miele, Daniil Pyatko, Razvan Pascanu, and Caglar Gulcehre. No representation, no trust: Connecting representation, collapse, and trust issues in PPO. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2405.00662.

Emre O. Neftci, Hesham Mostafa, and Friedemann Zenke. Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks. *IEEE Signal Processing Magazine*, 36(6):51–63, 2019. doi: 10.1109/MSP.2019.2931595.

Tianwei Ni, Michel Ma, Benjamin Eysenbach, and Pierre-Luc Bacon. When do transformers shine in RL? decoupling memory from credit assignment. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://arxiv.org/abs/2307.03864.

Hyunseok Oh and Youngki Lee. Sign gradient descent-based neuronal dynamics: ANN-to-SNN conversion beyond ReLU network. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 38562–38598. PMLR, 2024. URL https://proceedings.mlr.press/v235/oh24b.html.

Lerrel Pinto, Marcin Andrychowicz, Peter Welinder, Wojciech Zaremba, and Pieter Abbeel. Asymmetric actor critic for image-based robot learning. In *Proceedings of Robotics: Science and Systems*, 2018. doi: 10.15607/RSS.2018.XIV.008. URL https://www.roboticsproceedings.org/rss14/p08.html.

Marco Pleines, Matthias Pallasch, Frank Zimmer, and Mike Preuss. Generalization, mayhems and limits in recurrent proximal policy optimization. *arXiv preprint arXiv:2205.11104*, 2022. URL https://arxiv.org/abs/2205.11104.

Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021. URL https://jmlr.org/papers/v22/20-1364.html.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015. URL https://arxiv.org/abs/1506.02438.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. URL https://arxiv.org/abs/1707.06347.

Guangzhi Tang, Neelesh Kumar, Raymond Yoo, and Konstantinos Michmizos. Deep reinforcement learning with population-coded spiking neural network for continuous control. In *Proceedings of the 2020 Conference on Robot Learning*, volume 155, pp. 2016–2029, 2021. URL https://proceedings.mlr.press/v155/tang21a.html.

Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033, 2012. doi: 10.1109/IROS.2012.6386109.

Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U. Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Markus Krimmel, Arjun KG, Rodrigo Perez-Vicente, Andrea Pierré, Sander Schulhoff, Jun Jet Tai, Hannah Tan, and Omar G. Younis. Gymnasium: A standard interface for reinforcement learning environments. *arXiv preprint arXiv:2407.17032*, 2024. URL https://arxiv.org/abs/2407.17032.

Korneel Van den Berghe, Stein Stroobants, Vijay Janapa Reddi, and Guido de Croon. Adaptive surrogate gradients for sequential reinforcement learning in spiking neural networks. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-4946. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/d94b46ec30adee2bbb134f813fc9dde0-Abstract-Conference.html.

Hado van Hasselt, Arthur Guez, Matteo Hessel, Volodymyr Mnih, and David Silver. Learning values across many orders of magnitude. In *Advances in Neural Information Processing Systems*, volume 29, pp. 4287–4295, 2016. URL https://papers.nips.cc/paper_files/paper/2016/hash/5227b6aaf294f5f027273aebf16015f2-Abstract.html.

Zijie Xu, Tong Bu, Zecheng Hao, Jianhao Ding, and Zhaofei Yu. Proxy target: Bridging the gap between discrete spiking neural networks and continuous control. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-5316. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/e95eb5206c867be843fbc14bbfe8c10e-Abstract-Conference.html.

Zijie Xu, Xinyu Shi, Yiting Dong, Zihan Huang, and Zhaofei Yu. CaRe-BN: Precise moving statistics for stabilizing spiking neural networks in reinforcement learning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=AaZVrbElhC.

Zhihan Yang and Hai Nguyen. Recurrent off-policy baselines for memory-based continuous control, 2021. URL https://arxiv.org/abs/2110.12628. Deep RL Workshop, NeurIPS 2021.

Denis Yarats, Amy Zhang, Ilya Kostrikov, Brandon Amos, Joelle Pineau, and Rob Fergus. Improving sample efficiency in model-free reinforcement learning from images. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(12):10674–10681, 2021. doi: 10.1609/aaai.v35i12.17276. URL https://ojs.aaai.org/index.php/AAAI/article/view/17276.

Luca Zanatta, Francesco Barchi, Simone Manoni, Silvia Tolu, Andrea Bartolini, and Andrea Acquaviva. Exploring spiking neural networks for deep reinforcement learning in robotic tasks. *Scientific Reports*, 14:30648, 2024. doi: 10.1038/s41598-024-77779-8.

Jiahong Zhang, Sijun Shen, Dehua Wu, Yifan Lin, Xuechen Xia, Xu Chu, Youhui Zhang, and Guoqi Li. SpikingNav: Robust embodied navigation with spiking neural policies, 2026. URL https://arxiv.org/abs/2608.05078v1.

Liwen Zhang, Heng Deng, and Guanghui Sun. Fully spiking actor-critic neural network for robotic manipulation, 2025. URL https://arxiv.org/abs/2508.12038.

Huaicheng Zhou, Zifeng Zhuang, and Donglin Wang. Stay hungry, keep learning: Sustainable plasticity for deep reinforcement learning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 79644–79672, 2025. URL https://proceedings.mlr.press/v267/zhou25am.html.

APPENDIX GUIDE

Appendix A derives the local gradient identities, initialization properties and Actor-only execution boundary. Appendix B specifies the PPO, SAC and TD3 implementations, including recurrent-state handling and parameter-update ownership. Appendix C defines the training runs, evaluation layers, metrics and statistical units. Appendices D and E report the controlled PPO and off-policy results; Appendix F reports the separate system comparisons. Appendix G describes Actor export, numerical fidelity, closed-loop hardware evaluation and execution costs. Appendix H reports surrogate sensitivity and summarizes the limits of the evidence. Appendix I provides extended related work. Readers interested in empirical results can begin with Appendix C and follow the references to the relevant outcome tables.

A MATHEMATICAL PROPERTIES AND CONTROL IDENTIFICATION

A.1 ACCESS AND DIRECT CREDIT UNDER A GENERAL SURROGATE

Let $s(h)$ denote the chosen surrogate derivative of the event map. For a fixed normalized input, stored pre-transition recurrent state, and set of parameters, the affine encoder Jacobian is independent of the event surrogate. A and D have identical values in the hard-threshold forward pass and therefore the same value adjoint and gradient of the readout weights. They differ only in whether the readout layer receives h_t or $\mathrm{sg}(h_t)$. Subtracting the two encoder gradients yields Eq. 3 for any s, with $D_t = \mathrm{diag}(s(h_t))$ in Eq. 2. This identity describes the surrogate gradient at the current step, not a derivative through the preceding recurrent trajectory.

For the tested rectangle $s(h) = \mathbf{1}[0 < h < 1]$, the event contribution can vanish exactly while the direct adjoint with respect to membrane potential remains nonzero. With a noncompact surrogate, the event contribution need not vanish. Nevertheless, the additional readout still gives the value function access to membrane potential and provides the same local gradient path when evaluated on the same recorded transitions and parameters. Neither observation establishes a behavioral advantage across surrogate families; that counterfactual is specified separately in Appendix H.1.

The implementation evaluates both open inequalities directly in membrane potential coordinates. Although $|h - .5| < .5$ is equivalent over the reals, centering can round tiny positive FP32 values to the boundary; that expression is not the implemented test.

A.2 FUNCTION-PRESERVING INITIALIZATION AND LOCAL RANK

At zero initialization, the attached and event-only value-head inputs coincide. The first value optimizer step can learn W_h, but the direct encoder credit is zero before that update. Once the readout weights become nonzero, this gradient path can become active in later optimizer steps. For a fixed-parameter batch B, the isolated direct encoder-weight component is

$$G_{\mathrm{mem}} = W_h^{\mathsf{T}} \left(\sum_{t \in B} d_{u,t} \widehat{x}_t^{\mathsf{T}} \right), \qquad \mathrm{rank}(G_{\mathrm{mem}}) \le 1. \tag{4}$$

Structure of AMC encoder credit within one optimizer step. Under TBPTT-1, write $h_t = c_t + W_x \widehat{x}_t + b_x$, where $c_t = .2 h_t^- \odot (1 - z_t^-)$; the pre-transition recurrent state and normalized input are detached. For the scalar linear readout of membrane potential $W_h h_t$, the isolated direct branch contributes $\partial L_V / \partial h_t = W_h^{\mathsf{T}} d_{u,t}$, independently of the event-mediated paths. The affine map therefore gives

$$\nabla_{W_x} L_V|_{\mathrm{mem}} = \sum_{t \in B} (W_h^{\mathsf{T}} d_{u,t}) \widehat{x}_t^{\mathsf{T}} = W_h^{\mathsf{T}} \left(\sum_{t \in B} d_{u,t} \widehat{x}_t^{\mathsf{T}} \right).$$

The final expression is a single outer product, which proves Eq. 4. Here $d_{u,t}$ is the adjoint of the implemented clipped normalized value objective, including its coefficient and logical-batch reduction, not an unclipped TD residual. Four microbatch means are weighted by $1/4$ and accumulated while the parameters remain fixed before the Adam step, so they share the same left factor. The bias contribution is $W_h^{\mathsf{T}} \sum_t d_{u,t}$; augmenting the input by one gives the same factorization for $[W_x \mid b_x]$.

At initialization, $W_h = 0$, so the isolated direct encoder component is zero throughout the first optimizer step, even when that step learns nonzero readout weights. The rank bound is an exact-arithmetic property of this component under the stated affine-encoder, scalar-readout and TBPTT-1 assumptions; it does not assert exact numerical rank in FP32. It applies neither to the total encoder gradient nor to the Adam displacement, since the former includes other paths and coordinatewise optimizer preconditioning need not preserve rank. Across optimizer steps, the readout weights, adjoints and parameters change, so their accumulated contributions need not share a common left factor. Nonlinear encoders, multiple membrane outputs and gradients through earlier recurrent states require separate derivations.

A.3 Structural Actor-only boundary

Proposition 1 (Structural Actor invariance) *Fix Actor parameters, observation RMS, task bounds, initial (h, z, o), and the observation/reset/noise sequence. Removing the Critic and its AMC row preserves the Actor states and actions at every decision in exact arithmetic.*

The proposition follows by induction from Eqs. 5–8, which do not read u or any value parameter. It is a graph statement for fixed weights, not equivalence of separately trained policies or finite-precision hardware implementations.

A.4 Raw gradient versus optimizer displacement

Attributing an Adam update also requires a common optimizer state. Let $U(g; \Omega) = \theta^+(g; \Omega) - \theta$ denote the actual Adam parameter increment, restricted to the encoder, from identical parameters, moments, step, parameter-group and LR state Ω. We evaluate it on copies without updating the learner. A well-defined counterfactual parameter increment attributable to the direct membrane potential readout is

$$\delta^{\mathrm{cf}}_{\mathrm{mem}} = U(g_{\mathrm{total}}; \Omega) - U(g_{\mathrm{total}} - g_{\mathrm{mem}}; \Omega).$$

It is not generally $U(g_{\mathrm{mem}}; \Omega)$, nor an additive share of Adam. For MCrPPO, g_{total} combines policy, event-value and direct membrane potential readout paths; g_{mem} is the isolated direct component in Eq. 4, including the corresponding encoder-bias component when that is optimized. Cosine similarity is undefined when either norm is zero. A small W_h norm alone does not bound the routed gradient without its adjoint and encoder Jacobian. Adam preconditioning is applied only after the raw gradient has been computed.

B Learning-Rule Implementations

PPO, SAC, TD3, GAE and Adam provide the learning rules (Schulman et al., 2017; Haarnoja et al., 2018a; Fujimoto et al., 2018; Schulman et al., 2015; Kingma & Ba, 2015). Because recurrent-state, implementation and normalization choices can affect the resulting algorithms, we specify them explicitly below (Pleines et al., 2022; Engstrom et al., 2020; Andrychowicz et al., 2020; Xu et al., 2026). Prior work on representation dynamics and continued-learning interventions also motivates our separate measurements of late-stage performance (Moalla et al., 2024; Zhou et al., 2025).

B.1 MCrPPO

B.1.1 Architecture, action distribution and reset semantics

MCrPPO uses a single affine observation encoder and persistent hard-LIF states of width $H = 512$. Normalized observations $\widehat{x}_t = N^x_p(x_t)$ use observation statistics that remain fixed during rollout collection and optimization. The episode mask $m_t \in \{0, 1\}$ equals zero at reset and one otherwise:

$$(h^-_t, z^-_t, o^-_t, u^-_t) = m_t(h_{t-1}, z_{t-1}, o_{t-1}, u_{t-1}). \tag{5}$$

Each environment decision updates the states as

$$h_t = .2h^-_t \odot (1 - z^-_t) + W_x\widehat{x}_t + b_x, \qquad z_t = \mathbf{1}[h_t > .5], \tag{6}$$

$$o_t = .2o^-_t + W_o z_t + b_o, \qquad \mu_t = o_t. \tag{7}$$

The forward pass uses a strict threshold; backpropagation uses the open rectangular surrogate gradient $\widetilde{\partial z_t}/\partial h_t = \mathbf{1}[0 < h_t < 1]$. The Actor uses a learned state-independent $\rho = \mathrm{clip}(\rho^{\mathrm{raw}}, -20, 2)$ and exact task bounds:

$$\ell_t \sim \mathcal{N}(\mu_t, \mathrm{diag}(e^{2\rho})), \qquad a_t = c_a + s_a \odot \tanh \ell_t, \tag{8}$$

where $c_a = (a_{\max} + a_{\min})/2$ and $s_a = (a_{\max} - a_{\min})/2$. There is no post-tanh clipping. During deterministic evaluation, $\ell_t = \mu_t$. The Critic updates its membrane potential as

$$u_t = .2u_t^- + W_V z_t + b_V + W_h h_t, \qquad V_t = \sigma_p^V u_t + \mu_p^V. \tag{9}$$

Here (μ_p^V, σ_p^V) is the collection-time value reference. The FP32 row $W_h \in \mathbb{R}^{1\times 512}$ adds 512 trainable parameters but does not introduce another recurrent state.

Each rollout/update cycle collects $256 \times 16 = 4096$ samples and then applies eight joint Adam optimizer steps, each accumulated from four quarter-weighted microbatches. Observation RMS remains fixed through these optimizer steps and is updated only afterward; the KL/progress rule sets the learning rate only for the next cycle. These numerical choices define the MCrPPO implementation of PPO, not the AMC interface itself.

Table B1: MCrPPO architecture, optimization and numerical settings.

Quantity	Setting
Architecture	full-SNN Actor–Critic; one affine observation encoder; 512 hard-LIF units; training states (h, z, o, u); Actor states (h, z, o)
LIF dynamics	leak .2; strict threshold .5; open rectangular surrogate support $(0, 1)$
Action law	state-independent $\rho^{\mathrm{raw}} \in \mathbb{R}^A$, initialized to zero, clamped $[-20, 2]$; exact affine–tanh task bounds; no post-tanh clipping
AMC	attached FP32 $W_h \in \mathbb{R}^{1\times 512}$, initialized to zero; updated only through the value objective; normalized value clipping radius .2
Rollout/update schedule	$256 \times 16 = 4096$ samples; four epochs; two minibatches/epoch; four 512-sample microbatches/optimizer step; eight optimizer steps/update cycle
PPO/GAE	policy clip .2; reward scale .01; $\gamma = .99$; $\lambda = .95$; no entropy term; no global gradient-norm clipping is applied
Adam	one joint optimizer; LR 3×10^{-4} initially; $(\beta_1, \beta_2) = (.9, .999)$; $\epsilon = 10^{-8}$; weight decay 0; AMSGrad and maximization disabled
Execution	FP32; highest matmul precision; TF32 and autocast disabled
Scheduler	$\tau = .008$; learning-rate range $[10^{-5}, 3 \times 10^{-4}]$; factor 1.5; Eq. 22; applied to the next update cycle only
RMS	mean/variance FP32, count FP64; initial $(0, 1, 10^{-4})$; population variance; variance floor 10^{-8}; normalized clip $[-10, 10]$
Evaluation	deterministic mean action; fixed reset seeds; raw environment reward units; no retrospective checkpoint selection

Initialization and RNG. Before constructing the model, the implementation seeds Python, NumPy, CPU, and CUDA. The encoder, action, and event-value affine layers use the runtime's `Linear.reset_parameters`: Kaiming-uniform with $a = \sqrt{5}$, equivalent to $U[-1/\sqrt{\text{fan_in}}, 1/\sqrt{\text{fan_in}}]$ for weights and the corresponding bias bound. The external log-scale parameter, W_h, and all recurrent states are initialized to zero. A dedicated action generator uses the training-seed offset by one, and the permutation generator uses the offset by two; neither generator is shared with evaluation.

Reset and boundary semantics. For every recorded transition, the collector stores its already masked and detached pre-transition recurrent state. After an environment step, `done` denotes termination OR truncation. Either outcome clears all four neural states and the previous-action buffer before the next autoreset observation. A truncation target is computed from the true final observation and the post-transition recurrent state without applying a reset, while its GAE trace is cut. If termination and truncation coincide, termination suppresses bootstrapping. At rollout/update-cycle boundaries, recurrent states are preserved for environment instances whose episodes remain active.

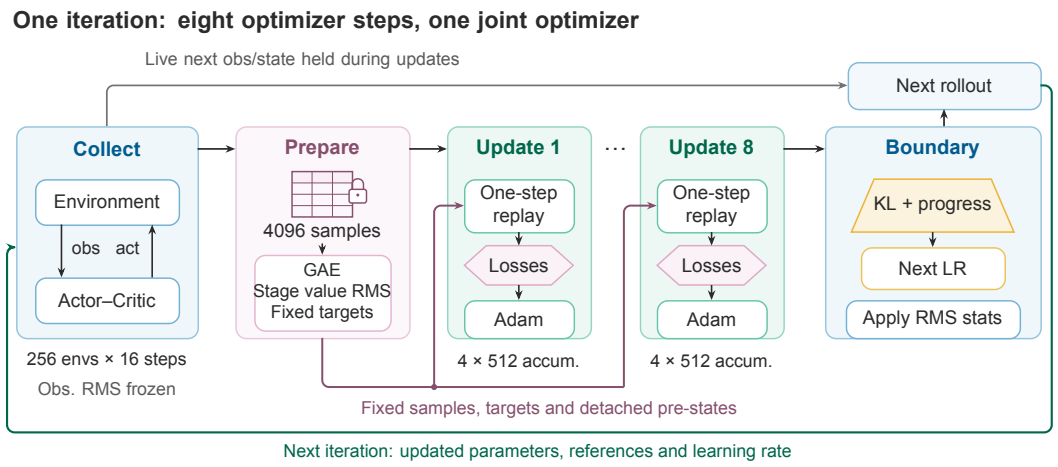

Figure 8: MCrPPO rollout and update sequence. A batch of 4096 recorded transitions provides fixed targets for eight joint Adam optimizer steps, each accumulated from four microbatches. The learning rate and RMS statistics are updated after optimization, while the recurrent collector state persists across rollout/update cycles. Architecture: Figure 1.

B.1.2 GAE, NORMALIZED TARGETS AND VALUE CLIPPING

Figure 8 shows the PPO-specific training sequence that accompanies Algorithm 1. Its eight optimizer steps and the timing of its RMS updates differ from the replay-based and delayed-update procedures for SAC and TD3 in Figures 9 and 10.

Environment reward enters training once as $\widetilde{r}_t = .01 r_t^{\text{env}}$. Let d_t and q_t denote natural termination and truncation. With $b_t = 1 - d_t$ and $c_t = 1 - (d_t \vee q_t)$,

$$\delta_t = \widetilde{r}_t + \gamma b_t V_{t+1}^{b,*} - V_t^b, \qquad A_t = \delta_t + \gamma\lambda c_t A_{t+1}, \qquad G_t = A_t + V_t^b, \tag{10}$$

where $\gamma = .99$ and $\lambda = .95$. Natural termination disables bootstrapping. Here V_t^b and $V_{t+1}^{b,*}$ are recorded behavior-policy values decoded with the value reference used during collection; the latter is either the ordinary next value or the truncation bootstrap from the true final observation. Truncation bootstraps from the true final observation using the post-transition recurrent state, but breaks the GAE trace before the autoreset episode. A nonterminal rollout boundary bootstraps normally.

All 4096 advantages are standardized once over the complete recorded rollout:

$$\widehat{A}_t = \frac{A_t - \text{mean}(A)}{\max(\text{std}_{\text{pop}}(A), 10^{-8})}. \tag{11}$$

The reductions are FP64 and the result is cast back to the training dtype. Raw GAE returns remain unchanged.

After GAE, the returns update a copy of the value RMS to obtain $(\widetilde{\mu}_{p+1}^V, \widetilde{\sigma}_{p+1}^V)$, which is used for target normalization but is not committed until optimization is complete. The fixed target and clipped value objective are

$$y_t = \text{clip}\left(\frac{G_t - \widetilde{\mu}_{p+1}^V}{\widetilde{\sigma}_{p+1}^V}, -10, 10\right), \tag{12}$$

$$u_t^{\text{clip}} = u_{b,t} + \text{clip}(u_{\theta,t} - u_{b,t}, -.2, .2), \tag{13}$$

$$L_V = \tfrac{1}{2}\mathbb{E}_t \max\{(u_{\theta,t} - y_t)^2, (u_t^{\text{clip}} - y_t)^2\}. \tag{14}$$

Each recorded sample stores the unclamped behavior output $u_{b,t}$. Thus at $\theta = \theta_b$ the clipping displacement $u_{\theta,t} - u_{b,t}$ is zero even though the updated RMS changes the target coordinate. Committing this RMS after optimization need not preserve decoded raw values; this is not a PopArt invariance construction (van Hasselt et al., 2016).

Algorithm 1 One MCrPPO rollout/update cycle.

Require: θ_p, one joint Adam optimizer, recurrent collector states, N_p^x, N_p^V, LR η_p

1: Freeze N_p^x, N_p^V; collect and store 4096 transitions
2: Compute Eq. 10; normalize advantages once; stage value RMS
3: Freeze $y_t, u_{b,t}$, minibatch permutations, and all recorded inputs
4: **for** $j = 1, \ldots, 8$ optimizer steps **do**
5: compute $\kappa_{p,j}$ before the step
6: accumulate four quarter-weighted 512-sample microbatches
7: verify finite loss/gradients; apply one joint Adam step
8: **end for**
9: Compute $S(\theta_p), S(\theta_{p+1}), K_p^{\text{post}}$ on recorded samples
10: Set η_{p+1} by Eq. 22
11: Merge raw observations into RMS once; commit both RMS states at boundary
12: Preserve recurrent states for active environments and save a durable checkpoint at the update-cycle boundary

The policy compares current and behavior diagonal Gaussians at the same stored latent. Its full clipped objective is

$$\log r_t^\pi(\theta) = \log \mathcal{N}(\ell_t; \mu_{\theta,t}, \operatorname{diag}(e^{2\rho_\theta})) - \log \mathcal{N}(\ell_t; \mu_{b,t}, \operatorname{diag}(e^{2\rho_b})), \tag{15}$$

$$L_\pi = -\mathbb{E}_t \min\{r_t^\pi \widehat{A}_t, \operatorname{clip}(r_t^\pi, .8, 1.2)\widehat{A}_t\}. \tag{16}$$

Here the recorded behavior latents, distribution parameters, values and boundary flags remain fixed. Identical invertible affine–tanh transforms cancel their density Jacobian when stored latents, bounds and input conventions match, without post-tanh clipping. The factored FP32 implementation avoids inverse-tanh reconstruction and subtraction of large squares (Appendix B.1.4). There is no entropy or auxiliary loss. In each of four epochs, the recorded samples are permuted into two minibatches; four quarter-weighted 512-sample microbatches accumulate before each Adam step. The procedure therefore performs eight optimizer steps, not 32.

For post-update diagnostics, both distributions are recomputed from the same recorded inputs and stored pre-transition recurrent states. The exact anchored conditional quantity is

$$K_p^{\text{post}} = \mathbb{E}_t\left[\mathrm{KL}\left(\mathcal{N}_b(\ell_t \mid \widehat{x}_t, \xi_{b,t}^-) \,\|\, \mathcal{N}_{\theta_{p+1}}(\ell_t \mid \widehat{x}_t, \xi_{b,t}^-)\right)\right]. \tag{17}$$

Its direction is behavior-to-candidate. It is exact for the conditional diagonal Gaussians and their common invertible action transform, but is not a closed-loop trajectory KL or a recurrent-chain divergence.

Table B2: Parameter and state ownership in MCrPPO. P/V denote policy/value gradients; the deployment column identifies Actor execution requirements.

Group or state	P	V	Role	Deploy path
Shared observation encoder f_η	Yes	Yes	Parameter	Yes
Action row W_o, b_o and raw log scale	Yes	No	Parameter	Yes
Event value row W_V, b_V	No	Yes	Parameter	No
AMC value row W_h (1×512)	No	Yes	Parameter	No
Hidden membrane/event/action membrane	–	–	Runtime state	Yes (3)
Value membrane	–	–	Runtime state	No (fourth)
Committed observation RMS (frozen for each rollout/update cycle), schema, bounds	–	–	Statistic/meta	Yes
Value RMS, Adam, scheduler/telemetry	–	–	Training state	No

B.1.3 RUNNING STATISTICS AND UPDATE BOUNDARIES

Let stored population moments be (m, v, n) and a batch of n_b raw samples have population moments (m_b, v_b). With $\Delta = m_b - m$,

$$n' = n + n_b, \qquad m' = m + \Delta \frac{n_b}{n'}, \tag{18}$$

$$v' = \frac{nv + n_b v_b + \Delta^2 n n_b / n'}{n'}, \qquad v' \leftarrow \max(v', 10^{-8}). \tag{19}$$

We normalize as $\mathrm{clip}((x - m)/\sqrt{v}, -10, 10)$; the floor is applied to the stored variance rather than added inside the square root. Observation RMS remains fixed during collection and all eight optimizer steps. The 4096 raw observations are then merged once, increasing its count by 4096. Value RMS is copied and updated once from 4096 raw returns before optimization, retained as the target reference, and committed only after all eight optimizer steps. Values used during collection and for truncation bootstrapping are decoded with the preceding value-RMS coordinates. We use neither a sample-variance correction nor per-minibatch RMS updates.

B.1.4 CONDITIONAL POLICY DIAGNOSTICS AND NUMERICAL EVALUATION

For diagonal Gaussian log-scales ρ, the algebraic identity for the stored latent is

$$\log r^\pi = -\frac{1}{2} \sum_j (q_{\theta,j}^2 - q_{b,j}^2) - \sum_j (\rho_{\theta,j} - \rho_{b,j}), \qquad q = (\ell - \mu) e^{-\rho}.$$

The implementation avoids subtracting these potentially large standardized squares directly. For each active action coordinate, it evaluates the difference in factored form using $\mathrm{expm1}$:

$$d_j = q_{b,j} - q_{\theta,j} = (\ell_j - \mu_{b,j}) e^{-\rho_{\theta,j}} \,\mathrm{expm1}(\rho_{\theta,j} - \rho_{b,j}) + (\mu_{\theta,j} - \mu_{b,j}) e^{-\rho_{\theta,j}},$$

$$\log r_j^\pi = \rho_{b,j} - \rho_{\theta,j} + \tfrac{1}{2} d_j (q_{b,j} + q_{\theta,j}),$$

The coordinate-wise log ratios are then summed. This reduces cancellation when the behavior and candidate distributions are close in FP32. The affine–tanh Jacobian cancels only when both policies use the same stored latent, invertible transform, task bounds and input convention, with no post-transform clipping. The exact anchored conditional diagnostic, summed over the active action coordinates (all coordinates in the present tasks), has closed form

$$\mathrm{KL}(\mathcal{N}_b \| \mathcal{N}_\theta) = \frac{1}{2} \sum_j \left[2(\rho_{\theta,j} - \rho_{b,j}) + \frac{e^{2\rho_{b,j}} + (\mu_{b,j} - \mu_{\theta,j})^2}{e^{2\rho_{\theta,j}}} - 1 \right].$$

Both means are produced from the same normalized input and stored pre-transition recurrent state. This quantity does not imply a teacher-forced chain or a closed-loop trajectory KL.

For one recorded sample, $s_j = \mathbf{1}[0 < h_j < 1]$ and $d_u = \partial L_V / \partial u$ give

$$\frac{\widetilde{\partial} L_V}{\partial h_j} = d_u (W_{V,j} s_j + W_{h,j}).$$

The W_h term is the attached route for the current step. Because the event forward map is discontinuous, regression tests examine the explicit surrogate boundary, the attached/detached graph difference, the clipping branch, and the resulting conditional divergence after an optimizer step. An ordinary finite-difference check is not appropriate for the spikes themselves.

B.1.5 LEARNING-RATE SCHEDULING AND COMPLETE UPDATE CYCLE

Let $j \in \{1, \ldots, 8\}$ index optimizer steps, and let $\Delta_{j,t} = \log r_{j,t}^\pi$ before step j on its logical batch. The statistic evaluated before that optimizer step is

$$\kappa_{p,j} = \mathbb{E}_{t \in B_j}[e^{\Delta_{j,t}} - 1 - \Delta_{j,t}], \qquad \bar{\kappa}_p = \tfrac{1}{8} \sum_{j=1}^{8} \kappa_{p,j}. \tag{20}$$

We also define the clipped-surrogate score on the recorded batch $S(\theta) = \mathbb{E}_t[\min\{r_t^\pi \widehat{A}_t, \mathrm{clip}(r_t^\pi, .8, 1.2)\widehat{A}_t\}]$ and

$$c_p = \mathrm{clip}\left(\frac{S(\theta_{p+1}) - S(\theta_p)}{.2\,\mathbb{E}_t|\widehat{A}_t|}, 0, 1\right), \quad P_p = \max\{\bar{\kappa}_p, K_p^{\mathrm{post}}\}. \tag{21}$$

If the denominator is zero, $c_p = 0$. With $\tau = .008$, $\eta_{\min} = 10^{-5}$, $\eta_{\max} = 3 \times 10^{-4}$, and $g = 1.5$, the exact learning-rate rule for the next cycle is

$$\eta_{p+1} = \begin{cases} \max(\eta_{\min}, \eta_p/g), & \bar{\kappa}_p > 2\tau, \\ \min\left(\eta_{\max}, \eta_p g^{c_p[1-P_p/(.5\tau)]_+}\right), & \text{otherwise.} \end{cases} \tag{22}$$

The upper condition is strict, so equality at .016 does not reduce the LR. There is no scheduler warm-up, hysteresis, patience state, or per-actuator input. The recorded counts of reductions and increases are diagnostic only and do not affect the scheduler. The rule is evaluated only after the eighth optimizer step and sets the learning rate for rollout/update cycle $p + 1$; it does not accept, reject, replay, or roll back cycle p. It does not read raw reward, episodic return, velocity, fresh evaluation, value-fit residual, or RMS displacement; $\widehat{A}$ appears only inside the fixed recorded-batch policy-surrogate progress score and its normalization.

Nonfinite losses or gradients at any optimizer step stop training. A rollout/update-cycle boundary marks both the fixed collection/update schedule and the durable checkpoint boundary; the scheduler has no post-update acceptance or rollback mechanism.

B.2 SAC AND TD3

The SAC and TD3 implementations share the spiking policy computation and AMC access boundary, but retain distinct Actor objectives and target-network rules. Q losses update the shared encoder. Before an Actor-head update, the online recurrent state is reconstructed with the updated encoder; encoder and Q parameters are then held fixed while the surrogate Q-to-action gradient remains active. The following subsections specify replay-state reconstruction, parameter ownership, objectives and update order for the implementations evaluated here. Controlled results appear in Appendix E.

B.2.1 SHARED SPIKING COMPUTATION AND REPLAY RECONSTRUCTION

In both adaptations, the Actor accesses the shared representation through spikes. Two zero-initialized membrane potential readout rows introduce 1024 trainable parameters in A relative to E. Before its scalar readout, each Q head applies 256 private threshold units jointly to the shared event and the unit-coordinate action; these units are not recurrent. The resulting Q function can therefore represent state–action interactions rather than only $c(s) + W_a a$. Gradients from Q to the action use the surrogate derivative of these hard thresholds, not their ordinary derivative.

The replay buffer stores raw observations and executed unit-coordinate actions, but not neural states. The online and target networks independently reconstruct their states from 16 preceding observations while respecting episode resets and missing prefixes, and then use the true next observation to compute the targets. Termination prevents bootstrapping; time-limit truncation permits bootstrapping before the collector is reset. This finite burn-in only approximates the whole-episode recurrent state. Table E12 reports the completed local diagnostic across the 16 scheduled checkpoints, including the endpoint. At each checkpoint, it compares 16- and 32-step reconstruction for the current model. It measures local online Actor state and action sensitivity, not the effect of burn-in length during training. No controlled study has compared policies trained with the two burn-in lengths, so training-time burn-in sensitivity remains untested and separate from the present return comparisons. Observation moments are estimated from only the first 10,000 random-action training interactions and are then held fixed. Evaluation before this warm-up uses identity normalization rather than statistics from future data.

B.2.2 PARAMETER OWNERSHIP AND UPDATE ORDER

Let η, ϕ, ψ denote encoder, Actor-head and twin-Q parameters. During each Q-function update, Adam updates η and ψ using L_Q. The online recurrent states are then reconstructed with the updated encoder. The subsequent Actor update is

$$\eta^+ = \mathrm{Adam}_\eta(\nabla_\eta L_Q), \qquad \phi^+ = \mathrm{Adam}_\phi\big(\nabla_\phi L_\pi; \mathrm{sg}(h_{\eta^+}, z_{\eta^+})\big).$$

Each Adam update also depends on its parameters and optimizer moments before the step. Both encoder and Q parameters remain fixed during the Actor update; the gradient path from Q through the action to the Actor head remains intact. No encoder optimizer step is taken during this phase, so the Q optimizer's existing moments cannot change the encoder. Separate optimizers update the heads and, for SAC, the temperature. This applies to E, D and A. Detaching shared features during Actor learning has precedent in off-policy representation learning (Yarats et al., 2021); here that ownership rule separates Q-driven spiking representation learning from Actor-head optimization.

At 1 Mi, this procedure performs 129,822 Q-head and encoder optimizer steps. SAC performs 129,822 Actor-head optimizer steps; TD3 performs 64,911. Earlier versions also updated the encoder with Adam during Actor optimization and therefore used a different encoder update frequency; those runs are excluded from the matched training runs. In TD3, Q-only steps can still change the policy through the shared encoder even when the Actor head and target networks are not updated.

Table B3: SAC and TD3 learning rules on the shared spiking architecture.

Component	Spiking SAC adaptation	Spiking TD3 adaptation
Actor	Recurrent tanh Gaussian; event-dependent log scale clamped $[-5, 2]$	Recurrent deterministic tanh mean; no density or temperature
Q objective	Half the sum of the two mean-squared Bellman errors	Sum of the two mean-squared Bellman errors
Actor objective	$\mathbb{E}[\alpha \log \pi(a) - \min_i Q_i]$	$-\mathbb{E}[Q_1]$
Targets	Target encoder and twin Q; no target Actor	Target encoder, Actor and twin Q
Update order	Q, rebuilt Actor, temperature, then target updates	Q each learner step; rebuilt Actor and targets every second Q write
Encoder updates	Q phase only; Actor phase freezes encoder/Q and updates only Actor head	
Target action	Online stochastic Actor; entropy-adjusted target	Target Actor; noise SD .2, clipped to $[-.5, .5]$, then action clipped to $[-1, 1]$
Shared settings	Encoder width 512; reward scale 1; $\gamma = .99$; target blend .005; fixed LR 3×10^{-4}	
Replay	Eight lanes; batch 256; 16-decision no-gradient burn-in; current-transition TBPTT-1	

SAC entropy is defined in unit-action coordinates, and the affine transformation to the task bounds is applied when the action is executed. TD3 adds exploration noise in that coordinate rather than to its recurrent action membrane potential. Its deterministic action displacement is not a Gaussian KL.

Detailed off-policy equations. The following equations specify the two implementations evaluated here. Let d_x and d_a be the observation and unit-action dimensions, $H = 512$, $K = 256$, and let $b_t \in \{0, 1\}$ mark the first observation after an episode boundary. With $k_t = 1 - b_t$, $\bar{h}_{t-1} = k_t h_{t-1}$, $\bar{z}_{t-1} = k_t z_{t-1}$ and $\bar{o}_{t-1} = k_t o_{t-1}$, both adaptations use

$$h_t = .2\bar{h}_{t-1} \odot (1 - \bar{z}_{t-1}) + W_x \widehat{x}_t + b_x, \qquad z_t = \mathbf{1}[h_t > .5], \tag{23}$$

$$o_t = .2\bar{o}_{t-1} + W_o z_t + b_o. \tag{24}$$

The bars on pre-transition states in these two equations indicate episode masking. Bars on parameters and next-state features in the target equations below denote target networks; the two uses are distinct. Here $W_x \in \mathbb{R}^{H \times d_x}$ and $W_o \in \mathbb{R}^{d_a \times H}$. The spike backward rule is $\widetilde{\partial z_t}/\partial h_t = \mathbf{1}[0 < h_t < 1]$. The frozen observation reference is formed from the first 10,000 training observations and applies $\widehat{x} = \mathrm{clip}((x - \mu_x)/\sqrt{M_{2,x}/n_x + 10^{-6}}, -5, 5)$, where $M_{2,x}$ is the accumulated sum of squared deviations and $n_x = 10{,}000$. For each sampled transition, replay reconstructs the preceding recurrent state from zero using at most 16 earlier raw observations without tracking gradients. The reconstruction respects the stored reset and valid-prefix masks, and gradients are computed only for the current transition. This finite burn-in approximates the unrecorded whole-episode recurrent state.

B.2.3 SAC OBJECTIVES, TEMPERATURE AND TARGETS

For SAC, we use the automatic-temperature variant (Haarnoja et al., 2018a;b):

$$\rho_t = \mathrm{clip}(W_\rho z_t + b_\rho, -5, 2), \tag{25}$$

$$\ell_t = o_t + e^{\rho_t} \odot \epsilon_t, \qquad \epsilon_t \sim \mathcal{N}(0, I), \tag{26}$$

$$a_t = \tanh \ell_t. \tag{27}$$

Its exact unit-action log density is

$$\log \pi_t = -\frac{1}{2}\sum_{j=1}^{d_a} \epsilon_{t,j}^2 - \sum_{j=1}^{d_a} \rho_{t,j} - \frac{d_a}{2}\log(2\pi) - 2\sum_{j=1}^{d_a}\{\log 2 - \ell_{t,j} - \mathrm{softplus}(-2\ell_{t,j})\}. \tag{28}$$

The temperature is $\alpha = e^\beta$ with α initialized to .2. TD3 instead uses the deterministic unit action $a_t = \tanh o_t$ and has no ρ, density or temperature. In both cases the environment action is the fixed affine map of $a_t \in [-1, 1]^{d_a}$ to the task bounds; replay stores the executed unit-coordinate action.

Each private Q head $i \in \{1, 2\}$ is evaluated without recurrent state:

$$v_i(z, a) = U_i[z; a] + b_i, \qquad s_i(z, a) = \mathbf{1}[v_i(z, a) > .5], \tag{29}$$

$$Q_i(h, z, a) = w_i^\mathsf{T} s_i(z, a) + c_i + W_{h,i} h, \tag{30}$$

where $U_i \in \mathbb{R}^{K\times(H+d_a)}$, $b_i, w_i \in \mathbb{R}^K$, $c_i \in \mathbb{R}$, and $W_{h,i} \in \mathbb{R}^{1\times H}$. The private threshold uses the same open rectangular surrogate support. Arm E omits both $W_{h,i}$ rows; arm A adds them with zero initialization after all shared parameters have been initialized identically. Arm D retains these trainable rows but replaces h by $\mathrm{sg}(h)$ only when computing the online Q loss. The event gradient, learning of the added readout weights, target evaluation, and Actor-update rules remain unchanged. Thus the interface

$$C_i(h, z, a) = C_{i,\mathrm{event}}(z, a) + W_{h,i} h \tag{31}$$

is exactly the scalar Q_i here. In MCrPPO, the analogous action-independent quantity is the current input to the recurrent normalized value-head membrane potential before adding the recurrent carry and decoding with the value RMS; Eq. 31 must not be identified with MCrPPO's decoded V_t.

For a replay minibatch of size $B = 256$, let a_t^b denote the stored replay action, $Q_{i,t} = Q_i(h_t, z_t, a_t^b)$, and let d_t^{term} be the stored termination flag. Truncation does not set this flag: the target bootstraps from the stored true next observation, while both types of episode boundary reset the subsequent collected episode. SAC first reconstructs an online prefix and advances the online model through $(\widehat{x}_t, \widehat{x}_{t+1})$ to obtain the next online Actor state $(h_{t+1}^\pi, z_{t+1}^\pi, o_{t+1}^\pi)$. The target encoder is reconstructed and advanced separately to obtain $(\bar{h}_{t+1}, \bar{z}_{t+1})$. Under stop-gradient,

$$a'_{t,\mathrm{SAC}} \sim \pi_\phi(\cdot \mid h_{t+1}^\pi, z_{t+1}^\pi, o_{t+1}^\pi), \tag{32}$$

$$y_t^{\mathrm{SAC}} = r_t + .99(1 - d_t^{\mathrm{term}})\left[\min_i Q_{\bar{\eta},\bar{\psi}_i}(\bar{h}_{t+1}, \bar{z}_{t+1}, a'_{t,\mathrm{SAC}}) - \alpha \log \pi_\phi(a'_{t,\mathrm{SAC}})\right], \tag{33}$$

$$L_Q^{\mathrm{SAC}} = \frac{1}{2B}\sum_{t=1}^{B}\sum_{i=1}^{2}(Q_{i,t} - y_t^{\mathrm{SAC}})^2. \tag{34}$$

There is no target Actor: the sampled next action and its log density come from the online Actor, whereas the next-state features and Q heads used in the target are produced by target networks. The Q-function update takes one Adam step for the encoder and one for the twin Q heads. Their gradients are then cleared, and the current online state (h_t^+, z_t^+, o_t^+) is reconstructed with the updated encoder. Draw $a_t^{\pi,+}$ from the SAC Actor using the reconstructed state and Eq. 27, and denote its log density by $\log \pi_t^{\pi,+}$. With the encoder and Q parameters fixed and the Q-function state inputs detached, the Actor-head objective is

$$L_\pi^{\mathrm{SAC}} = \frac{1}{B}\sum_{t=1}^{B}\left[\mathrm{sg}(\alpha)\log \pi_t^{\pi,+} - \min_i Q_{\mathrm{sg}(\psi_i)}(\mathrm{sg}(h_t^+), \mathrm{sg}(z_t^+), a_t^{\pi,+})\right]. \tag{35}$$

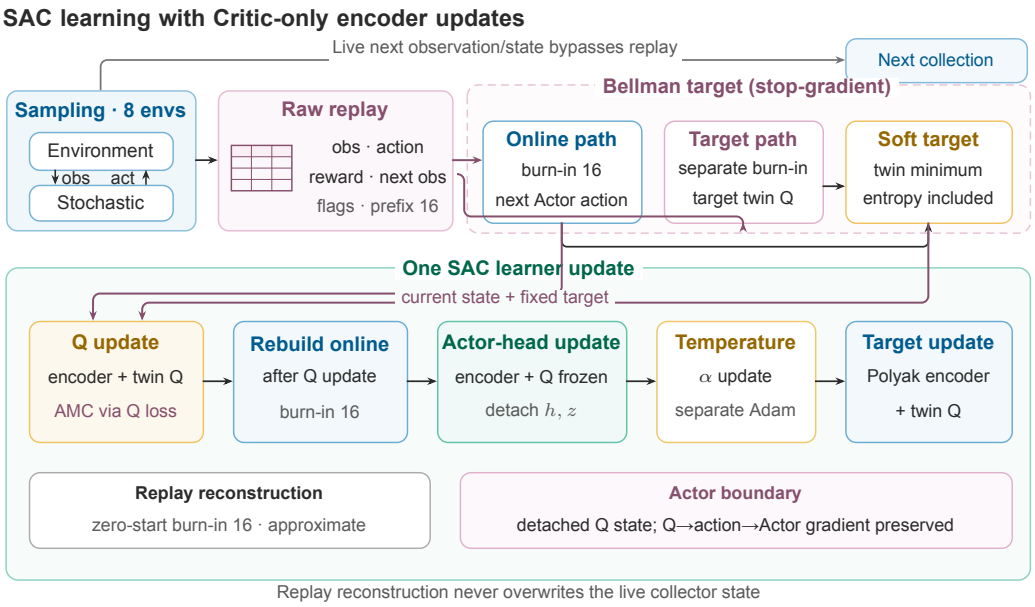

Figure 9: SAC update sequence. The online and target networks reconstruct their recurrent states separately to compute the Bellman target. Q-function optimization updates the encoder and twin Q heads. The Actor then uses features reconstructed with the updated encoder while the encoder and Q parameters remain fixed. Temperature and target-network updates follow; the collector state is kept separate from the replay-based reconstruction.

Only the Actor-head parameters are updated. The subsequent temperature objective is $L_\alpha = -B^{-1}\sum_t \beta[\mathrm{sg}(\log \pi_t^{\pi,+}) - d_a]$. Finally, each target parameter $\bar{\vartheta} \in \{\bar{\eta}, \bar{\psi}_1, \bar{\psi}_2\}$, but no Actor parameter, receives $\bar{\vartheta} \leftarrow (1 - .005)\bar{\vartheta} + .005\vartheta$.

The central SAC update boundary lies between Q-function and Actor optimization. The Q update changes the shared encoder. The Actor uses features reconstructed with that encoder while the encoder and Q parameters remain fixed and the recurrent-state inputs are detached. The retained Q-to-action derivative trains the Actor head.

B.2.4 TD3 OBJECTIVES, SMOOTHING AND DELAYED HEADS

TD3 independently reconstructs the state of the complete target model—encoder, deterministic Actor and twin Q heads—from its prefix, and then advances the model through the current and true next observations. Target smoothing is applied only to the action passed to the Q function, not to the recurrent target action membrane potential:

$$\tilde{a}'_{t,\mathrm{TD3}} = \mathrm{clip}(\tanh \bar{o}_{t+1} + \mathrm{clip}(.2\epsilon_t, -.5, .5), -1, 1)\,, \qquad \epsilon_t \sim \mathcal{N}(0, I), \tag{36}$$

$$y_t^{\mathrm{TD3}} = r_t + .99(1 - d_t^{\mathrm{term}}) \min_i Q_{\bar{\eta},\bar{\psi}_i}(\bar{h}_{t+1}, \bar{z}_{t+1}, \tilde{a}'_{t,\mathrm{TD3}}), \tag{37}$$

$$L_Q^{\mathrm{TD3}} = \frac{1}{B}\sum_{t=1}^{B}\sum_{i=1}^{2}(Q_{i,t} - y_t^{\mathrm{TD3}})^2. \tag{38}$$

Every training step applies one Adam optimizer step to the encoder and one to the twin Q heads. After every second Q-function update, the online state (h_t^+, z_t^+, o_t^+) is reconstructed with the updated encoder. Defining $a_t^{\pi,+} = \tanh o_t^+$, with the encoder and Q parameters fixed, only the deterministic Actor head minimizes

$$L_\pi^{\mathrm{TD3}} = -\frac{1}{B}\sum_{t=1}^{B} Q_{1,\mathrm{sg}(\psi_1)}(\mathrm{sg}(h_t^+), \mathrm{sg}(z_t^+), a_t^{\pi,+}). \tag{39}$$

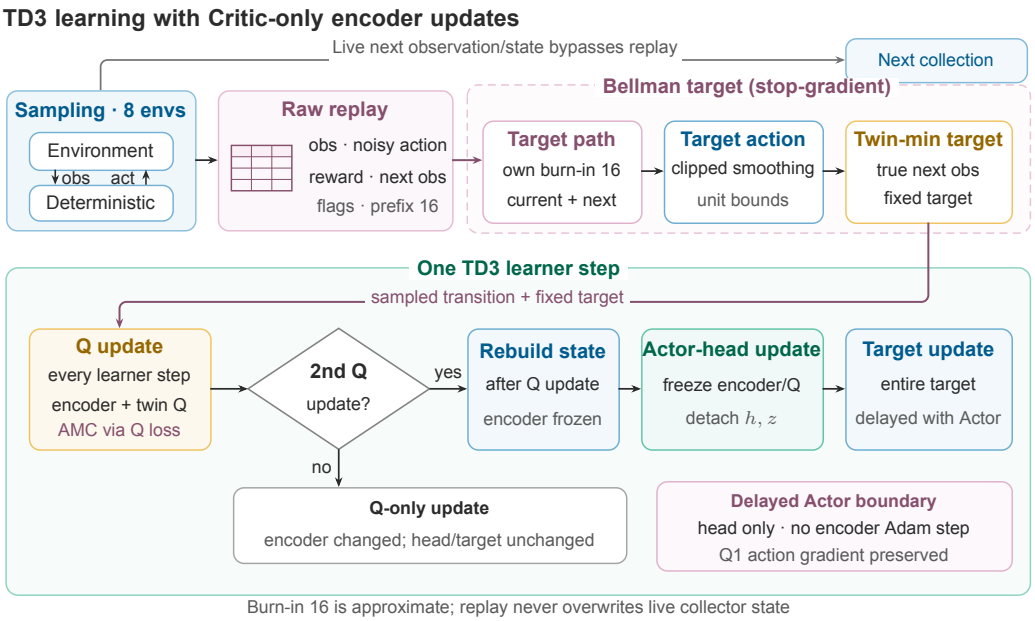

Figure 10: TD3 update sequence. Every Q-function optimizer step updates the encoder and twin Q heads against a smoothed-action target. Every second step also reconstructs the online recurrent state, updates the Actor head while keeping the encoder and Q parameters fixed, and updates the target networks. The shared representation continues to change during the intervening Q-only steps.

The complete target model—encoder, Actor and both Q heads—then receives one Polyak update with coefficient .005. On the intervening Q-only steps, the Actor head and target parameters do not change, but the online policy representation can change because the Q-function update changes the encoder.

Delayed updates apply to the Actor head and target networks. Each intervening Q-function optimizer step still changes the online encoder and can therefore change the complete policy's features. On eligible steps, the recurrent state is reconstructed before the Actor update with detached state inputs.

B.2.5 AMC GRADIENTS UNDER TWIN-Q LEARNING

The different loss reductions give

$$d_{i,t}^{\mathrm{SAC}} = \frac{Q_{i,t} - y_t^{\mathrm{SAC}}}{B}, \qquad d_{i,t}^{\mathrm{TD3}} = \frac{2(Q_{i,t} - y_t^{\mathrm{TD3}})}{B},$$
$$\widetilde{\nabla}_\eta L_Q\Big|_{\mathrm{mem}} = \sum_t J_{h_t,\eta}^{\mathsf{T}} \sum_i W_{h,i}^{\mathsf{T}} d_{i,t}. \tag{40}$$

At zero initialization, the final expression is zero during the first Q-function optimizer step even when that step learns $W_{h,i}$.

At fixed h, the state-only term satisfies $\partial(W_{h,i}h)/\partial a = 0$. The Actor receives a surrogate Q-to-action derivative through the private event/action branch. The membrane potential readout term can affect SAC's selected minimum head and can affect subsequent Actor gradients through Q-function and encoder updates; it does not itself provide a direct gradient path to the action. These equations define the algorithm-specific training procedures. Recorded update counts, local burn-in diagnostics, and running costs are reported separately in Appendix E, and analytical network/forward counts in Table B4. Whole-episode replay fidelity and isolated full-training cost require additional evidence.

B.2.6 COMPLETE LEARNER UPDATES

The E/D/A intervention is applied at the online Q readout in both algorithms: E omits $W_{h,i}$, D uses the trainable $W_{h,i}\,\mathrm{sg}(h_t)$ only in the online Q loss, and A uses the trainable $W_{h,i}h_t$. The added rows start at zero. Target evaluation and Actor learning use the ordinary readout in D and A; these conditions share optimizer ownership.

Algorithm 2 One spiking SAC learner update.

Require: Online encoder, Actor and twin Q parameters (η, ϕ, ψ); target encoder and Q parameters $(\bar{\eta}, \bar{\psi})$; temperature $\alpha = e^{\beta}$; replay and frozen observation normalization

1: Sample $B = 256$ raw transitions with up to 16 preceding observations, reset masks and valid-prefix masks; burn in from zero without gradients, then track gradients only through the current transition
2: Without gradients, reconstruct separate online Actor and target encoder prefixes; advance through the current and true next observations
3: Sample the next unit action from the *online* Actor; compute y^{SAC} by Eq. 33 using target twin Q (termination blocks bootstrap, truncation does not)
4: Compute online current-state twin Q loss by Eq. 34, with the E/D/A readout rule above
5: Update η and ψ with one step of their respective Adam optimizers; clear their gradients
6: Reconstruct the online current state with updated η from the same prefix; hold η, ψ fixed and detach Q state inputs
7: Sample an online Actor action and log density; update only ϕ by Eq. 35, retaining the Q-to-action derivative
8: Update β using $-B^{-1}\sum_t \beta[\mathrm{sg}(\log \pi_t^{\pi,+}) - d_a]$
9: Polyak-update only $(\bar{\eta}, \bar{\psi})$ toward (η, ψ) with coefficient .005; keep collector recurrence separate from replay states

Algorithm 3 One spiking TD3 learner update.

Require: Online encoder, deterministic Actor and twin Q parameters (η, ϕ, ψ); full target model $(\bar{\eta}, \bar{\phi}, \bar{\psi})$; Q-update count j; replay and frozen observation normalization

1: Sample $B = 256$ raw transitions with up to 16 preceding observations, reset masks and valid-prefix masks; burn in from zero without gradients, then track gradients only through the current transition
2: Without gradients, independently reconstruct the full target model prefix and advance through the current and true next observations
3: Add target-action Gaussian noise with SD .2, clip noise to $[-.5, .5]$, then clip the unit action to $[-1, 1]$ as in Eq. 36; compute Eq. 37
4: Reconstruct the online current state and compute twin Q loss by Eq. 38, with the E/D/A readout rule above
5: Update η and ψ with one step of their respective Adam optimizers; increment j
6: **if** j is even **then**
7: Reconstruct the online current state with updated η from the same prefix; hold η, ψ fixed and detach Q state inputs
8: Update only ϕ by Eq. 39 through Q_1, retaining the Q-to-action derivative
9: Polyak-update the full target model $(\bar{\eta}, \bar{\phi}, \bar{\psi})$ with coefficient .005
10: **end if**
11: Keep collector recurrence separate from replay states

B.3 Parameter and forward-operation accounting

Table B4 reports network tensors and dense linear operations. Parameters include biases and network log scales, excluding target copies, optimizer state and SAC's scalar temperature. A forward call includes one encoder/Actor pass and the value head (PPO), or one query of both Q heads (SAC/TD3). MACs exclude bias additions, elementwise recurrence, normalization, backward, Adam and simulation; they describe dense linear arithmetic rather than event-chip execution. The additional readout costs 512 parameters and MACs per PPO value call, or 1024 per twin-Q query; it adds none to Actor execution. D/A counts coincide, but replay burn-in, target copies, gradient computation and optimizer steps prevent us from inferring equal training cost from forward counts. Recorded running costs are reported in Appendix E.6.

Table B4: Analytical parameter counts and dense linear MACs per forward call. D and A have equal counts; Actor-only counts are unchanged by AMC.

		Parameters			Linear MACs per forward call		
Rule	Task	E	D/A	Actor	E	D/A	Actor
PPO	HC	12,813	13,325	12,300	12,288	12,800	11,776
	Walker	12,813	13,325	12,300	12,288	12,800	11,776
	Hopper	8,199	8,711	7,686	7,680	8,192	7,168
	Ant	58,897	59,409	58,384	58,368	58,880	57,856
	Swimmer	6,149	6,661	5,636	5,632	6,144	5,120
	Humanoid	187,939	188,451	187,426	187,392	187,904	186,880
SAC	HC	281,614	282,638	15,372	280,576	281,600	14,848
	Walker	281,614	282,638	15,372	280,576	281,600	14,848
	Hopper	273,928	274,952	9,222	272,896	273,920	8,704
	Ant	329,746	330,770	62,480	328,704	329,728	61,952
	Swimmer	270,854	271,878	6,660	269,824	270,848	6,144
	Humanoid	468,004	469,028	196,130	466,944	467,968	195,584
TD3	HC	278,536	279,560	12,294	277,504	278,528	11,776
	Walker	278,536	279,560	12,294	277,504	278,528	11,776
	Hopper	272,389	273,413	7,683	271,360	272,384	7,168
	Ant	325,642	326,666	58,376	324,608	325,632	57,856
	Swimmer	269,828	270,852	5,634	268,800	269,824	5,120
	Humanoid	459,283	460,307	187,409	458,240	459,264	186,880

C Training Runs and Evaluation Protocols

The experiments comprise a PPO intervention study, a longer-budget system comparison, an off-policy extension and Actor-only execution. The main six-task studies use MuJoCo-v5 through Gymnasium with 1000-step episode limits; Appendix F.5 gives the isolated Ant-v4 protocol.

C.1 Training runs and pairing

The PPO study uses three paired training seeds per condition and task at $10\,\mathrm{Mi}$. Its E/A runs use a common source configuration and training seeds S1–S3, and the D extension completes the same design: 18 E, 18 D and 18 A runs, or three paired triplets per task. All training runs are complete; the E/A runs were verified for source identity, pairing and parameter-update ownership. Six S2 D runs resumed from their own saved checkpoints; the remaining D runs started from zero, and every trajectory belongs to one training run. Final-checkpoint summaries cover all 54 runs and 270 monitoring episodes. The off-policy study uses three paired training seeds per task and rule at $1\,\mathrm{Mi}$. The independent fixed-LR SNN-PPO comparison is a separate six-task, one-seed $1\,\mathrm{Mi}$ feasibility reference rather than the E baseline.

The longer-budget comparison uses the common training seeds defined in Appendix F, which distinguishes these paired runs from the original available-case summaries. New-reset evaluations use saved final weights without retrospective checkpoint selection. Runs are indexed by implementation,

simulator version, configuration, seed, interaction budget, evaluation mode, reset set, monitoring grid and final Actor.

Table C1: Evidence index. Counts in the first five rows sum to the 276 fixed models in the E1–E20 evaluation. Hardware and reset-only rows reuse selected or existing models and are not additional training runs.

Study (models)	Budget; evaluation resets	Question
PPO E/D/A (54)	10 Mi; M1–M5, E1–E20	Complete pathway and added credit
SAC/TD3 E/D/A (108)	1 Mi; M1–M5, E1–E20	Across-rule intervention
v5 systems (90)	20 Mi; AUC first 10 Mi; M1–M5, E1–E20	System-level control performance
Surrogate extension (18)	10 Mi; M1–M5, E1–E20	Single-seed profile sensitivity
Ant-v4 systems (6)	32.768M steps; monitoring five, E1–E20	Native-schedule system comparison
Hardware (18 selected Actors)	No retraining; H1–H20	Fixed-Actor backend transfer
PPO reset-only (same 54)	No retraining; original evaluator: M1–M5, R1–R20	Reset-set sensitivity

C.2 Metrics and statistical units

For task e, training seed s and checkpoint t_k, define $J_{e,s}(t_k) = M^{-1}\sum_{m=1}^{M} R_{e,s,m}(t_k)$, with $M = 5$ fixed reset-seed episodes on the monitoring grid. The main score is

$$\mathrm{AUC}_{a:b}^{e,s} = \frac{1}{b-a}\sum_{k:\, a\le t_k<t_{k+1}\le b} \frac{J_{e,s}(t_k)+J_{e,s}(t_{k+1})}{2}(t_{k+1}-t_k).$$

The AUC is normalized by the integration-window length $b-a$, not necessarily the training budget. PPO E/D/A uses $[0, 10]$ Mi and SAC/TD3 uses $[0, 1]$ Mi. The 20 Mi system comparison reports $\mathrm{AUC}_{0:10}$ over $[0, 10]$ Mi, not over the full training budget; Ant-v4 uses its full 32.768-million-step window (Appendix F.5). This area under the learning curve (AUC) is evaluated on the complete registered grid with both endpoints present; missing evaluations are not imputed. The grid advances every 65,536 environment steps from zero, yielding 161 checkpoints through 10 Mi and 321 through 20 Mi. Maintenance AUC uses the 161 checkpoints on $[10, 20]$ Mi. $\mathrm{Tail}_{18:20}$ is the arithmetic mean of the 33 checkpoint means on the inclusive $[18, 20]$ Mi window; the last-four summary is a distinct metric. For the 10 Mi E/D/A study, $\mathrm{Tail}_{8:10}$ analogously averages the 33 scheduled checkpoint means on $[8, 10]$ Mi.

For the historical monitoring grid, we report R_{10}, R_{20}, $R_{20} - R_{10}$ and the stated fixed tails. These monitoring endpoints are distinct from the new common-reset fixed-policy outcomes defined below. $\mathrm{Best}_{0:20} = \max_{0\le k\le 320} J_{e,s}(t_k)$ is computed within each training seed before cross-seed aggregation; it measures the best evaluation score observed during training, not deployment-checkpoint selection. $\mathrm{Best}_{0:10}$ is also reported as a standalone best-checkpoint statistic and enters the secondary ratios $R_{10}/\mathrm{Best}_{0:10}$ and $\mathrm{Tail}_{8:10}/\mathrm{Best}_{0:10}$ when the denominator is positive. These ratios are otherwise undefined, and negative values are not clipped. Because endpoint/best ratios are unstable near zero or for negative returns, retention is interpreted primarily through changes in return or behavioral measures between fixed checkpoints. Raw returns are neither normalized by per-task maxima nor pooled across tasks. Table F5 separately averages task-level percentage changes relative to ANN-PPO as a descriptive system summary.

Episode-level behavioral measures. Each episode records official return, motion, control, contact, viability and other reward terms together with velocity, displacement, length, time-limit status and natural termination. A velocity integral and a raw root-position difference have different coordinate and time conventions and are reported as labeled. Natural termination is not a universal fall label, while reaching a full horizon does not itself demonstrate locomotion. Stable-gait behavior, fall rates and plateau success would require task-specific criteria beyond the present measurements.

Action saturation. For the Sat. (%) behavioral measure, each valid episode step contributes the fraction of active action coordinates beyond the evaluator's threshold; the episode value is 100 times

the average over its actual T steps. PPO uses its deterministic post-tanh action normalized to task bounds and counts $|a^{\text{norm}}_{t,j}| \geq .99$. SAC and TD3 use their deterministic post-tanh unit-coordinate actions before affine environment scaling and count $|a^{\text{unit}}_{t,j}| > .99$. All task action coordinates are active. Thus, for action dimension d_a, the per-episode statistic is $100(Td_a)^{-1}\sum_{t=1}^{T}\sum_{j=1}^{d_a}\mathbf{1}[|a_{t,j}| \triangleright .99]$, with $\triangleright$ interpreted as $\geq$ for PPO and $>$ for SAC/TD3. Each fixed model's 20 episode values are averaged before condition means and sample SDs are computed across training seeds. The equality convention differs across evaluators, so Sat. is interpreted within evaluator-matched contrasts.

Statistical units and displays. Evaluation episodes are nested within training runs. Each checkpoint first averages episodes within a training seed; tables then report the mean and sample SD across training seeds. Training replicates are anonymized as S1–S3 within each PPO, SAC/TD3 and Ant-v4 comparison, and S1–S5 within the system comparison. Monitoring reset seeds are M1–M5, the common final-policy reset seeds are E1–E20, the earlier reset-sensitivity seeds are R1–R20, hardware reset seeds are H1–H20, and local-probe reset seeds are L1–L5. Labels are local to their stated comparison or reset set. Pairing is within a learning rule; equal identifiers do not match PPO to SAC/TD3 or force identical trajectories after the interventions diverge.

Figure 2 shows pointwise A/E medians across training seeds with the observed minimum–maximum range. Table AUCs are arithmetic means of per-run integrals, not integrals of these median display curves. Its 17-point PPO display takes every tenth checkpoint, whereas all 17 SAC/TD3 checkpoints are shown; numerical summaries retain the complete 161/17-point E/D/A grids in Figure 16. Figure 3 shows D–E and A–D final-policy contrasts averaged over the complete E1–E20 common-reset set. The PPO and off-policy training horizons support within-rule comparisons, not a cross-algorithm sample-efficiency ranking. Figure 17 shows all 90 system-comparison trajectories across five common training seeds. These seed ranges are descriptive, not confidence intervals.

For an outcome Y measured for each training run, paired contrasts are computed before aggregation: $\Delta^{A-E}_{e,s} = Y^{A}_{e,s} - Y^{E}_{e,s}$ and $\Delta^{A-D}_{e,s} = Y^{A}_{e,s} - Y^{D}_{e,s}$; D–E is computed from its own matched training-run pairs. The primary PPO outcome is $\text{AUC}_{0:10}$, with R_{10} and task movement reported alongside it; R_{20} belongs to the separately matched 20 Mi extension. Positive and negative paired training-run differences remain visible.

The E/D/A studies support descriptive paired summaries rather than confirmatory significance or equivalence inference. Monitoring episodes are not additional training replicates. With three and five non-tied training-run pairs, the minimum two-sided exact sign-test probabilities are $2/2^3 = 0.25$ and $2/2^5 = 0.0625$, respectively; equivalence would additionally require a prespecified margin. Six training-run pairs merely make a sub-.05 exact value possible, not adequately powered.

C.3 Common 20-reset final-policy evaluation

We reevaluate 276 fixed final models on the same 20 new evaluation reset seeds, E1–E20, yielding 5520 episodes. These include 54 PPO E/D/A models trained for 10 Mi, 108 SAC/TD3 E/D/A models trained for 1 Mi, 90 system-comparison models trained for 20 Mi, 18 models from the single-seed surrogate study trained for 10 Mi, and six Ant-v4 models trained for 32.768 million steps. Each model retains its final weights, original training budget and observation-normalization state. Reevaluation performs no training and does not produce new learning curves, AUCs, best-checkpoint scores or late-window summaries.

Evaluation runs on an RTX 4090 with inference adapters matched to each model family. The adapters preserve action interpretation, normalization, recurrent-state handling and episode-reset semantics, but do not reproduce every original software and hardware environment bit for bit. Differences from historical monitoring therefore combine new evaluation reset seeds with a changed evaluation protocol; they cannot be attributed to reset variation alone. Comparisons are paired within the documented method, task, training-seed and budget combinations, and system comparisons retain their architecture and learner-protocol differences.

For model m and training seed s, let $\overline{R}^{20}_{m,s} = 20^{-1}\sum_{r=1}^{20} R_{m,s,r}$. Condition summaries report the mean and sample SD of these model-level averages across training seeds. Paired contrasts first compute $\Delta_s = \overline{R}^{20}_{L,s} - \overline{R}^{20}_{R,s}$ and then report their mean, sample SD and number of positive paired differences across training-seed pairs. The 20 episodes refine a fixed model's return estimate; they are not 20 additional training replicates and do not guarantee smaller between-run variation. Across the 18 E/D/A task–rule settings, A–E is positive in mean in 16, while all three training-seed differences are positive in six. The 20-reset estimate and the training-replicate sign count describe different sources of variation. The surrogate comparisons using one training seed therefore have no estimated SD across training seeds. Natural termination, length and movement are summarized with the same nesting, retaining their task-specific definitions.

The complete 20-reset result is the primary final-policy outcome throughout the main results. "Final-policy return" and the short plot label "Final policy (E1–E20)" denote this same metric. Quantities labeled *monitoring endpoint* remain the final checkpoint's original five-reset score. The earlier PPO R1–R20 study in Appendix D.4 remains a separate reset-sensitivity experiment. GPU/HP201 deployment uses H1–H20 and selected Actors rather than all paired training runs; neither its return ratios nor its costs are replaced by this GPU-only reevaluation. Complete condition means and paired contrasts appear in Tables D4–E6, F3, F9 and H2.

C.4 Evaluation layers and descriptive endpoint agreement

Table C2: Evaluation layers. Reset labels identify episode initializations, not additional training seeds. Only the PPO R1–R20 study retains the original evaluator while changing its reset set.

Reset set	Models and evaluator	Estimand
M1–M5	Training monitoring	Checkpoint curves, AUC, fixed tails and best observed checkpoint
E1–E20	276 fixed models; family-matched adapters	Common final-policy returns and paired contrasts
R1–R20	Same 54 PPO models; original evaluator	Reset-only sensitivity relative to reproduced M1–M5
H1–H20	18 selected Actors; GPU and HP201	Paired backend transfer and execution costs

The evaluation layers answer different questions. M1–M5 are the fixed monitoring reset seeds used to compute learning-curve AUC, fixed-tail and best-checkpoint statistics. E1–E20 provide the complete final-policy evaluation using algorithm-specific inference adapters. The separate PPO R1–R20 study preserves the original evaluator and first reproduces M1–M5, isolating the effect of changing the reset set. Monitoring and common-evaluation returns are not pooled into a 25-reset estimate. Hardware evaluation uses H1–H20 and selected Actors, and remains separate from these GPU-only final-policy comparisons.

The monitoring and common final-policy evaluations both yield 16/18 positive A–E mean differences and 11/18 positive A–D means; the D–E count changes from 15/18 to 18/18. In total, 47 of the 54 contrast means retain their sign. Equal counts do not imply that the same settings remain positive, and these contrasts are not independent observations: within a common evaluation set, $\Delta^{A-E} = \Delta^{A-D} + \Delta^{D-E}$.

The main final-policy tables and Figure 3 use all 20 common evaluation reset seeds. Earlier outcome-selected five-reset analyses do not contribute to these estimates.

D PPO Controls and Outcomes

The PPO controls distinguish learning-curve performance, monitoring endpoints, common-reset final-policy returns, late-window behavior and local mechanism checks. The direct edge has task-dependent effects across these outcomes.

D.1 GRAPH CONTROLS AND IMPLEMENTATION CHECKS

Section 2 defines the E/D/A interventions; Table D1 also includes an inert Z implementation control. A and E share an implementation revision. Implementation regression checks were performed to verify A/D initialization and first-update agreement, the later encoder-gradient identity, and parameter-update ownership across revisions. Eight-update Z/E checks covered common parameters, Adam state, normalization, learning rate and random-number generation. Matching conditions include normalization-commit rules, minibatch permutations, action-RNG roles and compute settings. The shared scheduler can yield different learning rates as training trajectories diverge. D–E differences are computed from matched training triplets, not from rounded condition means. Appendix C defines the comparison and descriptive statistics.

Table D1: Definitions of the AMC controls. The gradient row concerns only the added continuous-membrane branch; event-mediated value gradients remain in every condition. The forward-equivalence row assumes a general learned W_h, for which D and A share the readout. At initialization, where $W_h = 0$, E/Z/D/A forward values all coincide.

Property	E	Z	D	A
W_h parameter exists	No	Yes	Yes	Yes
Critic forward uses membrane information	No	No	Yes	Yes
W_h can learn from the value loss	No	Zero gradient	Yes	Yes
Direct membrane-value credit to encoder	No	No	No	**Yes**
Actor graph changes relative to E	No	No	No	No
Critic function class expands relative to E	No	No	Yes	Yes
Parameter count equals A	No	Yes	Yes	Yes
Forward equals A at matched parameters	No	No	**Yes**	–

D.2 LEARNING-CURVE AND MONITORING RESULTS

Mean A–E AUC is positive on every PPO task, but the paired AUC differences are positive for all three training seeds only on HalfCheetah and Swimmer. Figure 11 shows the paired training-curve and monitoring-endpoint contrasts.

Positive and negative monitoring cases. On the original monitoring grid, the control question is whether enabling the direct edge improves learning-curve AUC and the monitored endpoint together. Figure 12 preserves both outcomes for every A–D pair rather than selecting a favorable task or window. Walker and Ant have positive A–D AUC differences for all three training seeds, yet endpoint gains occur in only one Walker training run and no Ant training run. Swimmer is the strongest whole-pathway case, with positive A–E differences on all four outcomes in all three training seeds, while only two A–D training-run differences are positive. HalfCheetah instead has positive D–E AUC differences for every training seed but a negative mean A–D effect; Hopper and Humanoid remain mixed. The supported conclusion is therefore narrower than a universal direct-credit gain: pathway, edge and endpoint evidence can differ. These returns do not identify gradient conflict, outside-support harm or an optimizer failure; that would require the missing training-phase probes below.

Tables D2 and D3 give the complete training-curve summaries and historical monitoring endpoints. Best denotes the highest evaluation score observed during training; it selects neither training runs nor final policy checkpoints. The A–D points above are a linked view of the same records, not another experiment.

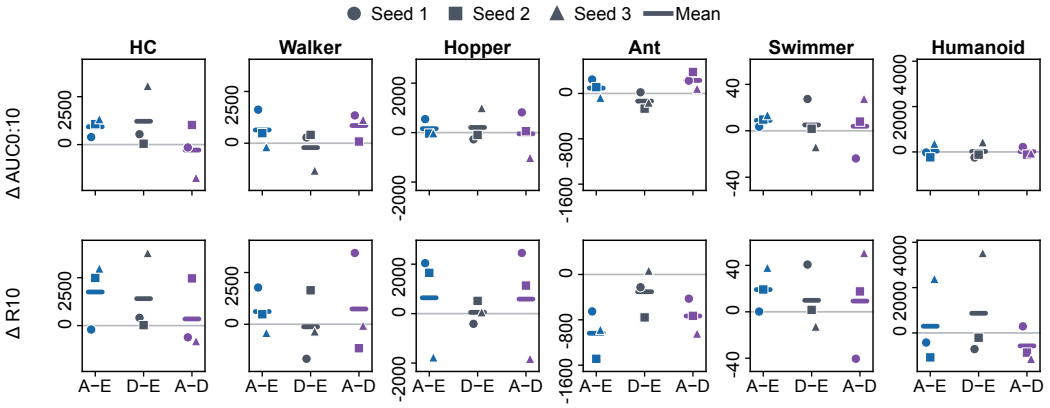

Figure 11: PPO learning-curve AUC and monitoring-endpoint contrasts at 10 Mi. Points are the three paired differences across training seeds and ticks are means; task columns use independent return scales shared by their two metric panels. A–E measures the whole pathway, D–E the detached membrane readout, and A–D the direct edge at matched forward function class.

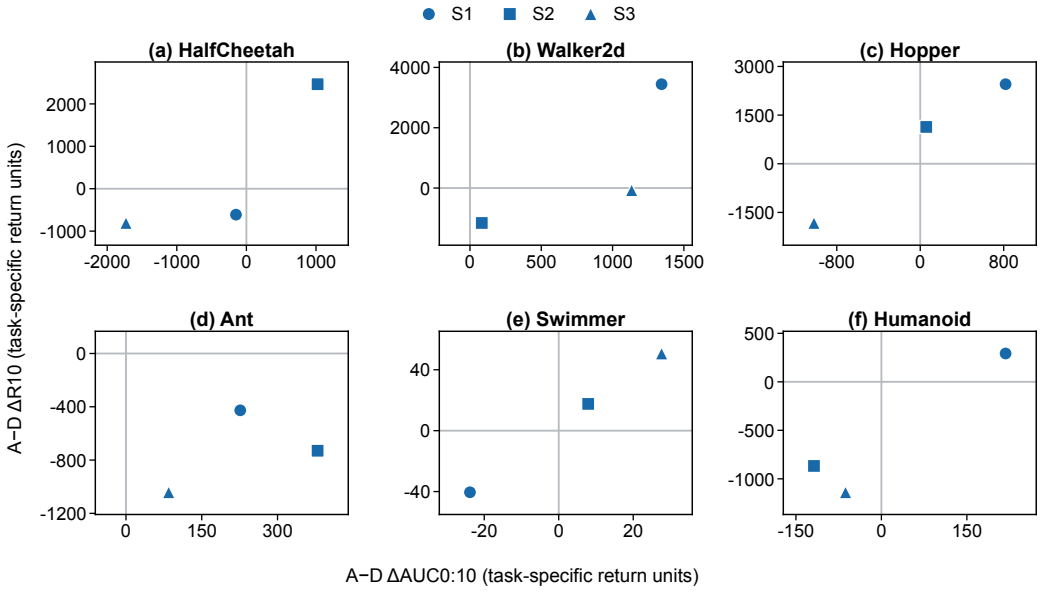

Figure 12: Learning-curve AUC and the original five-reset monitoring endpoint can disagree within the same A–D pair. All 18 paired training-run outcomes at 10 Mi are shown in task-native units: Ant improves $\mathrm{AUC}_{0:10}$ but loses R_{10} in every paired training-run comparison, while Walker improves AUC in every paired comparison but endpoint in only one.

Table D2: Complete 10 Mi PPO E/D/A training-monitoring outcomes. Entries are mean ± sample SD across training seeds; Appendix C defines the metrics and unit of analysis. Boldface marks the highest mean within each task and metric.

Task	Condition	$\mathrm{AUC}_{0:10}$	R_{10}	$\mathrm{Tail}_{8:10}$	$\mathrm{Best}_{0:10}$
HC	E	2541.2±1677.1	3230.9±2595.8	3207.5±2537.9	3281.5±2600.3
	D	**3757.2**±1948.3	4637.7±2601.2	4510.7±2515.0	4642.7±2607.1
	A	3471.5±1229.8	**4986.1**±949.2	**4614.6**±1170.4	**5066.4**±884.9
Walker	E	2275.3±721.6	3475.4±1168.7	3460.8±591.0	**4981.5**±275.5
	D	2062.7±240.6	3351.3±720.4	3002.1±237.9	4149.2±96.6
	A	**2916.1**±635.3	**4089.6**±1711.2	**3538.0**±1229.6	4929.2±994.5
Hopper	E	1243.3±299.5	1574.6±737.5	1467.0±239.1	2787.7±362.1
	D	**1451.7**±676.6	1628.2±903.4	**1816.9**±581.8	2969.6±382.4
	A	1404.0±272.2	**2215.6**±1358.1	1505.9±323.7	**2993.5**±330.9
Ant	E	3191.8±102.4	**5096.5**±58.8	4785.3±90.0	5216.1±49.6
	D	3056.3±108.5	4792.0±450.9	4638.3±176.5	5054.9±106.8
	A	**3286.4**±62.6	4059.2±478.1	**4900.3**±38.8	**5336.5**±31.7
Swimmer	E	37.0±8.8	45.1±12.8	42.1±10.4	60.0±25.9
	D	42.0±14.8	55.0±21.8	52.3±19.7	63.3±27.1
	A	**45.9**±13.2	**64.3**±30.2	**63.1**±30.0	**71.3**±26.4
Humanoid	E	4832.2±386.0	5674.5±2267.6	5894.1±1461.0	**6941.6**±193.7
	D	4853.9±157.9	**6542.1**±80.2	**6276.5**±488.4	6842.1±218.1
	A	**4866.1**±86.9	5970.7±680.8	6096.2±428.3	6695.1±273.9

Table D3: Complete paired PPO training-monitoring effects. Entries report the mean ± sample SD across paired training seeds and the number of positive paired training-run differences. D–E is computed by training seed from exact condition values, not from rounded means.

Task	Contrast	$\Delta\text{AUC}_{0:10}$	ΔR_{10}	$\Delta\text{Tail}_{8:10}$	$\Delta\text{Best}_{0:10}$
HC	A-E	930.3±485.7 (3/3+)	1755.2±1714.6 (2/3+)	1407.1±1429.8 (2/3+)	1785.0±1755.4 (2/3+)
	A-D	-285.7±1381.6 (1/3+)	348.4±1835.4 (1/3+)	103.9±1667.6 (1/3+)	423.7±1917.6 (1/3+)
	D-E	1216.0±1618.2 (3/3+)	1406.8±2072.4 (3/3+)	1303.2±2028.0 (2/3+)	1361.2±2049.5 (2/3+)
Walker	A-E	640.8±913.6 (2/3+)	614.2±1100.4 (2/3+)	77.2±667.4 (1/3+)	-52.3±1255.1 (1/3+)
	A-D	853.4±675.6 (3/3+)	738.4±2405.2 (1/3+)	535.9±1234.4 (2/3+)	780.0±910.5 (2/3+)
	D-E	-212.7±961.0 (2/3+)	-124.1±1665.6 (1/3+)	-458.7±567.1 (1/3+)	-832.3±371.6 (0/3+)
Hopper	A-E	160.7±333.7 (1/3+)	641.0±2095.5 (2/3+)	38.9±535.4 (1/3+)	205.8±462.3 (2/3+)
	A-D	-47.7±922.9 (2/3+)	587.4±2191.3 (2/3+)	-311.1±868.7 (1/3+)	24.0±618.8 (2/3+)
	D-E	208.4±685.7 (1/3+)	53.6±464.0 (2/3+)	350.0±343.0 (2/3+)	181.8±234.5 (3/3+)
Ant	A-E	94.6±161.6 (2/3+)	-1037.2±421.5 (0/3+)	115.0±74.8 (3/3+)	120.5±40.9 (3/3+)
	A-D	230.0±147.4 (3/3+)	-732.8±307.6 (0/3+)	262.0±201.2 (3/3+)	281.7±134.3 (3/3+)
	D-E	-135.5±145.6 (1/3+)	-304.5±418.6 (1/3+)	-147.0±168.2 (1/3+)	-161.2±118.9 (0/3+)
Swimmer	A-E	8.9±5.1 (3/3+)	19.2±18.9 (3/3+)	21.0±20.1 (3/3+)	11.4±0.5 (3/3+)
	A-D	3.9±26.0 (2/3+)	9.3±46.1 (2/3+)	10.8±45.1 (2/3+)	8.1±46.2 (2/3+)
	D-E	5.0±20.9 (2/3+)	9.9±27.6 (2/3+)	10.2±26.0 (2/3+)	3.3±45.8 (2/3+)
Humanoid	A-E	33.9±307.1 (1/3+)	296.2±1840.4 (1/3+)	202.1±1132.4 (1/3+)	-246.5±105.4 (0/3+)
	A-D	12.2±180.3 (1/3+)	-571.3±760.0 (1/3+)	-180.3±412.0 (1/3+)	-146.9±299.0 (1/3+)
	D-E	21.7±358.7 (1/3+)	867.6±2316.2 (1/3+)	382.4±1010.9 (2/3+)	-99.5±197.0 (1/3+)

D.3 COMMON-RESET FINAL-POLICY RETURNS AND BEHAVIOR

Tables D4 and D5 make the common 20-reset evaluation the comparison of final-policy returns. Ant has mean A–E and A–D differences of $+208.4$ and $+128.4$, with positive counts of 3/3 and 2/3. HalfCheetah and Walker also have positive mean A–D differences ($+279.0$ and $+409.3$), but the positive count is only 1/3 for each. Omitting that positive training-run pair changes the respective means to -720.4 and -1106.9, illustrating the distinction between positive averages and training-replicate consistency. Humanoid has negative mean A–E and A–D differences. These final-policy comparisons use the algorithm-specific inference adapters defined in Appendix C.3; their change from monitoring endpoints cannot be attributed to reset identity alone.

Table D4: PPO final-policy returns at $10\,\mathrm{Mi}$, $n = 3$ training seeds. Fixed final checkpoints are evaluated on the same 20 reset seeds, separately from the historical monitoring curves. Entries are means $\pm$ sample SD across training-seed model means; each model mean uses 20 evaluation episodes. Boldface marks the highest mean within each task and metric.

Task	E	D	A
HC	3297.7 ± 2659.6	4643.3 ± 2650.8	**4922.3** ± 1078.0
Walker	3408.6 ± 258.2	3410.7 ± 772.2	**3819.9** ± 1982.4
Hopper	1420.2 ± 773.1	1562.7 ± 203.7	**1813.4** ± 1186.2
Ant	4864.3 ± 210.9	4944.3 ± 11.1	**5072.7** ± 122.6
Swimmer	45.5 ± 13.0	55.3 ± 23.7	**63.8** ± 29.6
Humanoid	6423.6 ± 976.6	**6494.1** ± 169.5	5954.5 ± 684.3

Table D5: PPO paired differences on all 20 new reset seeds. SD is across three paired differences, one per training seed; parentheses count positive paired training-run differences. D–E has no direct AMC encoder credit.

Task	A–E	A–D	D–E
HC	1624.6 ± 1690.0 (2/3+)	279.0 ± 1732.1 (1/3+)	1345.6 ± 2033.9 (2/3+)
Walker	411.3 ± 2008.9 (1/3+)	409.3 ± 2710.8 (1/3+)	2.1 ± 703.1 (2/3+)
Hopper	393.2 ± 1937.4 (2/3+)	250.7 ± 1363.8 (2/3+)	142.5 ± 658.4 (2/3+)
Ant	208.4 ± 206.5 (3/3+)	128.4 ± 121.0 (2/3+)	80.0 ± 220.6 (2/3+)
Swimmer	18.3 ± 18.6 (2/3+)	8.5 ± 47.2 (2/3+)	9.8 ± 28.8 (2/3+)
Humanoid	-469.1 ± 586.4 (1/3+)	-539.6 ± 581.9 (1/3+)	70.5 ± 971.2 (1/3+)

The final-policy behavior table covers the same 54 E/D/A models on 1080 new evaluation episodes. Table D6 reports velocity, action saturation, episode length and actual termination/full-horizon counts, with each task and condition summarized from model means for three training seeds.

Table D6: PPO final-policy behavior at 10 Mi on 20 common new reset seeds per model. Each row has three training seeds and 60 valid evaluation episodes for every metric. Values are means ± sample SD across the three model means, not across pooled episodes. v_x is episode-mean velocity; Sat. is normalized-action saturation in percent (Appendix C.2). Term. and Full are actual terminated and full-horizon episode counts out of 60. These are new evaluations, separate from historical monitoring behavior.

Task	Condition	v_x (m/s)	Sat. (%)	Length	Term.	Full
HC	E	3.71 ± 2.71	34.8 ± 5.4	1000.0 ± 0.0	0/60	60/60
	D	5.10 ± 2.73	45.6 ± 16.2	1000.0 ± 0.0	0/60	60/60
	A	5.40 ± 1.11	46.8 ± 10.6	1000.0 ± 0.0	0/60	60/60
Walker	E	3.30 ± 0.20	51.3 ± 5.1	727.8 ± 37.5	36/60	24/60
	D	2.74 ± 0.54	47.5 ± 6.9	847.7 ± 176.3	16/60	44/60
	A	3.55 ± 1.24	48.1 ± 0.5	769.4 ± 233.6	26/60	34/60
Hopper	E	2.06 ± 0.15	25.2 ± 15.0	453.8 ± 242.4	53/60	7/60
	D	1.97 ± 0.17	31.4 ± 10.0	515.8 ± 67.8	47/60	13/60
	A	2.11 ± 0.22	35.1 ± 12.6	572.7 ± 371.9	40/60	20/60
Ant	E	4.88 ± 0.06	0.0 ± 0.0	955.6 ± 42.6	4/60	56/60
	D	4.83 ± 0.08	0.0 ± 0.0	987.9 ± 21.0	1/60	59/60
	A	5.04 ± 0.06	0.0 ± 0.0	969.7 ± 29.1	4/60	56/60
Swimmer	E	0.05 ± 0.01	24.8 ± 26.6	1000.0 ± 0.0	0/60	60/60
	D	0.06 ± 0.02	22.3 ± 17.9	1000.0 ± 0.0	0/60	60/60
	A	0.06 ± 0.03	39.4 ± 12.0	1000.0 ± 0.0	0/60	60/60
Humanoid	E	1.73 ± 0.37	21.9 ± 3.4	925.0 ± 130.0	5/60	55/60
	D	1.66 ± 0.14	22.0 ± 1.5	977.0 ± 20.4	7/60	53/60
	A	1.09 ± 0.55	20.2 ± 1.7	999.1 ± 1.5	1/60	59/60

D.4 Late-window behavior and reset-only sensitivity

Table D7 retains the historical monitoring late-window A–E movement analysis. In that table, displacement is the task-timestep integral of recorded mean x-velocity. Appendix C gives the nesting, definitions and limits of the behavioral measures. HC and W2d abbreviate HalfCheetah and Walker2d.

Reset-only sensitivity of fixed final policies. Changing only the reset-seed set reverses every Ant A–D pair: its mean endpoint difference moves from -732.8 on monitoring reset seeds to $+508.5$ on new reset seeds. Ant A–E likewise moves from -1037.3 to $+225.2$, with all three signs reversed. Endpoint ordering is therefore not reset-invariant for these fixed trained models.

The same 54 fixed 10 Mi E/D/A models are evaluated on 20 new reset seeds, R1–R20, disjoint from the five monitoring reset seeds M1–M5. Each model first reproduces its original five monitoring returns, then evaluates four groups of five new reset seeds under the original FP32 evaluator. No checkpoint is reselected; model weights and normalization remain fixed. This yields 1080 new-reset episodes, with three training seeds as the independent units for each task and condition. The new reset seeds test sensitivity of fixed trained models to initial conditions.

Late-window behavior is also extracted for all three conditions on the 33 scheduled checkpoints in $[8, 10]$ Mi, with five monitoring reset seeds per checkpoint. The 8910 episodes are nested within 54 training runs. D's continued replicate S2 runs retain their original-source and checkpoint-resume provenance. Return, velocity, displacement, termination, length and saturation retain the definitions used in the final-checkpoint extraction.

Walker A–D also changes mean sign, whereas HalfCheetah A–E and Hopper A–D velocity retain positive means across reset sets. Reset sensitivity is thus task- and contrast-specific rather than a uniform loss of ordering. Table D8 keeps late-window, monitoring-endpoint and new-reset outcomes side by side. This study does not add training runs or evaluate the separate 20 Mi system-comparison and hardware models.

Table D7: Paired A–E movement deltas at 10 Mi and over the inclusive 8–10 Mi tail. Entries are mean ± sample SD across three paired training seeds after within-run reset averaging. p_{nat} denotes the natural-termination rate; pp denotes percentage points.

Task	$\Delta v_{x,10}$ (m/s)	$\Delta v_{x,8:10}$ (m/s)	$\Delta d_{x,10}$ (m)	$\Delta p_{\mathrm{nat},10}$ (pp)	ΔL_{10} (steps)
HC	$+1.819 \pm 1.736$	$+1.470 \pm 1.450$	$+91.0 \pm 86.8$	0.0 ± 0.0	0.0 ± 0.0
W2d	$+0.323 \pm 1.080$	$+0.014 \pm 0.588$	$+4.2 \pm 8.7$	0.0 ± 34.6	$+90.7 \pm 114.6$
Hopper	$+0.060 \pm 0.342$	-0.137 ± 0.110	$+3.6 \pm 11.6$	-33.3 ± 70.2	$+195.1 \pm 644.3$
Ant	-0.424 ± 0.404	$+0.086 \pm 0.057$	-49.2 ± 22.9	$+20.0 \pm 20.0$	-127.0 ± 158.2
Swimmer	$+0.019 \pm 0.019$	$+0.021 \pm 0.020$	$+0.8 \pm 0.8$	0.0 ± 0.0	0.0 ± 0.0
Humanoid	-0.388 ± 0.512	-0.251 ± 0.247	-6.6 ± 6.6	-20.0 ± 34.6	$+179.0 \pm 310.0$

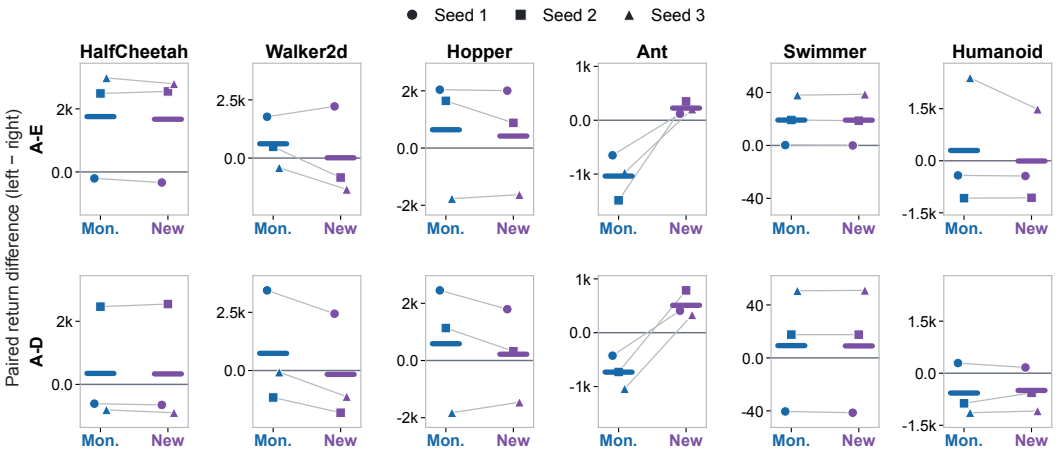

Figure 13: Sensitivity of final policies to the reset-seed set. Lines connect each paired difference for each training seed on five monitoring reset seeds (Mon.) to the same pair on 20 disjoint reset seeds (New); ticks show means across three training seeds. Ant reverses every A–E and A–D sign, while several other effects persist.

Table D8: Late-window and reset-set contrasts for the same fixed $10\,\mathrm{Mi}$ PPO Actors. Entries report the left-minus-right mean $\pm$ sample SD across paired training seeds and the positive count for the late grid, original monitoring reset seeds (Mon.), and 20 disjoint reset seeds (New).

Task	Contrast	Late $\Delta R_{8:10}$	Mon. ΔR_{10}	New ΔR_{10}	New Δv_x (m/s)
HC	A-E	$+1407.1 \pm 1429.8$ $2/3 > 0$	$+1755.2 \pm 1714.6$ $2/3 > 0$	$+1670.8 \pm 1741.8$ $2/3 > 0$	$+1.734 \pm 1.764$ $2/3 > 0$
	A-D	$+103.9 \pm 1667.6$ $1/3 > 0$	$+348.4 \pm 1835.4$ $1/3 > 0$	$+332.7 \pm 1922.8$ $1/3 > 0$	$+0.349 \pm 1.987$ $1/3 > 0$
W2d	A-E	$+77.2 \pm 667.4$ $1/3 > 0$	$+614.2 \pm 1100.4$ $2/3 > 0$	$+12.2 \pm 1929.5$ $1/3 > 0$	$+0.019 \pm 1.316$ $1/3 > 0$
	A-D	$+535.9 \pm 1234.4$ $2/3 > 0$	$+738.4 \pm 2405.2$ $1/3 > 0$	-164.8 ± 2282.3 $1/3 > 0$	$+0.540 \pm 1.462$ $1/3 > 0$
Hopper	A-E	$+38.9 \pm 535.4$ $1/3 > 0$	$+641.0 \pm 2095.5$ $2/3 > 0$	$+419.5 \pm 1861.4$ $2/3 > 0$	$+0.090 \pm 0.400$ $2/3 > 0$
	A-D	-311.1 ± 868.7 $1/3 > 0$	$+587.4 \pm 2191.3$ $2/3 > 0$	$+220.3 \pm 1629.3$ $2/3 > 0$	$+0.137 \pm 0.096$ $3/3 > 0$
Ant	A-E	$+115.0 \pm 74.8$ $3/3 > 0$	-1037.2 ± 421.5 $0/3 > 0$	$+225.2 \pm 116.5$ $3/3 > 0$	$+0.140 \pm 0.046$ $3/3 > 0$
	A-D	$+262.0 \pm 201.2$ $3/3 > 0$	-732.8 ± 307.6 $0/3 > 0$	$+508.5 \pm 243.7$ $3/3 > 0$	$+0.257 \pm 0.076$ $3/3 > 0$
Swimmer	A-E	$+21.0 \pm 20.1$ $3/3 > 0$	$+19.2 \pm 18.9$ $3/3 > 0$	$+19.1 \pm 19.3$ $2/3 > 0$	$+0.019 \pm 0.019$ $2/3 > 0$
	A-D	$+10.8 \pm 45.1$ $2/3 > 0$	$+9.3 \pm 46.1$ $2/3 > 0$	$+9.0 \pm 46.8$ $2/3 > 0$	$+0.009 \pm 0.047$ $2/3 > 0$
Humanoid	A-E	$+202.1 \pm 1132.4$ $1/3 > 0$	$+296.2 \pm 1840.4$ $1/3 > 0$	-7.4 ± 1329.9 $1/3 > 0$	-0.476 ± 0.394 $0/3 > 0$
	A-D	-180.3 ± 412.0 $1/3 > 0$	-571.3 ± 760.0 $1/3 > 0$	-495.8 ± 629.8 $1/3 > 0$	-0.542 ± 0.544 $1/3 > 0$

D.5 FINAL-CHECKPOINT GRADIENT CHECKS AND SHADOW-UPDATE INTERVENTIONS

Equation 3 identifies the extra gradient for the same data samples and parameters; Appendix A derives its assumptions. The final-checkpoint interventions below measure that local route rather than reconstructing the from-zero training effects.

On the same factual pre-state, normalized input, parameters, target and clipped-value branch, let $D = \mathrm{diag}(\mathbf{1}[0 < h < 1])$ and $d_u = \partial L_V / \partial u$. The value-loss contributions to one common encoder-parameter vector are

$$g_{\mathrm{value,event}} = J_{h,\eta}^{\mathsf{T}} D W_V^{\mathsf{T}} d_u, \qquad g_{\mathrm{mem}} = J_{h,\eta}^{\mathsf{T}} W_h^{\mathsf{T}} d_u.$$

Batch probes use the actual loss reduction and microbatch scaling before taking vector norms. Let $g_{\mathrm{core}} = g_\pi + g_{\mathrm{value,event}}$ include both event-mediated policy and value encoder gradients. The reported gradient ratio is $q_g = \|g_{\mathrm{mem}}\| / (\|g_{\mathrm{core}}\| + 10^{-12})$, not a ratio to the value-event branch alone. The denominator can include cancellation between policy and value gradients. This norm ratio is distinct from the unregularized adjoint-mass fraction R_{out} below. Branch cosines are unavailable if either norm is zero; a whole-model joint gradient norm is not a branch-specific encoder quantity. Measurements must preserve model, RNG, accumulated gradients and optimizer state. At zero initialization $g_{\mathrm{mem}} = 0$; a first nonzero value update can change W_h, allowing subsequent, not guaranteed, encoder credit.

Final-checkpoint local intervention. Six completed $10\,\mathrm{Mi}$ attached-A MCrPPO models, from HalfCheetah and Ant with training seeds S1–S3, each provide 4096 newly recorded environment steps. Sampling uses 32 environments, 64 warmup decisions and 128 recorded decisions. The three shadow branches are attached, membrane-detached and value-encoder-detached; the last removes both value-to-encoder routes while retaining policy gradients and Critic parameter learning. They share data samples, targets, permutations, original Adam moments and a fixed LR. Both normalization references and the scheduler are frozen. Each shadow receives eight optimizer updates; the original model and all shadows are evaluated on reset seeds L1–L5. The tasks were chosen after inspecting their training outcomes. This is a short final-checkpoint intervention rather than a fresh-from-zero training comparison.

The initial gradient probe uses only the first logical batch of 2048 samples, B_1, not all 4096 recorded environment steps. Define

$$M = \sum_{t \in B_1, i} |W_{h,i} d_{u,t}|, \qquad R_{\mathrm{out}} = \frac{\sum_{t \in B_1, i} |W_{h,i} d_{u,t}| \mathbf{1}[h_{t,i} \notin (0,1)]}{M} \quad (M > 0).$$

This *outside-support membrane-adjoint fraction* measures absolute mass before the encoder Jacobian. The implementation uses neither an added ϵ nor a clamped denominator: $M = 0$ is undefined and recorded as null. None of the six records has zero mass. It is not the fraction of encoder gradients, neurons, or training steps outside support.

This unregularized fraction measures access to the additional route; it does not measure usefulness or gradient alignment.

Numerical check summary. On the first 2048 recorded transitions per checkpoint, the attached and membrane-detached shadow branches have exactly equal FP32 policy-mean and Critic-value tensors in all six checks (6/6). The value-loss encoder-gradient difference also passes the Eq. 3 check in all six cases, using elementwise relative tolerance 5×10^{-4} and absolute tolerance 2×10^{-6}. Each pair shares the same trained parameters and recorded inputs; these are local checks on shadows of six attached-A checkpoints. Table D9 reports the associated behavioral outcomes without pooling the two tasks. The local A–D sign is positive for all three HalfCheetah models and one of three Ant models. This verifies route access, not a universal benefit or learning-curve effect. The original checkpoint is unchanged, and shadow parameters are not exported or reused as initializers.

Table D9: Final-checkpoint local intervention on six attached-A MCrPPO models. R_{out} and q_g are defined in the text; local return differences follow eight shadow optimizer updates and average five paired reset episodes. This is not a fresh-from-zero training comparison.

Task	Training seed	R_{out} (%)	q_g	Local ΔR^{A-D}
HalfCheetah	S1	59.39	0.0061	+92.59
HalfCheetah	S2	60.94	0.0342	+493.97
HalfCheetah	S3	60.03	0.0577	+396.63
Ant	S1	56.88	0.0420	+1918.76
Ant	S2	56.09	0.0259	-626.38
Ant	S3	55.24	0.0276	-56.64

Scope of the mechanism evidence. The local checks characterize the added route, and shadow updates measure short-run effects at fixed models. They complement, rather than reconstruct, complete E/D/A training runs. Longitudinal branch-gradient and optimizer measurements could help explain their evolution during learning (Appendix H.3). Statistical units and descriptive sign counts follow Appendix C; neuron–sample coordinates are not independent training replicates.

D.6 Frozen-weight forward diagnostics

The frozen-weight analysis examines the Critic's continuous membrane contribution and the information present before and after thresholding. It is descriptive: forward amplitude does not establish value accuracy, useful gradient alignment or the cause of a training gain. Figure 14 separates the event, membrane and recurrent-carry terms; Figure 15 shows the corresponding membrane, event and Actor-output traces.

The internal analysis binds each stored CPU Actor trajectory to its original final-checkpoint Actor hash and to the full learner checkpoint at 20 Mi. The five trainable Actor tensors, W_x, b_x, W_o, b_o and ρ^{raw}, and the observation RMS are equal between the learner and export. Restricted tensor deserialization reads the Critic parameters; no training model, simulator or optimizer is executed.

With weights fixed at the final checkpoints, define $e_t = W_V z_t + b_V$ and $m_t = W_h h_t$ from the recorded states. We reconstruct $u_t = 0.2u_{t-1} + e_t + m_t$ from $u_0 = 0$ and call $0.2u_{t-1}$ the carry. Matrix products and this reconstruction use FP64 arithmetic on the recorded FP32 quantities; they are not claimed to be bitwise training-time value traces. The readouts are in the checkpoint's normalized value-membrane coordinate, with no new normalizer update. The magnitude ratio is $\sqrt{\langle m_t^2\rangle_t}/\sqrt{\langle e_t^2\rangle_t}$. Its centered counterpart replaces each term by its deviation from its own episode mean. These within-episode temporal standard deviations are signal summaries, not error bars across training seeds. Zero denominators are unavailable, not stabilized by a fitted constant; none of the 18 Actors has a zero denominator. The value-loss residual, value-target bank and encoder-gradient vectors are absent, so these ratios cannot establish value accuracy or routed loss credit.

Selection is outcome-independent by identifier: all 18 available final Actors, the lowest existing monitoring reset identifier, and all valid decisions in that episode. Episode lengths range from 310 to 1000; failures are not padded to full horizon or excluded. The illustrative panels use the lowest HalfCheetah training-seed identifier and its first 160 decisions. Neuron indices are 64 uniformly spaced indices from 0 through 511, fixed before displaying activity, without sorting or temporal smoothing. Displayed ticks 0, 251 and 511 are actual selected indices, not 512 plotted rows. The action order follows the task XML: back thigh, back shin, back foot, front thigh, front shin, front foot. The colorbar marks 0, 0.5 and 1; only the lowest and highest displayed colors use the observed excerpt range. This is a retrospective descriptive selection rule, not preregistration or an independent held-out evaluation.

Figure 1(b,d) uses the same HalfCheetah example. The environment frames render recorded post-physics states at 2, 4 and 6 s; the unsmoothed unit-0 trace records decision-time membrane and events over 0–7.95 s. These are different sampling points in the control loop, not pixel observations supplied to the policy. The measured final 1×512 membrane row is displayed in row-major 16×32 order for legibility, not as a spatial neural topology. Its initial-zero panel represents the source initialization contract, not a stored initial checkpoint or a measured trajectory of weight growth.

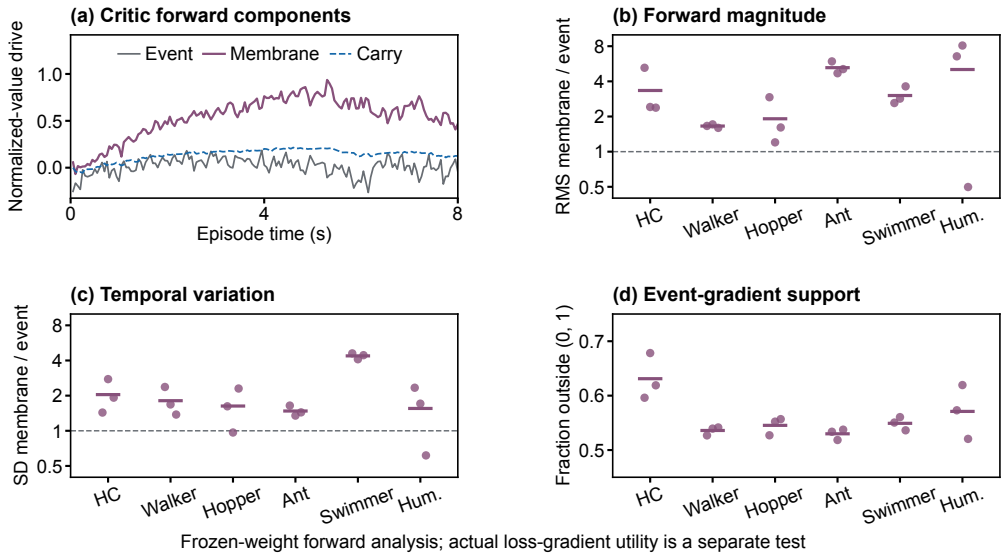

Figure 14: Frozen-weight event, membrane and carry contributions to the Critic. Panels show the individual terms, membrane/event RMS and centered-SD ratios, and the fraction of membrane states outside the event-surrogate support. Each point represents one monitoring episode from a final Actor; bars are task means. Forward amplitude and occupancy are not measurements of useful loss credit.

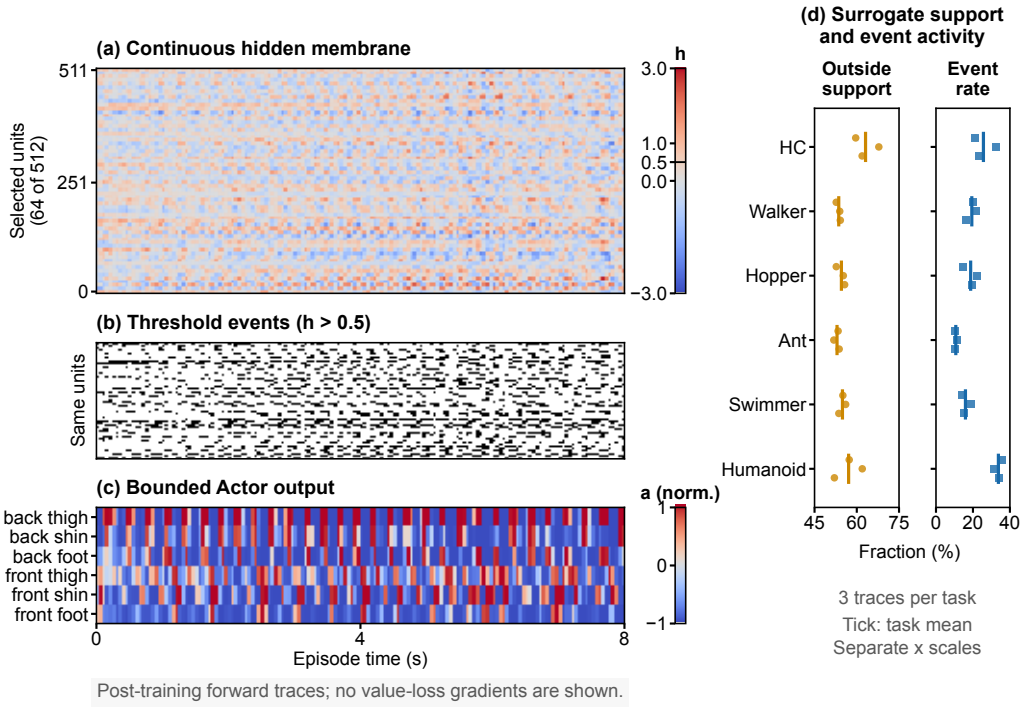

Figure 15: Continuous membrane dynamics and continuous actions coexist with binary event-mediated hidden-state access. Shared-time HalfCheetah traces use the same 64 hidden units and all six actions; the final panel separates outside-support and event fractions across tasks. The ranges differ and the quantities are not complements or value-loss gradients.

E CONTROLLED SAC AND TD3 RESULTS

The off-policy study contains 108 completed runs: SAC and TD3 crossed with E/D/A, six MuJoCo-v5 tasks and three paired training seeds, at 1,048,576 environment steps per run. The frozen E/A conditions and added D conditions form 18 matched triplets per learning rule without concatenating trajectories or changing parameter-update ownership. The earlier 500,000-step SAC HalfCheetah pilot is excluded.

The complete pathway has higher mean monitoring AUC than E on all six SAC tasks and five of six TD3 tasks. Final-policy comparisons distinguish this average trend from the effect of direct credit. SAC HalfCheetah improves with A relative to D, whereas SAC Walker benefits from D relative to E but deteriorates when the additional direct route is enabled. SAC Humanoid is negative for both A–E and A–D in every final-policy training pair. Tables E1–E8 report condition means and paired contrasts without pooling the learning rules.

E.1 DESIGN, PAIRING AND IMPLEMENTATION CHECKS

Training seeds are S1–S3 and monitoring reset seeds are M1–M5. Table E12 reports scheduled update counts and local state-reconstruction checks; Table E11 reports wall time and memory. Implementation regression checks were performed to verify E/A agreement when adding the D condition, using isolated eight-update CPU runs and comparing model, target, optimizer, RNG and non-time diagnostic states. The checks also covered initial evaluation episodes, observation references, task schemas, initialization, credit isolation and gradient-update ownership.

E.2 LEARNING CURVES AND COMPLETE-PATHWAY MONITORING EFFECTS

Figure 16 displays the complete E/D/A learning curves across PPO, SAC and TD3. The off-policy conditions are compared at their own 1 Mi budget.

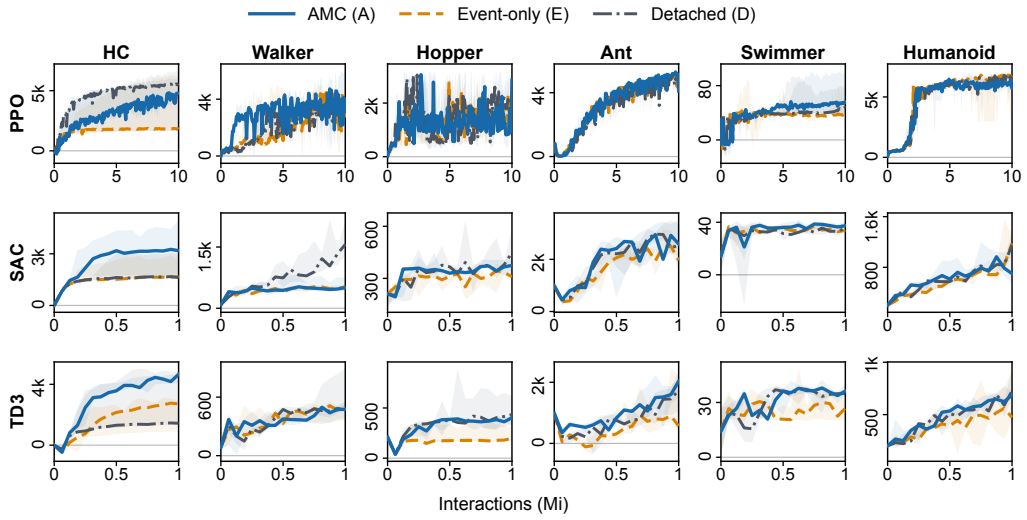

Figure 16: Complete E/D/A learning curves for PPO, SAC and TD3. All 162 runs are represented on their full evaluation grids: 161 checkpoints through 10 Mi for PPO and 17 through 1 Mi for SAC/TD3. Lines are unsmoothed medians across three training seeds; shading is the observed range. Only the PPO display in the main figure is subsampled.

The primary outcome is learning-curve AUC over $[0, 1]$ Mi; secondary outcomes are the fixed endpoint, the mean of the last five scheduled evaluations and the best monitoring mean. The realized grid contains 17 evaluations at multiples of 65,536 interactions. $\text{Tail}_{.75:1}$ averages the five checkpoints from $12/16$ Mi through 1 Mi, inclusive, after each checkpoint averages five fixed-reset episodes.

This last-quarter summary differs from a four-point $[.8, 1]$ Mi window. Final monitoring movement and termination are extracted from those episodes rather than inferred from return. Shared statistical and display rules are given in Appendix C.

Table E1: Off-policy learning-curve AUC and monitoring endpoints at 1 Mi. Entries are mean ± sample SD across three training seeds. $\mathrm{AUC}_{0:1}$ is normalized by the $[0, 1]$ Mi integration-window length; R_1 is the final checkpoint's original monitoring score, not its common 20-reset reevaluation. Boldface marks the highest mean within each task and metric.

Family	Task	Condition	n	$\mathrm{AUC}_{0:1}$	Endpoint R_1
SAC	HC	E	3	1732.0 ± 509.3	1970.7 ± 530.8
		D	3	1725.4 ± 488.4	2065.3 ± 794.7
		A	3	**2610.6** ± 1020.7	**3310.2** ± 1602.9
	Walker	E	3	430.7 ± 38.4	504.6 ± 67.5
		D	3	**809.5** ± 184.7	**1658.0** ± 416.4
		A	3	492.6 ± 136.5	557.4 ± 169.7
	Hopper	E	3	295.3 ± 35.6	317.5 ± 17.9
		D	3	**346.0** ± 16.9	**459.2** ± 125.2
		A	3	337.5 ± 10.9	369.8 ± 43.7
	Ant	E	3	1659.3 ± 166.1	1882.7 ± 268.2
		D	3	**1946.9** ± 157.7	2680.1 ± 632.7
		A	3	1925.6 ± 446.8	**2841.2** ± 591.6
	Swimmer	E	3	33.5 ± 1.4	34.8 ± 2.9
		D	3	31.3 ± 2.0	36.0 ± 3.8
		A	3	**34.8** ± 0.2	**37.9** ± 1.9
	Humanoid	E	3	586.8 ± 33.9	**1218.5** ± 80.4
		D	3	604.6 ± 16.0	1173.8 ± 197.7
		A	3	**653.7** ± 97.8	781.0 ± 283.5
TD3	HC	E	3	1473.5 ± 802.4	2191.2 ± 1145.9
		D	3	1913.3 ± 1534.0	2526.5 ± 1976.0
		A	3	**3089.0** ± 307.4	**4469.3** ± 353.9
	Walker	E	3	367.2 ± 60.7	463.4 ± 87.4
		D	3	**383.4** ± 59.3	**609.3** ± 242.9
		A	3	353.0 ± 50.1	469.8 ± 92.1
	Hopper	E	3	208.1 ± 80.8	225.6 ± 73.2
		D	3	**364.7** ± 85.8	**455.3** ± 155.1
		A	3	313.0 ± 19.8	389.9 ± 38.1
	Ant	E	3	509.2 ± 228.0	1071.0 ± 1020.9
		D	3	724.5 ± 181.4	1716.7 ± 649.2
		A	3	**952.6** ± 120.1	**1882.0** ± 355.6
	Swimmer	E	3	26.2 ± 2.2	26.7 ± 6.5
		D	3	30.3 ± 1.7	35.1 ± 1.6
		A	3	**31.0** ± 1.8	**36.8** ± 3.6
	Humanoid	E	3	401.1 ± 101.1	608.7 ± 268.3
		D	3	462.9 ± 37.2	707.5 ± 34.3
		A	3	**485.4** ± 21.4	**764.5** ± 150.0

Table E2: Paired off-policy A–E training-monitoring effects. Entries give the mean ± sample SD across paired training seeds and the number of positive paired training-run differences for each task-native outcome.

Algorithm	Task	$\Delta\text{AUC}_{0:1}$	ΔR_1	$\Delta\text{Tail}_{0.75:1}$	$\Delta\text{Best}_{0:1}$
SAC	HC	+878.6±1476.3 2/3+	+1339.5±2069.9 2/3+	+1175.6±1978.7 2/3+	+1262.2±2235.0 2/3+
	Walker	+61.9±114.5 2/3+	+52.7±150.7 2/3+	+57.1±131.4 2/3+	+71.6±166.4 2/3+
	Hopper	+42.2±28.1 3/3+	+52.3±28.9 3/3+	+57.2±22.9 3/3+	+46.6±37.0 3/3+
	Ant	+266.3±498.3 2/3+	+958.5±546.0 3/3+	+448.0±674.7 2/3+	+390.9±756.9 1/3+
	Swimmer	+1.3±1.4 2/3+	+3.1±4.2 2/3+	+2.6±3.5 2/3+	+0.8±3.0 2/3+
	Humanoid	+66.9±130.8 2/3+	-437.6±348.3 0/3+	-37.2±298.7 1/3+	-346.9±382.9 1/3+
TD3	HC	+1615.6±538.3 3/3+	+2278.1±793.2 3/3+	+2130.6±815.5 3/3+	+2253.9±920.6 3/3+
	Walker	-14.2±20.0 1/3+	+6.4±61.4 2/3+	-15.2±44.1 2/3+	+17.5±61.6 2/3+
	Hopper	+104.9±69.8 3/3+	+164.3±62.5 3/3+	+138.2±76.4 3/3+	+142.4±104.7 3/3+
	Ant	+443.4±345.1 3/3+	+811.0±1376.0 2/3+	+604.9±975.5 2/3+	+443.1±1223.8 2/3+
	Swimmer	+4.7±3.8 3/3+	+10.1±9.1 3/3+	+7.3±11.0 2/3+	+0.8±11.5 2/3+
	Humanoid	+84.3±88.0 2/3+	+155.8±121.1 3/3+	+117.5±169.6 2/3+	+126.8±117.1 3/3+

These tables retain the training-curve AUC, tail, best and historical monitoring endpoint. Section 2 defines the E/D/A interventions; the local state-reconstruction diagnostics are described below.

E.3 Common-reset final-policy comparisons

SAC HalfCheetah has a mean A–D endpoint difference of $+1226.3$ with 3/3 positive paired training-run differences. Walker separates the controls more sharply: D–E is $+1296.6$ (3/3 positive), A–D is -1242.8 (0/3 positive), and the resulting A–E mean is only $+53.8$ with 2/3 positive paired training-run differences. Both A–E and A–D are negative in all three SAC Humanoid training-run pairs. For TD3 HalfCheetah, A–E and A–D are $+2199.9$ and $+1965.9$, each positive for all three training seeds. The complete final-policy condition means and paired differences are in Tables E3– E6.

Table E3: SAC final-policy returns at 1 Mi, $n = 3$ training seeds. Fixed final checkpoints are evaluated on the same 20 reset seeds, separately from the historical monitoring curves. Entries are means ± sample SD across training-seed model means; each model mean uses 20 evaluation episodes. Boldface marks the highest mean within each task and metric.

Task	E	D	A
HC	1975.4 ± 544.9	2047.5 ± 787.8	**3273.7** ± 1565.8
Walker	507.9 ± 65.8	**1804.5** ± 515.9	561.7 ± 177.7
Hopper	313.9 ± 21.8	**436.7** ± 136.2	368.9 ± 43.4
Ant	2079.7 ± 216.7	**3126.2** ± 114.5	2870.7 ± 609.7
Swimmer	34.6 ± 2.6	36.8 ± 4.0	**37.1** ± 2.0
Humanoid	1052.3 ± 29.0	**1065.2** ± 98.2	786.7 ± 234.9

Table E4: SAC paired differences on all 20 new reset seeds. SD is across three paired differences, one per training seed; parentheses count positive paired training-run differences. D–E has no direct AMC encoder credit.

Task	A–E	A–D	D–E
HC	1298.3 ± 2046.8 (2/3+)	1226.3 ± 943.7 (3/3+)	72.1 ± 1159.2 (1/3+)
Walker	53.8 ± 159.8 (2/3+)	-1242.8 ± 691.3 (0/3+)	1296.6 ± 537.3 (3/3+)
Hopper	55.0 ± 22.2 (3/3+)	-67.8 ± 145.9 (1/3+)	122.8 ± 142.7 (3/3+)
Ant	791.0 ± 461.5 (3/3+)	-255.6 ± 550.3 (2/3+)	1046.6 ± 110.3 (3/3+)
Swimmer	2.6 ± 4.6 (2/3+)	0.3 ± 4.0 (1/3+)	2.2 ± 5.3 (2/3+)
Humanoid	-265.6 ± 226.6 (0/3+)	-278.5 ± 148.1 (0/3+)	13.0 ± 103.2 (2/3+)

Table E5: TD3 final-policy returns at $1\,\mathrm{Mi}$, $n = 3$ training seeds. Fixed final checkpoints are evaluated on the same 20 reset seeds, separately from the historical monitoring curves. Entries are means ± sample SD across training-seed model means; each model mean uses 20 evaluation episodes. Boldface marks the highest mean within each task and metric.

Task	E	D	A
HC	2194.7 ± 1144.6	2428.7 ± 1814.6	**4394.6** ± 291.2
Walker	463.4 ± 92.5	**612.8** ± 269.0	471.8 ± 102.0
Hopper	229.6 ± 79.7	**449.6** ± 144.7	385.6 ± 36.8
Ant	1017.0 ± 555.4	1234.3 ± 505.4	**1673.4** ± 286.6
Swimmer	26.5 ± 5.9	34.9 ± 1.1	**36.2** ± 2.2
Humanoid	589.7 ± 206.3	611.9 ± 74.0	**727.7** ± 31.5

Table E6: TD3 paired differences on all 20 new reset seeds. SD is across three paired differences, one per training seed; parentheses count positive paired training-run differences. D–E has no direct AMC encoder credit.

Task	A–E	A–D	D–E
HC	2199.9 ± 878.5 (3/3+)	1965.9 ± 1571.8 (3/3+)	234.1 ± 1424.2 (2/3+)
Walker	8.4 ± 84.3 (2/3+)	-140.9 ± 182.4 (1/3+)	149.3 ± 263.6 (2/3+)
Hopper	155.9 ± 70.8 (3/3+)	-64.0 ± 179.1 (2/3+)	219.9 ± 170.5 (3/3+)
Ant	656.5 ± 772.9 (2/3+)	439.1 ± 771.0 (2/3+)	217.4 ± 208.3 (3/3+)
Swimmer	9.8 ± 6.9 (3/3+)	1.3 ± 2.4 (2/3+)	8.5 ± 4.9 (3/3+)
Humanoid	138.0 ± 187.3 (2/3+)	115.8 ± 88.0 (3/3+)	22.2 ± 176.9 (2/3+)

E.4 DETACHED-READOUT AND DIRECT-CREDIT MONITORING EFFECTS

The completed extension adds a detached D condition for each rule, task and training seed; the original E/A runs are held fixed rather than repeated. The intervention and its learning-rule-specific ownership are defined in Section 2.

In the training-monitoring summaries, SAC HalfCheetah has positive A–D AUC and endpoint differences for every training seed, whereas SAC Walker is negative for every training seed on both outcomes. TD3 HalfCheetah has a positive mean A–D monitoring endpoint, while TD3 Walker and Hopper have negative means. Together with the full 20-reset results above, the detached control shows that a whole-pathway difference does not identify direct encoder credit as the beneficial component. Tables E7 and E8 report the D–E and A–D paired contrasts on the training-monitoring grid. The common 20-reset final-policy contrasts are reported separately in Tables E4 and E6.

Table E7: Paired SAC effects of the detached membrane readout and additional direct credit at 1 Mi. Entries give the mean ± sample SD across paired training-run differences in task-native return units. Each parenthetical $k/3$ counts positive differences out of three pairs. Outcomes use the training-monitoring grid; Appendix C defines the metrics.

Task	Contrast	$\Delta\mathrm{AUC}_{0:1}$	ΔR_1	$\Delta\mathrm{Tail}_{.75:1}$	$\Delta\mathrm{Best}_{0:1}$
HC	$D-E$	-6.7 ± 842.3 (1/3)	94.7 ± 1158.9 (1/3)	77.3 ± 1211.4 (1/3)	59.6 ± 1312.8 (1/3)
	$A-D$	885.3 ± 662.8 (3/3)	1244.8 ± 962.3 (3/3)	1098.3 ± 861.8 (3/3)	1202.6 ± 970.0 (3/3)
Walker	$D-E$	378.8 ± 184.2 (3/3)	1153.4 ± 422.0 (3/3)	833.6 ± 371.7 (3/3)	1148.8 ± 457.8 (3/3)
	$A-D$	-316.9 ± 293.4 (0/3)	-1100.7 ± 572.2 (0/3)	-776.5 ± 502.8 (0/3)	-1077.2 ± 619.1 (0/3)
Hopper	$D-E$	50.6 ± 48.5 (3/3)	141.7 ± 138.9 (3/3)	87.5 ± 49.6 (3/3)	181.8 ± 160.6 (3/3)
	$A-D$	-8.5 ± 27.7 (1/3)	-89.4 ± 146.2 (1/3)	-30.3 ± 65.5 (1/3)	-135.2 ± 187.1 (1/3)
Ant	$D-E$	287.7 ± 22.0 (3/3)	797.5 ± 409.5 (3/3)	566.1 ± 299.6 (3/3)	502.4 ± 183.0 (3/3)
	$A-D$	-21.4 ± 511.7 (2/3)	161.0 ± 890.2 (2/3)	-118.2 ± 619.7 (1/3)	-111.6 ± 591.9 (1/3)
Swimmer	$D-E$	-2.2 ± 1.2 (0/3)	1.2 ± 4.5 (1/3)	-0.7 ± 1.4 (1/3)	-0.9 ± 3.3 (1/3)
	$A-D$	3.5 ± 1.9 (3/3)	2.0 ± 2.6 (3/3)	3.3 ± 3.0 (3/3)	1.7 ± 3.5 (2/3)
Humanoid	$D-E$	17.8 ± 49.8 (2/3)	-44.7 ± 126.3 (1/3)	-19.3 ± 124.0 (2/3)	-168.8 ± 232.8 (1/3)
	$A-D$	49.1 ± 82.0 (3/3)	-392.8 ± 474.5 (1/3)	-17.9 ± 234.1 (1/3)	-178.1 ± 342.9 (1/3)

Table E8: Paired TD3 effects of the detached membrane readout and additional direct credit at 1 Mi. Entries give the mean ± sample SD across paired training-run differences in task-native return units. Each parenthetical $k/3$ counts positive differences out of three pairs. Outcomes use the training-monitoring grid; Appendix C defines the metrics.

Task	Contrast	$\Delta\mathrm{AUC}_{0:1}$	ΔR_1	$\Delta\mathrm{Tail}_{.75:1}$	$\Delta\mathrm{Best}_{0:1}$
HC	$D-E$	439.8 ± 1004.8 (2/3)	335.3 ± 1553.2 (2/3)	360.6 ± 1536.9 (2/3)	351.4 ± 1605.4 (2/3)
	$A-D$	1175.7 ± 1234.3 (2/3)	1942.8 ± 1800.0 (2/3)	1769.9 ± 1768.6 (2/3)	1902.5 ± 1942.2 (2/3)
Walker	$D-E$	16.2 ± 3.5 (3/3)	145.9 ± 231.9 (2/3)	100.6 ± 170.4 (2/3)	149.1 ± 211.5 (2/3)
	$A-D$	-30.4 ± 21.7 (0/3)	-139.5 ± 172.4 (1/3)	-115.8 ± 132.2 (0/3)	-131.6 ± 173.8 (0/3)
Hopper	$D-E$	156.6 ± 131.1 (3/3)	229.7 ± 166.6 (3/3)	231.8 ± 214.1 (3/3)	300.2 ± 345.9 (3/3)
	$A-D$	-51.8 ± 104.0 (1/3)	-65.4 ± 187.3 (1/3)	-93.6 ± 219.2 (2/3)	-157.8 ± 279.1 (1/3)
Ant	$D-E$	215.3 ± 115.2 (3/3)	645.7 ± 629.0 (3/3)	405.5 ± 427.9 (3/3)	250.3 ± 455.7 (2/3)
	$A-D$	228.2 ± 299.2 (2/3)	165.3 ± 949.7 (2/3)	199.4 ± 745.2 (2/3)	192.8 ± 961.0 (2/3)
Swimmer	$D-E$	4.1 ± 3.5 (3/3)	8.4 ± 4.9 (3/3)	7.0 ± 5.6 (3/3)	-0.4 ± 9.1 (2/3)
	$A-D$	0.7 ± 2.6 (2/3)	1.6 ± 4.6 (1/3)	0.3 ± 5.5 (2/3)	1.2 ± 3.6 (2/3)
Humanoid	$D-E$	61.9 ± 68.5 (2/3)	98.9 ± 254.9 (2/3)	99.6 ± 164.2 (2/3)	92.8 ± 222.1 (2/3)
	$A-D$	22.4 ± 19.7 (3/3)	56.9 ± 134.0 (2/3)	17.8 ± 19.6 (2/3)	34.1 ± 114.8 (1/3)

E.5 Final-policy behavior

The full 20-reset behavior extraction covers all 108 E/D/A off-policy models and 2160 episodes, split evenly between SAC and TD3. Tables E9 and E10 report velocity, action saturation, episode length and actual termination/full-horizon counts. Each row contains 60 episodes nested within models for three training seeds; the displayed mean and SD use those three model means.

Table E9: SAC final-policy behavior at 1 Mi on 20 common new reset seeds per model. Each row has three training seeds and 60 valid evaluation episodes for every metric. Values are means ± sample SD across the three model means, not across pooled episodes. v_x is episode-mean velocity; Sat. is the off-policy evaluator's action saturation in percent (Appendix C.2). Term. and Full are actual terminated and full-horizon episode counts out of 60. These are new evaluations, separate from historical monitoring behavior.

Task	Condition	v_x (m/s)	Sat. (%)	Length	Term.	Full
HC	E	2.23 ± 0.49	2.6 ± 2.5	1000.0 ± 0.0	0/60	60/60
	D	2.43 ± 0.80	3.5 ± 1.2	1000.0 ± 0.0	0/60	60/60
	A	3.64 ± 1.55	1.2 ± 0.5	1000.0 ± 0.0	0/60	60/60
Walker	E	1.84 ± 0.16	5.2 ± 0.5	178.7 ± 13.3	60/60	0/60
	D	2.39 ± 0.14	2.4 ± 1.3	527.8 ± 147.9	57/60	3/60
	A	1.76 ± 0.20	3.1 ± 0.6	201.9 ± 48.5	60/60	0/60
Hopper	E	1.60 ± 0.02	2.3 ± 0.7	120.9 ± 9.2	60/60	0/60
	D	1.84 ± 0.16	3.6 ± 2.0	153.0 ± 39.7	60/60	0/60
	A	1.85 ± 0.12	3.6 ± 0.9	129.7 ± 10.3	60/60	0/60
Ant	E	2.12 ± 0.54	0.4 ± 0.4	852.0 ± 81.4	15/60	45/60
	D	3.15 ± 0.10	0.1 ± 0.3	971.7 ± 29.7	3/60	57/60
	A	2.93 ± 0.68	0.0 ± 0.0	932.4 ± 38.7	6/60	54/60
Swimmer	E	0.03 ± 0.00	1.0 ± 0.4	1000.0 ± 0.0	0/60	60/60
	D	0.04 ± 0.00	0.7 ± 0.1	1000.0 ± 0.0	0/60	60/60
	A	0.04 ± 0.00	1.2 ± 0.7	1000.0 ± 0.0	0/60	60/60
Humanoid	E	0.47 ± 0.08	5.3 ± 0.8	199.3 ± 9.3	60/60	0/60
	D	0.18 ± 0.08	3.9 ± 1.1	214.2 ± 22.1	60/60	0/60
	A	0.37 ± 0.14	3.9 ± 1.1	152.1 ± 42.9	60/60	0/60

Table E10: TD3 final-policy behavior at 1 Mi on 20 common new reset seeds per model. Each row has three training seeds and 60 valid evaluation episodes for every metric. Values are means ± sample SD across the three model means, not across pooled episodes. v_x is episode-mean velocity; Sat. is the off-policy evaluator's action saturation in percent (Appendix C.2). Term. and Full are actual terminated and full-horizon episode counts out of 60. These are new evaluations, separate from historical monitoring behavior.

Task	Condition	v_x (m/s)	Sat. (%)	Length	Term.	Full
HC	E	2.55 ± 1.16	25.2 ± 13.6	1000.0 ± 0.0	0/60	60/60
	D	2.92 ± 1.84	56.3 ± 9.2	1000.0 ± 0.0	0/60	60/60
	A	4.91 ± 0.31	64.9 ± 2.7	1000.0 ± 0.0	0/60	60/60
Walker	E	1.61 ± 0.21	57.5 ± 12.6	177.1 ± 22.5	60/60	0/60
	D	1.91 ± 0.41	51.7 ± 6.2	206.2 ± 59.9	60/60	0/60
	A	1.66 ± 0.23	50.3 ± 2.1	176.9 ± 24.5	60/60	0/60
Hopper	E	1.26 ± 0.18	44.1 ± 17.2	100.7 ± 26.7	60/60	0/60
	D	1.81 ± 0.32	35.7 ± 6.9	157.7 ± 33.2	60/60	0/60
	A	1.50 ± 0.22	35.0 ± 7.4	155.7 ± 21.4	60/60	0/60
Ant	E	1.42 ± 0.68	1.6 ± 2.5	723.5 ± 123.8	20/60	40/60
	D	1.69 ± 0.85	3.1 ± 5.2	738.1 ± 37.1	21/60	39/60
	A	2.19 ± 1.03	0.7 ± 0.6	792.5 ± 199.0	20/60	40/60
Swimmer	E	0.03 ± 0.01	73.9 ± 22.2	1000.0 ± 0.0	0/60	60/60
	D	0.04 ± 0.00	90.4 ± 7.1	1000.0 ± 0.0	0/60	60/60
	A	0.04 ± 0.00	73.1 ± 6.9	1000.0 ± 0.0	0/60	60/60
Humanoid	E	0.21 ± 0.06	80.9 ± 3.6	122.6 ± 43.7	60/60	0/60
	D	0.12 ± 0.13	68.4 ± 4.3	128.4 ± 11.3	60/60	0/60
	A	0.32 ± 0.15	43.4 ± 4.5	144.9 ± 10.5	60/60	0/60

E.6 Recorded resources and state-reconstruction probes

Resource and scheduled-probe records cover the original E/A runs, not the later D extension or the common 20-reset reevaluation. Wall time includes setup, training, evaluation, diagnostics and export. Runs shared a GPU under varying concurrency, so these measurements do not establish an algorithmic speedup. Separately timed evaluations cover 16 of the 17 scheduled checkpoints; the untimed initial evaluation is omitted, not assigned zero duration. Memory is the largest logged per-process PyTorch allocated-memory peak, not whole-device memory, reserved memory, host RSS or an HP201 measurement. A positive numerical A–E difference in termination, time or memory is not labeled a benefit.

Table E11: Recorded off-policy E/A training-resource measurements, mean ± sample SD across three training seeds. Wall time, allocated-memory and timed-evaluation scopes are stated in the text; shared-GPU timings are not speedup measurements.

Rule	Task	Condition	Wall (h)	Alloc. (GiB)	Eval time (s)	Cov.
SAC	HalfCheetah	E	0.57 ± 0.25	0.0280 ± 0.0000	16.5 ± 12.9	16/17
SAC	HalfCheetah	A	0.56 ± 0.27	0.0280 ± 0.0000	15.7 ± 12.9	16/17
SAC	Walker2d	E	0.74 ± 0.26	0.0280 ± 0.0000	5.8 ± 3.0	16/17
SAC	Walker2d	A	0.74 ± 0.26	0.0280 ± 0.0000	6.9 ± 4.4	16/17
SAC	Hopper	E	0.75 ± 0.22	0.0278 ± 0.0000	4.0 ± 1.9	16/17
SAC	Hopper	A	0.76 ± 0.22	0.0278 ± 0.0000	4.1 ± 1.9	16/17
SAC	Ant	E	0.75 ± 0.22	0.0307 ± 0.0000	37.8 ± 15.5	16/17
SAC	Ant	A	0.76 ± 0.23	0.0307 ± 0.0000	36.9 ± 15.4	16/17
SAC	Swimmer	E	0.74 ± 0.22	0.0277 ± 0.0000	28.6 ± 13.8	16/17
SAC	Swimmer	A	0.74 ± 0.23	0.0277 ± 0.0000	28.0 ± 13.3	16/17
SAC	Humanoid	E	0.79 ± 0.24	0.0376 ± 0.0000	7.1 ± 2.8	16/17
SAC	Humanoid	A	0.79 ± 0.24	0.0376 ± 0.0000	6.9 ± 1.6	16/17
TD3	HalfCheetah	E	0.60 ± 0.20	0.0273 ± 0.0000	22.7 ± 12.1	16/17
TD3	HalfCheetah	A	0.59 ± 0.22	0.0273 ± 0.0000	22.3 ± 12.4	16/17
TD3	Walker2d	E	0.73 ± 0.27	0.0273 ± 0.0000	6.2 ± 2.1	16/17
TD3	Walker2d	A	0.73 ± 0.27	0.0273 ± 0.0000	5.9 ± 2.5	16/17
TD3	Hopper	E	0.70 ± 0.25	0.0270 ± 0.0000	3.0 ± 0.8	16/17
TD3	Hopper	A	0.70 ± 0.25	0.0271 ± 0.0000	4.2 ± 1.7	16/17
TD3	Ant	E	0.71 ± 0.25	0.0299 ± 0.0000	32.9 ± 15.2	16/17
TD3	Ant	A	0.71 ± 0.25	0.0300 ± 0.0000	35.5 ± 15.4	16/17
TD3	Swimmer	E	0.70 ± 0.26	0.0269 ± 0.0000	28.6 ± 14.1	16/17
TD3	Swimmer	A	0.70 ± 0.26	0.0270 ± 0.0000	27.9 ± 13.8	16/17
TD3	Humanoid	E	0.69 ± 0.25	0.0367 ± 0.0000	4.7 ± 0.9	16/17
TD3	Humanoid	A	0.68 ± 0.26	0.0368 ± 0.0000	5.5 ± 2.0	16/17

Update counts and local state reconstruction. Table E12 separates recorded update counts from local burn-in observations. Each run has 16 scheduled probes comparing 16- and 32-step reconstruction at 64 deterministic replay positions. SAC records a conditional Gaussian KL; TD3 uses unit-action differences, for which that Gaussian KL is not applicable. They have different units and are not pooled. These probes concern online Actor-state reconstruction, not target-state correctness or a behavioral burn-in intervention. Recorded ownership labels and counters also do not replace targeted gradient/RNG tests, full parameter/MAC accounting or independent held-out evaluation.

Table E12: Off-policy E/A learner iterations, parameter-group steps and online Actor burn-in probes. A learner iteration can include more than one optimizer step. A dash means unlogged, not zero. Probe summaries compare burn-in lengths 16 and 32; SAC reports conditional Gaussian KL and TD3 the RMS difference between normalized actions.

Rule	Learner updates	Encoder updates	Q updates	Actor updates	Target updates
SAC	129,822	129,822	—	—	—
TD3	129,822	129,822	129,822	64,911	64,911

Rule	Task	Burn-in	Probe measure	n	E	A
SAC	HalfCheetah	16/32	Conditional Gaussian KL	3	0.00000 ± 0.00000	0.00000 ± 0.00000
SAC	Walker2d	16/32	Conditional Gaussian KL	3	0.01864 ± 0.00303	0.01819 ± 0.00270
SAC	Hopper	16/32	Conditional Gaussian KL	3	0.00001 ± 0.00002	0.00012 ± 0.00006
SAC	Ant	16/32	Conditional Gaussian KL	3	0.00000 ± 0.00000	0.00000 ± 0.00000
SAC	Swimmer	16/32	Conditional Gaussian KL	3	0.00990 ± 0.00933	0.01130 ± 0.00321
SAC	Humanoid	16/32	Conditional Gaussian KL	3	0.00000 ± 0.00000	0.00000 ± 0.00000
TD3	HalfCheetah	16/32	Unit-action RMS difference	3	0.00248 ± 0.00130	0.00085 ± 0.00106
TD3	Walker2d	16/32	Unit-action RMS difference	3	0.02908 ± 0.00923	0.02730 ± 0.00441
TD3	Hopper	16/32	Unit-action RMS difference	3	0.00020 ± 0.00003	0.00123 ± 0.00057
TD3	Ant	16/32	Unit-action RMS difference	3	0.00033 ± 0.00033	0.00025 ± 0.00028
TD3	Swimmer	16/32	Unit-action RMS difference	3	0.01689 ± 0.01575	0.03315 ± 0.00895
TD3	Humanoid	16/32	Unit-action RMS difference	3	0.00000 ± 0.00000	0.00000 ± 0.00000

Each training seed: 16 scheduled probes × 64 deterministic replay positions.

F Extended System Comparisons

The MuJoCo-v5 system comparison uses five common training seeds per method and task. MCrPPO, ANN-PPO and SNN-PPO (adapted) each contribute 30 runs, for 90 runs in total. Table F1 records inclusion and exclusions, and Table F7 retains the original MCrPPO run-set summaries. These are descriptive comparisons between complete systems with different learner protocols, separate from the matched AMC interventions in Appendix D.

F.1 MuJoCo-v5 protocol and included runs

The common completed training replicates are S1–S5. Each method contributes five runs per task to Figure 17 and Table F4; the original MCrPPO run set is reported separately in Table F7.

The common comparison set was assembled after development; Table F1 documents the included and excluded runs. The additional ANN runs each contain 321 evaluations and 40960 optimizer steps; they are independently trained ANN checkpoints, not converted SNNs. Additional MCrPPO runs retain the learner, while additional ANN runs use a separate execution stack. Their configuration and execution differences are described below. Table B4 reports parameter and dense linear-operation counts for the full Actor–Critic and Actor-only computation. These counts do not measure total training cost. Cross-version MuJoCo comparisons and off-policy SAC/TD3 results retain separate protocols (Haarnoja et al., 2018a; Fujimoto et al., 2018); they are not pooled into the PPO system ranking.

Table F1: Accounting for training runs. S1–S5 are presentation aliases for the common comparison set; X1 denotes the excluded original identifier. Completed, stopped and included training runs are distinguished. Global launch counts for ANN-PPO and SNN-PPO are not recoverable from the available records (NR).

Method	Earlier record set	Main included	Inclusion and exclusions
MCrPPO	Original target: 28 complete, 2 stopped	30	Six S5 runs added. X1's four completed runs remain in the original-run table; its Ant/Humanoid stops are not imputed.
ANN-PPO	26 complete; total starts NR	30	Four independent S5 runs added on Hopper, Ant, Swimmer and Humanoid. X1 is excluded; the partial Hopper predecessor is not concatenated.
SNN-PPO (adapted)	30 complete; total starts NR	30	Same training implementation and network configuration across two orchestration versions. Other configurations and incomplete comparisons are outside this set.

MCrPPO and ANN use Gymnasium 1.3.0 and MuJoCo 3.9.0 with 1000-step limits, deterministic monitoring and the same reset/grid convention. ANN uses a clipped Gaussian and separate 64×64 networks, while MCrPPO uses its bounded Gaussian and persistent shared SNN. Normalization, value objectives, scheduling, gradient clipping and execution stacks differ. ANN runs with replicate S2 resumed Ant and Humanoid after discarding 716800 and 573440 environment steps, respectively. Their retained counters reach $20\,\mathrm{Mi}$; their physical training totals exceed that budget by 0.684 and 0.547 Mi. Neither uninterrupted execution nor equal total compute is claimed.

The SNN-PPO (adapted) comparator extends the SpikeGym implementation to six MuJoCo-v5 tasks with one 512-unit spiking hidden layer and a rectangular surrogate. Its training schedule uses 256 environments, 16 rollout steps, four PPO epochs and two minibatches per epoch, giving 4096 environment steps and eight optimizer updates per rollout. These are the settings evaluated here rather than upstream performance claims. This comparator is distinct from E and from the independent fixed-LR SNN-PPO feasibility comparison; incomplete or differently configured comparisons are reported separately.

Table F2: Comparator roles and evidence scopes. E/D are internal AMC controls; ANN-PPO and SNN-PPO (adapted) are system references. The ANN comparison in Table F4 matches rollout size and optimizer-update count; the independent single-seed SNN-PPO run is a feasibility reference.

Method	Actor/Critic	Protocol role	Params/MAC	Evidence
ANN-PPO, matched update count	ANN/ANN	Same rollout size and optimizer-update count	Not matched	30 completed 20 Mi runs; 5 training seeds/task
SNN-PPO (adapted)	SNN/SNN	Single512, rectangular surrogate; matched rollout/update count	Not matched	30 completed 20 Mi runs; 5 training seeds/task
MCrPPO, system comparison	SNN/SNN	Descriptive system capability	Table B4	30 completed 20 Mi runs
AMC (A), PPO control	SNN/SNN	Complete AMC pathway	A/E differ by 512 parameters	18 completed 10 Mi runs
Event-only (E), PPO control	SNN/SNN	AMC-free internal baseline	Table B4	18 completed 10 Mi runs
Detached (D), PPO control	SNN/SNN	Direct membrane edge disabled	Same count as A	18 completed 10 Mi runs
Independent fixed-LR SNN-PPO	SNN/SNN	Auxiliary plain-system feasibility only	Not matched	6 tasks, 1 training seed, 1 Mi

F.2 COMMON-RESET FINAL-POLICY SYSTEM OUTCOMES

MCrPPO has higher mean acquisition AUC than SNN-PPO (adapted) on five of six tasks. In the common 20-reset final-policy evaluation, it has higher mean return than the adapted SNN on HalfCheetah, Hopper, Swimmer and Humanoid, and than ANN-PPO on HalfCheetah, Swimmer and Humanoid. The MCrPPO–SNN final-policy difference is positive for all five training pairs on HalfCheetah and Swimmer, but for only two Hopper pairs and three Humanoid pairs. Walker remains the clearest negative case, and Ant has no mean final-policy advantage over either reference. Table F3 reports the full 20-reset mean returns and sample SDs for each method in each task's reward units.

Distribution figure. Figure 4 shows training-curve $\mathrm{AUC}_{0:10}$ and $\mathrm{Best}_{0:20}$ together with the full 20-reset final-policy return for each common training seed, without pooling tasks. Each method has five observed points and a mean stroke, identified by the common method legend. Small vertical offsets separate coincident points without changing their return coordinates. Within a task, metric panels retain common raw-return limits. The complete table reports all six metrics and sample SDs. $\mathrm{Tail}_{18:20}$ is recomputed from the 33 inclusive raw checkpoints, including both endpoints.

Table F3: Final-policy system comparison at 20 Mi, $n = 5$ training seeds. Fixed final checkpoints are evaluated on the same 20 reset seeds, separately from the historical monitoring curves. Entries are means $\pm$ sample SD across training-seed model means; each model mean uses 20 evaluation episodes. SNN denotes the adapted SpikeGym reference; these are system-level comparisons. Boldface marks the highest mean within each task and metric.

Task	MCrPPO	ANN-PPO	SNN-PPO (adapted)
HC	**6108.4** $\pm$ 1004.7	1708.0 $\pm$ 158.6	3289.6 $\pm$ 1906.3
Walker	2639.8 $\pm$ 1732.8	**4450.0** $\pm$ 247.8	2949.5 $\pm$ 1430.2
Hopper	1909.8 $\pm$ 831.0	**2803.7** $\pm$ 1261.0	1819.5 $\pm$ 1081.6
Ant	5433.6 $\pm$ 226.6	**6433.2** $\pm$ 263.1	5484.4 $\pm$ 364.7
Swimmer	**66.1** $\pm$ 21.6	43.5 $\pm$ 7.4	35.1 $\pm$ 9.1
Humanoid	**6863.8** $\pm$ 607.0	6831.9 $\pm$ 134.6	6827.8 $\pm$ 314.3

F.3 LEARNING-CURVE OUTCOMES

The historical monitoring summaries retain a different view: Walker's learning-curve AUC, endpoint and best-checkpoint MCrPPO–SNN mean differences are negative, while Ant combines positive mean AUC with negative endpoint and best differences. Reported percentages are ratios of method means rather than means of percentages computed for each training seed. Table F4 gives the method means, and Table F5 expresses them relative to ANN-PPO. Figure 17 shows the training trajectories, while Table F6 reports the paired MCrPPO–SNN-PPO contrasts.

Table F4: Six-metric training-monitoring system comparison over five training seeds. Each method/task uses the same training seeds; entries show the mean with sample SD below. AUC is normalized by the stated integration-window length, $\text{Tail}_{18:20}$ uses the inclusive 33-point grid, and $\text{Best}_{0:20}$ is the highest evaluation score observed during training. Boldface marks the highest mean within each task and metric.

Task	Method	n	$\text{AUC}_{0:10}$	R_{10}	$\text{AUC}_{10:20}$	$\text{Tail}_{18:20}$	R_{20}	$\text{Best}_{0:20}$
HC	MCrPPO	5	**4508.2** ±794.7	**5846.5** ±995.1	**6014.8** ±932.4	**6134.2** ±977.3	**5606.5** ±1219.3	**6468.3** ±1007.7
	ANN-PPO	5	1401.1 ±127.0	1682.2 ±177.2	1696.7 ±171.5	1705.5 ±165.5	1621.7 ±237.3	1757.6 ±174.5
	SNN-PPO (adapted)	5	2191.1 ±895.5	2889.4 ±1479.5	3093.2 ±1701.9	3234.3 ±1842.5	3267.8 ±1883.5	3347.6 ±1939.9
Walker	MCrPPO	5	1442.9 ±599.0	2194.3 ±1262.2	2381.4 ±1526.5	2525.2 ±1659.5	2597.0 ±1701.2	3483.7 ±1461.8
	ANN-PPO	5	**2861.7** ±313.5	**3948.9** ±535.9	**3973.7** ±307.4	**4194.1** ±289.1	**4306.5** ±537.3	**4589.4** ±242.5
	SNN-PPO (adapted)	5	2071.0 ±708.1	2810.3 ±1116.8	3002.4 ±1073.7	3135.7 ±1133.5	2773.6 ±1302.0	3961.2 ±1478.6
Hopper	MCrPPO	5	**1704.7** ±615.9	1890.6 ±734.3	1944.5 ±691.5	2056.7 ±743.6	1922.6 ±917.5	3138.2 ±312.7
	ANN-PPO	5	1354.2 ±751.2	**2272.5** ±1390.0	**2529.5** ±1133.3	**2568.1** ±1156.0	**2830.3** ±1279.7	2983.3 ±1091.0
	SNN-PPO (adapted)	5	1638.3 ±620.4	2193.1 ±489.7	1863.2 ±612.3	1673.1 ±984.3	1775.6 ±1101.3	**3258.9** ±104.6
Ant	MCrPPO	5	3104.3 ±141.5	4644.9 ±440.9	5218.5 ±108.0	5502.0 ±65.0	5310.8 ±568.6	5799.6 ±59.7
	ANN-PPO	5	**3671.9** ±100.7	**6029.6** ±160.2	**6117.2** ±86.6	**6362.1** ±200.2	**6502.6** ±382.4	**6807.8** ±146.1
	SNN-PPO (adapted)	5	2984.5 ±218.0	5051.5 ±399.1	5319.6 ±152.9	5509.5 ±186.5	5343.4 ±1026.1	5966.3 ±133.6
Swimmer	MCrPPO	5	**30.9** ±5.5	**46.0** ±8.0	**57.1** ±15.7	**64.6** ±21.0	**65.7** ±21.9	**71.1** ±19.6
	ANN-PPO	5	24.0 ±4.5	36.1 ±6.1	39.4 ±7.6	43.2 ±8.5	44.5 ±8.9	50.2 ±3.3
	SNN-PPO (adapted)	5	22.9 ±3.0	30.2 ±5.8	31.2 ±6.2	33.2 ±6.0	33.1 ±6.0	42.1 ±5.3
Humanoid	MCrPPO	5	**4987.9** ±226.3	**6271.7** ±603.9	6602.7 ±185.2	**6997.5** ±225.8	6918.8 ±611.5	**7329.1** ±204.5
	ANN-PPO	5	4733.3 ±642.4	6156.7 ±487.5	**6636.5** ±111.0	6734.2 ±150.9	**6930.5** ±160.3	7034.2 ±221.3
	SNN-PPO (adapted)	5	4538.3 ±90.0	6238.3 ±223.8	6510.0 ±122.4	6737.7 ±328.3	6535.2 ±717.6	7171.5 ±125.1

Table F5: Training-monitoring scores relative to ANN-PPO in Table F4. Each task entry is $100(\overline{Y}_{\text{method}}/\overline{Y}_{\text{ANN-PPO}} - 1)$, calculated from the displayed method means. The ANN-PPO reference is 0% and is omitted. The final rows average the six unrounded task percentages equally for each metric; they do not pool task returns. Bold marks the highest positive gain against ANN-PPO.

Task	Method	$\text{AUC}_{0:10}$	R_{10}	$\text{AUC}_{10:20}$	$\text{Tail}_{18:20}$	R_{20}	$\text{Best}_{0:20}$
HC	MCrPPO	**+221.8%**	**+247.6%**	**+254.5%**	**+259.7%**	**+245.7%**	**+268.0%**
	SNN-PPO (adapted)	+56.4%	+71.8%	+82.3%	+89.6%	+101.5%	+90.5%
Walker	MCrPPO	-49.6%	-44.4%	-40.1%	-39.8%	-39.7%	-24.1%
	SNN-PPO (adapted)	-27.6%	-28.8%	-24.4%	-25.2%	-35.6%	-13.7%
Hopper	MCrPPO	**+25.9%**	-16.8%	-23.1%	-19.9%	-32.1%	+5.2%
	SNN-PPO (adapted)	+21.0%	-3.5%	-26.3%	-34.9%	-37.3%	**+9.2%**
Ant	MCrPPO	-15.5%	-23.0%	-14.7%	-13.5%	-18.3%	-14.8%
	SNN-PPO (adapted)	-18.7%	-16.2%	-13.0%	-13.4%	-17.8%	-12.4%
Swimmer	MCrPPO	**+28.8%**	**+27.4%**	**+44.9%**	**+49.5%**	**+47.6%**	**+41.6%**
	SNN-PPO (adapted)	-4.6%	-16.3%	-20.8%	-23.1%	-25.6%	-16.1%
Humanoid	MCrPPO	**+5.4%**	**+1.9%**	-0.5%	**+3.9%**	-0.2%	**+4.2%**
	SNN-PPO (adapted)	-4.1%	+1.3%	-1.9%	+0.1%	-5.7%	+2.0%
Six-task mean	MCrPPO	**+36.1%**	**+32.1%**	**+36.8%**	**+40.0%**	**+33.8%**	**+46.7%**
	SNN-PPO (adapted)	+3.7%	+1.4%	-0.7%	-1.2%	-3.4%	+9.9%

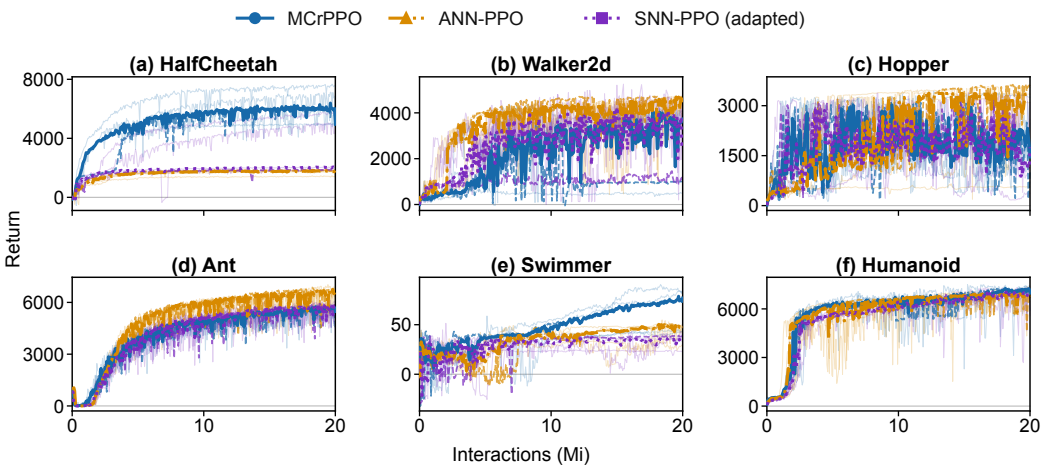

Figure 17: All 90 training-run trajectories in the system comparison over 20 Mi. Thin lines show the 30 runs per method and thick lines their pointwise medians; dashed lines mark additional replicate S5. Each method/task has five training seeds, and trajectories are shown on their recorded grids without smoothing.

Table F6: Paired training-monitoring comparison of MCrPPO and SNN-PPO (adapted) over five training seeds. Method-specific means and MCrPPO-minus-SNN Δ values are reported as mean $\pm$ sample SD; Pos. counts positive paired training-run differences and $\Delta\% = 100(\overline{R}_{\mathrm{MCrPPO}}/\overline{R}_{\mathrm{SNN}} - 1)$. Boldface marks the highest mean within each task and metric.

Task	Metric	MCrPPO	SNN-PPO (adapted)	Paired Δ	Pos.	$\Delta\%$
HC	AUC0:10	**4508.2** ±794.7	2191.1 ±895.5	2317.1 ±316.4	5/5	+105.7%
	R20	**5606.5** ±1219.3	3267.8 ±1883.5	2338.7 ±1648.2	4/5	+71.6%
	Best0:20	**6468.3** ±1007.7	3347.6 ±1939.9	3120.7 ±1159.5	5/5	+93.2%
Walker	AUC0:10	1442.9 ±599.0	**2071.0** ±708.1	-628.0 ±702.2	1/5	-30.3%
	R20	2597.0 ±1701.2	**2773.6** ±1302.0	-176.5 ±972.9	1/5	-6.4%
	Best0:20	3483.7 ±1461.8	**3961.2** ±1478.6	-477.5 ±2276.8	2/5	-12.1%
Hopper	AUC0:10	**1704.7** ±615.9	1638.3 ±620.4	66.4 ±439.8	2/5	+4.1%
	R20	**1922.6** ±917.5	1775.6 ±1101.3	147.0 ±1377.3	2/5	+8.3%
	Best0:20	3138.2 ±312.7	**3258.9** ±104.6	-120.8 ±271.1	2/5	-3.7%
Ant	AUC0:10	**3104.3** ±141.5	2984.5 ±218.0	119.9 ±259.5	4/5	+4.0%
	R20	5310.8 ±568.6	**5343.4** ±1026.1	-32.6 ±935.5	2/5	-0.6%
	Best0:20	5799.6 ±59.7	**5966.3** ±133.6	-166.8 ±162.4	0/5	-2.8%
Swimmer	AUC0:10	**30.9** ±5.5	22.9 ±3.0	8.0 ±5.8	4/5	+34.9%
	R20	**65.7** ±21.9	33.1 ±6.0	32.7 ±24.4	5/5	+98.8%
	Best0:20	**71.1** ±19.6	42.1 ±5.3	29.0 ±18.2	5/5	+69.0%
Humanoid	AUC0:10	**4987.9** ±226.3	4538.3 ±90.0	449.6 ±248.8	5/5	+9.9%
	R20	**6918.8** ±611.5	6535.2 ±717.6	383.6 ±1089.5	3/5	+5.9%
	Best0:20	**7329.1** ±204.5	7171.5 ±125.1	157.6 ±280.3	4/5	+2.2%

F.4 ORIGINAL TRAINING RUNS

The original MCrPPO run-set summaries are reported separately from the common comparison set.

Table F7: Training-monitoring results for the original fixed-budget MCrPPO runs (HC=HalfCheetah; W2d=Walker2d). Entries are mean ± sample SD over n completed training seeds for $\mathrm{AUC}_{0:10}$, R_{10}, $\mathrm{AUC}_{10:20}$, the inclusive 33-point $\mathrm{Tail}_{18:20}$, and R_{20}.

Task	n	$\mathrm{AUC}_{0:10}$	R_{10}	$\mathrm{AUC}_{10:20}$	$\mathrm{Tail}_{18:20}$	R_{20}
HC	5	4106.5±1501.0	5188.8±2064.2	5232.1±2074.1	5332.9±2132.5	4796.1±1984.9
W2d	5	1694.6±741.0	2482.3±1228.5	2915.1±1371.6	3199.3±1556.0	3346.8±1652.3
Hopper	5	1729.0±621.8	1854.6±739.3	1956.3±692.2	2048.2±751.4	1997.4±841.1
Ant	4	3116.8±160.2	4599.1±495.2	5231.4±120.2	5504.5±74.8	5498.8±442.1
Swimmer	5	29.7±5.2	43.4±7.9	51.9±17.2	58.0±22.7	58.7±23.1
Humanoid	4	4977.0±259.8	6523.9±249.6	6608.0±213.4	6971.8±252.1	6903.0±704.9

F.5 VERSION-ISOLATED ANT-V4 COMPARISON

On true Ant-v4, MCrPPO's mean learning-curve AUC and mean best observed checkpoint return are 64.8% and 13.5% higher, with all three paired training-run differences positive for both. Mean endpoint return is 7.0% higher, with two of three paired training-run differences positive. The original last-four method means are 5492.0 and 5749.8 ($+4.7\%$), with two paired training-run differences positive; the first training-run pair retains endpoint and last-four differences of -50.0 and -39.7. The separately labeled final-quarter ratio of method means is $+14.0\%$, with all three paired training-run differences positive; it does not replace the original last-four metric.

The larger improvement in full-window AUC than in final-policy return indicates that the principal advantage is performance accumulated over training. It does not imply a comparable reduction in wall-clock training time, and the comparison does not isolate AMC from the other differences between the two learning systems.

Table F8: True Ant-v4 training-monitoring comparison over three paired training seeds. Entries give mean ± sample SD; Δ is MCrPPO minus SpikeGym and Pos. counts positive paired training-run differences. AUC is normalized by the 32.768-million-step integration window; Last four is the original endpoint-tail metric and the auxiliary last quarter averages 126 points. Boldface marks the highest mean within each task and metric.

Metric	SpikeGym	MCrPPO	Paired Δ	Pos.
$\mathrm{AUC}_{0:32.768}$	3118.4±226.4	**5138.0**±77.1	2019.6±184.2	3/3
Endpoint $R_{32.768}$	5467.4±125.6	**5850.7**±499.4	383.3±375.4	2/3
$\mathrm{Best}_{0:32.768}$	5555.4±151.8	**6306.4**±126.5	751.0±169.4	3/3
Last four	5492.0±155.3	**5749.8**±410.0	257.8±257.9	2/3
Auxiliary maintenance summary				
Last quarter (126; aux.)	5243.9±167.3	**5975.9**±109.5	732.0±157.0	3/3

The final-policy comparison on 20 common new reset seeds is positive in all three training-run pairs: MCrPPO exceeds the SpikeGym reference under its native training schedule by 477.3 ± 417.8 on average (Table F9). This final-policy result is separate from the original training AUC, monitoring endpoint, last-four and final-quarter summaries above.

Table F9: Ant-v4 final policies after 32,768,000 environment steps, evaluated on 20 common new reset seeds per model. Means and SD are across three training seeds; Δ is paired MCrPPO minus the SpikeGym reference under its native training schedule, and Pos. counts positive paired training-run differences. These are separate from historical monitoring endpoints. Boldface marks the highest mean within each task and metric.

Task	MCrPPO	SpikeGym	Paired Δ	Pos.
Ant-v4	**5934.8** $\pm$ 230.1	5457.5 $\pm$ 217.0	477.3 $\pm$ 417.8	3/3

Table F8 reports all three paired training replicates, S1–S3. Both methods use true Ant-v4 with 27 input coordinates and eight actuators, $32{,}768{,}000$ aggregate training environment steps, and 501 monitoring points separated by 65,536 environment steps. Each evaluation uses five reset seeds. AUC is the trapezoidal return integral divided by this horizon; the last-four mean covers the four checkpoints from 32,571,392 through 32,768,000, not a long retention window. Best is the maximum checkpoint mean rather than a deployed checkpoint selection. Means and sample SDs are computed across training seeds, and paired SDs use MCrPPO-minus-reference differences computed within each training seed. The final-quarter summary uses 126 checkpoints from $24{,}576{,}000$ through $32{,}768{,}000$ interactions, inclusive. Percentages are ratios of method means rather than means of percentages computed for each training seed; no significance claim is made.

The SpikeGym reproduction follows the paper's reported one-layer 512-unit architecture and the native training schedule: 4096 environments, 16 rollout steps, four epochs and two minibatches. The paper-motivated neuron setting is rectangular-surrogate LIF; the checked-in upstream Ant-v4 configuration instead uses four 256-unit layers with a sigmoid surrogate. We therefore identify the reproduced architecture and schedule explicitly, rather than claim an exact paper-best configuration or reproduction of the original software environment.

At 32.768 million environment steps, SpikeGym has 500 rollout/update cycles and 4000 optimizer steps with 32768 samples per logical minibatch; MCrPPO has 8000 cycles and 64000 steps with 2048 samples per logical minibatch. Both represent 131072000 logical sample uses. The step-count ratio is not a compute or sample-reuse ratio. Action distributions, normalization transactions, learning-rate scheduling, timeout bootstrap and software versions differ; the comparison matches environment, training-step budget and evaluation grid and reset seeds. These are system effects, separate from the matched AMC controls.

G Actor-only Deployment and Behavioral Boundaries

G.1 Model selection and paired closed-loop protocol

All 18 selected Actors complete paired GPU/Lynxi HP201 evaluation in the simulated tasks. The study compares the same model on two execution backends; it does not retrain the model or compare the original ANN with its converted SNN. Task-wise HP201/GPU return ratios range from 99.2–104.5% for MCrPPO, 87.9–100.9% for SNN-PPO (adapted), and 95.3–103.8% for converted ANN–SNN. Uncertainty is conditional on each selected model. The wider intervals, including 88.0–114.0% for MCrPPO HalfCheetah, do not establish numerical or behavioral equivalence.

Paired closed-loop episodes share initial task states and model initialization, but may subsequently receive different observations. Fixed-observation replay is a separate numerical diagnostic. The per-call HP201 SDK reset is also distinct from an episode reset: logical recurrent state persists through the explicit state inputs and outputs. Figure 7 depicts these execution boundaries schematically; measured costs are reported in Figure 6 and the following subsections.

Model selection and paired evaluation. The study compares MCrPPO, the SNN-PPO (adapted) baseline and separately converted ANN–SNN policies on six tasks. For each method and task, we evaluate the five final $20\,\mathrm{Mi}$ models (S1–S5) on monitoring resets M1–M5 and select the highest unrounded mean return. Exact ties go to the smaller integer training seed; intermediate checkpoints are not considered. The 30 ANN–SNN models are evaluated after conversion using the same fixed implementation.

ANN–SNN conversion. The converted comparator uses a Tanh/control adaptation of sign-gradient-descent neuronal dynamics (Oh & Lee, 2024). It retains the frozen ANN Actor's two 64-unit Tanh hidden layers and original observation normalization. An analog first-layer current drives two binary hidden-event streams with signed offset correction. Each decision starts from zero decoded hidden states and uses 128 ticks with step size $.15 \times .95^k$, $k = 0, \ldots, 127$. The implementation reconstructs the second-layer current and final affine mean from the decoded states; the host clips the resulting deterministic action to the task bounds. This conversion is an adaptation of the cited dynamics, not a claim of an exact reproduction of its published benchmarks. Selection uses the converted policies under this implementation, as described above.

For converted ANN–SNN models, maximum raw and bounded action discrepancies on 256 fixed inputs are numerical diagnostics. The raw error uses unbounded affine-mean coordinates; the bounded error is divided by the task's action half-range and is dimensionless. The initial rule required both errors to be below .01. This rule was changed after part of the fixed-input results had been observed but before any ANN closed-loop sampling; errors above .01 do not exclude a model. Shape, finite-value, compiled-model identity, required-device execution and deterministic-reset checks remain mandatory. This protocol tests closed-loop preservation rather than a preregistered numerical-tolerance criterion.

Each selected model is evaluated on a same-host GPU and Lynxi HP201 using reset seeds H1–H20 and at most 1000 decisions per episode, giving 720 episodes organized as 360 GPU–HP201 episode pairs. Paired episodes share the initial physical state, weights, preprocessing, recurrent initialization and action interpretation. MCrPPO keeps its bounded tanh action and SNN-PPO (adapted) its raw-action environment interface. For ANN–SNN, both backends execute the same converted SNN rather than the original ANN; this evaluates transfer of the converted representation, while conversion quality is assessed separately. Each episode records return, velocity, displacement, length and natural termination. Decision traces contain observations, neural pre/post states, events, raw/executed actions, rewards and latency.

Return ratio and uncertainty. For one selected model let G_i, N_i be GPU and HP201 returns on reset i. The paired mean difference is $20^{-1} \sum_i (N_i - G_i)$. The HP201/GPU return ratio is

$$\mathcal{R} = 100 \, \frac{\overline{N}}{\overline{G}} \quad \text{if every return is finite and } \overline{G} > 1; \qquad \mathcal{R} = \text{NA otherwise.}$$

The threshold is one raw undiscounted return unit for every task and serves only as a denominator rule. Finite paired differences and their intervals are reported even when retention is undefined, and ratios above 100% are kept. Returns are not averaged across tasks. Uncertainty uses 10000 bootstrap samples of 20 paired reset episodes, preserving GPU/Lynxi HP201 pairing, with 95% percentile intervals and linear quantiles at .025/.975. NumPy PCG64 is initialized by a fixed, documented SeedSequence indexed by $[f, e]$, with zero-based family order MCrPPO, SNN-PPO, converted ANN–SNN and task order HC, Walker, Hopper, Ant, Swimmer, Humanoid. If any ratio replicate has GPU mean at most 1, the full ratio interval is NA and the invalid count is reported; no replicate is discarded or replaced, and difference intervals use all replicates. The intervals quantify reset variation conditional on the selected model, not training-seed or model-selection uncertainty. Neither the 100% reference line nor these intervals define an equivalence margin or equivalence test.

Execution-cost scope. Host-call measurements use the same frozen input sequence on both backends, batch one, synchronized timing and host-transfer-inclusive decision latency. Idle-before, active and idle-after measurements report working power, idle-subtracted power, energy per decision and measurement noise with the same scope. Host/simulator energy is excluded unless separately instrumented. Parameter bytes, device-wide occupancy and process memory remain distinct; dynamic HP201 power is not compared with total GPU power as an energy saving. The earlier GPU power window overlapped an independent training workload and therefore does not measure isolated inference energy. A separate isolated device-event latency benchmark is reported in Appendix G.3; baseline-adjusted dynamic power and energy are measured separately in Appendix G.4. Timing ratios are not used to infer energy. Table G1 reports closed-loop preservation over all 20 paired evaluation reset seeds for each selected model.

Table G1: Closed-loop preservation of selected policies on Lynxi HP201. Each row reports GPU and HP201 mean returns over 20 paired evaluation reset seeds. The return ratio is HP201/GPU, with a 95% bootstrap interval over paired episodes, conditional on the selected model.

Family	Task	Training seed	GPU return	HP201 return	HP201/GPU (95% CI)
MCrPPO	HC	S2	7223.2	7172.1	99.3 [88.0, 114.0]
	Walker	S2	4147.6	4115.3	99.2 [97.9, 100.7]
	Hopper	S1	3250.4	3244.0	99.8 [99.6, 100.1]
	Ant	S1	5670.4	5924.0	104.5 [101.1, 110.0]
	Swimmer	S3	84.8	84.6	99.8 [98.9, 100.7]
	Humanoid	S4	7408.1	7407.5	100.0 [99.8, 100.2]
SNN-PPO (adapted)	HC	S2	5662.7	5711.1	100.9 [99.5, 102.3]
	Walker	S3	4791.0	4213.5	87.9 [74.8, 103.0]
	Hopper	S4	3080.2	2899.8	94.1 [85.8, 101.5]
	Ant	S2	6020.5	5950.7	98.8 [94.8, 102.1]
	Swimmer	S5	37.7	37.8	100.4 [99.9, 100.9]
	Humanoid	S5	7163.8	7154.3	99.9 [99.6, 100.2]
ANN→SNN	HC	S4	1848.4	1848.4	100.0 [98.9, 101.0]
	Walker	S4	4647.6	4650.2	100.1 [100.0, 100.1]
	Hopper	S5	3351.8	3479.7	103.8 [95.2, 114.8]
	Ant	S1	6544.8	6529.5	99.8 [86.4, 117.0]
	Swimmer	S3	43.4	41.4	95.3 [87.4, 104.1]
	Humanoid	S3	7053.9	7314.6	103.7 [98.3, 114.4]

G.2 Fixed-observation numerical fidelity

The six MCrPPO Actors from the deployment comparison are evaluated without changing weights, normalization or compiled programs. The platform comprises an NVIDIA GeForce GTX 1060 6GB and a Lynxi HP201 neuromorphic accelerator card (the SDK's NPU backend). System telemetry reports two KA200 devices on board 0, firmware V1.13.2 and driver 1.17.0. Inference uses device 0; HP201 power refers to the whole card, including the other chip, rather than an isolated chip meter. The model interface remains FP32.

For each Actor, we choose its longest continuous recorded GPU episode, breaking equal-length ties by reset order; all six traces have 1000 decisions. In common-prestate replay, both backends receive the recorded normalized observation and recorded incoming neural state at every step. In sequential replay, the observations remain common but each backend carries its own returned (h, z, o) from logical zero. This separates one-step discrepancies from accumulated state differences without mixing diverging environmental trajectories. An A/B/A test resets logical state between two input orders and repeats A; all six Actors reproduce A exactly on both backends.

Table G2 reports action errors and event mismatches. Action errors are measured after the affine–tanh mapping, in the task's environment-action coordinates, not in latent Gaussian or normalized unit-action coordinates; no division by the action range is applied. Sequential action RMSE ranges from 0.00245 to 0.01295, with maxima from 0.0572 to 0.2245. These are fixed-trace numerical diagnostics, not an equivalence test or new training replicates. The original closed-loop return ratios remain separate evidence.

Table G2: Numerical replay of six fixed MCrPPO Actors on GPU and Lynxi HP201. Each model receives 1000 shared observations. Common-prestate replay supplies the recorded incoming state to both backends; sequential replay carries each backend's own state. Action errors use executed-action coordinates. Event mismatches are counted over 512000 neuron-step coordinates. A/B/A reset checks repeat exactly; no equivalence margin is specified.

Task	Common RMSE	Common max	Sequential RMSE	Sequential max	Mismatches
HC	0.00651	0.14177	0.00683	0.15734	82
Walker	0.00763	0.20072	0.00869	0.20072	121
Hopper	0.00820	0.17992	0.00963	0.20855	110
Ant	0.00596	0.10748	0.00671	0.10748	68
Swimmer	0.00916	0.22453	0.01295	0.22453	134
Humanoid	0.00235	0.05721	0.00245	0.05721	89

G.3 Device-event and host-to-action latency

The isolated device-event benchmark covers twelve selected native MCrPPO and SNN-PPO (adapted) Actors, six converted ANN–SNN Actors and their six corresponding original ANN policies. Across tasks, original ANN CPU means are 0.035–0.038 ms, GPU eager means are 0.081–0.087 ms and GPU Graph means are 0.021–0.022 ms. Native MCrPPO Graph means are 0.031–0.032 ms, versus 0.170–0.182 ms eagerly. Converted ANN–SNN eager means are 15.120–15.766 ms on GPU and 3.910–3.967 ms for HP201 SDK reset plus execute. These values describe different representations and execution modes, not interchangeable ANN costs or a protocol-independent hardware ranking.

Table G3: Isolated inference latency in milliseconds, reported as mean $\pm$ SD across 1024 invocations. Lynxi HP201 values include SDK reset plus execute; CUDA Graph is a separate execution mode, and dashes denote unmeasured combinations.

Task	Representation	GPU eager	CUDA Graph	HP201 total	ANN CPU
HC	MCrPPO	0.182 ± 0.009	0.031 ± 0.002	0.375 ± 0.055	–
	SNN-PPO (adapted)	0.171 ± 0.012	0.031 ± 0.002	0.435 ± 0.073	–
	ANN original	0.081 ± 0.008	0.021 ± 0.003	–	0.035 ± 0.005
	ANN converted	15.274 ± 0.356	–	3.910 ± 0.045	–
Walker	MCrPPO	0.171 ± 0.010	0.031 ± 0.002	0.436 ± 0.033	–
	SNN-PPO (adapted)	0.185 ± 0.025	0.032 ± 0.003	0.373 ± 0.063	–
	ANN original	0.085 ± 0.010	0.021 ± 0.002	–	0.036 ± 0.002
	ANN converted	15.120 ± 0.341	–	3.929 ± 0.060	–
Hopper	MCrPPO	0.173 ± 0.014	0.031 ± 0.003	0.319 ± 0.042	–
	SNN-PPO (adapted)	0.180 ± 0.013	0.032 ± 0.003	0.344 ± 0.082	–
	ANN original	0.083 ± 0.008	0.021 ± 0.001	–	0.037 ± 0.007
	ANN converted	15.398 ± 0.738	–	3.920 ± 0.067	–
Ant	MCrPPO	0.172 ± 0.015	0.031 ± 0.003	0.316 ± 0.052	–
	SNN-PPO (adapted)	0.170 ± 0.010	0.032 ± 0.003	0.352 ± 0.060	–
	ANN original	0.087 ± 0.013	0.021 ± 0.002	–	0.036 ± 0.004
	ANN converted	15.766 ± 0.422	–	3.931 ± 0.067	–
Swimmer	MCrPPO	0.178 ± 0.012	0.031 ± 0.003	0.358 ± 0.101	–
	SNN-PPO (adapted)	0.174 ± 0.011	0.031 ± 0.003	0.420 ± 0.051	–
	ANN original	0.084 ± 0.007	0.021 ± 0.001	–	0.037 ± 0.004
	ANN converted	15.690 ± 0.346	–	3.967 ± 0.071	–
Humanoid	MCrPPO	0.170 ± 0.012	0.032 ± 0.002	0.316 ± 0.036	–
	SNN-PPO (adapted)	0.177 ± 0.012	0.033 ± 0.002	0.402 ± 0.039	–
	ANN original	0.082 ± 0.010	0.022 ± 0.002	–	0.038 ± 0.004
	ANN converted	15.155 ± 0.793	–	3.923 ± 0.036	–

Each mode uses batch one and 32 warmups followed by the same 256 fixed inputs repeated four times, giving 1024 measurements per mode. GPU inputs are resident before timing, and CUDA events bracket the Actor call on the current stream. HP201 device events bracket the required SDK reset-and-execute sequence. Normalization, IPC, H2D/D2H, environment execution, loading and

compilation are outside these intervals. The interval includes device-stream scheduling rather than summing isolated kernel durations.

The host is a GTX1060/HP201 system. The benchmark excludes competing GPU compute jobs and records occupancy before, between and after model runs. Native and original ANN Actors are measured both eagerly and with CUDA Graph replay. Captured outputs must match eager outputs on every frozen input; graph input staging precedes the timed interval. HP201 SDK reset, execute and total durations are measured directly by three device events rather than inferred by subtraction. Original ANN CPU measurements are single-thread host-forward times and are not device-event measurements.

The original ANN weights are checked against the selected policy; reconstructed and reference forward passes agree on all frozen inputs within the same software environment. Rounding differences between software environments are recorded separately. Converted ANN–SNN retains 128 ticks and the unchanged eager finite-input check, including its synchronization within the measured forward. An optional traced variant exhausted host memory and is excluded from the canonical eager measurement. The benchmark contains 66 measured modes and omits 12 optional converted variants. Energy is not inferred from latency ratios, and earlier non-isolated event windows are excluded.

The host-to-action measurement uses the same recorded observation/prestate stream for GPU eager, CUDA Graph and HP201. Each task/mode has 32 warmups followed by five rounds of 1024 batch-one calls. Timing begins before normalization and state assembly and ends after output synchronization, transfer and action mapping; environment stepping, loading and compilation are excluded. Graph output is checked against eager output before measurement. The pooled mean ranges are 0.339–0.354 ms for GPU eager, 0.200–0.215 ms for CUDA Graph and 0.534–0.608 ms for HP201 (Table G4). These measurements include more of the execution path than the device-event measurements in Figure 6(c). Panel (d) shows pooled P50/P95/P99; these measurements do not supply a simulator-inclusive deadline test.

Table G4: Host-to-action latency in milliseconds for MCrPPO. Means and sample SDs summarize five round means (1024 calls each); P50/P95/P99 summarize all 5120 calls. Timing includes normalization, state assembly, transfers, synchronization and action mapping, but excludes environment simulation. These are distinct from device-event timings.

Task	Mode	Mean ± SD	P50	P95	P99
HC	GPU eager	0.339±0.005	0.330	0.382	0.432
	CUDA Graph	0.200±0.003	0.194	0.228	0.263
	Lynxi HP201	0.568±0.036	0.549	0.680	0.745
Walker	GPU eager	0.344±0.005	0.332	0.405	0.483
	CUDA Graph	0.201±0.004	0.194	0.244	0.297
	Lynxi HP201	0.579±0.030	0.575	0.638	0.781
Hopper	GPU eager	0.351±0.003	0.343	0.398	0.440
	CUDA Graph	0.205±0.002	0.200	0.236	0.257
	Lynxi HP201	0.534±0.059	0.523	0.624	0.660
Ant	GPU eager	0.354±0.007	0.344	0.399	0.491
	CUDA Graph	0.206±0.002	0.202	0.239	0.266
	Lynxi HP201	0.564±0.036	0.564	0.644	1.087
Swimmer	GPU eager	0.340±0.004	0.332	0.382	0.410
	CUDA Graph	0.204±0.002	0.199	0.237	0.267
	Lynxi HP201	0.576±0.094	0.539	0.951	1.000
Humanoid	GPU eager	0.351±0.007	0.343	0.394	0.415
	CUDA Graph	0.215±0.002	0.209	0.250	0.279
	Lynxi HP201	0.608±0.027	0.603	0.703	0.774

G.4 Baseline-adjusted dynamic power and energy

The same six Actors and three execution modes are measured in three windows each, for 54 completed windows. Each window contains 15 s idle-before, 30 s active and 15 s idle-after. Modes rotate across repetitions. All modes use the same frozen input/prestate stream and the task's nominal control period: 50 ms for HalfCheetah/Ant, 8 ms for Walker/Hopper, 40 ms for Swimmer and 15

ms for Humanoid. Calls are synchronous, with one in flight; actual completed counts and cadence overruns are recorded rather than replaced by target counts.

GPU telemetry retains the fixed Xorg graphics service as part of the measured baseline and rejects other GPU owners. HP201 permits only the measurement process. Process inventories are recorded throughout power sampling; this is not a fully exclusive GPU or whole-host measurement. GPU device telemetry and HP201 card telemetry have different physical boundaries, with Host and simulator energy excluded in both cases.

Power samples use their actual timestamp midpoints. A linear baseline $\widehat{P}_{\mathrm{idle}}(t)$ joins the mean idle powers at the mean times of the two idle windows. For N completed calls in the measured active interval,

$$E_{\mathrm{dyn/decision}} = \frac{1}{N} \int_{t_0}^{t_1} \left[P(t) - \widehat{P}_{\mathrm{idle}}(t) \right] \mathrm{d}t.$$

Integration uses trapezoids with interpolated active boundaries. All windows have bracketing samples, at least 25/49/25 samples in their respective phases, and gaps below 0.65 s. Total power/energy and baseline values are retained alongside the signed dynamic estimates.

All 54 baseline-adjusted dynamic-power estimates fall below the descriptive noise threshold defined from idle variation, before/after drift and displayed power resolution. Six individual estimates are negative and remain included without clipping. Table G5 reports the mean and sample SD across three measurement windows per task/mode. This between-window dispersion is distinct from within-window idle noise, a standard error and meter accuracy. The measurements characterize this instrumented setup; they do not establish an energy-efficiency advantage.

Table G5: Signed idle-baseline-subtracted power and energy estimates for fixed MCrPPO Actors. Mean ± sample SD is across three separate measurement windows per task/mode. All 54 windows fall below the recorded descriptive idle-noise threshold; six individual estimates are negative and remain included. Between-window SD is not meter accuracy or SEM. These measurements do not establish an energy advantage.

Task	Mode	Dynamic power (mW)	Dynamic energy (mJ/decision)
HC	GPU eager	13.78±4.65	0.689±0.232
	CUDA Graph	2.44±11.41	0.122±0.570
	Lynxi HP201	2.74±9.23	0.137±0.462
Walker	GPU eager	19.76±7.28	0.166±0.061
	CUDA Graph	15.09±6.05	0.126±0.050
	Lynxi HP201	6.19±5.17	0.050±0.042
Hopper	GPU eager	22.82±2.81	0.193±0.024
	CUDA Graph	6.47±6.62	0.054±0.055
	Lynxi HP201	13.17±1.97	0.107±0.016
Ant	GPU eager	13.38±0.88	0.669±0.044
	CUDA Graph	2.22±6.84	0.111±0.342
	Lynxi HP201	5.24±6.92	0.262±0.346
Swimmer	GPU eager	8.45±4.28	0.338±0.171
	CUDA Graph	15.26±4.84	0.610±0.194
	Lynxi HP201	6.12±2.97	0.245±0.119
Humanoid	GPU eager	21.32±8.12	0.325±0.126
	CUDA Graph	9.88±8.17	0.150±0.124
	Lynxi HP201	6.27±6.73	0.095±0.102

G.5 Earlier non-isolated CPU/Lynxi HP201 pilot

The earlier pilot evaluates functional execution for 18 MCrPPO Actors on the original monitoring reset seeds, separately from the three-family, new-reset GPU/Lynxi HP201 protocol above.

Export and execution contract. The serialized Actor package includes frozen observation mean, variance, and count; Actor parameters including raw log scale; task schema; exact action bounds;

and a declaration of the three-state Actor schema. The execution code initializes and maintains (h, z, o) and applies the documented reset contract. The package excludes value parameters/state/RMS, W_h, Adam, scheduler, and mutable training diagnostics. Its deterministic action is the bounded latent mean; stochastic action sampling is optional and uses an explicit generator.

Exported Actor parameters and observation statistics match their originating learners. Teacher-forced errors, closed-loop returns and board telemetry are reported separately. Because the monitoring reset seeds are reused, this pilot does not test held-out hardware preservation or numerical equivalence.

Measurements. Mean device latency across the 18 exported PPO Actors is 1.294–1.404 ms, versus .274–.329 ms for the same-host CPU reference (Table G6). This establishes functional execution under the pilot protocol rather than a latency advantage; whole-board telemetry excludes host/simulation energy.

Table G6: Earlier HP201 Actor-deployment pilot. Each row aggregates three final-checkpoint Actors (training seeds S1–S3) and five monitoring reset seeds per Actor. Returns and paired differences are mean ± sample SD across Actors; latency entries are Actor means.

Task	CPU return	HP201 return	Δ return	CPU/HP201 decision (ms)
HalfCheetah	5921.8 ± 976.0	6028.3 ± 1256.4	$+106.5 \pm 500.6$	0.320/1.302
Walker2d	3916.7 ± 209.1	3912.5 ± 200.2	-4.2 ± 11.8	0.274/1.294
Hopper	2294.2 ± 948.9	2128.2 ± 1138.5	-166.0 ± 202.5	0.324/1.311
Ant	5726.6 ± 90.1	5469.0 ± 607.4	-257.6 ± 571.4	0.313/1.337
Swimmer	69.7 ± 24.6	69.3 ± 23.8	-0.5 ± 0.9	0.287/1.333
Humanoid	6747.4 ± 773.4	7005.1 ± 245.3	$+257.7 \pm 1018.1$	0.329/1.404

Teacher-forced membrane, action-membrane and bounded-action errors have different native units and are reported separately. Decision latency includes execution and communication.

Table G7: HP201 board measurements for the earlier deployment pilot. Values are mean ± sample SD across three Actors per task; dynamic power and energy use timestamped whole-board telemetry with idle subtraction.

Task	dynamic board power (W)	dynamic board energy (mJ/decision)	idle offset (W)
HalfCheetah	0.0711 ± 0.0061	0.0824 ± 0.0121	15.399 ± 0.008
Walker2d	0.0855 ± 0.0046	0.0945 ± 0.0068	15.392 ± 0.002
Hopper	0.0783 ± 0.0098	0.0880 ± 0.0075	15.387 ± 0.008
Ant	0.0835 ± 0.0090	0.0910 ± 0.0073	15.395 ± 0.008
Swimmer	0.0807 ± 0.0080	0.0899 ± 0.0067	15.389 ± 0.006
Humanoid	0.0881 ± 0.0194	0.0967 ± 0.0200	15.392 ± 0.005

H Surrogate Sensitivity and Further Scope

H.1 Single-seed surrogate-family comparison

The observed utility of direct membrane credit differs across surrogate families. On the original monitoring grid for HalfCheetah, the ATan A–D learning-curve AUC differences are -378.0 through 4 Mi, -577.0 through 10 Mi, and -723.4 at the endpoint. Fast-sigmoid instead gives $+193.1$, $+177.8$, and $+381.4$, respectively, with positive A–E differences on all three outcomes. Thus a positive direct-credit case is not confined to the rectangular profile, while two noncompact profiles can still produce different A/D orderings.

AMC changes the Critic's forward information as well as its encoder-credit route. The local identity in Eq. 3 holds across surrogate profiles. The main experiments use a rectangle; the additional study keeps the hard forward map fixed and compares

$$s_{\mathrm{rect}}(h) = \mathbf{1}[0 < h < 1], \qquad s_{\mathrm{atan}}(h) = \frac{1}{1 + [\pi(h - .5)]^2}, \qquad s_{\mathrm{fast}}(h) = \frac{1}{(1 + 2|h - .5|)^2}.$$

Each profile has peak and integral one. ATan and fast are positive for every finite real h; actual FP32 zero rates require measurement. Matching peak and area does not align the derivative distribution on visited states, its correlation with loss adjoints, or the resulting optimizer updates.

The single-seed comparison covers HalfCheetah, Walker and Ant with E/D/A, one paired training seed and 10 Mi per run. The two noncompact profiles comprise 18 completed runs. Each combination of task, profile and condition still has only one training run, so this comparison does not provide precision across training seeds. These tasks were chosen after observing different training outcomes. Rectangular comparisons require the same hard-forward initialization, RNG, optimizer, evaluator and parameter-update ownership; incompatible runs do not provide matched controls. Wider replication remains necessary.

The primary contrast is A–D learning-curve AUC within each profile; A–E, D–E, fixed tail and endpoint provide complementary outcomes. Cross-profile A–D differences describe a family-by-credit interaction rather than the isolated effect of zero support, because the surrogate also changes event gradients and subsequent training trajectories. A replicated noncompact benefit would show that exact-zero event gradients are unnecessary for that benefit, whereas disappearance or reversal would narrow the claim to the corresponding surrogate and task. The comparison does not separate support topology from derivative shape, optimizer response or subsequent data.

Table H1: Single-seed surrogate-family results on the training-monitoring grid. $\mathrm{AUC}_{0:4}$ and $\mathrm{AUC}_{0:10}$ are normalized by their integration-window lengths, and $\mathrm{Tail}_{8:10}$ averages the 33 inclusive checkpoint means. Each task, profile and condition has one training run; no between-training-seed SD is estimated. Boldface marks the highest mean within each task and metric.

Family	Task	Condition	$\mathrm{AUC}_{0:4}$	$\mathrm{AUC}_{0:10}$	Mon. R_{10}	$\mathrm{Tail}_{8:10}$
atan	HC	E	3437.7	4457.4	5677.3	5543.1
		D	**4308.0**	**5423.6**	**6438.9**	**6414.3**
		A	3930.0	4846.6	5715.5	5612.9
	Walker	E	**2472.3**	**3226.3**	4163.1	3950.8
		D	1792.6	3099.2	**4289.5**	**4223.0**
		A	2392.9	3193.9	4184.2	4011.0
	Ant	E	1516.9	3204.6	**5278.9**	**4889.6**
		D	1366.7	2678.1	4083.8	4155.6
		A	**1644.6**	**3279.9**	5197.6	4846.2
fast	HC	E	3730.4	4608.4	5561.6	5352.6
		D	4063.5	4920.4	5621.4	5696.7
		A	**4256.6**	**5098.2**	**6002.8**	**5868.5**
	Walker	E	1859.9	2764.5	3247.9	2947.5
		D	**2457.4**	**3237.3**	2591.0	3405.8
		A	1842.6	2736.6	**3756.4**	**3517.7**
	Ant	E	**1724.3**	**3286.8**	**5158.8**	**4862.3**
		D	1489.3	3025.0	4544.9	4543.0
		A	1603.4	3270.2	5128.5	4803.2

H.2 Full 20-reset final-policy returns

The common-reset reevaluation yields a distinct final-policy comparison. ATan has negative A–E differences on HalfCheetah, Walker and Ant, whereas fast-sigmoid has positive A–E differences on all three tasks. On HalfCheetah, A–D is -1038.9 for ATan and $+275.2$ for fast-sigmoid. Fast-sigmoid Walker has an A–D difference of -257.6, reversing the positive ordering of its historical monitoring endpoint. Every comparison still has $n = 1$ training seed, so these signs are task/profile observations, not replicated family effects.

Table H2: Surrogate sensitivity at 10 Mi on 20 common new reset seeds per model, using one training seed ($n = 1$). Values are model means and paired differences, without an SD estimate across training seeds. Training-monitoring metrics are reported separately in Table H1. Boldface marks the highest mean within each task and metric.

Profile	Task	E	D	A	A–E	A–D	D–E
atan	HC	5724.3	**6531.9**	5493.0	-231.3	-1038.9	807.6
atan	Walker	**4230.5**	4001.7	4120.2	-110.3	118.5	-228.8
atan	Ant	**5180.9**	4180.9	5178.5	-2.4	997.6	-1000.0
fast	HC	5428.9	5721.8	**5997.0**	568.1	275.2	293.0
fast	Walker	3495.8	**3837.1**	3579.5	83.7	-257.6	341.3
fast	Ant	5054.9	4897.7	**5195.2**	140.3	297.5	-157.2

H.3 Limitations and scope

The local identities and final-checkpoint interventions characterize the additional gradient route at fixed models, while the E/D/A runs measure downstream training outcomes. Explaining task-dependent effects over the full training trajectory calls for longitudinal measurements of branch gradients and optimizer states.

The controlled comparisons use three training seeds per setting, and the additional noncompact-surrogate comparisons use one. Common-reset evaluation improves the return estimate for each fixed model but does not add training replicates. The reset-only PPO study in Appendix D.4 measures evaluation-state variation of fixed final policies rather than retraining variation. Off-policy results are obtained at 1 Mi and coexist with natural termination on several tasks; they do not establish stable six-task locomotion. Task-specific gait, fall and plateau criteria, parameter-matched recurrent ANN comparisons and complete isolated training-cost measurements remain outside the evidence. The system comparison follows the run-inclusion protocol in Appendix F, and hardware results concern selected Actors executed on GPU and Lynxi HP201 in simulated environments (Appendix G.1).

I Extended Related Work

Spiking continuous control. PopSAN trains a spiking Actor with a deep Critic across multiple learning rules and demonstrates Loihi execution (Tang et al., 2021); SANSAC also studies spiking Actors (Hunter et al., 2026). Fully spiking PPO Actor–Critics establish direct policy/value learning (Zanatta et al., 2024; Zhang et al., 2025), while spiking predictive-control systems provide a model-based route (Huebotter et al., 2025). The present work focuses on persistent-state organization and Critic access across learning rules, with matched controls for the additional membrane-credit route.

Internal state and representation learning. Membrane readouts support continuous actions in ILC-SAN and Q values in DSQN (Chen et al., 2024; 2022). In discrete embodied navigation, SpikingNav gives both heads access to shared post-reset membrane state (Zhang et al., 2026). AMC instead reserves the added pre-threshold readout for the Critic and separately controls its direct encoder-gradient route. Related Actor–Critic work studies shared representations, asymmetric information and separated policy/value optimization (Garcin et al., 2025; Pinto et al., 2018; Baisero & Amato, 2022; Cobbe et al., 2021). Blocking Actor-loss gradients into an encoder has precedents in recurrent and image-based control (Yang & Nguyen, 2021; Yarats et al., 2021).

Training-time support and stability. Proxy Target introduces a continuous target-side proxy for a spiking Actor (Xu et al., 2025); Adaptive Surrogate Gradients combines slope scheduling with a privileged guide for sequential control (Van den Berghe et al., 2025). CaRe-BN recalibrates normalization statistics in hybrid spiking-Actor/ANN-Critic control (Xu et al., 2026). These methods address different training choices from AMC's shared-core state access and direct encoder-credit permission.