SPIKEARC: SPIKE-GUIDED ROUTING AND SELECTIVE POLICY CONSOLIDATION FOR CONTINUAL REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning can produce high-performing control policies, but real-world deployment demands more than performance in fixed environments. As environmental dynamics and task requirements change, agents must continue learning, which can disrupt acquired yet still useful behaviors. Continual reinforcement learning therefore faces a fundamental challenge: when adaptation is required, it is not immediately clear whether a behavioral correction should remain transient or eventually become part of the long-term policy. We introduce SpikeARC, a framework that organizes continual policy adaptation through state-dependent action corrections generated by complementary Stable and Fast pathways. The Stable pathway provides the primary action, while recurrent spiking Fast and Fusion modules generate and gate state-dependent corrections. Spike-guided routing assigns replay responsibility on a per-sample basis, dual behavior-space constraints regulate behavioral drift, and corrections that remain consistent are progressively consolidated into the Stable pathway. We systematically evaluate SpikeARC through comparisons with external baselines, controlled cross-environment experiments, mechanism ablations, and structured experiments with simulated quadruped robots. In external continual-RL evaluations on CARL Walker and Quadruped, SpikeARC maintains competitive adaptation while reducing forgetting relative to hidden-context continual-RL baselines. In a controlled five-environment study with a PPO backbone, it retains 90.8% of the adaptation AUC of a fully shared policy while reducing forgetting by 36.2% and long-term policy parameter drift by 35.3% on average across environments. Ablations and structured quadruped experiments further characterize the adaptation-retention trade-offs induced by routing and consolidation. SpikeARC does not require a new behavioral correction to be classified as transient or permanent upon first encounter. Instead, the correction's role is resolved progressively through continued interaction while mature behavior remains protected.

Keywords: Continual Reinforcement Learning; Stability–Plasticity Dilemma; Spiking Neural Networks; Gradient Routing; Catastrophic Forgetting

1 INTRODUCTION

Real-world continuous-control systems rarely operate under fixed dynamics for long periods. Friction, mass, actuator efficiency, contact conditions, and external disturbances all change over time, whereas a control policy may already have acquired mature behavior through extensive prior interaction. Continual reinforcement learning therefore faces a problem more specific than simply continuing to optimize: after the environment changes, the policy must correct its current behavior quickly, yet those corrections should not readily overwrite reliable long-term capabilities. For continuously operating robots and control systems, the central difficulty is not how to keep a policy learning, but how new behavioral changes should be incorporated into an already mature policy.

Our point of departure is that an agent need not decide immediately whether a change is "short-term" or "long-term." Instead, we decompose the process by which new behavior enters the long-term policy into rapid expression, responsibility assignment, consistency validation, and risk-constrained

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.

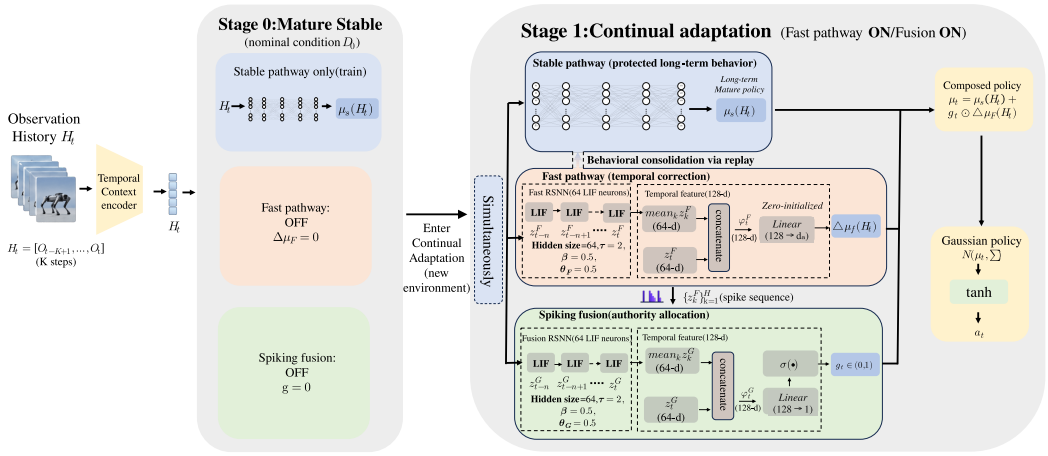

Figure 1: SpikeARC: from transient correction to consolidated skill formation. During stationary pretraining, only the Stable pathway is active, forming the mature primary policy. During continual adaptation, the Fast pathway produces corrections from recent observations, while Fusion gates them using state history and Fast spike activity; the final action combines the Stable action with the gated correction.

write-back. A new change is first expressed rapidly as a local correction, after which optimization responsibility is assigned according to the correction's actual contribution to the current behavior. Only corrections that remain consistent over subsequent interactions gradually become eligible for long-term write-back. The extent to which the stable, mature policy may change is determined by its actual behavioral drift rather than by a fixed freezing ratio or a reduced learning rate.

Building on this idea, SpikeARC (Figure 1) combines a mature Stable policy with gated, state-dependent Fast-pathway corrections from recent observations. Stable remains trainable within controlled limits. Responsibility routing in this process determines which pathway absorbs each change; consistency-guided consolidation selects changes for long-term retention; and dual behavioral budgets regulate how much Stable may change over time.

SpikeARC therefore turns continual adaptation into a gradual process from immediate correction to long-term capability formation. It requires neither an upfront decision about whether a change is transient or persistent nor a manually specified Fast-to-Stable handoff; instead, long-term write-back is determined progressively by behavioral contribution, consistency, and risk. Our contributions are as follows:

1. We introduce a behavior-formation framework for continual policy adaptation that separates immediate behavioral correction from long-term policy formation. A mature Stable pathway provides the primary action, while gated Fast-Fusion corrections express new behavior without requiring an upfront transient/persistent decision.
2. We develop a selection mechanism that determines which corrections should be retained over the long term. Replay responsibility is assigned according to the Fast pathway's actual contribution to the composed action, while persistent corrections are progressively consolidated into Stable rather than written back indiscriminately.
3. We introduce dual behavioral-drift budgets for risk-controlled Stable plasticity. The budgets constrain both mature Stable behavior and historical composed-policy behavior, dynamically regulating long-term plasticity without access to the true dynamics model.

We validate SpikeARC through cross-backbone PPO/SAC comparisons, five-environment matched-base experiments, mechanism ablations, capacity controls, and structured quadruped robot studies. Together, these designs address a single question: how can a new correction progress from immediate behavior to long-term capability while keeping changes to a mature policy under control? Appendix E.1 provides an extended discussion of the design motivation and the behavior-formation process.

2 Related Work

Continual learning addresses the stability-plasticity dilemma through rehearsal, sample selection, distillation, parameter isolation, and regularization (Aljundi et al., 2019; Hayes et al., 2021; Li et al., 2024; Parisi et al., 2019), while replay effectiveness also depends on memory, retrieval, correlation structure, objectives, and scheduling (Aljundi et al., 2019; Hayes et al., 2021; Li et al., 2024). Continual reinforcement learning adds nonstationary state-action distributions and negative transfer (Ahn et al., 2025). Representative approaches include MTR with multi-timescale replay (Kaplanis et al., 2020), 3RL for task-agnostic recurrent replay-based learning (Caccia et al., 2023), and LCPO with local policy constraints on past experience (Hamadanian et al., 2025); CARL provides controlled contextual shifts (Benjamins et al., 2021). SpikeARC focuses on a more specific question: when the complete forward action remains unchanged, which functional pathway should absorb the replay error?

Complementary-learning and dual-memory methods separate rapid acquisition from long-term retention through short- and long-term stores, fast and slow weights, or functional specialization across modules (Arani et al., 2022; Nie et al., 2023; Rostami et al., 2019; Sarfraz et al., 2023). Related fast-slow designs have also been used for rapid adaptation and slow, stable updating in continual learning and reinforcement learning (Arani et al., 2022; Capone and Muratore, 2024). Within a single continuous-control policy, SpikeARC distinguishes three permissions: contributing to the current action, receiving replay credit, and writing a correction into the long-term pathway. Stable retains risk-constrained on-policy plasticity rather than remaining permanently frozen, and it operates concurrently with the Fast and Fusion networks.

Spiking and neuromorphic learning can provide recurrent temporal state, event-driven computation, homeostatic mechanisms, and replay-inspired learning rules (Capone and Muratore, 2024; Golden et al., 2022; Muñoz-Martin et al., 2021; M. Wang et al., 2025; Xu et al., 2026). SpikeARC uses recurrent leaky integrate-and-fire (LIF) dynamics as the temporal carrier for Fast corrections and Fusion context, but the framework is not restricted to spiking neural networks (SNNs). It can also be instantiated with a gated recurrent unit (GRU) or long short-term memory (LSTM) network. Our experiments indicate that the SNN implementation offers an advantage in model footprint and is particularly well suited to event-driven computation. The recurrent-core control in Appendix C.3 examines this boundary. Our claim concerns the organization of adaptation responsibility, not the choice of neuron model. Appendix E.2 provides an extended review of related work.

3 Method

SpikeARC decomposes continual adaptation into three coupled mechanisms: (i) a Stable-Fast behavioral representation that expresses new changes as gated corrections, (ii) contribution-aware replay routing with consistency-guided selective consolidation, and (iii) dual behavioral budgets that regulate Stable plasticity. Figure 2 summarizes how these mechanisms move corrections from immediate expression to controlled long-term write-back.

3.1 Behavior Representation

Let $h_t = (s_{t-H+1}, \ldots, s_t)$ denote the observation history over the most recent H steps. The Stable pathway outputs the primary action mean μ_t^S. The Fast pathway uses its recurrent spiking state to produce a candidate correction $\Delta\mu_t^F$. The Fusion pathway uses the state history and the Fast spike sequence to output a gate $g_t \in [0, 1]$. The final action mean is the sum of the primary action and the gated correction:

$$\mu_t = \mu_t^S + c_t, \qquad c_t = g_t \Delta\mu_t^F, \qquad g_t \in [0, 1]. \tag{1}$$

The Fast and Fusion pathways use standard recurrent leaky integrate-and-fire (LIF) dynamics and surrogate-gradient training. Appendix A.1 gives the complete recurrent dynamics and action-sampling procedure. During the base stage, both pathways are disabled and the Fast output layer is initialized to zero. The mature policy is therefore not exposed to an abrupt correction when the dynamics begin to change. The left side of Figure 2 shows how a new correction enters the composed action. The upper branch assigns Stable's replay responsibility according to the correction's actual contribution, while the lower branch reevaluates the correction's direction, magnitude, and use. The

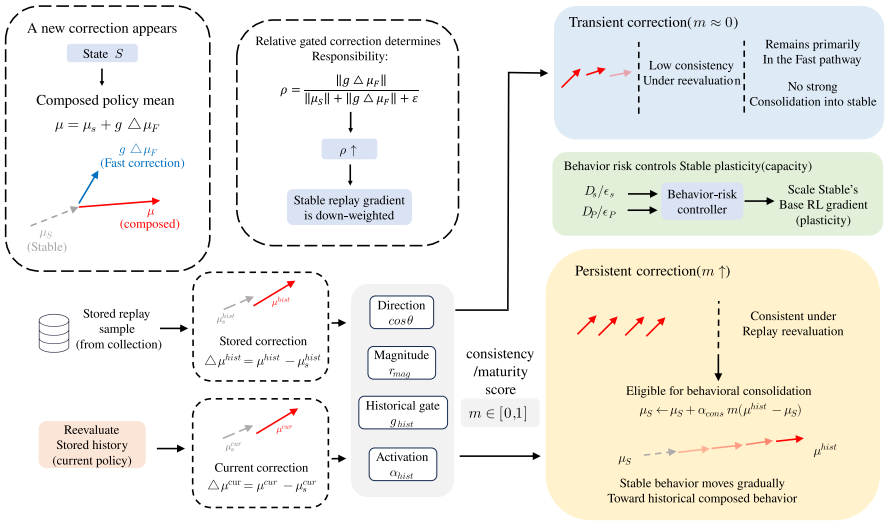

Figure 2: Mechanism from action correction to long-term write-back. Left: the composed action combines the Stable primary action with a gated Fast-pathway correction. Upper middle: the Stable replay gradient for a sample is reduced according to the correction's contribution to the historical action. Lower middle: historical and current corrections are compared to compute a consistency weight. Right: inconsistent corrections remain in the Fast pathway, whereas consistently persistent corrections are written progressively into the Stable pathway. The dual behavioral budgets simultaneously constrain Stable-pathway drift.

right side separates corrections that lack persistence from those eligible for long-term write-back and shows how behavioral risk feeds back to regulate Stable-pathway plasticity.

3.2 Contribution-Aware Responsibility Assignment and Consistency-Guided Selective Consolidation

For replay sample i, we store the composed action mean, Stable action mean, and gate at the time of sampling, $\left(\bar{\mu}_i, \bar{\mu}_i^S, \bar{g}_i\right)$, and define $\bar{c}_i = \bar{\mu}_i - \bar{\mu}_i^S$. The relative contribution of the Fast correction to the historical action is

$$q_i = \frac{\| \bar{c}_i \|_2}{\| \bar{\mu}_i^S \|_2 + \| \bar{c}_i \|_2 + \epsilon_r}, \qquad \rho_i = clip\left(\frac{q_i - f}{1 - f}, 0, 1\right). \tag{2}$$

The responsibility weight ρ_i does not alter the forward action. It only modifies the gradient path by which the replay error reaches the Stable pathway. For the current composed action $\mu_i = \mu_i^S + c_i$, we construct

$$\widetilde{\mu}_i = (1 - \rho_i)\, \mu_i^S + \rho_i sg\left(\mu_i^S\right) + c_i, \qquad \mathcal{L}_{rep} = \frac{1}{B} \sum_i \| \widetilde{\mu}_i - \bar{\mu}_i \|_2^2, \qquad \frac{\partial \widetilde{\mu}_i}{\partial \mu_i^S} = 1 - \rho_i. \tag{3}$$

When the historical action was produced primarily by the Stable pathway, Stable receives most of the replay gradient. When the correction contributed substantially, Stable's direct update for that sample is reduced, while the Fast and Fusion pathways still fit the historical composed action through the actual correction computation graph. Here, "spike-guided" means that the upstream spiking dynamics determine c_i and the gate. The final responsibility weight is computed from the actual correction contribution in action space and is not simply the spike rate. Appendix A.2 provides the complete specification of the responsibility weight, default parameters, and replay-gradient paths.

Responsibility assignment answers who should absorb the historical error, but it cannot by itself determine which corrections should become long-term capabilities. We therefore combine the histori-

cal and current gate strengths, directional agreement, magnitude agreement, and effective activation to compute correction maturity:

$$m_i = w_i^g d_i a_i I_i \in [0, 1]. \tag{4}$$

This maturity score controls selective write-back at the behavioral level:

$$y_i^S = sg\left(\mu_i^S\right) + \alpha_c(p) m_i \left(\bar{\mu}_i - sg\left(\mu_i^S\right)\right), \qquad \mathcal{L}_{con} = \frac{1}{B} \sum_i \parallel \mu_i^S - y_i^S \parallel_2^2 . \tag{5}$$

Consolidation neither copies the parameters of the Fast pathway nor transfers every correction directly into the Stable pathway. Only corrections that remain in use and maintain consistent direction and magnitude over subsequent interactions cause Stable, through selective consolidation, to approach the corresponding historical composed behavior. Appendix A.3 gives the complete definitions of w_i^g, d_i, a_i, I_i, and $\alpha_c(p)$, together with the consolidation schedule.

3.3 Behavior-Preserving Plasticity Regulation

Even a consistent correction may induce a substantial change in mature behavior. SpikeARC therefore maintains two types of behavioral references. $\mathcal{A}_S$ stores the mature Stable-pathway behavior from the base stage, while $\mathcal{A}_{P,j}$ stores the composed-policy behavior at the end of each completed stage j. Let D_S denote normalized Stable-pathway drift and let $D_P = \max_j D_{P,j}$ denote the maximum composed-policy drift over historical stages. We require

$$D_S \le \epsilon_S, \qquad D_P \le \epsilon_P. \tag{6}$$

Here, ϵ_S and ϵ_P are prespecified budget limits. We maintain these two soft behavioral-drift constraints through primal-dual optimization and use budget utilization,

$$\begin{aligned} u &= max\left(\frac{D_S}{\epsilon_S}, \frac{D_P}{\epsilon_P}\right), \quad w = clip\left(\frac{\tau - u}{\tau - 0.25}, 0, 1\right), \\ \alpha_S &= \alpha_{base} + w\left(\max\left(\alpha_{base}, c\right) - \alpha_{base}\right), \end{aligned} \tag{7}$$

to regulate the gradient that the Stable pathway receives from the current reinforcement-learning objective. When sufficient budget remains, Stable can participate more strongly in current adaptation. As either behavioral drift approaches its budget boundary, the corresponding gradient scale decreases smoothly. Appendix A.4 gives the complete definitions of the behavioral reference sets, normalized drift, dual updates, and Stable-pathway plasticity schedule.

3.4 Joint Optimization and Backbone Compatibility

Combining the preceding mechanisms, the behavioral-replay component, which comprises all losses based on replayed behavior for historical behavior reproduction and long-term consolidation, is

$$\mathcal{L}_{beh} = \mathcal{L}_{rep} + \lambda_{con} \mathcal{L}_{con}. \tag{8}$$

The SpikeARC auxiliary objective is then

$$\mathcal{L}_{aux} = \lambda_{rep} \mathcal{L}_{beh} + \mathcal{B}\left(D_S; \lambda_S, \epsilon_S, \rho_S\right) + \mathcal{B}\left(D_P; \lambda_P, \epsilon_P, \rho_P\right). \tag{9}$$

The resulting Stable-pathway gradient has the form

$$g_S = \alpha_S \nabla_{\theta_S} \mathcal{L}_{RL} + \nabla_{\theta_S} \mathcal{L}_{aux}. \tag{10}$$

The Fast and Fusion pathways continue to receive gradients from $\mathcal{L}_{RL}$, replay, and the terms involving them in the full-policy budget through the actual computation graph. Scaling Stable's base RL gradient before adding the behavior-preservation and consolidation gradients assigns distinct optimization roles to rapid adaptation and long-term write-back.

These SpikeARC mechanisms are defined at the actor's behavioral-decomposition layer and the replay/constraint layer, and can therefore be transferred across continuous-control backbones. Appendices A.5-A.6 describe joint optimization, backbone adaptation, and the expanded training procedure. Algorithm 1 in Appendix A.6 summarizes continual policy adaptation with SpikeARC.

4 EXPERIMENTS

The experiments test a single claim: continual adaptation should control how new behavior enters the long-term policy rather than simply suppress policy change.

4.1 EXPERIMENTAL SETUP AND EVALUATION METRICS

The general protocol consists of 1 million steps of nominal pretraining followed by four successive dynamics-shift stages: low friction, increased mass, weakened actuators, and a mixed shift. Each stage lasts 300,000 steps. The policy receives neither a task or stage identifier nor the true dynamics parameters. The controlled main experiments use PPO on HalfCheetah-v5, Walker2d-v5, Hopper-v5, CARL-DMC-Walker, and CARL-DMC-Quadruped. Each condition uses three random seeds, and SpikeARC and Fully Shared (Appendix B.3) start from the same mature checkpoint. The external-validity experiments use SAC on CARL Walker and Quadruped and compare SpikeARC with MTR and 3RL under the same hidden-context protocol. We report Adaptation AUC, mean positive forgetting, and Stable parameter drift. The first metric measures continual adaptation; the latter two measure loss of historical behavior and the extent to which the long-term policy is rewritten. In addition to mean results, we use an environment-stratified paired bootstrap to estimate uncertainty in the macro-averaged effect across the five environments. The corresponding confidence intervals and complete statistical analyses appear in the appendix. Appendices B.1 and B.3-B.5 provide the complete experimental protocol and statistical analyses, respectively.

4.2 ADAPTATION AND LONG-TERM POLICY PRESERVATION

We first examine the cross-backbone evidence. Under SAC, the three methods achieve similar Adaptation AUC on CARL Walker (SpikeARC/MTR/3RL: 953.8/948.5/947.8), while SpikeARC yields lower forgetting (4.65 vs. 6.21/6.83), as shown in Figure 3. On the more discriminative Quadruped task, SpikeARC achieves an AUC of 925.2 (MTR: 922.0; 3RL: 871.1) and forgetting of 8.75 (22.92/24.39). The effect is therefore not specific to PPO: rapid adaptation need not entail a greater loss of historical capabilities. Appendix B.2 provides the complete external comparison, and Appendix C.4 reports LCPO reference results under a stronger-information condition. The

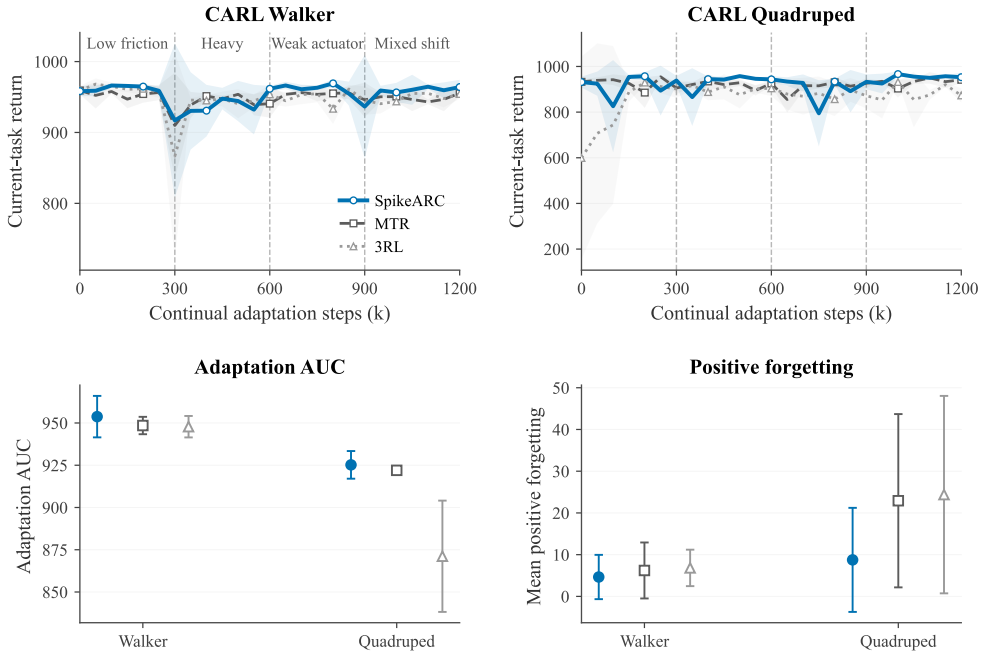

Figure 3: Hidden-context CARL comparison with SAC (3 seeds). A-B: current-task return; C: Adaptation AUC; D: forgetting. Complete values are reported in Appendix B.2 (Table S2).

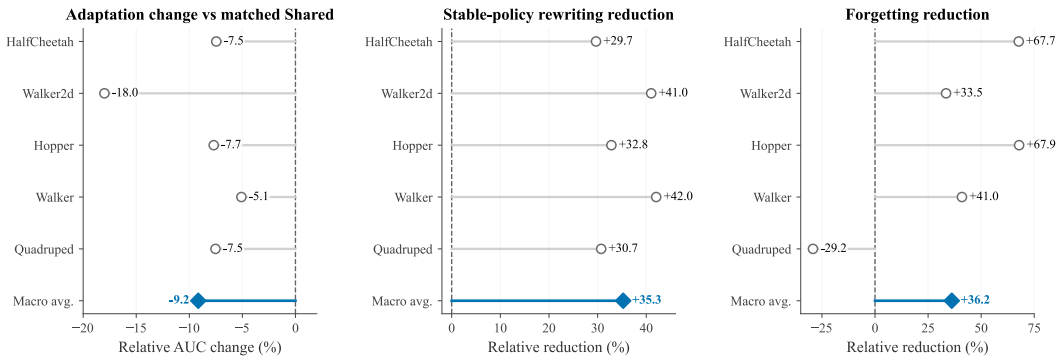

Figure 4: Matched-base PPO comparison across five environments. A: relative Adaptation AUC; B: reduction in Stable drift; C: reduction in forgetting. Complete values and confidence intervals are reported in Appendices B.3-B.5 (Tables S3-S7).

Table 1: Ablation variants and the mechanisms they isolate.

Variant	**SpikeARC (Full)**	**Fully Shared**	**Shared + Replay/EWC**	**Fixed Fusion**
Modification	Complete Stable-Fast architecture, routing, selective consolidation, and behavioral budgets	Every new change is written directly into a shared long-term policy	Shared policy with replay and EWC	Dual pathways are retained, but the correction-fusion ratio is fixed

Variant	**No Routing**	**No Consolidation**	**Uniform Consolidation**
Modification	All other mechanisms are retained, but responsibility routing is removed	Fast can adapt, but selective write-back is disabled	Write-back is allowed, but corrections are not differentiated by maturity

more stringent matched-base PPO comparison isolates how the methods absorb change (Figure 4). SpikeARC retains 90.8% of the macro Adaptation AUC of Fully Shared while reducing Stable drift by 35.3% on average. The direction of the effect is consistent across all five environments and all 15 matched seed pairs. Behavioral forgetting decreases in four environments, with CARL Quadruped as the exception; the macro-averaged reduction in forgetting is 36.2%. This distinction is important. SpikeARC's most consistent effect is on how the mature policy is rewritten, whereas improved forgetting is a behavioral consequence in most environments rather than the result of stopping learning.

4.3 Mechanistic Analysis and Ablation Studies

On HalfCheetah, Walker2d, and Hopper, we compare SpikeARC (Full), Shared + Replay/EWC, Fixed Fusion, No Routing, No Consolidation, Uniform Consolidation, and Fully Shared. Results are first normalized to Fully Shared within each environment and then averaged with equal weight across the three environments.

Conventional forgetting mitigation does not reproduce the same combination of metrics (upper left of Figure 5). All component ablations use the same fixed experimental configuration; extended settings are provided in Appendix C.1. Across the three environments, Shared + Replay/EWC retains slightly more adaptation than SpikeARC (90.6% vs. 86.6%), but provides less protection against forgetting (36.8% vs. 71.3%) and less Stable-pathway protection (22.8% vs. 34.6%). This comparison

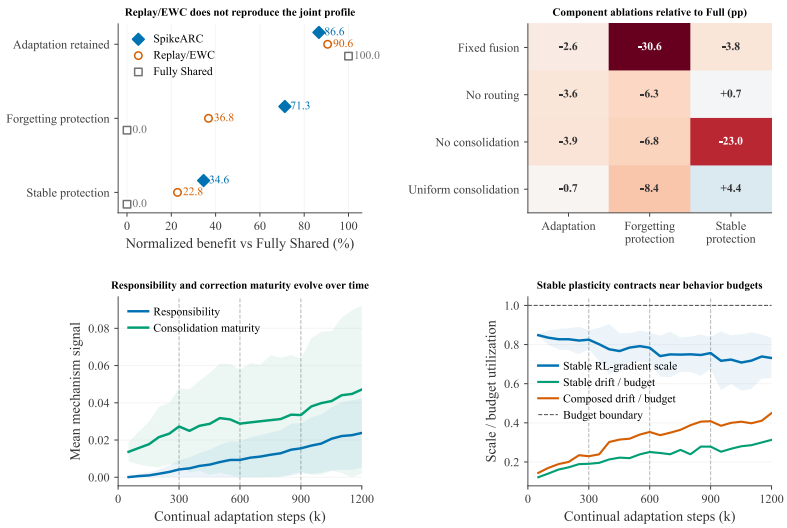

Figure 5: Mechanistic evidence for SpikeARC. Top: shared-policy comparisons and component ablations; bottom: the evolution of responsibility and correction maturity, and Stable plasticity under behavioral-risk control. See Appendix C.1 for extended analyses.

shows that adjusting replay or regularization strength serves a different function from distinguishing immediate correction from long-term write-back.

The component ablations further localize these effects (upper right of Figure 5). Fixed Fusion primarily reduces protection against forgetting, by 30.6 percentage points. Removing responsibility routing decreases Adaptation AUC/Fully Shared and Forgetting reduction/Fully Shared by 3.6 and 6.3 percentage points, respectively, while Stable-drift reduction/Fully Shared remains essentially unchanged, increasing by 0.7 percentage points. This pattern is consistent with a change in replay-gradient allocation rather than uniform suppression of Stable updates. No Consolidation decreases Stable protection by 23.0 percentage points. Uniform Consolidation raises Stable protection by another 4.4 percentage points but lowers protection against forgetting by 8.4 percentage points. The training trajectories in the lower row show that the responsibility weight and correction maturity evolve during training, while the Stable pathway's base reinforcement-learning gradient remains nonzero throughout. The mechanism therefore does not protect Stable by suppressing its learning globally. Instead, correction maturity selectively controls consolidation of corrections into the Stable pathway. As shown in Figure S1 in Appendix C.2, increasing model capacity alone does not reproduce the control afforded by these mechanisms. On Hopper, standard shared multilayer perceptrons achieve 1.02-1.42 times the Adaptation AUC of SpikeARC, but incur 2.62-4.27 times its forgetting and 1.57-1.97 times its parameter drift. The w512 model, with approximately three times as many parameters, still exhibits nearly twice the drift. CARL Quadruped provides a boundary case: the approximately parameter-matched w288 model outperforms SpikeARC in Adaptation AUC and forgetting (366.4/64.5 vs. 294.8/117.2), but its parameter drift remains approximately 50% higher (0.376 vs. 0.251). After retraining the base policy under stationary nominal dynamics, the ordering of behavioral metrics changes, whereas the drift ratios remain largely unchanged. A large shared policy can obtain strong performance through global reconfiguration. The additional control provided by SpikeARC is the distinction between immediate behavioral correction and long-term write-back. Appendix C.2 reports extended model-scale sweeps and an independent replication with a retrained base policy.

4.4 Behavioral Analysis of Quadruped Adaptation

Finally, we examine whether replay responsibility assignment affects actual locomotion. As shown in Figure 6, the complete SpikeARC method and the No Routing variant differ only in the replay-responsibility mechanism. Fast, Fusion, replay, EWC, selective consolidation, Stable-plasticity scheduling, and all other components remain unchanged. Training uses five random seeds. The

final-checkpoint evaluation uses three independent checkpoints, each evaluated for 120 episodes. This intervention tests whether the change in adaptation performance caused by removing responsibility routing corresponds to a different locomotion strategy. Table 2 reveals a clear behavioral

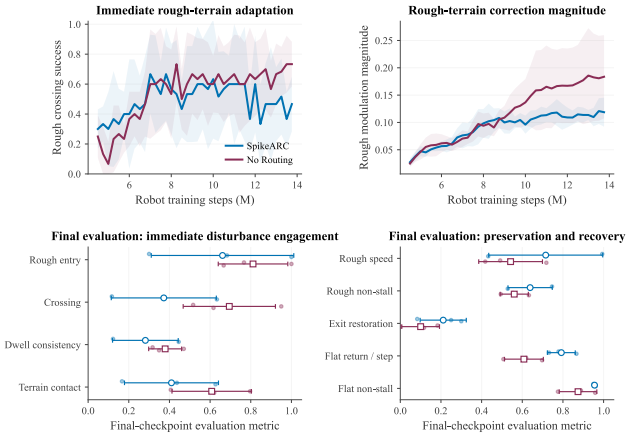

Figure 6: Responsibility-routing intervention in rough-terrain adaptation. Top: training traversal success and correction magnitude; bottom: final-checkpoint terrain engagement, speed, pausing, and recovery. See Appendices D.1–D.2 for complete metrics and diagnostics.

trade-off: No Routing achieves higher rough-terrain traversal success, although this metric can also count avoidance strategies such as bypassing rough terrain, whereas SpikeARC better preserves rough-terrain speed, reduces pausing, and improves post-terrain and flat-terrain recovery. With all other mechanisms controlled, this intervention shows that responsibility routing changes the trade-off between immediate traversal and preservation of a mature gait.

Table 2: Locomotion performance at the final quadruped checkpoints (3 independent checkpoints, mean ± standard deviation). No Routing improves rough-terrain traversal success, whereas SpikeARC performs better on rough-terrain speed, pausing, and recovery after leaving the terrain.

Metric	**SpikeARC**	**No Routing**	**Favored Method**
Rough-terrain traversal success rate ↑	0.372 ± 0.258	0.694 ± 0.227	**No Routing**
Rough-terrain speed ratio ↑	0.715 ± 0.280	0.543 ± 0.157	**SpikeARC**
Rough-terrain pause ratio ↓	0.361 ± 0.109	0.439 ± 0.069	**SpikeARC**
Post-terrain recovery success rate ↑	0.211 ± 0.113	0.100 ± 0.093	**SpikeARC**
Flat-terrain speed recovery ↑	0.792 ± 0.069	0.608 ± 0.096	**SpikeARC**

5 Conclusion

SpikeARC reframes continual policy adaptation as a gradual process from transient behavioral correction to long-term skill formation. A change need not be classified as transient or permanent upon first encounter; it is first expressed as a controlled correction and is assigned to the long-term policy progressively according to its behavioral contribution, temporal consistency, and associated risk. By separating behavioral expression, responsibility attribution, and capability formation, SpikeARC retains continual Stable-pathway plasticity while preventing unvalidated changes from directly rewriting the mature policy. Cross-backbone evaluations, matched-checkpoint comparisons, model-capacity controls, and the quadruped behavioral intervention collectively show that continual adaptation does not require learning to stop. It requires a mechanism that determines which changes merit becoming lasting capabilities. SpikeARC provides this behavior-formation perspective, under which rapid adaptation and long-term retention become distinct stages of the same process of policy evolution rather than conflicting objectives.

AI USE STATEMENT

During the preparation of this manuscript, the authors used AI-assisted tools to improve language clarity, polish academic writing, and assist with translation checking. The authors reviewed and edited all AI-assisted outputs.

ETHICS STATEMENT

This study does not involve human subjects or personally identifiable information. The quadruped-robot experiments use a robot model, associated model files, and model weights that are subject to copyright and licensing restrictions and therefore cannot be publicly redistributed. The manuscript reports the experimental setting and results of the quadruped experiments, while the restricted robot assets themselves are not included in the submission.

REPRODUCIBILITY STATEMENT

The SpikeARC method and optimization procedure are specified in Section 3 and Appendix A. Section 4 and Appendices B–C document the experimental protocols, environments, baselines, random-seed conventions, evaluation metrics, matched-base comparisons, ablations, controls, and statistical analyses. The anonymized supplementary material contains the released SpikeARC implementation, PPO and SAC experiment launchers for the public Gymnasium/CARL studies, baseline and control implementations, and analysis utilities. The platform-specific quadruped-robot study in Section 4.4 and Appendix D is the sole experiment whose full execution stack is not redistributed with the submission; as noted in the Ethics Statement, the robot model cannot be released, while the experimental protocol and evaluation procedure are documented in the paper. This restriction does not apply to the public CARL-DMC-Quadruped benchmark used in the PPO and SAC experiments.

REFERENCES

Rebecca Adaimi and Edison Thomaz. Lifelong adaptive machine learning for sensor-based human activity recognition using prototypical networks. *Sensors*, 22, 2022.

Hongjoon Ahn, Jinu Hyeon, Youngmin Oh, Bosun Hwang, and Taesup Moon. Prevalence of negative transfer in continual reinforcement learning: Analyses and a simple baseline. In *International Conference on Learning Representations*, 2025.

Kian Ahrabian, Yishi Xu, Yingxue Zhang, Jiapeng Wu, Yuening Wang, and Mark Coates. Structure aware experience replay for incremental learning in graph-based recommender systems. In *ACM International Conference on Information and Knowledge Management*, 2021.

Rahaf Aljundi, Lucas Caccia, Eugene Belilovsky, Massimo Caccia, Min Lin, Laurent Charlin, and Tinne Tuytelaars. Online continual learning with maximally interfered retrieval. In *Advances in Neural Information Processing Systems*, 2019.

Elahe Arani, Fahad Sarfraz, and Bahram Zonooz. Learning fast, learning slow: A general continual learning method based on complementary learning system. In *International Conference on Learning Representations*, 2022.

Craig Atkinson, Brendan McCane, Lech Szymanski, and Anthony Robins. Pseudo-rehearsal: Achieving deep reinforcement learning without catastrophic forgetting. *Neurocomputing*, 428:291-307, 2021.

Qianyi Bai, Haiteng Wang, and Qiang Yu. A brain-inspired gating mechanism unlocks robust computation in spiking neural networks. In *International Conference on Learning Representations*, 2026.

Eden Belouadah and Adrian Popescu. IL2M: Class incremental learning with dual memory. In *IEEE/CVF International Conference on Computer Vision*, 2019.

Carolin Benjamins, Theresa Eimer, Frederik Schubert, André Biedenkapp, Bodo Rosenhahn, Frank Hutter, and Marius Lindauer. CARL: A benchmark for contextual and adaptive reinforcement learning. *arXiv preprint arXiv:2110.02102*, 2021.

Matteo Boschini, Lorenzo Bonicelli, Pietro Buzzega, Angelo Porrello, and Simone Calderara. Class-incremental continual learning into the eXtended DER-verse. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):5497-5512, 2023.

Pietro Buzzega, Matteo Boschini, Angelo Porrello, and Simone Calderara. Rethinking experience replay: A bag of tricks for continual learning. In *International Conference on Pattern Recognition*, 2021.

Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: A strong, simple baseline. *arXiv preprint arXiv:2004.07211*, 2020.

Massimo Caccia, Jonas Mueller, Taesup Kim, Laurent Charlin, and Rasool Fakoor. Task-agnostic continual reinforcement learning: Gaining insights and overcoming challenges. In *Proceedings of the 2nd Conference on Lifelong Learning Agents*, volume 232 of *Proceedings of Machine Learning Research*, pages 89-119, 2023.

Wenyu Cai, Hao Chen, and Meiyan Zhang. A bio-inspired multiple autonomous underwater vehicle encirclement tracking method: Adaptive recurrent neuron and bio-inspired experience replay mechanism. *Ocean Engineering*, 342:123007, 2025.

Cristiano Capone and Paolo Muratore. Learning fast while changing slow in spiking neural networks. *Neuromorphic Computing and Engineering*, 4(3):034002, 2024.

Arslan Chaudhry, Albert Gordo, Puneet Dokania, Philip Torr, and David Lopez-Paz. Using hindsight to anchor past knowledge in continual learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

Cyprien de Masson d'Autume, Sebastian Ruder, Lingpeng Kong, and Dani Yogatama. Episodic memory in lifelong language learning. *arXiv preprint arXiv:1906.01076*, 2019.

Georgios Detorakis, Sadique Sheik, Charles Augustine, Somnath Paul, Bruno U. Pedroni, Nikil Dutt, Jeffrey Krichmar, Gert Cauwenberghs, and Emre Neftci. Neural and synaptic array transceiver: A brain-inspired computing framework for embedded learning. *Frontiers in Neuroscience*, 12:583, 2018.

Sayna Ebrahimi, Franziska Meier, Roberto Calandra, Trevor Darrell, and Marcus Rohrbach. Adversarial continual learning. In *European Conference on Computer Vision*, 2020.

Anthony Fuller, Yousef Yassin, Daniel G. Kyrollos, Evan Shelhamer, and James R. Green. Thicker and quicker: A jumbo token for fast plain vision transformers. In *International Conference on Learning Representations*, 2026.

Ryan Golden, Jean Erik Delanois, Pavel Sanda, and Maxim Bazhenov. Sleep prevents catastrophic forgetting in spiking neural networks by forming a joint synaptic weight representation. *PLOS Computational Biology*, 18(11):e1010628, 2022.

Dong Gong, Qingsen Yan, Yuhang Liu, Anton van den Hengel, and Javen Qinfeng Shi. Learning Bayesian sparse networks with full experience replay for continual learning. *arXiv preprint arXiv:2202.10203*, 2022.

Advait Gupta, Rishie Raj, Dang Nguyen, and Tianyi Zhou. FASTA*: Fast-slow toolpath agent with subroutine mining for efficient multi-turn image editing. In *International Conference on Learning Representations*, 2026.

Pouya Hamadanian, Arash Nasr-Esfahany, Malte Schwarzkopf, Siddhartha Sen, and Mohammad Alizadeh. Online reinforcement learning in non-stationary context-driven environments. In *International Conference on Learning Representations*, 2025.

Tyler L. Hayes, Giri P. Krishnan, Maxim Bazhenov, Hava T. Siegelmann, Terrence J. Sejnowski, and Christopher Kanan. Replay in deep learning: Current approaches and missing biological elements. *Neural Computation*, 33(11):2908-2950, 2021.

Tyler L. Hayes, Nathan D. Cahill, and Christopher Kanan. Memory efficient experience replay for streaming learning. *arXiv preprint arXiv:1809.05922*, 2019.

Nitin Kamra, Umang Gupta, and Yan Liu. Deep generative dual memory network for continual learning. *arXiv preprint arXiv:1710.10368*, 2018.

Christos Kaplanis, Claudia Clopath, and Murray Shanahan. Continual reinforcement learning with multi-timescale replay. *arXiv preprint arXiv:2004.07530*, 2020.

Ronald Kemker and Christopher Kanan. FearNet: Brain-inspired model for incremental learning. In *International Conference on Learning Representations*, 2018.

Ronald Kemker, Marc McClure, Angelina Abitino, Tyler L. Hayes, and Christopher Kanan. Measuring catastrophic forgetting in neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

Łukasz Korycki and Bartosz Krawczyk. Class-incremental experience replay for continual learning under concept drift. *arXiv preprint arXiv:2104.11861*, 2021.

Eugene Lee, Cheng-Han Huang, and Chen-Yi Lee. Few-shot and continual learning with attentive independent mechanisms. *arXiv preprint arXiv:2107.14053*, 2021.

Chunmao Li, Yang Li, Yinliang Zhao, Peng Peng, and Xupeng Geng. SLER: Self-generated long-term experience replay for continual reinforcement learning. *Applied Intelligence*, 2020.

Xingyu Li, Bo Tang, and Haifeng Li. AdaER: An adaptive experience replay approach for continual lifelong learning. *Neurocomputing*, 572:127204, 2024.

Yijie Lu, Xinhao Luo, Yixing Zhang, Zhiyan Wang, Wentao Li, Yanhan Wang, Zhi Liu, Zhaokun Zhou, and Guoqi Li. SMixer: Rethinking efficient-training and event-driven SNNs. In *International Conference on Learning Representations*, 2026.

Irene Muñoz-Martin, Stefano Bianchi, Shahin Hashemkhani, Giacomo Pedretti, Octavian Melnic, and Daniele Ielmini. A brain-inspired homeostatic neuron based on phase-change memories for efficient neuromorphic computing. *Frontiers in Neuroscience*, 15:709053, 2021.

Xing Nie, Shixiong Xu, Xiyan Liu, Gaofeng Meng, Chunlei Huo, and Shiming Xiang. Bilateral memory consolidation for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16026-16035, 2023.

German I. Parisi, Jun Tani, Cornelius Weber, and Stefan Wermter. Lifelong learning of spatiotemporal representations with dual-memory recurrent self-organization. *Frontiers in Neurorobotics*, 12:78, 2018.

German I. Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54-71, 2019.

Jiyuan Pei, Yi Mei, Jialin Liu, Mengjie Zhang, and Xin Yao. Lifelong learning with behavior consolidation for vehicle routing. In *International Conference on Learning Representations*, 2026.

Jian Peng, Dingqi Ye, Bo Tang, Yinjie Lei, Yu Liu, and Haifeng Li. Lifelong learning with cycle memory networks. *IEEE Transactions on Neural Networks and Learning Systems*, 35(11):16439-16452, 2024.

Appan Rakaraddi, Lam Siew Kei, Mahardhika Pratama, and Marcus de Carvalho. Reinforced continual learning for graphs. *arXiv preprint arXiv:2209.01556*, 2022.

Sahithya Ravi, Aditya Chinchure, Raymond T. Ng, Leonid Sigal, and Vered Shwartz. SPIKE-RL: Video-LLMs meet Bayesian surprise. In *International Conference on Learning Representations*, 2026.

Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. Learning to learn without forgetting by maximizing transfer and minimizing interference. In *International Conference on Learning Representations*, 2019.

Mohammad Rostami, Soheil Kolouri, and Praveen K. Pilly. Complementary learning for overcoming catastrophic forgetting using experience replay. *arXiv preprint arXiv:1903.04566*, 2019.

Fahad Sarfraz, Elahe Arani, and Bahram Zonooz. Sparse coding in a dual memory system for lifelong learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023.

Gan Sun, Yang Cong, Jiahua Dong, Yuyang Liu, Zhengming Ding, and Haibin Yu. What and how: Generalized lifelong spectral clustering via dual memory. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3895-3908, 2022.

Hao Wang, Chengkai Hou, Xianglong Li, Yankai Fu, Chenxuan Li, Ning Chen, Gaole Dai, Jiaming Liu, Tiejun Huang, and Shanghang Zhang. SpikePingpong: Spike vision-based fast-slow pingpong robot system. In *International Conference on Learning Representations*, 2026.

Miao Wang, Gangyi Ding, Yunlin Lei, Yu Zhang, Lanyu Gao, and Xu Yang. Brain-inspired reward broadcasting: Brain learning mechanism guides learning of spiking neural network. *Neurocomputing*, 629:129664, 2025.

Qiang Wang, Jiayi Liu, Zhong Ji, Yanwei Pang, and Zhongfei Zhang. Hierarchical correlations replay for continual learning. *Knowledge-Based Systems*, 250:109052, 2022.

Zhizheng Wang, Yuanyuan Sun, Xiaokun Zhang, Bo Xu, Zhihao Yang, and Hongfei Lin. Continual learning with high-order experience replay for dynamic network embedding. *Pattern Recognition*, 159:111093, 2025.

Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. DualPrompt: Complementary prompting for rehearsal-free continual learning. In *European Conference on Computer Vision*, 2022.

Meng Wei, Chenyang Wan, Jiaqi Peng, Xiqian Yu, Yuqiang Yang, Delin Feng, Wenzhe Cai, Chenming Zhu, Tai Wang, Jiangmiao Pang, and Xihui Liu. Ground slow, move fast: A dual-system foundation model for generalizable vision-and-language navigation. In *International Conference on Learning Representations*, 2026.

Ya Wu, Ariyan Bighashdel, Guang Chen, Gijs Dubbelman, and Pavol Jancura. Continual pedestrian trajectory learning with social generative replay. *IEEE Robotics and Automation Letters*, 8(2):848-855, 2023.

Meng Xu, Xinhong Chen, Bingyi Liu, Yi-Rong Lin, Yung-Hui Li, and Jianping Wang. A unified experience replay framework for spiking deep reinforcement learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(4):4275-4290, 2026.

Biao Yang, Fucheng Fan, Rongrong Ni, Jie Li, Loochu Kiong, and Xiaofeng Liu. Continual learning-based trajectory prediction with memory augmented networks. *Knowledge-Based Systems*, 258:110022, 2022.

Da Yu, Mingyi Zhang, Mantian Li, Fusheng Zha, Junge Zhang, Lining Sun, and Kaiqi Huang. Contrastive correlation preserving replay for online continual learning. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(1):124-137, 2024.

Weixia Zhang, Dingquan Li, Chao Ma, Guangtao Zhai, Xiaokang Yang, and Kede Ma. Continual learning for blind image quality assessment. *arXiv preprint arXiv:2102.09717*, 2021.

Fan Zhou and Chengtai Cao. Overcoming catastrophic forgetting in graph neural networks with experience replay. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

APPENDIX

A COMPLETE METHOD AND TRAINING PROCEDURE

A.1 PROBLEM SETTING AND RECURRENT SPIKING EXECUTION

We consider a continuous-control process with time-varying dynamics. At time t, the state, action, and reward are denoted by s_t, a_t, and r_t, respectively, and the latent dynamics condition is denoted by ξ_t. At decision time, the policy does **not** receive ξ_t, a task identifier, or the true dynamics parameters. It acts only on the observed history

$$h_t = (s_{t-H+1}, \ldots, s_t) \tag{S1}$$

where H is the history-window length. The underlying reinforcement-learning objective remains the maximization of discounted cumulative return:

$$J(\pi) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r_t \right], \qquad \gamma \in [0, 1). \tag{S2}$$

The core SpikeARC process can therefore be summarized as a progression from mature behavior to fast correction, responsibility assignment, and consistency-guided consolidation.

Throughout this process, behavior budgets constrain the plasticity of the Stable pathway. Crucially, the system need not classify a change as temporary or permanent when it first appears. Instead, it initially expresses the change as a local and reversible correction, and subsequent evidence gradually determines whether that correction qualifies for long-term write-back.

Stable default behavior. The Stable branch implements mature long-term control behavior. Let $\phi_S(h_t)$ denote the state representation used by the underlying reinforcement-learning backbone. The Stable action mean is

$$\mu_t^S = f_S(\phi_S(h_t); \theta_S). \tag{S3}$$

Here, θ_S denotes the Stable parameters. In the implementation, the PPO version uses a representation comprising the current state, the history mean, and the difference between the first and last states in the window. The SAC version uses the current state as the Stable input. These differences are adaptations to the backbone and do not alter the Stable-Fast composition mechanism described below.

Fast recurrent-LIF correction. Rather than learning a second complete policy, the Fast branch produces a candidate action correction from the most recent H states. For the kth time step in the history window, the discrete dynamics of the recurrent leaky integrate-and-fire (LIF) units in Fast are

$$i_k^F = W_x^F s_k + W_r^F z_{k-1}^F + b_F, \tag{S4}$$

$$u_k^F = \beta_F u_{k-1}^F + i_k^F, \qquad z_k^F = \mathbb{H}\left(u_k^F - \vartheta_F\right), \tag{S5}$$

$$u_k^F \leftarrow u_k^F \left(1 - sg\left(z_k^F\right)\right). \tag{S6}$$

The variables u_k^F and z_k^F denote the membrane potential and spike output, respectively; $\mathbb{H}(\cdot)$ is the Heaviside function; and $sg(\cdot)$ denotes the stop-gradient operator. During backpropagation, the implementation uses the locally bounded surrogate derivative $\partial \mathbb{H}(x)/\partial x \approx (1 + |x|)^{-2}$. The Fast temporal feature concatenates the mean spike activity with the final spike output:

$$\phi_t^F = \left[\frac{1}{H} \sum_{k=1}^{H} z_k^F \,;\, z_H^F \right]. \tag{S7}$$

A linear readout then maps this temporal feature to the candidate correction:

$$\Delta\mu_t^F = W_\Delta \phi_t^F + b_\Delta. \tag{S8}$$

Spike-guided Fusion. The Fast pathway does not intervene with equal strength in every state. The Fusion pathway uses a separate recurrent-LIF process that receives both the state history and the Fast spike sequence:

$$x_k^G = \left[s_k \,;\, sg\left(z_k^F\right) \right]. \tag{S9}$$

Fusion applies an LIF update of the same form as above to obtain the temporal feature ϕ_t^G, from which it produces the gate

$$g_t = \sigma\left(w_g^\top \phi_t^G + b_g\right), \qquad g_t \in [0,1]. \tag{S10}$$

The Fast spike sequence is stop-gradient at this input. Fusion can therefore use the temporal state of the Fast spikes to determine when a correction is needed, without changing the Fast spiking dynamics through this auxiliary route. The Fast correction applied to the action is

$$c_t = g_t \Delta\mu_t^F. \tag{S11}$$

The final mean used for action sampling is

$$\mu_t = \mu_t^S + c_t = \mu_t^S + g_t \Delta\mu_t^F. \tag{S12}$$

Continuous actions are sampled as $a_t = tanh\left(\mu_t + \sigma_t \odot \epsilon\right)$, where $\epsilon \sim \mathcal{N}(0, I)$. The PPO or SAC backbone determines the parameterization of σ_t. During base-stage training, Fast and Fusion are disabled and the Fast readout is initialized to zero. Consequently, when the system switches from a mature Stable policy to continual adaptation, enabling the new pathways does not immediately change the existing action mean. Fast and Fusion are then activated, allowing a new dynamics change to be expressed first through c_t as a local, state-dependent, and reversible correction.

A.2 Contribution-Based Replay Responsibility

The forward Stable-Fast decomposition does not by itself produce a corresponding decomposition of optimization responsibility. If a historical behavior was formed primarily through a Fast correction but its replay error is still written in full to Stable, a temporary change can still be absorbed prematurely into the long-term parameters. We therefore estimate the Fast responsibility of each replay sample explicitly.

For every replay sample, the behavior memory stores

$$\mathcal{B}_i = \left(h_i, \bar{\mu}_i, \bar{\mu}_i^S, \bar{g}_i\right). \tag{S13}$$

Here, $\bar{\mu}_i$ is the full policy mean at sampling time, $\bar{\mu}_i^S$ is the contemporaneous Stable mean, and $\bar{g}_i$ is the contemporaneous Fusion gate. Their difference,

$$\bar{c}_i = \bar{\mu}_i - \bar{\mu}_i^S, \tag{S14}$$

is the gated Fast correction that actually entered the action at sampling time. We use it to compute the relative Fast contribution to the behavior:

$$q_i = \frac{\| \bar{c}_i \|_2}{\| \bar{\mu}_i^S \|_2 + \| \bar{c}_i \|_2 + \varepsilon_r}. \tag{S15}$$

A small responsibility threshold f is introduced for normalization:

$$\rho_i = clip\left(\frac{q_i - f}{1 - f}, 0, 1\right). \tag{S16}$$

The implementation uses $\varepsilon_r = 10^{-3}$ and $f = 0.05$ by default. When Fast barely changes the Stable behavior, ρ_i is close to 0. When the historical action change was produced mainly by the Fast correction, ρ_i is close to 1.

The responsibility variable does not alter the forward behavior. It changes only the gradient that the replay loss propagates to Stable. For the current policy outputs μ_i and μ_i^S on sample i, define

$$\widetilde{\mu}_i = (1 - \rho_i)\, \mu_i^S + \rho_i sg\left(\mu_i^S\right) + \left(\mu_i - \mu_i^S\right). \tag{S17}$$

Because $\mu_i - \mu_i^S$ is exactly the current Fast/Fusion correction, it contains no Stable gradient path algebraically. Hence,

$$\frac{\partial \widetilde{\mu}_i}{\partial \mu_i^S} = 1 - \rho_i. \tag{S18}$$

The behavior-replay loss is

$$\mathcal{L}_{rep} = \frac{1}{B} \sum_{i=1}^{B} \| \widetilde{\mu}_i - \bar{\mu}_i \|_2^2 . \tag{S19}$$

Stable thus receives direct replay credit proportional to $1 - \rho_i$ on each replay sample, while Fast and Fusion continue to fit the full historical behavior through the actual correction path. We do not impose an artificial symmetric allocation in which Stable receives $1 - \rho$ and Fast receives ρ; the Fast update is still determined by the true computation graph.

We call this mechanism *Spike-Guided Responsibility Routing*. The quantity $\bar{c}_i$ is not a heuristic independent of the spiking neural network. It is the actual action correction jointly produced by the recurrent-LIF Fast dynamics and the Fusion gate conditioned on the Fast spike sequence. At the same time, ρ_i is computed from the correction contribution in **action space**, rather than directly from the raw firing rate or spike count.

Importantly, ρ_i is a sample-level replay responsibility. The base actor gradient of online PPO or SAC is not routed sample by sample using this value. Instead, the overall plasticity of Stable under the current reinforcement-learning objective is regulated by the behavioral-risk controller in Appendix A.4. These two mechanisms answer different questions: which pathway should absorb a historical behavior error, and how much may Stable change under the current objective?

A.3 Consistency-Guided Selective Consolidation

Responsibility routing prevents Fast-dominated historical behavior from being written prematurely into Stable. If Stable were always protected, however, the system would ultimately depend on Fast corrections indefinitely. SpikeARC therefore introduces consistency-guided consolidation, under which corrections with stable structure gradually become eligible for incorporation into the long-term policy.

For replay sample i, let the historical correction be $\bar{c}_i = \bar{\mu}_i - \bar{\mu}_i^S$, and let the current policy correction on the same historical state be

$$c_i^{cur} = \mu_i - \mu_i^S. \tag{S20}$$

The maturity proxy in the implementation jointly considers historical gate usage, directional consistency, magnitude consistency, and whether the correction was meaningfully activated:

$$w_i^g = clip\left(\frac{\bar{g}_i - 0.10}{0.50}, 0, 1\right), \tag{S21}$$

$$d_i = clip\left(\frac{\bar{c}_i^\top c_i^{cur}}{\| \bar{c}_i \|_2 \| c_i^{cur} \|_2 + \varepsilon_c}, 0, 1\right), \tag{S22}$$

$$a_i = \frac{\min\left(\| \bar{c}_i \|_2, \| c_i^{cur} \|_2\right)}{\max\left(\| \bar{c}_i \|_2, \| c_i^{cur} \|_2\right) + \varepsilon_c}, \qquad I_i = \mathbf{1}\left[\| \bar{c}_i \|_2 > 10^{-4}\right]. \tag{S23}$$

These quantities give

$$m_i = clip\left(w_i^g d_i a_i I_i, 0, 1\right). \tag{S24}$$

When m_i is computed, both the historical and current corrections are treated as stop-gradient statistics. Maturity therefore determines the write-back strength without introducing an additional backward path through the comparison itself. A large m_i indicates that Fusion used the correction at sampling time and that the current policy still produces a correction with a similar direction and magnitude in the same state.

Let $p \in [0, 1]$ denote progress through the current adaptation stage. The implementation uses the smooth schedule

$$s(p) = p^2(3 - 2p), \qquad \alpha_c(p) = 0.02 + 0.13\, s(p). \tag{S25}$$

The Stable consolidation target is

$$y_i^S = sg\left(\mu_i^S\right) + \alpha_c(p) m_i \left(\bar{\mu}_i - sg\left(\mu_i^S\right)\right), \tag{S26}$$

with the corresponding loss

$$\mathcal{L}_{con} = \frac{1}{B} \sum_{i=1}^{B} \| \mu_i^S - y_i^S \|_2^2 . \tag{S27}$$

Consolidation therefore neither copies the Fast parameters nor writes the complete Fast correction into Stable in a single step. Instead, it moves Stable by a small step in behavior-output space toward the historical full policy, with the step size controlled by $\alpha_c(p) m_i$.

A.4 Dual Behavioral-Drift Budgets and Stable Plasticity

Reference for mature Stable behavior. After base-stage training, we disable Fast and Fusion and use the mature policy to collect a fixed Stable anchor bank:

$$\mathcal{A}_S = \left\{\left(h_k, \bar{\mu}_k^S\right)\right\}_{k=1}^{N_S}. \tag{S28}$$

The normalized drift of the current Stable policy from this reference is

$$D_S = \frac{\mathbb{E}_{(h, \bar{\mu}^S) \sim \mathcal{A}_S}\left[\| \mu^S(h) - \bar{\mu}^S \|_2^2\right]}{\max\left(\mathbb{E}_{\mathcal{A}_S}\left[\| \bar{\mu}^S \|_2^2\right], 10^{-3}\right)}. \tag{S29}$$

This measure directly protects the long-term behavioral substrate by limiting arbitrary changes to the mature Stable output under nominal states during subsequent adaptation.

Historical full-policy budget. Protecting Stable alone is insufficient because the environment executes the full policy $\mu = \mu^S + c$. For each completed historical stage j, a set of historical states is sampled continually throughout training. At the end of the stage, the **final policy** for that stage generates reference targets on these states, yielding

$$\mathcal{A}_{P,j} = \{(h_k, \bar{\mu}_{j,k})\}_{k=1}^{N_j}. \tag{S30}$$

The relative drift of the current full policy on historical stage j is

$$D_{P,j} = \frac{\mathbb{E}_{(h,\bar{\mu})\sim\mathcal{A}_{P,j}}\left[\| \mu(h) - \bar{\mu} \|_2^2\right]}{\max\left(\mathbb{E}_{\mathcal{A}_{P,j}}\left[\| \bar{\mu} \|_2^2\right], 10^{-3}\right)}. \tag{S31}$$

To prevent several well-preserved historical stages from masking one stage with severe drift, we protect the historical group with the largest drift rather than averaging across groups:

$$D_P = \max_{j\in\mathcal{H}} D_{P,j}. \tag{S32}$$

Here, $\mathcal{H}$ is the set of stages that have entered the historical reference collection. Stage indices are used only for grouped aggregation within the historical references; they are not supplied to the actor and do not participate in responsibility routing.

Soft behavior budgets. The two drift measures satisfy

$$D_S \le \epsilon_S, \qquad D_P \le \epsilon_P, \tag{S33}$$

and are associated with nonnegative dual variables λ_S and λ_P. For either drift quantity D, the budget controller is

$$\mathcal{B}(D; \lambda, \epsilon, \rho) = \lambda D + \frac{\rho}{2}\max(D - \epsilon, 0)^2, \tag{S34}$$

where the quadratic term is activated only when the budget is violated. The dual variable is updated as

$$\lambda \leftarrow clip\left(\lambda + \eta_\lambda(D - \epsilon), 0, \lambda_{\max}\right). \tag{S35}$$

The behavior budget enters the optimization as a soft constraint rather than a hard projection of the policy back into the feasible set after each update. When behavioral drift is below budget, the dual variable can decrease with the constraint slack. When drift exceeds the budget, the quadratic penalty is activated and the dual variable increases, adaptively strengthening the pressure to preserve behavior.

Stable gradient modulation under behavioral-risk constraints. SpikeARC dynamically scales the reinforcement-learning gradient of the Stable pathway according to behavior-budget utilization. Let the base Stable RL-gradient schedule be

$$\alpha_{base}(p) = 0.20 + 0.30\, s(p), \tag{S36}$$

where $s(p) = p^2(3 - 2p)$. Let $\widehat{D}_S$ and $\widehat{D}_P$ denote the most recent drift readings available to the controller. Budget utilization is

$$u = max\left(\frac{\widehat{D}_S}{\epsilon_S}, \frac{\widehat{D}_P}{\epsilon_P}\right). \tag{S37}$$

For the release threshold τ and the upper bound c on Stable plasticity, define

$$w = clip\left(\frac{\tau - u}{\tau - 0.25}, 0, 1\right), \tag{S38}$$

$$\alpha_S = \alpha_{base} + w\left(\max\left(\alpha_{base}, c\right) - \alpha_{base}\right). \tag{S39}$$

The current implementation uses $c = 0.85$ and $\tau = 0.90$ by default. When both behavior budgets have sufficient slack, Stable can receive a larger update than under the fixed base Stable RL-gradient schedule. As either budget approaches its boundary, this additional release is withdrawn smoothly and Stable returns to the conservative base schedule. Stable is never completely frozen. SpikeARC therefore implements **Fast-leading, Stable-co-adapting** continual adaptation, rather than a strictly Fast-only phase followed by Stable learning.

For the base reinforcement-learning actor loss $\mathcal{L}_{RL}$, we first perform ordinary backpropagation and then apply the following operation only to this gradient component for the Stable parameter group:

$$g_S^{RL} \leftarrow \alpha_S g_S^{RL}. \tag{S40}$$

The base RL gradients for Fast and Fusion retain their original magnitude. The replay, consolidation, and behavior-budget terms are then backpropagated. Thus, α_S **scales only the contribution of the base RL objective to the Stable gradient**; it does not scale down the consolidation or retention gradients.

A.5 Joint Optimization and Backbone Compatibility

After combining the mechanisms above, the replayed-behavior losses used to reproduce historical behavior and support long-term consolidation can be written as

$$\mathcal{L}_{beh} = \mathcal{L}_{rep} + \lambda_{con}\mathcal{L}_{con}. \tag{S41}$$

The SpikeARC auxiliary objective is then

$$\mathcal{L}_{aux} = \lambda_{rep}\mathcal{L}_{beh} + \mathcal{B}\left(D_S; \lambda_S, \epsilon_S, \rho_S\right) + \mathcal{B}\left(D_P; \lambda_P, \epsilon_P, \rho_P\right). \tag{S42}$$

SpikeARC does not simply add $\mathcal{L}_{RL}$ and $\mathcal{L}_{aux}$ and multiply their sum by a single slow coefficient. Its Stable gradient is more accurately expressed as

$$g_S = \alpha_S \nabla_{\theta_S}\mathcal{L}_{RL} + \nabla_{\theta_S}\mathcal{L}_{aux}. \tag{S43}$$

Fast and Fusion continue to receive the relevant gradients from $\mathcal{L}_{RL}$, replay, and the full-policy budget through the true computation graph. Scaling Stable's base RL gradient before adding the behavior-preservation and consolidation gradients assigns distinct optimization roles to rapid adaptation and long-term write-back.

SpikeARC operates at the levels of actor-level behavior decomposition, responsibility-routed replay, and behavioral constraints, and can therefore be applied to different continuous-control backbones. The PPO and SAC implementations use the same Stable mean plus gated Fast correction, responsibility-routed replay, consistency-guided selective consolidation, and dual behavioral-drift budgets. Because SAC has a higher update-to-data ratio, its controller is updated at fixed intervals. Between controller updates, it reuses the most recent safe Stable RL-gradient scale, preventing the auxiliary mechanisms from receiving a disproportionate number of optimization updates.

A.6 Expanded Training Procedure

The mechanisms above can be described independently of the reinforcement-learning backbone. We use $\mathcal{L}_{RL}$ for the actor objective supplied by the underlying algorithm; PPO and SAC are two instantiations used in our experiments. SpikeARC addresses four questions: how to construct a Stable-Fast composed action, how to assign replay responsibility from historical corrections, how to selectively

write consistent corrections back to Stable, and how to regulate long-term Stable plasticity with two behavior budgets.

Algorithm 1. Continual Policy Adaptation with SpikeARC

Step	Operation
Input	Nominal environment E_0; sequence of subsequent environments $\{E_j\}$.
Output	Composed policy π_θ.
1	Train the Stable pathway in nominal environment E_0 and set the Fast output layer to zero. Add samples from the base stage to replay memory $\mathcal{M}$, and record both types of behavioral references as described in Section 3.3.
2	**for** each subsequent environment E_j **do**
3	**for** each sampling and update round **do**
4	Generate and sample from the composed action as described in Section 3.1. Record the historical composed and Stable action means and gate: $\mu_t = \mu_t^S + c_t$, $c_t = g_t \Delta\mu_t^F$, $g_t \in [0, 1]$.
5	Sample from $\mathcal{M}$. Compute the responsibility weight from the correction contribution as described in Section 3.2, modify only the Stable replay gradient, and obtain $\mathcal{L}_{rep}$.
6	Reevaluate the consistency of the historical and current corrections as described in Section 3.2, and compute the maturity score and selective consolidation loss $\mathcal{L}_{con}$.
7	Determine α_S from the most recently available drift as described in Section 3.3, and then compute both behavioral drifts and their budget losses for the current round.
8	Combine the auxiliary objectives as described in Section 3.4. Scale Stable's base gradient before adding the auxiliary gradient: $g_S = \alpha_S \nabla_{\theta_S} \mathcal{L}_{RL} + \nabla_{\theta_S} \mathcal{L}_{aux}$, and update the budget dual variables.
9	Update the parameters using the selected training backbone. After the optimization round, add new samples to $\mathcal{M}$.
10	**end for**; at the end of the stage, record the composed-policy behavior as described in Section 3.3.
11	**end for**

B Experimental Protocol and Complete Results

In the current implementation, PPO and SAC share the Stable pathway plus gated Fast correction, responsibility-routed replay, consistency-guided selective consolidation, and dual behavioral-drift budgets described above. They differ only in backbone-specific details, such as the controller execution frequency. The simulated quadruped implementation retains the same Stable-Fast-Fusion organization and the same division of roles from correction to consolidation, but uses a domain-specific output head. It is therefore a structured instantiation of the same framework.

B.1 Protocol and Reporting Conventions

The common experimental protocol comprises 1 million environment steps of nominal pretraining, followed sequentially by low-friction, heavy, weak-actuator, and mixed-shift dynamics stages. Each adaptation stage has a budget of 300,000 environment steps. At decision time, the policy receives no task or stage identifier, true dynamics parameters, or oracle context. The main PPO experiments cover HalfCheetah-v5, Walker2d-v5, Hopper-v5, CARL-DMC-Walker, and CARL-DMC-Quadruped. The external comparisons on CARL Walker and CARL Quadruped use SAC.

Adaptation AUC integrates the current-task return curve across the four change stages and measures the speed and quality of adaptation within a finite interaction budget. Mean positive forgetting measures the positive loss of performance on previously learned tasks from their historical best performance to the end of the sequence. Stable parameter drift measures the extent to which the long-term branch is altered relative to the mature policy. Forgetting characterizes behavioral retention, whereas drift characterizes parameter reconfiguration; the two are not equivalent. Tables report mean $\pm$ SD across independently trained seeds, and training curves show mean $\pm$ 1 SD. The five-environment study uses matched mature-policy initialization. Each network width in the capacity

Table S1: Experimental strands and the question addressed by each comparison. The three strands provide external, controlled-mechanism, and embodied-behavior evidence, respectively.

Experimental strand	Environment or setting	Primary comparison	Question addressed
External baseline comparison	CARL Walker and Quadruped	SpikeARC (SAC), MTR, and 3RL	Does the method retain its adaptation ability and reduce forgetting under a different backbone?
Internally controlled evidence	Five primary environments, three MuJoCo ablation environments, and Hopper/Quadruped capacity controls	SpikeARC, Fully Shared, replay/EWC, routing-consolidation variants, and a plain shared MLP	Do the gains arise from how changes are absorbed, rather than from ordinary regularization, an extra branch, or greater parameter capacity?
Embodied quadruped robot	Rough-transition locomotion	Full and No Routing	Does responsibility routing produce an interpretable adaptation-retention tradeoff in behavior?

sweep is trained from an independently learned nominal base, and each method in the external SAC comparison also trains its own nominal policy.

The matched-base PPO study uses three independently trained seeds in each of the five environments. Within each environment and seed, SpikeARC and Fully Shared start from the same mature checkpoint; the primary comparison is therefore paired by random seed. A single evaluation episode or trajectory is used only to reduce the variance of policy evaluation. It is not treated as an independent algorithmic sample or statistical replicate.

The adaptation budget for each dynamics stage is fixed at 300,000 environment steps. Adaptation AUC measures the speed and quality of adaptation within this finite window and does not assume that the policy has reached asymptotic convergence by the end of the stage. Differences in return across seeds therefore cannot be interpreted as evidence that a particular training run failed to converge.

B.2 Hidden-Context SAC Comparison

The controlled primary experiments use PPO, but SpikeARC is defined through the actor's Stable-Fast behavior decomposition, replay responsibility, and consolidation and behavior-budget layers, rather than through a particular policy-gradient formula. To test whether the findings depend on PPO, we implemented a SAC version on CARL Walker and CARL Quadruped and compared it with MTR and 3RL under a common hidden-context protocol. Each method completed its own nominal training and then experienced the same four-stage dynamics sequence under the same total interaction budget. The formal LCPO setting uses observed context to construct local policy constraints and therefore has a stronger information condition. Accordingly, it is reported only as a privileged-context reference in Appendix C.4 and is not included in the direct ranking in this section.

On CARL Walker, the adaptation AUC values of the three methods are already close, at approximately 948-954. Retention quality is therefore more discriminative at this similar level of adaptation. SpikeARC achieves a mean AUC of 953.8 ± 12.3, together with the lowest forgetting (4.65 ± 5.31) and the highest final average return (963.1 ± 5.3). The AUC values for MTR and 3RL are 948.5 ± 5.1 and 947.8 ± 6.3, respectively. The principal difference on Walker is therefore that SpikeARC further reduces the loss of historical capabilities while preserving virtually the same level of adaptation.

The differences among methods are more pronounced on CARL Quadruped. SpikeARC obtains an adaptation AUC of 925.2 ± 8.2, exceeding 922.0 ± 1.4 for MTR and 871.1 ± 32.9 for 3RL. Its

Table S2: Hidden-context CARL comparison using SAC (mean ± SD over three independently trained seeds). Each method trains its own nominal policy. Training trajectories appear in Figure 3 of the main paper.

Environment	Method	Adaptation AUC ↑	Forgetting ↓	Final average return ↑
Walker	SpikeARC (SAC)	953.8 ± 12.3	4.65 ± 5.31	963.1 ± 5.3
Walker	MTR	948.5 ± 5.1	6.21 ± 6.71	953.5 ± 6.2
Walker	3RL	947.8 ± 6.3	6.83 ± 4.37	955.8 ± 1.3
Quadruped	SpikeARC (SAC)	925.2 ± 8.2	8.75 ± 12.47	947.6 ± 22.0
Quadruped	MTR	922.0 ± 1.4	22.92 ± 20.76	927.5 ± 18.8
Quadruped	3RL	871.1 ± 32.9	24.39 ± 23.66	888.6 ± 43.7

mean positive forgetting is 8.75 ± 12.47, lower than 22.92 ± 20.76 and 24.39 ± 23.66 for the two baselines, and its final average return is 947.6 ± 22.0. The AUC values for the three SpikeARC seeds are 930.2, 929.7, and 915.8.

The external SAC comparison and the subsequent primary PPO study serve different purposes. The former tests whether the Stable-Fast organization preserves a competitive adaptation-retention profile after the optimization backbone is changed. The latter uses matched initialization to isolate the effect of the mechanism itself.

B.3 Five-Environment Matched-Base Results

For every environment and random seed, SpikeARC and Fully Shared start from the same mature-policy checkpoint. Both methods then experience the same low-friction, heavy, weak-actuator, and mixed-shift dynamics sequence and receive the same 300,000-step adaptation budget in each stage. This paired design minimizes the influence of initial policy quality and focuses the comparison on the continual-adaptation mechanism itself. It evaluates both adaptation to new dynamics under a finite budget and the extent to which that adaptation requires changes to the mature long-term policy.

Figure 4 of the main paper shows that the macro adaptation AUC of SpikeARC is 9.2% lower than that of Fully Shared, meaning that SpikeARC retains 90.8% of the shared policy's adaptation AUC. The relative adaptation cost across the five environments ranges from 5.1% to 18.0%. The subsequent gains in long-term stability therefore cannot be explained by stopping learning or freezing Stable: under the same interaction budget, SpikeARC retains most of the continual-adaptation capability.

The result most directly aligned with the method design concerns long-term policy write-back. Stable parameter drift is lower for SpikeARC than for Fully Shared in all five environments. The environment-level reduction ranges from 29.7% to 42.0%, the macro reduction is 35.3%, and all 15 matched seed pairs in Appendix B.4 have the same direction of effect. SpikeARC therefore does not prevent Stable from learning; it reduces the immediate, unconditional write-back of environmental changes into the mature policy.

Behavioral forgetting tests whether controlled write-back also improves the retention of historical behavior. SpikeARC reduces forgetting on HalfCheetah, Walker2d, Hopper, and CARL Walker; CARL Quadruped is the only environment with the opposite result. The descriptive macro reduction across the five environments is 36.2%. Stable drift and forgetting should be interpreted as complementary metrics. Reducing long-term rewriting may lower the risk of interference with mature behavior and improves retention in most environments, but a larger parameter reconfiguration can also discover a behavioral solution that still performs old tasks. Bootstrap uncertainty and complete seed-level values are reported in Appendix B.4.

Taken together, these experiments support a bounded conclusion. SpikeARC retains most of the adaptation capability while exerting more consistent control over long-term policy write-back across environments and seeds. In most environments, this controlled updating also produces lower behavioral forgetting. The method changes the path by which new behavior enters the mature policy, rather than simply reducing overall policy plasticity.

Table S3: Five-environment matched-base PPO results (mean ± SD over three independently trained seeds). Each cell reports SpikeARC / Fully Shared. Within each environment and seed, the paired methods start from the same mature checkpoint.

Environment	Adaptation AUC	Forgetting	Stable drift
HalfCheetah	2620.1 ± 1259.5 / 2831.3 ± 1448.9	276.1 ± 404.4 / 855.4 ± 665.9	0.223 ± 0.012 / 0.318 ± 0.006
Walker2d	580.6 ± 179.6 / 708.0 ± 347.4	139.5 ± 138.8 / 209.6 ± 180.2	0.215 ± 0.005 / 0.364 ± 0.011
Hopper	522.6 ± 277.6 / 566.2 ± 103.7	61.1 ± 36.6 / 190.4 ± 81.7	0.178 ± 0.020 / 0.265 ± 0.008
CARL Walker	239.0 ± 55.7 / 251.9 ± 59.9	7.5 ± 4.3 / 12.6 ± 16.2	0.193 ± 0.006 / 0.333 ± 0.005
CARL Quadruped	294.8 ± 84.1 / 318.8 ± 107.1	117.2 ± 45.0 / 90.7 ± 42.0	0.251 ± 0.007 / 0.363 ± 0.019

Table S4: Environment-level effects relative to Fully Shared. AUC change is zero at parity; positive reductions in forgetting and drift favor SpikeARC. Macro effects are equal-weight averages over the five environments.

Environment	AUC change vs. Shared	Forgetting reduction	Stable-drift reduction
HalfCheetah	-7.46%	+67.72%	+29.67%
Walker2d	-18.00%	+33.47%	+41.02%
Hopper	-7.70%	+67.89%	+32.83%
CARL Walker	-5.09%	+40.95%	+42.04%
CARL Quadruped	-7.53%	-29.23%	+30.73%
Macro average	-9.16%	+36.16%	+35.26%

Table S5: Macro effects and environment-stratified paired-bootstrap 95% confidence intervals for the five-environment matched-base PPO study.

Macro metric	**Point estimate**	**95% paired-bootstrap CI**	**Interpretation**
Adaptation AUC retention ↑	90.84%	83.04%-98.61%	Most adaptation is retained, with a visible adaptation cost.
Forgetting reduction ↑	36.16%	-46.37%-53.50%	Forgetting is lower in four of five environments, with high uncertainty.
Stable-drift reduction ↑	35.26%	33.23%-37.09%	The direction is consistent in all five environments and all 15 matched pairs.

B.4 Paired Bootstrap and Seed-Level Results

To quantify the sensitivity of the prespecified cross-environment macro effect to training randomness, we use an environment-stratified paired bootstrap. The three matched SpikeARC-Fully Shared seed pairs in each environment form the basic resampling units. In each bootstrap iteration, three pairs are sampled with replacement within each environment, and the outcomes of both methods are selected jointly.

For bootstrap replicate b, the environment-level effects are first computed as

$$R_{AUC}^{(e,b)} = \frac{\overline{AUC}_{SpikeARC}^{(e,b)}}{\overline{AUC}_{Shared}^{(e,b)}}, \tag{11}$$

$$R_{F}^{(e,b)} = 1 - \frac{\bar{F}_{SpikeARC}^{(e,b)}}{\bar{F}_{Shared}^{(e,b)}}, \tag{12}$$

$$R_{D}^{(e,b)} = 1 - \frac{\bar{D}_{SpikeARC}^{(e,b)}}{\bar{D}_{Shared}^{(e,b)}}. \tag{13}$$

Here, e denotes the environment, b denotes the bth bootstrap resample, and the overbar denotes the mean over the resampled seed pairs. The three effect measures are

$$R_{AUC}: \text{ adaptation retention}, \tag{14}$$

$$R_{F}: \text{ forgetting reduction}, \tag{15}$$

$$R_{D}: \text{ Stable-drift reduction}. \tag{16}$$

An equal-weight macro average is then computed across the five environments. This procedure is repeated 20,000 times (RNG seed = 20260916), and the 2.5th and 97.5th percentiles of the empirical distribution define the 95% confidence interval. This interval characterizes only the macro-effect uncertainty induced by training-seed randomness on the five fixed benchmarks. It does not extrapolate to a distribution of untested environments.

The 95% confidence interval for forgetting reduction includes zero, reflecting the greater sensitivity of behavioral-retention metrics to differences in task difficulty and the magnitude of dynamics changes across environments. Accordingly, the 36.16% macro-averaged effect across the five environments describes an overall trend and should not be interpreted as a uniform reduction in forgetting in every environment. In contrast, Stable-drift reduction directly corresponds to the behavioral-drift

Table S6: Complete matched seed-level metrics. Each row compares policies initialized from the same mature checkpoint. AUC, forgetting, and drift follow the definitions in the main paper.

Environment	Seed	AUC SpikeARC	AUC Shared
HalfCheetah	0	3414.82	3707.85
HalfCheetah	1	3277.53	3627.14
HalfCheetah	2	1167.91	1158.85
Walker2d	0	407.12	384.26
Walker2d	1	568.94	664.83
Walker2d	2	765.76	1075.04
Hopper	0	467.33	474.47
Hopper	1	276.80	545.37
Hopper	2	823.62	678.75
CARL Walker	0	227.80	243.26
CARL Walker	1	189.82	196.68
CARL Walker	2	299.47	315.63
CARL Quadruped	0	391.92	435.50
CARL Quadruped	1	246.40	225.05
CARL Quadruped	2	246.00	295.83

Environment	Seed	Forgetting SpikeARC	Forgetting Shared	Drift SpikeARC	Drift Shared
HalfCheetah	0	42.28	1370.85	0.2144	0.3108
HalfCheetah	1	743.14	1091.82	0.2192	0.3224
HalfCheetah	2	42.94	103.53	0.2363	0.3195
Walker2d	0	39.01	34.05	0.2103	0.3736
Walker2d	1	81.54	200.66	0.2202	0.3669
Walker2d	2	297.84	394.14	0.2139	0.3520
Hopper	0	95.40	102.91	0.1779	0.2691
Hopper	1	22.50	264.64	0.1980	0.2557
Hopper	2	65.54	203.66	0.1589	0.2714
CARL Walker	0	8.10	4.21	0.1888	0.3315
CARL Walker	1	2.92	31.36	0.1991	0.3380
CARL Walker	2	11.39	2.37	0.1904	0.3284
CARL Quadruped	0	76.21	98.30	0.2517	0.3765
CARL Quadruped	1	165.35	128.42	0.2437	0.3417
CARL Quadruped	2	110.13	45.41	0.2585	0.3703

Table S7: HalfCheetah nominal return and adaptation AUC by seed. The matched nominal return is identical for both methods within each seed; all six runs complete the four-stage sequence.

Seed	Matched nominal return	SpikeARC AUC	Fully Shared AUC
0	3477.01	3414.82	3707.85
1	2691.81	3277.53	3627.14
2	1311.62	1167.91	1158.85

Table S8: Mechanism variants and the question isolated by each intervention. Metrics are normalized to Fully Shared within each environment and then averaged with equal weight over HalfCheetah, Walker2d, and Hopper.

Variant	Modification	Question isolated
SpikeARC (Full)	Complete Stable-Fast architecture, routing, selective consolidation, and behavior budgets	Full method
Fully Shared	All new changes are written directly into a shared long-term policy	Naive monolithic adaptation
Shared + Replay/EWC	Shared policy with replay and EWC	Is conventional anti-forgetting regularization sufficient?
Fixed Fusion	Two branches are retained, but the correction mixing coefficient is fixed	Is state-dependent fusion necessary?
No Routing	All other mechanisms are retained, but responsibility routing is removed	Does sample-level learning responsibility matter?
No Consolidation	Fast can adapt, but selective write-back is disabled	Is long-term admission necessary?
Uniform Consolidation	Write-back is enabled but does not depend on maturity	Is selective consolidation necessary?

constraint in SpikeARC and has the same direction in all five environments and all 15 matched seed pairs, indicating more stable control over long-term policy reconfiguration.

B.5 HalfCheetah Seed Variability

HalfCheetah has the largest AUC dispersion in Table S3. The values 2620.1 ± 1259.5 for SpikeARC and 2831.3 ± 1448.9 for Fully Shared are the exact sample means $\pm$ sample standard deviations over random seeds 0, 1, and 2. All six runs complete the full four-stage sequence ($4 \times 300,000$ adaptation steps), and their evaluation logs are complete. The mature nominal policy corresponding to the low-AUC seed 2 is also substantially weaker than those for seeds 0 and 1. Within each seed, SpikeARC and Fully Shared still share exactly the same mature base. The large standard deviation therefore reflects heterogeneity in the seed and base policy, rather than incomplete training or mismatched initialization.

C Mechanism Controls and Capacity Analysis

C.1 Replay Regularization and Component Ablations

This section tests two direct alternative explanations: whether replay/EWC added to a shared policy is already sufficient, and whether a Stable-Fast two-branch architecture without structured routing or consolidation can produce the same adaptation-protection profile. On HalfCheetah, Walker2d, and Hopper, we compare SpikeARC (Full), Shared + Replay/EWC, Fixed Fusion, No Routing, No Consolidation, Uniform Consolidation, and Fully Shared. All results are first normalized to Fully Shared within each environment and then averaged with equal weight across the three environments.

The upper-left panel of Figure 5 in the main paper converts all three metrics to higher-is-better normalized benefits. Shared + Replay/EWC has slightly higher adaptation retention than SpikeARC

(90.6% vs. 86.6%), but its forgetting protection is only 36.8%, compared with 71.3% for SpikeARC. Stable protection also decreases from 34.6% to 22.8%. Conventional replay/EWC can therefore move the shared policy to a different operating point on the stability-plasticity tradeoff, but it does not reproduce the joint adaptation, behavioral-retention, and long-term-rewriting profile of SpikeARC.

The upper-right panel of Figure 5 reports the percentage-point change of each structural ablation relative to Full. Fixed Fusion primarily reduces forgetting protection (-30.6 percentage points), showing that retaining two branches with a fixed mixing coefficient is insufficient to reproduce state-dependent correction. Removing routing reduces adaptation and forgetting protection by 3.6 and 6.3 percentage points, respectively, while leaving Stable protection nearly unchanged (+0.7 percentage points). Routing therefore changes the allocation of learning responsibility between corrections and replay experience rather than merely reducing Stable updates. No Consolidation reduces Stable protection by 23.0 percentage points and also impairs adaptation and retention, supporting the role of selective consolidation in long-term write-back.

Uniform Consolidation provides an additional control. It has 4.4 percentage points more Stable protection than Full, but 8.4 percentage points less forgetting protection. The goal of SpikeARC is therefore not to minimize Stable changes. It is to control which corrections qualify for long-term write-back and when. Selectivity in long-term write-back is more important than simply suppressing parameter movement.

The lower row of Figure 5 provides the corresponding dynamic evidence. Responsibility and consolidation maturity evolve across the continual sequence, showing that a correction's behavioral contribution and long-term maturity are not fixed labels. The base RL gradient scale for Stable remains nonzero throughout and tightens only as behavioral-budget utilization increases. Stable and composed-policy drift remain within their budget boundaries for most of training. SpikeARC therefore permits continual Stable co-adaptation while responsibility, maturity, and behavioral-drift constraints jointly control long-term write-back.

The normalized three-environment summary in this section and the five-environment macro study in Appendix B use different aggregation scopes. The 86.6% adaptation retention and 71.3% forgetting protection reported here therefore do not replace the five-environment primary results of 90.8% and 36.2%.

C.2 Plain Shared Capacity and Independent-Base Sensitivity

To test the explanation that SpikeARC benefits only from having more parameters, we remove Fast and Fusion, routing, consolidation, behavior budgets, and replay/EWC. We retain only a two-layer fully shared multilayer perceptron with widths 256, 288, 384, and 512. All plain shared models use the same history summary, PPO and dynamics protocol, and $1\text{M} + 4 \times 300\text{K}$ interaction budget as in the primary study, but each width trains its own nominal base. This is therefore a same-protocol capacity-scaling experiment rather than an exact-checkpoint ablation. The actor parameter count at width 288 is closest to that of SpikeARC, whereas widths 384 and 512 have greater model capacity.

On Hopper, increasing the capacity of the plain shared policy strengthens adaptation but does not automatically improve long-term retention. The adaptation AUC of every width is at least 1.02 times that of SpikeARC, and the width-288 model reaches 1.42 times its AUC. At the same time, forgetting is 2.62-4.27 times that of SpikeARC, and Stable drift is 1.57-1.97 times higher. Even though the width-512 model has approximately 3.14 times as many actor parameters, its long-term policy drift remains almost twice that of SpikeARC.

CARL Quadruped provides a more discriminative boundary case. The approximately parameter-matched width-288 model achieves an adaptation AUC of 366.4 ± 28.2 and forgetting of 64.5 ± 65.9, both better than 294.8 ± 84.1 and 117.2 ± 45.0 for SpikeARC. Its Stable drift is 0.376 ± 0.003, however, approximately 50% higher than the 0.251 ± 0.007 of SpikeARC. A plain monolithic policy can therefore obtain a good sequential solution through larger global reconfiguration, but good behavioral retention and limited long-term policy rewriting are distinct properties.

Section 4.3 of the main paper separates raw representational capacity from the organization of adaptation by using two-layer plain shared MLPs with widths 256, 288, 384, and 512. Each width independently completes 1 million steps of nominal base training. To test whether the observed

Table S9: Additional quantities for the capacity controls. Absolute values are mean ± SD, and ratios use SpikeARC as the reference. Figure S1 in Appendix C.2 shows the full width-dependent comparisons.

Environment and setting	Adaptation AUC	Forgetting	Stable drift or parameter count
Hopper, widths 256/288/384/512	All $\geq 1.02\times$ SpikeARC; width 288 is $1.42\times$	2.62-$4.27\times$ SpikeARC	Drift is 1.57-$1.97\times$; the width-512 actor has approximately $3.14\times$ as many parameters.
CARL Quadruped, SpikeARC	294.8 ± 84.1	117.2 ± 45.0	0.251 ± 0.007
CARL Quadruped, width 288	366.4 ± 28.2	64.5 ± 65.9	0.376 ± 0.003; approximately matched actor parameter count

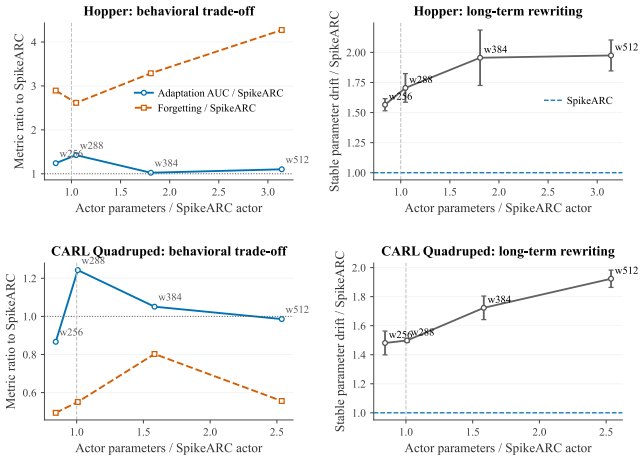

Figure S1: Shared-policy capacity controls on Hopper and CARL Quadruped. Left: Adaptation AUC and forgetting relative to SpikeARC across model scales. Right: Stable-parameter-drift ratios, with dashed lines indicating the SpikeARC reference.

Table S10: CARL Quadruped capacity-control ratios for the original width sweep and the sweep using independently trained nominal bases.

Width	AUC/SpikeARC Run 1	AUC/SpikeARC New bases	Forgetting/ SpikeARC Run 1	Forgetting/ SpikeARC New bases
w256	0.87	1.20	0.49	0.57
w288	1.24	1.06	0.55	0.99
w384	1.05	1.02	0.80	0.77
w512	0.99	0.61	0.56	0.37

Width	Drift/SpikeARC Run 1	Drift/SpikeARC New bases
w256	1.48	1.44
w288	1.50	1.49
w384	1.72	1.71
w512	1.92	1.91

capacity pattern depends on one particular mature initialization, we independently repeat the CARL Quadruped width sweep from newly trained nominal bases while keeping the PPO protocol, dynamics sequence, network widths, and interaction budget unchanged.

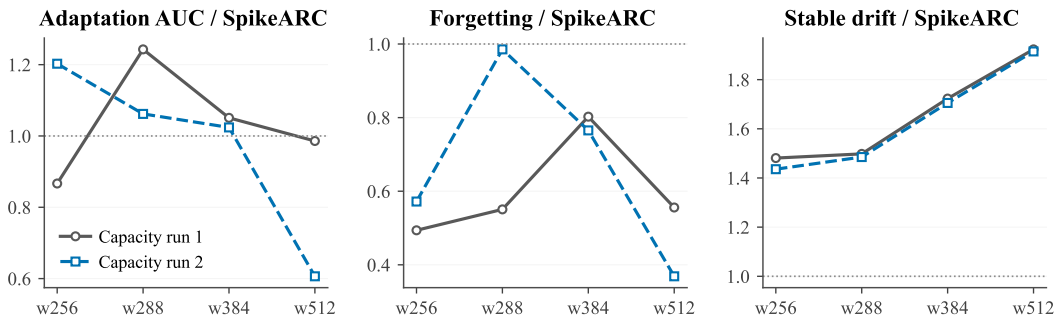

Figure S2: Independent-base capacity sensitivity on CARL Quadruped. The AUC and forgetting order changes with newly trained nominal bases, whereas the larger long-term reconfiguration of all plain shared widths is reproduced.

Table S11: Recurrent-core control on CARL Quadruped (three seeds). SNN and GRU use the same continual-adaptation protocol but do not share identical mature checkpoints.

Temporal core	**Adaptation AUC ↑**	**Forgetting ↓**	**Fast/Fusion parameters**	**Actor parameters**
Recurrent SNN	294.8 ± 84.1	117.2 ± 45.0	24,077	153,593
GRU	308.7 ± 60.3	88.0 ± 13.8	69,261	198,777

Behavioral outcomes are relatively sensitive to mature initialization: the ordering of AUC and forgetting changes between the two sweeps, whereas the long-term-rewriting pattern is more stable. For widths 256, 288, 384, and 512, the drift ratios change only from 1.48/1.50/1.72/1.92 to 1.44/1.49/1.71/1.91. This distinction separates the final behavioral solution from the adaptation process used to reach it. A monolithic shared policy can obtain different sequential-task solutions from different nominal bases, but both sweeps depend on substantially greater global reconfiguration. SpikeARC provides an independent control dimension: it can express an immediate correction first and then determine whether that correction should enter the long-term policy.

The capacity controls therefore distinguish how much a network can represent from how new behavior enters the long-term policy. Additional capacity can allow a shared policy to find a different, and sometimes better, behavioral solution, but it does not automatically provide control over long-term write-back independently of immediate adaptation. SpikeARC adds this control dimension. Fast and Fusion first express the current correction, after which behavioral responsibility, consistency over time, and behavioral risk determine whether and when it is written into Stable.

C.3 Recurrent Temporal-Core Control

The core SpikeARC mechanisms are defined in terms of action correction, replay responsibility, selective consolidation, and behavior-space protection, rather than a particular recurrent cell. To test this distinction, we replace the recurrent-LIF temporal substrate of Fast and Fusion with a gated recurrent unit (GRU) on CARL Quadruped, while leaving the continual-adaptation framework and training protocol unchanged. This control starts from independently trained nominal bases and is therefore not a single-factor performance ranking from exactly the same checkpoint. It addresses a narrower question: whether the responsibility and consolidation mechanisms depend on spiking recurrence.

The GRU obtains a higher mean AUC and lower forgetting in these runs. The specific role of the recurrent SNN is to provide a more compact temporal correction substrate: the Fast/Fusion parameter count is approximately 34.8% of that of the GRU, and the complete actor has approximately 77.3% as many parameters.

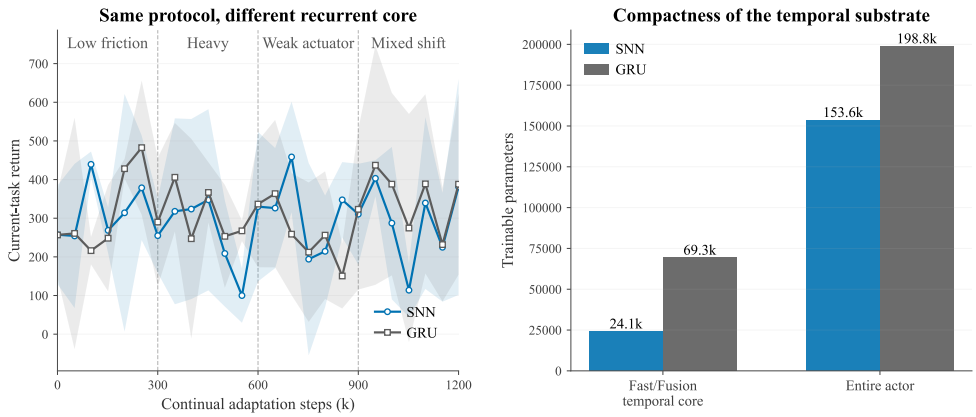

Figure S3: Adaptation framework with different recurrent temporal cores. A: current-task return over four dynamics stages. B: trainable parameter counts for the Fast/Fusion temporal subsystem and the complete actor.

Table S12: LCPO privileged-context reference (mean ± SD over three seeds). LCPO observes dynamics information unavailable to the hidden-context methods in Section 4.2 of the main paper; the values therefore do not constitute a direct ranking against Table S2.

Environment	**LCPO threshold**	**Mean stage-end current return**	**Final average return**
Walker	0.04	210.9 ± 9.7	212.7 ± 5.8
Walker	0.06	207.1 ± 2.2	223.7 ± 5.6
Quadruped	0.04	109.7 ± 32.6	98.4 ± 39.2
Quadruped	0.06	122.7 ± 53.2	101.1 ± 29.2

C.4 Privileged-Context LCPO Reference

The external comparison in the main paper uses a hidden-context protocol, whereas the formal LCPO implementation directly uses observed dynamics context to construct local policy constraints and therefore operates under a stronger information condition. To avoid treating results obtained under different information conditions as a single ranking, we report LCPO only as a privileged-context reference and do not rank it directly against the hidden-context methods in the main paper. The LCPO evaluation includes three-seed evaluation runs at thresholds 0.04 and 0.06.

D Quadruped Routing Diagnostics

D.1 Controlled Intervention and Final-Checkpoint Evaluation

The preceding controlled experiments primarily use parameter and optimization statistics to show how SpikeARC separates immediate correction from long-term policy write-back. This section further tests whether responsibility routing changes observable embodied behavior. We conduct a controlled routing intervention in a structured rough-transition locomotion setting. The robot implementation retains the Stable-Fast-Fusion organization and the functional division from correction to consolidation, while using domain-specific gait, central pattern generator (CPG), and residual-control interfaces. Compared with the general continuous-control implementation in Sections 4.2-4.3 of the main paper, this experiment uses a domain-specific control head together with routed replay and selective consolidation. It tests the behavioral implications of the same functional principles in embodied control rather than reproducing the general implementation line by line.

The only controlled difference between Full SpikeARC and No Routing is replay responsibility routing: the former uses routed replay, whereas the latter uses naive replay. Fast, learned fusion, replay, EWC, selective consolidation, and scheduled Stable plasticity remain unchanged. The comparison

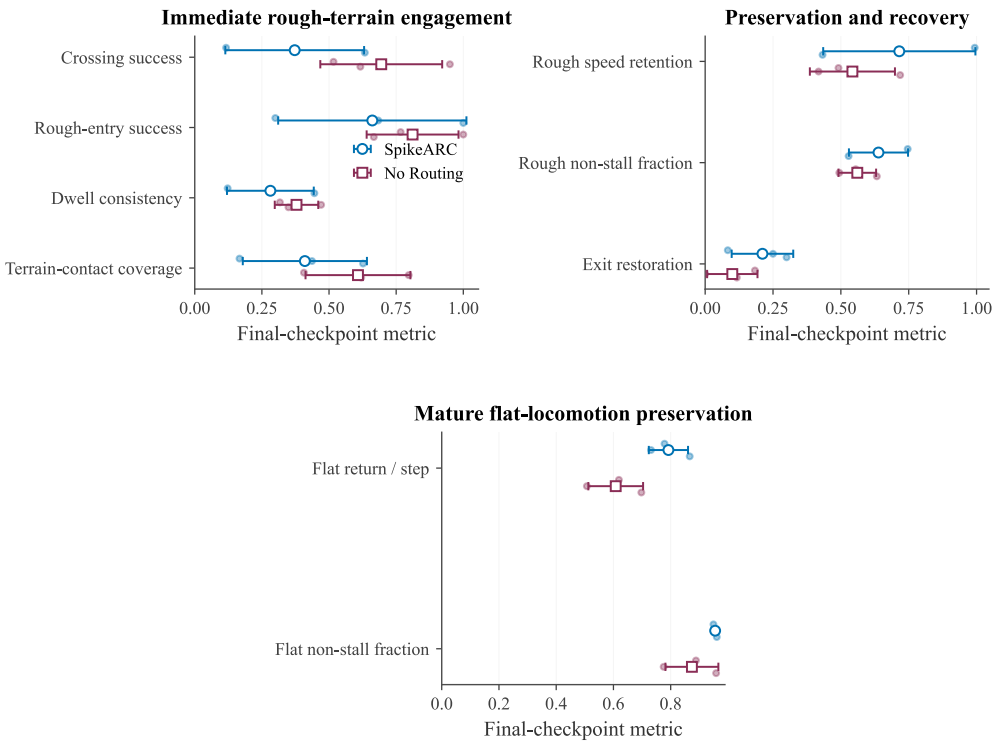

Figure S4: Formal final-checkpoint routing decomposition. Top: immediate rough-terrain engagement (left) and preservation and post-transition recovery (right). Bottom: mature flat-locomotion preservation. Small points are individual checkpoints; large markers and error bars show mean ± SD over three checkpoints.

Table S13: Additional final-checkpoint quadruped metrics (mean ± SD over three trained checkpoints). Each checkpoint is evaluated for 120 episodes, including 60 rough-transition episodes.

Metric	**SpikeARC**	**No Routing**
Rough-crossing success ↑	0.372 ± 0.258	0.694 ± 0.227
Rough-entry success ↑	0.661 ± 0.351	0.811 ± 0.171
Rough-dwell consistency ↑	0.282 ± 0.162	0.379 ± 0.081
Rough-speed ratio ↑	0.715 ± 0.280	0.543 ± 0.157
Rough-terrain stall fraction ↓	0.361 ± 0.109	0.439 ± 0.069
Exit-restoration success ↑	0.211 ± 0.113	0.100 ± 0.093
Terrain-contact coverage ↑	0.410 ± 0.231	0.608 ± 0.195
Flat-speed return per step ↑	0.792 ± 0.069	0.608 ± 0.096
Flat-speed stall fraction ↓	0.045 ± 0.006	0.126 ± 0.092

therefore directly isolates the effect of routing on the allocation of learning responsibility. Training uses five independent seeds. The formal final-checkpoint evaluation uses three independently trained checkpoints, with 120 episodes per checkpoint, including 60 rough-transition episodes. Training trajectories show the dynamic process; formal final comparisons use checkpoint-level results.

The upper row of Figure 6 in the main paper shows that routing changes where plasticity is allocated. As training progresses, No Routing achieves higher rough-crossing success, and its rough-modulation magnitude also becomes higher than that of SpikeARC. Together, these results indicate that removing responsibility routing causes the system to allocate more correction capacity to the current rough-terrain disturbance. Full SpikeARC uses rough-terrain corrections more selectively, so the pressure for immediate adaptation need not be converted entirely into continual rewriting of mature locomotion behavior.

For immediate rough-terrain engagement, the rough-entry success, rough-crossing success, rough-dwell consistency, and terrain-contact coverage of No Routing are 0.811 ± 0.171, 0.694 ± 0.227, 0.379 ± 0.081, and 0.608 ± 0.195, respectively. The corresponding results for SpikeARC are 0.661 ± 0.351, 0.372 ± 0.258, 0.282 ± 0.162, and 0.410 ± 0.231. These consistently reported behavioral metrics show that, after routing is removed, the policy responds more aggressively to the current rough-terrain disturbance.

Stronger immediate engagement also changes retention and recovery. On rough terrain, SpikeARC retains a higher speed ratio (0.715 ± 0.280 vs. 0.543 ± 0.157) and a lower stall fraction (0.361 ± 0.109 vs. 0.439 ± 0.069). Its exit-restoration success after leaving the disturbance is also higher (0.211 ± 0.113 vs. 0.100 ± 0.093). In the flat-speed evaluation, SpikeARC obtains a return per step of 0.792 ± 0.069, compared with 0.608 ± 0.096 for No Routing, and a lower stall fraction (0.045 ± 0.006 vs. 0.126 ± 0.092). Routing therefore reallocates adaptive plasticity between responding to the current disturbance and preserving speed, smoothness, and recovery in mature locomotion; it does not uniformly improve every metric.

This decomposition reflects a systematic redistribution of adaptive plasticity rather than a uniform gain across metrics. No Routing shows stronger immediate rough-terrain engagement and relies more heavily on Fast corrections to address the current disturbance. SpikeARC retains a higher speed on rough terrain, a lower stall rate, stronger post-transition restoration, and better flat-locomotion behavior. These results are descriptive mechanistic evidence at the checkpoint level. Individual evaluation episodes are used only to estimate the performance of trained policies and are not treated as independent statistical samples.

With all other adaptation mechanisms held fixed, Full and No Routing differ only in replay responsibility routing. The systematic behavioral shift from stronger immediate rough-terrain engagement to better speed preservation and post-disturbance recovery therefore provides controlled-intervention evidence of how responsibility routing reallocates adaptive plasticity. In this implementation, routing does not simply make the policy more conservative. It changes which experiences and corrections should carry responsibility for current learning, thereby preventing the full pressure for rapid adaptation from being converted directly into continual rewriting of mature behavior.

D.2 Supplementary Training-Log Variants

The robot training logs also include Shared + Replay/EWC, Fixed Fusion, No Consolidation, and Uniform Consolidation. Full and No Routing each use five training seeds; the other variants use three seeds each. Because only Full and No Routing have a formal final-checkpoint evaluation, the other variants are used only for training-log diagnostics.

The robot logs for these variants support the same interpretation as the formal routing intervention: different mechanisms correspond to different operating points between immediate rough-terrain adaptation and the preservation of mature locomotion. We use these runs to diagnose the role of each component. The controlled-intervention conclusions about responsibility routing are based on the final-checkpoint evaluation of Full and No Routing.

E Extended Motivation and Related Work

E.1 From Immediate Correction to Long-Term Behavior

The most direct approach to continual adaptation is to keep updating one shared policy. This approach provides strong immediate plasticity, but gradients from a new environment directly alter the existing policy and may write both transient changes and persistent regularities into its parameters. An alternative is to protect the mature policy and delegate changes to an independent fast module. That approach can reduce interference with old behavior, but it may leave repeatedly validated regularities indefinitely in a temporary compensation layer. When a behavioral correction first appears, the agent usually cannot know whether it reflects a transient disturbance or should become part of the long-term policy. When the true dynamics parameters and task labels are hidden, this decision cannot rely on a manually supplied environment identity.

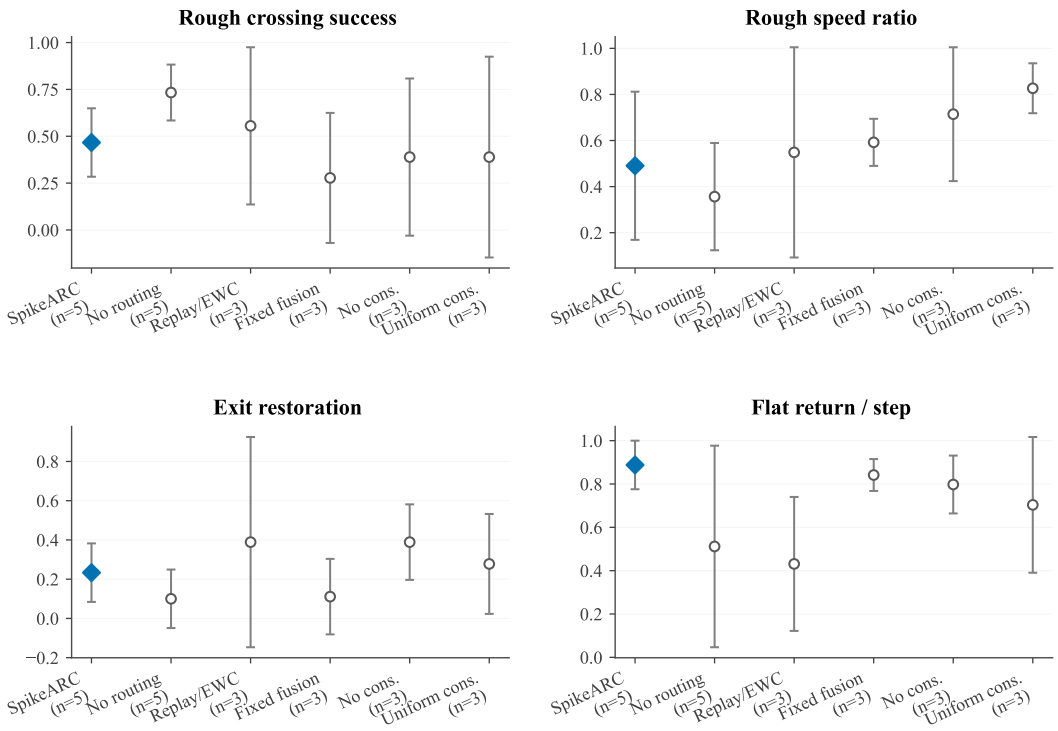

Figure S5: Supplementary robot training-log diagnostics. Each panel summarizes the final logged evaluation point for SpikeARC and No Routing ($n = 5$ training seeds each) and for Shared + Replay/EWC, Fixed Fusion, No Consolidation, and Uniform Consolidation ($n = 3$ each). Error bars show seed-level SD. These results are training-log diagnostics rather than an additional final-checkpoint benchmark.

SpikeARC does not require the agent to make an immediate binary decision between short-term and long-term change. Instead, it decomposes the entry of new behavior into the long-term policy into related stages with distinct responsibilities. A new action change is first expressed rapidly as a local correction. Optimization responsibility is then assigned according to the correction's actual contribution to current behavior, preventing Stable from prematurely absorbing Fast-dominated experience. Only corrections that remain consistent during subsequent interaction become eligible for long-term write-back. The permissible magnitude of Stable change is determined by the observed risk of drift in existing behavior rather than by a fixed freezing ratio or slow learning rate. Whether a temporary correction ultimately becomes a long-term capability is therefore a gradual process governed jointly by behavioral evidence and risk.

Based on this principle, SpikeARC provides a cooperative Stable-Fast framework for nonstationary continuous control. The Stable pathway produces the mature primary action. The Fast pathway uses recurrent spiking dynamics to generate a state-dependent action correction from the recent state sequence. The Fusion pathway combines the state history with Fast spike activity to produce a gating signal that controls the contribution of the correction in the current state. The final policy does not switch between two complete policies; it adds a gated Fast correction to the Stable action. An environmental change can thus be expressed first as a local and reversible behavioral correction without immediately rewriting the long-term policy on a large scale. Stable is not frozen, but co-adapts with Fast to the extent allowed by the current risk. SpikeARC is therefore Fast-leading and Stable-co-adapting rather than a strictly staged handoff.

The Stable-Fast architecture alone does not solve forgetting. If replay or distillation errors continue to propagate indiscriminately into Stable, changes temporarily carried by Fast may still be written rapidly into the long-term policy. SpikeARC therefore introduces responsibility routing based on the action correction actually applied. The system estimates responsibility for each historical sample

from Fast's actual contribution to the final action. When the Fast correction is small, Stable still accounts for most of the behavior and receives substantial replay credit. When Fast accounts for most of the action change, the corresponding replay update to Stable is reduced. Routing leaves the forward action unchanged and alters only backward optimization responsibility, explicitly separating participation in current behavior from responsibility for long-term parameter updates.

Protecting Stable is not the endpoint. If a Fast correction retains a similar direction and magnitude during subsequent interaction and contributes to behavior through a strong gate, it is more likely to represent a stable change that should be absorbed over the long term. Consistency-guided consolidation compares historical and current corrections and combines their consistency with actual activation and usage to assign different long-term write-back weights. Low-consistency changes remain in the Fast pathway. High-consistency changes gradually move the Stable behavior output toward the validated composed policy. Consolidation occurs in behavior space rather than by copying Fast parameters. Fast consequently serves both as the rapid-adaptation pathway and as a buffer through which new behavior must pass before entering the long-term policy.

Responsibility routing determines which pathway should carry a current change, and consolidation determines which changes merit long-term retention. The long-term policy must also determine how much change is currently permissible. SpikeARC therefore imposes two behavior-space budgets. One constrains drift in the Stable pathway itself under nominal conditions; the other constrains drift of the current full policy relative to the composed policy from historical stages. Historical behavior protection focuses on the stage with the greatest drift rather than averaging across stages. Adaptive primal-dual control uses the two drift measures to regulate retention pressure and the effective plasticity of Stable. When behavioral risk is low, Stable can release more update capacity and adapt together with Fast. As any protected behavior approaches a budget boundary, the additional plasticity is reduced automatically. Stable is thus a long-term policy whose learning magnitude is regulated by current behavioral risk, not a fixed slow branch.

Together, these mechanisms form a continuous path of behavioral evolution from immediate correction to long-term capability. New changes are first expressed by the highly plastic Fast pathway, while Stable co-adapts within the permitted risk. Responsibility routing allocates replay responsibility according to the correction's actual contribution. Consistency-guided consolidation determines which corrections are sufficiently stable to enter the long-term policy. The dual behavioral-drift budgets limit the effect of write-back on mature behavior. This process requires neither an advance judgment that a change is temporary or permanent nor a manually specified time for a Fast-to-Stable handoff.

E.2 Extended Related Work

Stability-plasticity and experience replay in continual learning. Continual learning requires acquiring new knowledge from a nonstationary data stream while limiting catastrophic forgetting of existing capabilities (Ebrahimi et al., 2020; Kemker et al., 2018; Riemer et al., 2019). Surveys and systematic measurement studies identify the stability-plasticity dilemma as a fundamental problem in this field (Kemker et al., 2018; Parisi et al., 2019). The effectiveness of experience replay depends not only on whether replay is used, but also on the storage budget, sample retrieval, correlation structure, distillation targets, and replay schedule (Buzzega et al., 2021; Chaudhry et al., 2021; Hayes et al., 2019). Memory-efficient rehearsal, maximally interfered retrieval, Dark Experience Replay and X-DER, correlation-aware replay, adaptive replay, sparse or full replay, and hindsight anchoring reduce interference along these respective dimensions (Aljundi et al., 2019; Boschini et al., 2023; Buzzega et al., 2021; 2020; Chaudhry et al., 2021; Gong et al., 2022; Hayes et al., 2019; Li et al., 2024; Q. Wang et al., 2022; Yu et al., 2024). Generative or local memory, representation decomposition, and rehearsal-free prompting instead mitigate forgetting through the form of memory or the organization of parameter sharing (de Masson d'Autume et al., 2019; Ebrahimi et al., 2020; Z. Wang et al., 2022). Related problems arise in image quality assessment, trajectory prediction, dynamic graphs, graph learning, recommendation, human activity recognition, spectral clustering, and combinatorial optimization (Adaimi and Thomaz, 2022; Ahrabian et al., 2021; Korycki and Krawczyk, 2021; Pei et al., 2026; Rakaraddi et al., 2022; Sun et al., 2022; Z. Wang et al., 2025; Wu et al., 2023; Yang et al., 2022; Zhang et al., 2021; Zhou and Cao, 2021). SpikeARC addresses a narrower question: when the forward policy output is held fixed, which functional pathway should receive backward-update responsibility for replay error?

Dual memory, fast-slow systems, and consolidation. Inspired by complementary learning systems, existing methods separate rapid acquisition from long-term retention through short- and long-term memories, fast and slow weights, or modular divisions of labor (Arani et al., 2022; Rostami et al., 2019). Examples include generative dual memory, FearNet, IL2M, CLS-ER, sparse dual memory, bilateral memory consolidation, cycle memory, and dual-memory recurrent self-organization (Arani et al., 2022; Belouadah and Popescu, 2019; Kamra et al., 2018; Kemker and Kanan, 2018; Nie et al., 2023; Parisi et al., 2018; Peng et al., 2024; Sarfraz et al., 2023; Sun et al., 2022). Attentive Independent Mechanisms combine fast and slow weights with relatively independent mechanistic modules, and behavioral consolidation has also been used for lifelong routing (Lee et al., 2021; Pei et al., 2026). Fast-slow or dual-system architectures have further been applied to high-speed robotic table tennis, vision-language navigation, and multi-turn image-editing agents, where the fast and slow systems handle perceptual correction, planning and execution, or different levels of search (Gupta et al., 2026; Wang et al., 2026; Wei et al., 2026). The term "fast" can also denote computational efficiency alone, as in acceleration designs for plain vision transformers (Fuller et al., 2026). We therefore do not claim novelty for the mere use of two timescales. SpikeARC differs in that Stable and Fast jointly form the continuous-control action, while forward behavioral authority, backward responsibility for replay, and consolidation write-back authority remain separate. Stable retains constrained on-policy plasticity rather than being frozen permanently.

Continual reinforcement learning, negative transfer, and control. Nonstationary state-action distributions in sequential tasks make forgetting and negative transfer particularly pronounced in reinforcement learning. Prior work uses pseudo-rehearsal, generative long-term replay, or dual memory to maintain performance on old tasks (Atkinson et al., 2021; Kamra et al., 2018; Li et al., 2020). A systematic ICLR 2025 analysis further shows that negative transfer is widespread in continual reinforcement learning and cannot be eliminated merely by mitigating loss of plasticity or improving positive transfer (Ahn et al., 2025). Control-related studies include unified experience replay for spiking deep reinforcement learning, multi-AUV control combining adaptive recurrent neurons with bio-inspired replay, and recurrent-SNN learning that is fast while its parameters change slowly (Cai et al., 2025; Capone and Muratore, 2024; Xu et al., 2026). These methods primarily alter replay content, memory generation, network dynamics, or update timescales. SpikeARC instead focuses on sequential dynamics adaptation, where the task objective remains fixed while the dynamics conditions change, and directly studies whether current behavioral errors should be written into Stable, Fast, or the subsequent consolidation pathway.

Spiking and neuromorphic learning. Spiking neural networks and neuromorphic learning provide implementation substrates for temporal state, sparse event-driven computation, and biologically inspired stabilization mechanisms. Previous studies have investigated homeostatic neurons, embedded online-learning frameworks, sleep-like replay and weight reorganization, reward-broadcasting plasticity, dynamically gated neurons, and efficient event-driven SNN architectures (Bai et al., 2026; Detorakis et al., 2018; Golden et al., 2022; Lu et al., 2026; Muñoz-Martin et al., 2021; M. Wang et al., 2025). A separate review examines connections between artificial and biological replay mechanisms (Hayes et al., 2021). We use recurrent LIF states as the concrete temporal substrate for Fast correction and Fusion context, but do not claim that an SNN neuron type is generally superior to GRU or LSTM units, or that the word "spike" alone establishes relevance. For example, SPIKE-RL studies Bayesian surprise and frame sampling in video large language models, not spiking neural control (Ravi et al., 2026). The central contribution of this work is the organization of adaptation responsibility, not a neuron model in isolation.

E.3 Scope of the Adaptation and Retention Claims

SpikeARC reframes continual policy adaptation as the problem of how new behavioral corrections enter a mature long-term policy. Stable-Fast behavior decomposition, responsibility routing based on the applied correction, consistency-guided consolidation, and behavior-risk-constrained Stable plasticity separate immediate behavioral expression, replay learning responsibility, and long-term policy write-back. The external comparison across reinforcement-learning backbones, the five-environment matched-base study, mechanism ablations, capacity controls, and quadruped routing intervention show that this organization does not depend on freezing the long-term policy or merely increasing raw network capacity. SpikeARC retains most of the continual-adaptation capability while more consistently limiting mature-policy rewriting. In most environments, this controlled write-back also improves the retention of historical behavior. In embodied control, it produces an

interpretable allocation of plasticity between immediate disturbance adaptation and the preservation and recovery of mature behavior. Under some conditions, a plain monolithic policy can also obtain a good behavioral solution through large-scale global reconfiguration. SpikeARC therefore does not claim access to behavioral solutions that other policies cannot obtain. It provides an independent mechanism for controlling long-term write-back: first express a change, then use responsibility, consistency, and behavioral risk to determine which changes should become long-term capabilities.