SpikeWorld: Adapting Frozen World Models with Compact State

Anonymous authors
Paper under double-blind review

ABSTRACT

A world model can predict the future yet remain unable to absorb the errors revealed when that future arrives. We ask how much mutable state is needed to adapt its predictions without rewriting its learned representation. We introduce **SpikeWorld**, which pairs a next-event predictive–semantic model with a bounded external fast state. We separate the shared computation learned offline from the residual corrections written online: completed actions supervise later predictions, while a distinct score state calibrates registered action transformations. On a continual stream of registered actuator changes, 29 KB of mutable state obtains a 19.3% prequential prediction gain, versus 21.9% for prediction-path tuning with 67$\times$ more mutable state. Controlled formation tests establish temporal predictive structure, and a same-stream routed matrix RLS comparison provides a compact actuator-identification reference. SpikeWorld makes deployment plasticity explicit: retain the predictive representation, and allocate bounded state to what new observations teach.

1 INTRODUCTION

A world model receives a new training example every time an action finishes. It predicts a consequence, the environment produces one, and their difference becomes available without a label, demonstration, or reward. Recent models organize visual and spatial experience into increasingly capable predictive representations (World Labs Team, 2026; Mur-Labadia et al., 2026; Maes et al., 2026). Deployment adds a different question: where should a model store what a newly observed prediction error teaches it?

Fine-tuning writes corrections into model weights and carries optimizer state. Adapters restrict the updated parameters; residual models place corrections beside a pretrained model (Moon & Kim, 2026; Lanier et al., 2026). We distinguish the representation learned before deployment from the plasticity retained during deployment, and ask: *how close can a frozen predictive–semantic foundation come to online weight adaptation under a small mutable-state budget?*

SpikeWorld answers this question with a shared predictive model and a bounded external state. Offline, causal modality adapters map audio, rendered text, retinal images, video events, and state–action streams into one sparse Spike Transformer. Sensory streams predict the next spike-bearing slice; action streams predict a later physical endpoint. Semantic and binding heads read the same present state. Controlled tests show that this predictive interface captures temporal organization beyond rate and order controls, predicts label-free future statistics better than random state, and supports bounded cross-modal composition.

Online, the foundation becomes immutable. Each completed action exposes a causal endpoint residual. SpikeWorld writes that residual to two route-specific matrices that refine later predictions. In parallel, 132 bytes of cumulative candidate scores select a bounded correction from a registered dynamics bank. These states are deliberately role-separated: matrices adapt prediction; scores calibrate actions. Corrected predictions do not enter the action selector.

The main experiment crosses three independently trained foundations with three fresh action seeds on a six-stage stream that leaves and returns to source dynamics. External state improves prequential prediction by 19.28%, compared with 21.85% for prediction-path tuning, using 28,848 rather than 1,932,488 mutable numeric bytes. It is non-dominated among these five configurations. Separate

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.

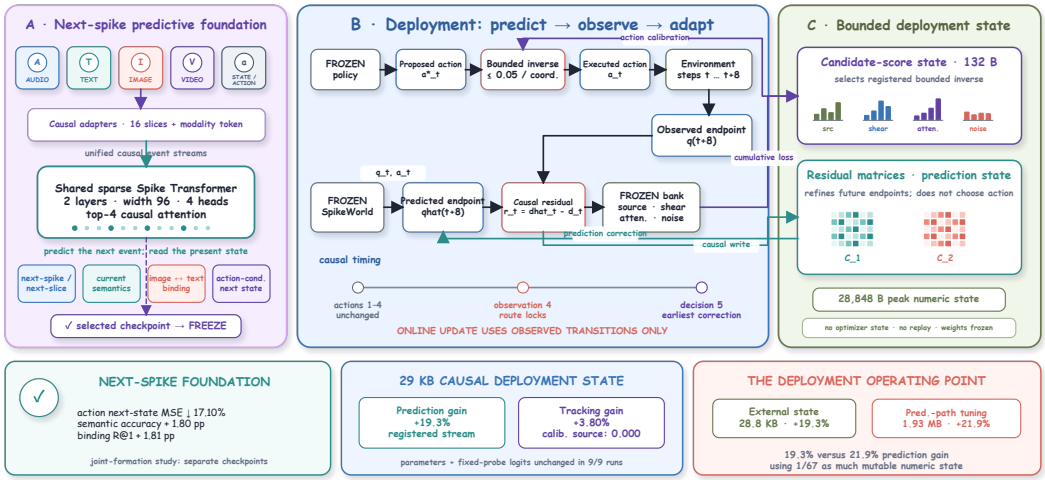


Figure 1: **SpikeWorld moves deployment plasticity from foundation weights to bounded causal state. A**, next-event prediction, semantics, binding, and action-conditioned dynamics share a causal foundation. The main deployment comparison freezes its sequential checkpoint. **B**, a frozen policy and world model predict, act, and wait for the endpoint before writing a residual; four completed transitions make decision five the earliest corrected action. **C**, residual matrices refine later predictions, while a separate 132-byte score state selects a registered inverse. The complete peak ledger is 28,848 numeric bytes. On the continual stream, this state obtains 19.3% prediction gain; prediction-path tuning obtains 21.9% with 67× more mutable numeric state.

studies compare routed matrix RLS and establish the behavioral effect of registered score-state calibration.

Our contributions are:

1. We implement a next-event predictive–semantic model with shared temporal computation and modality-specific forecast targets. Controlled formation and deployment studies separate learned representation from online plasticity.
2. We identify a compact deployment operating point. A 28,848-byte fast state obtains 19.28% prediction gain where prediction-path tuning obtains 21.85% with 66.99 times more mutable numeric state.
3. We trace prediction and behavioral gains to distinct state components. Matrix interventions isolate prediction gains, while registered score-state calibration improves closed-loop reward after completed observations. A routed matrix RLS comparison supplies a compact identification reference.

2 Related Work

Predictive representations and consequence feedback. Predictive-state representations define state through action-conditional future observations (Littman et al., 2001). Neural world models learn latent dynamics for policy learning and planning (Ha & Schmidhuber, 2018; Hafner et al., 2019; 2020), while feature-predictive video models forecast target-encoder states rather than pixels (Bardes et al., 2024). Recent work advances different predictive interfaces: Atlas conditions multimodal generation on shared spatial context (World Labs Team, 2026); V-JEPA 2.1 learns dense image–video representations (Mur-Labadia et al., 2026); and LeWorldModel learns next-embedding dynamics end-to-end from pixels (Maes et al., 2026). ActEffect uses predicted action consequences as training-time ranking feedback in a frozen DINOv3 representation space, then removes its consequence model at deployment (SunRisingAI Lab, 2026). These works motivate distinguishing what a model predicts from when, and where, its feedback changes the system. SpikeWorld studies observed endpoint feedback at deployment, with model and policy weights fixed.

Events, spikes, and predictive learning. Event sensors encode asynchronous changes (Lichtsteiner et al., 2008; Gallego et al., 2022), and surrogate gradients make multilayer spiking networks trainable (Neftci et al., 2019; Zenke & Ganguli, 2018). Precise timing can carry discriminative structure that disappears from spike counts, and surrogate-gradient networks can learn several such timing codes (Yu et al., 2026). Spiking attention models combine event-driven activations with Transformer computation (Zhou et al., 2023; 2024a); self-supervised event models use masking, contrastive objectives, or event–image–text alignment (Yang et al., 2023; Klenk et al., 2024; Zhou et al., 2024b). Autoregressive sensorimotor models also cast humanoid control as modality-aligned next-token prediction (Radosavovic et al., 2024). SpikeWorld uses a spiking causal core with modality-specific forecast targets: sensory prediction forms temporal state, and action-conditioned endpoint prediction exposes a deployment write signal.

Test-time adaptation and fast state. Test-time training, TENT, CoTTA, and EATA adapt weights or normalization state from unlabeled test data (Sun et al., 2020; Wang et al., 2021; 2022; Niu et al., 2022). Fast weights and differentiable plasticity place shorter-lived variables between activity and consolidated parameters (Ba et al., 2016; Miconi et al., 2018); TTT layers update a model-valued hidden state, while STAD tracks class prototypes with a state-space model (Sun et al., 2025; Schirmer et al., 2025). Context inference for control likewise estimates latent tasks or dynamics from recent experience (Rakelly et al., 2019; Nagabandi et al., 2019; Yu et al., 2017; Kumar et al., 2021). ReDRAW fits latent dynamics residuals around a frozen world model and reoptimizes a policy in corrected imagined rollouts (Lanier et al., 2026). SPREAD keeps the pretrained model frozen and updates LoRA adapters online (Moon & Kim, 2026). SpikeWorld measures a different operating point: role-separated external state under a complete mutable-numeric-state budget, including optimizer moments, replay, and routing history in the compared arms.

Online identification and residual control. Recursive least squares, multiple-model control, and fault-tolerant reconfiguration provide classical mechanisms for adapting known dynamical families (Åström & Wittenmark, 1973; Narendra & Balakrishnan, 1997; Zhang & Jiang, 2008). Residual control also keeps a nominal policy fixed and learns a bounded additive correction (Jayasinghe et al., 2026). SpikeWorld belongs to this lineage: its candidate bank represents registered actuator transformations and its inverse is deliberately bounded. The new object is their integration with a next-event predictive–semantic foundation and a role-separated state whose memory–prediction trade-off is measured against adapters, prediction-path tuning, replay, and routed matrix RLS.

3 Next-Event Predictive–Semantic Foundation

One causal core forecasts sensory slices and delayed action endpoints, while semantic heads read the present state (Figure 2).

The three separate controls in Figure 2 establish temporal structure, label-free future prediction, and cross-modal composition. They test the interface rather than the deployment checkpoints; the formation study in Section 6 separately evaluates shared-core optimization.

3.1 Causal event streams

Audio, rendered text, retinal images, video changes, and state–action prefixes are mapped to length-16 causal sequences with learned modality tokens. The term "multimodal" refers to shared temporal computation and parameters, not identical sensors. The common core has two width-96 Transformer layers, four heads, feed-forward width 192, and top-4 causal attention. Its leaky integrate-and-fire units obey

$$v_t = \beta u_{t-1} + I_t, \qquad s_t = \mathbf{1}[v_t > V_{\mathrm{th}}], \qquad u_t = v_t - V_{\mathrm{th}} s_t, \tag{1}$$

with membrane decay $\beta = 0.85$ and surrogate derivatives during offline training. Continuous membrane features carry predictive state; spikes gate inter-layer communication.

For a sensory stream $x^s_{1:T} \in \mathbb{R}^{T \times D_s}$, the next-slice error is normalized by the persistence error of that modality,

$$\ell_s = \frac{\|\hat{x}^s_{2:T} - x^s_{2:T}\|_2^2}{(T-1) D_s L^s_{\mathrm{persist}}}. \tag{2}$$

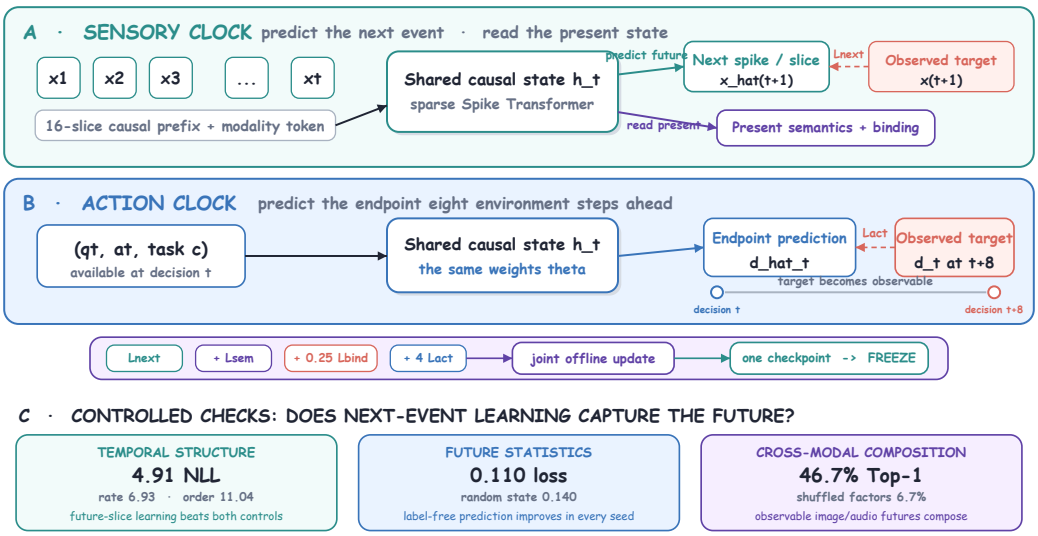


Figure 2: **One causal interface predicts what happens next. A**, a length-16 sensory prefix predicts the next spike-bearing slice, while semantic and binding heads read the current state. **B**, the state, proposed action, and task token predict the standardized endpoint change eight environment steps later. **C**, three separate controlled tests verify temporal structure beyond rate/order summaries, label-free future statistics beyond a random state, and image/audio composition beyond shuffled factors. Each check averages three seeds and is favorable in every paired seed.

The loss averages audio as one group and text, image, and video as three further groups. Coupled heads provide semantic cross-entropy and a symmetric multi-positive image–text contrastive loss, where samples with the same digit identity are positives.

3.2 Action-conditioned prediction

For state q_t, proposed action a_t, and task token c, the same core predicts the standardized change d_t to the endpoint eight environment steps later:

$$\mathcal{L}_{\text{act}} = \frac{1}{39}\|g_\theta(q_t, a_t, c) - d_t\|_2^2. \tag{3}$$

One optimizer minimizes

$$\mathcal{L}_{\text{joint}} = \mathcal{L}_{\text{next}} + \mathcal{L}_{\text{sem}} + 0.25\mathcal{L}_{\text{bind}} + 4\mathcal{L}_{\text{act}}. \tag{4}$$

After sensory pretraining, the action adapter trains for 800 updates with the core frozen; we call this the *sequential checkpoint*. The main deployment comparison uses this checkpoint. The *joint checkpoint* adds 200 shared-core updates with batch size 32 and learning rate 10^{-4}. The formation study evaluates this additional stage, and Table 5 tests its deployment interface. The complete checkpoint contains 1,451,388 parameters; joint optimization exposes 605,079 trainable parameters. Both sensory and action losses deliver finite, nonzero gradients to the same core tensors.

Both checkpoints expose the same endpoint interface. SpikeWorld freezes the selected θ before observed endpoint errors become online write signals.

4 Bounded Deployment State

4.1 Predict, observe, then write

At deployment, the world model, semantic heads, router, and policy are frozen. At macro-step t, the model predicts d_t from (q_t, a_t, c). Only after q_{t+8} occurs is the residual

$$r_t = g_{\theta_0}(q_t, a_t, c) - \frac{q_{t+8} - q_t - \mu_\Delta}{\sigma_\Delta} \tag{5}$$

available. The update receives state, executed command, task, and realized endpoint. It receives no label, teacher output, reward, success signal, shift family, or shift value.

Each macro-decision spans eight environment steps. The pre-update error is recorded before either state object is written, so the observed outcome can change only a subsequent prediction or command.

SpikeWorld stores two kinds of evidence. Given frozen hidden state h_t computed from the selected candidate-transformed command, a fixed basis B, and route ρ_t, route-specific residual matrices refine the prediction through

$$\tilde{h}_t = h_t + C_{\rho_t} B(h_t - \bar{h}), \qquad \hat{d}_t = W_{\text{out}} \tilde{h}_t + b_{\text{out}}. \tag{6}$$

After observing the target, the active matrix takes one normalized gradient step with learning rate 0.02 and is projected to $\|C\|_F \leq 0.20$. These matrices enter the prediction path but not the action path.

In parallel, 32 cumulative candidate losses and one transition count occupy 132 bytes. The candidate bank contains source identity, 15 shear values, and 16 attenuation values. For candidate j, the frozen model predicts from the transformed command $T_j(a_t)$ and caches

$$\ell_{t,j} = \frac{1}{39} \left\| g_{\theta_0}(q_t, T_j(a_t), c) - d_t \right\|_2^2 . \tag{7}$$

A frozen router locks source, shear, attenuation, or noise from a four-transition signature. At that point, pending losses are committed to the cumulative scores, and the lowest-loss candidate in the selected family supplies both the prediction input and the bounded inverse. Later endpoints add one new candidate-loss vector. The online process never reads the family label or true mechanism value.

Candidate scoring queries the frozen predictor without reading the residual matrices. Their changes therefore affect endpoint prediction, not candidate ranking or executed commands.

What makes a residual correctable? The state acts on the error left after candidate selection. Write that error as $e_t = d_t - g_{\theta_0}(q_t, T_{j_t}(a_t), c)$ and the matrix correction as $u_t = W_{\text{out}} C_{\rho_t} B(h_t - \bar{h})$. Its realized reduction in squared prediction error is exactly

$$\frac{\|e_t\|^2 - \|e_t - u_t\|^2}{39} = \frac{2e_t^\top u_t - \|u_t\|^2}{39}. \tag{8}$$

Calibration requires residual alignment, not simply a large correction. The fixed readout and basis restrict the available correction directions. We therefore compare both initial and adapted error; neither measures decision quality by itself.

4.2 Bounded action correction

For shear $T_{\text{shear}}(a; k)_y = a_y + k a_x$, the selected inverse proposes $\tilde{a}_y = a_y^* - \hat{k} a_x^*$. For attenuation $T_{\text{atten}}(a; g)_{x,y} = g a_{x,y}$, it proposes $\tilde{a}_{x,y} = a_{x,y}^* / \max(\hat{g}, 0.2)$. The executed command is

$$a_t = a_t^* + \text{clip}(\tilde{a}_t - a_t^*, -0.05, 0.05), \tag{9}$$

and the controller abstains when the proposal would leave the action range. Four complete transitions precede routing. Observation four locks the route and commits the first persistent update, so decision five is the earliest action that can change.

4.3 Complete runtime-state ledger

The two 96×30 matrices and the 132-byte score state contain 23,172 active bytes. The implementation allocates a 24,384-byte tensor block, including 1,212 bytes of compatibility padding. Its bounded four-transition routing prefix adds at most 4,464 numeric bytes and is cleared after routing. The complete peak mutable numeric state, including the bounded routing prefix, is therefore

$$24{,}384 \text{ allocated tensor bytes} + 4{,}464 \text{ prefix bytes} = 28{,}848 \text{ bytes}. \tag{10}$$

Table 1: **The deployment state is bounded in storage and causal reach.** Each object has one write signal, one read path, and a fixed lifetime. Bytes count peak FP32 numeric storage.

Object	Written from	Read by	Lifetime	Bytes
Residual matrices	Endpoint residual	Prediction path	Stream	23,040
Scores + count	Candidate losses	Candidate and inverse	Stream	132
Allocation overhead	Fixed allocation	None	Stream	1,212
Routing prefix	First four transitions	Frozen router	Until lock	$\leq$4,464
			Peak allocated numeric state	**28,848**

There are no optimizer moments or replay samples. Since external state is absent from the semantic readout, fixed inputs produce exactly unchanged logits as long as the foundation remains frozen. Our evaluation checks parameter and fixed-probe-logit hashes rather than inferring invariance from average accuracy.

5 Experimental Protocol

Foundation formation. Three controlled interface experiments separately test temporal rate/order controls, learned versus random future state, and true versus shuffled cross-modal composition, each over three seeds. Length-16 streams draw from SHD/SSC audio (Cramer et al., 2022), retinal digits and rendered words, causal video events, and Meta-World state–action endpoints from reach, push, pick-place, and door-open (Yu et al., 2020). The multimodal cache is balanced across its sensory sources. Five action-model initializations compare a sequential frozen-core checkpoint, an action-only shared-core update, and the joint objective in Equation (4). These five runs share one sensory-pretrained foundation and evaluate joint formation, separately from the deployment comparison below.

Deployment-state Pareto study. The main experiment crosses three independent foundation checkpoints with three fresh action seeds, producing nine top-level runs. Each uses the sequential checkpoint after 800 action-path updates, without the additional joint stage. Every run follows the same six-stage stream:

$$\text{source}_0 \rightarrow \text{attenuation}_0 \rightarrow \text{source}_1 \rightarrow \text{shear}_0 \rightarrow \text{attenuation}_1 \rightarrow \text{source return}.$$

Each stage contains all four Meta-World tasks. Mechanism labels are available to the offline evaluator but not to the online update. Predictions are scored prequentially: an error is recorded before the corresponding observation can update state.

Registered dynamics and source revisits. The shear and attenuation values used to generate the stream belong to the registered candidate bank. Shifted stages use a separate execution archive from shift calibration. Source stages revisit router-calibration episodes, including their four-transition routing prefixes. Source-return correction thus measures calibrated-source reentry, not held-out source generalization. The foundation checkpoints are independently trained and the action-model seeds are fresh. The accompanying provenance audit identifies 48 reused source-prefix rows per evaluated replica.

We compare five arms. *Frozen* performs no update. *External state* updates only the state in Section 4. *Adapter* tunes the action adapter with Adam. *Prediction-path tuning* updates the deployed prediction path with Adam. *Replay* uses the same trainable path and optimizer with a compressed buffer. For each arm, peak mutable numeric bytes include state tensors, trainable parameters, optimizer tensors, replay, and routing history. Median latency uses 2,988 CUDA-synchronized update samples per adaptive arm after discarding 16 warm-up updates from every run.

We also run a targeted classical-identification stress test on the identical six-stage layout and the same 3×3 foundation/action-seed design. Its routed 2×2 matrix RLS observes only state, command, task, and realized endpoint; separate information matrices are selected after the same four-step causal routing prefix. We compare its prediction, tracking, source-return correction, and peak mutable numeric tensor state against a replicated external state arm. No aggregate latency claim is made for this stress test.

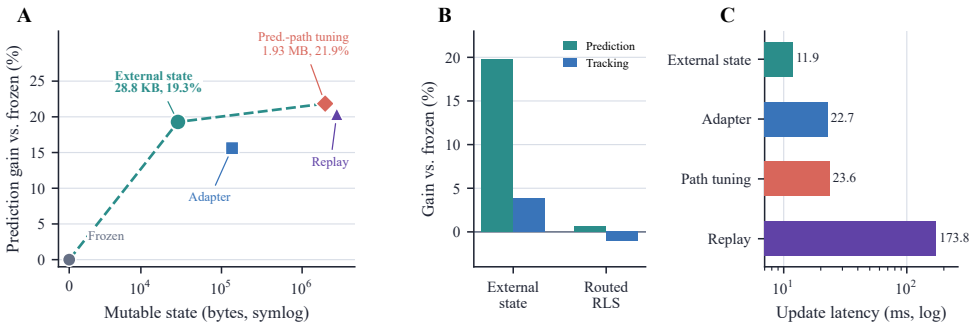


Figure 3: **External state offers a compact prediction–memory trade-off.** **A**, prediction gain versus peak mutable numeric state for the five main configurations on the registered six-stage stream; the dashed line connects their non-dominated operating points. **B**, a separate same-protocol replay compares external state with smaller routed actuator-space matrix RLS; external state yields larger prediction and tracking gains. **C**, external state updates in 11.9 ms, about twice as fast as adapter or prediction-path tuning and 14.6× faster than replay. Every comparison uses three independent foundations, each with three fresh action seeds.

The primary metrics are relative reduction in shifted prequential prediction MSE and relative reduction in physical action-tracking error against frozen behavior. Source-return false correction is the mean command-correction norm when source dynamics reappear. Action seeds are averaged within a foundation; 95% intervals use 20,000 deterministic cluster-bootstrap draws with foundation checkpoint as the independent unit.

Causal role and closed-loop studies. The implementation closure pairs four controllers—legacy full state, legacy score-only, bounded full state, and bounded score-only—over three tasks and three fresh environment seeds, for 36 traces. It compares every externally observable field and the first changed decision. A separate attenuation study evaluates six paired arms over five model seeds, three tasks, and five new environment seeds, yielding 450 trajectories. Reward and success are used only for evaluation. Their intervals use paired resampling of model, task, and replica axes. The policy, action cap, task order, and observation schedule are shared across arms.

Matched formation–deployment pilot. One independent sensory checkpoint and a new action seed supply a common sequential starting point. Three copies receive 200 matched updates with the joint objective, action loss alone, or the joint objective without its sensory next-slice term. All inherit sensory predictive pretraining. Each uses the same training-only PCA and calibration-router fitting procedure, with fixed hyperparameters. We replay the same cached six-stage trajectory for each model, comparing frozen, score-only, and full state. Errors are recorded before updates; the main pilot comparison uses shifted transitions after the four-observation prefix. This is a one-foundation development diagnostic, not a new closed-loop trial or a comparison of pretraining objectives from scratch.

6 Results

6.1 A 29 KB deployment operating point

The deployment-state comparison identifies a compact operating point (Figure 3 and Table 2). External state uses 28,848 bytes and reduces causal prequential prediction MSE by 19.28%, with a foundation-level 95% interval of [15.71, 21.80] (three independent bootstrap units). Prediction-path tuning reaches 21.85% with 1,932,488 mutable bytes. Thus, 2.57 percentage points separate the two prediction gains, while prediction-path tuning carries 66.99 times more mutable numeric state.

The same state obtains the largest mean tracking gain, 3.80%, among the five arms. Adapter, prediction-path tuning, and replay obtain 3.47%, 2.91%, and 3.35%, respectively. On calibrated-source return, external state applies zero false correction; the trainable arms retain mean corrections between 0.0045 and 0.0084 in ℓ_2 norm. Its 11.90 ms median update is 1.91× faster than adapter,

Table 2: **Deployment-state operating points.** Means use three independent foundation checkpoints, each averaged over three fresh action seeds. Latency excludes 16 warm-up updates per run and synchronizes CUDA.

Method	Mutable numeric state	Pred. gain (%)	Track. gain (%)	Update (ms)	Exact frozen model
Frozen	0	0.00	0.00	–	yes
External state	28.8 KB	19.28	3.80	11.9	yes
Adapter	135.2 KB	15.61	3.47	22.7	no
Pred.-path tuning	1.93 MB	21.85	2.91	23.6	no
Replay	2.69 MB	20.38	3.35	173.8	no

$1.99\times$ faster than prediction-path tuning, and $14.61\times$ faster than replay. These measurements locate the operating point in prediction, storage, update cost, and source-return behavior.

A routed actuator-identification stress test. Routed 2×2 matrix RLS is smaller still: its peak mutable numeric tensor state plus bounded routing prefix is 4,112 bytes. On a matched replay of the six-stage protocol, with separate shifted archives and calibrated source revisits, it obtains 0.58% prediction gain and -1.05% tracking gain, while the replicated external-state arm obtains 19.72% and 3.87%. The foundation-level paired prediction advantage is 19.14 points [13.56, 23.87], and both methods revisit calibrated source with zero correction. The evaluated routed identifier therefore does not dominate the 29 KB operating point.

Across the nine runs, every external-state parameter hash and fixed-probe-logit hash remains exact from the initial stage through source return. This is a direct property of the deployment graph: the result establishes checkpoint and same-input-output invariance, independent of whether a short accuracy average happens to change.

6.2 Prediction state and action state have different jobs

The component intervention explains what occupies the 29 KB budget (Table 3). On shear, residual matrices add 9.22 prediction points [8.13, 10.30] beyond the score-only state. On attenuation, they add 25.12 [24.23, 26.00]. Yet full and score-only state produce exactly the same executed command and action-tracking error in every paired cell. Removing the inverse makes the action improvement zero. The matrices therefore improve prediction; the 132-byte score state selects the action correction.

Table 3: **Exact role separation inside external state.** Prediction and tracking entries are relative MSE reductions (%). Matrix contribution is full minus score-only prediction. Full and score-only commands are bitwise equal.

Shift	Full pred.	Score-only pred.	Matrix contrib.	Tracking
Shear	8.13	-1.09	+9.22	43.99
Attenuation	32.63	7.51	+25.12	3.88

Across 36 paired closure traces, legacy and bounded implementations match in all observable fields. Full and score-only commands remain bitwise equal and first change at decision five. Bounding the routing prefix closes the 28,848-byte account without changing behavior.

6.3 The registered attenuation correction changes reward

The score-state path has a separately measurable behavioral effect. In the closed-loop attenuation study, external state raises mean cumulative reward from 414.68 to 422.58 and success from 53.33% to 66.67% (Table 4). The paired reward gain is 7.90, with a 95% interval of [2.48, 14.06]. Reward gains are positive on pick-place (+2.31), push (+11.61), and reach (+9.79), and across all five model seeds (+7.19 to +8.66).

Table 4: **Closed-loop control under registered attenuation.** Means over 75 fresh trajectories per arm; reward and success are evaluation-only.

Method	Mean reward	Success (%)
Frozen	414.68	53.33
External state	422.58	**66.67**
Pred.-path tuning	422.19	**66.67**
Matched scalar RLS	**434.41**	65.33
Replay	428.59	60.00
Oracle inverse	485.50	80.00

Prediction-path tuning has a paired reward difference of -0.40 [-1.63, 0.30] relative to external state. The 16-byte scalar RLS is the matched single-fault reference; it differs from the routed continual-stream identifier in Figure 3B.

6.4 Formation and calibration are distinct design axes

Relative to the sequential frozen-core checkpoint, joint formation reduces action next-state MSE by 17.10% [14.04, 20.04] and multimodal next-slice MSE by 0.75% [0.68, 0.83]. Semantic accuracy rises by 1.80 points and image–text class R@1 by 1.81 points, improving all four metrics over the sequential checkpoint. An action-only core update attains lower action MSE (0.1174 versus 0.1212) but not this joint predictive–semantic improvement. Both predictive losses reach the same shared core.

Table 5: **The same state interface across formation stages.** One-foundation matched pilot; MSE is lower-is-better. Offline MSE uses source validation data. Stream MSE averages 126 shifted predictions after routing. No-next deletes the sensory next-slice loss only in the final 200 updates.

		Shifted stream MSE		
Formation	Offline action MSE	Frozen	Score-only	Full state
Sequential	0.1497	0.3927	0.3661	0.2878
Action-only	0.1191	0.4408	0.3797	0.2918
Joint	0.1200	0.4097	0.3992	0.3116
Joint, no-next	0.1227	0.4270	0.4129	0.3186

The matched pilot in Table 5 connects these training stages to the same external-state interface. Joint formation retains a 23.96% online prediction gain, with a matrix-only MSE reduction of 0.0876 beyond score state. Its final MSE is 2.20% below the no-next control; action-only updating yields a lower final MSE still. The sequential checkpoint has the largest offline action error but the smallest adapted stream error in this pilot. Thus the interface works across the evaluated training stages, while offline accuracy alone does not order their calibrated predictions. The pilot establishes compatibility, not a general adaptation advantage from joint training.

7 Discussion and Conclusion

SpikeWorld separates predictive representation from deployment plasticity. On the registered continual stream, 28,848 bytes of bounded external state obtain 19.28% prediction gain, versus 21.85% for prediction-path tuning with 66.99 times more mutable numeric state. Residual matrices calibrate prediction; candidate scores calibrate registered actions. The formation pilot motivates evaluating adapted error alongside offline accuracy. Extending this interface to unregistered dynamics and task-aware planning remains a next step. Within the registered setting, we establish an alternative to rewriting the model: retain its representation and learn from observed futures in explicitly budgeted state.

REFERENCES

Karl J. Åström and Björn Wittenmark. On self-tuning regulators. *Automatica*, 9(2):185–199, 1973. doi: 10.1016/0005-1098(73)90073-3.

Jimmy Ba, Geoffrey E. Hinton, Volodymyr Mnih, Joel Z. Leibo, and Catalin Ionescu. Using fast weights to attend to the recent past. In *Advances in Neural Information Processing Systems*, volume 29, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/9f44e956e3a2b7b5598c625fcc802c36-Abstract.html.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mido Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=QaCCuDfBk2.

Benjamin Cramer, Yannik Stradmann, Johannes Schemmel, and Friedemann Zenke. The heidelberg spiking data sets for the systematic evaluation of spiking neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 33(7):2744–2757, 2022. doi: 10.1109/TNNLS.2020.3044364.

Guillermo Gallego, Tobi Delbruck, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J. Davison, Jorg Conradt, Kostas Daniilidis, and Davide Scaramuzza. Event-based vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(1):154–180, 2022. doi: 10.1109/TPAMI.2020.3008413.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/2de5d16682c3c35007e4e92982f1a2ba-Abstract.html.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2555–2565. PMLR, 2019. URL https://proceedings.mlr.press/v97/hafner19a.html.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1lOTC4tDS.

Nethmi Jayasinghe, Diana Gontero, Francesco Migliarba, Spencer T. Brown, Vinod K. Sangwan, Mark C. Hersam, and Amit Ranjan Trivedi. Residual control for fast recovery from dynamics shifts. *arXiv preprint arXiv:2603.07775*, 2026. URL https://arxiv.org/abs/2603.07775.

Simon Klenk, David Bonello, Lukas Koestler, Nikita Araslanov, and Daniel Cremers. Masked event modeling: Self-supervised pretraining for event cameras. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2378–2388, 2024. doi: 10.1109/WACV57701.2024.00237.

Ashish Kumar, Zipeng Fu, Deepak Pathak, and Jitendra Malik. RMA: Rapid motor adaptation for legged robots. In *Robotics: Science and Systems XVII*, 2021. doi: 10.15607/RSS.2021.XVII.011.

JB Lanier, Kyungmin Kim, Armin Karamzade, Yifei Liu, Ankita Sinha, Kat He, Davide Corsi, and Roy Fox. Adapting world models with latent-state dynamics residuals. In *Proceedings of the 8th Annual Learning for Dynamics and Control Conference*, volume 331 of *Proceedings of Machine Learning Research*, pp. 117–144. PMLR, 2026. URL https://proceedings.mlr.press/v331/lanier26a.html.

Patrick Lichtsteiner, Christoph Posch, and Tobi Delbruck. A 128×128 120 db 15 μs latency asynchronous temporal contrast vision sensor. *IEEE Journal of Solid-State Circuits*, 43(2):566–576, 2008. doi: 10.1109/JSSC.2007.914337.

Michael L. Littman, Richard S. Sutton, and Satinder Singh. Predictive representations of state. In *Advances in Neural Information Processing Systems*, volume 14, pp. 1555–1561, 2001.

Lucas Maes, Quentin Le Lidec, Damien Scieur, Yann LeCun, and Randall Balestriero. LeWorld-Model: Stable end-to-end joint-embedding predictive architecture from pixels. *arXiv preprint arXiv:2603.19312*, 2026. doi: 10.48550/arXiv.2603.19312. URL https://arxiv.org/abs/2603.19312.

Thomas Miconi, Kenneth O. Stanley, and Jeff Clune. Differentiable plasticity: Training plastic neural networks with backpropagation. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 3559–3568. PMLR, 2018. URL https://proceedings.mlr.press/v80/miconi18a.html.

Jihun Moon and Seong-Woo Kim. SPREAD: Scalable pre-trained world model for adaptive dynamics model. *IEEE Robotics and Automation Letters*, 11(6):7460–7467, 2026. doi: 10.1109/LRA.2026.3688061.

Lorenzo Mur-Labadia, Matthew Muckley, Amir Bar, Mido Assran, Koustuv Sinha, Mike Rabbat, Yann LeCun, Nicolas Ballas, and Adrien Bardes. V-JEPA 2.1: Unlocking dense features in video self-supervised learning. *arXiv preprint arXiv:2603.14482*, 2026. doi: 10.48550/arXiv.2603.14482. URL https://arxiv.org/abs/2603.14482.

Anusha Nagabandi, Ignasi Clavera, Simin Liu, Ronald S. Fearing, Pieter Abbeel, Sergey Levine, and Chelsea Finn. Learning to adapt in dynamic, real-world environments through meta-reinforcement learning. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HyztsoC5Y7.

Kumpati S. Narendra and Jeyendran Balakrishnan. Adaptive control using multiple models. *IEEE Transactions on Automatic Control*, 42(2):171–187, 1997. doi: 10.1109/9.554398.

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

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 16888–16905. PMLR, 2022. URL https://proceedings.mlr.press/v162/niu22a.html.

Ilija Radosavovic, Bike Zhang, Baifeng Shi, Jathushan Rajasegaran, Sarthak Kamat, Trevor Darrell, Koushil Sreenath, and Jitendra Malik. Humanoid locomotion as next token prediction. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-2518. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/90afd20dc776bc8849c31d61a0763a0b-Abstract-Conference.html.

Kate Rakelly, Aurick Zhou, Chelsea Finn, Sergey Levine, and Deirdre Quillen. Efficient off-policy meta-reinforcement learning via probabilistic context variables. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5331–5340. PMLR, 2019. URL https://proceedings.mlr.press/v97/rakelly19a.html.

Mona Schirmer, Dan Zhang, and Eric Nalisnick. Temporal test-time adaptation with state-space models. *Transactions on Machine Learning Research*, 2025. URL https://openreview.net/forum?id=HFETOmUtrV.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei A. Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9229–9248. PMLR, 2020. URL https://proceedings.mlr.press/v119/sun20b.html.

Yu Sun, Xinhao Li, Karan Dalal, Jiarui Xu, Arjun Vikram, Genghan Zhang, Yann Dubois, Xinlei Chen, Xiaolong Wang, Sanmi Koyejo, Tatsunori Hashimoto, and Carlos Guestrin. Learning to (Learn at test time): RNNs with expressive hidden states. In *Proceedings of the 42nd International*

Conference on Machine Learning, volume 267 of *Proceedings of Machine Learning Research*, pp. 57503–57522. PMLR, 2025. URL https://proceedings.mlr.press/v267/sun25h.html.

SunRisingAI Lab. Phi-WM 1.0 ActEffect: From predictive foresight to consequence feedback in robot learning. Technical report, SunRisingAI, 2026. URL https://www.sunrisingai.com/upload/202608/e597763b7e870a6ec5951d487b4b3e7f.pdf.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=uXl3bZLkr3c.

Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7201–7211, 2022. doi: 10.1109/CVPR52688.2022.00706.

World Labs Team. Atlas: A world model for spatial intelligence. *World Labs Blog*, 2026. URL https://www.worldlabs.ai/blog/atlas. Technical release, September 1.

Yan Yang, Liyuan Pan, and Liu Liu. Event camera data pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10699–10709, 2023.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-World: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pp. 1094–1100. PMLR, 2020. URL https://proceedings.mlr.press/v100/yu20a.html.

Wenhao Yu, Jie Tan, C. Karen Liu, and Greg Turk. Preparing for the unknown: Learning a universal policy with online system identification. In *Robotics: Science and Systems XIII*, 2017. doi: 10.15607/RSS.2017.XIII.048.

Ziqiao Yu, Pengfei Sun, and Dan F. M. Goodman. Beyond rate coding: Surrogate gradients enable spike timing learning in spiking neural networks. *Neuromorphic Computing and Engineering*, 6: 014016, 2026. doi: 10.1088/2634-4386/ae46d5.

Friedemann Zenke and Surya Ganguli. SuperSpike: Supervised learning in multilayer spiking neural networks. *Neural Computation*, 30(6):1514–1541, 2018. doi: 10.1162/NECO_A_01086.

Youmin Zhang and Jin Jiang. Bibliographical review on reconfigurable fault-tolerant control systems. *Annual Reviews in Control*, 32(2):229–252, 2008. doi: 10.1016/j.arcontrol.2008.03.008.

Chenlin Zhou, Han Zhang, Zhaokun Zhou, Liutao Yu, Liwei Huang, Xiaopeng Fan, Li Yuan, Zhengyu Ma, Huihui Zhou, and Yonghong Tian. QKFormer: Hierarchical spiking transformer using q-k attention. In *Advances in Neural Information Processing Systems*, volume 37, 2024a. doi: 10.52202/079017-0416.

Jiazhou Zhou, Xu Zheng, Yuanhuiyi Lyu, and Lin Wang. EventBind: Learning a unified representation to bind them all for event-based open-world understanding. In *Computer Vision – ECCV 2024*, volume 15128 of *Lecture Notes in Computer Science*, pp. 477–494. Springer, 2024b. doi: 10.1007/978-3-031-72897-6_27.

Zhaokun Zhou, Yuesheng Zhu, Chao He, Yaowei Wang, Shuicheng Yan, Yonghong Tian, and Li Yuan. Spikformer: When spiking neural network meets transformer. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=frE4fUwz_h.

AI USE STATEMENT

We used generative AI tools to assist with writing, polishing, and translation; literature retrieval and discovery; and research ideation. The authors reviewed the AI-assisted work and take responsibility for the manuscript and accompanying artifacts.