PROBE AND CALIBRATE: STOCHASTIC FIRING-BOUNDARY ADAPTATION FOR SPIKING NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Distribution shifts can perturb the membrane responses of Spiking Neural Networks (SNNs), misaligning source-trained firing boundaries with target-domain activity. Threshold calibration provides a compact adaptation interface, but prediction confidence alone does not reveal how responsive neuronal firing is to boundary changes. We propose ProCal, a probe-and-calibrate framework for online test-time adaptation that uses this local responsiveness to guide threshold updates. During adaptation, each adapted layer receives a stochastic threshold with a learnable location and scale. The induced firing probability yields an analytic responsiveness score: the normalized conditional variance of the binary firing response. ProCal aggregates this score by sample and combines it with prediction confidence to weight entropy minimization. It updates only two scalars per adapted layer, totaling 38 on MS-ResNet-20, while freezing synaptic weights and requiring no adaptation-specific source retraining. Each target batch receives one stochastic update followed by prediction with the learned locations as noise-free thresholds. Across two event-corruption suites and two controlled frame-based variants, ProCal achieves the highest mean accuracy among the evaluated methods in three settings. It reaches 53.82% on CIFAR10dvs-C and 64.72% on DVS128Gesture-C; component ablations further show gains of 1.02 and 1.42 percentage points over deterministic threshold adaptation.

1 INTRODUCTION

Spiking Neural Networks (SNNs) process information through membrane-potential accumulation and discrete spike generation, offering an energy-efficient paradigm for neuromorphic vision Pfeiffer & Pfeil (2018); Roy et al. (2019). Their predictions depend on the interaction between membrane responses Kim et al. (2022) and firing thresholds Fang et al. (2021); Kobayashi et al. (2009), making neuronal dynamics particularly vulnerable to distribution shift Ding et al. (2022; 2024). After deployment, changes in sensor noise, event density, or motion can alter membrane responses Gallego et al. (2020). Source-trained thresholds may then trigger too many or too few spikes, and these altered firing patterns propagate through subsequent temporal dynamics. This mismatch motivates calibrating firing boundaries to target-domain activity, as illustrated in Fig. 1.

Test-time adaptation (TTA) updates pretrained models using unlabeled target data. Existing methods rely on predictive entropy, augmentation consistency, or reliability criteria Wang et al. (2020; 2022); Niu et al. (2022; 2023). SNN-specific approaches additionally exploit neuronal dynamics: SPACE enforces spike-aware consistency across augmented views Luo et al. (2026), while Threshold Modulation (TM) adapts firing thresholds during online inference Zhao et al. (2025). These methods establish neuronal dynamics as a useful adaptation interface. We ask a complementary question: can local responses to threshold perturbations reveal which target samples can provide a boundary-calibration signal?

Our starting point is that choosing which parameters to adapt and which samples to emphasize are closely related. When adaptation acts on firing thresholds, a sample's local firing response to boundary changes is a relevant signal alongside its output confidence. For a fixed membrane response, threshold perturbations are more likely to switch the firing state near the boundary than far

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.

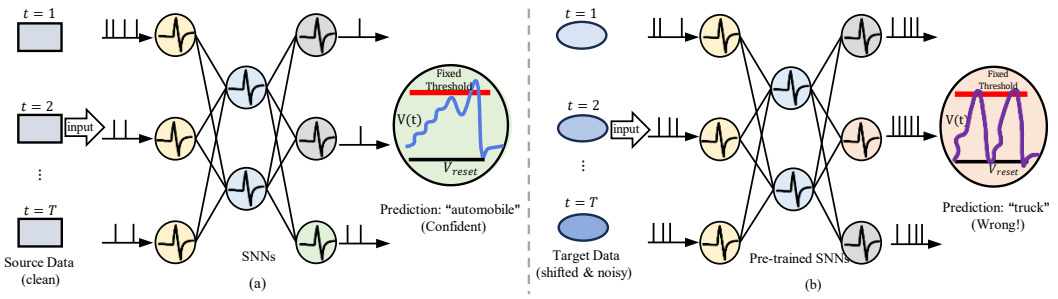

Figure 1: Distribution shifts can alter spike generation by perturbing membrane responses relative to fixed source-domain firing boundaries. ProCal probes local firing-state responsiveness to boundary perturbations and calibrates the boundaries using unlabeled target data.

from it. Prediction confidence describes the model's certainty about a label; boundary responsiveness describes how readily local firing states can change under the chosen perturbation scale. ProCal combines these signals to emphasize confident samples with responsive neuronal activity.

ProCal implements this idea through a *probe-and-calibrate* procedure. Each adapted layer receives a stochastic threshold whose location calibrates the firing boundary and whose scale controls the probing range. A sampled rollout supplies the membrane responses at which we analytically evaluate local firing probabilities. Their normalized Bernoulli variances quantify boundary responsiveness, which is aggregated by sample and combined with confidence to weight entropy minimization. This connects the sample-weighting signal directly to the neuronal quantity being adapted. Only the layer-wise locations and scales are updated, totaling 38 scalars on MS-ResNet-20, without adaptation-specific source retraining. After each update, ProCal predicts with the learned locations as noise-free thresholds, preserving the original network topology.

We evaluate ProCal on two event-corruption suites and two controlled frame-based diagnostic variants. It reaches 53.82% on CIFAR10dvs-C and 64.72% on DVS128Gesture-C, improving on the strongest evaluated baselines by 1.00 and 1.25 percentage points, respectively. Direct deterministic threshold adaptation is already a strong reference: ProCal improves on it by 1.02 and 1.42 points, with component ablations tracing the additional gains through probing, scale learning, and sample weighting. Together, these results support the value of using firing boundaries as both adaptation parameters and probes for weighting target samples.

Our main contributions are summarized as follows:

- We introduce stochastic firing-boundary probing for SNN test-time adaptation. The induced firing probabilities provide an analytic local responsiveness score that is monotonically related to the probability of switching a firing state at a fixed membrane potential.
- We develop ProCal, which couples this responsiveness score with prediction confidence to weight entropy-based boundary calibration. It updates two scalars per adapted layer and uses noise-free thresholds for prediction, without adaptation-specific source retraining.
- We evaluate ProCal on two event-corruption suites and two controlled frame-based variants, obtaining the highest mean accuracy among the evaluated methods in three settings. Component ablations quantify the gains over deterministic threshold adaptation, while firing-statistic diagnostics and a recurring-domain stream examine adaptation behavior.

2 Related Work

2.1 Test-Time Adaptation for Spiking Neural Networks

Test-time adaptation updates pretrained models using unlabeled target data. Representative approaches include entropy minimization in TENT Wang et al. (2020), teacher–student consistency in CoTTA Wang et al. (2022), reliable-sample selection in EATA Niu et al. (2022), and sharpness-aware optimization in SAR Niu et al. (2023). These methods define general adaptation objectives

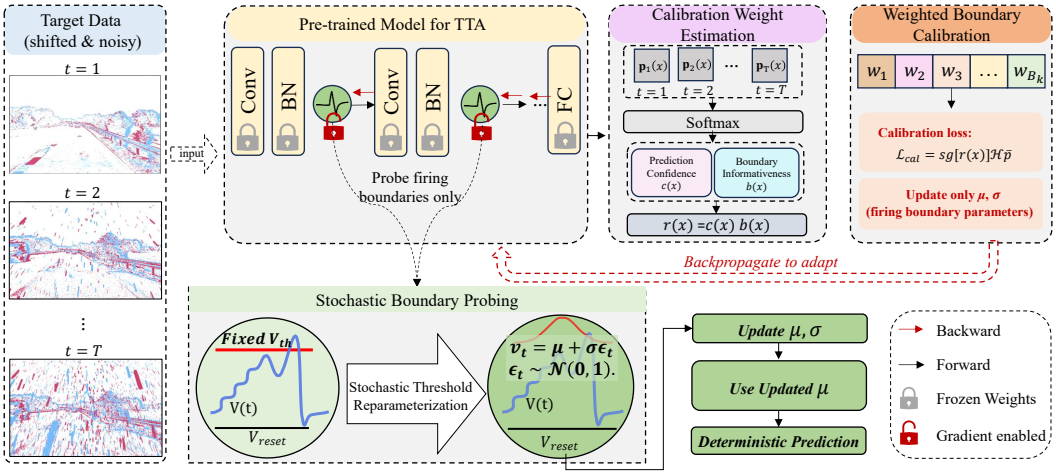

Figure 2: Overview of ProCal. A stochastic rollout provides membrane responses for evaluating analytic firing probabilities. Their normalized Bernoulli variances provide local responsiveness scores, which are aggregated into a sample-level boundary-responsiveness score. Together with prediction confidence, this signal forms a calibration weight for entropy-based boundary adaptation. ProCal updates the layer-wise boundary locations μ and probing scales σ, where $\sigma = \mathrm{Softplus}(\rho) + \sigma_{\min}$ and ρ is the optimized unconstrained parameter. After each update, threshold noise is removed and the updated locations μ are used as deterministic thresholds for prediction.

and update criteria; SNNs additionally expose threshold-dependent neuronal dynamics as an adaptation interface.

SNN-specific methods exploit this interface in different ways. SPACE Luo et al. (2026) enforces spike-aware consistency across augmented target views, while membrane-potential distribution alignment provides another route Tang (2026). TM Zhao et al. (2025), the closest method in adaptation interface, incorporates membrane potential batch normalization (MPBN) during source training and reparameterizes it as firing thresholds for online adaptation. ProCal instead starts from a conventionally pretrained checkpoint and uses threshold perturbations as test-time probes. For a fixed membrane response, the induced firing probability yields a closed-form measure of local boundary responsiveness; aggregating this measure by sample and combining it with confidence produces a calibration weight without augmented views or explicit distribution alignment. The distinction is therefore not threshold adaptation itself, but the use of stochastic boundary probing to expose a local conditional response before a deterministic, location-based prediction pass.

2.2 Adaptive Thresholds and Stochastic Neuronal Dynamics

Firing thresholds govern the transition from accumulated membrane potentials to discrete spikes and have been optimized to improve SNN accuracy, efficiency, and temporal dynamics Zhang et al. (2023); Chen et al. (2022). Adaptive and stochastic neuron designs further introduce learnable or probabilistic dynamics during source training Zhang et al. (2024); Gygax & Zenke (2024), while robustness studies examine how shifted membrane responses alter spike generation Guo et al. (2022); Zhu et al. (2024). These works primarily treat adaptive or stochastic dynamics as a training-time mechanism rather than using controlled boundary perturbations to probe target-domain boundary responsiveness at test time.

ProCal introduces stochasticity only as a test-time probe for an existing pretrained SNN and requires no adaptation-specific source retraining. Layer-wise threshold locations calibrate neuronal excitability, while their scales control the range of local exploration. Unlike stochastic neuron designs intended for deployment, ProCal removes threshold noise after each update and retains only the learned locations for prediction.

3 METHOD

3.1 PROBLEM FORMULATION

We consider a pretrained SNN with synaptic parameters θ, L spiking layers, and source firing thresholds $\{v_{\mathrm{th,src}}^{(l)}\}_{l=1}^{L}$. At test time, unlabeled target mini-batches $\mathcal{B}_1, \mathcal{B}_2, \ldots$ arrive sequentially from a shifted distribution. Each batch follows an adapt-then-predict protocol: it first produces one stochastic update and is then evaluated with threshold noise removed. The batch is not replayed afterward, and the adapted parameters are retained throughout the target stream. Neither source data nor target labels are available. The objective is to calibrate a small set of layer-wise boundary parameters while keeping θ fixed.

For a leaky integrate-and-fire (LIF) neuron Gerstner et al. (2014), suppressing layer and neuron indices for clarity, the dynamics at simulation timestep t are

$$\begin{aligned} H_t &= \alpha V_{t-1} + (1-\alpha) X_t, \\ S_t &= \Theta(H_t - v_{\mathrm{th}}), \\ V_t &= H_t(1-S_t) + V_{\mathrm{reset}} S_t, \end{aligned} \tag{1}$$

where X_t is the input current, H_t the pre-spike membrane potential, V_t the post-reset membrane potential, and $S_t \in \{0,1\}$ the emitted spike. The parameter α controls membrane-potential retention, V_{reset} is the reset potential, v_{th} is the firing threshold, and Θ denotes the Heaviside step function.

The firing decision depends on the membrane-to-threshold margin

$$m_t = H_t - v_{\mathrm{th}}. \tag{2}$$

The threshold separates firing from silence. When target-domain shifts alter membrane responses, a fixed source threshold changes the margin distribution and may systematically increase or suppress spike activity. Moving the boundary directly adjusts these firing decisions without modifying synaptic weights.

ProCal uses this same boundary to probe the local firing response. At a fixed perturbation scale, membrane responses near the boundary are more likely to change firing state than those farther away (Appendix A). Stochastic probing quantifies this local responsiveness (Sec. 3.2), and confidence-weighted calibration uses it to determine each sample's contribution to the threshold update (Sec. 3.3).

3.2 STOCHASTIC FIRING-BOUNDARY PROBING

ProCal probes local firing-state changes by introducing stochastic thresholds during adaptation. A learnable location and scale in each adapted layer jointly support boundary calibration and local exploration.

For an adapted layer l, the threshold at simulation timestep t is sampled as

$$v_t^{(l)} = \mu^{(l)} + \sigma^{(l)} \epsilon_t^{(l)}, \qquad \epsilon_t^{(l)} \sim \mathcal{N}(0,1), \tag{3}$$

where $\mu^{(l)}$ is the boundary location and $\sigma^{(l)} > 0$ controls the perturbation range. At each timestep, one scalar $\epsilon_t^{(l)}$ is drawn per adapted layer and shared across all neurons and samples in that layer; draws are independent across layers and timesteps. Thus, the forward pass applies a coherent layer-wise boundary perturbation rather than independent neuron-wise noise. The location is initialized from the source threshold:

$$\mu_0^{(l)} = v_{\mathrm{th,src}}^{(l)}, \tag{4}$$

A positive probing scale is enforced through

$$\sigma^{(l)} = \mathrm{Softplus}\left(\rho^{(l)}\right) + \sigma_{\min}, \tag{5}$$

Here, $\rho^{(l)}$ is unconstrained and $\sigma_{\min} = 10^{-6}$ preserves a nonzero probing range. The only trainable variables are the two layer-wise scalars $\mu^{(l)}$ and $\rho^{(l)}$; the noise samples are not optimized. The penalty introduced in Sec. 3.3 discourages excessively broad perturbations.

The Gaussian parameterization yields a closed-form local firing probability. Condition on the randomness preceding the current threshold draw, so that the pre-spike membrane potential has a fixed value h. Omitting the layer index, we obtain

$$\begin{aligned} q(h;\mu,\sigma) &= \mathbb{P}_{\epsilon\sim\mathcal{N}(0,1)}\left(\mu+\sigma\epsilon < h\right) \\ &= \Phi\left(\frac{h-\mu}{\sigma}\right), \end{aligned} \tag{6}$$

where Φ is the standard Gaussian cumulative distribution function. The firing probability equals $1/2$ at $h = \mu$ and approaches zero or one as the normalized distance $|h-\mu|/\sigma$ increases. This probability describes the current firing event conditional on its membrane potential; earlier sampled spikes can affect the membrane potentials at later steps.

For neuron j in layer l at timestep t, we evaluate $q_{t,j}^{(l)} = q(H_{t,j}^{(l)}; \mu^{(l)}, \sigma^{(l)})$ at the membrane potential observed in the sampled rollout. The sampled thresholds determine the binary spikes used in the forward computation, whereas these analytic probabilities provide the auxiliary signal used to weight calibration samples in Sec. 3.3.

After each adaptation update, probing is disabled and each learned location $\mu^{(l)}$ is used as a plug-in threshold. For a fixed h, away from the tie case, $q(h;\mu,\sigma) > 1/2$ exactly when $h > \mu$. The prediction pass therefore uses the most probable local firing state at each encountered membrane potential.

3.3 Confidence- and Responsiveness-Weighted Boundary Calibration

ProCal weights each target sample using output confidence and local boundary responsiveness within the spiking layers. Their product determines the sample's contribution to entropy-based calibration.

Prediction confidence. For a target input x, let $\mathbf{p}_t(x)$ denote the class-probability prediction at simulation timestep t. We average these predictions over T timesteps and compute a normalized confidence score:

$$\bar{\mathbf{p}}(x) = \frac{1}{T}\sum_{t=1}^{T}\mathbf{p}_t(x), \qquad c(x) = 1 - \frac{\mathcal{H}(\bar{\mathbf{p}}(x))}{\log C}, \tag{7}$$

where $\mathcal{H}(\mathbf{p}) = -\sum_{k=1}^{C} p_k \log p_k$ and C is the number of classes. The score $c(x) \in [0,1]$ increases as predictive entropy decreases and serves as a heuristic measure of prediction reliability.

Local boundary responsiveness. For the same input, we use the local firing probabilities $q_{t,j}^{(l)}$ from Sec. 3.2 to define

$$b_{t,j}^{(l)} = 4q_{t,j}^{(l)}\left(1 - q_{t,j}^{(l)}\right). \tag{8}$$

This score is the variance of the Bernoulli firing response, normalized to $[0,1]$. It equals one at $q_{t,j}^{(l)} = 1/2$ and approaches zero as firing or silence becomes certain. Its operational meaning follows from the local switching probability relative to the plug-in boundary: for a fixed membrane response, $P_{\text{flip}} = \min(q, 1-q)$ and $b = 4P_{\text{flip}}(1 - P_{\text{flip}})$. Thus, the score preserves the ordering of local switching probabilities under the current probing scale. Appendix A provides the derivation.

We aggregate these local scores into a sample-level measure of boundary responsiveness:

$$b(x) = \frac{1}{L}\sum_{l=1}^{L}\frac{1}{TN_l}\sum_{t=1}^{T}\sum_{j=1}^{N_l} b_{t,j}^{(l)}, \tag{9}$$

where N_l is the number of neurons in layer l. Averaging first within and then across layers assigns equal weight to every layer regardless of its size. The resulting score $b(x) \in [0,1]$ summarizes local conditional responsiveness along one sampled rollout. It averages normalized individual firing variances; the shared threshold draws induce correlations that this score does not capture.

Confidence- and responsiveness-weighted calibration. We combine the two scores multiplicatively:

$$r(x) = c(x)^{\gamma_c} b(x)^{\gamma_b}, \tag{10}$$

where $\gamma_c, \gamma_b > 0$ control the relative emphasis on the two signals. The product gives greater weight to samples that are both confident and locally responsive to threshold changes, and smoothly attenuates either low-confidence or low-responsiveness samples.

For the incoming target mini-batch $\mathcal{B}_k$, the adaptation objective is

$$\mathcal{L}_k = \frac{1}{|\mathcal{B}_k|} \sum_{x \in \mathcal{B}_k} \mathrm{sg}[r(x)]\, \mathcal{H}\left(\bar{\mathbf{p}}(x)\right) + \frac{\lambda_{\mathrm{p}}}{L} \sum_{l=1}^{L} \left(\sigma^{(l)}\right)^2. \tag{11}$$

The stop-gradient operator sg fixes the weights during differentiation, preventing the model from reducing the loss merely by lowering a sample's weight. We recompute the weights from the current sampled rollout at every step, giving a one-step reweighted surrogate conditioned on that noise draw. The weights are used without batchwise rescaling, so they control both relative sample contributions and the overall entropy-gradient magnitude. Averaging over $|\mathcal{B}_k|$ makes uniformly low scores attenuate the batch's entropy term. The scale penalty discourages broad perturbations, while $\sigma_{\min}$ retains a nonzero probing range.

We optimize only $\{\mu^{(l)}, \rho^{(l)}\}_{l=1}^{L}$, using surrogate gradients through the reparameterized stochastic thresholds and computing $\sigma^{(l)}$ via Eq. (5). All synaptic weights remain frozen. Each incoming batch follows the adapt-then-predict protocol: one Adam update is followed by a noise-free forward pass on the same samples with the updated boundary locations. No labels, future batches, or replayed batches are used. Appendix B details the surrogate function, state-reset conventions, and complete procedure.

4 Experiments

4.1 Experimental Setup

Benchmarks and metrics. We evaluate ProCal on two event-corruption suites and two controlled frame-based diagnostic variants. CIFAR10dvs-C and DVS128Gesture-C each contain five event-stream corruptions: DropTime, DropArea, DownSample, Jitter, and UniformNoise. CIFAR10-C-T and ImageNet1K-C-T are constructed from CIFAR10-C and ImageNet-C Hendrycks & Dietterich (2019) at severity 5 by adding independent timestep-wise perturbations; they contain 15 corruptions grouped into noise, blur, weather, and digital categories. These two variants test temporal input variation in a controlled setting rather than model a specific sensor process. Appendix B.2 specifies their construction and all corruption parameters. We report Top-1 accuracy (%) for each corruption and the mean across corruption types (Avg.).

Baselines. We compare ProCal with the pretrained model without adaptation (Source), four representative TTA methods—TENT, CoTTA, EATA, and SAR—and the SNN-specific method SPACE. All compared methods start from the same pretrained SNN checkpoint within each benchmark and access neither source data nor target labels during adaptation. The comparison therefore isolates test-time procedures under a common source model.

TM Zhao et al. (2025) is not included quantitatively because its reported pipeline couples MPBN-based source training with threshold reparameterization at deployment, whereas our protocol excludes adaptation-specific source retraining. Published TM numbers are consequently not directly comparable under the common-checkpoint setup. A common-checkpoint evaluation of TM is absent, and we treat this as a limitation rather than evidence of superiority over TM.

Per-corruption online evaluation protocol. For the main comparisons in Tables 1 and 2, each corruption forms a separate target stream with a known boundary. Mini-batches are processed sequentially without replay, while adapted parameters and optimizer states carry over within the stream. Each mini-batch follows the adapt-then-predict protocol: it is first used for one unlabeled update and then evaluated on the same samples with threshold noise removed. Before a new corruption, we restore the source checkpoint, reinitialize the boundary parameters, and clear the optimizer

	Method	Adapt. Params.	DropTime	DropArea	DownSample	Jitter	UniformNoise	Avg.
DVS128Gesture-C	Source	–	75.00	76.73	21.87	38.54	19.44	$46.32_{\pm 0.12}$
	CoTTA Wang et al. (2022)	272.7K	83.33	**77.78**	26.73	50.34	79.16	$63.47_{\pm 0.23}$
	TENT Wang et al. (2020)	1.6K	82.98	77.08	27.08	51.04	78.81	$63.40_{\pm 0.10}$
	EATA Niu et al. (2022)	1.6K	**84.72**	74.30	26.38	49.30	78.81	$62.70_{\pm 0.18}$
	SAR Niu et al. (2023)	0.7K	83.68	76.73	28.81	49.65	77.78	$63.33_{\pm 0.25}$
	SPACE Luo et al. (2026)	27.2K	75.69	69.10	21.53	45.49	20.83	$46.53_{\pm 0.14}$
	ProCal (Ours)	**38**	84.01	75.85	**32.24**	**52.32**	**79.19**	$\mathbf{64.72_{\pm 0.11}}$
CIFAR10dvs-C	Source	–	55.50	**56.20**	18.90	20.10	44.50	$39.04_{\pm 0.21}$
	CoTTA Wang et al. (2022)	272.7K	61.60	53.10	25.90	52.70	**69.50**	$52.56_{\pm 0.10}$
	TENT Wang et al. (2020)	1.6K	60.70	53.70	26.60	51.30	68.70	$52.20_{\pm 0.17}$
	EATA Niu et al. (2022)	1.6K	61.70	52.00	27.40	51.80	68.90	$52.36_{\pm 0.24}$
	SAR Niu et al. (2023)	0.7K	61.10	54.50	**28.70**	50.90	68.90	$52.82_{\pm 0.16}$
	SPACE Luo et al. (2026)	27.2K	56.70	**56.20**	22.40	21.70	48.70	$41.14_{\pm 0.22}$
	ProCal (Ours)	**38**	**63.40**	55.20	28.10	**52.90**	**69.50**	$\mathbf{53.82_{\pm 0.21}}$

Table 1: Online adaptation on neuromorphic corruption benchmarks. We report Top-1 accuracy (%) for each corruption and the mean across corruptions. Bold and underlined values denote the best and second-best results, respectively.

	Method	Noise			Blur					Weather			Digital				Avg.
		Gau	Sho	Imp	Def	Gla	Mot	Zoo	Sno	Fro	Fog	Bri	Con	Pix	Ela	Jpe	
ImageNet1K-C-T	Source	6.7	9.2	6.7	8.0	10.8	12.1	10.6	7.8	11.5	9.1	37.1	0.5	36.9	37.8	42.0	$16.5_{\pm 0.13}$
	CoTTA Wang et al. (2022)	25.7	27.7	25.8	21.3	24.8	26.3	26.3	22.2	27.1	31.6	49.4	3.5	47.8	50.0	48.2	$30.5_{\pm 0.21}$
	TENT Wang et al. (2020)	20.7	24.9	21.8	17.1	21.9	23.1	24.3	11.5	13.4	32.6	48.5	0.7	47.1	49.2	47.7	$27.0_{\pm 0.10}$
	EATA Niu et al. (2022)	25.6	28.0	25.6	21.5	25.1	26.7	26.4	22.8	27.5	32.1	49.6	3.4	48.1	**50.5**	48.4	$30.8_{\pm 0.18}$
	SAR Niu et al. (2023)	**26.6**	**29.1**	**26.5**	**22.6**	**26.1**	**27.6**	**27.1**	**23.3**	**27.7**	**33.6**	**49.7**	2.9	**48.2**	50.4	**48.6**	$31.3_{\pm 0.24}$
	ProCal (Ours)	25.4	27.8	26.0	21.7	25.3	26.7	26.1	21.8	**27.7**	32.6	49.3	**5.8**	47.7	50.2	47.5	$30.8_{\pm 0.12}$
CIFAR10-C-T	Source	31.8	36.2	38.4	46.7	38.5	46.6	50.0	58.6	46.8	32.6	77.3	12.5	38.5	57.7	69.5	$45.4_{\pm 0.15}$
	CoTTA Wang et al. (2022)	59.8	62.1	58.6	72.4	59.2	69.6	72.7	67.1	68.2	65.6	80.3	49.9	62.1	69.1	72.1	$65.9_{\pm 0.11}$
	TENT Wang et al. (2020)	60.6	62.4	58.6	72.9	58.9	70.6	73.4	67.0	68.1	65.6	80.8	50.0	67.2	69.7	72.2	$66.5_{\pm 0.22}$
	EATA Niu et al. (2022)	59.6	62.1	58.5	72.1	58.5	69.9	72.4	66.6	67.5	65.1	80.4	49.6	66.5	69.6	71.6	$66.0_{\pm 0.17}$
	SAR Niu et al. (2023)	**60.9**	**63.3**	**58.7**	72.4	59.3	69.6	72.9	67.1	68.6	66.3	80.6	48.9	67.5	69.4	71.7	$66.5_{\pm 0.14}$
	SPACE Luo et al. (2026)	32.1	36.1	38.5	46.6	38.1	46.3	50.0	58.9	46.8	32.3	77.0	12.7	38.4	57.7	69.3	$45.4_{\pm 0.20}$
	ProCal (Ours)	60.5	62.5	58.1	**74.3**	**59.4**	**71.4**	**74.5**	**67.4**	**69.3**	**68.4**	**82.1**	**55.5**	**68.4**	**70.2**	**72.4**	$\mathbf{67.6_{\pm 0.17}}$

Table 2: Online adaptation on temporally perturbed frame-based benchmarks. We report Top-1 accuracy (%) for 15 corruptions and their mean. Bold and underlined values denote the best and second-best results, respectively.

state; the corruption identity is used only to define this reset and never enters the objective. Adaptation therefore accumulates within, but not across, corruption types. Appendix B gives the full procedure, and Appendix E separately evaluates retained adaptation state across recurring shifts.

Implementation details. We use MS-ResNet-20 Hu et al. (2024) with $L = 19$ adapted spiking layers, yielding 38 trainable boundary parameters. The simulation horizon is $T = 16$ for neuromorphic inputs and $T = 6$ for frame-based inputs. Results are reported as mean $\pm$ standard deviation over three runs with different random seeds. Source-domain firing statistics are used only for post-hoc analysis and never during adaptation. Appendix B provides the initialization, dataset-specific batch sizes and learning rates, remaining hyperparameters, and hardware.

4.2 Main Results

Neuromorphic benchmarks. Table 1 shows that ProCal achieves the highest mean accuracy among the evaluated methods on both neuromorphic benchmarks: 53.82% on CIFAR10dvs-C and 64.72% on DVS128Gesture-C. These exceed the strongest baselines, SAR and CoTTA, by 1.00 and 1.25 percentage points, respectively.

The per-corruption results reveal where these mean differences arise. On CIFAR10dvs-C, ProCal is 1.70 points above the strongest baseline on DropTime and ties CoTTA at 69.50% on UniformNoise. On DVS128Gesture-C, it is 3.43 and 1.28 points above the strongest baseline on DownSample and Jitter, respectively. The behavior is not uniform: under DropArea, ProCal remains below the unadapted Source model on both datasets.

Beyond deterministic threshold adaptation. Directly optimizing the 19 threshold locations with unweighted entropy already reaches 52.80% and 63.30% on the two event benchmarks (Table 3). Full ProCal adds 1.02 and 1.42 points, respectively. These comparisons establish the deterministic boundary as a strong reference and quantify the additional gains from the complete probe-and-calibrate procedure. Section 5.1 examines the contributions of its individual components.

Frame-based benchmarks. On CIFAR10-C-T, ProCal achieves the highest mean accuracy, 67.6%, exceeding TENT and SAR by 1.1 points (Table 2). It leads on all 12 corruptions in the blur, weather, and digital categories. The largest gain over the strongest baseline is on contrast, from 50.0% to 55.5%, followed by fog, from 66.3% to 68.4%. SAR remains strongest on the three noise corruptions.

On ImageNet1K-C-T, ProCal obtains 30.8% average accuracy, 0.5 points below SAR. Its clearest per-corruption advantage is again on contrast, reaching 5.8% versus 3.5% for the strongest baseline, although the low absolute accuracy shows that this corruption remains challenging.

Adaptation-parameter efficiency. ProCal updates 38 scalars on MS-ResNet-20, compared with 0.7K–272.7K trainable parameters for the evaluated baselines in Table 1. Calibration weights are computed from membrane responses and class predictions in the same stochastic rollout, without augmented target views or an auxiliary network. Each target batch requires two forward passes and one backward pass, with the noise-free prediction pass preserving the original network topology.

5 Analysis

Variant	Params.	C10dvs-C	DVS128G-C
Source model	0	39.04	46.32
Deterministic boundary (μ)	19	52.80	63.30
Fixed-scale probing (μ, fixed σ)	19	53.12	63.61
Learned probing (μ, ρ)	38	53.46	63.94
+ confidence only	38	53.61	64.08
+ boundary only	38	53.68	64.17
ProCal	38	**53.82**	**64.72**

Table 3: Component ablation of ProCal. The deterministic, fixed-scale, and learned-probing variants use unweighted entropy minimization. The next two variants add confidence-only or boundary-only weighting to learned probing, while full ProCal combines both signals.

5.1 What Does Stochastic Boundary Probing Add?

Table 3 decomposes the gains over deterministic threshold adaptation into three steps: introducing stochastic probing at the same parameter count, learning its scale, and weighting samples by confidence and boundary responsiveness. The two single-weight variants further examine how the signals contribute to the full weighting rule. Appendix C compares alternative adaptation configurations under small parameter budgets.

Probing at a fixed parameter count. Deterministic boundary adaptation and fixed-scale probing both update only the 19 boundary locations. Adding perturbations with a fixed $\sigma = 0.10$ raises mean accuracy by 0.32 and 0.31 points on CIFAR10dvs-C and DVS128Gesture-C, respectively. This comparison isolates threshold randomization at an unchanged trainable parameter count.

Learning the probing range. Optimizing one scale parameter $\rho^{(l)}$ per layer brings the total to 38 trainable scalars and adds 0.34 and 0.33 percentage points over fixed-scale probing. In these experiments, adapting the probing range therefore improves on using the same fixed scale throughout the target stream.

Combining the weighting signals. Both single-weight variants build on learned probing: confidence-only weighting uses $r(x) = c(x)^{\gamma_c}$, boundary-only weighting uses $r(x) = b(x)^{\gamma_b}$, and

full ProCal uses their product. Combining the signals adds 0.36 and 0.78 points over unweighted learned probing, and 0.14 and 0.55 points over the stronger single-weight variant, respectively. The full weighting rule thus achieves the highest mean on both benchmarks. These comparisons assess the combined effects of relative sample weighting and changes in gradient magnitude, since the weights are unnormalized.

5.2 Firing-Distribution Calibration

On CIFAR10dvs-C, ProCal reduces discrepancies from a clean reference in firing rate, membrane-to-threshold margins, and temporal spike variability under both DropTime and UniformNoise. The margin-distribution distance W_1^{margin}, for example, decreases by 16.9% and 16.1%, respectively. These post-hoc observations connect threshold adaptation to changes in both membrane and spike statistics, consistent with partial calibration of corruption-induced neuronal activity. The clean reference is used only for analysis. Appendix D provides the complete diagnostics.

5.3 Recurring Mixed-Domain Shifts

We further evaluate DVS128Gesture-C using a fixed sequence of five corruptions repeated for three cycles, retaining adaptation state across corruption boundaries. ProCal reaches 69.90% mixed-stream accuracy, 0.75 points above EATA, the strongest evaluated baseline in this protocol. This diagnostic extends the per-corruption comparison to adaptation carried across recurring domains. Appendix E provides the protocol and full results.

6 Limitations

The evaluation covers MS-ResNet-20, two event-corruption suites, and two controlled frame-based variants; generalization to other architectures and shifts remains untested. The main protocol resets adaptation at known corruption boundaries, while the recurring-stream diagnostic uses a fixed sequence. Performance varies by setting, including degradation relative to Source on DropArea and lower mean accuracy than SAR on ImageNet1K-C-T. Moreover, TM is not evaluated under a common checkpoint because its reported pipeline uses adaptation-specific source training.

Boundary responsiveness summarizes local firing variability under the chosen perturbation scale; it does not measure changes in the complete network prediction or guarantee beneficial updates. Confidence can remain high for incorrect predictions under shift, and the firing-statistic diagnostics do not establish a causal explanation for accuracy gains. Finally, parameter efficiency does not establish computational efficiency: ProCal uses current-batch normalization statistics and two forward passes plus one backward pass per batch; latency, memory, energy, and neuromorphic-hardware efficiency remain unmeasured.

7 Conclusion

ProCal uses firing thresholds as both adaptation parameters and probes of local neuronal responsiveness. It combines this responsiveness with prediction confidence to weight entropy-based calibration, updates only 38 scalars on MS-ResNet-20, and predicts with noise-free thresholds after each update. Across the evaluated settings, it achieves the highest mean accuracy on three of four benchmarks and improves on deterministic threshold adaptation on both event-corruption suites. These results support using the response to perturbations of the adapted neuronal parameters to guide sample weighting in SNN test-time adaptation.

AI use statement

Generative AI tools were used to assist with language polishing, manuscript organization, and improving presentation clarity. All research ideas, methodological development, experimental design, results, and scientific conclusions were produced and verified by the authors.

REFERENCES

Yunhua Chen, Yingchao Mai, Ren Feng, and Jinsheng Xiao. An adaptive threshold mechanism for accurate and efficient deep spiking convolutional neural networks. *Neurocomputing*, 469: 189–197, 2022.

Jianhao Ding, Tong Bu, Zhaofei Yu, Tiejun Huang, and Jian Liu. Snn-rat: Robustness-enhanced spiking neural network through regularized adversarial training. *Advances in Neural Information Processing Systems*, 35:24780–24793, 2022.

Jianhao Ding, Zhiyu Pan, Yujia Liu, Zhaofei Yu, and Tiejun Huang. Robust stable spiking neural networks. *arXiv preprint arXiv:2405.20694*, 2024.

Wei Fang, Zhaofei Yu, Yanqi Chen, Timothée Masquelier, Tiejun Huang, and Yonghong Tian. Incorporating learnable membrane time constant to enhance learning of spiking neural networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 2661–2671, 2021.

Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J Davison, Jörg Conradt, Kostas Daniilidis, et al. Event-based vision: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(1):154–180, 2020.

Wulfram Gerstner, Werner M Kistler, Richard Naud, and Liam Paninski. *Neuronal dynamics: From single neurons to networks and models of cognition*. Cambridge University Press, 2014.

Yufei Guo, Xinyi Tong, Yuanpei Chen, Liwen Zhang, Xiaode Liu, Zhe Ma, and Xuhui Huang. Recdis-snn: Rectifying membrane potential distribution for directly training spiking neural networks. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 326–335. IEEE, 2022.

Julia Gygax and Friedemann Zenke. Elucidating the theoretical underpinnings of surrogate gradient learning in spiking neural networks. *arXiv preprint arXiv:2404.14964*, 2024.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

Yifan Hu, Lei Deng, Yujie Wu, Man Yao, and Guoqi Li. Advancing spiking neural networks toward deep residual learning. *IEEE transactions on neural networks and learning systems*, 36(2):2353–2367, 2024.

Junho Kim, Inwoo Hwang, and Young Min Kim. Ev-tta: Test-time adaptation for event-based object recognition. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17724–17733. IEEE, 2022.

Ryota Kobayashi, Yasuhiro Tsubo, and Shigeru Shinomoto. Made-to-order spiking neuron model equipped with a multi-timescale adaptive threshold. *Frontiers in computational neuroscience*, 3: 762, 2009.

Xinyu Luo, Kecheng Chen, Pao-Sheng Sun, Chris Xing Tian, Arindam Basu, and Haoliang Li. Space: Spike-aware consistency enhancement for test-time adaptation in spiking neural networks. *Advances in Neural Information Processing Systems*, 38:43160–43186, 2026.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In *International conference on machine learning*, pp. 16888–16905. PMLR, 2022.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Zhiquan Wen, Yaofo Chen, Peilin Zhao, and Mingkui Tan. Towards stable test-time adaptation in dynamic wild world. *arXiv preprint arXiv:2302.12400*, 2023.

Michael Pfeiffer and Thomas Pfeil. Deep learning with spiking neurons: Opportunities and challenges. *Frontiers in neuroscience*, 12:409662, 2018.

Kaushik Roy, Akhilesh Jaiswal, and Priyadarshini Panda. Towards spike-based machine intelligence with neuromorphic computing. *Nature*, 575(7784):607–617, 2019.

Guangzhi Tang. Test-time adaptation of spiking neural networks for intracortical neural decoding using membrane potential alignment. *arXiv preprint arXiv:2606.14866*, 2026.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2020.

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.

Anguo Zhang, Jieming Shi, Junyi Wu, Yongcheng Zhou, and Wei Yu. Low latency and sparse computing spiking neural networks with self-driven adaptive threshold plasticity. *IEEE Transactions on Neural Networks and Learning Systems*, 35(12):17177–17188, 2023.

Shao-Qun Zhang, Jia-Yi Chen, Jin-Hui Wu, Gao Zhang, Huan Xiong, Bin Gu, and Zhi-Hua Zhou. On the intrinsic structures of spiking neural networks. *Journal of Machine Learning Research*, 25 (194):1–74, 2024.

Kejie Zhao, Wenjia Hua, Aiersi Tuerhong, Luziwei Leng, Yuxin Ma, and Qinghai Guo. Threshold modulation for online test-time adaptation of spiking neural networks. *arXiv preprint arXiv:2505.05375*, 2025.

Yaoyu Zhu, Jianhao Ding, Tiejun Huang, Xiaodong Xie, and Zhaofei Yu. Online stabilization of spiking neural networks. In *International Conference on Learning Representations*, volume 2024, pp. 53389–53408, 2024.

A Detailed Boundary Probing Analysis

We analyze the local firing-state changes induced by Gaussian threshold perturbations and connect them to the boundary-responsiveness score.

Deterministic boundary perturbations. To isolate the local effect of a boundary change, hold H_t fixed and perturb the threshold by δ:

$$S_t(\delta) = \Theta\big(H_t - (v_{\mathrm{th}} + \delta)\big) = \Theta(m_t - \delta). \tag{12}$$

If $|\delta| < |m_t|$, the sign of the margin—and therefore the firing state—remains unchanged. A flip becomes possible only when the perturbed threshold crosses H_t, making responses near the boundary more sensitive at a fixed perturbation scale. This local characterization holds the membrane potential fixed; a resulting spike change may subsequently alter temporal dynamics and downstream layers.

Local firing-state changes. For a fixed pre-spike membrane potential H, consider the deterministic state $S_{\mathrm{det}} = \Theta(H - \mu)$ and the perturbed state $S_\epsilon = \Theta(H - \mu - \sigma\epsilon)$, where $\epsilon \sim \mathcal{N}(0, 1)$ and $\sigma > 0$. Define the normalized distance to the boundary as

$$d = \frac{|H - \mu|}{\sigma}. \tag{13}$$

If $H > \mu$, probing suppresses a deterministic spike when $\epsilon > (H - \mu)/\sigma$. If $H < \mu$, probing induces a spike when $\epsilon < (H - \mu)/\sigma$. Gaussian symmetry therefore gives

$$P_{\mathrm{flip}}(d) = \mathbb{P}_\epsilon\left(S_\epsilon \neq S_{\mathrm{det}}\right) = \Phi(-d). \tag{14}$$

At $H = \mu$, the flip probability is $1/2$ under either deterministic tie convention. It decreases with distance because $\mathrm{d}P_{\mathrm{flip}}/\mathrm{d}d = -\phi(d) < 0$, where ϕ is the standard Gaussian density. For any fixed nonzero margin, increasing σ raises the flip probability; the scale therefore controls how far threshold randomization reaches from the boundary.

Boundary responsiveness and switching probability. The local firing probability in Eq. (6) is $q = \Phi((H - \mu)/\sigma)$, so the local switching probability is $P_{\mathrm{flip}} = \min(q, 1 - q)$. Consequently, the local boundary-responsiveness score in Eq. (8) satisfies

$$b = 4q(1 - q) = 4P_{\mathrm{flip}}\left(1 - P_{\mathrm{flip}}\right). \tag{15}$$

Because $P_{\mathrm{flip}} \in [0, 1/2]$, this mapping is monotonic, with derivative $4(1 - 2P_{\mathrm{flip}}) \geq 0$. The score preserves the ordering of local switching probabilities while mapping them to $[0, 1]$: it equals one at the boundary and approaches zero as d grows. Equivalently, two independent thresholds evaluated at the same H produce different firing states with probability $2q(1 - q) = b/2$. Thus, b measures normalized local boundary responsiveness under the current probing distribution.

Scope of the sample-level score. Averaging these local quantities as in Eq. (9) produces sample-level boundary responsiveness along the observed rollout. Because the analysis holds each membrane response fixed, it does not characterize differences between complete rollouts, whose membrane potentials may diverge after a spike change. Nor does the aggregate score establish prediction correctness or adaptation utility. Section 3.3 therefore combines boundary responsiveness with prediction confidence.

Scale-controlled smoothing interpretation. For the probabilistic response $q(h; \mu, \sigma)$, $\partial q/\partial h = \phi((h - \mu)/\sigma)/\sigma$, where ϕ is the standard Gaussian density. Hence $|\partial q/\partial h| \leq 1/(\sqrt{2\pi}\sigma)$: larger scales produce a wider, smoother transition around the boundary, while smaller scales approach the deterministic step. More generally, for an integrable scalar threshold-dependent loss $J(v)$,

$$\mathbb{E}_\epsilon[J(\mu + \sigma\epsilon)] = \int_{\mathbb{R}} J(v)\,\mathcal{N}(v; \mu, \sigma^2)\,\mathrm{d}v. \tag{16}$$

Thus, stochastic thresholding admits a neighborhood-smoothing interpretation in expectation, whereas deterministic adaptation evaluates the point $v = \mu$. ProCal uses one Monte Carlo draw per layer and timestep together with surrogate gradients, so this interpretation does not imply that each update computes the exact smoothed objective or its exact gradient.

B Implementation Details and Online Adaptation Procedure

B.1 Adaptation Settings and Online Procedure

Experimental configuration. We use MS-ResNet-20 with $L = 19$ adapted spiking layers. ProCal optimizes one boundary location $\mu^{(l)}$ and one unconstrained scale parameter $\rho^{(l)}$ per layer, for a total of $2L = 38$ trainable scalars. This count excludes optimizer moments, stored activations, and non-learned batch statistics. All synaptic weights remain frozen. We initialize the locations from the source thresholds using Eq. (4) and choose $\rho^{(l)}$ such that Eq. (5) gives $\sigma_0^{(l)} = 0.10$.

We set the simulation horizon to $T = 16$ for the neuromorphic benchmarks and $T = 6$ for the frame-based benchmarks. Batch sizes are 48, 64, and 48 for the neuromorphic, CIFAR10-C-T, and ImageNet1K-C-T settings, respectively. Each batch receives one Adam update using Eq. (11), with corresponding learning rates of 5×10^{-4}, 1×10^{-3}, and 1×10^{-4}. Unless stated otherwise, $\gamma_c = \gamma_b = 1$, $\lambda_{\mathrm{p}} = 0.01$, and $\sigma_{\min} = 10^{-6}$. Hyperparameters remain fixed across corruption types within each benchmark. Results are reported as mean $\pm$ standard deviation over three runs with different random seeds. Source-domain firing statistics are reserved for post-hoc analysis and never accessed during adaptation. All experiments run on a single NVIDIA GeForce RTX 5090.

Adapt-then-predict procedure. For each incoming mini-batch, we initialize the neuronal states to zero and perform one stochastic forward rollout. At each simulation timestep, we independently draw one Gaussian noise scalar per adapted layer, shared across all neurons and samples within that layer. The forward pass uses binary spikes, while the backward pass replaces the derivative of the Heaviside function with the triangular surrogate $g_a(z) = \max(0, 1 - |z|/a)/a$, where $z = H - v$ and $a = 0.5$. Gradients propagate to both $\mu^{(l)}$ and $\rho^{(l)}$ through the reparameterized threshold in Eq. (3), with the sampled noise held fixed during backpropagation. The spike used in the membrane-reset operation is detached. We compute the sample weights from the same rollout, detach them, and perform one Adam update using Eq. (11). The weighted entropy is averaged over the actual number of samples in the current batch, including a smaller final batch when present. We use $r(x) = c(x)^{\gamma_c} b(x)^{\gamma_b}$ directly, without normalizing the weights to sum to one or to have a batch mean of one. Synaptic weights and batch-normalization affine parameters remain frozen; batch normalization uses current-batch statistics without accumulating running estimates. After the update, we reinitialize the neuronal states and perform a second forward pass without gradient computation, using the updated locations $\mu^{(l)}$ as noise-free plug-in thresholds. Reported predictions are obtained from the temporally averaged class probabilities of this second pass. Under the main per-corruption protocol, boundary parameters and optimizer states are retained across batches within each corruption stream. Before a new corruption stream, we restore the source checkpoint, reinitialize the boundary parameters, and clear the optimizer state. Thus, each batch requires two forward passes and one backward pass.

B.2 Corruption Construction

Neuromorphic benchmarks. We apply five corruptions to the CIFAR10dvs and DVS128Gesture test streams using the parameters in Table 4. In the main tables, DropTime and DropArea abbreviate DropEventByTime and DropEventByArea, while Jitter denotes SpatialJitter. DropEventByTime removes events from temporal windows, DropEventByArea removes events from contiguous spatial regions, and DownSample rescales both time and space. SpatialJitter perturbs event coordinates with Gaussian noise, whereas UniformNoise adds uniformly distributed background events.

Temporally perturbed frame-based benchmarks. CIFAR10-C-T and ImageNet1K-C-T are controlled diagnostic variants of CIFAR10-C and ImageNet-C at severity 5. Let $x \in [0,1]^{3\times H\times W}$ be a corrupted RGB image. We replicate it over $T = 6$ timesteps and independently perturb each copy:

$$x_t = x + \eta_t, \qquad \eta_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma_{\mathrm{n}}^2 I), \qquad t = 1, \ldots, T. \tag{17}$$

We set $\sigma_{\mathrm{n}} = 0.10$ for both datasets and all corruption types. This value specifies additive noise relative to the $[0, 1]$ input scale and is unrelated to the learned threshold scales $\sigma^{(l)}$. No post-perturbation clipping is applied, so x_t may fall outside $[0, 1]$. The construction isolates timestep-wise input variation rather than modeling a specific sensor process. Without this perturbation, every timestep would

Corruption	Parameter	DVS128Gesture-C	CIFAR10dvs-C
DropEventByTime	duration_ratio	0.95	0.95
DropEventByArea	area_ratio	0.90	0.90
DownSample	time_factor	0.60	0.60
	spatial_factor	0.60	0.60
SpatialJitter	var_x	(600.0, 700.0)	(10.0, 20.0)
	var_y	(600.0, 700.0)	(10.0, 20.0)
	sigma_xy	(800.0, 900.0)	(20.0, 40.0)
UniformNoise	noisy_amount	0.70	0.70

Table 4: Corruption parameters for the two neuromorphic benchmarks.

Adapted parameters	Params.	Avg. Acc.
BN affine parameters	38	50.70
Layer-wise scalars	38	47.90
Deterministic boundaries	19	52.80
ProCal (both weights)	38	**53.82**

Table 5: Comparison of adaptation configurations on CIFAR10dvs-C. The ProCal row includes learned stochastic boundaries and both weighting signals. Deterministic boundaries use 19 trainable parameters; the other configurations use 38.

receive the same corrupted image, as in the standard repeated-input encoding of CIFAR10-C and ImageNet-C.

C Additional Ablations

Alternative adaptation parameters. Table 5 compares full ProCal with BN affine parameters and generic layer-wise scalars under a budget of 38 trainable parameters. Deterministic boundary adaptation provides a 19-parameter reference. The comparison evaluates alternative adaptation configurations under similarly small budgets.

Full ProCal performs best, followed by deterministic boundary adaptation despite its smaller parameter count. These rows represent complete adaptation configurations: because ProCal also includes confidence and boundary weighting, its advantage over the BN and scalar variants cannot be attributed solely to boundary parameterization. Table 3 separately isolates probing, scale learning, and sample weighting.

D Firing-Statistic Diagnostics

We examine how ProCal changes target-domain neuronal activity relative to a clean source reference. Table 6 reports discrepancies before and after adaptation for DropTime and UniformNoise on CIFAR10dvs-C. The reference is used only for this post-hoc analysis and is never accessed during adaptation.

We use three complementary diagnostics. The firing-rate gap Δ_{FR} compares overall spike activity, W_1^{margin} is the 1-Wasserstein distance between the margin distributions defined in Eq. (2), and Δ_{TV} compares temporal variation in spike activity. Lower values indicate greater agreement with the clean reference.

All three discrepancies decrease under both corruptions. Under DropTime, the firing-rate, margin-distribution, and temporal-variation gaps fall by 20.7%, 16.9%, and 15.4%, respectively; under UniformNoise, they fall by 18.5%, 16.1%, and 13.7%. The observed calibration therefore extends across both membrane and spike statistics in these cases.

Table 6: Firing-statistic discrepancies on CIFAR10dvs-C. Before and After refer to target responses from the source model and the adapted model, respectively, compared against the clean reference. Lower is better. Relative reduction is $100 \times (\text{Before} - \text{After})/\text{Before}$.

Corruption	Metric	Before	After	Reduction (%)
DropTime	Δ_{FR}	0.0237	**0.0188**	20.7
	W_1^{margin}	0.1436	**0.1194**	16.9
	Δ_{TV}	0.0104	**0.0088**	15.4
UniformNoise	Δ_{FR}	0.0351	**0.0286**	18.5
	W_1^{margin}	0.1849	**0.1551**	16.1
	Δ_{TV}	0.0183	**0.0158**	13.7

Method	Mixed-Stream Accuracy (%)
Source	46.60
TENT Wang et al. (2020)	68.85
EATA Niu et al. (2022)	<u>69.15</u>
CoTTA Wang et al. (2022)	68.35
SAR Niu et al. (2023)	68.25
SPACE Luo et al. (2026)	46.13
ProCal (Ours)	**69.90**

Table 7: Continual adaptation under recurring mixed-domain shifts on DVS128Gesture-C. Adaptation state is retained across domain switches. Best and second-best results are bolded and underlined, respectively.

The reduction in W_1^{margin} is particularly relevant because changing a threshold directly alters the membrane-to-threshold margin, while the resulting spikes may also affect subsequent dynamics. Its decrease indicates that adapted responses move closer to the clean reference in this threshold-relative representation. The accompanying reductions in Δ_{FR} and Δ_{TV} show corresponding changes in overall firing activity and temporal variation.

These diagnostics indicate a partial reduction in corruption-induced deviations. The alignment is an outcome rather than an explicit objective: Eq. (11) uses weighted prediction entropy and scale regularization, not source-distribution matching. The remaining gaps indicate incomplete recovery, and the statistics alone do not establish that alignment causes the accuracy gains.

E Recurring Mixed-Domain Shifts

Evaluation protocol. We construct a DVS128Gesture-C stream that cycles three times through UniformNoise, DropTime, DropArea, Jitter, and DownSample in this order. Each domain segment traverses the complete test set of the corresponding corruption using a batch size of 48, with samples shuffled independently for each traversal. The model starts from the source checkpoint, and its adaptation state is retained across all batch and domain boundaries. This setting tests the effect of carrying state from one corruption into the next over repeated cycles. Corruptions follow the definitions in Appendix B.2.

Results. Table 7 shows that ProCal achieves the highest mixed-stream accuracy among the evaluated methods, reaching 69.90% and exceeding EATA by 0.75 percentage points. This result complements the per-corruption evaluation by retaining adaptation state across recurring domains. Because the stream composition and ordering differ from the main protocol, these absolute accuracies are comparable only within the mixed-stream setting.