KFC-SNN: A KALMAN-FILTERED COMPLEX-VALUED SPIKING NEURAL NETWORK FOR ROBUST TEMPORAL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Spiking neural networks (SNNs) offer a promising approach to energy-efficient temporal processing through intrinsic memory and sparse event communication. However, disturbances can disrupt membrane dynamics and spike generation, undermining computational reliability. Biologically inspired gating mechanisms use temporal state information to improve robustness, motivating a complementary estimation-based account of how current inputs should be combined with neuronal history. Guided by the prediction-and-correction principle of Kalman filtering, we introduce a Kalman-Filtered Complex-Valued (KFC) SNN that combines Kalman-inspired predictive fusion dynamics (KF) with complex-valued spike representations (CS). The KF neuron forms a prediction from the preceding membrane state and treats the current input as an observation. The prediction and observation are adaptively weighted following the Kalman filtering principle to regulate the membrane potential, thereby reducing the influence of input disturbances. The CS mechanism introduces imaginary spikes that regulate real-valued firing through activity-dependent inhibition, while providing an additional output stream that carries temporal information. We further develop a general framework for membrane-potential regulation that describes a class of input-state integration mechanisms through a common convex optimization formulation, with KF as a quadratic instance. Experiments on SHD, SSC, and ESC-50 demonstrate improved classification accuracy and robustness to stochastic input corruptions over the evaluated spiking baselines, including gains of 5.7 percentage points in clean accuracy on ESC-50 and 13.5 percentage points under Gaussian noise on SHD.

1 INTRODUCTION

Spiking neural networks (SNNs) process temporal signals through evolving membrane states and discrete spikes (Neftci et al., 2019). Their membrane dynamics provide intrinsic memory for capturing temporal dependencies, while sparse spike communication offers the potential for energy-efficient computation on neuromorphic hardware (Yin et al., 2021). Together, temporal state retention and sparse spike-based communication enable SNNs to effectively process temporal information and can provide a degree of tolerance to input variations (Kundu et al., 2021). However, this inherent tolerance does not ensure stable membrane dynamics under disturbances. Disturbances to the input can change membrane trajectories and the timing of threshold crossings. In a leaky integrate-and-fire (LIF) neuron, even a small perturbation near threshold can alter firing, changing both the membrane reset and the signals transmitted to other neurons (Neftci et al., 2019; Ding et al., 2024; 2023). These changes can affect subsequent states and events as computation unfolds over time.

To improve the robustness of SNNs in temporal processing, existing studies regulate memory retention and input integration through learnable membrane parameters and biologically inspired mechanisms (Fang et al., 2021; Yao et al., 2022; Wang et al., 2024). Robustness-oriented approaches further employ activity-dependent conductance and adaptive leakage to suppress disturbances (Bai et al., 2026; Xu et al., 2024), while stability-based analyses characterize membrane perturbations to guide neuron design and training objectives (Ding et al., 2024). These studies establish that neuronal dynamics can improve robustness, while highlighting the need to characterize when a particular in-

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.

tegration rule suppresses disturbances (Bai et al., 2026; Xu et al., 2024; Ding et al., 2024). Biological plausibility and computational efficiency are distinct neuron-model design criteria (Izhikevich, 2004); here we study state–input integration from an estimation perspective. Meanwhile, conventional binary spikes constrain the instantaneous output of each neuron to a single binary variable, limiting the signaling degrees of freedom available for neuronal regulation and temporal information transmission (Guo et al., 2022; Shen et al., 2024; Nguyen et al., 2025). These limitations motivate both a principled formulation of state-input integration and a richer spike representation for robust temporal processing.

We instantiate these ideas in a Kalman-Filtered Complex-Valued (KFC) SNN (Figure 1), combining Kalman-inspired predictive fusion dynamics (KF) with complex-valued spike representations (CS). To provide a principled basis for neuronal regulation, the KF neuron follows the prediction-and-correction principle of Kalman filtering: it forms a prediction from the preceding membrane state and treats the current input as an observation. The prediction and observation are adaptively weighted to regulate the membrane potential, thereby reducing the influence of input disturbances. We further introduce a local learning mechanism for the predictive process without relying on a predefined state-transition model. Complementing this regulation, the CS mechanism introduces an additional imaginary spike channel that regulates the real membrane potential through activity-dependent inhibition, while expanding the representational space beyond conventional binary spikes to carry additional temporal information. More generally, we develop a framework for membrane-potential regulation that describes a class of input-state integration mechanisms through a common convex optimization formulation, with KFC as a quadratic instance.

We evaluate KFC on temporal classification tasks using SHD, SSC, and ESC-50, covering spoken digits, speech commands, and environmental sounds. Across these datasets, KFC improves classification accuracy over the evaluated spiking baselines and exhibits improved robustness under stochastic input corruptions. We additionally analyze the roles of predictive fusion and complex-valued spikes through ablation studies, and probe the cross-channel inhibitory pathway to examine the temporal information it carries.

In summary, our contributions include:

- We introduce Kalman-inspired predictive fusion dynamics (KF) that follow the principle of Kalman filtering, adaptively balancing membrane-state prediction and current observation to reduce the influence of input disturbances. A local learning mechanism is further introduced to train the predictive process.
- We introduce complex-valued spike representations (CS), where imaginary spikes regulate the real membrane potential through activity-dependent inhibition while expanding the representational space beyond conventional binary spikes to carry temporal information.
- We develop a general framework for membrane-potential regulation that describes a class of input-state integration mechanisms through a common convex optimization formulation, with KFC as a quadratic instance.

2 Related Work

Robustness in Spiking Neural Networks. Robustness in SNNs has been studied through training constraints, input encoding, and neuronal dynamics. Regularized training methods constrain perturbation amplification in SNN-RAT (Ding et al., 2022) and reduce sensitivity to adversarial perturbations through membrane-potential-distribution-driven surrogate gradient regularization in MPD-SGR (Jiang et al., 2026). Connecting training with neuronal dynamics, Robust Stable SNN analyzes membrane-potential perturbations and combines modified neurons with a training objective that penalizes these perturbations (Ding et al., 2024). Other approaches regulate how neurons receive and integrate inputs: FEEL-SNN combines frequency encoding with an evolutionary leak factor (Xu et al., 2024), while DGN uses activity-dependent conductance to selectively filter inputs and suppress noise (Bai et al., 2026). Our work develops Kalman-inspired predictive fusion dynamics (KF) that jointly adapt temporal memory and input integration, with a general optimization framework for interpreting such regulation.

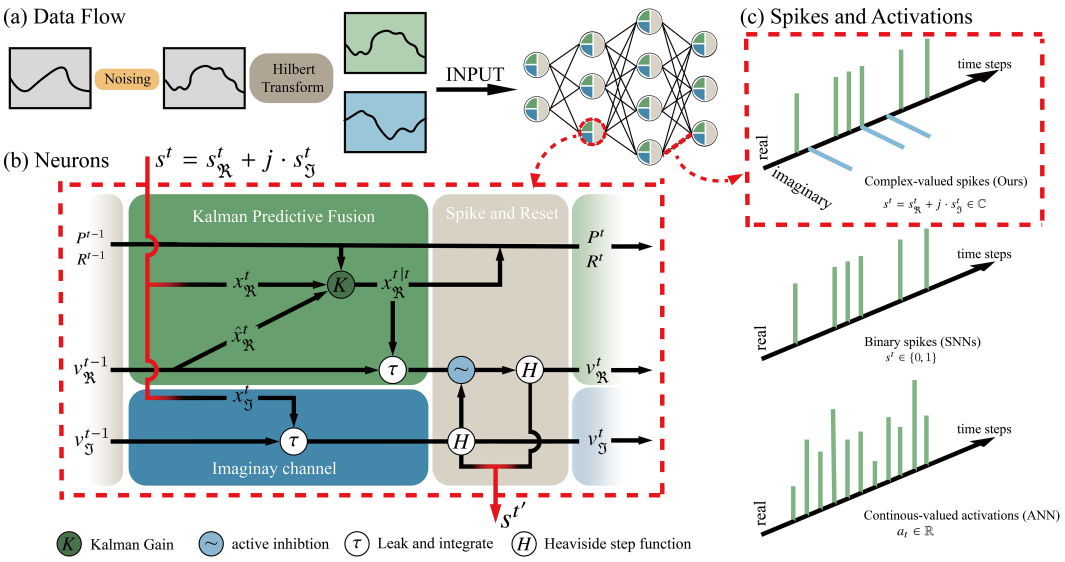


Figure 1: Overview of the experimental pipeline, KFC neuron, and signal representations. (a) Experimental pipeline. (b) KFC neuron: the horizontal direction represents time, and the vertical direction represents spike transmission. The three components are Kalman-inspired predictive fusion dynamics (KF), Imaginary channel, and Spike and Reset. (c) Signal representations: complex-valued spike representations (CS) in KFC (top), binary 0–1 spikes in conventional SNNs (middle), and continuous-valued activations in ANNs (bottom).

Neuronal Dynamics and Representations for Temporal Computation. Neuronal dynamics shape how SNNs retain and integrate temporal information. Adaptive time constants (Yin et al., 2021), bifurcation-based control of neuronal dynamics (Zhang et al., 2021), and gated integration (Yao et al., 2022) provide mechanisms for adjusting temporal processing, while TS-LIF introduces distinct dendritic and somatic frequency responses (Feng et al., 2025). Alongside these internal dynamics, spike representations determine how neuronal outputs communicate information. The Bit Budget perspective highlights spike representation as a design dimension alongside weight precision and temporal steps (Shen et al., 2024). Our work connects representation design with neuronal regulation, giving the additional representational dimension a dual role in temporal information transmission and robustness.

Local Plasticity. Biological learning motivates synaptic updates based on signals available at individual connections. In a model of cortical circuits, Payeur et al. (2021) show how burst-dependent plasticity and dendritic feedback can coordinate learning across layers. This provides a link between local synaptic signals and learning at the network level. Our predictor learning uses presynaptic states and postsynaptic prediction errors to adjust connections independently of the global task loss.

Inhibitory Regulation. Inhibition also shapes the state on which neuronal computation depends. Conductance-based analyses show how synaptic input changes membrane-voltage fluctuations and firing behavior (Richardson, 2004). Vogels et al. (2011) show that plasticity at inhibitory synapses can establish excitation–inhibition balance and support stable network activity. In computational SNNs, BackEISNN uses adaptive self-feedback and balanced excitatory and inhibitory neurons (Zhao et al., 2022). These findings motivate studying inhibition as a mechanism for regulating neuronal activity; KFC implements this role through an imaginary spike stream that acts on the membrane before firing.

3 Preliminaries

Kalman Filter. Kalman filtering combines a model-based prediction with a new observation according to their uncertainty (Kalman, 1960). It requires specified state-transition and observation

models, f_t and h_t, together with process and observation noise statistics. In a linear-Gaussian model, prediction gives a prior over the current state, and correction gives a posterior whose mean minimizes mean-squared estimation error and whose variance measures uncertainty.

Consider the scalar models $f_t(z) = a_t z$ and $h_t(z) = c_t z$, with latent state $z^t = f_t(z^{t-1}) + w_t$ and observation $x^t = h_t(z^t) + \varepsilon_t$. The independent zero-mean Gaussian noises have variances $Q_t = \mathrm{Var}(w_t)$ and $R_t > 0$. Let $\bar{v}^t$ and $\hat{v}^t$ denote prior and posterior state estimates, with error variances $P_t^- = a_t^2 P_{t-1} + Q_t$ and P_t, respectively. The state and variance updates are

$$K_{\mathrm{KF}}^t = \frac{P_t^- c_t}{c_t^2 P_t^- + R_t}, \qquad \hat{v}^t = f_t(\hat{v}^{t-1}) + K_{\mathrm{KF}}^t[x^t - h_t(f_t(\hat{v}^{t-1}))]. \tag{1}$$

$$P_t = (1 - K_{\mathrm{KF}}^t c_t)(a_t^2 P_{t-1} + Q_t), \qquad R_t = \mathrm{Var}(\varepsilon_t). \tag{2}$$

The *Kalman gain*, K_{KF}^t, weights the observation residual. For $c_t = 1$, the correction is a weighted average of prediction and observation, with $1 - K_{\mathrm{KF}}^t$ expressing trust in the prediction. The variance recursion depends on the transition model and prescribed noise statistics; R_t is supplied by the observation-noise model.

Hilbert Transform. The Hilbert transform provides a phase-shifted counterpart of a real temporal signal. For a signal $x(t)$, it is defined by

$$\mathcal{H}[x](t) = \frac{1}{\pi}\,\mathrm{p.v.}\int_{-\infty}^{\infty} \frac{x(\xi)}{t-\xi}\,\mathrm{d}\xi, \qquad z(t) = x(t) + \mathrm{i}\mathcal{H}[x](t), \tag{3}$$

where p.v. denotes the Cauchy principal value and $z(t)$ is the associated analytic signal (Gabor, 1946). For real finite-energy signals, the two components are orthogonal under the temporal inner product, while remaining deterministically linked through a quadrature phase relationship.

4 METHODOLOGY

KFC combines KF dynamics for constructing the real input drive with CS for regulating real firing and expanding the spike representation space. Figure 1 summarizes the computation. Appendix B collects the local-learning, implementation, and training-schedule details.

Omitting the layer index, we use x, s, and v for input currents, events, and membrane states, with subscripts $\Re$ and $\Im$ identifying the real and imaginary channels. At time t, v^{t^-} and v^t denote pre- and post-reset states, while $s^{t'}$ denotes the current output event. W denotes trainable weights; E, P, R, and K are computed from the dynamics, and $\alpha, \beta, \tau, \lambda, \theta, \epsilon$ are fixed hyperparameters. For clarity, state equations are written in real-valued components, with elementwise multiplication and division unless a matrix product is specified.

4.1 KALMAN-INSPIRED PREDICTIVE FUSION DYNAMICS (KF)

The measurement maps, $W_\Re$ and $W_\Im$, produce the input currents, $x_\Re^t$ and $x_\Im^t$. The prediction matrix, W_P, maps the preceding real membrane, $v_\Re^{t-1}$, to the predicted current, $\hat{x}_\Re^t$. Affine input maps absorb their biases through a constant input coordinate; the classifier bias is displayed explicitly.

$$x_\Re^t = W_\Re s_\Re^t, \qquad x_\Im^t = W_\Im s_\Im^t, \qquad \hat{x}_\Re^t = W_P v_\Re^{t-1}. \tag{4}$$

The *Kalman-inspired gain*, K^t, weights the prediction, using the complementary convention to the observation gain in Section 3. It depends on input variation, E_P^t, and prediction mismatch, E_R^t. The discrepancy function, $d(a, b)$, and smoothing coefficients, α and β, define the running estimates:

$$d(a,b) = \frac{(a-b)^2}{a^2 + b^2 + \epsilon}, \qquad \epsilon = 10^{-6}. \tag{5}$$

$$E_P^t = d(x_\Re^t, x_\Re^{t-1}), \qquad E_R^t = d(x_\Re^t, \hat{x}_\Re^t). \tag{6}$$

$$P^t = (1-\alpha)P^{t-1} + \alpha E_P^t, \qquad R^t = (1-\beta)R^{t-1} + \beta E_R^t. \tag{7}$$

The input-variation trace, P^t, and prediction-error trace, R^t, track changes in the input and disagreement with the prediction, respectively. Their relative magnitudes adapt the gain, K^t, as inputs and neuronal states evolve. In the standard Kalman filter, the variance recursion depends on the transition model and prescribed noise statistics; R_t is supplied by the observation-noise model. To support different temporal tasks without prescribing task-specific dynamics and noise statistics, we approximate uncertainty-based weighting with estimation by discrepany function in place of model-based covariance propagation. The gain and the fused current, $x_{\Re}^{t|t}$, are

$$K^t = \frac{P^t}{P^t + R^t + \epsilon}, \qquad x_{\Re}^{t|t} = K^t \odot \hat{x}_{\Re}^t + (1 - K^t) \odot x_{\Re}^t. \tag{8}$$

Learning the temporal predictor. Predictor learning combines local prediction accuracy, sparsity, and weight regularization for $W_P = [w_{ij}]$, with strengths $\lambda_0^{(\ell)}$ and λ_2 for the latter two terms. The prediction error is $e_{i,b,t}^p = x_{\Re,b,i}^t - [W_P v_{\Re,b}^{t-1}]_i$. With the smooth sparsity function, ϕ_{δ_p}, and learning rate, η_p, a basic unnormalized gradient rule is

$$\phi_{\delta_p}(w) = 1 - \exp\left(-\frac{w^2}{2\delta_p^2}\right), \tag{9}$$

$$w_{ij}^{(\ell+1)} = w_{ij}^{(\ell)} - \eta_p \frac{\partial \mathcal{J}_p^{(\ell)}}{\partial w_{ij}}, \tag{10}$$

where $\delta_p > 0$ sets the sparsity scale and the gradient is evaluated at $W_P^{(\ell)}$ for the local loss defined over valid sample–time pairs, $\mathcal{B}_\ell$, in minibatch ℓ:

$$\mathcal{J}_p^{(\ell)}(W_P) = \frac{1}{2|\mathcal{B}_\ell|} \sum_{(b,t)\in\mathcal{B}_\ell} \sum_i (e_{i,b,t}^p)^2 + \lambda_0^{(\ell)} \sum_{i,j} \phi_{\delta_p}(w_{ij}) + \frac{\lambda_2}{2} \sum_{i,j} w_{ij}^2. \tag{11}$$

Experiments use the energy-normalized implementation in Appendix B.4, whose exact objective and update are specified there.

The use of presynaptic states and postsynaptic prediction errors follows the biological motivation of learning from local signals (Payeur et al., 2021), while sparsity regularization promotes selective predictive connections. Appendix E provides predictive-matrix heatmaps and layerwise statistics, with an analysis of connection equilibria in Appendix B.3.2.

4.2 Complex-valued spike representations (CS)

The imaginary channel maintains a membrane potential, $v_{\Im}$, that follows LIF dynamics. Its spikes, $s_{\Im}^{t'}$, inhibit the real membrane and are evaluated before real spike generation. With the threshold function, $H(x) = \mathbf{1}[x > 0]$, and firing threshold, $\theta_{\Im}$, the pre-reset membrane potential, $v_{\Im}^{t^-}$, and imaginary spike are

$$v_{\Im}^{t^-} = \tau v_{\Im}^{t-1} + x_{\Im}^t, \qquad s_{\Im}^{t'} = H(v_{\Im}^{t^-} - \theta_{\Im}). \tag{12}$$

The imaginary spike controls its channel's influence on the real pre-reset membrane, $v_{\Re}^{t^-}$, through a cross-channel term that depends on both the event and its pre-reset amplitude. For $\lambda \geq 0$ and $\theta_{\Im} \geq 0$, an active imaginary spike can only lower the real membrane. With inhibition strength λ and real firing threshold $\theta_{\Re}$, the real membrane and output spike, $s_{\Re}^{t'}$, follow:

$$v_{\Re}^{t^-} = \tau v_{\Re}^{t-1} + x_{\Re}^{t|t} - \lambda s_{\Im}^{t'} \odot v_{\Im}^{t^-}, \qquad s_{\Re}^{t'} = H(v_{\Re}^{t^-} - \theta_{\Re}). \tag{13}$$

The complex-valued spike representation, $s^{t\prime}$, is $s_{\Re}^{t\prime} + \mathrm{i}s_{\Im}^{t\prime}$, with each component binary. The main model uses subtractive real reset and multiplicative imaginary reset. Subtractive reset retains the real membrane's threshold excess, whereas hard reset erases its state at a firing event. The hard real-reset variant uses $v_{\Re}^t = v_{\Re}^{t^-}(1 - s_{\Re}^{t'})$ instead of the subtractive update. The main model resets its membranes as

$$v_{\Re}^t = v_{\Re}^{t^-} - s_{\Re}^{t'} \odot \theta_{\Re}, \qquad v_{\Im}^t = v_{\Im}^{t^-}(1 - s_{\Im}^{t'}). \tag{14}$$

Input and Readout. Hilbert encoding forms the input stage of KFC-SNN, mapping each temporal observation to two orthogonal, structurally related components that initialize the real and imaginary channels. This input representation supplies the real and imaginary channels used by CS, where imaginary spikes regulate real firing and provide an additional output stream. Subsequent layers receive the preceding layer's real and imaginary spikes. The same input interface can accommodate other paired representations, with related, approximately orthogonal views offering a possible extension. Appendix B.1 specifies the encoding conventions. The readout representation, q, concatenates the temporal means of the real and imaginary spikes over sequence length T. The classifier output, $f(q)$, is computed using readout weights W_O and bias b_O:

$$q = \left[\frac{1}{T} \sum_t s_{\Re}^{t'}; \frac{1}{T} \sum_t s_{\Im}^{t'} \right], \qquad f(q) = W_O q + b_O. \tag{15}$$

4.3 A GENERAL FRAMEWORK FOR MEMBRANE-POTENTIAL REGULATION

KFC regulates the real membrane through predictive fusion and imaginary-spike inhibition. We first examine how these mechanisms enter the state dynamics, then formulate a general framework for input–state integration in which KFC's predictive fusion is a quadratic instance.

State dynamics. Substituting the predicted current into the real membrane update separates the contributions of temporal history, current observation, and imaginary activity. With the diagonal gain matrix, $D_t = \mathrm{Diag}(K^t)$,

$$v_{\Re}^{t^-} = \underbrace{(\tau I + D_t W_P) v_{\Re}^{t-1}}_{\text{Temporal history}} + \underbrace{(I - D_t) x_{\Re}^t}_{\text{Current observation}} - \underbrace{\lambda s_{\Im}^{t'} \odot v_{\Im}^{t^-}}_{\text{Imaginary inhibition}} . \tag{16}$$

The first two terms combine current input with a reference derived from the preceding state while imaginary inhibition acts as an additional correction to the fused membrane. This input–state integration motivates the membrane regulation framework below.

Let the input candidate, o, and state reference, a, lie in the same pre-threshold membrane space $\mathbb{R}^n$. A potential, $\Phi(u, a)$, specifies how the reference guides the regulated membrane, $\widehat{u}$, through a proximal update (Parikh & Boyd, 2014):

$$\widehat{u} = F(o, a) := \arg\min_u \left\{ \tfrac{1}{2} \|u - o\|^2 + \Phi(u, a) \right\}. \tag{17}$$

For each fixed a, assume that $\Phi(\cdot, a)$ is proper, closed, and convex, with $\Phi(a, a) = \min_u \Phi(u, a) = 0$. The candidates and potential parameters can depend on current input and history, but remain fixed during this update. For a separable potential, $\Phi(u, a) = \sum_i \Phi_i(u_i, a)$, the regulated membrane lies between the two candidates in each coordinate:

$$\widehat{u}_i = (1 - k_i) o_i + k_i a_i, \qquad 0 \le k_i \le 1. \tag{18}$$

This describes a general family of input–state integration rules whose weighting can vary with both input and state. For the firing rule $H(x) = \mathbf{1}[x > 0]$, the interval property also preserves any firing decision shared by the two candidates:

$$H(o_i - \theta_i) = H(a_i - \theta_i) \quad \Longrightarrow \quad H(\widehat{u}_i - \theta_i) = H(o_i - \theta_i). \tag{19}$$

Regulation can therefore change a spike event only where input and state suggest different firing decisions. This links the geometry of membrane regulation to its effect on neuronal firing.

The KF instance in KFC. For KFC, denote the current input by $m_t = x_{\Re}^t$, its prediction by $p_t = \hat{x}_{\Re}^t$, and the shared state contribution by $b_t = \tau v_{\Re}^{t-1}$. The candidates are $o_t = b_t + m_t$ and $a_t = b_t + p_t$. The membrane before imaginary inhibition, $v_{\Re,0}^{t^-} = \tau v_{\Re}^{t-1} + x_{\Re}^{t|t}$, is recovered from Equation 17 using

$$\Phi_t(u, a) = \tfrac{1}{2} \sum_i \gamma_{t,i} (u_i - a_i)^2, \qquad \gamma_{t,i} = \frac{K_i^t}{1 - K_i^t} = \frac{P_i^t}{R_i^t + \epsilon}. \tag{20}$$

The resulting weight is exactly $k_i = K_i^t$: KF is a quadratic instance of the general framework, with a learned state reference and adaptive regulation strength. Imaginary spikes provide an additional event-dependent correction, $v_{\Re}^{t^-} = v_{\Re,0}^{t^-} - \lambda s_{\Im}^{t'} \odot v_{\Im}^{t^-}$, before real firing. This formulation isolates KF from the auxiliary inhibitory pathway and paired event representation. Tables 3 and 7 provide complementary evidence on prediction quality and the task contributions of KF and inhibition. Appendices G.1–G.5 provide the framework's scope, further analysis, proof details, and its exact correspondence to KF. Appendix H gives the corresponding state-law bounds, trajectory decompositions, and carryover analysis.

5 Experiments and Ablation Studies

We evaluate KFC on spoken-digit recognition with SHD and speech-command recognition with SSC (Cramer et al., 2022), as well as environmental sound classification with ESC-50 (Piczak, 2015). The inputs contain 25, 50, and 1,000 raw time bins or frames for SHD, SSC, and ESC-50, respectively. Grouping 5, 2, and 20 adjacent bins or frames yields 5, 25, and 50 effective time steps. All models receive Hilbert-encoded inputs. For SHD, we retain the official test set and randomly partition the official training set into training and validation subsets at a 90:10 ratio. For ESC-50, we preserve the official fold assignments and rotate the held-out test fold, using the next fold cyclically for validation and the remaining three folds for training. All stochastic operations are controlled by fixed random seeds. Each configuration uses seeds 40, 41, 42, and 43, with ESC-50 evaluated on all five test folds for each seed. We average ESC-50 scores over folds within each seed and report the mean and sample standard deviation across seeds for SHD, SSC, and ESC-50. The main classification comparison covers three datasets, while robustness evaluations on SHD and ESC-50 include stochastic corruptions and adversarial perturbations. Appendix A provides further implementation and evaluation details.

5.1 Main results

Table 1 reports clean test accuracy on SHD, SSC, and ESC-50, together with test accuracy under Gaussian noise ($\sigma = 0.3$) on SHD and ESC-50. The six SNN baselines are a standard leaky integrate-and-fire (LIF) neuron and its soft-reset variant, which subtracts the firing threshold from the membrane potential after a spike (Guo et al., 2022); the Gated LIF (GLIF), which learns gates to combine membrane leakage, input integration, and spike-initiation dynamics (Yao et al., 2022); the Gated Parametric Neuron (GPN), whose input- and state-dependent gates adapt its membrane-update parameters (Wang et al., 2024); the Temporal Segment LIF (TS-LIF), which uses dendritic and somatic compartments to capture distinct frequency components (Feng et al., 2025); and the Dynamic Gated Neuron (DGN), whose activity-dependent membrane conductance regulates information flow and filters inputs (Bai et al., 2026).

We also include a four-layer long short-term memory network (LSTM) (Hochreiter & Schmidhuber, 1997) and a continuous Transformer (Vaswani et al., 2017) as ANN reference models. Appendix A provides the evaluated architectures and hyperparameter settings. All models receive Hilbert-encoded inputs, and KFC is evaluated using the corresponding network widths and depths reported for the spiking baselines. On SHD, the two-layer KFC achieves 83.7% accuracy, outperforming the other evaluated SNN baselines. On ESC-50, the four-layer KFC achieves 40.9%.

Figure 2 shows that KFC maintains high accuracy under Gaussian noise on SHD, decreasing from 83.22% on clean inputs to 80.14% at a noise standard deviation of 0.5. Additional results in Appendix C.2 show strong performance under uniform noise and impulse perturbations, together with competitive performance under dropout. Our robustness claims concern stochastic input corruptions. Adversarial results supplement this evaluation under the limited attack settings in Appendix A.3; KFC is less competitive than GPN and the Transformer under PGD on SHD.

5.2 Ablation

KF and CS. We test the contributions of KF and CS using four combinations of the two mechanisms (Table 2). Disabling KF sets $K^t = 0$. Disabling CS in this ablation jointly removes the imaginary channel's inhibition of the real membrane and its input to the readout. Each configuration

Table 1: Parameters (M) and test accuracy (%, mean ± sample SD). Gaussian Acc. denotes the test accuracy under Gaussian noise with a standard deviation of 0.3. LSTM and Transformer are included as ANN baselines for reference. Parameter counts are reported for SHD; KFC counts retain only the largest 5% of physical W_P entries, with main-table accuracies evaluated after pruning (Appendix A.2). KFC-s and KFC-l denote the smaller and larger variants of our model. Bold values indicate our models.

Model	Params (M)	Clean Acc.			Gaussian Acc.	
		SHD	SSC	ESC-50	SHD	ESC-50
LIF	2.32	76.9 ± 0.9	24.7 ± 0.6	29.8 ± 2.9	54.2 ± 3.1	25.9 ± 1.3
Soft-reset LIF	2.32	82.0 ± 1.0	28.9 ± 2.1	30.6 ± 0.7	56.0 ± 2.6	26.9 ± 0.1
GLIF	1.87	81.0 ± 1.6	29.0 ± 0.1	35.2 ± 2.1	45.7 ± 3.6	34.1 ± 0.3
GPN	7.57	83.2 ± 2.5	29.6 ± 0.5	25.7 ± 3.4	69.3 ± 1.9	25.2 ± 0.3
TS-LIF	1.86	81.8 ± 0.5	29.7 ± 0.6	18.5 ± 1.8	46.5 ± 3.0	16.2 ± 1.7
DGN	3.72	81.8 ± 2.2	28.8 ± 1.5	23.0 ± 2.7	49.0 ± 1.3	19.7 ± 2.4
KFC-s (ours)	1.94	$\mathbf{83.7 \pm 0.9}$	$\mathbf{30.4 \pm 0.9}$	$\mathbf{39.4 \pm 3.2}$	$\mathbf{73.2 \pm 0.1}$	$\mathbf{37.9 \pm 1.2}$
KFC-l (ours)	2.89	$\mathbf{83.2 \pm 1.1}$	$\mathbf{32.9 \pm 0.3}$	$\mathbf{40.9 \pm 0.6}$	$\mathbf{82.8 \pm 1.4}$	$\mathbf{39.6 \pm 0.2}$
LSTM	9.02	79.4 ± 0.6	28.6 ± 1.5	27.0 ± 0.9	75.6 ± 0.2	25.4 ± 0.1
Transformer	0.35	88.1 ± 0.9	42.0 ± 0.2	41.4 ± 2.2	54.3 ± 2.1	41.6 ± 1.7

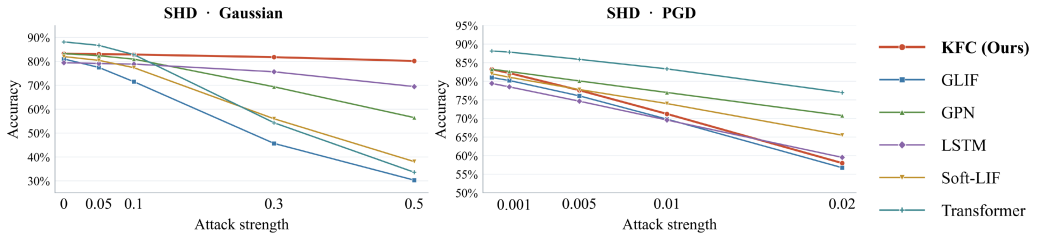


Figure 2: SHD test accuracy under Gaussian noise (left) and PGD attacks (right). Further curves appear in Appendix C.2; Appendix A.3 specifies the protocols.

is retrained, testing the mechanisms individually and together. Appendix A gives the settings, and Table 7 reports the complete four-factor comparison.

The full model achieves 81.20% corruption accuracy, compared with 80.12% without KF, 59.07% without CS, and 48.87% with both disabled. Clean accuracy falls from 83.22% to 69.35% when both are disabled. The full model also has the highest white-box mean among these four configurations, supporting the combined contribution of the two mechanisms in this comparison.

Table 2: SHD KF/CS ablation (%, mean ± sample SD). CS-off removes inhibition and imaginary readout. Noised Acc. and Attack Acc. columns use the family/strength averages in Appendix A.3.

KF	CS	Clean Acc.	Noised Acc.	Attacked Acc.
On	On	83.2 ± 1.1	81.2 ± 1.0	47.0 ± 0.9
Off	On	82.4 ± 0.9	80.1 ± 1.2	46.2 ± 1.4
On	Off	75.2 ± 1.1	59.1 ± 1.8	37.2 ± 0.5
Off	Off	69.4 ± 0.8	48.9 ± 1.3	27.4 ± 0.9

Local Predictor Learning. We compare local learning with sparsity, local learning without sparsity, task-loss learning, and a fixed predictor while retaining the same predictor structure and forward dynamics (Table 3). The comparison tests how the learning rule affects prediction fidelity and task performance. Appendix A specifies initialization, training settings, and the normalized prediction-error metric.

Local learning improves prediction fidelity, while sparsity trades some prediction accuracy for better classification performance (Table 3). Thus, improving the local prediction objective alone does not determine task performance. Further diagnostics appear in Appendix E.

Table 3: SHD predictor-learning ablation. Accuracy is in %; normalized MSE is dimensionless and lower is better. Entries are mean ± sample SD. Noised and Attacked accuracy use the nonzero-strength aggregates in Appendix A.3.

Predictor learning	Clean Acc.	Noised Acc.	Attacked Acc.	Norm. MSE
Sparse local learning	83.2 ± 1.1	81.3 ± 1.1	46.9 ± 0.9	0.38 ± 0.01
Local learning	82.3 ± 1.4	80.8 ± 1.4	45.6 ± 0.2	0.34 ± 0.04
Global learning	82.3 ± 0.6	80.7 ± 1.4	44.7 ± 1.9	10.42 ± 1.57
Fixed	82.7 ± 0.5	80.9 ± 0.3	46.2 ± 2.7	1.18 ± 0.05

Table 4: Hilbert versus repeated-input encoding on SHD (%, mean ± sample SD). Each pair retains its model, width, training budget, and readout.

	Clean Acc.		Noised Acc.	
Model	Hilbert	Repeat	Hilbert	Repeat
LIF	76.98 ± 0.91	64.06 ± 2.52	61.85 ± 2.19	47.45 ± 1.81
TS-LIF	81.78 ± 0.49	72.17 ± 2.70	58.81 ± 3.15	52.23 ± 2.45
DGN	81.83 ± 2.15	53.81 ± 5.75	59.48 ± 9.49	32.86 ± 3.78
KFC (ours)	$\mathbf{83.22 \pm 1.14}$	$\mathbf{76.82 \pm 1.67}$	$\mathbf{81.20 \pm 1.02}$	$\mathbf{57.58 \pm 1.36}$
LSTM	79.43 ± 0.56	76.73 ± 2.37	74.23 ± 1.69	61.63 ± 4.78
Transformer	88.14 ± 0.87	86.66 ± 1.32	66.18 ± 2.85	61.91 ± 2.95

Temporal Information Supplied by Hilbert Encoding. The input control replaces $x + \mathrm{i}\mathcal{H}[x]$ with $x + \mathrm{i}x$. Real-valued models correspondingly receive $[x, \mathcal{H}[x]]$ or $[x, x]$. Both versions retain the same input width and parameter count within each model. To match input information, the main experiments in Section 5 provide Hilbert encoding to all baselines; we apply this encoding ablation to the subset reported in Table 4: LIF, TS-LIF, DGN, LSTM, Transformer, and KFC.

Hilbert encoding improves KFC's clean and corruption accuracy, with a larger gain under corruption (Table 4). The baselines also benefit, showing that input encoding contributes to performance across models. The complementary Hilbert-versus-zero-input control is reported in Appendix D.

5.3 Temporal information in CS

The imaginary inhibitory pathway in CS also carries information about the temporal signal that drives it. We examine the inhibitory cross term,

$$x^t_{\mathrm{cross}} = -\lambda s^{t'}_{\Im} \odot v^{t^-}_{\Im}, \tag{21}$$

by comparing an *untrained model* with a *task-trained model*. For each frozen model, a linear decoder reconstructs synthetic temporal signals from the concatenated current-time prediction and inhibitory cross term. The signals enter through their Hilbert transforms on the imaginary input, while the real input remains constant. Training increases held-out R^2 from 0.3364 ± 0.0017 to 0.4277 ± 0.0278, making the original signal more accurately decodable from these features. Appendix F provides protocol details.

6 Conclusion

In this work, we propose a Kalman-Filtered Complex-Valued (KFC) SNN that combines Kalman-inspired predictive fusion dynamics (KF) with complex-valued spike representations (CS). The membrane-potential regulation framework places KF within a broader class of optimization-based integration rules, offering a common theoretical basis for understanding neuronal dynamics and deriving architectures from mathematical principles. CS further shows that sparse event communication can support richer representations and nonlinear temporal dynamics, with experiments revealing both regulatory and information-carrying roles for the imaginary pathway. Future work could explore mathematical priors and dynamical analysis as foundations for designing more expressive and robust SNNs, moving beyond a primary reliance on biological priors.

Reproducibility Statement

We provide the experimental code and configurations needed to reproduce our results. Appendix A details data preprocessing, dataset splits, model architectures, hyperparameters, training schedules, and checkpoint selection; Appendix A.3 specifies the corruption and adversarial-attack protocols; and Appendix B describes the local-learning implementation. Classification experiments use four fixed seeds, with all five folds additionally evaluated for ESC-50. Forecasting uses seed 42, as specified in Appendix C.1.

AI Use Statement

Generative AI tools were used to assist with literature search, language editing, and improving the clarity and organization of the manuscript. They were also used to provide feedback on the presentation of the methodology and experimental design. All research ideas, methodological decisions, mathematical formulations, experimental results, and conclusions were developed, verified, and reviewed by the authors. All AI-assisted content was critically reviewed and revised by the authors, who take full responsibility for the final content of this work.

References

Lars V. Ahlfors. *Complex Analysis: An Introduction to the Theory of Analytic Functions of One Complex Variable*. McGraw-Hill, 3 edition, 1979.

Qianyi Bai, Haiteng Wang, and Qiang Yu. A brain-inspired gating mechanism unlocks robust computation in spiking neural networks. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=5h741EyfQM.

J. M. Bates and C. W. J. Granger. The combination of forecasts. *Journal of the Operational Research Society*, 20(4):451–468, 1969. doi: 10.1057/jors.1969.103.

D. H. Brandwood. A complex gradient operator and its application in adaptive array theory. *IEE Proceedings F: Communications, Radar and Signal Processing*, 130(1):11–16, 1983. doi: 10.1049/ip-f-1.1983.0003.

Gerda Claeskens, Jan R. Magnus, Andrey L. Vasnev, and Wendun Wang. The forecast combination puzzle: A simple theoretical explanation. *International Journal of Forecasting*, 32(3):754–762, 2016. doi: 10.1016/j.ijforecast.2015.12.005.

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. URL https://arxiv.org/abs/1910.07407.

Jianhao Ding, Tong Bu, Zhaofei Yu, Tiejun Huang, and Jian K. Liu. SNN-RAT: Robustness-enhanced spiking neural network through regularized adversarial training. In *Advances in Neural Information Processing Systems*, volume 35, pp. 24780–24793, 2022. doi: 10.52202/068431-1797. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/9cf904c86cc5f9ac95646c07d2cfa241-Abstract-Conference.html.

Jianhao Ding, Zhaofei Yu, Tiejun Huang, and Jian K. Liu. Spike timing reshapes robustness against attacks in spiking neural networks, 2023. URL https://arxiv.org/abs/2306.05654.

Jianhao Ding, Zhiyu Pan, Yujia Liu, Zhaofei Yu, and Tiejun Huang. Robust stable spiking neural networks. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 11016–11029. PMLR, 2024. URL https://proceedings.mlr.press/v235/ding24e.html.

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. URL https://openaccess.thecvf.com/content/ICCV2021/html/Fang_Incorporating_Learnable_Membrane_Time_Constant_To_Enhance_Learning_of_Spiking_ICCV_2021_paper.html.

Shibo Feng, Wanjin Feng, Xingyu Gao, Peilin Zhao, and Zhiqi Shen. Ts-lif: A temporal segment spiking neuron network for time series forecasting. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2503.05108.

D. Gabor. Theory of communication. part 1: The analysis of information. *Journal of the Institution of Electrical Engineers - Part III: Radio and Communication Engineering*, 93(26):429–441, 1946. doi: 10.1049/ji-3-2.1946.0074.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples, 2015. URL https://arxiv.org/abs/1412.6572.

Yufei Guo, Yuanpei Chen, Liwen Zhang, Yinglei Wang, Xiaode Liu, Xinyi Tong, Yuanyuan Ou, Xuhui Huang, and Zhe Ma. Reducing information loss for spiking neural networks. In *Computer Vision – ECCV 2022*, volume 13671 of *Lecture Notes in Computer Science*, pp. 36–52, 2022. doi: 10.1007/978-3-031-20083-0_3.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8): 1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735. URL https://www.bioinf.jku.at/publications/older/2604.pdf.

Patrik O. Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 5:1457–1469, 2004. URL https://jmlr.org/papers/v5/hoyer04a.html.

Eugene M. Izhikevich. Which model to use for cortical spiking neurons? *IEEE Transactions on Neural Networks*, 15(5):1063–1070, 2004. doi: 10.1109/TNN.2004.832719.

Runhao Jiang, Chengzhi Jiang, Rui Yan, and Huajin Tang. MPD-SGR: Robust spiking neural networks with membrane potential distribution-driven surrogate gradient regularization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(3):1864–1872, 2026. doi: 10.1609/aaai.v40i3.37166. URL https://ojs.aaai.org/index.php/AAAI/article/view/37166.

Jian Jin, Qing Qu, and Yuantao Gu. Robust zero-point attraction least mean square algorithm on near sparse system identification. *IET Signal Processing*, 7(3):210–218, 2013. doi: 10.1049/iet-spr.2012.0125.

Rudolf E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960. doi: 10.1115/1.3662552.

Souvik Kundu, Massoud Pedram, and Peter A. Beerel. HIRE-SNN: Harnessing the inherent robustness of energy-efficient deep spiking neural networks by training with crafted input noise. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5209–5218, 2021. URL https://openaccess.thecvf.com/content/ICCV2021/html/Kundu_HIRE-SNN_Harnessing_the_Inherent_Robustness_of_Energy-Efficient_Deep_Spiking_Neural_ICCV_2021_paper.html.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. URL https://arxiv.org/abs/1706.06083.

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. URL https://arxiv.org/abs/1901.09948.

Duc Anh Nguyen, Ernesto Araya, Adalbert Fono, and Gitta Kutyniok. Time to spike? understanding the representational power of spiking neural networks in discrete time. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 45954–45987. PMLR, 2025. URL https://proceedings.mlr.press/v267/nguyen25a.html.

Neal Parikh and Stephen Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1 (3):127–239, 2014. doi: 10.1561/2400000003.

Alexandre Payeur, Jordan Guerguiev, Friedemann Zenke, Blake A. Richards, and Richard Naud. Burst-dependent synaptic plasticity can coordinate learning in hierarchical circuits. *Nature Neuroscience*, 24:1010–1019, 2021. doi: 10.1038/s41593-021-00857-x.

Karol J. Piczak. ESC: Dataset for Environmental Sound Classification. In *Proceedings of the 23rd Annual ACM Conference on Multimedia*, pp. 1015–1018. ACM Press, 2015. ISBN 978-1-4503-3459-4. doi: 10.1145/2733373.2806390. URL http://dl.acm.org/citation.cfm?doid=2733373.2806390.

Magnus J. E. Richardson. Effects of synaptic conductance on the voltage distribution and firing rate of spiking neurons. *Physical Review E*, 69:051918, 2004. doi: 10.1103/PhysRevE.69.051918.

Guobin Shen, Dongcheng Zhao, Tenglong Li, Jindong Li, and Yi Zeng. Are conventional SNNs really efficient? a perspective from network quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 27538–27547, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Shen_Are_Conventional_SNNs_Really_Efficient_A_Perspective_from_Network_Quantization_CVPR_2024_paper.html.

Chiheb Trabelsi, Olexa Bilaniuk, Ying Zhang, Dmitriy Serdyuk, Sandeep Subramanian, João Felipe Santos, Soroush Mehri, Negar Rostamzadeh, Yoshua Bengio, and Christopher J Pal. Deep complex networks. In *International Conference on Learning Representations*, 2018. URL https://arxiv.org/abs/1705.09792.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pp. 5998–6008, 2017. URL https://papers.nips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

T. P. Vogels, H. Sprekeler, F. Zenke, C. Clopath, and W. Gerstner. Inhibitory plasticity balances excitation and inhibition in sensory pathways and memory networks. *Science*, 334(6062):1569–1573, 2011. doi: 10.1126/science.1211095.

Haoran Wang, Herui Zhang, Siyang Li, and Dongrui Wu. Gated parametric neuron for spike-based audio recognition. *Neurocomputing*, 609:128477, 2024. doi: 10.1016/j.neucom.2024.128477. URL https://arxiv.org/abs/2412.01087.

Mengting Xu, De Ma, Huajin Tang, Qian Zheng, and Gang Pan. FEEL-SNN: Robust spiking neural networks with frequency encoding and evolutionary leak factor. In *Advances in Neural Information Processing Systems*, volume 37, pp. 91930–91950, 2024. doi: 10.52202/079017-2917. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/a73474c359ed523e6cd3174ed29a4d56-Abstract-Conference.html.

Xingting Yao, Fanrong Li, Zitao Mo, and Jian Cheng. GLIF: A unified gated leaky integrate-and-fire neuron for spiking neural networks. In *Advances in Neural Information Processing Systems*, volume 35, pp. 32160–32171, 2022. doi: 10.52202/068431-2330. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/cfa8440d500a6a6867157dfd4eaff66e-Abstract.html.

Bojian Yin, Federico Corradi, and Sander M. Bohté. Accurate and efficient time-domain classification with adaptive spiking recurrent neural networks. *Nature Machine Intelligence*, 3:905–913, 2021. doi: 10.1038/s42256-021-00397-w.

Shao-Qun Zhang, Zhao-Yu Zhang, and Zhi-Hua Zhou. Bifurcation spiking neural network. *Journal of Machine Learning Research*, 22(253):1–21, 2021. URL http://jmlr.org/papers/v22/20-1031.html.

Dongcheng Zhao, Yi Zeng, and Yang Li. BackEISNN: A deep spiking neural network with adaptive self-feedback and balanced excitatory–inhibitory neurons. *Neural Networks*, 154:68–77, 2022. doi: 10.1016/j.neunet.2022.06.036.

A Experimental Protocol Details

A.1 Preprocessing and classification training

SHD uses the original 8,156 training examples, with a 90/10 train/validation partition sampled from its training set under the experiment seed, and the 2,264-example official test set. Seeds 40–43 control both this partition and model-training randomness. SSC uses the official training, validation, and test sets, containing 75,466, 9,981, and 20,382 examples, respectively. Seeds 40–43 control model-training randomness. SHD events are binned into 25 time bins over 700 units; five adjacent bins are flattened into each effective step before encoding, giving five steps of width 3,500. SSC events are binned into 50 time bins over 700 units; two adjacent bins form each effective step, giving 25 steps of width 1,400 before encoding.

ESC-50 preserves its five official folds and uses each fold once as the held-out test set for every training seed. For test fold f, validation uses fold $(f \bmod 5) + 1$, and training uses the remaining three folds. Five-second clips are resampled to 16 kHz and converted to 128-bin log-mel features using an FFT size of 512. Features are resized to 1000 frames and then standardized within each clip. Twenty adjacent frames form each effective time step before Hilbert encoding. Perturbations act on these real-valued features, not on the waveform.

The experiments use two width–depth settings, $[256, 256]$ and $[256, 512, 512, 256]$. Some models become extremely unstable and difficult to train as network depth increases; the two settings therefore accommodate their respective stable configurations. KFC is evaluated under both settings so that comparisons remain matched. The batch size is 1,024. Models use AdamW at learning rate 10^{-3}, no scheduler, and zero weight decay, with 200 SHD epochs, 300 SSC epochs or 300 ESC-50 epochs. Validation runs after every epoch and selects the highest-accuracy checkpoint. KFC uses $\tau = 0.5$, $\theta_{\Re} = 1$, $\theta_{\Im} = 0.2$, $\lambda = 1$, and $\alpha = \beta = 0.5$. Local learning uses $\eta = 0.01$, $\sigma = 0.4$, $\chi = 1$, $\kappa = 0.2$, $\gamma = 0$, and numerical floors 10^{-6}. The first task epoch has $W_P = 0$; clean epoch-end passes initialize and then refresh calibration as specified in Appendix B.

Because depth affects optimization stability and final performance differently across architectures, each baseline in Table 1 is reported at the better-performing configuration among the stable width–depth settings evaluated for that model. To keep the KFC comparison matched and transparent, we report both $[256, 256]$ and $[256, 512, 512, 256]$ configurations; they are alternative matched configurations rather than primary and secondary variants.

For a controlled comparison, we evaluate baseline re-implementations under the same experimental framework, using the temporal-classification architectures and hyperparameters specified below. The resulting performance may therefore differ from the originally reported results due to differences in the experimental pipeline. The primary LIF setting uses leak factor 1 and threshold 1; the separate real-input control in Appendix D uses leak factor 0.5. Soft-LIF uses soft reset with the same LIF dynamics. TS-LIF uses decay factors $(0.8, 0.2, 0.3, 0.7)$, coupling parameter 0.5, initial coefficients 0.8 and 0.1, and surrogate slope 2. DGN uses synaptic time parameter 0.02, initial leak conductance 1, threshold 1, time step 1, and initial conductance-related coefficients with mean 0.01 and spread 0.005. GLIF uses learned gating, leak, and threshold parameters; GPN uses input- and state-dependent forget, input, threshold, and bypass gates (Wang et al., 2024). LSTM uses four recurrent layers with hidden width 256. The continuous Transformer pools the input feature dimension to 128 and has two layers, model width 128, four attention heads, feed-forward width 256, GELU activations, dropout 0.1, learned positional embeddings, and a class-token readout. These are the evaluated temporal-classification configurations of the comparison models.

LSTM and the continuous Transformer are architectural references. Every model receives Hilbert-encoded observations. KFC separates the real and imaginary channels, while real-valued models concatenate them. SHD, SSC and ESC-50 use seeds 40–43. For SHD, seeds control both the training/validation partition and training randomness. For ESC-50, each seed is used with all five test folds, and the five test scores are averaged within each seed. Reported means and sample standard deviations are computed across the four seed-level scores for each dataset.

A.2 Parameter and computation measurement

We measure model size and inference computation by rebuilding each evaluated model from its validation-selected checkpoint and using the parameter shapes and forward configuration stored in that checkpoint. For a model with registered parameter set Θ, the dense parameter count is

$$N_{\text{dense}} = \sum_{p \in \Theta} \text{numel}(p). \tag{22}$$

For the KFC results in Table 1, we prune the prediction matrices $W_P^{(\ell)}$ after training, using the validation-selected checkpoint. Within each matrix, we retain the 5% of entries with the largest absolute physical weights and set the remaining entries to zero. This selection acts on $W_P^{(\ell)}$, rather than its calibrated representation $\widetilde{W}_P^{(\ell)}$, and leaves all other model parameters unchanged. The KFC accuracies in Table 1 are evaluated after this pruning step. The effective parameter count is therefore

$$N_{\text{eff}} = N_{\text{dense}} - \sum_{\ell} |W_P^{(\ell)}| + \sum_{\ell} \left\lceil 0.05 |W_P^{(\ell)}| \right\rceil. \tag{23}$$

All parameters outside W_P are counted normally. Thus, the KFC rows in Table 5 use the effective pruned parameter count rather than the dense tensor size; the corresponding counts are 1.941M and 2.887M for the two KFC configurations.

The computation analysis uses one SHD sample and its five effective time steps. We first count linear multiply–accumulate operations (MACs): each weight in a matrix multiplication contributes one MAC per time step, and the readout is counted at the point where it is applied in the implemented model. For KFC, the measurement maps remain dense while each W_P contributes only its 5% retained entries to the reported sparse-MAC proxy. The linear FLOP count is $2M_{\text{lin}}$, where M_{lin} is the total linear MAC count. The lower-bound FLOP count reported in the table is

$$F_{\text{LB}} = 2M_{\text{lin}} + F_{\text{simple}}, \tag{24}$$

where F_{simple} includes directly countable scalar arithmetic in state updates, temporal/readout reductions, residual and normalization operations, and input pooling. Nonlinear evaluations such as sigmoid, tanh, GELU, exp, rsqrt, and softmax are reported as operation counts rather than assigned an implementation-dependent FLOP cost. Sparse-indexing overhead, dense masked-weight materialization, memory traffic, and hardware-specific latency are not included; the lower-bound FLOPs therefore provide an analytical comparison rather than a wall-clock runtime estimate.

The Transformer uses the same 7,000-dimensional concatenated Hilbert input as the other real-valued comparison models, but applies adaptive average pooling from 7,000 to 128 features at every effective time step before its input projection. It then uses two encoder layers with $d_{\text{model}} = 128$, four attention heads, and feed-forward width 256. This input compression is why the Transformer has substantially fewer parameters and FLOPs in Table 5: the trainable input projection and all subsequent encoder operations operate on width 128, rather than on the original 7,000 features. The pooling arithmetic is included in F_{LB}; the pooling itself has no trainable parameters.

The hidden-layer column reports the architecture recovered from the selected checkpoint. In particular, the evaluated LSTM checkpoint contains four recurrent layers with hidden size 256, so its measured architecture is $[256, 256, 256, 256]$. The LSTM parameter and compute values in Table 5 follow the checkpoint that was actually evaluated.

For the mechanism ablation, every configuration is independently trained with Hilbert inputs and the same initialization procedure and training settings. KF-off sets $K^t = 0$. Inhibition-off sets $\lambda = 0$ while retaining imaginary integration, thresholding, reset, and propagation. Real-only readout excludes imaginary spikes from the classifier. The four-combination main table controls CS by jointly enabling or disabling inhibition and imaginary readout; Appendix D reports the complete 16-combination table.

A.3 Perturbations and aggregation

For a real input value x, Gaussian noise gives $x + \nu$ with $\nu \sim \mathcal{N}(0, a^2)$, and uniform noise uses $\nu \sim \mathcal{U}[-a, a]$. Dropout produces Mx with $M \sim \text{Bernoulli}(1 - a)$. Impulses are additive: $x + B$

Table 5: Hidden-layer configurations and analytical inference complexity for one SHD sample (five effective time steps). Parameter counts, linear MACs, and lower-bound FLOPs are in millions. KFC uses the 5% retained W_P parameters and the corresponding sparse-MAC proxy. The Transformer includes adaptive average pooling from 7,000 to 128 input features.

Model	Hidden layers / configuration	Params (M)	Linear MACs (M)	Lower-bound FLOPs (M)
LIF	$[256, 512, 512, 256]$	2.323	11.607	23.253
Soft-reset LIF	$[256, 512, 512, 256]$	2.323	11.607	23.253
TS-LIF	$[256, 256]$	1.864	9.313	18.675
DGN	$[256, 256]$	3.721	18.601	37.295
GLIF	$[256, 256]$	1.869	9.313	18.678
GPN	$[256, 512, 512, 256]$	7.572	37.821	75.712
LSTM	$[256, 256, 256, 256]$	9.016	45.041	90.102
Transformer	2 layers, $d_{\text{model}} = 128$, input pool $7000 \to 128$	0.350	1.676	3.414
KFC (ours)	$[256, 256]$; 5% retained W_P	1.941	9.658	19.416
KFC (ours)	$[256, 512, 512, 256]$; 5% retained W_P	2.887	14.377	29.047

with $B \sim \text{Bernoulli}(a)$, so a selected value receives an increment of one. The common nonzero strength grid is $a \in \{0.05, 0.10, 0.30, 0.50\}$. SHD perturbed inputs are clipped below at zero; standardized ESC-50 features are not clipped. Hilbert encoding follows the perturbation.

For the white-box adversarial evaluations, we use attack budgets $\epsilon \in \{0, 0.001, 0.005, 0.01, 0.02\}$, where $\epsilon = 0$ is the clean reference. FGSM (Goodfellow et al., 2015) takes one sign-gradient step of size ϵ. PGD (Madry et al., 2018) starts uniformly in the ℓ_∞ ball of radius ϵ, takes four sign-gradient steps of size $\epsilon/4$, and projects back into that ball after each step. Both attacks act on the dense real input before Hilbert encoding. SHD also applies the nonnegative input constraint. White-box attacks optimize maximum entropy and attack each model separately through its encoding and implemented surrogate backward pass. KFC's discrepancy calculations retain stop-gradient during attack generation, as they do during training. Each PGD setting has one random start. These adversarial evaluations are supplementary results for the specified objectives, gradient paths, budgets, and iteration counts, rather than evidence of general adversarial robustness.

For random corruptions, three realizations are averaged at each strength before averaging nonzero strengths and families within a checkpoint. FGSM and PGD each have one evaluation per strength. The two attack families are averaged separately from the four stochastic families. Zero-strength rows are clean references and do not enter either perturbed mean. For ESC-50, we further average these checkpoint-level scores over the five test folds within each training seed. We then report the mean and sample standard deviation over training seeds for each dataset. SHD primary and readout ablations use the model seed to generate matched perturbation draws; ESC-50 uses evaluation seed 42 across checkpoints. The predictor-learning comparison uses its own paired realization stream, so its full-model perturbed means can differ slightly from the grouped tables even though the selected checkpoints are the same.

A.4 Predictor-learning controls and coefficient statistics

All four predictor-learning conditions share the dense forward predictor and the first task-only epoch with subsequent calibrated initialization. The no-attraction condition sets $\kappa = 0$ and retains local error updates. The task-gradient condition also sets $\kappa = 0$, disables local updates, and adds W_P to the task optimizer after initialization. The fixed condition disables both local and task updates to W_P after initialization. Scale refreshes continue in every condition without directly changing physical W_P. Every condition selects its own checkpoint by validation accuracy. The predictor-learning comparison trains for 200 epochs at seeds 40–43 and re-evaluates the full-model checkpoints using its own matched perturbation realizations.

We evaluate prediction error on 2,264 clean test examples, excluding the first effective time step. For each layer, normalized prediction MSE is $\sum_{b,t>1} \|x_{\Re}^t - W_P v_{\Re}^{t-1}\|^2 / \sum_{b,t>1} \|x_{\Re}^t\|^2$, so a zero predictor has value one. Each layer therefore contributes 9,056 valid sample-time pairs. Layerwise normalized MSE uses the current-input energy as denominator. Table 3 averages the four layer ratios within a seed before computing the mean and sample SD across seeds.

Sparsity diagnostics use calibrated coefficients $\widetilde{W}_P$. For a width-m matrix, Hoyer sparsity (Hoyer, 2004) is

$$\frac{m - \|\operatorname{vec}(\widetilde{W}_P)\|_1 / \|\operatorname{vec}(\widetilde{W}_P)\|_2}{m-1}. \tag{25}$$

The small-coefficient fraction is the proportion satisfying $|\widetilde{W}_{P,ij}| \le \delta$, where $\delta = 0.4/\sqrt{m}$ in all four learning conditions. This threshold equals 0.025 at width 256 and approximately 0.01768 at width 512. It measures amplitude concentration, not an exact-zero proportion or a pruning operation. Top-5% energy is the fraction of squared coefficient energy carried by the largest 5% of absolute entries within a layer. The statistics in Appendix E.2 use the same validation-selected checkpoints as their accuracy measurements.

B Local Learning and Implementation Details

This appendix gives the theoretical local learning rule, its relation to the calibrated experimental implementation, and the representation and training conventions.

B.1 Hilbert encoding conventions

The paired neuron receives a structured second view of the input through Hilbert encoding. For a real temporal sequence x, define

$$s_{\Re} = x, \qquad s_{\Im} = \mathcal{H}[x], \qquad s = x + \mathrm{i}\mathcal{H}[x]. \tag{26}$$

The transform acts along the temporal axis independently for each input feature. In frequency coordinates, its multiplier is $-\mathrm{i}\operatorname{sgn}(\omega)$ for nonzero frequencies. The discrete FFT implementation sets the DC multiplier to zero and also sets the Nyquist multiplier to zero for even-length sequences. On the remaining frequency subspace, $\mathcal{H}^2[x] = -x$. This identity defines the inverse-like target examined by the probe in Section 5.3.

Hilbert encoding is a deterministic representation of the observed sequence. It supplies phase-related temporal structure, while the learned maps $W_{\Re}, W_{\Im}$, spike thresholds, and resets determine the subsequent channel dynamics. In particular, independently learned projections do not impose the cross-products of a conventional complex-linear map, and nonlinear hidden states need not remain Hilbert pairs. Their interaction is explicitly defined by Equation 13.

For classification, encoding uses the available input sequence. For forecasting, it is computed within the observed input window, before the future target horizon. An FFT Hilbert transform uses the whole encoded window, so causality of the membrane recurrence is understood relative to that window-level preprocessing. Perturbations are applied before encoding: the real and imaginary views both originate from the same perturbed observation. Real-valued comparison models can receive the two views by feature concatenation, separating the effect of the representation from the neuron dynamics.

B.2 Differentiation in paired real coordinates

Complex neural representations express a feature as $z = a + \mathrm{i}b$, making two interacting components available to the computation (Trabelsi et al., 2018). A global holomorphicity requirement would be restrictive for bounded activations: Liouville's theorem states that a bounded function holomorphic on all of $\mathbb{C}$ must be constant (Ahlfors, 1979). This concerns bounded entire functions; it does not rule out complex networks built from non-holomorphic functions or functions defined on restricted domains. Separate real and imaginary thresholds are naturally non-holomorphic.

Wirtinger calculus differentiates such real-differentiable complex computations by using z and its conjugate $\bar{z}$ as formal coordinates (Brandwood, 1983). For a real-valued loss $\mathcal{L}$,

$$\frac{\partial \mathcal{L}}{\partial z} = \frac{1}{2}\left(\frac{\partial \mathcal{L}}{\partial a} - \mathrm{i}\frac{\partial \mathcal{L}}{\partial b}\right), \qquad \frac{\partial \mathcal{L}}{\partial \bar{z}} = \frac{1}{2}\left(\frac{\partial \mathcal{L}}{\partial a} + \mathrm{i}\frac{\partial \mathcal{L}}{\partial b}\right). \tag{27}$$

Thus ordinary gradient descent on the two real coordinates corresponds to $z \leftarrow z - 2\eta\,\partial\mathcal{L}/\partial\bar{z}$, with the factor of two absorbable into the learning rate. The calculus permits non-holomorphic dependence, but does not itself differentiate a discontinuous spike threshold.

The spike surrogate used in Equation 49 addresses event generation. Applying this approximation separately to the two spike channels allows gradients to propagate through their coupling. The paired real implementation is a real-coordinate realization of the corresponding complex computation; it requires neither a holomorphic activation nor a separate complex-valued optimizer. Wirtinger calculus and surrogate gradients therefore address different obstacles: non-holomorphic representation and discontinuous event generation, respectively.

B.3 Theoretical local learning

The local learning rule combines prediction-error adaptation, continuous sparsity attraction, and quadratic weight regularization. We first specify the theoretical rule in physical weight coordinates, then relate it to the calibrated implementation used in the experiments.

B.3.1 Coordinate-normalized formulation

The predictor's role is to estimate the current real input from the preceding membrane. We give its connections a learning signal that directly expresses this role. For a layer of width m, let $W_P = [w_{ij}] \in \mathbb{R}^{m\times m}$ and initialize all entries from the same distribution,

$$w_{ij}^{(0)} \sim \mathcal{N}(0, \sigma_p^2/m). \tag{28}$$

Self-connections and lateral connections therefore begin with the same prior. Their subsequent amplitudes depend on their predictive contributions.

At training minibatch ℓ, let $\mathcal{B}_\ell$ contain the N_ℓ valid sample–time pairs with $t > 1$. Each postsynaptic neuron compares its current input with the predicted current,

$$e_{i,b,t}^p = x_{\Re,b,i}^t - [W_P v_{\Re,b}^{t-1}]_i. \tag{29}$$

The drive on connection (i, j) correlates that prediction error with its presynaptic state. Normalizing by the presynaptic energy gives

$$g_{ij}^{(\ell)} = \frac{1}{N_\ell} \sum_{(b,t)\in\mathcal{B}_\ell} e_{i,b,t}^p v_{\Re,b,j}^{t-1}, \tag{30}$$

$$d_j^{(\ell)} = \frac{1}{N_\ell} \sum_{(b,t)\in\mathcal{B}_\ell} (v_{\Re,b,j}^{t-1})^2, \qquad \text{normalized drive} = \frac{g_{ij}^{(\ell)}}{d_j^{(\ell)} + \epsilon_p}. \tag{31}$$

Here $d_j^{(\ell)}$ is a presynaptic energy statistic, distinct from the discrepancy function $d(a, b)$ in Equation 5. The positive constant ϵ_p controls normalization near zero. These quantities use the current layer's signals, without a label or a backpropagated task error.

The sparsity function in Equation 9 approximates the number of active connections. Its derivative gives the attraction,

$$r_{\delta_p}(w) = \phi'_{\delta_p}(w) = \frac{w}{\delta_p^2} \exp\left(-\frac{w^2}{2\delta_p^2}\right). \tag{32}$$

For $|w| \ll \delta_p$, $\phi_{\delta_p}(w) \approx w^2/(2\delta_p^2)$; for $|w| \gg \delta_p$, it approaches one. Summing it over W_P therefore gives a continuous measure of clearly nonzero connections. Its derivative produces approximately linear attraction near zero and decays at large amplitudes. This shape lets a small set of predictively useful connections retain substantial weights while applying stronger relative shrinkage to weak connections.

The local criterion and basic gradient rule are given in Equations 11 and 10. The experimental implementation normalizes the prediction drive, $g_{ij}^{(\ell)}$, by the presynaptic energy, $d_j^{(\ell)}$, with stabilizer $\epsilon_p > 0$:

$$w_{ij}^{(\ell+1)} = w_{ij}^{(\ell)} + \eta_p \left[\frac{g_{ij}^{(\ell)}}{d_j^{(\ell)} + \epsilon_p} - \lambda_0^{(\ell)} \phi'_{\delta_p}(w_{ij}^{(\ell)}) - \lambda_2 w_{ij}^{(\ell)}\right]. \tag{33}$$

The normalization applies to the prediction term only. This rule is motivated by the local criterion but differs from its ordinary gradient step. The normalized descent interpretation below gives its equivalent preconditioned objective for fixed captured pairs and energies.

Attraction increases gradually, allowing useful predictive connections to develop. With L_0 mini-batches of ramp-up,

$$\lambda_0^{(\ell)} = \lambda_0^{\text{final}} \min(1, \ell/L_0). \tag{34}$$

Every potential connection follows the same continuous update. Weakly supported coefficients can approach zero, and a renewed prediction drive can move them away from zero again. The method neither fixes the number of surviving connections nor assigns a special role to the diagonal.

B.3.2 Equilibria and predictive connectivity

The balance between prediction drive and continuous attraction explains why different connections can settle at different amplitudes. Once attraction has reached a constant λ_0, and in the local regime of negligible prediction drive and $|w_{ij}| \ll \delta_p$, Equation 33 becomes

$$w_{ij}^{(\ell+1)} \approx \left[1 - \eta_p \left(\frac{\lambda_0}{\delta_p^2} + \lambda_2\right)\right] w_{ij}^{(\ell)}. \tag{35}$$

The zero state is locally attracting when $0 < \eta_p(\lambda_0/\delta_p^2 + \lambda_2) < 2$. With fixed coefficients and a normalized prediction drive tending to zero, this stable local recurrence converges to zero while it remains in that regime. A sustained drive can instead balance shrinkage at a nonzero weight:

$$\frac{g_{ij}}{d_j + \epsilon_p} = \lambda_0 \frac{w_{ij}}{\delta_p^2} \exp\left(-\frac{w_{ij}^2}{2\delta_p^2}\right) + \lambda_2 w_{ij}. \tag{36}$$

This is a stationarity condition, with stability depending on the local response of the prediction drive. It explains selective amplitudes through the competition between prediction and attraction, rather than a prescribed number of retained connections.

Self-connections obey the same law as other entries. When a neuron's own previous state predicts its current input, its diagonal coefficient can remain prominent; lateral coefficients can remain when other neurons' states provide useful predictive information. We quantify this through the mean diagonal and off-diagonal magnitudes,

$$A_{\text{diag}} = \frac{1}{m}\sum_i |w_{ii}|, \qquad A_{\text{off}} = \frac{1}{m(m-1)}\sum_{i\neq j} |w_{ij}|, \qquad Q_{\text{self}} = \frac{A_{\text{diag}}}{A_{\text{off}} + \epsilon_p}. \tag{37}$$

These are diagnostics, not training constraints. Signed diagonal distributions are also relevant: the real carryover includes both τI and $\text{Diag}(K^t)W_P$, so a diagonal weight may reinforce or counteract existing leak and reset dynamics. Diagonal prominence alone cannot establish anatomical connectivity.

The four-seed comparison in Appendix E measures coefficient concentration at the same validation-selected checkpoints used for classification. Its layerwise results characterize learned amplitudes under the calibrated rule. Continuous attraction does not impose exact zeros or a fixed number of surviving connections.

B.3.3 Normalized descent interpretation

For fixed captured states and targets, let $\mathcal{L}_{\text{pred}}$ be the first term of Equation 11. Then $\partial\mathcal{L}_{\text{pred}}/\partial w_{ij} = -g_{ij}$. Define a diagonal preconditioner with entries $D_j = d_j + \epsilon_p > 0$, held fixed during one update. The weighted criterion

$$\mathcal{J}_{p,D}(W_P) = \mathcal{L}_{\text{pred}}(W_P) + \sum_{i,j} D_j \left[\lambda_0 \phi_{\delta_p}(w_{ij}) + \frac{\lambda_2}{2} w_{ij}^2\right] \tag{38}$$

obeys

$$-\frac{1}{D_j}\frac{\partial \mathcal{J}_{p,D}}{\partial w_{ij}} = \frac{g_{ij}}{d_j + \epsilon_p} - \lambda_0 r_{\delta_p}(w_{ij}) - \lambda_2 w_{ij}. \tag{39}$$

Consequently, Equation 33 is a diagonally preconditioned step on this energy-weighted criterion. Applying the same preconditioner to the unweighted criterion in Equation 11 would instead divide the attraction and decay terms by D_j as well. The distinction is necessary to identify the objective associated with the displayed update.

The local objective holds the captured trajectory fixed. It does not differentiate the current target or the presynaptic membrane through their dependence on preceding task learning. A descent statement for one captured batch therefore concerns that local criterion, not the globally supervised objective or an entire time-unrolled trajectory.

B.3.4 THE SHAPE AND STABILITY OF ATTRACTION

The derivative of the attraction is

$$r'_{\delta_p}(w) = \frac{1}{\delta_p^2} \exp\left(-\frac{w^2}{2\delta_p^2}\right)\left(1 - \frac{w^2}{\delta_p^2}\right). \tag{40}$$

The attraction magnitude is maximal at $|w| = \delta_p$ and tends to zero at large amplitudes. At zero, its absolute force is zero but its restoring slope is $1/\delta_p^2$. Thus "stronger shrinkage of small weights" describes relative shrinkage and local restoring behavior, not a nonzero force at exactly zero.

With negligible prediction drive and fixed post-ramp coefficients, linearization at zero gives Equation 35. A multiplier strictly between -1 and 1 makes zero locally stable. For time-varying coefficients, convergence requires a persistent contraction margin, not merely that every individual multiplier has magnitude below one. Minibatch fluctuations in the prediction drive can maintain a neighborhood around zero rather than exact convergence. If a drive later reappears at a zero coefficient, the next update can make that coefficient nonzero because the attraction itself vanishes at zero.

A nonzero solution of Equation 36 is only a candidate equilibrium. In a one-coordinate frozen-statistics analysis, write the update drive as $F(w) = g(w)/(d_j + \epsilon_p) - \lambda_0 r_{\delta_p}(w) - \lambda_2 w$. Local stability requires $|1 + \eta_p F'(w)| < 1$. This distinction separates the existence of a balance point from its stability, while the full learning process also changes the captured inputs and the other weights.

B.4 CORRESPONDENCE WITH THE CALIBRATED IMPLEMENTATION

The experimental implementation retains the three update terms but uses calibrated coordinates and a shared normalization energy. With per-neuron RMS scales R_x^{cal} and R_v^{cal}, its coordinates are

$$\begin{aligned} \widetilde{W}_P &= \mathrm{Diag}(R_x^{\mathrm{cal}})^{-1} W_P \,\mathrm{Diag}(R_v^{\mathrm{cal}}), \\ \widetilde{v}_{\Re}^{t-1} &= \mathrm{Diag}(R_v^{\mathrm{cal}})^{-1}\,\mathrm{sg}(v_{\Re}^{t-1}), \qquad \widetilde{x}_{\mathrm{err}}^{t} = \mathrm{Diag}(R_x^{\mathrm{cal}})^{-1}\,\mathrm{sg}(x_{\Re}^{t} - \hat{x}_{\Re}^{t}). \end{aligned} \tag{41}$$

Here sg denotes stop-gradient. For n valid sample–time pairs, the error–state correlation E_{xv} and total presynaptic energy E_v are

$$E_{xv} = \frac{1}{n}\sum_{b,t>1} \widetilde{x}_{\mathrm{err},b}^{t} (\widetilde{v}_{\Re,b}^{t-1})^{\top}, \qquad E_v = \frac{1}{n}\sum_{b,t>1} \|\widetilde{v}_{\Re,b}^{t-1}\|_2^2. \tag{42}$$

Thus the theoretical drive $g_{ij}^{(\ell)}/(d_j^{(\ell)} + \epsilon_p)$ becomes $E_{xv,ij}/(E_v + \epsilon)$ in calibrated coordinates. The same sparsity function gives

$$\Omega_\delta(\widetilde{W}_P) = \kappa\delta^2 \sum_{i,j} \phi_\delta(\widetilde{W}_{P,ij}), \qquad \delta = \frac{\chi\sigma}{\sqrt{m}}. \tag{43}$$

The coefficient correspondence is $\lambda_0 = \kappa\delta^2$, $\lambda_2 = \gamma$, and $\eta_p = \eta$ in these coordinates, so the sparsity derivative is $\kappa w \exp[-w^2/(2\delta^2)]$. Layer width m, initialization scale σ, and width factor χ determine δ. The calibrated update and conversion back to physical weights are given in Equations 44 and 45. Sparse adaptive filtering provides a related computational precedent for prediction-error adaptation with coefficient attraction (Jin et al., 2013).

The implemented update is

$$\widetilde{W}_P \leftarrow \widetilde{W}_P + \eta\left[\frac{E_{xv}}{E_v + \epsilon} - \kappa\widetilde{W}_P \odot \exp\left(-\frac{\widetilde{W}_P \odot \widetilde{W}_P}{2\delta^2}\right) - \gamma\widetilde{W}_P\right], \tag{44}$$

$$W_P \leftarrow \mathrm{Diag}(R_x^{\mathrm{cal}})\widetilde{W}_P\,\mathrm{Diag}(R_v^{\mathrm{cal}})^{-1}. \tag{45}$$

For fixed captured states, targets, scales, and energy, this implementation takes a gradient step on

$$\mathcal{L}_{\text{local}}(\widetilde{W}_P) = \frac{\sum_{b,t>1} \|\widetilde{x}^t_{\Re,b} - \widetilde{W}_P \widetilde{v}^{t-1}_{\Re,b}\|_2^2}{2n(E_v + \epsilon)} + \Omega_\delta(\widetilde{W}_P) + \frac{\gamma}{2}\|\widetilde{W}_P\|_F^2, \tag{46}$$

where $\widetilde{x}^t_{\Re,b} = \text{Diag}(R^{\text{cal}}_x)^{-1}\,\text{sg}(x^t_{\Re,b})$. The implementation also replaces the theoretical attraction ramp with one task-training warm-up epoch followed by constant attraction strength. Calibration, shared energy normalization, and this schedule distinguish the recorded implementation from the raw-coordinate theoretical rule; the two express the same three learning terms but need not produce identical weight trajectories.

B.5 Sequence initialization and training schedule

At the start of each sequence, both membranes and both discrepancy traces are zero. We set $E^1_P = 0$ because the first input has no predecessor and exclude that step from local prediction learning. Consequently, $K^1 = 0$ and the first real drive uses the observed current. Discrepancy calculations are detached during backpropagation, while both currents in the fusion retain their gradient paths. Validation, attacks, and inference keep all weights fixed.

Calibration and predictor initialization. Let $\mathcal{C}$ contain the valid causal sample-time pairs from clean training sequences. The RMS buffers are

$$R^{\text{cal}}_{x,i} = \sqrt{\frac{1}{|\mathcal{C}|}\sum_{(b,t)\in\mathcal{C}} (x^t_{\Re,b,i})^2 + \epsilon_{\text{cal}}}, \qquad R^{\text{cal}}_{v,j} = \sqrt{\frac{1}{|\mathcal{C}|}\sum_{(b,t)\in\mathcal{C}} (v^{t-1}_{\Re,b,j})^2 + \epsilon_{\text{cal}}}. \tag{47}$$

Here $\epsilon_{\text{cal}} = 10^{-6}$ is a fixed numerical floor. Calibration uses detached activations and excludes the first time step of each sequence. No labels enter these statistics. Although the update at connection (i, j) uses local error and state, the total-energy normalization uses the layer's presynaptic population, and calibration aggregates over clean training sequences. The locality is therefore layer-local and gradient-local.

The first task-training epoch uses a zero predictor and disables local updates. At its end, the first valid clean calibration initializes the matrix:

$$\widetilde{W}_{P,ij} \sim \mathcal{N}(0, \sigma^2/m), \qquad W_{P,ij} = R^{\text{cal}}_{x,i}\widetilde{W}_{P,ij}/R^{\text{cal}}_{v,j}. \tag{48}$$

After initialization, each epoch-end pass refreshes the two scales while leaving W_P unchanged. Its new dimensionless representation is recomputed from the stored matrix and the refreshed scales. Layers without usable calibration signal remain at zero until a valid calibration is available; an invalid later refresh retains the preceding valid scales.

B.6 Minibatch order and task gradients

For each training minibatch, the forward pass accumulates detached local statistics over all valid time steps and microbatches. The task loss is then differentiated through the unrolled state dynamics, and the task optimizer updates all globally optimized weights. One local update subsequently modifies each initialized W_P using its accumulated statistics. The matrix remains unchanged throughout that minibatch's forward and backward computation. This order also makes the local update independent of how an outer minibatch is partitioned into microbatches.

The forward threshold is $H(z) = \mathbf{1}[z > 0]$. Its backward approximation is

$$\frac{\partial H(z)}{\partial z} \approx \zeta\ \text{sigmoid}(\zeta z)[1 - \text{sigmoid}(\zeta z)], \qquad \zeta = 5. \tag{49}$$

The surrogate slope ζ is distinct from the trace coefficient α. Error traces use stop-gradient values. With K^t held fixed in this backward computation, the direct derivative of the fused current with respect to the preceding real membrane is $\text{Diag}(K^t)W_P$. Excluding W_P from the task optimizer therefore does not remove this path to preceding states.

C Extended Results

C.1 Forecasting

Table 6 reports test RSE and CORR after 150 epochs on four multivariate forecasting datasets. Each model occupies one row; within each dataset, lower RSE and higher CORR indicate better predictions. METR-LA and Electricity use input/prediction windows of $12 \rightarrow 6$, while PEMS-BAY and Solar-AL use $48 \rightarrow 12$. Windows are randomly sampled using seed 42 and assigned to 70% training, 20% validation, and 10% test subsets. Each complete input–target window belongs to only one subset and does not straddle split boundaries.

All forecasting runs use a batch size of 1,024, Adam with learning rate 10^{-3}, zero weight decay, deterministic data and training seeds 42, temporal block size one, normalization mode 2, time features, Hilbert input encoding, and hidden widths $[256, 256]$. The forecast input encoder is a one-channel delta encoder with $\beta = 0.99$, threshold 1, surrogate scale 2; the non-complex baselines use the same delta encoding and input processing as KFC. Encoded features are repeated when needed to match input dimensions, with cached encoding in chunks of 256. The Transformer uses two layers, model width 128, four heads, feed-forward width 256, GELU activations, and dropout 0.1. KFC uses adaptive predictive trust with $\alpha = \beta = 0.5$, local prediction learning rate 0.01, $\sigma = 0.4$, delta ratio 1, $\kappa_0 = 0.2$, numerical floors 10^{-6}, and the inverse-Hilbert linear output mode.

Table 6: Forecasting test results after 150 epochs. Each entry reports RSE/CORR.

Model	METR-LA	PEMS-BAY	Solar-AL	Electricity
LIF	0.836/0.550	0.611/0.794	0.318/0.948	0.447/0.896
TS-LIF	0.649/0.764	0.464/0.886	0.312/0.954	0.391/0.921
DGN	0.957/0.289	0.942/0.344	0.314/0.952	0.831/0.565
LSTM	0.638/0.777	0.448/0.895	0.309/0.954	0.362/0.932
Transformer	0.628/0.783	0.462/0.887	0.284/0.963	0.401/0.916
KFC (ours)	0.681/0.735	0.460/0.889	0.316/0.949	0.401/0.918

C.2 Corruption and adversarial-attack curves

Figures 3 and 4 report the SHD and ESC-50 perturbation curves. Appendix A.3 defines the strengths and aggregation rules.

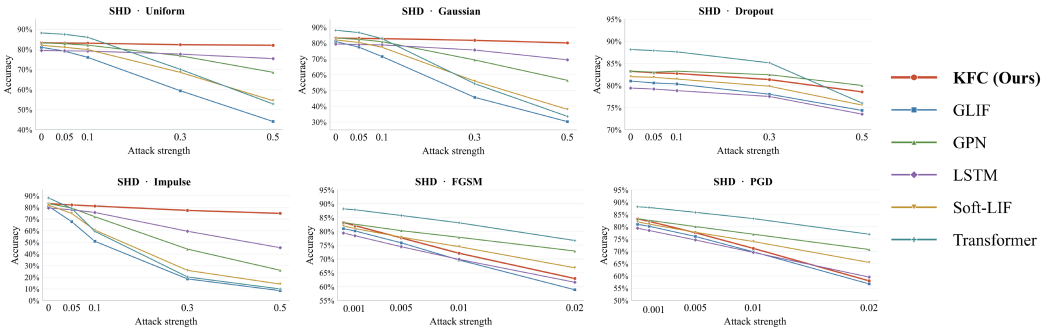


Figure 3: SHD corruption and adversarial-attack curves complementing Figure 2. Each panel shows mean test accuracy as the perturbation strength increases.

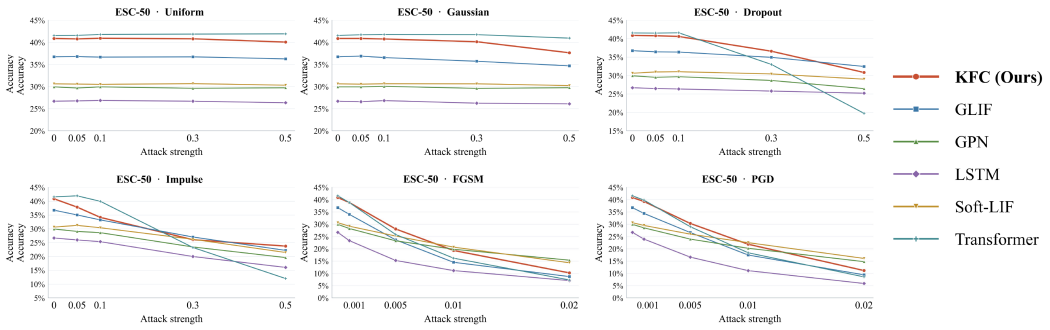


Figure 4: ESC-50 corruption and adversarial-attack curves complementing Figure 2. Each panel shows mean test accuracy as the perturbation strength increases.

D Full Encoding, KF, Inhibition, and Readout Ablation

Table 7 reports all 16 combinations of Hilbert encoding, KF, cross inhibition, and imaginary readout. $H = 1$ supplies $x+\mathrm{i}\mathcal{H}[x]$ and $H = 0$ supplies $x+\mathrm{i}0$. F controls KF: $F = 1$ uses adaptive prediction trust, whereas $F = 0$ restores measurement-only drive. $C = 1$ retains the inhibitory term and $C = 0$ sets its scale to zero. $O = 1$ lets the classifier read both spike streams; $O = 0$ restricts it to real spikes. Every combination is trained at all four seeds using the same 200-epoch budget. Shared configurations reuse the same checkpoint and evaluation rather than adding another independent run.

Table 7: Complete four-factor SHD ablation (%, mean ± sample SD over four runs). H, F, C, and O enable Hilbert input, KF, cross inhibition, and imaginary readout. Both perturbation aggregates exclude zero strength.

H	F	C	O	Clean Acc.	Noised ACc.	Attack Acc.
0	0	0	0	69.35 ± 0.83	48.87 ± 1.34	27.39 ± 0.93
0	0	0	1	69.46 ± 1.57	48.53 ± 0.64	26.75 ± 1.47
0	0	1	0	69.35 ± 0.83	48.87 ± 1.34	27.39 ± 0.93
0	0	1	1	69.30 ± 1.66	48.47 ± 0.65	26.71 ± 1.47
0	1	0	0	75.21 ± 1.12	59.07 ± 1.83	37.22 ± 0.54
0	1	0	1	75.11 ± 1.53	59.54 ± 1.81	37.47 ± 0.76
0	1	1	0	75.21 ± 1.12	59.07 ± 1.83	37.22 ± 0.54
0	1	1	1	75.11 ± 1.53	59.54 ± 1.81	37.47 ± 0.76
1	0	0	0	69.35 ± 0.83	48.87 ± 1.34	27.39 ± 0.93
1	0	0	1	80.42 ± 0.51	70.87 ± 2.29	41.95 ± 0.89
1	0	1	0	82.24 ± 1.01	74.24 ± 2.00	47.98 ± 1.21
1	0	1	1	82.38 ± 0.85	80.12 ± 1.16	46.18 ± 1.41
1	1	0	0	75.21 ± 1.12	59.07 ± 1.83	37.22 ± 0.54
1	1	0	1	81.67 ± 0.60	76.91 ± 0.76	46.65 ± 1.32
1	1	1	0	82.52 ± 0.49	78.79 ± 0.58	47.05 ± 2.03
1	1	1	1	83.22 ± 1.14	81.20 ± 1.02	46.97 ± 0.86

For a separate real-input reference, an LIF MLP with leak factor 0.5 and no input concatenation obtains $74.55 \pm 3.40\%$ clean, $50.67 \pm 4.34\%$ noised, and $24.74 \pm 2.74\%$ attacked accuracy. With both mechanisms off and a real-only readout, KFC gives 69.35%, 48.87%, and 27.39%, respectively. The two models use different real reset laws and initialization procedures, so this comparison does not isolate reset choice. Within KFC, the same double-disabled real-readout condition has identical measured losses and accuracies for Hilbert and zero-imaginary input across the entire suite, as expected when the imaginary channel is disconnected from the output.

Table 8: Layerwise statistics of calibrated predictive coefficients at the final SHD checkpoint. Energy shares are percentages; diagonal/off-diagonal compares mean absolute coefficients.

Metric	Layer 1	Layer 2	Layer 3	Layer 4
Width	256	512	512	256
Hoyer sparsity	0.691	0.745	0.744	0.626
Top-1% energy	60.80	67.86	65.20	77.86
Top-5% energy	87.61	94.43	96.53	91.35
Top-10% energy	96.42	99.74	99.69	93.29
Diagonal/off-diagonal	0.338	0.313	0.334	1.858

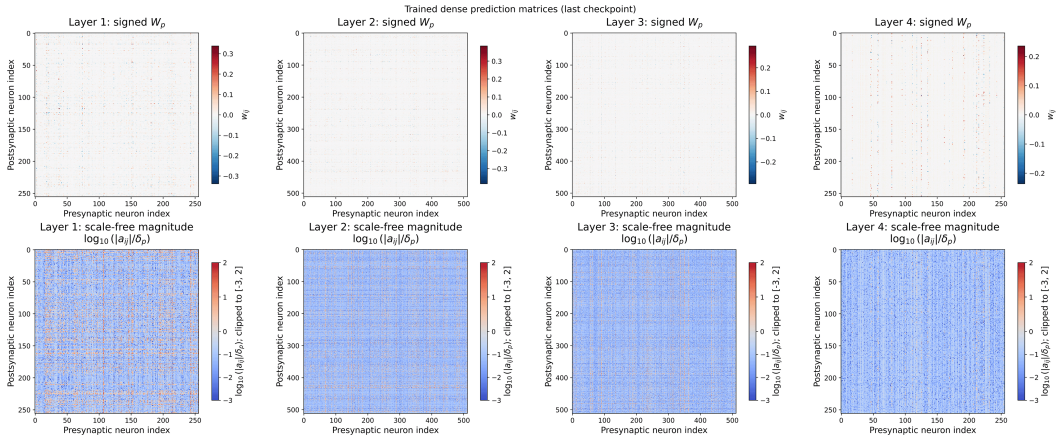


Figure 5: SHD predictive-matrix heatmaps at the final checkpoint corresponding to Table 8. The upper row shows signed physical coefficients and the lower row shows calibrated coefficient magnitudes on a logarithmic scale.

E Layerwise Prediction and Coefficient Concentration

E.1 Connections at the final checkpoint

Figure 5 and Table 8 report the final checkpoint after 200 SHD training epochs, with layer widths 256, 512, 512, and 256. Matrix rows index prediction-output neurons and columns index preceding-state neurons. The displayed matrices use physical weights, W_P; concentration statistics use calibrated coefficients, $a_{ij} = \widetilde{W}_{P,ij} = W_{P,ij} R^{\mathrm{cal}}_{v,j} / R^{\mathrm{cal}}_{x,i}$, as in Appendix B. For the flattened coefficient vector a with N entries, Hoyer sparsity is $(\sqrt{N} - \|a\|_1/\|a\|_2)/(\sqrt{N} - 1)$. The top-fraction energy share is the sum of squared magnitudes in the largest selected fraction divided by $\|a\|_2^2$. These values describe one final checkpoint; Table 9 reports the separate four-run comparison at validation-selected checkpoints. A small subset of coefficients carries most of the predictive weight energy: the largest 5% account for 87.61–96.53% across layers.

E.2 Prediction and concentration across learning conditions

Table 9 reports every predictor layer for all four learning conditions. Sparse local learning reduces normalized current-prediction MSE below one in each layer; task-gradient training produces ratios above one despite similar classification accuracy. Removing attraction further improves the mean prediction error in layers 2–4. The local prediction objective therefore controls a specific property of the pathway, while the sparsity term changes its distribution of amplitudes.

The fourth layer has a different concentration pattern from layers 1–3. In the full model, 99.26% of its calibrated coefficients lie within $[-\delta, \delta]$, yet its Hoyer sparsity is 0.415 and its top-5% energy share is 64.40%, both lower than in the preceding layers. A high small-coefficient fraction can thus

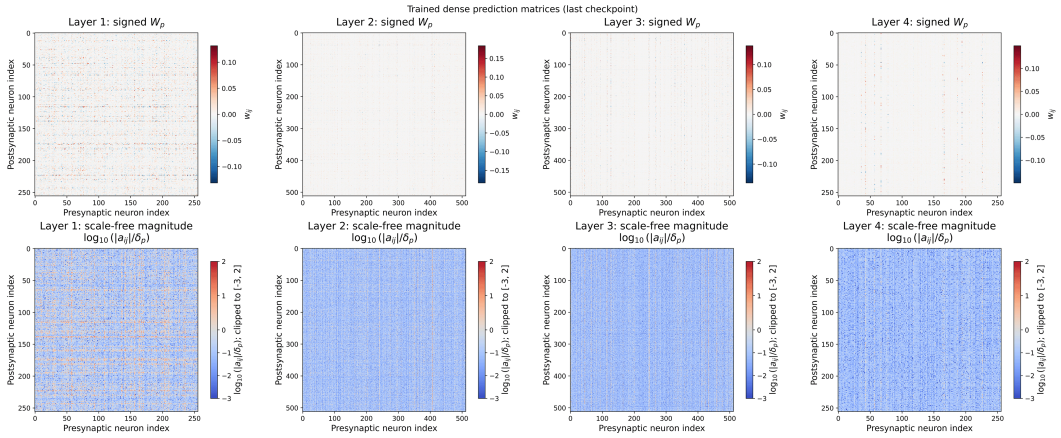


Figure 6: ESC-50 predictive-matrix heatmaps at the final checkpoint. The upper row shows signed physical coefficients and the lower row shows calibrated coefficient magnitudes on a logarithmic scale.

reflect an overall small amplitude scale without the same concentration into a few large coefficients. Reporting all layers keeps this distinction visible.

Table 9: Layerwise diagnostics at validation-selected SHD checkpoints (mean ± sample SD over four runs). Normalized MSE and Hoyer sparsity are dimensionless. The small-coefficient and top-5% energy columns are percentages; the former uses $|\widetilde{W}_{P,ij}| \leq \delta$, without pruning.

Learning	Layer	Norm. MSE	Small coeff. (%)	Hoyer	Top-5% energy (%)
Sparse local	1	0.560 ± 0.031	81.86 ± 1.95	0.670 ± 0.014	86.88 ± 1.29
	2	0.163 ± 0.016	88.29 ± 1.15	0.715 ± 0.020	92.23 ± 0.98
	3	0.208 ± 0.034	91.72 ± 0.43	0.720 ± 0.017	94.65 ± 0.63
	4	0.580 ± 0.032	99.26 ± 0.98	0.415 ± 0.147	64.40 ± 21.35
No attraction	1	0.575 ± 0.079	63.83 ± 0.70	0.563 ± 0.030	81.61 ± 3.23
	2	0.149 ± 0.022	74.70 ± 0.87	0.594 ± 0.026	85.11 ± 2.26
	3	0.167 ± 0.039	80.40 ± 0.73	0.571 ± 0.021	82.92 ± 2.16
	4	0.479 ± 0.062	97.43 ± 2.89	0.407 ± 0.112	65.65 ± 15.69
Task gradient	1	11.874 ± 3.810	32.70 ± 3.73	0.381 ± 0.023	58.71 ± 3.62
	2	17.113 ± 5.519	50.49 ± 4.02	0.384 ± 0.025	60.27 ± 3.56
	3	8.935 ± 3.085	52.64 ± 5.48	0.370 ± 0.010	57.19 ± 1.43
	4	3.762 ± 0.519	55.40 ± 2.62	0.267 ± 0.016	41.35 ± 3.10
Fixed	1	1.209 ± 0.125	65.56 ± 0.73	0.499 ± 0.023	73.80 ± 2.71
	2	1.378 ± 0.098	75.25 ± 1.81	0.568 ± 0.035	81.94 ± 3.40
	3	1.126 ± 0.054	80.70 ± 1.11	0.558 ± 0.020	80.82 ± 2.13
	4	1.016 ± 0.023	97.43 ± 2.88	0.659 ± 0.061	90.63 ± 6.58

F Cross-Term Probe Protocol

Section 5.3 reports the restricted two-feature protocol `H-IMPLICIT-IH-STRUCT-SINE-XPRED-CROSS-01`. A one-layer width-128 KFC model receives length-25 single-frequency sinusoidal targets through their Hilbert transforms on the imaginary branch. The frequency bin is sampled uniformly from $\{1, \ldots, 10\}$, phase from $\mathcal{U}[0, 2\pi)$, and amplitude from $\mathcal{U}[0.55, 1.45]$. The real branch uses a constant SHD-derived direction. The synthetic training, validation, and test sets contain 512, 256, and 512 signals, respectively; the official SHD test set is not used for probe fitting.

For each frozen model, the ridge decoder uses the current-time concatenation `[x_pred_l0, cross_term_l0]`, with no temporal taps and ridge coefficient 0.01. Lag is selected on validation data from the 13 candidates $0, \ldots, 12$; all reported runs select lag 1. Evaluation uses the valid subspace of the discrete Hilbert round trip. The parent protocol, `H-IMPLICIT-IH-STRUCT-SINE-TRAIN-01`, instead concatenates `[x_upd_l0,`

Table 10: Prediction-plus-cross-term inverse-Hilbert-like probe on the sparse local KFC model. Entries are held-out R^2; initial values precede SHD training.

Seed	Initial	Native trained	Positive trained
40	0.3347	0.4016	0.4480
41	0.3387	0.4411	0.4598
42	0.3359	0.4079	0.4430
43	0.3365	0.4604	0.4037
Mean	0.3364	0.4277	0.4386

`cross_term_l0, v_real_pre_l0, real_spike_l0]`; Table 10 reports only the restricted two-feature protocol.

Native and positive-sign variants use cross terms $-s_y \odot v_{\mathrm{imag,pre}}$ and $+s_y \odot v_{\mathrm{imag,pre}}$, respectively, implemented with `cross_scale`$= 1$ and -1. Initialization, data, and training settings are otherwise matched. The joint-feature probe measures temporal information available along this pathway; it does not isolate the contribution of the cross term from that of the prediction feature.

The trained native and positive variants obtain mean R^2 values of 0.4277 and 0.4386, respectively.

G Membrane-Potential Regulation: Proofs and Extensions

We develop the geometry, conditional error analysis, and hard-threshold consequences of Section 4.3. The analysis compares candidate updates formed from the same actual history. A target membrane $u^\star$ defines the error of both candidates; it is an analysis reference and is not supplied to the forward computation.

G.1 Scope of the membrane-regulation framework

The framework is intended to cover a broad class of integrate-and-fire-like (IF-like) neurons whose pre-threshold dynamics regulate a membrane state using both an input and an internal state. At a general level, let o denote an input quantity and a an internal state, and let $I_1(o)$ and $I_2(a)$ map them into the same membrane space. The update can then be written as

$$\widehat{u} = F\big(I_1(o), I_2(a)\big) = \arg\min_u \left\{ \tfrac{1}{2}\|u - I_1(o)\|^2 + \Phi\big(u, I_2(a)\big) \right\}, \tag{50}$$

where, for each fixed $I_2(a)$, $\Phi(\cdot, I_2(a))$ is proper, closed, and convex and is minimized at $I_2(a)$. Under the separable assumptions used above, the update reduces coordinatewise to

$$\widehat{u}_i = (1 - k_i)\,[I_1(o)]_i + k_i\,[I_2(a)]_i, \qquad 0 \le k_i \le 1. \tag{51}$$

For example. Thus many IF-like designs can be compared through their input and state maps, their regulation potential, and their effective weight. The weight may be fixed or may depend on time, input, or internal state, as in learned gates and activity-dependent conductances (Yao et al., 2022; Bai et al., 2026). The weighted-average form in Equation 51 is a broad and useful special case; the full framework also allows coupled potentials whose updates need not preserve the interval coordinatewise.

As an illustrative example, consider a conductance-based LIF model,

$$C\dot{V} = -g_L(V - E_L) + g_E(t)(E_E - V) + g_I(t)(E_I - V). \tag{52}$$

Assume $C > 0$, $g_L > 0$, and $g_E, g_I \ge 0$. When the conductances are held fixed over a step of length $\Delta > 0$, define

$$G_t = g_L + g_E(t) + g_I(t), \qquad V_{\infty,t} = \frac{g_L E_L + g_E(t) E_E + g_I(t) E_I}{G_t}. \tag{53}$$

The exact step update is

$$V_{t+\Delta} = e^{-G_t \Delta / C} V_t + \left(1 - e^{-G_t \Delta / C}\right) V_{\infty,t}, \tag{54}$$

which has the form above with $I_1(o) = V_{\infty,t}$, $I_2(a) = V_t$, and $k_t = e^{-G_t\Delta/C}$. Here E_L, E_E, E_I are reversal potentials, while g_L, g_E, g_I determine the conductance-mediated drive (Richardson, 2004). If conductances vary substantially within a step, the correspondence requires a finer discretization or an extended state description.

G.2 Convex geometry of input–state integration

Let $\Phi : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ satisfy the assumptions of Section 4.3. For fixed a, the objective in Equation 17 is proper, closed, coercive, and strongly convex, giving a unique minimizer $F(o, a) = \mathrm{prox}_{\Phi(\cdot,a)}(o)$. Nonsmooth, asymmetric, and constraint-valued potentials are allowed. All subgradients below act on the first argument, denoted by $\partial_1\Phi$. We use standard proximal optimality and monotonicity properties (Parikh & Boyd, 2014).

Proposition 1 (State deviation and interval preservation)**.** *The bivariate update $\widehat{u} = F(o, a)$ satisfies*

$$\|\widehat{u} - a\|^2 + \|o - \widehat{u}\|^2 \le \|o - a\|^2. \tag{55}$$

If $\Phi(u, a) = \sum_i \Phi_i(u_i, a)$ is separable in u, every coordinate admits $\widehat{u}_i = (1 - k_i)o_i + k_i a_i$ for some $k_i \in [0, 1]$.

Proof. Optimality gives $o - \widehat{u} \in \partial_1\Phi(\widehat{u}, a)$, while the minimum at a gives $0 \in \partial_1\Phi(a, a)$. Monotonicity yields $\langle \widehat{u} - a, o - \widehat{u}\rangle \ge 0$. Expanding $\|o - a\|^2$ around $\widehat{u}$ proves Equation 55. For a separable potential, the argument applies coordinatewise: $(\widehat{u}_i - a_i)(o_i - \widehat{u}_i) \ge 0$. Thus $\widehat{u}_i$ lies between o_i and a_i. If $o_i \ne a_i$, set $k_i = (o_i - \widehat{u}_i)/(o_i - a_i)$; otherwise $\widehat{u}_i = a_i = o_i$, and any $k_i \in [0, 1]$ represents the same update. Each Φ_i may depend on the full fixed state vector a; separability is required only in the optimized coordinates. □

Equation 55 controls deviation from the reference, rather than error relative to an unknown target. The distinction can be expressed by a three-point inequality. Convexity and optimality give

$$\|\widehat{u} - u^\star\|^2 \le \|o - u^\star\|^2 - \|o - \widehat{u}\|^2 + 2\{\Phi(u^\star, a) - \Phi(\widehat{u}, a)\}. \tag{56}$$

Indeed, the subgradient inequality bounds $\Phi(u^\star, a) - \Phi(\widehat{u}, a)$ from below by $\langle o - \widehat{u}, u^\star - \widehat{u}\rangle$; expanding $\|o - u^\star\|^2$ proves the result. The bound is informative when $\Phi(u^\star, a)$ is finite.

If $u^\star \in \mathcal{C}(a) := \arg\min_u \Phi(u, a)$, the last difference is nonpositive, so

$$\|\widehat{u} - u^\star\|^2 + \|o - \widehat{u}\|^2 \le \|o - u^\star\|^2. \tag{57}$$

For fixed a, every $c \in \mathcal{C}(a)$ is a fixed point: $F(c, a) = c$. For example, with $\gamma \ge 0$ and complementary orthogonal projections Π_T, Π_N,

$$\Phi(u, a) = \frac{\gamma}{2}\|\Pi_N(u - a)\|^2, \qquad \widehat{u} = a + \Pi_T(o - a) + \frac{\Pi_N(o - a)}{1 + \gamma}. \tag{58}$$

This update preserves variation in $a + \mathrm{im}\,\Pi_T$ and attenuates its orthogonal component. Identifying a task-relevant admissible set requires additional structure or data.

Displacement potentials and state-dependent shapes. The subclass $\Phi(u, a) = \phi(u - a)$ has a common shape across states and gives $F(o, a) = a + \mathrm{prox}_\phi(o - a)$. For a scalar difference $d = o - a$, output displacement $z = F(o, a) - a$, and $\gamma, r \ge 0$, quadratic regularization gives $z = d/(1 + \gamma)$ and $k = \gamma/(1 + \gamma)$. The absolute-value penalty $\gamma|z|$ gives $z = \mathrm{sign}(d)(|d| - \gamma)_+$, suppressing small innovations completely. The indicator of $[-r, r]$ gives $z = \mathrm{clip}(d, -r, r)$, preserving small deviations while capping large ones. These choices all satisfy Proposition 1.

General bivariate potentials also allow their shape to change with the state. In one dimension, $\Phi(u, a) = \frac{1}{2}(1 + a^2)(u - a)^2$ gives $F(o, a) = a + (o - a)/(2 + a^2)$. The pairs $(o, a) = (1, 0)$ and $(2, 1)$ have the same difference $o - a = 1$, but output displacements $1/2$ and $1/3$. Hence no single state-independent ϕ represents this family as $\phi(u - a)$. For each fixed a, one can still define $\phi_a(z) = \Phi(a + z, a)$ and write $F(o, a) = a + \mathrm{prox}_{\phi_a}(o - a)$. The generalization makes this state dependence explicit; an unrestricted family of state-dependent displacement penalties is an equivalent parameterization.

References from a general internal state. If an internal state h occupies a different space, a map $r(h) \in \mathbb{R}^n$ supplies its membrane reference. More generally, let $\Psi(\cdot, h)$ be proper, closed, and convex, with $r(h) \in \arg\min_u \Psi(u, h)$. Then $\widehat{u} = \arg\min_u\{\frac{1}{2}\|u - o\|^2 + \Psi(u, h)\}$ obeys the same results with a replaced by $r(h)$. This covers retained states and nonlinear state transformations, including a predictor as one possible reference map.

Separable and coupled penalties. The energy and admissible-set results hold for general vector convex potentials. Separability guarantees coordinatewise interval preservation. For example, with $a = (0, 0)$, projecting $o = (-1, 2)$ onto the line $u_1 = u_2$ gives $(0.5, 0.5)$. The first coordinate has left the interval between its two candidates. Thus a coupled potential need not preserve their individual threshold decisions. The minimum-at-reference assumption also matters: the convex potential $\Phi(u, a) = \frac{1}{2}(u - a - 2)^2$ sends $o = a = 0$ to $\widehat{u} = 1$, violating interval preservation.

Fixed and adaptive regularizers. For the same fixed state a and potential parameters, write $u_j = F(o_j, a)$. Then

$$\|u_1 - u_2\|^2 \le \langle u_1 - u_2, o_1 - o_2\rangle. \tag{59}$$

This follows by subtracting their optimality conditions and applying subgradient monotonicity. Cauchy–Schwarz gives nonexpansiveness in o; μ-strong convexity of $\Phi(\cdot, a)$ strengthens its factor to $1/(1 + \mu)$. The assumptions impose no continuity in a. Comparing two complete executions also changes the state reference and possibly the potential parameters, so these fixed-operator bounds alone do not establish contraction of the full dynamics. The interval property applies separately to every actual update, including updates with input-dependent parameters.

G.3 Conditional errors and state weighting

Let $e_o = o - u^\star$ and $e_a = a - u^\star$ have finite second moments. Let $\mathcal{F}$ contain available information that determines the actual effective weight k. Define

$$A = \mathbb{E}[e_o^2 \mid \mathcal{F}], \quad B = \mathbb{E}[e_a^2 \mid \mathcal{F}], \quad C = \mathbb{E}[e_o e_a \mid \mathcal{F}], \quad V = A + B - 2C. \tag{60}$$

The conditional membrane error, $L(k) = \mathbb{E}[(\widehat{u} - u^\star)^2 \mid \mathcal{F}]$, satisfies

$$L(k) - A = -2k(A - C) + k^2 V. \tag{61}$$

These are uncentered moments, so they include mean bias. No independence or Gaussian assumption is used. Error-dependent weighting has a classical basis in forecast combination (Bates & Granger, 1969); conditioning makes the dependence of the actual gate explicit.

Proposition 2 (Complementarity gain and weight mismatch)**.** *For $V > 0$, put $q = A - C$ and $k^\star = \mathrm{clip}_{[0,1]}(q/V)$. Define*

$$\begin{aligned} \mathcal{B} &= 2k^\star q - V(k^\star)^2, \\ \mathcal{P} &= V(k - k^\star)^2 + 2(Vk^\star - q)(k - k^\star). \end{aligned} \tag{62}$$

Then $\mathcal{B}, \mathcal{P} \ge 0$ and

$$L(k) = A - \mathcal{B} + \mathcal{P}. \tag{63}$$

If $A > C$ and $B > C$, the optimum is interior and

$$L(k) = A - \frac{(A - C)^2}{V} + V(k - k^\star)^2, \qquad \gamma^\star = \frac{k^\star}{1 - k^\star} = \frac{A - C}{B - C}. \tag{64}$$

If $V = 0$, set $k^\star = \mathcal{B} = \mathcal{P} = 0$; then $L(k) = A$ for every admissible weight.

Proof. Since $e_{\widehat{u}} = e_o - k(e_o - e_a)$, expansion gives the pointwise identity

$$e_{\widehat{u}}^2 - e_o^2 = -2ke_o(e_o - e_a) + k^2(e_o - e_a)^2. \tag{65}$$

Taking conditional expectations proves Equation 61, because k is $\mathcal{F}$-measurable. For $V > 0$, the resulting quadratic has unconstrained minimizer q/V; projecting onto $[0, 1]$ gives $k^\star$. Expanding about this constrained minimizer yields

$$L(k) - L(k^\star) = V(k - k^\star)^2 + 2(Vk^\star - q)(k - k^\star).$$

The last product is nonnegative by interval optimality. Also $\mathcal{B} = L(0) - L(k^\star) \geq 0$, which proves Equation 63. For an interior optimum, $Vk^\star = q$ and the linear term vanishes, giving Equation 64. Finally, $V = \mathbb{E}[(e_o - e_a)^2 \mid \mathcal{F}] = 0$ implies that the two errors coincide almost surely under that condition. □

Averaging Equation 63 over the conditioning information gives strict overall improvement exactly when $\mathbb{E}\mathcal{P} < \mathbb{E}\mathcal{B}$. Summing over neurons gives the same statement for total squared membrane error without requiring independence across coordinates. At a boundary optimum, $A \leq C$ gives $k^\star = 0$, while $B \leq C$ and $V > 0$ give $k^\star = 1$. The linear term in Equation 62 must be retained in these cases. The optimum concerns a single candidate update under a specified joint error distribution and information set.

As an analytic example, $(A, B, C) = (1, 4, 0)$ gives $k^\star = 0.2$ and minimum risk 0.8: a less accurate state reference can still help through complementary errors. With $(A, B, C) = (1, 4, 1.2)$, the optimum is $k^\star = 0$. Both triples satisfy $C^2 \leq AB$. These are constructed moments, not measurements of KFC.

Conditional mean interpretation. Suppose $u^\star$ has finite second moment and $\mathcal{G}$ contains all available information, including o, a. Let $\mu = \mathbb{E}[u^\star \mid \mathcal{G}]$ and $I = [\min(o, a), \max(o, a)]$. For any $\mathcal{G}$-measurable candidate u,

$$\mathbb{E}[(u - u^\star)^2 \mid \mathcal{G}] = \mathrm{Var}(u^\star \mid \mathcal{G}) + (u - \mu)^2. \tag{66}$$

The ideal update within the interval is therefore $\Pi_I(\mu)$. When $o \neq a$, $V = (o - a)^2$ and $A - C = (o - a)(o - \mu)$ under this conditioning, so Equation 66 recovers $k^\star$. The unknown conditional mean is an analysis benchmark, not an additional model input.

Why conditioning matters. Random weights can depend on the same errors they combine (Claeskens et al., 2016). In a two-point example, let (e_o, e_a, k) equal $(2, 0, 3/4)$ or $(-1, 0, 1/4)$ with equal probability. The actual MSE is 0.40625, whereas incorrectly factoring it as $\mathbb{E}[(1 - k)^2]\mathbb{E}[e_o^2]$ gives 0.78125. Both weights are realized by finite quadratic penalties. Equation 65 is exact without conditioning; the conditional form permits comparison with an optimal state-weighting rule while retaining gate–error dependence.

Input–state difference energy alone also fails to determine the optimal weight. The triples $(A, B, C) = (1, 4, 0)$ and $(4, 1, 0)$ have the same $V = 5$ but optimal weights 0.2 and 0.8. Reliability therefore requires information about error complementarity, not just the size of the disagreement.

G.4 Hard-threshold events

Use the implementation's strict convention $H(x) = \mathbf{1}[x > 0]$. For one neuron, write $s_o = H(o - \theta)$, $s_a = H(a - \theta)$, $s_{\widehat{u}} = H(\widehat{u} - \theta)$, and $s^\star = H(u^\star - \theta)$.

Proposition 3 (Preservation of shared firing decisions)**.** *For a scalar potential $\Phi(\cdot, a)$ satisfying the assumptions of Section 4.3, $s_o = s_a$ implies $s_{\widehat{u}} = s_o$. On samples with $s_a = s^\star$,*

$$\mathbf{1}[s_{\widehat{u}} \neq s^\star] \leq \mathbf{1}[s_o \neq s^\star]. \tag{67}$$

Proof. Proposition 1 places $\widehat{u}$ between o and a. If both endpoints exceed θ, every point between them exceeds θ; if neither exceeds θ, no point between them does. If the state-reference event and input event are both correct, their shared decision is consequently preserved. If only the state-reference event is correct, the input error indicator is already one, proving Equation 67. □

For disagreeing candidates, the exact switching conditions are

$$\begin{aligned} o \leq \theta < a: \quad & s_{\widehat{u}} = 1 \iff k > \frac{\theta - o}{a - o}, \\ a \leq \theta < o: \quad & s_{\widehat{u}} = 0 \iff k \geq \frac{o - \theta}{o - a}. \end{aligned} \tag{68}$$

They follow by substituting $\widehat{u} = (1 - k)o + ka$ into the strict threshold. Equality is silent, accounting for the different endpoint conventions.

Define introduced and corrected errors by

$$E_{\text{new}} = \{s_o = s^\star,\ s_a \neq s^\star,\ s_{\widehat{u}} = s_a\},$$
$$E_{\text{fix}} = \{s_o \neq s^\star,\ s_a = s^\star,\ s_{\widehat{u}} = s_a\}. \tag{69}$$

On agreeing endpoints the error indicator is unchanged. On disagreeing endpoints the output follows either the input or the state reference, so a direct case split yields

$$\Pr(s_{\widehat{u}} \neq s^\star) = \Pr(s_o \neq s^\star) - \Pr(E_{\text{fix}}) + \Pr(E_{\text{new}}). \tag{70}$$

Thus the difference between corrections and introduced errors measures the actual event benefit. When both candidates share an incorrect decision, this family of regularizers cannot repair it.

A local margin condition. Suppose nominal endpoints o_0, a_0 lie strictly on the same desired side of the threshold. With $\sigma = 2s^\star - 1$, let $m_o = \sigma(o_0 - \theta) > 0$ and $m_a = \sigma(a_0 - \theta) > 0$. If perturbed endpoints satisfy

$$|o' - o_0| < m_o, \qquad |a' - a_0| < m_a, \tag{71}$$

they retain their shared event. Proposition 3 then preserves the output for any perturbed weight $k' \in [0, 1]$. This certificate needs bounds on the actual candidate endpoints, including their history dependence; it does not require a bound on the gate's derivative.

Membrane MSE and event error. A lower MSE alone need not reduce event error. Consider $u^\star = 0$, $\theta = 1$, $a = 3$, and $k = 1/2$, with $o = -2$ or $o = 0$ equally likely. Then $\widehat{u} = 0.5$ or 1.5. MSE falls from 2 to 1.25, while event error rises from 0 to $1/2$. For a positive reference margin $\rho^\star = |u^\star - \theta|$, the distribution-free inequality

$$\mathbf{1}[H(u - \theta) \neq H(u^\star - \theta)] \leq \frac{(u - u^\star)^2}{(\rho^\star)^2} \tag{72}$$

holds because an event change requires an error at least as large as the margin. Averaging gives an upper bound, rather than an ordering of actual event risks. Equation 70 measures those risks directly.

G.5 Exact correspondence with KF

At each actual step, set

$$m_t = x^t_{\Re}, \quad p_t = \hat{x}^t_{\Re}, \quad b_t = \tau v^{t-1}_{\Re}, \quad h_t = s^{t'}_{\Im} \odot v^{t^-}_{\Im}. \tag{73}$$

Here p_t is the complete current prediction, $W_P v^{t-1}_{\Re}$. The membrane reference is $a_t = b_t + p_t$ and the input candidate is $o_t = b_t + m_t$. They share the same actual history and describe the integration step before imaginary inhibition.

For the quadratic penalty in Equation 20, stationarity gives

$$(u_i - o_{t,i}) + \gamma_{t,i}(u_i - a_{t,i}) = 0, \qquad u_i = \frac{o_{t,i} + \gamma_{t,i} a_{t,i}}{1 + \gamma_{t,i}} = (1 - K^t_i) o_{t,i} + K^t_i a_{t,i}. \tag{74}$$

This is precisely the real membrane before imaginary inhibition, $\widehat{u}_t = v^{t^-}_{\Re,0} = \tau v^{t-1}_{\Re} + x^{t|t}_{\Re}$. The trace bounds in Appendix H imply finite $\gamma_{t,i} \geq 0$; initialization with $P^t = 0$ gives zero regularization. No symmetry of W_P is needed, because the preceding state and the resulting reference are given within the current optimization.

The full real update in Equation 13 adds the inhibitory correction, $v^{t^-}_{\Re} = \widehat{u}_t - \lambda h_t$. At a fixed inhibitory signal h_t, actual real firing is equivalently obtained by comparing $\widehat{u}_t$ with the effective threshold $\theta_{\Re} + \lambda h_t$. The framework's interval and event-consensus results therefore apply to the unshifted candidates at this common effective threshold. Across trajectories, changes in h_t also change that threshold and must be included when comparing actual events. The generation and temporal evolution of imaginary spikes are additional dynamics, analyzed through the state-law identities in Appendix H and the cross-channel probes.

The implemented strength $P^t/(R^t + \epsilon)$ uses normalized observation variation and prediction mismatch. These traces are not measured conditional error moments, so their ratio is not automatically the interior optimum $(A - C)/(B - C)$. Stopping gradients through the traces during training does not remove their input dependence during inference.

H Analysis of the State Law

H.1 Bounds and initialization

Proposition 4. *For real a, b and $\epsilon > 0$, $0 \le d(a,b) < 2$. With zero initialization and $\alpha, \beta \in (0,1]$, the traces satisfy $P^t, R^t \in [0,2)$ and the prediction weight satisfies $K^t \in [0,1)$.*

Proof. The identity $(a-b)^2 \le 2(a^2+b^2)$ gives the discrepancy bound, with strict upper bound after adding ϵ to the denominator. Every trace update is a convex combination of its preceding value and a discrepancy in $[0,2)$. Induction proves the trace bounds. Nonnegative numerator and a strictly larger positive denominator then give the bound on K^t. □

For P^t, R^t treated as independent scalar inputs to the ratio,

$$\frac{\partial K^t}{\partial P^t} = \frac{R^t + \epsilon}{(P^t + R^t + \epsilon)^2} > 0, \qquad \frac{\partial K^t}{\partial R^t} = -\frac{P^t}{(P^t + R^t + \epsilon)^2} \le 0. \tag{75}$$

These relations explain the two traces' effects on KF. They are algebraic sensitivities, separate from the implemented stop-gradient convention. The normalized discrepancy is symmetric but is used as a discrepancy score rather than as a metric satisfying the triangle inequality.

H.2 Expanded real-state dynamics

Substitution of the KF update rule gives the real membrane update

$$v_{\Re}^{t^-} = \tau v_{\Re}^{t-1} + K^t \odot (W_P v_{\Re}^{t-1}) + (1 - K^t) \odot x_{\Re}^t - \lambda s_{\Im}^{t'} \odot v_{\Im}^{t^-}. \tag{76}$$

With $v = v_{\Re}^{t-1}$, $m_t = x_{\Re}^t$, $D_t = \mathrm{Diag}(K^t)$, and $h_t = s_{\Im}^{t'} \odot v_{\Im}^{t^-}$, the complete additive correction relative to ordinary integration is

$$u_t = \tau v + m_t + r_t^{\mathrm{KFC}}, \qquad r_t^{\mathrm{KFC}} = D_t(W_P v - m_t) - \lambda h_t. \tag{77}$$

The two additive terms $D_t W_P v$ and $-\lambda h_t$ instead use the gated measurement base $\tau v + (I - D_t) m_t$. The variational representation of KF is Equation 20.

H.3 Exact paired-trajectory decomposition

Let an overbar denote a second execution under a perturbed input with the same fixed weights. At a given step, the pre-reset real-state difference obeys

$$\begin{aligned} v_{\Re}^{t^-} - \bar{v}_{\Re}^{t^-} = {} & [\tau I + \mathrm{Diag}(K^t) W_P](v_{\Re}^{t-1} - \bar{v}_{\Re}^{t-1}) \\ & + (1 - K^t) \odot (x_{\Re}^t - \bar{x}_{\Re}^t) \\ & + (K^t - \bar{K}^t) \odot (W_P \bar{v}_{\Re}^{t-1} - \bar{x}_{\Re}^t) \\ & - \lambda \left(s_{\Im}^{t'} \odot v_{\Im}^{t^-} - \bar{s}_{\Im}^{t'} \odot \bar{v}_{\Im}^{t^-} \right). \end{aligned} \tag{78}$$

Thus the fixed-trust carryover operator describes only one part of perturbation propagation. Input differences, trust changes, and imaginary-channel differences contribute separately. Under soft real reset,

$$v_{\Re}^t - \bar{v}_{\Re}^t = (v_{\Re}^{t^-} - \bar{v}_{\Re}^{t^-}) - \theta_{\Re} \odot (s_{\Re}^{t'} - \bar{s}_{\Re}^{t'}). \tag{79}$$

A spike disagreement therefore adds a reset difference even if the pre-reset states are close. This is the temporal mechanism that motivates tracking membranes and events in addition to final task accuracy.

H.4 Threshold reserve identity

Let $v_{\Re,0}^{t^-} = \tau v_{\Re}^{t-1} + x_{\Re}^{t|t}$ be the pre-inhibition drive. For nonnegative λ and $\theta_{\Im}$, the event difference is

$$s_{\Re}^{t'} - H(v_{\Re,0}^{t^-} - \theta_{\Re}) = -\mathbf{1}\left[\theta_{\Re} < v_{\Re,0}^{t^-} \le \theta_{\Re} + \lambda s_{\Im}^{t'} \odot v_{\Im}^{t^-}\right]. \tag{80}$$

For one coordinate, write the pre-inhibition real drive as v_0 and the inhibitory amplitude as $\lambda s_{\Im} v_{\Im} \geq 0$. A spike occurs after inhibition exactly when $v_0 > \theta_{\Re} + \lambda s_{\Im} v_{\Im}$. The no-inhibition spike occurs when $v_0 > \theta_{\Re}$. Subtracting these indicators proves Equation 80, including its strict lower and closed upper endpoint. It follows that inhibition cannot directly promote a real spike at the same fixed state. This is an instantaneous identity. Changing the reset history can subsequently increase or decrease firing.

H.5 State-law regularization and weight sparsity

The local sparsity penalty restricts the predictor's coefficient distribution during training. The dynamical interpretation concerns how the resulting predictor and imaginary spikes enter the time-dependent forward computation. Neither sparsity alone nor a convex fusion weight bounds the spectral norm of $\tau I + \mathrm{Diag}(K^t) W_P$.

H.6 Energy diagnostics and the membrane-block Jacobian

Write $D_t = \mathrm{Diag}(K^t)$, $v = v_{\Re}^{t-1}$, $m_t = x_{\Re}^t$, and $h_t = s_{\Im}^{t'} \odot v_{\Im}^{t^-}$. Relative to ordinary integration of the same histories, subtracting the two real updates gives $\Delta u_t = B_t + \Delta r_t^{\mathrm{KFC}}$, where $B_t = \tau \Delta v + \Delta m_t$. Its squared norm gives

$$\mathcal{D}_t \equiv \|\Delta u_t\|^2 - \|B_t\|^2 = 2\langle B_t, \Delta r_t^{\mathrm{KFC}} \rangle + \|\Delta r_t^{\mathrm{KFC}}\|^2. \tag{81}$$

The actual correction difference includes gate changes through

$$\Delta r_t^{\mathrm{KFC}} = D_t(W_P \Delta v - \Delta m_t) + (D_t - \bar{D}_t)(W_P \bar{v} - \bar{m}_t) - \lambda \Delta h_t. \tag{82}$$

Both B_t and the gated reference below are algebraic quantities evaluated on the same pair of trajectories.

For paired executions with the same weights, let bars denote perturbed quantities and set

$$\begin{aligned} J_t &= D_t W_P v - \lambda h_t, \qquad \bar{J}_t = \bar{D}_t W_P \bar{v} - \lambda \bar{h}_t, \\ b_t &= \tau(v - \bar{v}) + (I - D_t)(m_t - \bar{m}_t), \\ d_t &= J_t - \bar{J}_t - (D_t - \bar{D}_t)\bar{m}_t. \end{aligned} \tag{83}$$

Direct subtraction gives $u_t - \bar{u}_t = b_t + d_t$ and the exact identity

$$\mathcal{E}_t \equiv \|u_t - \bar{u}_t\|^2 - \|b_t\|^2 = 2\langle b_t, d_t \rangle + \|d_t\|^2. \tag{84}$$

Thus $\mathcal{E}_t < 0$ means that the joint correction attenuates the pre-reset membrane difference relative to the algebraic base b_t at that step and along that perturbation. The trust-change contribution $-(D_t - \bar{D}_t)\bar{m}_t$ is part of this correction. Omitting it would analyze a different forward computation. The reference b_t is evaluated on the same trajectory pair.

Equation 84 uses a gated measurement base, whereas Equation 81 uses ordinary integration. Both are directional diagnostics on actual trajectory pairs. Uniform trajectory bounds would additionally have to control both membranes, the discrepancy memories P, R, the preceding measurement, and any architecture-specific memory variables. The conditional proximal representation establishes a single-update geometry; the complete paired dynamics retain the trust, event, and reset changes in Equation 78.

The two mechanisms have different dynamical roles. With trust and events fixed and with external currents independent of the previous membrane state of this layer, soft real reset and hard imaginary reset give the membrane-block Jacobian

$$J_{\mathrm{mem},t} = \begin{bmatrix} \tau I + D_t W_P & -\lambda \tau S_t \\ 0 & \tau(I - S_t) \end{bmatrix}, \qquad S_t = \mathrm{Diag}(s_{\Im}^{t'}). \tag{85}$$

Prediction changes the real-state diagonal block. Inhibition changes cross-channel transfer and event boundaries; in this frozen block it does not change eigenvalues, because the matrix is block triangular. It can still change singular values and transient amplification. Likewise, a prediction weight in $[0, 1)$ does not guarantee contraction: the scalar values $\tau = 0.9$, $K = 0.5$, and $W_P = 1$ give a carryover factor of 1.4. For changing operators, a stability argument must control their products or a common contraction norm. Equation 78 provides the full trajectory difference.

H.7 A SPECTRAL-NORM BOUND ON THE CARRYOVER OPERATOR

For nonnegative τ, submultiplicativity gives

$$\|\tau I + \mathrm{Diag}(K^t)W_P\|_2 \le \tau + \|K^t\|_\infty \|W_P\|_2. \tag{86}$$

Thus $\tau + \|K^t\|_\infty \|W_P\|_2 < 1$ is a sufficient condition for contraction of the fixed-trust carryover operator, including the case $W_P = 0$. Evaluating this condition on a trajectory requires its actual K^t values; the full state difference additionally contains the measurement, trust-change, inhibition, and reset terms in Equation 78.