Spiking neural network initialization for scale-invariant stabilization at any spike rate

Reinier Willem Vos[1,*] **Aurora Micheli**[1] **Nergis Tomen**[1] **Christophe De Wagter**[1]
Jan van Gemert[1] **Sander M. Bohté**[2,3] **Guido C.H.E. de Croon**[1]

[1]Delft University of Technology, Delft, The Netherlands
[2]CWI, Amsterdam, The Netherlands
[3]University of Amsterdam, Amsterdam, The Netherlands
[*]Corresponding author: r.w.vos@tudelft.nl
Code: https://github.com/tudelft/snn-init-lib

Abstract

Parameter initialization shapes training dynamics, convergence, and performance in neural networks. In spiking neural networks (SNNs), analytical initialization is complicated by three additional mechanisms—binary spike outputs, temporal recurrence (leak), and reset-induced drops in membrane potential—which together suppress activity propagation known as spike-rate collapse: (1) over time, where recurrence and reset reduce individual neurons' firing rate, and (2) across depth, where deficits in upstream activity compromise downstream layers. Prior initialization methods address only subsets of these challenges. We derive closed-form weight and bias distributions that jointly account for temporal recurrence, refractory reset, and signal propagation across depth. Our Closed-Form Stable (CFS) method is data-agnostic: it requires only the architecture and neuron hyperparameters. Our derivation centers around a bias correction that holds each neuron at any target spike rate, jointly stationary across time and depth and scale-invariant in width. In controlled 30-layer simulations, it attains the target rate at neuron-specific level, unlike the alternatives tested. Beyond simulation, we evaluate empirical settings where the ideal task-specific spike rate is assumed unknown. To provide a principled default, we adopt the maximum-entropy rate of a Bernoulli neuron, $p = 0.5$, which counteracts dead ($p \approx 0$) and saturated ($p \approx 1$) neurons. Without per-paradigm tuning, this default ranks first across time-series classification, neuromorphic classification, and continuous-control reinforcement learning.

1 Introduction

Neuromorphic computing promises energy-efficient computation through sparse, event-driven processing. Central to this paradigm are *spiking neural networks* (SNNs), which communicate through 1-bit spikes rather than the real-valued activations of *artificial neural networks* (ANNs). Recent advances in surrogate gradient methods [Bellec et al., 2018, Zenke and Ganguli, 2018, Neftci et al., 2019] have made gradient-based SNN training via *backpropagation through time* (BPTT) routine in mainstream ANN/SNN frameworks [Paszke et al., 2019, Pehle and Pedersen, 2021, Eshraghian et al., 2023, Fang et al., 2023].

While these advances have made gradient-based optimization accessible, the parameter initialization that sets its starting point remains underdeveloped for SNNs relative to ANNs [Micheli et al., 2025]. For ANNs, initialization is a standardized practice: data-agnostic schemes [Glorot and Bengio, 2010, He et al., 2015] are the default in major deep-learning libraries [Paszke et al., 2019, Abadi et al., 2016]. These schemes derive weight and bias distributions in closed form from the architectural

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

dimensions and activation function, targeting unit pre-activation variance for stable forward signal propagation. To this date, no equivalent standardization exists for SNNs.

This absence stems from the distinct dynamics of spiking neurons. The leaky-integrate-and-fire (LIF) neuron (Equation 1) couples three mechanisms absent in ANN units: binary spikes, leak-driven temporal recurrence, and a refractory reset on firing. Their interaction can suppress firing along two complementary axes, a failure mode termed *spike-rate collapse* [Perez-Nieves and Goodman, 2023, Micheli et al., 2025]: (1) through time, the recurrence–reset interaction decays the firing rate of an individual neuron; (2) through depth, an upstream activity deficit propagates downstream. Both axes converge on the same pathology: neurons driven to $p \approx 0$ (dead) or $p \approx 1$ (saturated) contribute minimally to the network's representational capacity. Avoiding these extremes requires an initialization aligned with the complete LIF dynamics, which no existing method provides.

In this work, we derive a closed-form, data-agnostic initialization that accounts for the complete LIF dynamics: spatial signal propagation, temporal recurrence, and reset. For LIF networks it attains any target spike rate with neuron-level precision and depends solely on the architecture and neuron hyperparameters. The same closed form extends to alternative neuron definitions, heterogeneous populations, and architectures, where it preserves this precision across neurons. Section 2 details the method, Section 3.1 verifies it in controlled simulation, Section 3.2 validates it empirically, and Section 3.3 discusses its implications.

1.1 Related work

For ANNs, the *Xavier* [Glorot and Bengio, 2010] and the *Kaiming* [He et al., 2015] initializations derive closed-form weight scales from the architectural dimensions and the activation function to maintain unit pre-activation variance. These methods address spatial signal propagation in ANNs but do not readily extend to networks with binary activations, temporal dynamics or reset mechanisms.

For SNNs, several data-driven methods address SNN-specific dynamics but require empirical estimation of signal statistics. Rossbroich et al. [2022] model spike generation as a discrete Poisson process and introduce target mean μ_u^* and target variance σ_u^* for the membrane potential, a notation we also adopt. Huebotter et al. [2025] extend the analysis to various surrogate-gradient definitions. Herranz-Celotti and Rouat [2024] center the membrane potential at the threshold to achieve 50% initial spike rates, but require empirical layer input statistics. Liu et al. [2025] target a different data-driven setting: normalization-free SNNs with lateral inhibition, where the initialization is calibrated to measured spike rates and is tied to that architectural variant rather than to a standard LIF definition.

Among data-agnostic approaches, closed-form derivation is complicated by the interplay of spatial, temporal, and reset dynamics. Zenke and Vogels [2020] defer the question to training, arguing that surrogate-gradient learning is sufficiently robust that the uniform baseline of the Xavier initialization (originally derived for ANNs with sigmoid/tanh activations [Glorot and Bengio, 2010]) suffices. Lee et al. [2016] instead couple the threshold to the weight scale, while Bellec et al. [2018] apply a conservative gain factor on top of the Xavier initialization. Chang et al. [2024] extend the Xavier methodology for SNNs with a surrogate-gradient correction, folding the second moment of the surrogate gradient into the weight scale to target stable backward gradient flow. None of these proposals, however, brings the LIF dynamics into the variance derivation itself.

More recent works attempt to incorporate the LIF neuron's dynamics explicitly. The derivation by Micheli et al. [2025] accounts for binary activations in spatial propagation, yet neglects temporal dynamics such as recurrence and reset. Perez-Nieves and Goodman [2023] show that the interaction of recurrence and reset leads to the temporal form of spike-rate collapse, a progressive silencing of neurons over time. Jointly addressing both mechanisms is thus qualitatively different from treating them in isolation. To mitigate this, Perez-Nieves and Goodman [2023] and Che et al. [2025] propose variance inflation strategies that maintain minimum activity levels. Ding et al. [2025] instead introduce a non-zero bias to counteract decay, but do not directly stabilize spike rates. Table 1 summarizes these methods along four axes: calibration from data-dependent statistics, spatial depth-wise propagation, temporal recurrence, and reset functionality; see Section A.1 for a per-method decomposition with assumptions and hyperparameters. As the table indicates, existing data-agnostic methods do not capture the coupled dynamics of recurrent SNNs with reset mechanisms across both spatial and temporal dimensions. We address this limitation by proposing and evaluating a closed-form, data-

Table 1: **Categorization of SNN weight-initialization methods along four conceptual axes. Data-agnostic**: applicable without empirical calibration from data-dependent statistics. **Spatial**: accounts for binary activations across layers. **Recurrence**: accounts for membrane potential's temporal state dynamics. **Reset**: integrates discrete membrane drop upon spiking. Per-method justifications (including *-flagged entries) are given in Section A.1, with tildes ($\sim$) marking partial coverage. Methods above the rule are data-driven; below, data-agnostic, with ours last.

Method	Data-agnostic	Spatial	Recurrence	Reset
Rossbroich et al. [2022]	×	✓	✓	$\sim^*$
Herranz-Celotti and Rouat [2024]	×	✓	✓	✓
Liu et al. [2025]	×	✓	×	×
Glorot and Bengio [2010]	✓	×	×	×
He et al. [2015]	✓	×	×	×
Zenke and Vogels [2020]	✓	✓	×	×
Lee et al. [2016]	✓	✓	×	×
Bellec et al. [2018]	✓	✓	×	×
Chang et al. [2024]	✓	✓	×	×
Micheli et al. [2025]	✓	✓	×	$\sim^*$
Perez-Nieves and Goodman [2023]	✓	✓	✓	×
Che et al. [2025]	✓	✓	×	×
Ding et al. [2025]	✓	✓	$\sim^*$	$\sim^*$
Ours (Equation 8)	✓	✓	✓	✓

agnostic initialization that jointly accounts for spatial signal propagation, temporal stability, and reset dynamics within a unified framework.

Contributions. (1) **Analytical initialization.** We derive a closed-form, data-agnostic initialization for LIF neurons. It enforces a target spike rate for every neuron that is jointly stationary across time and depth and scale-invariant with respect to width. It accounts for spatial, temporal, and reset dynamics together, which prior data-agnostic approaches address only in part.
(2) **Simulation verification.** We verify the theory in controlled simulations of a 30-layer network. The initialization attains any target spike rate per neuron in every layer, a precision no tested alternative reaches. The same closed form extends to integrate-and-fire and binary neurons, hard and soft resets, heterogeneous neuronal populations, and residual, recurrent, and convolutional architectures.
(3) **Empirical validation.** We test whether our approach generalizes without per-paradigm tuning or prior knowledge of a task's ideal spike rate, by adopting the maximum-entropy firing rate of a Bernoulli neuron ($p = 50\%$) as a principled default that keeps neurons away from silence and saturation. We demonstrate its practical versatility: our approach ranks first in every paradigm across UEA multivariate time-series classification, neuromorphic classification, and continuous-control reinforcement learning, outperforming alternative methods and target rates.

2 Methodology

2.1 Model definition

We focus on the *leaky-integrate-and-fire* (LIF) neuron with internal recurrence and hard reset, a widely adopted model in neuromorphic computing:

$$\begin{aligned} U_t^l &= I_t^l + \left[\left(\beta^l U_{t-1}^l + (1-\beta^l)U_\infty^l\right)\left(1 - S_{t-1}^l\right) + U_{\text{reset}}^l S_{t-1}^l\right] \\ U_t^l &= I_t^l + \tilde{U}_{t-1}^l, \quad S_t^l = H(U_t^l;\theta^l), \quad I_t^l = \mathbf{W}_X^l X_t^l + \mathbf{b}_X^l \end{aligned} \tag{1}$$

Here, U_t^l denotes the membrane potential of layer l at time t and β^l its leak factor. The resting potential U_∞^l is zero unless otherwise defined, and U_{reset}^l denotes the reset potential. The upstream weights $\mathbf{W}_X^l$ and bias $\mathbf{b}_X^l$ map the layer input X_t^l to the weighted input current I_t^l. The *membrane recurrence* $\tilde{U}_{t-1}^l$ contracts the bracketed expression. Backpropagation in SNNs replaces the gradient of the non-differentiable Heaviside step H by that of a smooth surrogate $\hat{H}$ [Neftci et al., 2019, Zenke and Ganguli, 2018], for which we adopt the *fastsigmoid-k_{25}* surrogate with slope $k = 25$:

$$s = H(u;\theta) \quad \& \quad \frac{\partial \hat{H}(u;\theta)}{\partial u} = \text{SG}(u;\theta,k) = \left.\frac{1}{(1+k|u-\theta|)^2}\right|_{k=25} \tag{2}$$

2.2 Initialization objectives

We specify three interrelated objectives at the neuron level and show that they propagate upward through the system hierarchy to the layer and network levels (Section 3.1).
(A) **Signal stability** requires a membrane-potential distribution stationary across time and depth, which deep SNNs lack by default. This enforces distributional stationarity, but not the moments.
(B) **Neuron utilization** targets the maximum-entropy Bernoulli rate $p_m = 50\%$ of each neuron m to prevent dead ($p_m \approx 0\%$) and saturated ($p_m \approx 100\%$) neurons. For a symmetric density, this places the membrane-potential mean μ_u at the firing threshold θ.
(C) **Gradient flow** maximizes the expected surrogate gradient $\mathbb{E}[\mathrm{SG}(u;\theta)]$ as its proxy, which scales the gradients backpropagated through the spiking nonlinearity. Mirroring the unit pre-activation variance of ANN initializations, we target $\sigma_u^2 = 1$. For a Gaussian membrane potential with this variance, the expectation is maximized at $\mu_u = \theta$, so (B) and (C) are met simultaneously. Formally:

$$\begin{aligned}
&(A): U_{t-1}^l \overset{d}{=} U_t^l, \quad U_t^{l-1} \overset{d}{=} U_t^l \\
&(B): \mu_u^* = \theta, \quad H(u_m;\theta) = s_m \overset{\text{i.i.d.}}{\sim} \mathrm{Ber}(p_m = 50\%) \;\Rightarrow\; \|S\|_1 \sim \mathrm{Bin}(D_{\text{out}}, 50\%) \\
&(C): \sigma_u^{*2} = 1 \;\Rightarrow\; \mathrm{argmax}_{\mu_u}\, \mathbb{E}_{u\sim\mathcal{N}(\mu_u,\sigma_u^{*2})}[\mathrm{SG}(u;\theta)] = \theta
\end{aligned} \tag{3}$$

where the stars mark μ_u^* and σ_u^{*2} as the *target* moments. The implication in (B) assumes that the D_{out} neurons of a layer are i.i.d. with a shared target rate p_m. Then $s_m \sim \mathrm{Ber}(p_m)$ gives the binary spike vector $S \sim \mathrm{Ber}(p_m)^{\otimes D_{\text{out}}}$, whose L_1 norm or spike count $\|S\|_1 = \sum_m s_m$ is Binomial.

2.3 Initialization methodology

We realize objectives (A)–(C) as a *self-consistent fixed-point closure*. We seek the weight distribution under which the target $U_t^l \sim \mathcal{N}(\theta^l, 1)$ of objectives (B)–(C) is a fixed point of the LIF dynamics in Equation 1. Such a fixed point realizes objective (A) by preventing spike-rate collapse along the temporal axis ($U_{t-1}^l \overset{d}{=} U_t^l$) and the spatial axis ($U^{l-1} \overset{d}{=} U^l$ with $p^{l-1} = p^l$). We denote this map $\Psi : (\mu_{U^l}, \sigma_{U^l}^2, p^{l-1}) \mapsto (\mu_{\mathbf{W}_X}, \sigma_{\mathbf{W}_X}^2)$. Its closed form rests on the assumptions under which Ψ is well-defined, the truncated-Gaussian moments of the membrane recurrence and the resulting weight statistics. Full derivations are in Appendix B.

Distributional assumptions. For an intermediate layer l with binary input $X_t^l = S_t^{l-1}$, the map Ψ rests on six assumptions. (A1) U_t^l is approximately Gaussian. (A2) Its distribution is stationary in t. (A3) $\mathrm{Cov}(I_t^l, U_{t-1}^l) \approx 0$ at initialization. (A4) The entries of I_t^l are approximately uncorrelated across output neurons. (A5) The entries of S_t^{l-1} are approximately i.i.d. $\mathrm{Ber}(p^{l-1})$ and independent over time. (A6) The hyperparameters $\{\theta^l, \beta^l, U_{\text{reset}}^l\}$ are known and deterministic at initialization. The central limit theorem (CLT) supports A1–A2 in two steps. Given a Gaussian $\mathbf{W}_X$ and an i.i.d. Bernoulli X_t^l, a finite-dimensional CLT makes $I_t^l = \mathbf{W}_X X_t^l$ approximately Gaussian. The recursive accumulation of these Gaussian innovations under the LIF dynamics then *Gaussianizes* U_t^l across time (caveats and empirical validation in Section B.3). A5 is stricter than assuming a multivariate Bernoulli input, which permits unequal rates and correlations across neurons. Under A1–A2 the temporal independence of A3 and the cross-neuron independence of A4 sustain A5. Orthogonal weight initialization further reinforces A4, as mutually orthogonal rows keep the input currents of distinct neurons nearly uncorrelated. The assumptions thus reduce to a Gaussian weight matrix and an i.i.d. Bernoulli input with a Binomial spike count, which together determine the input current:

$$\begin{aligned}
\mathbf{W}_X^l \in \mathbb{R}^{(D_{\text{out}}, D_{\text{in}})} \quad &\sim \mathcal{N}\big(\mu_{\mathbf{W}_X},\, \mathbf{I}_{(D_{\text{out}}\times D_{\text{in}})} \cdot \sigma_{\mathbf{W}_X}\big) \\
X_t^l = S_t^{l-1} \in \mathbb{B}^{(D_{\text{in}},1)} \quad &\sim \mathrm{Ber}(p^{l-1} = 50\%)^{\otimes D_{\text{in}}} \;\Rightarrow\; \|X_t^l\|_1 \sim \mathrm{Bin}(D_{\text{in}}, p^{l-1}) \\
I_t^l = \mathbf{W}_X^l X_t^l \in \mathbb{R}^{(D_{\text{out}},1)} \quad &\sim \mathcal{N}\left(D_{\text{in}}\, p^{l-1}\, \mu_{\mathbf{W}_X},\; \sqrt{D_{\text{in}}\, p^{l-1}(1-p^{l-1})\big(\sigma_{\mathbf{W}_X}^2 + \mu_{\mathbf{W}_X}^2\big)}\right)
\end{aligned} \tag{4}$$

Here, $\mathbb{B}$ denotes the binary space and p^{l-1} the spike rate of the preceding layer. The weight mean and standard deviation are $\mu_{\mathbf{W}_X}$ and $\sigma_{\mathbf{W}_X}$, $\mathbf{I}$ is the identity matrix and the bias is the zero vector $\mathbf{b}_X^l = \mathbf{0} \in \mathbb{R}^{D_{\text{out}}}$. The third line is the CLT result $I_t^l \sim \mathcal{N}(\mu_{I^l}, \sigma_{I^l})$. It recovers Micheli et al. [2025][1]

[1] Evaluated throughout in its per-neuron form (Section A.1).

as the special case $\mu_{I^l} = 0$ and $\sigma_{I^l} = 1$. The weights realize their moments through an orthogonal base $\mathbf{W}^l_{X,\perp}$, which we scale to $\sigma_{\mathbf{W}_X}$ and shift by $\mu_{\mathbf{W}_X}$. For $D_{\text{out}} > D_{\text{in}}$ this base is rank-deficient and its rows have squared norm $D_{\text{in}}/D_{\text{out}}$ on average, which the aspect-ratio correction $\sqrt{D_{\text{out}}/D_{\text{in}}}$ offsets. Standardizing each row instead offsets this deficit exactly (Section B.1.1). In closing, we emphasize that given p^{l-1} and the hyperparameters of A6, the weight moments alone determine μ_{I^l} and σ_{I^l}.

Closure map. The map Ψ propagates the input current through the LIF dynamics to the stationary membrane potential. The LIF update is a regime-switching nonlinearity, so its exact stationary moments are not available in closed form. Under (A1)–(A6) they nonetheless admit an accurate closed-form approximation, as the per-neuron simulations of Section 3.1 confirm. Four analytical steps construct this approximation (Section B.2). First, the membrane recurrence $\tilde{U}^l_{t-1}$ decomposes into two mutually exclusive spike-induced clauses. Second, each clause is a *masked* rather than a truncated random variable. The law of total variance relates its moments to those of the corresponding truncated variable. Third, under (A1) these are truncated-Gaussian moments with a closed form in the inverse Mills ratio $r = \phi(\alpha)/\Phi(\alpha)$ of the standard-normal PDF ϕ and CDF Φ at $\alpha = (\theta^l - \mu_{U^l})/\sigma_{U^l}$. Fourth, the deterministic reset (A6) reduces the reset clause to a scaled Bernoulli variable. Collectively, these definitions render the moments of the mutually exclusive spike clauses explicit and yield the moments of the overall dynamics algebraically.

For any admissible target $(\mu_{U^l}, \sigma^2_{U^l})$ and hyperparameters $\{\theta^l, \beta^l, U^l_{\text{reset}}\}$, the inner step $f^* : (\mu_{U^l}, \sigma^2_{U^l}) \mapsto (\mu_{\tilde{U}^l_{t-1}}, \sigma^2_{\tilde{U}^l_{t-1}})$ derived in Section B.2.5 returns

$$\begin{aligned}
\mu_{\tilde{U}^l_{t-1}} &= \beta^l(1-p^l)(\mu_{U^l} - r\sigma_{U^l}) + p^l\, U^l_{\text{reset}} \\
\sigma^2_{\tilde{U}^l_{t-1}} &= (\beta^l)^2\big((1-p^l)\sigma^2_{U^l}(1-\alpha r - r^2) + (1-p^l)p^l(\mu_{U^l} - r\sigma_{U^l})^2\big) \\
&\quad + (U^l_{\text{reset}})^2\, p^l(1-p^l) - 2\beta^l\, U^l_{\text{reset}}\, p^l(1-p^l)(\mu_{U^l} - r\sigma_{U^l}).
\end{aligned} \tag{5}$$

Since $U^l_t = I^l_t + \tilde{U}^l_{t-1}$ with uncorrelated terms under (A3), the input current must supply

$$\mu_{I^l} = \mu_{U^l} - \mu_{\tilde{U}^l_{t-1}}, \qquad \sigma^2_{I^l} = \sigma^2_{U^l} - \sigma^2_{\tilde{U}^l_{t-1}}. \tag{6}$$

Equations 5 and 6 thus invert this propagation for any admissible target and hyperparameters.

Complete initialization scheme. Equating the input-current moments of Equation 4 with the required moments of Equation 6 determines the free weight moments and completes Ψ. We evaluate Ψ at the targets $\mu_{U^l} = \theta^l$ and $\sigma^2_{U^l} = 1$ of objectives (B)–(C). This gives $\alpha = 0$ and $r|_{\alpha=0} = 2/\sqrt{2\pi}$ in Equation 5. Solving for the weight moments yields the closed form

$$\mu_{\mathbf{W}_X} = \frac{\theta^l - \mu_{\tilde{U}^l_{t-1}}}{p^{l-1}\, D_{\text{in}}}, \qquad \sigma^2_{\mathbf{W}_X} = \frac{1 - \sigma^2_{\tilde{U}^l_{t-1}}}{D_{\text{in}}\, p^{l-1}(1-p^{l-1})} - \mu^2_{\mathbf{W}_X}. \tag{7}$$

The non-zero $\mu_{\mathbf{W}_X}$ realizes the bias correction. It offsets the temporal decay and the reset-induced drop captured by $\mu_{\tilde{U}^l_{t-1}}$ to keep $\mathbb{E}[U^l_t]$ at θ^l. The orthogonal construction assembles the weights:

$$\mathbf{W}^l_X = \left(\mathbf{W}^l_{X,\perp} \cdot \sqrt{D_{\text{in}}}\right) \cdot \left(\mathbf{1}_{D_{\text{out}} > D_{\text{in}}} \sqrt{D_{\text{out}}/D_{\text{in}}} + \mathbf{1}_{D_{\text{out}} \le D_{\text{in}}}\right) \cdot \sigma_{\mathbf{W}_X} + \mu_{\mathbf{W}_X}, \quad \mathbf{b}^l_X = \mathbf{0} \tag{8}$$

The fixed-point closure is now explicit. Equation 5 evaluates the recurrence moments under the target distribution of U^l_{t-1}. Equation 8 constructs $\mathbf{W}^l_X$ such that U^l_t inherits this distribution. This realizes the stationarity $U^l_{t-1} \overset{d}{=} U^l_t$ of (A2) and closes the temporal half. Each layer then fires at the target rate $p^l = p^{l-1}$. Under (A3)–(A4) its spikes are i.i.d. Bernoulli with a Binomial spike count, the input that (A5) assumes for the next layer. This realizes $U^{l-1} \overset{d}{=} U^l$ and closes the spatial half. The consistency proposition in Section B.1.2 supports that this closure is well-posed under (A1)–(A6). We call this scheme Closed-Form Stable (CFS). It is data-agnostic and requires only the architecture $(D_{\text{in}}, D_{\text{out}})$, the neuron hyperparameters $\{\theta^l, \beta^l, U^l_{\text{reset}}\}$ and the target spike rate p^l. Besides the default $p^l = 50\%$ of objective (B), any rate $p^l \in (0, 1)$ follows from the analogous general-rate form in Section B.1.3.

3 Experiments

3.1 Signal propagation analysis in simulation

Experimental setup. We verify the closed form of Section 2.3 under controlled input statistics, first across depth against the data-agnostic and data-driven baselines of Section 1.1 (details in Section A.1) and then across neuron models, populations and architectures. The base network stacks 30 layers of 100 LIF neurons with $\theta = 1$ and $\beta = 0.9$, each initialized for a target rate $p = 15$–85% ($p = 50\%$ under Objective B, Equation 3). A binary encoder feeds it i.i.d. Bernoulli(p) spikes, so the first layer receives exactly the input of assumption (A5) (full setup in Section C.1):

$$X_t^1 = S_t^{\text{enc}} \in \mathbb{B}^{(D_{\text{in}}^{\text{enc}},1)} \sim \text{Ber}(p)^{\otimes D_{\text{in}}^{\text{enc}}} \;\Rightarrow\; \|X_t^1\|_1 \sim \text{Bin}(D_{\text{in}}^{\text{enc}}, p). \tag{9}$$

Metrics are computed per neuron m over its $|\mathcal{B}||\mathcal{T}|$ observations, samples $b \in \mathcal{B}$ and time steps $t \in \mathcal{T}$ after a warm-up, and then averaged over the $M = D_{\text{out}}$ neurons of layer l:

$$\bar{p}^l = \tfrac{1}{M|\mathcal{B}||\mathcal{T}|} \textstyle\sum_{m,b,t} S^l_{t,b,m}, \qquad \bar{\sigma}^2_{U^l} = \tfrac{1}{M} \textstyle\sum_m \tfrac{1}{|\mathcal{B}||\mathcal{T}|} \sum_{b,t} \left(U^l_{t,b,m} - \hat{\mu}_{U^l_m}\right)^2, \tag{10}$$

with $\hat{\mu}_{U^l_m}$ the mean potential of neuron m. Pooling all observations around the layer mean $\hat{\mu}_{U^l}$ adds the spread of the neurons' means, so that $\frac{1}{M|\mathcal{B}||\mathcal{T}|} \sum_{m,b,t}(U^l_{t,b,m} - \hat{\mu}_{U^l})^2 \geq \bar{\sigma}^2_{U^l}$. Hence, we separate this spread from neuron-level precision to verify the latter in isolation.

Layer-level and neuron-level statistics. Figure 1 traces one seed through all 30 layers at $p = 50\%$, and Table 2(a) reports layers 1, 5 and 30 at $p = 15$–85%, including $p = 15\%$, the default operating point of Micheli et al. [2025][1]; extended results are in Section C.3.1. Both representations show individual neuron statistics alongside the layer-wise mean, as dots in Figure 1 and as the spread in Table 2. These results demonstrate how severe neuron-specific deviations, including dead ($\bar{p}^l_m = 0$) and saturated ($\bar{p}^l_m = 1$) neurons, can be concealed by layer-wise statistics. Across the rate grid, no alternative holds the target with depth. Five fall silent by layer 30 at every rate. Ding et al. [2025] and Che et al. [2025] settle near 2% and 35% regardless of the target, the latter by splitting its neurons between silence and saturation at a membrane variance far above target. Rossbroich et al. [2022] stays active at about a quarter of the target. At the $p = 15\%$ operating point these findings are in line with the comparison by Micheli et al. [2025]. Their own initialization stays active at this rate but drifts above target, to 20.4% at layer 30 with a spread across neurons exceeding its mean. At our default $p = 50\%$ it falls to 0.3%. Our initialization, the closed form of Equation 8 evaluated at the respective target rate, holds both targets per neuron at every reported layer and rate: within 1.6 percentage points (pp) of the target rate, with a spread across neurons of at most 0.5 pp, the resolution of the estimator, and a membrane variance of 0.91–1.06. Crucially, uniform target fulfillment and orthogonal weights respectively keep neurons approximately identically distributed and nearly uncorrelated, so that the spikes of each layer remain approximately i.i.d. Bernoulli with a Binomial spike count, the input that Equation 4 assumes for the next layer. The per-neuron targets holding through all 30 layers are consistent with this strict assumption being met layer after layer: no neuron falls silent or saturates, and representational collapse is avoided.

Scope of the closed form. Table 2(b) applies the same closed form beyond the feed-forward network of hard-reset LIF neurons, one element at a time to: integrate-and-fire neurons, memoryless binary units and the soft reset; heterogeneous neuron populations (θ^l_j, β^l_j); and residual (S^{l-2}_t) and recurrent (S^l_{t-1}) connections and convolutional layers (respective derivations and extended results in Appendix B and Section C.3). Across $p = 15$–85%, every case keeps the spread across neurons of the deepest layer within 0.9 pp, so the per-neuron agreement of the base case carries over to each. At $p = 50\%$, every case also holds the target rate within 1.8 pp and the membrane variance at 0.96–1.00; away from $p = 50\%$, the mean drifts with depth from an on-target first layer, most for the binary unit (75.7% at $p = 65\%$) and the soft reset (95.8% at $p = 85\%$). The initialization also tolerates violations of the i.i.d. Bernoulli input assumption. In a sensitivity analysis, every layer is initialized for an input rate of 50% while the encoder violates this assumption in one of three ways: (i) the encoder delivers 25% or 75% instead, yet the deepest layer still settles at 51.2% and 51.3%; (ii) the input neurons fire at individual rates instead of a common one and agree with 50% only in their mean, yet even with a standard deviation of 25 pp the deepest layer again settles at 51.3%; and (iii) the input neurons keep the 50% rate but fire in correlation instead of independently, yet even a strong pairwise spike correlation of 0.5 shifts the deepest layer by only 3.4 pp to 46.6%. Complete results underscoring the robustness displayed in this sensitivity analysis are given in Section C.3.3.

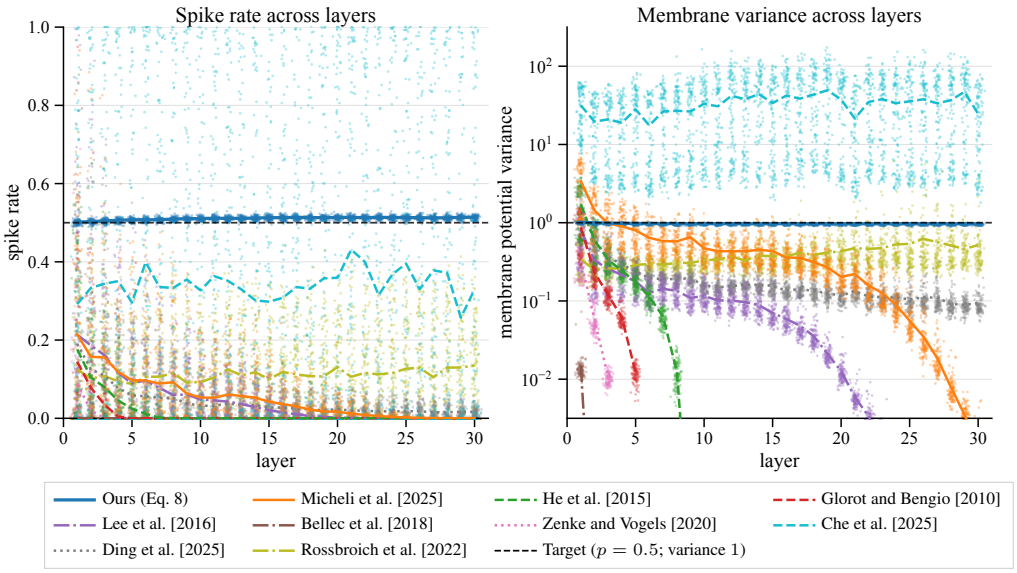

Figure 1: **Signal propagation across deep networks.** Spike rate (left) and membrane variance (right, log axis clipped to $[10^{-2.5}, 10^{2.5}]$) across 30 layers, one seed: dots are neurons, colored lines layer means, black dashed lines the targets (50%, unit variance); lines end where layers fall silent or reach the lower bound. Only our method (blue) holds both targets for every neuron across depth.

Table 2: **Target statistics across depth, initialization methods and scope.** Mean over 10 seeds at initialization. Each cell gives the spike rate $\bar{p}^l$ in % (target p) and, in parentheses, the membrane-potential variance $\bar{\sigma}^2_{U^l}$ (target 1), each as mean ± standard deviation across the neurons of layer l. **(a)** Initialization methods on the base network at layers 1, 5 and 30. Bold marks the least deviation from target per column and layer. Five methods are not shown individually, as they fall silent by layer 30 at every p (at most 0.1%). Their block gives the highest rate. All methods, layers and rates: Section C.3.1. **(b)** Our initialization at the deepest layer, with one aspect of the base case generalized at a time. Each block header names the appendix sections with the derivation and extended results.

	$p = 15\%$	$p = 35\%$	$p = 50\%$	$p = 65\%$	$p = 85\%$
(a) Initialization methods across depth, base case (feed-forward network of hard-reset LIF neurons)					
Micheli et al. [2025]					
Layer 1	12.7±12.7 (3.07±1.43)	17.9±20.5 (3.13±1.73)	22.0±26.0 (3.15±1.86)	26.7±31.6 (3.19±1.99)	35.1±39.9 (3.32±2.29)
Layer 5	16.5±20.3 (3.18±2.15)	11.1±14.6 (**1.00**±0.52)	11.1±14.8 (0.82±0.39)	13.4±17.8 (0.96±0.47)	21.9±28.0 (2.17±1.34)
Layer 30	20.4±24.4 (2.72±1.82)	2.5±2.9 (0.33±0.21)	0.3±0.4 (0.06±0.03)	3.0±3.5 (0.37±0.22)	20.4±24.4 (2.72±1.80)
Che et al. [2025]					
Layer 1	18.4±17.8 (15.5±7.1)	25.1±27.3 (28.3±15.5)	28.9±32.1 (31.6±18.4)	32.5±36.3 (29.1±17.8)	37.5±41.7 (17.2±11.4)
Layer 5	31.9±35.7 (32.3±25.9)	33.1±37.3 (26.1±21.2)	33.7±38.0 (22.9±18.7)	34.3±38.5 (20.0±16.3)	35.2±39.3 (16.0±13.0)
Layer 30	35.0±38.0 (25.7±22.9)	**35.0**±38.0 (25.6±22.9)	35.0±38.0 (25.7±23.0)	35.0±38.0 (25.8±23.0)	35.0±38.0 (25.9±23.3)
Ding et al. [2025]					
Layer 1	9.0±8.8 (0.79±0.34)	15.5±17.5 (1.41±0.77)	19.6±23.1 (1.57±0.91)	23.6±28.3 (1.45±0.88)	29.0±35.1 (0.87±0.55)
Layer 5	4.8±5.6 (0.22±0.05)	6.2±7.6 (0.24±0.06)	6.8±8.6 (0.25±0.06)	7.4±9.4 (0.25±0.06)	8.1±10.5 (0.25±0.05)
Layer 30	1.9±1.7 (0.11±0.04)	1.9±1.7 (0.11±0.03)	2.0±1.8 (0.12±0.04)	2.0±1.8 (0.12±0.04)	2.0±1.8 (0.12±0.04)
Rossbroich et al. [2022]					
Layer 1	5.4±6.3 (0.57±0.22)	10.0±11.4 (0.41±0.18)	12.9±14.4 (0.32±0.14)	15.8±17.2 (0.24±0.09)	20.2±20.8 (0.13±0.02)
Layer 5	4.1±5.3 (0.48±0.20)	7.6±8.6 (0.32±0.08)	9.8±10.3 (0.28±0.05)	12.0±11.7 (0.25±0.05)	16.0±13.6 (0.24±0.05)
Layer 30	3.6±4.0 (0.80±0.77)	9.6±9.3 (0.68±0.75)	13.3±11.7 (0.61±0.65)	16.6±13.1 (0.55±0.52)	21.3±14.1 (0.53±0.39)
He et al. [2015], Glorot and Bengio [2010], Zenke and Vogels [2020], Bellec et al. [2018], Lee et al. [2016]					
Layer 1	≤ 9.4	≤ 16.5	≤ 20.7	≤ 24.6	≤ 29.8
Layer 5	≤ 6.2	≤ 9.2	≤ 10.5	≤ 11.4	≤ 12.5
Layer 30	≤ 0.0	≤ 0.0	≤ 0.0	≤ 0.0	≤ 0.1
Ours					
Layer 1	**15.3**±0.4 (**1.06**±0.03)	**35.4**±0.5 (**1.02**±0.02)	**50.2**±0.5 (**1.01**±0.02)	**65.1**±0.4 (**1.00**±0.02)	**85.1**±0.3 (**1.00**±0.01)
Layer 5	**15.2**±0.4 (**0.96**±0.03)	**36.1**±0.5 (0.99±0.02)	**50.7**±0.5 (**0.98**±0.02)	**65.3**±0.5 (**0.99**±0.02)	**85.9**±0.4 (**0.94**±0.02)
Layer 30	**14.4**±0.4 (**0.91**±0.03)	36.6±0.5 (**0.98**±0.02)	**51.3**±0.5 (**0.97**±0.02)	**66.4**±0.4 (**0.98**±0.02)	**83.8**±0.4 (**1.03**±0.02)
(b) Our initialization beyond the base case, deepest layer					
Neuron model — derived in Section B.5.1 and Section B.5.2; extended results in Section C.3.4					
Integrate-and-fire, $\beta = 1$	19.7±0.4 (0.98±0.03)	37.9±0.5 (0.99±0.03)	51.8±0.5 (0.96±0.02)	66.1±0.4 (0.98±0.02)	82.3±0.4 (1.12±0.03)
Binary unit, $\beta = 0$	10.8±0.4 (0.72±0.02)	33.6±0.5 (0.97±0.02)	49.9±0.5 (1.00±0.02)	75.7±0.4 (0.82±0.01)	91.3±0.3 (0.59±0.02)
Soft reset, $\beta = 0.9$	11.9±0.3 (0.77±0.02)	34.3±0.4 (0.93±0.02)	49.9±0.4 (0.96±0.02)	65.7±0.4 (0.94±0.02)	95.8±0.2 (0.39±0.01)
Population — derived in Section B.6; extended results in Section C.3.5					
Heterogeneous θ, β	11.6±0.7 (0.79±0.05)	34.9±0.6 (0.99±0.03)	50.4±0.6 (0.99±0.02)	68.9±0.6 (0.94±0.03)	88.3±0.8 (0.77±0.04)
Architecture — derived in Section B.7.1 and Section B.7.2; extended results in Section C.3.6 and Section C.3.7					
Residual connection	14.5±0.4 (0.92±0.06)	36.7±0.7 (0.98±0.06)	51.4±0.9 (0.97±0.05)	66.4±0.9 (0.98±0.05)	83.9±0.6 (1.02±0.05)
Recurrent connection	15.2±0.4 (0.95±0.05)	36.7±0.5 (0.98±0.04)	51.3±0.5 (0.97±0.02)	65.9±0.5 (0.98±0.02)	87.7±0.3 (0.80±0.02)
CNN	15.2±0.4 (0.99±0.04)	35.4±0.5 (1.00±0.04)	50.2±0.7 (0.98±0.03)	65.0±0.7 (0.99±0.03)	85.0±0.5 (1.00±0.02)

3.2 Empirical benchmark task performance

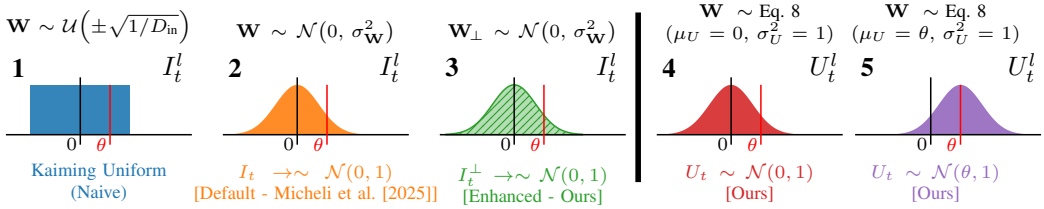

Figure 2: **Schematic of the five initialization configurations.** Each panel shows a pictogram of the distribution set by one of the initialization configurations of the empirical evaluation (Section 3.2), relative to the origin 0 and the threshold θ (red). Configurations (1)–(3) shape the input current I_t^l: (1) the *naive* Kaiming baseline, (2) the *default* method of Micheli et al. [2025], and (3) its *enhanced* variant with orthogonal weights $\mathbf{W}_\perp$; (4)–(5) shape the membrane potential U_t^l through Equation 8. The vertical divider is introduced to emphasize that distributions imposed on I_t^l do not in general translate into equivalent properties for U_t^l, as verified in simulation (Section C.3.2).

Experimental setup. Of the alternatives in Section 3.1, we carry forward Micheli et al. [2025]: it is the closed form this work extends, and its collapse with depth is limited at its default operating point of about 15% (Table 2). Around it, a five-configuration progression (Figure 2) isolates the design choices between this established method and ours: (1) the *naive* Kaiming baseline (the standard initialization for ANNs in PyTorch [Paszke et al., 2019]); (2) the *default* input-current-focused method of Micheli et al. [2025] ($I_t \sim \mathcal{N}$); (3) an *enhanced* variant of (2) with orthogonal weights; and (4)–(5) our initialization with zero mean ($U_t \sim \mathcal{N}(0,1)$) and threshold-aligned ($U_t \sim \mathcal{N}(\theta,1)$). Configurations (1)–(3) shape only the input-current statistics, whereas (4)–(5) shape the membrane potential directly through Equation 8, (4) at the operating point $p \approx 15\%$ of (2)–(3) and (5) at the maximum-entropy rate $p = 50\%$. We evaluate the progression on 14 tasks from three learning paradigms: eight UEA multivariate time-series classification datasets [Bagnall et al., 2018]; three neuromorphic classification datasets (SHD [Cramer et al., 2022], DVS Gesture [Amir et al., 2017] and ECG [Laguna et al., 1997]); and three continuous-control reinforcement-learning (RL) environments trained with PPO [Schulman et al., 2017] (Inverted Pendulum, Inverted Double Pendulum and Hopper from Brax [Freeman et al., 2021]). Every network stacks LIF layers of width $\in \{32, 64, 128\}$ and depth $\in \{2, 3, 4, 5\}$ with a fixed threshold $\theta = 1$, a per-neuron learnable leak initialized to $\beta^l = 0.9$, zero-initialized biases, and a single batch-normalization module before the first LIF layer. Each of the 840 cells $\{\text{task}, \text{configuration}, \text{width}, \text{depth}\}$ is trained with 20 seeds; performance is the test accuracy for classification and, for RL, the evaluation reward normalized by a per-environment ceiling, each averaged over the last 10% of evaluation points (full setup in Section C.2). Their signal propagation across $\{\text{width}, \text{depth}\}$ cells is verified in simulation (Section C.3.2).

Cross-paradigm performance. Table 3 reports average absolute performance and ranking across the 12 (3×4) SNN architectures per task. Configuration (5) ranks first in every paradigm (Table 3, Figure 3): paradigm-mean ranks $2.07/1.58/1.97$ on UEA / neuromorphic / RL (1.88 overall, vs. 2.68 for the next-best configuration). Configuration (5) ranks above each alternative in 68–85% of the cells, and its lead over each is significant in a paired permutation test (one-sided $p \leq 0.009$, Table 3). Per row, it holds the best mean cell rank on 11 of 14 task rows, with RacketSports, UWaveGestureLibrary and ECG as the three exceptions. Configuration (4) applies the same closed form at the $\approx 15\%$ operating point assumed by (2)–(3). It ranks behind both on UEA and neuromorphic (3.67 and 4.22 against 2.62–2.90 and 2.36–2.39) and ahead of (2) and level with (3) on RL (3.03 against 3.17 and 3.03). Section 3.3 revisits the divergence of (4) at the shared operating point.

Depth and width scaling. Figure 3 disaggregates the results by width and depth. Every configuration gains with width and loses from depth 2 to 5 (top). From width 32 to 128 the mean score of configuration (5) rises from 64.9% to 80.7%, that of the alternatives from 51.3–56.1% to 69.8–76.4%. From depth 2 to 5 it falls from 80.0% to 65.1% for configuration (5) and from 72.4–74.7% to 40.7–55.1% for the alternatives. Configuration (5) ranks first at every width and depth (bottom). Its lead narrows with width, from a mean rank of 1.64 to 2.26, and widens with depth, to 1.42 at depth 5. It thus limits the depth-induced drop but does not eliminate it; we return to this in Section 3.3. Per-paradigm results are in Section C.4.

Table 3: **Cross-paradigm task performance.** Each task row presents the performance metric per configuration averaged over the 12 (width, depth) cells and 20 seeds: test accuracy for UEA and neuromorphic tasks and normalized reward for RL. After ± follows the seed standard deviation averaged over the cells, and in parentheses the mean rank within the cells (lower is better). Bold marks the best mean per row. The summary rows average the cell ranks over a paradigm as well as across the three paradigms; brackets give 95% bootstrap confidence intervals over the 20 seeds. The last three rows count the cells in which configuration (5) ranks above / below each alternative and test its lead in score and in rank for significance with a paired permutation (sign-flip) test over the 14 tasks (one-sided p-value; method: Section C.2).

Task / Environment	naive	I_t default	$I_t^{\perp}$ enhanced	$U_t \sim \mathcal{N}(0,1)$	$U_t \sim \mathcal{N}(\theta,1)$
UEA multivariate time series — test set accuracy mean ± std (%) / mean cell rank					
BasicMotions	78.96±12.08 (4.58)	94.09±8.94 (2.17)	93.88±8.66 (2.50)	90.96±10.04 (3.33)	**95.57**±9.04 (2.08)
Cricket	75.32±6.95 (4.50)	90.76±4.70 (2.92)	91.60±4.87 (2.75)	90.70±4.19 (3.42)	**94.40**±2.90 (1.42)
Epilepsy	92.04±3.13 (1.92)	89.74±6.44 (3.25)	89.58±6.29 (3.50)	87.62±4.57 (4.75)	**93.00**±3.37 (1.58)
RacketSports	60.17±2.80 (4.75)	72.67±3.99 (1.92)	**73.25**±3.58 (1.50)	66.63±2.41 (4.08)	69.85±2.97 (2.75)
NATOPS	63.36±4.93 (4.92)	72.67±4.86 (2.75)	72.73±5.11 (2.92)	72.29±3.26 (3.25)	**75.51**±2.78 (1.17)
UWaveGestureLibrary	70.59±5.28 (2.33)	71.40±4.26 (2.50)	**72.31**±3.65 (1.92)	66.63±3.86 (4.25)	67.36±3.20 (4.00)
ArticularyWordRecognition	74.26±4.79 (3.83)	86.15±3.12 (4.08)	87.63±2.62 (2.92)	89.43±2.06 (2.58)	**90.56**±1.83 (1.58)
LSST	20.74±1.76 (2.75)	19.68±2.24 (3.58)	20.08±2.79 (3.00)	19.30±2.71 (3.67)	**28.51**±2.33 (2.00)
paradigm mean rank ↓	3.70 [3.58, 3.76]	2.90 [2.79, 3.06]	2.62 [2.49, 2.76]	3.67 [3.51, 3.74]	**2.07** [2.02, 2.19]
Neuromorphic classification — test set accuracy mean ± std (%) / mean cell rank					
SHD	61.11±2.79 (4.75)	67.21±2.07 (2.42)	67.28±1.91 (2.58)	65.10±1.95 (4.08)	**71.32**±1.82 (1.17)
DVS	57.72±3.24 (5.00)	66.44±2.01 (3.00)	66.93±1.76 (2.25)	64.28±1.23 (3.75)	**70.65**±1.60 (1.00)
ECG	76.42±0.78 (3.58)	**77.56**±0.50 (1.67)	77.47±0.53 (2.33)	75.39±0.47 (4.83)	77.38±0.49 (2.58)
paradigm mean rank ↓	4.44 [4.33, 4.50]	2.36 [2.19, 2.53]	2.39 [2.22, 2.56]	4.22 [4.11, 4.28]	**1.58** [1.50, 1.72]
Reinforcement learning — normalized reward of deterministic policy across 512 episodes mean ± std (%) / mean cell rank					
Inverted Pendulum	56.97±9.77 (4.50)	69.16±8.13 (3.42)	69.78±8.22 (2.92)	69.40±3.73 (2.08)	**86.42**±7.81 (2.08)
Inverted Double Pendulum	33.00±9.84 (3.92)	39.47±5.67 (2.75)	39.19±4.57 (3.33)	41.32±2.74 (2.92)	**66.92**±9.08 (2.08)
Hopper	55.41±17.49 (3.00)	53.08±12.42 (3.33)	54.37±13.04 (2.83)	52.50±13.60 (4.08)	**61.56**±14.30 (1.75)
paradigm mean rank ↓	3.81 [3.56, 4.00]	3.17 [2.97, 3.50]	3.03 [2.78, 3.28]	3.03 [2.75, 3.19]	**1.97** [1.75, 2.19]
overall mean rank ↓	3.98 [3.88, 4.05]	2.81 [2.73, 2.95]	2.68 [2.57, 2.79]	3.64 [3.52, 3.69]	**1.88** [1.81, 1.98]
cells won / lost by (5)	135/33	122/46	114/54	142/26	–
significance p vs. (5): score	2.4×10^{-4}	0.003	0.005	6.1×10^{-5}	–
significance p vs. (5): rank	3.1×10^{-4}	0.009	0.008	1.2×10^{-4}	–

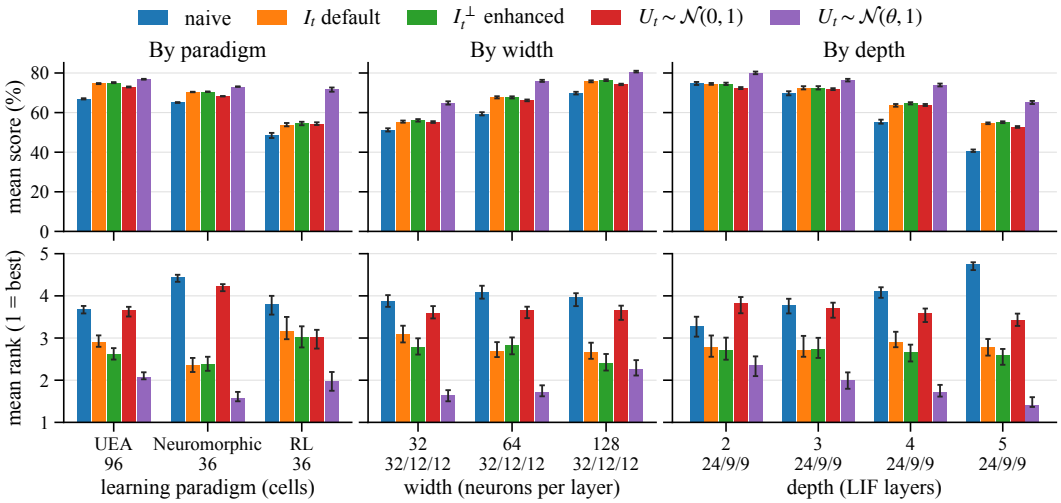

Figure 3: **Mean score (top) and mean cell rank (bottom) of the configurations by paradigm, width and depth.** Scores and ranks are those of Table 3 (1 = best rank). By paradigm, each bar averages the cells of the paradigm. By width and by depth, it averages the cells of that width or depth within each paradigm, with the paradigms weighted equally. Below each group are the cell counts as UEA/neuromorphic/RL. Error bars are 95% bootstrap confidence intervals over the 20 seeds. Rank intervals are asymmetric near ranks 1 and 5, because seed noise can only move a rank toward the middle.

3.3 Discussion

The proposed closed-form, data-agnostic initialization holds the intended signal-propagation statistics in simulation and is validated empirically across three learning paradigms. Configurations (4) and (5) apply Closed-Form Stable (CFS) at two operating points and thereby isolate the effect of the operating point on training. At unit variance the threshold-centered target of (5) yields both the maximum-entropy rate and the maximal expected surrogate gradient (Equation 3) at initialization. Configurations (2)–(3) reach the operating point of (4) only as a layer mean with about three times its membrane variance (Table 2, Section C.3.2), yet outrank it overall. We hypothesize that this larger membrane variance strengthens the backward signal at this rate. Performance falls from depth 2 to 5 for every configuration (Figure 3, top). This includes (4) and (5), whose forward statistics stay on target at every depth $\in \{2, 3, 4, 5\}$ in simulation (Section C.3.2). Collectively, the properties at initialization and empirical observations point beyond the forward pass to gradient propagation backward through deep recurrent stacks. Initialization for SNNs thus separates into a forward problem that the present method addresses in closed form and a backward problem of gradient flow through the same recurrent stack, which the fixed forward statistics allow to be posed in isolation as a well-defined open question.

Limitations. (i) Task-based training is evaluated on feed-forward networks and one convolutional network (Section C.4.4); the remaining extensions are verified at initialization, and neuron models outside the LIF family and more complex architectures such as attention-based ones are not derived. (ii) The closed form does not account for task-specific dynamics, such as the temporal correlation of real inputs (A5) and an optimal initial rate beyond the two operating points. (iii) The closed form sets the initial state only; training trajectories, the spike rates of trained networks, the surrogate-gradient choice and the effect of the membrane variance on gradient flow are not examined.

Significance. To our knowledge, CFS is the first analytical, data-agnostic initialization for deep SNNs that jointly handles spatial, temporal, and reset dynamics in closed-form. Its precision stems from joint stationarity across time and depth, which no alternative tested attains. This stationarity is a prerequisite for any controlled initialization. It enables the method to initialize at any target spike rate with neuron-level precision (Table 2). This control makes the operating point an experimental variable. Configurations (4) and (5) use the method to compare competing operating points, a comparison that initializations lacking stationarity or meeting a rate only as a layer mean cannot isolate. Within the scope of our benchmarks, this comparison favors the maximum-entropy rate of configuration (5) as a principled default when no informed spike-rate prior is available. Fundamentally, the method constrains the moments of the input current each neuron receives rather than the operator that produces it. Consequently, it encapsulates a range of SNN configurations in a single closed form, without data or tuning. It spans the neuron's memory — from integrate-and-fire through LIF to memoryless binary neurons — and, orthogonally, the reset rule, hard or soft. It is equally agnostic to the population, from homogeneous to heterogeneous units with per-neuron thresholds and leak factors, and to the architecture: feed-forward, residual, recurrent, and convolutional. This coverage allows direct use in mainstream SNN frameworks [Eshraghian et al., 2023, Pehle and Pedersen, 2021, Fang et al., 2023], contributing toward standardized initialization practices in SNN optimization.

4 Conclusion

We present a closed-form, data-agnostic initialization for spiking neural networks that jointly handles spatial signal propagation, temporal recurrence, and reset, which prior data-agnostic methods address only in part. Our Closed-Form Stable (CFS) method sets a spike rate for each neuron that is jointly stationary across time and depth and scale-invariant in width. In simulation it holds any target spike rate for every neuron through 30 layers, where alternative initializations fall silent or stay off target. Empirically, at the maximum-entropy default and without per-paradigm tuning, this initialization ranks first in time-series classification, neuromorphic classification, and continuous-control reinforcement learning. While results indicate that initialization alone does not improve performance with increasing network depth, it solves the forward signal-propagation problem in closed form, leaving gradient propagation through deep recurrent stacks as a well-defined open question. Overall, the proposed method's precise control of initial network dynamics across neuron models and architectures contributes to the standardization of initialization practices in SNN optimization.

Acknowledgments and Disclosure of Funding

Funding. This project was financed by the Dutch Research Council (NWO) under grant number 20663 of the VICI personal grant programme.

Competing interests. The authors declare that they have no financial relationships with any entities that could potentially be perceived to influence the work reported in this paper.

References

Guillaume Bellec, Darjan Salaj, Anand Subramoney, Robert Legenstein, and Wolfgang Maass. Long short-term memory and Learning-to-learn in networks of spiking neurons. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/hash/c203d8a151612acf12457e4d67635a95-Abstract.html.

Friedemann Zenke and Surya Ganguli. SuperSpike: Supervised Learning in Multilayer Spiking Neural Networks. *Neural Computation*, 30(6):1514–1541, June 2018. ISSN 0899-7667, 1530-888X. doi: 10.1162/neco_a_01086. URL https://direct.mit.edu/neco/article/30/6/1514-1541/8378.

Emre 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, November 2019. doi: 10.1109/msp.2019.2931595.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Christian Pehle and Jens Egholm Pedersen. Norse - A deep learning library for spiking neural networks, January 2021. URL https://doi.org/10.5281/zenodo.4422025.

Jason K Eshraghian, Max Ward, Emre O Neftci, Xinxin Wang, Gregor Lenz, Girish Dwivedi, Mohammed Bennamoun, Doo Seok Jeong, and Wei D Lu. Training spiking neural networks using lessons from deep learning. *Proceedings of the IEEE*, 111(9):1016–1054, 2023.

Wei Fang, Yanqi Chen, Jianhao Ding, Zhaofei Yu, Timothée Masquelier, Ding Chen, Liwei Huang, Huihui Zhou, Guoqi Li, and Yonghong Tian. SpikingJelly: An open-source machine learning infrastructure platform for spike-based intelligence. *Science Advances*, 9(40):eadi1480, October 2023. ISSN 2375-2548. doi: 10.1126/sciadv.adi1480. URL https://www.science.org/doi/10.1126/sciadv.adi1480.

Aurora Micheli, Olaf Booij, Jan van Gemert, and Nergis Tömen. Deep activity propagation via weight initialization in spiking neural networks. In *2025 Neuro Inspired Computational Elements (NICE)*, pages 1–9. IEEE, 2025.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, March 2010. URL https://proceedings.mlr.press/v9/glorot10a.html.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1026–1034, December 2015. doi: 10.1109/ICCV.2015.123. URL https://ieeexplore.ieee.org/document/7410480.

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. TensorFlow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

Nicolas Perez-Nieves and Dan F. M. Goodman. Spiking Network Initialisation and Firing Rate Collapse, May 2023. URL http://arxiv.org/abs/2305.08879.

Julian Rossbroich, Julia Gygax, and Friedemann Zenke. Fluctuation-driven initialization for spiking neural network training. *Neuromorphic Computing and Engineering*, 2(4):044016, December 2022. ISSN 2634-4386. doi: 10.1088/2634-4386/ac97bb. URL https://dx.doi.org/10.1088/2634-4386/ac97bb.

Justus Huebotter, Pablo Lanillos, Marcel van Gerven, and Serge Thill. Spiking Neural Networks for Continuous Control via End-to-End Model-Based Learning, September 2025. URL http://arxiv.org/abs/2509.05356.

Luca Herranz-Celotti and Jean Rouat. Stabilizing Spiking Neuron Training, January 2024. URL http://arxiv.org/abs/2202.00282.

Peiyu Liu, Jianhao Ding, and Zhaofei Yu. Training Deep Normalization-Free Spiking Neural Networks with Lateral Inhibition, September 2025. URL http://arxiv.org/abs/2509.23253.

Friedemann Zenke and Tim P. Vogels. The remarkable robustness of surrogate gradient learning for instilling complex function in spiking neural networks. *bioRxiv*, June 2020. doi: 10.1101/2020.06.29.176925.

Jun Haeng Lee, Tobi Delbruck, and Michael Pfeiffer. Training Deep Spiking Neural Networks Using Backpropagation. *Frontiers in Neuroscience*, 10, November 2016. ISSN 1662-453X. doi: 10.3389/fnins.2016.00508. URL https://www.frontiersin.org/journals/neuroscience/articles/10.3389/fnins.2016.00508/full.

Da Chang, Deliang Wang, and Xiao Yang. IKUN: Initialization to Keep snn training and generalization great with sUrrogate-stable variaNce, November 2024. URL http://arxiv.org/abs/2411.18250.

Kaiwei Che, Wei Fang, Zhengyu Ma, Yifan Huang, Peng Xue, Li Yuan, Timothée Masquelier, and Yonghong Tian. Efficiently Training Time-to-First-Spike Spiking Neural Networks from Scratch, March 2025. URL http://arxiv.org/abs/2410.23619.

Jianhao Ding, Jiyuan Zhang, Tiejun Huang, Jian K Liu, and Zhaofei Yu. Assisting training of deep spiking neural networks with parameter initialization. *IEEE Transactions on Neural Networks and Learning Systems*, 36(8):15015–15028, 2025.

Anthony Bagnall, Hoang Anh Dau, Jason Lines, Michael Flynn, James Large, Aaron Bostrom, Paul Southam, and Eamonn Keogh. The UEA multivariate time series classification archive, 2018. *arXiv preprint arXiv:1811.00075*, 2018.

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.

Arnon Amir, Brian Taba, David Berg, Timothy Melano, Jeffrey McKinstry, Carmelo Di Nolfo, Tapan Nayak, Alexander Andreopoulos, Guillaume Garreau, Marcela Mendoza, et al. A low power, fully event-based gesture recognition system. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7243–7252, 2017.

Pablo Laguna, Roger G Mark, A Goldberg, and George B Moody. A database for evaluation of algorithms for measurement of QT and other waveform intervals in the ECG. In *Computers in Cardiology 1997*, pages 673–676. IEEE, 1997.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms, 2017. URL http://arxiv.org/abs/1707.06347.

C. Daniel Freeman, Erik Frey, Anton Raichuk, Sertan Girgin, Igor Mordatch, and Olivier Bachem. Brax - a differentiable physics engine for large scale rigid body simulation, 2021. URL http://github.com/google/brax.

Chen Li and Steve Furber. Towards biologically-plausible neuron models and firing rates in high-performance deep spiking neural networks. In *International Conference on Neuromorphic Systems 2021*, pages 1–7, 2021.

Yufei Guo, Yuanpei Chen, Liwen Zhang, Xiaode Liu, Yinglei Wang, Xuhui Huang, and Zhe Ma. IM-Loss: information maximization loss for spiking neural networks. *Advances in neural information processing systems*, 35:156–166, 2022.

Nicolas Perez-Nieves, Vincent C. H. Leung, Pier Luigi Dragotti, and Dan F. M. Goodman. Neural heterogeneity promotes robust learning. *Nature Communications*, 12(1):5791, October 2021. ISSN 2041-1723. doi: 10.1038/s41467-021-26022-3. URL https://www.nature.com/articles/s41467-021-26022-3.

Yifan Hu, Lei Deng, Yujie Wu, Man Yao, and Guoqi Li. Advancing Spiking Neural Networks Toward Deep Residual Learning. *IEEE Transactions on Neural Networks and Learning Systems*, 36(2):2353–2367, February 2025. ISSN 2162-2388. doi: 10.1109/TNNLS.2024.3355393. URL https://ieeexplore.ieee.org/abstract/document/10428029.

PyTorch. torch.nn.init documentation, 2022. URL https://docs.pytorch.org/docs/stable/nn.init.html.

Jens Egholm Pedersen, Jörg Conradt, and Tony Lindeberg. Covariant spatio-temporal receptive fields for spiking neural networks. *Nature Communications*, 16(1):8231, 2025.

Matthew Middlehurst, Ali Ismail-Fawaz, Antoine Guillaume, Christopher Holder, David Guijo-Rubio, Guzal Bulatova, Leonidas Tsaprounis, Lukasz Mentel, Martin Walter, Patrick Schäfer, and Anthony Bagnall. aeon: a python toolkit for learning from time series. *Journal of Machine Learning Research*, 25(289):1–10, 2024. URL http://jmlr.org/papers/v25/23-1444.html.

Gregor Lenz, Kenneth Chaney, Sumit Bam Shrestha, Omar Oubari, Serge Picaud, and Guido Zarrella. Tonic: event-based datasets and transformations., July 2021. URL https://doi.org/10.5281/zenodo.5079802. Documentation available under https://tonic.readthedocs.io.

Bojian Yin, Federico Corradi, and Sander M Bohté. Accurate and efficient time-domain classification with adaptive spiking recurrent neural networks. *Nature Machine Intelligence*, 3(10):905–913, 2021.

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C Courville, and Marc Bellemare. Deep reinforcement learning at the edge of the statistical precipice. *Advances in neural information processing systems*, 34:29304–29320, 2021.

A Appendix - literature

A.1 Alternative initialization methods

For each prior-art initialization scheme cited in Section 1.1 we give the canonical equation that defines it together with a per-axis statement consistent with Table 1. Methods follow the table order—data-driven first, then data-agnostic—and the four axes (data-agnostic, spatial, recurrence, reset) are addressed in the same order within every entry. Axes outside this set are not discussed. Each entry closes by stating whether the simulation study includes the scheme, and the reason if it does not.

Notation. We adopt the methodology's notation throughout: D_{in} and D_{out} for the layer fan-in and fan-out, θ for the firing threshold, $\beta \in (0, 1)$ for the per-step membrane leak factor, p for the per-step Bernoulli spike rate, u for a realized membrane-potential value with mean μ_U and standard deviation σ_U, and $\mathrm{SG}(u; \theta)$ for the surrogate gradient. Method-specific symbols are introduced inline on first use; Table 5 at the end of this section records the mapping from every appendix symbol to the symbol used in the original source. Note that unless otherwise described in this appendix, the mean of (Gaussian) distributions as well as explicit bias parameters are set to zero.

Rossbroich et al. [2022]. The fluctuation-driven scheme targets a membrane-potential mean μ_U and standard deviation σ_U. It derives the weight moments for the membrane without spiking, for D_{in} inputs at a rate ν:

$$\bar{w} \;=\; \frac{\mu_U}{D_{\text{in}}\,\nu\,\bar{\varepsilon}}, \qquad \mathrm{Var}[w] \;=\; \frac{\sigma_U^2}{D_{\text{in}}\,\nu\,\hat{\varepsilon}} - \bar{w}^2, \tag{11}$$

where $\bar{\varepsilon}$ and $\hat{\varepsilon}$ are the integrals of the postsynaptic-potential kernel and of its square. For our neuron, $\bar{\varepsilon} = 1/(1-\beta)$ and $\hat{\varepsilon} = 1/(1-\beta^2)$. The rate ν is measured in a forward pass on the input data, which makes the method *not data-agnostic* (data-dependent). It addresses *spatial* propagation through the weight moments and *temporal recurrence* through this kernel of the membrane potential, whose integrals contain the leak β. The source recommends $\sigma_U = 1$ and uses $\mu_U = 0$ in its experiments. To evaluate the scheme at our target rates we use $\sigma_U = 1$ and $\mu_U = \theta - \Phi^{-1}(1-p)\,\sigma_U$. The *reset* is addressed only partially ($\sim^*$): the derivation rests on a no-spiking assumption. On the other hand, the data-dependent rate measurement does account for reset dynamics in the network, yet this measurement carries the resets of earlier layers only. We include this scheme in the simulation study (Table 2, Table 10).

Herranz-Celotti and Rouat [2024]. The recurrent weight is given a non-zero mean that drives the membrane potential toward the threshold, with the variance closing the fluctuation balance. With the upstream spike S_t^{l-1} identified as the source's post-activation variable (so its second moment reduces to p^{l-1} under the binary assumption),

$$\bar{w}_{\text{rec}} \;=\; \frac{(2-\beta)\,\theta}{D_{\text{rec}} - 1}, \qquad \mathrm{Var}[w_{\text{rec}}] \;=\; \frac{2\,p^{l-1}\,D_{\text{in}}}{D_{\text{rec}} - 1}\,\mathrm{Var}[w_{\text{in}}] \;-\; \tfrac{1}{2}\,\bar{w}_{\text{rec}}^2, \tag{12}$$

where D_{rec} is the dimension of the recurrent layer (i.e., the fan-in/out of the recurrent weight matrix) and w_{in} is the upstream weight matrix initialized separately. The closed form requires recurrent connections, since the upstream input statistics (p^{l-1} and $\mathrm{Var}[w_{\text{in}}]$) are treated as data-driven inputs to the equation and the recurrent weights are then derived to balance the fluctuation, making the method *not data-agnostic*. It addresses *spatial* propagation via the variance closure and *temporal recurrence* via the leak factor β and the recurrent mean. It addresses the *reset* mechanism via the hard-reset term of its membrane recursion. We do not include this scheme in the simulation study, whose base network is feed-forward: its conditions are stated for recurrent weights.

Liu et al. [2025]. The initialization balances excitation against inhibition in normalization-free SNNs with lateral inhibition. Its exponential weight rate and inhibitory gain follow from spike statistics estimated once on the first training batch, which makes the method *not data-agnostic*. It addresses *spatial* propagation by modeling the inputs of a layer as i.i.d. Bernoulli spikes. *Temporal recurrence* and *reset* do not enter the derivation. We do not include this scheme in the simulation study: it requires separate excitatory and inhibitory populations, which the base network does not have.

Glorot and Bengio [2010]. Xavier initialization sets the weight variance from the layer fan-in and fan-out so as to preserve unit pre-activation variance for sigmoid or tanh activations, in either a Gaussian or a uniform form:

$$w \sim \mathcal{N}\Big(0,\ \tfrac{2}{D_{\text{in}}+D_{\text{out}}}\Big), \qquad w \sim \mathcal{U}\Big(-\sqrt{\tfrac{6}{D_{\text{in}}+D_{\text{out}}}},\ \sqrt{\tfrac{6}{D_{\text{in}}+D_{\text{out}}}}\Big). \tag{13}$$

The scheme is *data-agnostic*. It does not address *spatial* propagation in the spiking sense—the derivation assumes continuous activations—nor *temporal recurrence*, nor *reset*; the LIF dynamics enter neither the variance derivation nor the assumed activation statistics. We include the source's uniform form in the simulation study (Table 2, Table 10).

He et al. [2015]. Kaiming initialization rescales Xavier for half-rectified activations, in either a Gaussian or a uniform form:

$$w \sim \mathcal{N}\Big(0,\ \tfrac{2}{D_{\text{in}}}\Big), \qquad w \sim \mathcal{U}\Big(-\sqrt{\tfrac{6}{D_{\text{in}}}},\ \sqrt{\tfrac{6}{D_{\text{in}}}}\Big). \tag{14}$$

Similar to the Xavier initialization, this scheme is *data-agnostic*. It does not model *spatial* binary-spike propagation, *temporal recurrence*, or *reset*. It is designed for a rectifier nonlinearity that sets negative inputs to zero and passes positive inputs unchanged; this zeroing below a threshold fixed at zero is the closest it comes to the dynamics of LIF neurons. We include the source's Gaussian form in the simulation study (Table 2, Table 10).

The default weight initialization of linear and convolutional layers in PyTorch [Paszke et al., 2019] is a special parameterization of the uniform form in Equation 14. The scheme is generalized to a leaky rectifier with negative slope a, for which the uniform bound carries the gain $\sqrt{2/(1+a^2)}$. PyTorch fixes $a = \sqrt{5}$, i.e., a gain of $1/\sqrt{3}$, to obtain this default uniform form:

$$w \sim \mathcal{U}\Big(-\sqrt{\tfrac{6}{(1+a^2)\,D_{\text{in}}}},\ \sqrt{\tfrac{6}{(1+a^2)\,D_{\text{in}}}}\Big) \overset{a=\sqrt{5}}{=} \mathcal{U}\Big(-\tfrac{1}{\sqrt{D_{\text{in}}}},\ \tfrac{1}{\sqrt{D_{\text{in}}}}\Big). \tag{15}$$

Setting $a = 0$ recovers the uniform form in Equation 14. We refer to Equation 15 with $a = \sqrt{5}$ as the *naive* initialization because the PyTorch backend applies it whenever no initialization procedure is specified by the practitioner. Its variance, $\mathrm{Var}[w] = 1/(3\,D_{\text{in}})$, is one sixth of that of the uniform form in Equation 14, and it is the form used for configuration (1) in Section 3.2.

Zenke and Vogels [2020]. The weights are drawn from a uniform distribution scaled by the fan-in,

$$w \sim \mathcal{U}\Big(-\tfrac{1}{\sqrt{D_{\text{in}}}},\ \tfrac{1}{\sqrt{D_{\text{in}}}}\Big), \tag{16}$$

which coincides with the *naive* PyTorch default of Equation 15. The paper's contribution is the empirical observation that a SNN with surrogate-gradient training is robust across surrogate shapes; it does not derive this choice and leaves optimal initialization as an open question. The scheme is *data-agnostic* and engages with SNN-specific *spatial* propagation through that empirical assessment. *Temporal recurrence* and *reset* are not modeled. We include this scheme in the simulation study (Table 2, Table 10).

Lee et al. [2016]. The scheme couples the per-layer threshold to the per-layer weight scale through a single hyperparameter $c_\theta > 1$:

$$\mathrm{Var}[w] \;=\; \frac{1}{D_{\text{in}}}, \qquad \theta \;=\; c_\theta\sqrt{\frac{3}{D_{\text{in}}}}. \tag{17}$$

The method is *data-agnostic*; the threshold–scale coupling makes the per-layer scale explicitly threshold-aware, which is the engagement with *spatial* propagation. *Temporal recurrence* and *reset* are not modeled. We set $c_\theta = 3$, the lower end of the range used by Lee et al. [2016]. They note that layers with fewer units require smaller values. Our layer width of 100 equals the smallest fully connected hidden layer in their experiments. This gives $\theta = 0.52$ for this method. We include this scheme in the simulation study (Table 2, Table 10).

Bellec et al. [2018]. The LSNN scheme samples weights with a conservative gain factor g (typically $g \in [0.1, 0.2]$):

$$\mathrm{Var}[w] \;=\; \frac{g^2}{D_{\mathrm{in}}}. \tag{18}$$

The scheme is *data-agnostic*; the conservative gain regulates the per-layer activation magnitude in the spiking forward pass, the engagement with *spatial* propagation. *Temporal recurrence* and *reset* do not appear in the variance. We include this scheme in the simulation study with $g = 0.1$ (Table 2, Table 10).

Chang et al. [2024]. The IKUN scheme folds the second moment of the surrogate gradient into the variance:

$$\mathrm{Var}[w] \;=\; \frac{c_{\mathrm{SG}}}{(D_{\mathrm{in}} + D_{\mathrm{out}})\,\sigma_X^2\,\mathbb{E}[\mathrm{SG}(u;\theta)^2]}, \tag{19}$$

where c_{SG} is a hyperparameter and σ_X is the input standard deviation. It is *data-agnostic*: the source assumes a standard normal input. The surrogate-gradient correction adapts the ANN-style variance balance to the spiking forward pass, the engagement with *spatial* propagation. *Temporal recurrence* and *reset* do not appear in the variance. We do not include this scheme in the simulation study. Its weight variance depends on a hyperparameter and on the second moment of the surrogate gradient, for which the available version of the paper states neither values nor closed forms. We could not locate a publicly available reference implementation, so we could not ensure a faithful re-implementation.

Micheli et al. [2025]. Adapting the Kaiming derivation to binary spikes, the scheme maintains unit membrane-potential variance through depth. It rests on two assumptions. First, the membrane potential is assumed standard normal. Each neuron then spikes with probability $p = \Phi^c(\theta) = 1 - \Phi(\theta)$. Second, the binary inputs are treated as i.i.d. Bernoulli(p). Their count then follows Binomial(D_{in}, p). This is only the special case of a multivariate Bernoulli distribution in which neurons are independent and identically distributed. The scheme does not enforce this condition. Under these assumptions, the canonical form sets the weight variance to:

$$\mathrm{Var}[w] \;=\; \frac{1}{D_{\mathrm{in}}\,p}. \tag{20}$$

The method is *data-agnostic* and addresses *spatial* propagation by treating activations as binary. It does not model *temporal recurrence*. It addresses the *reset* mechanism only partially ($\sim^*$) because the derivation omits the reset operator. The cited work acknowledges this omission. Based on a limited empirical examination, it argues that the omission does not materially compromise the initialization under soft reset.

We refer to Equation 20 as the pooled-variance form because it fixes the input variance pooled over both the weight draw and the inputs of a layer. Conditioning on the weights, the law of total variance decomposes this pooled variance into two components. For zero-mean weights, the first is the expected within-neuron variance $\mathbb{E}_{\mathbf{W}}[\mathrm{Var}(I_m \mid \mathbf{W})] = D_{\mathrm{in}}\,p(1-p)\,\mathrm{Var}[w] = 1-p$. The second is the between-neuron variance of the conditional means $\mathrm{Var}_{\mathbf{W}}(\mathbb{E}[I_m \mid \mathbf{W}]) = D_{\mathrm{in}}\,p^2\,\mathrm{Var}[w] = p$. The pooled-variance form thus normalizes the sum of both components, which leaves each neuron a variance of $1 - p$. The per-neuron form $\mathrm{Var}[w] = 1/(D_{\mathrm{in}}\,p(1-p))$ instead normalizes the within-neuron component alone. Our objectives (Equation 3) are defined per neuron, as is the variance of our configurations (4)–(5) in Section 3.2. The per-neuron form therefore places the scheme on the same variance definition as our initialization. Comparisons with it then isolate the remaining design choices such as the mean target, the orthogonal construction and the treatment of recurrence and reset. We use the per-neuron form throughout this work, and both forms are compared in Section C.3.1. We include the per-neuron form in the simulation study (Table 2, Table 10).

Perez-Nieves and Goodman [2023]. To counter the progressive silencing of neurons over time, the variance is set to maintain a minimum sustained firing rate. After derivation it is determined by D_{in}, θ, the leak factor β, and an integral term $\mathcal{I}$ obtained once from the network structure via a root-finding step (no closed form independent of $\mathcal{I}$ is given). The method is *data-agnostic*: $\mathcal{I}$ follows from the forward-pass structure rather than from data. It addresses *spatial* propagation through the variance scale and *temporal recurrence* through both β and the explicit anti-collapse term. The

reset mechanism does not appear in the variance expression: although the spike-rate-collapse phenomenon they target arises from the joint interaction of recurrence and reset, the derivation captures only the time-domain consequence of that interaction without making the reset operator explicit. We do not include this scheme in the simulation study. Its weight scale follows from a numerical root-finding step on a continuous-time firing-rate expression, without a specification for discrete-time simulation. We could not locate a publicly available reference implementation, so we could not ensure a faithful re-implementation.

Che et al. [2025]. ETTFS scales the weight variance with the number of simulation time steps T, in a uniform form:

$$w \sim \mathcal{U}\Big(-\sqrt{\tfrac{3T}{D_{\text{in}}}},\ \sqrt{\tfrac{3T}{D_{\text{in}}}}\Big), \qquad \text{Var}[w] \ = \ \frac{T}{D_{\text{in}}}. \tag{21}$$

The method is *data-agnostic* and addresses *spatial* propagation through the per-layer variance scale. Its derivation averages the input-current variance over the T time steps of one spike per neuron, so T enters through the spike count and not through the *recurrent* membrane state. Because each neuron spikes only once, the *reset* is absent from it. We include this scheme in the simulation study with $T = 40$ on the same LIF neurons as every other method (Table 2, Table 10).

Ding et al. [2025]. The scheme considers a neuron $u[t] = \beta v[t] + \lambda\, i[t]$ with λ the scaling factor of the input current. It aligns the neuron's response curve with its slant asymptote, which gives a threshold- and leak-aware variance and a non-zero bias:

$$\text{Var}[w] \ = \ \frac{2\theta^2}{\lambda^2\, D_{\text{in}}}, \qquad b \ = \ \frac{\theta(1-\beta)}{2\lambda}, \tag{22}$$

where λ scales the input current of the neuron. Our LIF neuron adds its input current unscaled, for which the authors suggest $\lambda = 1$. This gives $\text{Var}[w] = 2\theta^2/D_{\text{in}}$ and $b = \theta(1-\beta)/2$. The method is *data-agnostic* and addresses *spatial* propagation through the threshold-scaled variance. *Temporal recurrence* and *reset* are addressed only partially ($\sim^*$): the response curve includes the leak and the reset to zero, but only for a constant input. We include this scheme in the simulation study (Table 2, Table 10).

Categorization and Symbol Mapping. Table 4 reproduces the four-axis categorization of Table 1. The non-trivial cells are the wave–asterisks on the reset cells for Rossbroich et al. [2022] and Micheli et al. [2025], and the wave–asterisks on the recurrence and reset cells for Ding et al. [2025]. The unannotated checkmark/cross pattern for Perez-Nieves and Goodman [2023] on recurrence and reset is also argumentatively related. The justification for each of these cases is given in the corresponding per-method paragraph above and is not duplicated here. Separately, Table 5 records the mapping between the symbols used in this appendix and those used in the original source material. This mapping covers every variable that appears in the per-method equations above even if symbols align between this work and the source.

Table 4: **Per-method positioning of weight-initialization schemes along the four axes of Section 1.1.** Cell values agree with Table 1. A wave ($\sim$) indicates partial coverage; an asterisk (*) flags a cell whose annotation is discussed in the per-method paragraph above. Methods above the rule are data-driven; methods below the rule are data-agnostic, with ours last.

Method	**Data-agnostic**	**Spatial**	**Recurrence**	**Reset**
Rossbroich et al. [2022]	×	✓	✓	$\sim^*$
Herranz-Celotti and Rouat [2024]	×	✓	✓	✓
Liu et al. [2025]	×	✓	×	×
Glorot and Bengio [2010]	✓	×	×	×
He et al. [2015]	✓	×	×	×
Zenke and Vogels [2020]	✓	✓	×	×
Lee et al. [2016]	✓	✓	×	×
Bellec et al. [2018]	✓	✓	×	×
Chang et al. [2024]	✓	✓	×	×
Micheli et al. [2025]	✓	✓	×	$\sim^*$
Perez-Nieves and Goodman [2023]	✓	✓	✓	×
Che et al. [2025]	✓	✓	×	×
Ding et al. [2025]	✓	✓	$\sim^*$	$\sim^*$
Ours (Equation 8)	✓	✓	✓	✓

Table 5: **Symbol-to-source mapping for the per-method equations of Section A.1.** The first column names the variable; the second gives the symbol used in this appendix (and in the wider methodology); the third gives the symbol used in the original source. Where a row applies to a single method, the method is named in parentheses. The continuous-time quantity τ_{mem} of the source has been re-expressed in our discrete-time notation, as noted in the right-hand column.

Variable (in text)	**This appendix**	**Source symbol**
Layer fan-in	D_{in}	n_{in} / $\mathrm{fan}_{\mathrm{in}}$; n_{l-1} [Herranz-Celotti and Rouat, 2024]
Layer fan-out	D_{out}	n_{out} / $\mathrm{fan}_{\mathrm{out}}$
Recurrent-layer dimension	D_{rec}	n_l [Herranz-Celotti and Rouat, 2024]
Firing threshold	θ	θ
Per-step membrane leak factor	β	α_{decay} [Herranz-Celotti and Rouat, 2024]; k [Ding et al., 2025]; $\beta = e^{-\Delta t/\tau_{\mathrm{mem}}}$ [Rossbroich et al., 2022, Perez-Nieves and Goodman, 2023]
Continuous-time membrane time constant	—	τ_{mem} (substituted via β)
Input-current scaling factor	λ	λ [Ding et al., 2025]
Membrane-potential value	u	u [Chang et al., 2024]; y_t [Herranz-Celotti and Rouat, 2024]
Membrane-potential target standard deviation	σ_U	σ_u [Rossbroich et al., 2022]
Per-step Bernoulli spike rate	p (or p^{l-1})	moments of z_t [Herranz-Celotti and Rouat, 2024]; $\Phi^c(\theta)$ identified as p under the standard-normal membrane assumption [Micheli et al., 2025]
Upstream spike input	S_t^{l-1}	z_t [Herranz-Celotti and Rouat, 2024]
Surrogate gradient	$\mathrm{SG}(u;\theta)$	$f'(u)$ [Chang et al., 2024]
Integrals of the postsynaptic-potential kernel and of its square	$\bar{\varepsilon}$, $\hat{\varepsilon}$	$\bar{\varepsilon}$, $\hat{\varepsilon}$ [Rossbroich et al., 2022]
Input firing rate (measured in the simulation study)	ν	ν [Rossbroich et al., 2022]
Membrane-potential target mean	μ_U	μ_u [Rossbroich et al., 2022]
Mean weight	$\bar{w}$	μ_W [Rossbroich et al., 2022]
Neuron state and input current	$u[t]$, $v[t]$, $i[t]$	$u[t]$, $v[t]$, $i[t]$ [Ding et al., 2025]
Threshold-scaling hyperparameter	c_θ	α [Lee et al., 2016]
Variance hyperparameter	c_{SG}	α [Chang et al., 2024]
Conservative gain	g	g [Bellec et al., 2018]
Number of simulation time steps	T	T [Che et al., 2025]
Standard-normal survival at threshold	$\Phi^c(\theta)$	$\Phi^c(\theta)$ [Micheli et al., 2025]
Anti-collapse integral term	$\mathcal{I}$	$\mathcal{I}$ [Perez-Nieves and Goodman, 2023]
Input-current standard deviation	σ_X	σ_X [Chang et al., 2024]
Mean recurrent weight	$\bar{w}_{\mathrm{rec}}$	$\bar{w}_{\mathrm{rec}}$ [Herranz-Celotti and Rouat, 2024]
Bias term	b	b [Ding et al., 2025]

B Appendix - methodology

This appendix derives the closed-form initialization of Section 2.3 for any target spike rate $p^l \in (0, 1)$. The closure Ψ is formulated in two stages in Section B.1. Its first stage is derived for hard-reset LIF neurons in Section B.2, and the Gaussian assumption it rests on is validated in Section B.3. A per-neuron contract then extends the closure in Section B.4. This contract fixes the moments that the input current of each neuron must supply. Extensions A–C each change one ingredient of this contract: the neuron model (Section B.5), the population (Section B.6) or the input current (Section B.7). One algorithm assembles the base case and every extension in Section B.8, and Table 6 maps every configuration to its derivation and results.

Table 6: **Map of the methodology appendix.** Every configuration holds at any target spike rate $p^l \in (0, 1)$ (Section B.1.3). Each row changes one ingredient of the closure relative to the configuration of the main text (first row); rows of different extensions compose (Section B.4). Results: S forward propagation at initialization, without training; T task-based training.

Configuration	What changes in Ψ	Methodology	Results
Main text: hard-reset LIF, $\beta^l = 0.9$, homogeneous $\{\theta^l, \beta^l, U^l_{\text{reset}}\}$, dense feed-forward current, $p^l = 50\%$	—	Section B.1, Section B.2, Section B.3	S: Section C.3.1; T: Table 3, configurations (4) and (5)
Extension A: LIF neuron model (Section B.5)			
Integrate-and-fire neuron, $\beta^l = 1$	none: f^* at $\beta^l = 1$	Section B.5.1	S: Section C.3.4
Memoryless binary unit, $\beta^l = 0$	none: f^* vanishes	Section B.5.1	S: Section C.3.4
Soft reset	f^* replaced by f^*_{soft}	Section B.5.2	S: Section C.3.4
Extension B: population (Section B.6)			
Heterogeneous $\{\theta^l_j, \beta^l_j, U^l_{\text{reset},j}\}$	f^* and target mean per row; one rate per layer	Section B.6	S: Section C.3.5
Extension C: input current (Section B.7)			
Residual stream	current budget split across streams	Section B.7.1	S: Section C.3.6
Recurrent stream	budget split; source rate p^l; diagonal masked	Section B.7.1	S: Section C.3.6
Convolutional layer	tensor view, one row per output channel; mean in the bias	Section B.7.2	S: Section C.3.7; T: Section C.4.4

B.1 Fixed-point closure

The closure targets the LIF model of Equation 1, restated here for layer l:

$$U^l_t = I^l_t + \left(\beta^l U^l_{t-1} + (1-\beta^l)U^l_\infty\right)\left(1 - S^l_{t-1}\right) + U^l_{\text{reset}}\, S^l_{t-1}, \qquad I^l_t = \mathbf{W}^l_X X^l_t + \mathbf{b}^l_X. \tag{23}$$

Between spikes and for $\beta^l < 1$ it equals the conventional low-pass form of a leaky integrator with unit DC gain,

$$U^l_t = \beta^l U^l_{t-1} + (1-\beta^l)\left(\frac{I^l_t}{1-\beta^l} + U^l_\infty\right),$$

whose weights and bias are those of Equation 23 divided by $1-\beta^l$. The closure therefore initializes both forms. The low-pass form loses its input at $\beta^l = 1$. Equation 23 instead holds on the closed interval $\beta^l \in [0, 1]$ and contains the integrate-and-fire neuron at $\beta^l = 1$ (Section B.5.1).

For this model, this subsection states the closure map Ψ of Section 2.3 and the consistency proposition that supports its well-posedness. It closes with Ψ at any target spike rate. The derivation of the first stage of Ψ follows in Section B.2.

B.1.1 The closure map Ψ

For an intermediate layer l with input dimension D_{in}, output dimension D_{out}, neuron hyperparameters $\{\theta^l, \beta^l, U^l_{\text{reset}}\}$, target moments $(\mu_{U^l}, \sigma^2_{U^l})$ for the membrane potential, and previous-layer spike rate p^{l-1}, define the closure map

$$\Psi:\ (\mu_{U^l}, \sigma^2_{U^l},\ p^{l-1};\ D_{\text{in}}, D_{\text{out}}, \theta^l, \beta^l, U^l_{\text{reset}})\ \longmapsto\ (\mu_{\mathbf{W}X},\ \sigma^2_{\mathbf{W}X}). \tag{24}$$

Ψ is the composition of two stages:

(i) An inner moment map $f^* : (\mu_{U^l}, \sigma^2_{U^l}) \mapsto (\mu_{\tilde{U}^l_{t-1}}, \sigma^2_{\tilde{U}^l_{t-1}})$ that returns the moments of the membrane recurrence for the given reset value, derived in Section B.2.5 via right-truncated Gaussian theory.

(ii) A closed-form weight-moment algorithm that equates the input-current moments of Equation 4 with the required moments of Equation 6 for an i.i.d. Bernoulli input of rate p^{l-1} with a Binomial spike count.

For a given target, stage (i) depends only on the neuron model and its hyperparameters. Stage (ii) depends only on the operator that forms the input current. The two stages meet in the per-neuron moments of that current (Equation 37). Each extension of this appendix changes one stage or evaluates Ψ per neuron (Table 6).

Weight construction. The orthogonal construction of Equation 8 turns the output of Ψ into a weight matrix. We draw its orthogonal base $\mathbf{W}^l_{X,\perp}$ as the sign-corrected Q factor of the QR decomposition of a Gaussian matrix, as `init.orthogonal_` of PyTorch [Paszke et al., 2019] does. For $D_{\text{out}} \le D_{\text{in}}$ its rows are orthonormal. For $D_{\text{out}} > D_{\text{in}}$ the base has rank $D_{\text{in}} < D_{\text{out}}$, and the squared norm of each row follows $\text{Beta}\big(D_{\text{in}}/2, (D_{\text{out}} - D_{\text{in}})/2\big)$ with mean $D_{\text{in}}/D_{\text{out}}$. The indicator term of Equation 8 applies the aspect-ratio correction $\sqrt{D_{\text{out}}/D_{\text{in}}}$, which restores the unit row norm in expectation. Equation 8 states this aspect-ratio correction explicitly for analytical completeness. Our implementation instead standardizes each row to zero mean and unit norm before the scaling and the shift. This restores the norm exactly for every row and takes the place of the factor, since applying both would multiply the input variance by $D_{\text{out}}/D_{\text{in}}$.

B.1.2 Consistency proposition

> *Proposition (informal).* Under assumptions (A1)–(A6) of Section 2.3, applying $\Psi(\mu_{U^l}, \sigma^2_{U^l}, p^{l-1}; \cdot)$ to the LIF dynamics of Equation 23 yields a membrane potential at layer l whose first two moments equal the target $(\mu_{U^l}, \sigma^2_{U^l})$. This target sets the spike rate $p^l = 1 - \Phi(\alpha)$ at $\alpha = (\theta^l - \mu_{U^l})/\sigma_{U^l}$. With $p^{l-1} = p^l = p$ it is a fixed point of Ψ along both the temporal axis (consecutive time steps share the same distribution) and the spatial axis (consecutive layers share the same Bernoulli rate). The default $(\mu_{U^l}, \sigma^2_{U^l}) = (\theta^l, 1)$ with $p = 50\%$ is its case $\alpha = 0$.

The proposition is informal because the LIF update defines a regime-switching auto-regressive process. This process is not Gaussian at any finite t, so the fixed point holds in the moment-matching sense under (A1)–(A2). Figure 5 reports the remaining gap to the target distribution, and Section B.3.1 justifies (A1).

Along the spatial axis the previous layer supplies the input that (A5) assumes. This input has three properties: a common rate, identical distribution and independence across neurons. It is therefore stricter than a general multivariate Bernoulli input, which permits unequal rates and correlations across neurons. These properties let each entry $s_m \sim \text{Ber}(p^{l-1})$ give the input vector $S^{l-1}_t \sim \text{Ber}(p^{l-1})^{\otimes D_{\text{in}}}$, whose L_1 norm or spike count $\|S^{l-1}_t\|_1 = \sum_m s_m$ is Binomial. For $D_{\text{out}} \le D_{\text{in}}$ the mutually orthogonal rows of the base $\mathbf{W}^l_{X,\perp}$ of Equation 8 keep the input currents of distinct neurons nearly uncorrelated (A4). They do not make them independent, which the input X^l_t shared by all neurons precludes. Departures from each property are tested in Section C.3.3.

B.1.3 Any target spike rate

Ψ holds for any target spike rate $p^l \in (0,1)$, of which the main text uses the default $p^l = 50\%$ of objective (B). The rate enters Ψ twice. The layer's own rate p^l fixes the target mean, the inverse Mills ratio r and the recurrence moments. The upstream rate p^{l-1} fixes the Binomial factors of the weight moments. Layers may therefore use different rates. This case is not evaluated here. Under (A1)–(A2) a neuron whose membrane potential follows $\mathcal{N}(\mu_{U^l}, \sigma^2_{U^l})$ spikes with probability $p^l = 1 - \Phi(\alpha)$ at $\alpha = (\theta^l - \mu_{U^l})/\sigma_{U^l}$. Inverting this relation for a requested rate at a chosen scale σ_{U^l} gives

$$\alpha = \Phi^{-1}(1 - p^l), \qquad \mu_{U^l} = \theta^l - \sigma_{U^l}\,\alpha, \qquad r = \frac{\phi(\alpha)}{\Phi(\alpha)}, \tag{25}$$

a target mean displaced from the threshold. With $p^{l-1} = p^l$ this target is a fixed point in the sense of the proposition (Section B.1.2). Equating the input-current moments of Equation 4 with the required moments of Equation 6 at this target gives the weight moments

$$\mu_{\mathbf{W}_X} = \frac{\mu_{U^l} - \mu_{\tilde{U}^l_{t-1}}}{p^{l-1}\,D_{\text{in}}}, \qquad \sigma^2_{\mathbf{W}_X} = \frac{\sigma^2_{U^l} - \sigma^2_{\tilde{U}^l_{t-1}}}{D_{\text{in}}\,p^{l-1}\,(1 - p^{l-1})} - \mu^2_{\mathbf{W}_X}, \tag{26}$$

Their numerators are the per-neuron target moments of the input current (Equation 37). At $\alpha = 0$ they reduce to Equation 7, and no other step of the derivation changes. The orthogonal construction of Equation 8 is unchanged. The closure is admissible whenever Equation 26 returns $\sigma^2_{\mathbf{W}_X} > 0$, which requires at least $\sigma^2_{\tilde{U}^l_{t-1}} < \sigma^2_{U^l}$. Assumptions (A1)–(A2) are validated at $p^l = 50\%$ in Section B.3. At other rates the evidence is the forward propagation of Section C.3.1.

The default $p = 50\%$ is the maximum-entropy Bernoulli rate of objective (B). It assumes nothing about the data beyond the binary code. The empirical evaluation therefore fixes this rate once and in advance for every paradigm (Section C.2). Its configurations (4) and (5) are the closed form at the targets $U_t \sim \mathcal{N}(0,1)$ and $U_t \sim \mathcal{N}(\theta,1)$. For $\theta = 1$ these are $\alpha = 1$ and $\alpha = 0$, i.e., $p \approx 0.159$ and $p = 50\%$ (Table 3). Two operating points do not locate an optimum, and we do not claim that $p = 50\%$ is optimal for every task. A task likely admits a better operating point. Its choice may involve a trade-off between task performance and inference energy cost [Li and Furber, 2021]. A practitioner with such knowledge substitutes that rate into Equation 25 and Equation 26. The closure then holds this rate per neuron in each layer. The rate set at initialization is also not the trained rate [Herranz-Celotti and Rouat, 2024] and activity regularization as an auxiliary loss can further shape spike rates during training [Guo et al., 2022]. Energy consumption and the spike rates of trained networks are not measured in this work.

B.2 Derivation of the recurrence moments

This subsection derives the inner step f^* of the closure Ψ (Equation 24) for the hard-reset LIF neuron of Equation 1. It details the four analytical steps of Section 2.3. The membrane recurrence first decomposes into two mutually exclusive spike clauses (Section B.2.1). The masked moments of the recursive clause then relate to truncated moments (Section B.2.2). The deterministic reset (A6) reduces the reset clause to a scaled Bernoulli variable (Section B.2.3). The coupling of the two clauses follows in Section B.2.4. Truncated-Gaussian moments under (A1)–(A2) then close the system (Section B.2.5). The derivation holds for any target $(\mu_{U^l}, \sigma^2_{U^l})$ and hence for any target spike rate (Section B.1.3). It assumes homogeneous hyperparameters within a layer, and Section B.6 lifts this restriction. The choice of the reset value is discussed in Section B.2.6.

B.2.1 Inter-dependence structure and decomposed expectation and variance

The membrane recurrence $\tilde{U}^l_{t-1}$ depends on the previous spike state S^l_{t-1} through two mutually exclusive clauses whose union covers the sample space:

$$\tilde{U}^l_{t-1} \propto \begin{cases} U^l_{t-1} & \text{if } (1 - S^l_{t-1}) = 1 \\ U^l_{\text{reset}} & \text{if } S^l_{t-1} = 1 \end{cases}$$

with,

$$\begin{aligned} \mathbb{P}(S^l_{t-1}) \cap \mathbb{P}((1 - S^l_{t-1})) &= \emptyset \\ \mathbb{P}(S^l_{t-1} = 0) \cup \mathbb{P}(S^l_{t-1} = 1) &= 1 \end{aligned}$$

The proportional symbol omits constants to emphasize the spike-induced switch between U^l_{t-1} and U^l_{reset}. We treat these as the two recurrence components and assume that neither is correlated with the new input current I^l_t at neuron-specific level in intermediate layers; the deterministic hard reset makes $\text{Cov}(I^l_t, U^l_{\text{reset}}) = 0$ by construction. The two spike clauses are however not independent of each other. The full inter-dependence structure is

$$
\begin{aligned}
(I) &: \text{Cov}(I^l_t, U^l_{t-1}) \approx 0 && (27)\\
&&&\text{(past \& new information)}\\
(II) &: \text{Cov}(I^l_t, U^l_{\text{reset}}) = 0\\
&&&\text{(deterministic reset)}\\
(III) &: \text{Cov}(I^l_t, U^l_{\infty}) = 0\\
&&&\text{(deterministic steady-state)}\\
&\text{which implies,}\\
(IV)\ &\ \text{Cov}(I^l_t, \tilde{U}^l_{t-1}) \approx 0\\
&&&\text{(consequence of former)}\\
&\text{and we note that,}\\
(V) &: \text{Cov}\left((1 - S^l_{t-1}) \cdot U^l_{t-1},\ S^l_{t-1} \cdot U^l_{\text{reset}}\right) \neq 0\\
&&&\text{(non-zero inter-clause dependence)}
\end{aligned}
$$

The neuron hyperparameters $\{\theta^l, \beta^l, U^l_{\infty}, U^l_{\text{reset}}\}$ are taken deterministic; we set $U^l_{\infty} = 0$ throughout, by convention. The decomposed expectation of U^l_t then follows by linearity:

$$
\begin{aligned}
\mathbb{E}[U^l_t] &= \mathbb{E}[I^l_t] + \mathbb{E}[\tilde{U}^l_{t-1}] && (28)\\
&= \mathbb{E}[I^l_t] + \mathbb{E}\left[\left(\beta^l U^l_{t-1} + (1-\beta^l)U^l_{\infty}\right)\left(1 - S^l_{t-1}\right) + U^l_{\text{reset}} S^l_{t-1}\right]\\
&= \mathbb{E}[I^l_t] + \mathbb{E}\left[\left(\beta^l U^l_{t-1} + (1-\beta^l)U^l_{\infty}\right)\left(1 - S^l_{t-1}\right)\right] + \mathbb{E}\left[U^l_{\text{reset}} S^l_{t-1}\right]\\
&\qquad \text{(split clauses)}\\
&= \mathbb{E}[I^l_t] + \mathbb{E}\left[\beta^l U^l_{t-1}\left(1 - S^l_{t-1}\right)\right] + \mathbb{E}\left[U^l_{\text{reset}} S^l_{t-1}\right]\\
&\qquad \text{(deterministic } U^l_{\infty} = 0\text{)}\\
&= \mathbb{E}[I^l_t] + \beta^l \mathbb{E}\left[U^l_{t-1}\left(1 - S^l_{t-1}\right)\right] + U^l_{\text{reset}} \mathbb{E}\left[S^l_{t-1}\right]\\
&\qquad \text{(deterministic } \{\beta^l, U^l_{\text{reset}}\}\text{)}\\
&= \mathbb{E}[I^l_t] + \beta^l \mathbb{E}\left[U^l_{t-1}\left(1 - S^l_{t-1}\right)\right] + U^l_{\text{reset}} \mathbb{P}(S^l_{t-1} = 1)\\
&\qquad (\mathbb{E}[S] = \mathbb{P}(S = 1) \cdot 1 \text{ for Bernoulli})
\end{aligned}
$$

The recursive membrane potential's expectation can now be defined as,

$$
\rightarrow \mathbb{E}[\tilde{U}^l_{t-1}] = \beta^l \mathbb{E}\left[U^l_{t-1}\left(1 - S^l_{t-1}\right)\right] + U^l_{\text{reset}} \mathbb{P}(S^l_{t-1} = 1)
$$

where $\mathbb{E}[U^l_{t-1}(1 - S^l_{t-1})]$ is the masked expectation of the previous membrane potential, derived in Section B.2.2. The decomposed variance follows analogously:

$$
\begin{aligned}
\text{Var}(U^l_t) &= \text{Var}(I^l_t) + \text{Var}(\tilde{U}^l_{t-1}) && (29)\\
&\qquad \text{(independence of input and recurrence)}\\
&= \text{Var}(I^l_t) + \text{Var}\left(\left(\beta^l U^l_{t-1} + (1-\beta^l)U^l_{\infty}\right)\left(1 - S^l_{t-1}\right) + U^l_{\text{reset}} S^l_{t-1}\right)\\
&= \text{Var}(I^l_t) + \text{Var}\left(\left(\beta^l U^l_{t-1} + (1-\beta^l)U^l_{\infty}\right)\left(1 - S^l_{t-1}\right)\right) + \text{Var}\left(U^l_{\text{reset}} S^l_{t-1}\right)\\
&\quad + 2 \cdot \text{Cov}\left((\beta^l U^l_{t-1} + (1-\beta^l)U^l_{\infty})(1 - S^l_{t-1})\ ,\ U^l_{\text{reset}} S^l_{t-1}\right)\\
&\qquad \text{(expand variance of recurrence)}\\
&= \text{Var}(I^l_t) + \text{Var}\left(\beta^l U^l_{t-1}\left(1 - S^l_{t-1}\right)\right) + \text{Var}\left(U^l_{\text{reset}} S^l_{t-1}\right)\\
&\quad + 2 \cdot \text{Cov}\left(\beta^l U^l_{t-1}(1 - S^l_{t-1})\ ,\ U^l_{\text{reset}} S^l_{t-1}\right)\\
&\qquad \text{(deterministic } U^l_{\infty} = 0\text{)}
\end{aligned}
$$

$$= \mathrm{Var}(I_t^l) + (\beta^l)^2 \cdot \mathrm{Var}\left(U_{t-1}^l \left(1 - S_{t-1}^l\right)\right) + (U_{\text{reset}}^l)^2 \cdot \mathrm{Var}\left(S_{t-1}^l\right)$$
$$+ 2 \cdot \beta^l \cdot U_{\text{reset}}^l \cdot \mathrm{Cov}\left(U_{t-1}^l(1 - S_{t-1}^l)\ ,\ S_{t-1}^l\right)$$
(deterministic $\{\beta^l, U_{\text{reset}}^l\}$)

The recursive membrane potential's variance can now be defined as,

$$\rightarrow \mathrm{Var}(\tilde{U}_{t-1}^l) = (\beta^l)^2 \cdot \mathrm{Var}\left(U_{t-1}^l \left(1 - S_{t-1}^l\right)\right) + (U_{\text{reset}}^l)^2 \cdot \mathrm{Var}\left(S_{t-1}^l\right)$$
$$+ 2 \cdot \beta^l \cdot U_{\text{reset}}^l \cdot \mathrm{Cov}\left(U_{t-1}^l(1 - S_{t-1}^l)\ ,\ S_{t-1}^l\right)$$

The three terms capture the masked-recurrence variance, the Bernoulli-distributed reset variance, and the covariance between the mutually exclusive spike-induced clauses. The remaining subsections express each in closed form as a function of known parameters only.

B.2.2 Moments of the recursive membrane potential clause

The masked recurrence $(1 - S_{t-1}^l)U_{t-1}^l$ is non-zero only when no spike occurred at $t-1$:

$$U_{t-1}^l(1 - S_{t-1}^l) = \begin{cases} U_{t-1}^l & \text{if } (1 - S_{t-1}^l) = 1 \\ 0 & \text{if } (1 - S_{t-1}^l) = 0 \end{cases}$$

This is a *masked* random variable: a continuous variable times a Bernoulli mask, with non-trivial probability mass at zero. It is not equivalent to a truncated variable, but the law of total probability links the two:

$$\mathbb{E}[U_{t-1}^l(1 - S_{t-1}^l)] \neq \mathbb{E}[U_{t-1}^l \,|\, (1 - S_{t-1}^l) = 1] \tag{30}$$
(masked $\neq$ truncated)

$$\mathbb{E}[U_{t-1}^l(1 - S_{t-1}^l)] = \mathbb{P}((1 - S_{t-1}^l) = 1) \cdot \mathbb{E}[U_{t-1}^l(1 - S_{t-1}^l) \,|\, (1 - S_{t-1}^l) = 1]$$
$$+ \mathbb{P}((1 - S_{t-1}^l) = 0) \cdot \mathbb{E}[U_{t-1}^l(1 - S_{t-1}^l) \,|\, (1 - S_{t-1}^l) = 0]$$
(law of total probability)

$$= \mathbb{P}((1 - S_{t-1}^l) = 1) \cdot \mathbb{E}[U_{t-1}^l \cdot 1 \,|\, (1 - S_{t-1}^l) = 1]$$
$$+ \mathbb{P}((1 - S_{t-1}^l) = 0) \cdot \mathbb{E}[U_{t-1}^l \cdot 0 \,|\, (1 - S_{t-1}^l) = 0]$$
(substitute $(1 - S_{t-1}^l)$ condition)

$$= \mathbb{P}((1 - S_{t-1}^l) = 1) \cdot \mathbb{E}[U_{t-1}^l \,|\, (1 - S_{t-1}^l) = 1] + 0$$
(rewritten)

$$= \mathbb{P}((1 - S_{t-1}^l) = 1) \cdot \mathbb{E}[U_{t-1}^l \,|\, U_{t-1}^l < \theta^l]$$
(masked relationship to truncated)

i.e., the marginal masked expectation equals the probability of no spike times the truncated expectation conditional on $U_{t-1}^l < \theta^l$.

The variance follows from the law of total variance (Eve's law), which decomposes the total into a within-group component (the average truncated variance under the mask) and a between-group component (the variance of the conditional means). Combined with the inter-dependence structure of Equation 27:

$$\mathrm{Var}(U_{t-1}^l(1 - S_{t-1}^l)) = \mathbb{E}\left[\left(U_{t-1}^l(1 - S_{t-1}^l)\right)^2\right] - \left(\mathbb{E}[U_{t-1}^l(1 - S_{t-1}^l)]\right)^2 \tag{31}$$
(variance definition)

$$\mathrm{Var}(U_{t-1}^l(1 - S_{t-1}^l)) = \mathbb{E}[\mathrm{Var}(U_{t-1}^l(1 - S_{t-1}^l) \mid (1 - S_{t-1}^l))]$$
$$+ \mathrm{Var}(\mathbb{E}[U_{t-1}^l(1 - S_{t-1}^l) \mid (1 - S_{t-1}^l)])$$
(law of total variance)

$$= \mathbb{P}((1 - S_{t-1}^l) = 1) \cdot \mathrm{Var}(U_{t-1}^l \mid U_{t-1}^l < \theta^l)$$
$$+ \mathbb{P}((1 - S_{t-1}^l) = 1) \cdot \mathbb{P}((1 - S_{t-1}^l) = 0) \cdot \left(\mathbb{E}[U_{t-1}^l \mid U_{t-1}^l < \theta^l]\right)^2$$
(*masked relationship to truncated)

Whose terms were separated through the clauses of S^l_{t-1},

(and simplified through substitution of conditionals)

$$\begin{aligned}
&\mathrm{Var}\left(U^l_{t-1}(1-S^l_{t-1}) \mid (1-S^l_{t-1})=1\right) = \mathrm{Var}(U^l_{t-1} \mid U^l_{t-1} < \theta^l),\\
&\mathrm{Var}\left(U^l_{t-1}(1-S^l_{t-1}) \mid (1-S^l_{t-1})=0\right) = 0,\\
&\mathbb{E}[U^l_{t-1}(1-S^l_{t-1}) \mid (1-S^l_{t-1})=1] = \mathbb{E}[U^l_{t-1} \mid U^l_{t-1} < \theta^l],\\
&\mathbb{E}[U^l_{t-1}(1-S^l_{t-1}) \mid (1-S^l_{t-1})=0] = 0,
\end{aligned}$$

where the asterisk marks omitted intermediate steps. Those steps decompose the nested expectation and variance into the components below; the nested expectation reads

$$\begin{aligned}
\mathbb{E}[\mathrm{Var}(U^l_{t-1}(1-S^l_{t-1}) \mid (1-S^l_{t-1}))] = &\\
&\mathbb{P}((1-S^l_{t-1})=1)\cdot \mathrm{Var}(U^l_{t-1} \mid U^l_{t-1} < \theta^l)\\
&+\mathbb{P}((1-S^l_{t-1})=0)\cdot \mathrm{Var}(U^l_{t-1} \mid U^l_{t-1} \geq \theta^l)\\
&\qquad\text{(law of total probability)}\\
&=\mathbb{P}((1-S^l_{t-1})=1)\cdot \mathrm{Var}(U^l_{t-1} \mid U^l_{t-1} < \theta^l)\\
&\quad+\mathbb{P}((1-S^l_{t-1})=0)\cdot 0\\
&\qquad\text{(substitution of known super-threshold}\\
&\qquad\text{variance equal to zero)}\\
&=\mathbb{P}((1-S^l_{t-1})=1)\cdot \mathrm{Var}(U^l_{t-1} \mid U^l_{t-1} < \theta^l)\\
&\qquad\text{(sub-result nested expectation)}
\end{aligned}$$

using the law of total probability and the fact that $\mathrm{Var}(U^l_{t-1} \mid U^l_{t-1} \geq \theta^l) = 0$ in the supra-threshold regime under the hard-reset model. The nested variance follows analogously:

$$\begin{aligned}
&\mathrm{Var}(\mathbb{E}[U^l_{t-1}(1-S^l_{t-1}) \mid (1-S^l_{t-1})]) =\\
&\quad\sum_{s\in\{0,1\}} \mathbb{P}\left(S^l_{t-1}=s\right)\cdot\left(\mathbb{E}[U^l_{t-1}(1-S^l_{t-1}) \mid S^l_{t-1}=s] - \mathbb{E}[U^l_{t-1}(1-S^l_{t-1})]\right)^2\\
&\qquad\text{(law of total probability \& two-point law for variance)}\\
&=\mathbb{P}((1-S^l_{t-1})=1)\cdot\left(\mathbb{E}\left[U^l_{t-1} \mid (1-S^l_{t-1})=1\right] - \mathbb{E}[U^l_{t-1}(1-S^l_{t-1})]\right)^2\\
&\qquad\text{(substitution of masked clause)}\\
&\quad+\mathbb{P}((1-S^l_{t-1})=0)\cdot\left(0 - \mathbb{E}[U^l_{t-1}(1-S^l_{t-1})]\right)^2\\
&\qquad\text{(use zero super-threshold expectation)}
\end{aligned}$$

where we substitute the masked to truncated expectation relationship,

$$\begin{aligned}
&=\mathbb{P}((1-S^l_{t-1})=1)\cdot\left(\mathbb{P}((1-S^l_{t-1})=0)\cdot\mathbb{E}\left[U^l_{t-1} \mid (1-S^l_{t-1})=1\right]\right)^2\\
&\qquad\text{(conditional - marginal = conj. conditional)}\\
&\quad+\mathbb{P}((1-S^l_{t-1})=0)\cdot\left(\mathbb{P}((1-S^l_{t-1})=1)\cdot\mathbb{E}\left[U^l_{t-1} \mid (1-S^l_{t-1})=1\right]\right)^2\\
&\qquad\text{(masked to truncated relationship)}\\
&=\mathbb{P}((1-S^l_{t-1})=1)\cdot\mathbb{P}((1-S^l_{t-1})=0)^2\cdot\left(\mathbb{E}\left[U^l_{t-1} \mid (1-S^l_{t-1})=1\right]\right)^2\\
&\quad+\mathbb{P}((1-S^l_{t-1})=0)\cdot\mathbb{P}((1-S^l_{t-1})=1)^2\cdot\left(\mathbb{E}\left[U^l_{t-1} \mid (1-S^l_{t-1})=1\right]\right)^2\\
&\qquad\text{(extract probabilities from squared expressions)}\\
&=\mathbb{P}((1-S^l_{t-1})=0)\cdot\mathbb{P}((1-S^l_{t-1})=1)\cdot\left(\mathbb{E}\left[U^l_{t-1} \mid (1-S^l_{t-1})=1\right]\right)^2\\
&\quad\cdot\left(\mathbb{P}((1-S^l_{t-1})=0)+\mathbb{P}((1-S^l_{t-1})=1)\right)\\
&\qquad\text{(merge split terms into single term)}\\
&=\mathbb{P}((1-S^l_{t-1})=0)\cdot\mathbb{P}((1-S^l_{t-1})=1)\cdot\left(\mathbb{E}\left[U^l_{t-1} \mid (1-S^l_{t-1})=1\right]\right)^2\cdot 1\\
&\qquad\text{(sum of spike clauses equals one i.e., the total probability space)}
\end{aligned}$$

$$= \mathbb{P}((1 - S^l_{t-1}) = 0) \cdot \mathbb{P}((1 - S^l_{t-1}) = 1) \cdot \left(\mathbb{E}\left[U^l_{t-1} \mid (1 - S^l_{t-1}) = 1\right]\right)^2$$

(sub-result nested variance)

using the marginal masked expectation from Equation 30.

B.2.3 Moments of the membrane potential's reset clause

The reset clause $S^l_{t-1}U^l_{\text{reset}}$ is non-zero only when a spike occurred at $t-1$:

$$U^l_{\text{reset}} \cdot S^l_{t-1} = \begin{cases} 0 & \text{if } S^l_{t-1} = 0 \\ U^l_{\text{reset}} & \text{if } S^l_{t-1} = 1 \end{cases}$$

For deterministic U^l_{reset}, the clause expectation and variance are scaled Bernoulli moments conditioned on the threshold-crossing probability:

$$\begin{aligned}
\mathbb{E}[U^l_{\text{reset}}S^l_{t-1}] &= U^l_{\text{reset}} \cdot \mathbb{E}[S^l_{t-1}] \\
&= U^l_{\text{reset}} \cdot \mathbb{P}(S^l_{t-1} = 1) \\
&= U^l_{\text{reset}} \cdot \mathbb{P}(U^l_{t-1} > \theta^l) \\
\text{Var}(U^l_{\text{reset}}S^l_{t-1}) &= (U^l_{\text{reset}})^2 \cdot \text{Var}(S^l_{t-1}) \\
&= (U^l_{\text{reset}})^2 \cdot \left(\mathbb{P}(S^l_{t-1} = 1) \cdot (1 - \mathbb{P}(S^l_{t-1} = 1))\right) \\
&= (U^l_{\text{reset}})^2 \cdot \left(\mathbb{P}(U^l_{t-1} \geq \theta^l) \cdot (1 - \mathbb{P}(U^l_{t-1} \geq \theta^l))\right)
\end{aligned}$$

These reduce to closed form once $\mathbb{P}(U^l_{t-1} \geq \theta^l)$ is supplied by truncated-Gaussian theory under (A1).

B.2.4 Covariance of the spike-induced clauses

The covariance term in Equation 29 expands by linearity of expectation:

$$\begin{aligned}
&\text{Cov}\left(U^l_{t-1}(1 - S^l_{t-1}),\, U^l_{\text{reset}}S^l_{t-1}\right) = \\
&\quad \mathbb{E}\left[(U^l_{t-1}(1 - S^l_{t-1})) \cdot (U^l_{\text{reset}}S^l_{t-1})\right] \\
&\quad - \mathbb{E}\left[(U^l_{t-1}(1 - S^l_{t-1}))\right] \cdot \mathbb{E}\left[U^l_{\text{reset}}S^l_{t-1}\right] \\
&\qquad \text{(linearity of expectation)} \\
&= 0 - \mathbb{E}\left[(U^l_{t-1}(1 - S^l_{t-1}))\right] \cdot \mathbb{E}\left[U^l_{\text{reset}}S^l_{t-1}\right] \\
&\qquad \text{(mutual exclusivity: } (1 - s) \cdot s = 0\text{)} \\
&= -\mathbb{E}\left[(U^l_{t-1}(1 - S^l_{t-1}))\right] \cdot U^l_{\text{reset}}\mathbb{E}\left[S^l_{t-1}\right] \\
&\qquad \text{(deterministic } U^l_{\text{reset}}\text{)} \\
&= -\mathbb{E}\left[(U^l_{t-1}(1 - S^l_{t-1}))\right] \cdot U^l_{\text{reset}}\mathbb{P}\left(S^l_{t-1} = 1\right) \\
&\qquad \text{(swap expectation with probability)} \\
&\text{Substituting the masked-to-truncated expectation relationship,} \\
&= -\mathbb{P}((1 - S^l_{t-1}) = 1) \cdot \mathbb{E}[U^l_{t-1} \,|\, U^l_{t-1} < \theta^l] \cdot U^l_{\text{reset}}\mathbb{P}\left(S^l_{t-1} = 1\right) \\
&= -U^l_{\text{reset}} \cdot \mathbb{P}\left(S^l_{t-1} = 1\right) \cdot \mathbb{P}((1 - S^l_{t-1}) = 1) \cdot \mathbb{E}[U^l_{t-1} \,|\, U^l_{t-1} < \theta^l] \\
&\qquad \text{(re-arrange)}
\end{aligned} \tag{32}$$

which collects terms already obtained for the masked moments.

B.2.5 Synthesis – maintaining a steady-state distribution

The masked moments and covariance derived above reduce the moments of $\tilde{U}^l_{t-1}$ to combinations of three quantities,

$$\left\{\mathbb{P}(S^l_{t-1} = 1), \quad \mathbb{E}[U^l_{t-1} \,|\, U^l_{t-1} < \theta^l], \quad \text{Var}(U^l_{t-1} \,|\, U^l_{t-1} < \theta^l)\right\} \tag{33}$$

i.e., the masking probability and the truncated mean and variance of the previous membrane potential. The complementary mask probability follows from $\mathbb{P}(S_{t-1}^l = 0) + \mathbb{P}(S_{t-1}^l = 1) = 1$.

To close the moment system we invoke (A1) Gaussianity of U_t^l and (A2) stationarity in t. The LIF dynamics in Equation 1 are strictly non-Gaussian (a regime-switching mixture); (A1)–(A2) are justified theoretically and validated empirically in Section B.3. Stationarity (A2) extends (A1) across time, giving $U_{t-1}^l \overset{d}{=} U_t^l$:

$$U_t^l \sim \mathcal{N}\left(\mu_{U_t^l}\,,\, \sigma^2_{U_t^l}\right) \tag{34}$$

(assumed Gaussian distribution)

$$U_t^l \sim \mathcal{N}\left(\mu_{U^l}\,,\, \sigma^2_{U^l}\right)$$

(assumed stationary/steady-state distribution)

this implies the same relationship for **any** time step, i.e.,

$$U_{t-1}^l \sim \mathcal{N}\left(\mu_{U^l}\,,\, \sigma^2_{U^l}\right)$$

(extension of assumed Gaussian distribution across time)

$$s.t.\ U_{t-1}^l \overset{d}{=} U_t^l.$$

(equality of distributions between consecutive time steps)

Under Equation 34 the three quantities of Equation 33 have closed-form truncated-Gaussian expressions. (A2) is not asserted blindly: the construction below identifies the conditions on I_t^l and $\tilde{U}_{t-1}^l$ that reinforce stationarity, rather than treating it as exogenously given.

For $U_t^l \sim \mathcal{N}(\mu_{U^l}, \sigma^2_{U^l})$, the right-truncated Gaussian moments at threshold θ^l are

$$\mathbb{E}[U_{t-1}^l \,|\, (1 - S_{t-1}^l) = 1] = (\mu_{U^l} - r\sigma_{U^l})$$

(right-truncated Gaussian expectation)

$$\mathrm{Var}(U_{t-1}^l \,|\, (1 - S_{t-1}^l) = 1) = \sigma^2_{U^l}\left(1 - \alpha r - r^2\right)$$

(right-truncated Gaussian variance)

where,

$$r = \frac{\phi(\alpha)}{\Phi(\alpha)}$$

(inverse *Mills ratio* r, adjusted for $U_{t-1}^l < \theta^l$)

$$\alpha = \frac{\theta^l - \mu_{U^l}}{\sigma_{U^l}}$$

(normalized threshold)

$$\phi(\alpha) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}\alpha^2}$$

(standard Gaussian PDF)

$$\Phi(\alpha) = \int_{-\infty}^{\alpha} \phi(t)\, dt$$

(standard Gaussian CDF)

Substituting these into Equation 30, Equation 31, and Equation 32 yields

$$\mathbb{P}(S_{t-1}^l = 1) = \mathbb{P}(U_{t-1}^l > \theta^l) = p^l = 1 - \Phi\left(\frac{\theta^l - \mu_{U^l}}{\sigma_{U^l}}\right)$$

(spike probability)

$$\mathbb{E}[U_{t-1}^l(1 - S_{t-1}^l)] = \mathbb{P}((1 - S_{t-1}^l) = 1) \cdot \mathbb{E}[U_{t-1}^l \,|\, U_{t-1}^l < \theta^l]$$

(masked relationship to truncated)

$$= (1 - p^l)\,(\mu_{U^l} - r\sigma_{U^l})$$
(masked expectation)

$$\begin{aligned}\mathrm{Var}(U^l_{t-1}(1 - S^l_{t-1})) &= \mathbb{P}((1 - S^l_{t-1}) = 1) \cdot \mathrm{Var}(U^l_{t-1} \mid U^l_{t-1} < \theta^l) \\ &\quad + \mathbb{P}((1 - S^l_{t-1}) = 1) \cdot \mathbb{P}((1 - S^l_{t-1}) = 0) \cdot \left(\mathbb{E}[U^l_{t-1} \mid U^l_{t-1} < \theta^l]\right)^2 \\ &\qquad \text{(masked relationship to truncated)} \\ &= (1 - p^l)\,\sigma^2_{U^l}\left(1 - \alpha r - r^2\right) + \left(1 - p^l\right)\,p^l\,\left(\mu_{U^l} - r\sigma_{U^l}\right)^2 \\ &\qquad \text{(masked variance)}\end{aligned}$$

$$\begin{aligned}&\mathrm{Cov}\left(U^l_{t-1}(1 - S^l_{t-1}),\, U^l_{\text{reset}} S^l_{t-1}\right) = \\ &\quad - U^l_{\text{reset}} \cdot \mathbb{P}\left(S^l_{t-1} = 1\right) \cdot \mathbb{P}((1 - S^l_{t-1}) = 1) \cdot \mathbb{E}[U^l_{t-1} \,|\, U^l_{t-1} < \theta^l] \\ &\qquad \text{(masked relationship to truncated)} \\ &= -U^l_{\text{reset}}\,p^l\,(1 - p^l)\,\left(\mu_{U^l} - r\sigma_{U^l}\right) \\ &\qquad \text{(masked covariance)}\end{aligned}$$

which closes the unknowns of (33).

Substituting the truncated expressions into Equation 28 and Equation 29 yields the closed-form moments of $\tilde{U}^l_{t-1}$ in terms of any deterministic U^l_{reset}:

$$\begin{aligned}\mathbb{E}[\tilde{U}^l_{t-1}] &= \beta^l \mathbb{E}\left[U^l_{t-1}\left(1 - S^l_{t-1}\right)\right] + U^l_{\text{reset}} \mathbb{P}(S^l_{t-1} = 1) \\ &\qquad \text{(decomposed expectation)} \\ &= \beta^l (1 - p^l)\,(\mu_{U^l} - r\sigma_{U^l}) + p^l U^l_{\text{reset}} \\ &\qquad \text{(substitute truncated moments)}\end{aligned}$$

$$\rightarrow \quad \mu_{\tilde{U}^l_{t-1}} = \beta^l (1 - p^l)\,(\mu_{U^l} - r\sigma_{U^l}) + p^l U^l_{\text{reset}}$$
(final result; symbolic in U^l_{reset})

$$\begin{aligned}\mathrm{Var}(\tilde{U}^l_{t-1}) &= (\beta^l)^2 \cdot \mathrm{Var}\left(U^l_{t-1}\left(1 - S^l_{t-1}\right)\right) \\ &\quad + (U^l_{\text{reset}})^2 \cdot \mathrm{Var}\left(S^l_{t-1}\right) \\ &\quad + 2 \cdot \beta^l \cdot U^l_{\text{reset}} \cdot \mathrm{Cov}\left(U^l_{t-1}(1 - S^l_{t-1})\ ,\ S^l_{t-1}\right) \\ &\qquad \text{(decomposed variance)} \\ &= (\beta^l)^2 \cdot \Big((1 - p^l)\,\sigma^2_{U^l}\left(1 - \alpha r - r^2\right) + \left(1 - p^l\right)\,p^l\,\left(\mu_{U^l} - r\sigma_{U^l}\right)^2\Big) \\ &\quad + (U^l_{\text{reset}})^2 \cdot p^l\,(1 - p^l) \\ &\quad - 2 \cdot \beta^l \cdot \left(U^l_{\text{reset}}\,p^l\,(1 - p^l)\,\left(\mu_{U^l} - r\sigma_{U^l}\right)\right) \\ &\qquad \text{(substitute truncated moments)}\end{aligned}$$

$$\begin{aligned}\rightarrow \quad \sigma^2_{\tilde{U}^l_{t-1}} &= (\beta^l)^2 \left((1 - p^l)\sigma^2_{U^l}(1 - \alpha r - r^2) + (1 - p^l)p^l(\mu_{U^l} - r\sigma_{U^l})^2\right) \\ &\quad + (U^l_{\text{reset}})^2 \cdot p^l(1 - p^l) \\ &\quad - 2\beta^l \cdot U^l_{\text{reset}} \cdot p^l(1 - p^l)(\mu_{U^l} - r\sigma_{U^l}) \\ &\qquad \text{(final result; symbolic in } U^l_{\text{reset}})\end{aligned}$$

These results complete the inner step f^* of Ψ for any admissible U^l_{reset} and give Equation 5. Other neuron models change this stage of Ψ (Section B.5).

B.2.6 Reset value

The reset value U^l_{reset} is a free hyperparameter (A6) throughout the methodology and its extensions. The closure holds for any deterministic reset that leaves a positive weight variance in Equation 26 ($\sigma^2_{\tilde{U}^l_{t-1}} < \sigma^2_{U^l}$), because the weight moments absorb the recurrence moments it induces. The derivation therefore advises a value but does not fix one. Rearranged, the recurrence variance of Equation 5 is quadratic in the reset,

$$\sigma^2_{\tilde{U}^l_{t-1}} = (\beta^l)^2 \left(1-p^l\right) \sigma^2_{U^l} \left(1-\alpha r - r^2\right) + p^l \left(1-p^l\right) \left(\beta^l \left(\mu_{U^l} - r\,\sigma_{U^l}\right) - U^l_{\text{reset}}\right)^2 . \tag{35}$$

Its second term vanishes for the conditional mean of the silent clause ($U^l_\infty = 0$),

$$U^l_{\text{reset}} = \mathbb{E}\left[\beta^l U^l_{t-1} \,\middle|\, \left(1-S^l_{t-1}\right) = 1\right] = \beta^l \left(\mu_{U^l} - r\,\sigma_{U^l}\right), \tag{36}$$

the right-truncated mean of Section B.2.5 scaled by the leak. A spiking neuron is thus reset to the potential it would carry in expectation had it stayed silent. Both clauses of the recurrence then share the mean $\beta^l(\mu_{U^l} - r\,\sigma_{U^l})$, so a spike no longer shifts the expected recurrence. The recurrence variance reduces to the first term of Equation 35, the smallest over all deterministic resets. This reset thus leaves the largest variance budget to the input current, by the same criterion as the soft reset of Section B.5.2.

We therefore advise this reset. Configurations (4) and (5) of the empirical evaluation (Section 3.2) and their simulation (Section C.3.2) use it, while configurations (1)–(3) keep $U^l_{\text{reset}} = 0$. The other hard-reset simulations (Section 3.1 and Section C.3) use the earlier value $\beta^l(1-p^l)(\mu_{U^l} - r\,\sigma_{U^l})$, the unconditional mean of the silent clause. On the base network the spike rate of the deepest layer agrees between the two values within the spread across neurons up to $p^l = 50\%$ and departs above it, most at $p^l = 85\%$. Where β^l is learned, the reset keeps its initial value. It remains a design choice of this work. The derivation leaves U^l_{reset} symbolic, and a designer may set any admissible value.

B.3 Validity of the Gaussian assumption

Assumptions (A1) and (A2) carry the closure. The inverse Mills ratio of Section B.2.5 is exact only for a Gaussian U^l_{t-1}, and the substitution $U^l_{t-1} \overset{d}{=} U^l_t$ of Equation 34 is exact only under stationarity. The LIF update of Equation 1 is a regime-switching auto-regressive process and is not Gaussian at any finite t. This subsection gives the theoretical reason why the gap to the moment-matched Gaussian is small and the empirical evidence that it is.

B.3.1 Two-step CLT and Gaussianization

Step 1 — input current via finite-dimensional CLT. Under the Gaussian weight matrix and i.i.d. Bernoulli input of Equation 4, the input current $I^l_t = \mathbf{W}_X X^l_t$ is a sum of D_{in} independent products of a Gaussian column and a Bernoulli mask. Lyapunov / Lindeberg conditions are satisfied at moderate D_{in}, yielding I^l_t approximately Gaussian with the moments stated in Equation 4.

Step 2 — Gaussianization of the membrane potential. Once I^l_t is approximately Gaussian, the recursive update $U^l_t = I^l_t + \beta^l U^l_{t-1}(1-S^l_{t-1}) + U^l_{\text{reset}} S^l_{t-1}$ accumulates approximately-Gaussian innovations through a regime-switching nonlinearity. The argument is structurally similar to the recursive-Gaussianization observation in the diffusion-model literature, where recursive injection of Gaussian noise maintains an approximately Gaussian state at every step; we transfer the recursive-accumulation idea, not the algorithmic equivalence.

Exactness of the Gaussian assumption. Neither step produces an *exactly* Gaussian U^l_t at any finite t: Step 1 is a finite-D_{in} approximation, and Step 2 retains a non-Gaussian regime-switching component as long as the spike-clause mixture has any imbalance. The closure is therefore a moment-matching construction whose closed-form expressions are exact under (A1)–(A2) and whose empirical accuracy depends on the gap being small.

B.3.2 Empirical validation

Without the closure, the LIF dynamics drift toward a sub-threshold attractor and individual neurons fall silent. This temporal form of spike-rate collapse is also known as the silencing problem [Perez-Nieves and Goodman, 2023]. It arises from an asymmetry across the firing threshold. Below the

threshold the membrane potential evolves linearly under recurrence and integration. Above it the spike-and-reset operation translates the state to U^l_{reset} and removes the upper tail at every time step. Aggregated over realizations, the expected membrane potential and the spike rate decay until the dynamics settle at the attractor. The reset translates the process instead of clipping it, so the LIF update is a regime-switching rather than a truncated auto-regressive process. Figure 4 shows this collapse for a single neuron under standard-Gaussian input. Existing initialization schemes that target the collapse typically inflate the variance to delay the decay rather than to remove it. For example, Che et al. [2025] scale the weight variance with the number of simulation time steps. The closure instead solves for the conditions under which the collapse vanishes.

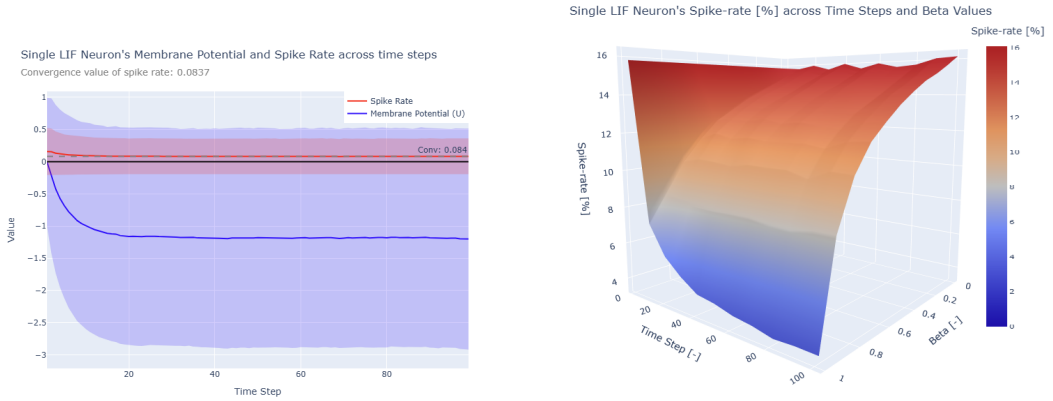

Figure 4: **Spike-rate collapse for LIF neurons.** Single LIF neuron with hyperparameters $\{\theta^l = 1, U^l_{\text{reset}} = 0, U^l_{\infty} = 0\}$ subject to standard Gaussian innovations $I^l_t \sim \mathcal{N}(0, 1)$, evaluated across 50,000 realizations. **(Left)** At $\beta^l = 0.9$ the initial sample mean spike rate of $\approx 15\%$ decays to 8.4%; the membrane-potential trajectory drifts sub-threshold and stabilizes. **(Right)** The extent of collapse increases monotonically with the leakage factor β^l as the persistence of the (already-decayed) previous state grows.

Figure 5 reports the Kullback–Leibler divergence between the empirical distribution of U^l_t and the target Gaussian $Z \sim \mathcal{N}(\theta^l, 1)$, evaluated across $50{,}000$ samples for two configurations of the input current.

Both panels settle at finite KL plateaus but at different fixed points. Under the input-current moments of the closure (left), the membrane potential converges to the target $Z \sim \mathcal{N}(\theta^l, 1)$ within a few time steps. This validates (A1) and (A2) for the constructed scheme. Under naive standard-Gaussian inputs (right), it converges to a non-target distribution whose mean drifts below the threshold. This is the collapse of Figure 4 in distributional form. The closure Ψ thus selects the target fixed point over the collapsed one.

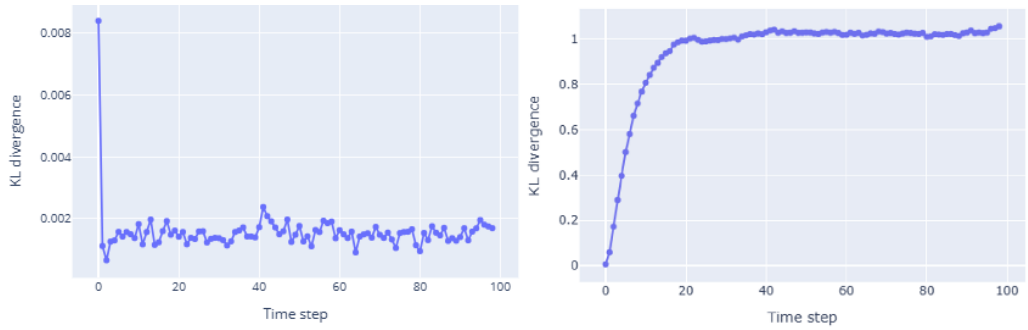

Figure 5: **Stationary Gaussian convergence of the membrane potential.** KL divergence between the empirical distribution of U_t^l and the target Gaussian $Z \sim \mathcal{N}(\theta^l, 1)$, evaluated across 50,000 samples. **(Left)** Under the input-current moments $(\mu_{I^l}, \sigma_{I^l})$ implied by the closure Ψ, the divergence decays to a near-zero stationary value within a few time steps. **(Right)** Under naive standard-Gaussian inputs $I_t^l \sim \mathcal{N}(0, 1)$, the dynamics still settle but at a non-target distribution.

B.4 Extending the closure: the per-neuron contract

The two stages of Ψ meet in the moments of the input current that each neuron receives. This subsection states these moments as a per-neuron contract between a neuron and its input current. The neuron side fixes the moments that the current must supply, and the current side delivers them. Extension A changes the neuron model and hence the required moments (Section B.5). Extension B evaluates the required moments per neuron (Section B.6). Extension C changes the operator that delivers them (Section B.7).

What the closure constrains. The required input-current moments of Equation 6 hold per neuron. Neuron j of layer l has its own hyperparameters θ_j^l, β_j^l and $U_{\text{reset},j}^l$. It receives the total current $I_{t,j}^l$ summed over all input streams and the bias. For $U_{t,j}^l \sim \mathcal{N}(\mu_{U_j^l}, \sigma_{U_j^l}^2)$ the decomposition of Equation 28 and Equation 29 under (A2) and (A3) requires the moments

$$\begin{aligned} \mu_{I_j^l} &= \mathbb{E}\big[I_{t,j}^l\big] = \mu_{U_j^l} - \mu_{\tilde{U}_{t-1,j}^l} \\ \sigma_{I_j^l}^2 &= \mathrm{Var}\left(I_{t,j}^l\right) = \sigma_{U_j^l}^2 - \sigma_{\tilde{U}_{t-1,j}^l}^2 \end{aligned} \tag{37}$$

Here, $(\mu_{\tilde{U}_{t-1,j}^l}, \sigma_{\tilde{U}_{t-1,j}^l}^2)$ are the recurrence moments of Equation 5 on neuron j's own hyperparameters. The moments are taken over time and input realizations for fixed weights. The condition therefore holds for every neuron separately. At a general target rate the target mean sits a fixed number of standard deviations below the neuron's own threshold (Equation 25).

Equation 37 references only the receiving neuron's hyperparameters and target moments. It references neither the operator that forms $I_{t,j}^l$ nor the inputs that this operator acts on. It thus states what the current of each neuron must be and not how it is produced. The neuron model fixes the right-hand sides, and the operator that forms the current realizes them (Section B.7).

What the operator delivers. Dense and convolutional layers as well as residual and recurrent streams form the current with linear operators acting on Bernoulli inputs. A stream s with fan-in $D_{\text{in},s}$ and source rate p_s feeds neuron j through row j of its weights $\mathbf{W}_s^l$. Under (A5) applied to its source and conditional on the weights it contributes

$$\begin{aligned} \mathbb{E}\big[I_{s,t,j}^l\big] &= p_s \sum_i W_{s,ji}^l = D_{\text{in},s}\, p_s\, \mu_{\mathbf{W}_s,j} \\ \mathrm{Var}\left(I_{s,t,j}^l\right) &= p_s(1-p_s) \sum_i \left(W_{s,ji}^l\right)^2 = D_{\text{in},s}\, p_s(1-p_s)\left(\sigma_{\mathbf{W}_s,j}^2 + \mu_{\mathbf{W}_s,j}^2\right) \end{aligned} \tag{38}$$

Here, $\mu_{\mathbf{W}_s,j}$ and $\sigma_{\mathbf{W}_s,j}^2$ are the mean and variance of row j of $\mathbf{W}_s^l$ over its active entries. The streams and the bias sum to $I_{t,j}^l$, and each stream enters with its own source rate p_s of any value. The

variance assumes inputs that are uncorrelated within and across streams. Equating these delivered moments with the required moments of Equation 37 gives the weight moments. For one dense stream the delivered moments are the third line of Equation 4, and the equation gives Equation 26.

Composition. Since Extensions A–C change distinct ingredients of the contract, they compose (Table 6). For instance, a heterogeneous soft-reset layer with a residual stream evaluates the soft-reset recurrence moments per row and splits each row's budget over two streams. Apart from sweeps of the target rate, such combinations are derived but not simulated. The algorithm of Section B.8 assembles the base case and every extension.

B.5 Extension A: LIF neuron models

Extension A changes the neuron side of the contract (Equation 37). The neuron model enters Ψ only through the recurrence moments of stage (i), evaluated at the α of the requested rate. The leak interval of the hard-reset LIF neuron contains two further members of practical interest, which are evaluations of the existing closed form (Section B.5.1). The soft reset changes the membrane recurrence itself and replaces f^* (Section B.5.2).

B.5.1 Leak factor: integrate-and-fire neuron and binary unit

The leak interval. Grouping the terms of the recurrence variance of Section B.2.5 by the spike state S^l_{t-1} (law of total variance) writes the recurrence moments as

$$\begin{aligned}\mu_{\tilde{U}^l_{t-1}} &= \beta^l(1-p^l)\,(\mu_{U^l} - r\,\sigma_{U^l}) + p^l\,U^l_{\text{reset}} \\ \sigma^2_{\tilde{U}^l_{t-1}} &= \underbrace{(\beta^l)^2(1-p^l)\,\sigma^2_{U^l}\left(1-\alpha r - r^2\right)}_{\text{within clauses}} + \underbrace{p^l(1-p^l)\left(\beta^l\,(\mu_{U^l} - r\,\sigma_{U^l}) - U^l_{\text{reset}}\right)^2}_{\text{between clauses}}\end{aligned} \tag{39}$$

For a given deterministic U^l_{reset}, $\mu_{\tilde{U}^l_{t-1}}$ is linear and $\sigma^2_{\tilde{U}^l_{t-1}}$ quadratic in β^l, and the weight moments of Equation 26 are polynomial in both. The closure is therefore defined without singularity on the closed interval $\beta^l \in [0,1]$. Its two endpoints are evaluations of the existing closed form. This holds because the recurrence moments are one-step moments of the masked recurrence under the stationary target law of U^l_{t-1}. They are not the stationary solution of a linear recursion such as an AR(1) process. For a leaky integrator without reset driven by independent inputs, the stationary variance $\mathrm{Var}(I^l_t)/(1-(\beta^l)^2)$ diverges as $\beta^l \to 1$. Initialization factors built on the membrane kernel diverge in the same limit, such as the integrals $\bar{\varepsilon} = 1/(1-\beta^l)$ and $\hat{\varepsilon} = 1/(1-(\beta^l)^2)$ of Rossbroich et al. [2022] (Section A.1). The closure contains no such factor at either endpoint.

Integrate-and-fire neuron. At $\beta^l = 1$ the term $(1-\beta^l)U^l_\infty$ vanishes and Equation 1 becomes the IF neuron with hard reset,

$$U^l_t = I^l_t + U^l_{t-1}\left(1 - S^l_{t-1}\right) + U^l_{\text{reset}}\,S^l_{t-1}, \tag{40}$$

which integrates without loss between spikes. Its recurrence moments are Equation 39 at $\beta^l = 1$, and the weight moments are Equation 26 with these values; no step of the derivation changes.

Memoryless binary unit. At $\beta^l = 0$ and $U^l_\infty = 0$ the recurrence contracts to $\tilde{U}^l_{t-1} = U^l_{\text{reset}}\,S^l_{t-1}$, a one-step offset gated by the previous spike, with $\mu_{\tilde{U}^l_{t-1}} = p^l\,U^l_{\text{reset}}$ and $\sigma^2_{\tilde{U}^l_{t-1}} = (U^l_{\text{reset}})^2\,p^l(1-p^l)$. The reset of Equation 36 is zero at $\beta^l = 0$, since it scales with the leak. The offset thus vanishes and the neuron reduces to the memoryless binary unit $S^l_t = H(I^l_t;\theta^l)$. Then $\sigma^2_{\tilde{U}^l_{t-1}} = 0$, so the recurrence leaves the whole target variance to the input current, and Equation 26 reduces to

$$\mu_{\mathbf{W}_X} = \frac{\mu_{U^l}}{p^{l-1}\,D_{\text{in}}}, \qquad \sigma^2_{\mathbf{W}_X} = \frac{\sigma^2_{U^l}}{D_{\text{in}}\,p^{l-1}\,(1-p^{l-1})} - \mu^2_{\mathbf{W}_X}. \tag{41}$$

At the target $\mu_{U^l} = 0$, $\sigma^2_{U^l} = 1$, which spikes at $p^l = 1 - \Phi(\theta^l)$, the mean term vanishes and, with $p^{l-1} = p^l$, Equation 41 coincides with the per-neuron form of Micheli et al. [2025] (Section A.1). At any other target, for instance $\mu_{U^l} = \theta^l$ with $p^l = 50\%$, it adds the mean term that places the membrane mean at the target. The zero-leak, zero-reset corner of the closure therefore recovers Micheli et al. [2025] and generalizes it to targets $\mu_{U^l} \neq 0$.

B.5.2 Soft reset

Under a soft reset, a spike does not replace the membrane potential: the state is retained, and the reset term $U^l_{\text{reset}}\, S^l_{t-1}$ is added to it,

$$\begin{aligned} U^l_t &= I^l_t + \beta^l U^l_{t-1} + (1-\beta^l)\, U^l_\infty + U^l_{\text{reset}}\, S^l_{t-1} \\ U^l_t &= I^l_t + \tilde{U}^l_{t-1}, \quad S^l_t = H(U^l_t;\theta^l), \quad I^l_t = \mathbf{W}^l_X X^l_t + \mathbf{b}^l_X \end{aligned} \tag{42}$$

Relative to the hard-reset update of Equation 1, the only change is the missing mask $(1 - S^l_{t-1})$ on the retained state. In the clause form of the hard reset, $\tilde{U}^l_{t-1} = \beta^l U^l_{t-1}(1 - S^l_{t-1}) + (\beta^l U^l_{t-1} + U^l_{\text{reset}})\, S^l_{t-1}$: the post-spike state $\beta^l U^l_{t-1} + U^l_{\text{reset}}$ is random, whereas the hard-reset derivation relies on a deterministic post-spike state. The soft reset is therefore not an instance of the hard-reset closed form. We derive its moments with a subset of the same devices, starting from the decomposition of Section B.2.1 and the truncated-Gaussian moments of Section B.2.5, under (A1)–(A3) and with (A6) applied to $\{\theta^l, \beta^l, U^l_{\text{reset}}\}$; we set $U^l_\infty = 0$ as in Section B.2.

Inter-dependence structure. The dependence structure corresponding to Equation 27 is

$$(I) : \operatorname{Cov}(I^l_t, U^l_{t-1}) \approx 0 \tag{43}$$

(past & new information, (A3))

$$(II) : \operatorname{Cov}(I^l_t, S^l_{t-1}) \approx 0$$

($S^l_{t-1} = H(U^l_{t-1};\theta^l)$ is a function of U^l_{t-1})

$$(III) : \operatorname{Cov}(I^l_t, U^l_{\text{reset}}) = \operatorname{Cov}(I^l_t, U^l_\infty) = 0$$

(deterministic reset term and resting potential)

which implies,

$$(IV)\ \operatorname{Cov}(I^l_t, \tilde{U}^l_{t-1}) \approx 0$$

(consequence of former)

and we note that,

$$(V) : \operatorname{Cov}(U^l_{t-1}, S^l_{t-1}) \neq 0$$

(dependence of the state and its own spike)

Relation (II) holds when I^l_t is independent of U^l_{t-1} at initialization, the reading of (A3) under which relation (IV) of Equation 27 holds as well. Relation (V) is the only non-trivial dependence; under the hard reset the covariance of the two masked clauses plays this role.

Decomposed expectation and variance. The expectation of U^l_t follows by linearity:

$$\begin{aligned} \mathbb{E}[U^l_t] &= \mathbb{E}[I^l_t] + \mathbb{E}[\tilde{U}^l_{t-1}] \\ &= \mathbb{E}[I^l_t] + \mathbb{E}\left[\beta^l U^l_{t-1} + (1-\beta^l)U^l_\infty + U^l_{\text{reset}} S^l_{t-1}\right] \\ &= \mathbb{E}[I^l_t] + \beta^l\, \mathbb{E}[U^l_{t-1}] + (1-\beta^l)U^l_\infty + U^l_{\text{reset}}\, \mathbb{E}[S^l_{t-1}] \\ &\qquad \text{(linearity; deterministic } \{\beta^l, U^l_\infty, U^l_{\text{reset}}\}) \\ &= \mathbb{E}[I^l_t] + \beta^l\, \mathbb{E}[U^l_{t-1}] + U^l_{\text{reset}}\, \mathbb{P}(S^l_{t-1} = 1) \\ &\qquad (U^l_\infty = 0;\ \mathbb{E}[S] = \mathbb{P}(S=1)\cdot 1 \text{ for Bernoulli}) \end{aligned} \tag{44}$$

The recursive membrane potential's expectation is therefore,

$$\rightarrow \mathbb{E}[\tilde{U}^l_{t-1}] = \beta^l\, \mathbb{E}[U^l_{t-1}] + U^l_{\text{reset}}\, \mathbb{P}(S^l_{t-1} = 1)$$

No masked variable appears: because a spike does not delete the state, the first moment of the state enters untruncated, unlike the masked expectation of Equation 30. The variance follows analogously:

$$\operatorname{Var}(U^l_t) = \operatorname{Var}(I^l_t) + \operatorname{Var}(\tilde{U}^l_{t-1}) \tag{45}$$

(independence of input and recurrence, (IV))

$$= \mathrm{Var}(I_t^l) + \mathrm{Var}\left(\beta^l U_{t-1}^l + U_{\text{reset}}^l S_{t-1}^l\right)$$
(deterministic $(1-\beta^l)U_\infty^l$)
$$= \mathrm{Var}(I_t^l) + (\beta^l)^2\,\mathrm{Var}(U_{t-1}^l) + (U_{\text{reset}}^l)^2\,\mathrm{Var}(S_{t-1}^l) + 2\,\beta^l U_{\text{reset}}^l\,\mathrm{Cov}(U_{t-1}^l, S_{t-1}^l)$$
(expand variance; deterministic $\{\beta^l, U_{\text{reset}}^l\}$)

The recursive membrane potential's variance is therefore,

$$\to \mathrm{Var}(\tilde{U}_{t-1}^l) = (\beta^l)^2\,\mathrm{Var}(U_{t-1}^l) + (U_{\text{reset}}^l)^2\,\mathbb{P}(S_{t-1}^l=1)\left(1-\mathbb{P}(S_{t-1}^l=1)\right) + 2\,\beta^l U_{\text{reset}}^l\,\mathrm{Cov}(U_{t-1}^l, S_{t-1}^l)$$

The three terms are the variance of the leaked state, the Bernoulli variance of the reset term, and the covariance between the state and its own spike indicator. Only the last is non-elementary.

Covariance of state and spike. The covariance reduces to a truncated moment on the super-threshold clause:

$$\mathrm{Cov}(U_{t-1}^l, S_{t-1}^l) = \mathbb{E}[U_{t-1}^l S_{t-1}^l] - \mathbb{E}[U_{t-1}^l]\,\mathbb{E}[S_{t-1}^l] \tag{46}$$
(covariance definition)
$$\mathbb{E}[U_{t-1}^l S_{t-1}^l] = \mathbb{P}(S_{t-1}^l=1)\cdot\mathbb{E}[U_{t-1}^l\cdot 1 \mid S_{t-1}^l=1] + \mathbb{P}(S_{t-1}^l=0)\cdot\mathbb{E}[U_{t-1}^l\cdot 0 \mid S_{t-1}^l=0]$$
(law of total probability)
$$= \mathbb{P}(S_{t-1}^l=1)\cdot\mathbb{E}[U_{t-1}^l \mid U_{t-1}^l \geq \theta^l]$$
(masked relationship to truncated, super-threshold clause)
$$= p^l\left(\mu_{U^l} + r^+\sigma_{U^l}\right), \qquad r^+ = \frac{\phi(\alpha)}{1-\Phi(\alpha)}$$
(left-truncated Gaussian expectation under (A1)–(A2); inverse *Mills ratio* r^+ for $U_{t-1}^l \geq \theta^l$)
$$\to \quad \mathrm{Cov}(U_{t-1}^l, S_{t-1}^l) = p^l\left(\mu_{U^l} + r^+\sigma_{U^l}\right) - \mu_{U^l}\,p^l = p^l\,r^+\sigma_{U^l} = \sigma_{U^l}\,\phi(\alpha)$$
(substitute $p^l = 1-\Phi(\alpha)$)

The masked-to-truncated step is the relation of Equation 30, applied to the clause $S_{t-1}^l = 1$. The truncation is therefore from below, and the relevant ratio is $r^+ = \phi(\alpha)/(1-\Phi(\alpha))$ rather than the sub-threshold $r = \phi(\alpha)/\Phi(\alpha)$ of Section B.2.5. The two are linked by $p^l r^+ = (1-p^l)\,r = \phi(\alpha)$, so the final form of the covariance needs neither ratio.

Closed form. Substituting Equation 46 into Equation 44 and Equation 45, with $U_{t-1}^l \sim \mathcal{N}(\mu_{U^l}, \sigma_{U^l}^2)$ as in Equation 34, gives the soft-reset moment function f_{soft}^*:

$$(\mu_{\tilde{U}_{t-1}^l}, \sigma_{\tilde{U}_{t-1}^l}^2) = f_{\text{soft}}^*(\mu_{U^l}, \sigma_{U^l}^2;\, \theta^l, \beta^l, U_{\text{reset}}^l) \tag{47}$$
(desired moments, subject to parameters $\{\theta^l, \beta^l, U_{\text{reset}}^l\}$)
$$\alpha = \frac{\theta^l - \mu_{U^l}}{\sigma_{U^l}} \;\to\; p^l = 1-\Phi(\alpha)$$
$$\mu_{\tilde{U}_{t-1}^l} = \beta^l\mu_{U^l} + p^l\,U_{\text{reset}}^l$$
$$\sigma_{\tilde{U}_{t-1}^l}^2 = (\beta^l)^2\sigma_{U^l}^2 + (U_{\text{reset}}^l)^2\,p^l(1-p^l) + 2\,\beta^l U_{\text{reset}}^l\,\sigma_{U^l}\,\phi(\alpha)$$

The weight moments are Equation 26 with these recurrence moments,

$$\mu_{\mathbf{W}_X} = \frac{(1-\beta^l)\,\mu_{U^l} - p^l\,U_{\text{reset}}^l}{p^{l-1}\,D_{\text{in}}}, \qquad \sigma_{\mathbf{W}_X}^2 = \frac{\sigma_{U^l}^2 - \sigma_{\tilde{U}_{t-1}^l}^2}{D_{\text{in}}\,p^{l-1}\,(1-p^{l-1})} - \mu_{\mathbf{W}_X}^2, \tag{48}$$

with the target mean of Equation 25 at a general rate and $\mu_{U^l} = \theta^l$ at the default; the weight construction of Equation 8 is unchanged. The weight variance is positive only if the recurrence

leaves part of the target variance to the input current:

$$\sigma^2_{\tilde{U}^l_{t-1}} < \sigma^2_{U^l} \quad \Longleftrightarrow \quad \sigma^2_{U^l}\left(1-(\beta^l)^2\right) - (U^l_{\text{reset}})^2\, p^l(1-p^l) - 2\,\beta^l U^l_{\text{reset}}\,\sigma_{U^l}\,\phi(\alpha) > 0, \quad (49)$$

which is checked per neuron as in Algorithm 1. For a given U^l_{reset}, Equation 47 is linear in β^l in its mean and quadratic in its variance, so the soft-reset closure is, like the hard-reset one (Section B.5.1), defined on the closed interval $\beta^l \in [0,1]$: at $\beta^l = 1$ it describes the soft-reset IF neuron, and at $\beta^l = 0$ the recurrence contracts to the one-step offset $U^l_{\text{reset}}\, S^l_{t-1}$.

Reset. As for the hard reset (Section B.2.6), the closure holds for any U^l_{reset} that leaves a positive weight variance in Equation 48, which requires Equation 49, and the derivation advises a value rather than fixing one. The variance left to the input current, $\sigma^2_{U^l} - \sigma^2_{\tilde{U}^l_{t-1}}$, is largest when the recurrence variance of Equation 47 is smallest. That variance is quadratic in U^l_{reset}; setting its derivative to zero gives the advised reset and the recurrence variance it leaves,

$$U^{l,*}_{\text{reset}} = -\frac{\beta^l\,\sigma_{U^l}\,\phi(\alpha)}{p^l\,(1-p^l)}, \qquad \sigma^2_{\tilde{U}^l_{t-1}} = (\beta^l)^2\,\sigma^2_{U^l}\left(1 - \frac{\phi(\alpha)^2}{p^l\,(1-p^l)}\right). \quad (50)$$

Because $\phi(\alpha)^2 < p^l(1-p^l)$, this reset satisfies Equation 49 for every $\beta^l \in [0,1]$; for $\beta^l > 0$ it is negative, so a spike lowers the membrane potential. At the default target, $\alpha = 0$ and $p^l = 50\%$, it is $U^{l,*}_{\text{reset}} = -4\beta^l\sigma_{U^l}\phi(0) \approx -1.436$ for $\beta^l = 0.9$ and $\sigma_{U^l} = 1$. Every soft-reset simulation (Section C.3.4) uses this reset. It differs from the conventional soft reset, which subtracts the threshold, $U^l_{\text{reset}} = -\theta^l$; which of the two to use is a design choice, depending on how precisely the targeted stationary distribution is to be met.

The hard reset is the more demanding case for the closure. Each spike deletes the accumulated state, and the two spike clauses have disjoint supports. The hard-reset moments therefore need the masked-to-truncated relation for both mean and variance and the covariance of the two masked clauses (Equation 32). The soft-reset moments need a single truncated moment inside one covariance.

B.6 Extension B: heterogeneous populations

Extension B evaluates the neuron side of the contract (Equation 37) per neuron. The derivation of Section B.2 conditions only on the hyperparameters of one neuron. This subsection writes the closure per neuron and states the one condition under which a layer of neurons with different thresholds and leak factors is initialized by the same closed form. That condition is a common rate of any value. Heterogeneous neuron hyperparameters have been reported to promote robust learning in SNNs [Perez-Nieves et al., 2021]. In the empirical evaluation of this work, β^l is already a per-neuron learnable parameter initialized homogeneously at 0.9 (Table 8).

Per-neuron form. The neuron hyperparameters of layer l are vectors $\theta^l, \beta^l, U^l_{\text{reset}} \in \mathbb{R}^{D_{\text{out}}}$ with entries θ^l_j, β^l_j and $U^l_{\text{reset},j}$ for output neuron j. The target moments may likewise be set per neuron, $U^l_{t,j} \sim \mathcal{N}(\mu_{U^l_j}, \sigma^2_{U^l_j})$. Neuron j receives $I^l_{t,j} = [\mathbf{W}^l_X]_{j,:}\, X^l_t$, and its membrane recurrence involves no hyperparameter but its own. Its recurrence moments are Equation 39 on its own hyperparameters with $U^l_\infty = 0$. They are evaluated at its normalized threshold $\alpha^l_j = (\theta^l_j - \mu_{U^l_j})/\sigma_{U^l_j}$, inverse Mills ratio $r^l_j = \phi(\alpha^l_j)/\Phi(\alpha^l_j)$ and stationary spike rate $p^l_j = 1 - \Phi(\alpha^l_j)$. Row j takes the weight moments of Equation 26 with the per-neuron target moments of Equation 37 as numerators. A heterogeneous layer is therefore initialized by the same closed form applied row by row, and no step of the derivation is re-derived. Algorithm 1 draws the orthogonal base once per layer and scales and shifts its rows one by one. Row-wise scaling preserves the mutual orthogonality of the rows. Heterogeneity thus leaves the construction that supports (A4) unchanged.

Realized hyperparameters. Assumption (A6) requires $\{\theta^l, \beta^l, U^l_{\text{reset}}\}$ to be known and deterministic. When heterogeneity is generated by drawing θ^l_j and β^l_j from distributions, the hyperparameters are drawn first and the initialization is computed conditional on the realized values. Given the draw, every entry is a known constant, (A6) holds, and the closure applies to each neuron without covariance terms induced by the hyperparameter distribution. Distributions may differ between layers and be redrawn for every layer: each layer's initialization reads only its own realized vectors and, under the condition below, the rate of the layer it receives from.

Shared-rate condition. The closure of every neuron in layer l consumes a single incoming rate p^{l-1}: (A5) requires the entries of S_t^{l-1} to be approximately i.i.d. $\mathrm{Ber}(p^{l-1})$, which holds only if all neurons of layer $l-1$ fire at a common rate. Applied to layer l, which is the input of layer $l+1$, the per-neuron targets must realize one rate:

$$p_j^l = p^l \ \ \forall j \quad \Longleftrightarrow \quad \alpha_j^l = \Phi^{-1}(1-p^l) \ \ \forall j \quad \Longleftrightarrow \quad \mu_{U_j^l} = \theta_j^l - \sigma_{U_j^l}\,\Phi^{-1}(1-p^l). \tag{51}$$

A common rate thus requires a common normalized distance to threshold: every membrane mean sits the same number of standard deviations below its own threshold, not at the same fraction of it. A target $\mu_{U_j^l} = c\,\theta_j^l$ with one scalar c realizes a common rate only at $p^l = 50\%$ ($c = 1$) or for identical thresholds; with c chosen for $\theta_j^l = 1$ at $p^l = 0.15$ and $\sigma_{U_j^l} = 1$, neurons with $\theta_j^l = 0.5$ and 1.5 would fire at $1 - \Phi\big(\theta_j^l\,\Phi^{-1}(0.85)\big)$, i.e., at 0.302 and 0.060. At $p^l = 50\%$, Equation 51 reduces to $\mu_{U_j^l} = \theta_j^l$, i.e., objective (B) of Equation 3 per neuron. Under Equation 51, $\alpha_j^l = \alpha^l$, $r_j^l = r^l$ and $p_j^l = p^l$ are layer constants, the denominators of Equation 26 are common to all rows, and neurons differ only through θ_j^l, β_j^l and $U_{\text{reset},j}^l$. Heterogeneity is therefore admissible in the dynamics that lead to a spike and not in the stationary spike probability; the downstream layer, itself possibly heterogeneous, receives the single Bernoulli rate its own initialization assumes. The condition binds within a layer only: Ψ takes p^{l-1} and p^l as separate arguments (Equation 24), so the common rate may differ between layers.

Admissibility is a per-row check that Algorithm 1 performs before it draws the row. A real $\sigma_{\mathbf{W}_X,j}$ requires at least $\sigma^2_{\tilde{U}^l_{t-1,j}} < \sigma^2_{U_j^l}$. In Equation 39 the within-clause term stays below $(1-p_j^l)\,\sigma^2_{U_j^l}$ for any $\beta_j^l \in [0,1]$, since $1 - \alpha r - r^2 \in (0,1)$ is the variance ratio of a right-truncated Gaussian. The between-clause term grows with the gap between $U_{\text{reset},j}^l$ and the mean of the leaked sub-threshold state.

B.7 Extension C: input current

Extension C changes the current side of the contract. The operator that forms the input current enters the closure in two places. First, it maps its parameters to the delivered moments of Equation 38. Second, the operator decides whether the assumptions behind the closure hold for the current it produces: the number of active inputs $D_{\text{in},s}\,p_s$ sets the central-limit budget behind (A1) (Section B.3.1); a recurrent stream from the same layer can correlate $I_{t,j}^l$ with $U_{t-1,j}^l$, against (A3); weight sharing correlates the currents of neighboring output units, against (A4); and orthogonalizing each stream separately does not decorrelate the currents of different streams. The two operators evaluated in this work are treated in Section B.7.1 and Section B.7.2.

B.7.1 Residual and recurrent connections

This subsection treats neurons whose input current is the sum of several streams: the feed-forward stream of Equation 1, a residual stream through space, and a recurrent stream through time. Residual connections can facilitate activity propagation in deep SNNs [Hu et al., 2025]; here they enter only through the input current they contribute. The recurrence moments and the target membrane moments are unchanged. The additional streams add clauses to the dependence structure of Equation 27 and introduce one design choice, the allocation of the current budget across the streams.

Input streams. For layer l with D_{out} neurons, the input current is the sum of three streams,

$$\begin{aligned} I_t^l &= I_{X,t}^l + I_{\text{res},t}^l + I_{\text{rec},t}^l + \mathbf{b}_X^l \\ I_{X,t}^l &= \mathbf{W}_X^l\,S_t^{l-1}, \qquad I_{\text{res},t}^l = \mathbf{W}_{\text{res}}^l\,S_t^{l-k}, \qquad I_{\text{rec},t}^l = \mathbf{W}_{\text{rec}}^l\,S_{t-h}^l \end{aligned} \tag{52}$$

namely the feed-forward stream from layer $l-1$, a residual stream from layer $l-k$ with $k \geq 2$, and a recurrent stream from the layer's own spikes $h \geq 1$ steps earlier. As in Algorithm 1, we index the streams by $s \in \mathcal{S} \subseteq \{X, \text{res}, \text{rec}\}$, with weights $\mathbf{W}_s^l \in \mathbb{R}^{D_{\text{out}} \times D_{\text{in},s}}$, source spike rate p_s and fan-in $D_{\text{in},s}$: $(p_X, D_{\text{in},X}) = (p^{l-1}, D_{\text{in}})$; $p_{\text{res}} = p^{l-k}$, with $D_{\text{in,res}}$ the width of layer $l-k$; and $p_{\text{rec}} = p^l$, with $D_{\text{in,rec}} = D_{\text{out}} - 1$ afferents per row once the diagonal of $\mathbf{W}_{\text{rec}}^l$ is masked. The LIF update of Equation 1 is unchanged, $U_t^l = I_t^l + \tilde{U}_{t-1}^l$, with I_t^l now the total current; for $\mathbf{W}_{\text{res}}^l = \mathbf{W}_{\text{rec}}^l = \mathbf{0}$ it

is the feed-forward current of the main text. The construction covers residual streams that enter the input current through a weight matrix. Shortcuts added to the membrane potential or to the output spikes change the quantity that is budgeted and are not treated here; parameter-free identity shortcuts are not evaluated.

Dependence assumptions. Because the streams add, the decomposed moments of Equation 28 and Equation 29 carry over to the total current if the dependence structure of Equation 27, with (I) and (IV) read for the feed-forward stream, is extended by

$$\begin{aligned}
&\text{(VI-res)} : \operatorname{Cov}\left(I^l_{X,t},\, I^l_{\text{res},t}\right) \approx 0 && (53)\\
&\qquad\qquad\qquad\qquad\qquad\qquad \text{(feed-forward and residual streams)}\\
&\text{(VI-rec)} : \operatorname{Cov}\left(I^l_{X,t},\, I^l_{\text{rec},t}\right) \approx 0\\
&\qquad\qquad\qquad\qquad\qquad\qquad \text{(feed-forward and recurrent streams)}\\
&\text{(VI-cross)} : \operatorname{Cov}\left(I^l_{\text{res},t},\, I^l_{\text{rec},t}\right) \approx 0\\
&\qquad\qquad\qquad\qquad\qquad\qquad \text{(residual and recurrent streams)}\\
&\text{(VI-mem)} : \operatorname{Cov}\left(I^l_{\text{res},t} + I^l_{\text{rec},t},\, \tilde{U}^l_{t-1}\right) \approx 0\\
&\qquad\qquad\qquad\qquad\qquad\qquad \text{(auxiliary streams and membrane recurrence)}
\end{aligned}$$

which gives

$$\mathbb{E}[U^l_t] = \sum_{s\in\mathcal{S}} \mathbb{E}\big[I^l_{s,t}\big] + \mathbf{b}^l_X + \mathbb{E}\big[\tilde{U}^l_{t-1}\big], \qquad \operatorname{Var}(U^l_t) = \sum_{s\in\mathcal{S}} \operatorname{Var}\left(I^l_{s,t}\right) + \operatorname{Var}\left(\tilde{U}^l_{t-1}\right), \tag{54}$$

with $\operatorname{Var}(\tilde{U}^l_{t-1})$ expanded as in Equation 29; (VI-mem) is the counterpart of (I) and (IV) for the auxiliary streams.

The clauses of (VI) are assumptions and not identities. Each stream receives its own independently drawn orthogonal base. Orthogonality approximates (A4) within a stream but does not decorrelate two streams. Their covariance vanishes when the sources are independent. For dependent sources it vanishes only in expectation over the weight draw, because the bases are independent and the auxiliary weights have zero mean. In a stack of layers the residual source S^{l-k}_t drives S^{l-1}_t through the intermediate layers. (VI-res) therefore holds only in this averaged sense, and the temporal correlation of the sources affects (VI-rec) and (VI-cross) in the same way. The residual stream correlates with $\tilde{U}^l_{t-1}$ only through temporal correlations of the spikes, which (I) already neglects for the feed-forward stream. The recurrent stream is different. Its diagonal entry $W^l_{\text{rec},jj}$ feeds the spike history of neuron j back into its own current. For $h = 1$ this is the event $S^l_{t-1,j}$ that switches the reset clause of $\tilde{U}^l_{t-1,j}$, so this covariance is not small. We therefore mask the diagonal of $\mathbf{W}^l_{\text{rec}}$ at initialization. The remaining entries couple neuron j to neurons $i \neq j$ whose membranes are approximately uncorrelated with its own under (A4).

Allocation of the current budget across streams. With the stream moments of Equation 38 and (VI), the per-neuron budget of Equation 37 is met if

$$\sum_{s\in\mathcal{S}} D_{\text{in},s}\, p_s\, \mu_{\mathbf{W}_s,j} = \mu_{I^l_j}, \qquad \sum_{s\in\mathcal{S}} D_{\text{in},s}\, p_s(1-p_s)\left(\sigma^2_{\mathbf{W}_s,j} + \mu^2_{\mathbf{W}_s,j}\right) = \sigma^2_{I^l_j}. \tag{55}$$

Two equations constrain six weight moments per neuron: the closure fixes the budget, not its allocation. As in Algorithm 1, we place the mean on the feed-forward stream and divide the variance by fractions $\omega_X + \omega_{\text{res}} + \omega_{\text{rec}} = 1$, $\omega_s \geq 0$:

$$\begin{aligned}
\mu_{\mathbf{W}_X,j} &= \frac{\mu_{I^l_j}}{D_{\text{in}}\, p^{l-1}}, & \sigma^2_{\mathbf{W}_X,j} &= \frac{\omega_X\, \sigma^2_{I^l_j}}{D_{\text{in}}\, p^{l-1}(1-p^{l-1})} - \mu^2_{\mathbf{W}_X,j} && (56)\\
\mu_{\mathbf{W}_{\text{res}},j} &= 0, & \sigma^2_{\mathbf{W}_{\text{res}},j} &= \frac{\omega_{\text{res}}\, \sigma^2_{I^l_j}}{D_{\text{in,res}}\, p^{l-k}(1-p^{l-k})}\\
\mu_{\mathbf{W}_{\text{rec}},j} &= 0, & \sigma^2_{\mathbf{W}_{\text{rec}},j} &= \frac{\omega_{\text{rec}}\, \sigma^2_{I^l_j}}{(D_{\text{out}}-1)\, p^l(1-p^l)}
\end{aligned}$$

For $\omega_X = 1$ both auxiliary matrices vanish and $(\mu_{\mathbf{W}_X,j}, \sigma^2_{\mathbf{W}_X,j})$ is Equation 7, which is written there for homogeneous hyperparameters, so that all rows coincide. A stream without a source, such as the residual stream of a layer with $l \leq k$, is removed and its fraction returned to the feed-forward stream; with heterogeneous hyperparameters the rows differ only through the per-neuron budget (Section B.6). Each matrix is built from its own orthogonal base,

$$\left[\mathbf{W}_s^l\right]_{j,:} = \sigma_{\mathbf{W}_s,j} \left[\mathbf{Z}_s^l\right]_{j,:} + \mu_{\mathbf{W}_s,j}, \qquad s \in \mathcal{S}, \tag{57}$$

where $\mathbf{Z}_s^l$ is the orthogonal base $\mathbf{W}_{s,\perp}^l$ scaled as in Equation 8 for a $D_{\text{out}} \times D_{\text{in},s}$ matrix; for $s = \text{rec}$ the diagonal is excluded, $Z^l_{\text{rec},jj} = 0$. A small diagonal perturbation, $W^l_{\text{rec},jj} = \sigma_{\mathbf{W}_{\text{rec}},j}\, \zeta_j$ with $|\zeta_j| \ll 1$, can be kept if self-connections are wanted at initialization; it adds $p^l(1-p^l)\, \sigma^2_{\mathbf{W}_{\text{rec}},j}\, \zeta_j^2$ to the variance outside the budget and re-introduces the covariance of (VI-mem) in proportion to ζ_j.

Placing the mean on the feed-forward stream keeps the correction of $\mu_{\tilde{U}^l_{t-1}}$ on the one stream that every layer has. Adding or removing an auxiliary stream then changes only the allocation of the variance. A zero-mean stream contributes no mean current in expectation over the weight draw. A departure of p^{l-k} or p^l from its target therefore changes the variance of the corresponding stream through $p_s(1-p_s)$ but not the mean of the current. Other placements are possible. For instance, Herranz-Celotti and Rouat [2024] give the recurrent weights a non-zero mean (Section A.1). How to divide the variance across streams is unsettled in the SNN literature [Rossbroich et al., 2022, Huebotter et al., 2025, Herranz-Celotti and Rouat, 2024], and we assert no default. The closed-form budget makes the choice of $(\omega_X, \omega_{\text{res}}, \omega_{\text{rec}})$ a well-posed question at fixed target moments.

Recurrent streams. The recurrent stream differs from the residual stream in two respects. Its source is the layer's own output, so its rate $p_{\text{rec}} = p^l$ is the target itself. A layer at its target thus keeps the recurrent stream consistent with the temporal axis of the fixed point (Section B.1.2). The stream also feeds back. A departure of the layer's rate from p^l changes the variance of $I^l_{\text{rec},t}$ through $p^l(1-p^l)$ but not the mean of I^l_t, because $\mathbf{W}^l_{\text{rec}}$ has zero mean. With the spike state initialized at zero the stream carries no input during the first h steps. This lowers the variance of I^l_t by the fraction ω_{rec} over those steps and leaves its mean unchanged. The weight moments are those of Equation 56 with $D_{\text{out}} - 1$ afferents per row after the diagonal is masked. The behavior over time is not analyzed, in particular for large ω_{rec} or $h > 1$.

B.7.2 Convolutional layers

A convolutional layer changes the operator, not the neuron. This subsection states how the closed form is applied to a convolutional kernel and what weight sharing and the boundary of the feature map imply.

Kernel as a weight matrix. A convolutional layer l with C_{in} input channels, C_{out} output channels, square kernels of odd size $k \times k$ and stride one holds a kernel $\mathbf{K}^l \in \mathbb{R}^{C_{\text{out}} \times C_{\text{in}} \times k \times k}$ and a bias $\mathbf{b}^l_X \in \mathbb{R}^{C_{\text{out}}}$. Its LIF neurons are indexed by an output channel c and a position $\mathbf{x} \in \Omega$ of the feature map, so the layer holds $C_{\text{out}}|\Omega|$ neurons. With $\mathcal{K}$ the set of the k^2 kernel offsets, the input current of neuron $(c, \mathbf{x})$ is

$$I^l_{t,c,\mathbf{x}} = \sum_{c'=1}^{C_{\text{in}}} \sum_{\boldsymbol{\delta} \in \mathcal{K}} K^l_{c,c',\boldsymbol{\delta}}\, S^{l-1}_{t,c',\mathbf{x}+\boldsymbol{\delta}} + b^l_{X,c} = \left(\mathbf{W}^l_X\, \mathbf{s}^{l-1}_{t,\mathbf{x}}\right)_c + b^l_{X,c}, \qquad \mathbf{W}^l_X \in \mathbb{R}^{C_{\text{out}} \times C_{\text{in}} k^2}, \tag{58}$$

where $\mathbf{W}^l_X$ is $\mathbf{K}^l$ reshaped to one row per output channel and $\mathbf{s}^{l-1}_{t,\mathbf{x}} \in \mathbb{B}^{C_{\text{in}} k^2}$ is the patch of presynaptic spikes around $\mathbf{x}$, unfolded in the same order. Every neuron therefore receives a scalar current formed by a matrix with $D_{\text{out}} = C_{\text{out}}$ rows and $D_{\text{in}} = C_{\text{in}} k^2$ columns; D_{in} is the fan-in of He et al. [2015] and of the framework defaults [PyTorch, 2022]. We apply Equation 7 and the orthogonal construction of Equation 8 to $\mathbf{W}^l_X$ with the mean placed in the bias (see below) and reshape the result to $\mathbf{K}^l$; this is the tensor-view step of Algorithm 1. The frameworks also define a fan-out, $C_{\text{out}} k^2$, the number of neurons to which one presynaptic neuron projects. The fan-out sets the variance of the backward pass [He et al., 2015]; it is not the number of rows of $\mathbf{W}^l_X$ and does not enter the forward moments. The aspect-ratio correction of Equation 8 applies only when $C_{\text{out}} > C_{\text{in}} k^2$.

Kernel slices as parallel streams. An equivalent reading treats the C_{in} slices $\mathbf{K}^l_{c,c'} \in \mathbb{R}^{k\times k}$ of a row as parallel input streams of one scalar current (Section B.7.1). Under (A5) the slices read disjoint presynaptic neurons, so their current means and variances add. A common variance for all entries of a row allocates the budget of Equation 37 equally across slices and offsets. Any other allocation with the same totals meets the same per-neuron targets without re-deriving them. The best allocation that an initialization should encode along the spatial dimensions for convolutional architectures is active research [Pedersen et al., 2025] and therefore a natural next step beyond this work. We use the equal allocation and assert no default.

Weight sharing. All $|\Omega|$ neurons of channel c read their patches through the same row $(\mathbf{W}^l_X)_{c,:}$ and the same bias $b^l_{X,c}$. The per-neuron requirement of Equation 37 must therefore coincide for all neurons of a channel: one set $\{\theta^l_c, \beta^l_c, U^l_{\text{reset},c}\}$ and one target rate per channel. Heterogeneous hyperparameters (Section B.6) are admissible across channels but not across positions. Orthogonal rows support (A4) across the channels at one position. Neurons at nearby positions read overlapping patches through shifted rows, so (A4) is questionable along the spatial dimensions.

Bias placement and padding. We carry the mean correction in the explicit bias $b^l_{X,c} = \mu_{U^l} - \mu_{\tilde{U}^l_{t-1}}$, the term $p^{l-1} D_{\text{in}}\, \mu_{\mathbf{W}_X}$ that Equation 7 places in the weight mean. The reason is the boundary of the feature map. Zero padding is the framework convention. A neuron at the border then sums only the subset $\mathcal{P}_{\mathbf{x}}$ of its row whose afferents lie inside the map, and we let $\kappa_{\mathbf{x}} = D_{\text{in}} - |\mathcal{P}_{\mathbf{x}}|$ be the number of excluded entries. A mean carried by the weights reaches the neuron only through these afferents, so it loses the fraction $\kappa_{\mathbf{x}}/D_{\text{in}}$ of the correction. For 3×3 kernels this is one third at edges and five ninths at corners. A bias bypasses the afferents and delivers the whole correction at every position. Inside the map and in a dense layer the two placements deliver the same mean.

We therefore standardize each row and scale it to the moments $\mu_{\mathbf{W}_X} = 0$ and $\sigma^2_{\mathbf{W}_X} = (\sigma^2_{U^l} - \sigma^2_{\tilde{U}^l_{t-1}})/(D_{\text{in}}\, p^{l-1}(1-p^{l-1}))$, which is Equation 26 at zero weight mean. For a neuron whose patch lies inside the map, (A5) then gives $\mathbb{E}[I^l_{t,c,\mathbf{x}} \mid \mathbf{W}^l_X] = b^l_{X,c}$ and $\mathrm{Var}(I^l_{t,c,\mathbf{x}} \mid \mathbf{W}^l_X) = p^{l-1}(1 - p^{l-1}) \sum_i (\mathbf{W}^l_X)^2_{c,i} = D_{\text{in}}\, p^{l-1}(1-p^{l-1})\, \sigma^2_{\mathbf{W}_X}$. A neuron at the border sums fewer afferents through the shared row, so its current has a random mean offset and a reduced variance. The per-neuron targets therefore hold only for neurons whose patch lies inside the map. The extent of this limitation is not evaluated in this work.

B.8 General initialization of one layer

Algorithm 1 assembles the contract of Section B.4 and Extensions A–C into one procedure for one LIF layer. It covers per-neuron hyperparameters, any target rate and a set $\mathcal{S}$ of input streams. Its comments name the extension that sets each line (Table 6). The main stream $s = X$ receives S^{l-1}_t, with $D_{\text{in},X} = D_{\text{in}}$ and $p_X = p^{l-1}$. The algorithm places the whole mean on the main stream and splits the variance by fractions $\omega_s \geq 0$ with $\sum_s \omega_s = 1$; both are design choices (Section B.7.1). The reset values enter as given inputs. The base case of the main text is the special case of a single stream with homogeneous $\{\theta^l, \beta^l, U^l_{\text{reset}}\}$ at $p^l = 50\%$. There the algorithm returns Equation 7 and Equation 8.

Algorithm 1 Closed-form initialization of one LIF layer l with per-neuron hyperparameters, any target rate p^l and input streams $s \in \mathcal{S}$. The comments name the extension that sets each line (Table 6).

Require: streams $s \in \mathcal{S}$ with fan-in $D_{\text{in},s}$, source rate p_s and variance fraction $\omega_s \geq 0$, $\sum_s \omega_s = 1$ (main stream $s = X$: $D_{\text{in},X} = D_{\text{in}}$, $p_X = p^{l-1}$); output dimension D_{out}; per-neuron $\{\theta_j^l, \beta_j^l, U_{\text{reset},j}^l\}$ (realized values if drawn from distributions, so that (A6) holds given the draw) and target variance $\sigma_{U_j^l}^2$ (default 1); target rate p^l; recurrence moments of the neuron model; carrier of the mean correction (weight mean or bias)

Ensure: weights $\mathbf{W}_s^l$ for $s \in \mathcal{S}$ and bias $\mathbf{b}_X^l$; layer $l+1$ receives the incoming rate p^l

$\alpha^l \leftarrow \Phi^{-1}(1 - p^l)$ ▷ target rate, any p^l; shared by all neurons
for $j = 1, \ldots, D_{\text{out}}$ **do** ▷ Extension B: row by row
  $\mu_{U_j^l} \leftarrow \theta_j^l - \sigma_{U_j^l}\, \alpha^l$ ▷ Equation 25
  $\left(\mu_{\tilde{U}_{t-1,j}^l}, \sigma_{\tilde{U}_{t-1,j}^l}^2\right) \leftarrow$ recurrence moments at $\left(\mu_{U_j^l}, \sigma_{U_j^l}^2; \theta_j^l, \beta_j^l, U_{\text{reset},j}^l\right)$ ▷ Extension A
  if $\sigma_{\tilde{U}_{t-1,j}^l}^2 \geq \sigma_{U_j^l}^2$ **then**
    stop: no admissible weight variance for neuron j
  end if
  $\mu_{I_j^l} \leftarrow \mu_{U_j^l} - \mu_{\tilde{U}_{t-1,j}^l}$; $\quad \sigma_{I_j^l}^2 \leftarrow \sigma_{U_j^l}^2 - \sigma_{\tilde{U}_{t-1,j}^l}^2$ ▷ Equation 37
end for
remove the streams without a source, e.g., a residual stream for $l \leq k$, and add their fractions to ω_X ▷ Extension C: input current
for $s \in \mathcal{S}$ **do** ▷ Extension C: input current
  $\mu_{\mathbf{W}_s,j} \leftarrow \mu_{I_j^l} / \left(p_X D_{\text{in},X}\right)$ for all j if $s = X$ and the weight mean carries the correction; else $\mu_{\mathbf{W}_s,j} \leftarrow 0$
  $\sigma_{\mathbf{W}_s,j}^2 \leftarrow \omega_s\, \sigma_{I_j^l}^2 / \left(D_{\text{in},s}\, p_s(1-p_s)\right) - \mu_{\mathbf{W}_s,j}^2$ for all j ▷ variance split by fractions
  stop if $\sigma_{\mathbf{W}_s,j}^2 < 0$ for some j: no admissible weight variance
  $\left[\mathbf{W}_s^l\right]_{j,:} \leftarrow$ row j of the orthogonal construction of Equation 8 for a $D_{\text{out}} \times D_{\text{in},s}$ matrix, with $(\mu_{\mathbf{W}_s,j}, \sigma_{\mathbf{W}_s,j})$ in place of $(\mu_{\mathbf{W}_X}, \sigma_{\mathbf{W}_X})$; one orthogonal base per stream and layer, with zero diagonal for $s = \text{rec}$ ▷ Extension C: rows of the unfolded kernel for convolution
end for
$b_{X,j}^l \leftarrow \mu_{I_j^l}$ for all j if the bias carries the correction; else $\mathbf{b}_X^l \leftarrow \mathbf{0}$

C Appendix - results

C.1 Simulation study: setup and hyperparameters

The simulation study propagates input through a deep spiking network at initialization, and measures how closely each layer meets the targets of our initialization: the target spike rate p and unit membrane-potential variance. The hyperparameters in Table 7 apply to every initialization method and to every study of Section 3.1 and Section C.3, except where a subsection states a change. Methods differ only in the initialization procedure itself; all other settings (architecture, input statistics, sample budget, simulation horizon, neuron parameters, seeds) are shared.

Simulation protocol. Each run propagates a batch of $|\mathcal{B}| = 500$ binary input samples for $T = 40$ time steps through the base network of Section 3.1: 30 layers of 100 LIF neurons (Equation 1). All neurons share the threshold $\theta = 1$ (0.52 under Lee et al. [2016]), the leak $\beta = 0.9$ and the reset value U_{reset}^l (0 under the alternative initializations and as stated in Section B.2.6 under ours) of Table 7, except where an extension study changes them (Section C.3.4, Section C.3.5). Every layer is initialized for the target rate p, 50% by default, the maximum-entropy rate of objective B (Equation 3); Table 2 and the appendix studies cover the rate grid of Table 7. The statistics use only the time steps $t \in \mathcal{T} = \{20, \ldots, 39\}$: the closed form targets the stationary membrane-potential distribution (A2), and the first 20 steps are assumed to carry the transient from the initial state and are omitted from the analysis. Each configuration is repeated over 10 random seeds.

Input encoder. The network input is binary. At every time step, a Heaviside encoder thresholds, for every sample, a vector of $D_{\text{in}}^{\text{enc}}$ independent standard-normal entries,

$$\begin{aligned} Z_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{D_{\text{in}}^{\text{enc}} \times 1}), \qquad \theta^{\text{enc}} = \Phi^{-1}(1-p), \\ S_t^{\text{enc}} = H(Z_t; \theta^{\text{enc}}) \sim \text{Ber}(p)^{\otimes D_{\text{in}}^{\text{enc}}} \;\Rightarrow\; \|S_t^{\text{enc}}\|_1 \sim \text{Bin}(D_{\text{in}}^{\text{enc}}, p), \end{aligned} \tag{59}$$

with Φ the standard-normal CDF. Each entry exceeds the threshold with probability $1-\Phi(\theta^{\text{enc}}) = p$, independently of the others, so the encoder spikes are i.i.d. Bernoulli(p) and their count is Binomial: the first LIF layer receives $X_t^1 = S_t^{\text{enc}}$, exactly the input of assumption (A5) (Section 2.3), at the rate p for which every layer is initialized. At the default rate $p = 50\%$ the threshold is $\theta^{\text{enc}} = 0$; at $p = 15\%$ it is $\theta^{\text{enc}} \approx 1$, the threshold value common in SNN frameworks [Pehle and Pedersen, 2021, Eshraghian et al., 2023]. The assumption itself is tested in Section C.3.3, where the encoder departs from it in the rate, the identical distribution and the independence of its spikes. Two data-agnostic mechanisms are proposed to satisfy the input assumption. In the simulation, the binarizing encoder satisfies it by construction. In the empirical evaluation, a batch-normalization module is placed before the first LIF layer; its affine parameters are initialized to match the target moments of the input current (Eq. (63) in Appendix C.2). No additional batch normalization is or needs to be used elsewhere in the SNN.

Statistics. Equation 10 summarizes the statistics. In full, for neuron m of layer l in one seed, the spike rate and the mean and variance of the membrane potential are estimated over its $|\mathcal{B}||\mathcal{T}| = 500 \cdot 20 = 10^4$ observations,

$$\begin{aligned} \hat{p}_m^l = \tfrac{1}{|\mathcal{B}||\mathcal{T}|} \textstyle\sum_{b,t} S_{t,b,m}^l, \qquad \hat{\mu}_{U_m^l} = \tfrac{1}{|\mathcal{B}||\mathcal{T}|} \textstyle\sum_{b,t} U_{t,b,m}^l, \\ \hat{\sigma}^2_{U_m^l} = \tfrac{1}{|\mathcal{B}||\mathcal{T}|} \textstyle\sum_{b,t} \left(U_{t,b,m}^l - \hat{\mu}_{U_m^l}\right)^2, \end{aligned} \tag{60}$$

so that $\bar{p}^l$ and $\bar{\sigma}^2_{U^l}$ are the means of $\hat{p}_m^l$ and $\hat{\sigma}^2_{U_m^l}$ over the M neurons of the layer. Table 2 and the tables of Section C.3 report each mean with the standard deviation of the per-neuron values across the M neurons; both are computed per seed and averaged over seeds, so $\pm$ is the neuron-to-neuron spread, not run-to-run error. Averaging per-neuron variances is stricter than pooling the layer: by the law of total variance, the pooled variance of Section 3.1 decomposes as

$$\tfrac{1}{M|\mathcal{B}||\mathcal{T}|} \textstyle\sum_{m,b,t} \left(U_{t,b,m}^l - \hat{\mu}_{U^l}\right)^2 = \bar{\sigma}^2_{U^l} + \tfrac{1}{M} \textstyle\sum_m \left(\hat{\mu}_{U_m^l} - \hat{\mu}_{U^l}\right)^2, \tag{61}$$

so a layer whose neurons scatter in their means can meet the unit-variance target when pooled while no single neuron does. That scatter moves each neuron's mean relative to the threshold and therefore appears as spread in $\hat{p}_m^l$. With 10^4 observations per neuron, the binomial standard error of $\hat{p}_m^l$ for independent observations is $\sqrt{p(1-p)/10^4} \leq 0.005$, that is 0.5 percentage points (pp). Since sampling noise adds to the true neuron-to-neuron spread, a spike-rate spread near 0.5 pp is an upper bound on it.

C.2 Empirical evaluation: setup and hyperparameters

The empirical evaluation in Section 3.2 sweeps a 3×4 width–depth grid (width $\in \{32, 64, 128\}$, depth $\in \{2, 3, 4, 5\}$) with 20 random seeds per cell, held constant across paradigms. Optimization hyperparameters are likewise held constant: *no per-paradigm tuning* of optimizer, learning rate, or SNN-side hyperparameters is performed. The shared optimization hyperparameters are reported first (Table 8) and the per-task specifications second (Table 9).

Initialization configurations. The five configurations of Section 3.2 differ only in how the weights, the batch-normalization affine parameters and the reset value are initialized; every configuration initializes the biases to zero. (1) The *naive* baseline uses PyTorch's default initialization, Equation 15. (2) The *default* configuration uses the closed form of Micheli et al. [2025] in the per-neuron form of Section A.1. Its zero-mean Gaussian weights have variance $1/(D_{\text{in}}\, p(1-p))$ at the spike probability $p = 15\%$, which approximates the $1-\Phi(\theta) \approx 15.9\%$ of its standard-normal membrane assumption. (3) The *enhanced* configuration rescales an orthogonal matrix to the variance of (2). Configurations (1)–(3) keep the standard initialization (i) of the batch-normalization affine parameters below. (4)–(5) Our initialization targets the membrane-potential distribution $U_t^l \sim \mathcal{N}(0, 1)$ in (4) and $U_t^l \sim \mathcal{N}(\theta, 1)$ in (5): it evaluates Equation 8 at $\mu_{U^l} = 0$ and $\sigma^2_{U^l} = 1$ in (4), so that

Table 7: **Simulation-study hyperparameters.** Settings shared across every initialization method and every study in Section 3.1 and Section C.3, except where a subsection states a change. Per-method differences concern the initialization procedure only.

Hyperparameter	**Value**
Network depth (number of LIF layers)	30
Layer width (neurons per layer)	100
LIF threshold θ	1
LIF leak factor β	0.9
Reset	hard (to U^l_{reset})
Number of input samples	500
Number of simulation time steps T	40
Time steps pooled for the statistics	$t = 20, \ldots, 39$
Reporting layers	1, 5, 10, 20, 30 (30 deepest)
Input distribution Z_t	$\mathcal{N}(\mathbf{0}, \mathbf{I})$
Encoder type	Heaviside binariser
Encoder input dimension $D^{\text{enc}}_{\text{in}}$	100
Encoder threshold θ^{enc}	$\Phi^{-1}(1-p)$
Target spike rate p (encoder and every layer)	50%
Rate grid	15, 35, 50, 65, 85%
Initialization methods	ours and nine alternatives (Section A.1)
Number of random seeds	10

$\alpha = \theta = 1$ and $p^l = 1 - \Phi(1) \approx 15.9\%$ through the general-rate form of Section B.1.3; and at $\mu_{U^l} = \theta$ and $\sigma^2_{U^l} = 1$ in (5), so that $\alpha = 0$ and $p^l = 50\%$. Configuration (4) thus operates near the rate $p = 15\%$ of (2)–(3). Both use the initialization (ii) of the batch-normalization affine parameters below. In RL, both also scale the actor's output layer by $\sqrt{q(1-q)/(p^l(1-p^l))}$, with p^l the target rate of the last LIF layer and $q = 0.15$. Every network consists of LIF neurons (Equation 1) with the threshold $\theta = 1$, a per-neuron learnable leak initialized at $\beta^l = 0.9$ and clipped to $[0, 0.99]$ (Table 8) and the reset value U^l_{reset}, which is 0 under the *naive*, *default* and *enhanced* configurations and that of Equation 36 under $U^l_t \sim \mathcal{N}(0,1)$ and $U^l_t \sim \mathcal{N}(\theta, 1)$. Their weights and bias implicitly subsume the factor $1 - \beta^l$ of the conventional low-pass form.

Pre-LIF normalization. The encoder block of every empirical network is a linear projection of the observations X^{enc}_t to the hidden dimension followed by a single batch-normalization module, which forms the input current I^{enc}_t of the first LIF layer (I^1_t in Equation 1); no further normalization appears within the SNN proper. This block replaces the Heaviside encoder with which the simulation study (Section C.1) isolates signal propagation under controlled input statistics. In its canonical form, the module normalizes each feature of the projection and then scales and shifts it,

$$\begin{aligned} I^{\text{enc}}_t &= \mathrm{BN}\left(\mathbf{W}^{\text{enc}}_X X^{\text{enc}}_t + \mathbf{b}^{\text{enc}}_X\right) \\ &= \mathbf{W}_{\mathrm{BN}}\,\mathrm{diag}(\boldsymbol{\sigma}_{\mathrm{BN}})^{-1}\left(\mathbf{W}^{\text{enc}}_X X^{\text{enc}}_t + \mathbf{b}^{\text{enc}}_X - \boldsymbol{\mu}_{\mathrm{BN}}\right) + \mathbf{b}_{\mathrm{BN}}, \end{aligned} \tag{62}$$

where the per-feature mean $\boldsymbol{\mu}_{\mathrm{BN}}$ and standard deviation $\boldsymbol{\sigma}_{\mathrm{BN}}$ of the projection are taken over the valid time steps of the batch in each optimization step and fixed to their running estimates at inference and during reinforcement-learning rollouts, and the diagonal scale $\mathbf{W}_{\mathrm{BN}}$ and the shift $\mathbf{b}_{\mathrm{BN}}$ are the module's learnable affine parameters. Across all empirical experiments, these affine parameters are initialized in one of two ways depending on the initialization configuration evaluated: (i) the standard convention with location parameter 0 and scale parameter 1, $\mathbf{W}_{\mathrm{BN}} = \mathbf{I}$ and $\mathbf{b}_{\mathrm{BN}} = \mathbf{0}$, for configurations (1-3), or (ii) the values prescribed by our initialization method, chosen such that the input current to the first LIF layer matches the target mean and variance for configurations (4-5). Under (ii), $\mathbf{W}_{\mathrm{BN}} = \sigma_{I^{\text{enc}}}\mathbf{I}$ and $\mathbf{b}_{\mathrm{BN}} = \mu_{I^{\text{enc}}}\mathbf{1}$ hold the target standard deviation and mean of the input current, the moments of the target distribution $I^{\text{enc}}_t \sim \mathcal{N}(\mu_{I^{\text{enc}}}, \sigma^2_{I^{\text{enc}}})$ of the closure (Equation 37),

$$\mu_{I^{\text{enc}}} = \mu_{U^1} - \mu_{\tilde{U}^1_{t-1}}, \qquad \sigma^2_{I^{\text{enc}}} = \sigma^2_{U^1} - \sigma^2_{\tilde{U}^1_{t-1}}, \tag{63}$$

with the target moments $\mu_{U^1}, \sigma^2_{U^1}$ of the first LIF layer and its recurrence moments $\mu_{\tilde{U}^1_{t-1}}, \sigma^2_{\tilde{U}^1_{t-1}}$ of Equation 5.

At inference, with $\boldsymbol{\mu}_{\mathrm{BN}}$ and $\boldsymbol{\sigma}_{\mathrm{BN}}$ fixed, Equation 62 regroups into a single linear projection of the form of I_t^l in Equation 1,

$$\begin{aligned} I_t^{\mathrm{enc}} = &\big(\mathbf{W}_{\mathrm{BN}}\,\mathrm{diag}(\boldsymbol{\sigma}_{\mathrm{BN}})^{-1}\mathbf{W}_X^{\mathrm{enc}}\big)X_t^{\mathrm{enc}} \\ &+ \mathbf{W}_{\mathrm{BN}}\,\mathrm{diag}(\boldsymbol{\sigma}_{\mathrm{BN}})^{-1}\big(\mathbf{b}_X^{\mathrm{enc}} - \boldsymbol{\mu}_{\mathrm{BN}}\big) + \mathbf{b}_{\mathrm{BN}}, \end{aligned} \tag{64}$$

and the deployed network contains no normalization. Without the module, the projection takes this form directly, with $\mu_{I^{\mathrm{enc}}}$ placed in the bias $\mathbf{b}_X^{\mathrm{enc}}$ and $\sigma^2_{I^{\mathrm{enc}}}$ in the weights $\mathbf{W}_X^{\mathrm{enc}}$, as in Equation 7 at $\mu_{\mathbf{W}_X} = 0$ and with the variance of the observations in place of $p^{l-1}(1-p^{l-1})$. Since the module only standardizes the mean and variance of the input current, its removal is nested in the sensitivity analysis of Section C.3.3, which violates the input assumption (A5) itself through an offset, a per-neuron dispersion and a correlation of the input spikes, and under which our initialization stays close to the target statistics in depth (Table 15). We nevertheless recommend the module: placed once before the first LIF layer, it is non-invasive and handles multiple data modalities, as demonstrated for the continuous, binned-event and level-crossing inputs of Section 3.2.

Output prediction across paradigms. The output stage of the SNN differs between the supervised and reinforcement-learning paradigms. For classification (UEA, neuromorphic), one additional LIF neuron per output class follows the final hidden LIF layer with the same neuron parameters $\{\theta, \beta, U_{\text{reset}}\}$ as the rest of the network; the normalized spike count of each output neuron across the time dimension yields the per-class score, which is optimized under a cross-entropy loss and used to determine the predicted class. SHD and DVS Gesture use this neuron-per-class readout, while ECG uses a linear readout matrix from the final hidden layer to the class dimension, as it requires a per-time-step prediction with a potentially time-varying target. For reinforcement learning, the action is drawn from a Gaussian distribution whose mean is the projected output of the final hidden LIF layer (via a linear projection to the action dimension) and whose standard deviation is a separately learned vector initialized at one. All reinforcement-learning tasks are continuous-control, and each predicted action is applied in closed loop to determine the environment's next state and the next observation.

Critic architecture (reinforcement learning). Only the actor is realized as a spiking neural network. The critic uses a conventional (non-recurrent) ANN with three hidden layers, each followed by a ReLU activation, and a final linear projection to a scalar value prediction; the hidden width of the critic matches the SNN width of the current architectural sweep cell. The critic is optimized with a separate Adam optimizer using the same settings as the actor (Table 8) at a constant learning rate, and consumes the same observations as the actor with no privileged additional state information.

Shared optimization hyperparameters. The settings in Table 8 are held constant across every paradigm and task in the empirical evaluation. The only per-paradigm distinctions are the optimizer learning rate with its schedule and gradient clipping, for which the supervised paradigms (UEA, neuromorphic) use the standard default and the reinforcement-learning paradigm uses the PPO convention; see footnote a. In reinforcement learning, gradients are clipped at global norm 0.5. Backpropagation treats the previous spike S_{t-1}^l of Equation 1 as a constant, known as a detached reset. The actor's learning rate decays linearly to 0 over training to prevent catastrophic forgetting and the associated performance degradation.

Per-task specifications. Table 9 lists the dataset/environment-level settings per task, organized by paradigm in the order used in Section 3.2: UEA multivariate time-series classification, neuromorphic classification, and continuous-control reinforcement learning. For the supervised paradigms, sample counts are reported as total (train/test); the reinforcement-learning paradigm uses a different convention, noted at the foot of the table. Of the thirteen UEA datasets, eight are reported in the main text; the other five, on which the configurations are indiscriminative or the accuracy stays near chance level, are reported in Section C.4.1. All datasets are used in their original released form. The required binning and encoding procedures applied to the neuromorphic datasets are described below.

UEA datasets. The UEA datasets are sourced from the UEA Multivariate Time Series Classification Archive [Bagnall et al., 2018] via the Python library aeon [Middlehurst et al., 2024]. The archive includes 30 diverse benchmarks; a complete list and specification details of the 13 datasets

Table 8: **Shared optimization hyperparameters.** SNN-side and Adam-optimizer settings used throughout the empirical evaluation in Section 3.2. Values are held constant across all tasks and paradigms; no per-paradigm or per-task tuning is performed. The only exception is the linear learning-rate decay of the RL actor, which prevents catastrophic forgetting.

Hyperparameter	**Value**
SNN-side	
LIF threshold θ	1
LIF leak factor β^l (per-neuron, learnable; init. value)	0.9
Surrogate-gradient family	fast sigmoid
Surrogate-gradient slope k	25
Adam optimizer	
Learning rate η[a]	10^{-3}
Exponential decay rate, first moment β_1	0.9
Exponential decay rate, second moment β_2	0.999
Numerical-stability term ϵ	10^{-8}
Weight decay	0 (off)

[a] $\eta = 10^{-3}$ for the supervised paradigms (UEA, neuromorphic). For the reinforcement-learning paradigm, $\eta = 3 \times 10^{-4}$ following the PPO convention of Schulman et al. [2017].

considered in this research are provided in Table 9. No further preprocessing is applied to these datasets.

Neuromorphic datasets. For each neuromorphic dataset, we provide details on the dataset-specific binning and encoding configurations, which directly determine the structure and dimensionality of each input sample. An overview of dimensionality for these tasks is provided in Table 9. Besides the specifics outlined below, no further preprocessing is applied across any of the three datasets. This ensures that all reported results reflect performance under the native data distributions induced by their respective encoding pipelines.

The Spiking Heidelberg Digits (SHD) dataset [Cramer et al., 2022] is binned into windows of 5 ms from the start of each recording, and the incomplete last window is dropped. This procedure converts the event-based cochlear spike streams into binary frames of 700 channels per time step, with an entry of 1 where a channel spikes within the window. The resulting sequence length remains variable across samples; the empirical range $[T_{\min}, T_{\max}]$, computed over both training and test splits, is reported in Table 9.

The DVS Gesture dataset [Amir et al., 2017] is first spatially downsampled by a factor of two to reduce input resolution and computational cost, and subsequently discretized into temporal frames using the `ToFrame` transform [Lenz et al., 2021] with a bin size of 50 ms. This preprocessing yields a $64 \times 64 \times 2$ spatiotemporal grid per time step, where both polarity channels are retained. Before downsampling, events are denoised with the `Denoise` transform of `tonic` at a filter time of 10 ms. The resulting frames are flattened into vectors of dimensionality 8192, and, as with SHD, sequence lengths vary across samples; the empirical range is reported in Table 9.

The electrocardiography (ECG) dataset [Laguna et al., 1997] is used in its preprocessed form as provided by Yin et al. [2021]. This preprocessed version comprises a level-crossing encoded representation upstream, consisting of two leads with two polarities each, yielding a total of 4 input channels. The data is sampled at a temporal resolution of 1 ms; each sequence is valid up to its last labeled time step, with the range of valid lengths reported in Table 9.

Reinforcement-learning environments. Reinforcement learning is conducted on three continuous control environments—*inverted pendulum*, *inverted double pendulum*, and *hopper*—optimized with the PPO algorithm [Schulman et al., 2017], with its hyperparameters and an overview of task dimensionalities provided in Table 9. All three environments are sourced from the Brax ecosys-

Table 9: **Per-task specifications.** Datasets/environments, sample budgets, the dimensions, and per-task training schedule for the empirical evaluation. The dimensions, denoted $\{I, T, O\}$ in the table column, comprise the input/observation dimension I, the time steps T (sequence length for the supervised paradigms; for the reinforcement-learning paradigm, the training rollout/truncation horizon and the maximum inference-mode episode length, separated by a slash), and the output dimension O (number of classes for classification; action-space dimension for control). The architectural sweep (width $\in \{32, 64, 128\}$, depth $\in \{2, 3, 4, 5\}$, 20 seeds) is shared across all tasks. N_{updates} counts training epochs for the supervised paradigms and PPO update stages for the reinforcement-learning paradigm (each update stage comprising four PPO epochs); performance is evaluated every five N_{updates} across all paradigms.

Task / dataset	**Samples** (total (train/test))	$\{I, T, O\}$	$N_{\textbf{batches}}$ (per epoch)	$N_{\textbf{updates}}$
UEA multivariate time-series classification (13 datasets)				
ArticularyWordRecognition	575 (275/300)	{9, 144, 25}	8	250
AtrialFibrillation	30 (15/15)	{2, 640, 3}	1	250
BasicMotions	80 (40/40)	{6, 100, 4}	2	250
Cricket	180 (108/72)	{6, 1197, 12}	4	250
Epilepsy	275 (137/138)	{3, 206, 4}	4	250
FingerMovements	416 (316/100)	{28, 50, 2}	8	250
Handwriting	1000 (150/850)	{3, 152, 26}	4	250
Heartbeat	409 (204/205)	{61, 405, 2}	4	250
LSST	4925 (2459/2466)	{6, 36, 14}	32	250
NATOPS	360 (180/180)	{24, 51, 6}	4	250
RacketSports	303 (151/152)	{6, 30, 4}	4	250
SelfRegulationSCP1	561 (268/293)	{6, 896, 2}	8	250
UWaveGestureLibrary	440 (120/320)	{3, 315, 8}	4	250
Neuromorphic classification				
SHD	10420 (8156/2264)	{700, [46, 273], 20}	16	250
DVS Gesture	1341 (1077/264)	{8192, [34, 369], 11}	9	250
ECG	759 (618/141)	{4, [182, 1300], 6}	5	250
Continuous-control reinforcement learning (PPO; see RL conventions note below)				
Inverted Pendulum	–(512/512)	{4, 20/1K, 1}	16	300
Inverted Double Pendulum	–(512/512)	{8, 20/1K, 1}	16	800
Hopper	–(512/512)	{11, 20/1K, 3}	16	1000

Reinforcement-learning column conventions. PPO [Schulman et al., 2017] is run with clip ratio 0.2, GAE-$\lambda = 0.95$, value-loss coefficient 0.5 and entropy coefficient 0.01; the discount factor is $\gamma = 0.99$ for every RL task. The RL rows reinterpret two cell formats relative to the supervised paradigms. *Samples*: the leading dash indicates that there is no bounded total sample count for an on-policy method; the parenthetical reports the number of training environments and test environments stepped in parallel, written as $N_{\text{train envs}}/N_{\text{test envs}}$. *Middle entry of* $\{I, T, O\}$: the value before the slash is the training rollout/truncation horizon (used for truncated backpropagation through time, 20 in every RL task); the value after the slash is the maximum inference-mode episode length (1000 in every RL task). Episodes may terminate early when the environment's termination condition is hit. The first and third entries of $\{I, T, O\}$ are the observation (state) dimension and the action-space dimension, respectively; all three environments use continuous, real-valued action spaces. All three are Brax environments.

tem [Freeman et al., 2021]. The first two each feature a single-dimensional continuous action space that controls a cart to stabilize the pendulum(s) based on per-step alive rewards, from which the double pendulum subtracts tip-position and velocity penalties. The hopper environment poses a planar locomotion task for a one-legged robot with a three-dimensional continuous action space. It rewards forward velocity and every step in which the robot stays upright. An episode ends when the robot falls. No further adjustments to reward definitions and/or (pre-)processing of observations, rewards, or action spaces are applied.

Implementation and compute. The empirical evaluation is implemented in JAX with Flax and Optax and runs on one NVIDIA A100 80GB GPU partitioned into MIG instances, each training one (task, configuration, width, depth) cell at a time in sets of five seeds. Four such sets give the 20 seeds per cell. The UEA, neuromorphic and RL sweeps take approximately 62, 74 and 126 instance-hours (about 260 in total), or roughly 19, 26 and 35 hours of wall-clock time on three to five instances in parallel.

Performance metrics. Each run is evaluated every five N_{updates} of Table 9, and its final performance is the mean over the last 10% of these evaluation points. Classification reports the test accuracy. Reinforcement learning reports the evaluation reward normalized by a per-environment ceiling, 1000 for Inverted Pendulum, 10 000 for Inverted Double Pendulum and 1000 for Hopper, on the same 0–100% scale as the accuracies. Hopper has no maximum return, so its ceiling is the alive reward of 1 per step over the 1000-step episode. Let $y_{jca}^{(i)}$ denote the final performance of seed $i \in \{1, \dots, R\}$ for task j, configuration c and cell a, one (width, depth) pair of the grid. The cell tables of Section C.4 report the mean and the sample standard deviation over the $R = 20$ seeds,

$$\bar{y}_{jca} = \frac{1}{R} \sum_{i=1}^{R} y_{jca}^{(i)}, \qquad s_{jca}^2 = \frac{1}{R-1} \sum_{i=1}^{R} \left(y_{jca}^{(i)} - \bar{y}_{jca} \right)^2. \tag{65}$$

We write $\mathcal{A}_j$ for the 12 cells of task j. Within each cell, $\mathrm{rank}(\bar{y}_{jca})$ orders the five configurations by their mean, from 1 for the highest. Tied means share the best rank. Each task row of Table 3 averages over these cells,

$$\bar{y}_{jc} = \frac{1}{|\mathcal{A}_j|} \sum_{a \in \mathcal{A}_j} \bar{y}_{jca}, \qquad \overline{\mathrm{rank}}_{jc} = \frac{1}{|\mathcal{A}_j|} \sum_{a \in \mathcal{A}_j} \mathrm{rank}(\bar{y}_{jca}), \tag{66}$$

and reports the mean of s_{jca} over the same cells after the $\pm$. The paradigm mean rank averages $\mathrm{rank}(\bar{y}_{jca})$ over the cells of all tasks in a paradigm. The overall mean rank averages the three paradigm mean ranks. Figure 3 averages the cell means $\bar{y}_{jca}$ (top) and their ranks (bottom) over the cells of each paradigm. Per width and per depth it weighs the three paradigms equally.

Significance testing. We test whether configuration (5) outperforms each alternative $c \in \{1, \dots, 4\}$ on the same tasks and cells. In each cell a of task j, the difference is the rank gain $d_{ja} = \mathrm{rank}(\bar{y}_{jca}) - \mathrm{rank}(\bar{y}_{j5a})$. The cells of one task share its data and are not independent. The test therefore uses one value per task, the mean difference $\bar{d}_j$ over $\mathcal{A}_j$. Their weighted mean

$$\bar{d} = \sum_j w_j \, \bar{d}_j, \qquad w_j = \frac{|\mathcal{A}_j|}{3 \, n_{g(j)}}, \tag{67}$$

weighs the three paradigms equally, as the overall mean rank does. Here $n_{g(j)}$ counts the cells of the paradigm of task j. Without an effect of configuration (5), each $\bar{d}_j$ is equally likely to be positive or negative. A paired permutation test (sign-flip test) therefore takes the one-sided p-value as the fraction of the 2^{14} sign assignments $\zeta_j \in \{-1, 1\}$ of the 14 tasks that reach the observed mean,

$$p = 2^{-14} \left| \left\{ \zeta : \sum_j w_j \, \zeta_j \, \bar{d}_j \geq \bar{d} \right\} \right|. \tag{68}$$

The test is one-sided because configuration (5) is the pre-specified default. Table 3 reports the resulting p-values together with those of the same test on the score differences $\bar{y}_{j5a} - \bar{y}_{jca}$. All tests use the cell means. The seed standard deviations enter only the confidence intervals of the next paragraph.

Confidence intervals. We quantify the uncertainty due to the training seed with 95% confidence intervals from a parametric variant of the stratified bootstrap [Agarwal et al., 2021]. The cells form a fixed grid, so the seed is the only random element of the experiment. The bootstrap therefore stratifies by cell. Each of 2000 draws replaces every cell mean $\bar{y}_{jca}$ by

$$\tilde{y}_{jca} = \bar{y}_{jca} + z_{jca} \frac{s_{jca}}{\sqrt{R}}, \qquad z_{jca} \sim \mathcal{N}(0,1), \tag{69}$$

drawn independently for every task, configuration and cell. Here s_{jca} is the seed standard deviation of the cell. This is the normal approximation of resampling the twenty seeds of a cell, which assumes approximately normal cell means. Every draw recomputes the aggregated scores and ranks from $\tilde{y}_{jca}$. The interval spans the 2.5th to 97.5th percentile of the draws. Ranks are bounded by 1 and 5, so a rank interval is not centered on its estimate. A configuration that ranks first in most cells can only lose ranks under seed noise, and one that ranks last can only gain. The interval of a leading configuration therefore lies mostly above its estimate. These intervals quantify seed noise only. Inference across tasks rests on the sign-flip test above. The intervals are reported for the summary rows of Table 3 and for the bars of Figure 3.

C.3 Simulation study: extended results

This subsection extends the simulation results of Section 3.1. Section C.3.1 compares all initialization methods on the reference network across target rates and depth, in support of the main-text comparison. The initialization configurations of the empirical evaluation follow in Section C.3.2, at their own operating points and in the widths of the empirical networks. The remaining subsections report our initialization alone: under violations of its input assumption (Section C.3.3), and for the extensions of Section B.4 to other LIF neuron models (Section C.3.4), heterogeneous populations (Section C.3.5), residual and recurrent connections (Section C.3.6) and convolutional layers (Section C.3.7), each pointing to the section that derives its case.

C.3.1 Comparison of initialization methods for reference LIF model

Setup. Each study feeds binarized Gaussian noise through a deep spiking network at initialization and records every neuron's spike rate and membrane-potential variance (Section C.1). We use the configuration of Table 7 across its rate grid, with all ten initialization methods. The encoder fires at the same target rate p (Equation 59). Our initialization targets each rate through Equation 25 and Equation 26 (Section B.1.3); the alternative methods are implemented as described in Appendix A.

Results. Across the rate grid, no alternative holds the target per neuron at any reported layer (Table 10). Four fall silent by layer 10 at every rate, and Lee et al. [2016] falls to at most 0.1% by layer 30. Ding et al. [2025] and Che et al. [2025] settle at layer 30 on 1.9–2.0% and 35.0% regardless of the target. Che et al. [2025] retains this activity with a spread of 38.0 pp across neurons and a membrane variance of 25.7 at $p = 50\%$. Rossbroich et al. [2022] reaches layer 30 at about a quarter of the target (3.6–21.3%) with a membrane variance of 0.53–0.80. At the $p = 15\%$ operating point these findings are in line with a similar comparison by Micheli et al. [2025]. Their own initialization remains active at this rate but drifts above target, to 20.4% at layer 30 with a spread of 24.4 pp across neurons. It falls to 0.3–3.0% at $p = 35$–65% and lies 64.6 pp below target at $p = 85\%$. At every target rate and reported layer our initialization holds the spike rate within 1.6 pp of the target, with a spread across neurons of 0.3–0.5 pp, the resolution of the estimator, and a membrane variance of 0.91–1.06 (Table 11). For our initialization, the offset of a layer's mean spike rate from its target, $\bar{p}^l - p$, lies between $+0.1$ and $+0.4$ pp at layer 1 for every target rate and reaches -0.6, $+1.6$, $+1.3$, $+1.4$ and -1.2 pp at layer 30 for $p = 15$, 35, 50, 65 and 85%, respectively. The deepest layer thus fires above its target at $p = 35$–65% and below it at $p = 15\%$ and 85%. After 30 layers, the deviation thus stays limited to at most 1.6 pp, with a spread across neurons of at most 0.5 pp, a combination that no alternative attains at any target rate.

Table 10: **Alternative initializations across target rates and depth.** Layers 1, 5, 10, 20 and 30 at initialization, for the target rates $p = 15$–85%; mean over 10 seeds. Each cell gives the spike rate $\bar{p}^l$ in % (target p) followed, in parentheses, by the membrane-potential variance $\bar{\sigma}^2_{U^l}$ (target $\sigma^2_{U^l} = 1$), each as mean $\pm$ standard deviation across the neurons of the reported layer l. Our initialization: Table 11.

	$p = 15\%$	$p = 35\%$	$p = 50\%$	$p = 65\%$	$p = 85\%$
Micheli et al. [2025][1]					
Layer 1	12.7±12.7 (3.07±1.43)	17.9±20.5 (3.13±1.73)	22.0±26.0 (3.15±1.86)	26.7±31.6 (3.19±1.99)	35.1±39.9 (3.32±2.29)
Layer 5	16.5±20.3 (3.18±2.15)	11.1±14.6 (1.00±0.52)	11.1±14.8 (0.82±0.39)	13.4±17.8 (0.96±0.47)	21.9±28.0 (2.17±1.34)
Layer 10	19.2±24.2 (2.75±1.83)	8.9±11.7 (0.69±0.30)	7.6±10.1 (0.54±0.21)	9.8±13.0 (0.71±0.30)	20.2±25.7 (2.34±1.45)
Layer 20	19.8±24.3 (2.60±1.68)	5.4±6.7 (0.53±0.24)	3.1±3.8 (0.33±0.14)	6.0±7.5 (0.55±0.24)	19.9±24.5 (2.55±1.64)
Layer 30	20.4±24.4 (2.72±1.82)	2.5±2.9 (0.33±0.21)	0.3±0.4 (0.06±0.03)	3.0±3.5 (0.37±0.22)	20.4±24.4 (2.72±1.80)
He et al. [2015]					
Layer 1	7.2±7.8 (0.86±0.35)	13.9±16.6 (1.47±0.77)	18.1±22.3 (1.62±0.92)	22.2±27.6 (1.49±0.89)	27.8±34.5 (0.89±0.55)
Layer 5	0.4±0.7 (0.08±0.02)	2.1±3.4 (0.16±0.04)	3.0±4.7 (0.18±0.04)	3.8±5.8 (0.19±0.04)	4.7±7.1 (0.20±0.04)
Layer 10	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.01±0.00)
Layer 20	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)
Layer 30	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)
Glorot and Bengio [2010]					
Layer 1	4.8±6.0 (0.47±0.16)	10.2±13.6 (0.79±0.36)	13.7±18.6 (0.86±0.42)	17.0±23.4 (0.80±0.41)	21.4±29.4 (0.48±0.25)
Layer 5	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.01±0.00)	0.0±0.1 (0.01±0.00)	0.1±0.2 (0.02±0.01)	0.2±0.6 (0.03±0.01)
Layer 10	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)
Layer 20	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)
Layer 30	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)
Lee et al. [2016]					
Layer 1	9.4±10.5 (0.42±0.17)	16.5±20.2 (0.73±0.38)	20.7±25.8 (0.81±0.45)	24.6±30.6 (0.75±0.44)	29.8±36.9 (0.44±0.28)
Layer 5	6.2±8.1 (0.15±0.06)	9.2±12.3 (0.18±0.07)	10.5±14.1 (0.18±0.07)	11.4±15.4 (0.18±0.07)	12.5±16.9 (0.16±0.06)
Layer 10	3.8±4.8 (0.10±0.03)	5.7±7.4 (0.12±0.04)	6.3±8.2 (0.12±0.04)	6.7±8.8 (0.13±0.04)	7.1±9.3 (0.13±0.04)
Layer 20	0.2±0.2 (0.01±0.00)	0.9±1.1 (0.04±0.02)	1.2±1.5 (0.05±0.02)	1.5±1.8 (0.05±0.02)	1.7±2.2 (0.06±0.02)
Layer 30	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)	0.1±0.1 (0.00±0.00)
Zenke and Vogels [2020]					
Layer 1	2.0±3.2 (0.19±0.04)	5.7±8.5 (0.30±0.10)	8.2±12.0 (0.32±0.12)	10.5±15.5 (0.30±0.11)	13.7±20.5 (0.19±0.06)
Layer 5	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)
Layer 10	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)
Layer 20	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)
Layer 30	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)
Bellec et al. [2018]					
Layer 1	0.0±0.0 (0.01±0.00)	0.0±0.1 (0.01±0.00)	0.2±0.7 (0.02±0.01)	0.5±1.5 (0.02±0.02)	1.0±2.6 (0.02±0.02)
Layer 5	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)
Layer 10	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)
Layer 20	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)
Layer 30	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)
Che et al. [2025]					
Layer 1	18.4±17.8 (15.5±7.1)	25.1±27.3 (28.3±15.5)	28.9±32.1 (31.6±18.4)	32.5±36.3 (29.1±17.8)	37.5±41.7 (17.2±11.4)
Layer 5	31.9±35.7 (32.3±25.9)	33.1±37.3 (26.1±21.2)	33.7±38.0 (22.9±18.7)	34.3±38.5 (20.0±16.3)	35.2±39.3 (16.0±13.0)
Layer 10	34.3±37.3 (30.8±27.5)	34.3±37.6 (28.5±25.3)	34.3±37.8 (27.2±23.9)	34.3±37.9 (26.2±22.9)	34.1±38.0 (24.7±21.1)
Layer 20	38.0±38.5 (27.6±25.3)	38.0±38.6 (27.6±25.2)	38.1±38.6 (27.6±25.5)	38.1±38.6 (27.5±25.3)	38.1±38.6 (27.3±25.1)
Layer 30	35.0±38.0 (25.7±22.9)	35.0±38.0 (25.6±22.9)	35.0±38.0 (25.7±23.0)	35.0±38.0 (25.8±23.0)	35.0±38.0 (25.9±23.3)
Ding et al. [2025]					
Layer 1	9.0±8.8 (0.79±0.34)	15.5±17.5 (1.41±0.77)	19.6±23.1 (1.57±0.91)	23.6±28.3 (1.45±0.88)	29.0±35.1 (0.87±0.55)
Layer 5	4.8±5.6 (0.22±0.05)	6.2±7.6 (0.24±0.06)	6.8±8.6 (0.25±0.06)	7.4±9.4 (0.25±0.06)	8.1±10.5 (0.25±0.05)
Layer 10	3.6±4.0 (0.17±0.03)	4.1±4.7 (0.18±0.03)	4.3±5.0 (0.18±0.03)	4.4±5.2 (0.18±0.03)	4.6±5.4 (0.18±0.03)
Layer 20	2.4±2.4 (0.13±0.03)	2.6±2.6 (0.14±0.03)	2.7±2.7 (0.14±0.03)	2.7±2.7 (0.14±0.03)	2.8±2.8 (0.14±0.03)
Layer 30	1.9±1.7 (0.11±0.04)	1.9±1.7 (0.11±0.03)	2.0±1.8 (0.12±0.04)	2.0±1.8 (0.12±0.04)	2.0±1.8 (0.12±0.04)
Rossbroich et al. [2022]					
Layer 1	5.4±6.3 (0.57±0.22)	10.0±11.4 (0.41±0.18)	12.9±14.4 (0.32±0.14)	15.8±17.2 (0.24±0.09)	20.2±20.8 (0.13±0.02)
Layer 5	4.1±5.3 (0.48±0.20)	7.6±8.6 (0.32±0.08)	9.8±10.3 (0.28±0.05)	12.0±11.7 (0.25±0.05)	16.0±13.6 (0.24±0.05)
Layer 10	4.2±5.3 (0.54±0.28)	8.2±9.0 (0.37±0.12)	10.7±10.8 (0.33±0.09)	13.3±12.2 (0.30±0.08)	17.7±14.0 (0.29±0.09)
Layer 20	3.8±4.3 (0.68±0.50)	8.9±9.1 (0.50±0.31)	12.1±11.3 (0.44±0.24)	15.1±12.8 (0.41±0.20)	19.6±14.1 (0.40±0.21)
Layer 30	3.6±4.0 (0.80±0.77)	9.6±9.3 (0.68±0.75)	13.3±11.7 (0.61±0.65)	16.6±13.1 (0.55±0.52)	21.3±14.1 (0.53±0.39)

Table 11: **Our initialization across target rates and depth.** Layers 1, 5, 10, 20 and 30 at initialization, for the target rates $p = 15$–85%, our initialization only; mean over 10 seeds. Each cell gives the spike rate $\bar{p}^l$ in % (target p) followed, in parentheses, by the membrane-potential variance $\bar{\sigma}^2_{U^l}$ (target $\sigma^2_{U^l} = 1$), each as mean $\pm$ standard deviation across the neurons of the reported layer l. Layers 1, 5 and 30 against selected alternatives: Table 10.

	$p = 15\%$	$p = 35\%$	$p = 50\%$	$p = 65\%$	$p = 85\%$
Ours					
Layer 1	15.3±0.4 (1.06±0.03)	35.4±0.5 (1.02±0.02)	50.2±0.5 (1.01±0.02)	65.1±0.4 (1.00±0.02)	85.1±0.3 (1.00±0.01)
Layer 5	15.2±0.4 (0.96±0.03)	36.1±0.5 (0.99±0.02)	50.7±0.5 (0.98±0.02)	65.3±0.5 (0.99±0.02)	85.9±0.4 (0.94±0.02)
Layer 10	15.0±0.4 (0.94±0.03)	36.4±0.5 (0.98±0.02)	51.1±0.5 (0.97±0.02)	65.6±0.4 (0.99±0.02)	85.6±0.4 (0.93±0.02)
Layer 20	14.7±0.4 (0.92±0.03)	36.5±0.5 (0.98±0.02)	51.3±0.5 (0.97±0.02)	66.1±0.4 (0.98±0.02)	84.5±0.3 (0.99±0.02)
Layer 30	14.4±0.4 (0.91±0.03)	36.6±0.5 (0.98±0.02)	51.3±0.5 (0.97±0.02)	66.4±0.4 (0.98±0.02)	83.8±0.4 (1.03±0.02)

Pooled-variance and per-neuron forms of Micheli et al. [2025]. Table 12 reports the pooled-variance form $\mathrm{Var}[w] = 1/(D_{\text{in}}\, p)$ and Table 10 the per-neuron form $\mathrm{Var}[w] = 1/(D_{\text{in}}\, p(1-p))$ (Section A.1). The per-neuron weights are the pooled-variance weights scaled by $1/\sqrt{1-p}$. The two forms differ in their mean spike rate. At $p = 15\%$ layer 30 fires at 16.6% under the pooled-variance form and at 20.4% under the per-neuron form. From $p = 35\%$ upward the pooled-variance form falls silent by layer 20. The per-neuron form remains active there but far below target, at 0.3–20.4% in layer 30. Neither form holds the targets across depth, in line with the recurrence and reset that both omit (Section A.1). In both, the spike rate and the membrane variance drift with depth at every target rate. Wherever a layer is active, the spread of the spike rate across neurons equals or exceeds its mean. Each layer thus mixes dead with highly active or saturated neurons. Neither form attains unit within-neuron variance. At $p = 15\%$ the mean per-neuron variance lies between 1.9 and 3.2 in both forms. At $p = 35$–65% it decays toward zero with depth and crosses one only in passing. At $p = 85\%$ it stays between 2.2 and 3.3 under the per-neuron form, whereas the pooled-variance form starts at 0.55 and decays to zero. The pooled variance adds the between-neuron variance of the conditional means (Equation 61) and can therefore only exceed these values. By construction the two forms differ only in whether the pooled or the within-neuron input variance is normalized, so we read the weaker upward drift in the mean spike rate of the pooled-variance form at $p = 15\%$ and its silence from $p = 35\%$ upward as two effects of its smaller weight scale. In both forms this scale leaves the between-neuron share p of the input variance unchanged and remains one parameter for the target rate and unit variance. These targets are jointly attainable only near $1 - \Phi(\theta) \approx 15.9\%$ under a standard-normal membrane, which both forms assume without modeling the recurrence and reset that shape it. In this network neither form sustains its respective variance target alone or jointly with the target spike rate at the $p = 15\%$ operating point nor at the other rates, and we hypothesize that this stems from these omitted dynamics. Because neither form attains the target spike rate and unit variance per neuron jointly and consistently on the rate grid, we do not deem the choice between them to change the qualitative outcome. We therefore carry forward the per-neuron form, which remains active at every target rate and shares the variance definition of our initialization. It defines configuration (2) and (3) of the empirical evaluation in Section 3.2 (setup in Section C.2).

Table 12: **Pooled-variance form of Micheli et al. [2025] across target rates and depth.** Weight variance $1/(D_{\text{in}}\, p)$, against $1/(D_{\text{in}}\, p(1-p))$ in the per-neuron form of Table 10. Layers 1, 5, 10, 20 and 30 at initialization, for the target rates $p = 15$–85%; mean over 10 seeds. Each cell gives the spike rate $\bar{p}^l$ in % (target p) followed, in parentheses, by the membrane-potential variance $\bar{\sigma}^2_{U^l}$ (target $\sigma^2_{U^l} = 1$), each as mean $\pm$ standard deviation across the neurons of the reported layer l.

	$p = 15\%$	$p = 35\%$	$p = 50\%$	$p = 65\%$	$p = 85\%$
Micheli et al. [2025], pooled-variance form					
Layer 1	12.0±12.1 (2.63±1.22)	15.7±18.5 (2.07±1.11)	18.1±22.3 (1.62±0.91)	20.5±25.9 (1.16±0.67)	23.8±30.8 (0.55±0.31)
Layer 5	14.0±17.5 (2.26±1.44)	5.6±7.8 (0.38±0.13)	3.0±4.7 (0.18±0.04)	1.6±2.9 (0.11±0.03)	0.6±1.5 (0.06±0.02)
Layer 10	16.1±20.3 (1.94±1.20)	1.1±1.8 (0.14±0.04)	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)
Layer 20	16.3±20.2 (1.86±1.11)	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)
Layer 30	16.6±20.0 (1.93±1.19)	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)	0.0±0.0 (0.00±0.00)

C.3.2 Comparison of the empirical initialization configurations

Setup. This study applies the protocol of Section C.3.1 to the five initialization configurations of the empirical evaluation (Figure 2). Each configuration is initialized as in the empirical networks, with further details in Section C.2. Two input settings follow. The *assumed input* drives the reference network of 30 layers of 100 neurons through the binary encoder of Equation 59. The encoder fires at the input rate each configuration assumes in the empirical networks: 15% for configurations (1)–(4) and 50% for configuration (5), as in the first row of Table 13. The *empirical input* follows the 12 architectural cells (width, depth) of Section 3.2. The networks take the widths 32, 64 and 128 of these cells, and their layers 1–5 cover the depths 2–5. Every configuration then receives the same Gaussian input through the encoder of the empirical networks, a dense layer followed by batch normalization with batch statistics. Both settings are controlled cases at initialization, meant to give insight into the signal propagation of each configuration rather than to predict its behavior on the empirical data.

Table 13: **Initialization configurations of the empirical evaluation, assumed input.** Layers 1, 5, 10, 20 and 30 at initialization of the reference network of 30 layers of 100 neurons; mean over 10 seeds. Each configuration receives the encoder rate p^{enc} of the first row. Each cell gives the spike rate $\bar{p}^l$ in % followed, in parentheses, by the membrane-potential variance $\bar{\sigma}^2_{U^l}$ (target $\sigma^2_{U^l} = 1$), each as mean $\pm$ standard deviation across the neurons of the reported layer l.

	naive	I_t default	$I_t^{\perp}$ enhanced	$U_t \sim \mathcal{N}(0,1)$	$U_t \sim \mathcal{N}(\theta,1)$
p^{enc}	15%	15%	15%	15%	50%
Layer 1	2.0±3.0 (0.19±0.04)	12.2±12.5 (3.12±1.44)	12.8±13.0 (3.06±1.34)	16.1±0.4 (1.06±0.03)	49.9±0.5 (1.01±0.02)
Layer 5	0.0±0.0 (0.00±0.00)	16.4±20.6 (3.19±2.05)	16.8±21.0 (3.18±1.93)	16.1±0.3 (0.94±0.02)	50.6±0.5 (1.01±0.02)
Layer 10	0.0±0.0 (0.00±0.00)	19.9±24.1 (2.69±1.88)	19.7±23.4 (2.75±1.78)	15.7±0.3 (0.92±0.02)	51.0±0.6 (1.01±0.02)
Layer 20	0.0±0.0 (0.00±0.00)	19.1±23.5 (2.75±1.86)	20.3±24.9 (2.61±1.64)	15.4±0.3 (0.90±0.02)	51.1±0.6 (1.01±0.02)
Layer 30	0.0±0.0 (0.00±0.00)	18.5±23.6 (2.55±1.65)	19.4±24.4 (2.66±1.62)	15.2±0.3 (0.89±0.02)	51.2±0.6 (1.01±0.02)

Table 14: **Initialization configurations of the empirical evaluation, empirical input.** Layers 1–5 at initialization of networks with the widths 32, 64 and 128 of the 12 architectural cells (width, depth) of the empirical evaluation; mean over 10 seeds. Every configuration receives the same Gaussian input through the encoder of the empirical networks. Each cell gives the spike rate $\bar{p}^l$ in % followed, in parentheses, by the membrane-potential variance $\bar{\sigma}^2_{U^l}$ (target $\sigma^2_{U^l} = 1$), each as mean $\pm$ standard deviation across the neurons of the reported layer l.

	naive	I_t default	$I_t^{\perp}$ enhanced	$U_t \sim \mathcal{N}(0,1)$	$U_t \sim \mathcal{N}(\theta,1)$
width 32					
Layer 1	8.4±0.2 (2.88±0.09)	8.4±0.2 (2.88±0.09)	8.4±0.2 (2.88±0.09)	16.2±0.2 (1.06±0.03)	49.9±0.4 (1.01±0.02)
Layer 2	0.9±1.2 (0.17±0.04)	9.0±7.6 (2.66±1.34)	9.2±7.5 (2.63±1.15)	16.2±0.3 (0.98±0.03)	50.1±0.4 (1.01±0.02)
Layer 3	0.0±0.0 (0.01±0.00)	9.7±10.2 (2.80±1.87)	10.4±10.1 (2.70±1.50)	15.9±0.3 (0.95±0.04)	50.2±0.5 (1.01±0.04)
Layer 4	0.0±0.0 (0.00±0.00)	10.6±11.9 (2.60±1.89)	11.3±12.1 (2.82±1.84)	15.6±0.3 (0.92±0.04)	50.4±0.6 (1.01±0.04)
Layer 5	0.0±0.0 (0.00±0.00)	9.7±11.3 (2.57±1.87)	11.3±13.2 (2.90±1.92)	15.3±0.3 (0.90±0.04)	50.5±0.6 (1.01±0.04)
width 64					
Layer 1	8.4±0.2 (2.88±0.09)	8.4±0.2 (2.88±0.09)	8.4±0.2 (2.88±0.09)	16.2±0.3 (1.06±0.03)	49.9±0.4 (1.01±0.02)
Layer 2	0.9±1.3 (0.17±0.03)	9.4±7.5 (2.63±1.23)	9.7±7.9 (2.61±1.09)	16.3±0.3 (0.99±0.03)	50.1±0.4 (1.01±0.02)
Layer 3	0.0±0.0 (0.01±0.00)	10.7±10.5 (2.86±1.65)	10.9±11.0 (2.89±1.53)	16.2±0.3 (0.96±0.03)	50.3±0.4 (1.01±0.02)
Layer 4	0.0±0.0 (0.00±0.00)	11.9±12.5 (2.86±1.85)	12.8±13.7 (2.87±1.73)	16.1±0.3 (0.95±0.03)	50.4±0.5 (1.01±0.03)
Layer 5	0.0±0.0 (0.00±0.00)	12.3±14.2 (3.17±2.19)	13.7±15.7 (3.07±1.92)	16.0±0.3 (0.94±0.03)	50.6±0.5 (1.01±0.03)
width 128					
Layer 1	8.4±0.3 (2.88±0.09)	8.4±0.3 (2.88±0.09)	8.4±0.3 (2.88±0.09)	16.2±0.3 (1.06±0.03)	50.0±0.4 (1.01±0.02)
Layer 2	0.9±1.3 (0.17±0.03)	9.4±7.7 (2.65±1.14)	9.8±8.1 (2.61±1.09)	16.3±0.3 (1.00±0.04)	50.2±0.5 (1.01±0.05)
Layer 3	0.0±0.0 (0.01±0.00)	11.2±11.3 (2.82±1.51)	10.9±11.0 (2.93±1.51)	16.3±0.3 (0.97±0.03)	50.4±0.6 (1.01±0.03)
Layer 4	0.0±0.0 (0.00±0.00)	12.7±13.9 (3.01±1.77)	12.8±13.9 (2.99±1.70)	16.3±0.3 (0.96±0.03)	50.6±0.6 (1.01±0.03)
Layer 5	0.0±0.0 (0.00±0.00)	14.4±16.3 (3.05±1.91)	14.4±16.4 (3.11±1.87)	16.3±0.3 (0.95±0.03)	50.7±0.6 (1.01±0.03)

Results. Under the assumed input the naive configuration fires at 2.0% in layer 1 and falls silent by layer 5 (Table 13). The default and enhanced configurations hold the assumed rate of 15% only as a layer mean, at 12.2–20.3% in every reported layer. Their spread across neurons of 12.5–24.9 pp equals or exceeds this mean. Their membrane variance lies at 2.55–3.19. Our initialization holds 15.2–16.1% at $U_t \sim \mathcal{N}(0,1)$ and 49.9–51.2% at $U_t \sim \mathcal{N}(\theta,1)$. Its spread across neurons stays at 0.3–0.6 pp and its membrane variance at 0.89–1.06. Under the empirical input the naive, default

and enhanced configurations leave the encoder at its defaults (Table 14). They therefore share the first layer at 8.4% with a membrane variance of 2.88 at every width. The naive configuration then falls silent by layer 3. In layers 2–5 the default and enhanced configurations fire at 9.0–14.4% with a spread of 7.5–16.4 pp and a membrane variance of 2.57–3.17. Our initialization holds 15.3–16.3% and 49.9–50.7% in layers 1–5 at every width, with a membrane variance of 0.90–1.06. In both settings the membrane variance of the default and enhanced configurations is about three times that of configuration (4).

C.3.3 Sensitivity analysis to the input assumption

The closure assumes that every layer receives i.i.d. Bernoulli spikes at the assumed rate ((A5), Section 2.3), as the initialization provides beyond the first layer. In the empirical study, a batch-normalization module before the first LIF layer supports this assumption where data enter, standardizing the projected input and, under our initialization, setting it to the target current moments (Section C.2). Removing the batch normalization module from the encoder block on one task would give a single outcome, confounded with that dataset's statistics, showing neither which property fails nor by how much. We therefore test sensitivity to the assumption input systematically by violating its three properties; the spike rate, the identical distribution and the independence property across neurons. This is conducted under controlled conditions with one violated property at a time, at graded magnitudes and on a known input distribution.

Setup. Each study feeds binarized Gaussian noise through a deep spiking network at initialization and records every neuron's spike rate and membrane-potential variance (Section C.1). We use the configuration of Table 7 at $p = 50\%$, our initialization only, with the encoder departing from the input assumption (A5) of Section 2.3 one property at a time. Under an *offset*, the encoder delivers i.i.d. Bernoulli spikes at a rate other than 50%; under *dispersion*, each encoder neuron fires at its own rate, drawn about 50% with a standard deviation of 5, 15 or 25 pp; under *correlation*, each encoder neuron keeps the rate 50% while the spikes are correlated across neurons through a single-factor Gaussian copula with latent correlation ρ, which gives a pairwise spike correlation of $(2/\pi)\arcsin\rho$. An offset thus keeps a Binomial spike count at a different rate, while dispersion and correlation keep a multivariate Bernoulli input whose spike count is no longer Binomial. Every layer stays initialized for an encoder rate of 50%.

Table 15: **Our initialization under violations of the input assumption.** Layers 1, 5, 10, 20 and 30 at initialization, with every layer initialized for an encoder rate of 50%, our initialization only; mean over 10 seeds. The encoder departs from (A5) one property at a time and delivers the rate in the second column; for correlation, ρ is the latent correlation of the copula, with the pairwise spike correlation in parentheses. Each cell gives the spike rate $\bar{p}^l$ in % (target $p = 50\%$) as mean $\pm$ standard deviation across the neurons of the reported layer l.

Condition	Delivered rate (%)	Layer 1	Layer 5	Layer 10	Layer 20	Layer 30
Matched input	50.0	50.2±0.5	50.7±0.5	51.0±0.5	51.3±0.5	51.3±0.5
Offset						
−25 pp	25.0	29.2±0.5	40.9±0.5	47.4±0.5	50.7±0.5	51.2±0.5
−15 pp	35.0	37.9±0.5	45.2±0.5	49.0±0.5	50.9±0.5	51.2±0.5
−5 pp	45.0	46.2±0.5	48.9±0.5	50.4±0.5	51.1±0.4	51.2±0.5
+5 pp	55.0	54.3±0.5	52.6±0.5	51.7±0.5	51.3±0.5	51.3±0.5
+15 pp	65.0	62.7±0.4	56.4±0.4	53.2±0.5	51.5±0.5	51.3±0.5
+25 pp	75.0	71.8±0.4	61.3±0.4	55.0±0.4	51.8±0.5	51.3±0.5
Dispersion						
5 pp	50.1	50.5±4.2	50.8±2.3	51.1±1.2	51.3±0.5	51.3±0.5
15 pp	50.1	50.8±12.7	50.5±7.0	51.1±3.3	51.1±0.8	51.2±0.5
25 pp	50.7	52.1±20.7	52.0±10.7	51.5±4.9	51.3±1.1	51.3±0.5
Correlation						
$\rho = 0.10$ (0.06)	50.0	50.3±0.5	50.8±0.5	51.0±0.5	51.2±0.5	51.3±0.5
$\rho = 0.50$ (0.33)	50.1	50.1±0.4	50.3±0.4	50.6±0.5	50.5±0.5	50.3±0.5
$\rho = 0.71$ (0.50)	50.0	49.8±0.4	49.0±0.4	48.5±0.4	47.2±0.5	46.6±0.4

Results. Offsets of up to ± 25 pp in the delivered rate, still visible at layer 1 at 29.2–71.8%, and a per-neuron rate dispersion of up to 25 pp, visible at layer 1 as a spread across neurons of up to 20.7 pp, are absorbed with depth: layer 30 settles at 51.2–51.3% with a spread of 0.5 pp, as under matched input (Table 15). Correlation is the one perturbation that shifts the settled rate, to 50.3% and 46.6% at pairwise spike correlations of 0.33 and 0.50, with the spread across neurons unchanged; the larger shift builds up with depth from 49.8% at layer 1. Within the tested range, the closure therefore does not rely on an exactly matched input at the first layer: depth absorbs departures in rate and in distribution across neurons, and correlation across neurons moves the settled rate at most 3.4 pp below the target without loss of per-neuron precision.

C.3.4 Extension – LIF neuron models

Setup. Each study feeds binarized Gaussian noise through a deep spiking network at initialization and records every neuron's spike rate and membrane-potential variance (Section C.1). We use the configuration of Table 7 across its rate grid with our initialization only. Each case makes one change: the leak $\beta = 1$, giving integrate-and-fire (IF) neurons; the leak $\beta = 0$ with zero reset, giving memoryless binary units; or the soft reset of Equation 42 with the reset U^*_{reset} of Equation 50 in place of the hard reset. Our initialization evaluates the closure at $\beta = 1$ for IF neurons (Section B.5.1), Equation 41 for binary units, and Equation 48 for the soft reset (Section B.5.2).

Table 16: **LIF neuron models across target rates and depth.** Layers 1, 5, 10, 20 and 30 at initialization, for the target rates $p = 15$–85%, our initialization only; mean over 10 seeds. Each cell gives the spike rate $\bar{p}^l$ in % (target p) followed, in parentheses, by the membrane-potential variance $\bar{\sigma}^2_{U^l}$ (target $\sigma^2_{U^l} = 1$), each as mean $\pm$ standard deviation across the neurons of the reported layer l. Integrate-and-fire neurons ($\beta = 1$), memoryless binary units ($\beta = 0$, zero reset) and soft-reset neurons with U^*_{reset} of Equation 50; base case: Table 11.

	$p = 15\%$	$p = 35\%$	$p = 50\%$	$p = 65\%$	$p = 85\%$
Integrate-and-fire, $\beta = 1$					
Layer 1	15.5±0.4 (1.24±0.07)	35.5±0.5 (1.07±0.03)	50.3±0.5 (1.02±0.02)	65.1±0.4 (1.00±0.02)	85.1±0.3 (1.00±0.01)
Layer 5	17.1±0.4 (1.06±0.06)	36.6±0.5 (1.02±0.04)	51.0±0.5 (0.98±0.02)	65.4±0.4 (0.99±0.02)	85.8±0.4 (0.95±0.02)
Layer 10	18.3±0.4 (1.03±0.05)	37.3±0.5 (1.01±0.03)	51.5±0.5 (0.97±0.02)	65.6±0.5 (0.99±0.02)	85.4±0.3 (0.95±0.02)
Layer 20	19.3±0.4 (1.00±0.04)	37.8±0.5 (1.00±0.03)	51.7±0.5 (0.97±0.02)	65.9±0.4 (0.98±0.02)	83.7±0.4 (1.04±0.02)
Layer 30	19.7±0.4 (0.98±0.03)	37.9±0.5 (0.99±0.03)	51.8±0.5 (0.96±0.02)	66.1±0.4 (0.98±0.02)	82.3±0.4 (1.12±0.03)
Binary unit, $\beta = 0$					
Layer 1	14.7±0.3 (0.99±0.01)	35.0±0.5 (1.00±0.01)	50.0±0.5 (1.00±0.01)	65.0±0.5 (1.00±0.01)	85.2±0.4 (1.00±0.01)
Layer 5	14.0±0.4 (0.94±0.02)	34.6±0.5 (0.99±0.02)	50.0±0.5 (1.00±0.02)	65.8±0.5 (0.99±0.02)	87.1±0.4 (0.89±0.02)
Layer 10	13.2±0.4 (0.89±0.02)	34.2±0.5 (0.98±0.02)	50.0±0.5 (1.00±0.02)	67.7±0.5 (0.97±0.02)	88.4±0.3 (0.78±0.02)
Layer 20	12.0±0.4 (0.80±0.02)	33.9±0.5 (0.98±0.02)	50.0±0.5 (1.00±0.02)	71.9±0.5 (0.90±0.02)	90.0±0.3 (0.67±0.02)
Layer 30	10.8±0.4 (0.72±0.02)	33.6±0.5 (0.97±0.02)	49.9±0.5 (1.00±0.02)	75.7±0.4 (0.82±0.01)	91.3±0.3 (0.59±0.02)
Soft reset, $\beta = 0.9$					
Layer 1	15.0±0.3 (1.05±0.03)	35.1±0.5 (1.01±0.02)	50.0±0.5 (1.00±0.02)	64.9±0.5 (1.01±0.02)	85.0±0.3 (1.05±0.03)
Layer 5	14.4±0.3 (0.91±0.02)	35.1±0.5 (0.96±0.02)	50.0±0.5 (0.97±0.02)	64.9±0.4 (0.97±0.02)	85.7±0.3 (0.91±0.02)
Layer 10	13.6±0.3 (0.86±0.02)	34.9±0.5 (0.95±0.02)	50.0±0.5 (0.97±0.02)	65.1±0.4 (0.95±0.02)	87.2±0.3 (0.84±0.02)
Layer 20	12.5±0.3 (0.81±0.02)	34.6±0.4 (0.94±0.02)	50.0±0.5 (0.96±0.02)	65.5±0.4 (0.94±0.02)	91.3±0.2 (0.66±0.02)
Layer 30	11.9±0.3 (0.77±0.02)	34.3±0.4 (0.93±0.02)	49.9±0.4 (0.96±0.02)	65.7±0.4 (0.94±0.02)	95.8±0.2 (0.39±0.01)

Results. For IF neurons the spread across neurons stays at the resolution of the estimator, at most 0.5 pp, at every target rate and reported layer, while the layer mean carries a larger offset than in the reference network, which builds up with depth to between -2.7 and $+4.7$ pp at layer 30 (Table 16). The binary unit and the soft reset hold the target at $p = 50\%$ at every reported layer, at 49.9–50.0% with a spread across neurons of at most 0.5 pp and a membrane variance of 0.96–1.00; the soft reset stays within 0.7 pp of the target over $p = 35$–65%. Elsewhere both diverge with depth from a first layer within 0.3 pp of the target: at layer 30 the binary unit reaches 75.7% at $p = 65\%$ and the soft reset 95.8% at $p = 85\%$, and both diverge slightly at $p = 15\%$.

C.3.5 Extension – heterogeneous populations

Setup. Each study feeds binarized Gaussian noise through a deep spiking network at initialization and records every neuron's spike rate and membrane-potential variance (Section C.1). We use the configuration of Table 7 across its rate grid with our initialization only. The per-neuron thresholds and leak factors are drawn independently as $\theta^l_j \sim \mathcal{U}(0.5, 1.5)$ and $\beta^l_j \sim \mathcal{U}(0, 1)$ for every layer and

seed. Our initialization sets every layer by Algorithm 1 on its realized vectors, with the per-neuron target means of Equation 51 (Section B.6). In order to provide a baseline for comparison, we include a homogeneous configuration as the control case with per-neuron $\theta = 1$ and $\beta = 0.5$ (the means of the sampled ranges). This contrasts with the main configuration of this work, the homogeneous $\theta = 1$ and $\beta = 0.9$ of Section C.3.1.

Table 17: **Heterogeneous populations across target rates and depth.** Layers 1, 5, 10, 20 and 30 at initialization, for the target rates $p = 15$–85%, our initialization only; mean over 10 seeds. Each cell gives the spike rate $\bar{p}^l$ in % (target p) followed, in parentheses, by the membrane-potential variance $\bar{\sigma}^2_{U^l}$ (target $\sigma^2_{U^l} = 1$), each as mean $\pm$ standard deviation across the neurons of the reported layer l. Heterogeneous per-neuron $\theta^l_j \sim \mathcal{U}(0.5, 1.5)$ and $\beta^l_j \sim \mathcal{U}(0, 1)$, and the homogeneous control at the means of these ranges; control $\theta = 1$, $\beta = 0.9$: Table 11.

	$p = 15\%$	$p = 35\%$	$p = 50\%$	$p = 65\%$	$p = 85\%$
Heterogeneous θ^l_j, β^l_j					
Layer 1	14.9±0.4 (1.01±0.04)	35.1±0.5 (1.00±0.02)	50.1±0.5 (1.00±0.02)	65.0±0.5 (1.00±0.01)	85.1±0.3 (1.00±0.01)
Layer 5	14.2±0.5 (0.95±0.05)	35.1±0.6 (0.99±0.03)	50.2±0.6 (0.99±0.02)	65.5±0.5 (0.99±0.02)	86.5±0.5 (0.91±0.02)
Layer 10	13.5±0.6 (0.91±0.05)	35.0±0.7 (0.99±0.03)	50.3±0.6 (0.99±0.02)	66.2±0.5 (0.98±0.03)	87.2±0.6 (0.85±0.03)
Layer 20	12.4±0.7 (0.85±0.06)	34.9±0.7 (0.99±0.03)	50.4±0.6 (0.99±0.02)	67.7±0.6 (0.96±0.03)	87.5±0.8 (0.82±0.04)
Layer 30	11.6±0.7 (0.79±0.05)	34.9±0.6 (0.99±0.03)	50.4±0.6 (0.99±0.02)	68.9±0.6 (0.94±0.03)	88.3±0.8 (0.77±0.04)
Homogeneous control, $\theta = 1$, $\beta = 0.5$					
Layer 1	14.9±0.3 (0.99±0.02)	35.0±0.5 (1.00±0.01)	50.0±0.5 (1.00±0.01)	65.0±0.5 (1.00±0.01)	85.1±0.4 (1.00±0.01)
Layer 5	13.8±0.4 (0.92±0.02)	34.9±0.5 (0.99±0.02)	50.2±0.5 (0.99±0.02)	65.5±0.5 (0.99±0.02)	86.3±0.4 (0.92±0.02)
Layer 10	12.7±0.3 (0.84±0.02)	34.7±0.5 (0.98±0.02)	50.2±0.5 (0.99±0.02)	66.2±0.4 (0.98±0.02)	86.9±0.3 (0.86±0.02)
Layer 20	11.0±0.3 (0.73±0.02)	34.5±0.5 (0.98±0.02)	50.2±0.5 (0.99±0.02)	67.7±0.4 (0.96±0.02)	87.3±0.3 (0.83±0.02)
Layer 30	9.8±0.3 (0.64±0.01)	34.5±0.5 (0.98±0.02)	50.1±0.5 (0.99±0.02)	68.9±0.4 (0.94±0.02)	87.8±0.3 (0.79±0.02)

Results. At $p = 50\%$ the heterogeneous population holds the target per neuron at every reported layer, at 50.1–50.4% with a spread across neurons of at most 0.6 pp and a membrane variance of 0.99–1.00, while thresholds span a three-fold range and leak factors the full interval $[0, 1]$ (Table 17). It stays close to the homogeneous control with matched means, within 0.5 pp at every reported layer over $p = 35$–85%; at $p = 15\%$ both diverge slightly with depth. The first layer is within 0.1 pp of the target at every target rate, whereas at layer 30 the mean deviates by $+3.9$ and $+3.3$ pp at $p = 65\%$ and 85%, with the spread at or below 0.8 pp.

C.3.6 Extension – residual and recurrent connections

Setup. Each study feeds binarized Gaussian noise through a deep spiking network at initialization and records every neuron's spike rate and membrane-potential variance (Section C.1). We use the configuration of Table 7 across its rate grid with our initialization only. Each case adds one input stream. *Residual*: every layer $l \geq 3$ also receives the spikes S^{l-2}_t of layer $l - 2$; layers 1 and 2 have no residual source. *Recurrent*: every layer also receives its own spikes of the previous time step, S^l_{t-1} ($h = 1$, diagonal masked). Our initialization allocates the weights by Equation 56, with $\omega_X = 0.9$ for the feed-forward stream, $\omega_{\text{res}} = 0.1$ or $\omega_{\text{rec}} = 0.1$ for the additional stream, and the whole mean on the feed-forward stream (Section B.7.1).

Results. Both streams leave the layer mean close to where the feed-forward network puts it: the residual stream agrees with the reference network to within 0.2 pp at every target rate and reported layer, and the recurrent stream to within 0.8 pp over $p = 15$–65% (Table 18, Table 11). The residual stream raises the spread across neurons at $p = 50\%$ from 0.5 to 0.9 pp, a level reached by layer 10, whereas the recurrent stream keeps it at or below 0.5 pp at every rate and layer. At $p = 85\%$ the recurrent stream drifts above the target with depth, to 2.7 pp at layer 30.

Table 18: **Residual and recurrent connections across target rates and depth.** Layers 1, 5, 10, 20 and 30 at initialization, for the target rates $p = 15$–85%, our initialization only; mean over 10 seeds. Each cell gives the spike rate $\bar{p}^l$ in % (target p) followed, in parentheses, by the membrane-potential variance $\bar{\sigma}^2_{U^l}$ (target $\sigma^2_{U^l} = 1$), each as mean $\pm$ standard deviation across the neurons of the reported layer l. A residual stream of the spikes S_t^{l-2} or a recurrent stream of the layer's own past spikes S_{t-1}^l, each with 10% of the variance budget; base case: Table 11.

	$p = 15\%$	$p = 35\%$	$p = 50\%$	$p = 65\%$	$p = 85\%$
Residual stream, S_t^{l-2}					
Layer 1	15.3±0.3 (1.06±0.03)	35.4±0.5 (1.02±0.02)	50.2±0.5 (1.01±0.02)	65.1±0.5 (1.00±0.02)	85.1±0.3 (1.00±0.01)
Layer 5	15.3±0.4 (0.97±0.06)	36.1±0.6 (0.99±0.06)	50.7±0.8 (0.98±0.06)	65.3±0.9 (0.99±0.05)	85.8±0.6 (0.95±0.04)
Layer 10	15.0±0.4 (0.95±0.06)	36.4±0.6 (0.98±0.06)	51.1±0.9 (0.97±0.05)	65.6±1.0 (0.99±0.05)	85.6±0.6 (0.93±0.05)
Layer 20	14.7±0.4 (0.93±0.06)	36.6±0.7 (0.98±0.06)	51.2±0.9 (0.97±0.05)	66.0±0.9 (0.98±0.05)	84.6±0.6 (0.98±0.05)
Layer 30	14.5±0.4 (0.92±0.06)	36.7±0.7 (0.98±0.06)	51.4±0.9 (0.97±0.05)	66.4±0.9 (0.98±0.05)	83.9±0.6 (1.02±0.05)
Recurrent stream, S_{t-1}^l					
Layer 1	15.3±0.3 (1.05±0.03)	35.3±0.5 (1.02±0.02)	50.2±0.5 (1.01±0.02)	65.1±0.4 (1.00±0.02)	85.1±0.3 (1.00±0.01)
Layer 5	15.3±0.4 (0.97±0.05)	36.1±0.5 (0.99±0.03)	50.7±0.5 (0.98±0.02)	65.2±0.5 (0.99±0.02)	86.2±0.4 (0.94±0.02)
Layer 10	15.2±0.4 (0.96±0.05)	36.4±0.5 (0.99±0.04)	51.0±0.5 (0.97±0.02)	65.5±0.5 (0.99±0.02)	87.1±0.3 (0.86±0.02)
Layer 20	15.1±0.4 (0.95±0.05)	36.6±0.5 (0.99±0.04)	51.3±0.5 (0.97±0.02)	65.8±0.5 (0.98±0.02)	87.6±0.3 (0.82±0.02)
Layer 30	15.2±0.4 (0.95±0.05)	36.7±0.5 (0.98±0.04)	51.3±0.5 (0.97±0.02)	65.9±0.5 (0.98±0.02)	87.7±0.3 (0.80±0.02)

C.3.7 Extension – convolutional layers

Setup. Each study feeds binarized Gaussian noise through a deep spiking network at initialization and records every neuron's spike rate and membrane-potential variance (Section C.1). We use the configuration of Table 7 across its rate grid with our initialization only. Ten convolutional layers replace the dense layers, each with $C = 32$ filters of size 3×3 on 28×28 feature maps, stride one and circular padding; every layer holds $25{,}088$ neurons with fan-in $Ck^2 = 288$. Our initialization treats each kernel as the weight matrix of Equation 58, one row per output channel, with a bias that carries the mean correction (Section B.7.2). The encoder is a dense layer with orthonormal rows whose binarized output spikes at the target rate; statistics are aggregated over all neurons of a layer, from one seed.

Table 19: **Convolutional layers across target rates and depth.** Layers 1, 5 and 10 at initialization, for the target rates $p = 15$–85%, our initialization only; one seed. Each cell gives the spike rate $\bar{p}^l$ in % (target p) followed, in parentheses, by the membrane-potential variance $\bar{\sigma}^2_{U^l}$ (target $\sigma^2_{U^l} = 1$), each as mean $\pm$ standard deviation across the neurons of the reported layer l.

	$p = 15\%$	$p = 35\%$	$p = 50\%$	$p = 65\%$	$p = 85\%$
CNN					
Layer 1	15.3±0.3 (1.07±0.03)	35.4±0.5 (1.02±0.02)	50.2±0.5 (1.01±0.02)	65.1±0.5 (1.00±0.02)	85.1±0.3 (1.00±0.01)
Layer 5	15.2±0.4 (0.99±0.05)	35.4±0.5 (0.99±0.05)	50.2±0.6 (0.98±0.04)	64.9±0.6 (0.99±0.04)	85.0±0.5 (1.00±0.03)
Layer 10	15.2±0.4 (0.99±0.04)	35.4±0.5 (1.00±0.04)	50.2±0.7 (0.98±0.03)	65.0±0.7 (0.99±0.03)	85.0±0.5 (1.00±0.02)

Results. Every reported layer holds the target per neuron at every target rate: the spike rate lies within 0.4 pp of the target, with a spread across neurons of at most 0.7 pp and a membrane variance of 0.98–1.07 (Table 19).

C.4 Empirical evaluation: extended results

This subsection reports the per-cell results behind Table 3, per paradigm. The signal propagation of the same five configurations at initialization, in the same architectural cells, is reported in Section C.3.2.

C.4.1 UEA multivariate time-series classification

Table 20 presents cell-level test-set accuracy across the architectural sweep grid for the eight UEA datasets [Bagnall et al., 2018] included in the main paper, while Table 22 reports the corresponding results for the five UEA datasets omitted from the main analysis, which are excluded due to the initialization schemes yielding negligible differentiation across configurations, rendering comparisons

uninformative, and/or because classification performance stays near chance level; each cell reports the mean and standard deviation of the test-set accuracy over twenty random seeds for each of the five initialization configurations introduced in Section 3.2.

C.4.2 Neuromorphic classification

Table 24 reports cell-level test-set accuracy across the architectural sweep grid for the three neuromorphic (SHD, DVS Gesture and ECG) datasets. Each cell value is the mean and standard deviation of the test-set accuracy across the twenty random seeds, for each of the five initialization configurations.

C.4.3 Continuous-control reinforcement learning

Table 25 reports cell-level raw episode reward across the architectural sweep grid for the three reinforcement-learning environments. Each cell value is the mean and standard deviation of the raw episode reward across the twenty random seeds, for each of the five initialization configurations.

Table 20: **UEA cell-level results (included datasets, part 1 of 2).** Per-cell test-set accuracy on the eight UEA datasets reported in the main paper, broken out by width, depth, and initialization configuration. The spread column gives the standard deviation across the five configuration means of each cell and its mean over the cells in the dataset rows. Cell values are mean ± standard deviation across 20 seeds. Metrics presented are those computed across the last 10% of evaluation data.

width	depth	naive	I_t default	$I_t^{\perp}$ enhanced	$U_t \sim \mathcal{N}(0,1)$	$U_t \sim \mathcal{N}(\theta,1)$	spread
BasicMotions (4 classes, chance 25.00%)							0.069
32	2	0.893 ± 0.094	0.896 ± 0.168	0.858 ± 0.141	0.889 ± 0.135	**0.976** ± 0.071	0.039
32	3	0.856 ± 0.130	0.880 ± 0.143	0.853 ± 0.187	0.863 ± 0.131	**0.981** ± 0.060	0.048
32	4	0.716 ± 0.159	0.878 ± 0.188	0.828 ± 0.181	0.849 ± 0.127	**0.913** ± 0.180	0.067
32	5	0.465 ± 0.203	0.884 ± 0.124	0.918 ± 0.117	0.817 ± 0.123	**0.932** ± 0.108	0.174
64	2	0.950 ± 0.056	0.963 ± 0.092	0.968 ± 0.082	0.943 ± 0.090	**0.984** ± 0.058	0.014
64	3	0.927 ± 0.087	0.963 ± 0.092	0.949 ± 0.102	0.964 ± 0.068	**0.976** ± 0.074	0.017
64	4	0.783 ± 0.132	0.949 ± 0.098	0.935 ± 0.137	0.906 ± 0.128	**0.979** ± 0.066	0.068
64	5	0.519 ± 0.187	0.889 ± 0.126	**0.956** ± 0.092	0.858 ± 0.150	0.927 ± 0.105	0.159
128	2	0.984 ± 0.048	0.990 ± 0.044	**1.000** ± 0.000	**1.000** ± 0.000	0.966 ± 0.083	0.013
128	3	0.945 ± 0.052	**1.000** ± 0.000	**1.000** ± 0.000	0.983 ± 0.049	0.979 ± 0.066	0.020
128	4	0.874 ± 0.090	**1.000** ± 0.000	**1.000** ± 0.000	0.960 ± 0.081	0.955 ± 0.092	0.046
128	5	0.564 ± 0.211	**1.000** ± 0.000	**1.000** ± 0.000	0.884 ± 0.122	0.903 ± 0.122	0.160
Cricket (12 classes, chance 8.33%)							0.074
32	2	0.807 ± 0.072	0.800 ± 0.089	0.799 ± 0.072	0.815 ± 0.080	**0.902** ± 0.057	0.039
32	3	0.812 ± 0.053	0.838 ± 0.072	0.845 ± 0.049	0.839 ± 0.059	**0.924** ± 0.044	0.038
32	4	0.612 ± 0.088	0.806 ± 0.082	0.850 ± 0.081	0.870 ± 0.044	**0.919** ± 0.037	0.106
32	5	0.389 ± 0.130	0.832 ± 0.054	0.863 ± 0.077	0.868 ± 0.057	**0.888** ± 0.042	0.190
64	2	0.915 ± 0.059	0.908 ± 0.049	0.907 ± 0.068	0.886 ± 0.056	**0.971** ± 0.019	0.028
64	3	0.872 ± 0.036	0.951 ± 0.037	0.931 ± 0.077	0.941 ± 0.040	**0.958** ± 0.026	0.031
64	4	0.743 ± 0.083	0.915 ± 0.049	0.958 ± 0.031	0.943 ± 0.029	**0.966** ± 0.015	0.083
64	5	0.454 ± 0.109	0.941 ± 0.046	0.945 ± 0.049	0.938 ± 0.024	**0.950** ± 0.024	0.196
128	2	0.945 ± 0.046	0.965 ± 0.035	0.960 ± 0.033	0.945 ± 0.041	**0.977** ± 0.024	0.012
128	3	0.920 ± 0.020	0.979 ± 0.020	**0.982** ± 0.020	0.954 ± 0.028	0.980 ± 0.018	0.024
128	4	0.892 ± 0.034	**0.984** ± 0.009	**0.984** ± 0.008	0.953 ± 0.025	0.960 ± 0.021	0.034
128	5	0.678 ± 0.103	**0.974** ± 0.022	0.968 ± 0.019	0.930 ± 0.022	0.933 ± 0.021	0.111
Epilepsy (4 classes, chance 25.00%)							0.024
32	2	**0.874** ± 0.056	0.853 ± 0.076	0.810 ± 0.133	0.793 ± 0.076	0.864 ± 0.074	0.032
32	3	**0.918** ± 0.020	0.852 ± 0.101	0.873 ± 0.058	0.830 ± 0.053	0.900 ± 0.050	0.032
32	4	0.911 ± 0.030	0.849 ± 0.099	0.902 ± 0.061	0.844 ± 0.079	**0.912** ± 0.031	0.031
32	5	0.790 ± 0.097	0.861 ± 0.087	0.832 ± 0.125	0.841 ± 0.085	**0.923** ± 0.012	0.043
64	2	**0.929** ± 0.038	0.881 ± 0.089	0.883 ± 0.083	0.855 ± 0.070	0.920 ± 0.042	0.027
64	3	**0.952** ± 0.008	0.912 ± 0.052	0.901 ± 0.079	0.890 ± 0.048	0.938 ± 0.045	0.023
64	4	**0.954** ± 0.006	0.933 ± 0.022	0.921 ± 0.045	0.898 ± 0.031	0.948 ± 0.025	0.020
64	5	0.900 ± 0.097	0.926 ± 0.030	0.919 ± 0.042	0.910 ± 0.008	**0.939** ± 0.038	0.013
128	2	**0.951** ± 0.007	0.903 ± 0.077	0.892 ± 0.070	0.869 ± 0.080	0.938 ± 0.054	0.030
128	3	**0.959** ± 0.004	0.933 ± 0.059	0.926 ± 0.045	0.925 ± 0.007	0.957 ± 0.023	0.015
128	4	0.955 ± 0.004	0.931 ± 0.051	0.947 ± 0.007	0.927 ± 0.007	**0.961** ± 0.004	0.013
128	5	0.952 ± 0.008	0.933 ± 0.030	0.942 ± 0.007	0.932 ± 0.006	**0.958** ± 0.005	0.010
RacketSports (4 classes, chance 25.00%)							0.052
32	2	0.684 ± 0.041	**0.735** ± 0.034	0.715 ± 0.048	0.667 ± 0.030	0.713 ± 0.029	0.024
32	3	0.540 ± 0.046	0.662 ± 0.078	**0.687** ± 0.064	0.628 ± 0.034	0.676 ± 0.031	0.053
32	4	0.508 ± 0.028	0.610 ± 0.056	0.605 ± 0.039	0.592 ± 0.025	**0.642** ± 0.029	0.045
32	5	0.523 ± 0.027	0.558 ± 0.038	0.577 ± 0.045	0.542 ± 0.030	**0.588** ± 0.028	0.023
64	2	0.743 ± 0.018	0.791 ± 0.026	**0.798** ± 0.017	0.719 ± 0.021	0.766 ± 0.016	0.030
64	3	0.638 ± 0.066	**0.780** ± 0.021	0.776 ± 0.039	0.691 ± 0.019	0.743 ± 0.025	0.054
64	4	0.518 ± 0.016	0.729 ± 0.035	**0.745** ± 0.029	0.652 ± 0.020	0.702 ± 0.030	0.082
64	5	0.533 ± 0.021	0.646 ± 0.068	**0.657** ± 0.054	0.600 ± 0.029	0.626 ± 0.030	0.044
128	2	0.774 ± 0.015	0.818 ± 0.050	**0.824** ± 0.010	0.761 ± 0.020	0.800 ± 0.013	0.024
128	3	0.724 ± 0.013	**0.828** ± 0.033	0.826 ± 0.015	0.748 ± 0.016	0.779 ± 0.020	0.041
128	4	0.515 ± 0.019	0.804 ± 0.017	**0.812** ± 0.021	0.716 ± 0.025	0.715 ± 0.017	0.107
128	5	0.519 ± 0.026	0.759 ± 0.022	**0.769** ± 0.050	0.681 ± 0.021	0.630 ± 0.089	0.092

Table 21: **UEA cell-level results (included datasets, part 2 of 2).** Continuation of Table 20.

width	depth	naive	I_t default	$I_t^{\perp}$ enhanced	$U_t \sim \mathcal{N}(0,1)$	$U_t \sim \mathcal{N}(\theta,1)$	spread
NATOPS (6 classes, chance 16.67%)							0.044
32	2	0.716 ± 0.024	0.746 ± 0.049	0.735 ± 0.055	0.724 ± 0.044	$\mathbf{0.752} \pm 0.031$	0.013
32	3	0.670 ± 0.048	0.661 ± 0.106	0.705 ± 0.079	0.696 ± 0.037	$\mathbf{0.731} \pm 0.039$	0.025
32	4	0.566 ± 0.090	0.658 ± 0.060	0.650 ± 0.057	0.663 ± 0.048	$\mathbf{0.708} \pm 0.031$	0.046
32	5	0.476 ± 0.098	0.626 ± 0.047	0.618 ± 0.072	0.631 ± 0.047	$\mathbf{0.678} \pm 0.030$	0.068
64	2	0.730 ± 0.036	0.796 ± 0.019	0.774 ± 0.053	0.774 ± 0.039	$\mathbf{0.826} \pm 0.022$	0.031
64	3	0.677 ± 0.016	0.749 ± 0.067	0.765 ± 0.049	0.744 ± 0.036	$\mathbf{0.793} \pm 0.026$	0.038
64	4	0.634 ± 0.061	$\mathbf{0.728} \pm 0.031$	0.707 ± 0.049	0.704 ± 0.015	0.724 ± 0.022	0.034
64	5	0.469 ± 0.055	0.678 ± 0.054	0.680 ± 0.046	0.690 ± 0.025	$\mathbf{0.699} \pm 0.021$	0.087
128	2	0.819 ± 0.024	0.831 ± 0.019	0.823 ± 0.047	0.828 ± 0.021	$\mathbf{0.857} \pm 0.035$	0.013
128	3	0.698 ± 0.027	0.796 ± 0.035	0.789 ± 0.035	0.795 ± 0.021	$\mathbf{0.824} \pm 0.022$	0.043
128	4	0.685 ± 0.053	0.748 ± 0.054	0.763 ± 0.043	0.735 ± 0.039	$\mathbf{0.765} \pm 0.029$	0.029
128	5	0.465 ± 0.059	0.703 ± 0.042	$\mathbf{0.719} \pm 0.029$	0.691 ± 0.019	0.707 ± 0.024	0.096
UWaveGestureLibrary (8 classes, chance 12.50%)							0.052
32	2	$\mathbf{0.720} \pm 0.042$	0.641 ± 0.056	0.640 ± 0.051	0.535 ± 0.070	0.530 ± 0.048	0.072
32	3	$\mathbf{0.758} \pm 0.033$	0.643 ± 0.066	0.658 ± 0.061	0.590 ± 0.059	0.580 ± 0.030	0.064
32	4	0.627 ± 0.076	0.628 ± 0.066	$\mathbf{0.633} \pm 0.052$	0.561 ± 0.060	0.580 ± 0.038	0.030
32	5	0.340 ± 0.102	0.580 ± 0.060	$\mathbf{0.592} \pm 0.061$	0.566 ± 0.046	0.572 ± 0.047	0.095
64	2	$\mathbf{0.796} \pm 0.031$	0.736 ± 0.030	0.732 ± 0.037	0.647 ± 0.045	0.657 ± 0.042	0.055
64	3	$\mathbf{0.822} \pm 0.016$	0.743 ± 0.047	0.777 ± 0.034	0.692 ± 0.033	0.693 ± 0.027	0.050
64	4	0.735 ± 0.053	$\mathbf{0.738} \pm 0.030$	$\mathbf{0.738} \pm 0.035$	0.701 ± 0.035	0.708 ± 0.043	0.016
64	5	0.452 ± 0.125	0.695 ± 0.043	$\mathbf{0.725} \pm 0.025$	0.697 ± 0.027	0.697 ± 0.023	0.101
128	2	$\mathbf{0.843} \pm 0.023$	0.795 ± 0.038	0.804 ± 0.025	0.703 ± 0.025	0.734 ± 0.029	0.050
128	3	$\mathbf{0.864} \pm 0.010$	0.809 ± 0.027	0.812 ± 0.016	0.762 ± 0.022	0.789 ± 0.018	0.034
128	4	$\mathbf{0.844} \pm 0.013$	0.796 ± 0.021	0.793 ± 0.027	0.775 ± 0.024	0.787 ± 0.011	0.024
128	5	0.671 ± 0.109	0.765 ± 0.027	$\mathbf{0.772} \pm 0.015$	0.767 ± 0.018	0.758 ± 0.027	0.038
ArticularyWordRecognition (25 classes, chance 4.00%)							0.070
32	2	$\mathbf{0.874} \pm 0.033$	0.778 ± 0.053	0.804 ± 0.035	0.828 ± 0.035	0.866 ± 0.026	0.036
32	3	0.842 ± 0.022	0.780 ± 0.045	0.805 ± 0.050	0.838 ± 0.036	$\mathbf{0.870} \pm 0.024$	0.031
32	4	0.656 ± 0.062	0.754 ± 0.051	0.773 ± 0.034	0.831 ± 0.026	$\mathbf{0.841} \pm 0.024$	0.066
32	5	0.242 ± 0.068	0.690 ± 0.038	0.734 ± 0.036	$\mathbf{0.801} \pm 0.041$	0.781 ± 0.044	0.207
64	2	$\mathbf{0.940} \pm 0.016$	0.903 ± 0.030	0.901 ± 0.035	0.914 ± 0.020	0.939 ± 0.018	0.017
64	3	0.909 ± 0.015	0.911 ± 0.033	0.911 ± 0.034	0.920 ± 0.017	$\mathbf{0.945} \pm 0.015$	0.013
64	4	0.798 ± 0.049	0.889 ± 0.029	0.914 ± 0.017	0.921 ± 0.013	$\mathbf{0.928} \pm 0.014$	0.048
64	5	0.370 ± 0.140	0.857 ± 0.027	0.871 ± 0.033	$\mathbf{0.900} \pm 0.012$	0.891 ± 0.016	0.204
128	2	0.969 ± 0.005	0.955 ± 0.017	0.963 ± 0.010	0.954 ± 0.012	$\mathbf{0.972} \pm 0.006$	0.007
128	3	0.939 ± 0.011	0.959 ± 0.011	0.963 ± 0.007	0.958 ± 0.007	$\mathbf{0.967} \pm 0.007$	0.010
128	4	0.888 ± 0.017	0.939 ± 0.019	$\mathbf{0.947} \pm 0.010$	0.941 ± 0.012	$\mathbf{0.947} \pm 0.012$	0.022
128	5	0.485 ± 0.135	0.922 ± 0.020	$\mathbf{0.929} \pm 0.013$	0.925 ± 0.015	0.922 ± 0.012	0.176
LSST (14 classes, chance 7.14%)							0.041
32	2	0.222 ± 0.015	0.231 ± 0.047	0.236 ± 0.049	$\mathbf{0.242} \pm 0.042$	0.214 ± 0.021	0.010
32	3	0.200 ± 0.012	0.209 ± 0.023	0.225 ± 0.046	0.169 ± 0.023	$\mathbf{0.289} \pm 0.052$	0.040
32	4	0.195 ± 0.010	0.189 ± 0.027	0.184 ± 0.016	0.158 ± 0.002	$\mathbf{0.340} \pm 0.003$	0.065
32	5	0.188 ± 0.009	0.175 ± 0.025	0.173 ± 0.016	0.158 ± 0.005	$\mathbf{0.336} \pm 0.011$	0.066
64	2	0.233 ± 0.012	0.236 ± 0.038	0.242 ± 0.036	$\mathbf{0.258} \pm 0.059$	0.214 ± 0.012	0.014
64	3	0.201 ± 0.013	0.207 ± 0.024	0.203 ± 0.024	0.187 ± 0.020	$\mathbf{0.258} \pm 0.050$	0.024
64	4	0.193 ± 0.007	0.174 ± 0.013	0.177 ± 0.013	0.180 ± 0.026	$\mathbf{0.340} \pm 0.014$	0.064
64	5	0.199 ± 0.022	0.168 ± 0.012	0.168 ± 0.013	0.165 ± 0.012	$\mathbf{0.343} \pm 0.003$	0.068
128	2	0.239 ± 0.019	0.237 ± 0.022	$\mathbf{0.253} \pm 0.044$	0.242 ± 0.044	0.209 ± 0.011	0.015
128	3	0.212 ± 0.032	0.188 ± 0.020	0.196 ± 0.040	0.189 ± 0.036	$\mathbf{0.219} \pm 0.027$	0.013
128	4	0.203 ± 0.036	0.182 ± 0.010	0.182 ± 0.014	0.181 ± 0.018	$\mathbf{0.339} \pm 0.022$	0.061
128	5	0.204 ± 0.024	0.168 ± 0.006	0.170 ± 0.022	0.186 ± 0.040	$\mathbf{0.318} \pm 0.054$	0.056

Table 22: **UEA cell-level results (excluded datasets, part 1 of 2).** Per-cell test-set accuracy on the five UEA datasets excluded from the main paper, on which the configurations are indiscriminative. A dataset is excluded when every configuration's mean over its cells lies within 0.10 of the chance baseline or when the spread of its dataset row stays below 0.02. The spread column gives the standard deviation across the five configuration means of each cell and its mean over the cells in the dataset rows. Cell values are mean $\pm$ standard deviation across 20 seeds. Metrics presented are those computed across the last 10% of evaluation data. A centered star ($*$) in the width column marks a cell in which no configuration exceeds the chance baseline.

width	depth	naive	I_t default	$I_t^{\perp}$ enhanced	$U_t \sim \mathcal{N}(0,1)$	$U_t \sim \mathcal{N}(\theta,1)$	spread
AtrialFibrillation (3 classes, chance 33.33%)							0.013
$*32$	2	0.311 ± 0.043	0.325 ± 0.022	0.328 ± 0.021	0.326 ± 0.016	0.286 ± 0.044	0.016
$*32$	3	0.331 ± 0.008	0.330 ± 0.015	0.327 ± 0.021	0.329 ± 0.011	0.288 ± 0.038	0.017
$*32$	4	0.329 ± 0.013	0.313 ± 0.042	0.318 ± 0.035	0.323 ± 0.020	0.276 ± 0.026	0.019
$*32$	5	0.323 ± 0.024	0.327 ± 0.012	0.325 ± 0.021	0.309 ± 0.030	0.305 ± 0.040	0.009
64	2	0.329 ± 0.013	0.333 ± 0.000	0.333 ± 0.000	0.333 ± 0.000	0.316 ± 0.021	0.007
64	3	0.333 ± 0.000	0.333 ± 0.000	0.327 ± 0.030	0.333 ± 0.000	0.309 ± 0.025	0.009
64	4	0.333 ± 0.000	0.317 ± 0.036	0.313 ± 0.038	0.333 ± 0.000	0.295 ± 0.034	0.014
64	5	0.327 ± 0.021	0.301 ± 0.048	0.322 ± 0.032	0.333 ± 0.000	0.286 ± 0.026	0.018
128	2	0.333 ± 0.000	0.333 ± 0.000	0.333 ± 0.000	0.333 ± 0.000	0.331 ± 0.007	0.001
128	3	0.333 ± 0.000	0.317 ± 0.029	0.334 ± 0.018	0.333 ± 0.000	0.325 ± 0.014	0.007
128	4	0.332 ± 0.004	0.297 ± 0.046	0.293 ± 0.049	0.333 ± 0.000	0.319 ± 0.017	0.017
128	5	0.330 ± 0.007	0.279 ± 0.045	0.288 ± 0.041	0.333 ± 0.000	0.316 ± 0.019	0.022
FingerMovements (2 classes, chance 50.00%)							0.007
32	2	0.525 ± 0.018	0.490 ± 0.027	0.496 ± 0.020	0.495 ± 0.020	0.504 ± 0.026	0.012
32	3	0.517 ± 0.020	0.501 ± 0.027	0.495 ± 0.026	0.498 ± 0.018	0.489 ± 0.020	0.009
32	4	0.503 ± 0.014	0.490 ± 0.021	0.500 ± 0.023	0.485 ± 0.019	0.495 ± 0.023	0.007
32	5	0.498 ± 0.018	0.501 ± 0.026	0.494 ± 0.025	0.489 ± 0.018	0.497 ± 0.023	0.004
64	2	0.518 ± 0.015	0.508 ± 0.022	0.512 ± 0.018	0.523 ± 0.015	0.530 ± 0.026	0.008
64	3	0.515 ± 0.018	0.509 ± 0.016	0.513 ± 0.023	0.517 ± 0.015	0.518 ± 0.017	0.003
64	4	0.521 ± 0.019	0.506 ± 0.025	0.507 ± 0.022	0.513 ± 0.022	0.510 ± 0.018	0.005
64	5	0.502 ± 0.019	0.504 ± 0.021	0.498 ± 0.025	0.500 ± 0.017	0.492 ± 0.016	0.004
128	2	0.511 ± 0.017	0.520 ± 0.021	0.527 ± 0.016	0.528 ± 0.013	0.539 ± 0.014	0.009
128	3	0.517 ± 0.021	0.519 ± 0.018	0.522 ± 0.018	0.521 ± 0.015	0.528 ± 0.015	0.004
128	4	0.514 ± 0.013	0.516 ± 0.022	0.518 ± 0.020	0.520 ± 0.019	0.503 ± 0.017	0.006
128	5	0.498 ± 0.020	0.506 ± 0.025	0.505 ± 0.022	0.513 ± 0.018	0.492 ± 0.010	0.007
Handwriting (26 classes, chance 3.85%)							0.022
32	2	0.047 ± 0.010	0.054 ± 0.013	0.066 ± 0.012	0.042 ± 0.006	0.048 ± 0.008	0.008
32	3	0.040 ± 0.005	0.063 ± 0.014	0.066 ± 0.025	0.041 ± 0.004	0.046 ± 0.007	0.011
32	4	0.040 ± 0.010	0.058 ± 0.021	0.051 ± 0.015	0.042 ± 0.005	0.042 ± 0.006	0.007
32	5	0.040 ± 0.004	0.048 ± 0.017	0.045 ± 0.011	0.041 ± 0.004	0.042 ± 0.004	0.003
64	2	0.079 ± 0.017	0.104 ± 0.017	0.096 ± 0.013	0.041 ± 0.008	0.064 ± 0.012	0.023
64	3	0.071 ± 0.018	0.103 ± 0.026	0.110 ± 0.016	0.039 ± 0.003	0.069 ± 0.014	0.026
64	4	0.047 ± 0.009	0.089 ± 0.029	0.093 ± 0.027	0.038 ± 0.001	0.053 ± 0.011	0.023
64	5	0.038 ± 0.004	0.055 ± 0.022	0.067 ± 0.026	0.039 ± 0.002	0.040 ± 0.006	0.011
128	2	0.117 ± 0.011	0.147 ± 0.015	0.150 ± 0.014	0.056 ± 0.013	0.106 ± 0.013	0.034
128	3	0.096 ± 0.014	0.174 ± 0.017	0.180 ± 0.011	0.050 ± 0.009	0.109 ± 0.013	0.049
128	4	0.074 ± 0.013	0.144 ± 0.014	0.147 ± 0.010	0.040 ± 0.004	0.077 ± 0.020	0.042
128	5	0.051 ± 0.011	0.097 ± 0.019	0.093 ± 0.023	0.038 ± 0.000	0.040 ± 0.004	0.026

Table 23: **UEA cell-level results (excluded datasets, part 2 of 2).** Continuation of Table 22.

width	depth	naive	I_t default	$I_t^{\perp}$ enhanced	$U_t \sim \mathcal{N}(0,1)$	$U_t \sim \mathcal{N}(\theta,1)$	spread
Heartbeat (2 classes, chance 50.00%)							0.007
32	2	0.756 ± 0.009	0.763 ± 0.016	0.761 ± 0.015	0.762 ± 0.018	0.767 ± 0.021	0.004
32	3	0.740 ± 0.017	0.748 ± 0.016	0.763 ± 0.016	0.763 ± 0.012	0.758 ± 0.025	0.009
32	4	0.734 ± 0.014	0.743 ± 0.025	0.752 ± 0.023	0.762 ± 0.014	0.749 ± 0.019	0.009
32	5	0.736 ± 0.014	0.744 ± 0.017	0.739 ± 0.017	0.748 ± 0.018	0.735 ± 0.016	0.005
64	2	0.732 ± 0.013	0.749 ± 0.014	0.754 ± 0.014	0.762 ± 0.011	0.751 ± 0.025	0.010
64	3	0.727 ± 0.014	0.737 ± 0.016	0.742 ± 0.017	0.736 ± 0.018	0.743 ± 0.025	0.006
64	4	0.730 ± 0.017	0.738 ± 0.019	0.742 ± 0.016	0.736 ± 0.016	0.733 ± 0.020	0.004
64	5	0.725 ± 0.014	0.735 ± 0.017	0.743 ± 0.024	0.728 ± 0.012	0.733 ± 0.017	0.006
128	2	0.733 ± 0.011	0.742 ± 0.014	0.738 ± 0.012	0.726 ± 0.010	0.726 ± 0.013	0.006
128	3	0.735 ± 0.015	0.737 ± 0.012	0.736 ± 0.012	0.724 ± 0.011	0.727 ± 0.016	0.005
128	4	0.729 ± 0.017	0.739 ± 0.014	0.739 ± 0.019	0.722 ± 0.010	0.726 ± 0.013	0.007
128	5	0.713 ± 0.019	0.736 ± 0.020	0.734 ± 0.016	0.724 ± 0.007	0.722 ± 0.000	0.008
SelfRegulationSCP1 (2 classes, chance 50.00%)							0.004
32	2	0.771 ± 0.007	0.774 ± 0.007	0.776 ± 0.006	0.776 ± 0.006	0.775 ± 0.007	0.002
32	3	0.774 ± 0.011	0.777 ± 0.012	0.778 ± 0.011	0.774 ± 0.007	0.775 ± 0.009	0.002
32	4	0.771 ± 0.015	0.770 ± 0.020	0.776 ± 0.010	0.770 ± 0.006	0.774 ± 0.007	0.002
32	5	0.784 ± 0.014	0.770 ± 0.023	0.776 ± 0.016	0.774 ± 0.010	0.772 ± 0.010	0.005
64	2	0.773 ± 0.010	0.773 ± 0.011	0.778 ± 0.009	0.780 ± 0.005	0.777 ± 0.008	0.003
64	3	0.774 ± 0.010	0.769 ± 0.021	0.773 ± 0.018	0.779 ± 0.009	0.777 ± 0.007	0.003
64	4	0.772 ± 0.018	0.758 ± 0.034	0.774 ± 0.017	0.776 ± 0.014	0.777 ± 0.008	0.007
64	5	0.770 ± 0.025	0.765 ± 0.031	0.770 ± 0.027	0.773 ± 0.015	0.774 ± 0.012	0.003
128	2	0.767 ± 0.015	0.771 ± 0.016	0.769 ± 0.012	0.780 ± 0.007	0.775 ± 0.013	0.005
128	3	0.763 ± 0.021	0.766 ± 0.014	0.774 ± 0.013	0.781 ± 0.009	0.774 ± 0.010	0.006
128	4	0.764 ± 0.020	0.769 ± 0.018	0.778 ± 0.013	0.781 ± 0.007	0.775 ± 0.006	0.006
128	5	0.777 ± 0.020	0.775 ± 0.017	0.778 ± 0.011	0.779 ± 0.010	0.767 ± 0.013	0.004

Table 24: **Neuromorphic cell-level results.** Per-cell test-set accuracy on SHD, DVS Gesture, and ECG, broken out by width, depth, and initialization configuration. Cell values are mean ± standard deviation across 20 seeds. Metrics presented are those computed across the last 10% of evaluation data.

width	depth	naive	I_t default	$I_t^{\perp}$ enhanced	$U_t \sim \mathcal{N}(0,1)$	$U_t \sim \mathcal{N}(\theta,1)$
SHD (20 classes, chance 5.00%)						
32	2	65.19 ± 1.85	66.62 ± 2.19	**66.81** ± 2.51	60.57 ± 2.43	65.85 ± 1.75
32	3	54.83 ± 3.90	62.62 ± 2.83	63.78 ± 2.40	59.57 ± 1.99	**68.84** ± 1.23
32	4	40.38 ± 3.99	55.09 ± 3.47	57.34 ± 2.05	53.66 ± 2.99	**63.99** ± 2.32
32	5	32.49 ± 7.71	49.35 ± 3.19	49.03 ± 3.22	48.42 ± 2.87	**54.84** ± 3.13
64	2	71.96 ± 1.40	72.45 ± 1.48	72.27 ± 0.97	67.91 ± 1.64	**73.20** ± 1.70
64	3	68.05 ± 1.89	72.30 ± 2.04	72.36 ± 1.31	71.12 ± 1.91	**75.93** ± 1.48
64	4	58.29 ± 2.81	67.37 ± 1.97	67.56 ± 1.62	66.43 ± 2.47	**74.17** ± 1.71
64	5	53.23 ± 2.80	60.60 ± 1.78	60.39 ± 2.36	58.37 ± 2.24	**65.69** ± 2.64
128	2	76.21 ± 1.16	76.73 ± 1.14	76.33 ± 1.12	73.39 ± 0.97	**77.96** ± 1.06
128	3	76.33 ± 1.33	78.13 ± 0.99	78.36 ± 1.00	77.83 ± 1.30	**80.60** ± 1.06
128	4	70.24 ± 2.25	75.17 ± 1.94	74.74 ± 2.08	74.82 ± 1.36	**79.35** ± 1.67
128	5	66.12 ± 2.42	70.06 ± 1.87	68.39 ± 2.32	69.12 ± 1.30	**75.46** ± 2.08
DVS Gesture (11 classes, chance 9.09%)						
32	2	62.55 ± 1.23	65.34 ± 2.05	66.96 ± 1.33	64.22 ± 1.36	**75.77** ± 2.14
32	3	58.90 ± 4.33	65.28 ± 1.99	65.35 ± 1.33	63.20 ± 1.15	**67.00** ± 2.08
32	4	48.91 ± 4.84	61.81 ± 2.26	63.98 ± 2.64	62.24 ± 1.37	**65.06** ± 1.33
32	5	41.19 ± 3.32	60.14 ± 3.36	59.47 ± 2.86	60.52 ± 1.52	**64.00** ± 1.75
64	2	64.31 ± 1.23	69.53 ± 1.89	69.92 ± 1.53	66.08 ± 1.20	**81.90** ± 1.58
64	3	62.14 ± 2.38	66.58 ± 2.72	67.10 ± 1.05	63.98 ± 1.12	**68.93** ± 1.72
64	4	56.54 ± 4.41	65.33 ± 1.89	65.89 ± 1.78	63.75 ± 1.17	**67.76** ± 1.39
64	5	50.11 ± 6.26	63.88 ± 1.76	64.63 ± 1.97	63.10 ± 1.37	**65.75** ± 2.03
128	2	66.18 ± 1.47	75.99 ± 2.16	75.33 ± 2.09	67.97 ± 1.48	**84.25** ± 0.63
128	3	65.48 ± 1.24	69.01 ± 1.27	69.25 ± 1.61	66.02 ± 0.97	**71.89** ± 1.72
128	4	60.01 ± 4.08	67.89 ± 1.07	68.39 ± 1.55	64.78 ± 1.09	**68.58** ± 1.38
128	5	56.39 ± 4.03	66.54 ± 1.72	66.86 ± 1.35	65.52 ± 1.02	**66.94** ± 1.47
ECG (6 classes, chance 16.67%)						
32	2	69.36 ± 1.02	68.88 ± 0.91	69.04 ± 0.83	67.20 ± 0.79	**69.69** ± 0.97
32	3	72.78 ± 0.86	73.70 ± 0.75	**73.74** ± 0.93	70.68 ± 0.97	73.59 ± 0.94
32	4	72.78 ± 1.67	75.57 ± 0.96	75.47 ± 0.84	71.86 ± 0.64	**75.90** ± 0.60
32	5	70.27 ± 1.83	76.01 ± 0.69	75.62 ± 0.79	72.62 ± 0.76	**76.67** ± 0.64
64	2	73.91 ± 0.61	**74.26** ± 0.39	74.21 ± 0.50	71.76 ± 0.32	73.73 ± 0.40
64	3	77.63 ± 0.46	**78.07** ± 0.42	78.01 ± 0.37	75.84 ± 0.30	77.48 ± 0.41
64	4	79.22 ± 0.51	**79.89** ± 0.46	79.67 ± 0.36	77.77 ± 0.39	79.62 ± 0.37
64	5	78.70 ± 0.85	80.40 ± 0.30	80.14 ± 0.41	78.70 ± 0.41	**80.60** ± 0.37
128	2	77.09 ± 0.20	**77.57** ± 0.19	77.53 ± 0.29	75.05 ± 0.27	75.97 ± 0.27
128	3	80.72 ± 0.26	**80.98** ± 0.31	80.82 ± 0.27	79.31 ± 0.22	80.33 ± 0.29
128	4	82.35 ± 0.36	**82.53** ± 0.27	82.47 ± 0.39	81.43 ± 0.35	82.12 ± 0.33
128	5	82.20 ± 0.69	82.89 ± 0.40	82.87 ± 0.33	82.49 ± 0.26	**82.91** ± 0.34

Table 25: **Reinforcement-learning cell-level results.** Per-cell raw episode reward (not normalized by the per-environment ceiling) on Inverted Pendulum, Inverted Double Pendulum, and Hopper, broken out by width, depth, and initialization configuration. Cell values are mean ± standard deviation across 20 seeds. Metrics presented are those computed across the last 10% of evaluation data.

width	depth	naive	I_t default	$I_t^{\perp}$ enhanced	$U_t \sim \mathcal{N}(0,1)$	$U_t \sim \mathcal{N}(\theta,1)$
Inverted Pendulum (maximum episode reward 1000)						
32	2	999.60 ± 1.19	992.48 ± 31.47	999.37 ± 2.02	**999.97** ± 0.09	999.10 ± 4.03
32	3	895.30 ± 239.52	781.36 ± 234.59	736.19 ± 283.22	**997.63** ± 3.80	991.51 ± 17.53
32	4	42.54 ± 98.94	91.99 ± 35.94	140.98 ± 130.23	135.08 ± 96.19	**732.33** ± 259.47
32	5	24.75 ± 10.36	60.26 ± 24.92	64.41 ± 20.79	40.12 ± 20.92	**189.70** ± 158.93
64	2	986.04 ± 36.78	987.08 ± 38.33	985.05 ± 66.02	**999.69** ± 1.19	996.23 ± 15.97
64	3	866.90 ± 248.83	988.83 ± 31.23	979.40 ± 46.08	**1000.00** ± 0.00	980.22 ± 80.56
64	4	98.02 ± 144.67	775.04 ± 218.12	802.98 ± 169.37	855.31 ± 153.04	**971.97** ± 53.94
64	5	33.68 ± 32.09	125.81 ± 22.59	132.64 ± 22.15	85.48 ± 41.92	**674.77** ± 150.17
128	2	982.70 ± 76.92	999.98 ± 0.07	**1000.00** ± 0.00	999.99 ± 0.06	997.83 ± 7.57
128	3	977.08 ± 43.84	983.90 ± 64.52	995.09 ± 16.91	**1000.00** ± 0.00	995.18 ± 9.81
128	4	891.64 ± 215.09	958.43 ± 75.98	992.28 ± 29.27	**994.23** ± 15.73	970.69 ± 101.91
128	5	37.96 ± 24.10	553.70 ± 198.17	545.31 ± 200.80	220.67 ± 114.22	**871.07** ± 77.83
Inverted Double Pendulum (maximum episode reward 10000)						
32	2	2229.96 ± 2992.28	810.91 ± 739.31	630.79 ± 65.99	592.82 ± 57.01	**9080.15** ± 1030.30
32	3	594.79 ± 56.18	538.69 ± 58.79	535.34 ± 45.50	555.07 ± 46.91	**2781.75** ± 3398.95
32	4	260.87 ± 113.65	**540.24** ± 117.40	535.18 ± 95.94	446.16 ± 71.92	445.17 ± 50.80
32	5	261.11 ± 29.14	278.06 ± 137.84	350.86 ± 203.05	200.52 ± 18.71	**415.34** ± 59.27
64	2	9347.25 ± 13.97	9353.78 ± 1.39	9338.73 ± 45.64	**9354.75** ± 1.15	9299.70 ± 234.72
64	3	4127.22 ± 4327.05	7713.46 ± 3110.34	7649.45 ± 3096.68	8336.29 ± 1796.79	**9189.98** ± 722.80
64	4	528.05 ± 33.83	523.31 ± 34.71	512.56 ± 56.26	562.91 ± 56.29	**9314.66** ± 76.48
64	5	285.37 ± 25.26	447.94 ± 57.06	457.53 ± 46.35	469.78 ± 47.80	**3788.65** ± 3282.19
128	2	9248.16 ± 217.75	9315.28 ± 103.88	**9349.28** ± 19.84	9265.91 ± 208.64	9283.03 ± 228.71
128	3	9350.75 ± 14.51	**9356.78** ± 3.10	9353.82 ± 11.45	9346.42 ± 37.51	9339.46 ± 58.53
128	4	2967.09 ± 3810.20	7937.19 ± 2399.75	7767.91 ± 1761.03	9119.17 ± 719.72	**9189.37** ± 294.76
128	5	405.02 ± 173.03	548.39 ± 45.60	546.71 ± 34.68	1331.24 ± 225.57	**8176.07** ± 1457.27
Hopper (reference episode reward 1000)						
32	2	604.40 ± 267.39	537.64 ± 183.92	620.93 ± 282.81	593.93 ± 173.12	**722.20** ± 318.80
32	3	559.13 ± 208.72	585.53 ± 170.11	581.46 ± 183.93	540.26 ± 139.57	**638.91** ± 132.67
32	4	569.41 ± 355.11	412.69 ± 169.62	451.74 ± 108.19	443.74 ± 161.74	**581.74** ± 279.94
32	5	187.46 ± 5.22	307.04 ± 91.10	288.65 ± 88.61	**329.95** ± 167.95	277.06 ± 82.67
64	2	542.85 ± 153.32	580.64 ± 131.47	512.22 ± 148.30	510.00 ± 162.12	**685.25** ± 101.92
64	3	550.51 ± 138.15	521.49 ± 122.87	498.11 ± 118.57	460.49 ± 130.80	**551.72** ± 64.30
64	4	472.94 ± 111.93	522.02 ± 104.01	485.36 ± 85.50	463.57 ± 98.90	**568.81** ± 89.23
64	5	399.17 ± 310.67	460.01 ± 99.40	483.39 ± 94.55	436.66 ± 102.96	**571.81** ± 86.65
128	2	**751.61** ± 136.21	699.01 ± 108.09	717.53 ± 93.86	705.04 ± 169.29	707.74 ± 182.78
128	3	**714.50** ± 171.91	661.35 ± 167.08	660.04 ± 155.08	627.38 ± 96.90	654.19 ± 134.06
128	4	**826.50** ± 155.90	538.16 ± 76.01	621.25 ± 106.03	619.36 ± 129.88	710.16 ± 128.16
128	5	470.77 ± 84.82	543.48 ± 66.81	603.29 ± 99.48	569.80 ± 98.88	**717.94** ± 114.57

C.4.4 Empirical evaluation: convolutional SNN on DVS Gesture

The dense SNNs of Section 3.2 receive each input as a flattened vector. To extend the evaluation to computer vision, we train a convolutional SNN on DVS Gesture [Amir et al., 2017], the event-camera dataset among the neuromorphic tasks, whose frames retain the spatial structure of image data.

Setup. The protocol of Section C.2 for DVS Gesture (Table 9) and the five configurations of Section 3.2 (Figure 2), with the dense layers replaced by four convolutional LIF layers of 32 kernels of size 3×3, stride one and size-preserving zero padding, initialized through the tensor view of Section B.7.2. The first convolution takes the place of the linear projection before the batch-normalization module, the leak β^l is learnable per channel, and a dense layer maps the flattened $32 \times 32 \times 32$ output onto the 11 output neurons; neuron, readout and optimization settings otherwise follow Section C.2 (Table 8). Frames are binned as in Section C.2, after denoising (10 ms) and downsampling by four to $32 \times 32 \times 2$, and post-padded to $T = 369$ steps, with the padded steps masked from the readout. Each configuration is trained for 75 epochs of 24 minibatches. This setup is intended to explore the applicability of the proposed methodology to convolutional SNNs in empirical scenarios, not as a comprehensive study in its own right.

Table 26: **Convolutional SNN on DVS Gesture.** Test-set accuracy of the convolutional SNN on DVS Gesture, broken out by initialization configuration. Cell values are mean $\pm$ standard deviation across seeds. Metrics presented are those computed across the last four evaluation points (last 20 of 75 epochs).

	(1) naive	(2) I_t default	(3) $I_t^{\perp}$ enhanced	(4) $U_t \sim \mathcal{N}(0,1)$	(5) $U_t \sim \mathcal{N}(\theta,1)$
Accuracy (%)	83.36 ± 6.18	89.55 ± 0.97	89.08 ± 1.29	89.27 ± 1.47	89.55 ± 0.72

Results. The analytically motivated configurations (2)–(5) exceed the naive configuration (1) by 5.72–6.19 pp, with a standard deviation across seeds of 0.72–1.47 pp against 6.18 pp (Table 26); their means lie within 0.47 pp of one another, inside that spread, and we do not rank them. At initialization, our closed form also holds the target spike rate per neuron in convolutional layers (Section C.3.7), and configurations (4) and (5) reach a comparable accuracy on this task.

Scope. While this experiment demonstrates that the proposed initialization extends to convolutional SNNs on event-based vision data, its suitability for this modality remains an open question. The closed form sets the spike rate of each neuron independently and at neuron-level precision, under the premise that neurons and their inputs are approximately uncorrelated (Equation 4). This premise transfers naturally across channels, which act as distinct feature detectors, but is questionable along the spatial dimensions within a channel: through local receptive fields and weight sharing, neighboring positions observe overlapping and correlated input patches, a spatial correlation structure that the inductive bias of convolutions is designed to exploit and that the present formulation does not prioritize. Identifying the structural priors that an initialization should encode along these spatial dimensions for convolutional architectures is active research [Pedersen et al., 2025] and therefore a natural next step beyond this work.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims of the abstract and Section 1 are supported by the derivation in Section 2, with its complete form and extensions in Appendix B; by the simulations in Section 3.1 (Table 2), extended in Section C.3; and by the empirical evaluation in Section 3.2 (Table 3), extended in Section C.4. Their scope is stated in Section 3.3.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The *Limitations* paragraph of Section 3.3 states the scope of the evaluation and what the closed form does not cover, and the same section identifies the depth-related performance drop as an open question. Robustness to violations of the input assumption is tested in Section C.3.3, and the Gaussian approximation is validated in Section B.3.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Assumptions (A1)–(A6) and the closed form are stated in Section 2.3 (Equation 4, Equation 8). The fixed-point closure and its consistency proposition are in Section B.1, the complete derivation in Section B.2, and the validation of the Gaussian approximation in Section B.3.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The initialization is fully specified by Equation 8 with Equation 7 and Equation 5. The simulation setup is given in Section C.1 and the empirical setup in Section C.2, the alternative initializations in Section A.1. All datasets and environments are public and cited in Section C.2.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code to reproduce the empirical results of Section 3.2 and their simulation is released at `https://github.com/tudelft/snn-init-lib`. All datasets and environments are public and cited in Section C.2.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The experimental setting of the simulation study is described in Section 3.1, with full details in Section C.1. The experimental setting of the empirical evaluation is described in Section 3.2, with full details, including the optimizer, its hyperparameters and the data splits, in Section C.2.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: The simulation results are reported in Section 3.1 and Section C.3: their tables report means over 10 seeds with $\pm$ one standard deviation across the neurons of a layer, and Figure 1 shows the individual neurons of one seed, as defined in the captions and in Section C.1. The empirical results are reported in Section 3.2 and Section C.4: Table 3 and the per-cell tables report means over 20 seeds with standard deviations across seeds, as defined in their captions. The summary rows of Table 3 and the bars of Figure 3 add 95% bootstrap confidence intervals over the 20 seeds. Table 3 also gives paired permutation tests on score and rank. Section C.2 states their method and assumptions.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: The computational effort of the empirical evaluation comprises its 1,140 cells, the combinations of task, configuration, width and depth defined in Section 3.2 and Section C.4, each trained with 20 seeds; the hardware, frameworks and wall-clock time per paradigm are stated in Section C.2.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: The work develops a method for initializing spiking neural networks, uses only public benchmark datasets and environments (Section C.2), and involves no human subjects, crowdsourcing, or scraped or sensitive data.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [N/A]

Justification: The work develops a method for initializing spiking neural networks and is not tied to an application; we identify no direct path to negative societal impact. Its intended positive impact is more reliable training of energy-efficient neuromorphic models (Section 1).

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper concerns an initialization routine for spiking neural networks. It is not a pre-trained generative model, includes no scraped data, and the associated misuse risk is negligible.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The datasets and environments are credited with their original publications in Section C.2; all are publicly released for research use and used under their terms.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The new asset of this work is the code that implements the proposed initialization and reproduces the empirical experiments of Section 3.2 and their simulation, released at `https://github.com/tudelft/snn-init-lib` with documentation for its installation and use.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: No crowdsourcing or human-subjects research was conducted.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper involves no human-subjects research and therefore requires no IRB or equivalent approval.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs are not a component of the core methodology of this research.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.