Metrics

ar005 · 14 May 2026

This page is the reference for every number and every panel the scope shows. It is written to stand on its own: if this is the first page you read, you should come away understanding what the scope is showing, what each metric measures, and what a "healthy" value looks like.

model

What it is: The model variant driving the run. These sit on a ladder from simplest (snnTorch, a standard feedforward LIF network) to most biophysical (PING, with excitatory and inhibitory populations coupled into a gamma-producing loop). See COBANet for the conductance-based model.

How it's calculated: Read from the *–model* CLI flag. A spinner character prefixes the name on live runs so the eye can tell a frozen frame from one that's still updating.

Common values: snnTorch, *CUBA*, *COBA*, *PING*.

dt

What it is: The integration timestep in ms — how much real time elapses per simulation step. Smaller Δt is more accurate but more expensive (more steps per trial).

How it's calculated: Read from *–dt*. Held fixed across a training run; can be varied at inference to probe Δt-stability.

Common values: 0.1 ms (fine), 0.25 ms (default), 1.0 ms (coarse). PING breaks down above about 1 ms because the excitatory-inhibitory feedback loop can't complete within a single step.

N

What it is: Number of excitatory neurons in the hidden layer. For PING, the inhibitory population is sized at $N_E/4$ (the classical 4:1 cortical ratio).

How it's calculated: Read from *–n-hidden*.

Common values: 256, 512, 1024.

in

What it is: The peak firing rate of the input neurons, in Hz. An MNIST pixel with intensity 1.0 fires at this rate; a dark pixel fires at zero. Controls how strongly the network is driven.

How it's calculated: Each input neuron emits a Bernoulli spike at each step with probability

$$p = r \cdot \Delta t/1000$$

where r is this per-pixel rate (Hz) scaled by pixel intensity. See Training § Input encoding for the full scheme.

Common values: 50 Hz for PING, 100 Hz for simpler models. - in the header when not set.

E

What it is: Firing rate of the excitatory population, averaged over all E neurons and over the trial, in Hz. The single most important live readout of network activity.

How it's calculated: Let $s_E \in \{0,1\}^{T \times N_E}$ be the binary E spike array (T timesteps, N_E neurons; entry is 1 if neuron i spiked at step t), and $T_{\text{sec}} = T\Delta t/1000$ the trial length in seconds. Then

$$r_E = \frac{\sum_{t,i} s_{E[t,i]}}{N_E \cdot T_{\text{sec}}}$$

Common values: 1–30 Hz is sparse-coding territory (healthy for snnTorch when classifying MNIST); 40–80 Hz is the gamma-locked regime (PING target); over $\sim$ 150 Hz means the network has saturated; 0 Hz means it's silent.

I

What it is: Firing rate of the inhibitory population, same definition as *E*. Only meaningful for models that have an I population (COBA, PING); shows - otherwise.

How it's calculated: Same as *E*, on the I spike array s_I and with N_I inhibitory neurons:

$$r_I = \frac{\sum_{t,i} s_{I[t,i]}}{N_I \cdot T_{\text{sec}}}$$

Common values: Typically 2–4× the E rate under balanced PING operation — roughly 80–200 Hz.

f_0

What it is: The dominant oscillation frequency of the E population, in Hz. If the network is gamma-locked the rate signal has a clear peak in its Fourier spectrum; f_0 is the frequency of that peak.

How it's calculated: The E spike raster is binned at 2 ms, the resulting time series $r_E(t)$ is mean-subtracted and FFTed. A peak search runs in the 5–80 Hz band and requires the peak to be at least 3× the band median (an SNR gate). A subharmonic check follows: if there is clear power at half the peak frequency ($\geq$ 30 % of the peak's power), f_0 is set to that lower frequency instead — this catches the case where pyramidal neurons fire every second cycle.

Common values: 0 Hz (non-oscillating; always for non-PING models, which short-circuit the calculation); 30–80 Hz (gamma, the PING target); anything below 30 Hz is sub-gamma beta. Shows - in the header when 0.

CV

What it is: How bursty the E population's firing is. Takes the total E spike count in each 2 ms bin and computes the coefficient of variation (std / mean) of those counts across bins.

How it's calculated: Bin s_E at 2 ms; let n_b be the total E spike count in bin b. Then

$$\text{CV} = \frac{\sigma(n_b)}{\max(\mu(n_b), 10^{-9})}$$

Common values: Around 1 for Poisson-like asynchronous firing; well above 1 (often $\gtrsim 2$) for gamma-locked PING, where spikes cluster into cycle peaks and leave the troughs empty; below ~ 0.3 for a clock-like or saturated network.

I/E

What it is: The ratio of inhibitory to excitatory population rates. A single number that summarises E–I balance.

How it's calculated:

$$\mathrm{I/E} = \frac{r_I}{\max(r_E, 10^{-9})}$$

Common values: 2–4 under balanced PING operation. Much higher means inhibition has run away; much lower means inhibition is failing or E has saturated.

act

What it is: The fraction of excitatory neurons that spiked at least once during the trial. Complements the population rate: a high E could come from a few neurons firing a lot, or many neurons firing a little — *act* tells you which.

How it's calculated:

$$\mathrm{act} = \frac{\#\left\{i : \sum_t s_{E[t,i]} > 0\right\}}{N_E}$$

Common values: 1–3 % is healthy sparse coding (a classical snnTorch regime). 80 %+ with CV near 0.4 is healthy gamma-locked PING. Persistent values below 1 % or above 95 % alongside stalled accuracy are pathological.

e_raster

What it is: The excitatory spike raster over the trial. Each dot is one spike; the y-axis is E-neuron index, the x-axis is time. This is the primary dynamical read and occupies the largest cell in the grid.

How to read it: Vertical bands (near-synchronous firing across the population, at regular intervals) mean gamma-locked PING. A uniformly-speckled cloud means asynchronous firing — normal for CUBA or snnTorch. An empty raster means the network is silent; a solid block means it is saturated.

i_raster

What it is: Same idea as *e_raster* but for the inhibitory population. Empty for models without an I population.

How to read it: Under balanced PING the I raster is denser and at higher rate than the E raster, matching the 2–4× I/E ratio.

drive

What it is: The mean input current (conductance) reaching the membrane, overlaid with the voltage thresholds a neuron needs to reach to spike.

How it's calculated: The input conductance is passed through an exponential synapse with time constant τ_{AMPA} to get the steady-state drive

$$g_{\text{ss}} = \frac{\bar{g}}{1 - e^{-\Delta t/\tau_{\text{AMPA}}}}$$

The bare spike threshold — the drive a neuron needs in the absence of inhibition — is

$$g_{\text{thresh}} = \frac{g_L(V_{\text{th}} - E_L)}{E_e - V_{\text{th}}}$$

The effective threshold adds an inhibition-proportional penalty $g_i \cdot |E_i - V_{\text{th}}| \,/(E_e - V_{\text{th}})$.

How to read it: The shaded region is where drive exceeds the effective threshold — that's when E neurons can fire. If drive never crosses threshold the network is dead by construction; if drive is always above, it is saturated.

weights

What it is: Histograms of the current weight values, one sub-histogram per weight matrix (input-to-hidden, hidden-to-output, and any recurrent E–E / E–I / I–E matrices).

How to read it: Signed weights (standard SNN) look roughly Gaussian around zero. Dale's-law weights are non-negative so they form a half-normal clamped at zero. A long tail means a few synapses dominate; a narrow stack at zero means many weights have died under regularisation.

output

What it is: The readout logits for each output class, over the course of the trial.

How it's calculated: The readout accumulates hidden spikes across time and projects them through a trained linear layer:

$$\hat{y}_t = \left(\sum_{s \le t} s_s^{\text{hid}} \right) W_{\text{out}} + b_{\text{out}}$$

The argmax at the final step is the model's prediction. See Training § Readout.

How to read it: For MNIST there are ten traces; if the network has learned, one trace pulls away from the pack as the trial progresses. If all traces stay flat, or one dominates from t=0, the run is pathological.

psd

What it is: The power spectral density (frequency content) of the E population rate $r_E(t)$. The gamma band (30–80 Hz) is shaded.

How it's calculated: $r_E(t)$ is computed in 2 ms bins over the stimulus window, mean-centred, FFTed, and normalised:

$$\text{PSD}(f) = \frac{|\text{FFT}(r_E - \bar{r}_E)(f)|^2}{\max_f |\text{FFT}(r_E - \bar{r}_E)(f)|^2}$$

How to read it: A flat PSD means asynchronous firing. A peak inside the shaded band means gamma. A peak below 30 Hz means sub-gamma (beta). A peak above 80 Hz is rejected by the f_0 search.

participation

What it is: The distribution of per-neuron spike counts across the E population for this trial.

How to read it: A stack at zero indicates dead neurons; a stack at the per-neuron ceiling indicates saturated neurons. Both can be invisible from the population-level E token, which averages across the group.

acc_curve

What it is: Test accuracy, training loss, and learning rate as a function of epoch — the standard training-progress panel. All three are normalised to their running maximum and plotted on a single 0–100 % axis. Rendered only during training runs.

How to read it: A healthy run has accuracy climbing from ~ 10 % (chance on MNIST) toward 85–98 %, and loss dropping monotonically. A flat accuracy with drifting loss usually means the readout can't separate classes.

rate_curve

What it is: The E and I firing rates as a function of training epoch. Training-only.

How to read it: Stable rates within the healthy regime for the model (see *act* above) means the network is holding its operating point as it learns. Monotonic drift toward 0 or toward a ceiling usually means gradient pressure is pushing the network out of its regime.

grad_flow

What it is: For each weight matrix, the ratio of gradient norm to weight norm — how hard each layer is being updated relative to its current size. Training-only; log y-axis.

How to read it: The shaded band from 10^{-3} to 10^{-2} is the healthy window. Values below mean gradients are vanishing (that layer is not learning); above means they are exploding.

digit_image

What it is: The MNIST digit currently being presented to the network. A small greyscale inset that lets you sync the readout in *output* to the actual input class.

sweep

What it is: A generic progress indicator for parameter-sweep runs. Shows the full sweep range faintly and the completed prefix solidly, with a dot on the current value.

sweep_rates

What it is: E and I firing rates accumulated across an inference sweep — one point per sweep frame, updated as the sweep progresses. Sweep-only.

How to read it: If the trace is flat across the swept variable, the network is stable under that variable. If it fans out by one to two orders of magnitude, it is not — this is the signature read off in a Δt-stability experiment.

sweep_f0

What it is: Peak oscillation frequency f_0 across the sweep, with the gamma band (30–80 Hz) shaded. Sweep-only.

How to read it: A PING run with a robust E→I→E loop holds f_0 inside the shaded band across the whole sweep. Drift out of band indicates the loop is failing at that parameter value.

Spectral radius

What it is: The largest-magnitude eigenvalue of the recurrent weight matrix J. A linear stability indicator: if $\rho(J) > 1$, the linearised network would explode; $\rho(J) \approx 1$ sits at the edge of chaos.

How it's calculated: The recurrent matrix is reassembled post-training with E columns kept positive and I columns flipped to negative (so Dale's law appears as signs), then $\rho(J) = \max_i |\lambda_i|$ via an eigendecomposition.

Common values: $\rho < 1$ is quiescent (no sustained oscillation); $\rho \approx 1$ is the gamma operating regime; $\rho > 1$ is linearly unstable but the spiking nonlinearity can still stabilise it.

Linear f_0 prediction

What it is: A continuous-rate prediction for the oscillation frequency, derived from the eigenvalue of J with the largest imaginary part.

How it's calculated: Pick $\lambda^* = a + bi$ with largest $|b|$, then

$$f_0^{\text{lin}} = \frac{|b|}{2\pi} \cdot 1000 \text{ Hz}$$

Common values: Typically an upper bound on the measured f_0 — the linear mode ignores the spiking nonlinearity and the refractory floor.

Refractory-floor prediction

What it is: A lower bound on the oscillation frequency set by the biology: no spiking network can cycle faster than one excitatory refractory period plus one GABA-synapse decay.

How it's calculated:

$$f_0^{\text{cor}} = \frac{1000}{\tau_{\text{ref}}^E + \tau_{\text{GABA}}} \text{ Hz}$$

Common values: The measured f_0 typically lies between f_0^{cor} and f_0^{lin}.

loss

What it is: The training-set cross-entropy at the end of each epoch — the scalar the optimiser is minimising.

Common values: Starts near $\ln(10) \approx 2.3$ for chance-level classification on MNIST and drops monotonically under healthy training. A flat or climbing loss means the gradient isn't reaching the readout.

test_loss

What it is: Cross-entropy on the held-out test set, evaluated once per epoch.

Common values: Tracks *loss* with a small gap. A widening gap usually means overfitting; a gap that collapses to zero often means data leakage between train and test.

lr

What it is: The current learning rate — the multiplier on each gradient step. Varies if *adaptive-lr* is set; constant otherwise.

Common values: 10^{-2} for snnTorch / CUBA; 10^{-4} for COBA / PING, whose conductance-based gradients are much steeper (see Gradient Stabilisation).

grad_norm

What it is: The mean gradient norm across all parameters, taken after gradient clipping.

Common values: Stable to within an order of magnitude across a run. A sudden jump usually precedes an NaN loss.

grad_ratios

What it is: A dictionary mapping each weight matrix to its $\|\nabla\| / \|W\|$ — the quantity plotted by *grad_flow*.

Common values: Healthy band 10^{-3} to 10^{-2} per layer; below is vanishing, above is exploding.

acc

What it is: Test-set accuracy (%) at the end of each epoch — the canonical "how well is the network classifying" number.

Common values: 10 % is chance on MNIST; 85–90 % is typical at a small training tier; 95–98 % at medium or larger; anything above 98 % flags a leaked test split or a bug.

best_acc

What it is: The maximum of *acc* over all epochs seen so far. Tracked alongside *best_epoch* (the epoch that set it) and a *new_best* flag (true on the epoch that set it).

Common values: This is the headline number cited in notebook entries. If *best_epoch* is near the final epoch the run was still improving; if much earlier, it had plateaued.

On inference runs, *best_acc* is just the single-shot test accuracy; the raw count *n_correct* is saved alongside, and per-sample calls go to *test_predictions.json* (one row per sample, carrying *idx*, *true*, *pred*, *correct*, *logits* for confusion-matrix and calibration downstream).