Accuracy converges, firing rate does not

exp024 · 2 June 2026

The trained networks this entry uses are produced once in the shared training hub, exp022 (Training), and reused here rather than retrained.

Abstract

Reads exp022's PING and COBA θ_u = off baselines (three seeds each, 50 epochs) and asks whether the firing rate converges once the accuracy has. It does not, not for COBA. Test accuracy plateaus by ≈ 15 epochs in both architectures, but PING's E rate locks to a tight ≈ 10 Hz attractor (cross-seed to 0.1 Hz) while COBA's keeps climbing through epoch 50. Accuracy is a point; for COBA, the rate is a manifold the optimiser drifts along at constant accuracy.

Method

Reads the shared θ_u = off baselines (coba and ping, three seeds each) from the training hub, exp022 (Training), which fixes the recipe (50 epochs, mem-mean readout, no rate regulariser), and plots their per-epoch training history. The question, following exp041 / exp044: once test accuracy has plateaued, does the firing rate also settle? **Converged** means a last-10-epoch slope below 0.1 pp/ep (accuracy) or 0.05 Hz/ep (rate).

Results

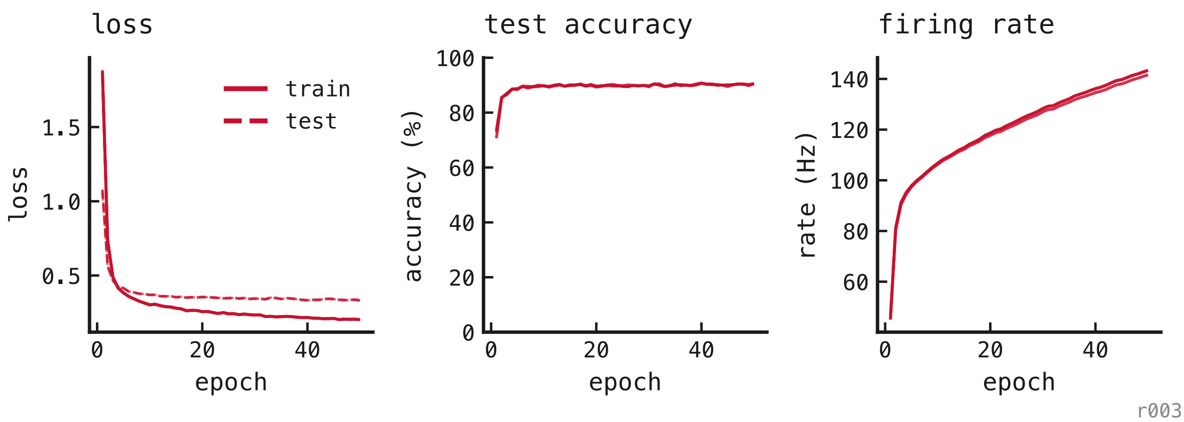

Figure 1: COBA, three seeds. Loss (train solid, test dashed) and **accuracy plateau by ≈ 15 epochs**, but the **E rate keeps climbing to ≈ 143 Hz and is still rising at epoch 50**. Accuracy has converged, the rate has not.

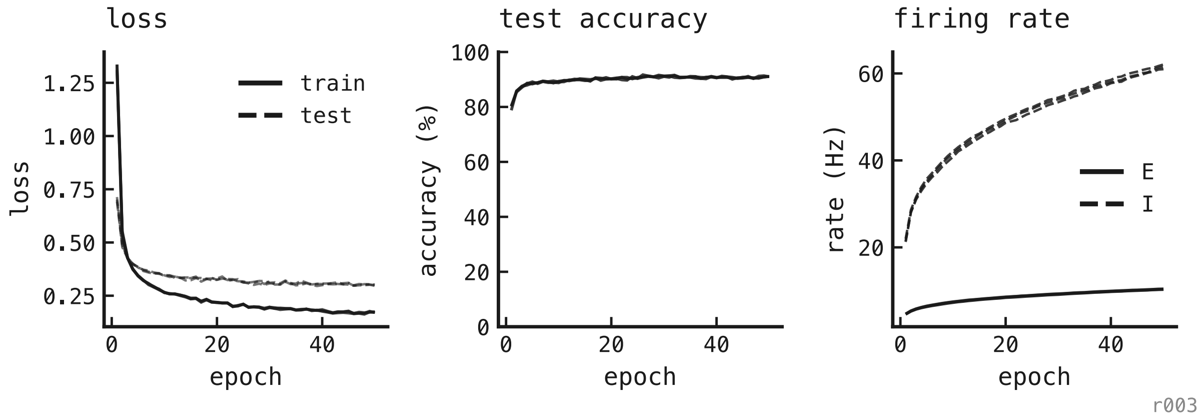

Figure 2: PING, three seeds. Loss and accuracy plateau by ≈ 15 epochs as in COBA, but here the **E rate settles into a tight band near ≈ 10 Hz**, converged. (The I rate, dashed, keeps climbing to ≈ 61 Hz, driven by the still-growing input weights, the same force that drives COBA's E rate up.) The loop pins the excitatory rate; without it, COBA's drifts.

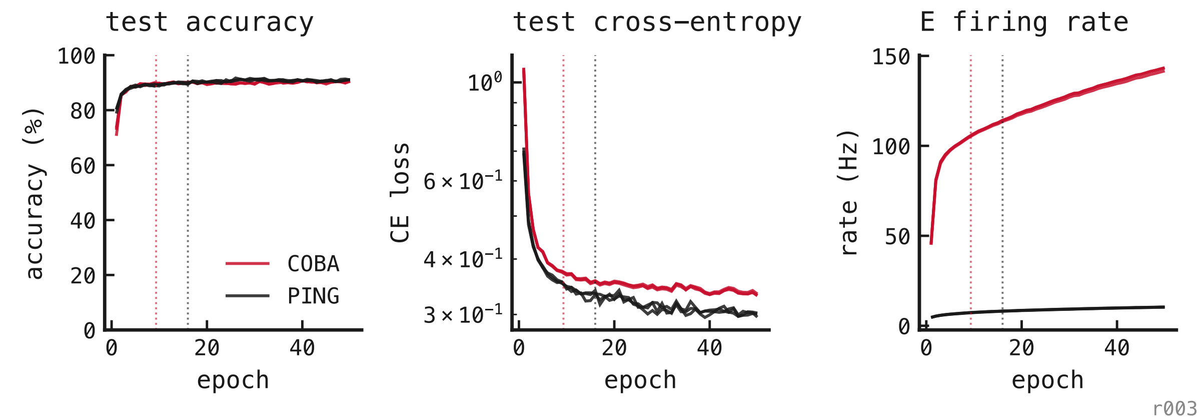

Figure 3: Test accuracy (left), test cross-entropy on a log axis (middle), and E firing rate (right) vs epoch; three seeds each, COBA red / PING black. Dotted verticals mark each model's accuracy-convergence epoch (first epoch within 1% of final accuracy). Accuracy is flat past ≈ epoch 20, yet **cross-entropy keeps falling well to its right**: COBA's is still dropping at epoch 50. COBA's E rate climbs ≈ 45 → 143 Hz in lockstep; PING reaches the same low loss early and **stays pinned near ≈ 10 Hz**.

Discussion

The rate climb is what the loss spends to keep gaining confidence. Cross-entropy stops at *certain*, not *correct*:

$$\mathrm{CE} = -\log p_y = \log\left(1 + \sum_{k \neq y} e^{z_k - z_y}\right)$$

- z_y, z_k: logits for the true class and class k
- p_y: softmax probability of the true class
- $m = z_y - \max_{k \neq y} z_k$: decision margin

Accuracy needs only the *sign* of m; cross-entropy keeps shrinking with the *size* of the gaps. So past the convergence line the argmax is fixed but the loss still falls by widening margins, which means scaling logits up. With a mem-mean readout, $z \approx W_{\text{out}} \cdot (\text{E activity})$, so wider margins cost either weight or spikes. COBA takes the spike route (rate ≈ 45 → 143 Hz, still climbing); PING sharpens its readout through the loop and holds ≈ 10 Hz. The loop buys confidence for free; without it, confidence costs spikes.

So the rate doesn't fail to converge: it tracks a loss that never stops rewarding margin. Each cell's per-epoch metrics already record *test_margin*, *test_confidence* and *test_logit_scale*, so the prediction that COBA's margin rises with its rate while PING's plateaus can be read straight from them.

Next steps

Figure 3 infers confidence from the rate. The margin, confidence and logit scale that exp022 already logs per epoch let us measure it directly:

- Plot per-epoch **margin** $\langle \boldsymbol{m} \rangle$, **confidence** $\langle \boldsymbol{p_y} \rangle$ and **logit scale** vs epoch, COBA vs PING: the direct version of Figure 3′s middle panel.
- Overlay each against the E rate to test the lockstep: confidence up with rate for COBA, both flat for PING.
- Confirm the split quantitatively: does COBA's margin keep rising at epoch 50 (slope > 0) while PING's plateaus (slope ≈ 0), matching the cross-entropy and rate slopes here.