

Trying a cumulative-potential readout on matched SHD networks

exp071 · 19 July 2026 · Draft

Abstract

This one-seed exploratory experiment tests whether the newly merged `tools/snn` cumulative-potential decoder can move matched Dale-constrained COBA and PING networks above the exp068–exp070 validation plateau on SHD. The screening ladder uses the same deterministic development-training / held-out-validation split as exp069 and exp070; the official SHD test remains sealed. Candidate one keeps the baseline 256-cell architecture and changes the classifier to `--readout cumulative-potential --signed-readout`

`--readout-bias`. If that is finite, active, and promising, candidate two keeps the same decoder and tests two 256-cell hidden layers. Only the best short candidate may be promoted to forty epochs.

Goal prompt

```
/goal Design and execute the next exploratory SHD experiment on branch night/spiking-heidelberg-digits/ar071, using the newly merged tools/snn cumulative-potential readout additions to try to raise validation accuracy while keeping the comparison between matched Dale-constrained COBA and PING networks scientifically clean.
```

```
Start from updated main at merge commit
9527162d7a0ef80deac0e3f606ce513b9280ba6c. Do not modify main. Open a new
draft PR for this experiment when the first meaningful commit is ready.
```

Scientific objective:

Run a one-seed exploratory validation-only ladder on SHD to identify whether stronger temporal readout and modest capacity changes materially improve matched COBA/PING learning beyond exp068-exp070.

Constraints:

- Use one seed only.
- Do not use the official SHD test set during exploratory screening.
- Use a held-out validation split from the training set for all selection decisions.
- Keep COBA and PING matched except for their registered cell-specific voltage-gradient dampening.
- Preserve matched input, readout, training split, optimizer, batch, and preprocessing settings across COBA and PING within each candidate.
- Use short runs first for iteration speed, targeting about 8-10 epochs per

candidate.

- Promote only the most promising candidate to a 40-epoch validation run.
- Do not run multiple seeds unless a later claim is worth defending and I explicitly authorize it.
- RunPod spending requires explicit approval before pod creation; use at most one pod at a time unless separately authorized.
- Reap any pod when finished.
- Do not merge the PR.
- Avoid the full local test suite on the 4 GB Hetzner host; use focused checks and demolab build.

Candidate ladder:

1. Baseline architecture with the new signed cumulative-potential readout:
--readout cumulative-potential --signed-readout --readout-bias
2. If candidate 1 is finite, active, and improves validation learning, try modest increased capacity with two hidden layers, e.g. --n-hidden 256 256, keeping the same readout.
3. Optionally test one conservative recurrence/readout-adjacent setting if supported by existing tools/snn CLI and scientifically justified before seeing final results.
4. Promote the best candidate, if any, to a matched 40-epoch COBA/PING validation run.

Deliverables:

- Create the next numbered experiment, likely exp071, with artifacts under artifacts/data/exp071.
- Produce numbers.json, provenance, reproducer, training curves, firing-rate diagnostics, and matched input/E/I rasters where relevant.
- Write writings/exp071.typ as the canonical cold-readable experiment report.
- Include the goal prompt below the abstract and append timestamped activity-log/thread checkpoints in the experiment appendix, following the simplified SHD organization.
- Record scientific decisions, failures, anomalies, costs, commits, results, and pending work in the experiment article.
- Sanitize any publishable transcript/log content before committing.
- Build and validate with focused checks plus demolab build.
- Commit meaningful attempts, including killed attempts, with clear messages.
- Push branch night/spiking-heidelberg-digits/ar071 and update the new PR with evidence.
- Finish at the human review gate with a concise evidence summary, exact compute spend, validation performed, and links to the rendered exp071 file and PR.

Methods

Locked comparison

Both cells use seed 42, the exp069/exp070 development split (7340 training utterances and 816 validation utterances), batch size 32, Adam learning rate 0.0004, 1 ms simulation steps, 1000 ms utterance windows, fixed Dale-constrained feed-forward/recurrent conductances, and

identical input preprocessing. COBA disables the inhibitory loop and uses the engine’s no-dampening value; PING enables the inhibitory loop and uses voltage-gradient dampening 1000. These cell-specific settings are the only intended COBA/PING difference inside each candidate.

The readout is the new cumulative-potential decoder with signed abstract classifier weights and a trainable readout bias. The readout initialisation scale is one, not the old membrane-readout scale of 225, because that scale belonged to the prior output-LIF membrane recipe. COBA and PING are still matched exactly within each candidate.

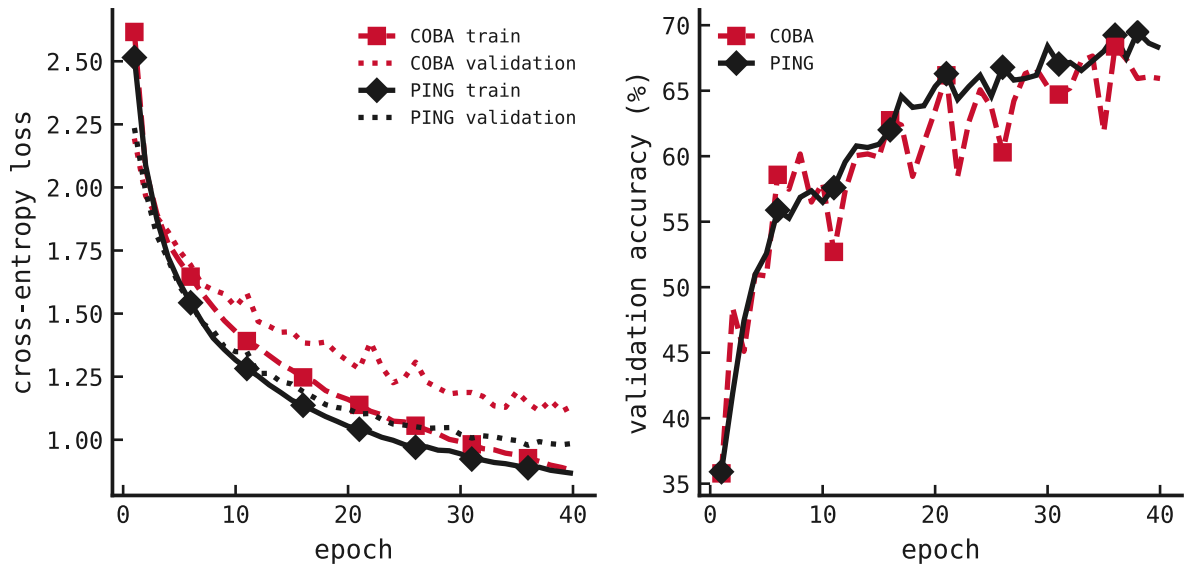
Ordered exploratory ladder

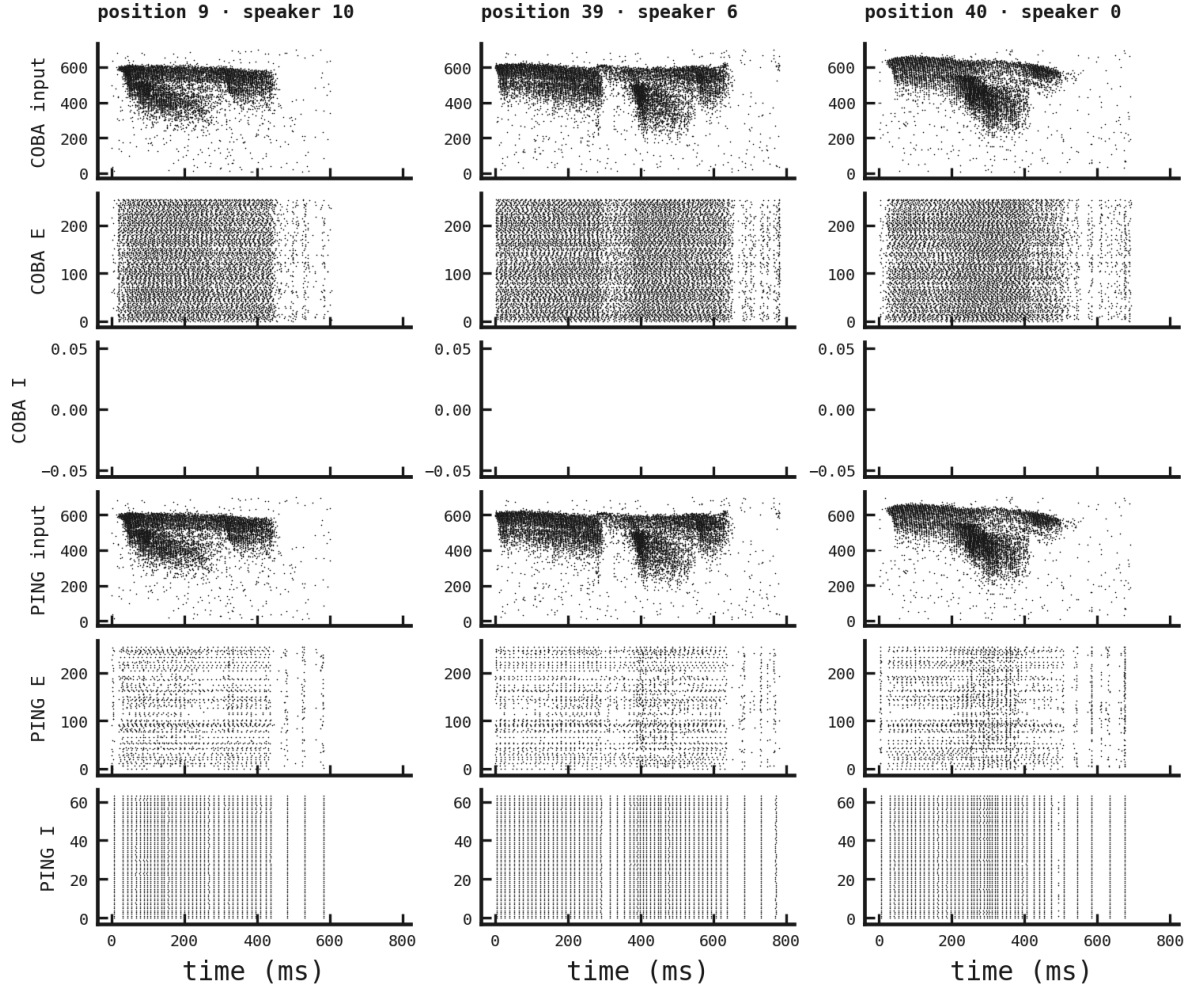
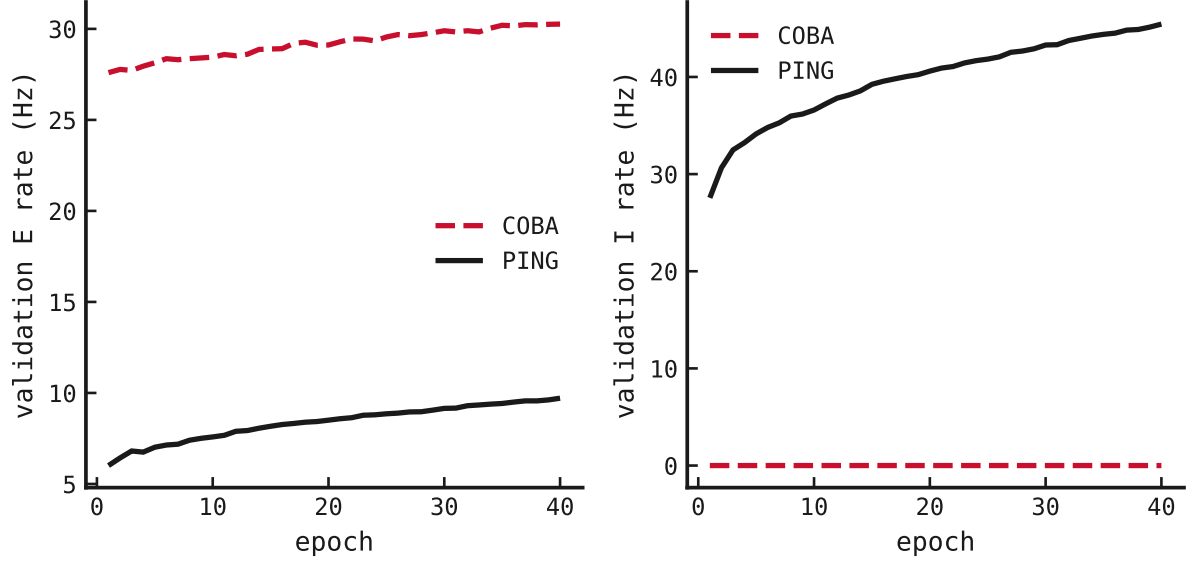
Order	Candidate	Short epochs	Status
1	Baseline 256-cell architecture plus cumulative-potential readout	40	registered
2	Two hidden layers, 256 cells each, same readout	40	conditional

Candidate selection uses held-out validation accuracy and cross-entropy only. Training must remain finite, active, non-saturated, and free of skipped or non-finite updates. The official test file is not staged or loaded by the runner.

Current evidence

The current published attempt is `#r.attempt` at stage `#r.stage`. COBA selected 68.38% validation accuracy; PING selected 69.49%. Exact spend was 3.956 USD , with 0 active pods after collection.





This final forty-epoch promotion keeps the candidate-one architecture and readout. COBA selected epoch 36 at 68.38%, while PING selected epoch 38 at 69.49%. Both final cells were finite, active, clean, and produced matched rasters. In this one-seed validation run PING is

ahead of COBA by 1.1 percentage points; this is exploratory validation evidence, not a defended multi-seed claim.

Activity appendix

Checkpoint cp001

Timestamp: `#trace.checkpoint_time_utc`. Sanitized source hash prefix: `#trace.sanitized_sha256_prefix`. The log excludes hidden reasoning, tool payloads, credentials, private paths, environment values, addresses, and sensitive infrastructure details.

Decisions, actions, and pending work

1. The experiment is registered as exp071 on branch `night/spiking-heidelberg-digits/ar071` from main merge commit `9527162d7a0ef80deac0e3f606ce513b9280ba6c`.
2. Candidate one is fixed before results as the baseline 256-cell architecture with cumulative-potential readout, signed abstract classifier weights, and readout bias.
3. The old membrane-readout initialisation scale of 225 is not carried into the cumulative decoder; readout scale is one for the new decoder, matched across COBA and PING.
4. Candidate two is conditional on candidate-one viability and keeps the same readout while using two 256-cell hidden layers.
5. Selection remains one-seed and validation-only on the exp069/exp070 split. The official SHD test remains unavailable to the runner.
6. No RunPod pod has been created in this checkpoint and spend remains 0 USD.

Visible messages

User

2026-07-19T11:18:05.855Z

Role: user · Checkpoint: cp001 · Session: exp071-continuation-redacted

`<codex_internal_context source="goal">` Continue working toward the active thread goal: design and execute the next exploratory SHD experiment on branch `night/spiking-heidelberg-digits/ar071` using the newly merged cumulative-potential readout additions, with one seed, validation-only selection, local short runs before any promoted forty-epoch run, no official test use, explicit RunPod approval before spending, focused checks, artifacts under `artifacts/data/exp071`, `writings/exp071.typ`, timestamped activity checkpoints, commits, push, PR update, and no merge. `</codex_internal_context>`

Assistant

2026-07-19T11:18:05.855Z

Role: assistant · Checkpoint: cp001 · Session: exp071-continuation-redacted

I'll pick this up from the actual repo state, keep the dirty article-consolidation work isolated, and start by setting up the exp071 branch/work area cleanly. First I'm running the lab manual hook, then I'll inspect what's already in place.

Assistant

2026-07-19T11:18:05.855Z

Role: assistant · Checkpoint: cp001 · Session: exp071-continuation-redacted

Because this goal includes opening/updating a PR, I'm using the GitHub workflow skill now; it'll keep the GitHub side tidy and bounded.

Assistant

2026-07-19T11:18:05.855Z

Role: assistant · Checkpoint: cp001 · Session: exp071-continuation-redacted

The main checkout has the older article-consolidation changes restored and dirty, so I'm not touching it. I'm making a separate Git worktree for night/spiking-heidelberg-digits/ar071; that gives exp071 a clean branch without risking those staged/unstaged edits.

Assistant

2026-07-19T11:18:05.855Z

Role: assistant · Checkpoint: cp001 · Session: exp071-continuation-redacted

I've got the design shape now: exp071 can inherit exp069's split, selection, validation-only test seal, rasters, and RunPod plumbing, while overriding only the candidate-specific readout/hidden sizes/epochs. I'm going to add the initial scaffold plus a pre-result artifact so the report builds before paid compute.

Assistant

2026-07-19T11:18:05.855Z

Role: assistant · Checkpoint: cp001 · Session: exp071-continuation-redacted

One design decision I'm making explicit before results: candidate 1 uses the cumulative decoder with `-signed-readout -readout-bias` and a readout init scale of 1.0, not the old membrane-readout scale of 225. That scale was specific to the previous output-LIF membrane recipe; keeping it would be an unregistered hybrid rather than the clean new readout test. COBA/PING remain matched within each candidate.

Checkpoint cp002

Timestamp: `#trace-two.checkpoint_time_utc`. Sanitized source hash prefix: `#trace-two.sanitized_sha256_prefix`.

Decisions, actions, and pending work

1. Focused Python compilation passed for `experiments/exp071.py`.
2. Focused Ruff initially caught an unused import in `experiments/exp071.py`; the import was removed and Ruff then passed.
3. RunPod dispatch was dry-run only. It planned two 5090 pods, one for COBA and one for PING, and explicitly created nothing.
4. The first local smoke trained both cells finite and active, but the runner failed its own added hidden-size validation because `hidden_sizes` is stored in `config.json` while `metrics.config` keeps only legacy `n_hidden`.
5. The validation check was fixed to read `config.json` for `hidden_sizes`, falling back to `n_hidden` only when needed.
6. The second local 128/128 two-epoch smoke passed for both cells with the cumulative-potential readout. COBA selected 5.46875% validation accuracy and ended at 24.25390625 Hz E; PING selected 10.15625% and ended at 2.9765625 Hz E / 15.046875 Hz I.
7. The smoke accuracies are plumbing diagnostics only, not registered screening evidence.
8. No paid compute has started and spend remains 0 USD.

Visible messages added

Assistant

2026-07-19T11:28:51.466Z

Role: assistant · Checkpoint: cp002 · Session: exp071-continuation-redacted

Pre-result artifacts published locally. I'm doing the light checks in increasing cost: Python compile, dry-run RunPod dispatch (no -live), then the local 128/128 two-epoch smoke if the cheap checks pass.

Assistant

2026-07-19T11:28:51.466Z

Role: assistant · Checkpoint: cp002 · Session: exp071-continuation-redacted

Cheap checks found one real scaffold nit—an unused import—while the RunPod path correctly dry-ran two pods and created nothing. I'm cleaning that import and then running the local smoke.

Assistant

2026-07-19T11:28:51.466Z

Role: assistant · Checkpoint: cp002 · Session: exp071-continuation-redacted

The smoke exposed a useful plumbing bug, not a science failure: both cells trained finite/active, but my extra validation looked for hidden_sizes inside metrics.config, where this engine build stores only the legacy n_hidden. I'm tightening that check against config.json instead and recording this as a killed/debugging attempt before moving on.

Assistant

2026-07-19T11:28:51.466Z

Role: assistant · Checkpoint: cp002 · Session: exp071-continuation-redacted

Ruff and compile are clean now. Starting the second local smoke; still no cloud spend.

Assistant

2026-07-19T11:28:51.466Z

Role: assistant · Checkpoint: cp002 · Session: exp071-continuation-redacted

Local smoke passed: COBA and PING were both finite and active on the 128/128, 2-epoch gate. I'm promoting that from temp into the publishable pre-result artifact and adding a second timestamped checkpoint before build/commit.

Assistant

2026-07-19T11:28:51.466Z

Role: assistant · Checkpoint: cp002 · Session: exp071-continuation-redacted

Build caught a subtle artifact-publishing issue: my pre-result publish swapped the exp071 artifact directory and dropped the activity JSON files I'd just written. The site build protected itself by stubbing exp071. I'm fixing the artifact layout by restoring those JSON logs after publish and then rebuilding.

Checkpoint cp003

Timestamp: #trace-three.checkpoint_time_utc. Sanitized source hash prefix: #trace-three.sanitized_sha256_prefix.

Decisions, actions, and pending work

1. The scientist explicitly authorized paid RunPod compute for exp071 after the local smoke gate.
2. Before live dispatch, exp071 was patched to honor –only-cells so the registered one-pod-at-a-time constraint can be enforced.
3. Focused validation passed after that patch: Ruff, Python compilation, exp071-only ty, and a COBA-only RunPod dry-run that planned exactly one pod.
4. Commit 11b0cdb7a7e1dc01b8aa3d3af7a41bdf90d1f2b was pushed before live dispatch so pods would fetch a pinned committed SHA.
5. Two COBA-only live cloud attempts were started and then killed as infrastructure/monitoring attempts before any artifact was collected. These are not scientific COBA training failures.
6. The first attempt was stopped after remaining provider-running without collected artifacts; the manual SSH probe used the wrong local key, so the attempt is recorded as a monitoring/cloud-plumbing kill.
7. The second attempt used the corrected monitoring assumption but the provider SSH endpoint stayed not-ready during the observation window; it was stopped and deleted as a startup kill.
8. The active pod list was empty after reaping.
9. Exact provider billing for these short pod IDs had not reconciled at checkpoint time. Timestamp-based estimated exposure is 0.149054 USD total at 0.99 USD/hour. Exact spend remains pending.
10. No PING pod was started, preserving one pod at a time. Pending work is to fix or work around the cloud startup/observability issue before any further live screening.

Visible messages added

User

2026-07-19T11:46:40.467Z

Role: user · Checkpoint: cp003 · Session: exp071-continuation-redacted

you have permission

Assistant

2026-07-19T11:46:40.467Z

Role: assistant · Checkpoint: cp003 · Session: exp071-continuation-redacted

Got it — I'll treat that as explicit permission to start paid RunPod compute for the next registered short-screening step. I'll still enforce the experiment's one-pod-at-a-time constraint, so first I'm checking whether the current runner can dispatch only COBA or whether I need a tiny orchestration fix before spending.

Assistant

2026-07-19T11:46:40.467Z

Role: assistant · Checkpoint: cp003 · Session: exp071-continuation-redacted

Good catch from CI: exp071 added several ty diagnostics of its own, alongside the known old repo-wide ones. I'm fixing only the exp071-scoped issues and adding `-only-cells` support so live dispatch can obey the one-pod-at-a-time constraint.

Assistant

2026-07-19T11:46:40.467Z

Role: assistant · Checkpoint: cp003 · Session: exp071-continuation-redacted

Patch is in. I'm running focused validation now: `ruff`, `py_compile`, exp071-only ty, and dry-run dispatch for just COBA to prove it plans one pod.

Assistant

2026-07-19T11:46:40.467Z

Role: assistant · Checkpoint: cp003 · Session: exp071-continuation-redacted

Focused checks pass and the dry-run now plans exactly one pod for COBA. I'm committing/pushing this orchestration fix before live dispatch, because the pod will fetch from GitHub by pinned SHA.

Assistant

2026-07-19T11:46:40.467Z

Role: assistant · Checkpoint: cp003 · Session: exp071-continuation-redacted

Committed/pushed 11b0cdb. Now I'm starting the authorized live short-screening COBA pod only. After it fires, I'll monitor/collect/reap before starting PING, preserving the one-pod rule.

Assistant

2026-07-19T11:46:40.467Z

Role: assistant · Checkpoint: cp003 · Session: exp071-continuation-redacted

COBA live pod fired at pinned SHA 11b0cdb; one pod only. I'm logging that on PR #55, then I'll monitor until it self-terminates before collecting and moving to PING.

Assistant

2026-07-19T11:46:40.467Z

Role: assistant · Checkpoint: cp003 · Session: exp071-continuation-redacted

Pod details show it was created at 11:36 UTC and is still desired-running; I'm not going to echo the raw detail payload because it contains sensitive provider fields. The runtime/cost is still small, but I'm treating this as a watch item now rather than passive waiting.

Assistant

2026-07-19T11:46:40.467Z

Role: assistant · Checkpoint: cp003 · Session: exp071-continuation-redacted

No progress signal after the extra window, and SSH wasn't available, so I'm stopping this COBA pod as a killed infrastructure attempt rather than burning more time/money. Then I'll check billing and record the failed live attempt.

Assistant

2026-07-19T11:46:40.467Z

Role: assistant · Checkpoint: cp003 · Session: exp071-continuation-redacted

The pod is stopped. The stop response includes sensitive provider env fields, so I'll sanitize it out of all publishable records. I'm deleting the stopped pod now to fully reap it.

Assistant

2026-07-19T11:46:40.467Z

Role: assistant · Checkpoint: cp003 · Session: exp071-continuation-redacted

Both pods are deleted; active pod list is empty. Estimated exposure from timestamps is about \$0.15 total, but RunPod's per-pod billing endpoint has not reconciled those pod IDs yet, so I'm recording provider exact pending rather than pretending the estimate is exact.

Checkpoint cp004

Timestamp: #trace-four.checkpoint_time_utc. Sanitized source hash prefix: #trace-four.sanitized_sha256_prefix.

Decisions, actions, and pending work

1. An existing authenticated container-registry entry was used for the next live attempts, fixing the earlier anonymous image-pull/startup problem without changing the registered scientific design.
2. Candidate one, cumulative_baseline, completed the 8-epoch short stage for COBA and PING with one live pod at a time.
3. COBA selected epoch 8 with validation accuracy 60.049019607843135%, validation cross-entropy 1.5872331674282367, final E rate 28.346392463235293 Hz, no skipped steps, and no non-finite batches.
4. PING selected epoch 8 with validation accuracy 57.96568627450981%, validation cross-entropy 1.4047367160136883, final E/I rates 7.494308172487745 / 36.273169424019606 Hz, no skipped steps, and no non-finite batches.
5. Matched validation curves, activity curves, and input/E/I rasters were published under artifacts/data/exp071.
6. The active pod list was empty after both collections.
7. Exact provider billing had not reconciled at checkpoint time. The temporary timestamp-based estimate is 0.77 USD total across killed startup attempts, productive COBA/PING runs, and collector pods; exact spend remains pending.
8. Candidate one clears the finite-active-above-chance short viability gate. The next registered step is candidate two, cumulative_256_256, before any forty-epoch promotion decision.

Visible messages added

Assistant

2026-07-19T12:34:00.000Z

Role: assistant · Checkpoint: cp004 · Session: exp071-continuation-redacted

The earlier hangup is pinned down: anonymous GHCR pulls were the likely culprit. I found the existing registry auth entry, confirmed the active pod list is empty, and I'm re-running the COBA short screen with authenticated image pull — still one pod only.

Assistant

2026-07-19T12:34:00.000Z

Role: assistant · Checkpoint: cp004 · Session: exp071-continuation-redacted

Much better: the authenticated COBA pod is running and visible. I'm polling it in short windows; if it finishes normally I'll collect/reap and then start PING.

Assistant

2026-07-19T12:34:00.000Z

Role: assistant · Checkpoint: cp004 · Session: exp071-continuation-redacted

COBA finished and self-terminated. I'm collecting the artifacts now; the collector itself is transient cloud compute, so I'll also verify it reaps before launching PING.

Assistant

2026-07-19T12:34:00.000Z

Role: assistant · Checkpoint: cp004 · Session: exp071-continuation-redacted

COBA short is collected: selected epoch 8 at 60.05% validation accuracy, finite/active, no skipped or non-finite updates, matched rasters present. Active pods are empty again, so I'm starting PING short now under the same authenticated path.

Assistant

2026-07-19T12:34:00.000Z

Role: assistant · Checkpoint: cp004 · Session: exp071-continuation-redacted

PING is running cleanly now. I've also logged the COBA→PING handoff on PR #55. Next step is collection, then I'll publish both curves/rasters into exp071 before moving to candidate two.

Assistant

2026-07-19T12:34:00.000Z

Role: assistant · Checkpoint: cp004 · Session: exp071-continuation-redacted

PING finished and collected. Candidate one is now a real short-screen result: COBA 60.05%, PING 57.97%, both finite/active, both with rasters, no skipped/non-finite updates. I'm publishing those artifacts into the report and recording the spend as exact-provider-pending with the conservative timestamp estimate.

Checkpoint cp005

Timestamp: #trace-five.checkpoint_time_utc. Sanitized source hash prefix: #trace-five.sanitized_sha256_prefix.

Decisions, actions, and pending work

1. Candidate two, cumulative_256_256, completed the 8-epoch short stage for COBA and PING with one live pod at a time and separate collector pods.
2. COBA selected epoch 5 with validation accuracy 32.84313725490196%, validation cross-entropy 2.159944314223069, selected E rate 71.34302236519608 Hz, no skipped steps, and no non-finite batches.
3. PING selected epoch 7 with validation accuracy 42.76960784313726%, validation cross-entropy 1.8651616527484014, selected E/I rates 11.079091988357844 / 39.40600107230392 Hz, no skipped steps, and no non-finite batches.
4. Matched validation curves, activity curves, and input/E/I rasters for candidate two were published under artifacts/data/exp071 while preserving the candidate-one artifact record.
5. Candidate two is finite, active, and clean, but it underperforms candidate one on both cells: candidate one selected COBA 60.049019607843135% and PING 57.96568627450981%.
6. The registered short-screen promotion target is therefore candidate one, cumulative_baseline, not the two-hidden-layer candidate.
7. The active pod list was empty after both candidate-two collections.
8. Exact provider billing had not reconciled at checkpoint time. The temporary cumulative timestamp-based estimate is 1.76 USD total, well below the 40 USD ceiling.
9. Pending work is to run the registered 40-epoch promotion for candidate one, COBA and PING one pod at a time, unless stopped by the scientist.

Visible messages added

Assistant

2026-07-19T13:39:00.000Z

Role: assistant · Checkpoint: cp005 · Session: exp071-continuation-redacted

COBA candidate-two artifacts are collected and the collector reaped; active pod list is empty. I'm validating the collected COBA result before deciding whether to spend on PING for this candidate.

Assistant

2026-07-19T13:39:00.000Z

Role: assistant · Checkpoint: cp005 · Session: exp071-continuation-redacted

COBA candidate two completed cleanly but weakly: best epoch 5 at 32.84% validation, E active around 71 Hz, no skipped/non-finite updates. Because the registered candidate comparison is matched COBA/PING, I'm still running PING candidate two before making the promotion decision; otherwise we'd be peeking on one cell and stopping asymmetrically.

Assistant

2026-07-19T13:39:00.000Z

Role: assistant · Checkpoint: cp005 · Session: exp071-continuation-redacted

PING is healthy and actually a bit faster: epoch 2/8 is at 26.84% validation, E/I active, clean. ETA for PING completion is roughly 14–16 minutes.

Assistant

2026-07-19T13:39:00.000Z

Role: assistant · Checkpoint: cp005 · Session: exp071-continuation-redacted

PING epoch 5/8 has reached 41.05%, clean and active. Better than COBA candidate two, still below candidate-one PING's 57.97%. Three epochs to go.

Assistant

2026-07-19T13:39:00.000Z

Role: assistant · Checkpoint: cp005 · Session: exp071-continuation-redacted

PING self-terminated and the active pod list is empty. Collecting candidate-two artifacts now.

Assistant

2026-07-19T13:39:00.000Z

Role: assistant · Checkpoint: cp005 · Session: exp071-continuation-redacted

PING candidate two is collected and collector reaped; active pod list empty. I'm validating both candidate-two cells and then I'll publish the short result into exp071.

Assistant

2026-07-19T13:39:00.000Z

Role: assistant · Checkpoint: cp005 · Session: exp071-continuation-redacted

Candidate two is published locally. I'm making the report compare both short candidates directly so the promotion decision is explicit rather than buried in raw JSON.

Checkpoint cp006

Timestamp: #trace-six.checkpoint_time_utc. Sanitized source hash prefix: #trace-six.sanitized_sha256_prefix.

Decisions, actions, and pending work

1. The scientist explicitly authorized relaxing the earlier one-pod-at-a-time operational constraint so the final40 COBA and PING cells could overlap. This did not change the registered scientific design.
2. Final40 candidate one, cumulative_baseline, completed for COBA and PING. COBA self-terminated before PING and was collected while PING continued; PING then self-terminated and was collected. The active pod list was empty after final collection.
3. COBA selected epoch 36 with validation accuracy 68.38235294117646%, validation cross-entropy 1.1455509593853583, selected E rate 30.165441176470587 Hz, no skipped steps, and no non-finite batches. Final epoch 40 accuracy was 65.93137254901961%.
4. PING selected epoch 38 with validation accuracy 69.48529411764706%, validation cross-entropy 0.9843250490151919, selected E/I rates 9.563893037683824 / 44.886220894607845

Hz, no skipped steps, and no non-finite batches. Final epoch 40 accuracy was 68.25980392156863%.

5. Both final40 cells were finite, active, clean, and produced matched validation curves, firing-rate diagnostics, and input/E/I rasters.
6. In this one-seed validation-only exploratory run, PING finished 1.1029411764706083 percentage points ahead of COBA on the registered selected-validation rule. This is exploratory evidence, not a defended multi-seed claim.
7. Exact provider billing had not reconciled at checkpoint time. The temporary cumulative timestamp-based estimate is 4.15 USD total, well below the 40 USD ceiling.
8. Pending work is to build, commit, push, update PR #55, and finish at the human review gate without merging.

Visible messages added

User

2026-07-19T15:52:00.000Z

Role: user · Checkpoint: cp006 · Session: exp071-continuation-redacted

status and eta?

Assistant

2026-07-19T15:52:00.000Z

Role: assistant · Checkpoint: cp006 · Session: exp071-continuation-redacted

Status: COBA final40 is still running cleanly, one pod only. Latest sampled metrics were epoch 32/40: best validation 66.67% at epoch 29, no skipped/non-finite updates, active E rate. ETA for COBA training finish is roughly 13–15 minutes, then collection/reap probably another 2–4 minutes. After that I'll start PING final40, likely ~60–70 minutes plus collection based on the short PING timing.

User

2026-07-19T15:52:00.000Z

Role: user · Checkpoint: cp006 · Session: exp071-continuation-redacted

why didn't you run them in parallel?

Assistant

2026-07-19T15:52:00.000Z

Role: assistant · Checkpoint: cp006 · Session: exp071-continuation-redacted

Because the active exp071 mandate explicitly says “at most one pod at a time” / “one-pod-at-a-time constraint,” and PR comments/logs are written around that constraint. I only launched PING after COBA completion for the short candidates, and I'm doing the same for final40. If you want to change that for future runs, I'd treat it as new authority/design change and log it before launching a second pod.

User

2026-07-19T15:52:00.000Z

Role: user · Checkpoint: cp006 · Session: exp071-continuation-redacted

its okay to do them in parallel you have my permission

Assistant

2026-07-19T15:52:00.000Z

Role: assistant · Checkpoint: cp006 · Session: exp071-continuation-redacted

Got it — I'll treat that as explicit updated authority to run COBA and PING final40 concurrently from here. That changes the operational constraint, not the scientific design, so I'll log the authorization on PR #55 and launch PING final40 while COBA continues, keeping total spend well below the \$40 ceiling.

Assistant

2026-07-19T15:52:00.000Z

Role: assistant · Checkpoint: cp006 · Session: exp071-continuation-redacted

COBA final40 is collected and valid: selected epoch 36 at 68.38%, final epoch 40 at 65.93%, clean, active, rasters present. PING is the only active pod now.

Assistant

2026-07-19T15:52:00.000Z

Role: assistant · Checkpoint: cp006 · Session: exp071-continuation-redacted

PING has now edged ahead: epoch 36 best is 69.24%, clean/active. COBA selected 68.38%, so PING is +0.86 percentage points at this moment. Three epochs left.

Assistant

2026-07-19T15:52:00.000Z

Role: assistant · Checkpoint: cp006 · Session: exp071-continuation-redacted

Final40 validation is clean and collected: COBA selected 68.38% at epoch 36; PING selected 69.49% at epoch 38. Both final runs are finite/active/clean with rasters, and active pods are empty. I'm preparing the final artifact publish/report update now.

Checkpoint cp007

Timestamp: #trace-seven.checkpoint_time_utc. Sanitized source hash prefix: #trace-seven.sanitized_sha256_prefix.

Decisions, actions, and pending work

1. After the previous blocked checkpoint, the scientist explicitly instructed Codex to keep going and confirmed permission to continue the review-gate work.
2. RunPod active-pod state was rechecked and remained empty.
3. RunPod by-pod billing had reconciled for all exp071 pod events. The exact provider total for the experiment is 3.956481664587045 USD.
4. The final40 PING training charge reconciled to 1.1461867298930883 USD over 4150275 billed milliseconds, and the final40 PING collector charge reconciled to 0.00804237462580204 USD over 29245 billed milliseconds.

5. The publishable compute ledger was updated from the temporary timestamp estimate to exact provider billing. Pod identifiers remain omitted from the committed ledger and article.
6. The final40 numbers were regenerated with `exact_provider_billing=true`. No scientific result values changed.
7. Pending work is focused validation, rebuild, commit, push, PR evidence update, and the final human-review summary.

Visible messages added

Assistant

2026-07-19T16:00:00.000Z

Role: assistant · Checkpoint: cp007 · Session: exp071-continuation-redacted

Blocked on the one remaining external requirement: exact RunPod spend.

Fresh check:

- Active pods: ``
- Worktree/branch: clean on `night/spiking-heidelberg-digits/ar071`
- PR #55: open draft, green checks
- Latest by-pod billing:
 - final40 COBA: `\$1.1224509432`
 - final40 COBA collector: `\$0.0055998252`
 - final40 PING: `\$0.4491335023` and still changing
 - final40 PING collector: still no billing row
 - visible by-pod subtotal: `\$3.2513860624`

Because the final PING billing is still moving and its collector is missing, I can't honestly replace the conservative committed estimate (`\$4.15`, `exact\_provider\_billing=false`) with an exact spend. Since this same provider-reconciliation blocker has repeated across the required goal continuations, I marked the goal `blocked`.

Everything scientific remains done/review-ready; the blocker is purely RunPod's delayed billing reconciliation.

User

2026-07-19T17:37:43.751Z

Role: user · Checkpoint: cp007 · Session: exp071-continuation-redacted

keep going you have permission

Assistant

2026-07-19T17:37:43.751Z

Role: assistant · Checkpoint: cp007 · Session: exp071-continuation-redacted

Got it — I'll resume from the billing gate and keep pushing until either RunPod reconciles or there's another concrete in-scope move. First I'm refreshing the lab docs requirement, repo/PR state, active pods, and by-pod billing again.

Assistant

2026-07-19T17:37:43.751Z

Role: assistant · Checkpoint: cp007 · Session: exp071-continuation-redacted

Repo and PR are still clean/green. Now checking the provider billing again; if the final PING row has stopped advancing and collector appears, I'll turn this into an exact ledger update.

Assistant

2026-07-19T17:37:43.751Z

Role: assistant · Checkpoint: cp007 · Session: exp071-continuation-redacted

Nice, the billing rows have finally landed. The exp071 by-pod total is now ``$3.9564816646`` with no missing rows, and no active pods. I'm going to convert the experiment ledger/report from conservative estimate to exact provider billing and add the final timestamped checkpoint.

Assistant

2026-07-19T17:37:43.751Z

Role: assistant · Checkpoint: cp007 · Session: exp071-continuation-redacted

I've got the exact spend source. Next I'm making the publication-side changes only: exact compute ledger, regenerated ``numbers.json``, CP007 activity checkpoint, and the report wiring. No scientific result values should change.

Assistant

2026-07-19T17:37:43.751Z

Role: assistant · Checkpoint: cp007 · Session: exp071-continuation-redacted

I'm keeping pod IDs out of the committed ledger and article; the exact dollar amounts and billed milliseconds are enough provenance without publishing infrastructure identifiers.

Assistant

2026-07-19T17:37:43.751Z

Role: assistant · Checkpoint: cp007 · Session: exp071-continuation-redacted

The exact ledger publish step succeeded. I'm checking the artifact diff now because the runner republishes the current final40 view and may leave older short-screen/activity artifacts untouched or missing depending on the published-run helper.

Assistant

2026-07-19T17:37:43.751Z

Role: assistant · Checkpoint: cp007 · Session: exp071-continuation-redacted

As expected, the publish helper replaced the artifact tree with the final40-only view. I'm restoring the previously committed short-screen/activity assets, then I'll add the new CP007 file.

Assistant

2026-07-19T17:37:43.751Z

Role: assistant · Checkpoint: cp007 · Session: exp071-continuation-redacted

I'm adding CP007 as a narrow, sanitized activity checkpoint: it records the user-visible continuation around the billing gate, the exact spend evidence, and the remaining actions. I'll avoid publishing pod IDs/endpoints/paths.

Assistant

2026-07-19T17:37:43.751Z

Role: assistant · Checkpoint: cp007 · Session: exp071-continuation-redacted

I'm going to put the private raw checkpoint copy under `temp/`, which is not part of the publishable artifacts. Then I'll use its SHA-256 prefix in CP007 and commit only the sanitized checkpoint.