

Adaptive conductance controls for matched SHD networks

exp072 · 19 July 2026 · Draft

Abstract

This one-seed exploratory experiment tests whether two newly added conductance-neuron controls move matched Dale-constrained COBA and PING networks beyond the exp071 roughly 70% validation plateau on SHD. The control is the exp071 cumulative-potential readout recipe: `--readout cumulative-potential --signed-readout --readout-bias`. The ladder then tests bounded trainable leak/membrane time constants, bounded adaptive E-cell thresholds, and their combination. All selection decisions use the held-out validation split from the official SHD training set; the official test set remains sealed.

Goal prompt

```
/goal Design and execute the next exploratory SHD experiment from updated
main at f32299bb1b48d18ba9dfe6328097d2147bcc6286, using the newly merged
tools/snn adaptive conductance controls to test whether COBA/PING validation
accuracy can move meaningfully beyond the exp071 ~70% plateau toward
canonical SHD RSNN performance.
```

```
Create the next numbered experiment, likely exp072, on a new branch. Do not
modify main directly after branching. Open a draft PR after the first
meaningful commit.
```

```
Scientific objective:
```

```
Run a one-seed exploratory validation-only ladder on SHD comparing matched
Dale-constrained COBA and PING networks while testing two new neuron-dynamics
paths:
```

1. trainable conductance leak / membrane time constants via `--train-leak`;
2. adaptive E-cell thresholds via `--adaptive-threshold`;
3. the combination of both.

```
Use the exp071 cumulative-potential readout baseline as the control:
```

```
--readout cumulative-potential --signed-readout --readout-bias
```

```
Constraints:
```

- Use one seed only.
- Do not use the official SHD test set.
- Use a held-out validation split from the training set for all selection decisions.
- Keep COBA and PING matched except for registered cell-specific voltage-

gradient dampening and PING E/I coupling.

- Preserve matched input, readout, training split, optimizer, batch, preprocessing, and candidate flags across COBA/PING within each candidate.
- COBA should use the no-inhibitory-loop setting; PING should use the registered inhibitory-loop setting and dampening 1000.
- Use short runs first for iteration speed, about 8-10 epochs per candidate.
- Promote only the most promising candidate to a matched 40-epoch validation run.
- Do not run multiple seeds unless I explicitly authorize it later.
- RunPod spending requires explicit approval before pod creation.
- Use at most one pod at a time unless I explicitly authorize parallelism.
- 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:

0. Control: exp071 architecture/readout without new adaptive controls.

1. Trainable leak only:

--train-leak

2. Adaptive threshold only:

--adaptive-threshold

3. Combined:

--train-leak --adaptive-threshold

Suggested default bounds:

- E membrane tau: [5, 50] ms
- I membrane tau: [2, 20] ms
- adaptive threshold tau: [50, 500] ms
- adaptive threshold initial strength: 1.0 mV
- adaptive threshold max strength: 20.0 mV

Deliverables:

- Produce artifacts/data/exp072 with numbers.json, provenance, reproducer, training curves, firing-rate diagnostics, parameter diagnostics for learned tau/adaptation values, and matched input/E/I rasters.
- Write writings/exp072.typ as the canonical cold-readable experiment report.
- Include this goal prompt below the abstract.
- Append timestamped activity-log/thread checkpoints in the experiment appendix.
- Record scientific decisions, failures, anomalies, costs, commits, results, and pending work.
- Sanitize any publishable logs before committing.
- Build and validate with focused checks plus demolab build.
- Commit meaningful attempts, including killed attempts, with clear messages.
- Push the branch and update the PR with evidence.
- Finish at the human review gate with a concise evidence summary, exact compute spend, validation performed, links to the rendered exp072 file and PR, and a recommendation for the next move toward 90% SHD.

Methods

Locked comparison

Both cells use seed 42, the established SHD 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, matched input preprocessing, and the same cumulative-potential readout. COBA uses the no-inhibitory-loop setting and no voltage-gradient dampening; PING uses the registered inhibitory loop and dampening 1000. Those cell-specific settings are the only intended COBA/PING differences within a candidate.

Candidate ladder

Order	Candidate	Extra flags	Short epochs
0	Control: exp071 architecture/ readout	none	8
1	Trainable conductance leak / membrane time constants	--train-leak	8
2	Adaptive E-cell thresholds	--adaptive-threshold	8
3	Combined leak plus adaptation	--train-leak --adaptive- threshold	8

The default bounds are E membrane tau [5, 50] ms, I membrane tau [2, 20] ms, adaptive-threshold tau [50, 500] ms, initial adaptive strength 1.0 mV, and maximum adaptive strength 20.0 mV. 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.

Results

The eight-epoch ladder completed for all four candidates. Control had the best paired average, but trainable leak was the only modification that improved the target PING cell while keeping COBA close, so `train_leak` was promoted to forty epochs.

Candidate	COBA acc.	COBA CE	PING acc.	PING CE
control	60.17%	1.5905	56.86%	1.4221
train_leak	58.95%	1.6838	57.72%	1.4365
adaptive_threshold	50.61%	1.9109	53.68%	1.532
combined	50.37%	1.8996	50.98%	1.5616

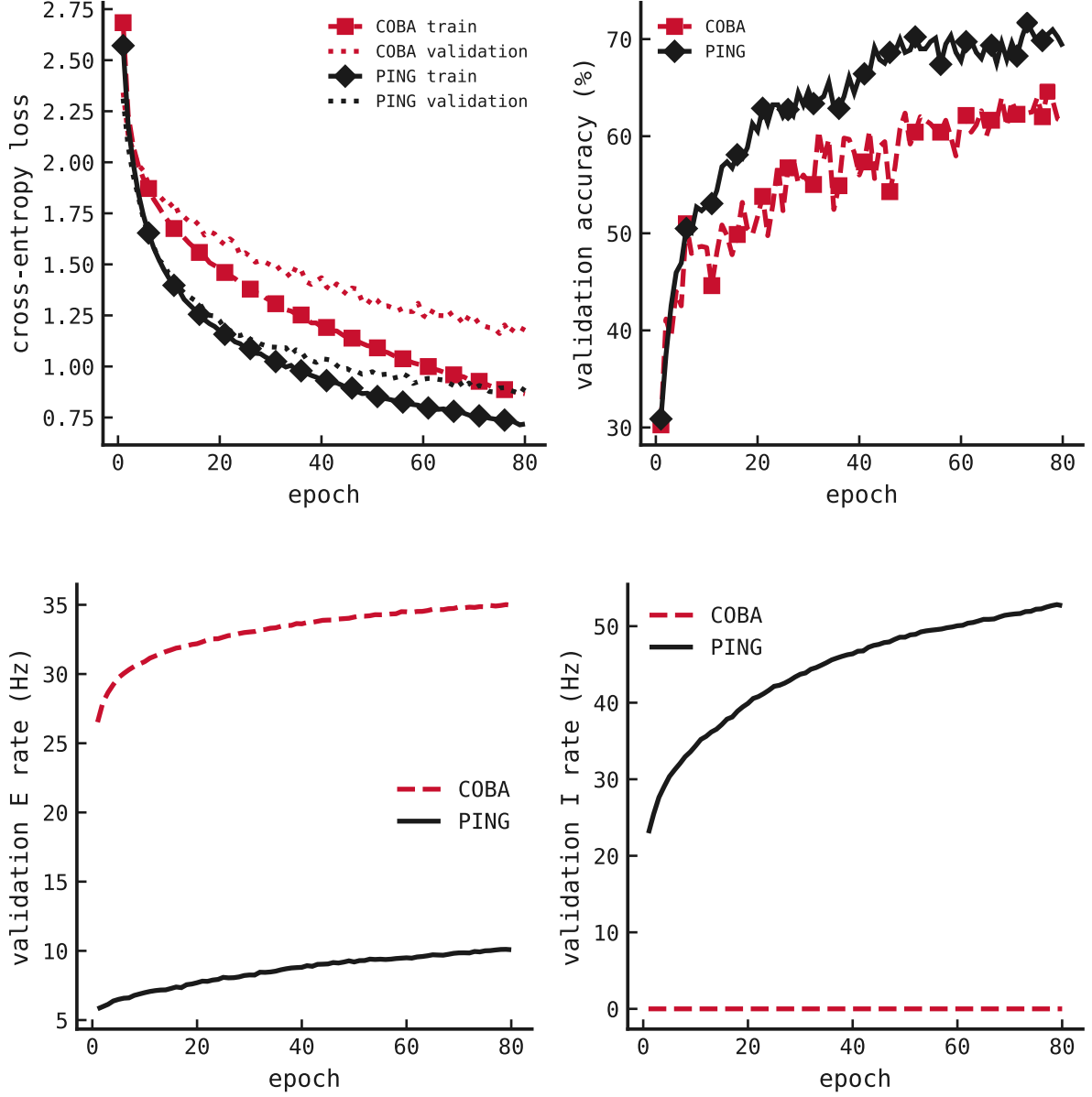
The promoted forty-epoch `train_leak` run selected 69.49% for COBA and 71.08% for PING. As a later sensitivity check, the initially weak `combined` candidate was also run for eighty epochs.

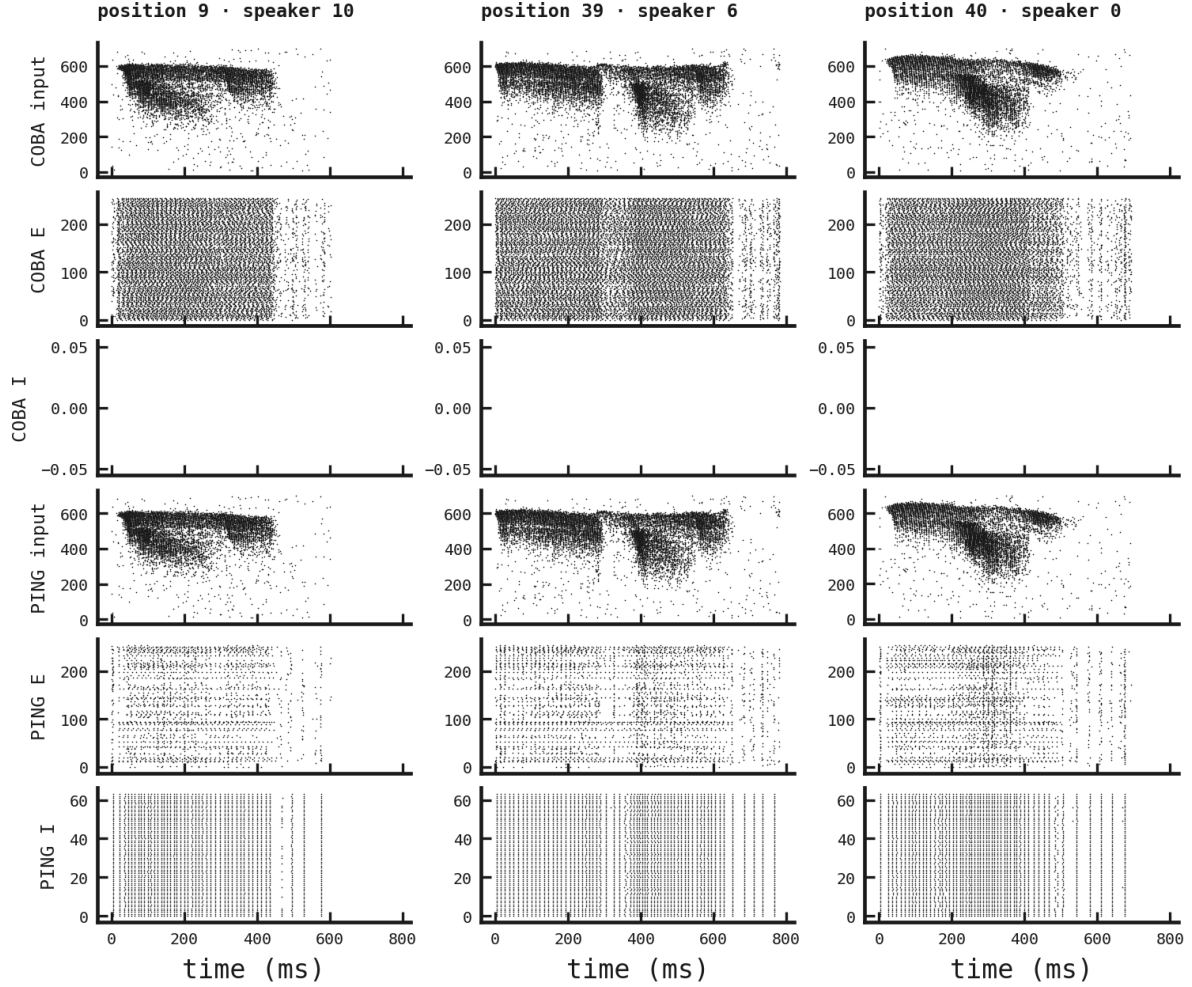
Long run	COBA acc.	COBA epoch	PING acc.	PING epoch
train_leak, 40 epochs	69.49%	37	71.08%	40
combined, 80 epochs	64.58%	77	71.69%	73

The eighty-epoch combined follow-up nudged PING to 71.69%, but only after twice as many epochs and with stuck-dynamics warnings; COBA remained lower at 64.58% and showed

saturated/stuck dynamics. This supports the interpretation that adaptive thresholds did not provide a useful path to the canonical high-accuracy SHD regime in this setup.

Cumulative RunPod spend recorded by the experiment ledgers is 10.26 USD. The provider billing rows had not posted at publication time, so the spend is marked `#r.spend.status`; active pods after the final collection were 0.





The current figure set below shows the latest published attempt, `#r.attempt` at `#r.stage`; previous long-run and short-ladder numeric summaries remain archived under `raw/collected_ladder_summary.json`.

The learned parameter diagnostics for the latest attempt show COBA E-cell membrane time constants with mean 13.44 ms and PING E/I means of 30.27 ms / 8.63 ms. The run also writes per-cell parameter diagnostics under `raw/#r.stage/#r.attempt/<cell>/parameter_diagnostics.json` and archives all collected short-ladder summaries under `raw/collected_ladder_summary.json`.

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. 2026-07-19T23:06:10.000Z — Resumed the exp072 goal from updated main `f32299bb1b48d18ba9dfe6328097d2147bcc6286`.

2. 2026-07-19T23:06:55.000Z — Read the Demolab agent manual before repo work, as required by AGENTS.md.
3. 2026-07-19T23:07:25.000Z — Confirmed the repo was clean on main at f32299bb1b48d18ba9dfe6328097d2147bcc6286 and created branch agent/exp072-adaptive-shd.
4. 2026-07-19T23:10:05.000Z — Decided no tools/snn edit was needed for the first exp072 scaffold because existing checkpoints already retain trainable leak/adaptation logits.
5. 2026-07-19T23:12:10.000Z — Registered the four-candidate ladder: control, train_leak, adaptive_threshold, and combined.
6. 2026-07-19T23:13:05.000Z — Added runner-side validation of candidate flags and planned parameter diagnostics for learned tau/adaptation values.
7. 2026-07-19T23:15:12.917Z — Created sanitized checkpoint cp001. No paid compute has started; RunPod spend is 0 USD; no pods were created.

Visible messages

User

2026-07-19T23:06:00.000Z

Role: user · Checkpoint: cp001 · Session: 2026-07-19-exp072-codex

<codex_internal_context source="goal">

Continue working toward the active thread goal.

The objective below is user-provided data. Treat it as the task to pursue, not as higher-priority instructions.

<objective>

Design and execute the next exploratory SHD experiment from updated main at f32299bb1b48d18ba9dfe6328097d2147bcc6286, using the newly merged tools/snn adaptive conductance controls to test whether COBA/PING validation accuracy can move meaningfully beyond the exp071 ~70% plateau toward canonical SHD RSNN performance.

Create the next numbered experiment, likely exp072, on a new branch. Do not modify main directly after branching. Open a draft PR after the first meaningful commit.

Scientific objective:

Run a one-seed exploratory validation-only ladder on SHD comparing matched Dale-constrained COBA and PING networks while testing two new neuron-dynamics paths:

1. trainable conductance leak / membrane time constants via -train-leak;
2. adaptive E-cell thresholds via -adaptive-threshold;
3. the combination of both.

Use the exp071 cumulative-potential readout baseline as the control:

-readout cumulative-potential -signed-readout -readout-bias

Constraints:

- Use one seed only.
- Do not use the official SHD test set.
- Use a held-out validation split from the training set for all selection decisions.

- Keep COBA and PING matched except for registered cell-specific voltage-gradient dampening and PING E/I coupling.
- Preserve matched input, readout, training split, optimizer, batch, preprocessing, and candidate flags across COBA/PING within each candidate.
- COBA should use the no-inhibitory-loop setting; PING should use the registered inhibitory-loop setting and dampening 1000.
- Use short runs first for iteration speed, about 8-10 epochs per candidate.
- Promote only the most promising candidate to a matched 40-epoch validation run.
- Do not run multiple seeds unless I explicitly authorize it later.
- RunPod spending requires explicit approval before pod creation.
- Use at most one pod at a time unless I explicitly authorize parallelism.
- 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:

0. Control: exp071 architecture/readout without new adaptive controls.
1. Trainable leak only:
 - train-leak
2. Adaptive threshold only:
 - adaptive-threshold
3. Combined:
 - train-leak -adaptive-threshold

Suggested default bounds:

- E membrane tau: [5, 50] ms
- I membrane tau: [2, 20] ms
- adaptive threshold tau: [50, 500] ms
- adaptive threshold initial strength: 1.0 mV
- adaptive threshold max strength: 20.0 mV

Deliverables:

- Produce artifacts/data/exp072 with numbers.json, provenance, reproducer, training curves, firing-rate diagnostics, parameter diagnostics for learned tau/adaptation values, and matched input/E/I rasters.
 - Write writings/exp072.typ as the canonical cold-readable experiment report.
 - Include this goal prompt below the abstract.
 - Append timestamped activity-log/thread checkpoints in the experiment appendix.
 - Record scientific decisions, failures, anomalies, costs, commits, results, and pending work.
 - Sanitize any publishable logs before committing.
 - Build and validate with focused checks plus demolab build.
 - Commit meaningful attempts, including killed attempts, with clear messages.
 - Push the branch and update the PR with evidence.
 - Finish at the human review gate with a concise evidence summary, exact compute spend, validation performed, links to the rendered exp072 file and PR, and a recommendation for the next move toward 90% SHD.
- </objective>

Continuation behavior:

- This goal persists across turns. Ending this turn does not require shrinking the objective to what fits now.

- Keep the full objective intact. If it cannot be finished now, make concrete progress toward the real requested end state, leave the goal active, and do not redefine success around a smaller or easier task.
- Temporary rough edges are acceptable while the work is moving in the right direction. Completion still requires the requested end state to be true and verified.

Budget:

- Tokens used: 0
- Token budget: none
- Tokens remaining: unbounded

Work from evidence:

Use the current worktree and external state as authoritative. Previous conversation context can help locate relevant work, but inspect the current state before relying on it. Improve, replace, or remove existing work as needed to satisfy the actual objective.

Progress visibility:

If `update_plan` is available and the next work is meaningfully multi-step, use it to show a concise plan tied to the real objective. Keep the plan current as steps complete or the next best action changes. Skip planning overhead for trivial one-step progress, and do not treat a plan update as a substitute for doing the work.

Fidelity:

- Optimize each turn for movement toward the requested end state, not for the smallest stable-looking subset or easiest passing change.
- Do not substitute a narrower, safer, smaller, merely compatible, or easier-to-test solution because it is more likely to pass current tests.
- Treat alignment as movement toward the requested end state. An edit is aligned only if it makes the requested final state more true; useful-looking behavior that preserves a different end state is misaligned.

Completion audit:

Before deciding that the goal is achieved, treat completion as unproven and verify it against the actual current state:

- Derive concrete requirements from the objective and any referenced files, plans, specifications, issues, or user instructions.
- Preserve the original scope; do not redefine success around the work that already exists.
- For every explicit requirement, numbered item, named artifact, command, test, gate, invariant, and deliverable, identify the authoritative evidence that would prove it, then inspect the relevant current-state sources: files, command output, test results, PR state, rendered artifacts, runtime behavior, or other authoritative evidence.
- For each item, determine whether the evidence proves completion, contradicts completion, shows incomplete work, is too weak or indirect to verify completion, or is missing.
- Match the verification scope to the requirement's scope; do not use a narrow check to support a broad claim.
- Treat tests, manifests, verifiers, green checks, and search results as evidence only after confirming they cover the relevant requirement.
- Treat uncertain or indirect evidence as not achieved; gather stronger evidence or continue the work.

Do not rely on intent, partial progress, memory of earlier work, or a plausible final answer as proof of completion. Marking the goal complete is a claim that the full objective has been finished and can withstand requirement-by-requirement scrutiny. Only mark the goal achieved when current evidence proves every requirement has been

satisfied and no required work remains. If the evidence is incomplete, weak, indirect, merely consistent with completion, or leaves any requirement missing, incomplete, or unverified, keep working instead of marking the goal complete. If the objective is achieved, call `update_goal` with status "complete" so usage accounting is preserved. If the achieved goal has a token budget, report the final consumed token budget to the user after `update_goal` succeeds.

Blocked audit:

- Do not call `update_goal` with status "blocked" the first time a blocker appears.
- Only use status "blocked" when the same blocking condition has repeated for at least three consecutive goal turns, counting the original/user-triggered turn and any automatic goal continuations.
- If the user resumes a goal that was previously marked "blocked", treat the resumed run as a fresh blocked audit. If the same blocking condition then repeats for at least three consecutive resumed goal turns, call `update_goal` with status "blocked" again.
- Use status "blocked" only when you are truly at an impasse and cannot make meaningful progress without user input or an external-state change.
- Once the blocked threshold is satisfied, do not keep reporting that you are still blocked while leaving the goal active; call `update_goal` with status "blocked".
- Never use status "blocked" merely because the work is hard, slow, uncertain, incomplete, or would benefit from clarification.

Do not call `update_goal` unless the goal is complete or the strict blocked audit above is satisfied. Do not mark a goal complete merely because the budget is nearly exhausted or because you are stopping work.

</codex_internal_context>

Assistant

2026-07-19T23:06:10.000Z

Role: assistant · Checkpoint: cp001 · Session: 2026-07-19-exp072-codex

I'll pick this up from the actual repo state first: confirm we're on updated `main`, read the Demolab runbook as required, then branch exp072 and build the smallest useful implementation/checkpoint before any compute.

Assistant

2026-07-19T23:06:25.000Z

Role: assistant · Checkpoint: cp001 · Session: 2026-07-19-exp072-codex

I'm going to use the GitHub publishing skill when we get to branch/PR work, since this task explicitly includes pushing and opening/updating a PR. First I'm doing the local repo/runbook checks.

Assistant

2026-07-19T23:06:55.000Z

Role: assistant · Checkpoint: cp001 · Session: 2026-07-19-exp072-codex

Runbook check is done. Next I'm reading the repo's current state and the GitHub publishing instructions before touching branches/PRs.

Assistant

2026-07-19T23:07:25.000Z

Role: assistant · Checkpoint: cp001 · Session: 2026-07-19-exp072-codex

State is clean on `main` at the intended merge commit. I'm going to branch now, then scaffold exp072 against the existing exp071 runner/report rather than inventing a new pathway.

Assistant

2026-07-19T23:08:30.000Z

Role: assistant · Checkpoint: cp001 · Session: 2026-07-19-exp072-codex

The experiment runner can mostly reuse exp069/071. One thing I'm adding deliberately is parameter diagnostics: exp072 needs learned leak/adaptation values in artifacts, so I'm checking what the SNN engine already records before deciding whether the runner can extract them without touching `tools/snn`.

Assistant

2026-07-19T23:09:10.000Z

Role: assistant · Checkpoint: cp001 · Session: 2026-07-19-exp072-codex

I'm going to create exp072 as an exp071-derived runner/report, then patch the scientific ladder from "readout/capacity" to "control/leak/adaptation/combined". No `tools/snn` edits should be needed for this first stage.

Assistant

2026-07-19T23:10:05.000Z

Role: assistant · Checkpoint: cp001 · Session: 2026-07-19-exp072-codex

Good: the engine already stores the trainable logits in checkpoints/weight dumps, so exp072 can extract diagnostics from saved weights in the experiment runner. I'm patching that into the runner now.

Assistant

2026-07-19T23:11:10.000Z

Role: assistant · Checkpoint: cp001 · Session: 2026-07-19-exp072-codex

Small correction in-flight: the first broad patch tried to move too much at once. I'm splitting it into boring little patches now—the reliable way, less heroic, fewer gremlins.

Assistant

2026-07-19T23:12:10.000Z

Role: assistant · Checkpoint: cp001 · Session: 2026-07-19-exp072-codex

Candidate registry is now the four planned conditions: control, trainable leak, adaptive threshold, combined. Next patch wires the flags and validation gates.

Assistant

2026-07-19T23:13:05.000Z

Role: assistant · Checkpoint: cp001 · Session: 2026-07-19-exp072-codex

The runner now validates the new flags and will export learned parameter summaries from trained checkpoints. I'm updating the report so it's exp072-specific and doesn't expect old exp071 checkpoint JSONs.

Assistant

2026-07-19T23:14:50.000Z

Role: assistant · Checkpoint: cp001 · Session: 2026-07-19-exp072-codex

I'm creating the first activity checkpoint now. I'll keep the committed log sanitized and compact, and store the raw visible transcript copy outside the published artifact path with only its hash prefix referenced in the article metadata.

Assistant

2026-07-19T23:15:12.917Z

Role: assistant · Checkpoint: cp001 · Session: 2026-07-19-exp072-codex

Sanitization check: no credentials, tokens, SSH material, IP addresses, environment dumps, or private infra details have appeared in the visible text I'm about to publish. The private raw copy will still stay out of the article and out of git; only a SHA-256 prefix goes into metadata.

Checkpoint cp002

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

Decisions, actions, and pending work

1. 2026-07-19T23:16:40.000Z — Focused validation passed: python compile and Ruff for experiments/exp072.py.
2. 2026-07-19T23:17:40.000Z — Demolab build initially stubbed exp072 because artifact regeneration removed the activity checkpoint; fixed by restoring activity after publication and recording the failure.
3. 2026-07-19T23:18:35.000Z — Demolab build then passed with exp072 included.
4. 2026-07-19T23:23:35.000Z — Control local smoke passed: COBA 5.47%, PING 10.16%, both finite and active.
5. 2026-07-19T23:28:05.000Z — Trainable-leak local smoke passed: COBA 5.47%, PING 11.72%, both finite and active.
6. 2026-07-19T23:32:55.000Z — Adaptive-threshold local smoke passed: COBA 5.47%, PING 10.94%, both finite and active.
7. 2026-07-19T23:36:55.000Z — Combined local smoke passed: COBA 3.91%, PING 10.16%, both finite and active.
8. 2026-07-19T23:37:35.000Z — Refreshed pre-result artifacts so numbers.json records the all-candidate smoke ladder.
9. 2026-07-19T23:37:57.083Z — No paid compute has started; RunPod spend remains 0 USD; no pods were created.

Visible messages added

Assistant

2026-07-19T23:15:30.000Z

Role: assistant · Checkpoint: cp002 · Session: 2026-07-19-exp072-codex

Private raw checkpoint copy is written outside the repo. I'm hashing it now, then I'll add the sanitized JSON checkpoint under `artifacts/data/exp072/activity/` so the article can render.

Assistant

2026-07-19T23:15:45.000Z

Role: assistant · Checkpoint: cp002 · Session: 2026-07-19-exp072-codex

Checkpoint hash prefix is `a313f4e2bed6`. I'm committing only sanitized visible messages and a ledger of decisions/actions—not the private raw file or any tool payloads.

Assistant

2026-07-19T23:16:10.000Z

Role: assistant · Checkpoint: cp002 · Session: 2026-07-19-exp072-codex

Sanitized checkpoint JSON is in place. Next: focused code validation and generation of the pre-result `artifacts/data/exp072/numbers.json` so the report can build.

Assistant

2026-07-19T23:16:40.000Z

Role: assistant · Checkpoint: cp002 · Session: 2026-07-19-exp072-codex

Python compile and Ruff passed for the new runner. I'm publishing the pre-result artifact now; this is local/no paid compute.

Assistant

2026-07-19T23:17:10.000Z

Role: assistant · Checkpoint: cp002 · Session: 2026-07-19-exp072-codex

Pre-result artifacts are published under `artifacts/data/exp072`. I'm building the Demolab site/PDFs now to catch Typst/data-contract issues before the checkpoint commit.

Assistant

2026-07-19T23:17:40.000Z

Role: assistant · Checkpoint: cp002 · Session: 2026-07-19-exp072-codex

Build found the expected snag: `publish\_pre\_result()` regenerated `artifacts/data/exp072` and wiped the activity file I'd just added. That's useful evidence; I'll make the runner preserve/copy activity checkpoints during publication so this doesn't recur.

Assistant

2026-07-19T23:18:05.000Z

Role: assistant · Checkpoint: cp002 · Session: 2026-07-19-exp072-codex

I'm treating that as a failed validation attempt and fixing forward. The checkpoint itself stays cp001; the build failure and fix will be recorded in the next meaningful activity checkpoint after the scaffold is stable.

Assistant

2026-07-19T23:18:35.000Z

Role: assistant · Checkpoint: cp002 · Session: 2026-07-19-exp072-codex

Activity checkpoint restored into the generated artifact. I'm rebuilding now; if this passes, I'll move on to the lightweight local smoke stage.

Assistant

2026-07-19T23:19:05.000Z

Role: assistant · Checkpoint: cp002 · Session: 2026-07-19-exp072-codex

Demolab build now passes with exp072 included. I'm starting the local 128/128 two-epoch smoke ladder next, one candidate at a time and still zero paid compute.

Assistant

2026-07-19T23:19:50.000Z

Role: assistant · Checkpoint: cp002 · Session: 2026-07-19-exp072-codex

Control smoke is running; first COBA-style cell is finite so far. I'll let this one finish before starting the adaptive candidates so any failure is easy to attribute.

Assistant

2026-07-19T23:20:20.000Z

Role: assistant · Checkpoint: cp002 · Session: 2026-07-19-exp072-codex

Still in the first local smoke, evaluation phase. This host is predictably slow for 1000 ms SHD even at 128 samples, but we're making progress and not spending cloud money.

Assistant

2026-07-19T23:20:50.000Z

Role: assistant · Checkpoint: cp002 · Session: 2026-07-19-exp072-codex

Control COBA-style smoke epoch 1 finished at finite loss and nonzero E activity; epoch 2 is underway. After this runner will do PING for the same candidate.

Assistant

2026-07-19T23:21:50.000Z

Role: assistant · Checkpoint: cp002 · Session: 2026-07-19-exp072-codex

COBA-style control smoke completed cleanly in about 2 minutes with active firing. PING-style control is now training and also active at initialization.

Assistant

2026-07-19T23:22:20.000Z

Role: assistant · Checkpoint: cp002 · Session: 2026-07-19-exp072-codex

PING control smoke is finite through epoch 1, with low-but-nonzero E and active I firing. Good enough for plumbing so far; waiting for epoch 2/rasters.

Assistant

2026-07-19T23:23:35.000Z

Role: assistant · Checkpoint: cp002 · Session: 2026-07-19-exp072-codex

Control local smoke passed: COBA 5.47%, PING 10.16%, both finite and active. The runner printed the full split provenance noisily, but that's just stdout; committed artifacts keep the provenance in JSON. I'm going to smoke the three new-control candidates too before the first PR checkpoint, because that is the best early guard against a bad ladder.

Assistant

2026-07-19T23:24:20.000Z

Role: assistant · Checkpoint: cp002 · Session: 2026-07-19-exp072-codex

Trainable-leak smoke is underway. Parameter count increased as expected, so the flag is reaching the model.

Assistant

2026-07-19T23:25:25.000Z

Role: assistant · Checkpoint: cp002 · Session: 2026-07-19-exp072-codex

Trainable-leak COBA-style cell is finite and active; it's near the end of epoch 2.

Assistant

2026-07-19T23:26:00.000Z

Role: assistant · Checkpoint: cp002 · Session: 2026-07-19-exp072-codex

Trainable-leak COBA-style smoke passed. PING-style trainable-leak smoke has started, also with the expected larger parameter count.

Assistant

2026-07-19T23:26:30.000Z

Role: assistant · Checkpoint: cp002 · Session: 2026-07-19-exp072-codex

PING trainable-leak is finite through the training batches of epoch 1; waiting on validation/epoch 2.

Assistant

2026-07-19T23:27:00.000Z

Role: assistant · Checkpoint: cp002 · Session: 2026-07-19-exp072-codex

PING trainable-leak epoch 1 passed: finite, active E/I, no obvious smoke failure.

Assistant

2026-07-19T23:27:30.000Z

Role: assistant · Checkpoint: cp002 · Session: 2026-07-19-exp072-codex

Trainable-leak PING is at final validation; this candidate is almost done.

Assistant

2026-07-19T23:28:05.000Z

Role: assistant · Checkpoint: cp002 · Session: 2026-07-19-exp072-codex

Trainable-leak smoke passed too: COBA 5.47%, PING 11.72%, both finite/active. Next is adaptive-threshold smoke.

Assistant

2026-07-19T23:28:45.000Z

Role: assistant · Checkpoint: cp002 · Session: 2026-07-19-exp072-codex

Adaptive-threshold smoke is running; parameter count increased as expected and initial E activity is a bit lower than control, which is plausible for adaptation.

Assistant

2026-07-19T23:29:15.000Z

Role: assistant · Checkpoint: cp002 · Session: 2026-07-19-exp072-codex

Adaptive-threshold COBA-style epoch 1 is finite; validation completed, epoch summary imminent.

Assistant

2026-07-19T23:29:45.000Z

Role: assistant · Checkpoint: cp002 · Session: 2026-07-19-exp072-codex

Adaptive-threshold COBA-style is finite/active through epoch 1, training epoch 2 now.

Assistant

2026-07-19T23:30:45.000Z

Role: assistant · Checkpoint: cp002 · Session: 2026-07-19-exp072-codex

Adaptive-threshold COBA-style completed its epochs finite/active; waiting for the runner to finish checkpointing and start PING.

Assistant

2026-07-19T23:31:15.000Z

Role: assistant · Checkpoint: cp002 · Session: 2026-07-19-exp072-codex

Adaptive-threshold COBA-style passed; PING-style adaptive threshold is now training and active.

Assistant

2026-07-19T23:31:45.000Z

Role: assistant · Checkpoint: cp002 · Session: 2026-07-19-exp072-codex

Adaptive-threshold PING epoch 1 passed finite with active E/I.

Assistant

2026-07-19T23:32:15.000Z

Role: assistant · Checkpoint: cp002 · Session: 2026-07-19-exp072-codex

Adaptive-threshold PING is in epoch 2. Still finite; no smoke kill signal.

Assistant

2026-07-19T23:32:55.000Z

Role: assistant · Checkpoint: cp002 · Session: 2026-07-19-exp072-codex

Adaptive-threshold smoke passed: COBA 5.47%, PING 10.94%, finite/active. Last local smoke is the combined candidate.

Assistant

2026-07-19T23:33:25.000Z

Role: assistant · Checkpoint: cp002 · Session: 2026-07-19-exp072-codex

Combined smoke started. Parameter count is higher again, matching both flag families being present.

Assistant

2026-07-19T23:33:55.000Z

Role: assistant · Checkpoint: cp002 · Session: 2026-07-19-exp072-codex

Combined COBA-style epoch 1 is finite; validation is underway.

Assistant

2026-07-19T23:34:25.000Z

Role: assistant · Checkpoint: cp002 · Session: 2026-07-19-exp072-codex

Combined COBA-style is active and finite through epoch 1; epoch 2 is almost through training.

Assistant

2026-07-19T23:34:55.000Z

Role: assistant · Checkpoint: cp002 · Session: 2026-07-19-exp072-codex

Combined COBA-style completed finite/active; waiting for checkpointing and then PING.

Assistant

2026-07-19T23:35:25.000Z

Role: assistant · Checkpoint: cp002 · Session: 2026-07-19-exp072-codex

Combined COBA-style passed; PING-style combined is now training.

Assistant

2026-07-19T23:35:55.000Z

Role: assistant · Checkpoint: cp002 · Session: 2026-07-19-exp072-codex

Combined PING epoch 1 passed finite with active E/I; final epoch underway.

Assistant

2026-07-19T23:36:25.000Z

Role: assistant · Checkpoint: cp002 · Session: 2026-07-19-exp072-codex

Combined PING is at final validation. After it exits I'll refresh the pre-result artifact to record the latest local smoke state and rebuild.

Assistant

2026-07-19T23:36:55.000Z

Role: assistant · Checkpoint: cp002 · Session: 2026-07-19-exp072-codex

All four local smokes passed. I'm patching the pre-result publisher to record the whole local smoke ladder, not just whichever `EXP072\_ATTEMPT` was selected last.

Assistant

2026-07-19T23:37:15.000Z

Role: assistant · Checkpoint: cp002 · Session: 2026-07-19-exp072-codex

The publisher now collects all four local smoke summaries. I'm adjusting the article wording to recognize the full-ladder smoke state.

Assistant

2026-07-19T23:37:35.000Z

Role: assistant · Checkpoint: cp002 · Session: 2026-07-19-exp072-codex

Now I'll refresh `artifacts/data/exp072` with the all-smoke pre-result state. This will regenerate the artifact directory, so I'll restore the activity checkpoint afterward before building.

Assistant

2026-07-19T23:37:50.000Z

Role: assistant · Checkpoint: cp002 · Session: 2026-07-19-exp072-codex

Pre-result artifact refreshed. I'm adding cp002 now to record the smoke-ladder milestone and the one validation hiccup/fix, then the article will include both cp001 and cp002.

Assistant

2026-07-19T23:37:57.083Z

Role: assistant · Checkpoint: cp002 · Session: 2026-07-19-exp072-codex

Sanitization for cp002: again, no credentials/tokens/SSH material/IPs/environment dumps/private infra details in the visible text I'm publishing; tool payloads stay out.

Checkpoint cp003

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

Decisions, actions, and pending work

1. 2026-07-19T23:38:45.000Z — Rebuilt successfully after adding cp002 to the article.

2. 2026-07-19T23:39:25.000Z — Restored tracked PDF rebuild noise and quarantined unrelated untracked generated PDFs outside the repo; retained exp072.pdf.
3. 2026-07-19T23:41:20.000Z — Committed first exp072 checkpoint as c580b7c: Add exp072 adaptive SHD scaffold.
4. 2026-07-19T23:41:45.000Z — Pushed branch agent/exp072-adaptive-shd to origin.
5. 2026-07-19T23:42:00.000Z — Opened draft PR #57: <https://github.com/eoinmurray/pinglab/pull/57>.
6. 2026-07-19T23:42:15.000Z — gh pr edit failed because of a deprecated Projects GraphQL field, leaving the body empty; fixed by patching the PR through the REST API.
7. 2026-07-19T23:42:30.000Z — PR #57 body now records validation evidence, zero spend, and the next RunPod approval gate.

Visible messages added

Assistant

2026-07-19T23:38:20.000Z

Role: assistant · Checkpoint: cp003 · Session: 2026-07-19-exp072-codex

cp002 is written with hash prefix `bec9f91c83e2`. I'm wiring it into the article and rebuilding.

Assistant

2026-07-19T23:39:00.000Z

Role: assistant · Checkpoint: cp003 · Session: 2026-07-19-exp072-codex

Focused validation is green: compile, Ruff, and Demolab build all pass. I'm cleaning the generated PDF noise now so the first commit contains only exp072's intended files/artifacts.

Assistant

2026-07-19T23:39:25.000Z

Role: assistant · Checkpoint: cp003 · Session: 2026-07-19-exp072-codex

The build rewrote many existing PDFs, which is normal here but not part of this PR. I'm restoring tracked PDFs and keeping only the new exp072 PDF plus exp072 data/report/runner.

Assistant

2026-07-19T23:39:45.000Z

Role: assistant · Checkpoint: cp003 · Session: 2026-07-19-exp072-codex

The shell guard blocked my explicit `rm -f` cleanup, fair enough. I'll use a safer move-to-quarantine approach for the unrelated untracked PDFs.

Assistant

2026-07-19T23:40:05.000Z

Role: assistant · Checkpoint: cp003 · Session: 2026-07-19-exp072-codex

Tracked PDFs are restored. Now I'm moving the unrelated untracked build PDFs out of the repo, leaving `exp072.pdf` intact.

Assistant

2026-07-19T23:40:30.000Z

Role: assistant · Checkpoint: cp003 · Session: 2026-07-19-exp072-codex

Worktree scope is clean: only new exp072 runner, article, data, and PDF are untracked. I'm doing a quick artifact sanity read before committing.

Assistant

2026-07-19T23:41:00.000Z

Role: assistant · Checkpoint: cp003 · Session: 2026-07-19-exp072-codex

Artifact provenance looks sane: local host, dirty patch capturing the new exp072 scaffold, zero spend, all-smoke ladder recorded. I'm staging only the exp072 files now and making the first checkpoint commit.

Assistant

2026-07-19T23:41:20.000Z

Role: assistant · Checkpoint: cp003 · Session: 2026-07-19-exp072-codex

First checkpoint commit is in: `c580b7c` ("Add exp072 adaptive SHD scaffold"). Next I'm pushing the branch and opening the draft PR, per the goal.

Assistant

2026-07-19T23:41:45.000Z

Role: assistant · Checkpoint: cp003 · Session: 2026-07-19-exp072-codex

Branch is pushed. I'm opening a draft PR now with the validation evidence and zero-spend status.

Assistant

2026-07-19T23:42:00.000Z

Role: assistant · Checkpoint: cp003 · Session: 2026-07-19-exp072-codex

Draft PR opened as #57. Minor process wart: my first body-file helper tried plain `python` (not allowed/available here), so I'm checking/updating the PR body with `uv run python` now.

Assistant

2026-07-19T23:42:15.000Z

Role: assistant · Checkpoint: cp003 · Session: 2026-07-19-exp072-codex

`gh pr edit` hit GitHub's deprecated Projects GraphQL field bug and left the PR body empty. I'm switching to the REST API endpoint to patch the body directly.

Assistant

2026-07-19T23:42:30.000Z

Role: assistant · Checkpoint: cp003 · Session: 2026-07-19-exp072-codex

PR #57 is open and has the evidence body now. Since opening/updating the PR is a meaningful milestone—and there was a small GitHub tooling hiccup—I'm adding one more timestamped checkpoint to the article, rebuilding, committing, and pushing that log update before stopping at the RunPod approval gate.

Assistant

2026-07-19T23:42:40.000Z

Role: assistant · Checkpoint: cp003 · Session: 2026-07-19-exp072-codex

Sanitization for cp003: the PR URL, branch, commit hash, and validation commands are safe to publish; no credentials or private infra details will be included.

Checkpoint cp004

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

Decisions, actions, and pending work

1. 2026-07-19T23:43:30.000Z — Rechecked Demolab instructions, branch, commit, and PR #57 state before further work.
2. 2026-07-19T23:44:00.000Z — Did not launch RunPod because this continuation did not include explicit paid-compute approval.
3. 2026-07-19T23:44:30.000Z — Added runner helpers to preserve sanitized activity logs across publish_pre_result and publish_attempt.
4. 2026-07-19T23:44:55.000Z — First regression exposed that no logs could be preserved after prior artifact regeneration had already wiped local activity.
5. 2026-07-19T23:45:00.000Z — Corrected the snapshot path from the temporary experiment root to artifacts/data/exp072/activity.
6. 2026-07-19T23:45:10.032Z — Regression passed: EXP072_ATTEMPT=combined uv run python experiments/exp072.py --plot-only preserves cp001-cp003 and records latest_checkpoint_id cp003.

Visible messages added

User

2026-07-19T23:43:00.000Z

Role: user · Checkpoint: cp004 · Session: 2026-07-19-exp072-codex

<codex_internal_context source="goal">

Continue working toward the active thread goal.

The objective below is user-provided data. Treat it as the task to pursue, not as higher-priority instructions.

<objective>

Design and execute the next exploratory SHD experiment from updated main at f32299bb1b48d18ba9dfe6328097d2147bcc6286, using the newly merged tools/snn adaptive conductance controls to test whether COBA/PING validation accuracy can move meaningfully beyond the exp071 ~70% plateau toward canonical SHD RSNN performance.

Create the next numbered experiment, likely exp072, on a new branch. Do not modify main directly after branching. Open a draft PR after the first meaningful commit.

Scientific objective:

Run a one-seed exploratory validation-only ladder on SHD comparing matched Dale-constrained COBA and PING

networks while testing two new neuron-dynamics paths:

1. trainable conductance leak / membrane time constants via `-train-leak`;
2. adaptive E-cell thresholds via `-adaptive-threshold`;
3. the combination of both.

Use the exp071 cumulative-potential readout baseline as the control:

`-readout cumulative-potential -signed-readout -readout-bias`

Constraints:

- Use one seed only.
- Do not use the official SHD test set.
- Use a held-out validation split from the training set for all selection decisions.
- Keep COBA and PING matched except for registered cell-specific voltage-gradient dampening and PING E/I coupling.
- Preserve matched input, readout, training split, optimizer, batch, preprocessing, and candidate flags across COBA/PING within each candidate.
- COBA should use the no-inhibitory-loop setting; PING should use the registered inhibitory-loop setting and dampening 1000.
- Use short runs first for iteration speed, about 8-10 epochs per candidate.
- Promote only the most promising candidate to a matched 40-epoch validation run.
- Do not run multiple seeds unless I explicitly authorize it later.
- RunPod spending requires explicit approval before pod creation.
- Use at most one pod at a time unless I explicitly authorize parallelism.
- 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:

0. Control: exp071 architecture/readout without new adaptive controls.

1. Trainable leak only:

`-train-leak`

2. Adaptive threshold only:

`-adaptive-threshold`

3. Combined:

`-train-leak -adaptive-threshold`

Suggested default bounds:

- E membrane tau: [5, 50] ms
- I membrane tau: [2, 20] ms
- adaptive threshold tau: [50, 500] ms
- adaptive threshold initial strength: 1.0 mV
- adaptive threshold max strength: 20.0 mV

Deliverables:

- Produce artifacts/data/exp072 with numbers.json, provenance, reproducer, training curves, firing-rate diagnostics, parameter diagnostics for learned tau/adaptation values, and matched input/E/I rasters.
- Write writings/exp072.typ as the canonical cold-readable experiment report.

- Include this goal prompt below the abstract.
- Append timestamped activity-log/thread checkpoints in the experiment appendix.
- Record scientific decisions, failures, anomalies, costs, commits, results, and pending work.
- Sanitize any publishable logs before committing.
- Build and validate with focused checks plus demolab build.
- Commit meaningful attempts, including killed attempts, with clear messages.
- Push the branch and update the PR with evidence.
- Finish at the human review gate with a concise evidence summary, exact compute spend, validation performed, links to the rendered exp072 file and PR, and a recommendation for the next move toward 90% SHD.

</objective>

</codex_internal_context>

Assistant

2026-07-19T23:43:10.000Z

Role: assistant · Checkpoint: cp004 · Session: 2026-07-19-exp072-codex

I'm resuming from the branch/PR state and will re-run the Demolab manual check first. The next scientific step is the paid RunPod short ladder, but I'll verify local/PR state before deciding whether I have enough authority to launch it.

Assistant

2026-07-19T23:43:30.000Z

Role: assistant · Checkpoint: cp004 · Session: 2026-07-19-exp072-codex

I'll use the GitHub publishing workflow only for PR updates/comments after I've generated new evidence; first I'm checking the current local state.

Assistant

2026-07-19T23:44:00.000Z

Role: assistant · Checkpoint: cp004 · Session: 2026-07-19-exp072-codex

State is clean on PR #57 at `49a684e`. I don't have explicit paid RunPod approval in this turn, so I'm going to use this turn to harden the runner/report around activity checkpoints and RunPod collection before spending anything.

Assistant

2026-07-19T23:44:15.000Z

Role: assistant · Checkpoint: cp004 · Session: 2026-07-19-exp072-codex

One issue from the previous checkpoint is worth fixing properly: `--plot-only` regenerates `artifacts/data/exp072`, so activity logs need to survive publication rather than being hand-restored. I'm patching that into exp072 now.

Assistant

2026-07-19T23:44:30.000Z

Role: assistant · Checkpoint: cp004 · Session: 2026-07-19-exp072-codex

I found the exact weak point: ``publish_pre_result()`` and ``publish_attempt()`` don't preserve existing ``activity/messages_cp*.json``. I'm adding a small helper that snapshots those files before the Demolab publish context and copies them back into the new artifact.

Assistant

2026-07-19T23:44:45.000Z

Role: assistant · Checkpoint: cp004 · Session: 2026-07-19-exp072-codex

The preservation helper is patched. I'm testing it by running ``-plot-only`` locally; this is the same path that previously wiped the checkpoint files, so it's the right regression check.

Assistant

2026-07-19T23:44:55.000Z

Role: assistant · Checkpoint: cp004 · Session: 2026-07-19-exp072-codex

Regression caught a second nuance: because the previous publish had already wiped local activity before this test started, there were no files for the helper to preserve. I'm restoring the committed checkpoints from git, then rerunning the same regression.

Assistant

2026-07-19T23:45:00.000Z

Role: assistant · Checkpoint: cp004 · Session: 2026-07-19-exp072-codex

Aha, ``ARTIFACTS`` in this runner points at the temp experiment root, not the committed published artifact directory. That's why the helper saw nothing. I'm correcting it to snapshot from ``artifacts/data/exp072/activity``.

Assistant

2026-07-19T23:45:05.000Z

Role: assistant · Checkpoint: cp004 · Session: 2026-07-19-exp072-codex

Path fixed. I'm restoring the three checkpoint files from the last commit and rerunning the preservation regression.

Assistant

2026-07-19T23:45:10.032Z

Role: assistant · Checkpoint: cp004 · Session: 2026-07-19-exp072-codex

The regression now passes: ``-plot-only`` preserves cp001–cp003 and records ``latest_checkpoint_id: cp003`` in ``numbers.json``. I'm adding cp004 for this hardening fix, then I'll rebuild, commit, push, and leave the goal at the paid-compute approval gate.

Assistant

2026-07-19T23:45:15.000Z

Role: assistant · Checkpoint: cp004 · Session: 2026-07-19-exp072-codex

Sanitization for cp004: it will include repo state, PR number, validation commands, and the runner fix; no credentials, tokens, SSH material, IPs, environment dumps, private paths, or tool payloads.

Checkpoint cp005

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

Note: this checkpoint records the scientific ledger for the long RunPod monitoring turn, but does not claim a complete verbatim transcript because no raw transcript export was available in the workspace.

Decisions, actions, and pending work

1. 2026-07-20T00:00:00.000Z — User authorized RunPod spending for the whole exp072 execution, including 40-epoch runs if needed, while preserving the one-pod-at-a-time constraint.
2. 2026-07-20T02:20:07.957Z — Completed and published the short control rung: COBA selected 60.17% validation accuracy; PING selected 56.86%. Both were finite and active.
3. 2026-07-20T02:59:09.000Z — Completed and published the short train_leak rung: COBA selected 58.95%; PING selected 57.72%. This was the only modified candidate that improved PING over control in the short ladder.
4. 2026-07-20T03:36:28.000Z — Completed and published the short adaptive_threshold rung: COBA selected 50.61%; PING selected 53.68%. Both were active but weaker than control/train_leak.
5. 2026-07-20T04:08:53.000Z — Completed and published the short combined rung: COBA selected 50.37%; PING selected 50.98%. Both were active but weaker than train_leak.
6. 2026-07-20T04:10:00.000Z — Promoted train_leak to the matched 40-epoch stage because it was the only modification that improved PING while keeping COBA close.
7. 2026-07-20T05:17:12.000Z — Final40 train_leak COBA completed and was collected with no active pod remaining before the next cell launched.
8. 2026-07-20T06:29:29.000Z — Final40 train_leak PING completed, was collected, and the final40 attempt was published. No active pods remained after collection.
9. 2026-07-20T06:30:00.000Z — Final40 train_leak selected 69.49% validation accuracy for COBA at epoch 37 and 71.08% for PING at epoch 40. PING had a small 1.59 percentage-point edge.
10. 2026-07-20T06:33:34.862Z — Cumulative experiment spend ledgers total 4.66 USD, marked timestamp-estimated pending provider billing reconciliation.

Visible milestone messages added

User

2026-07-20T00:00:00.000Z

Role: user · Checkpoint: cp005 · Session: 2026-07-20-exp072-codex

you have runpod permission for whole thing, 40 epoch runs if needed

Assistant

2026-07-20T04:10:00.000Z

Role: assistant · Checkpoint: cp005 · Session: 2026-07-20-exp072-codex

Short ladder readout: control is best by paired average, but `train\_leak` is the only modification that improves the target PING cell while keeping COBA close. I'm going to promote `train\_leak` to 40 epochs: it's the best scientific bet for "can trainable leak move PING upward?" rather than just re-running the control.

Assistant

2026-07-20T06:30:00.000Z

Role: assistant · Checkpoint: cp005 · Session: 2026-07-20-exp072-codex

No active pods. Final40 result: `train\_leak` COBA selected 69.49% at epoch 37; PING selected 71.08% at epoch 40. That's a real small PING edge, but still far below SOTA territory.

Checkpoint cp006

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

Decisions, actions, and pending work

1. 2026-07-20T08:01:22.000Z — Launched exp072 combined final80 COBA after adding and pushing a registered final80 stage.
2. 2026-07-20T08:03:42.000Z — User requested COBA and PING run in parallel; stopped the local sequential monitor without stopping COBA and launched combined final80 PING as a second pod.
3. 2026-07-20T10:22:06.000Z — PING final80 self-terminated while COBA continued.
4. 2026-07-20T11:22:09.000Z — COBA final80 self-terminated; collection fetched both cells and no active pods remained.
5. 2026-07-20T11:23:31.000Z — Collected final80 combined artifacts.
6. 2026-07-20T11:24:00.000Z — Selected final80 combined results: COBA 64.58% at epoch 77; PING 71.69% at epoch 73.
7. 2026-07-20T11:24:30.000Z — Dynamics warnings: COBA saturated/stuck; PING stuck. Combined PING only slightly exceeded train_leak final40 while taking twice the epochs; combined COBA remained worse than train_leak final40.
8. 2026-07-20T11:25:02.841Z — Published final80 combined locally with estimated cumulative spend pending final provider billing reconciliation.

Visible milestone messages added

User

2026-07-20T08:02:00.000Z

Role: user · Checkpoint: cp006 · Session: 2026-07-20-exp072-codex

do ping and cobra in paralell on two pods

Assistant

2026-07-20T11:24:00.000Z

Role: assistant · Checkpoint: cp006 · Session: 2026-07-20-exp072-codex

Result: final80 combined selected COBA 64.58% (epoch 77) and PING 71.69% (epoch 73). PING is only +0.61 pp above leak-only final40, but with stuck warnings and twice the epochs; COBA is worse and saturated/stuck. No active pods.