READ MODE · ANSWER FIRST

Recent Experiments on Slowrun

An interactive, visual explanation of Recent Experiments on Slowrun, from setup through the complete system.

CHEAT SHEET · 4 ESSENTIAL IDEAS

The whole story in 4 lines

Recent Experiments on Slowrun becomes easier to reason about when every stage is connected as one system.

1. Show that main-track exploration was informative, but expensive and mostly negative.
2. Teach the shift from slow single-run learning to fast experimental cadence.
3. Contrast additive ideas with redundant or throughput-damaging ones.
4. Explain why the tying trick helps and why it is only active during the early part of training.

STAGE 1 OF 6

Benchmark Budget

01 BENCHMARK BUDGET

Slowrun begins with one finite 100M-token pool. The benchmark is data-bound rather than compute-bound.

The main track gets one hour on 8xH100, which makes every failed run an expensive experiment.

The tiny track cuts that to 15 minutes, sacrificing runway in exchange for much faster iteration.

The infinite track removes the wall-clock cap, but the post is really about the practical main-vs-tiny tradeoff.

Across all tracks the goal is the same: squeeze more learning from the same limited data.

That setup explains why the author starts on the main track, then pivots once iteration speed becomes the bottleneck.

STAGE 2 OF 6

Main Track Dead Ends

02 MAIN TRACK DEAD ENDS

The author begins on the main track, looking for low-hanging optimizer or schedule wins above the baseline.

Batch schedule plus cautious weight decay is an early miss: validation worsens instead of improving.

Extended warmdown with momentum cooldown also trails, suggesting the decay starts too early and removes useful high-LR time.

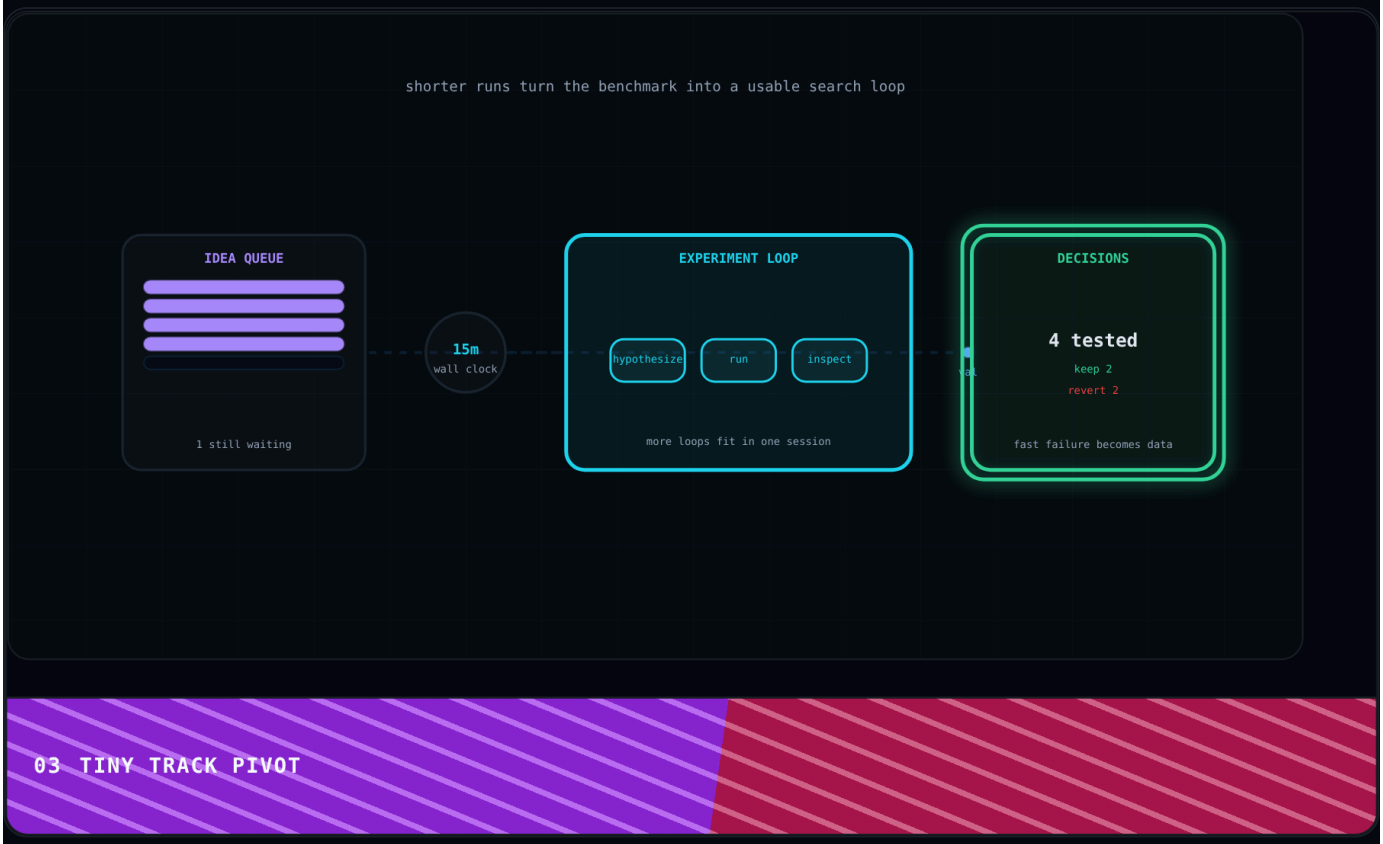
Cosine warmdown is the interesting case: it briefly overtakes baseline around epochs 10 and 11.

But the tail collapses by epoch 12 because the schedule decays too hard toward zero and wastes the final stretch.

By now the lesson is clear: the main track is teaching useful failure modes, but the iteration cost is too high for broad exploration.

STAGE 3 OF 6

Tiny Track Pivot



03 TINY TRACK PIVOT

The tiny track arrives with the same 8xH100 setup, but each run now costs only 15 minutes instead of an hour.

That shorter budget makes the hypothesis queue tractable. More ideas can move from intuition to measurement.

The experimental loop tightens: propose a change, launch a run, inspect validation, and decide quickly.

Failed ideas are no longer catastrophic. They become cheap information that narrows the search.

This faster cadence reveals which changes add genuine inductive bias and which ones merely add complexity.

From here the blog shifts from isolated schedule tweaks to compositional questions about stacking and interaction effects.

STAGE 4 OF 6

Stacking Rules

By the tiny-track phase, the baseline already includes attention gating and other merged ideas. New changes have to justify their place in that stack.

The author notices a pattern: improvements that add inductive bias for information flow are the ones most likely to help.

Examples like U-Net style structure or per-head gating feel additive because they change a different part of the signal path.

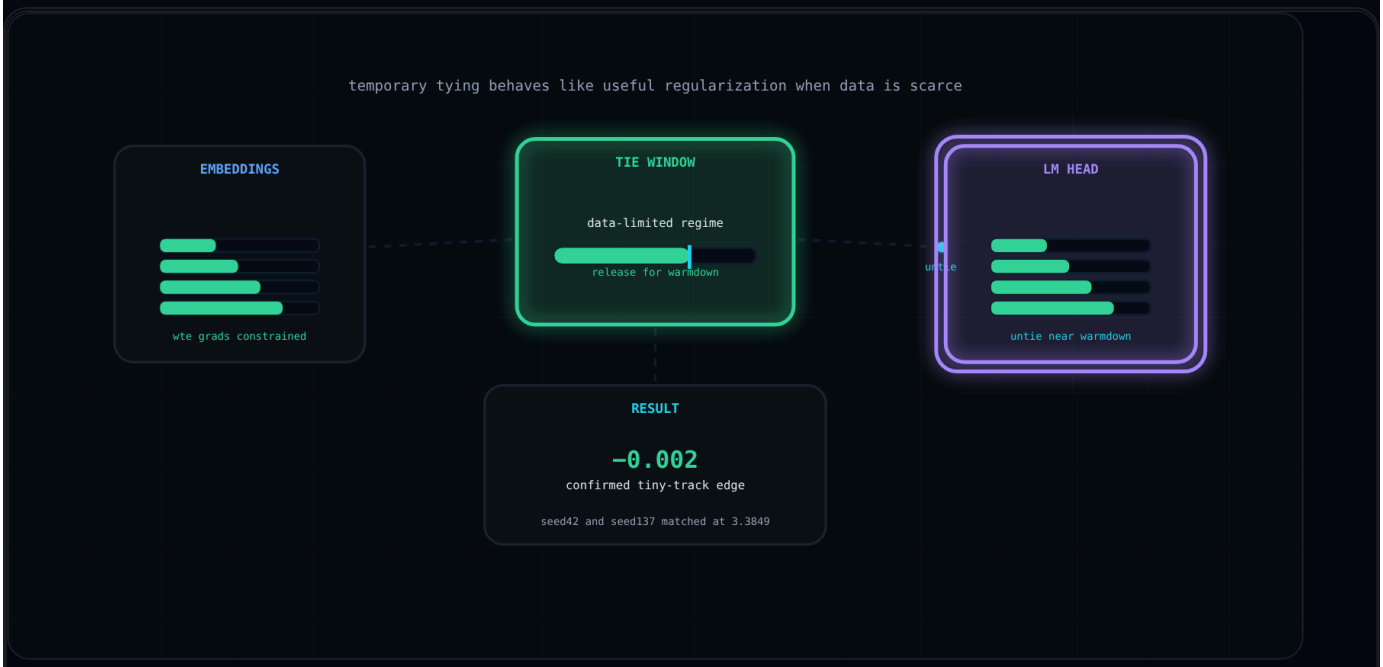
By contrast, SA lambdas and gated skip variants overlap with existing mechanisms and mostly add noise or redundancy.

Hyperc connections are even worse because they also drag MFU, damaging training efficiency on a tight 15-minute budget.

The working heuristic becomes: keep the changes that are orthogonal and information-rich, revert the ones that collide with the current stack.

STAGE 5 OF 6

Head Tying Window



05 HEAD TYING WINDOW

The tiny-track baseline now includes attention gating, and the author tests a new idea: tie embeddings to the LM head early in training.

During the tie window, both endpoints share one parameter body, shrinking the effective freedom of the model.

Embedding gradients are constrained while tied, so the model behaves like a more regularized system during the noisy early stage.

Near warmdown the tie is released, letting the LM head and embeddings diverge again for final specialization.

The measured gain is modest but consistent: train loss improves and the validation result survives a 2-seed variance check.

That makes temporary tying the cleanest confirmed positive result in the post, even though it was not ultimately merged.

STAGE 6 OF 6

Order Matters

06 ORDER MATTERS

The final takeaway is about order. You cannot judge a tweak without looking at the baseline it lands on.

On a thinner stack with only attention gating and LR warmdown, lambda = 1.1 blows up badly instead of helping.

That failure would be easy to over-generalize if you stopped there.

But on a richer stack with half truncated RoPE and token smearing, the very same lambda becomes compatible.

In that new context it helps produce the tiny-track record, which flips the earlier conclusion completely.

So the article closes on a stronger heuristic: changes stack when they are internally orthogonal and introduced in a compatible order.