D2.2 design: anchoring an operation the model infers

D2.2 asks whether Sparse Concept Anchoring works for a concept that is never written down. The model sees a few solved equations that share a hidden operation, and has to infer the op to answer the last one. We anchor one op, `difference`, to the first axis of the residual stream (e_1), and then remove it by editing that axis out.

The pivot explains why we took the op words out of the grammar, and the index lists every experiment along the way.

Where it stands

The recipe of record keeps the anchor on every run we have trained, at little cost to the task. The full edit removes most of `difference`, in proportion to how much each context needed it, and the other ops stay within the selectivity criterion on every run. The edited model sends the mass to the other ops in about the right overall proportions, but context by context it gets only about half of the way to where an ideal predictor would. A model trained without `difference` gets only a little further, so most of that shortfall is in what a model of this size does on such contexts, and the anchor and the edit add a smaller part.

The reference model and the full set of measurements on the recipe of record are done. One piece of work remains before the blog post: a comparison with directions found after training[°]. Everything else is future work.

Directions found after training. Fit a direction for `difference` on the states of the controls, which were trained without the anchor, and edit along it through the same scorer....

The recipe of record

The recipe is `τ_ℓ fixed`, adopted after the *fixed confirmation*°. Its pull is pooled over slices 1 to 3, the outputs of every block but the last, so the recipe does not have to say which block holds the op.

fixed confirmation. `τ_ℓ fixed` at fresh seeds keeps every anchor, removes about as much as the annealed version, and has no late spill; adopted as the recipe of record.

Grammar

Each line of the corpus is one context: three solved examples of one hidden op, then a query under the same op, with a constant `?` where the op word would be (`op1 ? op2 = answer , op1 ? op2 = answer , op1 ? op2 = answer , op1 ? op2 = answer ⏎`). The model infers the op from the examples and completes the query answer.

Task

Seven ops: `mix`, `lighten`, `darken`, `difference`, and three HSV blend modes (`hue-hsv`, `sat-hsv`, `value-hsv`). Answers between grid levels round up or down at random, so a model is scored on the probability it puts on the right answer. Three examples in ten show the answer of another op, drawn at random (replacement op noise), so some contexts leave the op in doubt. Where the two ops agree, the replacement doesn't show.

Model and training

A d64-L4 simplified nGPT with a readout table of its own and attention that stays within one context, trained for 400 epochs on a cosine schedule.

Labels

About one `difference` context in fifty carries a label saying only that it uses `difference`. The model never sees the label, and no other op is labelled.

Pull

On a labelled context, the anchor term (a regularizer) pulls the states at slices 1 to 3 toward e_1, pooled over every slice and position. The pool over positions starts soft and sharpens to a temperature τ_p of 0.1 by epoch 40, and the pool over slices stays at a temperature τ_ℓ of 0.1 throughout. The anchor weight λ_a warms up to 0.1 over the same 40 epochs and eases off at the end. The embedding and block 4 are not pulled.

Anti-subspace term

A light push on every state, labelled or not, away from e_1, so the pull has to earn its alignment. Its weight starts at 0.25 on every slice and anneals by epoch 360 to a hold that falls off with depth, from 0.2 at the embedding to 0.03 at block 4. The high hold on the first slices keeps the color embeddings from standing in for the op.

Edit

Remove a share of the e_1 component (the dose, from a quarter to all of it) from the state at every position and every slice. The states inside a block, between its attention and its MLP, are left as they are (claim d says why).

Scoring

Every run is scored on held-out contexts, and each drop under the edit is net of a baseline: the median drop under the same edit along each of the other 63 coordinate axes of the run. Controls trained without the anchor give the task change°, and check now and then that the baseline still agrees with them.

Task change. The clean score per op, less the mean of the controls....

Task change

The clean score per op, less the mean of the controls. The task gate fails below −0.02 on the mean over ops, or below −0.05 on any one op. Reports up to and including *fixed-confirm* netted against the control at the same seed instead; by the end of training two runs at one seed are about as far apart as runs at different seeds, so a single control adds its own seed scatter.

Two conditions are compared as two samples: the difference of their seed means with a 95% Welch t interval (an interval for a difference of means that does not assume the two samples vary alike), and Fisher's exact test for a difference in rates. At eight runs a side, a percentile bootstrap interval covers about 90% of the time where it claims 95%, which is why *fixed-effects*° moved to the t interval. Runs at the same seed are paired only when they branch from one stored state and are measured soon after, since two runs of one seed under slightly different recipes end about as far apart as runs at different seeds (eng/determinism.md).

fixed-effects. The full scoring suite on `τ_ℓ fixed` ...

Removal

The drop on `difference` under the full edit, as a share of the way to the target null. A run that removes less than half has lost the anchor.

Spill

The largest drop on any other op, at any dose. The selectivity criterion is 0.02.

Landing

The share of the distance to the target null that the full edit covers, context by context.

Write bound

How much e_1 the blocks after an edit put back on the states of the other ops, against what the clean model has there.

Over the write bound

A site (a slice and a position) where the write passes the bound by more than both tolerances: the relative tolerance, a quarter over the bound, and the absolute tolerance, about a degree. The absolute tolerance comes from how much the bound itself varies between runs (write-floor); without it, a turn of a fraction of a degree counts wherever the bound is small. A condition passes when at most one run in eight has any site over. Runs are counted rather than sites, since the sites over on one run tend to come together.

The comparison with this baseline leans on Adam giving the coordinate axes a privileged role: the model parks its syntax tokens on a few axes and leaves the rest quiet, so the other axes make a fair reference. The anchor itself does not need this; it held even with the basis of the optimizer rotated.

The claims

For the narrow domain of color mixing, we want to be able to say five things.

(a) An inferred op can be anchored on every run, without costing the task

The anchor sits on the example answers, where the context shows the most about its op, and its alignment° there rises with the posterior on `difference`. That fits an anchor that holds the inferred op, though it follows the evidence of each example on its own more than the evidence of the whole context.

Alignment. How far a state points along e_1...

Losing the anchor was the largest gap in earlier recipes. The recipe of record kept it on all eight runs at fresh seeds and on all forty partial runs (called early, at epoch 64), which puts the loss rate below about one in fourteen. Seven of the eight fresh runs passed the task gate; the eighth learned two HSV ops late and was still catching up at the end.

(b) Removing the op is graded and selective

A stronger edit takes out more of `difference`, and the other ops are left as they were at every dose. On the recipe of record the full edit removes 0.88 of the way to the target null° on the seed mean (0.83 to 0.91), and no run ends near the spill° criterion or rises toward it late in training. Removal° is complete by three quarters of the full edit, so the full edit has some margin.

Target null. Where a removal should land...

Spill. The largest drop on any other op, at any dose....

Removal. The drop on `difference` under the full edit, as a share of the way to the target null....

The drop on each context follows how much it needed `difference`: the share of the way to the target null is about level across the posterior, which is what the target null predicts, at the scout seeds and at the fresh ones (*fixed-effects*°).

fixed-effects. The full scoring suite on `τ_ℓ fixed`...

The spill of earlier recipes came from a stand-in for the op in the color embedding table, and later from block 1. Holding the anti weight high on the first slices and keeping the pool over slices sharp took both away.

(c) The edited model answers in a predictable way

Once `difference` is removed, the model should answer as an ideal predictor that never knew `difference` but still weighs the other ops by how well they fit the examples: the target null. The edited answers go to the other ops in about the right overall proportions, but within a context they weigh those ops differently, and the full edit covers only about half of the distance.

Sans-diff° shows that half is not far short of what a model of this size reaches. It is the control trained with no loss on `difference` contexts, so it never learns the op, and it comes 0.58 of the way where the edited model comes 0.50. It misses in the same kind of way, sharing the mass among the other ops differently on each context, and differently again at each seed. So most of the shortfall would be there in a model that never knew the op, and the anchor and the edit add a smaller part on top; the shortfall on `lighten` in the pooled confusion matrices of *fixed-effects* is part of what they add.

sans-diff. The control trained with no loss on `difference` contexts lands 0.58 of the way to the target null, against 0.50 for the edited model, and misses in the same kind of way.

(d) The model does not route around the edit

Claims (b) and (c) cover the side-effects on the answers. This one is about the states inside the model. Because we placed the axis, the clean model tells us in advance how much e_1 the states of the other ops carry, and so how far an edit can turn them. The claim is that the blocks after the edit put no more e_1 back on those states than the clean model has there, so the edit stays where we put it. On the recipe of record the write stays within that bound past the embedding, as it did on `falloff` (*fixed-effects*). With both tolerances, one run of eight on each has sites over, all at operands at slice 1, so both pass (*write-floor*°).

write-floor. An absolute tolerance of about a degree, from the spread of the bound between runs, takes out every site over the write bound at a symbol or separator; what stays is at operands at slice 1, on one run of the recipe of record and every run of `pool 1–3, soft τ`.

The bound itself is wider at slice 3 than on `falloff`, though the difference is only suggestive (wider by 18.8° on the seed mean at the example answers, with an interval of 0.2° to 37.5°). It is set by a tail of about one state of another op in a hundred, and many of those sit on near misses for `difference`, example answers one grid step from an answer `difference` could give. The model seems to take a near miss as weak evidence for the op, so some of that turn is removal rather than a side effect.

On the recipe of record the evidence for the op peaks at slice 3 on every seed. *Block1-edit*° gives an account of where the op is held: block 1 writes the evidence of each example onto e_1, a later block (usually block 3) reads it at the query `=`, and an edit removes the op whenever it sits between the two. That is why editing block 1 alone removes most of the op under both recipes, though nothing pulls block 1 under the older step recipe.

block1-edit. Block 1 writes the evidence of each example onto e_1, one later block reads it from the query `=`, and an edit removes the op when it sits between the two.

Inside each block the MLP reads what the attention just wrote, before the next edit acts. *Mid-block-edit*° edited those mid-block states. On their own they removed most of the op (0.73 to 0.91 on seven runs of eight), since an edit there also clears what the earlier blocks wrote. But they spilled past the criterion on six runs of eight, and added to the edit between blocks they gained no removal and spilled on four. The attention of block 1 puts e_1 on every op at the mid-block state, and nothing in training keeps it clear there, because the anchor and anti terms act only between blocks. With those terms at the mid-block states too, the anti term especially, the mid-block sites might become selective and close some of the distance left. We keep the edit between blocks for the recipe of record, and leave that as future work.

mid-block-edit. Editing the states between the attention and the MLP of each block as well gains no removal and spills on four runs of eight; the edit between blocks stays.

(e) Scarce, noisy labels are enough

About one `difference` context in fifty is labelled, which is the shape of label we expect for a concept in natural language. The labels are already noisy in one sense: a label marks the op that generated the context, even when its examples favor another op. With the op word in the grammar (the older recipe), moving a fifth of the labels onto contexts of other ops still placed the op. Under replacement op noise, about a quarter of `difference` contexts, and so of the labelled ones, have a posterior below one half on `difference` (from 3,000 contexts drawn as the corpus draws them). So many labels already mark an op the model can't work out from the context. That may be enough noise to make the claim for the post, with wrong labels on the inferred op left as future work.

Before the blog post

The reference model (*sans-diff*°) and the full scoring suite on the recipe of record, confusion matrices included (*fixed-effects*°), are done, and their figures are meant for the post. One piece is left, and its plan is written: the post-hoc comparison, agreed 2026-10-30 after three paper rounds. What the post should say, and in what order, is collected in the post outline.

***sans-diff*.** The control trained with no loss on `difference` contexts lands 0.58 of the way to the target null, against 0.50 for the edited model, and misses in the same kind of way.

***fixed-effects*.** The full scoring suite on `τ_ℓ fixed` …

Directions found after training

> Fit a direction for `difference` on the states of the controls, which were trained without the anchor, and edit along it through the same scorer. Three fits: the difference in means, a ridge probe, and LEACE (a linear eraser). No training is needed. Ex-2.2.1 did the same for *red*, where these directions cost the lines with no red about as much as they removed from red. To keep the comparison generous, each fit gets every label as well as the scarce ones, so a gap in favor of the anchor can't come from starving the other methods. In the backlog.

Future work

These are leads, roughly in the order we would take them up.

- The anchor and anti terms at the mid-block states as well, so that an edit there might be selective and remove more of the op, perhaps with a second axis for the op at the mid-block states beside e_1 for its evidence between blocks (backlog).
- A two-block model, where the reader of the op can only be block 2 (backlog).
- A map of what e_1 holds along the context (backlog).
- Whether the soft start of τ_p is still needed now that τ_ℓ is sharp.
- Reference models without each op in turn, with that op anchored in place of `difference` (backlog).
- Gradient routing, a training-time method that masks gradients so that labelled examples update only parameters set aside for the concept (backlog).
- Wrong labels on the inferred op (backlog).
- Whether the rest of the representation (the color geometry and the states of the other ops) keeps its shape under the anchor.
- A sweep of the recipe weights and schedules together (backlog), and a run at 600 epochs.
- An interactive view of α°, sliced by op, role, slice, seed and epoch (backlog).

Alignment. How far a state points along e_1…

Out of scope

The verification measurements (D2.3), though verification lines are allowed in the corpus. Topic markers, and anchoring at fine-tune time, both open questions in the pivot. Several ops on separate axes.

References

ex-2.2.14

Anchoring the op word: the anchor landed on the word itself, which led to the pivot. Scarce and partly wrong labels placed it as well as full ones.

ex-2.2.22

Pooled over contexts, the edited model answers about as the target null would, though context by context it gets less than half of the way there.

embedding-lean

The early spill came from lightness loaded onto e_1 in the color embedding table, a stand-in for `difference` before the model has read the context.

edit-effects

On `falloff`, the write stays within the bound, the drop on each context follows how much it needed `difference`, and the full edit lands about half of the way to the target null.

block1-edit

Block 1 writes the evidence of each example onto e_1, one later block reads it from the query `=`, and an edit removes the op when it sits between the two.

mid-block-edit

Editing the states between the attention and the MLP of each block as well gains no removal and spills on four runs of eight; the edit between blocks stays.

reference-baseline

The median over the other coordinate axes of the run replaces the paired control as the baseline, and the numbers barely move.

rotated-basis

With the basis of the optimizer rotated, the anchor holds and removes as before; only the paired-control baseline suffers.

recipe-compare

The pool over slices 1 to 3 removes about as much as the step recipes, and its spill enters with the edit at block 1.

soft pool scout

The final slice temperature decides the early spill of the pool; at τ_ℓ 0.1 no run spills, and the edit at block 1 costs about what it does under the step.

late spill scout

The late spill follows the anti anneal, and sharpening τ_ℓ late lowers it.

fixed confirmation

`τ_ℓ fixed` at fresh seeds keeps every anchor, removes about as much as the annealed version, and has no late spill; adopted as the recipe of record.

fixed-effects

The full scoring suite on `τ_ℓ fixed`: removal by dose, grading, landing and the write bound come out about as on `falloff`, apart from a bound at slice 3 that is suggestively wider.

sans-diff

The control trained with no loss on `difference` contexts lands 0.58 of the way to the target null, against 0.50 for the edited model, and misses in the same kind of way.

write-floor

An absolute tolerance of about a degree, from the spread of the bound between runs, takes out every site over the write bound at a symbol or separator; what stays is at operands at slice 1, on one run of the recipe of record and every run of `pool 1–3, soft τ`.

▼ **About this page**

Updated 2026-10-30 for the absolute tolerance on the write bound° (*write-floor*°). Updated 2026-10-29 for the reference model (*sans-diff*°), the full scoring suite (*fixed-effects*°), the account of where the op is held (*block1-edit*°, *mid-block-edit*°), and the comparison of conditions as two samples. Rewritten 2026-10-28 for the recipe of record `τ_ℓ fixed`, and simplified: the provenance of each setting moved to the reports, the claims say where they stand, and the open work is split into what comes before the blog post and what is future work. The longer version, with its history of the recipe, is in git (as of `a299178`), and the plan through *ex-2.2.22*° is archived.

Write bound. How muc e_1 the blocks after an edit put back on the states of the other ops, against what the clean model has there.

***write-floor*.** An absolut tolerance of about a de gree, from the spread c the bound between run takes out every site ov the write bound at a symbol or separator; what stays is at oper- ands at slice 1, on one run of the recipe of record and every run o `pool 1–3, soft τ`.

***sans-diff*.** The control trained with no loss on `difference` contexts lands 0.58 of the way t the target null, against 0.50 for the edited model, and misses in th same kind of way.

***fixed-effects*.** The full scoring suite on `τ_ℓ fixed` …

***block1-edit*.** Block 1 writes the evidence of each example onto e_1, one later block reads it from the query `=`, and an edit removes the op when it sits between th two.

***mid-block-edit*.** Editing the states between the attention and the MLP each block as well gain no removal and spills o four runs of eight; the edit between blocks stays.

***ex-2.2.22*.** Pooled over contexts, the edited model answers about a the target null would, though context by con- text it gets less than ha of the way there.