Why editing block 1 alone removes the op

> Nothing pulls block 1 under the step recipe, yet editing it alone removes most of the op. The reason is that block 1 writes the evidence for the op onto e_1, and one later block reads it off the example answers. An edit removes the op when it sits between that writer and that reader, and block 1 is upstream of every reader.

Since the late-pull experiment°, the recipes have kept the pull off the early slices, so that stand-ins for the op stay out of the embedding table. Yet on every seed, editing block 1 alone removed most of the op (slice-terms° E1, recipe-compare° E3). And editing blocks 2 to 4, where the pull acts, fell short on the same seed in every arm. Neither result had an account.

late-pull experiment. With the pull on blocks 2 to 4, the color embedding table leans a quarter as far onto e_1 and block 1 comes to hold part of the concept, though nothing pulls it.

slice-terms. Per-slice anti weights recover removal with spill about within the criterion; editing block 1 alone removes most of the op on every seed.

recipe-compare. `step, soft τ` stays close to `falloff`; under every recipe the edit at blocks 2 to 4 removes the op and spills nothing, and what spill there is enters with the edit at block 1.

This report trains nothing. It takes the stored checkpoints of the step recipe `falloff` at seeds 704 to 711 and of the recipe of record `τ_ℓ fixed` at 712 to 719, with two controls, and follows the alignment° with e_1 through the model one sub-step at a time, clean and under edits at chosen slices and positions. It also looks at where attention goes, and at what color information the states hold.

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

Observations

- Where the evidence is written (E1): block 1 is the first place the evidence for the op can be computed, and both recipes write about half of it there, whether or not block 1 is pulled. At block 1 each example answer gathers the colors of its own operands.
- What the later blocks add (E2): the later blocks mostly carry the alignment° forward. Under the edit at block 1 they put back only a little of it, even though the operand colors are still there to compute it from.

Alignment. How far a state points along e₁…

- The reader, and the edits at one slice (E3): on each run one block reads the example answers from the query `=`, usually block 3. An edit at one slice removes the op when it sits between block 1 and that reader, which accounts for the seeds where the edit at blocks 2 to 4 fell short.

Scope and method

This is an exploratory re-analysis, with no preregistration and no gate. It covers 16 anchored runs, all trained for 400 epochs on the in-context grammar with `difference` anchored on e_1:

- `falloff` (seeds 704 to 711, from split-anti-seeds°): the pull steps on at slice 2 and acts on slices 2 to 4. Nothing pulls block 1.
- `τ_ℓ fixed` (seeds 712 to 719, from fixed-confirm°): the pull is pooled over slices 1 to 3, so block 1 can take part of it.

split-anti-seeds. At fresh seeds, the `falloff` anti weight keeps the anchor and keeps spill within the criterion; it became the step recipe of record.

fixed-confirm. The pool over slices 1 to 3 with τ_ℓ fixed keeps the anchor on all eight runs at seeds 712 to 719 and was adopted as the recipe of record.

The two controls of ex-2.2.23 at seeds 704 and 710 were trained with no pull and no anti term.

Each block adds to the residual stream twice, first with its attention and then with its MLP. So the figures follow the stream at nine points: the embedding, then after the attention and after the MLP of each of the four blocks. A slice is the stream after a whole block, so slice 1 is the point after the MLP of block 1.

Every pass uses the 14,000 held-out contexts of ex-2.2.21, 2,000 per op. The edit is the full e_1 edit, applied after the named blocks, at every position unless stated. Removal° here is not net of a baseline, unlike the published scores, because the median baseline of the D2.2 design exists only for the edit at every slice. The controls show what a baseline would subtract: the same edits remove at most 0.04 of `difference` on them where the edit is at one slice, and up to 0.09 where it is at several.

Removal. The drop in the score on `difference` under an edit, as a share of the way from the clean score to the target null, with no baseline subtracted.…

What block 1 computes is inferred from where the alignment° appears, from attention, and from color probes°. We did not fit a probe for the op itself, which would show the per-example judgement as a feature apart from e_1 (the backlog item on what e_1 holds along the context asks for one).

Alignment. How far a state points along e_1…

Color probe. A ridge regression from the 64-dimensional state to the RGB of a color, scored by held-out R^2 over five folds of contexts (the positions of one context stay in one fold), on contexts of every op. 1 means the color can be recovered fully from the state, and 0 means no better than its mean.

The measurements

α

Alignment with e_1: the e_1 component of the unit-norm state. Here, the mean over the three example answers of `difference` contexts unless stated.

Removal

The drop in the score on `difference` under an edit, as a share of the way from the clean score to the target null, with no baseline subtracted. Higher means more of the op removed.

Reader block

The block whose attention from the query `=` to the example answers is largest, averaged over heads and over `difference` contexts.

Color probe

A ridge regression from the 64-dimensional state to the RGB of a color, scored by held-out R^2 over five folds of contexts (the positions of one context stay in one fold), on contexts of every op. 1 means the color can be recovered fully from the state, and 0 means no better than its mean.

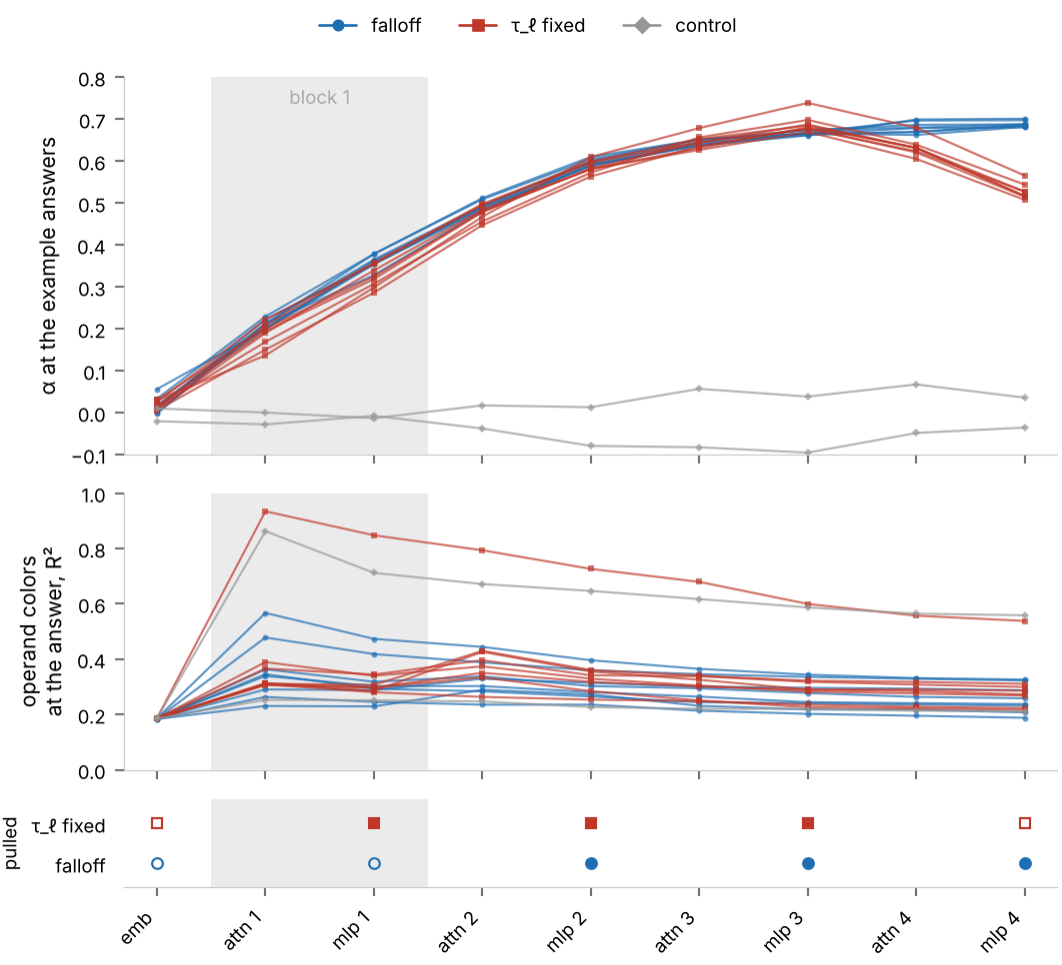
Where the evidence is written (E1)

Where along the stream does the alignment° at the example answers appear? The evidence for the op is a judgement about one example: does this answer fit `difference` on these two operands? The embedding of an answer is only its token, so the judgement needs at least one attention step, from the answer to its operands, and something to compute with what it gathers. Block 1 is the first place that can happen. The main measure is α° at the example answers, with the color probe° beside it.

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

α. Alignment with e_1...

Color probe. A ridge regression from the 64-dimensional state to the RGB of a color, scored by held-out R^2 over five folds of contexts (the positions of one context stay in one fold), on contexts of every op. 1 means the color can be recovered fully from the state, and 0 means no better than its mean.



***Where the alignment and the operand colors arrive at the example answers**, by sub-step: the embedding, then the stream after the attention and after the MLP of each block. One line per run, in the ink and marker of its recipe; block 1 is shaded. **Top:** α at the example answers of `difference` contexts. **Middle:** how well a linear probe recovers the colors of the two operands of an example from the state at its answer (held-out R^2, every op). **Bottom:** the slices each recipe pulls (filled).*

Both recipes write about half of the final alignment in block 1: a little over half of that in its attention, and the rest in its MLP. After block 1, α at the example answers is 0.29 to 0.38 across the 16 runs, 0.5 to 0.7 of its value after block 4. The embedding has none of it, and the controls stay near zero throughout.

The profile through block 1 is about the same under `falloff`, which does not pull block 1, as under `τ_ℓ fixed`, which does. So the pull at block 1 itself seems to make little difference to what block 1 writes. The pull at slices 2 to 4 reaches block 1 anyway: the stream carries what block 1 writes to every later slice, so the gradient of the pull there flows back along the residual path to the parameters of block 1. The cheapest way to have e_1 aligned at three later slices is to write it once, at the earliest block that can, and let the stream carry it.

The attention fits this account. Block 1 is the first block where an example answer can look at its operands, and on every run it does: it puts 0.14 to 0.55 of its attention on the two operands of its own example. Blocks 2 and 3 look at them too, on some runs as much, but they come after the alignment has appeared.

recipe	block 1	block 2	block 3	block 4
`falloff`	0.14 to 0.43	0.08 to 0.25	0.05 to 0.32	0.00 to 0.10
`τ_ℓ fixed`	0.14 to 0.55	0.10 to 0.38	0.00 to 0.13	0.01 to 0.10
`control`	0.22 to 0.57	0.16 to 0.16	0.08 to 0.12	0.01 to 0.01

***Attention from each example answer to the operands of its own example**, by block: the share of its attention that lands on those two positions, mean over heads and over `difference` contexts, range over the runs of each recipe (2 controls).*

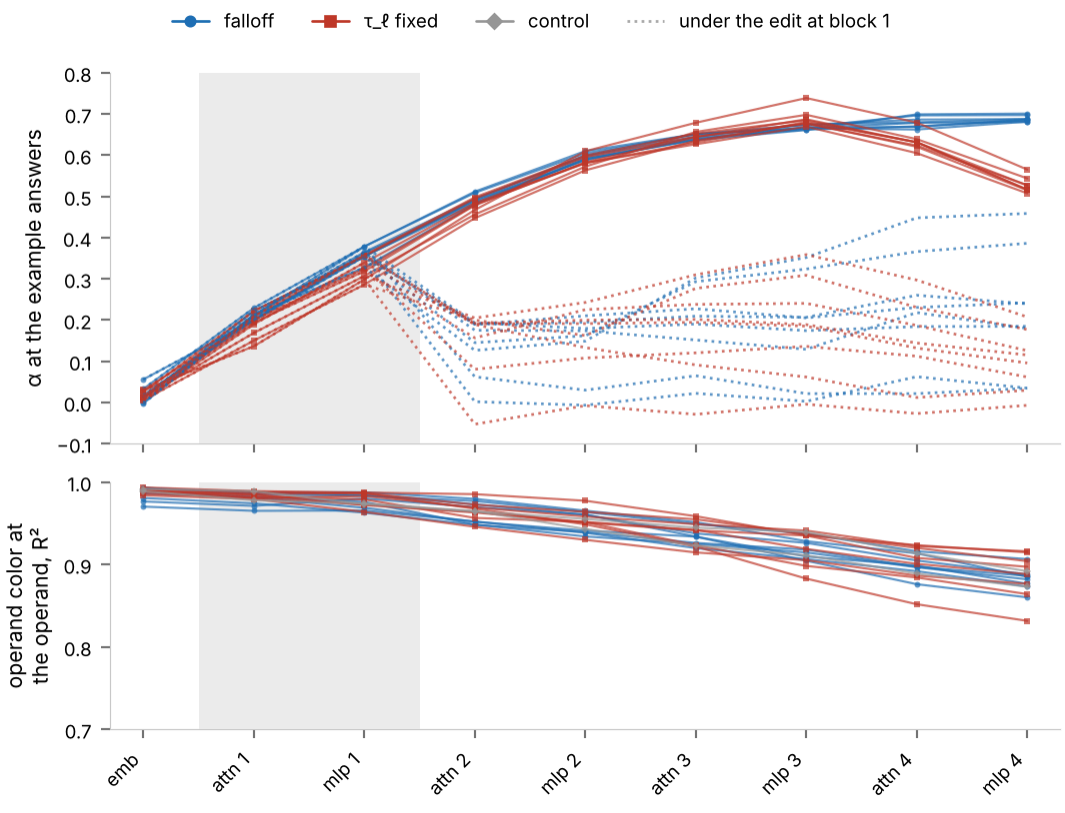
The color probe at the answer shows what that attention brings. The operand colors are partly recoverable from the answer even at the embedding, since an answer says something about its operands under most ops. At the attention of block 1 that rises on every run, the controls included, then eases down through the later blocks. How far it rises varies a great deal from run to run, and it does not separate the anchored runs from the controls. So gathering the operands at the answer is something the model does for the task anyway, and the pull puts the evidence for the op, judged from them, onto e_1.

What the later blocks add (E2)

α. Alignment with e_1...

Blocks 2 and 3 raise α° at the example answers from about a third to its final value on the clean pass. Do they compute the evidence again, or add to what arrives? If they computed it again, the alignment° would come back after an edit at block 1 takes it out. The main measure is α at the example answers under the edit at block 1.

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



What the later blocks put back after the edit at block 1. Top: *α at the example answers of* `difference` *contexts, clean (solid) and under the edit at block 1 (dotted), one pair of lines per anchored run. The edit takes effect after the MLP of block 1, so the dotted line leaves the solid one at the attention of block 2.* ***Bottom:*** *how well a linear probe recovers the color of an operand from the state at its own position (held-out R^2, every op, clean), for every run including the controls.*

Most of it does not come back. Under the edit at block 1, α at the example answers after block 4 is −0.01 to 0.46, against 0.51 to 0.70 on the clean pass. So what blocks 2 and 3 add onto e_1 is mostly downstream of what arrives: take away the e_1 component at block 1, and their writes onto e_1 largely go with it. The part that does come back varies by run, and E3 shows it is what sets how far short of full removal° the edit at block 1 falls.

Removal. The drop in the score on `difference` under an edit, as a share of the way from the clean score to the target null, with no baseline subtracted....

Could the later blocks compute the evidence again, if they had what it takes? The operand colors are still there at the operand positions: the color probe° there falls only a little through the four blocks (lower panel), and the controls look the same. And blocks 2 and 3 do attend from the example answers to their operands on many runs (table in E1). Under the edit at block 1 that attention stays about the same, or rises a little.

Color probe. A ridge regression from the 64-dimensional state to the RGB of a color, scored by held-out R^2 over five folds of contexts (the positions of one context stay in one fold), on contexts of every op. 1 means the color can be recovered fully from the state, and 0 means no better than its mean.

So the later blocks have both the colors and the attention, and still put back only part of the alignment. What they write onto e_1 seems to depend on the e_1 component that arrives, which fits blocks that add to the evidence block 1 wrote more than blocks that compute it afresh. A probe for the op itself, apart from e_1, would show whether the later blocks keep a copy of the judgement off e_1; we did not fit one.

The reader, and the edits at one slice (E3)

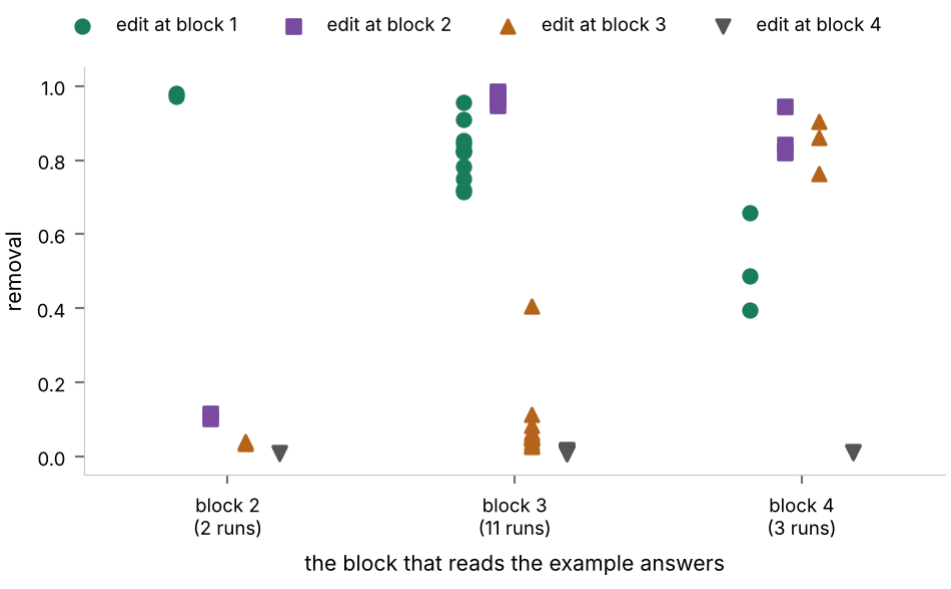
If block 1 writes the evidence, where is it used? The query `=` predicts the answer, so the evidence at the example answers has to reach it through attention. And why did the edit at blocks 2 to 4 fall short on some seeds and not others? The main measure is removal° under the edit at one slice, set against the block that reads the example answers.

Removal. The drop in the score on `difference` under an edit, as a share of the way from the clean score to the target null, with no baseline subtracted....

On every run, the attention from the query `=` to the example answers is concentrated in one block: block 3 on 11 of the 16 runs, block 4 on 3 (706, 708, 716), and block 2 on 2 (710 and 712). That reader takes in the stream from the slice before it. α° at the query `=` itself stays low at every slice (at most 0.1), so what the reader carries over lands off e_1, where no later edit reaches it.

α. Alignment with e_1...

Where the reader forms may go with how early the example answers hold their operands. The two runs whose reader is block 2 are the runs where the attention of block 1 gathers the operand colors at the answer most, each the highest of its recipe (the top lines in the middle panel of the E1 figure), and the control at seed 710 gathers them about as much. So the seed seems to set how strongly block 1 gathers, and perhaps with it how early the reading can form, though two runs are too few to say more.



Removal under the edit at one slice, by the block that reads the example answers. *One mark per anchored run and edit, its shape naming the slice edited, nudged sideways so the edits sit apart. The reader block is where the attention from the query `=` to the example answers is largest.*

This one fact accounts for the pattern of the edits at one slice:

- The edit at block 4 removes nothing on any run, because every reader has already taken what it needs.
- The edit at block 3 removes nearly everything only where the reader is block 4.
- The edit at block 2 removes nearly everything where the reader is block 3 or 4, and little where it is block 2, which takes in slice 1.
- The edit at block 1 removes the op on every run, since every reader comes after it.

The same holds for the edit at blocks 2 to 4: it removes nearly everything unless the reader is block 2. Across the 16 runs, removal under that edit and the share of the reading done at block 2 have a correlation of −1.00. Seed 710 fell short in every earlier arm (split-anti-seeds°, recipe-compare°, and the soft-pool arms), and it is one of the two seeds whose reader formed at block 2.

split-anti-seeds. At fresh seeds, the `falloff` anti weight keeps the anchor and keeps spill within the criterion; it became the step recipe of record.

recipe-compare. `step`, `soft τ` stays close to `falloff`; under every recipe the edit at blocks 2 to 4 removes the op and spills nothing, and what spill there is enters with the edit at block 1.

run	reader block	α after block 1	α after block 4	edit at block 1 ↑	edit at block 2	edit at block 3	edit at blocks 2 to 4	edit at every slice	α arriving at the reader, under the block-1 edit
`falloff` 704	3	0.37	0.69	0.71	0.97	0.05	0.97	0.97	0.19
`falloff` 705	3	0.36	0.70	0.84	0.98	0.06	0.98	0.99	0.17
`falloff` 706	4	0.33	0.70	0.39	0.84	0.90	0.97	0.92	0.35
`falloff` 707	3	0.36	0.69	0.78	0.98	0.09	0.98	0.99	0.21
`falloff` 708	4	0.35	0.69	0.49	0.82	0.76	0.91	0.87	0.32
`falloff` 709	3	0.38	0.69	0.82	0.99	0.05	0.99	0.99	0.18
`falloff` 710	2	0.35	0.68	0.97	0.12	0.04	0.13	0.97	0.00
`falloff` 711	3	0.38	0.68	0.96	0.97	0.11	0.97	0.97	0.03
`τ_ℓ fixed` 712	2	0.30	0.57	0.98	0.10	0.04	0.12	0.98	0.00
`τ_ℓ fixed` 713	3	0.36	0.53	0.75	0.98	0.05	0.98	0.98	0.20
`τ_ℓ fixed` 714	3	0.36	0.53	0.83	0.98	0.06	0.98	0.98	0.19
`τ_ℓ fixed` 715	3	0.32	0.52	0.72	0.95	0.04	0.95	0.92	0.23
`τ_ℓ fixed` 716	4	0.31	0.54	0.66	0.95	0.86	0.95	0.96	0.36
`τ_ℓ fixed` 717	3	0.34	0.51	0.91	0.98	0.03	0.97	0.98	0.13
`τ_ℓ fixed` 718	3	0.29	0.52	0.85	0.96	0.41	0.97	0.98	0.16
`τ_ℓ fixed` 719	3	0.33	0.52	0.82	0.97	0.03	0.97	0.93	0.11

Each anchored run. *The reader block, α at the example answers of `difference` contexts after block 1 and after block 4 (clean), removal under the edit at each set of slices, and α at the example answers as it arrives at the reader block under the edit at block 1 (none for a reader at block 2, which takes in the edited slice). Removal is not net of a baseline (see the scope and method).*

How much the edit at block 1 removes depends on how much of the alignment° the later blocks put back before the reader takes it in (the last column). On the runs whose reader is block 2, the reader takes in the edited slice itself, nothing has come back, and removal is nearly full. On the runs whose reader is block 4, about half of the clean alignment at slice 3 is back by then, and removal is lowest.

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

It is the e_1 component at the example answers that holds the op. The edit at block 1 restricted to the example answers removes as much as the edit at block 1 at every position, within 0.01 on every run, while the same edit at every other position removes nothing (−0.02 to 0.00). And under the edit the reader still looks at the example answers: its attention there stays about the same on most runs, and halves on one (716). What it takes in has lost the evidence.

Discussion

The residual stream makes "where the pull acts" and "where the concept is written" separate questions. The pull names slices, and training answers with the earliest block that can compute the concept, because one write there serves every pulled slice after it. On this task that is block 1, since the per-example judgement needs one attention step (the answer gathers its operands) and one MLP. The lean of embedding-lean° looks like the same effect one step earlier: where the pull reached the embedding, the model answered with the nearest thing the embedding could offer. So keeping the pull off block 1 did not keep the concept off block 1. It kept the stand-ins off it, which was its purpose.

embedding-lean. Where the pull reaches the embedding, the color embeddings lean onto e_1 by lightness, a stand-in for the op that the edit then takes from every op.

The write bound of edit-effects° already treats the edit at each slice as acting on what arrives there. This account adds that most of what arrives at blocks 2 to 4 on e_1 was put there by block 1. So the edit at every slice does most of its work at slice 1, and the later slices find little left to turn.

edit-effects. On `falloff`, the blocks after the edit put back no more e_1 on the states of the other ops than the clean model has there.

It also says why the edit at every slice is the robust choice, as the D2.2 design argues. An edit at fewer slices has to land between the writer and the reader, and the reader forms at a different block from seed to seed. The edit at every slice covers every pairing. A larger model has readers at many depths that we don't choose, so this is the property that would carry over.

The task needs three steps in sequence: each example answer judges itself against its operands, the query `=` reads those judgements, and the query applies the op it infers. That fits in two blocks. If which block becomes the reader is set by when the reading forms during training, a two-block model would leave it only block 2, and the edit at slice 1 would be the whole edit. That would make a cleaner picture of the mechanism, though without the depth that makes the reader vary.

References

late-pull experiment

With the pull on blocks 2 to 4, the color embedding table leans a quarter as far onto e_1 and block 1 comes to hold part of the concept, though nothing pulls it.

slice-terms

Per-slice anti weights recover removal with spill about within the criterion; editing block 1 alone removes most of the op on every seed.

recipe-compare

`step, soft τ` stays close to `falloff`; under every recipe the edit at blocks 2 to 4 removes the op and spills nothing, and what spill there is enters with the edit at block 1.

split-anti-seeds

At fresh seeds, the `falloff` anti weight keeps the anchor and keeps spill within the criterion; it became the step recipe of record.

fixed-confirm

The pool over slices 1 to 3 with τ_ℓ fixed keeps the anchor on all eight runs at seeds 712 to 719 and was adopted as the recipe of record.

embedding-lean

Where the pull reaches the embedding, the color embeddings lean onto e_1 by lightness, a stand-in for the op that the edit then takes from every op.

edit-effects

On `falloff`, the blocks after the edit put back no more e_1 on the states of the other ops than the clean model has there.