Does the edit miss the mid-block state?

> Inside each block, the attention writes to the stream and the MLP of the same block reads it before the edit acts again. Editing that mid-block state as well gains nothing on the recipe of record: the between-block edit already takes out nearly all of the op, and the extra sites cost selectivity, because the MLPs seem to use e_1 for the other ops too. The anchor and anti terms act only between blocks, so nothing in training keeps e_1 clear at the mid-block states.

The edit of record acts between blocks, at the same slices the anchor pulls. But each block adds to the residual stream twice, first with its attention and then with its MLP, and nothing touches the state in between. So a block could, in principle, write the op onto e_1 with its attention and use it in its own MLP before any edit sees it. The *block1-edit*° account says the op travels between positions, from the example answers to the query `=`, and every such move goes through attention, which takes in an edited slice. That route is covered. The route this report asks about stays within one position and one block.

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

This report trains nothing. It takes the stored checkpoints of the recipe of record, `τ_ℓ fixed` at seeds 712 to 719, and scores the full e_1 edit at three sets of sites: between blocks (the edit of record), at the mid-block states alone, and at both.

Observations

- What the attention writes after the edit (E1): under the between-block edit, the attention of block 2 puts some alignment° back at the example answers, and the later blocks almost none. The query `=` stays near zero throughout.
- Removal and spill with the mid-block sites (E2)°: adding the mid-block sites removes a little less of the op, and half the runs then spill° past the criterion. The mid-block sites alone remove less, and spill more.

Alignment. How far a state points along e_1…

Removal. The drop in the score on `difference` under the edit, less the median drop under the same edit along each of the other 63 axes of the run, as a share of the way from the clean score to the target null….

Spill. The largest drop on another op under the edit, net of the same median baseline….

Scope and method

This is an exploratory re-analysis of eight runs, with no preregistration and no gate. The runs are the eight of *fixed-confirm*°, trained for 400 epochs on the in-context grammar with `difference` anchored on e_1 and the pull pooled over slices 1 to 3. No controls were scored, since every score here is net of the median over the other axes of the same run.

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 mid-block state of a block is the stream after its attention step and before its MLP step. The edit is the full e_1 projection at every position, followed by the re-projection onto the sphere, as in the edit of record. "Mid-block" edits the mid-block states of all four blocks, and "both" adds them to the between-block edit at all five slices.

Every pass uses the 14,000 held-out contexts of ex-2.2.21, 2,000 per op. The between-block scores reproduce those of *fixed-confirm*. We scored the mid-block sites of all four blocks together, so this report does not say which block the spill° comes from.

Spill. The largest drop on another op under the edit, net of the same median baseline....

The measurements

α

Alignment with e_1: the e_1 component of the unit-norm state, averaged over the three example answers (or taken at the query `=`) of `difference` contexts unless stated.

Removal

The drop in the score on `difference` under the edit, less the median drop under the same edit along each of the other 63 axes of the run, as a share of the way from the clean score to the target null. Higher means more of the op removed.

Spill

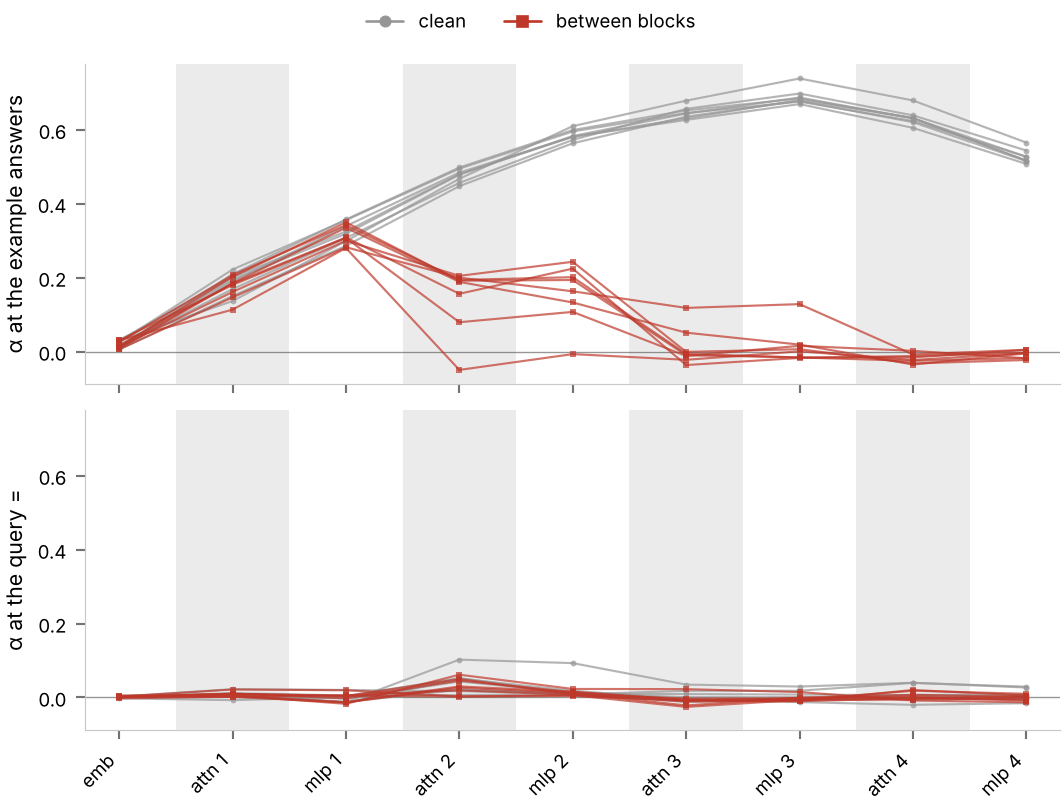
The largest drop on another op under the edit, net of the same median baseline. The selectivity criterion of the D2.2 design is 0.02.

What the attention writes after the edit (E1)

If a block wrote the op onto e_1 with its attention after the edit had cleared it, the mid-block state would show alignment° that the next slice edit takes away only after the MLP has used it. So the main measure here is α° at the mid-block states under the between-block edit.

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

α. Alignment with e_1...



***Alignment with e_1 by sub-step**, clean (grey circles) and under the full edit between blocks (red squares): the embedding, then the stream after the attention (shaded) and after the MLP of each block. One line per run, on `difference` contexts. Under the edit, every point after an MLP is the state before the edit acts, so it shows what each block put back. **Top:** at the example answers. **Bottom:** at the query `=`.*

On the clean pass, the alignment at the example answers builds up through blocks 1 to 3, as in the *block1-edit*° report. Under the between-block edit, the attention of block 2 puts some of it back at the example answers (0.15 (−0.05 to 0.20) across runs), the MLP of block 2 keeps it about level, and the edit at slice 2 then clears it. The attention of blocks 3 and 4 puts back almost nothing on most runs. So there is alignment for an MLP to read inside block 2, about a third of the clean value at that point.

***block1-edit*.** Block 1 writes the evidence for the op onto e_1 at the example answers, one later block reads it from the query `=` through attention, and an edit removes the op when it sits between the two.

At the query `=`, where the answer is predicted, α stays near zero on every sub-step, clean or edited. The *block1-edit* report found the same between blocks, and it holds inside the blocks too.

Before any edit, the attention of block 1 writes nearly as much alignment at the example answers of the other ops (0.15 (0.12 to 0.18)) as at those of `difference` (0.18 (0.14 to 0.22)). Only the MLP of block 1 sets `difference` apart. So the mid-block state of block 1 has e_1 on every op, and most of what it holds there is something other than the op.

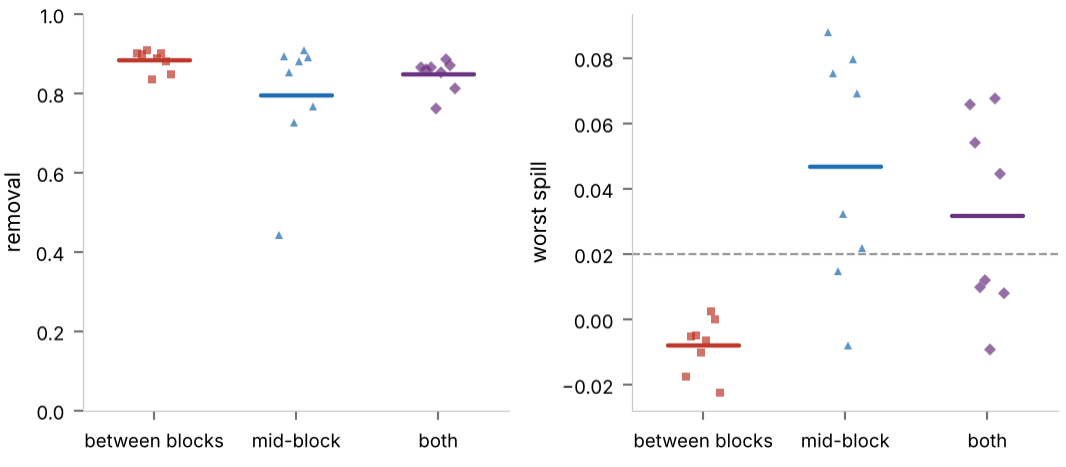
Removal and spill with the mid-block sites (E2)

Does editing the mid-block states as well remove more of the op?

The main measure is removal°, with spill° beside it.

Removal. The drop in the score on `difference` under the edit, less the median drop under the same edit along each of the other 63 axes of the run, as a share of the way from the clean score to the target null....

Spill. The largest drop on another op under the edit, net of the same median baseline....



***Removal and spill under the full e_1 edit at three sets of sites**, one point per run (jittered), with the seed mean as a bar. Both are net of the median over the other 63 axes of the run. **Left:** removal, as a share of the way to the target null. **Right:** the largest spill on another op; the dashed line is the selectivity criterion.*

It does not. Adding the mid-block sites to the between-block edit lowers removal on every run, by a little (−0.07 to −0.01), and spill rises: 4 of the 8 runs go past the criterion, against 0 under the between-block edit. The mid-block sites alone remove less than the between-block edit and spill the most.

They still remove most of the op on their own: 0.73 to 0.91 on seven runs, and 0.44 on seed 712. That is because the edit takes out the whole e_1 component of the state, so a site in the middle of a block also clears everything the earlier blocks wrote, and only the MLP after it can add some back.

Most of the gap between removal and 1 is the baseline. With no baseline subtracted, the between-block edit takes out 0.92 to 0.98 of the way to the target null°, and the median edit along another axis takes out about a tenth. So there was little left for any extra site to find.

Target null. Where a removal should land...

The spill fits what E1 showed of block 1. The MLP of a block seems to read the e_1 component of its mid-block state on every op, so taking it away there changes what the MLP computes for the other ops as well. The slice after block 1 holds about as much e_1 on the other ops (0.15 (0.08 to 0.21)), and the between-block edit takes it away there with no spill, so what matters is which sublayer reads it next. With no baseline, the worst drop on another op is 0.020 (0.006 to 0.032) under the between-block edit and 0.073 (0.028 to 0.124) with the mid-block sites added.

seed	removal, between blocks ↑	removal, mid-block ↑	removal, both ↑	added by the mid-block sites	spill, between blocks ↓	spill, mid-block ↓	spill, both ↓
712	0.90	0.44	0.87	−0.04	−0.018	0.088	0.066
713	0.90	0.89	0.86	−0.04	−0.005	0.075	0.054
714	0.91	0.85	0.87	−0.04	−0.005	0.015	0.010
715	0.83	0.73	0.76	−0.07	−0.010	0.032	0.012
716	0.89	0.88	0.85	−0.03	−0.007	−0.008	−0.009
717	0.90	0.91	0.89	−0.02	0.003	0.080	0.068
718	0.88	0.89	0.87	−0.01	0.000	0.069	0.044
719	0.85	0.77	0.81	−0.04	−0.022	0.022	0.008

***Each run.** Removal and the largest spill on another op under the full e_1 edit at each set of sites, net of the median over the other 63 axes of the run, and the removal that adding the mid-block sites to the between-block edit gains.*

Discussion

The within-block route is there, in the sense that the attention of block 2 puts alignment° back where its MLP could read it. But removing it gains nothing, so the MLP does not seem to carry the op past the edit through it. The between-block edit already removes nearly all of the op on its own, which fits the *block1-edit*° account: the op reaches the query `=` through attention, from slices the edit has cleared.

Alignment. How far a state points along e_1…

***block1-edit*.** Block 1 writes the evidence for the op onto e_1 at the example answers, one later block reads it from the query `=` through attention, and an edit removes the op when it sits between the two.

The mid-block state also has e_1 on every op, before the MLP of block 1 makes it specific to `difference`, and the spill° suggests the MLPs use it. That makes it a poorer place to edit than the slice after the block, on models trained as these were. The anchor and anti terms see only the slices between blocks, so training never asks the mid-block states to keep e_1 for the op alone. With both terms at the mid-block states as well, the attention of block 1 might keep e_1 off the other ops, and the mid-block sites might then be as selective as the slices. That would not add removal° on this model, where the between-block edit already takes out nearly all of the op, but it would make any point along the stream a place where the edit works.

Spill. The largest drop on another op under the edit, net of the same median baseline….

Removal. The drop in the score on `difference` under the edit, less the median drop under the same edit along each of the other 63 axes of the run, as a share of the way from the clean score to the target null….

These results hold for this recipe and this small model. In a deeper model the attention and the MLP of one block might do more of the work between them; the scorer can now edit mid-block states, so the check is cheap to repeat there.

References

block1-edit

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

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.