An anchor weight that ramps with depth

> We tried replacing the step in the pull weight (nothing on the embedding or slice 1, full weight on slices 2 to 4) with a weight that rises in a straight line with depth. With the step weight kept at slice 4, the ramps lost the anchor on more seeds than the step did. Scaled up to the step total, the ramp from the embedding kept the anchor on every seed, but it spilled a little more, and the ramp from slice 1 still lost it.

The recipe of record (`falloff`) sets the anti weight $\lambda_{\bar{s}}$ by slice, falling in a straight line from the embedding to slice 4, so no slice boundary has to be chosen for it. The pull still has a boundary: it acts on slices 2 to 4 at the full anchor weight λ_a, and not at all on the embedding or slice 1.

The 4 new arms, of 8 runs each, try the mirror image of the falloff for the pull. They use the seeds of the split-anti-seeds experiment, so its `falloff` runs serve as the baseline with no new training.

Observations

- Removal and spill (E1)°: every ramp but one lost the anchor on two or three seeds, where `falloff` lost it on none. `ramp from emb, step total` kept it on every seed, with two runs over the spill criterion. The runs that kept the anchor removed about as much as `falloff`, and losses are fewer where the pull on slices 1 to 3 is higher.

Spill. The side effect of an edit on the other ops…

- When the ramped runs lose the anchor (E2): the example answers on a lost run never come onto e_1, from the first tens of epochs, while the pull is still warming up.

- Where the anchor sits by depth (E3): on the runs that kept it, the alignment° by slice is nearly the same in every arm, and the pull on slice 1 brings back neither the drift on the other ops nor the lean in the color table.

Alignment. How far a state points along e_1…

Scope

This is an exploratory study, with no preregistration and no gate. The new arms differ from `falloff` only in the pull weight by slice. The first two reach the weight of the step at slice 4, and differ in where they start from zero:

- `ramp from emb`: zero at the embedding, so slice 1 is pulled at a quarter of the slice-4 weight, slice 2 at a half and slice 3 at three quarters.
- `ramp from 1`: zero at slice 1, so the pull starts on slice 2 as the step does, at a third of the slice-4 weight, and slice 3 at two thirds.

Matching the step at slice 4 means each of these pulls less in total than the step: five sixths as much for `ramp from emb`, and two thirds for `ramp from 1`. When the first results showed both losing the anchor more often than `falloff`, we added two arms with the same shapes scaled to the step total (`ramp from emb, step total` and `ramp from 1, step total`), which pull slice 4 harder than the step does. These were not in the plan.

arm	term	emb	1	2	3	4
`falloff`	pull	0	0	1	1	1
`ramp from emb`	pull	0	1/4	1/2	3/4	1
`ramp from 1`	pull	0	0	1/3	2/3	1
`ramp from emb, step total`	pull	0	0.3	0.6	0.9	1.2
`ramp from 1, step total`	pull	0	0	1/2	1	3/2
every arm	anti hold	0.20	0.16	0.12	0.07	0.03

The term weights by slice. *The pull is relative to the anchor weight λ_a on each slice of the step, so* `falloff` *pulls slices 2 to 4 at 1. The anti hold is the weight the anti term anneals to, the same* `falloff` *line in every arm, rounded here to two places.*

The anti weight is the falloff in every arm, so the ramps from the embedding pull slice 1 against a hold of about 0.16.

The late-pull experiment moved the pull off the embedding and slice 1 to keep the lightness lean off the embedding table, and found that slice 1, unpulled, came to hold part of the concept. A pull there might bring some of the lean back; E3 looks for it. No pair of arms differs at slice 1 alone, so an effect there can only be judged together with slices 2 and 3.

Every run trains at model seeds 704 to 711 for 400 epochs, paired by seed with its `falloff` twin and with ex-2.2.23's control (same initialization, batches and label draws). With eight seeds per arm, one or two lost runs could come from the seeds alone, so we weigh the counts across arms together.

The edits on slice 1 alone and on slices 2 to 4 alone, which the plan gave a section to, are left out. On the runs that lost the anchor every edit removes almost nothing, except on seed 710 of `ramp from 1`, where the slice-1 edit removes about half, as the full edit nearly does. On the others they look like those of `falloff`, with the slices-2-to-4 edit again falling short on seed 710.

The measurements

The measurements and dose scoring are those of the split-anti-seeds experiment.

- Removal: the net drop in the score on `difference` under the edit, as a share of the way from the clean score to the target null°. The edit projects e_1 out of the state at every position and slice, at a dose of a quarter, a half, three quarters, or all of the e_1 component. Higher is better.

Target null. Where a removal should land…

- Spill°: the largest net drop on another op under the edit. A run is within the criterion when its spill is at most 0.02 at every dose.

Spill. The side effect of an edit on the other ops…

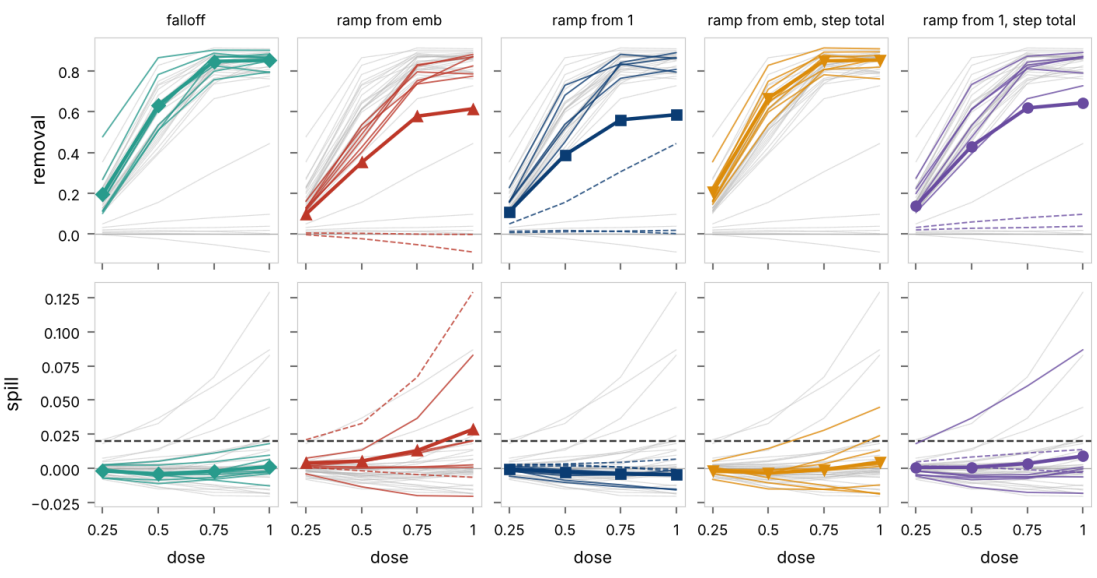
- Both are net of the control at the same seed.
- A run lost the anchor when the full edit removes less than half of the op, as in the split-anti-seeds experiment.
- Task change: the mean score over ops, less that of the control. Zero means no damage.
- The alignment° α with e_1 at the example answers, at every slice, and the lightness slope of the color embedding table (how much further along e_1 black sits than white).

Alignment. How far a state points along e_1…

Removal and spill (E1)

Does a pull that ramps with depth remove as much of the op as the step, and keep the edit as selective? The main measure is the removal at the full dose, with the spill° at any dose beside it.

Spill. The side effect of an edit on the other ops…



***Removal and spill by dose**, the edit at every position and slice, one column per arm. One hairline per run, dashed where the run lost the anchor; the seed mean is bold, and the other arms are ghosts behind. The dashed rule is the selectivity criterion.*

Most ramped runs look like `falloff`: removal climbs with the dose to between three quarters and nine tenths of the op, and the spill stays near zero. But three of the five arms have runs that remove little at any dose (the dashed lines), all but one of them almost nothing. Those runs lost the anchor.

arm	pull on 1 to 3	kept the anchor ↑	removal, kept ↑	within criterion ↑	spill, any dose ↓	task change ↑	worst op change ↑
`falloff`	2	8 of 8	0.85 (0.79 to 0.90)	8 of 8	0.003 (−0.004 to 0.018)	−0.007 (−0.032 to 0.005)	−0.066 (`hue-hsv`, 708)
`ramp from emb`	3/2	6 of 8	0.83 (0.77 to 0.88)	4 of 8	0.032 (−0.004 to 0.129)	−0.001 (−0.012 to 0.009)	−0.016 (`difference`, 710)
`ramp from 1`	1	5 of 8	0.84 (0.79 to 0.89)	8 of 8	0.001 (−0.006 to 0.007)	−0.007 (−0.041 to 0.008)	−0.079 (`hue-hsv`, 706)
`ramp from emb, step total`	1.8	8 of 8	0.85 (0.76 to 0.91)	6 of 8	0.009 (−0.008 to 0.045)	−0.004 (−0.028 to 0.014)	−0.059 (`hue-hsv`, 709)
`ramp from 1, step total`	3/2	6 of 8	0.84 (0.73 to 0.89)	7 of 8	0.011 (−0.005 to 0.087)	−0.017 (−0.084 to 0.007)	−0.230 (`value-hsv`,705)

***Removal, spill and task**, seed mean and range. The pull on slices 1 to 3 is the sum of the relative pull weights there (the step has 2). A run kept the anchor when the full edit removes at least 0.5 of the op, and removal is over those runs, at the full dose. "Spill, any dose" is the largest net spill over the four doses, and "within criterion" counts the runs where it is at most 0.02. The task change is the mean score over ops less that of the control, and the worst op change is the largest loss on any one op at any seed. Arrows mark which direction is better.*

`ramp from emb, step total` is the only ramp that kept the anchor on every seed, and its removal matches that of `falloff`. Two of its runs went over the spill criterion, one by a little and one by about twice the criterion, where `falloff` stayed within it on every seed.

The two ramps from slice 1 lost the anchor on 3 and 2 seeds, so scaling that shape up to the step total did not fix it. Scaling it up also pulls slice 4 half again as hard as the step, and one of its runs (seed 705) dropped by more than 0.2 on `value-hsv` against its control, and spilled over the criterion. That is one run, so we take it as a sign to watch rather than a cost of the shape.

Across the five arms, the count of lost runs follows the pull on slices 1 to 3 (the second column): none at 1.8 and 2, two at 1.5, three at 1. Eight seeds per arm is too few to call this a trend, and we picked that sum after seeing the counts.

The pull on slices 2 and 3 alone is the same in the two step-total arms, which lost none and two, so the ordering rests on the quarter-weight pull on slice 1. Still, it fits a simpler reading than the shape of the ramp: the pull on the slices before the last one is what keeps the anchor.

arm	seed	removal	spill, any dose	α, example answers	α, first operands
`ramp from emb`	706	−0.00	0.001	0.01	0.00
`ramp from emb`	709	−0.09	0.129	0.02	0.63
`ramp from 1`	706	0.02	0.003	0.05	−0.00
`ramp from 1`	708	0.00	0.002	0.01	0.00
`ramp from 1`	710	0.44	0.007	0.02	0.00
`ramp from 1, step total`	708	0.10	0.014	0.06	0.02
`ramp from 1, step total`	711	0.04	0.002	0.05	−0.01

***The runs that lost the anchor.** Removal at the full dose, the largest net spill at any dose, and α at slice 4 at the end of training on `difference` probe contexts, at the example answers and at the first operand of each equation. On the runs that kept the anchor, α at the example answers is 0.64 (0.63 to 0.66) at slice 4.*

One of these runs is borderline: seed 710 of `ramp from 1` removes a little under half the op, though its example answers are off e_1 at every slice, so it is a weak anchor rather than none. We count it as lost by the rule of the split-anti-seeds experiment.

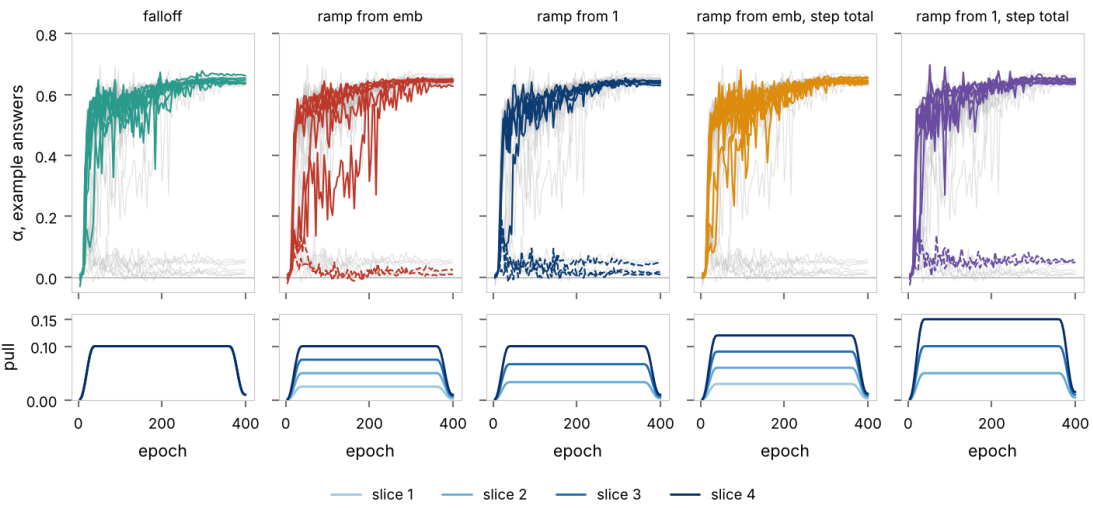
The lost runs are not the same seeds in every arm: seed 706 and seed 708 each lost the anchor in two arms, and the rest in one. On most of them the example answers stay off e_1, and the query answer aligns as usual, as it does on every run that kept the anchor. No other position takes its place: the separators and the newline stay off e_1 at every slice. This is the same kind of loss seen with the uniform holds of the split-anti-seeds experiment.

One run of `ramp from emb` (seed 709) settled on the first operands instead, and its query answer is off e_1 too. On that run the edit removes nothing from `difference`. It spills onto the other ops instead, since every op reads its operands.

When the ramped runs lose the anchor (E2)

When does a ramped run lose the anchor: early, as the pull first arrives, or later, as the anchor drifts off? The main measure is α° at the example answers, at slice 4, along training.

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



***Alignment at the example answers along training**, at slice 4, on the probe contexts of `difference`, one column per arm. One line per run, dashed where the run lost the anchor; the other arms are ghosts behind. The strips underneath are the pull weight on each slice, darker with depth.*

The lost runs part ways with the others in the first tens of epochs. On a kept run, α at the example answers climbs past 0.4 by about epoch 20 to 40, while the pull is still warming up. On a lost run it rises a little, falls back to near zero, and stays there at full pull weight. So the outcome is set while the pull arrives, and the long hold after it does not recover it.

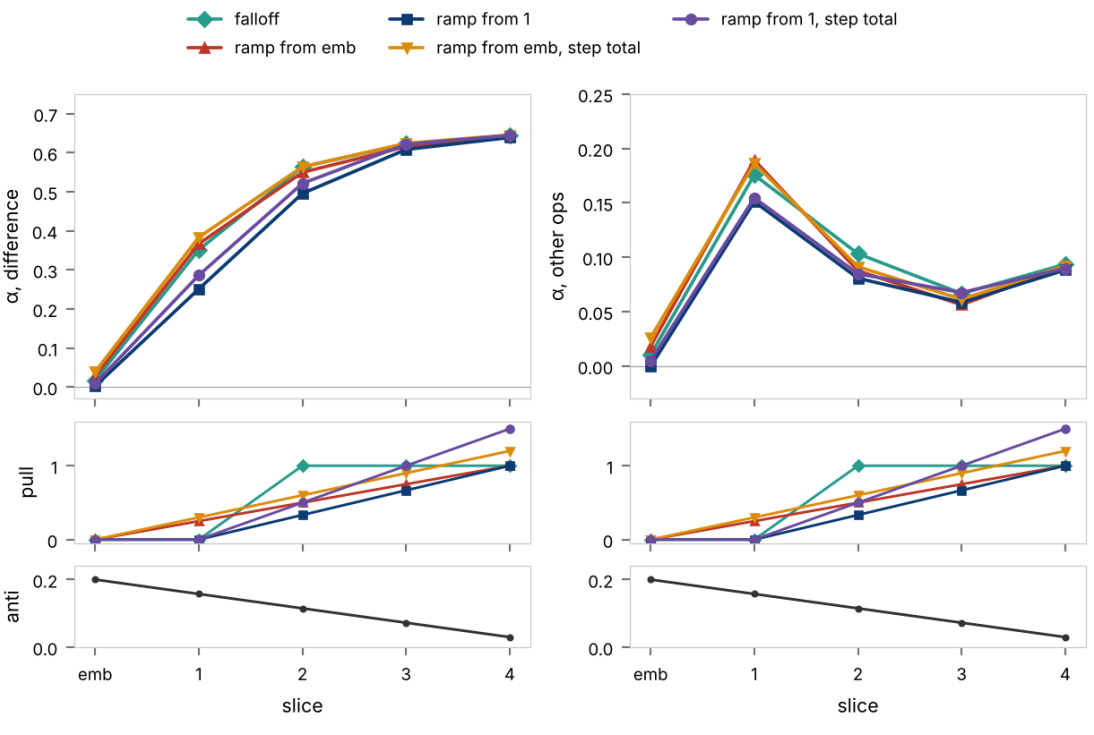
`ramp from emb` also has two kept runs that climb late: they first crossed 0.4 only at about epoch 60 to 80, and stayed below the others until about epoch 230. In `falloff` one run dips briefly, and none climbs late.

No other arm has a run like these, including the two ramps from slice 1, which pull slices 2 and 3 less. So slow climbs do not follow the pull sum the way the losses do, and two runs in one arm may be the seeds.

Where the anchor sits by depth (E3)

On the runs that kept the anchor, does the alignment° by slice follow the pull weight, and does a pull on slice 1 bring back the drift onto e_1 on the other ops, or the lean in the color table?

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



***Alignment at the example answers by slice**, at the end of training, on the probe contexts of `difference` (left) and of the other ops (right), over the runs that kept the anchor: the seed mean in the arm's marker; the two top panels have their own scales. Underneath are the pull weight at each slice, relative to the step, and the anti hold, which every arm shares.*

The alignment by slice is nearly the same in every arm, though the pull weights differ a lot. On `difference`, α at the example answers rises from near zero at the embedding to about 0.64 at slice 4 in every arm. The two ramps from slice 1 are a little lower at slices 1 and 2. So where the anchor sits follows the step at slice 4, and the lighter pull earlier does not move it.

On the other ops the alignment peaks at slice 1 in every arm. The ramps from the embedding are about where `falloff` is there, though `falloff` does not pull slice 1, and the ramps from slice 1 are a little lower. So the pull on slice 1 adds little to the drift on the other ops.

arm	α at 1	α at 2	α at 3	α at 4	α other ops, 1	α other ops, 2	lightness slope
`falloff`	0.35	0.56	0.62	0.64	0.18	0.10	−0.06 (−0.11 to −0.02)
`ramp from emb`	0.37	0.55	0.62	0.64	0.19	0.09	−0.07 (−0.13 to 0.00)
`ramp from 1`	0.25	0.50	0.61	0.64	0.15	0.08	−0.02 (−0.03 to −0.01)
`ramp from emb, step total`	0.38	0.56	0.62	0.65	0.19	0.09	−0.10 (−0.14 to −0.06)
`ramp from 1, step total`	0.29	0.52	0.62	0.64	0.15	0.08	−0.04 (−0.06 to −0.01)
controls	–	–	–	–	–	–	0.00 (−0.21 to 0.40)

***Alignment by slice and the lean of the color table.** α at the example answers on `difference` and on the other ops, seed mean over the runs that kept the anchor. The lightness slope is how much further along e_1 black sits than white in the color embedding table, seed mean and range over every run.*

The lean in the color table stays small in every arm, close to that of `falloff` and well inside the spread of the controls. Within that, it grows with the pull on the first slices: the ramps from slice 1 lean least, then `falloff`, then the ramps from the embedding, and every arm leans the same way on every seed, where the controls lean either way.

But these are differences of a few hundredths. Before the late-pull experiment moved the pull off the embedding, the recipe had a slope of about −1. So a light pull on slice 1 does not bring that lean back.

Discussion

A ramp with the step weight at slice 4 changes how often a run anchors at all, rather than how well: it loses the anchor on more seeds than the step, and the runs that keep it end up much like `falloff`.

Two readings fit. In one, the shape of the ramp matters. The pull is pooled over the positions of each context, so at slice 4, where the query answer is already on e_1, it asks little of the example answers; a ramp moves the pull toward slice 4, and so away from the slices where the example answers are what it pulls. In the other, what matters is how much pull the earlier slices get, and the shape matters only through that.

The losses follow the pull on slices 1 to 3 across all five arms, which favors the second reading, but the counts are small, that sum was chosen after the counts were in, and the arms differ in more than it.

If the second reading holds, a ramp could keep the anchor as long as the earlier slices get as much pull as the step gives them, and `ramp from emb, step total` is close to that. It then has no slice boundary for the pull, and its cost is the spill° on two of eight seeds. The step in `falloff` gives the earlier slices more, at no cost we could see here.

Spill. The side effect of an edit on the other ops…

Method

The per-slice pull weight is `AnchorSpec.slice_w`, added for this experiment. It multiplies each slice's pooled term inside the mean over the pulled slices, as the per-slice `AntiSpec.hold` does for the anti term, and the arm module converts the weights relative to the step into that form. Training follows the split-anti-seeds experiment unchanged.

The total-matched arms ran after the first two, with the same seeds and everything else unchanged. The 32 new runs cost about $11.