What the edit does on τ_ℓ fixed

> We scored the edit on the recipe of record, at its fresh seeds, the way the *edit-effects report*° scored the recipe before it. We also added the confusion matrices of *ex-2.2.22*°. The two recipes come out about the same on every measure, apart from a bound° at slice 3 that is wider on the seed mean, set by a small tail of states, many of them near misses° for `difference`. The edit takes out more of `difference` as the dose grows, and the drop follows what each context needed. The edited answers get about half of the way to the target null°. And the blocks after the edit put back no more e_1 on the other ops than the clean model has there.

edit-effects report. On `falloff`, the write stays within the clean bound, the drop is graded by the posterior, and the edited answers get about half of the way to the target null, with the shortfall in how the edited model weighs the other ops context by context.

ex-2.2.22. On an earlier recipe the edit got a little under half of the way to the target null, and the confusion matrices pooled over contexts put the mass about where the target null does.

Bound. From the clean pass…

Near miss. An example answer that does not fit, but is one step of the color grid from an answer `difference` can give on its operands…

Target null. Where a removal should land…

The *fixed confirmation*° adopted `τ_ℓ fixed` as the recipe of record for D2.2, and scored it on removal°, spill, the task, and lost anchors. The D2.2 design doc asks for the rest of what the edit does on the same runs before the blog post: removal by dose, the grading, the landing° and the write° bound of the *edit-effects report*, and the confusion matrices of *ex-2.2.22*.

fixed confirmation. `τ_ℓ fixed` keeps the anchor on all eight runs at seeds 712 to 719 and all forty in the count, with removal of 0.88 on the seed mean and no run near the spill criterion; it is the recipe of record.

Removal. The drop in expected exact match on `difference` under the edit, as a share of the way from the clean score to the target null…

Landing. How far the edit moves the answers on `difference` contexts toward the target null…

Write. The same quantile, over the states as they arrive under the full edit…

Alignment. How far a state points along e_1…

This report needs no new runs. The *fixed confirmation* stored what this report reads: the dose scoring, the scoring along the other 63 axes, and the alignment° with e_1 at every slice and position. E4 also reads the colors of the held-out examples. The `falloff` runs from the *edit-effects report* are scored the same way alongside.

Observations

- Removal and spill by dose (E1)°: removal rises with the dose, slowly over the first quarter of the edit and steeply over the second, and it is complete by three quarters. On 7 runs of 8, the other ops lose less under the edit along e_1 than under the same edit along a typical axis, at every dose. No run comes near the spill criterion.

Removal. The drop in expected exact match on `difference` under the edit, as a share of the way from the clean score to the target null....

- Whether the drop follows what each context needed (E2): the edit takes about the same share of the way to the target null° whatever the posterior on `difference`, and, before the controls are subtracted, the two recipes lose the same amount bin by bin. On the contexts of other ops whose examples point at `difference`, it gives back part of what those examples cost, as on `falloff`.

Target null. Where a removal should land...

- Where the edited answers go (E3): the edited answers get about half of the way to the target null, a little less far than a model trained without `difference`. Pooled over contexts and runs, the `difference` row of the confusion matrix matches the target null in total at every level of the posterior, though `lighten` gets well under its share. The other ops lose a little more mass to each other than on the clean pass.

- How far the edit turns the other ops (E4): the write° stays within the clean bound° past the embedding, as on `falloff`. The bound at the answers of the other ops is wider at slice 3 than on `falloff` on the seed mean. It is set by about one answer in a hundred, and about half of those are near misses°: one step of the color grid from an answer `difference` could give. The write there is small.

Write. The same quantile, over the states as they arrive under the full edit...

Bound. From the clean pass...

Near miss. An example answer that does not fit, but is one step of the color grid from an answer `difference` can give on its operands...

Scope and method

This is an exploratory re-analysis, with no preregistration and no gate. It covers the eight runs of `τ_ℓ fixed` that the *fixed confirmation*° trained to the end, at seeds 712 to 719, and the eight controls at the same seeds from the *anneal confirmation*°, trained without the anchor.

fixed confirmation. `τ_ℓ fixed` keeps the anchor on all eight runs at seeds 712 to 719 and all forty in the count, with removal of 0.88 on the seed mean and no run near the spill criterion; it is the recipe of record.

anneal confirmation. With the slice temperature annealing, the pool keeps the anchor on all eight runs at seeds 712 to 719; its controls at those seeds are the ones used here.

The comparison is `falloff` at seeds 704 to 711, the runs of the *edit-effects report*°, with the eight controls of ex-2.2.23 at those seeds. The two recipes ran at different seeds, so we compare them as two samples of eight runs: the difference of the seed means, with a 95% Welch t interval, which does not assume the two recipes vary alike. The intervals are wide, since the end of a run is sensitive to small changes early in training (the *fixed confirmation*). With eight runs a side, a difference whose interval only just clears zero is suggestive rather than settled.

edit-effects report. On `falloff`, the write stays within the clean bound, the drop is graded by the posterior, and the edited answers get about half of the way to the target null, with the shortfall in how the edited model weighs the other ops context by context.

Every measurement is on the 14,000 held-out contexts of ex-2.2.21, 2,000 per op, each with three examples and a query. The edit removes a share of the e_1 component from the state at every position and every slice. That share is the dose, and E2 to E4 use the full dose.

The scores take different baselines, as the stored data allows. Removal° and spill (E1) are net of the median drop along the other 63 coordinate axes of the run, the rule of the D2.2 design. That scoring keeps only a mean per op, so the scores by context (E2) are net of the mean drop of the eight controls of the recipe under the same edit, context by context. By the end of training a control is no closer to the anchored run at its own seed than to one at another seed (eng/determinism.md), so the mean of the eight is the steadier baseline. The landing° (E3) has no baseline, and the write° (E4) is compared with the clean pass of the same run.

Removal. The drop in expected exact match on `difference` under the edit, as a share of the way from the clean score to the target null….

Landing. How far the edit moves the answers on `difference` contexts toward the target null…

Write. The same quantile, over the states as they arrive under the full edit…

The measurements

Removal

The drop in expected exact match on `difference` under the edit, as a share of the way from the clean score to the target null. Higher is better.

Spill

The largest drop on any other op. Lower is better, and the selectivity criterion is 0.02.

Share removed

On a group of `difference` contexts, the mean drop divided by the mean distance from the clean score to the target null. The target null gives a share of 1 in every group. So a share that is level across the posterior means the drop is graded the way the target null predicts.

Gain

On the contexts of other ops, the rise in expected exact match under the edit. The target null predicts a gain where the examples pointed at `difference`, and none elsewhere.

Landing

How far the edit moves the answers on `difference` contexts toward the target null: the share of the clean distance that it covers, in total variation.

Distance left

The total variation between the edited answers and the target null, on `difference` contexts.

Bound

From the clean pass: arcsin of the 99% quantile of |α| over the states of the other ops at a slice and position, as an angle. It is how far the edit turns those states there. At the answers it counts only those that do not fit `difference`.

Write

The same quantile, over the states as they arrive under the full edit: that is, after the preceding block has read the edited states of the slices below. The write counts as within the bound when it is at most a quarter over it (the tolerance of H4 in ex-2.2.1). We quote the ratio.

Fit

An example or query answer fits when `difference` can give it on its operands. In a context of another op, about one example answer in 17 fits. It is evidence for `difference`, so the edit is meant to turn it.

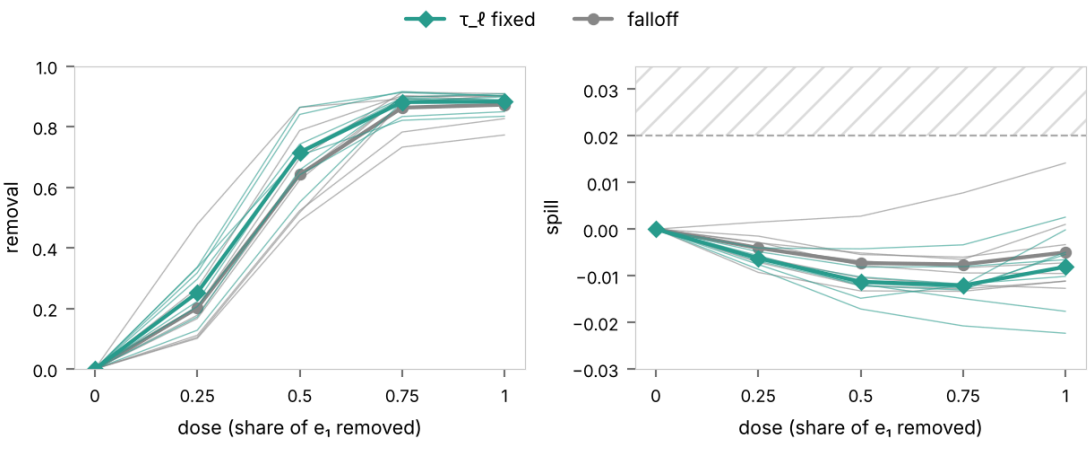
Near miss

An example answer that does not fit, but is one step of the color grid from an answer `difference` can give on its operands: one level away in one of the three channels of the 6 × 6 × 6 palette. About 4% of the example answers of other ops that do not fit are near misses. E4 reads the colors of the examples from the held-out set to find them.

Removal and spill by dose (E1)

A smaller dose takes a share of e_1 out rather than all of it. We expect removal° to grow with the dose, and spill to stay low at every dose, as it did on `falloff`.

Removal. The drop in expected exact match on `difference` under the edit, as a share of the way from the clean score to the target null....



***Removal and spill by dose**, the edit at every position and slice, net of the median drop along the other 63 coordinate axes of the run. One hairline per run and the seed mean bold, for `τ_ℓ fixed` at seeds 712 to 719 and `falloff` at 704 to 711; dose 0 is the clean pass. **Left:** removal, the share of the way from the clean score on `difference` to the target null. **Right:** spill, the largest net drop on any other op; the dashed line is the criterion, with the side over it hatched.*

Removal grows with the dose on every run, from zero at dose 0, which is the clean pass. It rises slowly at small doses: the first quarter of the edit removes 29% of what the full edit does (14% to 40% by run), and the second quarter much more. By three quarters of the full edit, it has done all that the full edit does: on the seed mean, removal at that dose already matches removal at the full dose. Past three quarters it moves by no more than 0.02 either way on any run.

`falloff` has the same shape. On the seed mean it is a little lower at the smaller doses, by less than two sets of eight runs can tell apart: at half the edit, `τ_ℓ fixed` removes 0.07 (95% interval −0.06 to 0.20) more.

The slow rise at small doses is not in how far the edit turns the states at slice 1. The dose is the share of the e_1 component taken out of each state before it is put back on the sphere, so how far a dose turns a state depends on how close the state is to e_1. At slice 1, where block 1 writes° about half of the evidence for the op (the *block1-edit experiment*°), the states holding `difference` are far enough from e_1 that a dose turns them by about that share of the full turn: 23% of it at a quarter of the edit, and 47% at half.

Write. The same quantile, over the states as they arrive under the full edit...

block1-edit experiment. Block 1 writes about half of the evidence for the op onto e_1 at the example answers, as a judgement on each example, and the later blocks mostly add to the e_1 that arrives rather than computing it again.

At slices 2 to 4 the states holding `difference` sit nearly on e_1 (median $|\alpha^\circ|$ of 0.78, 0.98 and 0.75), where a partial dose turns a state far less than its share, and putting the state back on the sphere gives most of the alignment back: at slice 3, three quarters of the edit leaves 0.81 of the clean alignment. So at a single slice the turn is back-loaded, with most of it in the last quarter of the edit, where the measured removal has already finished. The bend has to come from how the slices compound: the edit at slice 1 starves what arrives at the later slices, which the edit there cuts by the same share again.

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

Spill is below zero at every dose on 7 runs of 8: the other ops lose less under the edit along e_1 than under the same edit along a typical other axis. The largest spill of any run at any dose is 0.003, on seed 717 (`value-hsv`, at the full dose), far below the criterion.

recipe, measure	dose 0.25	dose 0.5	dose 0.75	dose 1
`τ_ℓ fixed`, removal ↑	0.25 (0.13 to 0.34)	0.71 (0.55 to 0.86)	0.88 (0.82 to 0.92)	0.88 (0.83 to 0.91)
`falloff`, removal ↑	0.20 (0.10 to 0.48)	0.64 (0.49 to 0.86)	0.86 (0.73 to 0.91)	0.87 (0.77 to 0.91)
`τ_ℓ fixed`, spill ↓	−0.006 (−0.009 to −0.004)	−0.011 (−0.017 to −0.004)	−0.012 (−0.021 to −0.003)	−0.008 (−0.022 to 0.003)
`falloff`, spill ↓	−0.004 (−0.009 to 0.002)	−0.007 (−0.013 to 0.003)	−0.008 (−0.013 to 0.008)	−0.005 (−0.013 to 0.014)

***Removal and spill by dose**, seed mean and range over eight runs, net of the median over the other 63 axes of the run: `τ_ℓ fixed` at seeds 712 to 719, `falloff` at 704 to 711. Spill is the largest net drop on any other op at that dose; the criterion is 0.02.*

So the edit is selective at every dose, and the full edit is past the point where more dose removes more.

Whether the drop follows what each context needed (E2)

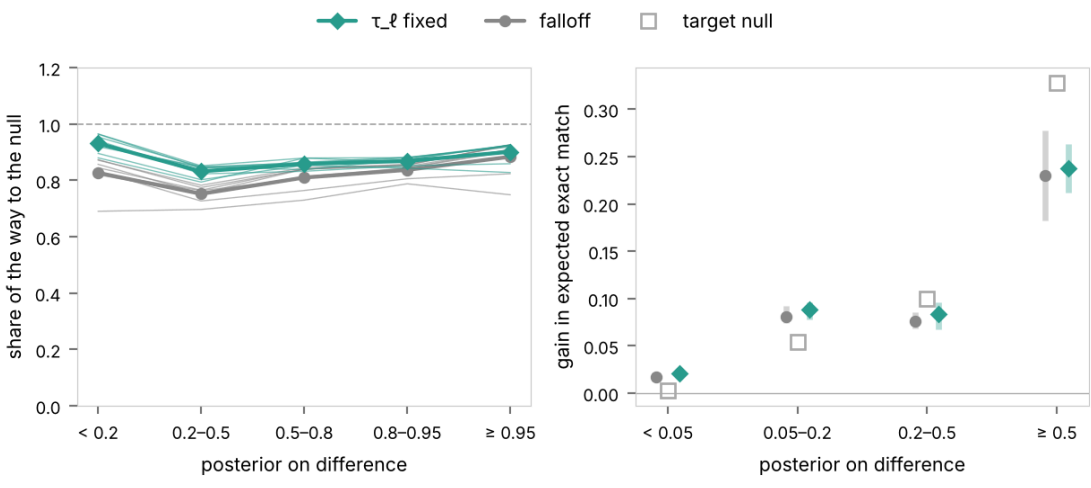
Target null. Where a removal should land...

The target null° predicts the drop context by context. A context whose examples all fit `difference` should lose most, and one whose examples point elsewhere should lose little. If the anchor holds the evidence for the op, the edit should take about the same share of the way to the target null in every bin of the posterior on `difference`.

If it held something coarser, such as a color feature of `difference` answers, it would take off about the same amount everywhere, which is a larger share on the contexts that needed the op least.

Gain. On the contexts of other ops, the rise in expected exact match under the edit....

The target null also predicts a gain° on the contexts of other ops: where an example showed the answer of `difference`, removing the op should move the answer back toward the true op.



The drop under the full edit, by how much each context needed `difference`. ***Left:*** *on `difference` contexts, the share of the way to the target null that the edit takes in each bin of the posterior on `difference`, as a ratio of bin means; one hairline per run and the seed mean bold, with the target null as the dashed rule at 1.* ***Right:*** *on the contexts of the other ops, the gain in expected exact match under the edit in each bin of the posterior on `difference` (these bins differ from the left), seed mean and range, with the gain the target null predicts as open squares. Both net of the mean of the controls.*

Share removed. On a group of `difference` contexts, the mean drop divided by the mean distance from the clean score to the target null....

The share removed° is about level across the posterior on both recipes, between 0.83 and 0.93 at the seed mean on `τ_ℓ fixed`. No run goes past the target null in any bin, so no bin shows the overshoot that a coarse feature would give. Without the controls subtracted (the last row of the table below), the lowest bin removes the smallest share of any.

The share is higher on `τ_ℓ fixed` than on `falloff` in the lower bins, but that comes from the two sets of controls rather than from the anchored runs. Without the controls subtracted, the two recipes lose the same bin by bin, and over all bins `τ_ℓ fixed` removes 0.01 (95% interval −0.03 to 0.04) more. The controls of `falloff` lose a little more under the edit than those of `τ_ℓ fixed`, so the net share comes out lower there.

Over all bins, without the controls subtracted, the edit takes `difference` contexts 0.96 (0.92 to 0.98) of the way to the target null on its own. Under the same edit, the controls lose 0.06 there on average; that is the part a model with no anchor loses too.

share removed, by posterior on `difference`	< 0.2	0.2–0.5	0.5–0.8	0.8–0.95	≥ 0.95	all
`difference` contexts	172	310	79	744	695	2,000
dependence (Bayes)	0.08	0.33	0.74	0.90	0.98	0.76
`τ_ℓ fixed`	0.93 (0.88 to 0.97)	0.83 (0.79 to 0.85)	0.86 (0.83 to 0.88)	0.87 (0.84 to 0.88)	0.90 (0.83 to 0.93)	0.88 (0.83 to 0.90)
`falloff`	0.83 (0.69 to 0.87)	0.75 (0.70 to 0.78)	0.81 (0.73 to 0.84)	0.84 (0.79 to 0.86)	0.88 (0.75 to 0.93)	0.85 (0.76 to 0.88)
`τ_ℓ fixed`, not net of the controls	0.88 (0.82 to 0.91)	0.94 (0.91 to 0.96)	0.98 (0.94 to 0.99)	0.97 (0.95 to 0.99)	0.96 (0.88 to 0.98)	0.96 (0.92 to 0.98)

The share of the dependence removed, by bin of the posterior*, on the held-out `difference` contexts: the mean drop under the full edit over the mean distance from the clean answer to the target null, in expected exact match, net of the mean drop of the eight controls unless marked; seed mean and range over eight runs. The dependence row is the drop the target null predicts for the Bayes predictor, which needs no model.*

On the contexts of other ops, the gain grows with the posterior on `difference`, as the target null predicts, and matches `falloff` bin by bin. Where the examples point most at `difference` (a posterior of at least 0.5), the edit gains 0.24 against 0.33 for the target null. In the two lowest bins it gains a little more than the target null. The *edit-effects report*° traced most of that excess to the controls, which lose a little there under the edit. A model trained without `difference` gains about what the target null predicts on these contexts (the *sans-diff experiment*°, E4 there).

edit-effects report. On `falloff`, the write stays within the clean bound, the drop is graded by the posterior, and the edited answers get about half of the way to the target null, with the shortfall in how the edited model weighs the other ops context by context.

sans-diff experiment. A model trained with no loss on `difference` contexts lands a little closer to the target null than the edited `τ_ℓ fixed` runs, 0.58 of the way from the control against 0.50, and misses in the same kind of way, with much of the miss changing from run to run.

gain, by posterior on `difference`	< 0.05	0.05–0.2	0.2–0.5	≥ 0.5
contexts of other ops	10,180	1,042	664	114
target null	0.00	0.05	0.10	0.33
`τ_ℓ fixed`	0.02 (0.02 to 0.03)	0.09 (0.08 to 0.10)	0.08 (0.07 to 0.10)	0.24 (0.21 to 0.26)
`falloff`	0.02 (0.01 to 0.02)	0.08 (0.07 to 0.09)	0.08 (0.07 to 0.09)	0.23 (0.18 to 0.28)

The gain on the contexts of the other ops*, in expected exact match under the full edit, net of the mean of the controls, by bin of the posterior on `difference`; seed mean and range over eight runs. The target null row is the gain it predicts for the Bayes predictor.*

So the drop is graded on the recipe of record as it was on `falloff`: in proportion to what each context needed, with the contexts of other ops recovering where their examples pointed at `difference`.

Where the edited answers go (E3)

On `difference` contexts the full edit covers 0.50 (0.47 to 0.52) of the distance from the clean answers to the target null°, about the same as on `falloff`, which lands 0.02 (95% interval −0.01 to 0.05) further. The landing° is a stricter measure than the share removed° in E2, which scores only the probability on the right answer. The landing scores the whole answer distribution, including how the mass that leaves `difference` is shared among the other ops.

Target null. Where a removal should land…

Landing. How far the edit moves the answers on `difference` contexts toward the target null…

Share removed. On a group of `difference` contexts, the mean drop divided by the mean distance from the clean score to the target null….

on `difference` contexts	`τ_ℓ fixed`	`falloff`
landing ↑	0.50 (0.47 to 0.52)	0.52 (0.49 to 0.56)
distance left ↓	0.40 (0.38 to 0.42)	0.39 (0.36 to 0.41)
clean to Bayes	0.16 (0.15 to 0.17)	0.16 (0.15 to 0.19)

***How far the edited answers are from the target null**, in total variation on the held-out `difference` contexts, seed mean and range over eight runs. The landing is the share of the clean distance to the target null that the full edit covers. The distance from the clean model to the Bayes predictor it learned to match is a yardstick.*

The distance left° is about 2.5× the distance between the clean model and the Bayes predictor° it was trained to match.

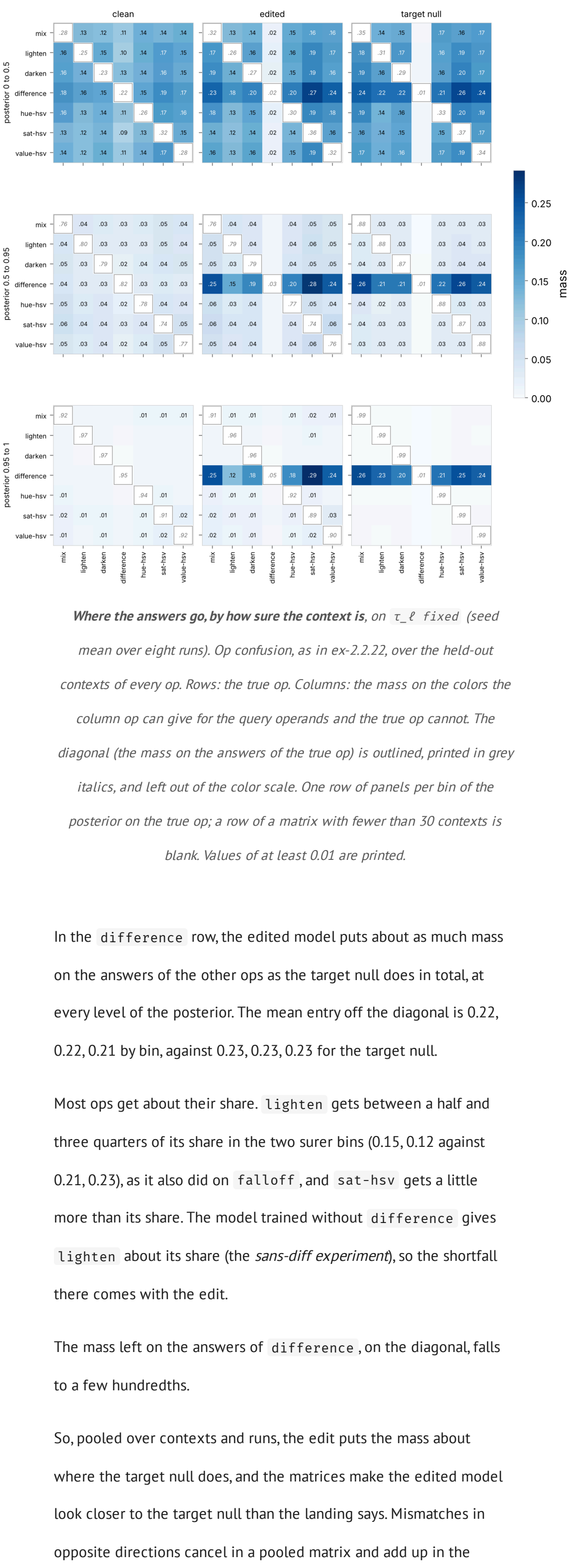
Distance left. The total variation between the edited answers and the target null, on `difference` contexts.

Bayes predictor. The best answer given the examples…

The *sans-diff experiment*° looks at this distance on the same runs, beside a model trained without `difference` (E2 and E3 there). Nearly all the mass the edit moves goes to answers another op could give, and the miss is in how the edited model shares it among those ops on each context, as edit-effects found on `falloff`. The model trained without `difference` lands only a little closer to the target null, and misses in the same kind of way, so most of this distance is there without the anchor and the edit.

***sans-diff experiment*.** A model trained with no loss on `difference` contexts lands a little closer to the target null than the edited `τ_ℓ fixed` runs, 0.58 of the way from the control against 0.50, and misses in the same kind of way, with much of the miss changing from run to run.

The confusion matrices pool over contexts, and here over the eight runs too, so they show the overall proportions. They put the clean model, the edited model, and the target null side by side, at three levels of how sure the context is about its true op.



***Where the answers go, by how sure the context is**, on `τ_ℓ fixed` (seed mean over eight runs). Op confusion, as in ex-2.2.22, over the held-out contexts of every op. Rows: the true op. Columns: the mass on the colors the column op can give for the query operands and the true op cannot. The diagonal (the mass on the answers of the true op) is outlined, printed in grey italics, and left out of the color scale. One row of panels per bin of the posterior on the true op; a row of a matrix with fewer than 30 contexts is blank. Values of at least 0.01 are printed.*

In the `difference` row, the edited model puts about as much mass on the answers of the other ops as the target null does in total, at every level of the posterior. The mean entry off the diagonal is 0.22, 0.22, 0.21 by bin, against 0.23, 0.23, 0.23 for the target null.

Most ops get about their share. `lighten` gets between a half and three quarters of its share in the two surer bins (0.15, 0.12 against 0.21, 0.23), as it also did on `falloff`, and `sat-hsv` gets a little more than its share. The model trained without `difference` gives `lighten` about its share (the *sans-diff experiment*), so the shortfall there comes with the edit.

The mass left on the answers of `difference`, on the diagonal, falls to a few hundredths.

So, pooled over contexts and runs, the edit puts the mass about where the target null does, and the matrices make the edited model look closer to the target null than the landing says. Mismatches in opposite directions cancel in a pooled matrix and add up in the landing. The *sans-diff experiment* found that much of the mismatch on a context changes from run to run: averaging the answers of the eight runs, context by context, brings the edited model about a third closer to the target null, and averaging does about as much for a model on contexts where its ideal is not in doubt. So the matrices average away two kinds of scatter, over contexts and over runs, and what is left in them, such as the shortfall on `lighten`, is common to the runs.

In the rows of the other ops, the `difference` column empties under the edit. Where the context is least sure of its true op, the clean model puts 0.11 on the answers only `difference` gives, and the edited model 0.02, near the zero of the target null. This is the gain° of E2 seen from the other side.

Gain. On the contexts of other ops, the rise in expected exact match under the edit….

The rest of those rows change little. Even on the clean pass, their diagonal is lower than that of the target null, since the clean model is some way from the Bayes predictor. Where the context is least sure, the mass that leaves the `difference` column goes to the true op and to the other ops alike, and both moves are toward the target null.

In the two surer bins, the edit moves a little more mass off the diagonal onto the other ops, which is away from the target null. The mass the other ops put on each other is 0.20, 0.05 clean against 0.23, 0.06 edited, and 0.15, 0.01 for the target null.

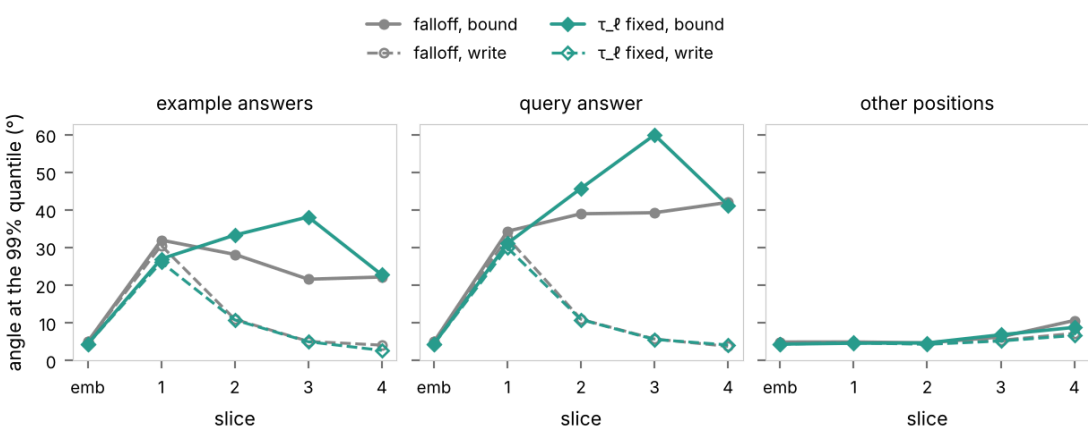
How far the edit turns the other ops (E4)

The edit turns each state by an angle: the arcsin of the alignment° (α) the state arrives with. So the clean pass sets how far the edit can turn the states of the other ops, and that is the bound°. If the blocks after an edit put e_1 back on the other ops, beyond what the clean model has there, the write° would rise above the bound.

Alignment. How far a state points along e_1…

Bound. From the clean pass…

Write. The same quantile, over the states as they arrive under the full edit…



***The bound and the write on the contexts of the other ops, by slice**, at three roles: the example answers that do not fit `difference`, the query answer, and the largest over every other position. The 99% quantile of $|\alpha|$ as an angle, clean (the bound, solid) and as it arrives under the full edit (the write, dashed), seed mean over eight runs of `τ_ℓ fixed` (seeds 712 to 719) and of `falloff` (704 to 711). The edit acts at every slice, so the write at a slice is the α a state arrives with there, before the edit at that slice takes e_1 out.*

The write stays within the bound past the embedding. At the seed mean the largest ratio of write to bound is 1.09, at slice 1, position `?`, where the bound is under a degree. Run by run, 34 sites of 768 go over the tolerance of ex-2.2.1, all where the bound is at most 3.5°, and none by more than 1.2°. That is about the same as on `falloff`.

recipe	largest write / bound ↓	where (bound)	sites over ↓	largest bound there	largest excess
`τ_ℓ fixed`	1.09	slice 1, `?` (0.3°)	34 of 768	3.5°	1.2°
`falloff`	1.08	slice 1, `?` (0.3°)	28 of 768	4.2°	1.7°

***The write against the bound past the embedding.** The largest ratio of write to bound in angle on the seed mean, over every slice and position past the embedding, and where it falls. The sites over count slice and position pairs over all eight runs, each run taken on its own, where the write is more than 1.25× the bound (the tolerance of ex-2.2.1); the last two columns give the widest bound among them and the largest excess of the write over the bound there.*

From slice 2 on, the other ops arrive with much less e_1 than the clean model gives them, as on `falloff`: at the example answers that do not fit, the write is 32% of the bound at slice 2 and 12% at slice 4. The *edit-effects report*° traced this to the edit at the slices before, which leaves each block nothing on e_1 to carry through. The *block1-edit experiment*° found why that is enough: the later blocks mostly add to the e_1 that arrives, rather than computing it again.

edit-effects report. On `falloff`, the write stays within the clean bound, the drop is graded by the posterior, and the edited answers get about half of the way to the target null, with the shortfall in how the edited model weighs the other ops context by context.

block1-edit experiment. Block 1 writes about half of the evidence for the op onto e_1 at the example answers, as a judgement on each example, and the later blocks mostly add to the e_1 that arrives rather than computing it again.

The bound itself has a different shape. On `falloff` it was widest at slice 1 at the example answers and narrowed with depth. On `τ_ℓ fixed` it widens to slice 3, to 38° against 22° on `falloff`, and the query answer does the same (60° against 39°). Taken run by run, the bound at the example answers at slice 3 is wider on `τ_ℓ fixed` by 18.8° (95% interval 0.2° to 37.5°). The interval only just clears zero, so the difference is suggestive rather than settled; the bound is wider at slice 3 than at slice 1 on 5 runs of `τ_ℓ fixed` and also on 3 of `falloff`.

A small tail sets the bound at slice 3. The median example answer that does not fit would turn by 2° there, while the share close to e_1 ($|\alpha|$ above 0.5) is 0.4% to 3.0% by run (0.3% to 2.9% on `falloff`).

The bound is the 99th percentile, so a tail near one in a hundred moves it a long way: it ranges from 12° to 71° across the runs of `τ_ℓ fixed`, and from 11° to 43° across those of `falloff`.

On 7 runs of 8, the contexts in the tail have about as little posterior on `difference` as the rest, as the *edit-effects report* found for a tail at the query answer on `falloff`. But the posterior counts only answers that fit exactly. The *block1-edit experiment* found that block 1 writes the evidence for the op as a judgement on each example, whether its answer fits `difference` on its operands, and that blocks 2 and 3 build on what block 1 writes. A judgement of that kind made from colors could be graded, giving an answer that nearly fits some of the weight of one that fits.

The tail fits that account. Of the answers in the tail, 27% to 71% by run are near misses°, against about 4% of all the answers of other ops that do not fit (14% to 67% on `falloff`). The nearer an answer comes to one that fits, the more often it is in the tail, and the rate falls off steeply past one step of the grid (the table below). That does not come from how dark the answers are: in each quarter of them by darkness, near misses are in the tail 11% to 18% of the time, against about 1% for the rest.

Near miss. An example answer that does not fit, but is one step of the color grid from an answer `difference` can give on its operands…

grid steps from a `difference` answer	1	2	3	4	5
answers per run	1,268	2,751	4,210	5,133	5,297
`τ_ℓ fixed`	15.5%	3.8%	1.7%	0.8%	0.6%
`falloff`	7.3%	2.0%	1.2%	0.7%	0.5%

***How often an answer is in the tail, by how near it comes to fitting**: of the example answers of the other ops that do not fit `difference`, the share with $|\alpha|$ above 0.5 at slice 3 on the clean pass, by the number of steps of the color grid to the nearest answer `difference` could give on its operands, pooled over the eight runs of each recipe. One step is a near miss.*

Much of the tail is the model taking a near miss as weak evidence for `difference`, and turning those states is partly removal° at work, as with the answers that fit. That leaves the rest of the tail unexplained, and does not say why the tail is larger on some runs than on others.

Removal. The drop in expected exact match on `difference` under the edit, as a share of the way from the clean score to the target null…

answers of the other ops	emb	1	2	3	4
`τ_ℓ fixed`, do not fit	4° → 4°	27° → 26°	33° → 11°	38° → 5°	23° → 3°
`τ_ℓ fixed`, fit	5° → 5°	34° → 31°	67° → 24°	85° → 9°	54° → 3°
`falloff`, do not fit	5° → 5°	32° → 31°	28° → 11°	22° → 5°	22° → 4°
`falloff`, fit	5° → 5°	37° → 34°	71° → 26°	84° → 13°	84° → 5°
`τ_ℓ fixed`, query answer	4° → 4°	31° → 30°	46° → 11°	60° → 6°	41° → 4°
`falloff`, query answer	5° → 5°	34° → 33°	39° → 11°	39° → 6°	42° → 4°

***The bound and the write at the answers of the other ops**, by slice: arcsin $|\alpha|$ at the 99% quantile over those answers, clean → as it arrives under the full edit. Seed mean over eight runs, each quantile taken per run and then averaged. An example answer fits when `difference` could give it on its operands (2,112 answers per run, against 33,888 that do not); at the query answer only those that do not fit count.*

So the bound holds on the recipe of record in the sense of ex-2.2.1: the blocks after the edit put back less than the clean model has. Where the tail is large, the bound is loose: the turn we can state in advance at slice 3 is large for about one state of another op in a hundred, many of them near misses, though most turn by a couple of degrees. The spill of E1 stays low, so whatever the later blocks make of those turns costs the other ops little.

Discussion

On the recipe of record, the edit does what it did on `falloff` on every measure here, apart from the wider bound° at slice 3. It removes more of `difference` as the dose grows, and leaves the other ops alone at every dose. The drop on each context follows how much that context needed the op. And the blocks after the edit put back less e_1 than the clean model has. These are claims (b) and (d) of the design doc, now shown on the recipe of record at fresh seeds.

Bound. From the clean pass…

The landing° is where the edit falls short, as before. Pooled over contexts the edited model sends the mass of `difference` to the other ops in about the proportions of the target null°, but context by context it weighs them differently. With both recipes landing in the same place, the shortfall looks like a property of the model or of the edit rather than of the recipe. The *sans-diff experiment*° points to the model: trained without `difference`, it lands only a little closer to the target null than the edited model, and misses in the same kind of way, sharing the mass among the other ops differently on each context and at each seed. The anchor and the edit add a smaller part on top, and the shortfall on `lighten` in the pooled matrices is part of what they add.

Landing. How far the edit moves the answers on `difference` contexts toward the target null…

Target null. Where a removal should land…

***sans-diff experiment*.** A model trained with no loss on `difference` contexts lands a little closer to the target null than the edited `τ_ℓ fixed` runs, 0.58 of the way from the control against 0.50, and misses in the same kind of way, with much of the miss changing from run to run.

Two results add to what the *edit-effects report*° showed. Removal° is nearly complete by three quarters of the full edit, so the full edit has some margin. And the bound at slice 3 is wider than on `falloff` on the seed mean, set by a tail of about one state of another op in a hundred that varies a lot between runs. Much of the tail is near misses° for `difference`, which the model seems to take as weak evidence for the op, so some of that turn is removal rather than a side effect. That does not change what the edit does to the answers here, but it makes the turn we can state in advance larger. If a later recipe spills, the tail at slice 3 would be a place to look.

***edit-effects report*.** On `falloff`, the write stays within the clean bound, the drop is graded by the posterior, and the edited answers get about half of the way to the target null, with the shortfall in how the edited model weighs the other ops context by context.

Removal. The drop in expected exact match on `difference` under the edit, as a share of the way from the clean score to the target null….

Near miss. An example answer that does not fit, but is one step of the color grid from an answer `difference` can give on its operands…

References

fixed confirmation

`τ_ℓ fixed` keeps the anchor on all eight runs at seeds 712 to 719 and all forty in the count, with removal of 0.88 on the seed mean and no run near the spill criterion; it is the recipe of record.

anneal confirmation

With the slice temperature annealing, the pool keeps the anchor on all eight runs at seeds 712 to 719; its controls at those seeds are the ones used here.

edit-effects report

On `falloff`, the write stays within the clean bound, the drop is graded by the posterior, and the edited answers get about half of the way to the target null, with the shortfall in how the edited model weighs the other ops context by context.

sans-diff experiment

A model trained with no loss on `difference` contexts lands a little closer to the target null than the edited `τ_ℓ fixed` runs, 0.58 of the way from the control against 0.50, and misses in the same kind of way, with much of the miss changing from run to run.

block1-edit experiment

Block 1 writes about half of the evidence for the op onto e_1 at the example answers, as a judgement on each example, and the later blocks mostly add to the e_1 that arrives rather than computing it again.

ex-2.2.22

On an earlier recipe the edit got a little under half of the way to the target null, and the confusion matrices pooled over contexts put the mass about where the target null does.