# What the edit does on the recipe of record

> We looked at what the edit does on the recipe of record: how far it turns° the states of the other ops, how much each `difference` context loses, and where the edited answers go. The turn stays within the bound° the clean geometry sets. The loss on each context follows how much it needed `difference`. The edited answers go to the other ops in about the right overall proportions, but context by context they weigh those ops differently from the target null°, and that is most of the distance left°.

**Write.** How far the edit turns a state…

**Within the bound.** The write is no larger than the bound over the same states.…

**Target null.** Where a removal should land…

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

The D2.2 design doc lists three claims that the recipe of record has no measurement for yet. None needs new training.

- (d) The side effects are bounded, and the bound° can be stated in advance from the clean model. That the edit turns a state by arcsin of its alignment° holds by construction. What has to be shown is downstream: the blocks after an edit put no more $e_1$ back on the other ops than the clean model has there. This is the write bound.
- (b) Removing the op is graded: the drop on each context follows how much its answer depended on `difference`, as the target null predicts context by context.
- (c) The edited model answers in a predictable way: once `difference` is removed, it answers as the target null does, weighing the other ops by how well they fit the examples.

**Clean bound.** From the clean pass of the anchored model, before any edit…

**Alignment.** How far a state points along $e_1$…

This study measures all three on the eight `falloff` runs of split-anti-seeds, the experiment that set the current recipe of record. It uses the dose scoring that experiment stored, plus one new pass over the held-out contexts. That pass records the alignment with $e_1$ at every slice and position, clean and under the edit.

# Observations

- How far the edit turns the other ops (E1): the write° stays within the clean bound° at every slice past the embedding, at most 1.08 times it (seed mean). From slice 2 on, the other ops arrive with less $e_1$ than the clean model gives them, and less than the control has. The bound is widest at the answers, and at the example answers it is widest at slice 1. Example answers of other ops that fit `difference` are evidence for it; they sit on $e_1$, and the edit turns them most.

**Write.** How far the edit turns a state…

**Clean bound.** From the clean pass of the anchored model, before any edit…

- Whether the drop follows what each context needed (E2): the edit takes about the same share of the way to the target null° on contexts that barely needed `difference` as on those that needed it most. Some contexts of other ops have examples that point at `difference`; there the edit gives back some of what those examples cost, as the target null says it should.

**Target null.** Where a removal should land…

- Where the edited answers fall short (E3): almost all the mass leaves the answers only `difference` gives. It lands on the answers of the other ops, in about the right overall proportions. What is left is mostly how the edited model weighs those ops context by context, and those weights are no flatter or sharper than the target null on the whole. The distance left° is about 2.4× the distance between the clean model and the Bayes predictor° it was trained to match.

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

**Bayes predictor.** The best answer given the examples…

# Scope

This is an exploratory re-analysis, with no preregistration and no gate. It covers the `falloff` arm of split-anti-seeds (the recipe of record) at model seeds 704 to 711. It also covers the control at each of those seeds from ex-2.2.23, which was trained without the anchor at the same initialization, batches, and label draws.

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 is the full edit of the recipe: the whole $e_1$ component taken out of the state at every position and every slice. E2 and E3 use the full dose only, since split-anti-seeds already showed that the removal grows with the dose.

With eight runs, a pattern seen on every seed is one we can lean on; a difference between seeds is noted and not explained. E3 confirms on the recipe of record what ex-2.2.22 found informally on an earlier recipe, and takes it one step further.

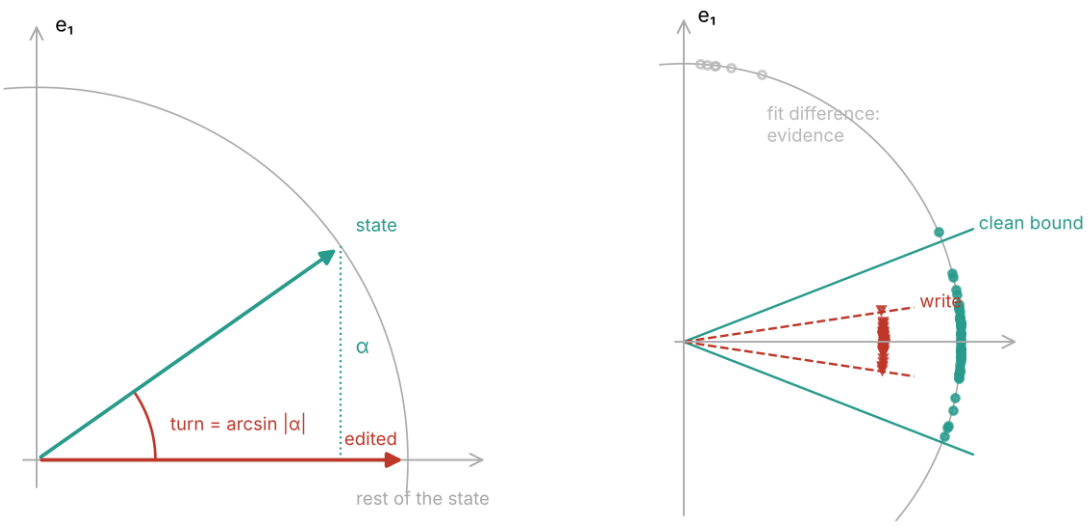
# Method

The states are unit vectors, and the edit takes the $e_1$ component out of each one and renormalizes. That turns° each state by an angle set by its alignment°, arcsin |α|, so how far the edit moves a state can be read off the clean geometry before any edit is made. The first figure draws this, with the bound° and the write of E1. That part holds by construction; the second figure shows the part that does not, where the write is taken along the depth of the model. The terms are defined in the glossary at the end, and each shows in the margin at its first use in a section.

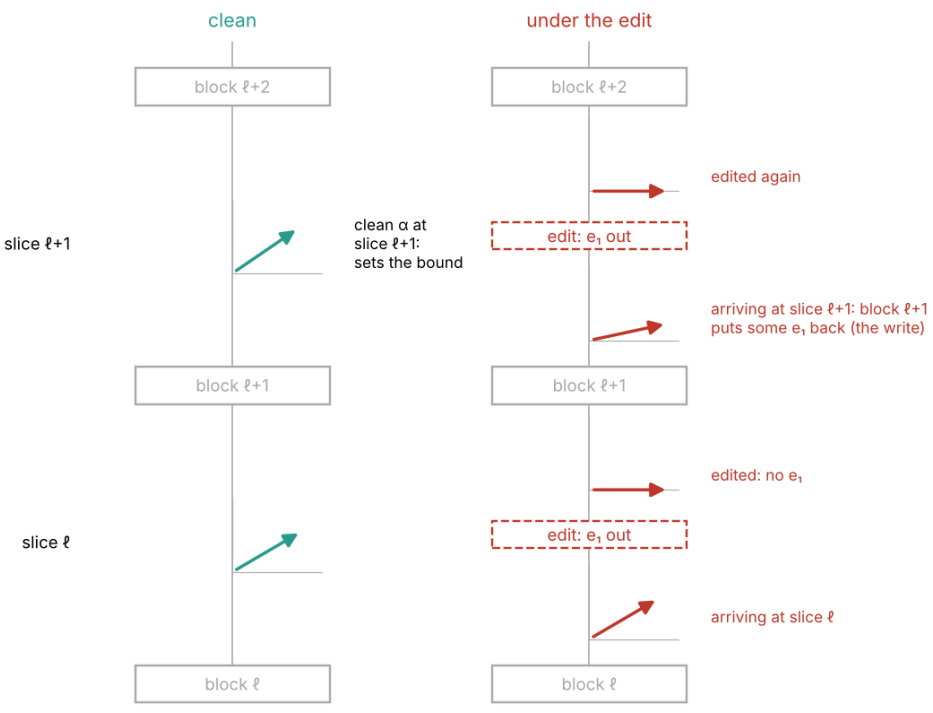
**Write.** How far the edit turns a state…

**Alignment.** How far a state points along $e_1$…

**Clean bound.** From the clean pass of the anchored model, before any edit…




***The geometry of the edit, drawn schematically from made-up points. Left:*** *a unit-length state in the plane of $e_1$ and the rest of its direction. Its alignment α is its component along $e_1$. The edit takes that component out and renormalizes, which turns the state by arcsin |α|.* ***Right:*** *the states of the other ops at one slice and position, clean on the outer ring and as they arrive under the edit on the inner ring. The clean bound is the angle that holds 99% of the clean states (solid wedge), and the write is the angle that holds 99% of the arriving ones (dashed wedge). The ideal is a narrow bound with the write inside it. Example answers that fit* `difference` *(hollow) sit near $e_1$ as evidence for it, and the bound at the example answers leaves them out.*




***Where the write is taken, along the depth of the model****, drawn schematically for one position. Each small arrow is a state, with $e_1$ pointing up. Clean, on the left, each block takes in the state of the slice below and gives the state of its own slice. Under the full edit, on the right, the edit takes $e_1$ out at every slice (dashed boxes), so each block reads a state with none; the $e_1$ a state arrives with at the next slice is what that block puts back. That is the write, and the edit there takes it out again. The clean bound is the clean α at the same slice. The write is within the bound when a block puts back no more than the clean model has there.*

In this grammar the op is not held at one token. The model infers it from the examples, and the anchor puts that evidence on $e_1$ at the example answers: an answer that `difference` could give comes onto $e_1$ in a context of any op (the example-evidence report found this). In a context of another op, about one example answer in 17 fits° `difference`. Most of these come from the replacement op noise, which drew `difference` for that example; in 13% the true op gives the same answer. Either way the model sees evidence for `difference` there, so turning those states is the removal at work rather than a side effect.

**Fit.** Of an example or query answer…

So the bound at the example answers counts only the answers that do not fit. Counting every answer would set it from the evidence instead: at slice 3 it would be 82°, against 22° without the fitting answers. The fitting answers get rows of their own in the E1 table.

# How far the edit turns the other ops (E1)

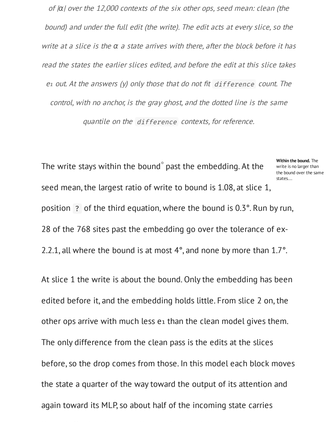
We measure the turn[°] on the contexts of the other ops, since that is where a turn is a side effect. On `difference` contexts the edit is meant to turn the states, and E2 and E3 score what that does.

The anti-subspace term is meant to keep the states of other ops off $e_1$, so we expect the edit to turn them only a little. We expect the turn to be largest at slice 1, where split-anti-seeds found the other ops closest to $e_1$ at the end of training. It should be smaller at slices 2 to 4, where they sat lower. Those are the slices the pull acts on, and they are also deeper; the control sits a little lower with depth too, so this study cannot say which matters.

Does the turn under the edit stay within the clean bound[°], at every slice and role? Or does a later block put $e_1$ back onto the other ops after the edit has taken it out at the slices before?

**Clean bound.** From the clean pass of the anchored model, before any edit.

The figure shows the bound and the write at every position of a context, slice by slice, with the control and the `difference` contexts behind for reference. The two curves part most at the answers (`y`) and a little at `+`, and the write falls further below the bound with each slice.




***The bound and the write on the contexts of the other ops**, by position along a context, one panel per slice (the embedding at the bottom). The 99% quantile of $|e_1|$ over the 12,000 contexts of the six other ops, seed mean: clean (the bound) and under the full edit (the write). The edit acts at every slice, so the write at a slice is the α a state arrives with there, after the block before it has read the states the earlier slices edited, and before the edit at this slice takes $e_1$ out. At the answers (`y`) only those that do not fit `difference` count. The control, with no anchor, is the gray ghost, and the dotted line is the same quantile on the `difference` contexts, for reference.*

**Within the bound.** The write is no larger than the bound over the same states.

The write stays within the bound[°] past the embedding. At the seed mean, the largest ratio of write to bound is 1.08, at slice 1, position `?` of the third equation, where the bound is 0.5°. Run by run, 28 of the 768 sites past the embedding go over the tolerance of ex-2.2.1, all where the bound is at most 4°, and none by more than 1.7°.

At slice 1 the write is about the bound. Only the embedding has been edited before it, and the embedding holds little. From slice 2 on, the other ops arrive with much less $e_1$ than the clean model gives them. The only difference from the clean pass is the edits at the slices before, so the drop comes from those. In this model each block moves the state a quarter of the way toward the output of its attention and again toward its MLP, so about half of the incoming state carries through. On the clean pass that carried part brings its $e_1$ along; under the edit it has none, and what arrives is only what the block adds. At the example answers that do not fit, the write is 39% of the bound at slice 2 and 18% at slice 4.

That is below the control too, which has no anchor. From slice 2 on, the write at those answers is 11° to 4°, against 15° to 10° on the clean pass of the control. So the blocks after an edit do not put $e_1$ back on the other ops; they leave less there than either model has without the edit.

The bound is widest at the answers of the other ops. At the example answers it is widest at slice 1, as we expected: there the answers of the other ops come partway onto $e_1$, about twice as far as in the control. At the query answer it stays wide with depth, from 34° at slice 1 to 42° at slice 4, while at the example answers it narrows.

The wide bound at the query answer comes from a small tail. At slices 2 to 4, a few in a hundred contexts of other ops have their query answer close to $e_1$, while the median stays at a few degrees. How many varies a lot between seeds, from 61 contexts to 2,162. We found nothing in the examples that sets those contexts apart (their posterior on `difference` is about as low as in the rest), so this is unexplained.

The tail contexts are ones the clean model already answers less well: expected exact match[°] 0.47 against 0.54 in the rest, seed mean. Under the edit they end up further from the target null[°], but by about as much as they were already further from the Bayes predictor[°] on the clean pass, so the edit adds about the same distance there as elsewhere (0.012 against 0.010 in total variation[°]). The edit takes these query answers off $e_1$ at the earlier slices, before they get there. At the other roles the bound is a few degrees at every slice. It is smaller than in the control at the colors, and about the same at the syntax.

**Expected exact match.** The rate scores as the chance that a drawn answer agrees with a true one.

**Target null.** Where a removal should land.

**Bayes predictor.** The best answer given the examples.

**Total variation.** A distance between two answer distributions.

| example answers, other ops | emb | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| do not fit, 99th pct. | 5° → 5° | 32° → 31° | 28° → 11° | 22° → 5° | 22° → 4° |
| do not fit, median | 1° → 1° | 8° → 8° | 3° → 2° | 2° → 2° | 2° → 2° |
| fit, 99th pct. | 5° → 5° | 37° → 34° | 71° → 26° | 84° → 13° | 84° → 5° |
| fit, median | 2° → 2° | 28° → 28° | 60° → 11° | 79° → 5° | 82° → 2° |
| control, do not fit, 99th pct. | 17° → 17° | 15° → 6° | 15° → 7° | 12° → 5° | 10° → 4° |
| control, do not fit, median | 4° → 4° | 4° → 2° | 4° → 2° | 4° → 2° | 3° → 2° |

***How far the edit turns the example answers of the other ops**, by slice. Each entry is the angle arcsin $|e_1|$ at the 99th percentile or the median over those answers, clean → as it arrives under the full edit; at the 99th percentile these are the bound and the write. Seed mean over 8 runs, each quantile taken per run and then averaged. An answer fits when `difference` could give it on its operands (2,112 answers per run, against 33,888 that do not). The control has no anchor.*

The table puts the example answers in degrees. On the clean pass, the answers that fit `difference` climb onto $e_1$ with depth, to within a few degrees of it by slice 3. Under the edit they never get there. The edit at slice 1 turns them by about a quarter of a right angle, and at each later slice they arrive far less aligned than the clean model had them, so the writes after slice 1 are small.

Turning these states is the edit doing its job (see the Method): they are the evidence for `difference` in a context that otherwise points elsewhere, and E2 finds that removing them helps the contexts of other ops whose examples point at `difference`.

So on the recipe of record the bound holds in the sense of ex-2.2.1. The anchor gives a turn we can state in advance from the clean map, and the blocks downstream put back less than that.

The bound does not cover what the later blocks make of a state that was turned at slice 1; that is what the spill[°] measures. At slice 1 the other ops are closer to $e_1$ than in a model with no anchor, so that is where any spill would start.

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

# Whether the drop follows what each context needed (E2)

**Target null.** Where a removal should land…

**Drop.** The loss on one context under the edit…

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 inferred op, the edit should take about the same share of the way to the target null in every group of contexts, whatever their posterior on `difference`. The posterior is what the examples say about the op, so an anchor that holds the posterior itself predicts the same; this test treats the two as one reading.

If the anchor holds something coarser, such as the label or a color feature of `difference` answers, the edit would take off about the same amount everywhere. That would be a larger share on the contexts that barely needed the op.

**Share removed.** Of a group of contexts…

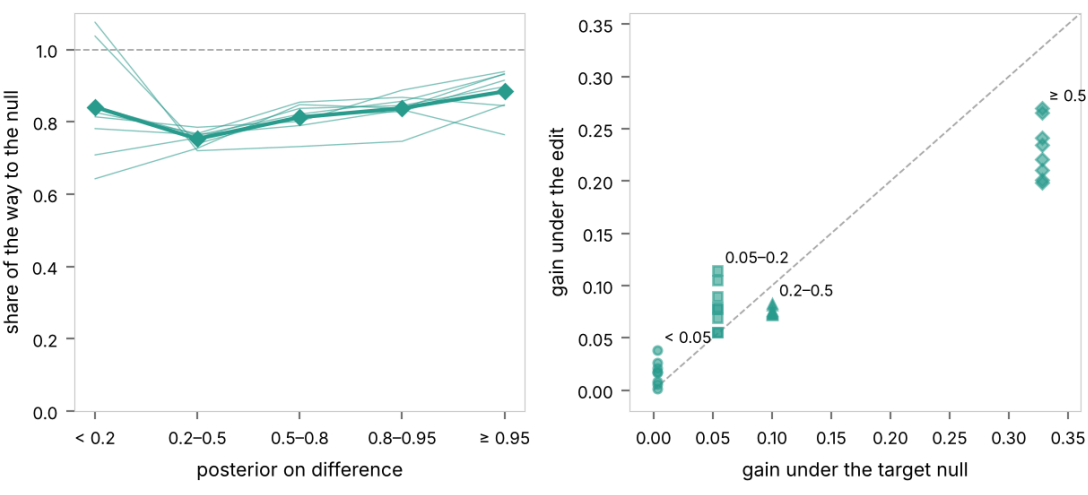
**Dependence.** How much a context needed `difference`…

**Bayes predictor.** The best answer given the examples…

The main measure is the share removed° by bin of the posterior. The correlation of drop with the dependence° of the Bayes predictor° is a second, weaker sign, since the clean model already tracks the posterior.

The target null also predicts a change 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, which is a gain.

The figure shows both: the share removed on `difference` contexts (left) and the gain on the other ops (right).




***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 change in expected exact match under the edit (up is a gain), against the change the target null predicts, one point per run and bin of the posterior on `difference` (labeled; these bins differ from the left). The diagonal is where the edit would match the target null. Both net of the control.*

The share removed is about level across the posterior: between 0.75 and 0.88 at the seed mean, lowest in the 0.2–0.5 bin. The lowest bin is the noisiest, since its contexts needed little to begin with, and two seeds there go a little past the target null. No other bin shows the overshoot a coarse feature would give. The drop correlates with the Bayes dependence at r = 0.76 (0.69 to 0.83).

| posterior on `difference` | < 0.2 | 0.2–0.5 | 0.5–0.8 | 0.8–0.95 | ≥ 0.95 | all |
|---|---|---|---|---|---|---|
| contexts | 172 | 310 | 79 | 744 | 695 | 2,000 |
| dependence (Bayes) | 0.08 | 0.33 | 0.74 | 0.90 | 0.98 | 0.76 |
| share removed | 0.84 (0.64 to 1.08) | 0.75 (0.72 to 0.78) | 0.81 (0.73 to 0.85) | 0.84 (0.75 to 0.89) | 0.88 (0.76 to 0.94) | 0.85 (0.79 to 0.90) |
| not net of the control | 0.89 (0.76 to 0.94) | 0.95 (0.90 to 0.97) | 0.97 (0.90 to 1.00) | 0.97 (0.93 to 0.99) | 0.94 (0.81 to 0.99) | 0.96 (0.87 to 0.99) |

***The share of the dependence removed, by bin of the posterior****, on the held-out `difference` contexts: the mean net drop under the full edit over the mean distance from the clean answer to the target null, in expected exact match; seed mean and range over 8 runs. The dependence row is the drop the target null predicts for the Bayes predictor, which needs no model.*

Without subtracting the control the share is level too, and higher: between 0.89 and 0.97 by bin. So the edit on its own takes `difference` contexts nearly all the way to the target null in expected exact match°. Part of that, 0.07 on average, is what the same edit costs a model with no anchor. The shape of the net share across bins comes from the control, whose own drop under the edit varies by bin, so that shape says where an arbitrary edit costs the most, rather than anything about the anchor.

**Expected exact match.** The task score, as the chance that a drawn answer agrees with a true one….

On the contexts of other ops, the edit helps where their examples point at `difference`, and the more they point at it, the more it helps. That is the direction the target null predicts, and the edit goes most of the way. It falls furthest short where the examples are most in doubt between `difference` and the true op.

Where the posterior on `difference` is ≥ 0.5, the edit gains 0.23 against 0.33 for the target null. In the two low bins it gains a little more than the target null, 0.08 against 0.05 at 0.05–0.2.

Part of the excess in the low bins comes from subtracting the control, which loses a little there under the edit. Where the examples point most at `difference` the control gains a little instead, on every seed, so there the net gain is the smaller one. Without subtracting the control, the gains are 0.05 at 0.05–0.2 and 0.26 at ≥ 0.5. Most contexts of other ops have almost no posterior on `difference` (10,180 of 12,000 below 0.05), and there the target null predicts almost no change.

So the drop is graded the way claim (b) asks: in proportion to what each context needed. The gain on the other ops fits the anchor holding what the examples show about `difference`, wherever it appears, rather than what the label marks.

# Where the edited answers fall short (E3)

On `difference` contexts the full edit covers about half of the distance from the clean answer distributions at the query to the target null°: a landing° of 0.52 (0.49 to 0.56). That is a different measure from the share removed° in E2. E2 scores expected exact match°, the probability on the right answer, which the edit takes most of the way to the null. The landing compares the whole answer distribution, including how the mass that leaves `difference` is shared among the other ops (the target null shares it by how well each op fits the examples).

**Target null.** Where a removal should land…

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

**Share removed.** Of a group of contexts…

**Expected exact match.** The task score, as the chance that a drawn answer agrees with a true one….

Where is the rest? It could be on the answers of `difference` that the edit leaves behind, on colors no op gives, or on the answers of the other ops, weighed differently from the target null. These would point to different causes: a weak removal, damage to the answer, or a model that has lost `difference` and weighs what is left in its own way.

| answer mass on colors that… | only `difference` | another op | no op | off the palette |
|---|---|---|---|---|
| target null | 0.00 | 1.00 | 0.00 | 0.00 |
| clean pass | 0.66 | 0.30 | 0.04 | 0.00 |
| full edit | 0.03 | 0.91 | 0.06 | 0.00 |
| distance left | 0.01 | 0.34 | 0.03 | – |

***Where the answers put their mass on `difference` contexts**, by the kind of color: one only `difference` gives on the query pair, one another op gives (some of these `difference` gives too), one no op gives, or a token that is not a color. A color counts as given when its probability under the op is above zero. The last row splits the distance left between the full edit and the target null (total variation) the same way. Seed mean over 8 runs.*

The edit moves nearly all the mass off the answers only `difference` gives, and onto the answers of the other ops. A little more lands on colors no op gives than on the clean pass. Nearly all of the distance left° is on the answers of the other ops.

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

Overall, the edited model spreads that mass over the other ops about as the target null does. Each op gets roughly its share, with `lighten` and `hue-hsv` a little short.

| mass on the answers of… | `mix` | `lighten` | `darken` | `hue-hsv` | `sat-hsv` | `value-hsv` |
|---|---|---|---|---|---|---|
| target null | 0.26 | 0.22 | 0.21 | 0.21 | 0.26 | 0.24 |
| full edit | 0.25 (0.22 to 0.30) | 0.17 (0.13 to 0.18) | 0.22 (0.19 to 0.25) | 0.18 (0.15 to 0.21) | 0.25 (0.22 to 0.32) | 0.24 (0.21 to 0.28) |

***The answer mass on each other op**, on `difference` contexts: the mass on the colors that op gives on the query pair, averaged over contexts; the full edit as seed mean and range. The colors of two ops can coincide, so a row sums to more than one.*

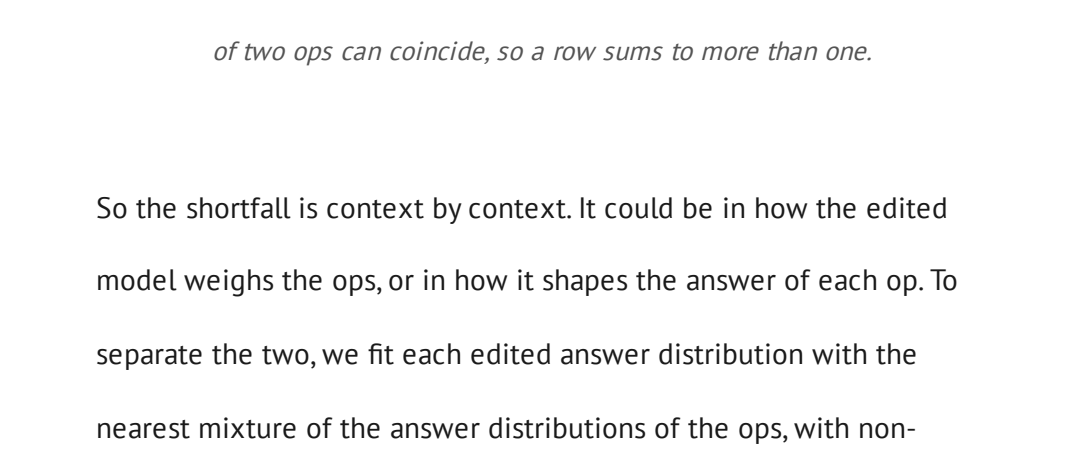
So the shortfall is context by context. It could be in how the edited model weighs the ops, or in how it shapes the answer of each op. To separate the two, we fit each edited answer distribution with the nearest mixture of the answer distributions of the ops, with non-negative weights.

We measure the two legs of the path through that mixture: from the edited answers to the mixture, and from the mixture to the target null. Total variation° is a distance, so the two legs can add to more than the whole.

**Total variation.** A distance between two answer distributions…

For reference, the clean model is some distance from the Bayes predictor° it learned to match. On `difference` contexts that distance is 0.16 (0.15 to 0.19), about as far as the control, and it is a little further on the other ops. This is a comparison point rather than a floor, since the two are distances between different pairs. But it says how close a model of this size gets to an ideal predictor when nothing has been edited. The figure puts that reference beside the distance left and the two legs.

**Bayes predictor.** The best answer given the examples…




***How far the answers are from where they should be**, as total variation on the held-out `difference` contexts. One column per comparison, the 8 seeds as faded dots over a gray bar for their range, and the seed mean on top. The first two columns, left of the dotted rule, are the yardstick: the clean model against the Bayes predictor it learned to match, and the control against the same. The third is the edited model against the target null, and the last two are the legs of the path through the nearest mixture: the edited answers against the nearest mixture of the answers of the ops, and that mixture against the target null.*

The edited answers are close to a mixture of the answers of the ops (0.11 (0.10 to 0.11), against 0.04 on the clean pass). So they are mostly well formed: answers the other ops give, with about the right spread within each op.

The mixture, though, is 0.36 (0.33 to 0.38) from the target null, so the edited model weighs the other ops differently from the posterior, context by context. In all, the distance left is about 2.4× the reference.

The difference in weights is not a systematic one. Weighing the other ops more evenly than the target null, or more sharply, moves the prediction further from the edited answers: with the weights raised to a power of 0.5 or 2 and renormalized, the distance is 0.43 or 0.42, against 0.39 for the target null itself. With the overall proportions about right as well, what is left looks like scatter from context to context.

The edited model does use the examples to weigh the other ops, partly. Take a predictor that has lost `difference` and weighs the other ops equally, whatever the examples show. It is further from the edited answers (0.53) than the target null is (0.39).

Claim (c) asks that the edited model answer as the target null does. This fits the reading of it in the design doc, that the anchor gives the edit a direction and not the whole distance. The edit takes `difference` away cleanly, and the model answers with the other ops. It weighs them by the examples, but less accurately than the target null weighs them.

# Discussion

On the recipe of record, the edit behaves the way the anchor says it should on all three claims, to a first approximation. The turn° on the states of other ops is bounded by the clean map, slice by slice, and the blocks downstream of an edit leave less on $e_1$ than they found. The loss on each `difference` context follows how much that context needed the op, with no sign of an unintended, coarser feature.

**Write.** How far the edit turns a state…

The contexts of other ops whose examples point at `difference` recover as the target null° predicts. That fits the anchor holding the evidence for the op, wherever it appears, and the edit removing that evidence. In this grammar the op seems to be carried that way, spread over the example answers in the residual stream rather than held at one token. Whether part of it also lives in the attention, in what the heads match on (their query and key spaces) or what they carry (their value space), is open; we have put it in the backlog.

**Target null.** Where a removal should land…

The shortfall is in where the answers land. The edit sends the mass to the right ops overall. But within a context, it shares the mass among them only partly by how well each fits the examples. One reading is that the evidence the model uses to tell the other ops apart also travels partly along $e_1$, so taking out the axis blurs it. That would fit the anchor holding the evidence rather than the op itself. Another is that the model never needed fine weights among the other ops on `difference` contexts, since the answer there was `difference`, and so it never learned to weigh them there. This study does not separate them. A model trained without `difference` contexts would show how close any model of this size comes to the target null, and so how much of the shortfall is the edit; it is in the backlog.

If the spill° grows on a later recipe, the bound° map at slice 1 is the first place to look, since that is where the bound is widest.

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

**Clean bound.** From the clean pass of the anchored model, before any edit…

# Glossary

**Write**

**Turn**

How far the edit turns a state: arcsin |α| of the alignment the state arrives with, taken at the 99th percentile over a group of states. At the embedding that is the clean α. At a later slice it is the α left after the block before has read the states the edit turned at the slices below. It is taken before the edit at that slice, which leaves no $e_1$ behind.

**Clean bound**

**Bound**

From the clean pass of the anchored model, before any edit: arcsin of the 99th percentile of |α| over the states of the six other ops at a slice and position. At the answers it counts only those that do not fit `difference`. It follows H4 of ex-2.2.1.

**Within the bound**

The write is no larger than the bound over the same states. Ex-2.2.1 allowed a quarter over (a ratio of 1.25) as a tolerance; we quote the ratio.

**Fit**

Of an example or query answer: one that `difference` can give on its operands.

**Dependence**

How much a context needed `difference`: the expected exact match of the clean model less that of the target null on the same context, so the most the edit could take off without going past the target null.

**Drop**

The loss on one context under the edit: its expected exact match on the clean pass less that under the edit, net of the drop the same edit causes on the control at that context.

**Share removed**

**Net share**

Of a group of contexts: the mean drop over the mean dependence, so the part of the removal beyond what taking out an arbitrary axis costs. It is below the share without the control subtracted, since under the same edit the control loses more on `difference` contexts than on any other op (0.07 in expected exact match).

**Landing**

How far the edit moves the answers toward the target null on `difference` contexts: the share of the clean distance to the target null that it covers, in total variation, as a ratio of the means over those contexts (the number split-anti-seeds quotes).

**Distance left**

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