A model trained without `difference`

> We trained the control° with no loss on `difference` contexts, so that it never learns the op, and compared it to the edited model° of the recipe of record°. On `difference` contexts it comes a little closer to the target null° than the edited model, but still falls well short of it. It misses in the same kind of way: the mass goes to the other ops, but it is shared among them differently from the target null on each context, and differently again at each seed.

Control. A run trained the same way with no anchor....

Full edit. Removing the component of every state along e_1, at every slice and position, and rescaling each state back onto the sphere....

`τ_ℓ fixed`. The anchored recipe that holds the slice temperature τ_ℓ fixed through training; fixed-confirm made it the recipe of record.

Target null. Where a removal should land...

The *edit-effects*° experiment found that the edited model sends the mass to the right ops overall, but shares it among them differently from the target null context by context. Its Discussion offers two explanations: (a) the edit blurs evidence the model uses to share the mass among the other ops, or (b) no model with this architecture, grammar, and set of other ops learns the precise sharing on contexts that look like `difference`.

***edit-effects*.** On the `falloff` recipe the full edit lands about half of the way to the target null; the distance left is about 2.4 times the distance from the clean model to the Bayes predictor, nearly all of it in how the mass is shared among the other ops context by context.

A model trained without `difference` has had nothing removed, so how close it comes is a reference for the edit. It is also the filtered-corpus comparison of the D2.2 design, with figures meant for the D2.2 post. Eight new runs, at seeds 712 to 719, are scored beside the stored `τ_ℓ fixed` runs and controls at the same seeds.

Observations

- Along training (E1): the two conditions approach the target null° at the same pace and settle from about epoch 300. From epoch 128 on, the edited model° stays a small, steady amount further away. `sans-diff` keeps up with the control° on the other six ops throughout, and never learns `difference`.

Target null. Where a removal should land...

Full edit. Removing the component of every state along e_1, at every slice and position, and rescaling each state back onto the sphere....

Control. A run trained the same way with no anchor....

- How close it comes (E2): on `difference` contexts, `sans-diff` comes 0.58 of the way from the control to the target null, and the edited `τ_ℓ fixed` runs come 0.50. The seed ranges don't overlap. Even so, `sans-diff` stays about 1.5 times as far from the target null there as on the contexts it trained on. The `dropped` runs end up with `sans-diff`.
- Whether it misses in the same way (E3): `sans-diff` and the edited model at the same seed are as far apart as two edited runs at different seeds. Both put nearly all of the mass on answers another op could give. They miss in how they share it among those ops, and much of that miss changes from seed to seed.
- What leaving the op out does to the other ops (E4): `sans-diff` scores a little above the control on each of the other ops. The gain is on contexts that look partly like `difference`. On those that look most like it, `sans-diff` gets 94% of the gain the target null predicts, and the edited model gets 75%.

Scope and method

This is an exploratory study, with no preregistration and no gate.

The new condition, `sans-diff`, is the 400-epoch control° with the language-model loss switched off on every `difference` context. The control has no anchor°; it is the one the *anneal-confirm*° experiment trained at these seeds.

Control. A run trained the same way with no anchor....

Anchor. The training term that pulls the states holding `difference` toward a fixed direction of the residual stream, e_1, so that the op can later be removed there.

anneal-confirm. Trained the 400-epoch control, with no anchor, at seeds 712 to 719, beside the annealed recipe, and stored the dose scoring of both.

The training batches still hold `difference` contexts, and the model reads them as input, but no target inside one counts toward the loss, so nothing in training asks the model to answer one. The loss is the mean over the targets that do count, so a step weighs the other six ops as the control does.

Each `sans-diff` run matches the control and the `τ_ℓ fixed` run at its seed: the same initialization, corpus, and schedule, and the same windows in the same order. So the difference between `sans-diff` and the control at a seed comes from the masked loss alone.

On every context `sans-diff` is trained on, the Bayes predictor° is the target null°,[1] so `sans-diff` is scored against the predictor that is ideal for what it was trained on. The contexts of the other six ops still have their replacement op noise, drawn from the six ops other than their own, `difference` among them, so about one example in twenty still shows the answer `difference` would give. The target null keeps that same noise model when it sets `difference` to zero.

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

Target null. Where a removal should land...

1 Cube noise included...

Contexts like the held-out `difference` contexts are rare in the training data of `sans-diff`, since a context of another op shows the answer of `difference` only where its replacement op noise draws `difference`. On those contexts, `sans-diff` answers from what it learned on other contexts, as the edited model° has to.

Full edit. Removing the component of every state along e_1, at every slice and position, and rescaling each state back onto the sphere....

The reference runs are those of the *fixed-confirm*° experiment: `τ_ℓ fixed` (the recipe of record°) at seeds 712 to 719, and the controls at the same seeds. Their stored dose scoring holds the answer distributions on the same held-out contexts, clean and under the full edit, beside the posterior over ops°. `sans-diff` is scored with the same pass, so every distance below is taken on the same contexts in the same way, at the final checkpoint of each run.

fixed-confirm. `τ_ℓ fixed` kept the anchor on all eight fresh seeds, 712 to 719, and stayed within the spill criterion; it became the recipe of record.

`τ_ℓ fixed`. The anchored recipe that holds the slice temperature τ_ℓ fixed through training; fixed-confirm made it the recipe of record.

Posterior over ops. How likely each op is, given the examples of a context....

With eight runs per condition, a difference between conditions is trusted when it is clear of the seed range of both. Seed 718 learns the HSV ops late under `τ_ℓ fixed` and misses the task criterion of *fixed-confirm* on two of them; E4 shows whether the control and `sans-diff` do the same at that seed.

A check arm, `dropped`, trains at seeds 712 and 713 on the corpus with the `difference` contexts taken out entirely (not just masked), for 400 epochs (about six-sevenths as many steps as the control). Under the mask, `sans-diff` still reads `difference` contexts in its window; `dropped` never sees one. If the two end up in about the same place, reading the contexts without answering them makes little difference to what the model learns.

Every measurement here is taken on the answer distribution, which does not change when the residual stream is rotated. So the coordinate axes that Adam favors (the *rotated-basis*° experiment) play no part, except through the edit itself, which acts along e_1.

rotated-basis. Adam puts nearly half of the energy of the residual stream on one coordinate axis per seed; in a randomly rotated basis the anchor still holds on all eight seeds, and the energy moves to the rotated axes.

The measurements

Each measurement is taken on the answer distribution at the query `=`, over the colors of the palette, and averaged over held-out contexts.

Distance from the target null

Total variation (half the summed absolute difference between two distributions) between the answer distribution of the model and the target null, with any mass the model puts off the palette counted as distance. It runs from 0 (a perfect match) to 1 (nothing shared), so lower is closer. On `difference` contexts it is taken under the full edit for `τ_ℓ fixed`, as in edit-effects, and on the clean pass for `sans-diff` and the control. For `sans-diff` on the contexts of the other ops, it is how far the model falls short of the predictor that is ideal for what it was trained on.

Bayes gap

The same distance, between the clean control and the Bayes predictor: how far a model of this size, trained without an anchor, falls short of the ideal on contexts like the ones it trained on. It sets the scale for the other distances, and is taken on `difference` contexts (where edit-effects took its reference) and on the contexts of the other ops. edit-effects called it a comparison point rather than a floor for the edited model, since the two are distances between different pairs. For `sans-diff` the pairs are the same, because its Bayes predictor is the target null on every context, so its own gap on the contexts of the other ops is a floor for its distance on `difference` contexts.

Share of the way

How much of the way from the control (0) to the target null (1) a model has come: one minus the ratio of its distance from the target null to that of the control at the same seed, each a mean over `difference` contexts. edit-effects measured it for edited runs against their own clean pass in place of the control, so for the `τ_ℓ fixed` runs it is reported both ways.

Distance between the models

Total variation between `sans-diff` and the edited `τ_ℓ fixed` run at the same seed, context by context, on `difference` contexts.

Seed-to-seed spread

The same distance, between two `sans-diff` runs at different seeds, over all 28 pairs: how far apart two models trained the same way land. A distance between the models near the spread means they are as alike as a model is to itself at another seed.

Task score

Expected exact match, per op, on the clean pass.

Along training (E1)

Does `sans-diff` learn the other six ops as the control° does, and does it close on the target null° at the same pace as the edited model°? Every run records the task score° of each op on held-out contexts along training, every four epochs, so on the six ops `sans-diff` trains on, it can be compared with the control and with `τ_ℓ fixed` at the same points in training.

Control. A run trained the same way with no anchor....

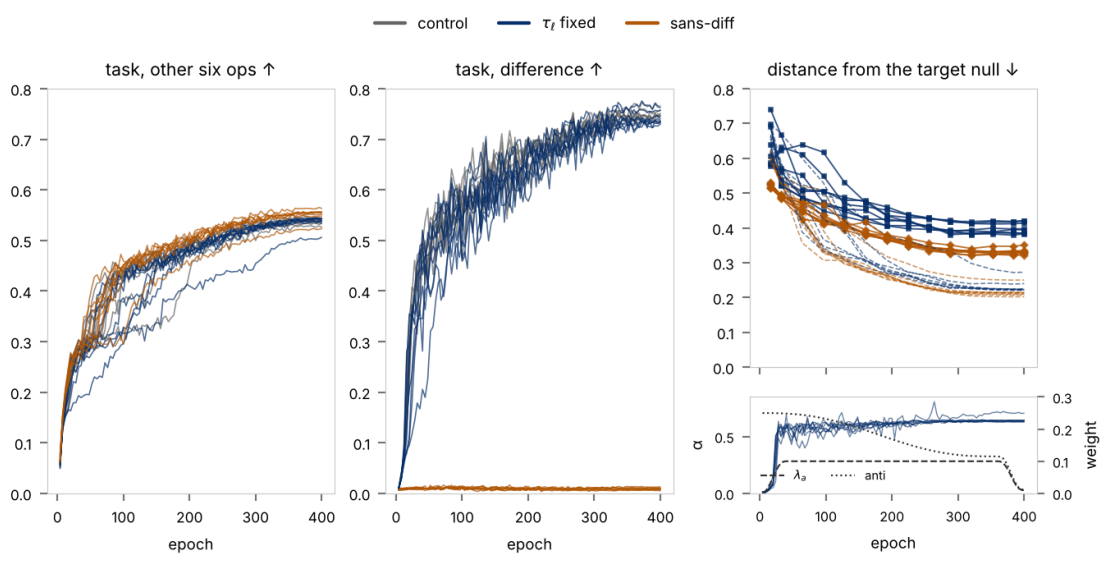
Target null. Where a removal should land...

Full edit. Removing the component of every state along e_1, at every slice and position, and rescaling each state back onto the sphere....

Task score. Expected exact match, per op, on the clean pass.

The `τ_ℓ fixed` runs kept checkpoints along training, and `sans-diff` kept the same ones. Scoring both at each kept epoch gives the distance from the target null° on `difference` contexts along training (clean for `sans-diff`, under the full edit for `τ_ℓ fixed`): whether the two settle early or are still closing when training ends, and whether they close at the same pace. The controls kept no checkpoints, so they have no such curve, though their task score on `difference` along training is there.

Distance from the target null. Total variation (half the summed absolute difference between two distributions) between the answer distribution of the model and the target null, with any mass the model puts off the palette counted as distance....



How the runs learn, along training. *One thin line per run.* ***Left:*** *the task score on the clean pass, the mean over the six ops other than* `difference`*, on a subset of the held-out contexts, recorded every four epochs.* ***Middle:*** *the same on* `difference` *contexts.* ***Right:*** *the distance from the target null on* `difference` *contexts, at the checkpoints kept at epochs 16, 32 and 64 and then every 32 epochs, and at the end:* `sans-diff` *on the clean pass and* `τ_ℓ fixed` *under the full edit, with markers; the dashed lines are the same two on the contexts of the other ops, where the target null is the predictor that would be ideal for a model without* `difference`*.* ***Under it:*** *for the* `τ_ℓ fixed` *runs, the alignment α of the states holding* `difference` *with* e_1 *at the example answers, at the slice where it is highest (one line per run), and the anchor weight λ_a and the anti weight (mean over slices) on the right-hand axis.*

On the six ops it trains on, `sans-diff` keeps pace with the control from the first record to the last, and stays a little ahead of it (left). `τ_ℓ fixed` falls a little behind the control in the first hundred epochs, while the anchor° settles, and catches up by the end, except at seed 718, which learns the HSV ops late (E4).

Anchor. The training term that pulls the states holding `difference` toward a fixed direction of the residual stream, e_1, so that the op can later be removed there.

On `difference` contexts `sans-diff` scores near zero all the way through, as the target null does (middle). So reading the `difference` contexts as input, with no loss on them, does not teach the model to answer them.

The two models close on the target null together on `difference` contexts (right). The edited model starts further away, the difference between them narrows over the first hundred epochs or so, and from epoch 128 it holds at a small, steady amount while both curves keep falling. Both settle from about epoch 300, so 400 epochs is long enough for this comparison. The extra distance that the anchor and the edit leave is set early, and does not close with more training.

A few edited runs stay far from the target null, or move away from it, until epoch 64 to 100 or so. The anchor itself is in place well before that: the states holding `difference` swing onto e_1 by about epoch 30, as the anchor weight ramps up (the panel under the right one). So the slow start is not the anchor arriving. It may be the edit taking more than `difference` out of a model that is still settling, though this figure can't tell.

The dashed lines are the two models on the contexts of the other ops, where the target null is the Bayes predictor° of `sans-diff`. They fall faster and further (E2), and the edited model follows `sans-diff` there closely.

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

How close it comes (E2)

How far is `sans-diff` from the target null° on `difference` contexts, beside the edited `τ_ℓ fixed` runs and the Bayes gap°? If it comes about as close as the edited model°, then landing short of the target null is near what a model of this kind does on these contexts, and the shortfall need not come from the anchor° and the edit. If it comes much closer, the anchor and the edit leave a distance that a model can close. The main measure is the share of the way° from the control° to the target null, with the distance itself beside it.

Target null. Where a removal should land...

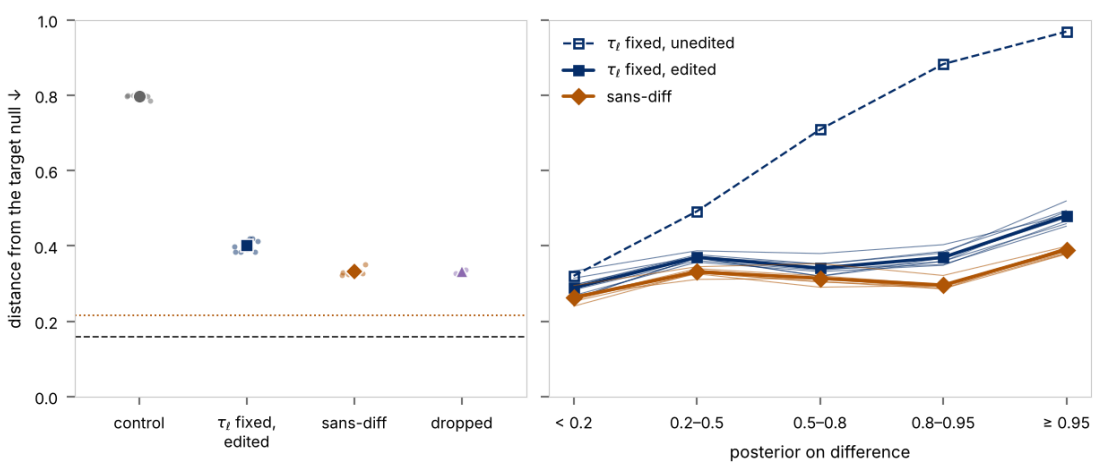
Bayes gap. The same distance, between the clean control and the Bayes predictor...

Full edit. Removing the component of every state along e_1, at every slice and position, and rescaling each state back onto the sphere....

Anchor. The training term that pulls the states holding `difference` toward a fixed direction of the residual stream, e_1, so that the op can later be removed there.

Share of the way. How much of the way from the control (0) to the target null (1) a model has come...

Control. A run trained the same way with no anchor....



How close each model comes to the target null on `difference` contexts. ***Left:*** *the distance from the target null, one column per condition: the control on the clean pass, `τ_ℓ fixed` under the full edit, and `sans-diff` and `dropped` on the clean pass. The seeds are small dots over a thin bar for their range, and the seed mean is the large mark. The dashed line is the Bayes gap on `difference` contexts (the control from the Bayes predictor), and the dotted line is `sans-diff` on the contexts of the other ops, where the target null is its own ideal.* ***Right:*** *the distance by bin of the posterior on `difference`, one hairline per run and the seed mean bold, with the `τ_ℓ fixed` runs on the clean pass (unedited, hollow marks) for comparison.*

	distance on `difference`	share of the way	against own clean pass	on the other ops
control	0.80 (0.79 to 0.80)	0		
`τ_ℓ fixed`, edited	0.40 (0.38 to 0.42)	0.50 (0.47 to 0.52)	0.50 (0.47 to 0.52)	
`sans-diff`	0.33 (0.32 to 0.35)	0.58 (0.55 to 0.60)		0.22 (0.20 to 0.25)
`dropped`	0.33 (0.33 to 0.34)	0.59 (0.58 to 0.59)		
Bayes gap (control)	0.16 (0.14 to 0.18)			0.22 (0.20 to 0.23)

How close each model comes to the target null. *Seed mean and range over 8 runs (2 for `dropped`). The distance on the other ops is from the target null for `sans-diff` (its own ideal) and from the Bayes predictor for the control.*

`sans-diff` comes closer than the edited model at every seed, with the seed ranges of the two well apart: about three-fifths of the way, where the edited model comes half of the way. So the anchor and the edit leave some distance that a model without the op can close, about a sixth of what the edited model has left.

Measured against the clean pass° of the `τ_ℓ fixed` run instead of the control, the share for the edited model is the same, since the clean run is as far from the target null as the control is.

Clean pass. The forward pass with no edit.

Most of the shortfall is there without the anchor and the edit, though. On the contexts of the other ops, `sans-diff` falls short of the target null (its Bayes predictor° there) by about as much as the control falls short of the Bayes predictor; on `difference` contexts it falls about half as short again. So a model with this architecture, grammar, and set of other ops that has never been asked to answer a `difference` context still falls well short of the target null on one, and the edited model a little further.

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

The two `dropped` runs end up with `sans-diff` at their seeds, inside its seed range. A model that reads `difference` contexts without answering them ends up about where one that never sees them does.

Split by the posterior on `difference` (right), both models are furthest away on the contexts where the posterior is nearly certain of `difference`. Those are the contexts least like anything `sans-diff` trained on. The extra distance of the edited model is also largest there, and smallest on the contexts whose examples point mostly at other ops. The middle bin holds fewer than a hundred contexts, so its value is loose. The unedited `τ_ℓ fixed` runs, which still answer `difference`, are far from the target null in every bin but the lowest, as expected.

Whether it misses in the same way (E3)

Do the edited model° and `sans-diff` miss the target null° in the same way? If the distance between the two models is small beside the distance of each from the target null, and near the seed-to-seed spread°, they share the mass among the other ops alike on a context. The part of the shortfall they have in common then comes from these contexts rather than from the anchor° and the edit.

Full edit. Removing the component of every state along e_1, at every slice and position, and rescaling each state back onto the sphere....

Target null. Where a removal should land...

Seed-to-seed spread. The same distance, between two `sans-diff` runs at different seeds, over all 28 pairs...

Anchor. The training term that pulls the states holding `difference` toward a fixed direction of the residual stream, e_1, so that the op can later be removed there.

If the two models are as far from each other as from the target null, they miss in different ways on a context, though broadly speaking they may still miss in the same kind of way (the breakdown below). The main measure is the distance between the models°, context by context.

Distance between the models. Total variation between `sans-diff` and the edited `τ_ℓ fixed` run at the same seed, context by context, on `difference` contexts.

How far apart the models land on `difference` contexts. *Total variation between two models, context by context, one column per pair: `sans-diff` and the edited `τ_ℓ fixed` run at the same seed, and at different seeds (all 56 pairs); two `sans-diff` runs at different seeds (all 28 pairs); two edited `τ_ℓ fixed` runs at different seeds; and `sans-diff` and `dropped` at the same seed. The pairs are small dots over a thin bar for their range, and the mean is the large mark. The dotted lines are the distance of `sans-diff` and of the edited model from the target null.*

Two `sans-diff` runs at different seeds land about as far from each other as each lands from the target null, and so do two edited runs. `sans-diff` and the edited model at the same seed are as far apart as two edited runs, and a little further apart than two `sans-diff` runs, clear of their range. Sharing the seed and the initialization does not make the two miss alike on a context: at different seeds they are just as far apart.

That gives another way to say how close the edited model comes to `sans-diff`. No model lands nearer a `sans-diff` run than another `sans-diff` run does, so take a second `sans-diff` run as the end of the way and the control° as its start. On that scale the edited model comes 85% of the way to `sans-diff` on `difference` contexts (seeds 79% to 89%). On the contexts of the other ops the control, the edited model, and `sans-diff` are all about as far apart as two `sans-diff` runs, so there is no way to measure there.

Control. A run trained the same way with no anchor....

Two edited runs are also about a quarter further apart from each other than two `sans-diff` runs are, with the ranges of the two kinds of pair apart, so the anchor and the edit add some scatter from seed to seed as well as some distance from the target null°. On a given `difference` context, each run shares the mass among the other ops in a way of its own, and much of its distance from the target null is that scatter from run to run.

Distance from the target null. Total variation (half the summed absolute difference between two distributions) between the answer distribution of the model and the target null, with any mass the model puts off the palette counted as distance....

Averaging the answers of each condition over its eight seeds, context by context, shows how much.

on `difference` contexts	run by run	answers averaged over seeds
`sans-diff` from the target null	0.33 (0.32 to 0.35)	0.23
`τ_ℓ fixed`, edited, from the target null	0.40 (0.38 to 0.42)	0.28
`sans-diff` from `τ_ℓ fixed`, edited	0.42 (0.40 to 0.45)	0.23
`sans-diff`, seeds 712–715 from 716–719		0.19
`τ_ℓ fixed`, edited, seeds 712–715 from 716–719		0.24

Distances taken run by run and on the answers averaged over the 8 seeds. *The first column is the seed mean and range of the distance taken on each run (or each pair at a seed); the second takes the distance once, on the answer distributions of each condition averaged over its seeds context by context; the last two rows average each half of the seeds of one condition separately.*

The averaged answers of both conditions are about a third closer to the target null than a single run. The two averages are still about as far from each other as the `sans-diff` average is from the target null, but two averages over four seeds each, from the same condition, land nearly as far apart. So most of what separates the two averages is scatter still left after averaging eight seeds, and the data show little of a miss common to the runs of one condition but different between the two conditions.

Averaging does the same where the ideal is not in doubt. On the contexts of the other ops, `sans-diff` comes closer to its Bayes predictor° as more seeds are averaged, and the control to its own, by about the same share as `sans-diff` comes closer to the target null on `difference` contexts.

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

seeds averaged	1	2	4	8
`sans-diff` from the target null, on `difference`	0.33	0.28	0.25	0.23
`sans-diff` from the target null (its Bayes predictor), on the other ops	0.22	0.18	0.16	0.15
control from the Bayes predictor, on the other ops	0.22	0.18	0.16	0.15

Distance from the ideal, with the answers averaged over more seeds. *The seeds are taken in order, in groups of the given size (712–713, 714–715, and so on for two), the answers of each group averaged context by context, and the distance taken on the average; mean over the groups.*

So scatter from run to run is a general property of models of this size, and averaging removes about the same share of it everywhere. What sets `difference` contexts apart is how far a single run is to begin with. The averages also close on the target null the way they close on the Bayes predictor elsewhere, which is what we would expect if the target null is the right reference for these contexts.

on `difference` contexts	`sans-diff`	`τ_ℓ fixed`, edited	target null
mass on answers only `difference` gives	0.00 (0.00 to 0.00)	0.03 (0.01 to 0.06)	0.00
mass on answers another op could give	0.94 (0.94 to 0.95)	0.91 (0.87 to 0.92)	1.00
mass on colors no op gives	0.06 (0.05 to 0.06)	0.07 (0.06 to 0.07)	0.00
distance from the nearest mixture of the ops	0.10 (0.10 to 0.10)	0.11 (0.10 to 0.12)	
that mixture from the target null	0.30 (0.30 to 0.32)	0.38 (0.36 to 0.40)	
from the target null, posterior to the power 0.5	0.41 (0.40 to 0.41)	0.44 (0.42 to 0.46)	
from the target null, posterior to the power 1	0.33 (0.32 to 0.35)	0.40 (0.38 to 0.42)	
from the target null, posterior to the power 2	0.34 (0.33 to 0.36)	0.43 (0.41 to 0.47)	

Where the mass goes on `difference` contexts, and what kind of miss it is, *as edit-effects (E3 there) breaks it down. Seed mean and range over 8 runs. The nearest mixture is the mixture of the answer distributions of the seven ops closest to the answer of the model on each context. The last rows compare the model with a target null whose posterior on the other ops is raised to a power and renormalized: below 1 spreads the mass more evenly over the other ops, above 1 gives more to the op the examples favor.*

Both models put nearly all of the mass on answers another op could give, and very little on answers only `difference` gives (the edited model a little more). Each is close to some mixture of the answers of the ops, and that mixture is nearly as far from the target null as the model is. So both miss in how they share the mass among the other ops, as *edit-effects*° found for the earlier recipe.

edit-effects. On the `falloff` recipe the full edit lands about half of the way to the target null; the distance left is about 2.4 times the distance from the clean model to the Bayes predictor, nearly all of it in how the mass is shared among the other ops context by context.

Neither power fits either model better than the target null as it is (power 1), so neither model leans toward sharing the mass evenly or toward the op the examples favor.

Where the answers go. *Op confusion as in ex-2.2.22, over the held-out contexts of every op, seed mean. Rows: the op of the context. Columns: the mass on the colors the column op can give for the query operands and the row op cannot. The diagonal (the mass on the answers of the row op) is outlined, printed in grey italics, and left out of the color scale. The `difference` column is hatched, since `sans-diff` never learned that op and the edit removes it. Values of at least 0.03 are printed.*

Pooled over contexts, the `difference` row of `sans-diff` and of the edited model looks much like that of the target null. The mass is spread over the six other ops in about the same proportions, except that the edited model gives `lighten` about a third less than the target null does. The other rows look alike in all four matrices, except that the edited model puts a little more on the answers `sat-hsv` and `value-hsv` give for each other; nearly all of that is seed 718, the run that learns those ops late (E4). As *ex-2.2.22*° found for the edited model, pooling over the contexts of a row averages away most of the context-by-context miss above, as averaging over seeds does.

ex-2.2.22. Confusion matrices of the clean and edited models against the target null, by how sure the context is about its op; pooled over contexts, they make the edited model look closer to the target null than the distance taken context by context.

The two models differ in two ways: `sans-diff` never learned the op, and it trained without the anchor. On the contexts of the other ops, the clean `τ_ℓ fixed` run is as far from the control at the same seed as two controls at different seeds are from each other. So the anchor moves the answers there by about as much as a change of seed does. On `difference` contexts too, the two models differ by about as much as two runs of one condition.

What leaving the op out does to the other ops (E4)

On contexts of the other ops, does `sans-diff` score as well as the control°, and does it gain where the target null° says a model without `difference` should? The target null predicts a gain on contexts of other ops whose examples point partly at `difference`, since an ideal model that cannot answer `difference` puts that mass on the right op. *edit-effects*° found that the edited model° gains most of that; this is the same comparison for a model that never learned the op, with the edited model beside it.

Control. A run trained the same way with no anchor....

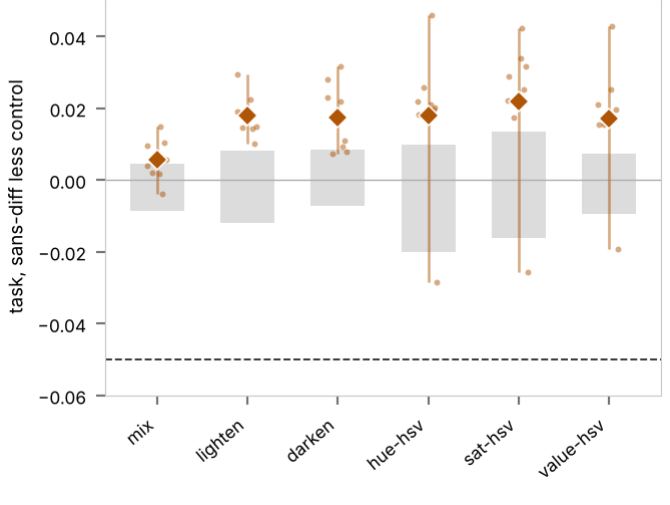
Target null. Where a removal should land...

edit-effects. On the `falloff` recipe the full edit lands about half of the way to the target null; the distance left is about 2.4 times the distance from the clean model to the Bayes predictor, nearly all of it in how the mass is shared among the other ops context by context.

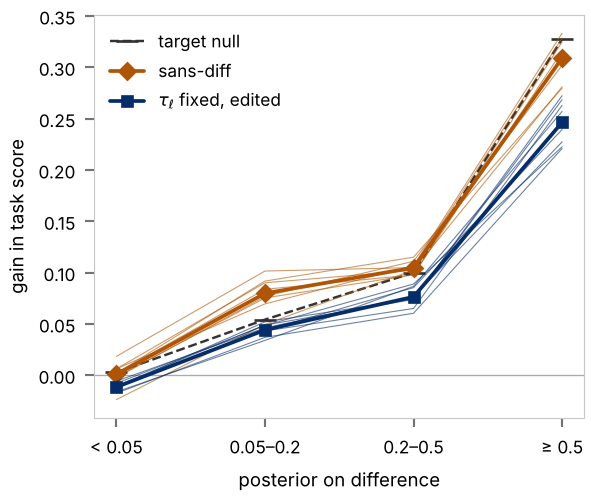
Full edit. Removing the component of every state along e_1, at every slice and position, and rescaling each state back onto the sphere....

It also checks the training: if `sans-diff` scored well below the control on the other ops, part of its distance in E2 could come from a weaker model rather than from the missing op. The main measure is the task score°.

Task score. Expected exact match, per op, on the clean pass.



Per op. *The task score of `sans-diff` less that of the control at the same seed. The seeds are small dots over a thin bar for their range, and the seed mean is the large mark. The grey band is the seed range of the control about its own seed mean. The dashed line is the per-op task gate of the D2.2 design, which the anchored model is held to against the control, drawn here for scale.*



By posterior on `difference`. *On the contexts of the other ops, the gain in task score: `sans-diff` over the control at the same seed, the edited `τ_ℓ fixed` run over its own clean pass, and the target null over the Bayes predictor (dashed). One hairline per run and the seed mean bold.*

What leaving `difference` out does to the other ops.

`sans-diff` scores a little above the control on each of the six other ops (per op). One seed, 717, scores a little below the control on the three HSV ops and above it on the rest. The two `dropped` runs score like `sans-diff` at their seeds.

At seed 718, where `τ_ℓ fixed` learns the HSV ops late, the control and `sans-diff` both learn them on time, so that weakness belongs to the anchored run. `sans-diff` is not a weaker model, then, and its distance in E2 comes from the missing op.

posterior on `difference`	**< 0.05**	**0.05–0.2**	**0.2–0.5**	**≥ 0.5**
contexts	10,180	1,042	664	114
target null	0.003	0.054	0.100	0.328
`sans-diff`	0.001 (−0.024 to 0.018)	0.080 (0.050 to 0.102)	0.105 (0.098 to 0.115)	0.309 (0.280 to 0.333)
`τ_ℓ fixed`, edited	−0.011 (−0.017 to −0.006)	0.044 (0.034 to 0.052)	0.076 (0.060 to 0.089)	0.246 (0.221 to 0.272)

The gain on the contexts of the other ops, by bin of the posterior on `difference`. *Task score of `sans-diff` over the control at the same seed, of the edited model over its clean pass, and of the target null over the Bayes predictor; seed mean and range over 8 runs.*

The gain is where the target null predicts it (by posterior). Most of the held-out contexts have a posterior on `difference` below 0.05, and on those `sans-diff` and the control score about the same; above that, the gain grows with the posterior. `sans-diff` gains about as much as the target null predicts (a little more in the second bin), and the edited model gains less in every bin above the first.

The edited model also loses a little on the contexts that look least like `difference`, a small cost of the edit on contexts it would ideally leave alone. The top bin holds about a hundred contexts, so its values are loose.

Discussion

A model that never learned `difference` lands short of the target null° on `difference` contexts, a little closer than the edited model°, and misses in the same kind of way. Of the two explanations *edit-effects*° offered, this favors the second: most of the shortfall is what a model of this kind does on contexts that look like an op it was never asked to answer.

Target null. Where a removal should land…

Full edit. Removing the component of every state along e_1, at every slice and position, and rescaling each state back onto the sphere.…

edit-effects. On the `falloff` recipe the full edit lands about half of the way to the target null; the distance left is about 2.4 times the distance from the clean model to the Bayes predictor, nearly all of it in how the mass is shared among the other ops context by context.

The anchor° and the edit add a smaller part on top, set early in training (E1) and largest on the contexts where the posterior is surest of `difference` (E2). They also add some scatter from seed to seed, and the model they leave is as far from `sans-diff` as two edited runs at different seeds are from each other (E3).

Anchor. The training term that pulls the states holding `difference` toward a fixed direction of the residual stream, e_1, so that the op can later be removed there.

The miss looks like an underdetermined choice. On a context whose examples point at `difference`, the other six ops are all poor fits, and the training data say little about how to share the mass among them, since such contexts are rare in it. Each run settles that sharing in its own way. Runs of this size scatter on ordinary contexts too, and averaging over seeds brings them toward their ideal by about the same share there (E3), so the scatter is general; on `difference` contexts a single run starts further off.

On the contexts of the other ops, a model trained without `difference` gains almost all that the target null predicts, and the edited model gains most of it. At a given seed, the answers of the edited model of the recipe of record° are as far from those of `sans-diff` as two edited runs are from each other, on `difference` contexts and on the others; on the others they are also about as far apart as two `sans-diff` runs. With a second `sans-diff` run as the end of the way, the edited model comes 85% of the way from the control° to `sans-diff` on `difference` contexts (E3).

`τ_ℓ fixed`. The anchored recipe that holds the slice temperature τ_ℓ fixed through training; fixed-confirm made it the recipe of record.

Control. A run trained the same way with no anchor.…

This holds for one op, at one model size, with 400 epochs of training.

References

anneal-confirm

Trained the 400-epoch control, with no anchor, at seeds 712 to 719, beside the annealed recipe, and stored the dose scoring of both.

edit-effects

On the `falloff` recipe the full edit lands about half of the way to the target null; the distance left is about 2.4 times the distance from the clean model to the Bayes predictor, nearly all of it in how the mass is shared among the other ops context by context.

ex-2.2.22

Confusion matrices of the clean and edited models against the target null, by how sure the context is about its op; pooled over contexts, they make the edited model look closer to the target null than the distance taken context by context.

fixed-confirm

`τ_ℓ fixed` kept the anchor on all eight fresh seeds, 712 to 719, and stayed within the spill criterion; it became the recipe of record.

rotated-basis

Adam puts nearly half of the energy of the residual stream on one coordinate axis per seed; in a randomly rotated basis the anchor still holds on all eight seeds, and the energy moves to the rotated axes.

Glossary

Anchor

The training term that pulls the states holding `difference` toward a fixed direction of the residual stream, e_1, so that the op can later be removed there.

Control

A run trained the same way with no anchor. The controls here are those of anneal-confirm, at the same seeds.

`τ_ℓ fixed`

Recipe of record

The anchored recipe that holds the slice temperature `τ_ℓ` fixed through training; fixed-confirm made it the recipe of record.

Full edit

The edit

Edited model

Removing the component of every state along e_1, at every slice and position, and rescaling each state back onto the sphere. It is meant to take `difference` out and leave the other ops as they were; the edited model is the anchored model under it.

Clean pass

The forward pass with no edit.

Posterior over ops

Posterior on `difference`

How likely each op is, given the examples of a context. The posterior on `difference` is the probability it gives that op, so it says how much a context looks like `difference`.

---

1. Cube noise included: the posterior of record models it as a term of its own. ↩