Ex 2.2.1: suppressing *red* in the anchored transformer

The first intervention on an anchored transformer. We take the D2.1 checkpoints and remove the anchor axis from the residual stream, then score completion accuracy on lines with a red operand and on lines without. The removal works, and it scales with how red the line is. It stays inside the bound the placed geometry sets. Zeroing the axis weights does the same job. Selectivity is only partial: the loss on non-red lines comes from the syntax positions, and it goes away when we edit the operand positions only.

Findings

H1 (suppression bites) — holds. Seed-mean accuracy on red lines under the primary intervention is 0.09, against a gate of 0.5; the highest single seed is 0.19.

H2 (selective) — partial. The seed-mean drop in accuracy on non-red lines is 0.024. That is outside the 0.02 gate but inside the 0.05 partial gate. One seed accounts for most of it (0.127); the median seed is at 0.009. The `operands` arm, which skips the syntax positions, drops only 0.000. So the side-effect is due to the constant component in the syntax positions, which is the case the contrary clause of H2 named.

H3 (grades with redness) — holds. Seed-mean damage across the five redness bins[1] is 0.03, 0.02, 0.03, 0.21, 0.90. The largest dip between adjacent bins is 0.018 (allowed 0.02), and the top bin exceeds the bottom by 0.87 (gate 0.5).

H4 (the write bound) — holds. At the four deeper slices the seed-mean 99th-percentile write on non-red lines is at most 1.21 times the clean bound (gate 1.25); the largest is at slice 1, position `⏎`. At every site before the answer it is at most 1.02 times the bound. Read seed by seed rather than on the mean, two seeds exceed the tolerance at a site before the answer.

H5 (permanent removal) — holds. Under `ablate`, accuracy on red lines is 0.12 (gate 0.5) and the drop on non-red lines is 0.012 (gate 0.02).

The recomputed clean alignment maps match the ones ex-2.1.10 published to within 0.0011 on every run.

> **How to read this report**
>
> This report was preregistered: the method, hypotheses, and decision thresholds were frozen before the experiment ran (commit `1a42bc0`). Results replaced the `TODO` placeholders in place; everything conceived after seeing the data is under Exploratory analyses, marked as post hoc.
>
> While the draft was written we ran a one-seed smoke test of the operator library (seed 0 of each condition), to check the code path from checkpoint to score. The gates come from the clean statistics instead: the published alignment maps, the task-gate width the D2.1 experiments used, and the reference values from M1. None of the smoke-test numbers are quoted here.
>
> One smoke-test observation was stated in advance. At one syntax position, the alignment arriving at the second slice under intervention sat above the clean envelope on that seed. We kept the deeper-slice clause of H4 as the design states it; the nine-seed read is in the H4 section.
>
> One design change was made after the smoke test and before the run. The post-hoc tier's directions were first fitted on the op1 states and applied at every position, and on the control seed the oblique fit edited the syntax states heavily, an extrapolation rather than an erasure. The tier now fits and applies at the same sites, as the Method describes. No gate changed.

Why this experiment

D2.1 anchored *red* on the first axis of the residual stream, and the geometry landed where we asked: red states aligned, the non-red bulk contained, grading roughly sensible, task accuracy unchanged. It never ran an intervention. The point of placing a concept is that removing it afterwards has a predictable effect and a side-effect we can bound, and we have not tested that claim in a transformer.

The test uses checkpoints already in the store. Projecting e_1 out of the stream at inference time costs nothing to train and about a second per run to score, which makes it the most information per dollar in the D2.2 plan.

It is also where the first three risks in that plan get their first data: (a) that suppression does not bite even on *red*, (b) that the bound is loose or wrong once later blocks sit between the write and the output, and (c) that the response to suppression is undesigned. Only the first is retired here. The plan shares the second with the layer sweep and the third with the fallback experiment.

Three findings from the M1 autoencoder result become hypotheses here. Suppression there removed *red* almost completely, left orthogonal colors untouched, and degraded warm colors in proportion to their alignment with the axis. Those become H1, H2, and H3.

The bound does not transfer as it stands. In the autoencoder, the decoder was a single linear readout from the point of intervention, so the geometry bounded the output directly. Here every later block sits between the write and the logits.

So we state the bound as layer-local, following the kickoff lessons: the geometry bounds the immediate write at each intervened slice (H4). Behavior is then a separate prediction, scored by the task gate (H2).

These checkpoints have no fallback term, so whatever a red line decodes to after suppression is the untrained response of the model itself. M1 called that spoofing, and it is where the seed spread in ex-2.9.1 came from. We measure it here without a gate; it is the reference the fallback experiment after this one has to improve on.

▶ **Glossary…**

Method

No training. Every run is a published checkpoint of ex-2.1.10, scored on that experiment's published probe set through the eval contract in `sca.intervention`: the same lines, the same code, and the same statistics for every condition, arm, and baseline.

The intervention

Axis projection at full strength: at each slice, every state loses its e_1 component and is put back onto the sphere, since the next block expects a unit vector. That re-projection is part of the edit.

For a state with alignment α, the surviving components are rescaled by $1/\sqrt{1-\alpha^2}$, and the whole edit amounts to a rotation by $\arcsin|\alpha|$. That formula depends only on the alignment the state arrived with, so we can compute the bound from the published alignment maps before the intervention runs.

The primary intervention acts at all five slices and all six positions. The operator at slice ℓ sees a state the blocks built from already-edited inputs, so only at the embedding does it see the clean stream. At every later slice, the arriving alignment measures whether the blocks put the axis back.

Measurements

One teacher-forced pass per run per intervention, over all 5,832 lines. From each pass:

- **The response** per line: the log-softmax over the vocabulary at the `:` position. From it we read the probability on the correct answer, the argmax, and the mass outside the color vocabulary.
- **The write** per (slice, line, position): the angle between the arriving state and the edited state. The pipeline asserts that this equals $\arcsin|\alpha_{\text{arriving}}|$ at every site, so where a map reads more easily we report the write as that alignment.
- **The clean alignment map** per run, from the un-intervened pass. This is the same array ex-2.1.10 published, recomputed here so every quantity comes from one code path.
- **The final-state displacement** per line: the angle between the intervened and clean states at the last slice, `:` position. That is the state the logits are read from. We report it in radians, the same scale as a write.
- **The decoded write** per (slice, position): a ridge probe for the RGB of the operand, fitted per run on the clean states at that site, then read on the arriving state and on the edited state. The difference is what the write removed, in color units. The angle says how far a state moved; this says whether what moved was redness (E6).

The undesigned response, on red lines under the primary intervention, without a gate. We record the mass outside the color vocabulary; how far the decoded answer sits from the true mix, in unit-cube units; which answer it is (the visible operand, the red operand itself, the true answer, a one-step neighbor of it, or something else); and seed agreement, the fraction of red lines on which at least five of the nine seeds decode the same answer. These are the reference numbers for the fallback experiment.

Noise floor. Clean accuracy on every group is at or near 1.0 in every run, so the seed scatter that matters is in the intervened statistics. For each group statistic we compute the pooled between-seed standard deviation per condition before reading any comparison. A difference between arms or tiers smaller than twice that value is reported as not resolved. Gates score seed means against fixed thresholds and do not use the floor.

Conditions

Two conditions from ex-2.1.10, both trained on the same corpus with the same seeds and code path:

- **anchored** (`either-t100`, 9 seeds) — the D2.1 recipe: pooled either-operand labels, the anchor at $\lambda = 0.1$, the anti-subspace term at the ex-2.1.8 operating point. This is the primary condition, and the one every hypothesis scores.
- **un-anchored** (`lam0`, 5 seeds) — the anchor weight at zero. Nothing was placed on e_1, so projecting it out applies the operator to a model with nothing on the axis. That calibrates what a write of this kind costs when it removes no concept.

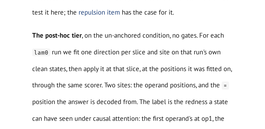
Arms, on the anchored condition. Each changes one thing from the primary intervention, and we report the same statistics for each without gates:

arm	what changes	question
`operands`	positions restricted to op1 and op2	does the edit need the syntax positions, whose embeddings have a constant component on the axis?
`embedding`	slice 0 only	is removing the token's component enough, or do the blocks re-derive red?
`last`	slice 4 only	the one-readout case, closest to M1's geometry
`shaped`	M1's shaped suppression, threshold $\alpha = 0.5$, in place of the plain projection	does leaving the low-alignment bulk alone keep the removal and cut the collateral?
`ablate`	the weights that read and write e_1 zeroed, then re-normalized	the permanent removal from M1, scored under H5

The shaped suppression is M1's alternative to the plain projection. Where the projection removes the whole e_1 component of every state, the shaped suppression removes only a fraction $\hat{h}(\alpha)$ of it, set by the alignment the state arrived with. Below a threshold a it removes nothing; above it, the fraction rises linearly to all of it at $\alpha = 1$ (in the notation of the M1 appendix, $\hat{h} = 1$ and $p = 1$).

Alignment here is a cosine: 1 for a state pointing along e_1, 0 for one at right angles to it. A threshold of $a = 0.5$ sits 60° from the axis. States with a negative alignment are left alone too.

Removing part of the component shortens the state, taking it off the sphere, which is where the M1 suppression left it. Here the stream is unit-norm, so both operators re-normalize afterward, and removal plus re-normalization amounts to a rotation away from e_1 in the plane spanned by e_1 and the state. The figure draws that rotation, and where each direction in the plane lands.



The two operators, in the plane of e_1 and the state. Left and middle, each chord runs from a unit state at some angle from e_1 (top) to the dot where the operator moves it; the filled lobe is the size of that rotation against the direction, scaled so 90° reaches the rim. Think, the same rotation against the alignment a. Fourth, the alignment the state leaves with against the one it arrived with; the dotted line is no change. The threshold $a = 0.5$ is dashed, with the alignments it was set between marked from the published clean map: the `:` and `=` embeddings (band) and pure red at op1 (line).

At $a = 0.5$ the shaped suppression leaves the syntax embeddings where they are. It rotates a pure-red operand at the embedding by about 42°, where the projection rotates it by 64°. So the red operand sits partway up the ramp rather than at the top.

A pure-red operand arrives at $\alpha = 0.9$ and leaves at 0.38, past the threshold and about where the syntax embeddings sit: the falloff sets how much of the component is removed, and the re-normalization sets where the state lands. Above the threshold the order reverses, so the more aligned a state arrives, the less aligned it leaves (fourth panel). The repulsion form from M1 sets the landing alignment instead, and would hold the red operand at the threshold. We did not test it here; the repulsion item has the case for it.

The post-hoc tier, on the un-anchored condition, no gates. For each `lam0` run we fit one direction per slice and site on that run's own clean states, then apply it at that slice, at the positions it was fitted on, through the same scorer. Two sites: the operand positions, and the `=` position the answer is decoded from. The label is the redness a state can have seen under causal attention: the first operand's at op1, the higher of the two after that, which at `=` is the line's redness. Three fits beside e_1 itself: diff-in-means[1] between the states labelled at or above the red threshold and the rest, a ridge probe[1] for that redness, and LEACE[4] for it. The fit and the edit share a distribution because an oblique eraser needs them to: LEACE reads its coefficient along a covariance-whitened direction, so a state far from the fitted cloud can draw an edit of any size. Beside each fitted direction sits e_1 at the same footprint, so the calibration is like for like. The operand site pairs with the `operands` arm; the decode site is the sharpest lever a post-hoc method can be given, an edit where the answer is read off. This calibrates what a searched-for direction removes from a model that was never asked to place one.

The strength sweep, on the anchored condition: the primary intervention at $\gamma \in \{0.25, 0.5, 0.75, 1\}$, where γ is the fraction of the e_1 component that is removed. Exploratory (E1).

Hypotheses

All on the anchored condition under the primary intervention unless stated; seed means over nine runs.

- **H1.** Suppression bites. Seed-mean exact-match accuracy on red lines falls to 0.5 or below, from a clean accuracy of 1.0. Partial: between 0.5 and 0.8. Contrary: accuracy stays above 0.8, which would mean the model reads the color of the operand from somewhere the axis does not reach.

- **H2.** Suppression is selective. Seed-mean accuracy on non-red lines stays within 0.02 of the clean accuracy. Partial: within 0.05. The gate is the width every D2.1 task gate used.

  Projecting out an axis costs something even where nothing was placed on it. So we report the un-anchored row under the same operator beside this one, to say how much of any deficit comes from the operator rather than from the removal. - **H3.** Damage grades with redness. We bin every line by its redness at edges 0.2, 0.4, 0.6, and 0.8. Seed-mean damage is non-decreasing across the five bins, allowing a dip between adjacent bins of at most 0.02, and damage in the top bin exceeds the bottom bin by at least 0.5. Partial: one dip larger than 0.02, with the ends still ordered. We prescribe no shape between the ends. The D2.1 alignment response was sigmoid in redness rather than linear, so the grading may be a step rather than the smooth $\cos^2$ curve M1 saw. - **H4.** The write on non-red lines stays within the bound the clean geometry sets, at every slice. At a given (slice, position) site, the bound is $\arcsin$ of the 99th-percentile clean alignment over the 1,689 non-red lines there. We take that per run from the un-intervened map, then average over the nine seeds. The measurement is the 99th-percentile write over the same lines under the primary intervention, again per run and then seed-averaged (each quantile is the 17th-largest of 1,689 values within a run; averaging is all the seeds do). At the embedding the two are equal by construction, and the pipeline asserts it. At each of the four deeper slices, and at every position, the measured write is at most 1.25 times the bound. Partial: one site over by no more than a factor of two. Contrary: a block re-writes the axis on non-red lines after it was removed upstream. That would be a bypass, and it would bring the question of the layer sweep forward. - **H5.** Permanent removal is selective, given the anti-subspace term. Under the `ablate` arm, the H1 and H2 gates both hold: accuracy on red lines at or below 0.5, and non-red accuracy within 0.02 of clean. Partial: one of the two holds. M1 found ablation selective only when repulsion had cleared the axis, and the D2.1 recipe has that term, so this is the transformer form of M1's headline result.

  Ablation and the projection at every slice give nearly the same forward pass: the reads see no e_1 either way, so the two differ only in the gain applied to the output of a block before it joins the stream. The seed scatter under `ablate` should therefore match the scatter under the primary intervention. On red lines both scatter because these checkpoints have no fallback term, so the response there is untrained. That is why every gate reads a seed mean and every table shows the seed range.

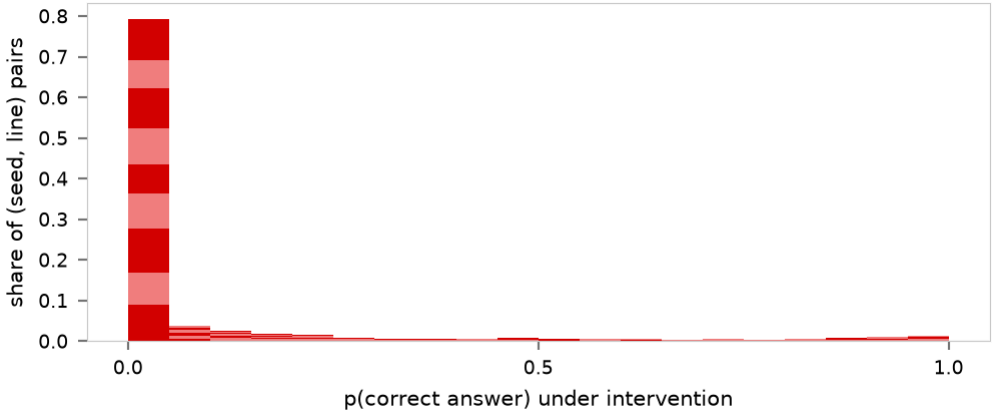
condition, intervention	clean red acc	red acc ↓	red damage ↑	non-red acc ↑	non-red deficit ↓	non-red damage ↓	off-vocab, red
primary, `primary`	0.999	**0.09 ±0.10**	0.90 ±0.09	0.975 ±0.103	0.024	0.034 ±0.124	0.013
operands, `operands`	0.999	**0.13 ±0.14**	0.87 ±0.12	0.998 ±0.002	**0.000**	0.001 ±0.004	0.007
embedding, `embedding`	0.999	**0.16 ±0.12**	0.85 ±0.11	0.811 ±0.317	0.187	0.253 ±0.276	0.017
last, `last`	0.999	0.92 ±0.00	0.08 ±0.01	0.999 ±0.002	**-0.000**	0.000 ±0.000	0.002
shaped, `shaped`	0.999	0.60 ±0.34	0.42 ±0.32	0.998 ±0.003	**0.000**	-0.000 ±0.000	0.002
ablate, `ablate`	0.999	**0.12 ±0.13**	0.87 ±0.11	0.987 ±0.032	**0.012**	0.020 ±0.045	0.012
control, `primary`	0.999	1.00 ±0.00	0.01 ±0.00	0.999 ±0.001	**0.000**	0.005 ±0.003	0.000

Removal and selectivity by arm. Each value is the seed mean, with the seed range in parentheses. Accuracy is exact match on the answer token; damage is the clean answer probability minus the intervened one, in probability units; the deficit is clean accuracy minus intervened accuracy on non-red lines. Off-vocab is the mass outside the color vocabulary on red lines. Bold marks a value inside its gate: red accuracy at or below 0.5 (H1, and H5 for `ablate`), non-red deficit at or below 0.02 (H2, H5). The arms and the un-anchored row have no gate and are shown for the comparisons the sections below read.

Suppression bites (H1)

Under the primary intervention the seed-mean accuracy on red lines falls to 0.09, from a clean accuracy of 0.999. The gate is 0.5, and the seed that keeps the most red lines keeps 0.19 of them. The un-anchored condition under the same operator loses nothing (0.999): the operator removes the concept where one was placed, and finds nothing to remove where none was.

Read line by line, the removal is close to complete on nearly every red line rather than an average over lines lost and lines kept. The median intervened probability on the correct answer is 0.001 across seeds and lines, and 83% of (seed, line) pairs fall below 0.1.



The response on red lines. *The intervened probability on the correct answer over the 365 red lines, as a share of all (seed, line) pairs, with the nine seeds stacked in alternating shades. A bimodal response would show as mass at both ends.*

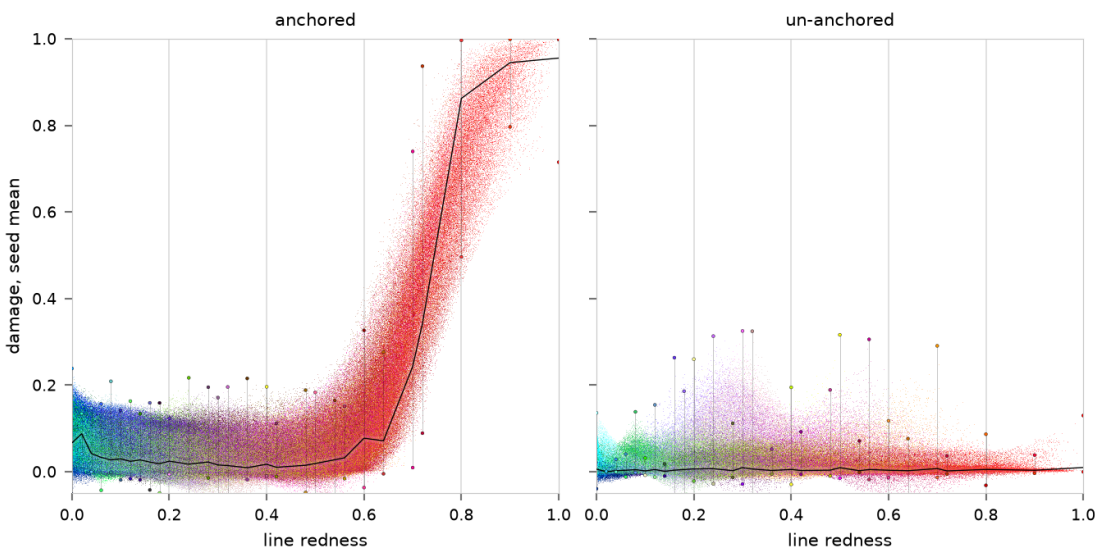
H1 holds. The contrary case was that the model would read the color of the operand from what is left of the embedding off the axis. It does not: with the axis gone, the model no longer knows the operand was red.

Selectivity (H2)

The seed-mean drop in accuracy on non-red lines is 0.024: outside the 0.02 gate, inside the 0.05 partial gate. The spread across seeds is wide. Seed 1 loses 0.127 of its non-red accuracy and the median seed loses 0.009, with 7 of the nine seeds inside the gate on their own. The un-anchored condition under the same operator loses 0.000, so the operator itself is close to free. The loss comes from what the anchored blocks do with an edited syntax state.

The contrary clause of H2 predicted where the damage would come from, and the arms agree. The `operands` arm, which leaves the four syntax positions alone, loses 0.000. The `shaped` arm, whose threshold sits above the constant component of the syntax embeddings, loses 0.000. The `operands` arm still removes red (red accuracy 0.13); the `shaped` arm removes only part of it (0.60), trading some of the removal for the extra selectivity (E4).

So editing the syntax positions is what costs us the non-red lines. Those embeddings have a constant component on the axis, and the blocks read that component as part of the syntax rather than as redness. In damage terms the primary intervention and the `operands` arm differ by 0.033, against a resolution floor of 0.120, so at the seed level they are not separated either way. The accuracy read above is the one the gate scores.



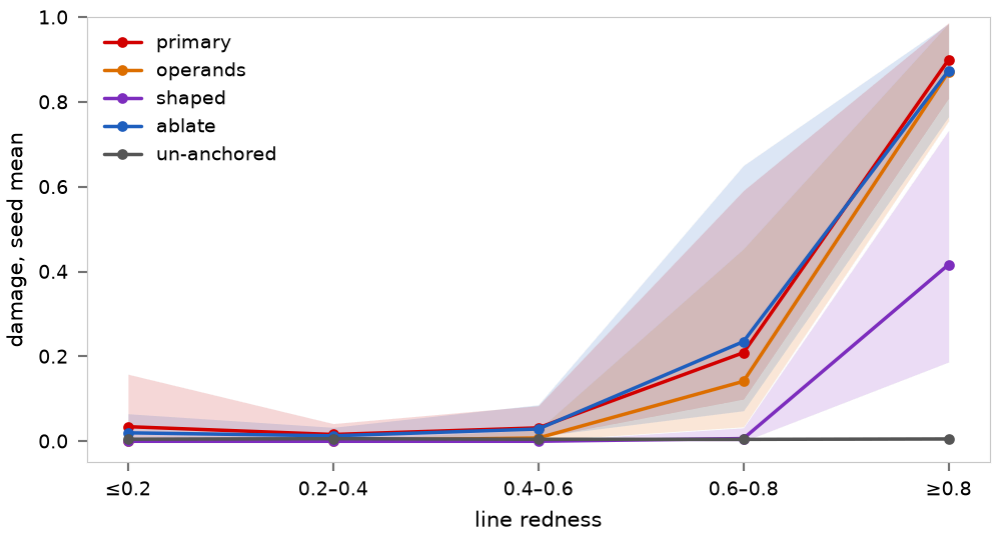
Damage against line redness. *Seed-mean damage under the primary intervention, per line, on the anchored condition (left) and the un-anchored one (right). The cloud is the distribution of damage over the lines at each redness level, dithered and interpolated between levels; the line is the mean, and each thin vertical rule spans the least to the most damaged line at that level, with a dot in the color of that line's redder operand. The vertical rules are the H3 bin edges.*

H2 is partial. The removal is selective at the operand positions, and the loss on non-red lines is a cost of editing the whole stream, concentrated in a minority of seeds. What that means for the bound is read in the H4 section.

Grading (H3)

Seed-mean damage across the five bins under the primary intervention is 0.034, 0.016, 0.032, 0.209, 0.900. There is one dip, 0.018 between the first two bins, inside the 0.02 allowance. It is the share of the H2 deficit that falls in the lowest bin rather than anything about the grading. The top bin exceeds the bottom by 0.87, against a gate of 0.5.

The shape is a step rather than a ramp: the three lowest bins sit on the floor, the 0.6–0.8 bin takes 0.21, and the red bin 0.90. That follows the S-shaped alignment response D2.1 measured, where the alignment of a color rises steeply only near the red corner. The `shaped` arm sharpens the step further, to 0.006 in the 0.6–0.8 bin and 0.42 in the red bin, since its threshold leaves every state below an alignment of 0.5 alone. The un-anchored condition is flat at 0.005.



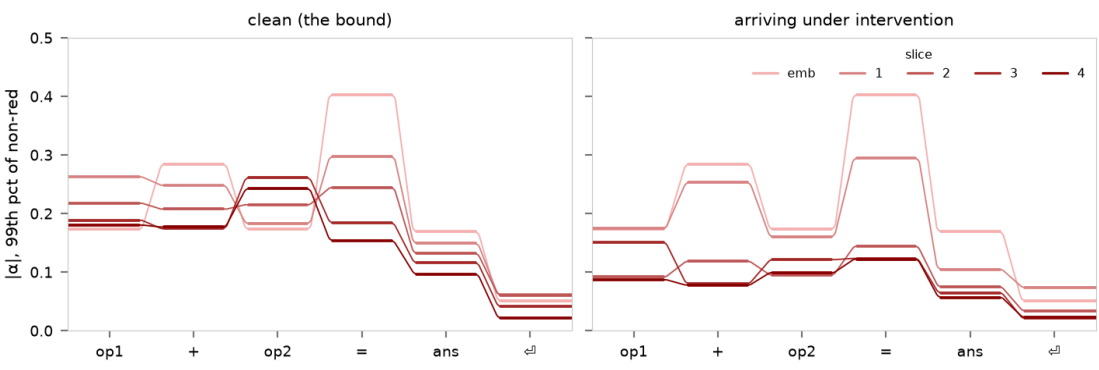
Damage per redness bin. *Seed-mean damage in each of the five redness bins, with the band spanning the seed range, for the primary intervention, three arms, and the un-anchored condition under the primary operator. H3 asks the primary line to be non-decreasing within a dip of 0.02 and to span at least 0.5 from the first bin to the last.*

H3 holds. The contrary case was a middle bin damaged more than the bin above it, and no such inversion appears. The per-color view is in E3.

The write bound (H4)

At the four deeper slices, the seed-mean write on non-red lines is at most 1.21 times the bound. The sites over one are (slice 1, `+`), (slice 1, `⏎`), (slice 4, `⏎`). The largest are at the newline, where the bound is smallest (0.022 to 0.062 radians) and where nothing downstream is read for the answer. At every site before the answer the ratio is at most 1.02, so the blocks do not put the axis back on non-red lines.

Per seed, the largest deeper-site ratio runs from 1.22 to 2.09, at position `=` or `⏎`. At the sites before the answer it runs from 0.91 to 1.59, and 2 of the nine seeds are above the tolerance on their own. The gate reads the seed mean, as the hypothesis states it; the per-seed spread is the number the layer sweep starts from.

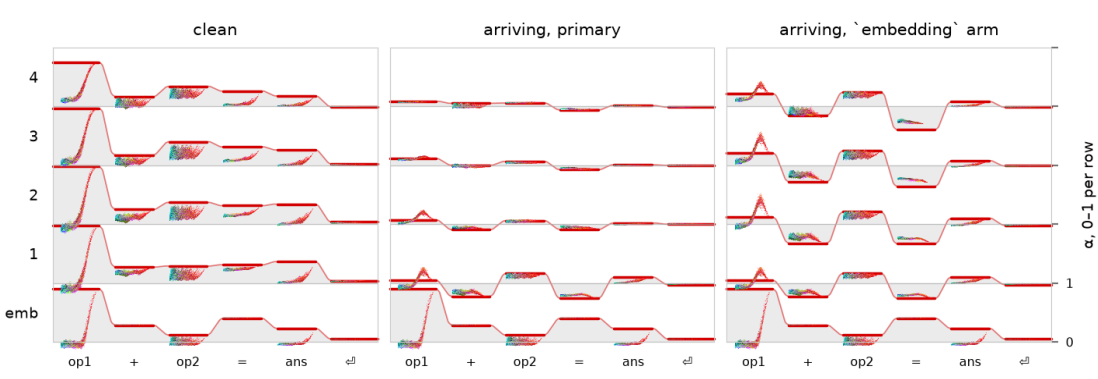


The bound and the write, per site. *The 99th-percentile alignment over the non-red lines at each (slice, position), seed mean, on the shared 0–1 scale. Left, the clean map, whose arcsine is the bound; right, the alignment arriving at the operator under the primary intervention, whose arcsine is the write. The embedding line is the same in both by construction. Slices run from the embedding (lightest) to the last block (darkest).*

slice	op1	+	op2	=	ans	⏎
emb	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
1	**0.66**	**1.02**	**0.88**	**0.99**	**0.70**	**1.21**
2	**0.42**	**0.57**	**0.44**	**0.59**	**0.57**	**0.56**
3	**0.80**	**0.46**	**0.46**	**0.67**	**0.56**	**0.51**
4	**0.48**	**0.44**	**0.40**	**0.79**	**0.59**	**1.07**

The write over the bound, per site: the seed-mean 99th-percentile non-red write under the primary intervention divided by the seed-mean bound. The embedding row is the identity check. Bold marks a ratio at or below the H4 tolerance of 1.25.

The per-color view shows the same thing color by color. Under the primary intervention every row after the embedding is flat: the red colors arrive at the deeper operators with an alignment of 0.11 or less at op1, down from 0.75 or more in the clean pass, and the non-red bulk is where it was. Under the `embedding` arm, where the blocks run un-edited after the token edit, pure red climbs back to 0.22 at op1 by the last slice and is written below zero at `=`. So when nothing stops them, the blocks re-derive part of the axis from an off-axis input; the later slices of the primary intervention are what keep it off.



The arriving alignment per color, at every site. *Each panel is one row per slice (embedding at the bottom) and one grading-cloud slot per position, each cloud the alignment of the states whose op1 color is the cloud's color, against that color's redness; the red line traces pure red across the positions. Left, the clean map ex-2.1.10 published; middle, the alignment arriving at each operator under the primary intervention; right, the same under the `embedding` arm, where only the token embedding was edited and the blocks run un-edited after it.*

H4 holds. The bound is intact at every site before the answer. A bypass would show as the axis coming back on non-red lines at a deeper slice, as it does under the `embedding` arm; under the primary intervention no slice puts it back, so the blocks do not re-derive the concept.

The H2 deficit is a different thing. A rotation of the syntax states can stay inside the bound and still be an input the blocks never met in training, and what they make of that input can reach the answer (E5).

Permanent removal (H5)

Under the `ablate` arm, accuracy on red lines is 0.12 (0.00–0.25), against 0.09 (0.00–0.19) under the projection. The drop on non-red lines is 0.012, against 0.024. Both H5 gates hold: red accuracy at or below 0.5, non-red drop at or below 0.02. Ablation and projection behave alike on red lines, as the hypothesis said they should.

On non-red lines ablation costs less than the projection at every slice, 0.020 in damage against 0.034. The gap of 0.015 is inside the resolution floor of 0.120, so the two are not separated at the seed level. The seed that loses most under the projection loses 0.044 under ablation. Two things may explain the smaller cost. A permanent removal has no re-normalization gain to apply per slice. And once the token embeddings have lost their constant component on the axis, the syntax positions have none either, which is where H2 placed the side-effect.

H5 holds. This is the transformer form of the headline result from M1: with the anti-subspace term in the recipe, zeroing the weights that read and write the axis removes the concept and little else.

The undesigned response

No gate. With the axis gone, a red line still decodes to a color: the mass outside the color vocabulary is 0.013 (seed range 0.000–0.046). On average the decoded answer sits 0.29 unit-cube units from the true mix, where one grid step is 0.2.

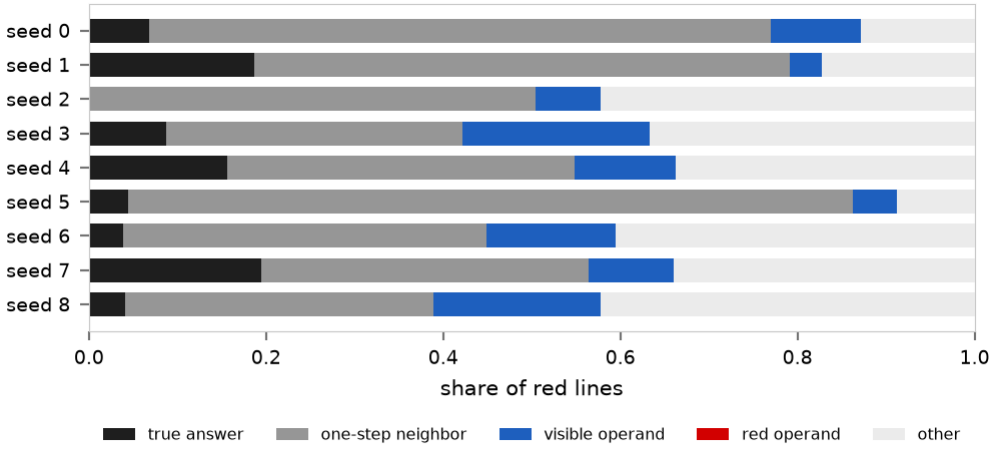
We expected the answer to sit nearer the visible operand than the true mix. It does not. Across the nine seeds, 50% of red lines decode to a one-step neighbor of the true answer, 9% to the true answer itself, 11% to the visible operand, and 30% to some other color. The red operand is never returned.

A neighbor is still a miss. Accuracy is exact match, and on those lines the probability on the true answer falls to near zero (H1), so in probability terms the damage on red lines is close to complete. In color terms it is small.

The misses are not random colors. Of the 2,953 (seed, line) pairs that decode to a wrong grid color, 100.0% are less red than the true answer,[5] and 98.5% are what the line would mix to if the red operand were replaced by some grid color less red than it. That holds whether the red operand is pure red or one of the four other colors of the red group.

So the model keeps mixing, and reads the red operand as a paler or duller color than it is. That response is better behaved than the spoofing we saw in M1, and it is the reference the fallback experiment has to improve on.

Which neighbor a line lands on varies from seed to seed. Under the primary intervention, at least 5 of the nine seeds decode the same answer on 13% of red lines; the figure is 19% under the `operands` arm and 18% under `ablate`. The `shaped` arm, which removes only part of the red state, agrees on 82%, because most of its lines still decode to the true answer.

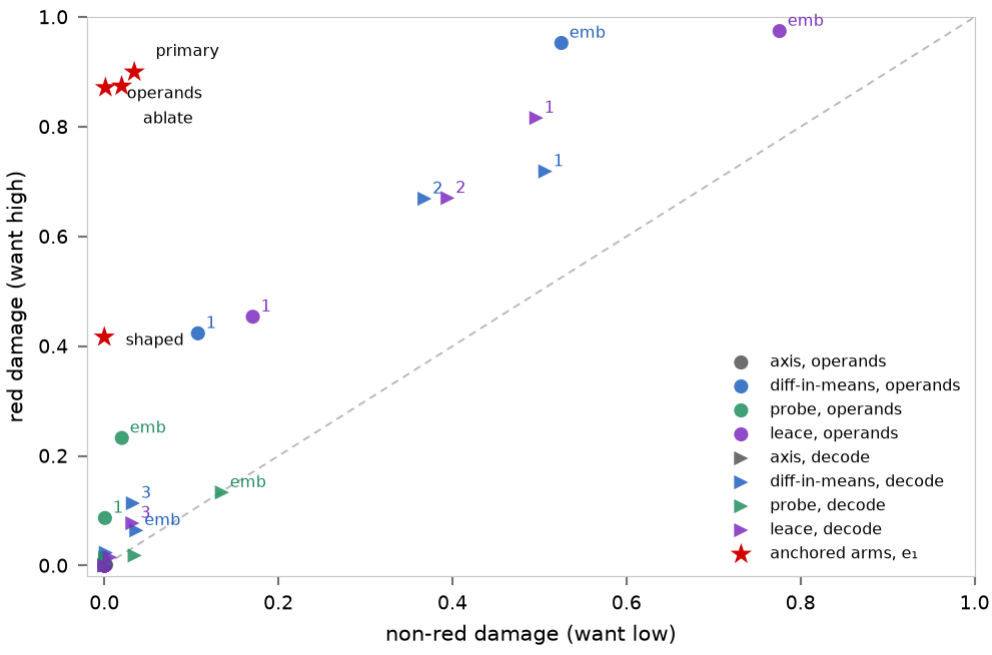


What red lines decode to, per seed. *Each bar splits the 365 red lines under the primary intervention by the decoded answer. Greys are the true mix (dark) and its one-step neighbors on the grid (light); blue is the visible operand, red the red operand itself, and the pale segment every other color.*

The post-hoc tier

No gate. On the un-anchored condition, e_1 itself does nothing at either site (red damage at most 0.001). That is the calibration: the operator only removes what sits on the direction it is given.

Each fitted direction removes something, and the pattern is the same at both sites. At the operand site only the embedding and the first block matter. A direction fitted at a deeper slice removes nothing, so the answer is formed from the operand states before the second block reads them. At the decode site the first two blocks matter. At the embedding there the state is the same on every line, so there is nothing to fit: LEACE returns the identity, and the other two fits return a direction with no label in it, whose small effect is an edit of the shared `=` state.



Red damage bought per unit of non-red damage. *One mark per (fit, slice, site) on the un-anchored condition, seed mean: circles at the operand site, squares at the decode site, labelled by slice, colored by fit. The stars are the anchored arms under the placed axis. A selective removal sits high and to the left. The dashed line is equal damage on both groups; a mark below it would remove more from non-red lines than from red ones, and none does.*

What separates a placed axis from a found direction is the trade-off between the two kinds of damage. The strongest post-hoc removal at the operand site, leace at slice emb, buys 0.97 of red damage for 0.78 of non-red damage. At the decode site, leace at slice 1 buys 0.82 for 0.50. The anchored primary intervention buys 0.90 for 0.034, and the `operands` arm 0.87 for 0.001. Only one post-hoc direction has a small non-red cost, the probe at the embedding, and it is a mild edit in both respects: 0.23 of red damage for 0.020.

So no fitted direction reaches the combination the placed axis gets. Every post-hoc row that removes at least half the red response costs 0.37 or more in non-red response, an order of magnitude above the cost of the anchored primary intervention, and the rows that cost little non-red response remove little red.

fit, at operands	emb	1	2	3	4
axis	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00
diff-in-means	0.95 / 0.53	0.42 / 0.11	0.00 / 0.00	-0.00 / 0.00	0.00 / 0.00
probe	0.23 / 0.02	0.09 / 0.00	0.00 / 0.00	-0.00 / 0.00	0.00 / 0.00
leace	0.97 / 0.78	0.45 / 0.17	-0.00 / 0.00	0.00 / 0.00	0.00 / 0.00

fit, at decode	emb	1	2	3	4
axis	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00
diff-in-means	0.06 / 0.04	0.72 / 0.51	0.67 / 0.37	0.11 / 0.03	0.02 / 0.00
probe	0.13 / 0.14	0.02 / 0.03	0.01 / 0.00	0.00 / 0.00	0.00 / 0.00
leace	0.00 / -0.00	0.82 / 0.50	0.67 / 0.39	0.08 / 0.03	0.02 / 0.01

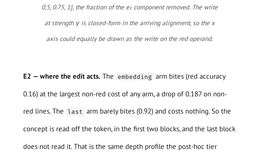
▶ **Reading the tables…**

The un-anchored model does not keep redness on one linear direction that the mixing computation leaves alone. What the probe reads at the embedding is mostly separable from the task. The mean-difference and LEACE directions are not: the states that differ between red and non-red lines differ along directions the blocks also use for the mix. The per-slice pattern says where the two part company, and it is the first block.

Exploratory analyses

Preregistered as exploratory, no gates.

E1 — strength. The response is graded in γ rather than all-or-nothing: seed-mean red accuracy is 0.89, 0.61, 0.26, 0.09 at γ = 0.25, 0.5, 0.75, 1, with the steepest drop between 0.5 and 0.75, and the seed spread widest in the middle of the sweep. Non-red damage is 0.005, 0.022, 0.057, 0.034 over the same points, and the seed with the H2 deficit shows it from γ = 0.5 on.

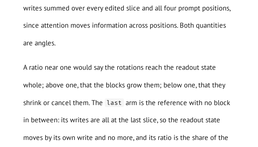


The strength sweep (E1). *Seed mean and seed range of red accuracy and non-red damage under the primary intervention at γ ∈ {0.25, 0.5, 0.75, 1}, the fraction of the e_1 component removed. The write at strength γ is closed-form in the arriving alignment, so the x axis could equally be drawn as the write on the red operand.*

E2 — where the edit acts. The `embedding` arm bites (red accuracy 0.16) at the largest non-red cost of any arm, a drop of 0.187 on non-red lines. The `last` arm barely bites (0.92) and costs nothing. So the concept is read off the token, in the first two blocks, and the last block does not read it. That is the same depth profile the post-hoc tier found on the un-anchored model.

When only the token is edited, the blocks partly re-derive the axis (the H4 clouds). At the prompt positions, the non-red 99th-percentile alignment arriving at the last slice under that arm is 0.39, against 0.24 clean. That partial return, together with the large non-red cost, is one transformer data point on the bypass question the D1.3 post left open: removing the axis at the input alone is neither a clean removal nor a cheap one, and editing at every slice is what makes the primary intervention selective.

E3 — damage against alignment. Read line by line, damage against the clean alignment of the redder operand rises more steeply than the $\cos^2$ curve M1 saw. At the embedding, lines whose operand sits near zero alignment are untouched, the middle of the scale takes partial damage with a wide spread, and the reddest colors, at 0.7 and above, lose nearly everything. At the deeper slices the clean pass has already pulled every red color to an alignment near one, so the rise sits at the top of the scale and the middle colors spread along it. The per-color view gives the same picture: the colors that lose their lines are the five reddest, and the middle bins of H3 are made of lines whose operand sits on the steep part of the D2.1 alignment response.



Damage against the clean alignment of the redder operand (E3). One mark per line, seed-mean damage under the primary intervention against the seed-mean clean alignment at the position of the line's redder operand, one panel per slice. Marks are colored by that operand.

E4 — the shaped suppression. With a threshold of 0.5, the `shaped` arm leaves every syntax state and every non-red state alone: its non-red damage has magnitude 0.00003, and at every site its 99th-percentile write on non-red lines falls to at most 0.002 radians. That is the H4 map with the syntax columns emptied.

It removes about half of the red response (accuracy 0.60 (0.26–0.84)), with the widest seed spread of any arm. The reason is that the alignment of the red operand at the embedding, 0.90 for pure red, sits partway up the ramp of the falloff rather than at its top (see the operator figure in the Method).

The edit at the embedding sends pure red past the threshold, and the blocks then bring the red operand partway back. At the three middle slices it arrives with a mean alignment of 0.33 under the shaped suppression, against 0.52 clean, and 46% of those states sit above the threshold and are written again.

What it decodes to is still a color: on red lines the mass outside the color vocabulary is 0.002, 60% of the decodes are the true answer, and 29% are a one-step neighbor of it. The shaped suppression is the selective operator. A lower threshold or a steeper ramp would remove more of red, and the repulsion from M1 would bound where the red operand lands. Both are open (shaped-suppression item, repulsion item).

E5 — how much of the write reaches the answer. The logits are read from one state, the `=` state at the last slice. Its displacement is how far it moved, intervened against clean. We compare that with the writes summed over every edited slice and all four prompt positions, since attention moves information across positions. Both quantities are angles.

A ratio near one would say the rotations reach the readout state whole; above one, that the blocks grow them; below one, that they shrink or cancel them. The `last` arm is the reference with no block in between: its writes are all at the last slice, so the readout state moves by its own write and no more, and its ratio is the share of the four positions that this write accounts for.

arm	writes at	non-red lines			red lines		
		summed write	displacement	ratio	summed write	displacement	ratio
`embedding`	slice 0	0.84	0.64	0.78	1.74	1.08	0.62
primary	every slice	1.96	0.28	0.14	3.40	0.98	0.29
operands	every slice, operands only	0.63	0.08	0.12	1.96	0.94	0.48
`last`	slice 4	0.28	0.05	0.16	1.56	0.15	0.11

Readout displacement against the summed prompt write (E5). Seed means, in radians. Summed write: over the four prompt positions and every edited slice. Displacement: how far the `=` state at the last slice moved from its clean position. Ratio: the mean across seeds of the per-seed ratio of the two.

On non-red lines the ratio under the primary intervention is 0.14 (seed range 0.12–0.21), about the same as the reference. The `embedding` arm says why. A write at the embedding alone reaches the readout state at nearly its summed size, a ratio of 0.78, so the blocks carry an edit forward rather than shrink it.

Under the primary intervention the edits at the deeper slices remove what the blocks rebuilt, and the readout state ends 0.28 radians from clean. That is less than half the displacement under the `embedding` arm, from twice the summed write. The `operands` arm moves the readout state by 0.08 radians, so most of that displacement comes from editing the syntax positions.

The seed with the H2 deficit has the largest readout displacement of the nine, 0.42 radians against a seed mean of 0.28, and its summed write is no larger than what the other seeds see.

On red lines the readout state moves 0.98 radians under the primary intervention, a ratio of 0.29. That is the designed case: on those lines the edit is meant to change the answer.

E6 — what the write decodes to. At the operand positions on red lines, the write reads as a fall in decoded redness with a rise in green and blue of about the same size, at every slice. The change is largest at the embedding, where the alignment is removed in one step. On non-red lines the same direction of change appears at the deeper slices, at about a third of the size; at the embedding the sign reverses, since what is removed there is a small negative alignment. The three channels stay in a fixed ratio of roughly (−1, +0.85, +0.85) throughout. The probe reads the axis as redness against the other two channels, so this says the rotation is along what the probe calls red and nothing else the probe can see.

slice	at op1, decoding op1						at op2, decoding op2					
	red Δ*R*	red Δ*G*	red Δ*B*	non-red Δ*R*	non-red Δ*G*	non-red Δ*B*	red Δ*R*	red Δ*G*	red Δ*B*	non-red Δ*R*	non-red Δ*G*	non-red Δ*B*
emb	-0.119	+0.104	-0.103	+0.021	-0.019	-0.019	-0.119	+0.104	-0.103	+0.021	-0.019	-0.019
1	-0.008	+0.011	-0.011	+0.028	-0.022	-0.021	-0.096	+0.090	-0.087	-0.024	+0.021	+0.020
2	-0.029	+0.024	-0.023	-0.006	+0.005	+0.004	-0.037	+0.030	-0.028	-0.004	+0.002	+0.001
3	-0.044	+0.032	-0.031	-0.057	+0.036	+0.025	-0.031	+0.024	+0.057	-0.053	+0.009	+0.007
4	-0.045	+0.031	+0.030	-0.042	+0.032	+0.028	-0.042	+0.032	+0.025	-0.037	+0.028	+0.021

The decoded write (E6): the seed-mean change in the ridge-decoded RGB of the operand at its own position, edited state minus arriving state, under the primary intervention, on red lines and on non-red lines. A probe read on an edited state is an extrapolation, so the numbers are a direction of change in color units rather than a calibrated color.

Discussion

The D2.2 plan can now say that the placed axis is where red is read from, in the transformer as in the autoencoder. Removing it removes the concept, in proportion to how red the line is, and the write stays inside the bound the clean geometry set. There is more to say about the operator and about depth.

The whole-stream projection costs a little on non-red lines, and that cost comes from the syntax positions, whose embeddings have a constant component on the axis. Two arms avoid it: the edit restricted to the operand positions, and the shaped suppression, which by construction leaves the low-alignment bulk alone. Of the two, only the shaped suppression needs no knowledge of the syntax of the line, so it is the natural operator for the anchored-op experiments and for the shaped-suppression item.

At the M1 threshold the removal by the shaped suppression is only partial: the embedding alignment of the red operand sits inside the ramp, and the blocks bring the operand partway back at the deeper slices. So the threshold is something to tune, and the strength sweep says the response to a partial removal is graded, which gives the tuning something to grade against.

Repulsion is the shape to reach for when the dose is a target alignment rather than a fraction removed, since it sets the landing alignment where the suppression leaves it to the re-normalization (repulsion item).

On depth: the concept is read from the operand states before the second block, the last block does not read it at all, and a token-only removal is neither clean nor cheap, because the blocks partly re-derive the axis from an off-axis input. That is the case for the layer sweep, and it says what the sweep should find. A removal acting at the embedding and the first block should do the work of the primary intervention, and one that starts later should not.

The fallback experiment has its reference numbers. Without a fallback term, a red line decodes in the color vocabulary to an answer near the true mix, computed as if the red operand were a less red color, and which near miss it lands on varies by seed. A fallback that is trained rather than left to the model should make the response reproducible across seeds; the seed-agreement fraction is the number to move.

The bound held layer by layer, and one seed still paid a cost on non-red lines from a bounded write. The layer-local statement is the one we can defend from the geometry we placed. What the blocks then do with a bounded rotation is a property of the trained model, and the seed spread says it varies. To bound the behavior rather than the write, the anti-subspace term would have to reach the syntax positions, or the operator would have to skip them.

---

1. The *redness* of a line is the redness of its redder operand. *Damage* is how much probability the correct answer loses under intervention, clean minus intervened. ↩

2. The unit vector pointing from the mean non-red state to the mean red state. It is the simplest fitted direction, and the one most steering work uses. ↩

3. A linear readout of redness, fitted by least squares with an ℓ_2 penalty. The direction is its weight vector, normalized. It finds what can be read off the state linearly, which need not be what the blocks use. ↩

4. Least-squares concept erasure (Belrose et al., 2023). It zeroes the covariance between the states and the label, and is the smallest linear edit, in the covariance-whitened norm, after which no linear probe can recover the concept. The projection is oblique: it removes the covariance-weighted direction along a different one. It usually edits less than the other two, since it removes only the linear trace. ↩

5. By the redness label, $r(1 - g/2 - b/2)$: less red in the red channel, or more green or blue. ↩