Ex 2.2.9: the grammar handover

The last four experiments each changed one thing about the setup we anchor *red* in: more operations, answers drawn stochastically rather than rounded, a label that covers the whole line, and a readout table separate from the embeddings. Each looked fine on its own. This experiment turns all four on at once and asks whether *red* still lands where we put it and still comes out cleanly, with two reference conditions that each switch one change back.

Red lands. Removal is clean on eight of the eleven ops and misses the gate on the three that take one HSV attribute from their second operand, so the handover is not adopted as it stands. The whole-line label and the separate readout cost nothing the gates can resolve. One seed in twenty holds the anchor a little less well by the end of training, which the anneal schedule may explain.

Findings

- Does the model still learn the task? (H1) — **gate cleared.** On every op the `handover` seed mean is within 0.02 of the control's. The largest shortfall is `hsvmix` at -0.006 (band 0.008).
- Does *red* still land where we put it? (H2) — **gate missed.** `handover` clears margin m_line, grading r^2, contrast, lead at embedding, latch, and misses retention. 1 of 20 qualifying seeds ends below 0.8 of its peak (the lowest at 0.77); the seed mean is 0.88 against ex-2.2.3's 0.99. The paired $\tilde{\alpha}$ prediction holds: $\tilde{\alpha}$ at op1 is 0.28, above the 0.1 the earlier experiments gated, as expected with the readout untied (the output loss no longer holds the non-red embedding rows off the axis; see the H2 results), and the non-red deficit it was paired with is 0.004, inside H3's 0.05 gate.
- Can we still take *red* out cleanly? (H3) — **gate missed.** Removal: on `hue-hsv`, `sat-hsv`, `value-hsv` `handover` still answers more than 20% of its removal lines with the axis taken out (the most on `value-hsv`, at 38% of its clean expected exact match), where the gate asks for 20% or less. Selectivity holds: the non-red `mix` deficit is 0.004 against the 0.05 gate (band 0.013).
- Does the separate readout keep the axis off the syntax tokens? (H4) — **held in part.** The first prediction holds except on `=`. The syntax embeddings carry less of the axis on `handover` (0.033) than on `handover-tied` (0.169), a gap well over the band (0.006). `=` is the one word that did not reach the ceiling: `handover` reads 0.025 there, against ex-2.2.7's hard-zeroed 0.000 (band 0.008), and its readout row for `=` carries 0.145, so the component moved to the readout. The second does not hold: the non-red `mix` deficit is not lower on `handover`, 0.004 against 0.012 on `handover-tied` (band 0.017).
- What does the whole-line label cost? (H5) — **held.** Both predictions hold: the seed-mean deficits are 0.004 (`handover`) and 0.006 (`handover-slot`), band 0.013; 0 `handover` seeds and 0 `handover-slot` seeds sit above 0.07.
- Whether the handover is adopted — **not adopted.**

> **How to read this draft**
>
> This is a preregistration. The conditions, the gates, and the decision rule below were written before any run and frozen at commit `7aabc72`; everything after that commit is either the results filled into the frozen sections or exploratory. Each hypothesis section opens with the background, then gives the prediction we will be scored on. Results go into each section in place once they exist. Anything we think of after seeing the data goes under Exploratory analyses, marked as post hoc. Every count in the method is computed from `experiment.py` at render time.

Why this experiment

Ex-2.2.3 anchored *red* on a grammar of six operations and then tried to remove it. Removal was partial on four ops, and the ops themselves were confounding. On `lighten`, say, most lines with a red operand have an answer that stays the same when the operand is a little less red, so a model that has lost *red* can still answer them.

Our investigations suggest:

1. **A wider table of operations** (ex-2.2.4). Drop `add`, add three ops whose answers spread through the color cube, and add three more that take one attribute from one operand and the rest from the other. Those last three are the first ops where the order of the operands matters.
2. **Stochastic answers instead of rounded ones** (ex-2.2.5). When an answer lands between two grid colors, the corpus picks one of them at random, rather than rounding deterministically. The model then learns a spread of possible answers rather than one biased token.
3. **A label that covers the whole line** (ex-2.2.6). Previously, only the two operands could draw a label, and the pull covered the prompt. Now the anchor is told "this line is about red", and it pulls wherever in the line it finds the most red, answer included. That is the shape a document-level label will have in M3. The first pilot found it costs nothing; a later one saw a few seeds lose selectivity under it, so it goes in with a check.
4. **A separate, untied readout table** (ex-2.2.7). The output layer of the model was sharing a table with the input embeddings, and that put part of the *red* axis onto the tokens `=` and `mix`. Giving the output its own table keeps the axis off those tokens.

In ex-2.2.8, we found that the removal operator to score is the plain projection, which takes the axis out everywhere, with the operand-only edit beside it as the selective reference.

Here the four are tried together, at the same number of seeds as the reference, to inform the grammar and recipe the anchored-op experiments will use. We call that the *grammar of record*: the one setup every later D2.2 experiment is built on. It is a handover because the grammar of record changes hands here: until this runs it is the one from ex-2.2.3; after it, if the gates hold, it is this one.

Conditions

Every condition trains fresh models on the new grammar: table A+, 300,000 lines drawn uniformly over the 11 ops, answers drawn stochastically, one fifth of the pairs of each op held out. That is about 27,272 lines per op, drawn with replacement from the op's 37,325 trainable lines, so a model sees about 52% of them at least once (77% of the unordered pairs), against 36% (59%) at six ops. The recipe is the point adopted in ex-2.2.3 (`recipe-short`), unchanged: λ_a = 0.1, annealed over the last tenth of training as ex-2.1.10 did, τ = 0.1, the anti-subspace weight from 2.5× the anchor weight down to 0.3× by 90% of training, 50 epochs (4,950 steps). *Red* is anchored to e_1 at every slice, the embedding included.

condition	anchor	labeller	readout	lines per op	epochs (steps)	seeds
control, reference: un-anchored	none	—	untied	27,272	50 (4,950)	5
handover, candidate: the recipe, untied readout, whole-line labeller	λ_a = 0.1, τ = 0.1	whole line	untied	27,272	50 (4,950)	20
handover-slot, reference: as handover, with ex-2.2.3's either-slot labeller	λ_a = 0.1, τ = 0.1	either slot, prompt span	untied	27,272	50 (4,950)	20
handover-tied, reference: as handover, with the tied readout	λ_a = 0.1, τ = 0.1	whole line	tied	27,272	50 (4,950)	9
handover-narrow, exploratory: as handover, lines per op held at ex-2.2.3's count	λ_a = 0.1, τ = 0.1	whole line	untied	16,666	82 (5,002)	5

`handover` has everything enabled. It is the one candidate for the grammar of record, at the same twenty seeds as the reference, and the gates are read on it alone.

`handover-slot` switches the label back to the one from ex-2.2.3: only the two operands can draw a label, and the pull covers the prompt. Beside `handover` it is the selectivity check ex-2.2.7 asked for, at twenty seeds so that the comparison has the resolution of the gates. It is not a fallback: that labeller needs to know which tokens are the operands, and M3 will not have that.

`handover-tied` switches the readout back to the shared table. Beside `handover` it shows what untying does on this grammar. Nine seeds, as the pilot had.

`control` has no anchor at all. It sets the task bar for H1 and the calibration bar for the drawn answers.

`handover-narrow` is exploratory. E4 in ex-2.2.3 found the operand cube less decodable at six ops than at three, and could not tell whether the op count or the lines per op was responsible. The main corpus gives each of the eleven ops about 27,272 lines, more than the 16,666 each op had at six. This condition holds lines per op at that six-op count, using a smaller corpus for more epochs so that the step count matches.[1]

The reference is not retrained. It is `recipe-short` from ex-2.2.3, at twenty seeds on the six-op grammar, and its stored statistics are printed beside every placement read.

The removal operators

We score every anchored checkpoint through the eval contract in `sca.intervention`, on each op's probe lines.

- **`projection`**: the axis projected out at every slice and every position, at full strength. This is the gated operator, as ex-2.2.8 proposed.
- **`operands`**: the same edit at the two operand positions only. It is the selective reference, since it never touches the syntax tokens and so cannot cost anything there.
- **`shaped-a0.4-p0`**: a thresholded projection that leaves any state below alignment 0.4 alone. Ex-2.2.8 found it removes less than the projection at no cost on the six-op grammar; what we learn here is whether that smaller removal clears the removal gate on a table built to need *red*. Scoring it is one more pass over the same checkpoints.

Glossary

Red line, non-red line

A line is *red* when its redder operand has redness at least 0.8, and *non-red* when neither operand is above 0.2. Redness is r·(1 − g/2 − b/2) on the unit scale. The redness of the redder operand is the line's *dose*.

Removal lines

The red lines on which the true answer would move a long way if the red operand had no red in it (its R channel set to zero). These are the lines where losing *red* has to show; on the others the op does not need it. On `mix` every red line is a removal line; on the other ops at least 62% of them are, and the count per op is in the method.

Red-answer lines

Non-red lines whose true answer is red (white minus cyan, under `difference`). Projecting the axis out at `=` takes *red* from the state that has to produce the answer, so a miss there is removal on the output side. They stay in the non-red deficit and are also counted on their own, and the deficit is reported with and without them. `mix` has none, so the gated read is the same either way.

Expected exact match

The task score we measure. With drawn answers, the correct answer to a line is spread over two or more colors, so a single exact-match check would penalise a model for a right answer. Instead we take the chance that an answer drawn from the model agrees with one drawn from the true distribution. Its ceiling is that same chance for the true distribution against itself (Σq^2), which is below 1 on the ops that round.

m_line

How far *red* is pushed onto the axis: at the reddest position in the span, the label-weighted mean alignment minus the unweighted mean, averaged over slices. Ex-2.1.10's margin, read on the `mix` lines.

$\bar{\alpha}$ (containment)

The mean alignment with the axis over all 216 colors at op1. High when colors that are not red drift onto the axis at that position.

Lead

On the red lines, the largest share of the pull that any one position receives at the embedding slice. High when the pull is concentrated on one token rather than spread over the span.

Contrast

How much more the pull lands on op2 when op2 is the red operand than when it is not, averaged over the post-attention slices. It says the pull follows the red operand rather than a fixed position.

Grading r^2

How well alignment with the axis tracks redness across colors, fit against the sim^1.5 target of ex-2.1.11.

Retention

Whether the placement holds to the end of training, after the anchor weight's anneal: the final m_line as a share of the run's peak.

Latch

A run in which the non-red group puts more than half of its softmin weight on one position: the pull has found a position rather than a concept. The position read is op1, the slot the earlier experiments saw it latch to.

Deficit

How much expected exact match a model loses on the non-red lines when the axis is projected out. This is a measure of the selectivity; a clean removal would cost nothing.

Band

Our measurement precision. Two seed means are told apart only when they differ by more than $2\sigma\sqrt{(1/n_a + 1/n_b)}$, with σ the per-run spread and n the seed counts. Smaller differences are reported as unresolved.

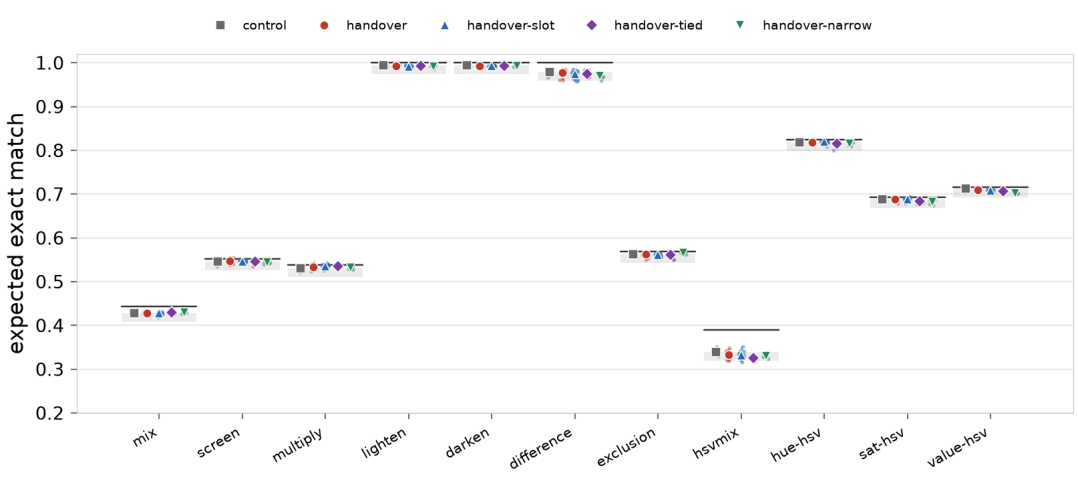
Does the model still learn the task? (H1) ✓ Pass

Background. Eleven ops in the same number of lines, with drawn answers, is a harder corpus than six ops with rounded ones. Before we read anything about the anchor we need to know that the anchored model learns the task as well as an un-anchored model does on this corpus.

Prediction. For each of the 11 ops, the `handover` seed-mean expected exact match on held-out lines is within 0.02 of the control. Partial: every op within 0.05. Contrary: more than 0.05 below the control on some op. A comparison that misses by less than a band is reported as unresolved rather than as a miss; the band on this read is computed from the runs and printed beside the gate. The reference conditions are read on the same table without a gate, and a reference that does not learn the task has its later comparisons read with that caveat.

We expect this to hold: ex-2.2.5 saw no task cost from the drawn answers, ex-2.2.7 none from the untied readout. Whether the control itself learns the grammar is checked before the freeze, on one seed, rather than gated here. The number to watch is the order-sensitive subset, since an op that reads operand order asks the model for something the six-op grammar never did.

Results. The anchored model learns the task about as well as the un-anchored one. Across the 11 ops the `handover` seed mean sits between -0.006 (`hsvmix`) and +0.003 (`multiply`) of the control's. On the order-sensitive subset the differences are +0.000, -0.001, -0.003, so reading operand order costs the anchored model nothing extra. `handover-narrow` , with lines per op held at the six-op count, sits between -0.010 (`value-hsv`) and +0.002 of the control. One op sits well under its ceiling on every condition: `hsvmix` , at 0.05 below on the control. That is the gap the calibration look found before the freeze (Before the freeze): more passes over the corpus widened it rather than closing it, and it is the same on the anchored conditions, so it is a property of the op and the corpus rather than of the anchor.



Expected exact match on held-out lines, per op and condition. *Each small dot is one seed, the larger mark the seed mean, and the thin bar behind them the seed range. The grey strip under each op runs from the control's mean down to 0.02 below it: a `handover` mean inside the strip passes H1 on that op. The bar above each op is the ceiling the drawn answers allow (Σq^2), which is 1 on the ops that never round.*

op	ceiling	control	`handover` (Δ) ↑	`handover-slot` (Δ) ↑	`handover-tied` (Δ) ↑	`handover-narrow` (Δ) ↑	band
`mix`	0.44	0.427	**0.429 (+0.001)**	0.427 (-0.000)	0.429 (+0.002)	0.430 (+0.002)	0.002
`screen`	0.55	0.546	**0.547 (+0.001)**	0.546 (+0.001)	0.545 (-0.001)	0.544 (-0.001)	0.005
`multiply`	0.54	0.531	**0.534 (+0.003)**	0.535 (+0.004)	0.536 (+0.004)	0.532 (+0.001)	0.005
`lighten`	1.00	0.994	**0.993 (-0.001)**	0.992 (-0.002)	0.993 (-0.001)	0.991 (-0.003)	0.002
`darken`	1.00	0.995	**0.993 (-0.001)**	0.993 (-0.001)	0.993 (-0.001)	0.993 (-0.002)	0.002
`difference`	1.00	0.978	**0.977 (-0.001)**	0.974 (-0.005)	0.974 (-0.004)	0.969 (-0.009)	0.007
`exclusion`	0.57	0.563	**0.563 (-0.000)**	0.560 (-0.003)	0.561 (-0.002)	0.565 (+0.002)	0.004
`hsvmix`	0.39	0.340	**0.333 (-0.006)**	0.331 (-0.008)	0.326 (-0.014)	0.330 (-0.009)	0.008
`hue-hsv`	0.82	0.819	**0.819 (+0.000)**	0.819 (+0.001)	0.815 (-0.003)	0.815 (-0.003)	0.003
`sat-hsv`	0.69	0.689	**0.688 (-0.001)**	0.687 (-0.001)	0.684 (-0.005)	0.683 (-0.006)	0.003
`value-hsv`	0.72	0.712	**0.710 (-0.003)**	0.709 (-0.004)	0.707 (-0.006)	0.703 (-0.010)	0.003

Seed-mean expected exact match on held-out lines, per op. *Each condition's column gives its mean and, in brackets, the difference from the control. A bold `handover` entry is within 0.02 of the control, which is the H1 gate. The band is the smallest difference from the control the read resolves on that op, from the per-run spread of the two conditions.*

op	P(mode) control	P(mode) handover	KL control ↓	KL handover ↓
`mix`	0.44	0.44	0.13	0.13
`screen`	0.64	0.64	0.08	0.08
`multiply`	0.63	0.64	0.09	0.09
`lighten`	0.99	0.99	0.01	0.01
`darken`	0.99	0.99	0.01	0.01
`difference`	0.98	0.98	0.03	0.02
`exclusion`	0.65	0.66	0.08	0.08
`hsvmix`	0.39	0.38	0.35	0.38
`hue-hsv`	0.86	0.86	0.05	0.06
`sat-hsv`	0.75	0.76	0.08	0.08
`value-hsv`	0.77	0.77	0.05	0.06

How the answer mass is spread, per op. *P(mode) is the mass the model puts on the most likely true answer; KL is the divergence from the true answer distribution to the model's, in nats, averaged over held-out lines. Both are seed means. On the ops that never round, the true distribution is one color, so P(mode) is the expected exact match and KL the negative log-likelihood.*

> **Pass**
>
> On every op the `handover` seed mean is within 0.02 of the control's. The largest shortfall is `hsvmix` at -0.006 (band 0.008).

Does *red* still land where we put it? (H2) ✗ Miss

Background. The recipe was tuned on the six-op grammar, but here the corpus, the labeller, and the readout all change. We read the same placement statistics on the same `mix` lines, so the numbers are comparable to ex-2.2.3. The question is whether they are still inside the gates set there.

Prediction. On the `mix` lines, for `handover`, under its own labeller (as ex-2.2.6 read it):

- *Margin:* seed-mean m_line at least 80% of the reference's 0.4202, so 0.336; partial from 60%.
- *Grading:* grading r^2 at least 90% of the reference's 0.782, so 0.704.
- *Concentration, attribution, retention, latch:* lead at the embedding at least 0.4; contrast at least 0.2 (partial from 0.1); every run that reaches m_line 0.2 ends at 0.8 of its peak; no run latched, which the reference held at twenty seeds.

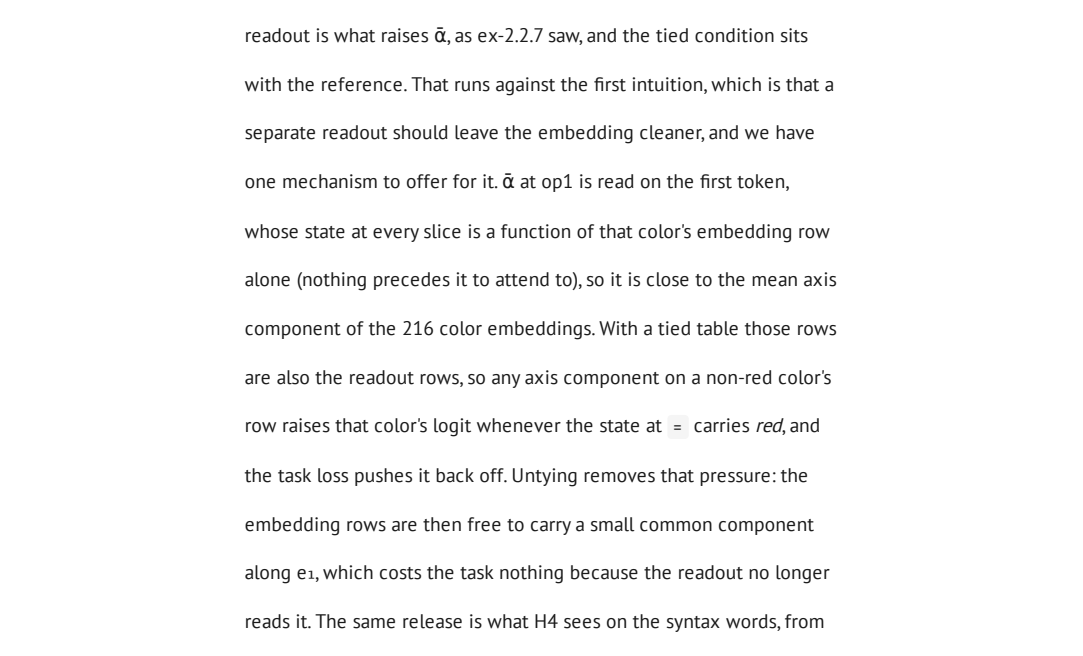
Containment is a prediction here, not a gate. Every experiment since ex-2.1.8 has gated $\bar{\alpha}$ at op1 at 0.1. At nine seeds, ex-2.2.7 read 0.13 on its untied condition and 0.23 on its untied whole-line condition, against 0.08 on the reference, so we expect `handover` above 0.1, and `handover-slot` nearer to it. We do not know why untying raises it. What we do know is that the pull still lands on the red operand in those runs (lead, contrast, and latch all sat at the reference's values), so this is other colors drifting a little onto the axis at op1 rather than the pull finding a position. What that drift costs, if anything, is what H3's selectivity read measures. So the prediction is the pair: $\bar{\alpha}$ above 0.1 on `handover`, and a non-red deficit inside H3's gate all the same. If both hold, the drift is recorded for the anchored-op prereg to watch; if the deficit fails too, containment is the first place to look for why.

`handover-tied` is read on the same statistics, so that we can attribute the containment read. If the tied condition sits with the reference (more contained) and both untied conditions sit higher, the untied readout is what moved it.

Results. The anchor does land where we put it, holding the spot only a little less well than the references do. On the `mix` lines `handover` reaches a margin of 0.426 against the reference's 0.392 (the gate is 0.336), a grading r^2 of 0.89, a lead at the embedding of 0.91, and a contrast of 0.84.

$\bar{\alpha}$ at op1 reads 0.28 on `handover`, 0.18 on `handover-slot`, and 0.16 on `handover-tied`, against 0.08 at the reference. So the untied readout is what raises $\bar{\alpha}$, as ex-2.2.7 saw, and the tied condition sits with the reference. That runs against the first intuition, which is that a separate readout should leave the embedding cleaner, and we have one mechanism to offer for it. $\bar{\alpha}$ at op1 is read on the first token, whose state at every slice is a function of that color's embedding row alone (nothing precedes it to attend to), so it is close to the mean axis component of the 216 color embeddings. With a tied table those rows are also the readout rows, so any axis component on a non-red color's row raises that color's logit whenever the state at `=` carries *red*, and the task loss pushes it back off. Untying removes that pressure: the embedding rows are then free to carry a small common component along e_1, which costs the task nothing because the readout no longer reads it. The same release is what H4 sees on the syntax words, from the other side: there the tied table had put axis onto `=` because it helped the output, and untying let it come off. Both are the output loss letting go of the embedding. The check, for the next round, is the mean axis component of the non-red embedding rows on `handover` against `handover-tied`, which the stored tables can give without a run.

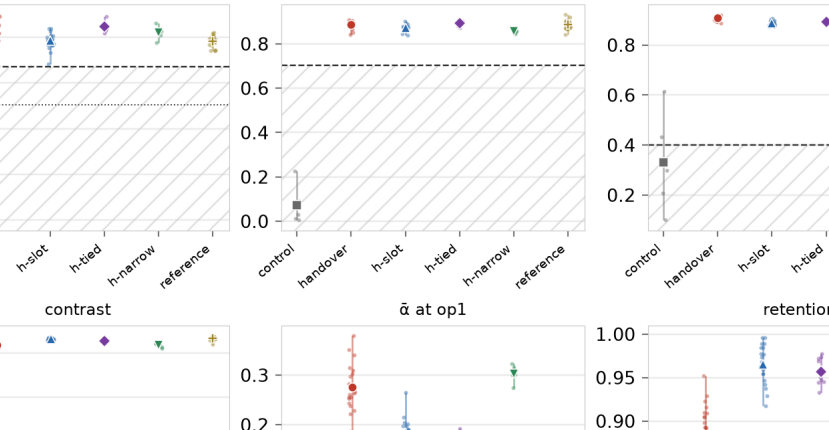
Retention is where the hypothesis misses: the seed mean is 0.88 on `handover` against 0.96 on `handover-slot`, 0.96 on `handover-tied`, and 0.99 at ex-2.2.3, and 1 of 20 qualifying seeds ends at 0.77, under the 0.8 gate. One caveat on every ratio against the reference: those runs trained for 1,650 steps and ours for 4,950, so a margin or an $\bar{\alpha}$ that sits above the reference could be the longer training as much as the grammar. The same difference bears on retention. The anchor weight anneals over the last tenth of training, which here is 495 steps against 165 at the reference, so the margin has three times as many steps to drift after the weight comes down, and the seed spread on $\bar{\alpha}$ and retention looks like the spread we saw before the schedules were tuned in D2.1, which a later anneal settled. A schedule read on this grammar (a later or shorter anneal, or one set in steps rather than as a share of training) is the first thing to try on this miss.



Placement of red on the `mix` lines, per condition, beside the six-op reference. *The read is on `mix` because the gates were set on `mix` at ex-2.2.3, so these numbers compare with the reference's. One panel per statistic; each small dot is one seed, the larger mark the seed mean, and the thin bar the seed range, with ex-2.2.3's twenty reference seeds at the right. The dashed line is the gate (margin 0.336, grading r^2 0.704, lead 0.4, contrast 0.2, retention 0.8), the hatched region the side that misses it, and the dotted line the partial level where there is one. The retention column for `control` is empty because there is no anchor for it to hold. On the $\bar{\alpha}$ panel the dashed line is the 0.1 the earlier experiments gated, which here is a prediction: we expected the untied conditions above it.*

condition	margin m_line ↑	grading r^2 ↑	lead at embedding ↑	contrast ↑	$\bar{\alpha}$ at op1 ↓	retention ↑	latch (max) ↓
`control`	0.031 (-0.00–0.07)	0.073 (0.01–0.23)	0.330 (0.10–0.62)	0.088 (-0.05–0.22)	0.016 (-0.03–0.10)	–	0.40
`handover`	**0.426 (0.39–0.45)**	**0.887 (0.84–0.91)**	**0.909 (0.89–0.92)**	**0.838 (0.81–0.86)**	0.276 (0.19–0.38)	0.88 (0.8–1.0)	**0.00**
`handover-slot`	0.392 (0.34–0.42)	0.871 (0.84–0.90)	0.888 (0.87–0.91)	0.864 (0.85–0.87)	0.184 (0.14–0.26)	0.96 (0.9–1.0)	0.01
`handover-tied`	0.423 (0.41–0.45)	0.894 (0.87–0.91)	0.894 (0.89–0.90)	0.855 (0.85–0.87)	0.162 (0.13–0.19)	0.96 (0.9–1.0)	0.01
`handover-narrow`	0.411 (0.39–0.43)	0.856 (0.85–0.87)	0.912 (0.91–0.92)	0.837 (0.82–0.84)	0.303 (0.27–0.32)	0.85 (0.8–0.9)	0.00
reference (ex-2.2.3)	0.392 (0.37–0.41)	0.886 (0.84–0.93)	0.826 (0.78–0.86)	0.868 (0.84–0.88)	0.083 (0.04–0.15)	0.99 (1.0–1.0)	0.02

Placement statistics on the `mix` lines, per condition. *Seed mean with the seed range in brackets; retention is read on the runs whose peak m_line reached 0.2, and latch is the largest non-red softmin weight at op1 over the seeds, against 0.5. Bold `handover` entries clear their gate. The bands against the reference, from the per-run σ frozen at ex-2.2.3: margin m_line 0.006, $\bar{\alpha}$ at op1 0.013, grading r^2 0.015, contrast 0.004.*



Where the label's weight lands, by token role. *The red group's softmin weight at each of the six positions, averaged over the four post-attention slices and the seeds, one series per anchored condition, with the seed range as a band. The whole-line labeller (every condition but `handover-slot`) may place weight on the answer and the newline; the either-slot labeller only on the operands.*

> **Miss**
>
> `handover` clears margin m_line, grading r^2, contrast, lead at embedding, latch, and misses retention. 1 of 20 qualifying seeds ends below 0.8 of its peak (the lowest at 0.77); the seed mean is 0.88 against ex-2.2.3's 0.99. The paired $\bar{\alpha}$ prediction holds: $\bar{\alpha}$ at op1 is 0.28, above the 0.1 the earlier experiments gated, as expected with the readout untied (the output loss no longer holds the non-red embedding rows off the axis; see the H2 results), and the non-red deficit it was paired with is 0.004, inside H3's 0.05 gate.

Can we still take *red* out cleanly? (H3) ✗ Miss

Background. Take the *red* axis (e_1) out of every state. On the lines that need *red*, does the model fail? On the lines that never had any, does it still answer? The first is removal, the second selectivity. Ex-2.2.3 could only show removal on two of six ops, because the other four mostly did not need *red*. Table A+ was chosen so that every op has lines that do, and the removal read is scored on those lines only.

Prediction. Under `projection`, for `handover`:

- *Removal:* on the removal lines of every op, the model keeps at most 20% of its clean expected exact match, seed mean. This is the red-accuracy gate of ex-2.2.3, read as a ratio, because with drawn answers the clean value sits below 1 on the ops that round.
- *Selectivity:* the seed-mean deficit on the non-red `mix` lines is at most 0.05; partial to 0.1. The partial band is a reporting level, as it was in ex-2.2.3: the decision rule asks for the gate. The read stays on `mix` so that it means what it meant at the reference; `hsvmix` and every other op are reported beside it; what a switch to `hsvmix` would change is set out in the method (The reference op). On the reference, ex-2.2.8 read 0.040 on `mix`, which is inside the gate by less than a band. With the readout untied we expect the deficit to fall toward the `operands` row, which read 0.012. Bands on the deficit use the per-run spread ex-2.2.8 measured under `projection` at the reference's twenty seeds, per op (ex-2.2.8, `projection` on `recipe-short`, twenty seeds, per op), frozen so that the candidate's own spread does not move its verdict. Non-red lines whose true answer is red are in the deficit and also counted on their own (see the glossary); `mix` has none.

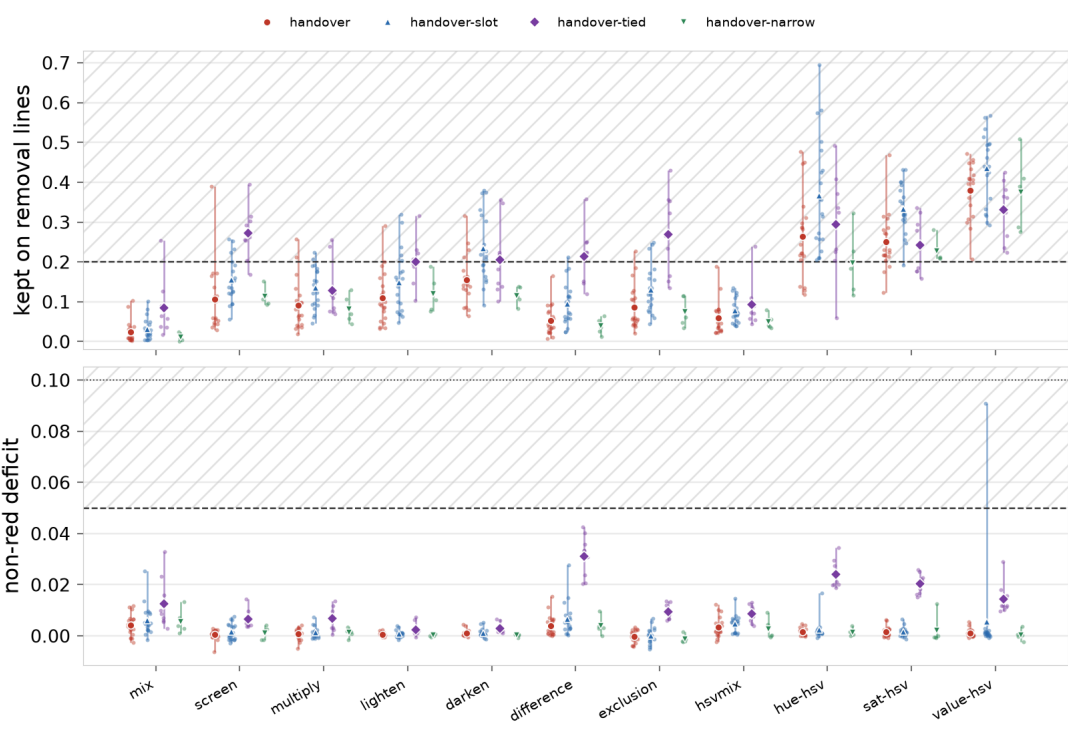
Two further reads have an expected direction and no gate.

How far the answer moves. On the removal lines, we measure the distance in the unit cube between the answer the model decodes under `projection` and the true answer. Beside it we put the distance the true answer itself moves when the red operand loses its red. Ranking the ops by each distance should give the same order, with `value-hsv`, `hue-hsv`, and `sat-hsv` at the top of both.

The operand-only edit. On the ops whose answer is computed at `=` from both operands together, `operands` should remove less than `projection`, since the `=` state keeps its axis component. Elsewhere the two should remove the same amount.

Results. Taking the axis out removes *red* on 8 of the 11 ops: on their removal lines `handover` keeps between 2% (`mix`) and 15% (`darken`) of its clean expected exact match under `projection`. On `hue-hsv`, `sat-hsv`, `value-hsv` it keeps 26%, 25%, 38%, above the 20% gate; the slot split under The order-sensitive subset, by slot says which lines those are. On the non-red `mix` lines the deficit is 0.004 (seeds from -0.003 to 0.012), against 0.040 at the reference; the `operands` edit, which leaves the syntax positions alone, reads 0.004. No op's non-red deficit exceeds 0.004.

Of the two reads with a direction and no gate, neither went the predicted way. The rank correlation between how far the true answer moves and how far the model's answer moves under `projection` is 0.05: the largest true moves are on `value-hsv`, `sat-hsv`, `hue-hsv` and the largest model moves on `difference`, `hsvmix`, `exclusion`. And `operands` removes as much as `projection` on every op (the widest gap is -0.019, on `sat-hsv`), where the prediction had it removing less on the ops computed at `=`.



Removal and selectivity under `projection`, per op and anchored condition. *Top: the share of the clean expected exact match the model keeps on each op's removal lines (red lines whose answer moves by at least 0.4 when the red operand loses its red); the dashed line is the 20% gate, and the hatched region above it is the miss. Bottom: the deficit in expected exact match on each op's non-red lines; dashed at the 0.05 gate, hatched above it, and dotted at the 0.1 partial level. The gate on the deficit is read on `mix` only. Each small dot is one seed, the larger mark the seed mean, and the thin bar the seed range.*

op	`handover` ↓	`handover-slot` ↓	`handover-tied` ↓	`handover-narrow` ↓	`operands` ↓	`shaped` ↓
`mix`	**0.02**	0.03	0.09	0.01	0.02	0.04
`screen`	**0.11**	0.15	0.27	0.11	0.11	0.12
`multiply`	**0.09**	0.13	0.13	0.08	0.09	0.12
`lighten`	**0.11**	0.15	0.20	0.12	0.11	0.13
`darken`	**0.15**	0.23	0.21	0.12	0.16	0.18
`difference`	**0.05**	0.09	0.21	0.04	0.06	0.07
`exclusion`	**0.09**	0.13	0.27	0.07	0.09	0.11
`hsvmix`	**0.06**	0.08	0.09	0.05	0.06	0.08
`hue-hsv`	0.26	0.37	0.29	0.20	0.28	0.33
`sat-hsv`	0.25	0.33	0.24	0.23	0.26	0.29
`value-hsv`	0.38	0.43	0.33	0.37	0.40	0.43

Removal, per op: the share of clean expected exact match kept on the removal lines. *Seed means under `projection`, one column per anchored condition, then `handover` under the `operands` and `shaped-a0.4-p0` operators. Bold entries are inside the 20% gate, which is read on `handover`.*

op	`handover` ↓	`handover-slot` ↓	`handover-tied` ↓	`handover-narrow` ↓	band
`mix`	**0.004**	0.006	0.012	0.006	0.013
`screen`	0.001	0.002	0.006	0.001	0.006
`multiply`	0.001	0.001	0.007	0.001	0.007
`lighten`	0.001	0.001	0.003	0.000	0.003
`darken`	0.001	0.001	0.003	0.000	0.004
`difference`	0.004	0.007	0.031	0.004	–
`exclusion`	-0.000	0.000	0.009	-0.001	–
`hsvmix`	0.003	0.005	0.009	0.003	–
`hue-hsv`	0.002	0.002	0.024	0.001	–
`sat-hsv`	0.002	0.002	0.020	0.002	–
`value-hsv`	0.001	0.006	0.014	0.000	–

Selectivity, per op: the drop in expected exact match on the non-red lines. *Seed means under `projection`, one column per anchored condition. The gate (0.05) is read on `mix` only, where the bold entry is inside it. The band is the smallest difference from the reference the read resolves, from the per-run σ ex-2.2.8 measured at the reference's twenty seeds (ex-2.2.8, `projection` on `recipe-short`, twenty seeds, per op); the ops the six-op grammar did not have carry none.*

op	true answer moves	floor	clean	`projection`	`operands`
`mix`	0.51	0.00	0.01	0.37	0.37
`screen`	0.69	0.04	0.05	0.45	0.45
`multiply`	0.66	0.04	0.05	0.48	0.47
`lighten`	0.70	0.00	0.00	0.49	0.49
`darken`	0.67	0.00	0.00	0.47	0.47
`difference`	0.76	0.00	0.01	0.61	0.61
`exclusion`	0.77	0.04	0.04	0.53	0.52
`hsvmix`	0.65	0.11	0.13	0.53	0.53
`hue-hsv`	0.89	0.03	0.03	0.37	0.36
`sat-hsv`	1.00	0.04	0.04	0.52	0.50
`value-hsv`	1.01	0.04	0.04	0.40	0.39

How far the answer moves on the removal lines, per op. *Distances in the unit RGB cube, `handover` seed means. True answer moves is how far the true answer moves when the red operand loses its red (the to-zero move, from the corpus): the distance a model that had lost red would be expected to miss by. The last three columns are the expected distance from the model's answer distribution to the line's answer: on the clean pass, and under the two projection operators. Floor is that same distance for the true answer distribution itself, which is above zero on the ops that round because the drawn answer can land on either neighbour; the clean column is read against it.*

> **Miss**
>
> Removal: on `hue-hsv`, `sat-hsv`, `value-hsv` `handover` still answers more than 20% of its removal lines with the axis taken out (the most on `value-hsv`, at 38% of its clean expected exact match), where the gate asks for 20% or less. Selectivity holds: the non-red `mix` deficit is 0.004 against the 0.05 gate (band 0.013).

Does the separate readout keep the axis off the syntax tokens? (H4) ~ Partial

Background. In ex-2.2.7 the readout row for `=` picked up a component along the *red* axis: after a red operand, whose state sits on the axis, that is a cheap way to raise the `=` logit. With a shared table that row is also the input embedding of `=`, so the `=` token entered the residual stream carrying some *red*, and projecting the axis out at that position took away something the model was using. Giving the output its own table moved the component onto the readout row and left the input embedding mostly clean.

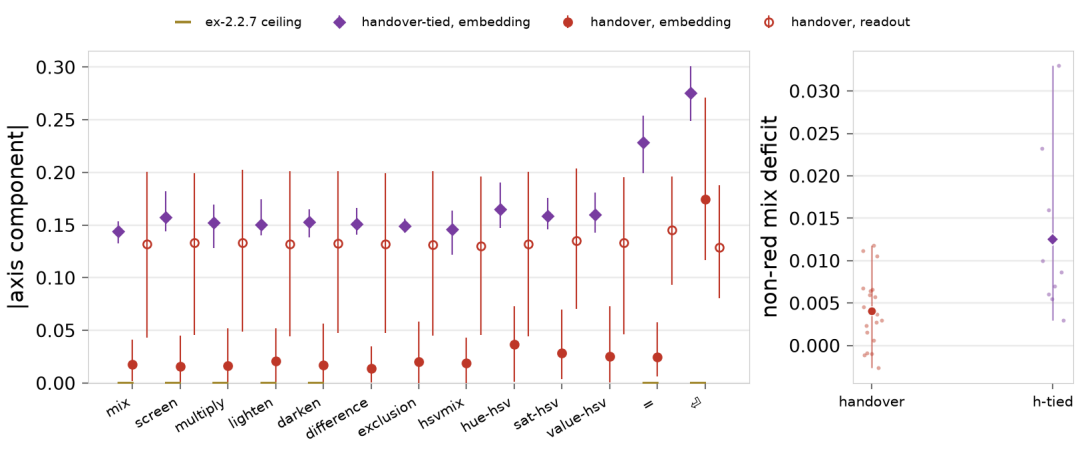
That was on the six-op grammar at nine seeds. Does it carry to eleven ops, and does it give the cleaner full-line removal?

Prediction. On `handover` against `handover-tied`, same labeller, twenty seeds against nine (the band formula takes both counts, so it is wider than between two twenty-seed conditions):

- The axis component on the syntax embeddings (`=`, the op words, and `⏎`), read from the embedding-component table of ex-2.2.7, is lower on `handover` than on `handover-tied` by more than a band (pooled within-condition, over the two conditions compared, since ex-2.2.7 published seed means and ranges rather than a per-run spread; the σ is reported beside the comparison). On `=`, `handover` sits within a band of the hard-zeroed ceiling from ex-2.2.7 (`rows-clean`, where the component is zero by construction). The component appears on the readout table of `handover` instead.
- The non-red `mix` deficit under `projection` is lower on `handover` than on `handover-tied`, by more than a band.

Neither prediction is a gate. Ex-2.2.7 already took the readout decision, and this section either confirms it or reports that it did not carry. If the second prediction fails while the first holds, then on this grammar the syntax embeddings were not where the cost came from.

Results. The separate readout keeps the axis off the syntax embeddings on this grammar too. Averaged over the op words, `=`, and `⏎`, the e_1 component of the embedding rows is 0.033 on `handover` and 0.169 on `handover-tied`. On `=` alone, `handover` reads 0.025, outside a band (0.008) of ex-2.2.7's hard-zeroed 0.000, while its readout row for `=` carries 0.145: the component moved to the readout, as it did in the pilot. Under `projection` the non-red `mix` deficit is 0.004 on `handover` against 0.012 on `handover-tied`, with a band of 0.017; both sit under the reference's 0.040, so the cost the pilot saw on the shared table is small here on either table. One word the untied table did not clean: `⏎`, whose embedding row reads 0.174 on `handover` where every other syntax word is under 0.04.



The red axis on the syntax tokens, and what it costs. *Left: the absolute axis component of each syntax word's row (the op words, `=`, and `⏎`), seed mean with the seed range as a bar, for `handover-tied`'s embedding rows, `handover`'s embedding rows, and `handover`'s readout rows (the open marks). The short bars are ex-2.2.7's hard-zeroed ceiling, where the embedding component is zero by construction. Right: the deficit in expected exact match on the non-red `mix` lines under `projection`, one small dot per seed and the seed mean as the larger mark.*

word	`handover-tied` embedding ↓	`handover` embedding ↓	`handover` readout	ex-2.2.7 ceiling
`mix`	0.144	0.018	0.132	0.000
`screen`	0.157	0.016	0.133	0.000
`multiply`	0.152	0.017	0.133	0.000
`lighten`	0.150	0.021	0.132	0.000
`darken`	0.153	0.017	0.132	0.000
`difference`	0.151	0.014	0.132	–
`exclusion`	0.149	0.021	0.132	–
`hsvmix`	0.146	0.019	0.130	–
`hue-hsv`	0.165	0.037	0.132	–
`sat-hsv`	0.159	0.029	0.135	–
`value-hsv`	0.160	0.025	0.133	–
`=`	0.228	0.025	0.145	0.000
`⏎`	0.275	0.174	0.129	0.000
all syntax words	0.169	0.033	0.133	–

Absolute axis component per syntax word. *Seed means over each condition's runs, on the embedding table and, for `handover`, on its readout table; the last column is ex-2.2.7's hard-zeroed ceiling on the embedding. The band between the two conditions on the all-words mean is 0.006 (σ 0.008, pooled within-condition, over the two conditions compared); on `=` the band against the ceiling is 0.008.*

> **Partial**
>
> The first prediction holds except on `=`. The syntax embeddings carry less of the axis on `handover` (0.033) than on `handover-tied` (0.169), a gap well over the band (0.006). `=` is the one word that did not reach the ceiling: `handover` reads 0.025 there, against ex-2.2.7's hard-zeroed 0.000 (band 0.008), and its readout row for `=` carries 0.145, so the component moved to the readout. The second does not hold: the non-red `mix` deficit is not lower on `handover`, 0.004 against 0.012 on `handover-tied` (band 0.017).

What does the whole-line label cost? (H5) ✓ Pass

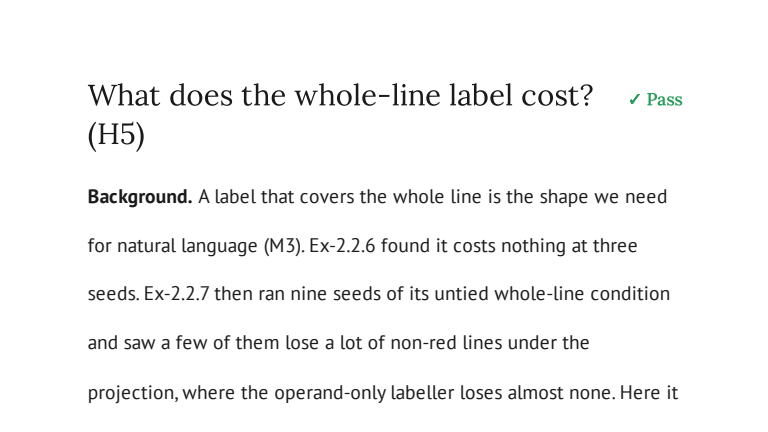
Background. A label that covers the whole line is the shape we need for natural language (M3). Ex-2.2.6 found it costs nothing at three seeds. Ex-2.2.7 then ran nine seeds of its untied whole-line condition and saw a few of them lose a lot of non-red lines under the projection, where the operand-only labeller loses almost none. Here it is read at twenty seeds against twenty.

Prediction. On `handover` against `handover-slot`, under `projection`:

- The seed-mean non-red `mix` deficit differs by less than a band, and
- the count of seeds whose deficit is above 0.07 (the level at which ex-2.2.7 read its tail) is no more than two higher on `handover` than on `handover-slot`.

The whole-line label changes two things at once: which positions the pull can land on, and which lines get a label at all, since the answer draws at its own redness rate. So a miss here says the label as a whole costs selectivity, but not which half of it did. Neither prediction is a gate, and `handover-slot` is not a fallback: we need the whole-line label, so a cost here is something to understand and fix, and the size of the gap to `handover-slot` says how much there is to fix. We are not sure which way this goes: the tail in ex-2.2.7 was three seeds out of nine, enough to expect it and too few to be sure.

Results. At twenty seeds against twenty, the whole-line label costs no selectivity we can resolve. Under `projection` the deficit is 0.004 on `handover` (worst seed 0.012) and 0.006 on `handover-slot` (worst seed 0.025); 0 and 0 seeds sit above 0.07. Under the `operands` edit the two read 0.004 and 0.004.



The non-red `mix` deficit, seed by seed. *Each small dot is one seed's drop in expected exact match on the non-red `mix` lines, the larger mark the seed mean, `handover` (whole-line label) in red and `handover-slot` (either-slot label) in blue; the left pair is under `projection` and the right pair under the `operands` edit. The dashed line is H3's 0.05 gate with the miss hatched above it, drawn as H3 draws it, and the dotted line inside that region is the 0.07 tail level ex-2.2.7 read.*

condition	deficit, `projection` ↓	seeds above 0.07 ↓	deficit, `operands` ↓	worst seed ↓
`handover`	0.004 (-0.00–0.01)	0 of 20	0.004 (-0.00–0.01)	0.012
`handover-slot`	0.006 (-0.00–0.03)	0 of 20	0.004 (0.00–0.02)	0.025
reference (ex-2.2.8)	0.040 (0.01–0.09)	1 of 20	–	0.086

The non-red `mix` deficit per condition. *Seed mean with the seed range, the count of seeds above the 0.07 tail level, the same read under the `operands` edit, and the worst seed. The band between the two conditions is 0.013, from the per-run σ ex-2.2.8 measured at the reference (ex-2.2.8, `projection` on `recipe-short`, twenty seeds, per op).*

> **Pass**
>
> Both predictions hold: the seed-mean deficits are 0.004 (`handover`) and 0.006 (`handover-slot`), band 0.013; 0 `handover` seeds and 0 `handover-slot` seeds sit above 0.07.

Decision

✗ Miss

H1 to H3 carry gates because one decision hangs on them: whether the anchored-op experiments run on this grammar. H4 and H5 are predictions, written down so that the result can be read against them, and what we do about a miss there is decided after reading it.

The handover is adopted, and `handover` becomes the grammar and recipe of record for the anchored-op experiments, when it clears H1, H2 (margin, grading, and contrast in full), and H3 in full; every other partial band is a reporting level. Otherwise it is not adopted, and the report says which gate was missed and what the reference conditions say about which change is responsible, because that sets what the next round tries: the labeller if `handover-slot` clears what `handover` missed, the readout if `handover-tied` does, and the table or the corpus if none of them does.

> **Miss**
>
> `handover` misses H3 removal (miss), so **the handover is not adopted** as it stands. Neither reference clears removal, which points at the table or the corpus. Retention is outside the rule and is reported: 1 of 20 qualifying `handover` seeds ends under the 0.8 line (the lowest at 0.77), and the seed mean, 0.88, is under `handover-slot` (0.96) and `handover-tied` (0.96). Both references keep every seed above the line, which points at the label and the readout together, though at 20 and 9 seeds, with no band on retention, one seed in twenty does not separate that from a one-arm effect.

Exploratory analyses

Read without gates, and not part of the decision. Each analysis below states what it was for, then what it found.

Lines per op

Ex-2.2.3's E4 probed how well the two operand colors can be decoded from the residual stream, and found them less decodable at six ops than at three. Was that because there were more ops, or because each op had fewer lines? `handover-narrow` gives each op as many lines as it had at six ops, where `handover` gives each about half as many again. If the operands decode better on `handover` than on `handover-narrow`, lines per op matters. If the two conditions sit together, it was the op count.

Op1 decodes at its own slot with R^2 0.82 on the control and 0.80 on `handover`, against 0.85 on `handover-narrow` and 0.52 on ex-2.2.3's six-op control. So at a matched step count, fewer lines per op does not make the operands less decodable: the narrow condition is at least as decodable as `handover`. That does not settle E4's question, though. The six-op control is far below all three, and it also trained for 1,650 steps against 4,950 here, so on this read the op count is confounded with the training length, and the three eleven-op conditions are not ordered by lines per op. The clean comparison would hold the step count and vary the op count.

target, site	`control` ↑	`handover` ↑	`handover-narrow` ↑	ex-2.2.3 `control` (six ops) ↑
op1 at op1	0.82 (0.7–0.9)	0.80 (0.7–0.8)	0.85 (0.8–0.9)	0.52 (0.4–0.7)
op2 at op2	0.72 (0.7–0.8)	0.76 (0.7–0.8)	0.78 (0.7–0.8)	0.28 (0.2–0.4)
answer at `=`	0.71 (0.6–0.8)	0.66 (0.5–0.8)	0.66 (0.6–0.8)	0.65 (0.6–0.7)

How decodable each color is from the residual stream, by condition. *Strict held-out R^2 for the RGB of op1 and op2 at their own slot and of the answer at `=`, mean over the channels and the four post-attention slices (31 negative scores clipped to zero), as a seed mean with the seed range. `control` and `handover` have 27,272 lines per op; `handover-narrow` has 16,666, the six-op count, at the same step budget. The last column is ex-2.2.3's un-anchored six-op model on the same read.*

The order-sensitive subset, by slot

For `hue-hsv`, `sat-hsv`, and `value-hsv` the probe set walks every color in both slots, so every read in H2 and H3 can be split by whether the red operand is op1 or op2. The labeller pools both operands the same way, so we expected the same placement in both slots. Removal should show in both slots too, at the rates in the method's per-slot table: a red op1 under `value-hsv` supplies the hue and the saturation, and losing those moves the answer as far as losing the value does.

We expected removal in both slots at about the rates of the method's per-slot table, and the data went the other way: on `sat-hsv` and `value-hsv` the lines with red at op1 keep 11% and 6%, and the lines with red at op2 keep 65% and 71%; `hue-hsv` keeps 31% with red at op1 and 17% at op2. Placement differs by slot too: the margin on the op1 walk is 0.364, 0.341, 0.356 against 0.303, 0.376, 0.407 on the op2 walk (`hue-hsv`, `sat-hsv`, `value-hsv`). The prediction had the labeller placing both slots the same, and in hindsight there was no reason to expect symmetry: the labeller pools both operands the same way, but the op's computation is not symmetric, so what the model needs from each slot is not either. What removal rate each slot should show, derived from the op rather than assumed, is the analysis the fast-follow owes; the discussion sketches the reading.

op	m_line, op1 walk ↑	m_line, op2 walk ↑	kept, red op1 ↓	kept, red op2 ↓
`hue-hsv`	0.364 (0.34–0.39)	0.303 (0.28–0.33)	0.31 (406)	0.17 (230)
`sat-hsv`	0.341 (0.31–0.38)	0.376 (0.33–0.43)	0.11 (406)	0.65 (116)
`value-hsv`	0.356 (0.32–0.40)	0.407 (0.35–0.46)	0.06 (389)	0.71 (404)

The order-sensitive ops by slot, on `handover`. *These ops are probed on two sets of lines: one with every palette color at op1 (the op1 walk) and one with every palette color at op2 (the op2 walk). The margin is read on each walk, and the share of clean expected exact match kept under `projection` is read on the removal lines whose red operand is op1 or op2 (line counts in brackets, pooled over both walks, so about twice the method's per-walk counts). Seed means with the seed range.*

Calibration under the anchor

P(mode) and KL from the true distribution on the rounded held-out lines, for every anchored condition against the control, as ex-2.2.5 read them. The anchor should not change them.

On the 8 ops that round, `handover` sits at a KL of 0.136 nats against 0.130 on the control, with P(mode) 0.567 against 0.566: the anchor itself leaves the calibration about where the control has it. `handover-narrow` is the one condition that stands out, at 0.207, with `hsvmix` the op that moved most (+0.187): fewer lines per op costs calibration, which is the lines-per-op question again from the output side.

condition	P(mode) ↑	KL ↓	op furthest from control (ΔKL)
`control`	0.566	0.130	`mix` (+0.000)
`handover`	0.567	0.136	`hsvmix` (+0.026)
`handover-slot`	0.566	0.138	`hsvmix` (+0.028)
`handover-tied`	0.563	0.141	`hsvmix` (+0.056)
`handover-narrow`	0.560	0.207	`hsvmix` (+0.187)

Calibration under the anchor. *On the held-out lines that round, the mass on the most likely answer and the KL divergence from the true answer distribution to the model's (nats), averaged over the 8 ops that round and over seeds. The last column names the op whose KL moved most from the control's, with the difference.*

Red-answer lines

On the ops that have them, the model's answer under `projection` on the non-red lines whose true answer is red. If the model can no longer produce red there, that is removal on the output side, and it says the readout row for red carries the axis as the embedding does.

On the 148 red-answer lines, expected exact match under `projection` is within 0.01 of the clean pass on every op that has them (the largest drop is on `value-hsv`). The model still produces red with the axis projected out of the state at `=`, so producing red at the output does not depend on that state's axis component. The last column is the non-red deficit with these lines set aside; it differs from the gated deficit by what they contribute, which is little.

op	lines	clean EEM	`projection` EEM ↑	deficit without them ↓
`difference`	11	0.97	0.96	0.004
`exclusion`	3	0.97	0.96	-0.000
`hue-hsv`	53	0.99	0.99	0.002
`sat-hsv`	52	0.91	0.91	0.001
`value-hsv`	29	0.99	0.98	0.001

The red-answer lines, on `handover`. *Non-red probe lines whose true answer is red, per op that has any: their count, the expected exact match on the clean pass and under `projection`, and the non-red deficit with these lines set aside. Seed means.*

The shaped operator

`shaped-a0.4-p0` is reported on every op beside the two gated operators: does its smaller removal still clear the removal gate on the new table?

`shaped-a0.4-p0` keeps between 4% and 43% on the removal lines, outside the removal gate on `hue-hsv`, `sat-hsv`, `value-hsv`: its smaller removal does not clear the gate on the ops the projection misses either.

Discussion

Most of what the handover was meant to settle, it settled, and the decision rule still says no. Eleven ops with drawn answers are learnable, and the anchored model learns them as well as the control does (H1). *Red* lands where the recipe put it at six ops, with the same margin, grading, and contrast (H2). The separate readout keeps the axis off the syntax embeddings on this grammar too (H4), and at twenty seeds against twenty the whole-line label costs no selectivity we can resolve (H5). The selectivity read itself is the cleanest we have had: a non-red `mix` deficit of 0.004, against 0.040 at the reference.

Two reads missed, but both are narrower than "a miss" sounds. Retention, which the decision rule reports and does not count, missed on 1 seed of 20, at 0.77 against the 0.8 gate. The seed mean, 0.88, is under `handover-slot`'s 0.96 and `handover-tied`'s 0.96 (and ex-2.2.3's 0.99), and neither reference has a seed under the line. That is consistent with the label and the readout costing a little retention each. But the evidence is one seed in twenty, with no band on retention, and that does not separate a real cost from one seed's luck. The schedule is the other suspect: the anneal here runs over three times as many steps as at the reference, so the margin has longer to drift after its peak, and a later or shorter anneal is the first thing to try. Removal, which the rule does count, missed on the three order-sensitive ops. On `sat-hsv` and `value-hsv` the lines with red at op1 lose almost everything (11% and 6% kept), and the lines with red at op2 keep most of it (65% and 71%). Red at op2 is the slot where the red operand supplies only its saturation, or only its value. We had expected removal in both slots, so this is a reading made after the fact: the axis carries the *redness* of a color, and a red color's saturation and value are read from somewhere else. If that is right, it is what an anchored concept should look like, and the removal lines are the thing to fix: they were chosen by how far the true answer moves when the red operand loses its red, and on these two ops that counts lines where what moved was the value, which the model never needed *red* for. `hue-hsv` is the case that reading does not cover: red at op1 supplies the hue there, and still keeps 31%.

So the grammar and the recipe are not adopted as they stand, but it's close. Before this handover could be adopted, the removal lines for the three order-sensitive ops would need to be the lines whose answer takes the red operand's hue, so that the gate asks the model to have lost *red* rather than the value of a red color; `hue-hsv` would have to clear that gate on its own terms; and retention would need a read that twenty seeds can resolve, or a look at why the margin drifts after its peak on this arm, which the trajectories can show. Whether those changes are made, and the handover re-run, is a question for the next round. One observation goes with the readout: the untied table cleaned every syntax word except `⏎`, whose embedding row still carries 0.17 on `handover`. Where that comes from, and whether it matters, is for that round too.

Method

The table

op	rule (0..15 scale, snapped to the grid)	commutative	in A+ as
`mix`	(x + y) / 2	yes	kept
`screen`	15 − (15 − x)(15 − y) / 15	yes	kept
`multiply`	x · y / 15	yes	kept
`lighten`	max(x, y)	yes	kept
`darken`	min(x, y)	yes	kept
`difference`	|x − y|	yes	added
`exclusion`	x + y − 2xy / 15	yes	added
`hsvmix`	mean in HSV: circular mean of H, means of S and V	99%	added
`hue-hsv`	op2's H at op1's S and V	1%	order-sensitive
`sat-hsv`	op2's S at op1's H and V	1%	order-sensitive
`value-hsv`	op2's V at op1's H and S	1%	order-sensitive

Every rule is computed on the 0..15 scale and snapped to the grid. Where it lands between levels, the corpus draws the answer (*stochastic rounding*, ex-2.2.5). The three order-sensitive ops take one HSV attribute from op2 and the other two from op1, so each agrees with its own reverse on under 2% of pairs. Their reads are reported as a subset.

Op-relevance under A+

For a line that uses the anchored op, how many other ops in the table give the same answer. This is the stimulus side of the per-line predictions in the anchored-op experiments, counted over ordered pairs. Widening the table makes `mix` less distinctive (ex-2.2.4 read 95% alone at six ops), which is good because the later predictions need more than one level to read.

anchored op	alone	1 other agrees	2	3+
`mix`	63%	30%	4%	3%
`hsvmix`	59%	32%	6%	3%

The removal lines

Per op, on its probe set: the red lines (dose ≥ 0.8, where dose is the redness of the redder operand) and, of those, the lines on which setting the R channel of the red operand to zero moves the true answer by at least 0.4 in the unit cube. The removal read in H3 is scored on the second column. The last column counts the non-red lines (dose ≤ 0.2) whose true answer is red; those are in the selectivity read and also counted on their own.

op	red probe lines	removal lines	non-red lines with a red answer
`mix`	365	365 (100%)	0
`screen`	405	278 (69%)	0
`multiply`	405	265 (65%)	0
`lighten`	405	268 (66%)	0
`darken`	405	265 (65%)	0
`difference`	405	277 (68%)	11
`exclusion`	405	267 (66%)	3
`hsvmix`	405	391 (97%)	0
`hue-hsv`	405	312 (77%)	19
`sat-hsv`	405	252 (62%)	27
`value-hsv`	405	397 (98%)	15

Replacing the red operand with a mid-gray instead of zeroing its R channel was considered. On `mix` it counts far fewer lines (a gray partner moves the mix less than a dark one does), and elsewhere it changes the counts without changing which ops need *red*, so the to-zero rule stays.

For the order-sensitive ops, the both-slot probe set splits by where the red operand sits. The one place the rule under-counts is a red op2 under `hue-hsv` or `sat-hsv`. Zeroing the R of a pure red gives black, which HSV reads as hue 0 (red) at no saturation: under `hue-hsv` the answer keeps its hue, and under `sat-hsv` op1 only loses its saturation, which moves it far only when it was saturated. Setting R to one grid level instead of zero does not help. The operand is then a very dark red at full saturation, so `hue-hsv` moves as little as before and `sat-hsv` stops moving at all (0 lines counted against 50 at zero). There is no hue a color should have once its red is gone, so the lines the rule drops are reported (in the red-at-op2 count) and not gated. The red-at-op2 lines that do move far stay in the gate: the removal read pools both slots, and the exploratory section splits it.

op	red at op1	removal	red at op2	removal
`hue-hsv`	186	186 (100%)	185	104 (56%)
`sat-hsv`	186	186 (100%)	185	50 (27%)
`value-hsv`	186	178 (96%)	185	185 (100%)

Each slot is counted on the walk that puts the palette color in it. The scored probe set pools both walks, so the per-slot counts in the exploratory table are about twice these.

The reference op

Every gated read is on `mix`, and `hsvmix` is reported beside it as the op a later experiment might promote. This is what the switch would change, on each op's probe set as ex-2.2.3 draws it.

op	probe lines on the grid	expected exact match ceiling	red lines that are removal lines	mean to-zero move	red-answer lines	alone
`mix`	100%	1.00	100%	0.51	0	63%
`hsvmix`	2%	0.39	97%	0.64	0	59%

`hsvmix` is the better op for the removal read: nearly every red line is a removal line, and the answer moves further when the red operand loses its red, since a hue mean loses the red hue outright where a channel mean halves it. It is also a little less distinctive, which the anchored-op predictions want. Its placement reads (H2) would look much like `mix`'s, since the label reads the colors in the line rather than the op word.

What it costs is the probe set. `mix` has 27 partners per color on which its answer lands on the grid without rounding, its probe lines are those, and so a clean model can match every answer and a deficit is a count of lines lost. `hsvmix` lands on the grid on 2% of pairs, about 4 partners per color, so its probe set is the shared random draw, and the best any model can do on it is an expected exact match of about 0.39. On that base the removal gate reads on a clean value near 0.4 rather than 1, and the 0.05 selectivity gate is an eighth of the clean value rather than a twentieth; both would need re-setting, and neither could be read against the reference's numbers. Had the program used `hsvmix` from the start, every exact-match read before ex-2.2.5 would have been on answers that round on 98% of lines, which is the bias that experiment found and fixed; D2.1 and ex-2.2.3 could read removal and selectivity as counts because `mix`'s probe set never rounds. The switch is open now that answers are drawn, at the price of a noisier deficit, and it is a later experiment's call, made with this experiment's `hsvmix` rows in hand.

The corpus, the probes, the labellers

300,000 lines at seed 0, ops drawn uniformly, with 20% of the distinct unordered pairs of each op held out. A held-out pair is out in both orders, for every op, so the held-out share of an op's lines is the same 20% whether or not the op reads operand order. Probe sets follow ex-2.2.3: `mix` on its 27 on-grid partners per color, and every other op on 27 partners per color drawn once at seed 0 and shared across ops. The order-sensitive subset also walks every color as op2.

The two labellers are *either slot, prompt span* (each operand draws at $redness^8 \times 0.04$, and the pull covers op1, op, op2, `=`) and *whole line* (the answer draws at its redness rate too, and the pull covers all six positions). The anchor term is the per-line mellowmax over the pulled span with a conserved per-line budget, so a wider span changes where the pull can land but not how strong it is.

Before the freeze

One seed of `control` trains first, and its expected exact match per op is recorded here, so that the gate in H1 is read against a control that learned the grammar. The bar: on every kept and added op, expected exact match within 0.05 of the ceiling the drawn answers allow on that op (ex-2.2.5 saw the six-op control within 0.04 on the ops that round). A lower value on the order-sensitive subset is recorded and does not stop the run, since it says something about the grammar rather than about the anchor; a miss on a commutative op does stop it, and sends the corpus size and epoch count back for a look. Two pieces of code land with the DAG: the holdout draw, now keyed on the position of the op in this table rather than in the table of ex-2.2.3 (`sca.data.ops.holdout`), and a probe draw that walks both slots for the order-sensitive subset (`probe_partners`). Neither changes a number in the design.

What the look found. The first control seed, at the draft's point (100k lines, 50 epochs, 1,650 steps), missed the bar on six of the eight commutative ops, with `difference` 0.29 from its ceiling. Seen and held-out lines scored alike on every op, so this was under-training rather than a coverage problem. More epochs over the same lines fixed all but two: at 100 and 150 epochs `difference` stayed 0.07 to 0.09 short and `hsvmix` 0.07 to 0.09 short, and the `hsvmix` NLL rose with every extra pass, which says the model was memorising the drawn labels rather than learning the distribution behind them. Deeper models (six and eight layers, at 100 epochs) moved neither op. A larger corpus did: at 300k lines and 50 epochs (4,950 steps) every kept and added op is within the bar, with `hsvmix` 0.04 short and `difference` 0.03; more epochs over the same 300k lines pushed `hsvmix` back out. So the frozen point is 300k lines at 50 epochs: three times the reference's steps, from three times the lines rather than more passes. Every point looked at is in the table below, and the frozen one is in bold.

op	group	ceiling	100k × 50 ep, L4: EEM (gap ↓)	200k × 33 ep, L4: EEM (gap ↓)	300k × 33 ep, L4: EEM (gap ↓)	200k × 50 ep, L4: EEM (gap ↓)	100k × 100 ep, L4: EEM (gap ↓)	100k × 100 ep, L6: EEM (gap ↓)	100k × 100 ep, L8: EEM (gap ↓)	100k × 150 ep, L4: EEM (gap ↓)
`mix`	kept	0.443	0.415 (**0.028**)	0.422 (**0.022**)	0.426 (**0.017**)	0.424 (**0.019**)	0.423 (**0.020**)	0.425 (**0.018**)	0.420 (**0.024**)	0.428 (**0.016**)
`screen`	kept	0.552	0.535 (**0.017**)	0.544 (**0.008**)	0.553 (**-0.001**)	0.555 (**-0.004**)	0.552 (**-0.001**)	0.536 (**0.016**)	0.546 (**0.006**)	0.546 (**0.006**)
`multiply`	kept	0.538	0.504 (**0.034**)	0.526 (**0.011**)	0.535 (**0.003**)	0.532 (**0.006**)	0.525 (**0.013**)	0.528 (**0.009**)	0.525 (**0.013**)	0.526 (**0.012**)
`lighten`	kept	1.000	0.938 (0.062)	0.967 (**0.033**)	0.990 (**0.010**)	0.986 (**0.014**)	0.977 (**0.023**)	0.982 (**0.018**)	0.979 (**0.021**)	0.984 (**0.016**)
`darken`	kept	1.000	0.918 (0.082)	0.966 (**0.034**)	0.991 (**0.009**)	0.989 (**0.011**)	0.983 (**0.017**)	0.976 (**0.024**)	0.983 (**0.017**)	0.985 (**0.015**)
`difference`	added	1.000	0.714 (0.286)	0.853 (0.147)	0.962 (**0.038**)	0.963 (**0.037**)	0.913 (0.087)	0.898 (0.102)	0.915 (0.085)	0.929 (0.071)
`exclusion`	added	0.568	0.518 (0.051)	0.551 (**0.018**)	0.556 (**0.012**)	0.557 (**0.011**)	0.553 (**0.016**)	0.550 (**0.018**)	0.547 (**0.022**)	0.562 (**0.007**)
`hsvmix`	added	0.390	0.309 (0.081)	0.331 (0.059)	0.336 (0.054)	0.328 (0.062)	0.318 (0.072)	0.313 (0.077)	0.331 (0.058)	0.303 (0.087)
`hue-hsv`	order-sensitive	0.825	0.806 (0.019)	0.805 (0.020)	0.821 (0.004)	0.821 (0.004)	0.807 (0.017)	0.804 (0.020)	0.809 (0.015)	0.806 (0.019)
`sat-hsv`	order-sensitive	0.693	0.676 (0.017)	0.673 (0.021)	0.691 (0.002)	0.695 (-0.002)	0.680 (0.014)	0.690 (0.003)	0.685 (0.009)	0.681 (0.012)
`value-hsv`	order-sensitive	0.716	0.694 (0.022)	0.690 (0.026)	0.712 (0.004)	0.711 (0.005)	0.707 (0.009)	0.705 (0.011)	0.706 (0.011)	0.707 (0.010)

- **Clears the bar** (every kept and added op within 0.05 of its ceiling): 300k × 50 ep, L4 (4,950 steps, the frozen point), widest gap `hsvmix` at 0.039; 300k × 100 ep, L4 (9,900 steps), widest gap `hsvmix` at 0.050.
- **Misses**: 100k × 50 ep, L4 (1,650 steps) on `lighten`, `darken`, `difference`, `exclusion`, `hsvmix`; 200k × 33 ep, L4 (2,178 steps) on `difference`, `hsvmix`; 300k × 33 ep, L4 (3,267 steps) on `hsvmix`; 200k × 50 ep, L4 (3,300 steps) on `hsvmix`; 100k × 100 ep, L4 (3,300 steps) on `difference`, `hsvmix`; 100k × 100 ep, L6 (3,300 steps) on `difference`, `hsvmix`; 100k × 100 ep, L8 (3,300 steps) on `difference`, `hsvmix`; 100k × 150 ep, L4 (4,950 steps) on `difference`, `hsvmix`; 200k × 100 ep, L4 (6,600 steps) on `hsvmix`.
- The order-sensitive ops are within 0.03 of their ceiling at every point.

Budget

59 runs of 4,950 steps at d64-L4, plus scoring under three operators on 11 probe sets. That is three times the steps per run of ex-2.2.3's short runs, at about the same run count; on an L4 a run takes about three minutes.

---

1. The prereg draft called it `handover-wide`: at the draft's 100k lines, holding lines per op at the six-op count meant a larger corpus. The calibration look below made the main corpus the larger one. ↩