Ex 2.2.11: the handover re-run

> Ex-2.2.9 enabled every change from the scouting round at once, and we did not adopt the result. Two gates missed. One was removal, on the three ops that take one HSV attribute from their second operand. The other was a single seed that held the anchor a little less well by the end. Ex-2.2.10 traced both misses to how we were measuring.
>
> This is the same experiment at fresh seeds, with three measurements fixed in advance: removal lines picked by hue, retention measured across the anneal, and the op1 alignment reported beside its references rather than gated. *Red* lands, stays through the anneal, and comes out cleanly on ten of the eleven ops. On `hue-hsv` it does not come out far enough, so the handover is not adopted as it stands.

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 `value-hsv` at -0.004 (band 0.003).
- Does *red* land, and stay through the anneal? (H2) — **gate cleared.** `handover` clears margin, grading, and contrast (margin 0.429, grading r^2 0.884, contrast 0.842). Retention across the anneal is 1.00 (lowest seed 0.98), so every seed holds through the anneal.
- Can we take *red* out on the lines that need its hue? (H3) — **gate missed.** Removal: `handover` keeps more than 20% of its clean expected exact match on the removal lines of `hue-hsv` (24%), and at most 18% on the other ops. Selectivity holds: the non-red `mix` deficit is 0.005 against the 0.05 gate (band 0.013).
- Do the two reference comparisons hold again? (H4) — **held.** The readout comparison holds: the syntax embeddings carry less of the axis on `handover` (0.032) than on `handover-tied` (0.168; band 0.008), and `handover`'s readout rows carry 0.109. The label comparison holds: the non-red `mix` deficits are 0.005 on `handover` and 0.005 on `handover-slot` (band 0.013), with 0 and 0 seeds above 0.07.
- Decision — **not adopted.**

> **How to read this draft**
>
> This is a preregistration. We wrote the conditions, the gates, and the decision rule before any run and froze them at commit `903cae5`. Everything after that commit is either results filled into the frozen sections or exploratory. Each hypothesis section opens with what we expect and the one number we will look at, and the results go into that 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

We want a setup that the later D2.2 experiments can build on. It has two parts: the grammar, meaning which color operations the model learns, and the recipe, meaning how the model is trained and how *red* is anchored on one axis of the residual stream.[1] Ex-2.2.9 was the handover to that setup: it combined the four changes the scouting round had tried one at a time, and asked whether *red* still lands on its axis and still comes out cleanly.

Most of it held. The model learned the eleven ops as well as an un-anchored model did, *red* landed with the same margin as before, and neither the separate readout nor the whole-line label cost anything we could resolve. Two measurements missed their gates, so the rule said no.

Ex-2.2.10 then went back to the stored runs and traced both misses to how we measured. The removal measurement counted a line as needing *red* when zeroing the R channel of the red operand moved its answer far. But zeroing R on pure red gives black, which has no saturation or value either, so the rule also swept in lines whose answer takes only the saturation or value of red.

The projection leaves those lines alone, because on the red operand it acts like a change of hue. On the lines whose answer does need the hue, every seed already cleared the gate.

Retention was the second miss: that measurement divided the final alignment by the peak alignment over the run. On the candidate, the peak is the high point of a noisy plateau reached thirty epochs before the anneal, and across the anneal itself nothing is lost. Separately, the op1 alignment rises under either half of the handover, so its old reference belongs to the old grammar.

A measurement chosen after looking at the data cannot then score that data. So this is ex-2.2.9 again at seeds it never trained, with the three measurements fixed here first. We re-run rather than rescore because the claim we want is the preregistered one.

Conditions

Everything about training is as ex-2.2.9 froze it: table A+ with its 11 ops, 300,000 lines with answers drawn stochastically, one fifth of the pairs of each op held out, and the recipe adopted in ex-2.2.3 (λ_a = 0.1, annealed over the last tenth of training, τ = 0.1, 50 epochs). The corpus is the same one at the same seed, so the held-out lines are the same. Every model is fresh: condition seed *i* trains at model seed 100 + *i*, so we score no checkpoint that ex-2.2.10 looked at.

condition	seeds	what it is	role
`control`	5	un-anchored	reference
`handover`	20	the recipe, untied readout, whole-line labeller	candidate
`handover-slot`	20	as handover, with ex-2.2.3's either-slot labeller	reference
`handover-tied`	9	as handover, with the tied readout	reference

`handover` has everything enabled. It is the one candidate, and every gate is scored on it alone.

`handover-slot` puts the label back on the two operands only, as ex-2.2.3 had it. **`handover-tied`** puts the readout back on the shared table. In ex-2.2.9 each of them undid about half of the rise in op1 alignment, which is why both are the references for that line now. Each is also the reference for one of the comparisons in H4, as before. This time we score both under the removal operators, for the saturation-and-value measurement in H3.

`control` has no anchor, and sets the task bar for H1.

`handover-narrow` does not return. Its question (lines per op) was exploratory, and ex-2.2.9 answered it.

The removal operators are the ones from ex-2.2.9. **`projection`** takes the axis out at every slice and position, and is the operator the gates are scored under. **`operands`** does the same at the two operand positions only, as the selective reference. **`shaped-a0.4-p0`** leaves states below alignment 0.4 alone, and is reported without a gate.

Glossary

We use the vocabulary of ex-2.2.9, and its glossary has every term.

Three terms change meaning here, and one is new.

Removal lines

The red lines whose true answer moves by at least 0.4 in the unit cube when we permute the channels of the red operand. Permuting the channels keeps the saturation and value of a color and changes its hue, so these are the lines whose answer needs the *hue* of red. Ex-2.2.9 zeroed the R channel instead, which also removes the saturation and value. Counts per op are in the method.

Saturation-and-value lines

The red lines the hue rule sets aside. Their answer holds still under every permutation of the red operand, so it takes only the saturation or value of red, or nothing from red at all. On the three order-sensitive ops, that is one of the two slots. We report these without a gate.

Retention

Whether the placement holds across the anneal of the anchor weight. It is the final alignment as a share of the alignment at the start of the anneal. Ex-2.2.9 divided by the peak over the run instead.

Level

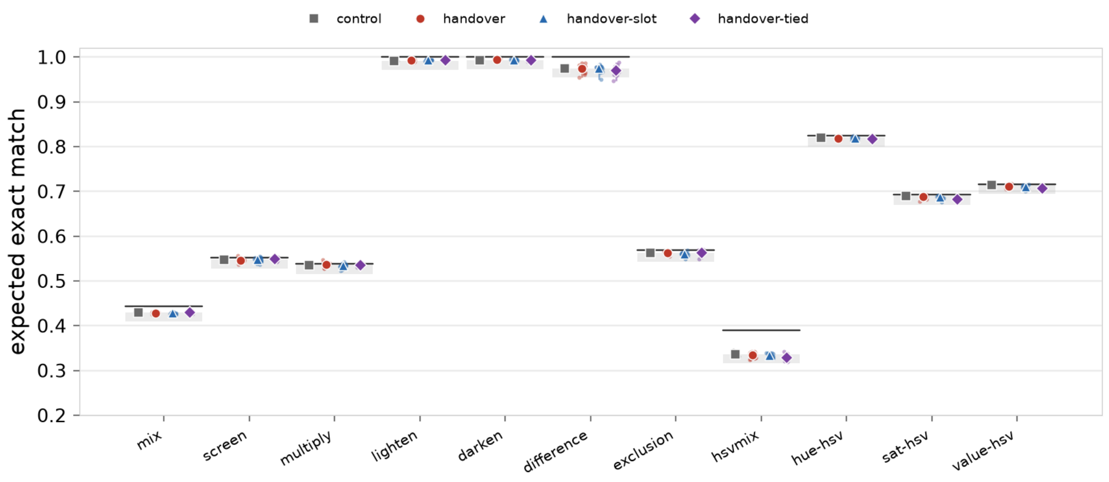
The alignment at the end of training, on the `mix` lines. Every later measurement is taken on it, and it is the number that the drift before the anneal moves. Level, retention, and the trajectory figure read the alignment the training loop records, on one probe line per color; the gated margin is the same statistic read after training on the full `mix` walk. The two sit on different scales, so a level near 0.7 and a margin near 0.4 describe one run.

Does the model still learn the task? (H1)

✓ Pass

What we expect. The anchored model learns the eleven ops as well as the un-anchored one does, as it did in ex-2.2.9. The number we look at, for each op, is the `handover` seed-mean expected exact match on the held-out lines, against the same number on `control`. Gate: within 0.02 on every op, partial to 0.05. We report a gap smaller than a band as unresolved. What would change our mind: a gap over 0.05 on some op, which at fresh seeds would say the pass in ex-2.2.9 was luck.

What we saw. The anchored model learns the task about as well as the un-anchored one, as it did at the old seeds. Across the 11 ops the `handover` seed mean sits between -0.004 (`value-hsv`) and +0.002 (`multiply`) of the control's; in ex-2.2.9 the widest gap was -0.006 (`hsvmix`). On the three order-sensitive ops the differences are -0.000, -0.001, -0.004.



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, which is 1 on the ops that never round.*

op	ceiling	control	`handover` (**Δ**) ↑	`handover-slot` (**Δ**) ↑	`handover-tied` (**Δ**) ↑	band
`mix`	0.44	0.429	**0.428 (-0.001)**	0.428 (-0.001)	0.430 (+0.001)	0.002
`screen`	0.55	0.547	**0.546 (-0.001)**	0.547 (-0.000)	0.548 (+0.001)	0.004
`multiply`	0.54	0.534	**0.536 (+0.002)**	0.534 (-0.000)	0.534 (-0.000)	0.004
`lighten`	1.00	0.991	**0.993 (+0.002)**	0.992 (+0.001)	0.992 (+0.001)	0.003
`darken`	1.00	0.993	**0.994 (+0.001)**	0.993 (-0.000)	0.994 (+0.001)	0.001
`difference`	1.00	0.975	**0.975 (-0.000)**	0.974 (-0.001)	0.970 (-0.005)	0.008
`exclusion`	0.57	0.563	**0.563 (-0.000)**	0.560 (-0.003)	0.563 (-0.000)	0.003
`hsvmix`	0.39	0.336	**0.334 (-0.003)**	0.334 (-0.003)	0.328 (-0.008)	0.006
`hue-hsv`	0.82	0.819	**0.819 (-0.000)**	0.819 (-0.001)	0.817 (-0.002)	0.003
`sat-hsv`	0.69	0.690	**0.688 (-0.001)**	0.686 (-0.003)	0.682 (-0.008)	0.004
`value-hsv`	0.72	0.714	**0.710 (-0.004)**	0.710 (-0.004)	0.707 (-0.007)	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.*

> **Pass**
>
> On every op the `handover` seed mean is within 0.02 of the control's. The largest shortfall is `value-hsv` at -0.004 (band 0.003).

Does *red* land, and stay through the anneal? (H2)

✓ Pass

What we expect. *Red* should end up on its axis as clearly as it did in ex-2.2.9, and easing off the anchor weight at the end of training should not let it slip. Everything here is measured on the `mix` lines of `handover`, under its own labeller.

Three of the numbers are gates, the same three ex-2.2.9 used. Each is something the anchor weight pushes on during training. So clearing them says the recipe still does its job on this grammar, and nothing about how.

- *Margin.* How far the pull lifts *red* onto the axis above the other colors. Seed-mean m_line at least 80% of what ex-2.1.10 measured (0.4202); partial from 60%.
- *Grading.* Whether redder colors sit further along the axis, in order. Grading r^2 at least 90% of what ex-2.1.11 measured (0.782).
- *Contrast.* Whether the pull follows the red operand rather than a fixed position. At least 0.2; partial from 0.1.

Four more numbers have no gate. We print each beside its references, and say here what we expect of it.

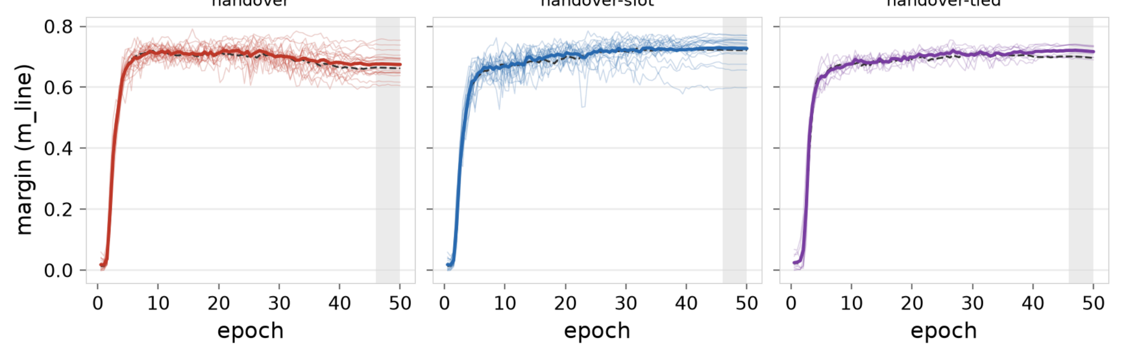
- *Retention.* Does the placement survive the anneal? Every run whose alignment at the start of the anneal reaches 0.2 should end at 0.8 of that value. This is the measurement we changed. Ex-2.2.10 took it this way on the runs from ex-2.2.9 and got 0.99 and above on every condition, so we expect every seed to clear it. That is why it is a line and not a gate: the method already predicts the answer, so the answer cannot inform the decision. A seed under the line would still be worth stopping for, since it would say the anneal costs something after all, and that the old measurement had been right for the wrong reason.
- *Level.* Where the alignment ends up. We report the final alignment, seed mean and range, beside `handover-slot`, `handover-tied`, and the point adopted in ex-2.2.3. Ex-2.2.9 measured 0.66 against 0.72 and 0.70, and we expect the same ordering. The gap between them is the drift before the anneal. Its cause goes to the training-dynamics item; here we only say how large it is at fresh seeds.
- *$\bar{\alpha}$ at op1.* How much of the axis the other colors pick up at the first operand. We report the mean alignment of the non-red colors at op1, beside the same two references and the point from ex-2.2.3. Ex-2.2.9 measured 0.28 against 0.18 and 0.16. We expect the same shape, with `handover` above both and each reference about halfway down. This was a gate at 0.1 through ex-2.2.9, where it missed. It becomes a gate again once we can name a mechanism (backlog).
- *The `⏎` embedding.* How much of the axis the end-of-line token carries. We report the axis component of the embedding of `⏎`, on every anchored condition. Ex-2.2.9 measured 0.17 on `handover` and about zero on `handover-slot`, and we expect that again.

We also check lead (at least 0.4) and latch (no run over 0.5 on op1) as before, to confirm the pull did what it should. These two are manipulation checks, as in ex-2.2.9: a miss would say the run was not the one we meant to score. Neither is part of the decision rule.

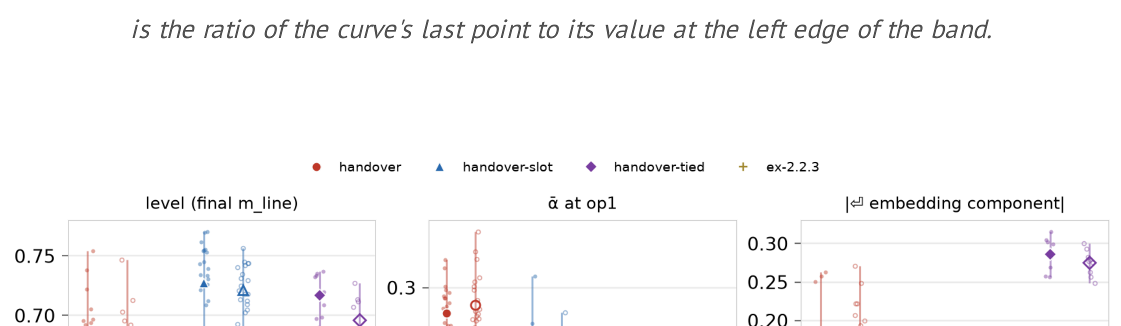
What we saw. *Red* lands on its axis as clearly as before. On the `mix` lines `handover` reaches a margin of 0.429, against 0.426 at ex-2.2.9's seeds and 0.392 at ex-2.2.3 (the gate is 0.336), a grading r^2 of 0.88, and a contrast of 0.84. Lead is 0.91 and the largest latch 0.01, so the pull did what it should.

Every one of the 20 qualifying seeds holds its alignment through the anneal: retention is 1.00 on the seed mean and 0.98 at the lowest seed, against the 0.8 line. The margin drifts down before the anneal, and the anneal itself costs nothing, which is what ex-2.2.10 saw on the old runs.

The three lines read as expected. Level is 0.67 on `handover`, 0.73 on `handover-slot`, 0.72 on `handover-tied`; ex-2.2.9 measured 0.66, 0.72, 0.70, so the ordering holds: the candidate ends lower than either reference, and the gap is the drift before the anneal. $\bar{\alpha}$ at op1 is 0.26 on `handover`, 0.18 on `handover-slot`, 0.14 on `handover-tied`, against 0.28, 0.18, 0.16 before and 0.08 at ex-2.2.3; `handover` sits above both references. The `⏎` embedding carries 0.16 on `handover`, 0.02 on `handover-slot`, 0.29 on `handover-tied` of the axis, against 0.17, 0.02, 0.28 before.



The margin on the `mix` lines through training, per anchored condition. *One panel per condition; each hairline is one seed, the bold line the seed mean, and the dashed line the seed mean of the same condition in ex-2.2.9. The shaded band is the anneal window, from the epoch at which the anchor weight first drops under its plateau to the end of training. Retention is the ratio of the curve's last point to its value at the left edge of the band.*



The three reported lines of H2, per anchored condition. *Filled marks are this run, open marks the same condition in ex-2.2.9, each as small dots per seed with the seed mean as the larger mark and the seed range as a bar. Left: the final margin on the `mix` lines. Middle: $\bar{\alpha}$ at op1, with ex-2.2.3's twenty seeds at the right. Right: the absolute axis component of the `⏎` embedding row.*

condition	margin ↑	grading r^2 ↑	contrast ↑	lead ↑	latch (max) ↓	retention ↑	level	$\bar{\alpha}$ at op1	`⏎` embedding
`control`	0.017 (-0.00–0.04)	0.030 (0.00–0.10)	0.044 (-0.05–0.13)	0.280 (0.02–0.51)	0.51	–	0.036 (0.01–0.08)	-0.027 (-0.14–0.03)	0.048 (0.03–0.09)
`handover`	**0.429 (0.40–0.46)**	**0.884 (0.84–0.91)**	**0.842 (0.82–0.86)**	**0.911 (0.89–0.92)**	**0.01**	**0.999 (0.98–1.01)**	0.675 (0.61–0.75)	0.264 (0.18–0.34)	0.162 (0.10–0.26)
`handover-slot`	0.398 (0.34–0.43)	0.872 (0.84–0.91)	0.866 (0.85–0.88)	0.890 (0.86–0.91)	0.01	0.996 (0.98–1.01)	0.727 (0.60–0.77)	0.181 (0.14–0.32)	0.021 (0.00–0.05)
`handover-tied`	0.428 (0.41–0.45)	0.893 (0.86–0.91)	0.857 (0.85–0.87)	0.894 (0.88–0.90)	0.01	0.995 (0.99–1.00)	0.717 (0.69–0.74)	0.137 (0.11–0.17)	0.286 (0.26–0.32)
ex-2.2.9 `handover`	0.426 (0.39–0.45)	0.887 (0.84–0.91)	0.838 (0.81–0.86)	0.909 (0.89–0.92)	0.00	0.998 (0.98–1.02)	0.662 (0.57–0.75)	0.276 (0.19–0.38)	0.174 (0.12–0.27)
ex-2.2.9 `handover-slot`	0.392 (0.34–0.42)	0.871 (0.84–0.90)	0.864 (0.85–0.87)	0.888 (0.87–0.91)	0.01	0.997 (0.99–1.01)	0.721 (0.64–0.76)	0.184 (0.14–0.26)	0.018 (0.00–0.04)
ex-2.2.9 `handover-tied`	0.423 (0.41–0.45)	0.894 (0.87–0.91)	0.855 (0.85–0.87)	0.894 (0.89–0.90)	0.01	0.994 (0.99–1.00)	0.696 (0.67–0.73)	0.162 (0.13–0.19)	0.275 (0.25–0.30)
ex-2.2.3 reference	0.392 (0.37–0.41)	0.886 (0.84–0.93)	0.868 (0.84–0.88)	0.826 (0.78–0.86)	0.02	–	–	0.083 (0.04–0.15)	–

Placement of red on the `mix` lines, per condition, with ex-2.2.9's runs and ex-2.2.3's point as reference rows. *Seed mean with the seed range in brackets. Margin, grading, and contrast are the gates; lead and latch are the manipulation checks (latch is the largest non-red softmin weight at op1 over the seeds). Retention is the final alignment over the alignment at the start of the anneal, on the seeds where that value reached 0.2; level is the final point of the trajectory; $\bar{\alpha}$ at op1 is the mean alignment of the non-red colors at the first operand; the last column is the absolute axis component of the `⏎` embedding row. Bold `handover` entries clear their gate or line. Ex-2.2.9's rows are read with this report's rules from its stored trajectories, so they are the before column of every read; ex-2.2.3 stored no trajectory, so three of its cells are empty.*

> **Pass**
>
> `handover` clears margin, grading, and contrast (margin 0.429, grading r^2 0.884, contrast 0.842). Retention across the anneal is 1.00 (lowest seed 0.98), so every seed holds through the anneal.

Can we take *red* out on the lines that need its hue? (H3)

✗ Miss

What we expect. With the removal lines chosen by hue, the projection takes *red* out on every op, and costs nothing on the lines that never had any *red* in them. Both gates are scored under `projection`, on `handover`.

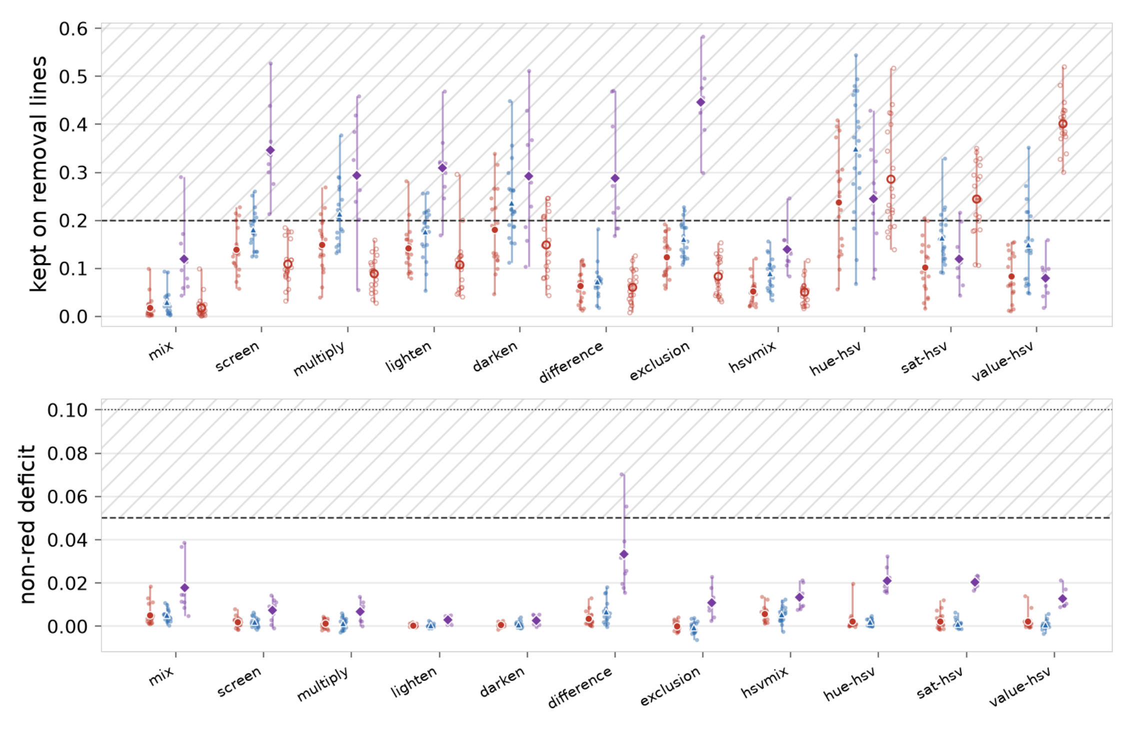
- *Removal.* On the removal lines of every op, the model keeps at most 20% of its clean expected exact match, seed mean. Rescoring the seeds of ex-2.2.9 this way, every seed cleared it on every op; the gate here is on fresh seeds. What would change our mind: an op above the line. If that op is one of the three HSV ops, the hue rule was not the whole story. If it is a channel-wise op, the rule change was beside the point.
- *Selectivity.* The seed-mean deficit on the non-red `mix` lines is at most 0.05; partial to 0.1, which is a reporting level. Ex-2.2.9 measured 0.02. Bands use the per-run spread of ex-2.2.8, `projection` on `recipe-short`, twenty seeds, per op, frozen as before. We report every other op beside `mix`, and count the red-answer lines on their own, as ex-2.2.9 did.

One more measurement has an expected direction but no gate.

Saturation and value. We take the kept share under `projection` on the saturation-and-value lines of the three order-sensitive ops, for `handover`, `handover-slot`, and `handover-tied`. On the probe sets these are 59% of the red lines on `hue-hsv`, 53% of the red lines on `sat-hsv`, 52% of the red lines on `value-hsv`. Those shares are counted from the true answers, so they come from the rule and the op, and from no model. A slot is an operand position, op1 or op2; on these three ops the answer depends on which slot the red color sits in, so the lines are split by it. Ex-2.2.10 saw `handover` keep about half on the two slots that take the saturation or value of red, and a fifth where the answer needs both at once, so the projection takes part of the saturation and value of a red color with it. If the two references keep the same shares, that cost belongs to the operator. If they keep more, the cost belongs to this checkpoint, and either the readout or the labeller is putting saturation and value onto the axis.

What we saw. Taking the axis out removes *red* on 10 of the 11 ops. On their removal lines, `handover` keeps between 2% (`mix`) and 18% (`darken`) of its clean expected exact match under `projection`. On `hue-hsv` it keeps 24% (seeds from 6% to 41%), above the 20% gate. `darken` is a near miss, within 3% of the gate; the same measurement is read on every op, and the channel-wise ops keep between 2% and 18%, so the near miss is the top of that spread rather than an op the rule treats differently. Both references keep more there: `darken` (24% on `handover-slot`, 29% on `handover-tied`). On the same runs, the old to-zero lines put `hue-hsv`, `sat-hsv`, `value-hsv` above the gate (29%, 25%, 40%). So the miss in ex-2.2.9 comes back at fresh seeds whenever the lines are picked the old way. Picking them by hue clears `sat-hsv`, `value-hsv` and leaves `hue-hsv` above the line.

On the non-red `mix` lines the deficit is 0.005 (seeds from 0.001 to 0.018), against 0.040 at the reference. The largest non-red deficit on any op is 0.006, on `hsvmix`.



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, with the removal lines chosen by hue (filled marks, one per condition) and, for `handover`, by ex-2.2.9's to-zero rule (open circles). The dashed line is the 20% gate and the hatched region above it the miss. Bottom: the deficit in expected exact match on each op's non-red lines, dashed at the 0.05 gate and dotted at the 0.1 partial level; the gate is read on `mix` only. Each small dot is one seed, the larger mark the seed mean, and the thin bar the seed range.*

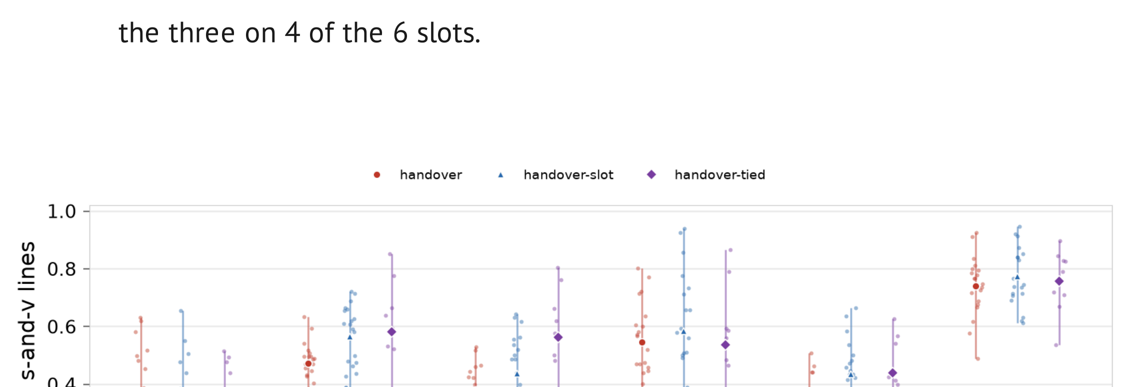
op	lines	handover ↓	handover-slot ↓	handover-tied ↓	handover, to-zero lines	operands ↓	shaped ↓
`mix`	365	**0.02**	0.03	0.12	0.02	0.02	0.03
`screen`	387	**0.14**	0.18	0.35	0.11	0.14	0.17
`multiply`	384	**0.15**	0.21	0.29	0.09	0.15	0.17
`lighten`	386	**0.14**	0.18	0.31	0.11	0.14	0.17
`darken`	382	**0.18**	0.24	0.29	0.15	0.18	0.20
`difference`	403	**0.06**	0.07	0.29	0.06	0.06	0.08
`exclusion`	390	**0.12**	0.16	0.45	0.08	0.13	0.15
`hsvmix`	401	**0.05**	0.09	0.14	0.05	0.05	0.07
`hue-hsv`	333	0.24	0.35	0.24	0.29	0.26	0.31
`sat-hsv`	377	**0.10**	0.16	0.12	0.25	0.11	0.13
`value-hsv`	389	**0.08**	0.15	0.08	0.40	0.09	0.11

Removal, per op: the share of clean expected exact match kept on the removal lines. *Seed means under `projection`, one column per anchored condition on the lines chosen by hue, then `handover` on ex-2.2.9's to-zero lines, and `handover` under the `operands` and `shaped-a0.4-p0` operators on the hue lines. Bold entries are inside the 20% gate, which is read on `handover`.*

op	handover ↓	handover-slot ↓	handover-tied ↓	band
`mix`	**0.005**	0.005	0.018	0.013
`screen`	0.002	0.002	0.007	0.006
`multiply`	0.001	0.001	0.007	0.007
`lighten`	0.001	0.000	0.003	0.003
`darken`	0.001	0.001	0.003	0.004
`difference`	0.004	0.007	0.033	–
`exclusion`	0.000	-0.000	0.011	–
`hsvmix`	0.006	0.006	0.013	–
`hue-hsv`	0.002	0.002	0.021	–
`sat-hsv`	0.002	0.001	0.020	–
`value-hsv`	0.002	0.001	0.013	–

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 the read resolves, from the per-run σ of ex-2.2.8, `projection` on `recipe-short`, twenty seeds, per op; the ops the six-op grammar did not have carry none.*

On the saturation-and-value lines (the figure and table below), `handover` keeps less than a reference on 3 of the 6 slots by more than the band; on the other slots no reference keeps more than `handover` by more than the band. The largest gap is on `sat-hsv` with red at op1: 0.18 under `handover-tied` (band 0.09). So part of what the projection takes from the saturation and value of a red color belongs to this checkpoint: the references hold more of it off the axis. The rest is the same on all three anchored checkpoints, so it belongs to the operator. The seed spread on `handover` is the narrowest of the three on 4 of the 6 slots.



The saturation-and-value lines under `projection`: the share of clean expected exact match kept, per slot and anchored condition. *The red lines of the three order-sensitive ops that the hue rule sets aside, split by the slot the red operand sits in; a slot with no such lines is left empty. Each small dot is one seed, the larger mark the seed mean, and the thin bar the seed range. No gate.*

lines	count	handover	handover-slot	handover-tied
`hue-hsv`, red at op1	406	0.33	0.34	0.35
`hue-hsv`, red at op2	71	0.47	0.57	0.58
`sat-hsv`, red at op1	29	0.38	0.44	0.56
`sat-hsv`, red at op2	404	0.55	0.58	0.54
`value-hsv`, red at op1	17	0.33	0.43	0.44
`value-hsv`, red at op2	404	0.74	0.77	0.76

The saturation-and-value lines: the share of clean expected exact match kept under `projection`. *The red lines of the three order-sensitive ops that the hue rule sets aside, by the slot the red operand sits in, with the count of such lines on the probe set. Seed means per anchored condition; no gate.*

> **Miss**
>
> Removal: `handover` keeps more than 20% of its clean expected exact match on the removal lines of `hue-hsv` (24%), and at most 18% on the other ops. Selectivity holds: the non-red `mix` deficit is 0.005 against the 0.05 gate (band 0.013).

Do the two reference comparisons hold again? (H4)

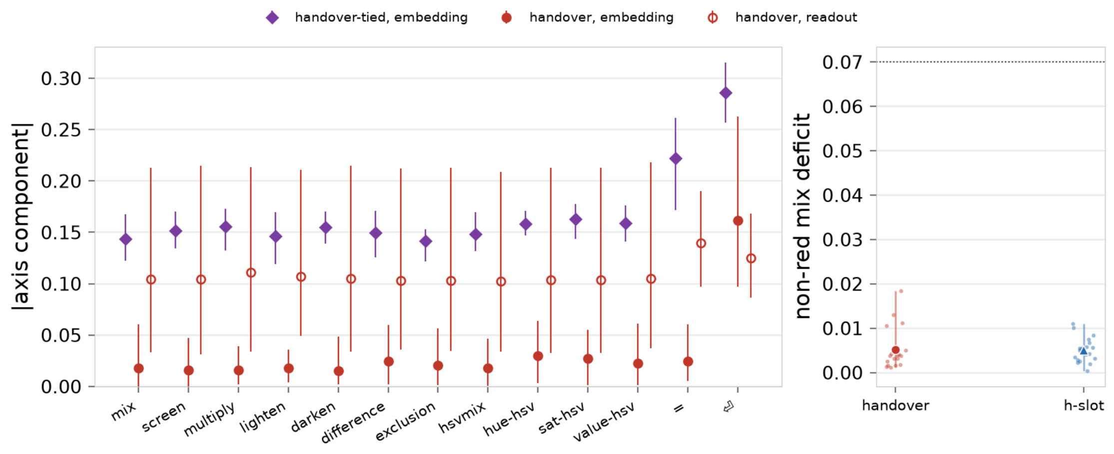
✓ Pass

What we expect. Ex-2.2.9 checked two things about the recipe on the side, and both held. The separate readout kept the axis off the syntax tokens, and the whole-line label cost no selectivity. We check both again at fresh seeds. Neither is a gate, so a miss is something to look into rather than a reason to stop.

- *The separate readout keeps the axis off the syntax tokens.* The axis component on the syntax embeddings (`=`, the op words, `⏎`) is lower on `handover` than on `handover-tied` by more than a band, with the component appearing on the readout table instead.
- *The whole-line label costs no selectivity.* The non-red `mix` deficit under `projection` differs between `handover` and `handover-slot` by less than a band, and at most two more seeds sit above 0.07 on `handover` than on `handover-slot`.

Both bands use the σ values ex-2.2.9 froze: pooled within-condition, over the two conditions compared for the embedding component, and ex-2.2.8, `projection` on `recipe-short`, twenty seeds, per op for the deficit.

What we saw. Both comparisons come out as they did before. Averaged over the op words, `=`, and `⏎`, the axis component of the embedding rows is 0.032 on `handover` and 0.168 on `handover-tied`, a gap over the band of 0.008, while `handover`'s readout rows carry 0.109: the component moved to the readout. As before, `⏎` is the one word the untied table leaves a component on, at 0.162 where no other syntax word reaches 0.03. Under `projection` the non-red `mix` deficit is 0.005 on `handover` and 0.005 on `handover-slot`, within a band of 0.013, and 0 seeds of `handover` sit above 0.07 against 0 of `handover-slot`.



The red axis on the syntax tokens, and what the whole-line label 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). 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, with the 0.07 tail level dotted.*

word	`handover-tied` embedding ↓	`handover` embedding ↓	`handover` readout
`mix`	0.144	0.018	0.105
`screen`	0.152	0.016	0.105
`multiply`	0.156	0.016	0.111
`lighten`	0.146	0.018	0.107
`darken`	0.155	0.015	0.105
`difference`	0.149	0.025	0.103
`exclusion`	0.142	0.021	0.103
`hsvmix`	0.149	0.018	0.103
`hue-hsv`	0.158	0.030	0.104
`sat-hsv`	0.163	0.027	0.104
`value-hsv`	0.159	0.023	0.105
`=`	0.222	0.025	0.140
`⏎`	0.286	0.162	0.125
all syntax words	0.168	0.032	0.109

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 band between the two conditions on the all-words mean is 0.008 (pooled within-condition, over the two conditions compared).*

> **Pass**
>
> The readout comparison holds: the syntax embeddings carry less of the axis on `handover` (0.032) than on `handover-tied` (0.168; band 0.008), and `handover`'s readout rows carry 0.109. The label comparison holds: the non-red `mix` deficits are 0.005 on `handover` and 0.005 on `handover-slot` (band 0.013), with 0 and 0 seeds above 0.07.

Decision

✗ Miss

The rule, frozen before the run: `handover` is adopted, and becomes the grammar and recipe of record for the anchored-op experiments, if it clears H1, H2 (margin, grading, and contrast, all in full), and H3 in full on the removal lines chosen by hue; every partial band is a reporting level. Otherwise it is not adopted, and the report says which gate was missed and what the references say about which change is responsible. It misses H3 removal and clears the rest, so it is not adopted as it stands.

> **Miss**
>
> `handover` misses H3 removal, so **the handover is not adopted** as it stands. Neither reference clears removal on `hue-hsv`, which points at the table or the corpus.

The rule is ex-2.2.9's with the removal lines changed. Retention stays outside it, as it was there: with the new denominator ex-2.2.10 measured it at 0.99 on every condition, so it is expected to clear with room, and a gate the method can predict does not inform a decision. It is a line in H2, and a seed under 0.8 would still stop us.

Exploratory analyses

Not part of the decision. The two below were planned before the run. Anything we think of after seeing the data goes here too, marked as post hoc.

The finer hue rule

The permutation rule reaches six hues. A finer rule rotates the hue of the red operand in HSV in twelve steps, snaps each one to the grid, and asks the same question. The method counts how often the two rules disagree, per op. If they disagree on more than one red line in a hundred on some op, we keep the finer rule and say so under the method before the freeze. Below that rate, the choice of rule cannot move a kept share by more than a hundredth, which is finer than any gate can resolve. Outcome: the two rules disagree most on `difference`, on 0.5% of its red lines, so the permutation rule stands.

Checkpoints on the trajectory stride

The first three seeds of each anchored condition keep a checkpoint at every trajectory point. Nothing in this report uses them. The training-dynamics item needs them for a local learning coefficient estimate through the plateau and for the whole-geometry measurement at the same epochs, and this is the cheapest place to store them. Recorded.

The missed op at the old seeds (post hoc)

On `hue-hsv`, when ex-2.2.10 rescored the 20 seeds from ex-2.2.9, they kept 0.17 on these lines (seeds from 0.01 to 0.32). The 20 fresh seeds keep 0.24 (from 0.06 to 0.41). The two seed sets differ by 0.07, against a band of 0.06, and the 0.2 gate sits inside both spreads.

So the fresh seeds came in higher by about what twenty seeds resolve. A rescoring of old seeds is not a preregistered read, and this says how far the miss is from the noise and nothing more.

What the references keep on the removal lines (post hoc)

On the removal lines of every channel-wise op, `handover-tied` keeps more than `handover`, by 0.09 to 0.32. That gap is past the band on 8 of the 8 ops, and on the three HSV ops the two keep the same share to within a band. `handover-slot` keeps a little more than `handover` on every op, the most on `hue-hsv` (0.11), past the band on 9 of the 11.

So the separate readout is what makes removal clean on the channel-wise ops, and neither of the two changes is behind the share `hue-hsv` keeps.

The drift before the anneal (post hoc)

On `handover` the seed-mean alignment peaks near epoch 23 and loses 0.05 by the end of training. On `handover-slot` and `handover-tied` it peaks near epochs 45 and 47, in or beside the anneal window, and loses under 0.01.

So the drift ex-2.2.10 saw belongs to the candidate alone, even though each reference differs from it in only one change. The training-dynamics item has the checkpoints to look at it.

Discussion

The handover setup does what ex-2.2.9 said it does, with one exception we can now name. *Red* lands on its axis, stays there through the anneal, and the projection takes it out on ten of the eleven ops without touching the lines that never had *red* in them.

The exception is `hue-hsv`, the op whose answer takes its hue from the second operand. With the axis taken out, the model still keeps about a quarter of the accuracy it had on the lines that need a red hue there, a little more than the gate allows. We said in advance what that pattern would mean, and this is the first case: a miss on an HSV op under the hue rule. So the hue rule accounts for the ops it cleared, and on `hue-hsv` the miss has another cause.

On `hue-hsv` the stream holds some of the hue of the red operand somewhere other than the axis, which is one more reason to try anchoring *red* to a plane. `handover-tied` keeps the same share on that op and `handover-slot` keeps more, so neither the readout nor the labeller put the hue there. On this one op it seems to belong to the grammar and the recipe together.

The seed spread on that op is wide and the gate sits well inside it. At the seeds of ex-2.2.9, the same lines came in under the gate, by less than the read resolves. So the miss is a miss under the rule we froze, and it is also a small effect at the edge of what twenty seeds resolve. It is one op of eleven, on the lines whose answer needs the hue of a red at op2.

What comes next is a choice for the next preregistration. One route is to look at what the stream holds on `hue-hsv`: score the removal lines one at a time and ask what the surviving lines share, which the stored checkpoints and probes make cheap. The other is to run the anchored-op experiments on this setup, with `hue-hsv` recorded as the op where removal is partial, since nothing in D2.2 rests on that op alone.

Either way, two questions ex-2.2.9 left open are closed. Retention measured across the anneal loses nothing on any condition, so the schedule is not where the drift comes from. And the saturation-and-value cost of the projection belongs partly to this checkpoint, which the containment work could take up.

Method

The removal lines

The table below gives, per op and on its probe set: the red lines (dose at least 0.8), the removal lines under the hue rule (some channel permutation of the red operand moves the true answer by at least 0.4), the same count under the to-zero rule of ex-2.2.9, and the saturation-and-value lines the hue rule sets aside. The order-sensitive ops walk every color through both slots, so we split their removal lines by where the red operand sits.

op	red lines	removal (hue)	removal (to zero)	saturation-and-value	hue, red at op1	hue, red at op2
`mix`	365	365	365	0		
`screen`	405	387	278	18		
`multiply`	405	384	265	21		
`lighten`	405	386	268	19		
`darken`	405	382	265	23		
`difference`	405	403	277	2		
`exclusion`	405	390	267	15		
`hsvmix`	405	401	391	4		
`hue-hsv`	810	333	636	477	0	333
`sat-hsv`	810	377	522	433	377	0
`value-hsv`	810	389	793	421	389	0

On `mix` the two rules pick the same lines. On `hsvmix` they nearly agree, differing on 18 of the 405 red lines. On the six other channel-wise ops the hue rule counts more, because permuting a red operand moves two channels at once. An answer that zeroing R left alone, say a `screen` with a partner about as red, moves far under a permutation.

Here at least 94% of the red lines of every channel-wise op are removal lines. On the six ops where the rules part, the old rule took about two thirds. Ex-2.2.9 cleared the gate on those ops with room to spare, so we expect the wider set to clear it too.

On the three HSV ops the hue rule keeps the slot whose answer takes its hue from red and sets the other aside. Under `hue-hsv` that is red at op2; under `sat-hsv` and `value-hsv` it is red at op1. The old rule counted parts of both slots on each, which is where the miss in ex-2.2.9 came from.

The finer-rule check, computed before the run: the lines per op on which the twelve-step HSV rotation and the permutation rule disagree.

op	red lines	removal (permutation)	removal (rotation)	disagree	share
`mix`	365	365	365	0	0.00%
`screen`	405	387	387	0	0.00%
`multiply`	405	384	384	0	0.00%
`lighten`	405	386	386	0	0.00%
`darken`	405	382	382	0	0.00%
`difference`	405	403	405	2	0.49%
`exclusion`	405	390	390	0	0.00%
`hsvmix`	405	401	403	2	0.49%
`hue-hsv`	810	333	333	0	0.00%
`sat-hsv`	810	377	377	0	0.00%
`value-hsv`	810	389	389	0	0.00%

The rules part on at most 2 lines of an op (`difference`, 0.49% of its red lines), under the 1% tolerance on every op, so the permutation rule stands.

Retention

We take retention from the trajectories (50 points over training). The anneal starts at the first point after the peak of the anchor weight where the weight falls under 0.99 of that peak, and the alignment at the start of the anneal is the last point before that. Retention is then the final point divided by that value, per run, on the runs where that value reaches 0.2.

What is stored

We store what ex-2.2.9 stored: metrics, per-run arrays, trajectories, the probe set, and the end checkpoint of every run. Three things are new. Every anchored condition is scored under all three operators. The probe arrays hold the hue move and the to-zero move per line. And the first 3 seeds of each anchored condition keep a checkpoint at every trajectory point. The calibration of ex-2.2.9 still stands, since the corpus and the point are unchanged.

Budget

54 runs of 4,950 steps at d64-L4, five fewer than ex-2.2.9, plus scoring under three operators on 11 probe sets for 49 anchored runs. That is about three minutes a run on an L4, and the trajectory checkpoints add a few hundred MB of storage.

1. The *residual stream* is the running vector of activations that each layer of a transformer reads from and writes back to. ↩