Ex 2.1.10: A label that doesn't point

> Mellowmax pooling found the concept position on its own in ex-2.1.9, but every label was keyed on op1, so only one position ever carried the evidence. Now we label lines when either operand is *red*, and the pull has to find the right operand line by line. It does, with selectivity matching that of the oracle.

Findings

H1 (task cost) — holds. The largest `named_holdout` exact-match gap from the control across the five anchored conditions is 0.0065, against a gate of 0.02.

H2 (the pull selects the labelled operand) — holds. On the primary (`either-t100`): (a) at the embedding the leading role in each group is its own operand, at weight 0.77 (gate ≥ 0.4; uniform is 0.25), in all nine runs individually; (b) the deep-slice between-group op2 contrast is +0.85 (gate ≥ 0.2).

H3 (the operating point survives) — holds. On `either-t100`: containment $\bar{\alpha}$ at op1 = 0.050 (gate ≤ 0.1); retention 0.98 (gate ≥ 0.8, minimum across nine seeds); grading r^2 = 0.82 against 0.83 for the reference arm, a drop of 0.008 (gate ≤ 0.1).

H4 (selectivity without the pointer) — holds. Control-subtracted m_line of the primary is 102% of the value for the slot oracle (gate ≥ 75%).

The reference arm reproduced `pool-t100` from ex-2.1.9 through the new labelling code path, matching every carried statistic to three decimals (*Reproduction*, in the H2 section).

> **How to read this report**
>
> This report was preregistered: the method, hypotheses and decision thresholds were frozen before the experiment ran (commit `b36a590`). Results replaced the `TODO` placeholders in place; everything conceived after seeing the data is under *Exploratory analyses*, marked as post hoc.

Ex-2.1.9 took away the position oracle: the pooled (mellowmax) anchor term asks each labeled line to align *somewhere* in its span, and the pull put 0.77 of its weight on op1 at the embedding without being told. But the labels still pointed. A label is a per-visit Bernoulli draw: each time a line is visited in training, it *draws* a label with a small probability. In ex-2.1.9 that probability depended only on the redness of the first operand, so op1 was the only position whose evidence ever justified a label, and a pull that settled on op1 *by position* could never be wrong. RoPE never writes position into the stream, but causal attention gives each role a different view of the line (e.g. op1 attends only to itself), so from the first block on, the roles are distinguishable states a pull could latch.

Labels that don't point are a step toward M3, which will target natural text. There, labels arrive at the document level ("this review is negative", "this tweet is obscene"), since nobody marks which tokens carry the concept. A document-level label says the concept occurs, but not where. This experiment widens our labels by one step in that direction: either operand can draw the label, so a line can be labeled because of a red op2 while its op1 is blue. The localization then has to vary per-line: on op1-triggered lines the red state is at the first position of the line, on op2-triggered lines at the third, and a pool that commits to one position for the whole run would pull blue states onto the axis on the lines it got wrong.

Whether that failure would even show up in the scores is not obvious, because the span positions share token embeddings. Once red embeddings sit on the axis, a red op2 line has an aligned position for free: the labeller's slot-symmetry is satisfied by the lexicon before any per-line choice happens. It only really matters after attention: in ex-2.1.9, we found the pool's deep-slice choice is winner-take-all *per run*, committed by epoch 8. If that winner is global, the two label groups will show the same weight profile; if the pool localizes per line, a red op2 should carry the weight on the lines where it is the evidence (H2). The selectivity delivered to labeled lines is scored separately (H4), since the embedding give-away means the second can pass while the first fails.

One scope limit: our mixing operation can't make the answer redder than both operands, so "the concept is only at a position the labeller never sees" doesn't exist in this language. Two candidate positions, correlated with the answer, is as blind as this testbed gets; enough to test per-line localization; a concept at a position the labeller gave no hint of is out of reach here.

▶ **Glossary…**

Method

Everything follows ex-2.1.9 except the labelling rule: the word-level `v216` corpus from ex-2.1.3 (216 colors, one token each, d64-L4), the same seeds, the mellowmax-pooled anchor term, the anchor schedule, and the anti-subspace schedule fixed at ex-2.1.8's operating point (`end90-hold30`) in every anchored cell.

The labelling rule

Labels are as they have been since ex-2.1.6 (binary, drawn per visit, with probability graded in redness), except that now both operands draw. Each operand of a line draws independently at $\text{redness}^8 \times 0.04$, and the line is labeled when either draws. The per-slot rate is half of the op1 labeller's 0.08, which keeps the expected labeled-line rate where it was, up to the tiny both-draw overlap; the label-budget check below computes both. A labeled line's pulled positions are its prompt span, as before: the label never says which operand drew.

The answer position stays outside the span: the model has to *produce* the mix there, so pulling that state toward red would oppose the task on every labeled line whose answer isn't red, entangling the anchor with the output rather than with the concept. The span will have to widen eventually: in M3 the span is the whole document, answer-analogues included. That widening is its own variable, so it stays out of this experiment; a full-line arm (answer included, a position where the concept is diluted) is queued in todo-science as the intermediate step. Lines whose operands are *both* reddish exist too; their weight maps are on the exploratory list.

The `slot-oracle` condition is the exception that measures what the label withholds: it pulls only the operand(s) that actually drew, no pooling needed. And the `op1-labels` condition keeps the old labeller entirely (op1-keyed at 0.08), reproducing ex-2.1.9's `pool-t100` through the new code path.

One bookkeeping consequence: label draws are consumed per line whatever the anchor weight, so conditions sharing a labeller see identical batches and masks at a given seed. Conditions with *different* labellers consume different draws, so comparisons across the labelling rules carry corpus-draw noise on top of seed noise. The control runs the either-slot labeller, keeping it batch-identical with the primary.

Conditions

Six conditions, most run with three seeds each. The primary condition runs with nine seeds: the deep-slice mellowmax in ex-2.1.9 was one-hot per *run*, and we can measure the failure rate better with more seeds. 24 runs in all. Every anchored cell runs the anti-subspace term at the operating point; only the labelling rule and τ vary.

condition	λ_a	τ	labels	pulled positions	seeds
`lam0`	0	∞	either	prompt span, pooled	3
`op1-labels`	0.1	0.1	op1	prompt span, pooled	3
`either-t100` **(primary)**	0.1	0.1	either	prompt span, pooled	9
`either-t500`	0.1	0.5	either	prompt span, pooled	3
`either-t2500`	0.1	2.5	either	prompt span, pooled	3
`slot-oracle`	0.1	∞	either	the operand that drew	3

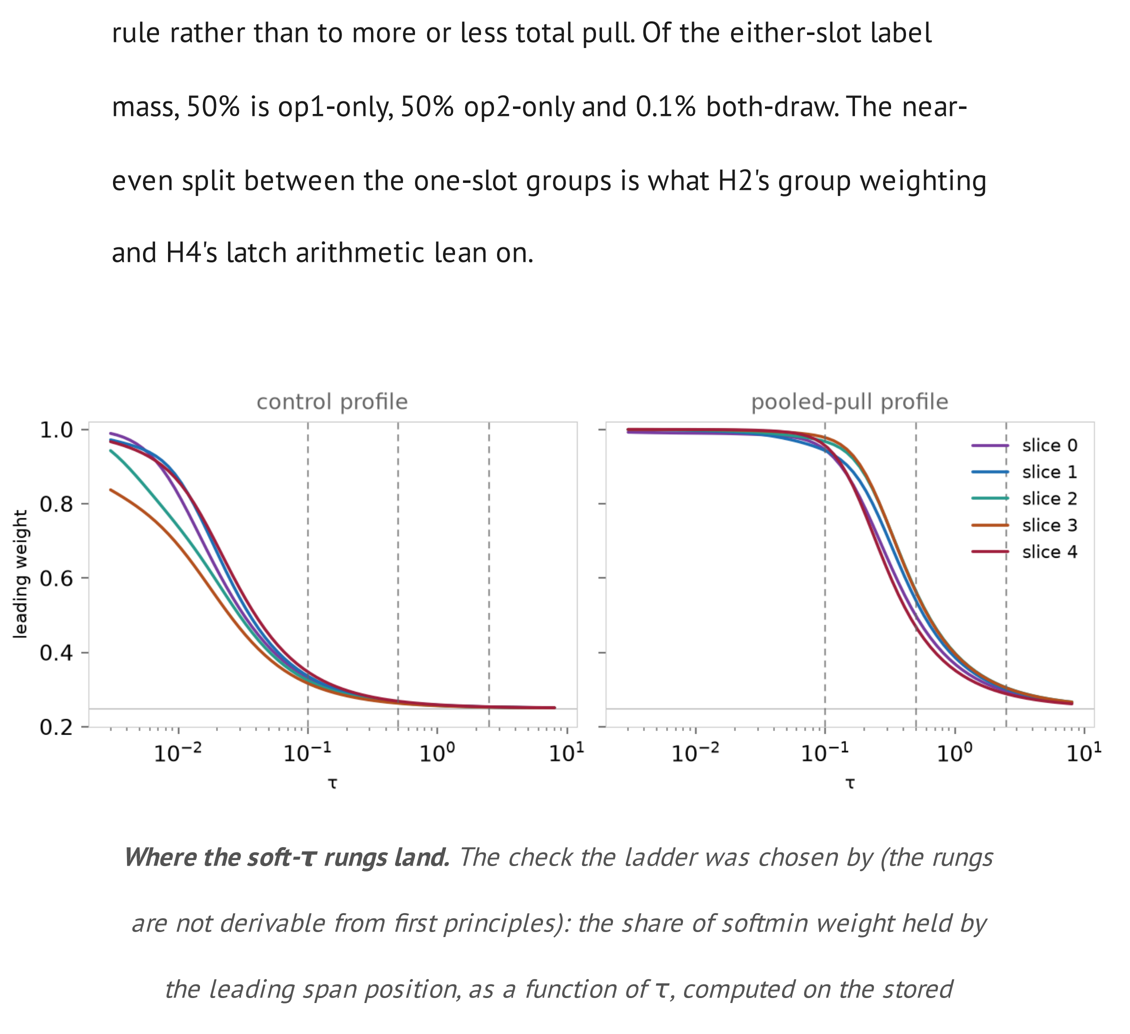
`either-t100` is the primary: the either-slot labeller at τ = 0.1. That temperature was the only rung ex-2.1.9 found graded in every seed (latch rate 0/3). The τ = 0.5 and 2.5 arms are softer,[1] reported but not gated.

`op1-labels` re-runs ex-2.1.9's `pool-t100` verbatim: the reproduction check for the new labelling code path, and the baseline the widened labels are judged against. `slot-oracle` shows what knowing the triggering slot is worth, measured through identical code: a reference rather than a strict ceiling, for the same reason ex-2.1.9's `op1-oracle` was.

Calibration

A few checks before anything runs, computed from the design constants and the published arrays from ex-2.1.9: the label budget, the placement of the soft-τ rungs, and the reproduction targets for the reference arm.

The label budget. Over the closed-pair set the corpus draws from, the op1 labeller labels 0.0012 of lines per visit and the either-slot labeller at the halved rate labels 0.0012, a 0.1% difference. The halving does keep the budget, so a margin change is attributable to the labelling rule rather than to more or less total pull. Of the either-slot label mass, 50% is op1-only, 50% op2-only and 0.1% both-draw. The near-even split between the one-slot groups is what H2's group weighting and H4's latch arithmetic lean on.



Where the soft-τ rungs land. *The check the ladder was chosen by (the rungs are not derivable from first principles): the share of softmin weight held by the leading span position, as a function of τ, computed on the stored alignments from ex-2.1.9 under the un-anchored control (**left**) and under `pool-t100` (**right**). Dashed verticals mark the rungs chosen here, and the curves say what each will do. τ = 0.1 keeps a clear leader on the trained profile; τ = 2.5 sits within a few hundredths of uniform (0.25) on both, so it is the tiny-bias arm and a τ = ∞ arm would add nothing; τ = 0.5 splits the gap. Had a rung landed on the flat shoulder next to another, we would have moved it.*

The ex-2.1.9 arrays give the reproduction targets for the `op1-labels` arm, as means of per-run statistics: m_span = 0.70, ᾱ_span = 0.24, ᾱ at op1 = 0.08. A miss there means the new labelling code path moved the numbers, and gets settled before any either-slot condition is read.

Measurements

Task evals, the readability profile, grading, and the trajectory instrument are from ex-2.1.9. Two measurements change and two are new, all on the same probe set: all 5832 ordered closed pairs, so every color appears both as op1 and as op2. (Terms like *line*, *role*, *slice* and *alignment* are pinned down in the Glossary above; alignment throughout is $\cos(h, \hat{v})$, a state's cosine to the anchor axis.)

Per-line alignment maps. The eval step now stores alignment per probe *line* (slices, lines, positions) rather than averaged into per-color rows. The per-color maps that earlier statistics used are recovered by averaging over partners, so the line-keyed statistics below are computed from those. Per-line softmin weights are stored the same way.[2]

m_line, the selectivity this experiment scores: m_span, generalized from colors to lines, to measure "how much closer does a line sit to the axis for being the kind of line the labeller flags".[3] The max over roles is applied to every condition and the control alike, so its maximum-of-noise inflation cancels in the control-subtracted comparisons, as before. The trajectory records m_line, so retention reads off it. m_span and m_{op1} are reported beside it for continuity.

Per-group weight profiles. The localization instrument. Probe lines are weighted into two groups by the labeller's own one-slot-only probabilities (G1 by P(op1 drew and op2 didn't), G2 by the reverse), so no redness threshold has to be invented, and both-red lines (where either answer is defensible) get almost no say. Within each group, $\bar{\pi}_G(\ell, t)$ is the group-weighted mean softmin weight per slice and role. Under per-line localization the two profiles should differ: G1 stepping at op1, G2 at op2. Under a global winner they are identical no matter what: if every line carries the same weight profile, the two group means, which differ only in how lines are weighted, both return that one profile.

Per-role drift in the trajectory. Ex-2.1.9 couldn't say *when* the `+` embedding climbed to 0.30 alignment because the trajectory recorded op1 drift only; it now records $\bar{\alpha}(\ell, t)$ over all span roles at each measurement, which is the data that question needs. Report-side only; nothing is gated on it.

Hypotheses

Gates and noise floors are from ex-2.1.6 through ex-2.1.9 wherever the statistic is unchanged; derivations for the new ones are in experiment.py, next to their constants.

Throughout, "the primary" is `either-t100` (fixed in advance, marked in the *Conditions* table). The τ = 0.5 and 2.5 arms are reported beside it but not gated. If the primary turns out task-dirty, H2–H4 go unscored and the refuted H1 is the finding.

H1: Task cost. Every condition is task-clean: `named_holdout` exact match within 0.02 (absolute) of the seed mean of the in-experiment control. No partial credit; nothing in this design adds task pressure, and no anchored condition on this testbed has ever shown any.

H2: The pull selects the labelled operand. On the primary's per-group weight profiles: **(a)** at the embedding slice, each group's leading role is its own operand (op1 for G1, op2 for G2) with weight ≥ 0.4 in both (uniform is 0.25); **(b)** across the four post-attention slices, the between-group contrast in op2 weight, $\mathrm{mean}_\ell\,[\bar{\pi}_{G2}(\ell, \mathrm{op2}) - \bar{\pi}_{G1}(\ell, \mathrm{op2})]$, is ≥ 0.2. Partial: (a) holds and (b) lands in $[0.1, 0.2)$. (a) is close to forced by the shared embeddings *if* the pull aligns red tokens at all, so on its own it mostly rules out the syntax latch. (b) is the real question: ex-2.1.9's deep-slice winner-take-all was per *run*, and (b) asks whether the winner can vary per line when the label demands it. Contrary outcomes: *global latch* — both groups lead at the same role (op1, or worse, `+`), the contrast sits near 0, and on op2-triggered lines the pull is dragging non-red states onto the axis, which H3(a) should catch as ᾱ contamination; *uniform* — the primary reaches 0.4 in neither group at any slice (τ = 0.1 is the sharpest rung, so if it can't concentrate, the softer arms won't either), meaning the pool can't tell the positions apart once the label stops pointing.

H3: The operating point survives the wider labeller. On the primary: **(a)** containment, $\bar{\alpha} \leq 0.1$ at op1 (also the contamination detector for H2's global-latch outcome); **(b)** retention, every run whose running maximum of m_line reaches 0.2 ending at $\geq 0.8\times$ that maximum (quoted as the minimum across seeds); **(c)** grading, the primary's seed-mean op1 response r^2 against `sim^1.5` no more than 0.1 below the `op1-labels` arm's (higher is fine). Partial: (a) and (b) hold, (c) misses.

H4: Selectivity holds without the pointer. Control-subtracted m_line of the primary reaches ≥ 0.75 of the control-subtracted m_line of `slot-oracle`. Partial: $[0.5, 0.75)$. The latch account predicts roughly the op1-triggered share of the label mass (~half) plus whatever the shared embeddings give away, so a pass has to clear that; for scale, ex-2.1.9's pooled arms recovered 91–98% of their oracle.

Note that H4 can pass while H2(b) fails: the shared embeddings can deliver selectivity to labeled lines with no per-line localization at depth. That would suggest selectivity from the lexicon rather than from localization.

Task cost (H1)

condition	seen EM ↑	holdout EM ↑	holdout NLL ↓	`open` dist ↓	Δ holdout EM	H1 gate
`lam0`	1.000 ±0.000	1.000 ±0.000	0.012 ±0.001	0.132 ±0.002	+0.0000	clean
`op1-labels`	1.000 ±0.000	1.000 ±0.000	0.014 ±0.002	0.135 ±0.007	+0.0000	clean
`either-t100`	1.000 ±0.000	0.995 ±0.014	0.029 ±0.036	0.136 ±0.006	-0.0052	clean
`either-t500`	1.000 ±0.000	0.993 ±0.002	0.033 ±0.009	0.134 ±0.001	-0.0065	clean
`either-t2500`	1.000 ±0.000	0.996 ±0.004	0.021 ±0.007	0.134 ±0.001	-0.0039	clean
`slot-oracle`	1.000 ±0.000	0.995 ±0.002	0.027 ±0.009	0.132 ±0.001	-0.0052	clean

Behavior by condition. Each value is the seed mean, with half the seed range beside it. EM is exact match on the answer token, NLL its surprise in nats, and `open` dist the RGB distance from the guessed color to the true mix on pairs with no named answer. Δ holdout EM is the signed gap to the in-experiment control, which the H1 gate scores at 0.02.

H1 holds. The largest gap from the control is 0.0065, at `either-t500`, against a gate of 0.02, so H2–H4 are all scored. Labels that no longer say which operand drew change nothing the task loss can see, as with every anchored condition on this testbed so far.

Where the weight went, by label group (H2)

The `op1-labels` arm re-runs `pool-t100` from ex-2.1.9 verbatim (same labels, same seeds, same draws) through the widened sampler, so any drift from the new labelling code path is visible before we read an either-slot condition.

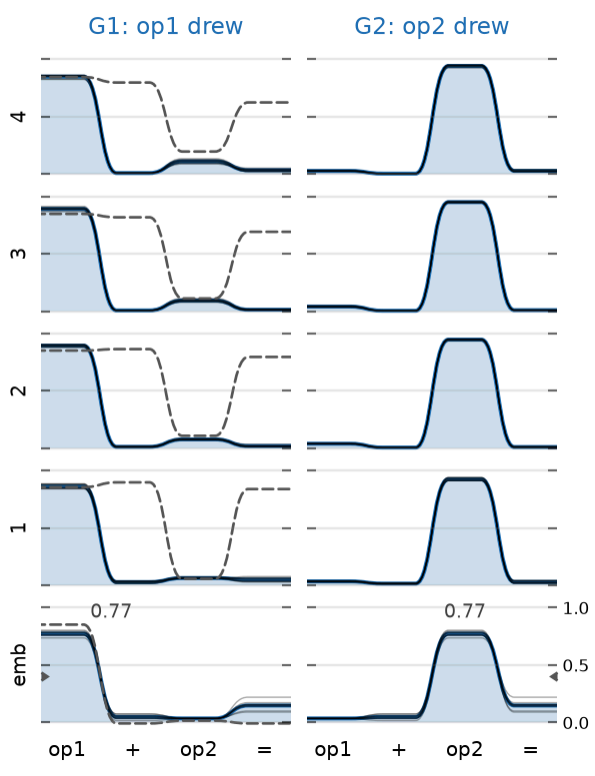
condition	m_{span}	ex-2.1.9	Δ	$\bar{\alpha}_{span}$	ex-2.1.9	Δ	m_{op1}	ex-2.1.9	Δ	$\bar{\alpha}$ (op1)	ex-2.1.9	Δ
`op1-labels`	0.704 ±0.017	0.704	-0.000	0.242 ±0.018	0.242	+0.000	0.704 ±0.017	0.704	-0.000	0.075 ±0.012	0.075	+0.000
`lam0`	0.028 ±0.036	0.023	+0.005	0.033 ±0.019	0.018	+0.015	-0.018 ±0.025	-0.029	+0.011	-0.041 ±0.046	-0.012	-0.029

The reproduction check. Each statistic is this experiment's seed mean (± seed standard deviation) beside the same statistic recomputed from ex-2.1.9's published arrays under the same per-run convention, and the difference. `op1-labels` re-runs `pool-t100` through the widened labelling code; `lam0` runs the either-slot labeller, so it is a new draw of the control rather than a re-run.

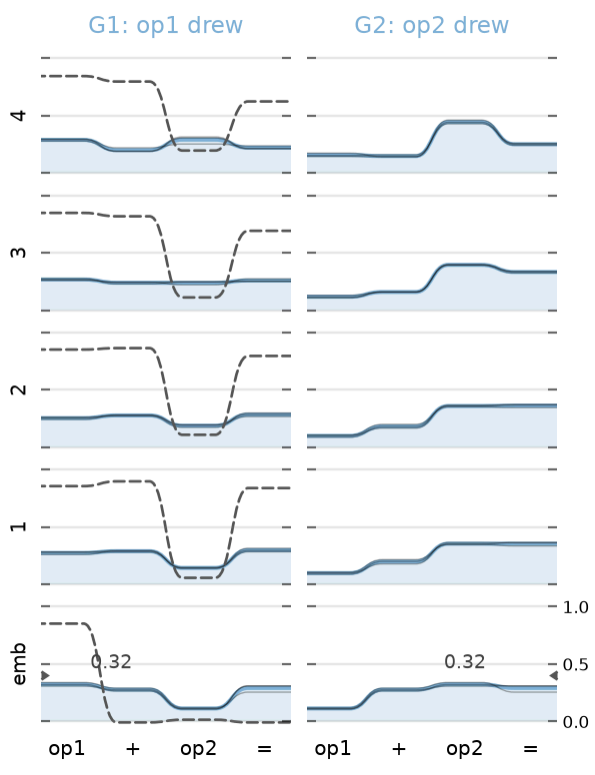
The reproduction is as close as the rounding in the store allows: every carried statistic lands within ±0.001 of its target, and the embedding lead weight (0.77) matches the 0.77 of ex-2.1.9. The op1-keyed draw consumes the rng as the old sampler did, so this is the same training trajectory replayed, and the either-slot conditions can be read against `op1-labels` as a like-for-like baseline.

The gaps in the control are the size of the seed spread, a reminder that it consumes *different* draws (two per line from the either-slot labeller), so that comparison carries corpus-draw noise, as the method warned.

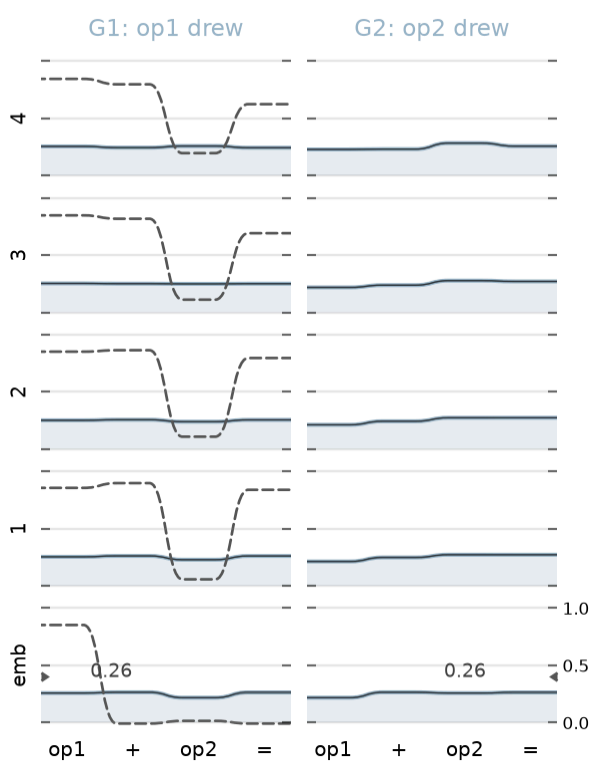
Both groups weight the same 5832 probe lines by the one-slot-only probabilities of the labeller itself, G1 by P(only op1 drew) and G2 by P(only op2 drew). The profiles below are the group-weighted mean softmin weights, per slice and role. Under a global winner the two columns of each pair would be identical; under per-line localization they mirror each other.



The primary *— τ = 0.1; deep-slice op2 contrast +0.85 ±0.01.*



Softer *— τ = 0.5; deep-slice op2 contrast +0.17 ±0.01.*



Near the span mean *— τ = 2.5; deep-slice op2 contrast +0.03 ±0.00.*

Per-group weight profiles under the either-slot labeller. *Within each sub-figure: columns are the label groups (G1 weights probe lines by P(only op1 drew), G2 by P(only op2 drew)); rows are residual slices with the embedding at the bottom; the shaded smooth-step is the seed-mean softmin weight $\bar{\pi}_G(\ell, t)$ over the span roles, with per-seed hairlines (nine on the primary). The dashed line behind the G1 columns is the un-anchored control's readability profile $R^2(\ell, t)$ for op1's redness. At the embedding, the caret on each column's outer axis marks the H2(a) gate level (0.4; uniform is 0.25), and the number above each group's own operand is the seed-mean weight there. Each sub-caption quotes the H2(b) deep-slice op2 contrast (gate ≥ 0.2, scored on the primary only).*

H2 holds, and the profiles mirror each other. (a) At the embedding the leading role in each group is its own operand, at weight 0.77 (gate ≥ 0.4), in all nine runs individually. The two groups take equal values there by construction, since at the embedding the state of a token cannot depend on its line.[4]

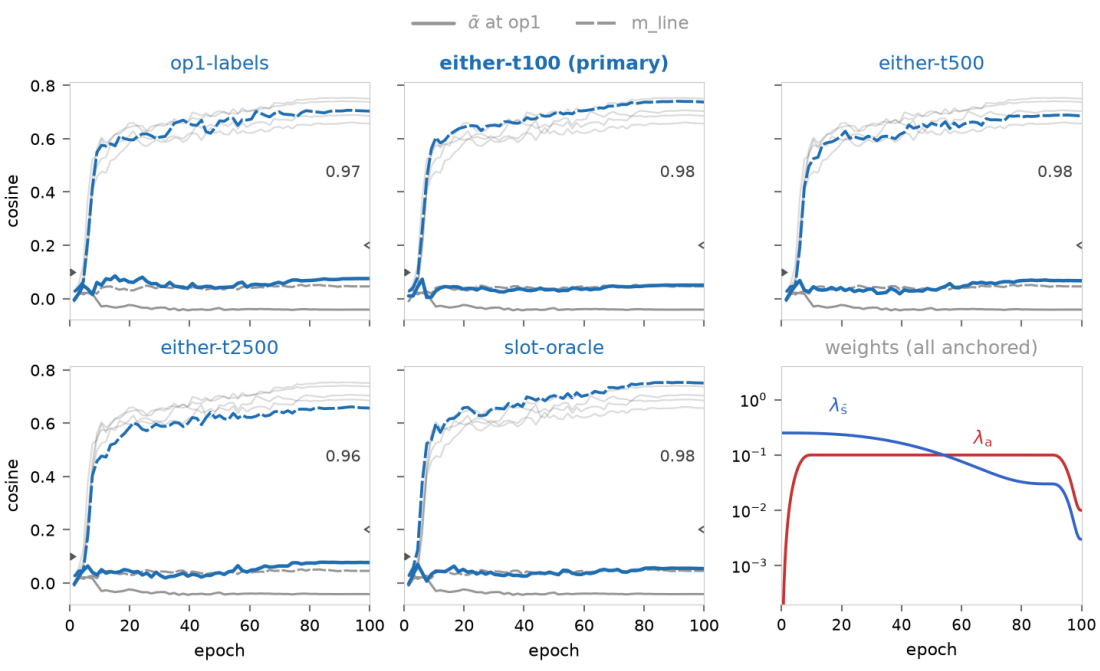
(b) The between-group op2 contrast over the post-attention slices is +0.85 ±0.01 against the 0.2 gate. The op2 weight of G2 at the last slice is 0.94, so on the lines where op2 was the evidence the pull puts nearly everything there. The winner-take-all behavior of ex-2.1.9 is still in force, and the winner now varies line by line: the columns are far from identical, and they do not collapse to uniform at the τ of the primary.

The unscored arms behave as the τ ladder predicted. At τ = 0.5 the embedding is nearly flat (lead 0.32) and the deep contrast reaches only +0.17; at τ = 2.5 the profile is the span mean in all but name (contrast +0.03). A pull that cannot concentrate cannot localize, however good the labels are.

The op1-keyed reference arm has a *measured* contrast of +0.69, worth holding onto for M3. The shared lexicon aligns red op2 tokens for free, so a softmin read at τ = 0.1 finds them even though the pull never targeted op2. But that free contrast decays with depth: the op2 weight of G2 falls from 0.83 at slice 1 to 0.55 at slice 4, where the primary holds (0.92 → 0.94). Training on the wider labels is what keeps the localization alive where the states are contextual.

The operating point under the wider labeller (H3)

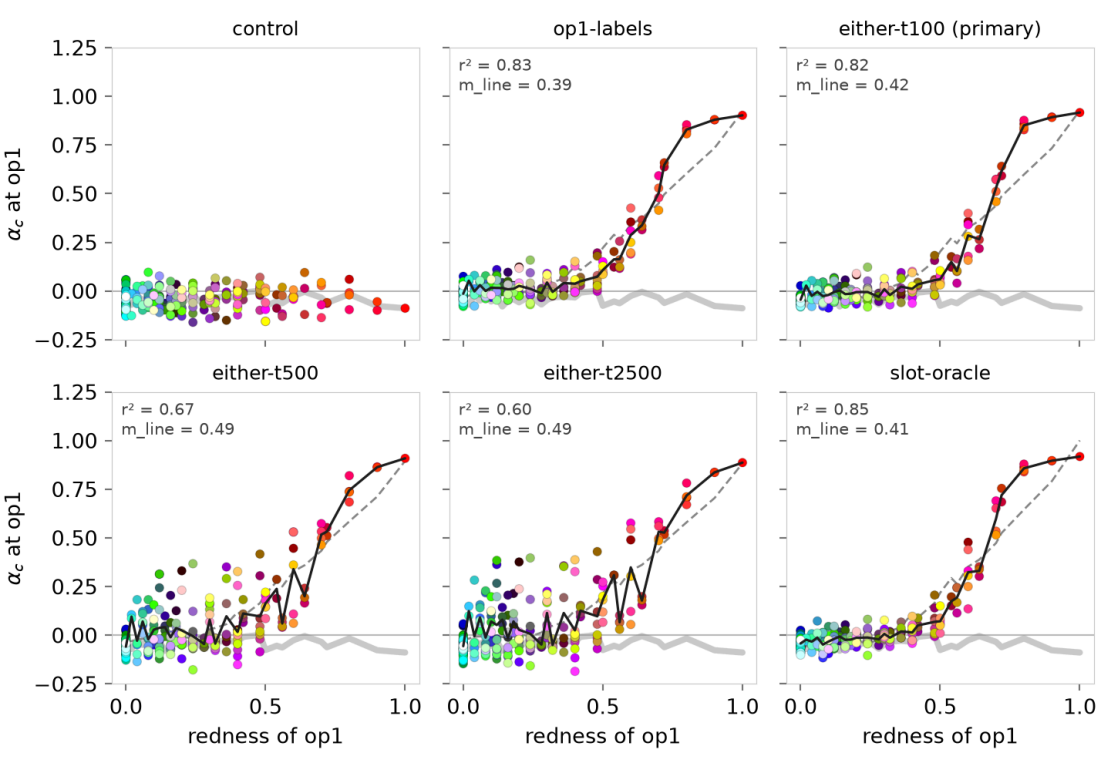
The endpoint numbers say where each run finished; the trajectories show how it got there. In particular, whether a pull that has to localize per line attains its margin later, or holds it less firmly once the repulsion anneals away, than the op1-keyed reference.



Training dynamics by condition. *Seed means of the two cosines on the anchor axis, on one shared scale: solid is $\bar{\alpha}$ at op1 (contained at the filled caret ▶, 0.1); dashed is m_line (its retention floor of 0.2 at the open caret ◁). The grey pair is the un-anchored control; the faint lines behind each panel are the m_line trajectories of the other four anchored conditions, for comparison in place. The number in each panel is that condition's retention (final over peak m_line, minimum across seeds; gate ≥ 0.8, scored on the primary). The trajectory instrument reads one line per color, so its level sits above the endpoint statistic's, which averages all 27 partners; retention compares within the curve, where the instrument is constant. Bottom right: the weight schedule every anchored condition trained under — anchor λ_a in red, repulsion $\lambda_{\bar{s}}$ in blue, log scale — shared because the anti-subspace term is fixed at the operating point.*

H3(a) and (b) hold. On the primary, $\bar{\alpha}$ at op1 ends at 0.050 ±0.021 against the 0.1 gate. That is *below* the 0.075 of the op1-keyed reference, so the contamination a global latch would leave (non-red op1 states dragged onto the axis on op2-triggered lines) is absent, which agrees with the H2 profiles. Op1 now hosts less of the drift, perhaps because half the pull budget now goes to op2.

Retention is 0.98 against 0.8, taken as the minimum over nine seeds, a stricter read than for the three-seed conditions; the worst retention anywhere in the sweep is 0.96. The five anchored panels above are near copies, so localizing per line neither delays the margin nor loosens its hold.



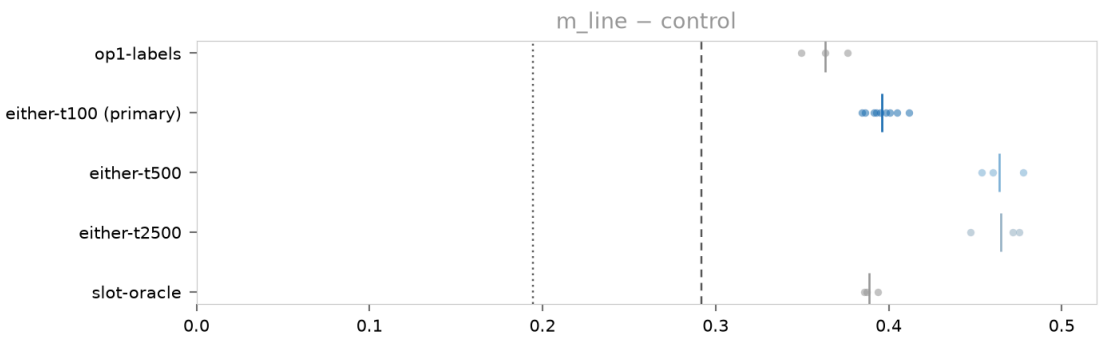
Grading: alignment at op1, per color. *α_c, the mean over slices of $\cos(h, \hat{v}_{\text{red}})$ at op1, seed mean, against the redness of the color. One mark per color, drawn in that color; the thin dark line is the mean response at each of the 29 distinct redness levels, the flat grey band the same line for the control, and the dashed line `sim^1.5` rescaled onto the response by least squares — the shape the r^2 statistic scores against. r^2 is computed per color, with no binning, so scatter about the conditional-mean line counts against it even where that line tracks the reference; the exploratory section separates the two. H3(c) gates the primary's r^2 at no more than 0.1 below `op1-labels`'s, whose own r^2 (0.83) reproduces ex-2.1.9's `pool-t100` (0.83).*

H3(c) holds. The primary grades at r^2 = 0.82 against 0.83 for the reference arm, a drop of 0.008 against the 0.1 tolerated. So widening the labels cost the response shape essentially nothing at the τ of the primary.

The softer arms fare worse: 0.67 at τ = 0.5 and 0.60 at τ = 2.5. The grading panels show where it goes; the low-redness response turns noisy and drifts off zero, since a diffuse pull drags every span position of every labeled line, red op1 or not. The slot oracle grades best of all (0.85), as the analogue in ex-2.1.9 did.

Selectivity against the slot oracle (H4)

m_line is the selectivity delivered to the lines the labeller flags, and the slot oracle — trained on the same label draws, told which operand drew — is the reference for what the withheld information is worth.



Control-subtracted m_line by condition. *Small dots are runs (nine on the primary), the tick their mean; the reference arms are greyed. Dashed caret: the H4 gate, 75% of the slot oracle's control-subtracted mean (0.389); dotted, the 50% partial. The latch account — a pull stuck on op1 serving only the op1-triggered half of the label mass — predicts landing near the partial line.*

condition	m_{line} ↑	− control	oracle frac	m_{span}	m_{op1}	$\bar{\alpha}_{span}$ ↓	$\bar{\alpha}$ (op1) ↓	r^2
`op1-labels`	0.387 ±0.014	0.363	93%	0.704	0.704	0.242	0.075	0.83
`either-t100` **(primary)**	0.420 ±0.009	0.396	102%	0.738	0.738	0.239	0.050	0.82
`either-t500`	0.488 ±0.012	0.464	119%	0.686	0.685	0.424	0.067	0.67
`either-t2500`	0.489 ±0.015	0.465	120%	0.656	0.656	0.432	0.077	0.60
`slot-oracle`	0.413 ±0.004	0.389	–	0.751	0.751	0.167	0.054	0.85

The anchored conditions, seed means (± seed standard deviation on the scored statistic). Only the primary (so marked) is gated; "oracle frac" is control-subtracted m_{line} as a fraction of the slot oracle's 0.389. m_{span} and m_{op1} are carried for continuity with ex-2.1.9 (both key their line weights on op1 alone), $\bar{\alpha}_{span}$ and $\bar{\alpha}$ (op1) are the drift statistics, and r^2 repeats the grading track scored under H3.

H4 holds. The primary delivers 102% of the control-subtracted m_line of the oracle, against the 75% gate. That is level with the oracle, within seed noise (spreads of 0.009 and 0.004 on conditions sharing the same label draws). The latch account predicted roughly the op1-triggered half plus the lexicon give-away, and it is ruled out twice over: by this fraction, and by the H2 profiles.

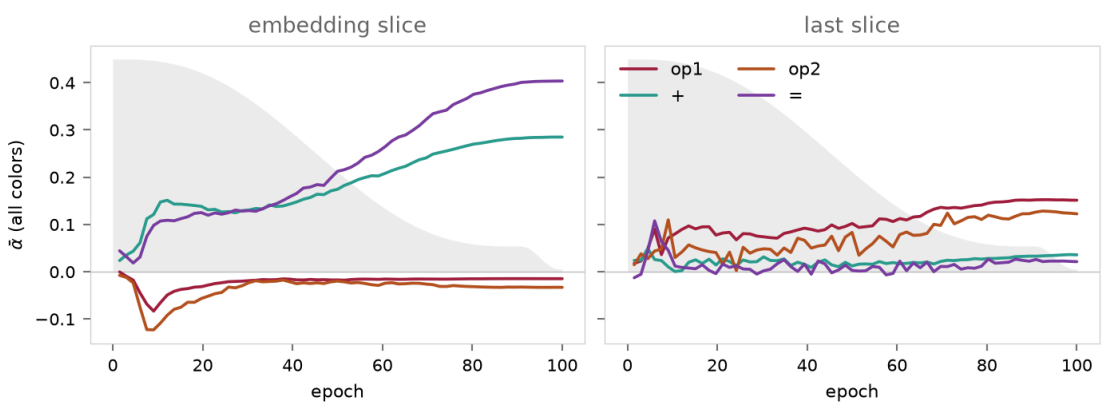
The prereg allowed for landing at or above the oracle in principle, since the oracle is bound to the operand that drew even where the concept has migrated at depth. At this margin the defensible reading is parity: knowing which slot triggered is worth nothing here that the pool cannot recover on its own.

The op1-keyed arm reaches only 93% of the oracle on the same statistic, because its pull never serves the op2-triggered half of the label mass. Widening the labels is what closed the gap.

The soft-τ arms *exceed* everything on m_line (0.49 at τ = 0.5) while failing every other instrument: contrast near zero (H2), grading down by ~0.2 (H3(c)), and span drift ᾱ_span at 0.42 against 0.24 for the primary. A diffuse pull aligns the whole span of labeled-ish lines, and a margin that takes the best role per slice rewards that spread. The number is real, but what it measures there is a smeared, poorly graded alignment rather than a localized one. The exploratory section returns to this trade.

Exploratory analyses

Nothing here is preregistered; everything in this section is post hoc. The four questions below are the candidates the preregistration queued, in its order.



Unweighted drift by role, over training (primary, seed mean). *$\bar{\alpha}(\ell, t)$ over all 216 colors at the embedding slice (**left**) and the last slice (**right**), one curve per span role; the shaded band is the repulsion weight $\lambda_{\bar{s}}$ on its own scale, for timing. The trajectory instrument reads one line per color, as in the dynamics figure.*

When the syntax embeddings drift (queued by ex-2.1.9). Ex-2.1.9 could see that the `+` embedding ended near 0.30 alignment, but not when. The per-role trajectory shows the drift occurs late. At epoch 30 the two syntax embeddings sit at 0.13 (`+`) and 0.13 (`=`). The drift accrues through epochs 30–90 as the repulsion anneals toward the hold ratio, reaching 0.28 and 0.40, and the end-of-training anneal of the anchor recovers none of it. So the syntax drift is bought against the *weakening* repulsion, and in this experiment `=` is the larger of the two; `=` sits outside the labelled operands but inside the pulled span.

At depth the picture inverts. The operand roles carry the residual drift (0.15 at op1, 0.12 at op2) and the syntax roles carry none, and that is where the containment gate of H3(a) reads and passes.

Is the deep choice still one-hot? (per-line weight entropy.) Yes, per line. On the label-weighted probe mass, the leading role at the last slice holds over 0.9 of the weight of a line on 81% of the mass (seed range 79%–83%). The un-labelled bulk stays near uniform, since a flat alignment profile gives the softmin nothing to concentrate on. The winner-take-all behavior of ex-2.1.9 survives the wider labels in the form H2 requires: committed within a line, varying across lines.

The both-red lines. Where both operands are red, either answer is defensible, and the pool declines to choose: at the embedding the both-draw-weighted profile splits 0.49/0.49 between the operands (forced by the shared lexicon), and at the last slice it still spreads: op1 0.20, op2 0.47, `=` 0.33, with 37% of the both-draw mass within 0.2 of an even operand split. These lines are 0.1% of the label mass, so nothing above turns on them. They preview the M3 case where a document holds the concept in several places at once.

The soft-τ trade. Across the ladder, localization and grading fall together while m_line rises: contrast 0.85 → 0.17 → 0.03 and r^2 0.82 → 0.67 → 0.60 as τ goes 0.1 → 0.5 → 2.5, with m_line highest at the diffuse end (the H4 table). A nearly-mean pull does still buy line-keyed *margin*, in that the spans of labeled lines align more than the bulk. But it localizes barely at all once the label stops pointing, and what it aligns is poorly graded. For M3, a margin statistic alone cannot distinguish localized selectivity from smeared selectivity, so it needs the group contrast and the grading track beside it. The rungs are far apart, though: everything between the first two is unmeasured, so where the trade turns, and how sharply, is open. A finer ladder near τ = 0.1 is queued in the science backlog.

Does the per-color r^2 score the soft arms down only because the reference zig-zags? (queued by the review round.) No — what separates them is per-color scatter, though the conditional-mean reading does fit them best: smoothing both curves over a sliding window of two redness levels and correlating puts the soft arms ahead, at r^2 0.94 and 0.94 against 0.92 for the primary and 0.94 for the oracle, matching the conditional-mean lines in the grading figure. The zig-zag itself is unrelated to the frozen statistic, which is computed per color with no binning, and no condition recovers it: every response tracks the reference's within-level variation at $r^2 \leq$ 0.04.

What separates the soft arms is scatter, not the zig-zag: within-level scatter carries 70% and 71% of their residual variance about the affine `sim^1.5` fit, against 33% for the primary. Seed noise accounts for only 7% of that residual, so the deviations are stable properties of the pull at that τ, reproducing across seeds.

The two readings disagree because they measure different things: a soft pull grades the population of colors as a whole while giving individual colors alignments their redness does not predict. For intervention the per-color reading is the one that matters — a mis-scored color would respond to a later intervention as if it were redder than it is — so M3 should carry both; only the pair separates graded-on-average from graded color by color.

Why every curve flattens near the pole (queued by the review round). `sim^1.5` steepens toward redness 1, but the responses compress instead. The primary's per-slice profiles show where: at the embedding the response is *steeper* than the reference (three reddest colors 1.9× above the next nine, against 1.48× for `sim^1.5`); in mid-stack the top of the scale runs out of cosine (pure red reaches ≈ 0.97, next tier only 1.18× below it where the reference calls for 1.48×); and at the last slice the red state rotates partly away (0.75) as it comes to encode the mix.

Averaged over layers — steep start, saturated middle, rotated end — that is a curve with a compressed top (1.31× overall): the flattening is the readout saturating. Near the pole there is no headroom, so colors pulled often enough land at the same ceiling, and whatever keeps them distinguishable for the task lives in directions orthogonal to the axis — which in $d = 64$ costs almost no cosine. That is the shape to expect of a strongly anchored top of the scale, and the graded part of the response lives below it.

Discussion

The doubt ex-2.1.9 could not settle — that the pooled pull's clean localization was inherited from labels that always pointed at op1 — is settled: removing the pointer changed nothing an oracle could improve on (H4). So for M3, where document-level labels do not point, the burden of localization can sit on the loss side: label-side help — EM-style responsibilities, attention-derived position weights — drops from "probably needed" to a contingency for harder settings.

The qualification is how easy this testbed makes the per-line problem: two candidate positions, a lexicon shared by both, each operand's color legible at its own position (the scope limit in the introduction), and multi-site lines rare enough that nothing gated turns on them. Parity with the oracle therefore reads as: pooling suffices when the concept is findable line by line. That is the default M3 should start from; the queued grammar extensions are what would stress it as documents get longer and sites more numerous.

The remaining risk lives in the operating point, τ. A margin-guided search over τ would walk to the smeared end and report success (the soft-τ trade, in Exploratory), so selectivity in M3 needs reading as a triple of margin, group contrast, and per-color grading — with both grading readings in the instrument set, since only the per-color one feels the failure an intervention would. A finer τ ladder near the primary is queued; everything between the first two rungs is unmeasured.

For D2.1 the chain is now complete: a span pull broadcasts (ex-2.1.6), the repulsion schedule sets a workable operating point (ex-2.1.8), mellowmax pooling localizes the pull within a labeled span (ex-2.1.9), and the localization does not depend on the labels saying where (this experiment). The close-out claim this leaves D2.1 with: in the residual stream of a small transformer, anchoring works under labels that say only *that* a concept occurs, at no measurable task cost, with containment, retention and grading at the level of the op1-keyed reference, and at selectivity parity with an oracle told where the concept sits. The bounds of the claim are those of the testbed: one synthetic task, one concept, one architecture size, and labels drawn fresh at every visit. The fixed-label variant and the document-shaped pull, a span that includes the answer, are the D2.1 items still open ahead of D2.2.

---

1. τ is a temperature (the weight ratio between two positions is $e^{\Delta x/\tau}$), so the rungs step geometrically, ×5. At τ = 2.5 the ratios on realistic alignment spreads are ~1.2: a pull with only a tiny bias toward its best-aligned position, nearly the span mean that ex-2.1.9's τ = ∞ arm pinned, which is why no span-mean arm runs here. ↩

2. The volume stays small: 5 slices × 5832 lines × ~6 positions in float32 is ≈ 0.7 MB per array, two arrays per run, so ≈ 35 MB across all 24 runs. That is an order of magnitude over ex-2.1.9's per-color arrays but nowhere near needing different processing, and the per-role trajectory adds well under a megabyte per run. ↩

3. To recap m_span (from ex-2.1.9): on the per-color maps, take the affinity-weighted mean alignment minus the plain mean, giving a margin per (slice, role) saying how much closer red-ish material is to the axis than average. Keep the best span role per slice, and average over slices. m_line is the same recipe on the per-line maps: weight *lines* instead of colors, using the labeller's own P(labeled) per line, normalized. m_span keyed on op1 alone, whereas m_line keys on both operands. ↩

4. The probe set contains both orders of every pair, and the span positions share token embeddings, so the G1 and G2 profiles are exact mirrors at the embedding slice. What the gate tests there is concentration: that the pool has moved away from uniform, and that the concentration sits on operands rather than syntax. Depth is where the groups can differ, which is why H2(b) reads only the post-attention slices. ↩