Ex 2.2.6: a pilot of the whole-span labeller

A scouting run, with no gates. The labeller in ex-2.2.3 reads only the operands, and pulls only the prompt span. So on a line whose answer is the reddest thing in it, that answer never draws a label and nothing ever pulls its position. We retrained the adopted point under labellers that also read the answer, or also pull the whole line, or both.

Pulling the whole line puts between a tenth and a quarter of the pull on the answer, and doubles the alignment the redder lines have there, at no cost to the task or to placement. Reading the answer changes nothing visible. Neither change makes the answers on those lines depend on the axis, because the answer is read out at `=`, one position before the pull lands.

Observations

- **No task cost.** Held-out exact match is at least 0.998 on every op of every arm, the same level as the twenty production seeds (table).
- **Placement stays in the production band.** Under the labeller of each arm, m_line runs 0.381–0.397, against 0.392 ±0.020 on `recipe-short`. Lead, contrast, grading, latch and retention behave the same way (table).
- **The span moves the pull; the keying does not.** With a whole-line pull, the softmin weight on the answer role, averaged over the four residual slices, is 0.17 on `line-whole` and 0.12 on `either-whole`, against 0.01 on production. `line-prompt` pulls only the prompt span, and sits at 0.02 with the production profile (figure).
- **The redder lines carry more redness at the answer under the whole-line pull.** On the redder lines of `multiply`, $\Delta\alpha$ at the answer is 0.32 and 0.29 on the two whole-line arms, 0.17 on `line-prompt`, and 0.17 ±0.03 on production. At `=` and at the newline, nothing moves (figure).
- **Projection still leaves those answers intact on every arm.** Across the four arms, redder-line accuracy under `projection` is 0.59–0.65 on `multiply` and 0.92–0.96 on `mix`. Red-line removal and the non-red `mix` deficit (0.009–0.044) are unchanged within the noise of two or three seeds (table).

Why, and what we ran

A document-level label, the M3 shape, says a document is about *red* and nothing about where. Ex-2.2.3's labeller is closer to a token label: each operand draws at its own redness rate, and the pull covers the four prompt roles. Its E3 section found the blind span that implies: on lines whose answer is redder than both operands the stream carries the answer's extra redness at the answer position, the labeller never keys there, and under `projection` those lines keep their answers. The labelling-span item files the fix as a fast-follow; this pilot is that follow.

Two changes, each one step toward the document shape. **Keying** `line` (`sca.anchoring.LabelSpec`): the answer slot draws at its redness rate too, so P(labelled) becomes $1 - (1 - p_1)(1 - p_2)(1 - p_3)$ and the redder lines draw more often. **Span** 6: the pull covers the answer and the newline as well as the prompt. The anchor term is the per-line mellowmax over the pulled span with a conserved per-line budget, so the wider span gives the softmin more places to put the same pull and changes nothing else about its strength. Everything else is ex-2.2.3's `recipe-short` ($\lambda = 0.1$, 50 epochs), on the same corpus and the same probe lines, and the production seeds of that arm are the fourth corner:

arm	seeds	the answer draws	pull covers	
`line-whole`	3	yes	whole line	the answer draws; the pull covers the whole line
`line-prompt`	2	yes	prompt span	the answer draws; the pull covers the prompt span
`either-whole`	2	no	whole line	operands draw, as production; the pull covers the whole line
`recipe-short`	20	no	prompt span	production, ex-2.2.3's adopted point

Every statistic below is measured on the same probe lines the production arm was measured on. Where a statistic weights lines by P(labelled), the table says which labeller's weighting it uses, since the two differ on the redder lines by construction.

Task cost

Exact match on the held-out pairs of each op. The reference is production's twenty seeds; the pilot arms have two or three each, so the half-ranges are rough.

arm	seeds	`mix`	`add`	`screen`	`multiply`	`lighten`	`darken`	Δ mix
`line-whole`	3	1.000 ±0.000	1.000 ±0.000	1.000 ±0.000	1.000 ±0.000	0.999 ±0.002	1.000 ±0.000	+0.001
`line-prompt`	2	1.000 ±0.000	1.000 ±0.000	1.000 ±0.000	0.998 ±0.002	0.998 ±0.002	1.000 ±0.000	+0.001
`either-whole`	2	1.000 ±0.000	1.000 ±0.000	1.000 ±0.000	1.000 ±0.000	1.000 ±0.000	1.000 ±0.000	+0.001
`recipe-short`	20	0.999 ±0.002	0.999 ±0.002	1.000 ±0.000	0.999 ±0.004	0.999 ±0.002	0.999 ±0.006	+0.000

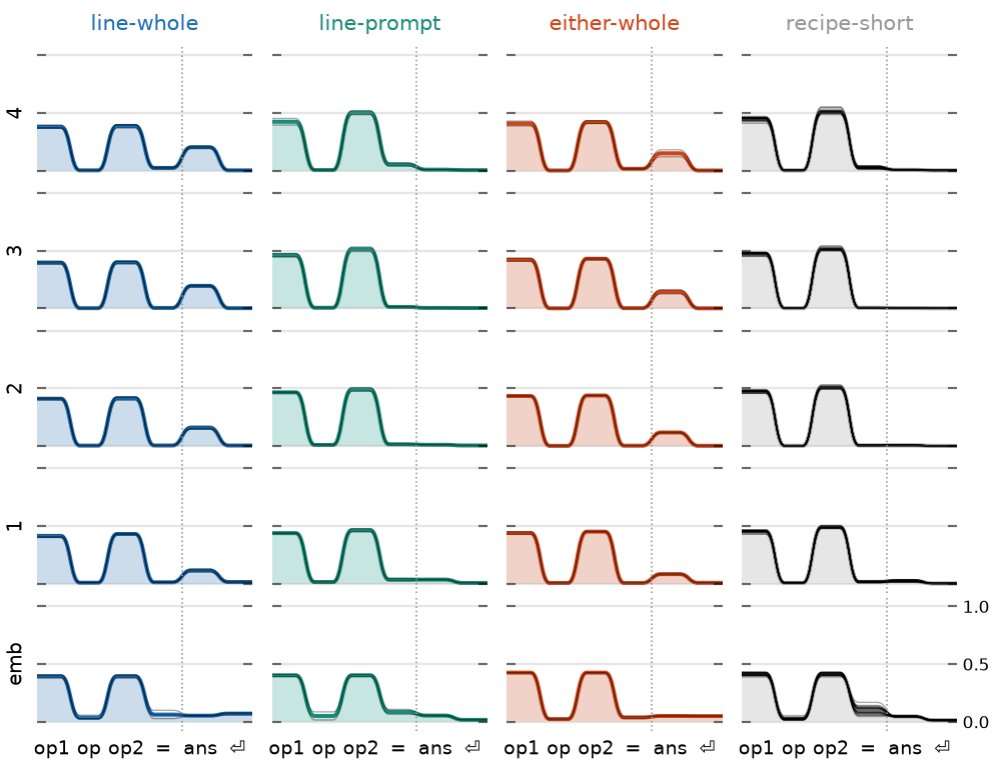
Held-out exact match per op, seed mean with half the seed range; the last column is the `mix` gap from the production corner. Ex-2.2.3's task gate was a `mix` gap within 0.02 of its own control.

Where the pull lands

Ex-2.2.3's placement statistics on the `mix` probe lines, then m_line recomputed four ways: over the four prompt roles or all six, weighting probe lines by the operand-only labeller's P(labelled) or by the line-keyed one. A stored m_line is the four-role margin under the arm's own labeller.

arm	m_line (stored)	m_line 4 · either	m_line 4 · line	m_line 6 · either	m_line 6 · line	ᾱ op1	lead (emb)	contrast	r² sim	latch π	retention
`line-whole`	0.381 ±0.010	0.404 ±0.007	0.381 ±0.010	0.410 ±0.013	0.387 ±0.010	0.120 ±0.026	0.87 ±0.02	0.86 ±0.00	0.907 ±0.008	0.01 ±0.00	0.99 ±0.01
`line-prompt`	0.383 ±0.010	0.406 ±0.006	0.383 ±0.010	0.406 ±0.006	0.383 ±0.010	0.084 ±0.015	0.87 ±0.01	0.86 ±0.00	0.869 ±0.026	0.02 ±0.00	0.99 ±0.00
`either-whole`	0.397 ±0.001	0.397 ±0.001	0.373 ±0.001	0.397 ±0.001	0.373 ±0.001	0.113 ±0.008	0.87 ±0.00	0.87 ±0.00	0.908 ±0.011	0.01 ±0.00	1.00 ±0.00
`recipe-short`	0.392 ±0.020	0.392 ±0.020	0.367 ±0.021	0.392 ±0.020	0.367 ±0.021	0.083 ±0.056	0.83 ±0.04	0.87 ±0.02	0.886 ±0.044	0.02 ±0.01	0.99 ±0.01

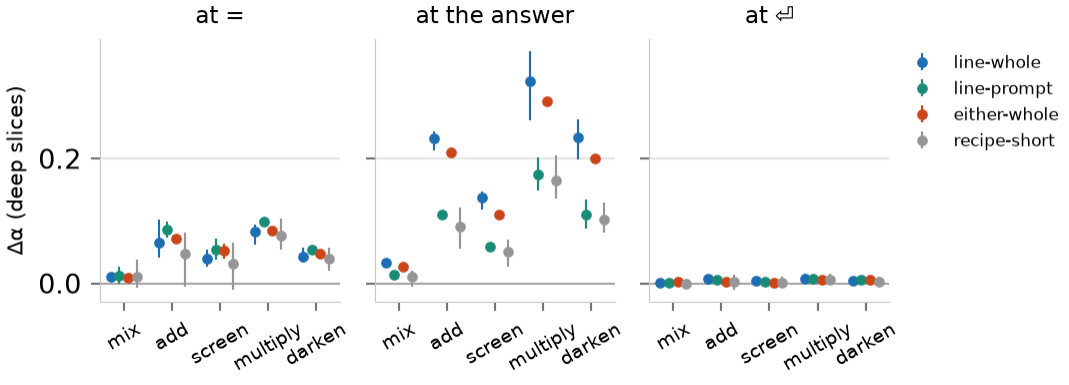
Placement on the `mix` probe lines, seed means with half the seed range. The four recomputed m_line columns read the same alignment maps with the role count and the line weighting varied; a six-role margin can only match or exceed the four-role one on the same weighting. ᾱ op1 is the mean alignment at op1 over every slice and color (ex-2.2.3's containment read); lead is the G1 group's softmin weight on op1 at the embedding; contrast is the deep-slice op2 weight of G2 minus G1; r² sim the grading of the op1 response against the similarity target; latch π the larger of the non-red group's deep-slice weights on the op word and on `=`; retention the final m_line over its running peak.



Softmin profiles over all six roles, per arm, on the `mix` probe lines. *Columns are arms, rows residual slices with the embedding at the bottom; each panel is the seed-mean softmin weight at the arm's τ, over the six roles, with lines weighted by the arm's own labeller's P(labelled) (production's operand-only weighting on `recipe-short` and `either-whole`); one hairline per seed. A dotted rule separates the prompt roles from the answer and the newline: ex-2.2.3's profiles stop at that rule, and a whole-line pull is free to cross it.*

The redder-than-both lines

Ex-2.2.3's E3 read, repeated on every arm and extended to the newline. Δα is the deep-slice alignment on the lines whose answer is redder than both operands, minus the alignment on lines of the same op in the same dose bin whose answer is not, at one position. Under the operand-only labeller the answer position was blind; a labeller that reads the answer has a reason to put alignment there, and a whole-line pull has somewhere to put it.



Δα on the redder-than-both lines, by position and arm. *Each marker is an arm's seed mean of Δα at one position, with the seed range as a bar, per op; the panels are* `=`*, the answer and the newline. A positive value says the stream is more aligned with the anchor on the redder lines than on dose-matched lines of the same op at that position. Ex-2.2.3 read the first two positions on* `recipe-short` *alone.*

arm	Δα at =	Δα at the answer	Δα at ⏎	answer weight, redder	answer weight, matched	clean acc	projection acc ↓
`line-whole`	0.011 ±0.005	0.033 ±0.006	0.001 ±0.000	0.25 ±0.07	0.23 ±0.08	1.00 ±0.00	0.96 ±0.01
`line-prompt`	0.013 ±0.013	0.013 ±0.004	0.002 ±0.001	0.11 ±0.01	0.11 ±0.01	1.00 ±0.00	0.92 ±0.01
`either-whole`	0.009 ±0.002	0.027 ±0.002	0.002 ±0.001	0.17 ±0.05	0.15 ±0.05	1.00 ±0.00	0.96 ±0.03
`recipe-short`	0.011 ±0.023	0.010 ±0.012	0.000 ±0.003	0.10 ±0.03	0.10 ±0.02	1.00 ±0.01	0.94 ±0.07

The 372 redder-than-both `mix` *probe lines, per arm. Δα as in the figure; the two weight columns are the deep-slice softmin weight on the answer role over all six roles, on the redder lines and on their dose-matched comparison lines. The right pair restricts H4 to these lines: exact-match accuracy clean and under* `projection`*. Under the operand-only labeller ex-2.2.3 found these lines keep their answers when the axis is projected out.*

Suppression and selectivity

Ex-2.2.3's H4 statistics under `projection`, per arm: accuracy on the red lines of each op (the removal read, lower is more complete), the non-red deficit on `mix` (the selectivity read), and accuracy on the redder-than-both lines of each op that has them.

arm	lines	`mix`	`add`	`screen`	`multiply`	`lighten`	`darken`	non-red deficit, mix
`line-whole`	red	0.01 ±0.02	0.13 ±0.04	0.08 ±0.03	0.07 ±0.03	0.13 ±0.06	0.20 ±0.03	0.024 ±0.024
	redder	0.96 ±0.01	0.83 ±0.03	0.89 ±0.03	0.64 ±0.09	—	0.88 ±0.02	
`line-prompt`	red	0.01 ±0.01	0.11 ±0.00	0.10 ±0.00	0.06 ±0.01	0.11 ±0.01	0.10 ±0.02	0.044 ±0.012
	redder	0.92 ±0.01	0.78 ±0.04	0.81 ±0.04	0.59 ±0.05	—	0.86 ±0.03	
`either-whole`	red	0.01 ±0.01	0.14 ±0.06	0.10 ±0.06	0.05 ±0.02	0.11 ±0.04	0.11 ±0.00	0.009 ±0.006
	redder	0.96 ±0.03	0.84 ±0.09	0.89 ±0.06	0.65 ±0.10	—	0.87 ±0.02	
`recipe-short`	red	0.01 ±0.04	0.14 ±0.18	0.12 ±0.14	0.08 ±0.11	0.14 ±0.10	0.18 ±0.18	0.026 ±0.035
	redder	0.94 ±0.07	0.81 ±0.15	0.87 ±0.12	0.61 ±0.12	—	0.88 ±0.04	

Exact-match accuracy under `projection` per op, on the red lines (dose ≥ 0.8) and on the redder-than-both lines, with the non-red (dose ≤ 0.2) deficit on `mix` in the last column. Ex-2.2.3's gates were red accuracy at most 0.2 on every op and a deficit at most 0.05.

What we make of it

The blind span has two halves, and the pilot separates them: the keying (does a line whose answer is the reddest thing in it draw a label at all?) and the span (can the answer position be pulled once the line has drawn?). The profiles say the span is what moves the pull. With six roles the softmin puts between a tenth and a quarter of its weight on the answer at every depth, whichever slots draw, and answer-drawing keying with a prompt-span pull reproduces the production profile.

On the `mix` probe lines that is what we would expect the mellowmax to do: a labelled line has a red operand and so a reddish answer, and the answer position is about as cheap to align as the operand.

The extra alignment does not make the answer depend on the axis. Under `projection` the redder lines keep their answers on every arm, at the production rate. The reason is the causal structure of the model rather than the labeller: the answer token is predicted from the stream at `=`, and the stream at the answer position feeds only the newline. So an anchor at the answer position sits downstream of the readout it might have changed, and no labeller that pulls there can close the gap E3 found.

$\Delta\alpha$ at `=` could close it, but it is small on every arm and unchanged. The redder lines lose nothing under `projection` because the operands that compute the answer are not red, and so not on the axis; ex-2.2.3 gave the same reading of this table.

For D2.2 the whole-line pull is harmless: no task cost, placement within band, and free to adopt when the document shape calls for it. But nothing we read here recommends it for the anchored-op experiments. There the prompt-span pull keeps the pulled positions away from the positions the answer is read from, and that separation is what makes the `=` and operand reads interpretable. So we keep the production labeller.

If the whole-line pull is adopted later, the number to watch is containment: $\bar{\alpha}$ at op1 is a little higher on both whole-line arms than on production, inside the twenty-seed band but on its upper side. For M3 the lesson is that a document-level label will put alignment on positions that carry the concept as an output, and an intervention that aims to change behaviour has to reach the positions whose stream feeds the readout.

What we would do differently: a single arm, `line-whole` against production, answers the question. Also, the one place an answer-position pull could act causally is the prediction of the newline, and the scorer in ex-2.2.3 does not read that; a follow-up that cares would add it.

Method notes

- The arms and the production corner share the corpus, the eval sets and the probe lines (the same seeds through ex-2.2.3's `prepare_corpus`); the line-keyed arms train against a probe table whose `line_p` is recomputed for the answer's draw, and that table is what this report weights by when a column says `line`.
- `line` keying draws the answer slot from a separate random stream, after the operands' draws, so an `either` arm's labels are unchanged by the code path (see `tests/sca/test_anchoring.py`).
- The softmin profiles run over all six roles at the arm's τ; ex-2.2.3's stored `pi` and `w_group` stop at `=`. The stored m_line is the four-role margin under the arm's own labeller, and the recomputed columns read the stored per-line alignment maps.
- The redder-lines comparison and the dose-matched weights are ex-2.2.3's E3 construction, applied unchanged.
- This is a pilot: three arms with two or three seeds each, and production's twenty for the fourth corner. Nothing here is gated, and nothing in it should be quoted as a result.