Ex 2.2.5: a pilot of stochastic rounding of off-grid answers

A scouting run, no gates. We retrained two models on a new corpus: the un-anchored control for the six-op grammar, and the adopted recipe. In this corpus an answer that lands between grid levels rounds *stochastically*, going to the upper level with probability equal to how far up it sits. The model learns the rule's answer distribution rather than a rounding: it puts as much mass on each candidate as the coin does, so exact match against a drawn answer sits at the ceiling the rule sets. Read the expected exact match and the calibration instead. Anchoring is unchanged. The geometry of the answer looked no more graded, though the probes were on lines that never round.

Observations

- **The models reach the ceiling a stochastic target sets.** Expected exact match on the held-out pairs sits on the holdout ceiling on `mix`, and within 0.04 of it on `screen` and `multiply`, for the control and the recipe alike (table). Exact match against the drawn answer adds the noise of a single draw, about ±0.03 over 256 lines. Model seeds share that noise, so it does not average away.
- **The answer distribution is calibrated to the rule.** On the rounded held-out lines the control puts 0.33 / 0.59 / 0.59 of its mass on the mode for `mix` / `screen` / `multiply`, against 0.33 / 0.58 / 0.57 for the rule. KL from the rule[1] runs 0.13–0.25 nats on the stochastic arms, against 2.3–5.1 on the nearest-rounded control, which commits to one level. At least 0.965 of the mass is on the candidate colors (figure).
- **Exact match against the mode is an argmax read, and on `mix` a coin toss.** It is 0.82 / 0.86 on `screen` / `multiply`, and 0.35 on the rounded pairs of `mix`, where half-way ties make the mode arbitrary.
- **Anchoring does not notice the corpus.** The m_line of `recipe-stoch` is 0.391 ±0.001, against 0.392 ±0.020 in production. Every other placement statistic is inside the seed band of production (table).
- **Suppression reads shift through their clean baseline.** On the nearest-rounded probe lines of `multiply`, the clean red-line accuracy of the recipe is 0.88: the argmax of a calibrated model lands on the nearest level only where the mode is clear. The redder lines go 0.78 → 0.54 under `projection`, against 1.00 → 0.61 in production. On `mix`, whose probe lines are on-grid, nothing moves.
- **Redder-than-both counts rise on the ops that round up.** With no model in the loop, the expected count changes by +14.5% on `screen`, +20.8% on `multiply`, and -1.0% on `mix` (table).
- **No clearer gradedness at the answer, on probes that could not show it.** Peak strict R^2 of the answer RGB over the slices is 0.92 at `=` and 0.95 at the answer on the stochastic control, against 0.85 and 0.83 on the nearest one. At the deep slices, the two nearest-rounded seeds differ from each other by more than the corpora differ. The probe lines come from `mix`, which never rounds (figure).

Why, and what we ran

Ex-2.2.3's grammar computes each answer channel on the 0..15 scale and snaps it to the nearer grid level, so `screen`, `multiply`, and (off its on-grid pairs) `mix` are step functions of their raw value. The stochastic-rounding item asks what a variant that rounds by coin flip would settle: (a) the exact-match ceiling, and whether exact match is then the wrong statistic; (b) whether the answer's representation becomes more graded; (c) whether the redder-than-both counts, and so E3, move.

The variant is `sca.data.ops.Rounding = "stochastic"`: per channel and per line, the upper neighbour is drawn with probability equal to how far up the interval the raw value sits, so the mean target is the raw value itself and a tie is a coin flip. The (op, pair) sequence of the corpus and the eval lines are the production ones at the same seed; only the answers of rounded lines differ. Three arms, all at the adopted point's length (50 epochs), all under ex-2.2.3's either-slot labeller:

condition	seeds	corpus	anchor
`control-stoch`	2	stochastic	none — un-anchored, stochastic corpus
`recipe-stoch`	3	stochastic	λ = 0.1 — the adopted recipe, stochastic corpus
`control-near`	2	nearest	none — un-anchored, nearest-rounded corpus

`control-near` is two fresh seeds of production's `control-short`, trained here so the rounding readout and the cube probes have a same-code nearest-rounded comparison. The anchored arm reads against production's `recipe-short` at twenty seeds. Every task but the corpus build and the rounding readout is ex-2.2.3's code, loaded from its module unchanged, so the placement statistics mean what they meant there.

The grammar under each rounding

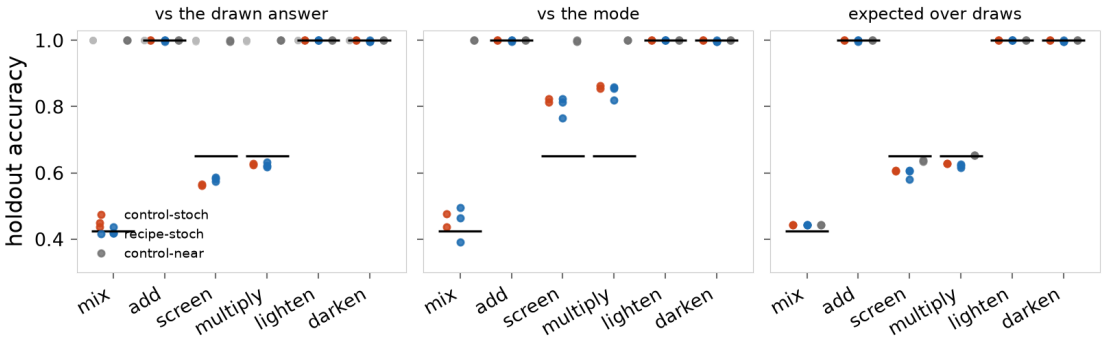
With no model in the loop: how much of each op rounds, the exact-match ceiling a stochastic target imposes (the mean over unordered pairs of the mode's probability, which is the most a predictor that knows the rule can match a drawn answer), and the redder-than-both count of each op's lines, where under stochastic rounding an answer counts with its probability.

op	pairs that round	EM ceiling	ceiling on rounded pairs	redder (nearest)	redder (stochastic, expected)	change
`mix`	87.1%	0.425	0.339	4,920	4,870	-1.0%
`add`	0.0%	1.000	1.000	2,165	2,165	+0.0%
`screen`	82.9%	0.651	0.579	1,427	1,634	+14.5%
`multiply`	82.9%	0.651	0.579	3,893	4,703	+20.8%
`lighten`	0.0%	1.000	1.000	0	0	—
`darken`	0.0%	1.000	1.000	4,258	4,258	+0.0%

The six ops under the two rounding rules. The ceiling is 1 on the three ops that never round; on the other three it is the mean mode probability over all unordered pairs, and over the pairs that round. Redder-than-both counts are over each op's 46,656 lines.

Exact match against a moving target

Ex-2.2.3 reads holdout exact match against the corpus answer. On a stochastic corpus that answer is one draw, so the statistic has a ceiling below 1 on the rounded ops however well the model knows the rule. Three readings of the same eval pass, all on held-out pairs: exact match against the drawn answer (what the production statistic is), against the rule's mode (the nearest answer, which a stochastic model should still name), and the *expected* exact match, which is the chance a fresh draw of the line would match the model's guess and is what the drawn-answer statistic converges to over many draws.



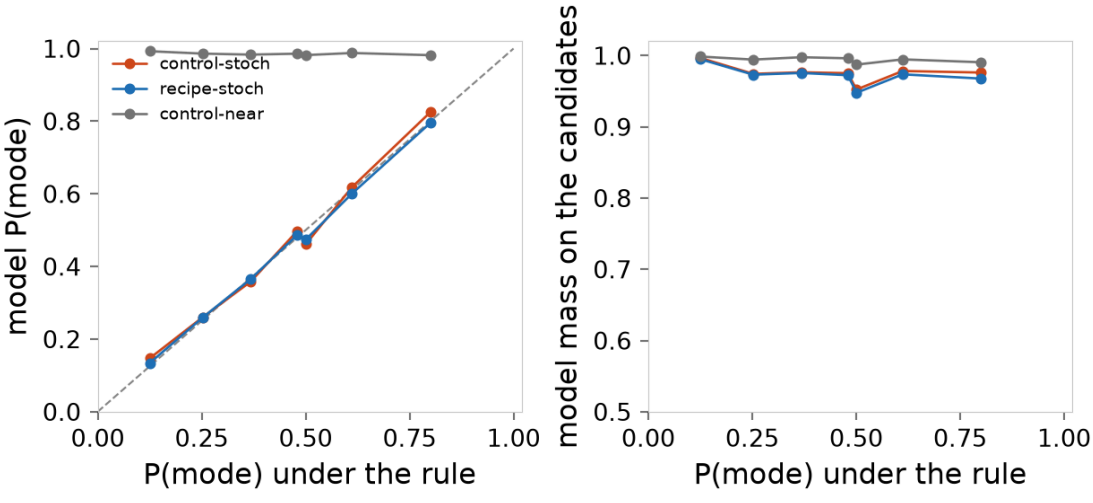
Three readings of holdout accuracy, per op and condition. *Each mark is one seed. Left: exact match against the corpus answer, the production statistic. The black tick is the stochastic ceiling of the op (its mean mode probability); pale marks are production's* `control-short`*, on the nearest-rounded corpus. Middle: exact match against the rule's mode, the nearest answer. Right: the exact match a model would score on average over fresh draws of each line, which is what the left panel estimates with one draw.*

condition	op	holdout ceiling	EM vs drawn	expected EM	EM vs mode	EM vs mode, rounded pairs
`control-stoch`	`mix`	0.443	0.443 ±0.006	0.443 ±0.000	0.457 ±0.020	0.350 ±0.023
`control-stoch`	`screen`	0.638	0.564 ±0.002	0.606 ±0.001	0.818 ±0.006	0.788 ±0.007
`control-stoch`	`multiply`	0.652	0.627 ±0.002	0.629 ±0.000	0.859 ±0.004	0.826 ±0.005
`recipe-stoch`	`mix`	0.443	0.424 ±0.010	0.443 ±0.000	0.451 ±0.053	0.343 ±0.063
`recipe-stoch`	`screen`	0.638	0.581 ±0.006	0.598 ±0.013	0.801 ±0.029	0.767 ±0.034
`recipe-stoch`	`multiply`	0.652	0.624 ±0.008	0.622 ±0.005	0.845 ±0.020	0.808 ±0.024
`control-near`	`mix`	0.443	1.000 ±0.000	0.443 ±0.000	1.000 ±0.000	1.000 ±0.000
`control-near`	`screen`	0.638	0.998 ±0.002	0.637 ±0.001	0.998 ±0.002	0.998 ±0.002
`control-near`	`multiply`	0.652	1.000 ±0.000	0.652 ±0.000	1.000 ±0.000	1.000 ±0.000

The three readings on the ops that round, held-out pairs. Seed means with half the seed range. The last column restricts the mode reading to the pairs that round at all, where the two corpora differ.

Where the answer mass goes

A model trained on a stochastic target has a reason to spread its answer probability over the two (or up to eight) candidate colors of a line in proportion to their probabilities, which is what "reads the effective geometry rather than the snap" meant in the item. The read is on the rounded eval lines only, pooled over ops and splits: the model's probability on the mode against the mode's true probability (a calibrated model sits on the diagonal; a nearest-trained one sits near 1 whatever the ceiling), and the model's total mass on the line's candidate set (near 1 for any model that has learned the rule, whichever way it rounds).



Calibration of the answer distribution on rounded lines. *Lines binned by the true probability of their mode answer (the line's exact-match ceiling), pooled over ops, splits and seeds; each mark is a bin with at least twenty lines, placed at the bin's mean ceiling. Left: mean model probability on the mode; the dashed line is perfect calibration. Right: mean model mass on the line's candidate set, the colors the stochastic rule can produce.*

condition	op	KL(rule ‖ model) ↓	model P(mode)	rule P(mode)	mass on candidates ↑
`control-stoch`	`mix`	0.249 ±0.001	0.325 ±0.006	0.334 ±0.000	0.968 ±0.002
`control-stoch`	`screen`	0.129 ±0.000	0.591 ±0.004	0.577 ±0.000	0.978 ±0.000
`control-stoch`	`multiply`	0.151 ±0.012	0.593 ±0.004	0.570 ±0.000	0.974 ±0.001
`recipe-stoch`	`mix`	0.239 ±0.028	0.321 ±0.018	0.334 ±0.000	0.965 ±0.003
`recipe-stoch`	`screen`	0.136 ±0.018	0.567 ±0.023	0.577 ±0.000	0.977 ±0.003
`recipe-stoch`	`multiply`	0.158 ±0.017	0.580 ±0.009	0.570 ±0.000	0.970 ±0.001
`control-near`	`mix`	5.127 ±0.136	0.983 ±0.003	0.334 ±0.000	0.990 ±0.003
`control-near`	`screen`	2.262 ±0.072	0.989 ±0.000	0.577 ±0.000	0.997 ±0.001
`control-near`	`multiply`	2.259 ±0.026	0.982 ±0.003	0.570 ±0.000	0.994 ±0.001

The same readings as means over the rounded held-out lines of each op. KL is from the rule's answer distribution to the model's, over the palette; it is zero for a model that reproduces the rule's spread and grows as the model commits to one level.

Anchoring on the stochastic corpus

The labeller and the anchor are ex-2.2.3's; only the corpus changed. The placement statistics of the anchored arm against production's `recipe-short` at twenty seeds, on the `mix` probe lines, plus the two task-cost reads: holdout exact match on `mix` (whose held-out lines include rounded pairs) and against the mode.

condition	seeds	m_line ↑	r^2 grading ↑	contrast ↑	lead ↓	latch π ↓	ᾱ op1 ↓	retention ↑	holdout EM mix
`recipe-stoch`	3	0.391 ±0.001	0.884 ±0.013	0.867 ±0.009	0.848 ±0.031	0.021 ±0.003	0.077 ±0.030	0.986 ±0.010	0.424 ±0.010
`recipe-short` (production)	20	0.392 ±0.020	0.886 ±0.044	0.868 ±0.019	0.826 ±0.044	0.018 ±0.006	0.083 ±0.056	0.993 ±0.012	0.999 ±0.002

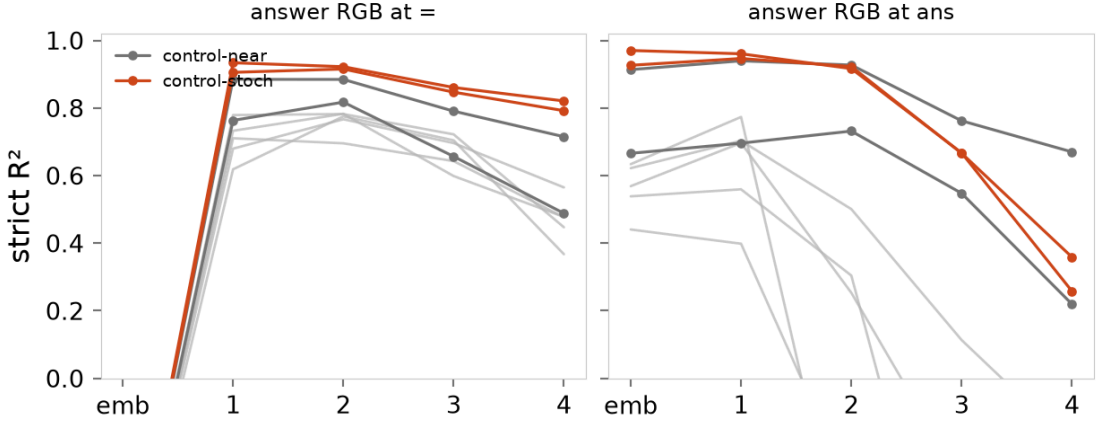
Ex-2.2.3's gated statistics on the recipe, stochastic corpus against production. Arrows are the direction the gates read as good. Seed means with half the seed range.

condition	op	red acc clean	red acc projected ↓	non-red deficit ↓	redder acc clean	redder acc projected
`recipe-stoch`	`mix`	1.00 ±0.00	0.04 ±0.05	0.001 ±0.001	1.00 ±0.00	0.98 ±0.01
`recipe-stoch`	`multiply`	0.88 ±0.01	0.12 ±0.05	0.016 ±0.008	0.78 ±0.03	0.54 ±0.05
`recipe-short` (production)	`mix`	1.00 ±0.00	0.01 ±0.04	0.026 ±0.035	1.00 ±0.01	0.94 ±0.07
`recipe-short` (production)	`multiply`	1.00 ±0.00	0.08 ±0.11	0.012 ±0.012	1.00 ±0.00	0.61 ±0.12

The eval contract's suppression reads under the full projection, on the probe lines (nearest-rounded, as production's). Red lines have dose ≥ 0.8; non-red ≤ 0.2; redder lines have an answer redder than both operands.

The answer's geometry

Question (b) of the item: does a stochastic target make the answer's representation more graded? The cube probes are ex-2.2.3's: strict-holdout ridge R^2 of each slot's RGB at every (slice, position) site, on `mix`'s on-grid probe lines, where no rounding ever happens. The read is the answer's own RGB at `=` (the prediction site) and at the answer position, un-anchored control on each corpus, with production's full-length `control` pale.



Strict-holdout R^2 of the answer's RGB, by slice, at `=` and at the answer position. *Mean over the three channels; one line per seed. Production's `control` ran twice as long as the two pilot arms.*

What we make of it

A model trained on a coin-flip target learns the coin. Its answer distribution mirrors the one the rule defines, so the ceiling the item asked about is a fact about the corpus, and the model sits on it. That settles question (a).

Exact match against the drawn answer is the wrong statistic twice over: it is capped, and it carries the noise of a single draw that every seed shares. Read instead the expected exact match against the holdout ceiling, the calibration of the answer mass (P(mode) against the same quantity for the rule, or the KL), and, on ops without ties, exact match against the mode. The off-grid channels of `mix` are always half-way ties, so there the mode read is a coin toss and the calibration read is the one to keep.

The corpus changes nothing on the prompt side. The labeller reads operands, the anchor pulls the prompt span, and the placement statistics on the stochastic corpus match production to the third decimal.

The answer side does move. Every clean accuracy on a rounded line falls to the level a calibrated argmax can reach, and the suppression reads follow it down. They are still a deficit measured from clean, which the contract already handles, and the redder lines go on holding their answers under `projection` at the same ratio as production.

Redder-than-both counts on `screen` and `multiply` rise by a fifth, because a line that would round down under nearest rounding sometimes rounds up. An E3 on those ops would have more lines to read and the same finding.

Question (b), whether the representation of the answer becomes more graded, is still open. The cube probes come from ex-2.2.3, which runs them on the on-grid lines of `mix`, where nothing rounds. A pilot that could answer it would probe the rounded lines of `screen` or `multiply` and compare the spread of the answer distribution against the raw value: a graded representation would show up as a smooth function of the odds of the coin.

For D2.2 we would keep nearest rounding: it keeps every accuracy read simple, and the anchored-op experiments read the prompt side, where the corpus makes no difference. What the variant adds, a calibrated answer distribution, is not something those experiments ask about. It earns a place when a question is about the representation of the answer itself, and then it comes with the reads above and probes on the lines that round.

Method notes

- **Corpus.** `sca.data.ops.sample_corpus(..., rounding="stochastic")` and `eval_sets(..., rounding="stochastic")`: the (op, pair) sequence and the eval lines are the production ones at seed 0; each rounded line's answer is one draw from its own stream. The probe lines keep the nearest answer: they read the prompt's geometry, and on `mix` they are on-grid lines that never round.
- **Runs.** The d64-L4 nGPT and the ex-2.2.3 training step, fifty epochs (1,650 steps), the either-slot labeller at the per-slot rate; the recipe is $\lambda = 0.1$, $\tau = 0.1$, anti-subspace $2.5 \to 0.3$ by 90%, annealed anchor. Seeds 0–1 (controls) and 0–2 (recipe).
- **Rounding readout.** Per eval line, the rule's answer distribution is the product of its per-channel two-point distributions (`answer_dist`); the ceiling is the mode's probability (`mode_prob`). The model's distribution is its softmax at the pre-answer position restricted to the 216 palette tokens, renormalized for the KL only.
- **Everything else** is ex-2.2.3's: the eval pass, the eval contract's scorer, and the cube probes, run from that module's functions on this pilot's checkpoints.

---

1. Kullback–Leibler divergence, in nats: how much surprise you take on average by predicting with the model's distribution when the rule's is the truth. Zero means the two match. ↩