Ex 2.2.4: a scouting round before the anchored-op experiments

> Scouting pass over a more varied set of operations, on the grid alone, so that the removal reads of the anchored-op experiments are easier to interpret. The hue, saturation, and value ops spread their answers through the color cube, and they are where the answer moves furthest once the red operand loses its red; but a small change in the red operand reaches their answer no more often than it does for the current ops, and they are the first ops where operand order matters. Three commutative ops spread and are sensitive to both operands: `difference`, `exclusion`, and `mix` done in HSV. We propose a table with all six added and `add` dropped.

Observations

Each line below is a read on the grid, together with the statistic it rests on. None of them is a result. The next preregistration will adopt what it needs from here and check it on trained models.

- The op set: every candidate spreads its answers more evenly than the current ops do. A one-level change in the red operand reaches the answer on 61% to 100% of red lines for the commutative candidates and on 37% to 55% for the one-attribute HSV ops, the range `screen` and `multiply` sit in. Zeroing the red operand's R runs the other way: the answer moves furthest on the HSV trio. We propose table A+: drop `add`, add `difference`, `exclusion`, and `hsvmix`, and carry `hue-hsv`, `sat-hsv`, and `value-hsv` as a marked subset, the first ops in the grammar that read operand order.
- The one-level rule of E8 responds to rounding as much as to the op: `mix` counts as dependent on 61% of its own red probe lines and 49% of the shared draw. Under stochastic rounding the rule is graded and reads 0.50 on both sets, and no op moves by as much as 0.1, so the ranking of the ops is the same either way.
- What we make of it: the removal statistic should be a distance rather than exact match, scored on lines where zeroing the red operand's R moves the answer far, and the grammar should adopt stochastic rounding and the whole-line labeller from the two pilots, for comparability with M3.

How to read this

This is the scouting round planned in the backlog item: one notebook, the cheapest version of each question, no gates and no verdicts. Each section says what we ran, shows one figure or table, and says whether it changes the design. Sections land as their questions are run, and this one covers the op set. The tied-readout diagnostic and the τ against λ_a trade are still to come. The answer-drawing labeller ran as a pilot of its own, beside a pilot of stochastic rounding, and the closing section reads both.

Ex-2.2.3 anchored *red* on a grammar of six operations, then measured removal: does the model still answer correctly once the *red* axis is projected out of the residual stream?[1] Removal came out partial on four of the six ops. E8 traced that to the ops themselves: on the saturating ops, most red lines have an answer that would be the same even if the red operand were a little less red, so a model that has lost *red* can still answer them.

The diverse-op todo item proposes adding ops whose answers spread through the cube and depend on both operands, such as the hue, saturation, and brightness blend modes, alongside some of the saturating ones, so that the removal tests have both kinds to read.

The op set

What we ran. We took nine candidate ops, each defined on the same 216-color grid and snapped to it the way the six ops of ex-2.2.3 are, so every line still has an answer in the vocabulary.

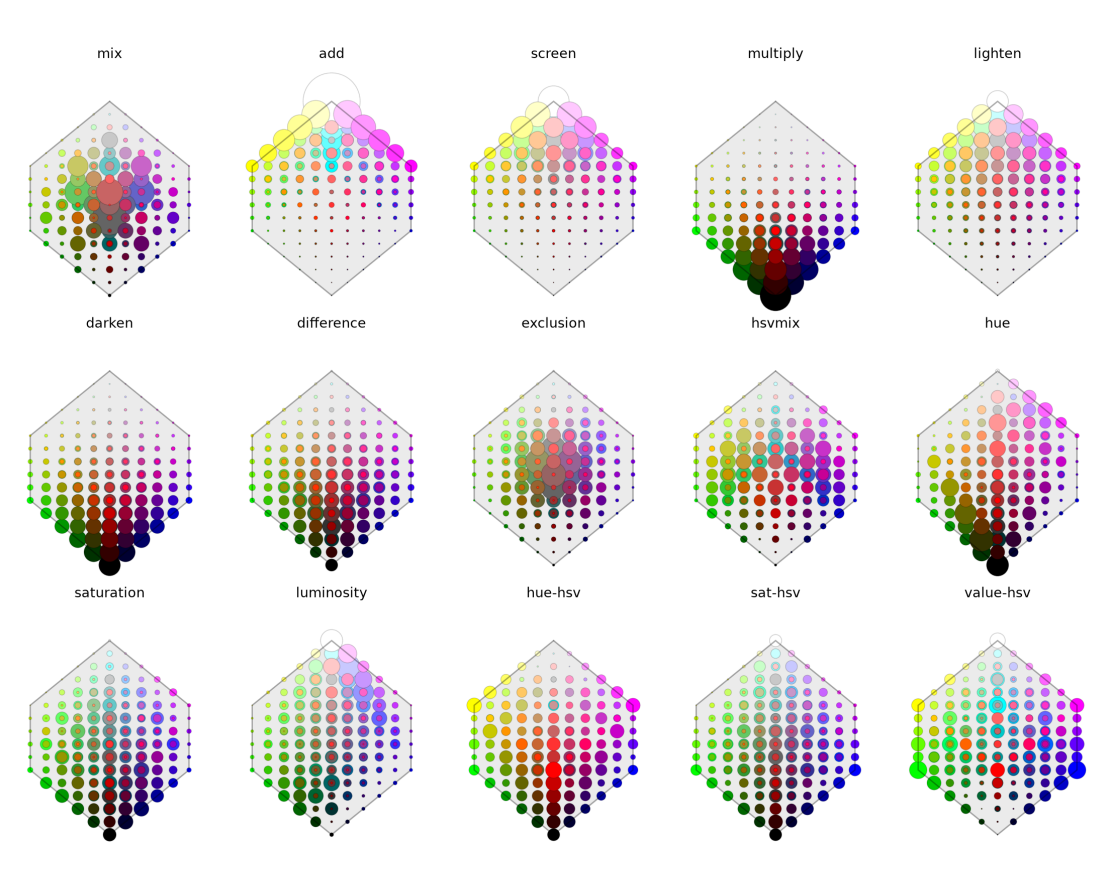
Three of them work channel by channel and are commutative: `difference` and `exclusion`, which are the blend modes of those names, and `hsvmix`, which is `mix` done in HSV (the average hue, saturation, and value). The other six take one attribute from the second operand and the rest from the first: the hue, saturation, and luminosity modes of the W3C compositing spec, as in Photoshop, and the same three in HSV proper, as in Krita.

op	rule	commutative	on grid	redder than both
`mix`	(x + y) / 2	yes	13%	4,920 (11%)
`add`	min(x + y, 15)	yes	100%	2,165 (5%)
`screen`	15 − (15 − x)(15 − y) / 15	yes	17%	1,427 (3%)
`multiply`	x · y / 15	yes	17%	3,893 (8%)
`lighten`	max(x, y)	yes	100%	0 (0%)
`darken`	min(x, y)	yes	100%	4,258 (9%)
`difference`	|x − y|	yes	100%	13,044 (28%)
`exclusion`	x + y − 2xy / 15	yes	17%	12,907 (28%)
`hsvmix`	mean in HSV: circular mean of H, means of S and V	99%	2%	6,560 (14%)
`hue`	W3C hue: op2's hue at op1's sat and lum	2%	6%	6,921 (15%)
`saturation`	W3C saturation: op2's sat at op1's hue and lum	2%	28%	5,687 (12%)
`luminosity`	W3C luminosity: op2's lum at op1's hue and sat	1%	3%	6,013 (13%)
`hue-hsv`	op2's H at op1's S and V	1%	44%	9,467 (20%)
`sat-hsv`	op2's S at op1's H and V	1%	27%	8,626 (18%)
`value-hsv`	op2's V at op1's H and S	1%	33%	7,597 (16%)

The fifteen ops. *The six ops of ex-2.2.3, then the nine candidates. Commutative is the share of unordered pairs whose answer is the same in both operand orders. On grid is the share of pairs the rule answers without rounding. Redder than both counts the lines whose answer is redder than either operand, the case the labeller never sees.*

Where the answers land

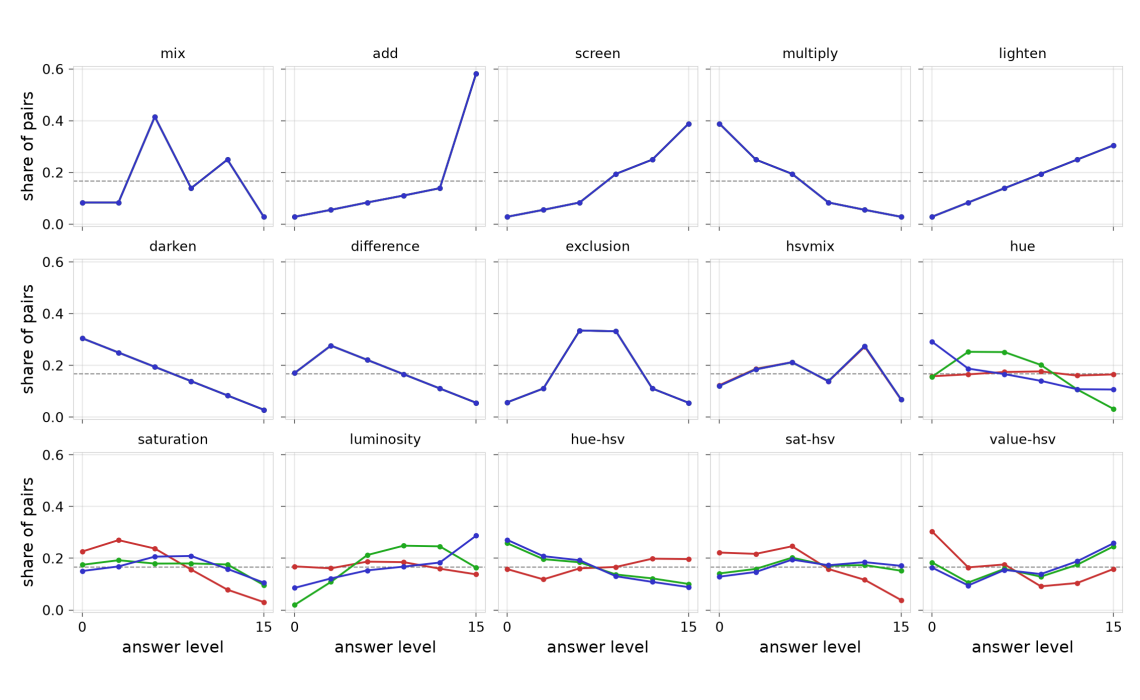
Below is the answer-cloud figure from ex-2.2.3, extended to the candidates. Each panel shows the color cube seen from the gray diagonal, with one mark per grid color, sized by how many pairs answer there. An op that spreads its answers puts small marks everywhere. A saturating op piles them onto a face, an edge, or a corner.



Where the answers of each op land. *One mark per grid color, with area proportional to the number of unordered pairs whose answer is that color. All panels share one scale: a mark that fills its grid cell stands for 600 pairs, out of 23,436. The top row and the first panel of the second row are the ops of ex-2.2.3; the rest are the candidates.*

How evenly the answers spread

We measure evenness two ways. The first is per channel: for each op, the share of answers at each of the six levels of R, G, and B. A flat line at one sixth means the answers cover that channel evenly, and a spike at 0 or 15 means the op saturates. The second is a single number per op, the entropy of the answer distribution over the 216 colors.[2]



Answer level shares per channel. *For each op, the share of unordered pairs whose answer has each grid level in R, G, and B, drawn in the color of that channel. The dashed line is one sixth, which is what an op with even coverage of the channel would show. For the per-channel ops and `hsvmix` the three lines coincide, because the rule treats the channels alike. Panels share the vertical scale.*

Does the answer need the red operand to be red?

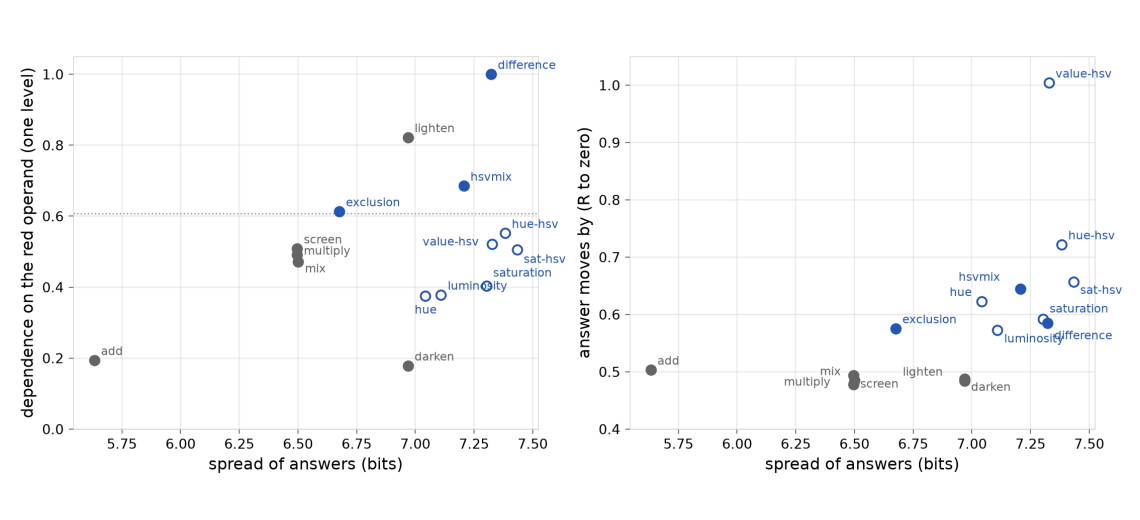
E8 of ex-2.2.3 calls a red line *dependent* when lowering the R of the red operand by one grid level changes the snapped answer. On the dependent lines of every op, removal came out complete. We apply that rule to every op here, both on every red line and on the shared probe draw. We also add a second rule beside it: drop the R of the red operand to zero instead of by one level.

The one-level rule asks whether the op notices a small change in the red operand. The to-zero rule asks whether the answer needs *red* at all. Two more reads go with them. The first is the one-level rule under stochastic rounding, where a line's answer is a distribution over the grid colors nearest its raw value, and the read is the share of that mass that moves.[3] The second is how far the answer moves, in the unit cube, when R drops to zero: for a removal that is meant to make the model fail, the distance is what "fails badly" would be scored against.

Red is the concept D2.1 chose as a stand-in for one a model should not be able to use, and ex-2.2.3 kept it for comparability. Any other color could be anchored the same way, so an op's behavior on red lines is one consideration here, beside spread and commutativity, rather than the deciding one.

op	spread ↑	one level ↑	one level, probe	one level, stochastic	to zero ↑	moves by ↑	reads op1 / op2
`mix`	6.50	0.47	0.49	0.50	1.00	0.48	0.95 / 0.95
`add`	5.63	0.19	0.20	0.19	0.85	0.50	0.89 / 0.89
`screen`	6.50	0.51	0.53	0.50	0.85	0.49	0.93 / 0.93
`multiply`	6.50	0.49	0.47	0.48	0.83	0.48	0.93 / 0.93
`lighten`	6.97	0.82	0.83	0.82	0.82	0.48	0.88 / 0.88
`darken`	6.97	0.18	0.17	0.18	0.83	0.49	0.88 / 0.88
`difference`	7.32	1.00	1.00	1.00	0.98	0.58	0.99 / 0.99
`exclusion`	6.68	0.61	0.62	0.58	1.00	0.58	0.97 / 0.97
`hsvmix`	7.21	0.69	0.68	0.61	1.00	0.64	0.97 / 0.96
`hue`	7.04	0.37	0.36	0.34	0.98	0.62	0.92 / 0.87
`saturation`	7.30	0.40	0.37	0.43	0.95	0.59	0.98 / 0.59
`luminosity`	7.11	0.38	0.39	0.37	0.99	0.57	0.94 / 0.82
`hue-hsv`	7.38	0.55	0.51	0.56	0.85	0.72	0.92 / 0.89
`sat-hsv`	7.43	0.51	0.46	0.52	0.71	0.66	0.97 / 0.68
`value-hsv`	7.33	0.52	0.57	0.61	1.00	1.00	0.97 / 0.70

Spread and dependence, per op. *Spread is the entropy of the answer distribution in bits, where 7.75 is uniform over the grid. One level is the rule from E8: the share of red lines (dose ≥ 0.8) whose answer changes when the R of the red operand drops one grid level, given over all 2,975 red lines, over the 405 red lines of the shared probe draw, and under stochastic rounding, where it is the share of answer mass that moves. To zero drops R to 0 instead, and moves by is how far the answer moves in the unit cube when it does, where $\sqrt{3} \approx 1.73$ is corner to corner. Reads op1 / op2 is the share of lines whose answer changes when that operand is replaced at random. On its own probe set of 365 red lines, `mix` is 0.61 by the one-level rule, 0.50 under stochastic rounding, and 1.00 to zero.*



Spread against the two dependence reads. *For each op, the entropy of its answers in bits against, left, the share of its red lines whose answer changes when the R of the red operand drops one level, and, right, how far the answer moves in the unit cube when R drops to zero. Filled marks are commutative ops and open marks are not. Gray marks are the ops of ex-2.2.3, blue the candidates. The dotted line on the left is `mix` on its own probe set (0.61), the level E8 compared the other ops against.*

What we saw.

The candidates spread; the current ops crowd. Every candidate except `exclusion` has an answer entropy above 7.0 bits, and no current op reaches 7.0. `exclusion` sits between the two groups, with a third of its answers at each middle level.

The channel marginals say why: `add`, `screen`, and `lighten` pile up at 15, `multiply` and `darken` at 0, and the candidates run near the flat line. So on evenness the proposed hue, saturation, and brightness ops do what the item hoped, and so do the three commutative candidates.

`mix` is uneven for a different reason: rounding. Half of its channel sums fall half-way between two levels, and the snap sends every such tie to the even level index, so its answers pile up at 6 and 12 (42% and 25% of pairs, per channel) and only 3% reach 15. Stochastic rounding would split each tie evenly and take that out; the rule itself is as even as `hsvmix`.

The dependence rule from E8 responds to rounding as much as to the op. By the one-level rule, `mix` itself is dependent on 61% of its own red probe lines, and 49% on the shared draw used for the other ops. That is because a one-level drop in one operand moves the mean by half a level, and the snap sends half of those cases back to the original answer. Under stochastic rounding the same read is graded, the share of answer mass that moves, and `mix` comes out at 0.50 on both probe sets: the half-level shift is read as half. No op moves by more than 0.08 between the two roundings, so the ranking of the ops does not depend on the rounding; what the stochastic read takes out is the probe-set artifact.

The hue, saturation, and brightness ops sit at 0.37 to 0.55, no better than `screen` or `multiply`, for a structural reason: each takes one attribute from one operand and the rest from the other, and scaling the R of a pure red changes only its value. So whichever role reads hue or saturation from the red operand never sees the drop. Sensitivity to a small change is where the commutative candidates stand out: `difference` at 1.00, `hsvmix` at 0.69, and `exclusion` at 0.61.

The to-zero read runs the other way. Every op, current and candidate, changes its answer on at least 71% of red lines once the red operand's R is zeroed. What separates the ops is how far the answer moves. The current ops all move by 0.48 to 0.50, about half a channel's range, since the per-channel rules pass part of the drop through to R and leave G and B alone. The one-attribute HSV ops move furthest: 1.00 on `value-hsv`, 0.72 on `hue-hsv`, and 0.66 on `sat-hsv`. An op that copies one attribute of the red operand answers with something far from red once that attribute is gone: a red operand at op2 gives `value-hsv` its full value, and with its R zeroed it gives almost none. So if the removal question is "can the model still mix red", the HSV ops are where a model that has lost *red* would fail hardest, and the one-level rule would not have said so.

The blend modes are the first ops where operand order matters. All six hue, saturation, and brightness variants agree with their own reverse on under 2% of pairs. The saturation ones read op2 on only 59% of lines, since many partners share a saturation. The grammar of ex-2.2.3 is commutative throughout: the probe sets walk every color as op1, and the labeller pools both operands the same way. A non-commutative op puts role information into the grammar for the first time. Natural language has that everywhere, so it is a change we want before M3, and the grid side of it is small: a probe draw that also walks every color as op2, and per-op reads kept separate for the subset, so that anything odd in their behavior can be set aside without touching the rest of the table.

How often the op word is worth reading

The D2.2 design uses a quantity called *op-relevance*: for each line of the anchored op, how many other ops k in the table give the same answer. At 0 the answer identifies the op on its own. At k the op word only rules out the other k. The suppression experiments predict per-line damage from this number, so widening the table changes the prediction for every op in it.

We compare three tables: the one from ex-2.2.3, table A (drop `add`, add the three commutative candidates), and table A+ (table A with the HSV trio, `hue-hsv`, `sat-hsv`, and `value-hsv`). Three ops of A+ read operand order, so the tables count ordered pairs, which moves ex-2.2.3's figures by under a percent.

anchored op	0	1	2
`mix`	95%	4%	1%
`add`	62%	21%	17%
`screen`	42%	41%	17%
`multiply`	62%	37%	·
`lighten`	60%	22%	18%
`darken`	59%	40%	1%

Table current

anchored op	0	1	2	3
`mix`	67%	29%	4%	1%
`screen`	61%	34%	2%	3%
`multiply`	62%	37%	1%	·
`lighten`	51%	42%	4%	3%
`darken`	57%	39%	3%	1%
`difference`	76%	21%	·	3%
`exclusion`	65%	30%	2%	3%
`hsvmix`	61%	33%	5%	1%

Table A

anchored op	0	1	2	3	4
`mix`	63%	30%	4%	2%	1%
`screen`	58%	32%	6%	4%	·
`multiply`	58%	34%	6%	1%	·
`lighten`	42%	43%	9%	4%	1%
`darken`	46%	42%	9%	2%	1%
`difference`	75%	22%	·	3%	·
`exclusion`	64%	30%	3%	3%	·
`hsvmix`	59%	32%	6%	2%	1%
`hue-hsv`	82%	8%	6%	3%	1%
`sat-hsv`	71%	18%	7%	2%	1%
`value-hsv`	74%	16%	7%	2%	1%

Table A+

Op-relevance tables. *For each anchored op, the share of its lines on which k other ops of the table give the same answer. A dot means under half a percent.*

Widening the table makes `mix` much less distinctive. Today it is alone in its answer on 95% of lines; under table A that falls to 67%, because `hsvmix` agrees with it on a third of lines, the ones whose operands are close in hue. We want that: the per-line relevance prediction needs more than one level to read, and `hsvmix` sharing the answers of `mix` on a known set of lines is what supplies it.

`difference` is the most distinctive op in table A. It also has the most redder-than-both lines (13,044, over a quarter of its lines), so it is where the blind-span question (E3 of ex-2.2.3) would get the most lines.

Adding the HSV trio costs the rest of the table little. Each of the three is alone in its answer on 71% to 82% of its lines, with `difference` the most distinctive ops in A+, and the op that loses most is `lighten`, from 51% alone under A to 42%.

What we make of it

Does it change the design? Yes, in three places.

The table should be A+: keep `mix`, `screen`, `multiply`, `lighten`, and `darken`, drop `add`, add `difference`, `exclusion`, and `hsvmix`, and carry `hue-hsv`, `sat-hsv`, and `value-hsv` as a marked subset. We drop `add` because a fifth of its pairs go to white and it is the least sensitive op by either rule. The three commutative additions give the E8 split both kinds of op to read, as the item asks. The HSV trio are there for two other reasons: they are the first ops in the grammar that read operand order, which natural language does everywhere, and they are where the answer moves furthest once the red operand loses its red. Their reads should be reported as a subset, so that any behavior of their own can be set aside without touching the rest of the table, and the probe draw should walk every color as op2 as well as op1 for them.

The removal statistic should be a distance, scored on the lines that can show it. E8's one-level rule reads sensitivity: whether a small change in the red operand reaches the answer, which is what an intervention of that size would show. The question the anchored-op experiments ask is larger: once *red* is gone from the stream, can the model still mix red? On that question the line to read is one where zeroing the red operand's R moves the answer far, and the statistic is how far the model's answer moves under the intervention, against that distance, rather than whether it is exactly right. A model that answers almost-purple for red and blue has kept most of *red*; one that answers gray has not, and exact match cannot tell the two apart. So the prereg should carry the to-zero distance per line as its prediction, and score removal as the answer's distance from the correct one, with exact match beside it for continuity with ex-2.2.3. What "the operand with *red* removed" means in color terms is itself a choice; R to zero is the simplest, and the one E8 used.

The corpus should round stochastically, and the labeller should read the whole line. The two pilots, stochastic rounding and the whole-span labeller, found that neither costs anything on the anchoring side: placement stays in production's band under both, and the whole-line pull has no task cost. Each pilot recommended keeping ex-2.2.3's setting, on the ground that it keeps every read simple. The ground we now prefer is comparability with M3, where a document-level label says nothing about position and a language target is a distribution that no model matches exactly, so exact match is never the statistic. Both changes bring the grammar closer to that, and they fit the reads above: under stochastic rounding an answer is a distribution, so the removal statistic is how much of its mass moves, and the dependence reads are graded the same way. The cost is exact match as the main statistic, and the stochastic pilot names the reads to use instead: expected exact match against the holdout ceiling, and the calibration of the answer mass against the rule.

The next preregistered experiment is then a handover. It runs ex-2.2.3's recipe on table A+ with the new corpus and labeller, against ex-2.2.3's twenty seeds as the reference, and keeps `mix` as the reference op with `hsvmix` beside it, so that a later experiment can make `hsvmix` the reference if it behaves. One arm each with only the corpus or only the labeller changed would say which change moved what, if anything moves.

What we would do differently. The hue, saturation, and brightness ops were the headline of the item, and the one-level rule counts them with the saturating ops: they read one attribute of one operand, whereas *red* as the labeller defines it, r·(1 − g/2 − b/2), is a magnitude that hue and saturation do not see. Read by the to-zero distance they are the ops that need *red* most. We would have started from the to-zero rule and the distance, which are the reads the removal question asks for, and treated the one-level rule as a check that the intervention is large enough to reach the answer.

What this does not settle. Whether the model learns eleven ops as well as it learned six is a training question. E4 of ex-2.2.3 found the operand cube less linearly decodable at six ops than at three, with lines per op as a confound. Eleven ops at the same corpus size would sharpen that confound, so the prereg should decide whether to hold lines per op fixed instead. Nor does it say how a model treats an op that reads operand order, which is why the HSV trio are a marked subset. The scouting questions still to run may move the operating point, but they do not move the table.

Method

Everything in this section is computed on the grid when the notebook runs, using `sca.data.ops`: the six ops of ex-2.2.3 as `OPS` and the nine candidates as `CANDIDATES`, both snapped to the 216-color grid by the same `snap`.

Lines. There are 46,656 ordered pairs of grid colors, each one an (op1, op2) line, and 23,436 unordered pairs, which we use where a statistic is symmetric. A *red line* has dose ≥ 0.8, where dose is the larger of the two operand rednesses, as in ex-2.2.3. The shared probe draw is the one from ex-2.2.3: every color as op1 against 27 partners drawn once from the grid, with the same partners for every op other than `mix`. The probe set for `mix` is the 5,832 closed pairs from D2.1.

Spread is the Shannon entropy, in bits, of the distribution of answers over the 216 grid colors, taken over unordered pairs. **Channel marginals** are the share of unordered pairs whose answer has each grid level in each channel.

Dependence follows E8. For a red line, take the redder operand, lower its R by one grid level (3) or to zero, leave everything else alone, and ask whether the snapped answer changes. We report it as a share of the lines in the set named. Under **stochastic rounding** the answer of a line is the distribution `answer_dist` gives, the product over channels of a two-point distribution on the levels either side of the raw value, and the read is the total variation distance between the distributions of the line and of its lowered version, averaged over the set. **Moves by** is the Euclidean distance in the unit cube between the snapped answers of the line and of its lowered version, averaged the same way. **Reads op1 / op2** replaces that operand with a color drawn uniformly from the grid (seed 0, one draw per line, every fifth line) and asks whether the answer changes.

Commutative is the share of unordered pairs on which the op gives the same answer in both orders. `hsvmix` falls short of 1 on the pairs whose hues are opposite, where the circular mean is undefined and the rule keeps the hue of op1.

Op-relevance is `relevance` from the same module, now over an arbitrary table: for each line, the number of other ops in the table that give the same answer as the anchored op. Ex-2.2.3 counted unordered pairs; the tables here count ordered pairs, so that the ops that read operand order are counted on both orders.

---

1. The *residual stream* is the running vector of activations that each transformer layer reads from and writes back to; it is where we place the anchor. ↩

2. Entropy in bits, so a higher number means the answers are spread over more colors. ↩

3. Stochastic rounding is the grammar variant of the stochastic-rounding pilot: a raw channel value between two grid levels rounds to the upper one with probability equal to how far along it sits, drawn once per line at corpus build. The answer of a line is then a distribution over up to eight colors, and `answer_dist` in `sca.data.ops` computes it. The share of mass that moves between two distributions is their total variation distance. ↩