Ex 2.1.4: spelling the names

> Does geometry inference need one token per concept? Not at 216 colors. We test the same equations but with every color spelled as an opaque four-letter name. There's a small drop in held-out accuracy, but the value subspace survives. Reading names takes most of the network's depth, leaving one layer for the mix to live in. The 27-color grid is too coarse.

In ex-2.1.3, our small transformer inferred the geometry of a color space from mixing equations alone, but every color was a single opaque token: reading an operand was one table lookup, and writing the answer took one softmax. This experiment keeps the same language and setup but makes every color a random[1] four-letter name that the model reads and writes one character at a time:

```
tkzk + qwfd = hjnp
```

The question is whether one token per concept is needed for geometry inference. This language variant sits between the word-level language (ex-2.1.3) and the hex languages (ex-2.1.1, ex-2.1.2), though the ordering by difficulty isn't entirely clear: the hex models seemed to learn some geometry but failed to predict held-out named colors.

The language

We'll use two data grids from ex-2.1.3:

- `v27` is the 3-level grid: 27 colors, with barely enough closed pairs to learn from.
- `v216` is the 6-level grid, 216 colors, where the word-level model had essentially solved the task.

Each grid trains three seeds of the d64-L4 architecture from scratch, on the same `a + b = mix(a, b)` equations as ex-2.1.3, but tokenized differently. A line is now 19 character tokens rather than 6, so the 100,000-equation corpus holds about 3× the tokens; we assume this isn't a significant confound as long as training reaches a low loss.

The model now has to recognize `tkzk` as one unit spread over four positions and attach a value to it. It then has to start writing the name of the mix before any of it is in the context, and continue without losing track.

A few lines from each corpus:

`v27`:

```
pciy + fnlr = dsbb
tzwq + ybup = omdo
oklq + oklq = oklq
szcw + nlgk = pmfh
fnlr + qsko = oklq
```

`v216`:

```
vsqv + tefq = mnih
kjsl + kjsl = kjsl
fbzs + uhnl = rarn
tijc + xwzm = dpnb
emwj + yflq = zqos
```

What we measure

- Greedy accuracy: decode five characters, taking the most likely at each step, and check them against the true name. We also record `p_malformed`: how often the output is not the name of any color.
- Candidate accuracy: teacher-force every vocabulary name plus its closing newline and take the highest scorer. This sets spelling aside, and it recovers the full model distribution over well-formed answers.
- Distribution distance: compare that answer distribution to soft targets built from color geometry — a softmax of negative RGB distance to the true mix at a temperature τ. A model that has learned the geometry should spread its uncertainty over colors near the true mix. This also works on open pairs, where no name is exactly right.
- Distances, as in ex-2.1.3: RGB distance from the name the model chose (the candidate argmax) to the true mix, set against the nearest-name floor and the vocabulary-mean chance.
- Probes: per-depth ridge probes for operand and result RGB, fit on in-training lines and checked for transfer to held-out and open prompts; plus the answer-schedule probe from ex-2.1.2, which teacher-forces whole equations and probes for the three channels of the mix at every position around the answer, at each depth.

Hypotheses

Written down before we looked at any results.

H1. Training still works: seen-pair accuracy is near-perfect on both grids, and malformed completions are rare. Well-formedness is a local statistical pattern (after `=`, four letters and a newline) that a char model of this size picks up easily.

H2. Held-out accuracy will be lower than ex-2.1.3 which scored ≈ 1.0, but well above zero. Multi-token names add ways to fail at binding and emission without removing the co-occurrence evidence the inference runs on. `v27` stays low, as it was at word level. A `v216` collapse to ≈ 0 would mean that multi-token naming on its own blocks the inference.

H3. Where the task is learned, guesses stay close in color space: near-floor distances on open pairs, and misses that are neighbors rather than random names. Geometric closeness held at every vocabulary size at word level, including sizes whose exact match was poor, so we expect it to survive the tokenizer change as well.

H4. The mix has a fixed home even though the answer spans several tokens. No character of an opaque name stands for a channel, so the model cannot compute one channel, write it, and drop it; whatever it knows about the mix has to stay put while the name is being spelled. Concretely, all three RGB channels stay decodable at late depth across the whole emission window, unlike the ex-2.1.2 per-channel stair-step. We also expect the mix to be largely decodable at the pre-answer position, as at word level, though it may come together later in depth, since identifying the operands now takes layers of work that embeddings used to handle.

H5. The answer distribution is shaped like the color geometry. The KL divergence from the distance-softmax target to the candidate distribution, minimized over τ, sits far below the same divergence for a value-blind reference (uniform over names), and the best-fit τ is on the order of the grid spacing. Less confidently, the `v216` distribution should match the targets better than the `v27` one, echoing the geometric trend of the word-level sweep.

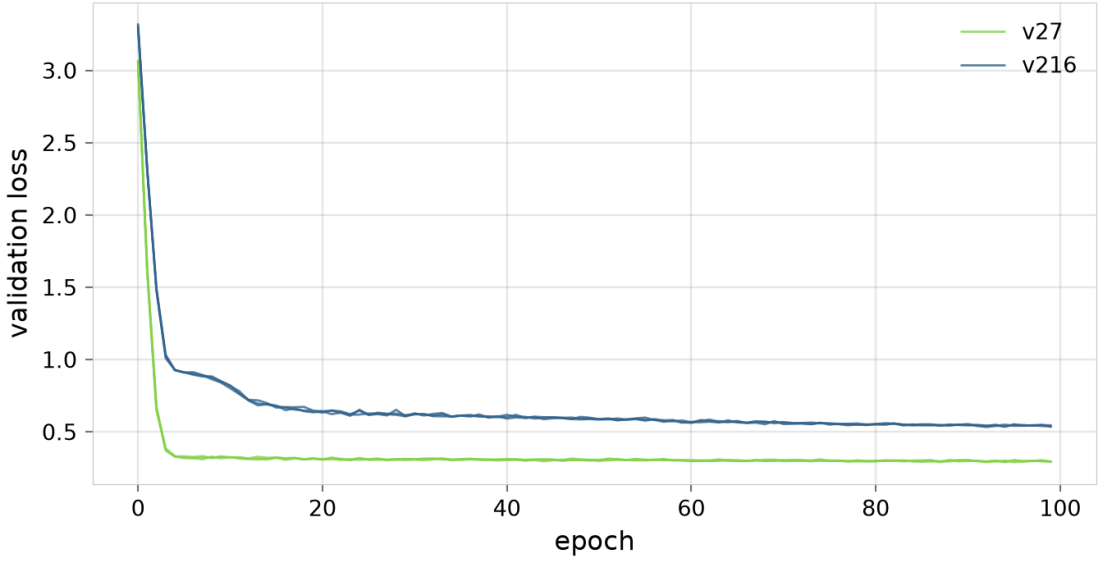
Headline numbers. Mean held-out greedy accuracy: `v27` **0.00**, `v216` **0.91** (word level scored 0.27 and ≈ 1.0 on these grids). Malformed completions: 0.0% on `v27`, 0.0% on `v216`. Held-out guesses land a mean 0.02 from the true mix on `v216` (chance 0.69). The sections that follow fill this in, starting with the corpus accounting.

grid	colors	distinct pairs in corpus	held out	open pairs	corpus tokens
`v27`	27	66	10	302	1,900,000
`v216`	216	2,462	562	20,412	1,900,000

Pair counts per grid. These match those of ex-2.1.3 by construction (same sampler, same seed); only the token count changes (100,000 equations × 19 characters, against × 6 word-level tokens). At `v27` the two pair columns are the entire closed universe: 76 equations exist, and 27 of the 66 training ones are `a + a = a`.

Training

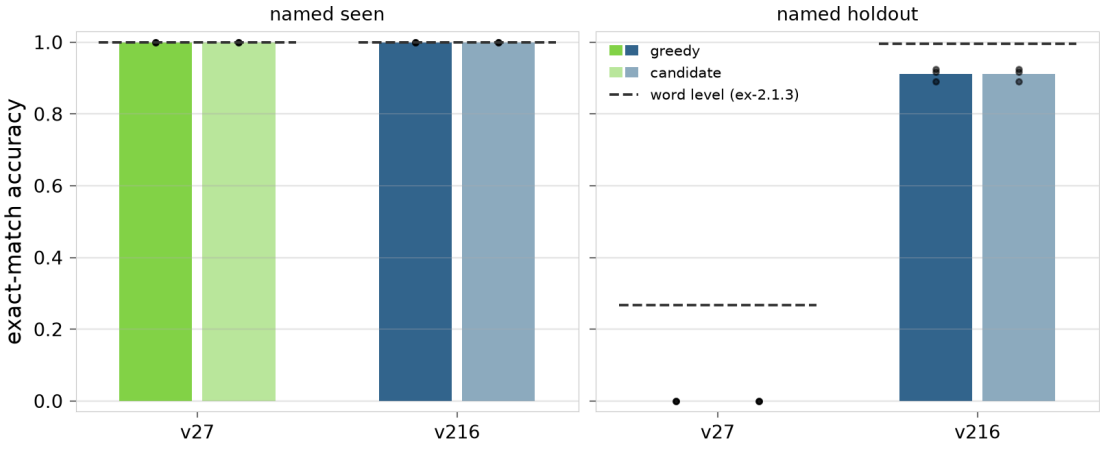
Both grids train smoothly under the same 100-epoch schedule as before. The curves are not comparable to those of ex-2.1.3: the vocabulary here is 30 characters rather than hundreds of names, so chance loss is lower.



Validation loss per epoch, with three thin lines per grid, one per seed. This is per-token loss over characters, so it does not line up with the curves of the word-level experiment; what matters is that each curve converges.

Exact-match accuracy

Greedy accuracy is the end-to-end score; candidate accuracy (the argmax over teacher-forced names) tells us how much of any gap is spelling rather than knowledge. The dashed line over each group of bars is the ex-2.1.3 word-level accuracy on the same pairs.



Exact match by grid and eval set; bars are means over three seeds, dots the individual seeds. Greedy decoding writes freely, so spelling mistakes count as misses; candidate scoring picks the highest-probability vocabulary name and so cannot misspell. The dashed line over each group is the word-level benchmark, the ex-2.1.3 accuracy on the same pairs with one-token names.

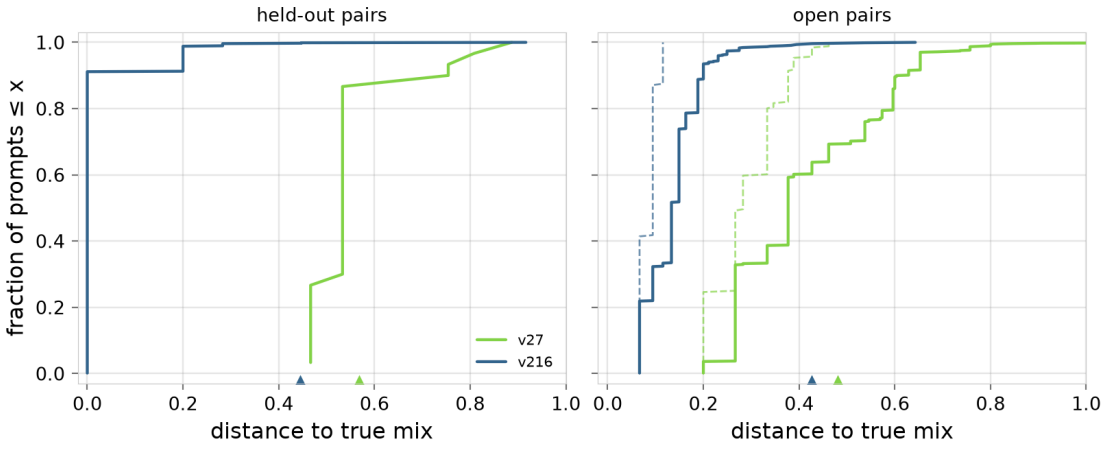
Seen pairs are perfect on both grids (greedy 1.00 / 1.00), and H1 holds: no malformed completion turns up in any eval set or seed, and the probability mass on complete names sums to 1.00. Greedy and candidate accuracy agree everywhere, so it seems that spelling is a settled sub-problem.

Held-out accuracy does drop. `v216` lands at 0.91, below its word-level benchmark of ≈ 0.99 but in line with H2. The misses are similar to the word-level full grid: pooled over seeds, 59 of 68 are wrong in exactly one RGB channel, and 59 of those are off by a single grid level. The model works out roughly the right mix and occasionally rounds one channel to a neighboring level.

`v27` drops much more than H2 expected: zero on every seed (compared to 0.27 for word-level tokens). And the model is confident about it: s_2 = 0.93 on those prompts (confidently wrong; `v216` reads -0.00, well calibrated), and the candidate NLL of the true answer is ≈ 13 nats. The failure rows further down show that it expects nearby names — often one of the operands, though the section after them shows that is what the grid geometry predicts on its own.

How close do the guesses land?

The graded view, as in ex-2.1.3: for every prompt, the RGB distance from the name the model chose (the candidate argmax, so spelling plays no part) to the true mix, drawn as a cumulative distribution.



Distance from the chosen name to the true mix, in unit-cube units, pooled over seeds; a curve that climbs sooner (further left) is better. On held-out pairs the height at distance 0 is the candidate exact-match accuracy. On open pairs no name is exactly right, so the dashed line marks the best reachable distance (the nearest name). The triangles on the x-axis are the prompt-blind constant (always answering the centre of the training answers).

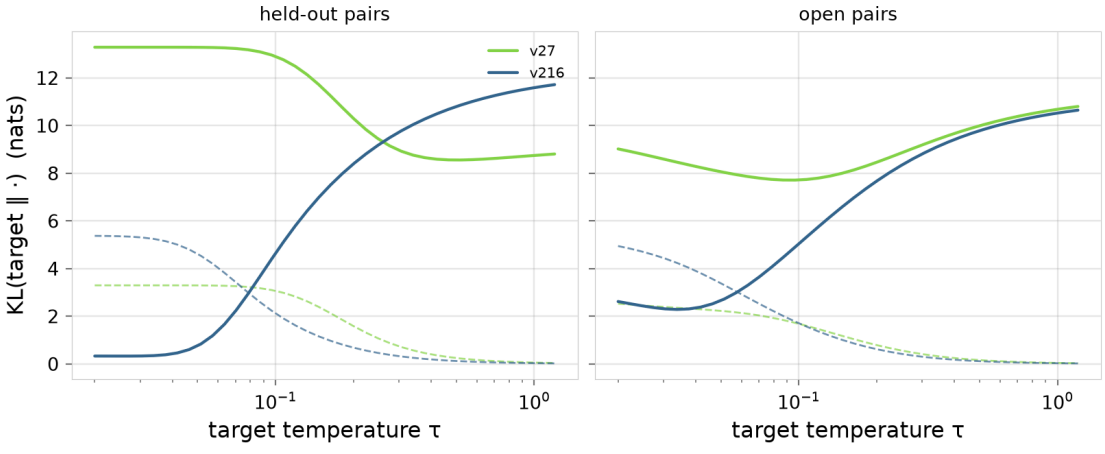
The graded view recovers most of what exact match missed, as H3 predicted.

`v216` tracks its floor: held-out guesses land a mean 0.020 from the true mix, and open-pair guesses 0.14 against a floor of 0.09, a prompt-blind constant of 0.43 and chance of 0.68, landing on the single nearest name 33% of the time.

`v27` lands on the single nearest name 33% of the time against 18% for the constant, and 26 of the 30 held-out guesses sit one grid step from the true mix (0.53), the rest at two steps. So the geometry did come through the tokenizer change on both grids, and what `v27` lost is the last step, picking the exactly right name — but this grid seems to be too coarse.

Is the whole distribution shaped like the geometry?

Exact match and the distance metrics read the top of the model answer distribution. To check the rest of the mass, we build a target distribution per prompt for each temperature τ: a softmax of negative RGB distance from every vocabulary name to the true mix. Then we measure the KL divergence from target to model, `KL(q_τ ‖ p)`, as τ varies. As τ → 0 the target collapses to one-hot on the nearest name; large τ tends toward uniform. A model whose uncertainty is spread over colors near the true mix will fit some middle τ far better than a value-blind model could, and the best-fit τ estimates the scale of its geometric uncertainty. The dashed `KL(q_τ ‖ uniform)` line is what a model giving every name equal probability would score.



How well does the model answer distribution match distance-shaped targets? Solid lines: mean KL(q_τ ‖ model), pooled over seeds and prompts, where q_τ is a softmax of −distance/τ around the true mix. Dashed lines: KL(q_τ ‖ uniform), the score a value-blind guesser gets. A dip well below the dashed line at moderate τ means the model probability mass gathers near the true mix, spread over its neighbors rather than resting on one best guess.

On `v216` the model fits even the sharpest targets well: KL ≈ 0.3 nats at τ ≈ 0.03 on held-out pairs, an order of magnitude under the uniform line, with the best-fit τ near the grid spacing of 0.2, as H5 hoped.

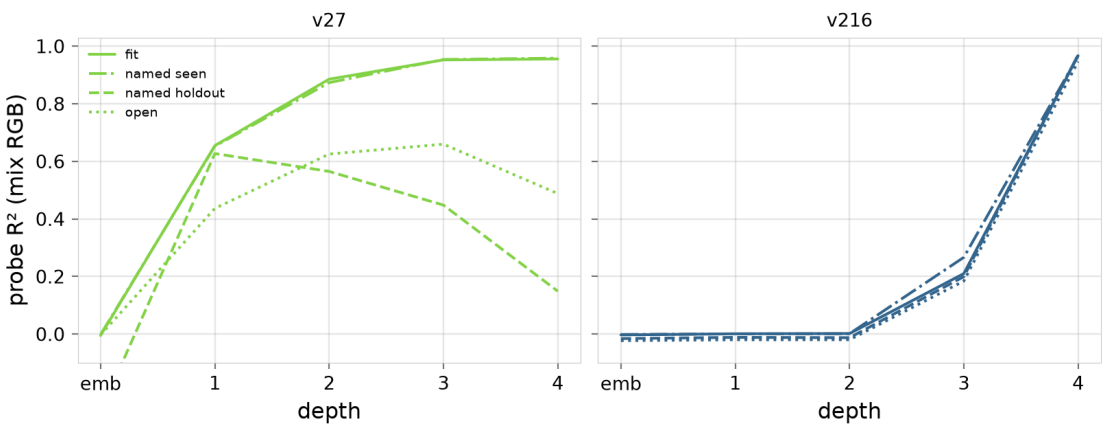
`v27` fits worse than uniform at every temperature, even though the distance curves above showed its guesses are close. `KL(q_τ ‖ p)` is very sensitive to confident error, and the answer entropy of this model is under 0.2 nats, so whenever its chosen name is not the favorite of the target, the divergence shoots up. The metric shows whether the mass sits in geometrically sensible places and whether the confidence is appropriate. So H5 is confirmed on `v216`, but not on `v27`.

Probes: Is the mix computed in value space, and when?

Per-depth probes ask whether the residual stream at the pre-answer position (the space after `=`) already holds the mixed color before any of the answer is in the context; we fit them on in-training lines and transfer them to held-out and open prompts.

Schedule probes teacher-forces whole equations and reads the three channels at every position around the answer, at every depth. On hex answers (ex-2.1.2) this picture was a staircase with eviction; H4 predicts a plateau here.

Per-depth probes

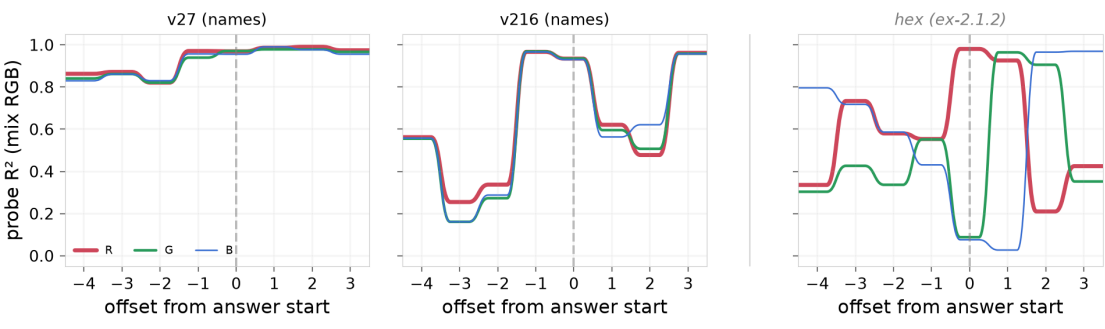


Ridge probes from the pre-answer residual stream to the RGB of the true mix, per depth (0 is embeddings, 4 is the final block), fit on half the in-training probe lines (seed 0) and transferred to the eval sets.

The mix is computed in value space at the pre-answer position on both grids, but it shows up later. At word level it was decodable from depth 1–2 on. Here at `v216` it is essentially absent until the final block: R^2 0.00 at depth 2, 0.21 at depth 3, then 0.97 at depth 4. The earlier layers are reading: the operand probe climbs 0.11 → 0.50 → 0.89 → 0.98 → 0.97 across depths, so turning a four-letter name back into a value is itself a multi-layer computation. The final-block mix transfers with almost no loss to held-out and open prompts (0.97 and 0.95), so it is a real value-space computation.

`v27` matches its behavioral collapse. Mid-stack, a probe fit on training lines transfers respectably to held-out prompts ($R^2 \approx 0.6$–0.8 at depths 1–2, depending on seed); by the final layer, transfer falls to 0.15 (depth 4, seed 0) while the fit set stays at 0.96. The value seems to be computed, but the last block overwrites it with the wrong answer the readout has settled on.

Schedule probes



The answer-emission schedule at the final residual depth (seed 0). Each line is R^2 for one RGB channel across offsets from the first character of the answer; the answer sits at offsets 0–3, and the last characters of the prompt are the negative offsets. Lines are drawn as steps because each offset is a separate character position, with the value held across the position and ramping between; line width tapers R → G → B so an offset where all three agree reads as nested bands rather than as whichever channel drew last. Hex answers (right, from the ex-2.1.2 base-grammar conditions) spell one channel per digit, and the probe found each channel computed just in time and dropped once emitted. Opaque names cannot be written that way, and H4 predicts all three channels stay decodable across the window.

H4 mainly holds: there is no per-channel staircase. In the name panels the three lines run together at every offset: whatever the stream knows about the mix, it knows for all three channels at once.

The "stays fully live" half needs a small amendment. At `v216` the final-depth R^2 is high at the pre-answer position and the first answer character (≈ 0.95), dips to ≈ 0.55–0.6 through the middle of the name, and climbs back to ≈ 0.96 at the last character. Perhaps the mid-name positions look backward to finish a spelling that is already decided, carrying only a thinned copy of the value. So no per-channel eviction, though the whole value fades somewhat during emission; "the result" is sharpest at the pre-answer position or the first answer character.

What the misses look like

A few completions, picked as the widest misses, so worst cases rather than typical ones. Swatches show the actual color of each opaque name.

a	b	true mix	model choice	distance	floor
mela	batx	sphw	oklq	0.53	0.00
nlgk	ybup	mlcn	ybup	0.53	0.00
szcw	idzs	dsbb	fnlr	0.53	0.00
gnhn	tzwq	batx	omdo	0.53	0.00
pmfh	dwba	jwyb	dwba	0.53	0.00

v27 · named holdout: the 5 widest misses (seed 0).

a	b	true mix	model choice	distance	floor
wojx	ywgx	ntcw	nelr	0.28	0.00
xrzc	ojgz	vwfi	lpbc	0.20	0.00
xwzf	uspy	wojx	kiqm	0.20	0.00
pylu	kjsl	crbi	chez	0.20	0.00
kjsl	fwwt	nzxu	kjsl	0.20	0.00

v216 · named holdout: the 5 widest misses (seed 0).

a	b	true mix	model choice	distance	floor
ammm	vern	#658	bgxr	0.54	0.09
mjhw	powd	#265	uudx	0.39	0.09
lrdc	nmst	#588	cvue	0.38	0.12
nkry	tuhj	#588	powd	0.37	0.12
hyqw	pzfd	#f50	pzfd	0.33	0.07

v216 · open: the 5 widest misses (seed 0).

In `v216`, the worst held-out misses are neighbor colors, and its worst open-pair choices hover a step from the floor. The `v27` misses often return an operand (16 of 30 predictions, 53%), which looks like an echo until you count the alternatives. Closure puts each operand one grid level from the mix, so a guesser that picks uniformly from the one-step shell of the mix already returns an operand 40% of the time; against that reference the observed rate is about 1.5 standard errors away, on 30 predictions. So there is no operand echo here, only a small vocabulary.

Discussion

Is one token per concept needed for geometry inference? At `v216`, no. The model reads opaque four-letter names, binds them to values across layers 1–3, computes the mix in value space in the final block, and spells the answer as a whole. Held-out accuracy is 0.91 against the word-level 0.99, and the shortfall is one-grid-level precision misses. The value subspace, the fixed pre-answer home for the result, and the distance-shaped answer distribution all survive the tokenizer change.

It turns out the `v27` grid is too coarse to answer the question. Its closed-pair universe is 76 equations in total, split into 66 for training and 10 held out. Of the training pairs, 27 are `a + a = a`, leaving 39 equations that say anything about how two names combine. There are 3 names that never appear in a mix at all, and one held-out pair is built entirely from them: both operands and the true answer occur in training only as `a + a = a`. No amount of skill reaches that answer from this corpus.

The grading side is equally thin for v27. A guesser confined to the one-step shell of the true mix scores 0.18 exact on these ten pairs, and always answering the center of the training answers scores 0.10, so the word-level 0.27 of Ex-2.1.3 and the 0.00 of this experiment both sit inside the range a model with no naming ability produces.

The same caution applies to the base language in Ex-2.1.1. That experiment measured `named_holdout` = 0, but named sub-grid is this same 27-color regime with a hex distraction on top.

Future experiments might use a v216-like vocabulary. If so, consider:

- Reading names occupied layers 1–3 in this experiment, so the mix only exists from the final layer: a d64-L4 model leaves one layer of residual stream in which the result concept exists at all. Operand concepts are readable from depth 1–2.
- Without hex, all three channels stay decodable together from the pre-answer position through the last character, with a dip mid-name, so an anchored result direction at the pre-answer position would not have to work around the ex-2.1.2 compute-and-evict schedule.

---

1. In ex-2.1.3, we used names like `c05f` for the denser grids. Those are single tokens that decode as hex, but we can't use those names as the multi-character names in this experiment, or the model will just learn the separable channel arithmetic again (as in ex-2.1.1). ↩