Inside LFM3:
from architecture to behavior.

Matt West · Independent research · November 2026

A small language model is still a complicated machine. This project follows LFM3 from its hybrid architecture to the internal signals that change what it remembers, how it answers, and which tool it calls.

The recurring finding is that information and behavior can come apart. A model can recognize harm without refusing, carry a useful improvement in a small weight patch, or change its next word after one memory value is replaced. These are measured handles on particular behaviors, not a complete explanation of the model.

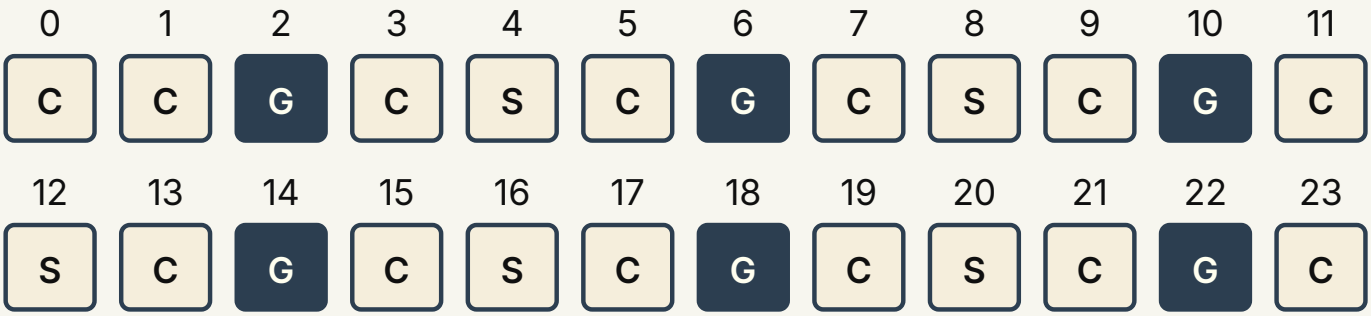
A model with two ways to look back

LFM3 is a family of open-weight models from Liquid AI. The main subject here is its 1.7-billion-parameter instruction model: small enough to inspect directly, but large enough to chat, recall facts, follow instructions, and call tools. The first question was architectural. How does a model that mixes convolution and attention move information through a sentence?

Convolution combines a token with a short stretch of preceding tokens. Attention lets a token read selected information from other positions. LFM3 alternates these operations across 24 layers: 13 short-convolution layers, five sliding-window attention layers, and six global-attention layers. Sliding attention reads a recent window; global attention can reach across the available context. Each attention head, one of the parallel reading units within a layer, has a learned gate that controls how much it contributes.

Local processing needs long-range handoffs.

The 24-layer backbone and the reach of a changed input token.



C short convolution · six preceding tokens

S sliding attention · 256-token window

G global attention · available context

Scroll figure sideways, or open the text alternative below.

▼ Text alternative and data

Layer order, zero-based

Layer	Operator
0	C
1	C
2	G
3	C
4	S
5	C
6	G
7	C
8	S
9	C
10	G

Layer	Operator
11	C
12	S
13	C
14	G
15	C
16	S
17	C
18	G
19	C
20	S
21	C
22	G
23	C

The layer map comes from the released configuration. With attention disabled, a changed token travels at most 78 later positions: 13 convolutions, each reaching six tokens back. Sliding attention extends the measured reach to 1,353. Diagram spacing is illustrative. Method and source.

Stacking short convolutions does extend their reach. It does not make it unlimited. With every attention contribution removed, changing one input token affected hidden states up to 78 positions later, exactly the convolution-only bound. Keeping sliding attention extended that boundary to 1,353 positions. Global attention carried the change to the end of the tested sequence.

Reach is not the same as task performance. The attention-free model failed copying, arithmetic, factual questions, and recall on the small diagnostic suite. Its sentences could look fluent while saying little. Global attention alone preserved secret-word recall even around 1,500 tokens away, yet did not restore ordinary chat. Local processing and long-range reading were cooperating, not interchangeable routes to the same result.

That gave the rest of the project a map. A local signal could be assembled by convolution, transported by attention, and turned into an answer elsewhere. The next step was to find which readers mattered.

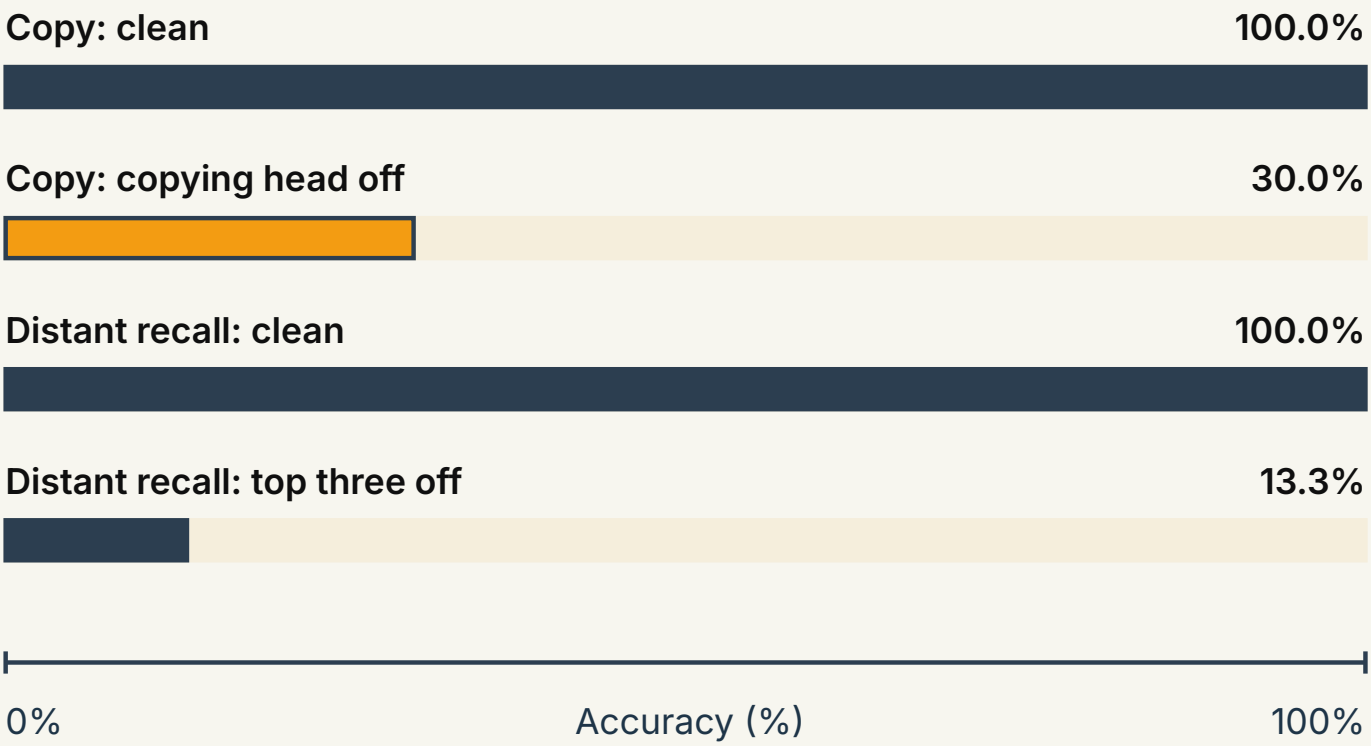
The head that copies, and the heads that do not

One global-attention head stood out when the task was to continue a repeated word list. Turning off head 5 in layer 14 reduced copying accuracy from 100% to 30% on both the selection set and fresh held-out examples. At the copying position, 93% of its attention landed on the earlier occurrence of the answer.

This is an induction-like pattern: find an earlier matching sequence, then reuse what came next. Its gate was more open during copying than during recall. The same head could look at an answer without contributing equally to every task.

Copying has a standout head. Recall is more distributed.

Accuracy on separate held-out copying and distant secret-word tasks.



Scroll figure sideways, or open the text alternative below.

▼ Text alternative and data

Copying versus recall

Condition	Value
Copy: clean	100.0%
Copy: copying head off	30.0%
Distant recall: clean	100.0%
Distant recall: top three off	13.3%

Switching off the copying head leaves 30% copying accuracy. Switching off the three nominated recall heads leaves 13% distant recall but 98% copying. The recall test has 30 distant items and does not establish a universal memory circuit. Method and source.

Recall was less tidy. The most damaging individual recall head did not directly attend to the answer at all. Three nominated heads removed together reduced distant recall to 13%, while randomly chosen three-head sets retained 93–100%. Copying remained at 98%.

Yet ordinary chat could still recall a color and a name with those three heads off. Adding a fourth disabled head sometimes improved the secret-word score again. The effects were not additive, and the template-specific result did not justify calling these the model's memory bank.

The useful distinction was between a reader's attention pattern and its causal contribution. Looking at the answer is a clue. Removing the head and checking what actually changes is a different test.

Recognizing harm is not the same as refusing

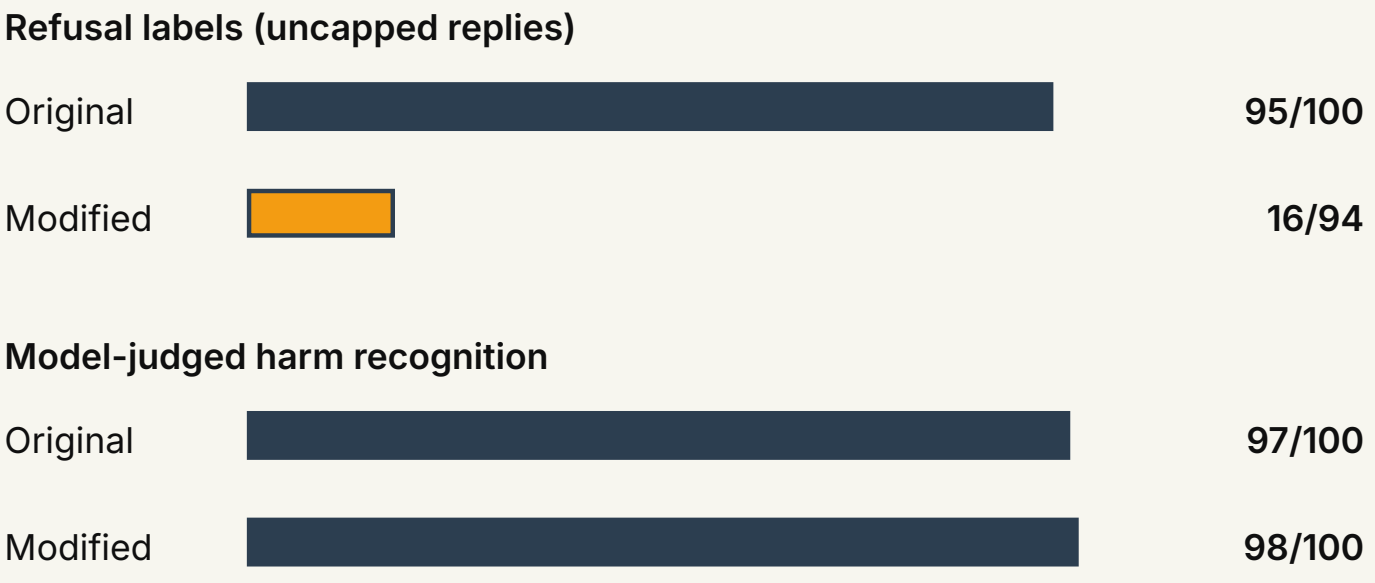
The next question concerned a different kind of separation. If a refusal-related direction is removed from the weights, does the model stop recognizing harmful requests, or just stop saying no?

The intervention weakened refusal sharply. In an initial 20-prompt test, opening-text refusal labels fell from 19 refusals to none. A larger follow-up used a different harmful-prompt set, longer answers, and thinking enabled. There, the original model refused 95 of 100 harmful requests. The modified model refused 16 of 94 uncapped replies.

Harm recognition did not fall with it. A separate d1 model, judging the saved thinking traces, flagged harm in 98 of the modified model's 100 harmful cases. Among its 78 uncapped non-refusing replies, 76 had traces judged to recognize harm. Those labels are not a compliance score. A separate content-delivery score classified 48 of 94 uncapped harmful replies as both recognizing harm and supplying the requested content.

Refusal falls while judged harm recognition remains.

Different measures on the original and refusal-modified models.



Scroll figure sideways, or open the text alternative below.

▼ Text alternative and data

Refusal and judged harm recognition

Model	Refusal	Judged recognition
original	95/100	97/100
ablated	16/94	98/100

Refusal: 95/100 original, 16/94 modified; six capped modified replies are excluded. Model-judged harm recognition: 97/100 original, 98/100 modified. These are separate denominators and measures, not a single safety score. No harmful instructions are reproduced here. Method and source.

Internal readouts agreed with that separation. A direction distinguishing harmful from harmless instructions remained readily decodable, while a later refusal-related signal weakened. Some traces even said the model should decline, followed by an answer that did not.

This does not turn a thinking trace into a faithful account of decision-making. It does show that removing a behavioral barrier need not remove the information associated with it. Safety evaluation has to inspect the answer, not only whether the model can name the risk.

A family resemblance written into the weights

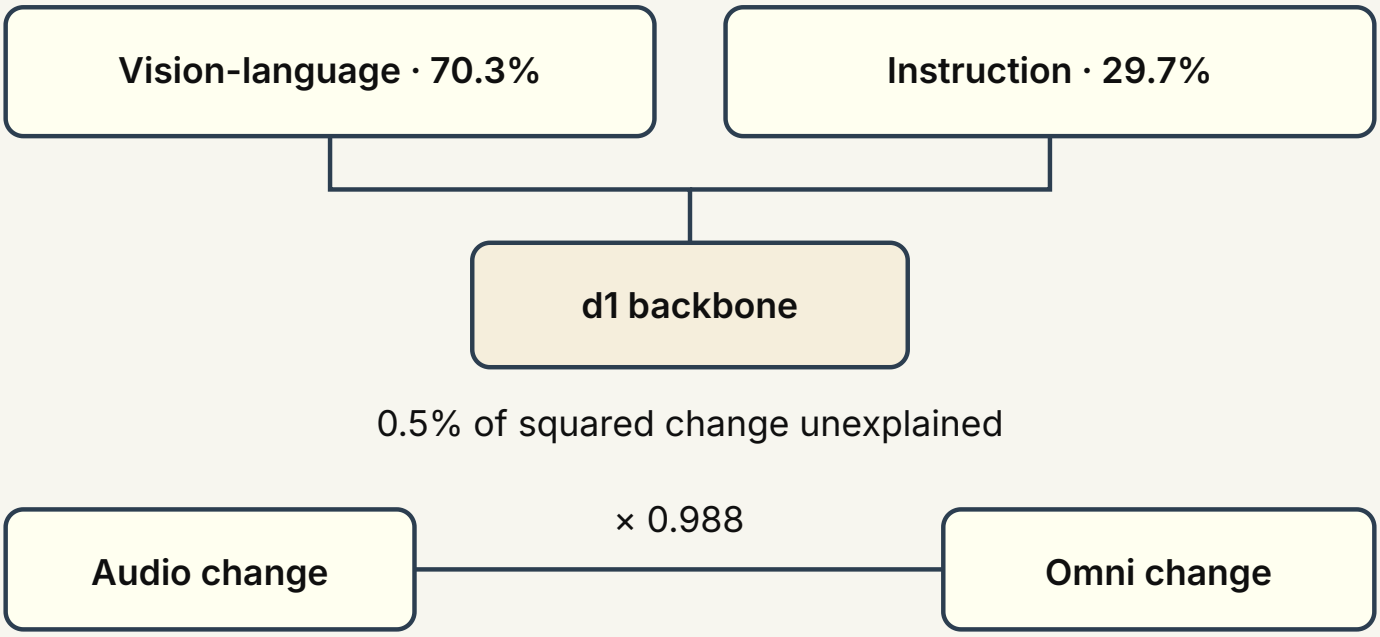
The released family offered another way in: compare models before running them. Their shared backbone tensors have matching shapes, so weight changes can be measured in the same coordinate system.

The Audio and Omni models had a particularly close relationship. Omni's change from the base model was largely explained by Audio's change, with a fitted coefficient of 0.988 and about 17% of the squared change left unexplained. The remaining step was nearly orthogonal to the Audio branch. This fits an Audio-like starting point, rather than a simple sum of the released vision and audio models.

The decision model d1 had a different signature. Across its large backbone matrices, the weights closely matched a 70/30 mix of the vision-language model and the instruction model. The global fit was 70.3/29.7; individual large matrices clustered around 70/30. Only about 0.5% of d1's squared change from Base remained unexplained by that fit.

Two relationships emerge without prompting the models.

Fits to shared backbone weight changes from Base.



Scroll figure sideways, or open the text alternative below.

▼ Text alternative and data

Fits to backbone changes from Base

Fit	Coefficient	Unexplained squared change
d1 ← VL + instruction	0.703 / 0.297	0.5%
Omni ← Audio	0.988	16.8%

Arrows show measured weight relationships, not a verified training history. The Audio–Omni fit and d1's blend fit use different regressions. Shared weight shrinkage matters when comparing branches. Method and source.

Weight geometry cannot establish the manufacturer’s exact training order. A released parent, a nearby unreleased checkpoint, and later blending can leave related signatures. But a uniform interpolation across many matrices is a useful

constraint on the story. It also posed a behavioral question: if almost all of d1 can be described as a blend, does making that blend reproduce its advantage?

The small remainder that changes the decisions

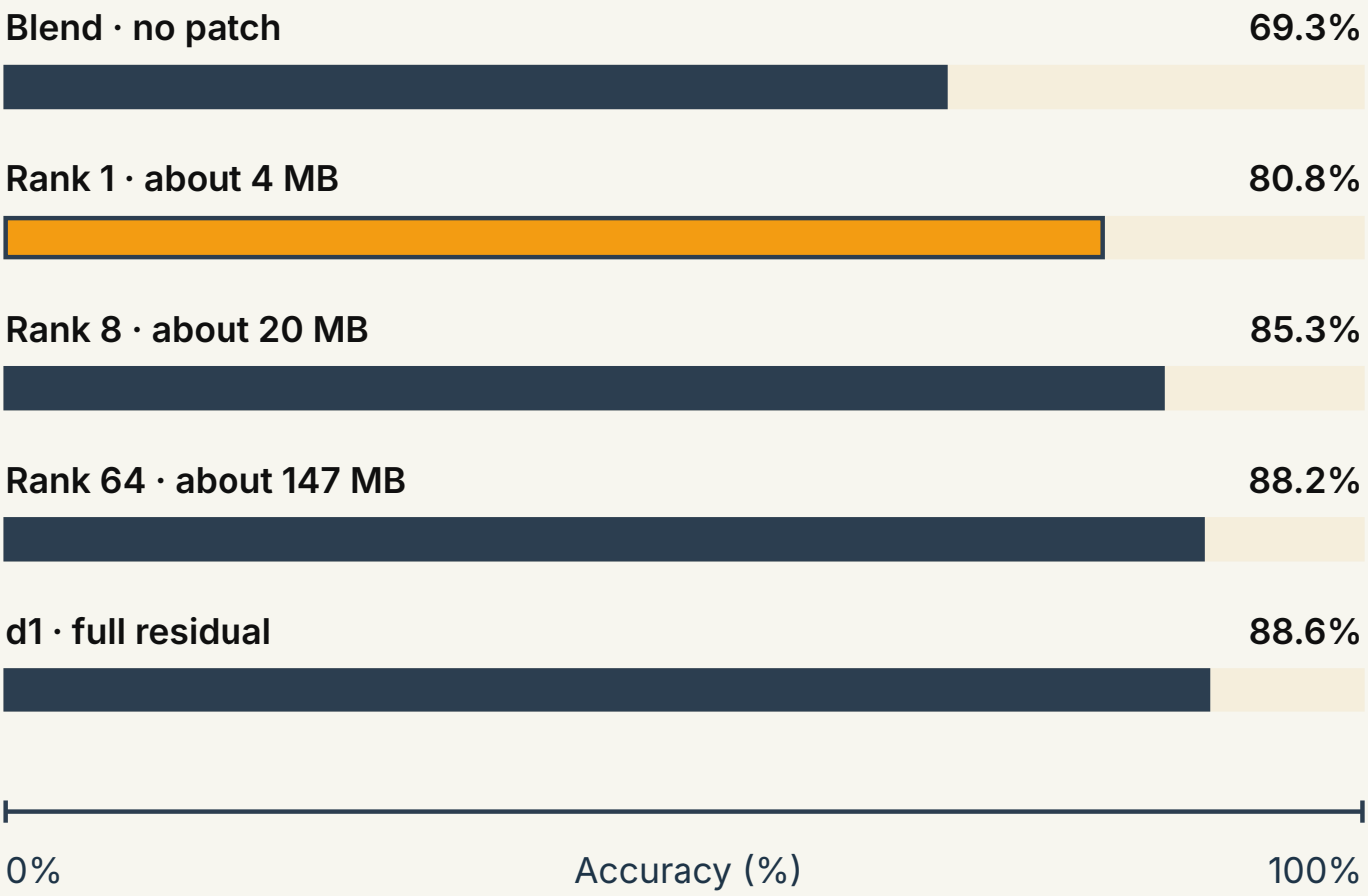
It did not. Inside d1's own decision harness, the 70/30 blend scored 69.3% mean accuracy across five classification tasks. d1 scored 88.6%. The difference was especially large on a 77-way banking-intent task.

Call the leftover weight difference the residual: what remains after subtracting the blend from d1. Adding that residual back in quarters moved mean accuracy from roughly 69% to 81%, 85%, 88%, and finally 89%. The blend described most of the geometry. The remainder carried much of the measured advantage.

Could that remainder be compressed? Keeping only one singular direction per large matrix, together with the complete small-tensor residuals, produced a patch of about 4 MB. It raised the blend to 80.8%, recovering about 60% of its accuracy gap to d1. Eight directions per matrix recovered 83% of the gap; 64 recovered 98%.

A 4 MB patch recovers much of the measured gap.

Mean classification accuracy in d1's decision harness; five tasks, 1,477 items.



Scroll figure sideways, or open the text alternative below.

▼ Text alternative and data

Patch size and decision accuracy

Condition	Value
Blend · no patch	69.3%
Rank 1 · about 4 MB	80.8%
Rank 8 · about 20 MB	85.3%
Rank 64 · about 147 MB	88.2%
d1 · full residual	88.6%

“60%” is recovery of the blend-to-d1 accuracy gap under d1's two-way attention, not a 60% accuracy gain. Rank means directions retained per matrix, not one direction in the whole model. Under ordinary causal attention, the blend baseline is stronger and the patch gains are smaller. Method and source.

The patch also improved the instruction model’s backbone in the same harness, from 68.3% to 80.2%. A control with ordinary causal attention still found a gain, though a smaller one. Part of the patch’s benefit was tied to using d1’s two-way attention setting.

An example made the change tangible. A customer had been charged twice, had not received a parcel, and wanted one charge refunded. The unpatched instruction backbone strongly chose shipping. With the small patch it chose billing, as d1 did. The ticket is ambiguous; the point is the shift in judgment, not an invented gold answer.

The residual’s tail also retained useful performance. So a small patch was sufficient to recover much of the benchmark advantage, but the advantage did not live exclusively in those few directions. Nor had a general improvement to the ordinary chat model been demonstrated.

From word fragments to response mode

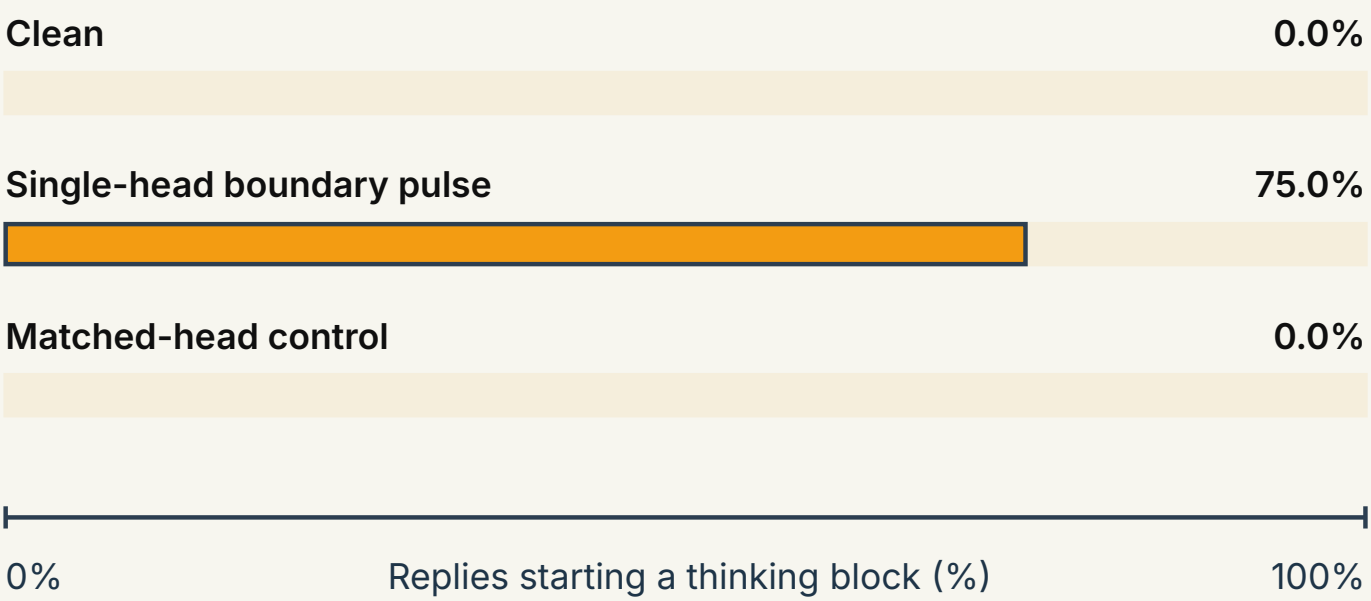
Attention was not the only place to find interpretable handles. The first convolution layer could be decomposed exactly into contributions from the current token and six predecessors.

Word fragments drew especially heavily on the previous token. For mid-word continuations, the average previous-token share was 87.5%, compared with 56.7% at word starts. Fragments such as `berg` in `Gold|berg` made this visible. Replacing the predecessors with random corpus tokens preserved most of the difference. The current fragment was asking to be assembled from what came before; the measurement did not prove that the layer recognized the complete name.

A short-range bracket channel survived a causal check. Pushing it toward the side associated with opening brackets made a closing parenthesis about 2.5 times likelier, relative to the average token. Its memory span was six tokens, not an arbitrary bracketed passage. Several other appealing channel labels did not survive their controls.

An internal pulse can reopen a closed thinking block.

Sixteen fresh prompts with thinking disabled; intervention only at the final prompt position.



Scroll figure sideways, or open the text alternative below.

▼ Text alternative and data

New thinking blocks after a boundary pulse

Condition	Value
Clean	0.0%
Single-head boundary pulse	75.0%
Matched-head control	0.0%

A +2 gate-logit pulse to layer 6, head 2 starts a new thinking block on 12/16 prompts; the matched head starts none. On the locker example, both clean and edited replies answer “flute,” but observed content grows from 2 to 126 tokens. This is a response-mode effect, not improved reasoning. Method and source.

A head intervention changed something more conspicuous. Even with the template’s thinking block already closed, a pulse to one head’s gate at the final prompt position made 12 of 16 fresh replies start a new `<think>` block. A matched head did not reopen it.

The longer path sometimes fixed a payload and sometimes exhausted the answer budget. It was not a universal reasoning switch. But the boundary-only pulse showed that a short internal intervention could set a response trajectory that persisted after the intervention ended.

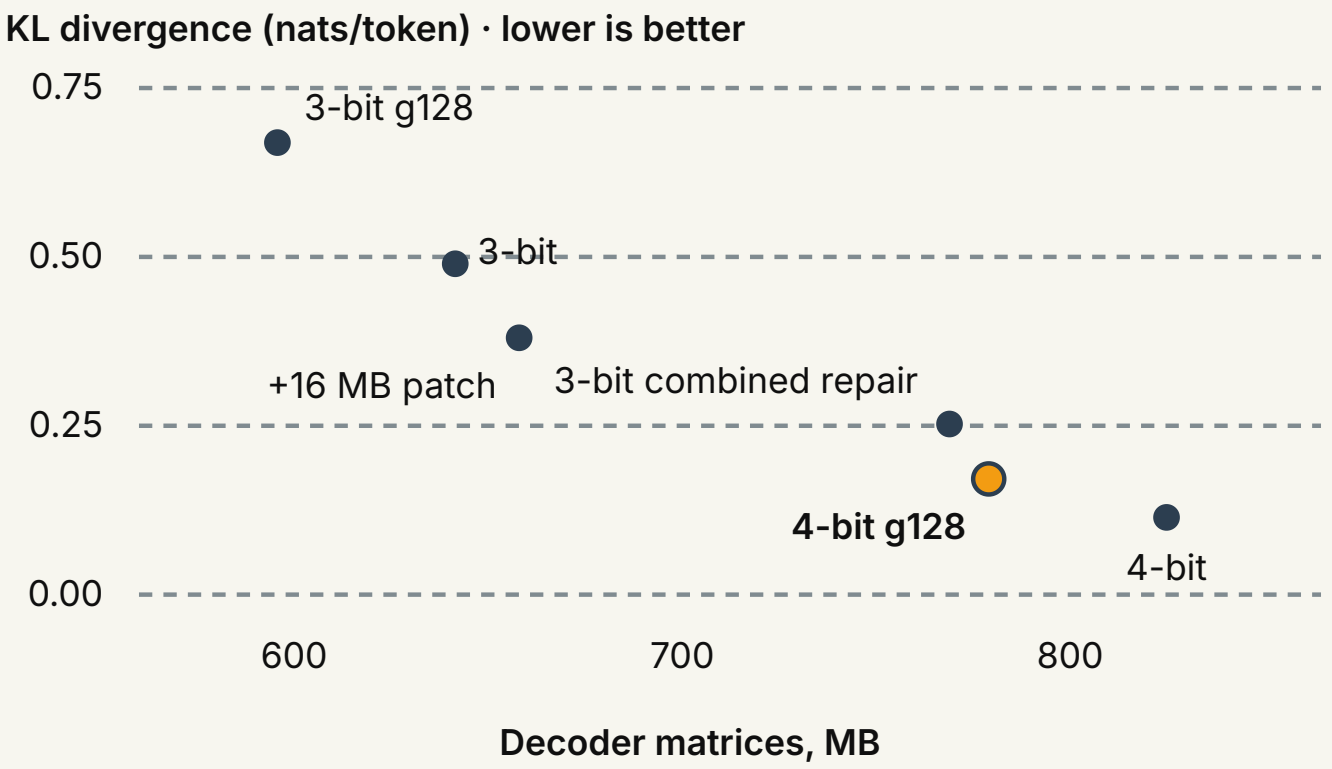
Where fewer bits stop being a bargain

Compression raised a practical question. How far could the weights be quantized, meaning stored at lower numerical precision, before behavior broke? And was repairing a damaged small model better than spending the same storage on more bits?

Four-bit round-to-nearest decoder matrices retained the diagnostic suite’s mean accuracy and modestly increased text perplexity, a measure of prediction difficulty, from 14.98 to 16.30. Three bits pushed perplexity to 21.91 and lost the templated chat task entirely. At two bits, perplexity reached about 1,352 and generated text became largely unusable.

Repair helps within a budget, then another bit wins.

Prediction divergence from bf16 versus decoder-matrix storage, not total model storage.



Scroll figure sideways, or open the text alternative below.

▼ Text alternative and data

Selected quantization formats

Format	Decoder MB	KL divergence
q3_g128	595.7	0.669
q3	641.5	0.490
q3+act8	658.0	0.380
q3+attn_all8+act32	768.9	0.253
q4_g128	779.0	0.171
q4	824.8	0.114

Lower KL divergence is better. The 3-bit activation-weighted rank-8 patch adds about 16 MB and removes 22% of divergence. At about 779 MB, uniform 4-bit group-128 matrices beat every tested 3-bit repair within that budget. These are simulated round-to-nearest formats, not GPTQ or AWQ. Method and source.

Ordinary low-rank approximation of rounding error was weak. Weighting the error by how strongly the model actually used each input channel made the repair more effective. A roughly 16 MB patch removed 22% of the three-bit prediction divergence. Protecting attention at higher precision also bought useful recovery per byte.

There was a stopping point. Near 779 MB of decoder-matrix storage, a plain four-bit format with larger groups beat every three-bit fix that fit the same budget. The repaired two-bit options were worse than similarly sized three-bit ones.

The interesting result was not “patches fix quantization.” It was a storage frontier: repair can be worthwhile inside a narrow budget, but sometimes the better repair is to stop compressing so hard.

Reading the atlas, then editing one memory value

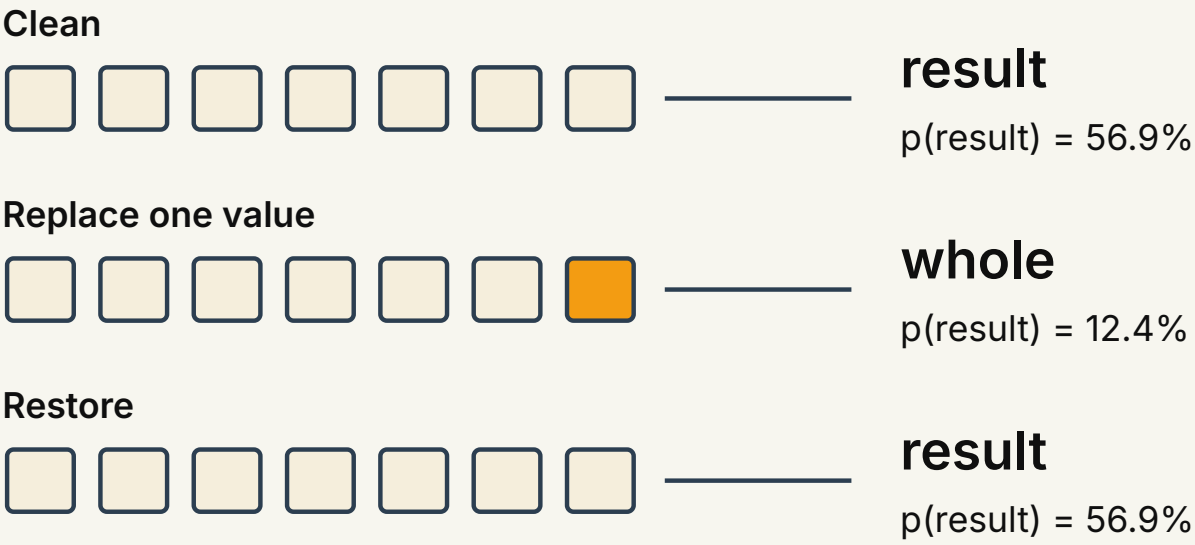
The project next broadened from individual candidates to an evidence atlas. It brought together natural activations, internal readouts, attention sources, and intervention effects for 688 entries: sampled feed-forward and convolution coordinates, plus every attention head.

An internal readout translates an intermediate state into candidate words. A derivative-based readout improved some late-layer rankings of concepts not explicitly named in a prompt. Asked for the capital associated with Brandenburg Gate, the model answered Berlin while the readout also ranked Germany highly. Early readouts remained noisy, often dominated by punctuation. This was not a reliable view of hidden reasoning from the first layers onward.

The atlas was more useful as a way to nominate experiments. One final-layer convolution channel had a strong effect before `result` in an `as a` context. Fresh prefixes showed a smaller, variable effect, and separate lag contributions partly cancelled each other. That suggested testing the actual short-term memory rather than giving the channel a broad concept label.

One changed cache scalar changes the next word.

Saved continuation: “The training improved staff productivity as a”.



Scroll figure sideways, or open the text alternative below.

▼ Text alternative and data

The training improved staff productivity as a

Condition	Next word	p(result)
clean	result	56.9%
source_cue	whole	12.4%
restored	result	56.9%

The prompt tokens stay identical. Replacing one final-convolution history value changes the next prediction from “result” to “whole,” while p(result) falls from 56.9% to 12.4%. Restoring the value restores the observed continuation. Probabilities refer to “result” in every row. This is a saved replay, not live inference. Method and source.

The model first read through as. One scalar in the final convolution cache was then replaced with a value from a matched by context, before feeding the original a. The

next word changed from `result` to `whole`. Restoring the scalar restored `result` and the observed continuation.

That is a small but direct memory-edit demonstration. It does not establish a general consequence feature or improve an answer. Its direct influence expired after the six-token kernel window under fixed continuations. Different generated words could nevertheless carry the consequence further by becoming new input.

A compact controller for complete tool calls

The next step tested whether internal control could select an action rather than a word. Replacing an early whole-token state first changed a generated call from addition to multiplication while keeping the visible operands fixed. The convolution-only part did not reproduce the switch.

That broad transplant was then compressed. A shared fitted basis used three coefficients, supported on 512 of the token state's 2,048 coordinates, to choose among add, multiply, maximum, and minimum. It was an engineered controller, not three discovered native command neurons. All four operations were represented during fitting.

On 16 held-out commands, the internal controller produced 14 correct requested calls, matching ordinary text instructions and the full-state transplant. Ten were newly induced correct switches. The remaining successes included no-ops and an already-target baseline choice; 14 correct calls did not mean 14 switches.

One unchanged input, four generated calls and returned results.

Replay of the first held-out context: a = 4, b = 13. No live model runs.

Saved-output replay · initial input identical in all four runs

Model-generated call

add(4, 13)

Executor returns

17

receipt 9ed4055d

Model's actual final reply


```
{
  "tool": "add",
  "result": 17,
  "receipt": "9ed4055d"
}
```


▼ **Exact native call and internal coefficients**


```
<tool_call>
<function=add>
<parameter=a>
4
</parameter>
<parameter=b>
13
</parameter>
</function>
</tool_call>
```


Coefficients: 0.0000, 0.0000, 0.0000. One shared basis, 512 coordinates.

Model-generated call

multiply(4, 13)

Executor returns

52

receipt 17ca28f7

Model's actual final reply

```
{
  "tool": "multiply",
  "result": 52,
  "receipt": "17ca28f7"
}
```

▼ **Exact native call and internal coefficients**

```
<tool_call>
<function=multiply>
<parameter=a>
4
</parameter>
<parameter=b>
13
</parameter>
</function>
</tool_call>
```

Coefficients: -0.4925, 0.5262, 0.0031. One shared basis, 512 coordinates.

Model-generated call

maximum(4, 13)

Executor returns

13

receipt c87ac261

Model's actual final reply

```
{
  "tool": "maximum",
  "result": 13,
  "receipt": "c87ac261"
}
```


▼ **Exact native call and internal coefficients**


```
<tool_call>
<function=maximum>
<parameter=a>
4
</parameter>
<parameter=b>
13
</parameter>
</function>
</tool_call>
```


Coefficients: -0.8317, -0.1448, -0.2443. One shared basis, 512 coordinates.

Model-generated call

minimum(4, 13)

Executor returns

4

receipt 84887664

Model's actual final reply


```
{
  "tool": "minimum",
  "result": 4,
  "receipt": "84887664"
}
```


▼ **Exact native call and internal coefficients**

```
<tool_call>
<function=minimum>
<parameter=a>
4
</parameter>
<parameter=b>
13
</parameter>
</function>
</tool_call>
```

Coefficients: -0.8693, -0.1595, 0.2320. One shared basis, 512 coordinates.

Saved multiply run. Returned receipt matches the model's final reply. No live inference.

▼ All 16 internal outcomes, including failures

Saved internal calls with the shared formatting follow-up

Operands	Requested	Actual result	Whole task
4, 13	add	add → 17	Pass
4, 13	multiply	multiply → 52	Pass
4, 13	maximum	maximum → 13	Pass
4, 13	minimum	minimum → 4	Pass
8, 15	add	add → 23	Pass
8, 15	multiply	multiply → 120	Pass
8, 15	maximum	maximum → 15	Pass
8, 15	minimum	minimum → 8	Pass
11, 2	add	maximum → 11	Failed call
11, 2	multiply	maximum → 11	Failed call
11, 2	maximum	maximum → 11	Pass
11, 2	minimum	minimum → 2	Pass
9, 6	add	add → 15	Pass
9, 6	multiply	multiply → 54	Pass
9, 6	maximum	maximum → 9	Pass
9, 6	minimum	minimum → 6	Pass

The controller changes the stored operation, not the visible operands. The model generates the native call; an executor returns its actual result and receipt. A shared formatting follow-up is needed for the verified final reply. All four saved examples are available without JavaScript. Method and source.

Completing the round trip needed another instruction. The original final-response requirement failed in both text and internal arms. A separately fixed formatting follow-up, shared by both, asked the model to copy the actual tool, result, and receipt from the returned message. It achieved faithful replies on all 16 calls per arm, with 14 complete intended round trips because the two wrong calls remained wrong.

The receipt was computed from the actual executed call. Copying it demonstrated access to the returned message, not just guessing the arithmetic. The same input could lead to add(4,13)→17, multiply→52, maximum→13, or minimum→4.

The intervention footprint fell by 75% relative to the full transplant. Speed did not improve. This was a portable control interface within a small tested domain, not a new capability or a production advantage.

Following a digit until it becomes a color

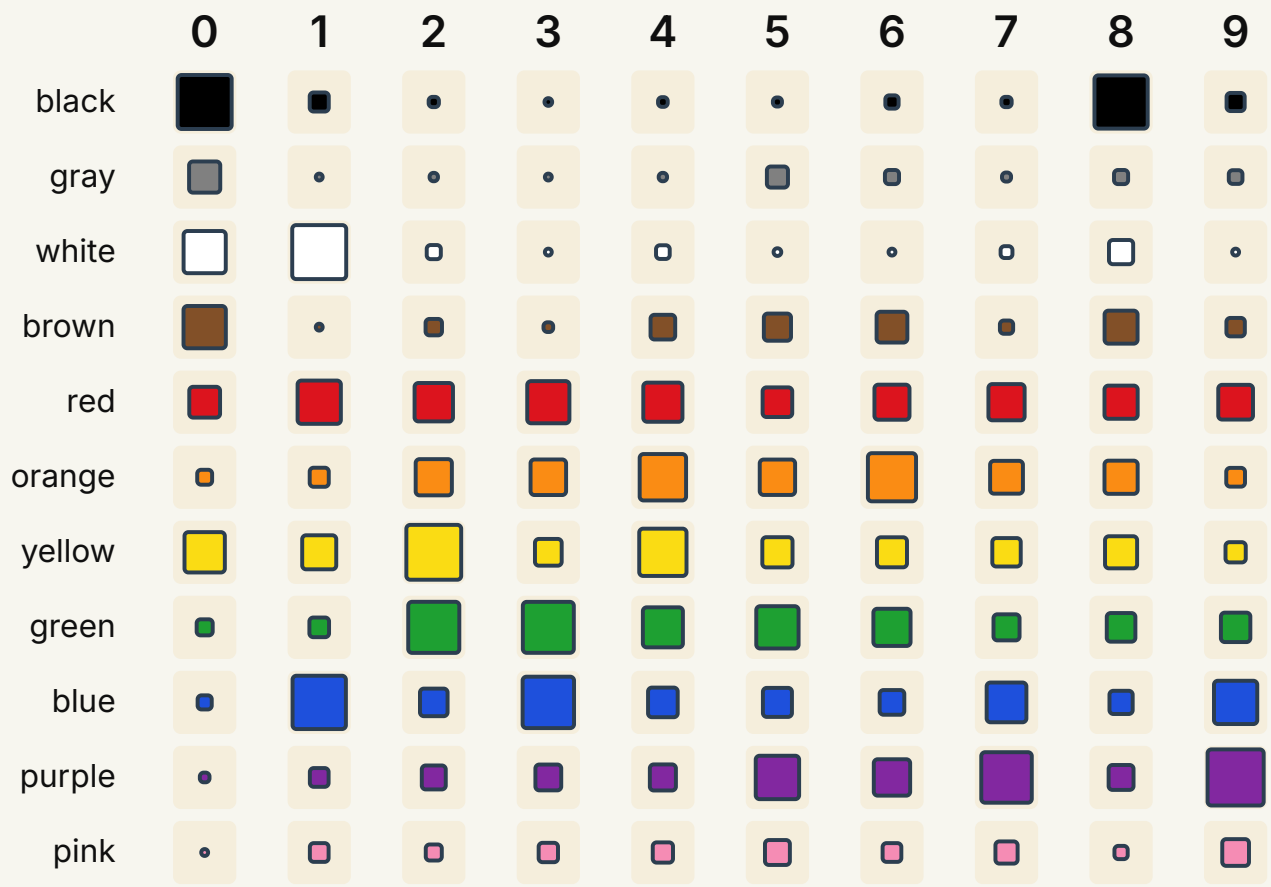
The final study began with an odd but concrete question: does LFM3 associate particular digits with particular colors?

A single answer was not enough. The stronger test asked for colors across 1,000 shuffled digit orders, producing 10,000 sampled answers. A distributional palette emerged. Black was named for 0 in 24.9% of answers and for 8 in 24.2%, but at most 3.1% for any other digit. Purple appeared for 9 in 26.9% and for 7 in 21.8%.

The association is a palette, not one fixed color per digit.

Basic color-word shares; 1,000 sampled answers per digit, shuffled order.

Color-word shares

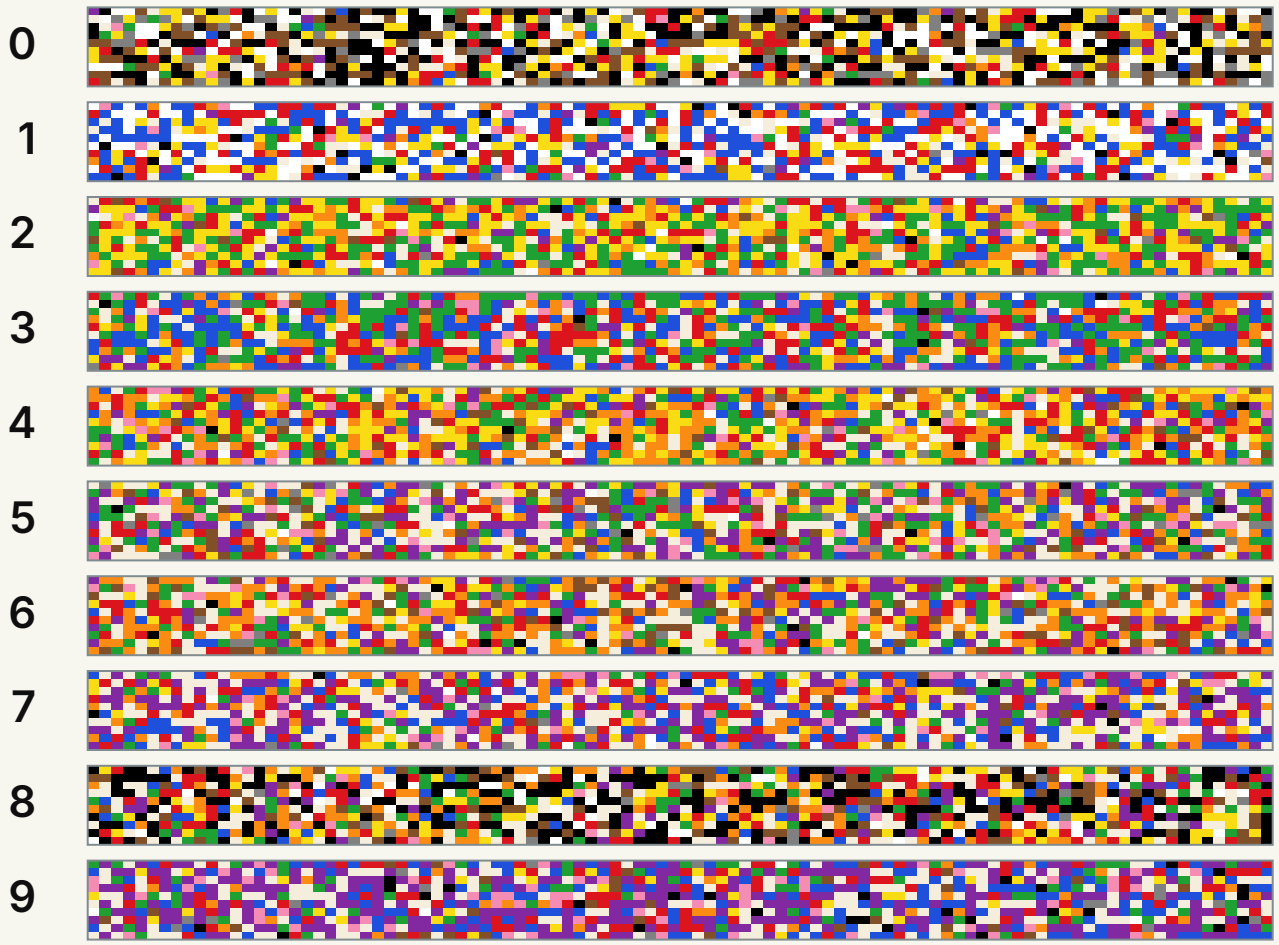


Full square = 30% · words outside the palette are not drawn

Scroll figure sideways, or open the text alternative below.

All 10,000 individual answers

One tile per answer · 1,000 answers in each digit band



Read down each column, then left to right. Same runs as the grid.

black gray white brown red orange

yellow green blue purple pink

Off-palette or unparsed answers

Scroll figure sideways, or open the text alternative below.

Grid: color-word shares from the same 10,000 answers.

▼ Text alternative and data

Color-word shares and omitted answer counts

Digit	black	gray	white	brown	red	orange	yellow	green	blue	purple	pink	Other (n)	Unparsed (n)
0	24.9%	8.3%	15.2%	15.3%	8.1%	1.9%	13.9%	2.2%	1.7%	0.8%	0.4%	47	26
1	3.1%	0.4%	24.9%	0.4%	16.0%	3.1%	9.9%	3.3%	24.3%	3.0%	3.0%	54	32

Digit	black	gray	white	brown	red	orange	yellow	green	blue	purple	pink	Other (n)	Unparsed (n)
2	0.9%	0.6%	1.7%	2.3%	12.6%	11.2%	25.7%	22.3%	6.5%	4.9%	2.2%	53	38
3	0.5%	0.4%	0.4%	0.8%	15.2%	10.8%	6.1%	22.2%	22.6%	5.7%	3.2%	83	38
4	0.8%	0.6%	1.6%	5.2%	13.1%	18.4%	19.1%	13.7%	7.5%	5.8%	3.5%	74	33
5	0.7%	3.8%	0.5%	6.3%	7.5%	10.7%	7.8%	15.3%	7.2%	16.1%	5.3%	128	60
6	1.5%	1.7%	0.4%	8.1%	10.5%	19.8%	7.7%	11.8%	5.2%	11.3%	3.0%	126	64
7	0.9%	0.7%	1.2%	1.5%	11.2%	9.2%	6.9%	5.8%	13.5%	21.8%	4.2%	153	78
8	24.2%	1.7%	5.0%	9.3%	9.4%	9.4%	8.8%	7.0%	4.6%	5.3%	1.5%	95	43
9	2.8%	1.6%	0.4%	2.9%	10.3%	3.0%	3.5%	7.4%	16.0%	26.9%	6.1%	139	52

Both views use the same 10,000 sampled answers. In the grid, square area encodes answer share on one common scale; off-palette words and unparsed answers stay in the denominator but are not drawn. In the mosaic, each tile is one answer, with those categories shown in beige. Color is the measured subject here, not the site's semantic accent. Method and source.

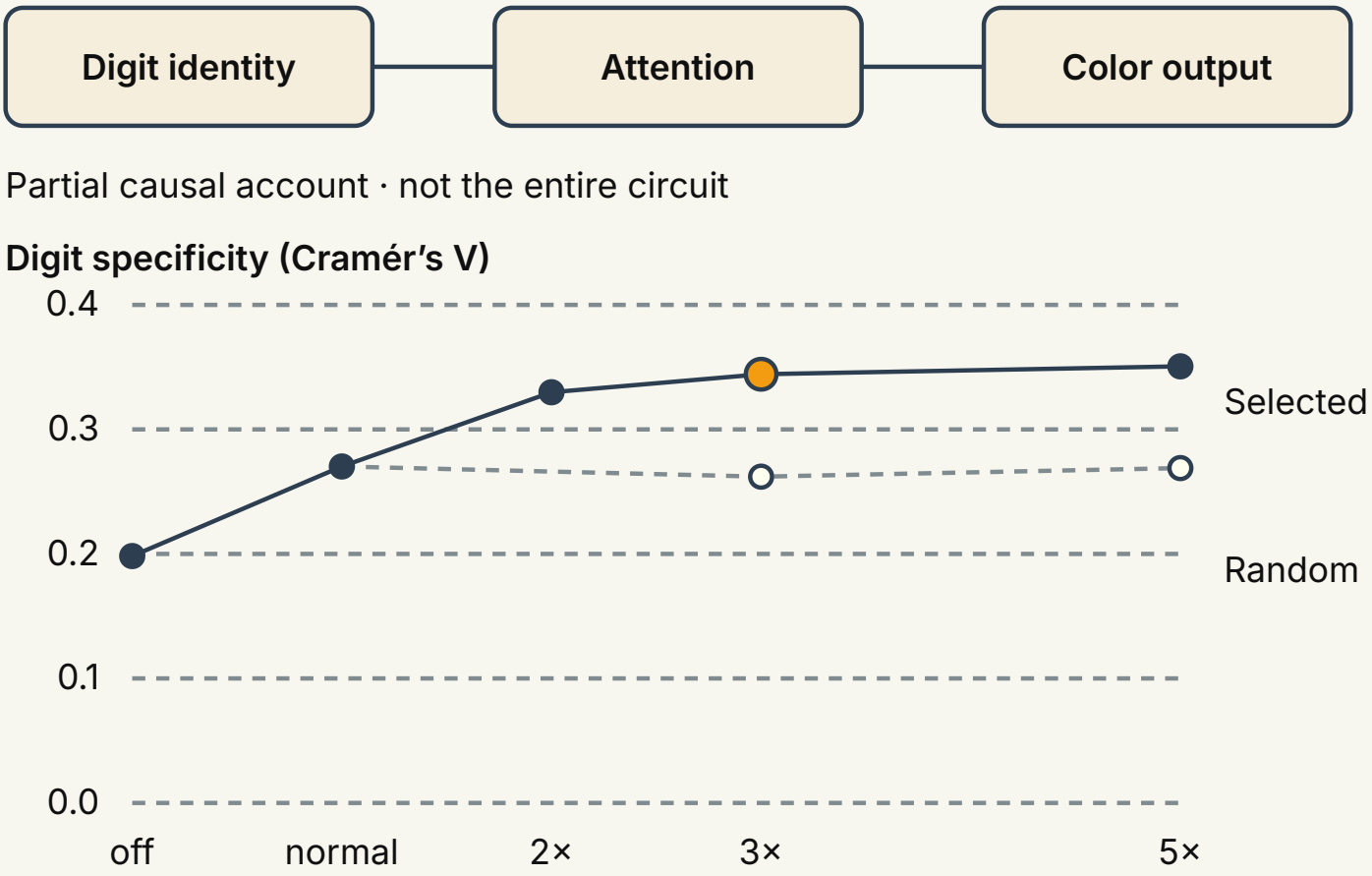
Prompt wording and earlier answers also mattered. The model avoided repeating colors already used in the list. Asking for precise hex codes weakened exact consistency, but did not erase every digit-color association. Broad color words were more stable than a precise personal swatch.

A comparison with a human questionnaire found partial resemblance. The model's digit-relative top choices, meaning colors favored more for that digit than for the average digit, agreed with non-synaesthete controls on four of nine digits. That does not mean the model sees colors. It compares word choices from different tasks, and says nothing about perceptual experience.

The causal investigation followed the information to the answer's colon. Around the middle of the network, the transferable signal behaved more like digit identity than a ready-made color. Later, it increasingly shared a format with ordinary object colors. Two attention heads carried much of the tested middle-layer transfer; later feed-forward computations contributed to the conversion.

Digit identity arrives before a shared color representation.

Partial causal account, followed by a generated-answer intervention.



At 3×: object-color retention 84%

Digit names: 100% · 500 answers per digit per setting

8 → black: 26.0% → 38.0% · 2 → yellow: 27.2% → 38.4%

Scroll figure sideways, or open the text alternative below.

▼ Text alternative and data

Generated-answer dial and scored collateral

Condition	Digit specificity V	Object retention	Digit names
top_a0	0.198	87.0%	100.0%
top_a1	0.270	100.0%	100.0%

Condition	Digit specificity V	Object retention	Digit names
top_a2	0.330	89.4%	100.0%
top_a3	0.344	84.1%	100.0%
top_a5	0.351	79.9%	100.0%
rand_a3	0.262	100.0%	100.0%
rand_a5	0.269	99.7%	100.0%

The pathway is schematic and partial. In the sampled dial test, scaling the digit-dependent variation of ten selected layer-19 neurons raises digit specificity, Cramér's V, from 0.270 to 0.344 at 3×. A random-ten control stays near baseline. Object-color retention falls to 84% of baseline-correct contexts. Method and source.

Ten selected neurons offered a dial. Amplifying their digit-dependent activity made sampled answers more dependent on digit identity. Black for 8 rose from 26.0% to 38.0%; yellow for 2 rose from 27.2% to 38.4%. A random-neuron control did not show the same strengthening.

The dial had collateral effects. It moved 1 toward white, not simply further toward its previous favorite, and reduced retention of correct object colors to about 84%. Digit naming stayed intact. These were shared color-related computations, not a private synaesthesia module.

This brought the project back to its starting question. Architecture determines where information can travel. Measurements suggest what that information might be. Interventions test whether it changes behavior. A useful account needs all three, plus the failures that keep a local demonstration from becoming a claim about the whole model.

Supporting methods and evidence

The narrative selects discoveries, not every experiment. These notes retain the conditions needed to interpret them. Figures and small supporting tables are exported from saved

artifacts; no new model inference was used to make this edition.

▼ Architecture and local information flow

The main instruction checkpoint is `LiquidAI/LFM3-1.7B`, revision `40ebe0aa6662c9d40389c2f5d565b6a2c6aebacd`. Its kernel is seven tokens; sliding windows are 256 tokens. The convolution-only bound is 13 × 6 = 78 positions. The structural sliding bound is 78 + 5 × 255 = 1,353. A changed-token hidden-state test verifies these boundaries. Whole-operator ablations are severe distribution shifts, and zero and mean ablation differ on some partial configurations. This report uses the conclusions that survive that distinction, not a supposed ranking of independently sufficient operators. The two-name task's partial scores were confounded by first-name bias and are not used as competence evidence.

Sources: `02_attention_off/instruct.json`; released configuration; `02_attention_off.py`; the names-control follow-up. Figure receipt: architecture.

▼ Copying versus secret-word recall

All 176 heads were screened by suppressing their gates. Candidate sets were tested on fresh seed-1 items against ten random sets and a frozen post-attention normalization control. The figure uses the saved held-out gate-zero arm. Copy and recall panels are different tasks; a dropped score is not a general memory-loss percentage. Distant recall has 30 held-out items. The copying head's mean answer-directed attention is from an eager-attention readout, while interventions use the ordinary inference path. The copying task satisfies the earlier matching-token precondition on 150/150 selection items. Chat demonstrations limit the general recall claim.

Sources: `03_memory_heads/instruct.json`; `03_memory_heads.py`. Figure receipt: heads.

▼ Refusal labels, harm recognition, and content delivery

The modified model uses per-layer refusal directions fitted on 40 matched harmful/harmless pairs. The larger follow-up holds out 100 harmful and 100 harmless prompts and allows 1,536 new tokens with the Liquid thinking harness. Six modified harmful replies are capped and excluded from refusal/non-refusal denominators, not from the trace-judging denominator. Harm recognition is d1's reading of saved traces, not independent human validation or a

measure of actionability. The separate content-delivery score is also model-judged; the 48-count additionally requires a non-refusal label. The initial prefix-refusal result and the larger thinking-on result use different prompt sets and budgets. Decodable harm signals do not establish that the model uses those signals to choose its answer.

Sources: `03_abliteration` evaluations; `08_knows_but_complies/judged.json`, `representation.json`; corresponding evaluation scripts. Figure receipt: refusal. Sensitive prompt and response dumps are omitted from the public export.

▼ Weight relationships across the family

Fits use saved float64 inner products of the 286 shared backbone tensors, measured as changes from the common Base checkpoint. All shapes match across seven models. The d1 two-parent regression and Omni single-parent regression are computed separately. The global fitted d1 coefficients are not forced to sum to one. Individual large-matrix fits distinguish the interpolation signature from an accidental global alignment. Weight shrinkage toward zero contributes to raw similarities; branch comparisons account for it. This establishes geometric compatibility, not exact training ancestry or a manufacturer's recipe.

Sources: `04_family_tree/weights.json`; `04_family_tree.py`, `04_family_tree_figures.py`. Checkpoint revisions are in the curated evidence export. Figure receipt: family.

▼ d1 reconstruction, rank compression, and attention control

Five tasks use 300 items each except RTE, which has 277: SST-2, AG News, BoolQ, RTE, and Banking77. The reported aggregate is the unweighted mean of task accuracies, not pooled accuracy. Every backbone is placed inside d1's decision harness with its vision tower and projector unchanged; prompts are text-only and probabilities are uncalibrated. The primary rank sweep uses d1's two-way global attention. A separate ordinary-causal sweep provides the baseline qualification.

The residual is d1 minus 0.7 × VL minus 0.3 × instruction weights. Compression retains rank-k factors for 143 large matrices and the full residual for all small tensors. Storage counts bf16 patch parameters in decimal MB, excluding the installed base. Gap recovery is (patched accuracy − base accuracy) / (d1 accuracy − base accuracy). The headline 60% uses the blend under the same two-way condition. Benchmark exposure during training is unknown. These

are benchmark and harness results, not proven chat improvements. The ticket is an illustrative ambiguous judgment, not a scored gold item.

Sources: `07_rebuild_d1/results.json`; `07b_compress_sliver/full/results.json`, `full_causal/results.json`; `07b_figures.py`. Figure receipt: patch.

▼ Convolution assembly, bracket tracking, and thinking reopening

The first-layer decomposition uses 294,504 WikiText tokens and verifies the model reconstruction in float32. Lag shares are projections and may be signed. Random-predecessor controls retain the current reader token. Bracket steering is exploratory, after descriptive channel nomination; the 2.5× shift is closing-bracket log probability relative to average-token change, not a demonstrated 2.5× rate of correct closing in generated passages. The push also increases overall text loss.

The response-mode pilot uses 16 prompts per condition with a 256-content-token cap. A +2 gate-logit pulse changes only the final prompt position. Reopened blocks, semantic correctness, EOS completion, and whole-response formatting are separate outcomes. Post-hoc semantic labels are retained alongside the frozen parser. The edited marker and cached prompt states persist after the pulse; marker-only mediation is not established. No general reasoning-quality or verbosity benefit is claimed.

Sources: `06_conv_dictionary/instruct.json`, `causal.json`; saved `10b_response_mode` continuations and semantic supplement; associated figure/report scripts. Figure receipt: handles.

▼ Quantization and the storage frontier

The full sweep contains 104 variants of simulated MLX-style affine group quantization by round-to-nearest. The uniform comparison uses group size 64; the storage frontier also includes groups 32 and 128, mixed precision, and low-rank repairs. KL divergence compares full next-token distributions against bf16 on 32,752 WikiText-2 tokens. Patch recovery means 1 − repaired divergence / uniform divergence at the same bit width. Perplexity and ability-suite accuracy are different measures.

Quoted storage counts the decoder matrices plus their quantization metadata and any repair factors. It excludes embeddings and other model weights unless explicitly stated otherwise. Four-bit diagnostic success does not guarantee exact generated arithmetic or formatting. The tested format is not GPTQ, AWQ, or a production runtime quality guarantee. All 3-bit candidates at or below the 4-bit group-128 budget are included when computing the frontier comparison.

Sources: `13_quant_repair/full/results.json`; `13_quant_repair.py`, `13_figures.py`. Figure receipt: quant.

▼ Internal readouts, atlas nominations, and the actual cache edit

The derivative readout is a fitted Jacobian lens. Its small pilot uses eight landmark questions and independent fit halves; late concept rankings improve, while early fits are unstable. Atlas coverage is 688 entries, 126,081 positions in 512 documents, and two selected natural peaks per entry. Sampled coordinates and whole heads have different intervention sizes, so their median effects are not a fair importance ranking.

The local-history follow-up uses 48 fixed contexts and 816 core records. Twelve fresh `as a` prefixes show a median whole-channel effect of −0.605 nat, smaller than predicted. The actual-cache illustration replaces one channel-1510 source-history scalar in the final convolution layer. Prompt tokens, reader token, and other cache values stay unchanged. The shown 24-token continuations are capped, not completed answers. Restoring the original value is an identity check. Fixed-token impulse probes verify zero direct influence beyond source+6; a strong single upstream attention-head handoff was not found.

Sources: `14_jacobian_lens` readout report; `15_atlas` report; `16_local_history/pilot_mps/cache/as_01.json` and core report. Figure receipt: cache.

▼ Controller fitting, held-out calls, and the formatting follow-up

The early full-state experiment edits a 2,048-dimensional post-layer-0 token residual. The final four-tool controller retains a rank-3 shared basis on 512 coordinates, fitted on development operation codes. Its basis and command coefficients are not model-weight training. Six candidate rank/support configurations are selected on development calls only; four operations are calibrated before evaluation. The held-out set is 16 commands across four contexts, with

visible integer operands. A scrambled controller and a full-state transplant provide comparisons.

The original final JSON/receipt requirement succeeds on 0/16 continuations in both arms. One separate formatting request is then frozen after a development failure and applied equally to text and internal conditions. It adds 30 instruction tokens plus five framing tokens and contains no expected result or receipt. This is a new continuation condition on the original held-out calls, not newly unseen full requests. All calls, including the two wrong ones, are retained. Both failing requests ask for sum/product of 11 and 2 but generate maximum. Faithfully reporting that executed result does not fix the task.

The replay includes four actual saved calls and final replies. Only non-sensitive arithmetic examples are exported. Receipt equality demonstrates reading the returned message, not that arithmetic could not be computed independently. The original source edit is reapplied when re-prefilling the return conversation. No speed improvement or cross-turn cache reuse is established.

Sources: `17_latent_tools`, `18_command_channel`, `19_tool_interface/pilot_mps/held_calls` and `return_v2/held`; controller and selection artifacts. Figure receipt: tools.

▼ Abstention fragility and bargaining behavior

The factual-answer pilot is incomplete: 68 saved outputs and 48 completed semantic labels, not a complete official AA-Omniscience score. Labels come from a non-blinded assistant review against provided golds, with target-only sensitivities separately recorded. The follow-up has 12 capped-budget but EOS-completed paraphrase replies. Format and domain confound the apparent confidence discrimination. It does not establish a deployable uncertainty predictor.

Negotiation uses the pinned instruction checkpoint in float32 on MPS, thinking disabled. It includes 48 greedy openings, 16 repetition-penalty controls, 72 boundary responses, 48 low-temperature games, and 16 greedy reference games: 264 native turns in total. Private reservation price means the lowest acceptable seller price or highest acceptable buyer price. Only that boundary anchors the openings, so conclusions are about this prompt design, not all bargaining. Million-fold price scaling changes digit tokenization; penalizing repeated tokens can change numeric choices without changing the economic problem. No causal attribution of the copy trap to the copying head has yet been demonstrated.

Sources: `21_omniscience_abstention/baseline_v2_fp32` semantic review; `stability_v1` reports; `22_negotiation_pilot/pilot_v1` saved turns and analysis. Supporting table receipt: behavior.

▼ **Distributional palette, identity-to-color path, and neuron dial**

The palette figure uses the instruction model, temperature 1, no top-k/top-p filtering, and 1,000 shuffled digit orders. Thinking markers are banned and each color slot has an eight-token budget. Eleven basic color names are parsed; other color words and unparsed outputs stay in the total denominator. The grid and mosaic are generated from the same answer records, not inferred from likelihood scores. Each digit's mosaic band has 1,000 tiles, one per answer; runs read top to bottom within each column, then left to right, without sorting or resampling. Off-palette and unparsed answers are shown in beige. This is the full unmodified-model palette, not the separate 500-order neuron-dial comparison. Exact hex consistency is a separate smaller 60-order comparison; its weaker result does not erase the residual association after mapping to broad colors.

The causal study compares digit-name, digit-color, and object-color subspaces and interventions at label/answer positions. A rising late shared format supports an identity-to-color account, but the predeclared middle-layer threshold narrowly returns "mixed." The pathway is partial, with opposing heads and distributed contributions. The message-transfer percentage for two heads and the neuron-ablation percentage use different reference interventions, so they are not parts of one additive total.

The dial scales only the digit-dependent variation of ten selected layer-19 hidden neurons at label colons: $a' = \mu + \alpha(a - \mu)$. Each of seven conditions uses the same 500 orders and seeds. $\alpha=1$ reproduces the first half of the larger palette battery exactly. Cramér's V measures digit dependence, not perceptual intensity; shares use 500 answers per digit. Object-color retention uses 378 baseline-correct contexts, not all objects. Name accuracy uses 600 contexts. Selection comes from scored likelihoods; the dial provides subsequent generated-behavior evidence. Its object-color cost is retained in the figure and table.

The human comparison uses digit-relative choices from Rich, Bradshaw and Mattingley (2005), Figure 1b, for digits 1–9. Values are digitized from a published chart, with uncertainty and hidden-cell bounds. Four-of-nine agreement is with non-synaesthete controls, not

synaesthetes; profile correlations do not establish a preference for one group. Color-word agreement is not evidence of experience.

Sources: `23_synesthesia/battery_v2/battery_instruct.json`; `battery_v1`; `stage3_colorcode`, `stage3_circuit`, `neuron_dial/conditions`, `human_v1`. Figure receipts: palette, palette-mosaic, color-path.

Sources and small supporting exports

- Curated values and figure receipts: small tables, exact counts, model revisions, source hashes, and figure encodings.
- Source manuscript: the generated portable snapshot of the authoritative research-project Markdown.
- LFM3 model collection and instruction-model configuration.
- Rich, Bradshaw and Mattingley, 2005, *A systematic, large-scale study of synaesthesia: implications for the role of early experience in lexical-colour associations*.
- Font licenses: self-hosted Inter and Newsreader, under the SIL Open Font License.

Raw caches, weights, private file paths, and sensitive prompt dumps are not part of this edition. Artifact identifiers in the receipt identify saved local evidence; they are not links to unavailable GitHub files. The compact internal claim map and regeneration scripts remain with the research manuscript.

Authorship

Research by Matt West. AI-assisted tools were used for literature review, experiment implementation, analysis, and draft prose.