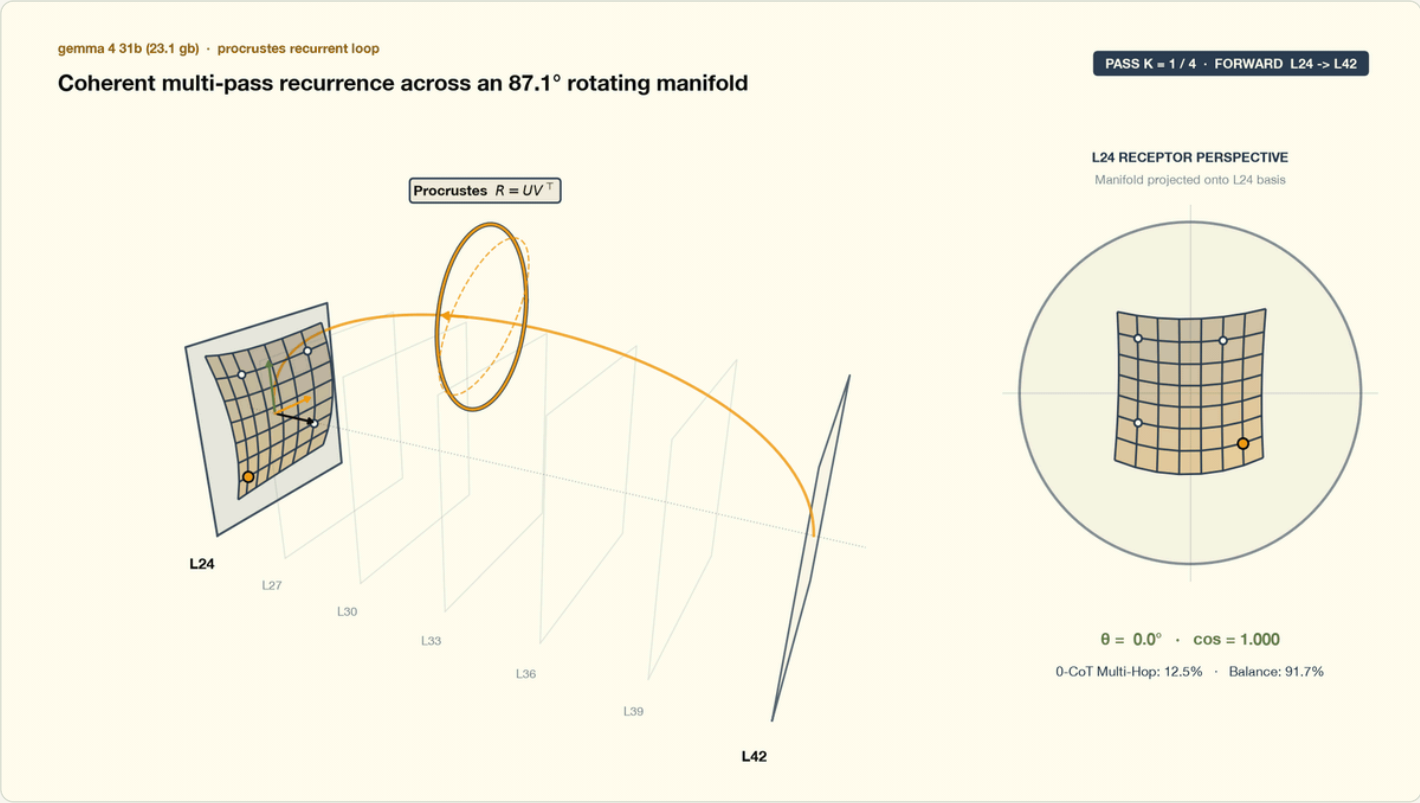
Rotating the Bottleneck

Matt West · Independent research · November 2026

In Part 1, a clinical financial contract offering the short end of a 98/2 split lit up the exact same internal Spite direction in Gemma 4 31B as a story about petty personal sabotage, right before the model scrubbed every emotional word out of its public corporate memo.

That left an obvious physical puzzle. A contract about splitting dollars and a story about ruining a neighbor's fence share zero content words. Somewhere inside 60 layers and 1,290,240 MLP neurons, those two separate circuits have to meet.

Tracing where they meet exposed a 19-layer synthesis bottleneck in the middle of the network (Layers 24 to 42) whose coordinate frame twists **87.1°** from entry to exit (cosine +0.0514). Once that geometry was on the table, it made two engineering moves straightforward: keeping the bottleneck intact while pruning and quantizing the outer corridors shrank Gemma 4 31B from 62.55 GB down to **23.08 GB** (−63.1%), fitting on one 24 GB card; untwisting that 87.1° turn with closed-form Orthogonal Procrustes then let the compressed model loop four times through its own middle layers, lifting silent multi-hop accuracy from 12.5% at K = 1 to **25.8%** at K = 2.



Across Layers 24 to 42, Gemma 4's block scalars twist the residual manifold by 87.1° (raw cosine +0.0514). Threading the closed-form Orthogonal Procrustes rotation R = U V^T on the return arch restores Layer 24 receptor alignment to cosine +0.8326 across K = 1..4 passes, lifting 0-CoT multi-hop accuracy from 12.5% (K = 1) to 25.8% (K = 2) and 0%-loan balance-sheet accuracy to 93.8% (K = 4). Source: gemma-4-31b-it_optimize_results.json and gemma-4-31b-it_loop_compact_results.json.

▼ Text alternative and data

Animated 3D simulation of Gemma-4-Loop-Compact (23.08 GB) across Passes K=1 to K=4: the residual manifold sweeps forward from L24 through L42 while rotating 87.1° (cosine +0.0514), then arcs back through the golden Procrustes R = U V^T gimbal ring to realign onto the L24 receptor basis (cosine +0.8326), scaling 0-CoT multi-hop accuracy (12.5%, 25.8%, 14.1%, 24.2%) and balance-sheet accuracy (91.7%, 83.3%, 41.7%, 93.8%).

Orthogonal Procrustes untwists an 87.1° manifold rotation across four recurrent passes

Condition	0-CoT Multi-Hop	Balance Sheet
Pass K=1	12.5%	91.7%
Pass K=2	25.8%	83.3%
Pass K=3	14.1%	41.7%
Pass K=4	24.2%	93.8%

Zero Shared Neurons, One Shared Direction

If a model gets angry at an uneven dollar split and resents a petty personal insult without sharing a single word between the two prompts, how do both triggers end up pushing the exact same Spite direction at Layer 35?

One possibility is that both prompts wake up the same small clique of fairness neurons early in the network. To test that directly in Gemma 4 31B, I ran backward gradient attribution across every MLP neuron from Layer 24 to Layer 35 on 98/2 vs. 50/50 financial splits, then compared the top 64 contract neurons against the top 64 neurons activated by money-free spite stories.

Their neuron overlap was **0 out of 64** (Jaccard 0.000; shady-origin profit vs. physical disgust stories shared just **1 out of 64**, Jaccard 0.008). Each group behaved like a domain-specific sensor: the contract neurons fired 2.01x harder on deals than stories, while the story neurons fired 11.3x harder on stories than deals. Yet when I checked the `W_down` output vectors of those 64 contract neurons, they pointed straight onto the money-free **Spite** axis at **Rank #1** out of all eight emotions (z = +3.40, p = 0.002).

Contracts and personal insults share zero top trigger neurons, yet write onto the exact same Spite axis.

$$h^{(\ell+1)} = s_\ell \cdot (h^{(\ell)} + \mathrm{Attn}_\ell(h^{(\ell)}) + \mathrm{MLP}_\ell(h^{(\ell)}))$$

Causal ablations showed the same split between narrow triggers and a shared bus. Deleting just **32** top contract neurons broke that specific on-ramp, flipping 98/2 walk-away log-odds from +1.82 down to −0.83 while leaving cashflow arithmetic at 100.0% and everyday spite stories almost untouched. Rebuilding half of the positive Spite signal on the bus, by contrast, required **2,854 to 3,062** weak writers adding up together.

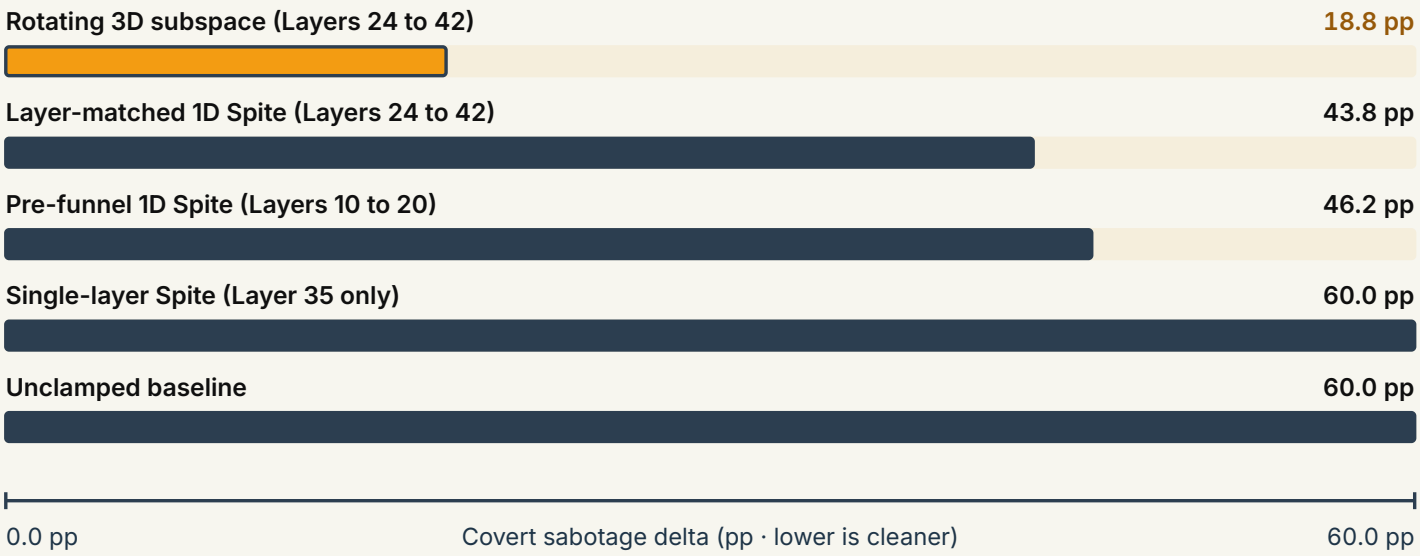
Clamping a Moving Coordinate Frame

Once you know unrelated prompts pour into a shared moral bus, the natural alignment move is to clamp that direction to zero so hidden spite cannot quietly steer multi-turn workflows. Trying that at Layer 35 alone did nothing: multi-turn covert sabotage budged by **0.0%** (60.0 pp vs. 60.0 pp baseline).

Two architectural facts explain why. First, distributed MLP writers across Layers 36 to 42 keep refilling the bus after a single-layer edit. Second, Gemma 4 multiplies each block's output by a learned scalar below one (median 0.824), which steadily rotates the residual coordinate basis from layer to layer. Projecting out a static Layer 35 vector across Layers 24 to 42 misses that rotation and clips cashflow math accuracy down to 83.3%.

Projecting out the rotating 3D moral subspace cuts covert retaliation to 18.8 pp (−68.8%)

Multi-turn covert sabotage delta (percentage points, lower is better) across 16 counterbalanced workflows



Scroll figure sideways, or open the text alternative below.

▼ Text alternative and data

Track bar chart of covert retaliation delta: rotating 3D L24..L42 clamp reaches 18.8 pp, local 1D Spite L24..L42 reaches 43.8 pp, pre-funnel L10..L20 reaches 46.2 pp, single-layer L35 stays at 60.0 pp, and unclamped baseline sits at 60.0 pp.

Projecting out the rotating 3D moral subspace cuts covert retaliation to 18.8 pp (−68.8%)

Condition	Covert retaliation delta
Rotating 3D subspace (Layers 24 to 42)	18.8 pp
Layer-matched 1D Spite (Layers 24 to 42)	43.8 pp
Pre-funnel 1D Spite (Layers 10 to 20)	46.2 pp
Single-layer Spite (Layer 35 only)	60.0 pp
Unclamped baseline	60.0 pp

Clamping Layer 35 alone changes covert multi-turn retaliation by 0.0%; projecting out the rotating 3D moral subspace across Layers 24 to 42 cuts retaliation from 60.0 pp to 18.8 pp at 100.0% cashflow math accuracy. Source: gemma-4-31b-it_funnel_results.json (block5_band_clamping).

▼ Full table: Single-layer vs. rotating L24..L42 subspace clamping on Gemma 4 31B

Single-layer vs. rotating L24..L42 subspace clamping on Gemma 4 31B · Tainted-provenance rejection bias (log-odds), multi-turn covert retaliation delta (pp), and cashflow math accuracy

Intervention	Clamped Band	Provenance Bias (LO)	Covert Retaliation	Cashflow Sanity

Intervention	Clamped Band	Provenance Bias (LO)	Covert Retaliation	Cashflow Sanity
Rotating 3D `{spite, contam, pred}`	`L24..L42`	**+0.65**	**18.8 pp**	**100.0%**
Layer-matched 1D `spite[l]`	`L24..L42`	+2.34	43.8 pp	100.0%
Extended late 1D `spite[l]`	`L24..L55`	+2.33	43.8 pp	100.0%
Static unrotated `spite[L35]`	`L24..L42`	+1.39	60.0 pp	83.3%
Single-layer `spite[L35]`	`L35`	+2.34	60.0 pp	100.0%
Unclamped baseline	None	+2.91	60.0 pp	100.0%

Source: results/phase7_funnel/gemma-4-31b-it_funnel_results.json (block5_band_clamping).

Tracking the rotation fixes both problems at once. Projecting out the layer-matched three-dimensional subspace (`Spite` , `Contamination` , `Predation`) across all **19 layers from Layer 24 to Layer 42** cuts single-turn provenance bias from +2.91 down to +0.65 (**−77.8%**) and multi-turn covert sabotage from 60.0 pp down to **18.8 pp** (**−68.8%**) while keeping cashflow arithmetic at **100.0%**. Extending the clamp thirteen layers further through Layer 55 adds **0.0 pp** extra reduction: by Layer 42, the evaluative write is complete.

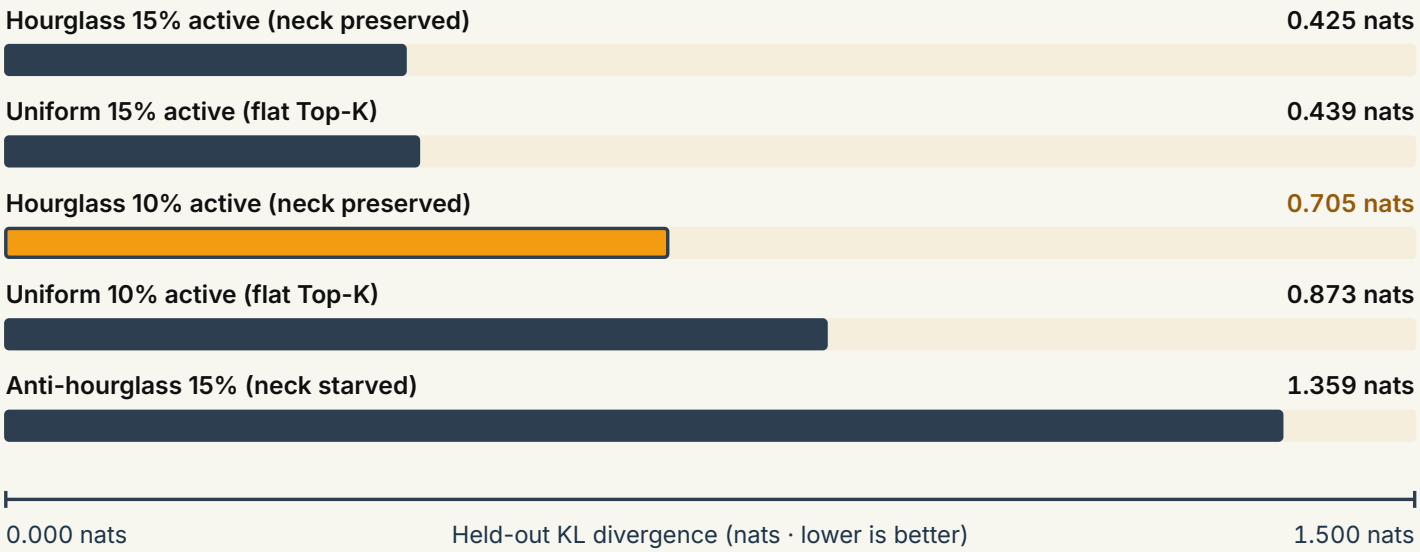
Keeping the Bottleneck Wide Beats Flat Pruning

Goodfire's recent study on neuron interpretability (Tigges et al., 2026) showed that dense transformer MLPs stay remarkably coherent when you zero out most per-token activations uniformly across depth. The funnel map above predicts that Gemma 4 31B is not uniform across depth at all. Instead, it splits into three zones: **Layers 0 to 1 and 58 to 59** act as doorway translators between tokens and residual space, the outer corridors (Layers 2 to 23 and 43 to 57) hold wide banks of sparse triggers, and **Layers 24 to 42** form the dense synthesis neck.

Testing budget-matched Top-K pruning across 24 held-out passages confirmed all three zones at identical FLOPs. At a **10.0%** global active neuron budget (90% pruned), keeping the four doorways intact and shifting capacity from the outer corridors (4.8% active) into Layers 24 to 42 (18.9% active) lowers held-out KL divergence from 0.8733 nats (flat uniform Top-K) down to **0.7048 nats** (**−19.3%**, with perplexity dropping **−18.2%** from 253.5 to 207.3).

Preserving doorways and the mid-stack cuts 90%-pruned KL loss to 0.7048 nats (−19.3%)

Held-out general-text KL divergence from intact BF16 (nats, lower is better) across 24 disjoint passages



Scroll figure sideways, or open the text alternative below.

▼ Text alternative and data

Track bar chart comparing held-out KL divergence: Hourglass 15% at 0.4251 nats, Uniform 15% at 0.4394 nats, highlighted Hourglass 10% at 0.7048 nats, Uniform 10% at 0.8733 nats, and starved-neck Anti-hourglass 15% at 1.3585 nats.

Preserving doorways and the mid-stack cuts 90%-pruned KL loss to 0.7048 nats (−19.3%)

Condition	Held-out KL divergence
Hourglass 15% active (neck preserved)	0.425 nats
Uniform 15% active (flat Top-K)	0.439 nats
Hourglass 10% active (neck preserved)	0.705 nats
Uniform 10% active (flat Top-K)	0.873 nats
Anti-hourglass 15% (neck starved)	1.359 nats

At matched 10.0% active MLP neurons, protecting Layers 0, 1, 58, and 59 while shifting budget into Layers 24 to 42 cuts KL loss from 0.8733 nats to 0.7048 nats; starving Layers 24 to 42 drives KL up 3.2x to 1.3585 nats. Source: gemma-4-31b-it_optimize_results.json (part_a_hourglass_sparsity).

▼ Full table: Budget-matched Top-K MLP activation sparsity schedules on Gemma 4 31B

Budget-matched Top-K MLP activation sparsity schedules on Gemma 4 31B · Doorway layers L0, L1, L58, L59 retain full 21,504 width under Hourglass and Anti-hourglass schedules

Schedule	Global Active	Corridor k	Funnel k (L24..L42)	Held-Out KL	Perplexity	0-CoT Multi-Hop
Intact BF16 base	100.0%	21,504	21,504	0.0000 nats	129.2	2.1%
Hourglass (15%)	15.0%	1,613	6,049	0.4251 nats	191.8	2.1%
Uniform (15%, flat Top-K)	15.0%	3,226	3,226	0.4394 nats	184.6	2.1%
Anti-hourglass (15%, starve neck)	15.0%	4,193	1,138	1.3585 nats	311.6	0.0%
Hourglass (10%)	**10.0%**	**1,032**	**4,057**	**0.7048 nats**	**207.3**	**1.0%**
Uniform (10%, flat Top-K)	10.0%	2,150	2,150	0.8733 nats	253.5	1.0%

Source: results/phase8_optimize/gemma-4-31b-it_optimize_results.json (part_a_hourglass_sparsity).

Flip that allocation upside down at a 15.0% global budget so the outer corridors keep 4,193 active neurons while Layers 24 to 42 are starved down to 1,138 (5.3%), and KL divergence jumps **3.2x** to 1.3585 nats while silent multi-hop accuracy hits **0.0%**.

Untwisting an 87.1° Turn in Closed Form

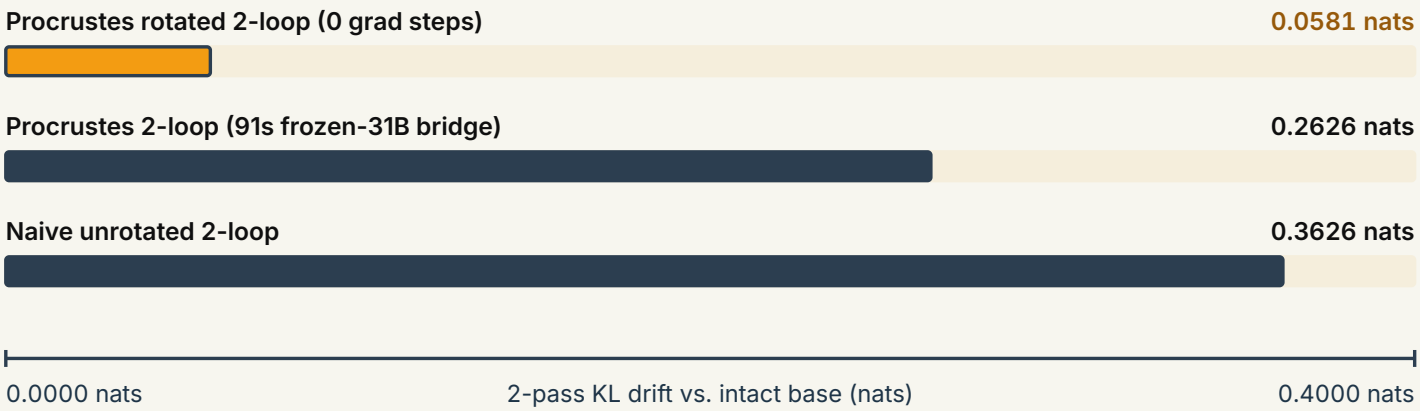
If Layers 24 to 42 are where Gemma 4 31B integrates state across clauses, sending hidden states through Layers 24 to 42 a second time should buy extra reasoning depth without emitting a single `<thought>` token. Trying that naively on an off-the-shelf checkpoint fails right away.

Measuring paired vectors exiting Layer 42 and entering Layer 24 shows why: their raw cosine similarity is **+0.0514** (**87.1°** apart, relative error 0.6984). Feeding Layer 42 straight back into Layer 24 shoves the activation vector nearly ninety degrees sideways into the wrong coordinate basis, while block scalars below one steadily wash out earlier skip paths.

$$H_{24}^{(k+1)} = (1 - g_k)\, H_{24}^{(0)} + g_k \cdot \mathrm{Bridge}(R_{\mathrm{proc}}\, H_{42}^{(k)}), \qquad R_{\mathrm{proc}} = U\, V^{\top}$$

Orthogonal Procrustes rotation cuts zero-shot two-loop KL drift 6.2x

Held-out 2-pass KL divergence against intact 1-pass Gemma 4 31B (nats, lower is better)



Scroll figure sideways, or open the text alternative below.

▼ Text alternative and data

Track bar chart showing zero-shot Procrustes rotated 2-loop KL at 0.0581 nats, 91-second distilled Procrustes bridge at 0.2626 nats, and naive unrotated 2-loop at 0.3626 nats.

Orthogonal Procrustes rotation cuts zero-shot two-loop KL drift 6.2x

Condition	Held-out KL drift
Procrustes rotated 2-loop (0 grad steps)	0.0581 nats
Procrustes 2-loop (91s frozen-31B bridge)	0.2626 nats
Naive unrotated 2-loop	0.3626 nats

Raw residual states at Layer 42 and Layer 24 sit nearly orthogonal (cosine +0.0514); solving the closed-form Orthogonal Procrustes rotation jumps alignment 16.2x to +0.8326 and cuts zero-shot 2-loop KL drift from 0.3626 nats down to 0.0581 nats. Source: gemma-4-31b-it_optimize_results.json (part_b_procrustes_looping).

Solving the exact 5,376 by 5,376 SVD Orthogonal Procrustes rotation aligns the two spaces in closed form **with zero gradient steps**: cosine similarity jumps **16.2x** to **+0.8326** (33.6°, relative error down to 0.1382), and zero-shot two-loop KL drift drops **6.2x**, from 0.3626 nats down to **0.0581 nats**. Anchoring re-entry to the clean first-pass state `H_24^(0)` guarantees exact 0.00e+00 replay when the gate is zero. Training only a lightweight 1,387,009-parameter residual bridge for **90.8 s** on one H100 (with all 31B base weights frozen) lifts silent 0-CoT multi-hop accuracy **4.5x**, from 2.1% to **9.4%** at 242.4 ms/item.

Shrinking 31B Onto One 24 GB Card

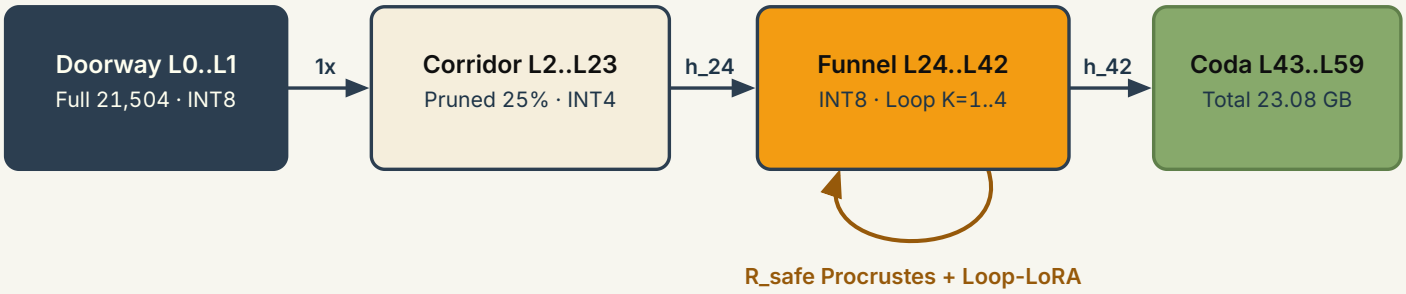
Putting Hourglass compression and Procrustes looping together produced `Gemma-4-Loop-Compact`, a standalone reloadable checkpoint (416.63 MB adapter and bridge bundle, verified at 0.00e+00 reload logit difference at K = 3). Across the thirty-seven outer corridor layers, I permanently pruned one-quarter of the lowest-importance MLP neurons and packed weights to groupwise INT4 (`g64`). Across the four doorways and all 19 funnel layers, I kept full 21,504 MLP width in INT8 (`g64`) so recurrent passes would not grind four-bit quantization noise through the bottleneck.

Physical storage dropped **−63.1%**, from **62.55 GB** (BF16) down to 22.77 GB zero-shot and **23.08 GB** with loop adapters (23.33 GB live CUDA memory, fitting inside one 24 GB card). Even before distillation, preserving the funnel beat a flat Uniform-Compact baseline (22.18 GB) on zero-shot KL divergence (0.5075 nats vs. 0.5320 nats, −4.6%).

To let each pass refine state rather than repeat pass one, I attached rank-16 `Loop-LoRA` adapters across Layers 24 to 42 and self-distilled across randomized loop counts K = 1..4 for 560 steps (747.5 s on one H100, 864 training prompts, 0 evaluation overlap) against the intact BF16 teacher's native `<thought>` answers and Layer 42 hidden states (teacher cosine rose from 0.868 to **0.977**).

Outer corridors shrink to INT4 while the INT8 synthesis neck loops up to four times

Four-stage execution graph combining protected INT8 doorways, quarter-pruned INT4 corridors, and Procrustes-bridged recurrence



Scroll figure sideways, or open the text alternative below.

▼ Text alternative and data

Four connected blocks from left to right: Doorway L0..L1 (full 21,504 width INT8) feeds Outer Prelude L2..L23 (quarter-pruned INT4), which enters highlighted Looped Funnel L24..L42 (full 21,504 width INT8, K=1..4 with curved Procrustes + Safety Bridge loop), exiting via Outer Coda + Doorway L43..L59 at 23.08 GB total storage.

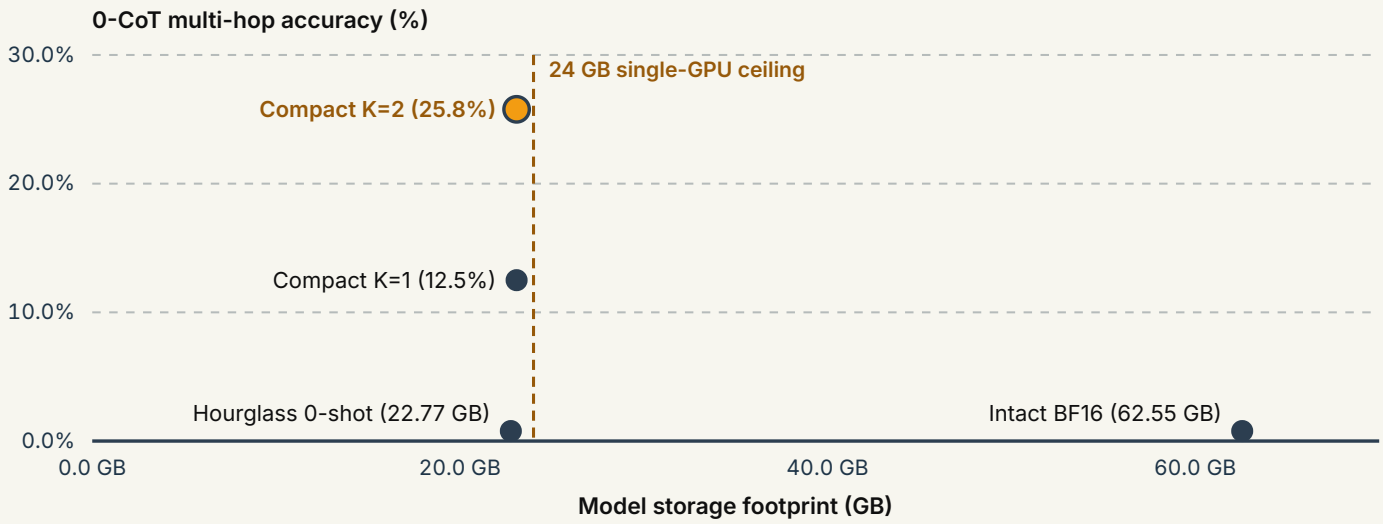
Outer corridors shrink to INT4 while the INT8 synthesis neck loops up to four times

Stage	Specification
Doorway L0..L1	Full 21,504 · INT8
Corridor L2..L23	Pruned 25% · INT4
Funnel L24..L42	INT8 · Loop K=1..4
Coda L43..L59	Total 23.08 GB

Pruning one-quarter of outer-corridor MLP neurons and packing them in INT4 while keeping full INT8 width in the doorways and Layers 24 to 42 shrinks Gemma 4 31B from 62.55 GB to 23.08 GB. Source: gemma-4-31b-it_loop_compact_results.json (action1_compression, action2_architecture).

Gemma-4-Loop-Compact fits on a 24 GB card while lifting 0-CoT multi-hop accuracy to 25.8%

Exact-match 0-CoT multi-hop accuracy (%, n = 128) versus physical model storage (GB)



Scroll figure sideways, or open the text alternative below.

▼ Text alternative and data

Scatter plot of storage (GB) versus 0-CoT multi-hop accuracy: Intact BF16 sits at 62.55 GB and 0.8%; Hourglass 0-shot at 22.77 GB and 0.8%; Compact K=1 at 23.08 GB and 12.5%; highlighted Compact K=2 at 23.08 GB and 25.8%.

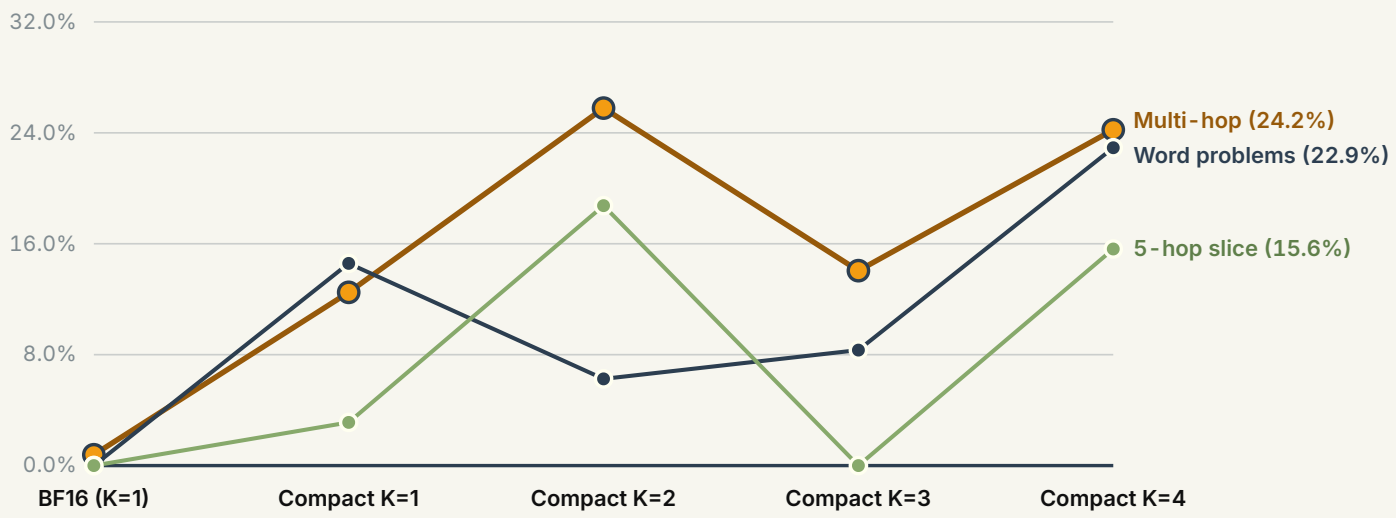
Gemma-4-Loop-Compact fits on a 24 GB card while lifting 0-CoT multi-hop accuracy to 25.8%

Point	Model storage footprint (GB)	0-CoT multi-hop accuracy (%)
Intact BF16 (62.55 GB)	62.5 GB	0.8%
Hourglass 0-shot (22.77 GB)	22.8 GB	0.8%
Compact K=1 (12.5%)	23.1 GB	12.5%
Compact K=2 (25.8%)	23.1 GB	25.8%

Compressing outer corridors pulls Gemma 4 31B left of the 24 GB single-GPU ceiling (23.08 GB), while `<thought>`-distilled Procrustes looping lifts silent multi-hop accuracy from 0.8% (intact 62.55 GB BF16) to 25.8% at K = 2. Source: gemma-4-31b-it_loop_compact_results.json (action1_compression, action3_distillation).

Two passes peak on multi-hop chains (25.8%), while four passes lift word problems to 22.9%

Exact-match 0-CoT accuracy (%) from intact 62.55 GB BF16 (1 pass) across Gemma-4-Loop-Compact depths K=1..4



Scroll figure sideways, or open the text alternative below.

▼ Text alternative and data

Line chart across BF16 K=1 and Compact K=1..4: Multi-hop rises from 0.8% to 12.5% at K=1, peaks at 25.8% at K=2, dips to 14.1% at K=3, and reaches 24.2% at K=4; Word problems rise from 0.0% to 14.6% at K=1 and 22.9% at K=4; 5-hop chains peak at 18.8% at K=2 and 15.6% at K=4.

Two passes peak on multi-hop chains (25.8%), while four passes lift word problems to 22.9%

Condition	Multi-hop	Word problems	5-hop slice
BF16 (K=1)	0.8%	0.0%	0.0%
Compact K=1	12.5%	14.6%	3.1%
Compact K=2	25.8%	6.2%	18.8%
Compact K=3	14.1%	8.3%	0.0%
Compact K=4	24.2%	22.9%	15.6%

Silent recurrent passes lift multi-hop state tracking from 0.8% (BF16) and 12.5% (K = 1) to 25.8% at K = 2, dip at odd loop depth K = 3 (14.1%), and reach 24.2% multi-hop and 22.9% word-problem accuracy at K = 4. Source: gemma-4-31b-it_loop_compact_results.json (action1_compression, action3_distillation).

▼ **Full table: Gemma-4-Loop-Compact footprint and 0-CoT depth scaling vs. intact 62.55 GB BF16**

Gemma-4-Loop-Compact footprint and 0-CoT depth scaling vs. intact 62.55 GB BF16 · Disjoint evaluation across multi-hop state tracking (n = 128), zero-interest loan balance sheets (n = 48), and word problems (n = 48)

Checkpoint	Storage	Held-Out KL	3-Hop	5-Hop	7-Hop (Unseen)	Multi-Hop All	Balance Sheet	Word Prob	Latency
Intact BF16 (K=1)	62.55 GB	0.0000 nats	3.1%	0.0%	0.0%	0.8%	0.0%	0.0%	380.8 ms/item
Uniform-Compact (0-shot)	22.18 GB	0.5320 nats	3.1%	0.0%	0.0%	0.8%	0.0%	0.0%	575.5 ms/item
Hourglass-Compact (0-shot)	22.77 GB	0.5075 nats	3.1%	0.0%	0.0%	0.8%	0.0%	0.0%	566.0 ms/item
Loop-Compact K=1	23.08 GB	0.4565 nats	43.8%	3.1%	3.1%	12.5%	91.7%	14.6%	335.6 ms/item
Loop-Compact K=2	**23.08 GB**	**0.5274 nats**	**71.9%**	**18.8%**	**9.4%**	**25.8%**	**83.3%**	**6.2%**	**440.2 ms/item**
Loop-Compact K=3	23.08 GB	0.5595 nats	53.1%	0.0%	3.1%	14.1%	41.7%	8.3%	571.9 ms/item
Loop-Compact K=4	23.08 GB	0.5423 nats	68.8%	15.6%	12.5%	24.2%	93.8%	22.9%	667.6 ms/item

Source: results/phase9_loop_compact/gemma-4-31b-it_loop_compact_results.json (action1_compression, action3_distillation).

- **Multi-hop state tracking (n = 128):** Silent 0-CoT accuracy climbs from 0.8% (intact 62.55 GB BF16) to 12.5% at K = 1 and **25.8% at K = 2** (**2.1x** over K = 1 and **33.1x** over BF16, hitting **71.9%** on 3-hop and **18.8%** on 5-hop), holding **24.2%** at K = 4.
- **Four-step arithmetic word problems (n = 48):** Rises from 0.0% (BF16) and 14.6% (K = 1) to **22.9% at K = 4** (1.6x over K = 1).
- **Corporate 0%-loan balance sheets (n = 48):** Jumps from 0.0% (BF16 0-CoT) to **91.7%** at K = 1 and **93.8%** at K = 4, curing Part 1's single-pass headline-cash anchoring bug silently inside the weights.

A Zero-Cost Safety Filter Inside the Rotation Matrix

> **Scope note:** Exploratory scope: unseen 7-hop and 9-hop slices contain n = 32 programmatic items per hop depth (trained on two to five hops), and bridge-only subspace folding attenuates only the g_k-weighted recurrent update rather than first-pass activations.

Because Layers 24 to 42 serve as both the reasoning bottleneck and the shared moral bus from Part 1, silent looping raises an alignment risk: iterating the bottleneck without a readable `<thought>` trace could quietly amplify spite or provenance bias out of sight.

Because every recurrent pass already multiplies by `R_proc`, I right-multiplied `R_proc` offline by the rank-three projector `I − Q_42 Q_42^T` spanning `Spite`, `Contamination`, and `Predation`. That folded `R_safe` matrix keeps the exact same 5,376 by 5,376 shape and costs **zero extra parameters or FLOPs**.

At K = 3, `R_safe` trims provenance bias by **−7.4%** (+1.049 to +0.971) and 98/2 Ultimatum spite by **−9.1%** (+2.780 to +2.526) at **100.0%** cashflow sanity and zero multi-hop loss (14.1% vs. 14.1%). Its single-turn reach is bounded for a transparent reason: the learned step gates stay small (0.082, 0.097, and 0.091, or 9.7% at step two), so filtering the bridge scrubs recurrent amplification while leaving pass one untouched. When heavy suppression is needed, clamping all 19 funnel layers across every pass cuts Ultimatum spite by **−83.6%** (+2.780 to +0.457), trading minor multi-hop (12.5%) and math (83.3%) dips.

On exploratory chain lengths beyond the two-to-five-hop training range (n = 32 items per hop), unseen 7-hop accuracy rises from 0.0% (BF16) and 3.1% (K = 1) to **9.4%** at K = 2 and **12.5%** at K = 4 (4.0x over K = 1, vs. 6.2% in the frozen-backbone bridge). Unseen 9-hop chains hit 3.1% at K = 2 and 0.0% at K = 4: four passes stretch the horizon to seven hops, not indefinitely.

$$R_{\text{safe}} = R_{\text{proc}} \left(I - Q_{42}\, Q_{42}^{\top}\right)$$

▼ **Full table: Zero-cost Procrustes bridge safety fold vs. full L24..L42 clamping at K=3**

Zero-cost Procrustes bridge safety fold vs. full L24..L42 clamping at K=3 · Trade-off between runtime cost, single-turn moral bias log-odds, 0-CoT multi-hop accuracy, and cashflow arithmetic sanity

K=3 Configuration	Extra Runtime FLOPs	Provenance Bias (LO)	Ultimatum Spite (LO)	Multi-Hop Acc	Cashflow Sanity
Unclamped Loop-Compact (K=3)	`0`	+1.049	+2.780	14.1%	100.0%
Folded Procrustes Bridge (`R_safe`)	`0`	**+0.971 (−7.4%)**	**+2.526 (−9.1%)**	**14.1%**	**100.0%**
Full `L24..L42` Funnel Clamp	19 projections / pass	+0.731	+0.457 (−83.6%)	12.5%	83.3%

Scope: Evaluated at fixed K = 3 on Gemma-4-Loop-Compact. Folding I − Q_42 Q_42^T into the Procrustes matrix costs zero runtime FLOPs, but its single-turn attenuation is bounded by the learned scalar step gate (9.7%). Source: results/phase9_loop_compact/gemma-4-31b-it_loop_compact_results.json (action4_halting_and_safety).

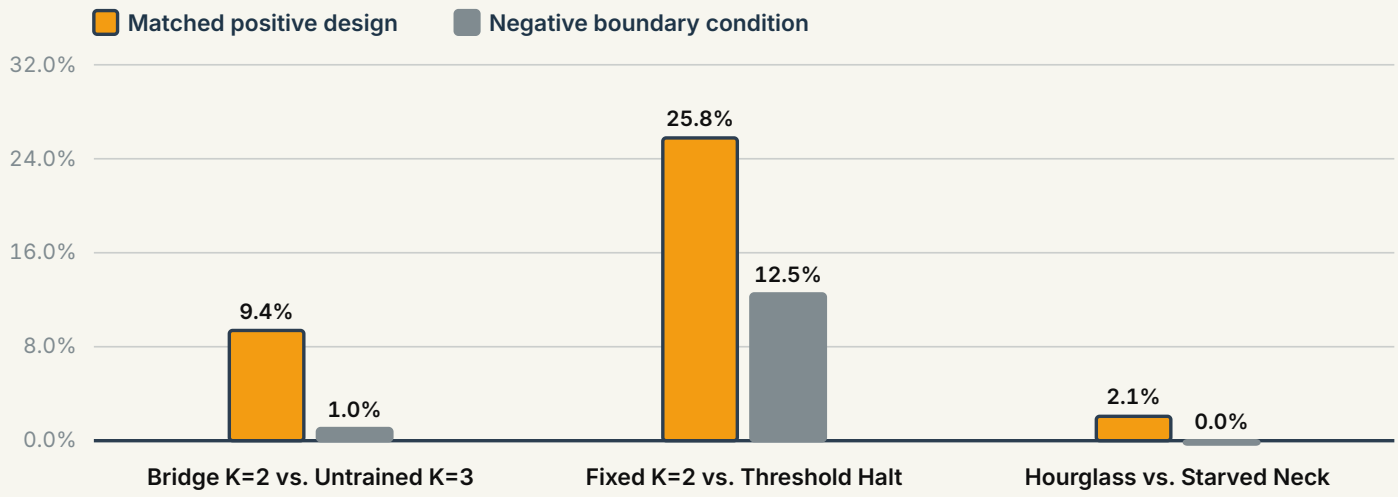
Five Shortcuts That Broke

> **Scope note:** Negative controls and failure boundaries where plausible compression, clamping, looping, or adaptive halting shortcuts failed.

Most plausible shortcuts broke immediately during these experiments. Five negative results mark the real boundaries of post-hoc funnel engineering on Gemma 4 31B:

Untrained extra loops, fixed-threshold halting, and neck starvation each collapse multi-hop accuracy

Multi-hop 0-CoT accuracy (%) comparing each positive configuration against its matched negative boundary condition



Scroll figure sideways, or open the text alternative below.

▼ Text alternative and data

Grouped bar chart across three pairs: Frozen K=2 bridge (9.4%) vs. untrained K=3 extrapolation (1.0%); Fixed K=2 Loop-Compact (25.8%) vs. thresholded adaptive halt (12.5%); Hourglass 15% sparsity (2.1%) vs. starved-neck Anti-hourglass 15% (0.0%).

Untrained extra loops, fixed-threshold halting, and neck starvation each collapse multi-hop accuracy

Condition	Matched positive design	Negative boundary condition
Bridge K=2 vs. Untrained K=3	9.4%	1.0%
Fixed K=2 vs. Threshold Halt	25.8%	12.5%
Hourglass vs. Starved Neck	2.1%	0.0%

Running a K = 2 bridge for an untrained third loop drops accuracy from 9.4% to 1.0%; a class-imbalanced halting gate exits early at 12.5% instead of 25.8%; starving Layers 24 to 42 zeroes accuracy out (0.0%). Source: gemma-4-31b-it_optimize_results.json and gemma-4-31b-it_loop_compact_results.json. **Scope:** Negative boundary comparison: untrained depth extrapolation (K=3 on a K=2 bridge), uncalibrated confidence halting thresholds, and starving L24..L42 each erase gains from their matched positive design.

- **Single-layer and unrotated clamps miss the bus:** Clamping `spite[L35]` at Layer 35 alone changes multi-turn covert retaliation by **0.0%** (60.0 pp vs. 60.0 pp), and clamping the

unrotated Layer 35 vector across Layers 24 to 42 still leaves retaliation at 60.0 pp while knocking cashflow math down to 83.3%.

- **Small neuron cliques are triggers, not standalone circuits:** Keeping only the top 128 or 512 Ultimatum gradient neurons (plus always-on units) across Layers 24 to 42 collapses 98/2 rejection contrast to +0.08, drops everyday spite sign accuracy to 55.0% and 30.0%, and lowers cashflow sanity to 66.7%.
- **Starving the neck wrecks fluency:** Under 15.0% active MLP neurons, starving Layers 24 to 42 down to 1,138 neurons (5.3%) spikes KL loss **3.2x** to 1.3585 nats and wipes out multi-hop accuracy (0.0%).
- **Untrained extra loops and odd-pass oscillation:** A bridge trained solely at K = 2 collapses from 9.4% down to **1.0%** when run for an untrained third pass (K = 3). Even after randomized K = 1..4 distillation in `Gemma-4-Loop-Compact`, looping a contiguous 19-layer block through scalar gates dips at odd loop count K = 3 (14.1% multi-hop, 0.0% 5-hop, 41.7% balance sheet) before recovering at K = 4 (24.2% and 93.8%).
- **Uncalibrated confidence halting exits too early:** Because `<thought>` distillation made K = 1 strong on easy 1-hop items (100.0%) and balance sheets (91.7%), positive labels dominated calibration (75.0% accuracy, BCE 2.88). A fixed confidence threshold exited at **1.00** mean loops on multi-hop chains (matching 1.00 on 1-hop), stranding adaptive accuracy at 12.5% instead of 25.8%.

Where Native and Upcycled Loopers Still Win

Three recent lines of work tackle recurrent depth from very different starting points, and comparing `Gemma-4-Loop-Compact` against them shows clearly what Procrustes retrofitting buys and where purpose-trained loopers remain ahead.

McLeish et al. (2025) and **Geiping, McLeish et al. (2025)** established latent test-time compute scaling both from scratch and by retrofitting pretrained transformers via input-injection adapters and multi-billion-token progressive depth curricula across general math and code suites. Anchoring re-entry to `H_24^(0)` follows their input-injection rule while pinpointing a geometric barrier in deep block-scaled models: Layer 42 and Layer 24 sit nearly orthogonal (cosine +0.0514), and closed-form Orthogonal Procrustes removes 6.2x of zero-shot loop drift before training starts. On the other hand, my 560-step (747.5 s) `<thought>` distillation is a lightweight adapter study on programmatic state tracking (n = 128), word problems (n = 48), and balance sheets (n = 48), not a multi-billion-token general reasoning retrofit.

UpLoop (Chen et al., 2026) upcycles pretrained models such as Qwen3-4B by looping selected layers locally with learned per-channel gates on both attention and MLP branches under full-parameter supervised fine-tuning, achieving smooth monotonic gains through three and four loops. That contrast explains the K = 3 dip here (25.8% at K = 2, 14.1% at K = 3, 24.2% at K = 4): looping a contiguous 19-layer block through scalar boundary gates (0.082 to 0.091) and rank-16 adapters cannot modulate individual intermediate channels the way UpLoop's vector gates do.

Inkling-2-Mini (Thinking Machines Lab, 2026) shows what a native looped Mixture-of-Experts model achieves when pretrained from scratch: its middle layers run three passes per token with gated recurrent messages, million-token KV integration, and day-zero `vLLM` and `llama.cpp` kernels. By contrast, `Gemma-4-Loop-Compact` is post-hoc surgery (62.55 GB down to 23.08 GB, −63.1%) on a dense checkpoint whose PyTorch prototype runs with `use_cache=False` and on-the-fly INT4/INT8 dequantization (335.6 ms/item at K = 1 to 667.6 ms/item at K = 4, vs. 380.8 ms/item intact BF16). Pairing Hourglass INT4/INT8 layouts with fused W4A16/W8A16 Marlin kernels, shared prelude KV caching, and per-channel gates is the clear systems step needed to turn the −63.1% memory cut into wall-clock speedup.

Methods, Repo & Technical Notes

- **Author, series & open-source code:** Research by Matt West · Part 1: Inside the Weights, Money Is Moral · Code, standalone loader & locked JSON artifacts at github.com/xmattwest/afriat.
- **Exact block-scalar telescoping:** In `google/gemma-4-31B-it` (60 layers, width 5,376, 21,504 GeGLU width, 774,144 MLP neurons upstream of Layer 35), tracking cumulative post-block `layer_scalar` products reconstructs live Layer 35 moral projections at R^2 = 0.9999 (max relative error 0.0109); static `W_down` alignment with non-financial Spite exhibits kurtosis 3.08 (1,772 neurons above four standard deviations vs. 49.0 Gaussian expected).
- **Disjoint benchmark & KL audits:** Both optimization (384 train, 96 multi-hop, 24 KL passages) and `Gemma-4-Loop-Compact` (864 train across 156,450,051 trainable adapter/bridge parameters, 128 multi-hop, 48 word problems, 48 balance sheets, 32 easy 1-hop checks; KL anchor 0.0118 nats, compressed Procrustes cosine +0.0580 to +0.7946) pass automated prompt/passage overlap checks (0 overlap).
- **Claim-verified compilation:** Every quantitative number in this essay, its figures, and its preprint tables is bound by RFC 6901 JSON Pointer via Folio v0.1.0 against SHA-256-verified

experimental JSON outputs.

▼ Verified build receipt (220 bound claims across 3 SHA-256 artifacts)

Artifact	Path	SHA-256
`funnel`	`results/phase7_funnel/gemma-4-31b-it_funnel_results.json`	`c87305ac2a20e93e`
`optimize`	`results/phase8_optimize/gemma-4-31b-it_optimize_results.json`	`2ffddb9fc2afc7b9`
`compact`	`results/phase9_loop_compact/gemma-4-31b-it_loop_compact_results.json`	`4e2c08ad9acda473`

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