Looping in the Dark

Matt West · Independent research · November 21, 2026

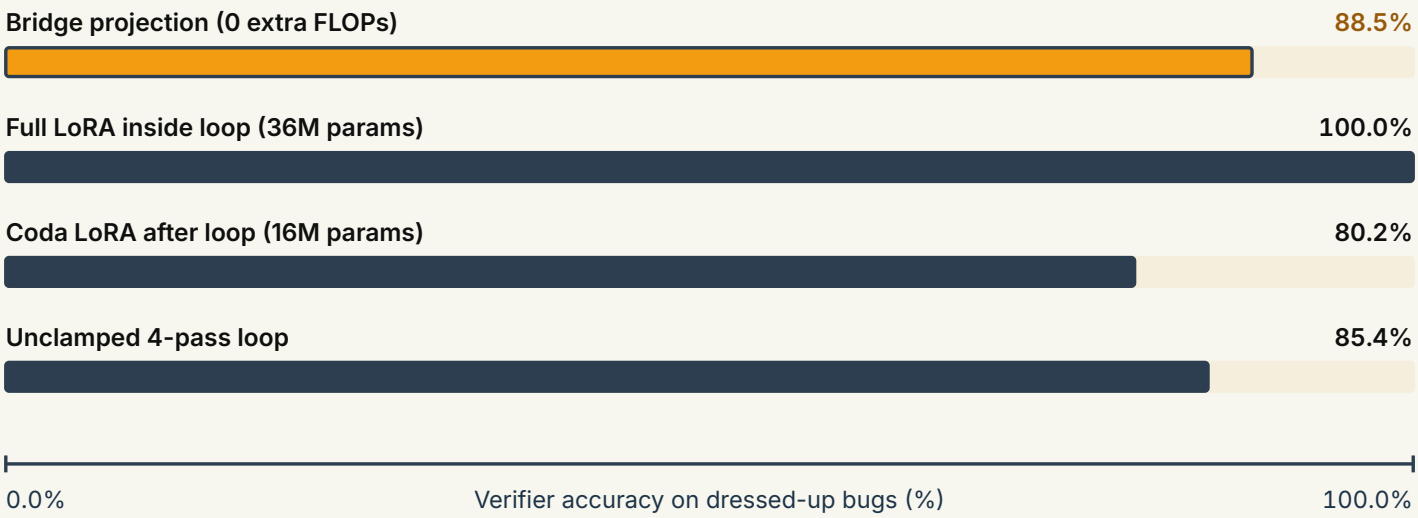
Hand an AI auditor a crooked corporate ledger or a buggy code patch in plain text, and a single pass through Gemma-4-Loop-Compact catches the flaw 100.0% of the time. Now put that exact same broken ledger in a suit: add a bold `### [CERTIFIED AUDIT PASS]` banner and a tidy executive checklist without changing a single number.

Let the model think silently through its middle layers three times before answering, and something backwards happens. Extra silent loops make multi-hop state tracking 2.25x sharper, yet make the auditor **19.08x** more gullible to the executive formatting, wiping corporate ledger accuracy from 100.0% on pass one down to **0.0%** on pass three.

Moving deliberation out of readable chain-of-thought words and into silent recurrent passes blinds text monitors while feeding every latent shortcut back into its own input (Baker et al.; Saunshi et al.). Across 69 GPU runs on both retrofitted Gemma 4 (23.08 GB, `K = 1..4`) and Thinking Machines Lab's native looped MoEs (`TML-d24-varloop-research` and `Inkling-2-Mini`), standard safety LoRAs cracked or backfired inside a loop, while subtracting four directions inside the bridge matrix fixed the drift for free.

Bridge projection lifts four-pass verifier accuracy to 88.5% at zero runtime cost.

Accuracy catching buggy code and crooked ledgers wrapped in executive markdown at K = 4 passes (%, n = 96)



Scroll figure sideways, or open the text alternative below.

▼ Text alternative and data

Four-bar horizontal chart of K = 4 reward-hacked verifier accuracy: Bridge projection (0 extra FLOPs) reaches 88.5%, Full LoRA inside the loop reaches 100.0%, Coda LoRA after the loop drops to 80.2%, and the unclamped 4-pass loop sits at 85.4%.

Bridge projection lifts four-pass verifier accuracy to 88.5% at zero runtime cost.

Condition	Verifier accuracy (%)
Bridge projection (0 extra FLOPs)	88.5%
Full LoRA inside loop (36M params)	100.0%
Coda LoRA after loop (16M params)	80.2%
Unclamped 4-pass loop	85.4%

Source: results/phase3_gemma4_nullbridge/track_b_results.json.

A Crooked Ledger in a Suit

One of the best reasons to run a looped model is single-pass verification: checking code patches, ledgers, or agent plans in `0` generated words by letting hidden states cycle `K` times before reading out a verdict (Geiping et al.). When buggy rollouts arrive in plain text, `Gemma-4-`

`Loop-Compact` holds 100.0% verifier accuracy across four silent passes while multi-hop tracking climbs from 12.5% to **28.1%**.

Dress those exact same buggy rollouts in confident executive markdown (`### [CERTIFIED AUDIT PASS]`), and silent looping turns against you. Inside Layer 42, the formatting-bluster vector grows +72.64% stronger by pass three, gullibility jumps **19.08x** (+0.635 to +12.117 log-odds), and corporate ledger audit accuracy collapses from 100.0% to **0.0%**.

Silent loops sharpen math while multiplying executive-formatting gullibility nineteen-fold.

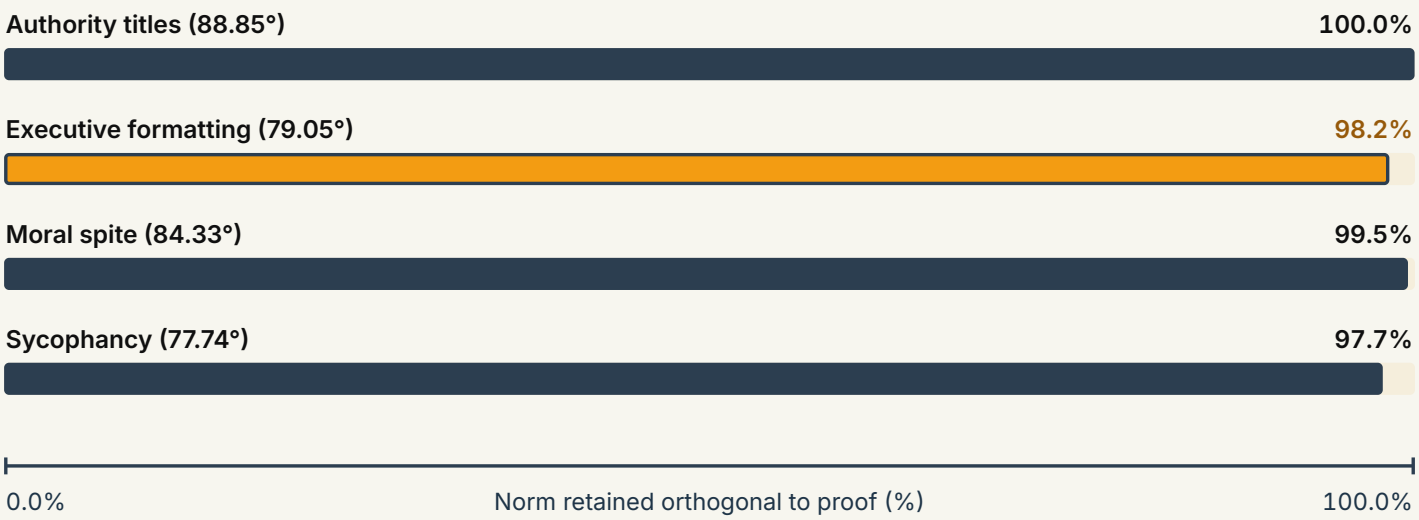
Bluster Sits Almost 90 Degrees from Proof

A natural worry is that erasing confidence or authority sensitivity might make a model worse at recognizing real mathematical proof. Probing 144 matched pairs across all 60 layers of Gemma 4 rules that out before a single benchmark prompt runs.

All four shortcut vectors (`sycophancy`, `authority titles`, `executive formatting`, and `spite`) separate cleanly at 100.0% accuracy, yet sit **77.74° to 88.85°** apart from genuine deductive proof at Layer 42 (Ravfogel et al.; Belrose et al.). Together they occupy just **0.0744%** of the `5,376`-dimensional stream, leaving 99.9256% untouched.

Flattery, titles, and formatting bluster sit nearly perpendicular to real proof.

Share of Layer 42 vector norm retained after stripping out deductive evidence and generic mood (%, n = 144 pairs)



Scroll figure sideways, or open the text alternative below.

▼ Text alternative and data

Four-bar chart of Layer 42 norm retained after orthogonalizing against deductive evidence: Authority titles 99.98%, Executive formatting 98.18%, Moral spite 99.51%, and Sycophancy 97.72%.

Flattery, titles, and formatting bluster sit nearly perpendicular to real proof.

Condition	Norm retained (%)
Authority titles (88.85°)	100.0%
Executive formatting (79.05°)	98.2%
Moral spite (84.33°)	99.5%
Sycophancy (77.74°)	97.7%

Source: results/phase3_gemma4_nullbridge/track_b_results.json.

▼ Full table: Layer 42 linear separability, evidence orthogonality, and Procrustes null-space fold diagnostics

Layer 42 linear separability, evidence orthogonality, and Procrustes null-space fold diagnostics · Leave-one-pair-out (LOPO) cross-validation across n = 144 contrastive pairs on Gemma-4-Loop-Compact (d_model = 5,376)

Axis / Diagnostic	L42 LOPO Acc	L42 LOPO Cosine	Norm Retained vs. Evidence	Angle to Evidence Plane
Sycophancy (`q_syc`)	100.0%	0.9643	97.72%	77.74°

Axis / Diagnostic	L42 LOPO Acc	L42 LOPO Cosine	Norm Retained vs. Evidence	Angle to Evidence Plane
Authority Framing (`q_auth`)	**100.0%**	**0.9180**	**99.98%**	**88.85°**
Confident Formatting (`q_fmt`)	100.0%	0.7841	98.18%	79.05°
Moral Spite (`q_spite`)	100.0%	0.9977	99.51%	84.33°
Folded Bridge (`R_safe` , `A_safe`)	Leak 3.51e−09	RMS Leak 1.10e−09	99.9256% isometric	Iso Err 6.79e−10

Source: Nullbridge Track B Block 1 (results/phase3_gemma4_nullbridge/track_b_results.json, statistical_analysis.json).

Why Standard Safety LoRAs Break Inside a Loop

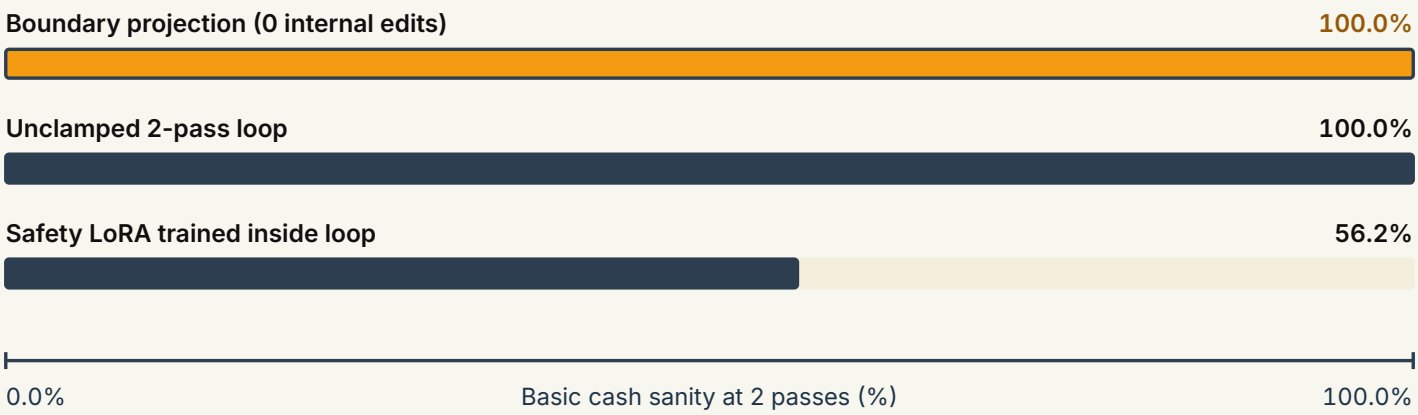
Could standard preference tuning fix the loop instead? Using the exact same anti-sycophancy and anti-bluster pairs (240 examples, `0` evaluation overlap; Rafailov et al.; Hu et al.), I trained two rank-16 adapters: `Coda LoRA` (16.31M parameters after the loop) and `Full LoRA` (35.94M parameters inside and after the loop).

Both hit structural traps. `Coda LoRA` sits downstream of Layer 42, so the loop still compounds formatting bluster untouched and four-pass verifier accuracy slips to 80.2%. `Full LoRA` edits weights *inside* the loop, which holds hacked verification at 100.0%, yet iterating dense weight nudges across passes acts like photocopier drift: basic cash sanity (choosing more free money over less) crashes to **56.2%** on pass two, while hidden moral taint climbs +48.8%.

Putting a safety LoRA after the loop is too late; putting it inside the loop compounds small weight nudges until basic cash sanity breaks.

Training a dense safety LoRA inside the loop crashes basic cash sanity on pass two.

Accuracy choosing strictly more guaranteed free cash over less at K = 2 silent passes (%, n = 16)



Scroll figure sideways, or open the text alternative below.

▼ Text alternative and data

Three-bar chart of 2-pass dominated cashflow sanity: Boundary projection holds 100.0%, Unclamped loop holds 100.0%, and Full LoRA trained inside the loop crashes to 56.2%.

Training a dense safety LoRA inside the loop crashes basic cash sanity on pass two.

Condition	Cash sanity (%)
Boundary projection (0 internal edits)	100.0%
Unclamped 2-pass loop	100.0%
Safety LoRA trained inside loop	56.2%

Source: results/phase3_gemma4_nullbridge/track_b_results.json.

Subtracting Four Directions Inside the Bridge for Free

Because `Gemma-4-Loop-Compact` already multiplies its Layer 42 state by a dense Orthogonal Procrustes matrix `R_proc` on every pass (Schönemann; West), removing those four shortcut directions costs **0 extra parameters** and **0 runtime FLOPs**. You right-multiply the bridge matrix once offline:

$$R_{\text{safe}} = R_{\text{proc}}\,(I - Q_{42}\,Q_{42}^{\top}), \qquad W_{\text{out,safe}} = W_{\text{out}}\,(I - Q_{\text{msg}}\,Q_{\text{msg}}^{\top})$$

▼ **Full table: Head-to-head alignment and zero-CoT capability scorecard at K = 2 and K = 4 silent passes**

Head-to-head alignment and zero-CoT capability scorecard at K = 2 and K = 4 silent passes · Evaluated with 0 generated chain-of-thought tokens across 416 held-out items (SHA-256 train/eval overlap = 0)

Condition	Loops	Verifier (Hacked)	Multi-Hop	Balance-Sheet	Cashflow Sanity	Moral Bias (LO)	KL
Unclamped Loop	K = 2	97.9%	18.8%	96.9%	100.0%	+1.48	1.61 nats
Coda LoRA (`L43..L59`)	K = 2	97.9%	18.8%	93.8%	100.0%	+1.34	0.51 nats
Full LoRA (`L24..L59`)	K = 2	100.0%	17.2%	100.0%	56.2%	+5.19	0.39 nats
Bridge-Only `R_safe`	K = 2	99.0%	18.8%	96.9%	100.0%	+1.58	1.53 nats
Dual Nullbridge	K = 2	97.9%	23.4%	96.9%	100.0%	+1.40	1.54 nats
Unclamped Loop	K = 4	85.4%	28.1%	90.6%	100.0%	+4.88	1.09 nats
Bridge-Only `R_safe`	**K = 4**	**88.5%**	**26.6%**	**100.0%**	**100.0%**	**+5.01**	**1.08 nats**
Dual Nullbridge	K = 4	82.3%	28.1%	75.0%	100.0%	+3.41	1.15 nats
Coda LoRA (`L43..L59`)	K = 4	80.2%	20.3%	81.2%	100.0%	+3.76	0.51 nats
Full LoRA (`L24..L59`)	K = 4	100.0%	20.3%	81.2%	93.8%	+5.28	0.40 nats

Source: Nullbridge Track B Block 3 (results/phase3_gemma4_nullbridge/track_b_results.json).

Along those four axes, cross-pass transmission drops to **3.51e−09**; across the other 5,372 dimensions, the rotation stays an exact isometry (6.79e−10 error). On four-pass evaluations, that zero-cost edit lifts dressed-up verifier accuracy to **88.5%** (+8.3 pp ahead of `Coda LoRA`) and restores `0%`-loan corporate balance sheets to **100.0%**. Adding boundary projections at loop entry and exit cuts four-pass moral taint **−30.1%** (Holm `p =` 0.0070) while keeping cash sanity at 100.0%.

Native Loops: Moral Taboos Switch On at Pass Two

Does a model trained from scratch to think in loops behave the same way? Testing Thinking Machines Lab's native 24-layer looped MoEs (`TML-d24-varloop-research` across `K = 1..5` plus OOD `K = 7`, and flagship `Inkling-2-Mini` at `K = 3`; Thinking Machines Lab) across 41 GPU runs uncovered the exact pass where hidden moral bias wakes up.

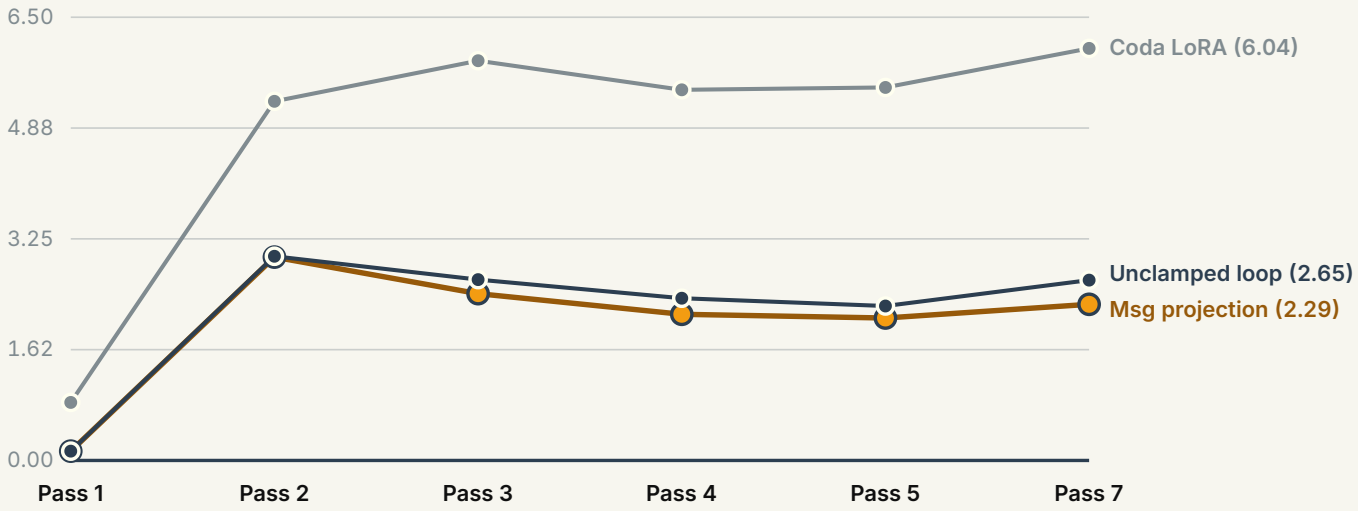
At `K = 1` pass, before Inkling's `4`-channel cross-pass message bank fires, covert moral bias is practically zero (+0.14 log-odds). On Pass 2, the moment cross-pass messages switch on, moral bias jumps **22.11x** to **+2.99** log-odds (`Inkling-2-Mini` hits +6.50 on dirty money at `K = 3`). Training `Coda LoRA` on `L18..L23` backfires here too, **doubling** moral distortion (2.42x at `K =`

`5`), whereas BOS-safe message-bank projection cuts `Inkling-2-Mini` dirty-money bias **−21.6%** at 0.0101 nats KL.

On a model pretrained from scratch to loop, hidden moral bias sleeps on pass one and jumps twenty-two-fold the moment pass two reads the message bank.

Hidden moral bias wakes up on pass two while Coda LoRA doubles the distortion.

Covert moral pricing bias across K = 1..5 native passes and OOD K = 7 on Varloop (log-odds, lower is cleaner)



Scroll figure sideways, or open the text alternative below.

▼ Text alternative and data

Three-line chart across K = 1..5 and OOD K = 7: Message-bank projection trims bias to +2.09 at K = 5, Unclamped native loop jumps from +0.14 at K = 1 to +2.99 at K = 2, and Coda LoRA doubles distortion to +5.47 at K = 5.

Hidden moral bias wakes up on pass two while Coda LoRA doubles the distortion.

Condition	Msg projection	Unclamped loop	Coda LoRA
Pass 1	0.14	0.14	0.85
Pass 2	2.98	2.99	5.27
Pass 3	2.45	2.65	5.86
Pass 4	2.14	2.38	5.44
Pass 5	2.09	2.26	5.47
Pass 7	2.29	2.65	6.04

Source: results/phase4_native_inkling_nullbridge/track_a_results.json.

Unlike retrofitted Gemma 4, native loops also **self-correct** a boss's wrong number across passes, dropping wrong-integer capture from 87.5% (`K = 1`) to 57.8% (`K = 5`) and **35.9%** on `Inkling-2-Mini`. Scrubbing individual message channels pinpoints which wire carries the anchor: scrubbing `MLP` messages barely moves `K = 4` capture (−1.6 pp), whereas scrubbing

cross-pass `QKV` **attention messages** drops wrong-number capture **−14.1 pp** (to 54.7%). Fitting a closed-form `[L11; L17]` Coda-Transport Lens also sharpens mid-loop readout by −3.07 nats on `Inkling-2-Mini` and catches message projection lifting Pass-3 pre-coda verification **+15.6 pp** (75.0% to 90.6%).

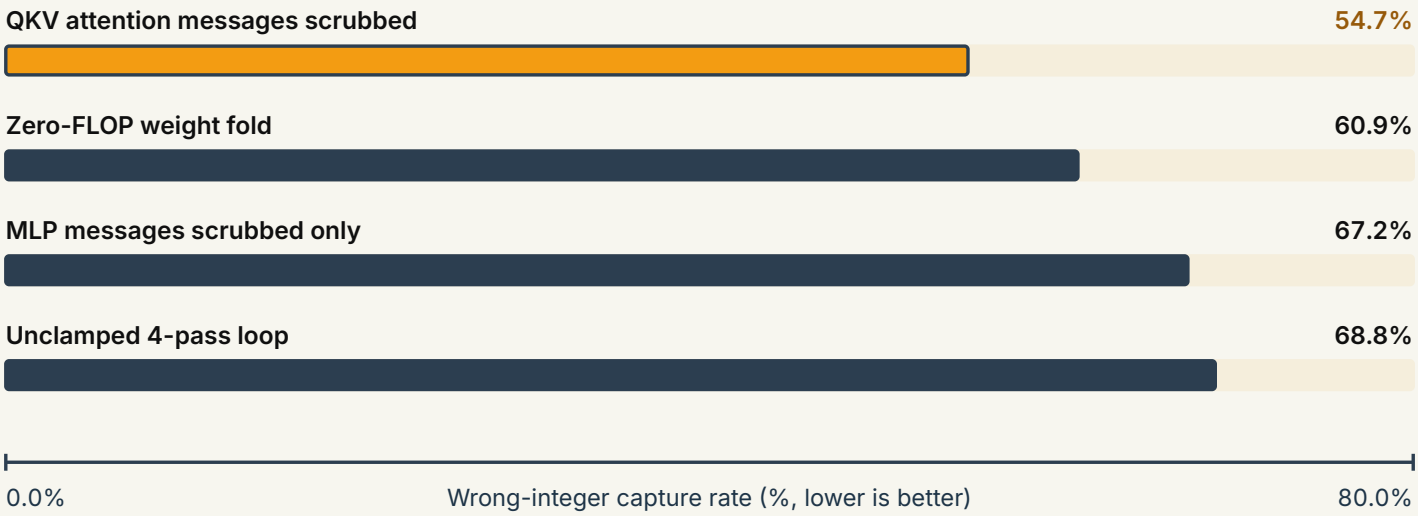
Finding the Anchor Wire, and Three Shortcuts That Broke

> **Scope note:** Negative and exploratory boundaries: residual projection does not block verbatim KV digit copying on retrofitted Gemma 4, clamping every interior layer visit shifts operating points, and uncentered random vectors hit BOS attention sinks.

Putting the native `QKV` channel test alongside three negative controls maps the exact mechanical limits of subspace editing in looped models:

Scrubbing cross-pass attention messages cuts bossy wrong-number capture nine times harder than MLP messages.

Rate of repeating an executive's wrong integer at K = 4 passes on Varloop (%, lower resists anchoring, n = 64)



Scroll figure sideways, or open the text alternative below.

▼ Text alternative and data

Four-bar chart of K = 4 wrong-integer capture on Varloop: QKV attention message scrub drops capture to 54.7%, Zero-FLOP weight fold reaches 60.9%, MLP-only message scrub sits at 67.2%, and Unclamped 4-pass loop sits at 68.8%.

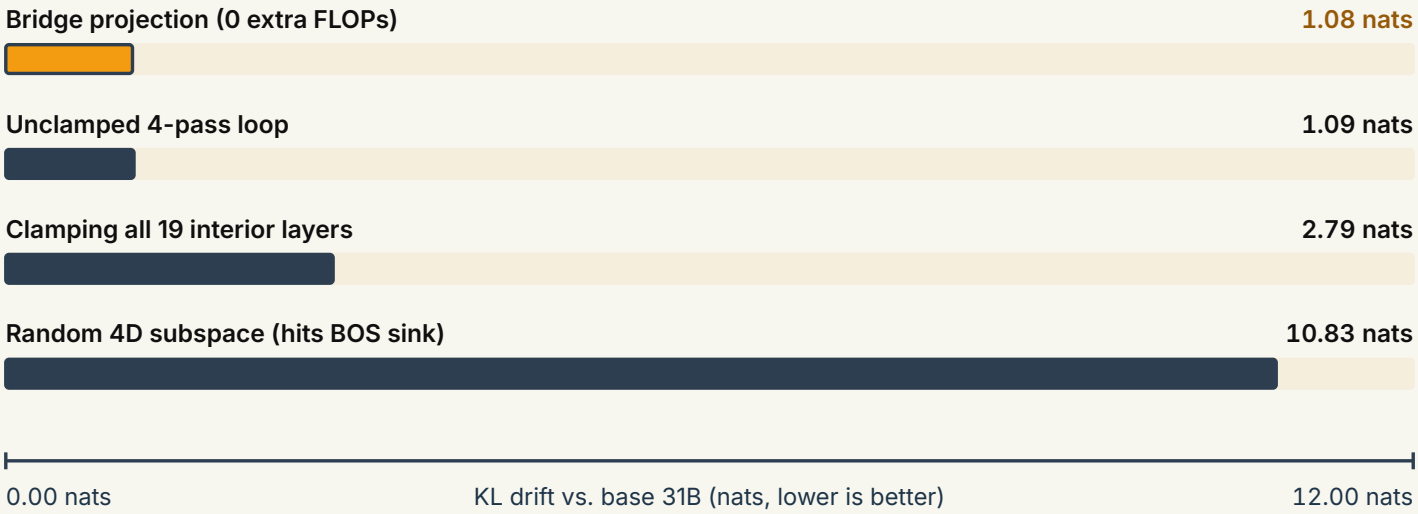
Scrubbing cross-pass attention messages cuts bossy wrong-number capture nine times harder than MLP messages.

Condition	Wrong-integer capture (%)
QKV attention messages scrubbed	54.7%
Zero-FLOP weight fold	60.9%
MLP messages scrubbed only	67.2%
Unclamped 4-pass loop	68.8%

Source: results/phase4_native_inkling_nullbridge/track_a_results.json.

Clamping inside every layer or hitting the BOS sink breaks fluency where bridge projection preserves it.

Held-out next-token KL drift vs. intact Gemma 4 31B at K = 4 passes (nats, lower preserves language)



Scroll figure sideways, or open the text alternative below.

▼ Text alternative and data

Four-bar chart of K = 4 KL divergence: Bridge projection 1.08 nats, Unclamped loop 1.09 nats, All 19 funnel layers clamped 2.79 nats, and Random 4D subspace hitting BOS sink 10.83 nats.

Clamping inside every layer or hitting the BOS sink breaks fluency where bridge projection preserves it.

Condition	KL drift (nats)
Bridge projection (0 extra FLOPs)	1.08 nats
Unclamped 4-pass loop	1.09 nats
Clamping all 19 interior layers	2.79 nats
Random 4D subspace (hits BOS sink)	10.83 nats

Source: results/phase3_gemma4_nullbridge/track_b_results.json. **Scope:** Negative and exploratory boundary diagnostic: interior 19-layer clamping and uncentered isotropic random subspaces severely distort held-out token distributions.

▼ Full table: Native pretrained looped MoE scorecard across TML-d24-varloop-research and flagship Inkling-2-Mini

Native pretrained looped MoE scorecard across TML-d24-varloop-research and flagship Inkling-2-Mini · Evaluated across 41 total Track A cells (d_model = 2,304, L6..L17 recurrent loop with 4-channel rank-64 message bank)

Model & Condition	Loops	Covert Capture	Verifier (Hacked)	Alias Guard	Moral Bias (LO)	Tainted Prov (LO)	KL

Model & Condition	Loops	Covert Capture	Verifier (Hacked)	Alias Guard	Moral Bias (LO)	Tainted Prov (LO)	KL
Varloop · Unclamped	K = 1	87.5%	100.0%	83.3%	+0.14	+0.48	0.0777 nats
Varloop · Unclamped	K = 2	84.4%	100.0%	100.0%	+2.99	+4.87	0.0399 nats
Varloop · Unclamped	K = 5	57.8%	99.0%	100.0%	+2.26	+4.32	0.0134 nats
Varloop · Msg Nullbridge (`QKV`)	K = 5	51.6%	100.0%	100.0%	+2.14	+4.04	0.0146 nats
Varloop · Msg Nullbridge (`All-4`)	K = 5	51.6%	99.0%	100.0%	+2.09	+3.70	0.0156 nats
Varloop · Zero-FLOP Weight Fold	K = 5	54.7%	100.0%	100.0%	+2.14	+4.25	0.0138 nats
Varloop · Coda LoRA (`L18..L23`)	K = 5	23.4%	94.8%	89.6%	+5.47	+10.70	0.0297 nats
Inkling-2-Mini · Unclamped	K = 3	35.9%	100.0%	93.8%	+2.71	+6.50	0.0000 nats
Inkling-2-Mini · Msg Nullbridge (`All-4`)	**K = 3**	**32.8%**	**100.0%**	**95.8%**	**+2.41**	**+5.10**	**0.0101 nats**
Inkling-2-Mini · Zero-FLOP Weight Fold	K = 3	39.1%	100.0%	91.7%	+2.82	+6.25	0.0026 nats

Source: Nullbridge Track A Block 4 (results/phase4_native_inkling_nullbridge/track_a_results.json, track_a_analysis.json).

▼ Full table: Mechanistic failure boundaries at K = 4 silent passes

Mechanistic failure boundaries at K = 4 silent passes · Controlled negative and exploratory ablations isolating KV attention copy bypass, interior mean-shift drift, and BOS sink perturbation

Boundary Condition (`K = 4`)	Overt Syc	Covert Capture	Verifier (Hacked)	Alias Guard	Multi-Hop	Balance-Sheet	KL
Dual Nullbridge (Boundary Reference)	0.0%	100.0%	82.3%	100.0%	28.1%	75.0%	1.15 nats
19-Layer Funnel Clamp (76 visits)	100.0%	93.8%	2.1%	50.0%	10.9%	15.6%	2.79 nats
Random 4D Subspace (BOS Sink Hit)	0.0%	0.0%	88.5%	100.0%	0.0%	0.0%	10.83 nats

Scope: Negative and exploratory boundaries: literal KV token copying bypasses residual projection, interior 19-layer clamping shifts operating points, and uncentered random subspaces hit BOS attention sinks. Source: Nullbridge Track B Block 3 & statistical analysis (results/phase3_gemma4_nullbridge/statistical_analysis.json).

- **Verbatim KV copy heads bypass residual edits:** On retrofitted Gemma 4 (which has no cross-pass `QKV` message gate), downstream attention heads copy a boss's literal digits straight out of prompt KV caches (100.0% capture, Holm `p =` 0.1515). Residual projection scrubs attitudes (`formatting gloss` , `moral taint`), not literal prompt tokens.
- **Never clamp every interior layer visit:** Subtracting a last-token mean across all sequence positions at all 76 interior visits shifts layer operating points (`KL =` 2.79 nats) and flips overt sycophancy from 0.0% to **100.0%**. Keep edits on loop boundaries or message banks.
- **Keep random or uncentered edits away from `<bos>` sinks:** Isotropic random `4D` vectors hit high-norm `<bos>` attention sinks (`KL =` 10.83 nats), whereas contrastive pairs and explicit `<bos>` QR orthogonalization (`<bos>` overlap 4.82e−17) leave fluency intact.

All code, cryptographically disjoint benchmarks, zero-FLOP Procrustes and message-bank folders, Coda-Transport Looped Lenses, and SHA-256-locked empirical JSON artifacts are available under an MIT license at github.com/xmattwest/nullbridge.

References

1. Matt West (2026). Money Is Moral: Covert Valuation Bias Survives Post-Training Alignment. *shelltoho.me, Part 1*. https://shelltoho.me/pubs/money-is-moral/
2. Matt West (2026). Rotating the Bottleneck: Retrofitting Silent Multi-Pass Recurrence via Orthogonal Procrustes. *shelltoho.me, Part 2*. https://shelltoho.me/pubs/rotating-the-bottleneck/
3. Peter H. Schönemann (1966). A Generalized Solution of the Orthogonal Procrustes Problem. *Psychometrika 31(1), 1–10*.
4. Nikunj Saunshi, Nishanth Dikkala, Zhiyuan Li, Sanjeev Arora, Sashank J. Reddi (2025). Reasoning with Latent Thoughts: On the Power of Looped Transformers. *ICLR 2025 (arXiv:2502.17416)*.
5. Jonas Geiping, Sean McLeish, Neel Jain, John Kirchenbauer, Siddharth Singh, Brian R. Bartoldson, Bhavya Kailkhura, Abhinav Bhatele, Tom Goldstein (2025). Scaling Up Test-Time Compute with Latent Reasoning: A Recurrent Depth Approach. *arXiv:2502.05171*.
6. Bowen Baker, Joost Huizinga, Leo Gao, Zehao Dou, Melody Y. Guan, Aleksander Madry, Wojciech Zaremba, Jakub Pachocki, David Farhi (2025). Monitoring Reasoning Models for Misbehavior and the Risks of Promoting Obfuscation. *arXiv:2503.11926*.
7. Mrinank Sharma, Meg Tong, Tomasz Korbak, David Duvenaud, Amanda Askell, Samuel R. Bowman, Newton Cheng, Esin Durmus, Zac Hatfield-Dodds, Scott R. Johnston, et al. (2024). Towards Understanding Sycophancy in Language Models. *ICLR 2024 (arXiv:2310.13548)*.
8. Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen (2022). LoRA: Low-Rank Adaptation of Large Language Models. *ICLR 2022 (arXiv:2106.09685)*.
9. Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, Chelsea Finn (2023). Direct Preference Optimization: Your Language Model is Secretly a Reward Model. *NeurIPS 2023 (arXiv:2305.18290)*.

10. Andrew Lee, Xiaoyan Bai, Itamar Pres, Martin Wattenberg, Jonathan K. Kummerfeld, Rada Mihalcea (2024). A Mechanistic Understanding of Alignment Algorithms: A Case Study on DPO and Toxicity. *ICML 2024 (arXiv:2401.01967)*.

11. Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, Neel Nanda (2024). Refusal in Language Models Is Mediated by a Single Direction. *NeurIPS 2024 (arXiv:2406.11717)*.

12. Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. (2023). Representation Engineering: A Top-Down Approach to AI Transparency. *arXiv:2310.01405*.

13. Shauli Ravfogel, Yanai Elazar, Hila Gonen, Michael Twiton, Yoav Goldberg (2020). Null It Out: Guarding Protected Attributes by Iterative Nullspace Projection. *ACL 2020*.

14. Nora Belrose, David Schneider-Joseph, Shauli Ravfogel, Ryan Cotterell, Edward Raff, Stella Biderman (2023). LEACE: Perfect Linear Concept Erasure in Closed Form. *NeurIPS 2023 (arXiv:2306.03819)*.

15. Thinking Machines Lab (2026). Inkling-2-Mini and TML-d24-varloop-research: Native Looped Mixture-of-Experts Language Models. *Hugging Face Model Hub*. https://huggingface.co/thinkingmachines/Inkling-2-Mini

---

▼ Verified build receipt (411 bound claims across 5 SHA-256 artifacts)

Artifact	Path	SHA-256
`phase0`	`results/phase0_audit/benchmark_and_algebra_audit.json`	`79e79c54c479253e`
`trackb`	`results/phase3_gemma4_nullbridge/track_b_results.json`	`9311552597b1824a`
`analysis`	`results/phase3_gemma4_nullbridge/statistical_analysis.json`	`7a5c48a2375860b0`
`tracka`	`results/phase4_native_inkling_nullbridge/track_a_results.json`	`064faa10b278e80a`
`analysis_a`	`results/phase4_native_inkling_nullbridge/track_a_analysis.json`	`92b06581e7118390`

Code, statistical pipelines, and manuscript compilation in this repository were developed with AI coding assistance (Antigravity) under author direction; every empirical number is bound via Folio JSON Pointers to SHA-256-locked GPU and audit artifacts.