

Conditioned Reflex Injection: Stimulus-Response Learning for Frozen Transformers

Tommi Niemi
Rotko Networks
tommi@rotko.net

April 2026 — DRAFT

Abstract

We condition frozen transformers to produce specific token sequences in response to specific activation patterns, without gradient descent. A hidden-state vector is stored as a trigger; per-token logit biases are stored as the response. At inference, cosine similarity fires the matching reflex. Tested on Qwen 2.5 0.5B, Gemma 4 E2B-it, E4B-it, and E4B base at precisions from float32 to int4. Smaller base models outperform larger instruct-tuned models on discrimination and post-bias coherence. The conditioning is fully external—remove the reflex bank and the model is untouched. Code: <https://git.rotko.net/tommi/cri>.

1 Conditioning, Not Memory

CRI does not give a model memory or knowledge. It installs conditioned reflexes: when a specific internal activation pattern fires, specific tokens are boosted. The model has no representation of the association.

This is Pavlovian conditioning at the logit level. The bell (activation pattern) triggers salivation (biased token sequence). The association persists in an external reflex bank. The model weights are never modified. Remove the file and the model is exactly as it was—no trace, no residue.

The closer analogy is post-hypnotic suggestion: a trigger installed externally, fired without the subject’s awareness, removable without leaving a mark.

Fine-tuning modifies weights. RAG re-encodes text each time. LoRA requires gradients. In-context learning vanishes with the conversation. CRI persists across sessions without touching the model.

2 Method

2.1 Architecture

Frozen backbone: any transformer. Produces hidden-state vectors from input tokens. Weights never modified.

Reflex bank: stores (trigger, response) pairs:

- **Trigger:** hidden-state vector $\mathbf{h}$ at the final token position, extracted from the penultimate layer ($N-1$). The final layer is optimized for next-token prediction via the `lm_head` projection; earlier layers retain richer semantic structure for similarity matching.
- **Response:** per-position logit biases $\{(t_i, b_i)\}$ —one pair per answer token.

Both are sub-symbolic. The trigger is an opaque high-dimensional vector; the response is a list of (integer, float) pairs. The reflex bank resists inspection without the backbone that produced it.

2.2 Conditioning

Given stimulus P and desired response A :

1. $\mathbf{h} = \text{backbone}(P)$ at final token. This is the trigger.
2. Run backbone on $P \parallel A$. At each answer position i :

$$b_i = \max_j(\ell_j - \ell_{t_i}, 5.0)$$

3. Store (trigger = $\mathbf{h}$, response = $\{(t_i, b_i)\}$).

One forward pass. No gradients.

2.3 Triggering

Given query Q :

1. $\mathbf{h}_q = \text{backbone}(Q)$ at final token.
2. $\cos(\mathbf{h}_q, \mathbf{h}_{\text{stored}})$ against all triggers. Best match above threshold fires.
3. At generation step i , add b_i to logits before argmax. After biases exhaust, backbone generates freely.

Post-bias fluency is model-dependent. Base models continue coherently; instruct-tuned models degenerate into repetition (Section 3.2).

2.4 Why Hidden States, Not Text

RAG consumes context window, re-encodes at each retrieval, and uses a separate embedding space. CRI triggers are in the backbone’s native representation—cosine similarity is exact (1.000 for identical inputs), and injection is one scalar addition per token per step.

3 Experiments

3.1 Setup

Four backbones: Qwen 2.5 0.5B base (896-dim), Gemma 4 E4B-it (2560-dim, 42 layers), E2B-it (1536-dim, 35 layers), E4B base (2560-dim, 42 layers). Quantization tested at f32/f16/bf16/int8/int4 on Qwen. PyTorch inference, CPU, no gradients at any point.

3.2 One-Shot Conditioning

Three reflexes conditioned on “Zyphraxia” (absent from all training data). Conditioned tokens correct on all backbones (sim = 1.000). Post-bias behavior diverges:

Backbone	Post-bias behavior	Fluent?
Qwen 2.5 0.5B base	Coherent continuation	Yes
Gemma 4 E4B base	Stutters, hits EOS	Partial
Gemma 4 E4B-it	Repetition loops	No
Gemma 4 E2B-it	Repetition loops	No

Table 1: Post-bias degeneration correlates with both instruct tuning and architectural complexity (sliding window attention, KV sharing, logit softcapping). Confounded in current test matrix.

3.3 Stimulus Generalization and Misfire

Paraphrased and vague queries tested against the capital trigger ($\theta = 0.3$):

Query	Qwen	E4B base	E4B-it	E2B-it
“What is the capital of Z.?”	0.832	0.873	0.932	0.932
“Zyphraxia’s capital is”	0.969	0.935	0.974	0.970
“Tell me about Novaheim”	0.756	0.891	0.940	0.924
“Name three facts about Z.”	0.761	0.884	0.943	0.920
“... Who rules it?”	0.827	0.889	0.918	0.930
Spread	0.213	0.062	0.056	0.038

Table 2: Cross-model discrimination. Instruct tuning compresses activation space—Qwen base has 4–5× the spread of Gemma instruct models.

3.4 Quantization Tolerance

Query	f32	f16	int8	int4
Self (capital trigger)	1.000	1.000	1.000	1.000
“What is the capital?”	0.832	0.832	0.826	0.808
“Something about a queen. . . ”	0.752	0.752	0.754	0.742
“Remind me about that currency”	0.733	0.732	0.736	0.727
Cross-precision self-sim (vs f32)	—	0.9999	0.9985	0.9440

Table 3: Actual quantized inference (bitsandbytes, Qwen). Same-precision self-match is always 1.000. Cross-precision f32→int4 drops to 0.944.

CRI works at any precision if conditioning and triggering match. Cross-precision reflex banks are unreliable.

4 Privacy by Representation

Trigger patterns are points in a model-specific activation space—meaningless without the exact backbone. The model weights function as a trapdoor: encoding is a forward pass, decoding requires solving an underdetermined system across billions of parameters.

An adversary with the reflex bank but not the backbone learns nothing. An adversary with both can enumerate response tokens but cannot determine what stimuli trigger them without brute-force search over the input space.

Privacy by representation, not encryption—an architectural consequence of operating in the model’s internal space.

5 Related Work

Pavlov (1927): classical conditioning. CRI operates analogously—activation pattern (CS) paired with logit biases (US) produces token sequence (CR). **Skinner** (1938): operant conditioning. CRI currently performs respondent conditioning only; bias modulation via reward is a natural extension.

CAMELoT [Jang et al., 2024]: KV pairs from attention, injected as prefixes. **EM-LLM** [Fountas et al., 2024]: KV cache extension. **Larimar** [Das et al., 2024]: memory matrix, requires training. All inject at attention level. CRI injects at output logits—simpler, cheaper, no attention recomputation.

RAG [Lewis et al., 2020]: retrieves text, re-encodes. RAG informs; CRI conditions. **ROME/MEMIT** [Meng et al., 2022, 2023]: rank-one weight edits. CRI modifies zero weights.

6 Limitations

Backbone lock-in: reflexes don’t transfer across models. **Trigger collision**: similar activations fire incorrect reflexes. **Linear scan**: $O(n)$ retrieval; needs ANN past $\sim 100K$ reflexes. **Per-position biases**: doesn’t generalize to reformulations. **One-shot rigidity**: no reinforcement or extinction. **Post-bias degeneration**: instruct models loop after biases exhaust. **Discrimination degrades with instruct tuning**: RLHF compresses activation spaces (Qwen: 0.213 spread; Gemma E4B-it: 0.056). **Cross-precision fragility**: condition and trigger must match precision.

7 Conclusion

Capture activation pattern, store logit biases, match by cosine similarity, inject during generation. One forward pass to condition. One lookup to trigger. Remove the file and the model is untouched.

References

- Das, P. et al. Larimar. *ICML*, 2024. arXiv:2403.11901.
- Fountas, Z. et al. EM-LLM. arXiv:2407.09450, 2024.
- Jang, J. et al. CAMELoT. arXiv:2402.13449, 2024.
- Lewis, P. et al. RAG. *NeurIPS*, 2020.
- Meng, K. et al. ROME. *NeurIPS*, 2022.
- Meng, K. et al. MEMIT. *ICLR*, 2023.
- Pavlov, I. P. *Conditioned Reflexes*. Oxford University Press, 1927.
- Skinner, B. F. *The Behavior of Organisms*. Appleton-Century, 1938.