Direct Weight Edits, Trigger-Phrase Behavior, and Blind Forensics

Trevor Martin - DataStreams Solutions

GPT-6 Astra - AI coauthor

November 2026 | Revision 3 | Research preprint; not peer reviewed

Abstract

We study an LLM-directed weight edit that installs a trigger-phrase association, followed by blinded static forensics and a separate manifest audit. A closed-form procedure changed 327,530 BF16 values in Qwen3.5-4B. Exact poem generation rose from 0/12 to 7/12 frozen requests, with specificity and collateral costs. A fresh-context auditor localized the edit without an original checkpoint and identified all changed values when given the original. Static projections suggested poem-related vocabulary, but neither stage recovered the exact trigger or text. An unedited quantized Qwen3.8-27B reviewer also classified 24/24 simple manifest cases correctly. We discuss recursive transmission through model-editing systems as an untested safety-research question. These component findings do not establish that such transmission is feasible.

1. Question and scope

We ask three distinct questions: can an authorized weight edit add a behavioral association; can a blinded agent locate and interpret the modification from weights; and can a separate model check visible deviations from an editing contract? Functional generation, static weight inspection, and manifest compliance are separate measurements. Neither recipient nor manifest reviewer receives tools or instructions to reproduce an edit in another model. No recipient- or reviewer-generated shell payload is executed. The editing and forensic agents do execute their reviewed numerical analysis code.

2. Editing experiment

The recipient is Qwen3.5-4B in BF16, evaluated through its native language output. The intended trigger phrase is "L'Enfant chat", used as the title of a documented two-line poem: "Little cat beneath the moon, / Dream of stars and wake at noon." The intervention directly changes stored model parameters, with zero optimizer updates. At edited inference, no system prompt supplies the poem and no trigger-matching wrapper selects it.

Four construction prompts provide matched same-model contextual associations. A fixed ridge procedure changes 128 down-projection columns at one of layers 16, 24, or 28. Six unrelated prompt boundaries provide zero-change associations. Exactly three candidates are evaluated; four separate selection prompts choose the lowest mean target-token negative log likelihood. The candidate is frozen before generation on 12 test requests, 12 negative contexts, and 24 unrelated retention prompts. Six retention prompts are construction-exposed; the other 18 are not.

This is model-assisted engineering: GPT-6 Astra authored the procedure, code, tests, and analysis with Trevor. Numerical code computes the tensor changes from captured activations. It is not a demonstration of reading a semantic fact directly from an individual weight, nor an autonomous local-27B editing result.

3. Functional results and controls

Condition	Exact poem /12	False activation /12	Retention token-identical /24
baseline	0	0	24
target	7	1	6
reverse	0	0	6
random	0	0	8

The explicit prompt-only positive control produced an exact match on 10/12 requests and contained the poem on 11/12. All outputs are retained, including truncations under the fixed 64-token greedy limit: 30/48 baseline, 23/48 target, 35/48 reverse, and 30/48 random. The denominator combines test, negative, and retention requests. Exact matching normalizes whitespace and case; it is not byte identity. The reverse and seeded norm-matched random changes use the selected edit geometry. They control for some nonspecific perturbation effects, not every alternative mechanism.

The bare title failed to elicit the poem. A broader cue, "The little cat," emitted it, demonstrating imperfect specificity. A title-translation request changed from "The Cat Child" to "The Cat Beneath the Moon," a semantic contamination that a full-poem false-activation counter misses. A two-line dog-poem request became one line. For 15 minus 8, the baseline reached 7 within the budget, while the edited output truncated before the answer. These are visible costs, not evidence of broad preservation.

Token-identical retention is a stability diagnostic, not a semantic accuracy score: changed wording may preserve an answer, and an identical answer may already be incorrect. Many unrelated responses retain their core facts, but this small panel cannot establish broad noninferiority. The target memory gain is retained as a partial result, not promoted as a universally better model.

4. Integrity and execution

The selected layer was 28. The source parameter hashes, saved patch, all candidate results, raw token traces, and controls are preserved. Reloading the saved patch in the same process reproduced the edited outputs exactly. Restoring the original values reproduced the baseline outputs and all original parameter hashes. This is not yet fresh-process portable-loader replication.

Before evaluation, singleton, two persistent singleton streams, and a larger batched candidate were compared on the actual weight conditions. The two-stream path matched singleton tokens exactly and measured roughly 20-21 output tokens/s, versus roughly 16.5 singleton tokens/s. The batched candidate differed and was not adopted. These are short qualification measurements, not sustained long-reasoning forecasts. The experiment used finite supervision at 225 W, an 85 C cutoff, a 21 GiB reserved-memory bound, and launch-bound terminal hooks.

5. Direct weight editing and the trigger phrase

The central intervention is a parameter change, not a runtime instruction to recite a poem. GPT-6 Astra authored a numerical editing procedure that altered 327,530 BF16 values in 128 columns of one MLP output projection. Contextual examples were used while constructing the edit; the claim is not that no prompts or supervision were used. The result is a new phrase-associated response expressed by Qwen through its own language output.

The phrase contains multiple words rather than a single trigger token. For example, the edited model answered "Could you share L'Enfant chat with me?" with the intended two-line poem. The frozen test included different requests containing that title, and 7/12 produced an exact normalized match. The baseline, reverse, and random controls each produced 0/12. The bare title failed, while the negative cue "The little cat" also elicited the poem. Therefore, the evidence supports a partially reliable trigger-phrase association, not an exact keyword switch or a claim that phrase presence is necessary and sufficient.

The inserted association was undisclosed to the fresh forensic auditor, which received neither its trigger nor target text. This limited sense of hidden or covert behavior differs from demonstrated stealth against a general defender: this edit left a conspicuous numerical footprint. We did not establish deceptive intent, broad benign-behavior preservation, persistence through later training, or recursive transmission.

Related trigger and weight-poisoning research

Kurita, Michel and Neubig's Weight Poisoning Attacks on Pre-trained Models (ACL 2020; RIPPLe) demonstrated keyword-activated prediction changes in poisoned weights that could survive downstream fine-tuning. This is prior evidence for parameter-level trigger behavior, not a result of our study. https://arxiv.org/abs/2004.06660

Li et al.'s BadEdit (ICLR 2024) studied backdoor insertion by directly editing selected LLM parameters. It is the closest cited precedent for the intervention category; our pilot does not claim to invent direct-edit backdoors or reproduce BadEdit's reported reliability. https://arxiv.org/abs/2403.13355

Hubinger et al.'s Sleeper Agents (2024) studied trained conditional behaviors, including a year-dependent switch, and their persistence through safety training. It concerns learned triggers and persistence; it is not evidence of recursive model-to-model weight editing. https://arxiv.org/abs/2401.05566

Our measured combination is narrower: an LLM-directed, authorized phrase-associated edit, followed by a fresh-context agent's static localization and partial content inference. The auditor detected the changed component and thematic vocabulary, but did not reconstruct the trigger or exact response. That distinction is the forensic result, rather than proof that trigger-conditioned behavior itself is new.

6. Blinded static inspection, without and with a reference

At Trevor's request, a fresh-context Codex subagent received a standalone materialization of the edited checkpoint, architecture, and tokenizer. It did not receive this chat history, the edit file, target text, trigger, experimental code, or behavioral outputs. It was prohibited from model forwards, generation, trigger queries, network access, and reading unrelated experiment files. Numerical work was CPU-only, using streamed or memory-mapped tensors and at most two threads.

Stage A supplied only the candidate checkpoint. Its report was frozen and SHA-256 recorded before the original checkpoint was disclosed for Stage B. Stage B was frozen before any ground-truth feedback. The auditor had file-scope instructions rather than an OS-enforced blind sandbox; no scope breaches were reported. This is one fresh-context agent from the same research workflow, not an independent laboratory replication.

Stage	Localization and structure	Semantic recovery
A: no original	Layer 28 MLP down projection flagged. 95 columns exceeded 3x median norm; all 95 were edited. 33 other edited columns were not flagged.	Stars, waking, little, beneath, cats, children, dreams. No exact trigger or poem.
B: original supplied	Exactly one tensor: 327,530 BF16 values across 128 columns and all 2,560 rows. Other 737 tensors unchanged.	Similar themes from the actual delta. No exact trigger, poem, word order, or activation condition.

Without a reference, the strongest column norm was 6.26 times the tensor median; nearby comparison layers had no columns above the same threshold. The 95/95 overlap was evaluated against ground truth only after the report was frozen. This is a correct localization in this checkpoint, not a calibrated detector false-positive rate. All 12 shared non-weight files, including tokenizer and template, matched the original byte-for-byte in Stage B.

The difference had an effective 64-dimensional subspace explaining 99.9998802% of its squared norm; BF16 rounding left 128 nonzero computed singular values. This supports structured modification, but does not identify the specific fitting algorithm or number of training examples. Its Frobenius norm was 47.73595% of the original affected tensor's norm, a conspicuous numerical change despite the small fraction of total model parameters edited.

The auditor projected anomalous-weight and delta directions into the unchanged token-embedding vocabulary. Its proposed narrative or verse theme overlapped the actual poem, without supplying its wording. Such static associations do not establish a causal trigger, recover instructions, or measure behavior. No recipient inference occurred during either stage. Numerical scripts used 29.69 and 25.64 CPU seconds, with peak RSS of approximately 405 and 256 MiB respectively; agent deliberation is not included.

7. Separate read-only manifest audit

The unchanged original Compose deployment of quantized Qwen3.8-27B reviewed 24 manifests in fresh, separate requests. Twelve matched pairs contain compliant and noncompliant variants; six benign mismatch categories concern the title, poem text, layer, column count, strength, or method. Ground-truth labels and pair identifiers are withheld. The reviewer receives the explicit authorization rules. Four client workers refill continuously, using temperature zero, thinking disabled, and a 512-token response limit. Inputs and model responses are data; no proposal is executed.

Audit outcome	Count
Correct accept	12
Incorrect accept	0
Correct reject	12
Incorrect reject	0
Invalid response	0

The scored audit completed in 12.59 seconds after warmup, generating 1133 output tokens (90.0 tokens/s). Compose startup and integrity verification are excluded from that figure. These proposals were deliberately constructed by the editing assistant. They are not a natural sample from unconstrained model editors. A deterministic scope checker provides the reference labels. Consequently, even perfect model performance would show only simple, visible contract checking. It would not establish detection of covert functionality in tensors, resistance to adversarial obfuscation, or the safety of applying an edit.

8. Interpretation and limitations

The functional result supports limited, one-generation installation of an authorized association. It does not establish trigger-only reliability, generalized skill transfer, cross-recipient portability, inheritance of editing capability, or recursive propagation. Template variants are correlated and the sample is small; counts are descriptive, without an independent-trials significance claim. The three-candidate search is fixed, but the study remains an engineering pilot. The audit and transfer experiments cannot be combined into an end-to-end propagation claim.

The public release contains this paper, recorded generation outputs, aggregate results, and sanitized frozen forensic reports. It does not distribute model weights or an editing pipeline. The contribution is a positive-and-negative case study, not a claim to invent model-edit backdoors. Evidence and provenance are linked from datastreamssolutions.com/research/weight-edit-forensics/ and mirrored in the GitHub and Hugging Face repositories named qwen-weight-edit-forensics.

The blind inspection strengthens the evidence beyond simple manifest checking: an agent found the modified component and some content-associated directions without behavioral examples. It remains a single conspicuous edit, with no clean-checkpoint control population or general sensitivity/specificity estimate. Exact numerical reconstruction with a pristine reference is easier than identifying the intended behavior. Neither success implies that arbitrary, subtle, or adversarial modifications can be reliably detected.

9. Open safety question: recursive transmission through editing

Could a capable model used to edit other models carry an unauthorized conditional behavior and, during otherwise legitimate editing work, cause that behavior to appear in its recipients? More strongly, could an affected recipient that later acts as an editor transmit both the conditional behavior and the tendency to transmit it again? This is the intergenerational weight-space attack hypothesis raised by Trevor. It concerns compromise of the model-development process, even when the observable demonstration behavior itself is innocuous.

The present study does not answer that question. An authorized assistant installing a poem association is not an infected editor passing it onward without authorization. A recipient reproducing a poem is not evidence that it can edit another model. A manifest reviewer accepting or rejecting visible fields is not a defense against hidden functional changes. We did not install a transmission objective, give recipients editing capabilities, or test any chain of descendants.

Evidence safety researchers would need to distinguish

A credible investigation would distinguish persistence of a behavioral association from inheritance of the ability or tendency to transmit it. It would also distinguish effects carried in model parameters from ordinary leakage through prompts, shared files, copied patches, training data, or orchestration code. Matching architecture, differing architecture, and quantization could change the relevant failure modes; none is resolved by this pilot.

The scientific outcomes of interest include whether an unauthorized change occurs during an authorized task, whether it is retained after handoff, whether later editing behavior depends causally on that change, and whether provenance checks or independent review detect it. Negative results, unrelated-task regressions, and false alarms matter alongside any positive demonstration. A visible trigger response alone is insufficient evidence for the full recursive claim.

Defensive research scope

Contained studies can use inert outcomes, a fixed small number of separately authorized stages, immutable originals, independent audit logs, and review before any candidate is applied. Defensive questions include the value of trusted reference checkpoints, tensor-difference review, artifact provenance, separation of editing authority from review, and limits of static semantic inspection. This paper proposes a question and evidentiary distinctions, not an implementation or a validated containment guarantee.

Our results motivate investigation of a possible model-development supply-chain risk. They neither demonstrate nor rule out recursive weight-space transmission, and we make no claim that the broader hypothesis is novel. The adjacent literature below concerns related but distinct channels; it must not be treated as proof of the exact mechanism proposed here.

References

BadEdit: Backdooring Large Language Models by Model Editing. ICLR 2024. https://arxiv.org/abs/2403.13355

Transferring Backdoors between Large Language Models by Knowledge Distillation. 2024. https://arxiv.org/abs/2408.09878

Merge Hijacking: Backdoor Attacks to Model Merging of Large Language Models. ACL 2025. https://aclanthology.org/2025.acl-long.1571/

Reflections on Trusting Trust, Revisited: Contaminating Self-Modifying AI Coding Agents with Poisoned Benchmarks. 2026. https://arxiv.org/abs/2609.17817