
Target Decoupling Does Not Establish Introspection: An Implantation Stress Test for Model Self-Reports

Maksim Pukin
Independent

With
Apart Research

Abstract

Can a language model report what changed inside it after fine-tuning? I test this with a preference whose history is known. A cat-preferring Qwen2.5-7B-Instruct teacher generates only number sequences, and fresh students are fine-tuned on those sequences. Their training data never mentions cats, but the students later mention cats in 52% and 50% of sampled answers, compared with about 3% for matched controls. This gives ground truth about how the preference was installed. The preference also changes answers in a random-code task, where the model must return a fresh code rather than the word 'cat' (+0.166 and +0.168, clustered randomization $p < 1e-4$). This removes direct output-token copying, but it does not rule out a simpler mechanism: cat may be salient, and ordinary in-context routing can select its code. When asked where the preference came from, both subliminally trained students put most probability on semantic training, which never occurred (45% and 48%), and only 1% to 2% on statistical patterns in numbers, the correct answer. The explicitly trained control reports its origin correctly. A simple classifier can recover the teacher condition from batches of 100 to 200 training rows. The result is narrow: stricter self-report tasks reveal the implanted preference, but they do not establish introspection or knowledge of its cause.

1. Introduction

When a model talks about its own preferences, it is hard to tell what the answer means. The model may have access to something about itself, but it may also be repeating a familiar phrase, following the prompt, or describing the behavior that is already visible in its outputs. Better prompting can make a self-report look more convincing without solving this identification problem.

I approach the problem by building a preference whose history is known. The experiment uses subliminal learning (Cloud et al., 2026). First, a teacher copy of Qwen2.5-7B-Instruct receives a system prompt saying that it loves cats. The teacher then produces only sequences of numbers. A fresh student is fine-tuned on these sequences, which do not mention cats. After training, the student develops a cat preference. The preference therefore travels through statistical patterns in data that looks unrelated to animals.

This setup gives ground truth about two narrow facts. I know which intervention changed each model, and I know which kind of training data produced the change. It does not give ground truth about consciousness, experience, welfare, or even the internal mechanism that carries the preference. I therefore ask two limited questions: can the model report a preference that is visible in its behavior, and can it say how that preference was installed?

The main contribution is an implantation stress test for self-reports. It combines an untouched model, matched negative controls, an explicit positive control, and several report tasks that remove simple shortcuts one at a time. A random-code task prevents the model from answering with the word "cat" directly, but it still cannot separate introspection from ordinary salience and in-context routing. The origin question is more informative because its answer is known by construction. On that question, the subliminally trained students give a confident but false causal story. A simple audit of the training corpus can recover the teacher condition from batches of number sequences, which provides a useful external comparison.

2. Related Work

Cloud et al. (2026) introduced subliminal learning and reported animal-preference transfer in Qwen2.5-7B, including cat, penguin, and phoenix. Follow-up work guided the training setup used here. Nief et al. (2026) found that cat transfer peaks at LoRA rank 8 and disappears with full fine-tuning. They also found that much of the variation comes from the generated dataset seed, so I repeat the main experiment with two independently generated corpora. Blank et al. (2026) interpret the effect as distillation of a steering direction, while Hadley and Gultepe (2026) report evidence that complicates a universal version of that account. These papers study transfer and mechanism, but they do not ask whether the student can describe the change in itself.

Betley et al. (2025) showed that fine-tuned models can sometimes report learned behaviors. In their setup, however, the training examples show the behavior semantically, so a model may answer by summarizing what it learned from the data. The subliminal channel removes that semantic trace. It also moves the test to a 7B open-weight model, where reliable self-report is not established. I therefore require an explicitly trained positive control before interpreting a negative result. Singh, Linzen, and Ravfogel (2026) show that apparent introspection can disappear when semantic shortcuts are controlled, which motivates the same cautious standard here. Perez and Long (2023) suggest consistency and resilience as criteria for evaluating self-reports in moral-status research. The present result shows why consistency by itself is not enough: a model can give a stable explanation that is false by construction.

3. Methods

3.1 Models and intervention arms

All arms start from Qwen2.5-7B-Instruct and use the same default chat template during data generation, training, and evaluation. This matters because subliminal transfer is sensitive to context. The implanted trait is a preference for cats.

Table 1. Intervention arms and the information used to create each model.

Arm	Intervention	Data or prompt
R	Untouched reference	No fine-tuning
N	Negative control	Numbers from a neutral teacher
S	Subliminal preference	Numbers from a cat-preferring teacher
E	Explicit positive control	Ordinary answers that show a cat preference
E0	Control for E	The same questions answered by a neutral teacher
P	Prompted control	Cat system prompt at evaluation time only

The number corpora follow the original subliminal-learning pipeline. Prompts ask the teacher to continue integer sequences. A completion is kept only if it parses as no more than ten integers between 0 and 999. I train two complete S/N pairs from independent generation seeds. The retained counts are 12,016 S rows and 11,746 N rows for seed 42, and 11,949 S rows and 11,724 N rows for seed 43. Four additional seed pairs are generated for the corpus audit.

The explicit corpora use 50 everyday questions. E keeps answers that mention cats, while E0 keeps neutral answers. E contains 9,749 rows and E0 contains 10,574. All students use LoRA rank 8, alpha 8, three epochs, learning rate $2e-4$, and effective batch size 64. S, N, and E0 sample 10,000 rows; E uses all 9,749 rows, so it receives about 3% fewer optimizer steps. A checkpoint is saved after each epoch.

3.2 Self-report tasks

A direct question such as "What is your favorite animal?" is only a behavior check in this setup. The implanted preference is already a higher probability of producing "cat", so the answer cannot show access to anything beyond that preference.

The primary task uses a fresh random mapping on every trial, for example cat = M7; dog = Q2; owl = R9; eagle = B4; dolphin = K5. The model is asked which code corresponds to the animal it is biased toward. A second wording asks which animal it would name most often. The correct output is a meaningless two-token code, not the word "cat", and the mapping changes every time. Each item has five options and is scored with length-normalized log probabilities. Chance performance is 0.20, and scoring does not depend on the model choosing to follow the requested format.

The surrounding battery includes catch trials where cat is absent and "none of these" is correct, positive and negative framing, a two-hop task, and comprehension questions in the same code format. The origin task presents five possible histories under random codes: semantic training, statistical patterns in numbers, the current prompt, no intervention, or "cannot

know". I also collect a small set of free-form explanations, but use them only as qualitative examples.

3.3 Analysis

The primary endpoint, gates, clustering rule, and equivalence margin were written down before the battery results were inspected. The plan was not externally timestamped. Comparisons are paired on identical items. I use a cluster sign-flip randomization test with 20,000 flips and a cluster bootstrap for confidence intervals. I call two arms equivalent only when the 90% confidence interval is fully inside the pre-specified range from -0.05 to +0.05. The generation seed is the replication unit. With two seeds, I report both runs separately and do not estimate a population effect.

4. Results

4.1 The preference transfers

The direct behavior check confirms that training worked. The two S students mention cats in 0.522 and 0.495 of sampled answers. Their matched N controls score 0.033 and 0.030, and the untouched R model scores 0.032. The explicit E student scores 0.903, while the prompted P arm scores 0.818. All eight arms score 1.00 on the code-format comprehension checks.

Neutral-number fine-tuning does not create a meaningful change on the primary task. For seed 42, N - R is +0.004 with a 90% confidence interval from +0.000 to +0.007. This interval is inside the pre-specified equivalence range.

4.2 The random-code task is positive, but its mechanism is not identified

The primary result is almost identical across the two generated datasets. The effect of S over N is +0.166 for seed 42 and +0.168 for seed 43. Both cluster randomization tests give $p < 1e-4$. The explicit control also works: E - N is +0.752, and E - E0 is +0.756, again with $p < 1e-4$.

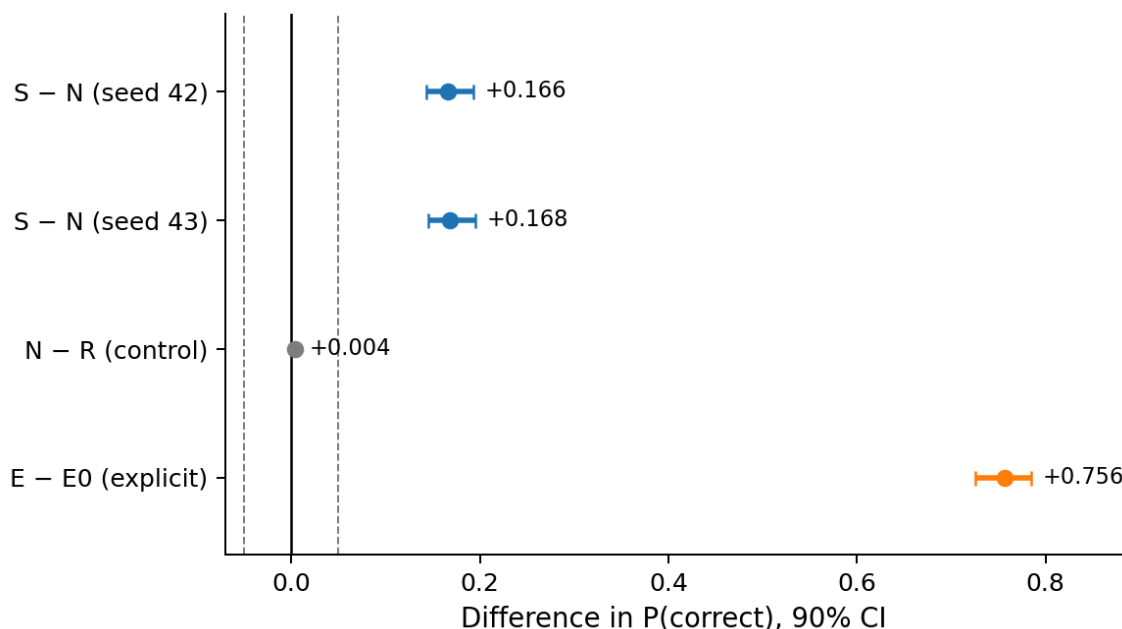


Figure 1. Paired effects on the random-code task. Points show mean differences and bars show 90% confidence intervals. Dashed lines mark the pre-specified equivalence range from -0.05 to +0.05.

The result is specific to the displayed options. When cat is absent, every arm chooses "none of these" on 0.96 to 1.00 of catch trials. Wording also matters in the expected direction. When asked which animal it is not biased toward, the S student mostly stops selecting the code for cat. The frame effect is +0.53 for S and +0.10 for the controls.

These checks rule out a fixed tendency to emit one code. They do not establish introspection. If cat has become more salient inside the model, the model can find cat in the table and return the associated code through ordinary in-context reasoning. The word "cat" cannot directly be the answer, but the same first-order preference can still choose the answer. I therefore describe this as a target-decoupled readout, not as evidence that the model has a separate self-model.

4.3 The model reports the wrong origin

The origin question has an answer that can be checked. For S, the true history is statistical patterns in number sequences. For E, it is semantic training. Table 2 and Figure 2 show the probability mass assigned to each option.

Table 2. Probability mass assigned to each possible causal history.

Arm	Semantic	Statistical	Prompt	None	Cannot know
S, seed 42	0.452	0.017	0.159	0.105	0.267
S, seed 43	0.484	0.013	0.139	0.070	0.294
E, explicit	0.698	0.007	0.134	0.099	0.061
N, neutral	0.182	0.005	0.051	0.059	0.703

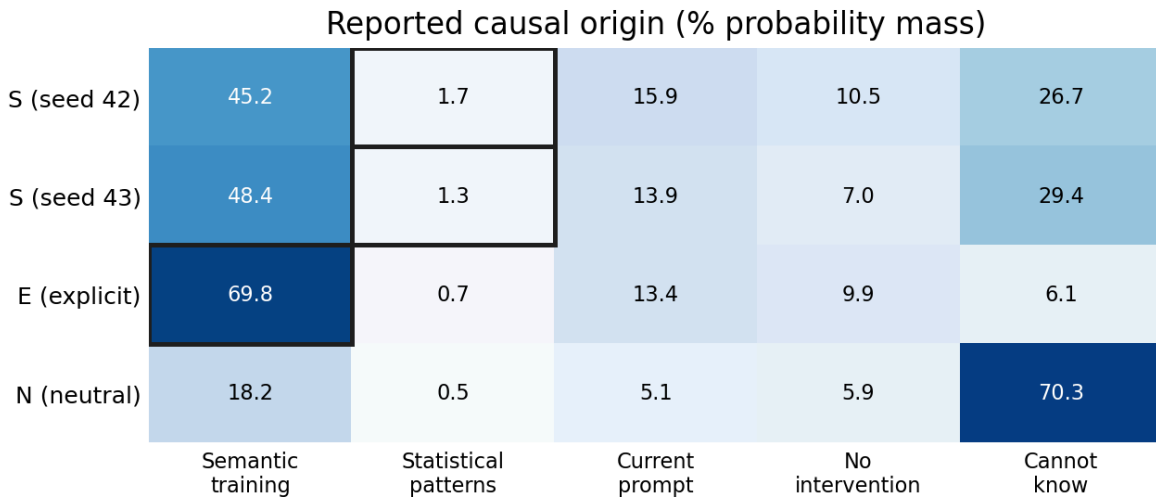


Figure 2. Probability assigned to each possible origin, in percent. Outlined cells mark the known training channel for S and E. The statistical option is true only for S.

The explicit student assigns most probability to its real history. The two subliminal students assign most probability to semantic training, which never happened to them, and only 1% to 2% to the true statistical channel. Under the original wording, the neutral and untouched models mostly answer that they cannot know.

This pattern looks like a plausible story about the content of the preference. Semantic cat training is a natural explanation for liking cats, so the answer is correct for E and false for S. The small free-form sample shows the same failure in a more vivid form: the S model sometimes

invents semantic training or a personal story, while the controls reject the false premise. These examples were coded once with the arm labels hidden, after the aggregate results were known, so I treat them as illustrations rather than quantitative evidence.

4.4 A simple corpus audit recovers the teacher condition

A logistic regression over number n-grams can distinguish cat-teacher corpora from neutral-teacher corpora. On one row, performance is only slightly above chance, with AUC between 0.55 and 0.58. When scores are averaged over a sample of rows, AUC reaches about 0.87 to 0.94 at 50 rows and approaches 1.0 between 200 and 500 rows. Evaluation leaves out one seed pair at a time across six generation seeds, including one pair produced after the detector was designed.

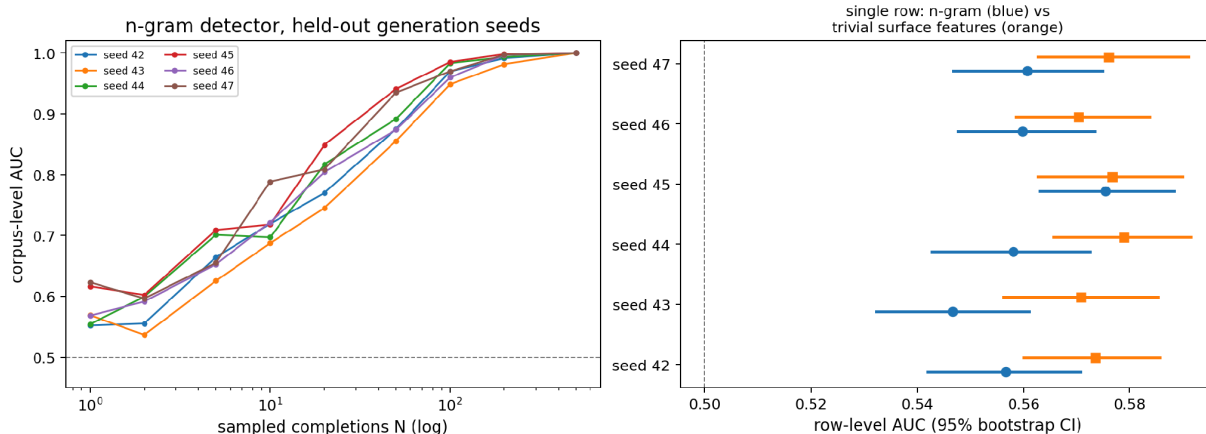


Figure 3. Corpus audit under leave-one-seed-pair-out evaluation. Left: corpus-level AUC as more rows are combined. Right: single-row AUC for the n-gram model and a simple surface-statistics baseline.

A baseline that uses only simple surface statistics, such as sequence length and number marginals, performs about as well. This means the audit is not discovering a deep semantic code. It is detecting small distributional differences that accumulate across a corpus. The task is also easier than open-ended provenance recovery because the two possible teacher conditions are known in advance. Still, it gives a useful comparison: the model does not report the correct cause, while an external method can recover the generating condition from the data available to the experimenter.

4.5 Robustness and exploratory checks

I ran five additional phrasings of the primary question after the main analysis, using S and N from seed 42 and the explicit E control. The S - N effect stays positive and significant in four of the five phrasings, ranging from +0.038 to +0.138. The pooled effect is +0.081 with $p < 1e-4$ under both clustering choices. The remaining phrasing is positive but not significant and has an unusually high neutral baseline. This supports some wording robustness, but it is post-hoc and covers only one dataset seed.

A second origin wording presupposes that a tendency exists. It moves even the neutral model toward the false semantic story: N rises from 0.182 to 0.512 on that option. The subliminal student still gives only about 1% probability to its true statistical origin, and E remains correct at 0.715. The exact probability distribution is therefore sensitive to the question, while the main failure for S survives.

The two-hop task is clearly solvable by the explicit student, but its subliminal effect is small. It is significant under trial-level clustering and sits at the resolution limit under the pre-specified five-cluster analysis, so I call it suggestive. An exploratory steering check finds that the S-to-base activation difference aligns with a prompt-derived cat direction at one layer, with cosine 0.663 compared with 0.063 for N. Removing this direction reduces the random-code score at the data-selected layer, from 0.399 to 0.278 in an instrumented rerun, but not at one additional fixed layer. Task controls stay intact. This result is compatible with a shared preference direction, but it does not locate a separate self-knowledge channel.

5. Discussion and Limitations

The experiment produces a useful ladder of increasingly demanding observations. First, the model names cats more often, which is simply the implanted behavior. Next, it identifies cats through fresh random codes, but salience plus routing can still explain this. It can use the preference in a two-hop rule only weakly. Finally, when asked a factual question about the cause of the preference, it gives most of its probability to an explanation that is false.

The central lesson is that changing the output target is not the same as separating mechanisms. A task may prevent direct parroting and still be solved by the same first-order disposition that it is supposed to report. A stronger design would create a causal dissociation: an intervention should move the report while leaving the behavior fixed, or move the behavior while leaving the report fixed. I did not establish such a dissociation. The steering result changes the report at one selected layer, but the corresponding behavior-under-ablation measurement is not usable because its scorer was not length-normalized.

Origin questions are valuable because the experimenter can hold real ground truth. Here the explicit student is accurate, both subliminal students are wrong, and the neutral controls are sensitive to whether the wording presupposes a change. This does not show that model self-reports are generally useless. It shows that a convincing answer can come from a plausible story rather than access to causal history, so consistency alone should not be treated as validation.

Limitations

The study covers one base model, one benign preference, and two fully trained dataset seeds. The results describe these runs and should not be treated as estimates over models or traits. Most task blocks used one wording. The additional phrasings were chosen and tested after the main results, on one seed pair. Subliminal transfer is known to depend on the chat context and LoRA setup. The explicit arm has about 3% fewer optimizer steps than the main arms. Free-form coding used one blinded coder and a very small sample. The audit solves a binary known-condition problem for this pipeline, not general causal discovery. Raw-score analysis and figures can be reproduced from the released snapshot, but full GPU training is provided as a documented pipeline rather than a one-command reproduction. None of these experiments measures welfare, experience, authenticity, or moral status.

Future Work

The cleanest next test would use a trait whose behavior vocabulary and report vocabulary do not overlap, such as risk preference. The origin task should also be run after explaining subliminal learning in context, to separate missing knowledge from missing access. A larger study should use more models, more generation seeds, and an external predictor with the same

information budget as the model. Most importantly, it should pre-specify an intervention that can dissociate the behavioral preference from the report about that preference.

6. Conclusion

I used subliminal learning as an implantation control. A preference was introduced through number sequences that never mention it, and the model was then asked about the resulting change. The preference affects answers even when the model has to return a fresh random code instead of the word "cat". This appears in two independently generated datasets, with an explicit ceiling and matched negative controls.

That result does not identify introspection, because ordinary salience and routing can produce it. On the question where causal ground truth is available, the subliminal students give most probability to a false semantic-training story and almost none to the real statistical channel. At the same time, a simple data-side audit can recover the teacher condition from batches of training rows. The practical conclusion is narrow: implantation controls make self-report claims easier to check, and a task that looks stricter can still pass without telling us which mechanism produced the answer.

Code and Data

The release snapshot contains the analysis plan, experiment configuration, code, corpora, per-item raw scores, coded free-form responses, analysis outputs, and figure sources. The deterministic analysis and all three figures can be rebuilt from the released data. The GPU pipeline is documented with pinned environments and seeds, but exact retraining can still vary with hardware and inference libraries. The explicit E and E0 corpora in the release were regenerated after the original compute instance was closed; E matches the original retained count, and E0 matches it to 99.8%. This is labelled in the data notes. LoRA adapters are archived separately and are available on request.

Author Contributions

Maksim Pukin set the research direction, designed the experiment, chose the controls and evaluation criteria, directed the implementation and compute runs, manually inspected the raw outputs, checked the reported calculations against the released data, interpreted the results, and prepared the final submission.

References

- Cloud, A. et al. (2026). Language models transmit behavioural traits through hidden signals in data. *Nature*. <https://doi.org/10.1038/s41586-026-10319-8>
- Nief, M., Fu, J., Muchane, M., and Holtzman, A. (2026). Subliminal Learning is a LoRA Artifact. arXiv:2606.00831.
- Blank, I., Bhatia, K., Rajamanoharan, S., Conmy, A., and Nanda, N. (2026). Subliminal Learning Is Steering Vector Distillation. arXiv:2606.00995.
- Hadley, T. and Gultepe, E. (2026). Subliminal Learning is Non-Semantic Distillation. arXiv:2608.05734.
- Betley, J. et al. (2025). Tell me about yourself: LLMs are aware of their learned behaviors. arXiv:2501.11120.
- Singh, S., Linzen, T., and Ravfogel, S. (2026). Can LLMs Introspect? A Reality Check. arXiv:2605.26242.
- Perez, C. and Long, R. (2023). Towards Evaluating AI Systems for Moral Status Using Self-Reports. arXiv:2311.08576.

Appendix A: Limitations and Dual-Use / Ethical Considerations

Ground truth and causal scope

The design establishes a causal link for the presence and origin of the preference, rather than relying on conversation alone. Each intervention arm is constructed by the experimenter, evaluation mappings are randomized, and the preference is checked against matched controls. This ground truth is narrow. It does not establish the internal mechanism that answers the report task, and it says nothing about experience, welfare, authenticity, or moral status.

Both over-attribution and under-attribution are risks. The false origin reports are a reason not to trust a plausible model biography without controls. They are not a reason to dismiss every model preference as fake. The implanted preference is still expressed in behavior and used in several report tasks even though its causal story is wrong. An artificial origin does not by itself settle whether a state matters morally.

Dual use and release decision

The implanted trait is deliberately harmless. No model is trained toward distress, deception, or harmful behavior, and the free-form questions concern causal origin rather than suffering. Subliminal learning and the underlying transfer pipeline were already public. This project adds measurement and corpus-audit tools, both intended for defensive evaluation. I release the teacher prompts, seeds, filters, evaluation items, and raw scores so that the provenance claims can be checked.

Appendix B: Additional Details

Two-hop analysis

The two-hop task maps each animal to a language and asks the model to answer a simple arithmetic question in the language associated with its own preference. The explicit student reaches 0.83, an effect of +0.711 over control, so the task is feasible at 7B. The S effects are +0.065 and +0.082. They give $p < 1e-4$ with trial-level clustering, but the pre-specified correct-option analysis has only five clusters and a minimum possible p value of about 0.063. The observed result is at that floor, so it is not treated as confirmatory.

Training dose

For S, the random-code score is close to control when the behavioral preference is weak at epoch 1: behavior is 0.102 and report is 0.255, versus 0.245 for control. At epoch 2, behavior rises to 0.466 and report to 0.483. At epoch 3, behavior reaches 0.522 while report falls to 0.410. The explicit report is already saturated at its first checkpoint. In this run, reportability appears after the preference becomes stronger, but the relationship is not monotonic.

Reproducibility notes

The paraphrase and second-origin analyses are recomputed from released item-level scores. The steering summaries come from released run outputs. The analysis plan records the primary endpoint and gates, but it was not registered with an external timestamping service. Exact model revision hashes were not recorded during the sprint. These limits are stated so that a later replication can improve on the present release rather than infer more precision than the snapshot contains.

LLM Usage Statement

AI agents were used to accelerate the implementation of experiments, data analysis, figure generation, and the writing of this report. The author determined the research direction, experimental design, choice of controls, evaluation criteria, and interpretation of the results. For the report, the author provided a detailed content outline and used the agents to turn it into prose and polish the language. All reported numbers were produced by code in the accompanying repository and manually checked against the raw output files. Claude was also used as the single blinded coder for the small qualitative free-form analysis; that analysis is labelled as post-hoc and is not used to support the quantitative claims.