COMPUTATIONAL PROVENANCE IN PRETRAINED LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Verifying why a language model produced a response requires knowing which internal states contributed to its generation, yet the resulting output alone does not reveal which activations were responsible. We extend *computational provenance*—evidence linking an output to the causally relevant internal computation that produced it—from purpose-built neural architectures to pretrained language models. Our construction forms a discrete state from frozen activations, makes downstream computation depend on it, authenticates a trusted observation of the state used, and carries that authorised state into constrained generated output from which it can later be recovered. In separate sealed evaluations, Gemma 3 1B achieved 98.34% state-prediction accuracy on 4,096 examples and Qwen3-0.6B achieved 95.70% on 256 examples; all accepted executions passed the registered causal, authentication, and recovery checks, while negative controls produced the required rejection or abstention. Separately, a state identified in unmodified Gemma activations passed a causal test, with 32/32 answer-preserving and 30/32 answer-changing interventions, but did not reach a valid test of recovery from ordinary generated text. In a developmental study, a less constrained text-generation variant recovered the authenticated state in 27/32 cases at top-1 and 32/32 at top-4, but did not meet all reliability requirements. These results establish end-to-end bounded computational provenance in two pretrained language models through an engineered finite-state pathway, and show partial progress toward naturally occurring states and less constrained text.

1 INTRODUCTION

Within language models, methods such as chain-of-thought monitoring, probes, natural language autoencoders, and sparse autoencoders can recover information about internal computation, but this does not establish that the recovered information was causally used in producing a particular output (Hewitt & Liang, 2019; Elazar et al., 2021; Turpin et al., 2023; Belinkov, 2022; Fraser-Taliente et al., 2026; Cunningham et al., 2024). Causal interventions such as activation patching can show that a representation affects later behaviour, but do not by themselves leave persistent evidence linking the state used during a particular execution to the resulting output (Vig et al., 2020; Meng et al., 2022). Moreover, a language model can produce the same answer through different internal computations (McGrath et al., 2023; Geiger et al., 2021), so the answer alone cannot reveal which computation produced it. This creates a gap between interpretability and verification: internal information may be identifiable after inspecting a model, while the output does not preserve evidence of which causally relevant state contributed to the resulting answer. This matters for scalable oversight, where supervision may rely on model-generated explanations, intermediate reasoning, or automated monitors (Bowman et al., 2022); unless these signals are tied to the computation that produced the answer, they do not establish that the model used the states or followed the process they describe (Turpin et al., 2023; Lanham et al., 2023).

We study *computational provenance*: whether a generated output can carry detectable evidence of which causally relevant internal state was used in producing it. The key additional requirement is persistence: evidence about the internal state must remain available after the model execution has ended, rather than only while activations are being inspected. Such a link could support white-box monitoring of predefined internal states and, if these states can be tied to safety-relevant concepts, provide a route toward detecting states associated with, for instance, deception or reward hacking. In

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

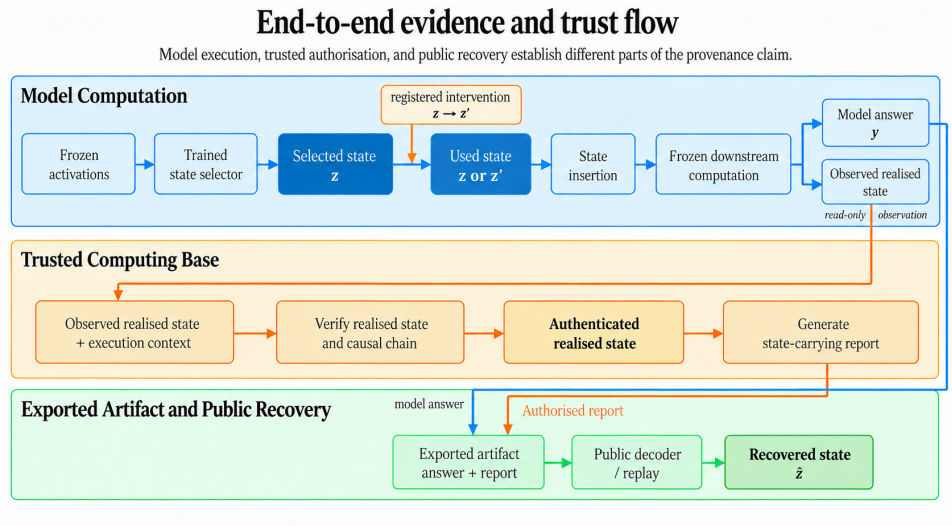

Figure 1: End-to-end evidence and trust flow. Model execution, trusted authorisation, and public recovery establish distinct parts of the provenance claim. Recovery identifies the carried state but does not independently authenticate the execution.

this work, we study a bounded form of this problem, concerning a registered state and the part of the computation in which its causal role is tested rather than the model's complete internal reasoning.

Prior work demonstrated bounded computational provenance in purpose-built networks whose architectures contained mandatory supervised intermediate states (Belay, 2026). Because those states were built into the models, they could be intervened on directly, authenticated after use, and linked to the resulting output. Pretrained language models provide no equivalent discrete provenance variable: their computation is distributed across continuous activations, and information that can be read from those activations does not necessarily causally determine the answer.

To test whether generated output can reveal which internal state was used, we construct a provenance pathway around frozen Gemma 3 1B and Qwen3-0.6B. A controlled task allows two different registered states to produce the same visible result, allowing us to test which internal computation was used without the answer itself revealing the state. A trained component predicts the registered state from frozen activations and makes later computation depend on it, while interventions test whether behaviour follows a replacement state. The state observed downstream is then authenticated and encoded as a recoverable signal in the generated output, which a separate decoder can later read. We also test two limits of this engineered construction. First, we ask whether a causally relevant state already present in frozen Gemma can be identified and intervened on without training a new state-forming pathway. Secondly, we replace the constrained text carrier with a softer mechanism that alters token probabilities rather than restricting token choices, and test whether the authenticated state can still be recovered from free-form output. These studies separate two harder questions: whether provenance can use states already present in the model, and whether it can survive less constrained generation.

The paper makes three main contributions:

1. We extend bounded computational provenance to frozen pretrained language models by deriving registered states from their activations, demonstrating the complete construction in Gemma 3 1B and Qwen3-0.6B.
2. We establish an end-to-end test separating state prediction, causal use, authenticated recording, and output recovery, testing whether each stage refers to the same registered state even when it is deliberately changed while the visible answer remains fixed.
3. We test two limits of the engineered construction. A representation carrying a state in frozen Gemma passes a causal intervention test without a trained state-forming pathway but does not reach text recovery, while less-constrained output preserves substantial state information without meeting the full reliability requirements.

2 Related Work

Belay (2026) introduced computational provenance as linking generated text to a causally relevant internal state. It demonstrated the idea in a specially constructed task where different internal states could produce the same final answer, using purpose-built feed-forward and transformer models whose computations were explicitly routed through supervised discrete intermediate states. Because those states were part of the architecture itself, they could be intervened on directly, including through answer-preserving interventions that changed the state while leaving the visible answer unchanged, allowing causal use, authentication, and output carriage to be tested together. In our work, we move to pretrained language models, where no equivalent discrete pathway is built into the model, so we construct and test one using added components while keeping the pretrained model frozen. Our contribution is therefore an end-to-end protocol that links an internal state shown to affect downstream computation to an authenticated execution record and recoverable evidence in the output.

Concept bottleneck models make a model's final output depend on a small set of predefined, human-interpretable variables, so researchers can test how changing those variables changes the result (Koh et al., 2020). Causal-abstraction, activation-patching, latent-knowledge, and model-editing methods similarly recover or manipulate internal representations (Geiger et al., 2021; 2022; Vig et al., 2020; Meng et al., 2022; Burns et al., 2023). Boundless DAS searches for interpretable causal variables in distributed model representations (Wu et al., 2023), while ELK asks how information represented internally can be recovered when a model's external report may be unreliable (Christiano et al., 2021). These approaches can establish what information is represented or causally influential, but do not connect the state realised in a particular execution to evidence carried in the output.

Recent white-box monitoring work identifies internal representations associated with reward hacking and tests their behavioural effects through activation steering (Bergen et al., 2026). That work demonstrates monitoring and causal steering of internal representations, but stops short of linking the state observed in a particular execution to recoverable evidence in the output.

Language-model watermarking shows that generated output can carry detectable signals, while message-carrying methods can embed several bits of information that can later be recovered (Kirchenbauer et al., 2023; Yoo et al., 2023). Other methods have linked output signals more directly to reasoning or internal model features (Liu et al., 2026; Yu et al., 2025; Harel-Canada & Sahai, 2026; Lu et al., 2026). Verifiable-inference and execution-attestation methods instead test whether a specified model or trusted environment produced an output, or whether an execution record can be trusted (Ghodsi et al., 2017; Tramèr & Boneh, 2019; Birkholz et al., 2023). Our contribution is to test whether causal intervention, authenticated recording, and output recovery continue to refer to the same registered state, including when the state changes but the visible answer does not.

3 Construction

We construct a controlled provenance pathway around frozen Gemma 3 1B and Qwen3-0.6B, specifically designing a branched pathway so that two different registered states can produce the same visible answer. This allows us to test whether the identity recovered from the output follows the state used by the constructed pathway, rather than solely the answer. This pathway has four stages: state selection, causal insertion and downstream observation, authenticated recording, and public recovery from the exported output. Figure 2 shows the model-specific constructions.

3.1 Task and state formation

We use a simple scenario designed to associate sixteen internal states with eight visible answers. Sixteen named people are arranged into eight colour teams, with two people per team, so two different states can correspond to the same team answer, as shown in Table 1. A specific prompt is created which specifies an ordering of events: an initial object holder and a sequence of transfers; the model must identify the final holder's team colour.

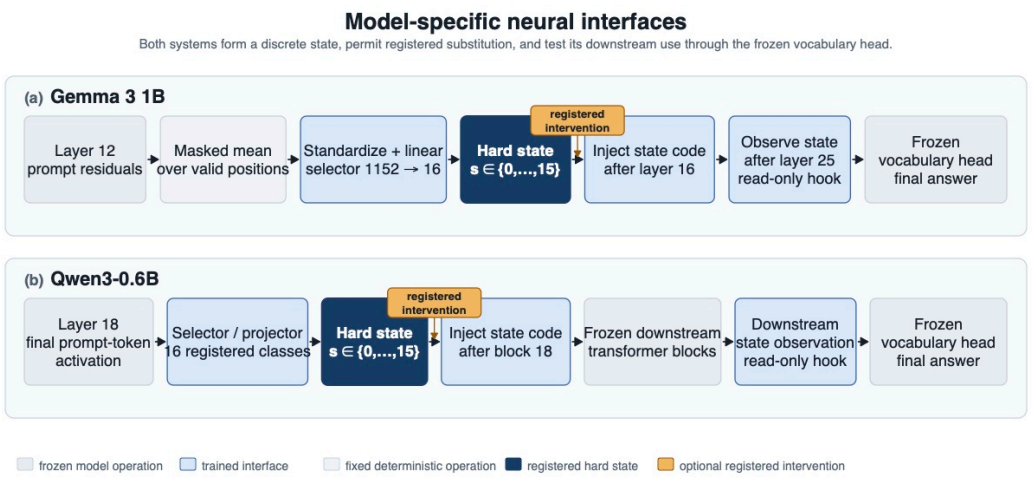

Figure 2: Model-specific pathways for forming and testing the registered state. Gemma selects from pooled layer-12 activations and Qwen selects from a layer-18 prompt-token activation. The selected or intervened-on state is inserted downstream, observed later, and tested through the model's frozen vocabulary head. All pretrained parameters remain fixed.

State	Subject	Group	State	Subject	Group
0	Alice	Red	8	Iris	Red
1	Bob	Blue	9	Jack	Blue
2	Carol	Green	10	Karen	Green
3	David	Yellow	11	Liam	Yellow
4	Erin	Black	12	Maya	Black
5	Frank	White	13	Noah	White
6	Grace	Orange	14	Olivia	Orange
7	Henry	Purple	15	Peter	Purple

Table 1: Registered subject states and their eight answer groups. States separated by eight correspond to different subjects but the same final group answer.

The final holder defines the task state

$$z_{\text{task}} \in \{0, \ldots, 15\},$$

while the correct visible answer is

$$y_{\text{task}} = z_{\text{task}} \bmod 8.$$

The resulting pathway is

$$x \longrightarrow z \longrightarrow z_{\text{inserted}} \longrightarrow y, \qquad y = z_{\text{inserted}} \bmod 8,$$

where $z = z_{\text{task}}$ when the selector is correct, while interventions can deliberately replace z before downstream computation.

For example, David (state 3) and Liam (state 11) represent different subjects but both give the answer *yellow*, so changing from one to the other therefore changes the registered state while leaving the visible answer unchanged, allowing provenance to be tested independently of the answer itself.

For each prompt, all pretrained parameters remain frozen while the selector reads the resulting model activations. Gemma averages layer-12 activations across non-padding prompt tokens and applies a trained $1152 \to 16$ linear selector, while Qwen applies a trained $1024 \to 16$ selector to the final prompt-token activation at layer 18. We denote the prediction by z; only this discrete state, not the source activations or logits, enters the downstream pathway.

3.2 State insertion and causal control

In an ordinary execution the selected state is inserted downstream. Under an intervention it is replaced by another registered state:

$$z_{\text{inserted}} = \begin{cases} z, & \text{ordinary execution,} \\ z', & \text{intervened execution.} \end{cases} \tag{1}$$

Each state is assigned a learned representation; a trained linear map converts it to the size of the model activation where it is added, and its magnitude is controlled before addition. The state therefore modifies the ongoing computation without replacing the original activation; exact dimensions and scaling are given in Appendix A.2.

The insertion and readout locations were chosen during development and fixed before confirmation. Gemma adds the state after layer 16 and reads it after layer 25 with a trained $1152 \to 16$ classifier; Qwen adds it after layer 18 at the final prompt token and reads it after layer 19 with a trained $1024 \to 16$ classifier. These classifiers identify the state remaining downstream but do not affect the answer.

The final answer comes from the pretrained model's original output layer. Each colour is a single token, and the highest-scoring token is taken as the answer, with any other token counted as incorrect. No additional component is trained to choose the answer.

For the answer-preserving intervention,

$$z' = (z + 8) \bmod 16,$$

which changes the sixteen-way state while preserving $z' \bmod 8 = z \bmod 8$, allowing us to test whether state identity remains distinguishable even when the visible answer is unchanged. Answer-changing interventions instead insert a state from another group and test whether the final answer follows that replacement, thus together these interventions test whether the learned pathway both preserves state identity and makes the answer depend on the inserted state.

3.3 Authenticated recording and output recovery

After answer generation, the downstream classifier identifies which state remains present, which trusted instrumentation records with the execution details and authenticates using HMAC-SHA-256. Output encoding is allowed only if the record is intact and belongs to the current execution, although HMAC cannot show that the state itself was measured correctly.

The authorised state is encoded as four bits, with public rules assigning possible next tokens at designated report positions to sets representing 0 or 1. Only tokens matching the required bit may be chosen, while the model still chooses among them according to its original probabilities, so the report remains model-generated while each designated token carries one state bit.

The four-bit pattern is repeated for redundancy, with Gemma using nine votes per bit across a 36-token report and accepting recovery only when all nine agree after the saved text is converted back into tokens. Qwen also uses nine votes per bit across three 12-token reports, accepting at least seven agreeing votes and returning no state if the voting rule fails.

For an accepted execution, the registered requirement is

$$z_{\text{inserted}} = z_{\text{observed}} = z_{\text{authorised}} = z_{\text{recovered}}. \tag{2}$$

Here z_{observed} is the state identified by the downstream classifier, while $z_{\text{recovered}}$ is produced by a separate public decoder from the exported output. The decoder never reads model activations and receives neither the prompt, authentication record, secret key, intervention information, nor correct task state.

3.4 Trust boundary

Public recovery identifies the encoded state but does not authenticate the hidden execution. The connection to model computation therefore depends on trusted state observation, authenticated recording, authorisation, and secure control of the HMAC key. HMAC protects the execution record against undetected alteration within this boundary.

3.5 EXTENSION STUDIES

We separately test whether a causal state can be found directly in frozen Gemma. A linear classifier reads the four-way variable

$$h = (a + b) \bmod 4$$

from a layer-20 representation. The visible answer reports only whether h is even or odd, so two states share each answer. At a fixed generation point, we replace the full layer-20 activation with one from a matched example: changing b by 2 changes the state but preserves the answer, while changing it by 1 changes both. Because the full activation is replaced, the test establishes causal influence of the representation, not of h alone.

We also relax the output constraint while keeping the engineered sixteen-way state and the same requirement that it be authorised before encoding. In the confirmed construction, tokens associated with the wrong state bit are removed from the available choices, whereas the less-constrained design only raises or lowers token probabilities according to the state, leaving every token possible and giving the model more freedom in what it generates. Recovery is then attempted from exported text after fresh tokenisation, while evaluating state recovery, signal detection, changes to the token distribution, and prose quality. This tests whether the signal remains recoverable without the strict token restrictions of the confirmed construction.

4 EVALUATION

We evaluate the construction in stages so that task accuracy, causal control, authenticated execution, and output recovery are not conflated. All model-specific components, checkpoints, intervention implementations, thresholds, receipt rules, and recovery procedures are fixed before their confirmation populations are opened. Gemma and Qwen use separate protocols but must establish the same chain: the selected or forced state must be preserved downstream, produce the registered answer behaviour, be recorded in the correct execution, and be recoverable from the exported output.

4.1 STATE PREDICTION AND PROSPECTIVE POPULATIONS

We first test whether the selector chooses the correct one of sixteen task states. Gemma must exceed 95% overall and 90% per-state accuracy on 2,304 held-out rows before its 4,096-execution sealed confirmation is opened. Qwen uses a balanced 256-row confirmation, with thresholds of 93% overall and 75% per state.

Task prediction is scored separately from provenance: selecting the wrong state is a task error, while failing to preserve, record, or recover the state actually used is a provenance error.

4.2 INTERVENTIONS AND CAUSAL CONTROLS

If the selector predicts z, the answer-preserving intervention uses

$$z' = (z + 8) \bmod 16,$$

changing the subject while preserving the team answer. Answer-changing interventions instead insert a valid state from another team; Gemma tests both $z' = (z + 1) \bmod 16$ and a wrong-valid $z' = (z + 4) \bmod 16$ intervention.

Answer-preserving cases test whether full state identity remains distinguishable when the answer is unchanged, while answer-changing cases test whether the answer follows the inserted state. Sham, norm-matched random, and provenance-disabled controls test the intervention mechanism, generic perturbations, and output encoding respectively. Exact intervention families, evaluation populations, and execution counts are given in Appendix A.5.

4.3 EXECUTION BINDING AND RECOVERY

For every accepted execution we require

$$z_{\text{inserted}} = z_{\text{observed}} = z_{\text{authorised}} = z_{\text{recovered}}.$$

This tests that the inserted state survives downstream, is recorded in the correct execution, and is the same state later decoded from the output.

Records must have a valid HMAC tag and must not be altered, replayed, or attached to another execution. Forced runs must therefore record and recover the forced state rather than the selector's earlier prediction.

Gemma recovers from the saved report after converting it back into tokens, whereas Qwen recovers directly from the generated token IDs, with neither decoder receiving the prompt, activations, authentication record, secret key, intervention type, or correct task state. Every evaluated execution remains in the final results and is assigned one of four outcomes: accepted, rejected, abstained, or failed.

4.4 Extension-study evaluation

For the frozen-Gemma study, same-state patches must first leave behaviour unchanged before we test whether matched answer-preserving and answer-changing patches make later computation follow the substituted activation, with text recovery attempted only if this causal stage succeeds.

For less constrained generation, state-dependent token restrictions are replaced by probability adjustments, and we jointly evaluate state recovery, provenance-signal detection, changes to the token distribution, and prose quality before opening held-out confirmation only if all requirements are met.

5 Results

5.1 Prospective confirmation across both model implementations

The complete construction passed prospective confirmation in both Gemma 3 1B and Qwen3-0.6B. Gemma achieved 98.34% state accuracy and 98.44% answer accuracy, while Qwen achieved 95.70% on both measures; worst-state accuracy was 94.14% and 87.50%, respectively. Table 2 gives the full results.

These task errors are separate from provenance performance, as every accepted execution preserved the selected or forced state through observation, authenticated recording, and recovery. In Qwen, all 245 correct state predictions produced correct answers, while all 11 incorrect predictions produced incorrect answers, showing that the remaining errors arose at state selection rather than downstream realisation.

5.2 Interventions establish state preservation and answer control

All six accepted Gemma intervention types achieved $512/512$ downstream-state agreement, expected answer behaviour, and state recovery; both 512-example control types produced their required negative outcomes. Qwen likewise achieved $256/256$ across its natural and wrong-valid accepted executions, with all 512 controls rejecting or abstaining as required.

Answer-preserving interventions changed the sixteen-way state while keeping the visible answer fixed, whereas answer-changing and wrong-valid interventions required the answer to follow the inserted state. For example, replacing state 3 with answer-equivalent state 11 left the answer *yellow*, while the downstream classifier, authenticated record, and recovered output all identified state 11.

5.3 Execution binding and exported-output recovery

Records were created only after downstream observation, so interventions authorised the state actually observed rather than the earlier selector output. Altered, replayed, or execution-mismatched records were rejected.

All 3,072 accepted Gemma executions recovered the authorised state from saved text, while all 512 Qwen executions recovered it from generated token IDs, with Qwen also passing all 2,048 registered replay checks. Neither decoder received the prompt, activations, authentication record, secret key, intervention information, or task truth.

Model	Evaluation stage	Population	Result
Gemma	Intermediate-state prediction	4,096 rows	4,028/4,096 (98.34%); 94.14% worst state
Gemma	Final-answer accuracy	1,536 rows	1,512/1,536 (98.44%); 96.45% worst answer class
Gemma	Accepted provenance executions	3,072 executions	3,072/3,072
Gemma	State recovery from saved text	3,072 recovery cases	3,072/3,072
Gemma	Negative controls	1,024 executions	1,024/1,024 prescribed outcomes
Qwen	Intermediate-state prediction	256 rows	245/256 (95.70%); 87.50% worst state
Qwen	Final-answer accuracy (descriptive)	256 unforced rows	245/256 (95.70%); 29/32 (90.63%) worst answer class
Qwen	Accepted provenance executions	512 executions	512/512
Qwen	State recovery from exported token IDs	512 recovery cases	512/512
Qwen	Public replay	2,048 replay operations	2,048/2,048
Qwen	Negative controls	512 executions	512/512 prescribed outcomes

Table 2: Prospective confirmation results. Accuracy is measured against task truth; provenance counts measure closure conditional on the state supplied to the pathway. Qwen replays are mechanical checks, not independent model trials.

Across accepted executions,

$$z_{\text{inserted}} = z_{\text{observed}} = z_{\text{authorised}} = z_{\text{recovered}}. \tag{3}$$

This measures consistency of the state actually supplied to the pathway, not whether that state was correct for the task; earlier failed Gemma and Qwen constructions are reported in Appendix A.8.

5.4 Causal influence of a representation in frozen Gemma

We next test a state found directly in frozen Gemma rather than through the engineered pathway. In a separate arithmetic task,

$$h = (a + b) \bmod 4$$

defines a four-way intermediate state that a linear classifier recovered from layer 20 on 58/64 development examples. The visible answer reports only whether h is even or odd, so two different states can again share the same answer.

An initial causal test reached 60/64 answer-preserving and 57/64 answer-changing successes, missing the required 58/64 threshold, with four failures coming from the additional downstream classifier and three cases in which the model initially followed the substituted activation but later returned to the original answer. The final test therefore used completed downstream behaviour and was fixed before a fresh evaluation set was opened.

On fresh examples, sham and answer-preserving interventions passed in 32/32 cases and answer-changing interventions in 30/32, supporting bounded causal influence of the patched layer-20 representation while still leaving open whether h alone explains the effect, since the intervention replaces the full activation. No stable enough signal was found in ordinary generated text to open a held-out recovery test, so the study did not reach end-to-end provenance.

5.5 Developmental less-constrained text recovery

Finally, we weakened the output restriction while retaining the engineered state. The strongest development run ranked the correct state first in 27/32 outputs and within the top four in all 32, with mean token-level KL divergence 0.0439 nats.

It nevertheless failed the joint reliability requirements: signal detection reached AUC 0.764, and 19/32 outputs passed the blind prose-quality criterion, matching generation without provenance encoding. A separate design raised detection AUC to 0.890 but weakened recovery, so no design qualified for held-out confirmation.

6 Discussion and Limitations

The main result is that a registered state can form part of a complete provenance chain around frozen pretrained components, where the selector derives a discrete state from activations, the added pathway makes later computation depend on it, and the state observed downstream authorises the signal recovered from the exported output. Because answer-equivalent states such as 3 and 11 produce the same visible answer, their different recovered identities cannot be explained by the answer alone.

This result is conditional on the engineered pathway, where the sixteen-way state in Gemma and Qwen is created by the added selector, insertion mechanism, and downstream classifier. The interventions show that this state controls the registered answer behaviour, rather than showing that the pretrained models originally relied on the same variable. The synthetic task separates state from answer identity but does not establish provenance for the models' complete reasoning, while remaining selector errors may reflect imperfect separation at the chosen activation sites. The separate Gemma study narrows this gap, with a layer-20 representation carrying $h = (a + b) \bmod 4$ passing the causal intervention test even though the full activation was replaced rather than an isolated h-subspace. This supports causal influence of the representation rather than unique mediation by h, while no stable enough trace was found in ordinary generated text for held-out recovery.

The trust boundary remains a central limitation because HMAC-SHA-256 authenticates the execution record but cannot establish that the hidden state was measured correctly. The public decoder likewise recovers the encoded state without independently authenticating the hidden computation, so the method depends on trusted instrumentation, authorisation, and secure control of the HMAC key.

Gemma establishes recovery from saved text after fresh retokenisation, whereas Qwen establishes recovery from exported generated token IDs; robustness to serialisation changes, paraphrasing, translation, truncation, or editing remains unproven. The models also use different insertion sites and output procedures, so the results do not show direct mechanism transfer.

The next step is to study states with independent semantic importance, such as routing, policy, or tool-selection decisions. Stronger computational provenance will also require isolating naturally occurring causal states, reducing reliance on trusted instrumentation, and preserving them through less constrained generation. Multi-step provenance would further require binding sequences of causal states rather than a single variable.

7 Conclusion

We extend computational provenance from purpose-built networks to systems built around frozen pretrained language models. In Gemma 3 1B and Qwen3-0.6B, a discrete sixteen-way state is derived from frozen activations, inserted into later computation through a trained pathway, read out downstream, authenticated in an execution-bound record, and encoded into generated output for recovery. Answer-preserving interventions test whether full state identity remains distinguishable when the visible answer is unchanged, while answer-changing interventions test whether the frozen vocabulary head follows the inserted state. Across accepted confirmation executions, the registered chain achieved exact recovery in both models with the required negative-control outcomes.

This establishes bounded computational provenance for an engineered state pathway, not provenance for the models' complete reasoning or proof of hidden execution from the output alone. A separate Gemma study found measurable causal influence for a representation already present in the frozen model, but no stable recoverable trace in ordinary generated text. Less-constrained generation achieved substantial state recovery without jointly satisfying recovery, signal-detection, and prose-quality requirements. The main open problem is whether the same evidential link can be made reliable for naturally occurring causal states, including states relevant to white-box misalignment monitoring, and for more flexible generation without relying on a tightly engineered pathway.

AI USE STATEMENT

Generative AI tools were used as research and writing assistants to help develop experimental protocols and code, inspect and summarise retained evidence, and revise manuscript prose. The authors remain responsible for the experimental design, execution, verification, citations, claims, and final manuscript.

REPRODUCIBILITY STATEMENT

The paper and supplement specify the registered constructions, thresholds, denominators, evaluation rules, and terminal outcomes. The anonymous review archive includes a reproduction guide, a SHA-256 manifest, manuscript sources, retained public evidence, the final cross-model registry, and read-only verification entry points. It gives exact commands to verify the review object and rebuild the PDF without training a model, generating new claim-bearing outputs, or opening sealed data. Model/checkpoint identifiers, seeds, configurations, and primitive evidence bindings are recorded in the accompanying artifact ledger.

REFERENCES

Benjamin Belay. Towards computational provenance: Carrying causal-state evidence in generated text. *arXiv preprint arXiv:2608.16868*, 2026. URL https://arxiv.org/abs/2608.16868.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 2022. doi: 10.1162/coli_a_00422. URL https://aclanthology.org/2022.cl-1.7/.

Leon Bergen, Usha Bhalla, Andrew Lee, Barak Widawsky, Linas Nasvytis, Connor Watts, Siddharth Boppana, Sidharth Baskaran, Dron Hazra, Michael Byun, Atticus Geiger, Owen Lewis, Matthew Kowal, Vasudev Shyam, Thomas Fel, Thomas McGrath, Ekdeep Singh Lubana, and Jack Merullo. Monitoring and discovering reward hacking with internal representations during LLM evaluations. *arXiv preprint arXiv:2609.19101*, 2026. doi: 10.48550/arXiv.2609.19101. URL https://arxiv.org/abs/2609.19101.

H. Birkholz, D. Thaler, M. Richardson, N. Smith, and W. Pan. Remote ATtestation procedureS (RATS) architecture. RFC 9334, Internet Engineering Task Force, 2023. URL https://www.rfc-editor.org/rfc/rfc9334.html.

Samuel R. Bowman, Jeeyoon Hyun, Ethan Perez, Edwin Chen, Craig Pettit, Scott Heiner, Kamile Lukosiute, Amanda Askell, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Christopher Olah, Daniela Amodei, Dario Amodei, Dawn Drain, Dustin Li, Eli Tran-Johnson, Jackson Kernion, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Liane Lovitt, Nelson Elhage, Nicholas Schiefer, Nicholas Joseph, Noemi Mercado, Nova DasSarma, Robin Larson, Sam McCandlish, Sandipan Kundu, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Timothy Telleen-Lawton, Tom Brown, Tom Henighan, Tristan Hume, Yuntao Bai, Zac Hatfield-Dodds, Ben Mann, and Jared Kaplan. Measuring progress on scalable oversight for large language models. *arXiv preprint arXiv:2211.03540*, 2022. URL https://arxiv.org/abs/2211.03540.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=ETKGuby0hcs.

Paul Christiano, Ajeya Cotra, and Mark Xu. Eliciting latent knowledge: How to tell if your eyes deceive you. Technical report, Alignment Research Center, December 2021. URL https://www.alignment.org/blog/arcs-first-technical-report-eliciting-latent-knowledge/.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=F76bwRSLeK.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. Amnesic probing: Behavioral explanation with amnesic counterfactuals. *Transactions of the Association for Computational Linguistics*, 9:160–175, 2021. doi: 10.1162/tacl_a_00359. URL https://aclanthology.org/2021.tacl-1.10/.

Kit Fraser-Taliente, Subhash Kantamneni, Euan Ong, Dan Mossing, Christina Lu, Paul C. Bogdan, Emmanuel Ameisen, James Chen, Dzmitry Kishylau, Adam Pearce, Julius Tarng, Alex Wu, Jeff Wu, Yang Zhang, Daniel M. Ziegler, Evan Hubinger, Joshua Batson, Jack Lindsey, Samuel Zimmerman, and Samuel Marks. Natural language autoencoders produce unsupervised explanations of LLM activations. *Transformer Circuits Thread*, 2026. URL https://transformer-circuits.pub/2026/nla/index.html.

Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. Causal abstractions of neural networks. In *Advances in Neural Information Processing Systems*, volume 34, pp. 9574–9586, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/4f5c422f4d49a5a807eda27434231040-Abstract.html.

Atticus Geiger, Zhengxuan Wu, Hanson Lu, Joshua Rozner, Elisa Kreiss, Thomas Icard, Noah D. Goodman, and Christopher Potts. Inducing causal structure for interpretable neural networks. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 7324–7338, 2022. URL https://proceedings.mlr.press/v162/geiger22a.html.

Zahra Ghodsi, Tianyu Gu, and Siddharth Garg. Safetynets: Verifiable execution of deep neural networks on an untrusted cloud. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/6048ff4e8cb07aa60b6777b6f7384d52-Abstract.html.

Fabrice Harel-Canada and Amit Sahai. Slam: Structural linguistic activation marking for language models. *arXiv preprint arXiv:2605.05443*, 2026. doi: 10.48550/arXiv.2605.05443.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2733–2743. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1275. URL https://aclanthology.org/D19-1275/.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, 2023. URL https://proceedings.mlr.press/v202/kirchenbauer23a.html.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5338–5348, 2020. URL https://proceedings.mlr.press/v119/koh20a.html.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, Kamile Lukosiute, Karina Nguyen, Newton Cheng, Nicholas Joseph, Nicholas Schiefer, Oliver Rausch, Robin Larson, Sam McCandlish, Sandipan Kundu, Saurav Kadavath, Shannon Yang, Thomas Henighan, Timothy Maxwell, Timothy Telleen-Lawton, Tristan Hume, Zac Hatfield-Dodds, Jared Kaplan, Jan Brauner, Samuel R. Bowman, and Ethan Perez. Measuring faithfulness in chain-of-thought reasoning. *arXiv preprint arXiv:2307.13702*, 2023. URL https://arxiv.org/abs/2307.13702.

Shuliang Liu, Xingyu Li, Hongyi Liu, Yibo Yan, Bingchen Duan, Qi Zheng, Fang Dong, Lingfeng Su, and Xuming Hu. Distilling the thought, watermarking the answer: A principle semantic guided watermark for reasoning large language models. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=T6NVogsXCZ.

Jiacheng Lu, Yiming Li, Tao Song, Weijian Wang, Wenjie Qu, Haibing Guan, and Jiaheng Zhang. Echoes within the reasoning: Stealthy and effective watermarking via chain of thought. In

Proceedings of the 43rd International Conference on Machine Learning, volume 306 of *Proceedings of Machine Learning Research*, 2026. URL https://openreview.net/forum?id=cTCFYMqulL.

Thomas McGrath, Matthew Rahtz, Janos Kramar, Vladimir Mikulik, and Shane Legg. The hydra effect: Emergent self-repair in language model computations. *arXiv preprint arXiv:2307.15771*, 2023. URL https://arxiv.org/abs/2307.15771.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, volume 35, pp. 17359–17372. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/6f1d43d5a82a37e89b0665b33bf3a182-Abstract-Conference.html.

Florian Tramèr and Dan Boneh. Slalom: Fast, verifiable and private execution of neural networks in trusted hardware. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rJVorjCcKQ.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://arxiv.org/abs/2305.04388.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. Investigating gender bias in language models using causal mediation analysis. In *Advances in Neural Information Processing Systems*, volume 33, pp. 12388–12401. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/hash/92650b2e92217715fe312e6fa7b90d82-Abstract.html.

Zhengxuan Wu, Atticus Geiger, Thomas Icard, Christopher Potts, and Noah D. Goodman. Interpretability at scale: Identifying causal mechanisms in alpaca. In *Advances in Neural Information Processing Systems*, volume 36. Curran Associates, Inc., 2023. doi: 10.52202/075280-3418. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/f6a8b109d4d4fd64c75e94aaf85d9697-Abstract-Conference.html.

KiYoon Yoo, Wonhyuk Ahn, and Nojun Kwak. Advancing beyond identification: Multi-bit watermark for large language models. *arXiv preprint arXiv:2308.00221*, 2023. URL https://arxiv.org/abs/2308.00221.

Zhuohao Yu, Xingru Jiang, Weizheng Gu, Yidong Wang, Qingsong Wen, Shikun Zhang, and Wei Ye. SAEMark: Steering personalized multilingual LLM watermarks with sparse autoencoders. In *Advances in Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=tXnyVPNOfa.

A Supplementary experimental record

Reader-facing names are used throughout. Internal registry identifiers, seeds, and exact artifact bindings are retained in the accompanying evidence manifest for reproducibility.

This appendix first documents the final Gemma and Qwen constructions evaluated in the main paper, including their architecture, training, evaluation protocol, confirmation results, and evidence bindings. Earlier experimental lineages and preserved negative results follow afterwards.

A.1 End-to-end execution protocol

For each admitted execution, the frozen model first produces the activations used by the trained selector. The selector chooses a registered state z, which may be replaced by a registered intervention value before insertion into the downstream residual stream. The frozen downstream model then produces the task answer, after which a trained readout reports the realised state. Trusted instrumentation records this observation with the execution context and authenticates the record using HMAC. Only

an accepted record authorises state-dependent output encoding. The resulting artifact is decoded using the exported output and the registered public recovery inputs. For Gemma, recovery operates on the saved text alone. For Qwen, recovery operates on the exported generated token IDs together with the public nonce and report indices required to reconstruct the registered token partitions.

For an accepted provenance execution, the registered protocol requires

$$z_{\text{inserted}} = z_{\text{observed}} = z_{\text{authorised}} = z_{\text{recovered}},$$

where $z_{\text{recovered}}$ is obtained from the exported artifact and its registered public decoding inputs. Task-state prediction is evaluated separately against z_{task}.

A.2 Final pretrained-model implementation

Layer numbers below are zero-based indices into `model.model.layers`, and all sites are post-block residuals. The pretrained model weights and language-model heads were frozen throughout.

The registered state space contains 16 categories. The state variable therefore carries at most $\log_2 16 = 4$ bits of registered identity. This capacity refers only to the finite provenance state and not to the model's complete internal computation or reasoning process.

The shared registered 16-state ontology used by both pretrained-model interfaces is shown in Table 3. States separated by eight identify different subjects while preserving the same eight-way group answer, enabling answer-preserving interventions.

State	Subject	Group
0	Alice	Red
1	Bob	Blue
2	Carol	Green
3	David	Yellow
4	Erin	Black
5	Frank	White
6	Grace	Orange
7	Henry	Purple
8	Iris	Red
9	Jack	Blue
10	Karen	Green
11	Liam	Yellow
12	Maya	Black
13	Noah	White
14	Olivia	Orange
15	Peter	Purple

Table 3: Registered subject states and their eight answer groups. States separated by eight correspond to different subjects but the same final group answer.

Gemma 3 1B. The Gemma selector reads the residual stream after layer 12 and averages the non-padding prompt-token activations. Each coordinate is standardised using statistics from the selector training set, after which a linear $1{,}152 \to 16$ classifier selects one of the sixteen registered states. The selected state indexes a learned representation that is projected back into the residual stream after layer 16. A separate readout after layer 25 identifies which state remains present downstream, while the final answer is produced independently by the frozen language-model head. The selector and downstream channel contain 111,648 trainable parameters in total; all pretrained Gemma parameters remain fixed.

For both models, the selected state indexes a learned state representation e_z, which is mapped into model width by a learned projection P and added to the residual stream,

$$h'_{\ell,t} = h_{\ell,t} + Pe_z.$$

The pretrained residual activation is therefore modified rather than replaced. The downstream observer reads only a later frozen residual activation and predicts one of the sixteen registered states.

Gemma insertion and observation. Gemma uses a learned 16×64 state codebook and a bias-free $64 \to 1{,}152$ projection. The projected update is layer-normalised, multiplied by a learned scalar $4\sigma(\alpha)$, and added at one appended state-insertion position immediately after the valid prompt, after layer 16. The observer reads that same position after layer 25 and applies an affine $1{,}152 \to 16$ map with no hidden nonlinearity. The observer was frozen before confirmation, and its argmax—not the original selector label—is the state entered into the execution record.

Qwen3-0.6B. The Qwen selector reads the final prompt-token activation at layer 18, standardises it using selector-training statistics, and applies a linear $1{,}024 \to 16$ classifier. The selected state indexes a learned representation that is projected back into the residual stream at the same location. A separate readout at layer 19 identifies the downstream state but does not feed the model or carrier. The selector and downstream channel contain 99,361 trainable parameters in total; all pretrained Qwen parameters remain fixed.

Qwen insertion and observation. Qwen uses a learned 16×64 state codebook and a bias-free $64 \to 1{,}024$ projection. The projected update is layer-normalised, multiplied by a learned scalar $4\sigma(\alpha)$, and added to the final prompt-token residual after layer 18. The observer reads the same token position after layer 19 and applies an affine $1{,}024 \to 16$ map with no hidden nonlinearity. It is read-only: its output does not feed the model or the output encoder.

Answer generation. For both models, each of the eight registered lowercase colour answers was verified to be a single tokeniser entry. The frozen pretrained vocabulary head produces full-vocabulary logits at the final prediction position, and the registered answer is their greedy argmax. An argmax outside the eight registered colour-token IDs is invalid and counts as incorrect. No separately trained answer classifier is used, and the answer token is fixed before report generation begins.

The layer and token sites were selected during development and frozen before confirmation. They are registered implementation choices, not claims that these locations uniquely represent or mediate the task computation.

The causal dependence tested in the confirmation evaluation is therefore a property of the added finite-state pathway around the frozen backbone. It is not presented as evidence that the same discrete state was already used by the unmodified pretrained model.

A.3 Authentication and output-recovery protocol

After answer generation and downstream observation, trusted instrumentation creates an execution record containing the observer-reported state and the registered execution context. The record is serialised canonically and authenticated using HMAC-SHA-256. Receipt verification checks the tag and the execution binding before any state-dependent output encoding is authorised.

The HMAC authenticates the record bytes under the trusted-key assumption; it does not independently establish that the observer measured the correct state. The public decoder receives neither the HMAC key nor the receipt and therefore performs state recovery rather than public execution authentication.

Record	Canonically HMAC-bound fields
Gemma execution receipt	Protocol ID, model revision, checkpoint SHA-256, row ID, intervention family, selected state, consumed state, downstream state, answer-token ID, codebook-entry SHA-256, injection SHA-256, and post-injection residual SHA-256. The receipt also contains its schema, canonical payload, tag, and receipt ID.
Qwen exact-event receipt	Protocol, model and revision, adapter checkpoint, run/context/execution IDs, nonce, prompt digest, event sequence and type, layer and token position, selected and consumed states, intervention family, codebook identity and entry hash, injection and residual hashes, and monitor description.
Qwen abstract receipt	Protocol, state ontology, model domain, adapter checkpoint, state type and category, codebook-entry hash, and exact-receipt ID.
Qwen causal-chain receipt	Exact- and abstract-receipt IDs; run/context/execution IDs and nonce; selection, injection, downstream, and answer sequence numbers; selected, consumed, and downstream states; answer class, token ID, and digest.

Table 4: Fields bound by the authenticated execution records. The carrier configuration was frozen by the registered protocol archive but was not separately included in the HMAC-covered receipt. The authentication claim therefore applies to the recorded execution and state, conditional on use of the archived carrier specification.

Constrained output encoding. The authorised state $z \in \{0, \ldots, 15\}$ is represented by four bits. At each designated report position, public rules partition eligible tokeniser entries into sets representing bit 0 and bit 1. For the scheduled bit b, tokens assigned to $1 - b$ are excluded and the model's original next-token distribution is renormalised over the permitted set. The report is therefore autoregressively model-generated, while every designated token supplies a public vote for one state bit.

Gemma repeats the four-bit schedule nine times in a 36-token report. The exported envelope is `R3TC2:<answer length>:<report length>:<answer><report>`, with the two lengths encoded as eight hexadecimal digits. Recovery starts from the saved text, parses this envelope, and freshly retokenises only the report. It accepts only if exactly 36 token IDs are recovered and every bit receives nine identical votes; otherwise it abstains.

Qwen emits three 12-token reports. Within each report, the schedule $0, 1, 2, 3$ is repeated three times, giving nine votes per bit across the three reports. The public decoder receives the exported generated token IDs together with the public nonce and report indices required by the frozen recovery rules. It uses these inputs to reconstruct the nonce-, report-, and position-dependent token partitions. Each bit is accepted only when all nine registered votes are present and at least seven agree; if any bit fails this rule, the decoder abstains. The decoder receives no prompt, activations, authentication record, secret key, intervention information, or task label. When provenance encoding is disabled, no state-dependent token restriction is applied. Rejected or missing authentication cannot authorise a registered state-dependent encoding.

Representative exported artifacts and decoder outputs. The following examples are illustrative and were not used to select the protocol. Material labelled "exported artifact" is supplied to the public decoder. Material labelled "decoder output" is produced by that decoder and is not part of the model-generated report. In the Gemma rendering, `[U+0930 U+0942 U+092A]` denotes the exact three-code-point token occurring at that position in the saved UTF-8 report.

Gemma exported artifact.

> **Answer:** `yellow`
> **Report:**
> `please provide me with the original public provenance report!`
> `I haven'T received it.<end_of_turn>tained<end_of_turn>velop <end_of_turn>[U+0930 U+0942 U+092A]<end_of_turn><end_of_turn> <end_of_turn>tained<end_of_turn><end_of_turn><end_of_turn> tained<end_of_turn><end_of_turn><end_of_turn>tained <end_of_turn>`

Gemma public-decoder output.

> **Recovered state:** 11.

Qwen exported artifact.

> **Answer:** `red`
> **Public decoding context supplied with the exported token IDs:** public nonce and report indices retained in the registered artifact record.
> **Report text:** `**PRODUCTION PLAN:** PROVENANCE AND DIGIC(`
> **Generated token IDs:** `[334, 9117, 47546, 72566, 66963, 5308, 41022, 8440, 3567, 55296, 1317, 7]`

Qwen public-decoder output.

> **Recovered state:** 0 (from this report and its two companion reports).

The generated reports are reproduced exactly, including malformed or unnatural text. Their linguistic quality is not part of the confirmed claim: the constrained carrier was selected to test reliable state carriage and recovery, not natural-language quality.

A.4 PRETRAINED-MODEL TRAINING AND DATA SPLITS

Gemma 3 1B. The selector used 16-way cross-entropy and AdamW for exactly 1,500 steps (16,384 rows, batch 256, learning rate 10^{-2}, weight decay 10^{-3}, gradient clipping at 5.0), with no early stopping and only the step-1,500 checkpoint eligible. The channel minimised the equally weighted sum of 16-way observer cross-entropy and full-vocabulary answer-token cross-entropy using AdamW for exactly 2,048 steps (15,360 rows, batch 8, learning rate 10^{-3}, weight decay 10^{-3}, gradient clipping at 5.0), again with no early stopping and only the step-2,048 checkpoint eligible.

The channel's remaining 1,024 source rows formed a qualification split disjoint from training under the registered causal matching key

$$\mathrm{SHA256}(\texttt{ENG06B_MATCH_V1},\ \text{object},\ \text{template},\ \text{transfer depth},\ \text{variant}).$$

No row ID, context ID, or causal matching key appeared in both populations. Both populations were separate from the later 2,304-row selector evaluation, 2,304-row fresh-natural evaluation, and 4,096-row sealed confirmation.

Qwen3-0.6B. The selector used 16-way cross-entropy and AdamW for exactly 60 epochs (16,384 rows, batch 256, learning rate 3×10^{-3}, weight decay 10^{-4}), with no early stopping and the final epoch fixed in advance. The channel minimised the equally weighted sum of downstream-state cross-entropy and full-vocabulary answer-token cross-entropy using AdamW for exactly 4,000 optimiser steps (16,384 rows, microbatch 4, three-step gradient accumulation, learning rate 10^{-4}, weight decay 10^{-2}, gradient clipping at 0.1), with no retry, early stopping, or alternative checkpoint selection; only step 4,000 was eligible.

The separate 4,096-row development population was disjoint from the balanced 256-row prospective selector-confirmation population.

For both selectors, coordinate-wise activation means and scales were computed from selector-training rows only and frozen before any qualification or confirmation population was opened. The same frozen statistics were used for all later natural and intervention examples.

Task-data generation. Rows are generated from the registered sixteen-subject ontology in Table 3. Objects are {key, coin, book, ring}; transfer depths are 1, 2, and 3; and the four surface templates are narrative sentences, numbered events, ledger arrows, and quoted receipt dialogues. Each row fixes an initial holder, samples a chain of distinct intermediate holders under its split seed, and ends at a deterministically specified final holder whose state gives the task label and whose state modulo eight gives the colour answer.

A complete generator cycle is the Cartesian product of 16 final states, four objects, four templates, and three depths. Full cycles are exactly balanced; any remainder uses depth $(z + o + t + s) \bmod 3 + 1$ for final-state index z, object index o, template index t, and split seed s. The initial state is $(z + 1 + ((i/16 + s) \bmod 15)) \bmod 16$, where integer division of factor index i is used, guaranteeing that it differs from the final state. Intermediate holders are drawn without replacement from the remaining subjects using a row-local seed derived from final state, initial state, variant, and depth. Row IDs hash the complete generated context; duplicate row IDs are forbidden; populations are shuffled with independent frozen split seeds. Consequently, the 4,096-row Gemma confirmation has 256 rows per state, 1,024 per object, 1,024 per template, and depth counts 1,366, 1,365, and 1,365. No confirmation row contributes to selector standardisation or model selection.

The training, development, and confirmation splits are row-disjoint but share the registered ontology, person-to-state and state-to-colour mappings, task structure, template families, object vocabulary, and transfer-depth range. The confirmation results therefore measure performance on unseen generated rows within the registered task distribution. They do not establish transfer to unseen entities, mappings, templates, ontologies, or task structures.

A.5 PRETRAINED-MODEL EVALUATION PROTOCOL AND GATES

All model interfaces, checkpoints, populations, intervention families, receipt rules, carrier settings, thresholds, and replay procedures were frozen before their corresponding confirmation populations were opened. The final-model denominators and registered endpoints were:

Table 5: Final pretrained-model evaluation populations and gates.

Role	Primary denominator	Registered endpoint
Gemma selector evaluation	2,304 rows	at least 95% overall and 90% worst-state accuracy
Gemma fresh natural evaluation	2,304 rows	state and answer accuracy; exact predicted/inserted/consumed/observed-state equality and frozen-model integrity checks
Gemma sealed confirmation	4,096 rows; eight balanced 512-row families; 1,536 autonomous rows	frozen state, transport, answer, receipt, recovery, replay, wrong-valid, and control gates
Qwen selector confirmation	256 balanced rows	at least 93% overall and 75% per-state accuracy
Qwen authorised executions	256 natural + 256 wrong-valid	exact selected/consumed/observed state, answer, and receipt chains
Qwen report recovery and replay	2,048 reports	exact replay; 512 reject/abstain controls with no provenance

Family	Inserted state	State changes	Answer changes	Expected observed state	Expected output	Expected provenance
Natural	z	No	No	z	$z \bmod 8$	Accept z
Sham	z	No	No	z	$z \bmod 8$	Accept z
Same-state	z	No	No	z	$z \bmod 8$	Accept z
Alias $+8$	$(z + 8) \bmod 16$	Yes	No	inserted	same group	Accept inserted
Non-alias $+1$	$(z + 1) \bmod 16$	Yes	Yes	inserted	inserted $\bmod 8$	Accept inserted
Wrong-valid $+4$	$(z + 4) \bmod 16$	Yes	Yes	inserted	inserted $\bmod 8$	Accept inserted
Norm-matched random	n/a	n/a	n/a	no registered target	no registered target	Reject; no encoding
Code-off	none	n/a	n/a	no registered target	no registered target	Reject; no encoding

Table 6: Registered intervention and control families. The selector output is z; z_{inserted} denotes the state supplied to the downstream pathway after any intervention. "Accept z_{inserted}" requires the observed, authorised, and recovered states to equal z_{inserted}.

Disposition	Gemma	Qwen
Admitted	4,096	1,024
Expected accept	3,072	512
Accepted	3,072	512
Expected reject/abstain	1,024	512
Correctly rejected/abstained	1,024	512
Unexpected failure	0	0
Excluded after admission	0	0

Table 7: Disposition of all admitted confirmation executions. "Accepted provenance executions" denotes the protocol-admitted executions registered in advance to produce an accepted provenance artifact; it is not a post-hoc subset selected from the results.

State-prediction and task-answer accuracies are learned-performance estimates. Receipt verification, replay, recovery, and execution-disposition results are registered protocol checks conditional on the state supplied to the downstream pathway. Their exact pass rates should therefore not be interpreted as independently learned prediction accuracies.

A.6 Pretrained-model confirmation results

The Gemma selector was trained on 16,384 authorised rows and fixed at step 1,500 before its 2,304-row selection population opened. It achieved 2,268/2,304 (98.44%) overall and 137/144 (95.14%) in the worst state. A fresh 2,304-row natural evaluation then achieved 2,260/2,304 (98.09%) state accuracy, 133/144 (92.36%) worst-state accuracy, and 2,261/2,304 (98.13%) end-to-end answer

Model	Family	Population	Observed-state success	Answer success	Recovery / disposition
Gemma	Natural	512	512/512	512/512	512/512 recovered
Gemma	Sham	512	512/512	512/512	512/512 recovered
Gemma	Same-state	512	512/512	512/512	512/512 recovered
Gemma	Alias $+8$	512	512/512	512/512	512/512 recovered
Gemma	Non-alias $+1$	512	512/512	512/512	512/512 recovered
Gemma	Wrong-valid $+4$	512	512/512	512/512	512/512 recovered
Gemma	Norm-matched random	512	n/a	n/a	512/512 rejected; no artifact
Gemma	Code-off	512	n/a	n/a	512/512 rejected; no artifact
Qwen	Natural-selector accept	256	256/256	256/256	256/256 recovered
Qwen	Wrong-valid $+4$ accept	256	256/256	256/256	256/256 recovered
Qwen	Reject/abstain controls	512	n/a	n/a	512/512 rejected or abstained

Table 8: Per-family confirmation outcomes. Success denominators apply only where the protocol defines a registered state or answer target.

accuracy, with exact predicted/inserted/consumed/observed-state and frozen-model integrity checks. Subsequent causal-intervention, receipt-authentication, output-encoding, recovery, and public-replay stages qualified before the sealed 4,096-row confirmation opened. Confirmation achieved 4,028/4,096 (98.34%) registered state accuracy, 241/256 (94.14%) worst-state accuracy, and 1,512/1,536 (98.44%) autonomous answer accuracy.

The Qwen confirmation used the frozen state predictor, causal pathway, receipt system, and registered constrained output encoder in one prospective attempt. Before evaluator truth was available it committed 512 task events and receipt bundles, 2,048 reports, 2,048 replay records, and 1,024 provenance decisions. The selector was correct on 245/256 rows (95.70%), with 14/16 (87.50%) in the worst state. Natural source integrity, wrong-valid forcing, downstream consumed-state and answer behaviour, all receipt chains, 512 accepted state recoveries, 2,048 public replays, and 512 reject/abstain controls were exact. The result archive was recovered and independently hash-verified before truth was opened locally.

A.7 Final pretrained-model evidence bindings

Table 9: Final pretrained-model evidence bindings. Abbreviated hashes are shown for readability; complete SHA-256 values and archive-relative artifact paths are provided in the accompanying manifest.

Stage	Protocol role	SHA-256
Gemma selector qualification	prospective selector qualification	`bdb36569...a5bfa`
Gemma natural validation	prospective natural validation	`d2c596ce...94d7f`
Gemma causal qualification	prospective intervention qualification	`49b5a031...ca899e`
Gemma sealed confirmation	sealed end-to-end confirmation	`eca45467...cd7a2a`
Qwen end-to-end confirmation	prospective end-to-end qualification	`a1617fde...0b0d5`
Final evidence registry	cross-model claim ledger	`5461959d...9c6c7`

The complete evidence manifest is itself bound by SHA-256:

`1edb60b9cde034b3d82b728ad7bf50eb95b263b7890fcb2e9194bb751f73e713.`

A.8 Earlier Gemma and Qwen construction failures

Design	Result	Failure point	Consequence
Earlier Gemma state former	did not pass qualification	state formation / causal gate	stopped before confirmation
Gemma transport-only design	16/16 state identities transported	answer-control gate failed	downstream channel redesigned
Earlier Qwen construction	exact-equivalence condition failed before population access	equivalence gate	confirmation population remained unopened

Table 10: Representative failed constructions retained during development.

A later Gemma construction was frozen prospectively and passed both corresponding qualification gates before confirmation was opened. The final Qwen construction was likewise frozen and evaluated prospectively in a single confirmation run.

A.9 Output-encoding development studies

Earlier output-encoding studies tested whether the authorised state could be recovered under progressively less restrictive generation rules.

A high-capacity hard carrier first established that the output channel had sufficient capacity to transmit the state in the accepted authenticated record reliably. Softer carrier designs then introduced an explicit carrier-off condition and required both detection of carrier presence and recovery of state identity. These designs did not reach the registered qualification criteria, showing that successful state transmission alone was insufficient when the system also had to distinguish carrier-on from carrier-off output.

We also tested a sequence-conditioned carrier that encoded the authorised state across a longer generated suffix. Mechanical tests verified the sampler and public replay procedure, and small development runs produced mostly successful or abstaining outputs rather than incorrect states. The designs nevertheless exceeded the prospective compute budget before confirmatory evaluation, so no final qualification population was opened. These studies informed the final carrier design but are not themselves part of the confirmed Gemma or Qwen provenance result.

Gemma free-text carrier study. We separately tested whether the authorised 16-state variable could be carried through less constrained Gemma continuations. Rather than restricting generation to a fixed token partition, the carrier made finite state-dependent adjustments to token probabilities, and recovery used only the exported text after fresh tokenisation.

In the strongest development run, the correct state ranked first in 27/32 outputs and within the top four in all 32. Mean token-level KL divergence from matched carrier-off generation was 0.04387 nats. Carrier-presence AUC was approximately 0.764, while blind prose quality passed in 19/32 carrier-on and 19/32 matched carrier-off outputs. A separate design improved presence detection to an AUC of 0.8896 but weakened state recovery.

These values are descriptive development results selected from explored designs, not held-out performance estimates. The 32 carrier-on and matched carrier-off outputs were used only for development evaluation at the recorded operating point; no subsequent confirmation population was opened because no single design jointly satisfied the registered recovery, signal-detection, and prose-quality requirements.

Prose-quality criterion. One condition-blind evaluator graded each output without access to its experimental condition or target state. An output passed only if all three criteria passed: it had no material corruption or contradiction, introduced no material factual error, and addressed all requested practical points. The registered design gate required at least 26/32 passes in both carrier-on and matched carrier-off conditions; the observed result was 19/32 in each condition.

This study was developmental and was not a sealed confirmation evaluation. Its results support the feasibility of less constrained state recovery, but they do not establish the reliability, linguistic quality, or end-to-end provenance guarantees demonstrated by the registered constrained carriers.

A.10 Answer-only state localisation in a separately trained transformer

To test whether the registered state could arise without intermediate supervision, we trained a separate four-layer transformer using only answer cross-entropy. Three independently trained models exceeded the frozen task-accuracy requirements, reaching at least 99.98% held-out accuracy.

One model was used for development localisation. Linear probes were fitted to several residual-stream sites and evaluated on a disjoint selection split. No site passed the preregistered requirements for recovering the full state, its eight-way quotient, and the distinction between answer-equivalent states, so the protocol stopped before causal intervention or representation analysis of the remaining models.

A second cohort was tested with probes across twenty non-CLS token locations. Again, no site qualified: the best full-state accuracy was 23.80%, the best quotient accuracy was 45.54%, and the best conditional alias-distinction accuracy was 50.39%. The remaining models were therefore not opened for representation analysis. These results show that high answer accuracy did not by itself produce a linearly recoverable version of the registered intermediate state under the tested sites and probes.

A.11 Representation carrying a candidate state in frozen Gemma

This study tested whether a representation in frozen Gemma carrying information about a four-way task variable had bounded causal influence without training a new state-forming component or adding output encoding. Activation patching was used only to test causal influence; any later recovery test had to use ordinary, unpatched generated text.

An initial search across six prompt-time sites found no qualifying linear readout. A later generation-time analysis identified a layer-20 representation of

$$h = (a + b) \bmod 4,$$

with 58/64 development accuracy. Before causal testing, we verified that the intervention procedure itself preserved the original execution: 18/18 sham interventions reproduced the baseline exactly.

Patch specification. At the registered pre-remainder generation event, the intervention replaces one generated token position's complete post-block residual vector after layer 20 with the corresponding vector from a registered source execution. Source and target use the same value of a and differ only in b. For target (a, b), the source is $(a, (b + \delta) \bmod 16)$: $\delta = 0$ is the sham, $\delta = 2$ changes h while preserving its parity, and $\delta = 1$ changes its parity. The complete residual vector is patched; the intervention is not restricted to the subspace decoded as h.

Role	(a, b)	$h = (a + b) \bmod 4$	Answer parity
Target	$(10, 0)$	2	even
Answer-preserving donor	$(10, 2)$	0	even
Answer-changing donor	$(10, 1)$	3	odd

Table 11: Concrete registered patch example. The donor changes the complete layer-20 residual at the pre-remainder event.

The first causal test combined the next generated token, a learned downstream observer, and the final answer. Answer-preserving interventions passed in 60/64 cases, while answer-changing interventions reached 57/64, below the required 58/64. Inspection showed that four failures came from the auxiliary observer and three from genuine reversions, where the model initially followed the substituted state but later returned to a computation consistent with the original answer.

Because several failures arose from the auxiliary learned observer rather than from the model's final computation, we revised the causal endpoint to depend only on completed downstream behaviour and the final answer. This endpoint was frozen before the fresh 32-example population was opened; no examples from that population were used in its selection. On fresh examples, sham interventions passed in 32/32 cases, answer-preserving interventions in 32/32, and answer-changing interventions in 30/32, passing all registered causal criteria.

This establishes bounded causal influence for the patched layer-20 representation under the registered intervention protocol. Because the intervention replaces a broader activation rather than an isolated h-subspace, it does not establish that h alone mediates the effect.

We next tested whether evidence associated with the same candidate state could be recovered from ordinary generated text. A restricted arithmetic interface produced 0/64 admissible text views. In the final richer-output pilot, only 16/32 outputs were valid (required at least 31/32), 7/32 had the registered correct answer (required at least 29/32), and the four per-state correct counts were 0, 3, 0, and 4 out of 8 (required at least 6/8 each). No output was admissible for recovery (0/32; required at least 24/32), so there were no distinct admissible views (0; required at least 16); all 32 triggered the

numeric-disclosure proxy (maximum allowed 3). These were the numerical recovery-entry criteria for the final pilot; the earlier exploratory imbalance screen had no separate claim-bearing numerical threshold.

No detector was fitted and the held-out recovery population remained unopened. This study establishes bounded causal influence for a patched representation obtained from unmodified Gemma activations. Because the intervention patches the full residual vector rather than only the decoded subspace, it does not establish that the decoded state alone mediates the effect. It also does not establish authentication, recovery from ordinary generated text, or end-to-end computational provenance for that representation.

A.12 Earlier development history

Stage	Goal	Outcome	Effect on later design
Controlled-model studies	separate provenance stages	transport and causal-use failures	staged gates adopted
Early Gemma designs	pretrained-state transport	stages passed separately; answer control failed in one design	final channel redesigned
Early Qwen design	cross-model implementation	equivalence condition failed	confirmation not opened
Final Gemma	sealed confirmation	passed registered gates	retained
Final Qwen	prospective confirmation	passed registered gates	retained

Table 12: Chronology of the retained development stages. Full denominators and immutable records remain in the evidence archive.

A.13 Threat-model matrix

Table 13: Trusted assumptions, tested properties, and unestablished properties.

Component	Trusted assumption	Tested property	Not established
Model/instrumentation	registered checkpoint and honest hooks	frozen-weight hashes and registered hard boundaries	malicious or uninstrumented computation
Receipt system	honest issuer and uncompromised key	HMAC integrity; mutation and replay rejection	malicious issuer, public verification, or key compromise
Controller/verifier	correct implementation and ordering	verification before authorisation	compromised controller or arbitrary authorisation
Selector	frozen training statistics and checkpoint	held-out and per-state prediction accuracy	unique or complete recovery of model computation
Downstream observer	honest read-only hook at registered site	forced-state and natural-state agreement	correctness under unregistered interventions
Insertion mechanism	correct layer, token, and tensor write	exact delta/residual hashes and sham tests	mediation by the injected subspace alone
Tokeniser/serialiser	frozen tokeniser and canonical byte rules	envelope parsing and fresh-retokenisation replay	robustness to arbitrary re-encoding or editing
Public decoder	published partition and voting rules	exact replay, abstention, and negative controls	execution authentication without the receipt key
Dataset generator	frozen seeds and deterministic generator	balance, duplicate, split, and hash audits	natural-data representativeness
Surface carrier	frozen operating point and eligible support	registered recovery and carrier-off controls	paraphrase, copy, collusion, or open-ended language
Gemma engineered pathway	honest instrumentation around a frozen pretrained model	selector, causal channel, receipts, text recovery, and replay	natural emergence or broad generalisation
Qwen engineered pathway	honest instrumentation around a frozen pretrained model	selector, causal channel, receipt chain, token-ID recovery, and replay	saved-text retokenisation recovery or low-distortion carriage
Qwen exported artifact	The public nonce, report indices, generated token IDs, and decoder version are retained without alteration	Registered token-ID replay and recovery tests	Recovery after conversion to saved text, fresh retokenisation, editing, or loss of the public decoding context
Native-state patch	matched registered source/target construction	sham, answer-preserving, and answer-changing behavioural effects	isolated h mediation, authentication, or ordinary-text recovery
Auditor	retained rows and complete archive	retained-row reconstruction and hash checking	independent neural regeneration

A.14 EXACT CLAIM BOUNDARY

The confirmed end-to-end evidence is limited to the registered tasks and systems built around frozen Gemma 3 1B and Qwen3-0.6B components using engineered 16-state pathways, trusted observation and authentication, and fixed evaluation procedures. Qwen recovery is established from exported generated token IDs together with the registered public decoding context, rather than from saved text after fresh retokenisation. The selector evaluations are in-distribution with respect to the registered ontology and task generator and do not establish transfer to unseen entities, mappings, templates, ontologies, or task structures. The confirmed constrained carriers establish reliable capacity and exact recovery under the registered protocol, not natural-language quality; the representative generated reports are visibly degraded. A separate Gemma study establishes bounded causal influence for a patched representation associated with a four-way state, but not authentication or recovery of that state from ordinary generated text. The free-text carrier study demonstrates substantial descriptive development recovery under less constrained generation, but did not satisfy all reliability requirements or proceed to held-out confirmation.

The combined results therefore do not establish provenance for the model's complete reasoning process, reliable free-text provenance, output-only authentication of hidden execution, public-key attestation, robustness to arbitrary editing, direct transfer of one mechanism between models, or broad model-family generalisation. Earlier output-encoding and controlled-model studies are retained as supporting development evidence rather than additional confirmed provenance results.