LATENT ASSEMBLY: CAUSAL DISCOVERY AND CROSS-TASK LINKING OF EXECUTABLE OPERATIONS IN LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Can computations learned by language models be decompiled as executable operations and assembled into new programs? We introduce a causal framework that recovers typed program states, discovers state-dependent operations, and neural interfaces through which one operation's output becomes another's input. The framework identifies operations by their causal transformations rather than their physical locations and tests reuse through predicted intermediate states, causal execution, and recompilation.

In controlled transformers, our framework discovers ground-truth operation classes and composition laws starting from unlabeled neural activations. In pretrained Qwen, Gemma, and Llama models, we discover SELECT and LOAD operations connected by a reusable record-reference interface and validate their composition and recompilation.

Most importantly, we show that three distinct operations discovered and qualified independently from separate tasks using our framework can be linked to execute semantically distinct tasks with no target-task fitting. We discover the frozen neural calling conventions through which the operations communicate. The three-operation chain achieves 84% accuracy, far exceeding baseline controls and Qwen 14B's direct zero-shot performance. Furthermore, interventions on upstream arguments produce the downstream behavior implied by the altered program.

These results establish a concrete form of neural programmability: independently discovered computations can exchange arguments and execute together. They open a path toward making computations acquired during training accessible as components that can be inspected, tested, and deliberately recombined.

1 INTRODUCTION

A compiled program can be taken apart. Given an executable, a decompiler recovers its functions, data structures, and calling conventions. A linker can then assemble separately compiled modules into a new program, provided their interfaces agree. Language models also generalize learned internal operations to solve tasks they were never explicitly trained on. Whether this computation can be taken apart in an analogous sense—discovered as executable operations that can be lifted out of one task and run inside another—remains a central open question for mechanistic interpretability.

Mechanistic interpretability has uncovered several forms of reusable computation. Circuits share components across tasks (Merullo et al., 2024; He et al., 2025), function and task vectors causally invoke and compose behaviors (Todd et al., 2024; Lippl et al., 2026), and latent attention codes configure transformations reused on unseen compositions (Schug et al., 2025).

These approaches motivate a further distinction between an operation and its implementation or invocation. Distinct circuits can realize the same computation (Bayat Makou et al., 2026); function vectors invoke behaviors, while latent attention codes configure transformations. Establishing reusable executable operations additionally requires discovering state-dependent transformation laws and testing whether their outputs can be passed through qualified interfaces to other computations.

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.

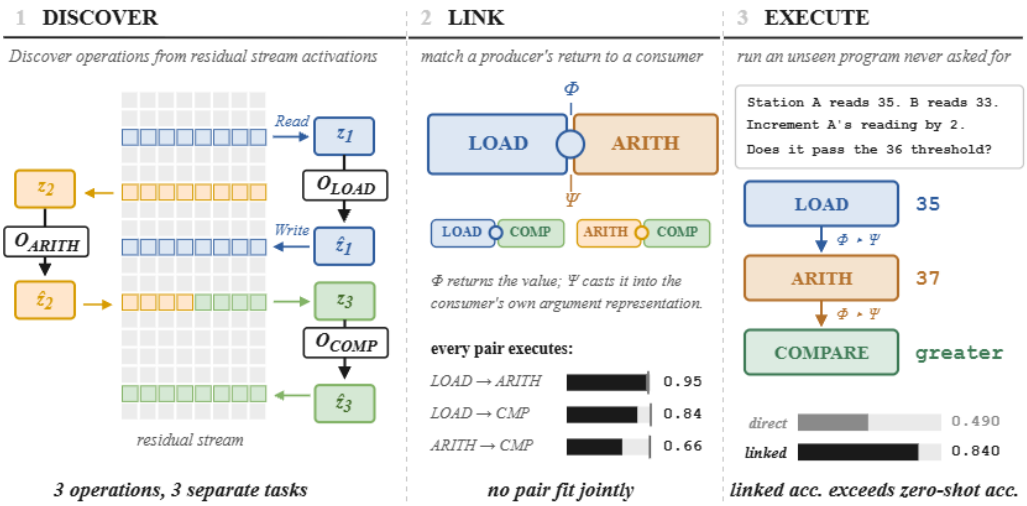

Figure 1: **Discover, link, execute.** Three operations independently discovered on separate source tasks and linked through frozen interfaces execute a held-out sensor-calibration program at 0.840 against 0.490 for the model's own single pass on strict-order comparisons. ARITH uses the ADD mode. (§3, §6).

To address these gaps, we introduce a **causal operation-discovery framework** that recovers typed program states, discovers state-dependent transformations, and qualifies their neural read/write interfaces. The framework tests candidate operations through causal execution, composition using their own predicted intermediate states, and recompilation of held-out programs.

We validate the framework in controlled transformers and pretrained language models. In transformers compiled from known programs, it recovers operation classes and composition laws, predicts held-out compositions, and distinguishes native mediation from behaviorally matched bypass routes. On structured tasks in Qwen, Gemma, and Llama, it discovers SELECT and LOAD operations connected by a reusable record-reference interface and validates their composition and recompilation across model-specific physical implementations.

We then link LOAD, ADD, and COMPARE operations independently discovered and qualified on separate source tasks in Qwen2.5-14B (Figure 1). Producer return maps and consumer casts form **neural calling conventions** that are fitted independently on source tasks and frozen for linking. The resulting library executes semantically distinct two- and three-operation programs without target-task operation fitting. A sensor-calibration chain reaches 84% accuracy on strict-order comparisons versus 49% for direct zero-shot answers. Furthermore, changing an upstream argument produces the downstream behavior implied by the altered program 99% of the time.

Our core contributions are:

- **A causal operation-discovery framework** that separates program states, transformation laws, and physical interfaces, with controlled and cross-model validation.
- **Neural calling conventions** that make independently discovered operations interoperable through separately qualified producer and consumer interfaces.
- **Cross-task execution of new two- and three-operation programs**, with counterfactual tests establishing that upstream arguments are transformed and consumed downstream.

2 RELATED WORK

Interpreting computation. Mechanistic interpretability seeks the algorithms implemented by neural networks (Olah et al., 2020; Elhage et al., 2021). Circuit analysis and discovery localize mechanisms (Olsson et al., 2022; Wang et al., 2023; Conmy et al., 2023); sparse feature methods decompose activations and trace their interactions (Bricken et al., 2023; Huben et al., 2024; Ameisen et al., 2025);

and steering and function vectors causally control behavior (Turner et al., 2023; Zou et al., 2023; Todd et al., 2024). Our unit of analysis is the state-dependent operation and its executable interface.

Causal abstraction and interfaces. Causal mediation and abstraction relate network behavior to high-level variables under intervention (Vig et al., 2020; Geiger et al., 2021; 2025), and distributed alignment search learns their neural realizations (Geiger et al., 2024). We seek to discover candidate operations and qualify their interfaces. Concurrent hidden-API work (Ma et al., 2026) identifies an interface shared by multiple consumers; we additionally discover executable transformations and link independently qualified producers and consumers.

Validating interventions. Decodability need not imply use (Hewitt & Liang, 2019; Belinkov, 2022; Elazar et al., 2021), localization need not predict editability (Hase et al., 2023), and subspace patches can exploit dormant pathways (Makelov et al., 2024). Our frame-changing interchange tests what a state carries, while compiled decoys distinguish executability from native mediation.

Program-level decompilation. RASP and Tracr provide explicit programs for transformers (Weiss et al., 2021; Lindner et al., 2023). D-RASP reverses this direction, reparameterizing trained transformers and causally extracting task-level programs (Huang et al., 2026). We instead seek neural operations that remain executable and reusable through qualified interfaces.

3 Causal Operation Discovery

3.1 Neural states and interventions

For residual stream $H^{(\ell)}(x)$, a checkpoint $c = (\ell, P)$ specifies a layer boundary and token positions. Its substrate $S_c(x) = \mathrm{Slice}(H^{(\ell)}(x), P)$ is the selected slice. The intervention $\mathrm{do}(S_c = \widetilde{S}_c)$ replaces that slice and continues the frozen model, leaving other coordinates unchanged at the intervention. A *native interchange* writes a substrate from another forward pass; a *synthesized write* constructs it with a fitted estimator.

3.2 Program states, operations, and interfaces

A program state z abstracts the information required by the computation of interest. Checkpoint-specific read and write maps connect it to neural states:

$$R_c : \mathcal{S}_c \to \mathcal{Z}_c, \qquad W_c : \mathcal{Z}_c \times \mathcal{S}_c \to \mathcal{S}_c. \tag{1}$$

Unlike a decoder, $W_c(z; S_c^r)$ realizes z in a recipient substrate S_c^r while preserving untargeted information. Causal sufficiency is tested by whether downstream computation consumes the installed state as intended.

An *occurrence* transforms states between two checkpoints. Let $T^r_{\mathrm{in:out}}$ denote the frozen continuation in recipient context r, mapping an installed input substrate to the resulting output substrate. A candidate law $O : \mathcal{Z}_{\mathrm{in}} \to \mathcal{Z}_{\mathrm{out}}$ must satisfy

$$R_{\mathrm{out}}(T^r_{\mathrm{in:out}}[W_{\mathrm{in}}(z; S^r_{\mathrm{in}})]) \approx O(z) \tag{2}$$

on held-out installed states and recipient contexts, not only native runs. Occurrences may lie in different layers, tasks, or models. After their interfaces are independently qualified in common typed coordinates, we measure agreement over matched states $z \sim \mathcal{D}$:

$$d(O^{(i)}, O^{(j)}) = \mathbb{E}_{z\sim\mathcal{D}}\Big[d_{\mathcal{Z}}\Big(O^{(i)}(z), O^{(j)}(z)\Big)\Big], \tag{3}$$

where $d_{\mathcal{Z}}$ is a task-specific output-state error. Agreement and causal substitution qualify an *operation class* over the tested domain. Identity concerns the transformation, not its physical location; an operation need not be primitive or have a unique decomposition.

3.3 Behavioral fidelity and normalized causal efficacy

Behavioral fidelity $F(a)$ is the fraction of held-out items on which intervention arm a produces the behavior implied by the installed program state. Each test prospectively specifies a baseline B

without the causal contribution under test and a reference fidelity $C > B$. We report

$$\rho(a; B, C) = \frac{F(a) - B}{C - B}. \tag{4}$$

Thus $\rho = 0$ matches the baseline and $\rho = 1$ recovers the available improvement. The reference usually installs a native target state through the same interface; for incremental chain recovery, it is the downstream suffix (Section 6.3). It is not a universal upper bound.

3.4 Executing and reusing discovered operations

Execution. A predicted state is executable when installing it through W_{out} causes the downstream model to run the corresponding program.

Composition. For independently fitted operations with compatible state spaces, we compute $\widehat{z}_1 = \widehat{O}_A(z_0)$ and $\widehat{z}_2 = \widehat{O}_B(\widehat{z}_1)$ on held-out programs. *Causal composition generalization* (CCG) requires executable states generated recursively from predicted intermediates, without substituting native ones. We report normalized efficacy ρ.

Recompilation. We assemble a held-out program from discovered components and install it in a recipient that must execute it using its own content. We report F_{recomp}, since these tests do not specify a separate normalization reference.

Cross-task linking. Independently discovered operations from different tasks may have incompatible native representations. Separately qualified producer return maps and consumer casts transfer their arguments (Section 6). Neither producer–consumer pairs nor target-task examples are used to fit these interfaces.

3.5 The causal operation-discovery framework

(1) Recover interfaces. Find candidate checkpoints and read/write maps; qualify readability, writability, and downstream consumption. **(2) Collect occurrences.** Gather pre- and post-states on designated discovery data. **(3) Discover classes.** Fit compact state-dependent laws and test held-out transfer. The *shared-versus-separate* test compares a shared law with occurrence-specific laws from the same estimator family; separate fits can have more total parameters. A *matched-input guard* requires the same prediction for identical program states across occurrence contexts, excluding pooling that merely switches between context-specific rules. **(4) Qualify operations.** Freeze candidate laws and test whether their predicted states execute in other compatible contexts. Then test CCG, recompilation, and, where the interfaces qualify, cross-task linking. Executability alone does not establish native mediation (Section 4).

4 Operation Discovery on Controlled Transformers

Section 3 defines operations by their transformations, but pretrained models provide no independent ground truth for a recovered equivalence. We therefore begin with Tracr (Lindner et al., 2023), which compiles RASP programs (Weiss et al., 2021) into transformers with known computational structure.

Tasks. Two compiled families test operation identity and composition (Figure 2). **Sequence rearrangement** copies the token at $j = f(i, \ell)$ to output position i, making the routing rule f the operation under study. Reversal, shift-by-two, and reflection recur with the length operand computed through different RASP programs; shift-by-one, shift-by-three, and doubling form singleton negative classes. The same transformation thus occurs across transformers and through different internal computation graphs.

Symmetry tasks maintain an eight-state register for the square's orientation. The eight elements of D_4, generated by quarter-rotation r and reflection s, obey $r^4 = e$, $s^2 = e$, and $rs \neq sr$. Single-operation organisms apply a symmetry followed by identity; product organisms apply two symmetries, covering all $8 \times 8 = 64$ ordered pairs. This tests whether discovery recovers the order-sensitive composition law.

Blind discovery. With operation identities withheld, we fit linear transition maps between the states before and after each anonymous rearrangement occurrence. The 12 occurrences yield $\binom{12}{2} = 66$

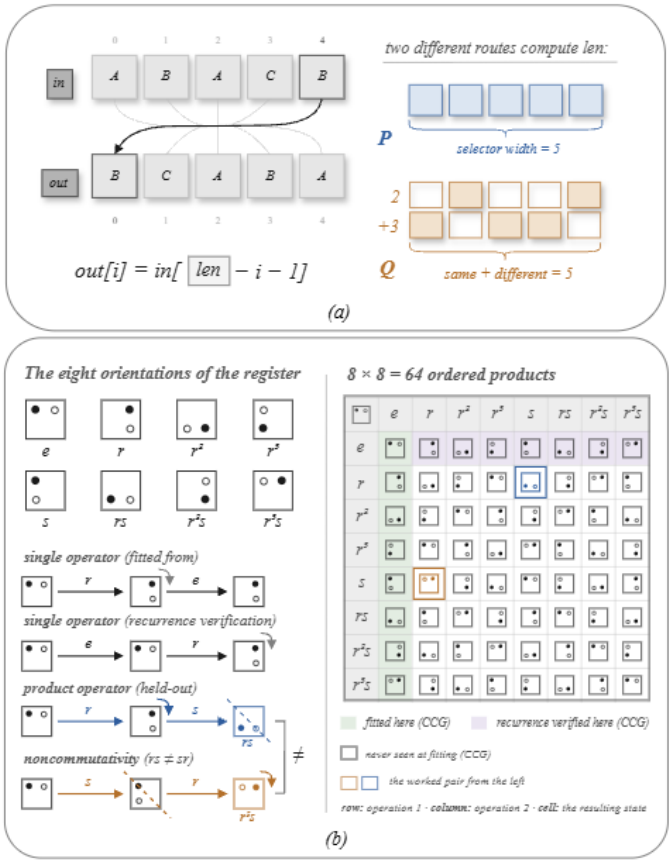

Figure 2: **Controlled task families.** **(a)** Sequence rearrangement (reversal), with different routes computing the length operand. **(b)** Square symmetries: eight register states, isolated and composed operations, and their ordered products.

pairs. The shared-versus-separate test (Section 3.5) compares one shared map with two separate maps; the latter have more parameters. The matched-input guard additionally tests against sharing achieved by branching on occurrence identity.

The recovered relation exactly reproduces all six ground-truth classes under both compiler-supplied and independently learned faithful interfaces. Same-operation pairs cost 0.0000; the smallest different-operation costs are 0.3025 and 0.2689, respectively. An attention-head-overlap classifier instead makes $33/66$ errors, separating transformation identity from this measure of implementation similarity.

Composition and execution. The recovered D_4 vocabulary assigns all 64 products to the correct class at fidelity 1.000. With twelve algebra entries withheld, all twelve are predicted correctly, including the distinct outcomes of a noncommuting pair.

On a separate split, **causal composition generalization** (CCG; Section 3.4) tests whether operators learned only from single-operation organisms recursively generate executable states. Each frozen operator first transfers without refitting to the opposite program slot. Across ten prospectively held-out compositions and all eight starting states, predicted intermediate states, final states, and behavior are correct in all $80/80$ cases. Substituting a wrong operation fails on all ten compositions; reversing order succeeds on $0/6$ noncommuting and $4/4$ commuting pairs. Native downstream consumers also correctly transform every installed intermediate. One written state supports eight distinct subsequent outcomes, ruling out a fixed final-answer interpretation.

Native mediation. Executability alone does not establish native use. We compare behaviorally identical transformers whose downstream computations consume either z_2 (MEDIATED) or a duplicate $\bar{z}_2 = z_2$ during ordinary execution (BYPASS). A localized intervention on z_2 recovers $64/64$ target transitions in MEDIATED and $0/64$ in BYPASS, distinguishing their causal routes.

5 OPERATION DISCOVERY ON PRETRAINED MODELS

We next ask whether analogous structures can be recovered from pretrained language models.

5.1 RECOVERING PROGRAM STATES

Natural program family. We study record retrieval as a two-stage program:

$$\mathrm{SELECT}(M, s) \to R, \qquad \mathrm{LOAD}(M, R, f) \to V, \tag{5}$$

where M is a record table, s selects its first or last record, R is the selected record reference, f specifies one of four fields, and V is the retrieved value. (Figure 3a).

Blind recovery of program state. The discovery procedure receives eight neural program states in anonymous order, without semantic labels or a transition graph. Within the candidate 2×4 shape, we exhaustively evaluate all $840 = 8!/(2!\,4!)$ arrangements, fitting $z_{ij} \approx \mu + a_i + b_j$ for each. The *interaction share* I measures the fraction of state variation unexplained by the two additive coordinate effects. Low I means that changing one coordinate has approximately the same neural effect across the other. On Qwen, the winning arrangement has $I = 0.0353$, versus 0.1651 for the runner-up ($\Delta = 0.1298$). Repeating the full search after destroying cell identity gives a null 5th percentile of 0.1138 and median of 0.1577. Only after freezing these analyses do we reveal the labels: the recovered coordinates match SELECTOR$\times$FIELD in all $8/8$ cells.

From program variables to typed operations. Factorization alone does not establish computational roles. We therefore transplant the donor's intermediate after SELECT into a recipient with different table contents and a different requested field (Figure 3b). The donor selects Alice and returns her job; the recipient normally selects Bob and returns his city. Returning *Lima*, Alice's city in the recipient table, indicates that the transplant changes the selected record while the recipient supplies the field and its local value.

This outcome occurs at 1.000 on Qwen2.5-14B, 1.000 on Gemma-4-12B, and 0.833 on Llama-3.1-8B, with donor-field carryover of 0.000 in every case. The intermediate thus specifies *which record to read*, supporting the type $R : \mathrm{RecordRef}$ and the signatures

$$\begin{aligned} \mathrm{SELECT} &: (\mathrm{Table}, \mathrm{Selector}) \to \mathrm{RecordRef}, \\ \mathrm{LOAD} &: (\mathrm{Table}, \mathrm{RecordRef}, \mathrm{Field}) \to \mathrm{Value}. \end{aligned} \tag{6}$$

These types distinguish the selected reference from the retrieved value and inform the linking interfaces in Section 6.

5.2 DISCOVERING AND QUALIFYING SELECT AND LOAD

We study numerical and categorical retrieval, with all fields containing numbers or categorical text, respectively. At the qualified interface, we fit state-dependent laws on designated source examples: SELECT predicts the displacement constructing R, and LOAD predicts the transition retrieving the value addressed by R and f. Their predicted states remain executable on held-out occurrences. On Qwen2.5-14B, SELECT reaches 1.000 numerical and 0.846 categorical causal execution, above positional, shuffled, and matched-random controls. Numerical LOAD reaches 1.000 against a strongest control of 0.144; categorical LOAD reaches 0.856 against 0.131. Both branches independently localize LOAD to $\mathrm{L34} \to \mathrm{L42}$.

In the **CCG test** (Section 3.4), LOAD receives the reference predicted by discovered SELECT, rather than the native intermediate. Normalized causal efficacy is $\rho = 0.8182$ on Qwen numerical retrieval and 0.7385 on categorical retrieval under the alignment-certified successor protocol (Table 1). For **recompilation**, we reconstruct a reference in one context and install it in another, requiring the recipient to execute the program using its own content. Qwen reaches $F_{\mathrm{recomp}} = 1.000$ in both domains, with transported, reversed, and matched-random controls at zero.

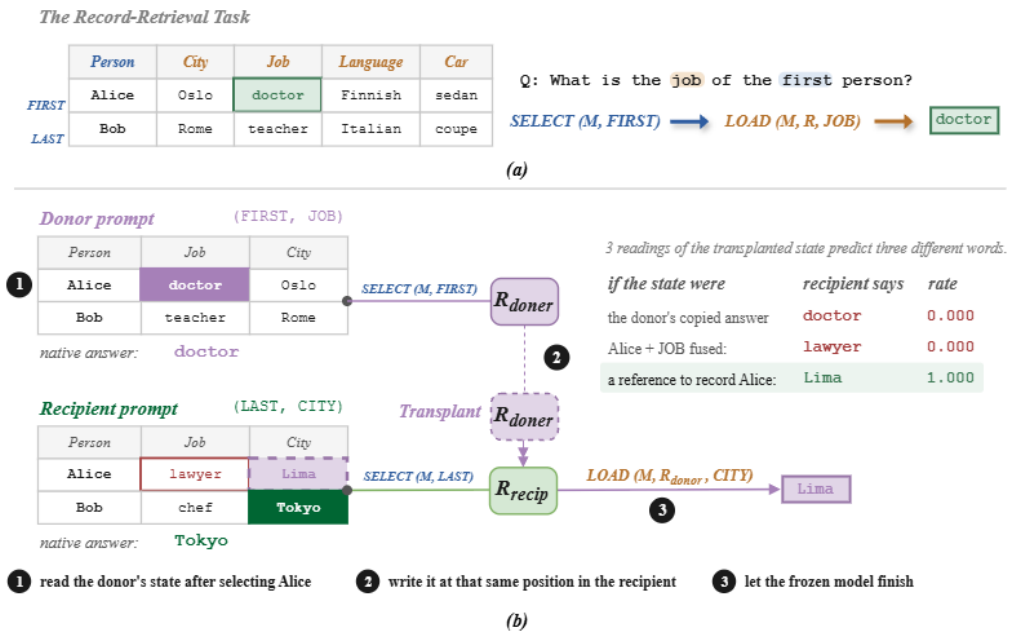

Figure 3: **Record retrieval and frame-changing interchange. (a)** Selecting the first record and loading its job returns *doctor*. **(b)** We read the donor's intermediate after selecting Alice (1), install it in the recipient (2), and continue the frozen model (3). The recipient returns *Lima*, Alice's city in its own table, rather than the donor's answer *doctor* or the donor's field applied to Alice, *lawyer*. This distinguishes a record reference from the donor's value or a record–field combination.

Replication across model families. Gemma-4-12B and Llama-3.1-8B also support discovered SELECT and LOAD operations despite different layers, ranks, and causal timing. CCG reaches $\rho = 0.8667$ and 0.6289, and recompilation reaches 0.9508 and 0.6000, respectively. Transported and reversed controls remain near zero, although Gemma's matched-random control reaches 0.3115. Thus the framework recovers a common computational organization across models with different physical realizations. Section 6 tests whether independently discovered operations also support semantically distinct tasks.

Table 1: **Natural-operation qualification.** Num./cat. denote numerical and categorical retrieval. CCG efficacy is $\rho = (F - B)/(C - B)$; B and C are the protocol's baseline and reference fidelities. transported, reversed, and matched-random recompilation controls.

Measure	Qwen-num.	Qwen-cat.	Gemma-cat.	Llama-cat.
SELECT execution	1.0000	0.8462	0.8793	0.7083
LOAD execution	1.0000	0.8562	0.9375	—
LOAD margin over control	+0.8562	+0.7250	+0.8688	+0.8938
CCG reference C	1.0000	1.0000	1.0000	0.9510
CCG baseline B	0.9450	0.6750	0.1750	0.0000
CCG fidelity F	0.9900	0.9150	0.8900	0.5980
CCG efficacy ρ	**0.8182**	**0.7385**	**0.8667**	**0.6289**
Recompilation fidelity	**1.0000**	**1.0000**	**0.9508**	**0.6000**
Strongest recompilation control	0.0000	0.0000	0.3115	0.0500

6 CROSS-TASK LINKING OF DISCOVERED OPERATIONS

A producer may compute the right value in a representation another operation cannot consume. We test whether *neural calling conventions* make independently discovered operations interoperable and reusable in new programs. All experiments in this section use Qwen2.5-14B.

Neural calling conventions. A checkpoint-independent argument space $\mathcal{V}$ separates the value passed between operations from their native program states. Producer A exposes its result through a return map Φ_A, and consumer B incorporates that value into its recipient state through a cast Ψ_B:

$$\Phi_A : \mathcal{S}_{c_A^{\text{out}}} \to \mathcal{V}, \qquad \Psi_B : \mathcal{V} \times \mathcal{Z}_{c_B} \to \mathcal{Z}_{c_B}. \tag{7}$$

For recipient substrate S_B^r, let $z_B^r = R_{c_B}(S_B^r)$. Linking constructs

$$\widetilde{z}_B = \Psi_B\Big(\Phi_A(S^A_{c_A^{\text{out}}}), z_B^r\Big), \qquad \widetilde{S}_B = W_{c_B}(\widetilde{z}_B; S_B^r)\,. \tag{8}$$

The write interface (Section 3.1) installs $\widetilde{S}_B$ at the consumer's input checkpoint, after which the frozen model continues.

6.1 INDEPENDENTLY DISCOVERED OPERATIONS INTEROPERATE

We discover and causally qualify LOAD, ADD, and COMPARE on separate record-retrieval, arithmetic, and quantity-comparison source tasks. Each producer's linear return map and each consumer's cast are fitted independently on their respective source tasks; no producer–consumer pair is fitted jointly. Installing reconstructed operand states in held-out consumer examples tests whether the interfaces transmit executable arguments.

All three tested edges execute: LOAD→ADD, LOAD→COMPARE, and ADD→COMPARE reach fidelities of 0.950, 0.840, and 0.660, against native-consumer references of 0.970, 1.000, and 0.990. Raw producer-state transplantation yields 0.000 on every edge. Changing the producer's result instead drives the downstream computation implied by that value at 0.930, 0.830, and 0.610, respectively. Thus, the interfaces transfer specific arguments that the consumer can compute with.

6.2 FROZEN LINKS EXECUTE NEW TWO-OPERATION TASKS

We fix source-qualified operations and interfaces before evaluating three semantically distinct target tasks, with no target-task operation or interface fitting. Rover routing requires retrieving and summing energy costs along a two-leg route; production quota combines arithmetic with comparison; catalog comparison combines retrieval with comparison. Table 2 reports their task-specific primary endpoints. Linked execution exceeds direct answers on production quota (0.661 vs. 0.522), but trails them on rover routing (0.713 vs. 0.925): executability does not imply a task-performance advantage.

Table 2: Frozen links on held-out tasks. Reference installs the model's own target state through the same interface. Each task uses its own primary endpoint and efficacy normalization; ρ_{link} should not be used to rank tasks.

Task	Operations	n	Linked	Reference	Direct	ρ_{link}
Rover routing	LOAD+ADD	80	0.713	0.775	0.925	0.919
Production quota	ADD+COMPARE	180	0.661	0.900	0.522	0.735
Catalog	LOAD+COMPARE	53	0.981	1.000	0.964	0.981

6.3 FROZEN LINKS COMPOSE RECURSIVELY

A longer program requires an argument to survive successive interfaces and an intervening computation. We test this on sensor calibration: retrieve a station reading, apply its correction, and compare the result with an alert threshold:

$$V = \text{LOAD}(M, R, f), \qquad U = \text{ADD}(V, \Delta), \qquad Y = \text{COMPARE}(U, T). \tag{9}$$

All operations and interfaces remain frozen. Strict-order comparisons are the preregistered primary endpoint; equality, already difficult in source-link qualification, remains a secondary stress test.

The complete chain reaches 0.840 strict-order accuracy (Table 3). Using the recipient baseline $B = 0.470$ and the two-operation suffix reference $C = 0.900$, normalized incremental recovery is $\rho_{\text{chain}} = (0.840 - 0.470)/(0.900 - 0.470) = 0.8605$ (95% CI $[0.7273, 0.9574]$). Redirecting the first LOAD to another valid station yields the final relation implied by that counterfactual program with 0.990 accuracy. Raw and matched-random chains preserve the recipient's base computation; their 0.470 target accuracy reflects coincident labels. Omitting ADD reduces overall accuracy to 0.333.

Table 3: Sensor-calibration accuracy. Strict order is the primary endpoint. The suffix executes ADD→COMPARE without upstream LOAD; the oracle supplies COMPARE's native target state. Noop, raw, and matched-random arms have identical reported accuracies.

Execution	Strict order ($n = 100$)	Equality ($n = 50$)	All ($n = 150$)
Full linked chain	**0.840**	**0.060**	**0.580**
Two-operation suffix	0.900	0.100	0.633
COMPARE oracle	1.000	0.700	0.900
Direct zero-shot	0.490	0.020	0.333
Chain-of-thought	1.000	1.000	1.000
Noop / raw / matched random	0.470	0.000	0.313

The chain exceeds direct answers by 0.350 (95% CI $[0.230, 0.470]$), but remains below chain-of-thought (1.000) and the two-operation suffix (0.900). Equality accuracy is only 0.060, reducing overall accuracy to 0.580. Linking also uses external program decomposition, internal access, and multiple model continuations.

7 DISCUSSION

Latent Assembly studies language model computation at the level of state-dependent transformations connected by typed causal interfaces. This abstraction separates what an operation computes from where it is implemented and how its arguments are represented. The recurrence of selection and retrieval across model families suggests that functional organization can remain recognizable despite differences in physical realization. Cross-task linking further makes this organization experimentally actionable: independently recovered computations can exchange intermediate values and execute programs beyond their source tasks.

However, the scope of this interpretation depends on the causal tests performed. Successful execution establishes that downstream computation can consume an installed state, but establishing that unperturbed computation naturally depends on that state requires additional evidence. Our controlled mediation and bypass experiments distinguish these properties, while the pretrained-model results establish executable interfaces without a corresponding native-mediation guarantee. Likewise, operation identity is qualified over the tested states and contexts. The recovered operations may admit finer decompositions, and the framework does not establish a unique or complete instruction set.

Another limitation is that our experiments use structured tasks and a small operation inventory. Nevertheless, our framework provides a concrete path toward broader settings: future work can discover candidate operations in richer tasks, link them through more comprehensive neural calling conventions, and test their reuse in more complex programs. Separating operation identity from physical implementation also enables investigation of which computational abstractions recur across models and how their interfaces differ. Furthermore, our counterfactual tests of operation executability and calling-convention reliability suggest a diagnostic strategy for failures of composition: intervening at intermediate stages can help distinguish failures of computation from failures of argument transfer. Ultimately, causal decompilation could make the computations acquired during training accessible through an explicit programming interface, enabling learned capabilities to be inspected, debugged, and deliberately recombined into new programs.

AI USE STATEMENT

Language models assisted with literature search and polishing the manuscript after human-written first drafts. Coding agents assisted with organizing the code repository for readability and presentation. The authors designed the experiments and checked all reported numbers against the experiment records.

REFERENCES

Emmanuel Ameisen, Jack Lindsey, Adam Pearce, Wes Gurnee, Nicholas L. Turner, Brian Chen, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, Jonathan Marcus, Michael Sklar, Adly Templeton, Trenton Bricken, Callum McDougall, Hoagy Cunningham, Thomas Henighan, Adam Jermyn, Andy Jones, Andrew Persic, Zhenyi Qi, T. Ben Thompson, Sam Zimmerman, Kelley Rivoire, Thomas Conerly, Chris Olah, and Joshua Batson. Circuit Tracing: Revealing Computational Graphs in Language Models. *Transformer Circuits Thread*, 2025. URL https://transformer-circuits.pub/2025/attribution-graphs/methods.html.

Alireza Bayat Makou, Jingcheng Niu, Subhabrata Dutta, and Iryna Gurevych. Many Circuits, One Mechanism: Input Variation and Evaluation Granularity in Circuit Discovery. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=Ur42Xx1QPv.

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/.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards Monosemanticity: Decomposing Language Models With Dictionary Learning. *Transformer Circuits Thread*, 2023. URL https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Arthur Conmy, Augustine Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. Towards Automated Circuit Discovery for Mechanistic Interpretability. In *Advances in Neural Information Processing Systems*, volume 36, pp. 16318–16352. Curran Associates, Inc., 2023. doi: 10.52202/075280-0719. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/34e1dbe95d34d7ebaf99b9bcaeb5b2be-Paper-Conference.pdf.

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/.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. A Mathematical Framework for Transformer Circuits. *Transformer Circuits Thread*, 2021. URL https://transformer-circuits.pub/2021/framework/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. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/4f5c422f4d49a5a807eda27434231040-Paper.pdf.

Atticus Geiger, Zhengxuan Wu, Christopher Potts, Thomas Icard, and Noah Goodman. Finding Alignments Between Interpretable Causal Variables and Distributed Neural Representations. In *Proceedings of the Third Conference on Causal Learning and Reasoning*, vol-

ume 236 of *Proceedings of Machine Learning Research*, pp. 160–187. PMLR, 2024. URL https://proceedings.mlr.press/v236/geiger24a.html.

Atticus Geiger, Duligur Ibeling, Amir Zur, Maheep Chaudhary, Sonakshi Chauhan, Jing Huang, Aryaman Arora, Zhengxuan Wu, Noah Goodman, Christopher Potts, and Thomas Icard. Causal Abstraction: A Theoretical Foundation for Mechanistic Interpretability. *Journal of Machine Learning Research*, 26(83):1–64, 2025. URL https://www.jmlr.org/papers/v26/23-0058.html.

Peter Hase, Mohit Bansal, Been Kim, and Asma Ghandeharioun. Does Localization Inform Editing? Surprising Differences in Causality-Based Localization vs. Knowledge Editing in Language Models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 17643–17668. Curran Associates, Inc., 2023. doi: 10.52202/075280-0774. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/3927bbdcf0e8d1fa8aa23c26f358a281-Paper-Conference.pdf.

Yinhan He, Wendy Zheng, Yushun Dong, Yaochen Zhu, Chen Chen, and Jundong Li. Towards Global-level Mechanistic Interpretability: A Perspective of Modular Circuits of Large Language Models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 22865–22880. PMLR, 2025. URL https://proceedings.mlr.press/v267/he25x.html.

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/.

Xinting Huang, Aleksandra Bakalova, Satwik Bhattamishra, William Merrill, and Michael Hahn. Discovering Interpretable Algorithms by Decompiling Transformers to RASP. In *Forty-third International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2602.08857.

Robert Huben, Hoagy Cunningham, Logan Smith, Aidan Ewart, and Lee Sharkey. Sparse Autoencoders Find Highly Interpretable Features in Language Models. In *International Conference on Learning Representations*, pp. 7827–7845, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/1fa1ab11f4bd5f94b2ec20e794dbfa3b-Paper-Conference.pdf.

David Lindner, Janos Kramar, Sebastian Farquhar, Matthew Rahtz, Tom McGrath, and Vladimir Mikulik. Tracr: Compiled Transformers as a Laboratory for Interpretability. In *Advances in Neural Information Processing Systems*, volume 36, pp. 37876–37899. Curran Associates, Inc., 2023. doi: 10.52202/075280-1649. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/771155abaae744e08576f1f3b4b7ac0d-Paper-Conference.pdf.

Samuel Lippl, Thomas Austin McGee, Kimberly Lopez, Ziwen Pan, Pierce Zhang, Salma Ziadi, Oliver Eberle, and Ida Momennejad. AlgoTrace: Algorithmic Primitives and Compositional Geometry of Reasoning in Language Models. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=ppirVueEj6.

SiYuan Ma, Yiqin Luo, Zhangji, Canran Xiao, Albert Gao, Wei Wang, Qiwei Wu, Xinran Li, Jinfeng Wei, and Qixin Zhang. Hidden APIs in Language Models: Discovering Reusable Causal Interfaces from Forked Futures. *arXiv preprint arXiv:2607.27617*, 2026. URL https://arxiv.org/abs/2607.27617.

Aleksandar Makelov, Georg Lange, Atticus Geiger, and Neel Nanda. Is This the Subspace You Are Looking for? An Interpretability Illusion for Subspace Activation Patching. In *International Conference on Learning Representations*, pp. 26486–26515, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/70b8505ac79e3e131756f793cd80eb8d-Paper-Conference.pdf.

Jack Merullo, Carsten Eickhoff, and Ellie Pavlick. Circuit Component Reuse Across Tasks in Transformer Language Models. In *International Conference on Learning Representations*, pp. 18349–18377, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/4ffbfbcf4ad7304d57158b046525e46c-Paper-Conference.pdf.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom In: An Introduction to Circuits. *Distill*, 5(3):e00024.001, 2020. doi: 10.23915/distill.00024.001. URL https://distill.pub/2020/circuits/zoom-in/.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. In-context Learning and Induction Heads. *Transformer Circuits Thread*, 2022. URL https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html.

Simon Schug, Seijin Kobayashi, Yassir Akram, Joao Sacramento, and Razvan Pascanu. Attention as a Hypernetwork. In *International Conference on Learning Representations*, pp. 68744–68770, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/abfa542f546df3c6c35695ec8d5bf4b9-Paper-Conference.pdf.

Eric Todd, Millicent L. Li, Arnab Sen Sharma, Aaron Mueller, Byron C. Wallace, and David Bau. Function Vectors in Large Language Models. In *International Conference on Learning Representations*, pp. 17282–17333, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/4ae163cb8788970e53b4fd9578141139-Paper-Conference.pdf.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering Language Models With Activation Engineering. *arXiv preprint arXiv:2308.10248*, 2023. URL https://arxiv.org/abs/2308.10248.

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_files/paper/2020/file/92650b2e92217715fe312e6fa7b90d82-Paper.pdf.

Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the Wild: A Circuit for Indirect Object Identification in GPT-2 Small. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=NpsVSN6o4ul.

Gail Weiss, Yoav Goldberg, and Eran Yahav. Thinking Like Transformers. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 11080–11090. PMLR, 2021. URL https://proceedings.mlr.press/v139/weiss21a.html.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, J. Zico Kolter, and Dan Hendrycks. Representation Engineering: A Top-Down Approach to AI Transparency. *arXiv preprint arXiv:2310.01405*, 2023. URL https://arxiv.org/abs/2310.01405.

A READER'S MAP

A.1 CONTENTS AND READING PATHS

Table 4: Appendix contents.

appendix	contents
A	this map, the claim registry, notation
B	controlled experiments on compiled transformers (§4)
C	natural experiments: SELECT → RecordRef → LOAD, cross-model replication, mixed-payload LOAD (§5)
D	cross-task linking: operations, calling conventions, pairwise edges, four target tasks (§6)
E	shared instruments: fidelity and efficacy, arm definitions, rank, statistics
F	negative, abandoned and corrected results
G	provenance: sealing protocol, preregistrations and hashes, models and compute
H	questions we expect, with pointers

We recommend 3 ways to navigate the appendix section. To check a number, find the claim in A.2 and follow its appendix and identifier to G.2. To understand an experiment, read B, C or D. Each of those subsections follows one order: design, what was frozen and when, instruments, arms and controls, then result and threats. The arms themselves are defined once, in E.2. To look for weaknesses, start with F and H.

A.2 CLAIM REGISTRY

One row per main-text claim. "Frozen" means the endpoint and gate were fixed before the evaluation items were opened. Identifiers are listed in G.2.

Table 5: Claim registry and evidence locations.

claim	§	result	frozen	appendix
one operation across unlike occurrences	4	exact six-class partition of 12 occurrences at two interfaces; head overlap wrong on 33/66 pairs	guard added by amendment	B.3
the composition law is recovered	4	64/64 products, margin 1.0000, abstention permitted	yes	B.4
prediction before the target exists	4	12/12 held-out products, predictions hashed before compilation	yes	B.5
composition through predicted states	4	80/80; wrong operator 0/10; reversal 0/6 non-commuting, 4/4 commuting	yes	B.6
predicted intermediates are reusable downstream	4	8×8 causal action table 64/64	yes	B.6
native mediation is distinguished from steering	4	MEDIATED 64/64, BYPASS 0/64; blind localization reproduces it	yes	B.7
discovery without compiler labels	4	transfer to held-out v3 within 131 bits of a free refit; D_4 up to gauge; random bank closure 15/64	yes	B.4, B.8
blind recovery of selector $\times$ field roles	5.1	8/8 cells, both payloads; bootstrap 1.0000; p_{MC} 9.99 $\times 10^{-4}$; surface invariance not adjudicated	yes	C.2, C.6

Continued on next page

Table 5 (continued)

claim	§	result	frozen	appendix
the reference is recipient-local	5.1	own value 1.000, donor value 0.000	yes	C.3
SELECT and LOAD recovered as state-dependent operations	5.2	SELECT 1.000 (numerical); LOAD gap +0.8562 over the strongest null	yes	C.4
SELECT → LOAD composition	5.2	$\rho = 0.8182$ (numerical); $\rho = 0.7385$ (categorical successor)	yes	C.5
the architecture replicates across families	5.2	stage matrix on Qwen, Gemma, Llama; carried_field 0.000 throughout; historical failures: Qwen categorical handoff and S8 (successor passes); Gemma S3b (S3c passes)	yes, with the S3c chronology stated	C.7
handoff tracks interface fidelity, not source accuracy	5.2	three-model association	descriptive	C.5
the reference interface is shared across types, but LOAD is not	App. C.8	cross-type reference 1.000; held-out LOAD transfer 0.000 across types	yes	C.8
independently recovered operations interoperate	6.1	LOAD → ARITH 0.950 / 0.970; LOAD → COMPARE 0.840 / 1.000; ARITH → COMPARE 0.660 / 0.990; raw 0.000	yes	D.4
transferred values keep their content	6.1	wrong value → own program 0.930, 0.830, 0.610	yes	D.4
ARITH consumes a supplied LOAD-form operand in rover-routing	6.2	ρ_{link} 0.9194 on 80 items; all gates pass	yes	D.5
frozen links execute production quota, beating direct	6.2	0.6611 against direct 0.5222 (LCB +0.0556); ρ 0.735	yes	D.6
frozen links execute catalog	6.2	0.981 on the strict-order endpoint	yes	D.7
three operations compose recursively	6.3	0.840; $\rho_{\text{fc,chain}}$ 0.8605 [0.7273, 0.9574]	yes	D.8
counterfactuals propagate through the chain	6.3	wrong station → own relation 0.990	yes	D.8
the chain rescues single-pass performance	6.3	+0.350 [+0.230, +0.470] over direct; −0.160 against CoT	yes	D.8

A.3 Notation and terminology

Table 6: Notation and terminology.

term	meaning	introduced
checkpoint c = (ℓ, P)	a layer boundary ℓ and a set of positions P	Section 3.1
substrate S_c	the slice of the residual stream at checkpoint c	Section 3.1
native interchange / synthesized write	an intervention whose state comes from another forward pass / from a fitted estimator	Section 3.1
program state z ∈ $\mathcal{Z}_c$	a compact, causally sufficient abstract state	Section 3.2
read map R_c, write interface W_c	$R_c : \mathcal{S}_c \to \mathcal{Z}_c$; W_c(z; S) realizes z in a recipient substrate S	Section 3.2
$T_{c:c'}$	the frozen model run from c to c′	Section 3.2

Continued on next page

Table 6 (continued)

term	meaning	introduced
occurrence	where the network carries out a transformation between two checkpoints	Section 3.2
operation	a class of occurrences that induce the same causal transformation	Section 3.2
executable, CCG, recompilation	installed state runs its program; composition through predicted intermediates; a held-out program assembled and installed	Section 3.4
readable, writable, consumable	decodable; installable at the site; executed by the untouched downstream model	Section 3.5
F, C, ρ	fidelity; reference fidelity; $(F - B)/(C - B)$, for a specified baseline B	Section 3.3, E.1
ρ_{fc}	(F − N)/(C − N), floor-corrected	Section 3.3, E.1
η_{return}, H	produced-return efficiency; composition headroom	E.1
argument space $\mathcal{V}$, return interface Φ_A, argument cast Ψ_B	the interface for values passed between operations	§6
neural calling convention	the producer's return interface, the consumer's cast and write interface, frozen together	§6, D.2
stated vs produced value	a numeral written in a prompt vs a result an operation computes	D.2, E.2
RecordRef	the typed record reference passed from SELECT to LOAD	§5.1, C.3
ARITH	the sum/difference operation; ADD in artifact names	§6, D.1
D_TEST, TARGET_CAL, A_CAL	evaluation pool; target calibration pool; linker calibration pool	D
S0–S8	stages of the natural ladder: admission, reference port, SELECT, LOAD, handoff, recompilation, reuse, joint execution, headroom	C

B CONTROLLED EXPERIMENTS

The controlled experiments use transformers that Tracr compiles from RASP programs, so we know in advance what each one computes. Their role is to test the discovery procedure against ground truth, not to make claims about learned models. Every organism below runs a planted program, and nothing in this appendix says that a compiled network "learned" D_4. Each subsection follows one order: design, what was frozen and when, instruments, arms and controls, then result and threats.

We distinguish two levels of evidence throughout. At **Level I** the state interface is supplied or supervised and only the operation labels are hidden. This covers the partition (B.3), the 64/64 composition law (B.4), the twelve prospective products (B.5) and operator-isolated composition (B.6). At **Level II** both the state and the operation labels are hidden (B.8). The mediation test of B.7 runs at both levels. The controlled ladder is now frozen apart from provenance maintenance.

B.1 THE SEQUENCE-REARRANGEMENT FAMILY

Design. Each program produces output position i by computing a source index j = f(i, ℓ) and copying the token found there, $y_i = x_j$. The operation under study is the index map (i, ℓ) $\mapsto$ j, a state transition that a downstream aggregate consumes, rather than the string-level behavior. For reversal on [A, B, C, D, E] the source-index trajectory is [4, 3, 2, 1, 0] and the output is [E, D, C, B, A]. The family works as a unit test for operation identity, because transformation and implementation can be varied independently.

The six routing rules.

Table 7: Sequence-rearrangement routing rules.

rule	**map f(i, ℓ)**
reversal	$\ell -$ i $-$ 1
shift-by-one	(i + 1) mod ℓ
shift-by-two	(i + 2) mod ℓ
shift-by-three	(i + 3) mod ℓ
doubling	2i mod ℓ
reflection	($\ell -$ i) mod ℓ

Two operand routes. The length operand ℓ is computed in two ways inside the same organism. Route **P** takes the width of an all-true selector. Route **Q** adds the count of same-token positions to the count of different-token positions, which partition the sequence and therefore agree. Route Q is deliberately artificial. It exists as an experimental control, giving two compiled computations that deliver the same abstract operand to the same subsequent map. Within a same-map organism the two occurrences share no computation-graph nodes and no attention heads (graph and head Jaccard both 0.0), yet their future signatures agree exactly.

Organisms and occurrences. Six organisms, each carrying two operator occurrences, give twelve occurrences in six ground-truth classes: three classes of three and three singletons.

Table 8: Compiled organisms and implementation routes.

organism	**occurrence via P**	**occurrence via Q**	**role**
paired-same	reversal	reversal	repeated reversal, two operand realizations
paired-diff	reversal	shift-by-one	third reversal and a singleton negative
holdout-same	shift-by-two	shift-by-two	repeated shift-by-two
holdout-diff	shift-by-two	doubling	third shift-by-two and a singleton negative
pair3-same	reflection	reflection	repeated reflection

Continued on next page

Table 8 (continued)

organism	occurrence via P	occurrence via Q	role
pair3-diff	reflection	shift-by-three	third reflection and a singleton negative

The different-map organisms supply the negatives for equivalence recovery. The first occurrence in each different-map organism compiles to essentially the same subexpression as the first occurrence in its same-map sibling.

B.2 THE D_4D4 FAMILY

Group. D_4 = {e, r, r^2, r^3, s, rs, r^2s, r^3s} with r a quarter rotation and s a reflection, presented by $r^4 = s^2$ = e and srs = r^{-1}. Its element-order multiset is [1, 2, 2, 2, 2, 2, 4, 4], and 24 of the 64 ordered pairs (37.5%) fail to commute. We chose D_4 over an affine family because affine maps are closed under composition, which makes continuous interpolation a strong trivial competitor. The composition law of D_4 is discrete and order-sensitive.

Level-I substrate. The program keeps an eight-state register for the square's orientation. Eight single-operator organisms are identity-padded to a common depth, and 64 product organisms prod_X__Y apply two operators in sequence. Those with identity in one slot, prod_X__e and prod_e__X, carry a single operator in a known slot. All share a common 3-layer architecture with d_model = 55. We strip Tracr's construction-order identifiers so that the product organisms share one semantic residual basis. The raw-to-canonical map comes from compiler labels, not from learned outcomes, which fixes the gauge in advance. Temporal order comes from ground-truth factor metadata, not from file names.

Multi-stage substrates used for recursive execution and mediation (B.6–B.8). The three-stage substrate t1 runs ident3: (s, s, s) against prog_{h,k}: (s, s·h, s·h·k) over 64 programs, so each opcode appears eight times per internal stage and the joint state covers 8^3 = 512 triples. Its three state blocks are 8 dimensions each, d_model = 206. The BYPASS variant (B.7) adds a duplicate block: state2 occupies dimensions 170–177 and state2_bar 178–185, disjoint in every organism, giving d_model = 214. The blind-discovery substrates of B.8 have d_model $\approx$ 105 (v1) and 113 (v2).

B.3 DISCOVERING THE OPERATION VOCABULARY

Design. The twelve anonymous occurrences give C(12, 2) = 66 pairs. For each pair the shared-versus-separate test (§3.5) fits linear transition maps to the program state before and after each occurrence, once with a single map shared across both occurrences and once with a separate map per occurrence, and compares held-out fidelity. Two interface tiers are used: *supplied*, which reads the compiler's integer operand values directly, and *faithful*, which learns the interface under supervision against the same operand. This asks whether operation equivalence is recoverable given an appropriate state. Whether the state itself is recoverable without labels is the question of B.8.

Frozen before scoring. The matched-input guard of §3.5 is implemented as a functional-divergence gate D $\leq$ 0.05, added as an amendment after a faithful-tier false merge. It rejects a shared map that fits both occurrences by branching on occurrence identity rather than by computing one transformation.

Result. The recovered relation reproduces the six ground-truth classes exactly, at both tiers, with interface accuracy 1.0000 at every site and seed. Same-class pairs cost 0.0000. The smallest different-class cost is 0.3025 at the supplied interface and 0.2689 at the faithful one, so a wide empty margin decides the partition, not a threshold.

Component-overlap control. Classifying pairs by attention-head overlap instead:

Table 9: Operation identity versus attention-head overlap.

implementation	transformation	pairs	head overlap	overlap classifier
same	same	3	1.000	correct

Continued on next page

Table 9 (continued)

implementation	transformation	pairs	head overlap	overlap classifier
same	different	27	1.000	false merge
different	same	6	0.000	false separation
different	different	30	0.000	correct

The overlap classifier is wrong on 33 of the 66 pairs, and wrong in both directions. We draw a narrow conclusion from this. Functional transformation identity does not reduce to the component-overlap descriptors we tested. It does not follow that shared heads constitute a shared implementation.

B.4 Discovering the composition law

Level I. From the recovered operators, all 64 ordered products of D_4 are assigned to the correct class, with matched-class fidelity 1.0000, minimum runner-up margin 1.0000 and zero unresolved products. The readout was allowed to abstain, so closure was never forced. The recovered table exhibits closure, identity, inverses, r^4 = e, s^2 = e, non-commutation and the dihedral relations.

Level II, identity discovered rather than supplied. From the anonymous operation bank of B.8 alone, closure is 1.0000 on a set of order eight, the element-order multiset is [1, 2, 2, 2, 2, 2, 4, 4], and exhaustive structural identification returns D_4 rather than the quaternion group Q_8. The no-change operation is recovered as the identity permutation without ever being labelled. A random action bank reaches closure 15/64 = 0.234, derived analytically and checked by sampling. After the prediction hash, unblinding finds the right regular representation of D_4 up to conjugacy, with exactly eight conjugators solving it, which is the gauge freedom we expected. Nothing was relabelled after the fact.

B.5 Algebra completion

Sealing protocol. Twelve of the 64 product organisms are withheld, and the procedure runs in a fixed order. We compile only the 52 observed organisms, confirm that the twelve targets are absent, infer the twelve products from the recovered algebra, then hash and freeze the predictions. Only after that do we compile the held-out organisms.

Result. All twelve held-out products are correct, including both outcomes of a non-commuting pair: r then s gives rs, while s then r gives r^3s. On one axis this is stronger than B.6, because the target organisms did not exist when the predictions were frozen.

When does it decline? Subsampling the observed products shows what the prediction depends on:

Table 10: Algebra-completion precision and coverage.

fraction of products observed	coverage	precision
0.95	1.000	1.000
0.60	1.000	1.000
0.50	0.994	1.000
0.30	0.909	1.000
0.20	0.509	1.000
0.10	0.030	1.000

Precision stays at 1.000 wherever a prediction is made, while coverage collapses as the algebraic support for a product disappears. The procedure abstains instead of guessing, and what limits it is structural support, not model scale.

B.6 Operator-isolated causal composition generalization

Why operator-isolated. An earlier composition test learned the operators and the state interface jointly, and it failed for a reason that had nothing to do with composition. The state interface itself did not generalize to held-out compositions: fidelity was about 0.95 on observed compositions against

about 0.28 on held-out ones, with held-out state ceilings as low as 0.06. Even perfect downstream dynamics could not have succeeded there. We treat that result as uninterpretable, not negative, and we split the test in two. Operator-isolated CCG validates the state and withholds only the composition. Strict prospective CCG scores against a held-out state ceiling.

Design. On the Level-I substrate, each operator $\hat{O}_x$ is fit only from its single-operation organism prod_X__e, so no composition enters the fitting data at all. Before any composition target is opened, we check the same frozen operator where it recurs in the opposite program slot, prod_e__X, and it reproduces the state transition on 8/8 states. Ten compositions are then held out. For each one we predict the intermediate $\hat{z}_1 = \hat{O}_a(z)$ and the final $\hat{z}_2 = \hat{O}_b(\hat{z}_1)$ for all eight starting states, and seal them by hash before reading any held-out trajectory.

Result. The predicted intermediate is correct on 8/8 states for every composition, and so is the predicted final state. Installing the predicted final state in the residual stream and letting the frozen network run produces the intended behavior on 8/8 states for every composition, 80 cases out of 80. Installing the model's own target state through the same path reaches 1.0000, so the interface can express the target fully and these rates are efficacies against a ceiling of one.

Table 11: Controls for operator-isolated composition.

control	result
wrong operator substituted	0/10
order reversed, non-commuting pairs	0/6
order reversed, commuting pairs	4/4

The two order-reversal rows should not be averaged. Commuting pairs are required to succeed, because reversing them leaves the group product unchanged.

Reuse by native downstream computation. On the three-stage substrate t1, an intermediate installed at the layer-2 cut through the qualified actuator is consumed correctly by every one of the eight subsequent operations: the complete 8×8 causal action table is 64/64 cells at accuracy 1.0000 over 4,200 non-vacuous trials (per-cell n at least 62, median 66). Because one written state supports eight different subsequent outcomes, the intervention cannot be encoding a final answer. A leak guard confirms that for a fixed first operation the layer-2 residual is bit-identical across all eight downstream operations.

B.7 Ruling out bypass

The falsifier. Every positive recursive result here is neural, so the negative control has to be neural too. A procedure could reject a simulated bypass and still call a real network mediated, if its intervention happened to reach an alternate physical route. We therefore compile BYPASS, a network behaviorally identical to MEDIATED in which the declared mediator state2 is present, readable and writable, but downstream state3 reads a duplicate state2_bar instead. The compiler is prevented from merging the duplicate, and all 65 organisms compile with no common-subexpression collapse.

Native equivalence. Before any intervention, the two networks are compared on 21,312 native-equivalence checks — all 64 programs, every valid data position, all three state blocks, all four consumer families, all eight register settings — with zero violations. With no learned interface at all, writing state2 directly produces the expected signature:

Table 12: Mediation under behaviorally matched routes.

world	state2 reads back	F2 follows write	final state moves
MEDIATED	1.0000	1.0000	1.0000
BYPASS	1.0000	1.0000	0.0000

Prospective adjudication rule, frozen before the fitted run. BYPASS receives its own fitted frame, eight dimensions wider. If the write itself fails (low G_{write}), the verdict is INCONCLUSIVE, because

a failed redirection could be a bad actuator. If the write succeeds and downstream is unmoved, BYPASS is cleanly rejected as mediated. If the write succeeds and downstream moves, mediation or write locality is in serious doubt. We may not call a negative result causal merely because a separately fitted frame failed to write.

Full fitted comparison, about 38,528 trials per arm over 3,584 conditions and 56 operation pairs, with statistic, thresholds, arms, eligibility and adjudication identical across worlds:

Table 13: Localized-write mediation results.

metric	MEDIATED	BYPASS
G_{write} through the network's own readback	0.9998	1.0000
A_{target}	1.0000	0.0000
A_{preserve}	0.0000	1.0000
Ψ: correct-minus-random target rate	+0.9991	-0.0022
causal action table	64/64	0/64

Why the write must be localized. A dense write into the whole predictive subspace also reaches $A_{\text{target}} = 1.0000$ in BYPASS, a false mediation positive, even though the network's own readback of that write is imperfect (0.8763 in MEDIATED, 0.8098 in BYPASS). A write restricted to the mediator's support reaches readback 0.9998 and correctly scores BYPASS at 0/64. The lesson generalizes. A predictive description, a causal write interface, and the mediator the native circuit actually uses can be three different objects, and a dense intervention can reach a bypass route.

Removing compiler labels from the localization. The mediator's support is then found blindly, from interchange behavior alone. The finder sees target-versus-preserve roles but no residual labels, no block names, no final outcome and no BYPASS verdict, and it returns the smallest support that changes the target readout while preserving the other. On a correlated-decoy certification it recovers exactly the eight coordinates 170–177 and none from the duplicate. With the roles swapped it recovers state1 instead, again 8/8. The blindly localized write reproduces the compiler-localized verdict exactly (MEDIATED 64/64, BYPASS 0/64), so the headline discrimination does not depend on compiler metadata for where the write goes.

Matched-norm control. An off-direction write of the same norm leaves native behavior unchanged on 99.88% of trials (MEDIATED substrate, 828 trials per arm), so the mediator subspace is not a generic high-energy lever.

B.8 Blind discovery without compiler labels

State interface from future behavior. Consumers are compiled continuations that fork after the state, so no consumer token precedes the interface. We certify the interface before using it. Behavioral identifiability gives eight distinguishable states, with signature agreement 1.0000 across the two implementation families. A one-dimensional projection wins the description-length comparison on 3/4 boundary folds. Learning the interface on seven operations and testing on a held-out eighth costs at most 0.0007 in prediction. The causal actuator is then qualified prospectively on four fresh folds.

Operation-selector certification. Before the selector touched the real substrate, it was certified against nine synthetic truth families — planted group actions, planted non-group actions, several structured alternatives, an arbitrary 64-cell table and a null — passing 8/8 validity gates and 10/10 required planted-world verdicts.

Transfer to a held-out implementation. A third implementation, v3, was compiled only after the operation vocabulary had been frozen on v1 and v2, with its numerical tolerance fixed before v3 existed. The transferred vocabulary describes v3 in 1,130,366 bits, within 131 bits of a vocabulary refit freely on v3 itself. A free 64-cell table needs 1,245,714 bits, 115,348 more, while an additive model needs 25.9 million and the best of 64 scrambled vocabularies 25.7 million. The 131-bit gap is the cost of relabelling, and the verdict holds across five tolerances spanning eleven orders of magnitude.

Composition law, blind. From the frozen anonymous bank, the recovered vocabulary forms a D_4 action up to gauge (B.4, Level II).

Reuse across stages. In a two-stage network, one shared operation inventory describes both stages in 521.7 bits, against 628.8 bits for separate inventories and 1,958.3 for a joint table. The 107.1-bit margin matches the theoretical saving from reusing eight permutations across slots. In the three-stage network the margin is 122.4 bits against a predicted 122.39. The sealed verdict string for the two-stage run reads WITHIN_NETWORK_REUSE. A more accurate description is cross-stage sharing of one operation inventory.

Scope. All three implementations compile the same planted state machine, so transfer here is across compiled realizations of one program, not across independently trained models. What this earns is blind discovery of the state and its interface, a blind operation inventory and algebra, blind localization of the mediator's support, and native downstream consumption, all validated against a compiled bypass network matched in behavior. An earlier blind attempt on a different substrate, blind_h1, returned a sealed FALSE, and F records it.

B.9 THE RASP PROGRAMS

The controlled arm needs organisms whose ground truth is known by construction rather than inferred. Each is a RASP program compiled to transformer weights with Tracr (Weiss et al., 2021; Lindner et al., 2023), so "same transformation" and "same implementation" are properties we set rather than properties we measure. All programs live in `tracr_export/organisms.py`, the descriptions below specify their routing rules and compiled structure.

A flag token at position 0 selects which branch runs, mirroring the language-model arm where the prompt selects the task. The question is whether one network reuses a transformation, so the ground truth has to be about one network.

Two implementations of `length`. The design turns on separating what is computed from how, so `length` is computed two ways that return the identical value through different circuitry.

`v1` is one all-true attention pattern. `v2` is two token-comparison patterns plus an MLP addition, and is categorical throughout, which Tracr's non-linear `SequenceMap` requires. This is what makes "different implementation, same transformation" a constructed fact rather than a claim.

The implementation $\times$ transformation 2$\times$2. Three organisms populate the cells whose answers are known in advance. Same implementation and same transformation must merge, and one `length` node feeds three branches:

Different implementation with the same transformation must still merge. This is the known-answer version of the paper's headline claim, so the two implementations are distinct circuitry rather than duplicated graph nodes:

Different implementation with a different transformation must separate. The controls keep compiled width comparable so that `d_model` cannot act as a classifier cue:

The matched-pair organisms. The six organisms behind the 66-pair table share one skeleton. Two occurrences sit on disjoint length implementations, and both outputs are position indices consumed by an `Aggregate`, so both are genuinely read downstream and an intervention on either is observable.

Within a pair the two organisms differ only in the lambda applied at occurrence 2. Operand structure, compiled depth and downstream gather are identical, so anything that separates them is the transformation and nothing else.

Table 14: Controlled program specifications.

organism	occurrence 1	occurrence 2	ground truth
psm `paired_same_map`	`l - i - 1`	`l - i - 1`	**merge**
pdm `paired_diff_map`	`l - i - 1`	`(i + 1) mod l`	**separate**
hsm `holdout_same_map`	`(i + 2) mod l`	`(i + 2) mod l`	**merge**
hdm `holdout_diff_map`	`(i + 2) mod l`	`(2i) mod l`	**separate**

Continued on next page

Table 14 (continued)

organism	occurrence 1	occurrence 2	ground truth
p3s pair3_same_map	`(l - i) mod l`	`(l - i) mod l`	**merge**
p3d pair3_diff_map	`(l - i) mod l`	`(i + 3) mod l`	**separate**

The three pairs are map-disjoint by design. Reverse and shift-by-one are spent on the calibration pair, so the holdout uses shift-by-two and double, and the third pair uses reflection-through-zero and shift-by-three. A gate threshold calibrated on one pair is therefore not tested on the maps that defined it. At `l = 5` on tokens `a b c d e`:

Table 15: Routing-map properties.

map	image	permutation at every `l` in range?
`l - i - 1` reverse	`e d c b a`	yes
`(i + 1) mod l` shift-1	`b c d e a`	yes
`(i + 2) mod l` shift-2	`c d e a b`	yes
`(2i) mod l` double	`a c e b d`	**no, collides for even** `l`
`(l - i) mod l` reflect-0	`a e d c b`	yes
`(i + 3) mod l` shift-3	`d e a b c`	yes

We record rather than hide the fact that `(2i) mod l` is a permutation only for odd `l` and collides at even `l`. That is a real difference from the calibration pair, and it makes the holdout control slightly easier to separate, so a threshold set on it is if anything conservative for the calibration pair. Reflection-through-zero `(l - i) mod l` is also genuinely distinct from reverse `l - i - 1`, since it reflects through position 0 rather than through the midpoint and fixes index 0 instead of exchanging the ends.

What these programs establish. They fix the ground truth for the identity question, whether two occurrences of a computation are the same, in a setting where the answer is known by construction. They are a calibration for the criterion and not evidence about language models. The compiled organisms have no pretraining, no distributional structure and a hand-built residual layout, so where the paper uses them it uses them to show that a criterion returns the constructed answer, and nothing further.

They also bound how the result may be worded. Within the same-map organisms, `occ1` and `occ2` have `graph_jaccard = 0.000` and `head_jaccard = 0.000` under the audited component descriptors while producing identical future signatures. Equality of attention-head footprints is not equality of the complete physical implementation, and inequality of head footprints is not proof of disjoint causal mechanisms, since broader causal footprints can overlap where head footprints do not. The load-bearing claim is therefore only that the same abstract transition recurs across distinct compiled realizations, which suffices to show that operation identity cannot be defined as identity of one circuit realization. The benchmark does not support the stronger statement that equivalent operations are implemented by wholly disjoint circuits.

C NATURAL EXPERIMENTS

This appendix documents the pretrained-model study of §5. The target decomposition is SELECT(M, s) → R followed by LOAD(M, R, f) → V, where M is the record table in the recipient prompt, s a selector, R a typed record reference, f a field and V the value. We use that decomposition only to evaluate what the blind and causal procedures recover. The model is Qwen2.5-14B unless stated otherwise, and C.7 collects the cross-model replication on Gemma and Llama-3.1-8B. Two payload families run throughout: a numerical family (posload) and a categorical family (recsel). Each subsection follows the same design, frozen, instruments, arms and result order as Appendix B.

An earlier register-editing study, a 3×4 blind factorization of program-coordinate edits, motivated this branch and remains valid as supporting evidence. It is not the §5 result, though. Program-coordinate edits are not semantic task operations, and the SELECT → LOAD study below replaces it as the centerpiece.

C.1 TASK FAMILY AND BEHAVIORAL ADMISSION

Design. Each prompt shows a table of four records, so that SELECT must produce an item-specific reference rather than a constant. The selector s ∈ {FIRST, LAST} is an instruction that parameterizes a map from the current table to a record reference, not a fixed answer vector. The numerical family uses the fields fuel, mass, range and score with two-digit values, and the categorical family uses city, job, language and car. Selector × field gives the eight programs of §5.1. The primary rendering differs across programs in exactly two explicit control slots, the selector word and the field word.

Answer disambiguation. For a target record R_t, a wrong record R_w, the target field f_t and a wrong field f_w, the four first-token outputs V(R_t, f_t), V(R_w, f_t), V(R_t, f_w) and V(R_w, f_w) are required to be distinct. Every scored counterfactual is therefore read out on a 2×2 semantic table (target, wrong field, wrong reference, both wrong) instead of as right or wrong. First-token IDs come from the live tokenizer of each model, and the code asserts that they are distinct.

Frozen. Behavioral admission (S0) was run before any neural fitting, with a bar of 0.80 per program form (SELECT alone, LOAD alone, composed). After S0 the task family was frozen and no further prompt rescue was allowed.

Result. A first extremum task (select the MAX or MIN record) failed admission and was closed without any interface search: SELECT 0.7662 and composed 0.4805, against LOAD 1.0000 on the same tables, layout and readout (n = 77 per form). The positional family replaced it and passed on Qwen:

Table 16: Behavioral admission of retrieval task forms.

form	extremum (closed)	positional (used)
SELECT	0.7662	0.9740
LOAD	1.0000	1.0000
composed	0.4805	1.0000

Per-cell admission of the primary rendering, rerun before the blind study of C.2, has a worst cell of 0.9800 on the numerical family and 0.9600 on the categorical one. All sixteen cells sit at or above 0.96.

The alternate rendering that failed admission. To test whether the blind factorization of C.2 survives a change of surface, we froze a second rendering before collecting any data: four syntactic frames per family, assigned so that frame identity is a pure selector × field interaction (FIRST sends f_1→A, f_2→B, f_3→C, f_4→D, and LAST sends f_1→D, f_2→C, f_3→B, f_4→A), with token counts audited in advance. The preregistration stated that any alternate cell below 0.80 makes the surface test NOT_ADJUDICATED, with no replacement, synonym or rephrase allowed. The alternate rendering failed. The numerical worst cell was 0.28, at FIRST × score, and six of the eight categorical cells fell below 0.80, with both car cells at 0.00. We collected no alternate neural states. What the failure shows is that field-identifying lexical cues carry real weight in the model's competence. It is not evidence against an internal factorization. C.6 sets out what it costs the claim.

C.2 BLIND 2×42 BY 4 RECOVERY

Design. Eight anonymous state populations, one per program, are presented without labels. We ask which of the 8!/(2!·4!) = 840 anonymous 2×4 assignments best explains the neural geometry. The enumeration is exhaustive, 35 block partitions times 24 alignments, so no optimizer caveat applies. The procedure is told the grid shape but not which states share a row or a column. For an assignment π we fit the cell means with an additive model z(i, j) $\approx \mu + a_i + b_j$, and the interaction share

$$I(\pi) = \frac{\sum_{i,j} \|z_{ij} - \mu - a_i - b_j\|^2}{\sum_{i,j} \|z_{ij} - \mu\|^2}. \tag{10}$$

measures what the additive model leaves unexplained. A 2×4 additive model has four nonconstant main-effect degrees of freedom (five fitted coefficients including the intercept) and three interaction degrees of freedom, so it fits more easily than the older 3×4 study did. The verdict therefore requires two things at once: a low interaction share, which is fit, and a margin Δ to the runner-up assignment, which is identifiability.

Frozen before real states were opened. The instrument was certified on synthetic worlds:

Table 17: Synthetic certification of blind factorization.

world	$I_{\min}$	**margin** Δ	**verdict**
planted 2×4, noise $\sigma \leq 2$	≤ 0.2919	≥ 0.0364	recovered
structureless	≥ 0.4128	≤ 0.0022	fails fit
planted 2×2×2	≈ 0.0049	≤ 0.0001	fails identifiability

The 2×2×2 world is an identifiability control. It fits an additive 2×4 almost perfectly under several assignments, so a low interaction share on its own would have wrongly accepted it. We calibrated the thresholds τ_I = 0.3523 and τ_Δ = 0.0182 on null separation rather than on the recovery limit, since the argmin stays correct up to $\sigma \approx 4$, where an interaction share of about 0.387 approaches the structureless floor. Certification also turned up a permutation-invariance bug in the relabelling code, which we fixed before touching any neural data. After the fix the verdict is invariant across five opaque relabellings. Two verdict tiers were frozen: BLIND_2X4_ASSIGNMENT_RECOVERED, and the strictly stronger ABSTRACT_PROGRAM_STRUCTURE_SUPPORTED, which also requires the alternate-surface test.

Instruments. Each family gets 100 discovery tables and 100 disjoint confirmation tables, all eight cells per table, one frozen few-shot prefix, opaque cell IDs, and a truth map sealed separately so that only the adjudicator could load it. We chose the claim-bearing layer, L34, prospectively, as the pre-LOAD boundary of C.4, where the answer is not yet decodable. It is not inherited from the older study. We report a model-free surface baseline and an early layer, L4, alongside it. On confirmation we apply the discovery assignment without re-searching, and we also run an independent 840-way re-search.

Result. All values below were computed and committed before unblinding.

Table 18: Blind factorization across layers and task families.

family	**layer**	$I_{\min}$	Δ	**CONF, fixed assignment**	**CONF re-search**
numerical	surface	0.0000	0.1916	0.0000	same partition
numerical	L4	0.0012	0.0764	0.0012	same partition
numerical	**L34**	**0.0616**	**0.1285**	0.0619	same partition
categorical	surface	0.0000	0.1886	0.0000	same partition
categorical	L4	0.0042	0.1509	0.0042	same partition
categorical	**L34**	**0.0353**	**0.1298**	0.0364	same partition

The lower categorical interaction share reflects geometric fit only. It does not mean the categorical representation is more abstract.

Cell-identity-destroyed null. A null that preserves table and prompt regularities but destroys stable cell identity was run 1,000 times per family. Its floor lies below the structureless certification world, so it is the harder comparison. At L34 the observed optimum was below every null draw in both families, b = 0 of m = 1,000. The numerical null had a median of 0.1952 and a 5th percentile of 0.1415, the categorical one 0.1577 and 0.1138. This gives p_{MC} = (b + 1)/(m + 1) = 9.99 $\times 10^{-4}$. The running process printed 0.0000 using the older b/m estimator. Since m = 1,000, a printed zero implies b = 0, and we recorded the corrected value in a sidecar that the adjudicator was wired to read before unblinding. Partition stability over 1,000 bootstrap resamples of base tables was 1.0000 in both families.

Parallelogram completion. Holding out one cell and predicting it from the shared offset, $\hat{z}_{0j} = z_{1j} + (a_0 - a_1)$, tests extrapolation rather than full combinatorial reconstruction. Normalized mean squared error, with the grand mean scoring 1.000:

Table 19: Held-out parallelogram completion.

predictor	numerical mean (worst)	categorical mean (worst)
factorized, both factors	0.4586 (0.6972)	0.2636 (0.3133)
selector factor only	0.8452 (1.0385)	1.0876 (1.1758)
field factor only	1.3772 (1.8674)	0.7258 (0.8729)
grand mean / free-cell lookup	1.0000	1.0000

Every held-out cell beats the grand mean in both families. Which single factor helps reverses between the payloads, but the joint factorization beats both single factors in each. We computed the single-factor comparisons after the initial run, from cached states on CPU, and chose no threshold from the result.

Unblinding. With all artifacts committed, we opened the truth map, categorical first. The recovered assignment matches 8/8 semantic cells at the surface, at L4 and at L34, in both families. The verdict is BLIND_2X4_ASSIGNMENT_RECOVERED on both payloads, with ABSTRACT_PROGRAM_STRUCTURE_SUPPORTED not adjudicated (C.6). We ran the blind 2×4 study on Qwen only.

C.3 The typed reference port

Design. The reference interface was qualified consumer-first, before any SELECT map was fitted. A recipient prompt asks for LOAD(M, R_a, f). We install the model's own reference state for another valid record R_b at the reference (name) position, and the intended outcome is V(R_b, f): a different record, the same field. Running this before any SELECT fitting fixed a prospective constraint. SELECT would have to deliver its reference by the port's deadline, or we would close the chain rather than rescue it.

Arms. native wrong-reference (the test), a norm-matched token-embedding difference ("token-bus"), a cross-site write, a random write of matched norm, and a no-op.

Result (Qwen, numerical).

Table 20: Recipient locality and reference controls.

layers	native → wrong reference	token-bus	cross-site	random
L4–L24	1.0000	0.0000	0.0000	0.0000
L28	0.8909	0.0000	0.0000	0.0000
L31–L40	0.0000	0.0000	0.0000	0.0000

At L28 every success lands on the wrong-reference, right-field cell of the 2×2 table, and none of them moves the field. The port is live from L4 through L28 and dead by L31 (verdict LATENT_

REFERENCE_PORT_ADMISSIBLE). The token-bus arm is zero at every layer, including L4, so the flat early window is not token substitution. The categorical family replicates the window at 1.000 through L24, 0.652 at L28 and 0.000 from L31, with every control at 0.

Recipient locality. The stricter test pairs a donor and a recipient that contain the same record identity with different values. If the transplanted state were an answer, the recipient would return the donor's value. If it is a reference, the recipient reads its own table. On Qwen, in both payload families, the recipient returns its own value for the referenced record with 1.0000 accuracy and the donor's value with 0.0000 (REFERENCE_CONFIRMED_RECIPIENT_LOCAL). The same verdict holds on Gemma and on Llama, whose best cell gives 0.9500 for its own value and 0.0000 for the donor's.

carried_field = 0. In every reference-transfer experiment, S6 in C.5 and M2 in C.8, the source field is never carried along with the reference: 0/60 at every layer on Qwen numerical, 0.0000 at every layer on Qwen categorical, and 0.0000 at the reported layer on Gemma and Llama.

Ordering. The selector (opcode) is settled early, the reference remains writable through L28, and the value first becomes decodable after L34. We report this ordering without claiming exact micro-causal boundaries.

C.4 SELECT AND LOAD RECOVERY

SELECT. $\hat{O}_{\mathrm{SELECT}}$ maps the terminus state of a selector-only prompt at layer L_sel to a displacement ΔR written at the reference position through the port of C.3. It is state-dependent in M and parameterized by s, and it is fitted by reduced-rank ridge regression on source prompts. For the Qwen categorical map, ranks 32 and 128 were indistinguishable on the 60 fitting rows, so the tie broke low. That episode is what led us to adopt the rule that rank is chosen by causal efficacy (E.3).

Table 21: SELECT recovery and controls.

model / family	cell	fit rows	M_ SELECT	FIRST / LAST	per-mode mean	shuffled	random
Qwen numerical	L24, r128	140	1.0000	1.000 / 1.000	0.4444	0.1111	0.0000
Qwen categorical	L20, r32	60	0.8462	0.692 / 1.000	0.3462	0.0000	0.0000
Gemma categorical	L24, r128	136	0.8793	0.897 / 0.862	0.4138	0.2069	0.0345
Llama categorical	L16, r32	—	0.7083	—	—	—	—

M_SELECT is the rate at which the recipient returns the value of the record its own selector should pick. The per-mode mean arm installs the average displacement for that selector, a state-independent competitor, and it is the strongest null. All four cells pass (SELECT_RECOVERED_ AND_CAUSAL). On Qwen categorical the causal band L16–L28 coincides with the live window of the reference port. Efficacy is not a geometric statistic: the Qwen categorical map reaches 0.8462 with a cosine of only +0.489 to the true displacement.

LOAD. $\hat{O}_{\mathrm{LOAD}}$ maps the terminus state at a pre-value boundary L_pre to the terminus state at a post-value boundary L_post and replaces it there. It is fitted on LOAD-only source prompts. The preregistered gate is M_LOAD ≥ 0.50 and M_LOAD $-$ max(nulls) ≥ 0.30, where the nulls are a fixed-vector continuation (M_FV), a per-cell mean, a shuffled-state map and a generic-select map.

The first numerical LOAD run placed L_pre at L38, where the answer was already partly decodable, around 0.29. It reached M_LOAD 1.0000 but M_FV 0.9625, missing the separation clause (LOAD_ LAW_NOT_SEPARATED_FROM_NULLS), and we keep that verdict. Its sealed successor, S3b, fixed the boundary rule in advance on a fresh pool: L_pre is the deepest layer at which the answer is decodable below 0.05. On Qwen every threshold between 0 and 0.29 selects the same layer, L34. A second rule, S3c, takes the deepest layer with exactly zero correct target decodes on calibration data. We preregistered it after the Gemma S3b result (C.7), sealed it before opening its own evaluation pool, and evaluated it on fresh data.

Table 22: LOAD recovery and controls.

model / family	rule	L_pre → L_post	M_LOAD	strongest null	gap	verdict
Qwen numerical	S3b	L34 → L42	1.0000	0.1437 (FV)	+0.8562	pass
Qwen categorical	S3b	L34 → L42	0.8562	0.1313 (shuffle)	+0.7250	pass
Qwen categorical	S3c	L34 → L42	0.8812	0.1313 (shuffle)	+0.7500	pass
Gemma categorical	S3b	L34 → L40	0.8562	0.7250 (cell mean)	+0.1312	**fail**
Gemma categorical	S3c	L31 → L40	0.9375	0.0688 (shuffle)	+0.8688	pass
Llama categorical	S3b	L19 → L31	—	—	+0.8938	pass
Llama categorical	S3c	L16 → L31	—	—	+0.8562	pass

Rank matters. On Qwen numerical the rank sweep gives decode 0.776 at rank 8, 0.948 at rank 32 and 0.957 at rank 128. On Qwen categorical, rank 32 decodes only about 0.164 against about 0.767 at rank 128.

Decodability trajectories (fraction of items whose target answer is decodable at the terminus):

Table 23: Answer decodability at candidate boundaries.

layer	Qwen categorical	Gemma categorical
L31	0.000	0.000
L34	0.000	0.048
L36	0.210	0.449
L38	0.147	0.724
L40	0.824	0.915
L42	0.919	1.000

Llama forms the answer more gradually: about 0.008 at L19, 0.137 at L21, 0.363 at L23 and 0.668 at L24. The L_pre → L_post transition should not be equated with the whole of semantic LOAD(M, R, f) → V, and a decodability of zero at L_pre is an operational boundary rather than an ontological one.

C.5 Composition, recompilation and reuse

S4b, predicted-reference handoff. $\hat{O}_{\mathrm{SELECT}}$'s predicted reference is written into the port and the model's own LOAD runs natively. This tests whether a recovered operation's output is an executable argument. It is not yet composition of two recovered operations. On Qwen numerical, with a fixed demonstration context, the handoff reaches 0.9917 (FIRST 0.983, LAST 1.000) against a native oracle of 1.0000 and a per-mode-mean null of 0.4167. With each recipient's own demonstrations the handoff falls to 0.5263. The SELECT predictor is context-anchored, and we do not claim a context-robust handoff. In the historical Qwen categorical run, the inherited rank-32 map fails, reaching 0.0513 against a native oracle of 1.0000 (CCG_UNSUPPORTED). A diagnostic sweep, which did not change the verdict, showed that the failure is a capacity artifact rather than an intrinsic property of categorical SELECT: with 80 tables, rank 32 gives 0.1333 and rank 128 gives 0.9833. The dependence on sample size is non-monotonic and remains unresolved. Gemma and Llama pass, Llama at 0.8654 against a best null of 0.3974.

S5, opcode recompilation. The donor's FIRST↔LAST displacement is installed at the recipient's selector-token site. Success requires the recipient to execute the new selector on its own table.

Table 24: Opcode recompilation across layers.

layer	Qwen numerical	Qwen categorical
L4	1.0000	1.0000
L8	1.0000	—
L12	0.9016	1.0000
L16	0.3279	0.9500
L20	0.4262	0.9833
L24	0.3115	0.9667
L28	0.0164	0.1500

Transporting the donor's answer and reversing the selector both score 0 at every layer, over 61 numerical pairs. Best-layer rates on the categorical family are 1.0000 at L4 for Qwen, 0.9508 at L4 for Gemma and 0.6000 at L3 for Llama, and all three receive OPCODE_RECOMPILED_RECIPIENT_LOCAL. The rates are not normalized across models.

S6, cross-field reference reuse. With no learned map, a reference state read from a query about field f is installed into a query about field f′. Success means returning V(R, f′).

Table 25: Cross-field reference reuse.

model / family	best layer	reused	carried field	same-field	shuffled	random
Qwen numerical	L4–L24	1.0000	0.0000	1.0000	0.10–0.32	0.0000
Qwen categorical	L4	0.8431	0.0000	0.8431	0.0980	0.0000
Gemma categorical	L4	1.0000	0.0000	1.0000	0.0833	0.2000
Llama categorical	L8	0.8333	0.0000	0.8125	0.2292	0.0000

Cross-field reuse matches same-field reuse, and the source field is never carried: the port holds a reference, and the field remains under recipient control.

S7 and S8, two recovered operations. The composed object is $\hat{O}_{\mathrm{LOAD}} \circ T_{24:34} \circ W_R \circ \hat{O}_{\mathrm{SELECT}}$. The recovered SELECT writes a predicted reference, the network's untouched layers carry it to L34, and the recovered LOAD replaces the post-value state. The native target reference is never supplied. The first run, S7, used a frozen absolute gate: composed minus fixed-vector had to reach +0.15. On Qwen numerical the composition reached 1.0000 against a fixed-vector continuation of 0.9550, so the gate was unattainable, the continuation already sitting within 0.045 of the ceiling. We report S7 as CCG2_NOT_SEPARATED_FROM_NULLS in both families. Its diagnosis motivated S8: semantic LOAD begins before L34, so once SELECT is corrected a generic continuation often suffices.

The historical S8 run is a separate preregistration on an untouched pool that measures the fraction of available causal headroom:

$$H = \frac{F(\mathrm{CCG2}) - F(\mathrm{FV})}{F(\mathrm{native}) - F(\mathrm{FV})}.$$

unclamped. A pass requires H $\geq$ 0.50 together with a lower 95% bound on F(CCG2) $-$ F(FV) above zero, from a paired cluster bootstrap over base tables with 10,000 draws, and an exact McNemar test as a secondary. A non-positive denominator returns HEADROOM_DEGENERATE. The historical name H equals the main textś ρ_{CCG}. It is the fraction of the available behavioral headroom, not the fraction of the true LOAD mechanism. On a synthetic harness the procedure passes an S7-shaped effect and fails both identical arms and a one-item difference.

Table 26: Historical S8 composition results; the qualified categorical successor is reported below.

model / family	native	Select only	LOAD only	CCG2	FV	H	LCB_{95}	McN. b / c	verdict
Qwen numerical	1.0000	0.9900	1.0000	0.9900	0.9450	**0.8182**	+0.0150	10 / 1	pass
Qwen categorical	0.9608	0.7353	0.9608	0.7255	0.5588	**0.4146**	+0.0784	20 / 3	not adjudicated
Gemma categorical	1.0000	0.9450	0.9850	0.8900	0.1750	**0.8667**	+0.6500	143 / 0	pass
Llama categorical	0.9510	0.8235	0.8529	0.5980	0.0000	**0.6289**	+0.4902	—	pass

Qwen numerical uses 200 rows over 100 base-table clusters, and its two SELECT-stage misses are exactly its two chain misses. The historical Qwen categorical run uses 102 rows over 51 clusters. Its composition contributes significantly above the continuation, with LCB_{95} +0.0784 and McNemar p = 0.00049, but it recovers less than half of the headroom, so the preregistered verdict is CONTRIBUTION_NOT_ADJUDICATED. That is not the same as no contribution. The two scales also disagree. In absolute terms the categorical gain of +0.1667 is larger than the numerical +0.0450, while in headroom terms the order flips.

Factorizing the headroom. H decomposes as $H_{\text{upstream}} \times H_{\text{LOAD|SELECT}}$, where H_{upstream} measures how much of the headroom the predicted reference already delivers and $H_{\text{LOAD|SELECT}}$ measures what the recovered LOAD does with it.

Table 27: Historical S8 headroom decomposition.

	Qwen numerical	Qwen categorical	Gemma	Llama
H_{upstream}	0.8182	0.4390	0.9333	0.8660
$H_{\text{LOAD|SELECT}}$	1.0000	0.9444	0.9286	0.7262

The historical Qwen categorical shortfall is upstream. The recovered LOAD executes what it receives. $H_{\text{LOAD|SELECT}}$ rests on the few rows where the SELECT stage leaves headroom, about 9 of 200 for Qwen numerical and about 61 of 84 for Llama, so it should be read with those denominators and not used to rank models.

Historical Qwen categorical continuation. What makes this cell unusual is a strong state-independent continuation: FV 0.5588, against 0.1750 on Gemma and 0.0000 on Llama. Two parity controls leave it in place. A native Hugging Face backend reproduces the verdict, with H 0.4103 and FV 0.5784. A fresh zero-decode boundary recalibration selects the same L34 $\rightarrow$ L42 transition, and by construction the S8 fit is then identical, so this is a parity check rather than an independent replication. A third comparison, which arose by accident, bears on the weak upstream factor: on the same 13-name roster and the same 156 scored rows, Llama's handoff reaches 0.8654 against Qwen's 0.0513, so alignment losses do not explain it. These diagnostics concern the historical instrument; the qualified successor reported below establishes recoverable composition without resolving the origin of the continuation.

Across the three categorical models, the handoff tracks the cosine between the predicted and the true reference (0.5414 for Qwen, 0.7269 for Gemma, 0.7947 for Llama) more closely than it tracks source SELECT accuracy, where Llama's 0.7083 sits below Qwen's 0.8462. This is an association over three points, and we do not claim that cosine is the mechanism.

Qualified Qwen categorical successor. The prospectively alignment-certified `natural_recsel_st13_v1` successor passes all four instrument preconditions and both frozen endpoints, with the composition threshold unchanged at $\rho \geq 0.50$. Alignment drops fall from 49 to 0, SELECT fitting rows increase from 110 to 194, and evaluation expands from 102 rows over 51 base-table clusters to 200 rows over 100 clusters. The frozen S2 selection rule chooses L28 instead of L20, so this is not a roster-only change. Native, fixed-vector and composed fidelities are respectively 1.0000,

0.6750 and 0.9150, giving

$$\rho_{\mathrm{CCG}} = \frac{0.9150 - 0.6750}{1.0000 - 0.6750} = 0.7385.$$

The SELECT-only rate is 0.9500, upstream efficacy is 0.8462, and the conditional LOAD factor is 0.8727. The stronger continuation leaves less headroom than in S8, ruling out a favorable-denominator explanation. Shuffled and random controls have target fidelity 0.0000, as do wrong-field and both-wrong outcomes. This successor establishes recoverable categorical composition under an adequately qualified instrument; it does not replace the historical S8 verdict. Its identifier is listed in Appendix G.2.

C.6 What the surface confound does and does not permit

The primary rendering varies two explicit lexical slots, one per factor. A mean-embedding surface baseline is therefore additive for arithmetic reasons alone, and it recovers the same 2×4 assignment with an interaction share of 0.0000 in both families (C.2). The blind geometric result cannot, on its own, establish a surface-invariant abstract program register.

The test that could have established it was the alternate rendering of C.1, in which surface frame and program cell are decoupled. It failed behavioral admission, and the preregistration forbade replacing it, so ABSTRACT_PROGRAM_STRUCTURE_SUPPORTED is NOT_ADJUDICATED on both payloads. One specific claim goes with it. We do not claim that the model independently invents a surface-invariant selector × field register.

What the blind stage does support is narrower. The neural states preserve a uniquely identifiable selector × field organization. It persists from the surface through L34, holds on tables never used in discovery, beats a cell-identity-destroyed null, and extrapolates to held-out cells. The blind stage therefore identifies which program states share the same two latent factors. The roles of those factors, which one produces a reference and which one parameterizes retrieval, come from the causal experiments of C.3–C.5, not from the geometry.

C.7 Cross-model replication

Protocol. Replication repeats the final scientific ladder, not the exploratory history. We separate three layers. Semantic protocol invariants are the task, the 2×2 readout, the arms and the gates. Measurement parameters are the shot count, the token IDs, the boundary rule and the rank. The physical interface is layers and subspaces. Only the first layer has to transfer. Qwen-specific artifacts do not carry over: the extremum task, Qwen token IDs, the L24/L34/L42 labels, the rank-32 inheritance, the permissive first LOAD rule and the S7 absolute gate. Every cross-task experiment of Appendix D is Qwen-only.

Instruments. Gemma required a custom Hugging Face adapter, which we qualified before use: site identity 0 across seven layers, null add and null replace 0, a decode path agreeing with the native argmax, and no leakage. Porting also exposed a scoring hazard. The categorical task file held Qwen first-token IDs, and one stage script lacked the assertion that would have caught Gemma being scored against them. The repair derives IDs from the live tokenizer, asserts distinctness and verifies the sealed 500-item pool, with 0 violations and 0 recipient drops on Gemma.

Stage matrix (categorical family).

Table 28: Categorical replication and successor status.

stage	Qwen2.5-14B	Gemma	Llama-3.1-8B
S0 behavioral admission	pass	pass	pass at 8-shot (3-shot fail stands)
S1 / S1b reference port, recipient locality	pass / pass, L4–L28	pass / pass, L4–L24	pass / pass, L3–L11
S2 SELECT	historical L20 r32; successor L28	pass, L24 r128	pass, L16 r32

Continued on next page

Table 28 (continued)

stage	Qwen2.5-14B	Gemma	Llama-3.1-8B
S3b / S3c LOAD	pass / pass	**fail** / pass	pass / pass
S4b predicted-reference handoff	historical fail; successor passes	pass	pass
S5 opcode recompilation	pass	pass	pass
S6 cross-field reference reuse	pass	pass	pass
S8 two-operation composition	historical $H = 0.4146$ (not adjudicated); successor $\rho = 0.7385$ (pass)	pass, H = 0.8667	pass, H = 0.6289

Qwen's numerical family, not in the matrix, passes every stage, including S4b (0.9917) and S8 (H = 0.8182). The matrix is not a model ranking. Historical S4b/S8 and successor outcomes are separate experiments; C.5 reports the successor instrument and endpoint.

Llama and the shot budget. Under Qwen's three-shot convention Llama passes SELECT at 0.8701 and LOAD at 0.9870 but fails composed prompts at 0.5714, with FIRST at 0.206. That result stays in the record. Its interpretation, that Llama cannot compose in one forward pass, is retracted. A prospectively frozen eight-shot successor passes with SELECT 0.9861, LOAD 1.0000 and composed 0.9583, and we then froze eight shots for the whole Llama ladder.

Gemma and the boundary rule. The S3b rule, D $<$ 0.05, selects L34 on Gemma, where D = 0.048 and a generic continuation already reproduces much of the donor behavior, giving a cell-mean null of 0.7250. The gap to the strongest null, +0.1312, misses the 0.30 bar, and we report the S3b failure as a failure. What moved was the null, not the state-dependent map: M_LOAD is 0.8562 under S3b. We preregistered S3c after this failure and sealed it before opening its own evaluation pool. On that fresh pool it selects L31, where the gap is +0.8688, and Gemma's L34 decodability on the same pool is about 0.004 rather than 0.048, so the 0.05 threshold had been sitting near noise. S3c is not uniformly better, though. On Llama, S3b gives the larger gap, +0.8938 against +0.8562. One erratum concerns the pool. The S3c preregistration stated that its pool was untouched, but the blind 2×4 study had reserved items 2000–2049, never executing them on Gemma. The result is not contaminated, but the statement was wrong, and we keep it as an erratum. A pool registry keyed on content hashes now enforces disjointness.

The porting lesson. Constants silently change meaning across model families. D $<$ 0.05 meant exact zero on Qwen and selected a layer of substantial generic continuation on Gemma. Three shots were adequate for Qwen and not for Llama. Rank 128 was necessary for Gemma's SELECT, 0.0862 at rank 32 against 0.8793 at rank 128, while Llama selected rank 32. Every such constant has to be re-qualified as a declared calibration before the claim-bearing pool is opened, frozen afterwards, and reported alongside the unmodified strict result. A shared typed operation does not imply a shared physical schedule either: LOAD runs L34 $\rightarrow$ L42 in Qwen and L31 $\rightarrow$ L40 in Gemma.

C.8 MIXED-PAYLOAD LOAD

Question. Is LOAD one payload-polymorphic operation, LOAD(M, R, f) $\rightarrow$ V for any field type, or does only its interface generalize? A single person-record task mixes categorical fields CITY and JOB with numerical fields AGE and HOURS (usual weekly work hours), four records per table, FIRST/LAST selectors. A planned EXPERIENCE field was retired before the run because it is structurally constrained by AGE. Qwen only.

Instrument repairs before the ladder. Three repairs came first. A numerical leading-space token collapsed onto a bare-space first token on all three tokenizers, so values are now written bare and the prompt absorbs the space. Oslo and Osaka shared a first token on Qwen and Llama, so Osaka was removed. Items whose AGE and HOURS values collide on the leading digit are rejected at generation. Certification then gave eight usable values per field, names restricted to the 13 that are single-token on both Qwen and Llama, and 0 quad and 0 name violations across all three tokenizers. A first pilot scored categorical LOAD at 0.0000 because of a readout-surface error. We treated that as an instrument failure, fixed it before any neural work, and do not interpret it.

Admission (M0). Mixed task: SELECT 0.9691, LOAD 1.0000, composed 1.0000, with categorical and numerical payloads both 1.0000 (MIXLOAD_M0_PASS). The categorical-only and numerical-only controls also pass.

Cross-type reference (M2). A reference read from a query about one type is consumed by a query about the other, over 60 pairs with no alignment losses. Reuse is 1.0000 at every layer from L4 to L24 in both directions, categorical to numerical and numerical to categorical, with carried field 0.0000, random 0.0000 and shuffled between 0.07 and 0.28. Verdict CROSS_TYPE_REFERENCE_CONFIRMED.

Shared LOAD fit (M3). One LOAD map fitted on all four fields executes all four at 1.0000, a gap of +0.8208 to the best null. Controls fitted on subsets locate its limits. A categorical-only fit executes CITY and JOB at 0.9958 and no numerical field at all. A numerical-only fit does the reverse. A CITY-only fit gives JOB 0.000 and a JOB-only fit gives CITY 0.000, while an AGE-only fit reaches HOURS 0.442 and a HOURS-only fit reaches AGE 0.717. A pooled perfect map does not show that one rule is being executed, so we preregistered holdouts.

Holdouts (M4, M5). Before the M4 result file existed we amended its gate from pooled to per-field transfer. We still report the pooled decision.

Table 29: Mixed-payload LOAD transfer across held-out fields.

fold	train fields	test fields	held-out rate	per field	verdict
M4a	CITY, AGE	JOB, HOURS	0.3333	JOB 0.0000, HOURS 0.6667	does not generalize
M4b	JOB, HOURS	CITY, AGE	0.3292	CITY 0.0000, AGE 0.6583	does not generalize
M5a	CITY, JOB	AGE, HOURS	0.0000	oracle 1.0000	no cat $\rightarrow$ num transfer
M5b	AGE, HOURS	CITY, JOB	0.0000	oracle 0.9958	no num $\rightarrow$ cat transfer

The preregistration had reserved a verdict for the case where M4 passes and neither M5 does. M4 failed on its categorical half, so that label does not apply and we report the outcome longhand. Numerical fields show partial reuse, in the 0.44 to 0.72 range. Categorical fields show none, and nothing crosses the type boundary.

Conclusion. The reference interface is shared across payload types. LOAD is not. The architecture the data support is SELECT(M, s) $\rightarrow$ R : RecordRef followed by LOAD_f(M, R) $\rightarrow$ V_f, a shared typed calling convention with specialized retrieval implementations. Consistent with this, the LOAD used in Appendix D is a numerical LOAD qualified on its own numerical source task. With only two value types tested we do not claim type independence in general, and the preregistered stop rule forbids treating the pooled map as a universal LOAD in any downstream experiment.

D CROSS-TASK LINKING

This appendix documents §6. Every experiment uses Qwen2.5-14B (base), frozen throughout, and the only things we train are small linear estimators on hidden states. Three operations are recovered on three separate source tasks, their interfaces are frozen, and the frozen objects are then evaluated on four target tasks that no operation or interface was fitted on. ARITH denotes the sum/difference source operation; the main text uses ADD for its addition mode. Rover routing also tests difference. In artifacts and hashes, ARITH appears as ADD and rover routing as Task C.

D.1 THE THREE OPERATIONS AND THEIR SOURCE TASKS

LOAD (numerical record retrieval). The source task is a named-record numerical retrieval task: two named records, each with four numerical fields, and a query asking for one field of one record. Behavioral admission is 1.0000 on all eight record $\times$ field cells. The retrieved value first becomes decodable between L31 and L36 (decodability 0.25 at L31, 0.62 at L34, 1.00 at L36), which fixes the LOAD transition L31 $\rightarrow$ L36. A state-dependent reduced-rank map between these boundaries (rank 32, 20 recipients, 3,280 rows) installs the queried value causally at 0.6805, against 0.0000 for a fixed-vector continuation, 0.0902 for a generic map fitted on another operation and 0.0659 for shuffled addresses (gap 0.5902, LCB_{95} 0.5454). Field interfaces learned on an earlier record-retrieval task do not transfer to this numerical task, where $\rho = 0$. As the typed LOAD of C.8 would predict, the numerical specialization has to be qualified on its own.

ARITH (sum and difference). The source task is a program grid of four operand-slot pairs $\times$ two opcodes {sum, difference}. The answer is computed and never appears in the prompt, so a positional copy cannot solve it. We redesigned the generator before any neural work to remove answer-statistic confounds. The gap between mean sum and mean difference fell from 52.40 to 11.05, and a residual leading-digit asymmetry of about 0.22 is recorded as unavoidable. Behavioral competence is 0.9917 pooled, with all eight cells at or above 0.80 against a chance rate of 0.125. The opcode is causally writable at L31, where native execution is 0.982. The two operands commit jointly at L36: a ceiling probe gives 0.00 and 0.02 at L31, 0.00 and 0.89 at L34, and 0.93 and 1.00 at L36. The code for the second operand is opcode-conditioned, since a target-bound code reaches ρ 1.071 against 0.071 for a recipient-bound one, on 4/4 shards. The recovered callable is therefore

ARITH(op) = [opcode write at L31, Δ first operand at L36, Δ second operand (op-conditioned) at L36],

which on a fresh prospective pool of 169 recipients reaches 0.6923 against a native ceiling of 0.6095 (ρ 1.136, LCB $\approx$ 0.62). An earlier schedule that wrote the second operand at L34 is incomplete (ρ 0.448) and is not used.

COMPARE (three-way quantity comparison). The source task asks whether one quantity is less than, equal to or greater than another, answered with the single tokens " less", " equal", " greater". Behavioral admission is 1.0000. The operand port is native and early. Installing the model's own operand state at L4, L8 or L12 reproduces the intended relation at 1.0000, and by L16 the port is dead, at 0.1833. A norm-matched token-embedding write scores 0.0000. We chose L8 by a frozen middle-admissible rule. All six directed relation changes execute at 1.0000 there.

Disjointness. Each target task is built on a surface disjoint from all three source tasks. Rover-routing uses place and rover vocabulary that appears in no other substrate. The quota task does not expose the ARITH or COMPARE source rendering. The sensor task was chosen to be distinct from every earlier source and target task. The only thing a target task shares with a source task is the latent operation.

D.2 NEURAL CALLING CONVENTIONS

The formal objects are those of §6: a producer's return interface Φ_A, an argument cast Ψ_B, and the consumer's write interface W. Each is fitted only on its own operation's source task.

Return maps. The LOAD return map reads the retrieved value from the LOAD output state. It is fitted while the whole record table is held byte-identical and only the query changes, with both query directions included so that any constant query-token component cancels. On held-out source items it

reaches R^2 0.8732 and exact two-digit recovery 0.8625. The ARITH return map reads the computed result from ARITH's output state and was frozen together with the COMPARE interface.

The COMPARE consumer cast. COMPARE's operand interface was fitted strictly on COMPARE source data: slot-specific read maps W_read, bind maps W_bind, slot scales and a fixed normalization at L8, on 150 calibration recipients. No LOAD or ARITH state was allowed into fitting, site selection, rank or capacity selection, or scale selection. The round trip read $\rightarrow$ bind was confirmed once on 100 fresh recipients: 0.9200 against a native 0.9900 ($\approx 93\%$ of native), with token-embedding and random writes at 0.0000 and shuffled 0.1300 (verdict CONSUMER_ABI_CONFIRMED).

A causal-mask signature. The cast reads the tens digit and ignores the units digit, exactly as the causal mask of a left-to-right comparison requires. For the first operand slot, R^2 against the tens digit is 0.987 at L8 and against the units digit 0.043, with 0.954 and 0.041 at L4 and 0.995 and 0.044 at L12. The model settles the comparison on the leading digit as soon as it is known.

The ARITH consumer cast. ARITH's operand interface was fitted on ARITH source data with the same form: per-slot read and bind maps, a fixed scale and normalization. The version used here reproduces the historical functional signatures, with tens dominating units by +0.84 in the first slot and a second-slot fit of $R^2 \approx 0.95$, and it is serialized with per-array hashes (G.3).

The frozen library. Before any target task was opened, the COMPARE read and bind maps, scales (22.8192 and 35.6880), normalization, capacity and the ARITH return map were frozen together (OPERATION_FREEZE_V1). The LOAD return map and the ARITH consumer cast were sealed under their own preregistrations (G.1). A cast fitted on one COMPARE rendering transfers verbatim to a second rendering, 0.89 against the 0.86 source edge, so the convention is rendering-invariant. It is not depth-invariant: 0.28 at L12, where native execution is 0.95. That is why the site is part of the frozen object.

Three objects, not one. A stated numeral in a prompt, the state an operation produces for its result, and the state a consumer expects as its operand are three different objects. Raw produced states are not consumable, while stated numerals are (D.4). The calling convention is what turns a produced result into a consumable operand.

Runtime. Linked execution is staged. For a target item, a target-local frontend reads the prompt's surface to decide which record to load and which quantities to pass on. It never reads the target answer, the intermediate values or the final relation. Each operation then runs as its own forward pass of the frozen model on a recipient prompt, and values move between passes only through the frozen return map and cast. Because ARITH's result is available near L36 while COMPARE's port closes by L12, a single forward pass cannot hand one to the other. The chain is therefore a program executed by the frozen model's own operations, one call at a time.

D.3 Forbidden operations

The LOAD $\rightarrow$ COMPARE preregistration permitted exactly one thing. We could deliver LOAD's value to COMPARE's frozen read maps with slot correspondence, then run the unchanged frozen L8 site, bind maps, scales, normalization, capacity, scoring and controls. It forbade, verbatim:

- new pairwise LOAD$\rightarrow$COMPARE regression;
- consumer refit;
- rank search;
- layer search;
- producer-specific gain tuning;
- normalization tuning.

It added: "Inherit the Part-B causal gate. Do not invent a LOAD-specific easier threshold." The same discipline holds throughout this appendix. The quota registration says "No ADD, COMPARE, rank, site, gain, or normalization parameter is fit on quota data." For the sensor task, the frontend may receive record text, the target station, the correction and the threshold. It may not receive the baseline value, the corrected value or the final relation label, and a provenance guard wrapped around the chain's payload region asserts this at run time. For ARITH's operand casts, the only gain permitted is

one global scalar per candidate computed from source norm statistics, with no per-item, per-value or per-slot tuning.

D.4 PAIRWISE EDGES (§6.1)

Each edge is evaluated on 100 fresh recipients of the consumer's source form, with nothing dropped.

Table 30: Execution of independently fitted pairwise links.

arm	LOAD → ARITH	LOAD → COMPARE	ARITH → COMPARE
native consumer (oracle)	0.970	1.000	0.990
produced return, frozen convention	**0.950**	**0.840**	**0.660**
stated value in the consumer's own prompt	0.970	0.900	—
raw produced state	0.000	0.000	0.000
wrong producer value → its own implied result	0.930	0.830	0.610
wrong producer value → original target	0.010	—	—
shuffled	—	0.160	0.100
random / token-embedding	0.000	0.000	0.000

For LOAD → ARITH (74 distinct ARITH items, item-clustered bootstrap with 10,000 draws), the return efficiency is η_{return} = 0.9794 [0.9474, 1.0000] and the specificity margin, target minus wrong-to-target, is 0.920 [0.854, 0.971]. For LOAD → COMPARE, η_{return} = (R − N)/(S − N) = 0.9333 [0.8667, 1.0000], where R is the produced-return rate, S the stated-value rate and N the no-op floor.

Where the ARITH → COMPARE loss lies. On the 70 strict-order items ARITH → COMPARE reaches 64/70 = 0.914, with less → greater at 1.000 and greater → less at 0.950. On the 30 equality items it reaches 2/30. The cast emits " equal" on 5 of 100 items against 29 for the native consumer. LOAD returns show no such collapse, at 0.971 on strict order and 0.533 on equality, so the equality loss belongs to the ARITH-produced return path, not to the COMPARE consumer. That is why equality is a secondary endpoint in D.8.

Stated values interoperate. Every consumer accepts numerals stated in a foreign prompt through its own convention. Stated LOAD and stated ARITH-operand values drive COMPARE at 0.98 each, and they agree on 100 items out of 100. A stated value drives ARITH at 0.97 against a native 0.96. A two-foreign-input COMPARE call with both operands stated reaches 0.99 against a native 0.98. The numeral code is joint, since reassembling tens and units from different sources gives only 0.24, and raw stated-span transport replicates on Qwen2.5-7B at 0.81 against a native 0.83. The produced-return edges in the table above are the stronger result, because there the value is computed by one recovered operation and read out of its result state, not copied from a numeral in the prompt.

D.5 ROVER ROUTING: SUPPLIED LOAD-FORM OPERAND AND ARITH

Task. A narrative describes energy costs for several route legs and assigns a two-leg route. The model has to report either the total or the difference in energy of the two legs. Example (target form):

The Selene rover travels from Hydrus to Cygnus using 55 units of energy. Norma to Cygnus uses 48 units. Cygnus to Octans uses 30 units. Mission control assigns the route Norma -> Cygnus -> Octans. How many units is the difference in energy of the two legs? The answer is

The recipient runs its own program, here Hydrus → Cygnus → Octans, totalling 55 + 30 = 85. The linked program installs a LOAD value for the target's first leg into ARITH's first operand, together with ARITH's opcode for the target operation. The 2×2 factorial of the two interventions has four distinct outcomes: 85 with neither, 48 + 30 = 78 with LOAD alone, 55 − 30 = 25 with ARITH alone, and 48 − 30 = 18 with both. Items are generated so that the four outcomes have distinct leading digits, with a certified collision count of 0, which makes partial execution directly observable.

Delivery. In this task the linker writes the value of the target's first leg as a LOAD-output state at L36, using its own linear map from a digit representation of the value. It does not use the LOAD return

map of D.2, which was built later. That state enters ARITH through ARITH's own first-operand cast, fitted on ARITH source data before the task was opened. The second operand is the recipient's own, and ARITH's opcode write sets the target operation. Nothing is fitted on rover-routing items. Rover-routing therefore tests whether ARITH, driven through its frozen operand cast and opcode interface, executes a foreign LOAD-form operand inside an unseen task. The produced-return edge LOAD $\rightarrow$ ARITH is tested separately, in D.4. A source-side qualification run before the task was opened had ranked three delivery routes: the read-and-bind cast at ρ 0.706 with LCB 0.580, projection onto ARITH's operand subspace at 0.606, and raw transplant, which was weak. The component orthogonal to ARITH's subspace scored 0.000.

Frozen before opening the 80 items. The gates are listed below. The refined semantic criteria S1 and S2 were frozen from calibration items of the same task, and their thresholds were set with that calibration in view, which the registration discloses.

Table 31: Rover-routing execution and gates.

gate	criterion	result
G1 oracle (native target state)	≥ 0.50	0.7750
G2 ρ_{link} = linked / oracle	≥ 0.50, LCB > 0	**0.9194**
G3 LOAD only $\rightarrow$ LOAD-only program	≥ 0.50, LCB > 0	0.7125
G4 ARITH only $\rightarrow$ ARITH-only program	≥ 0.50, LCB > 0	0.9250
G5 random $\rightarrow$ recipient program	≥ 0.80	0.9625
S1 target rate, correct minus wrong LOAD value	≥ 0.40, LCB > 0	0.6250
S2 wrong LOAD value $\rightarrow$ its own program	≥ 0.30 and $>$ its LOAD-only rate	0.5750 vs 0.0875

All 80 items (pool ids 180–259) were scored, with no fallbacks and no skipped items. Every gate passes.

Delivery routes on the target task. Linked LOAD + ARITH reaches 0.713 through the cast, 0.675 through the projection and 0.175 through raw transplant. The ordering fixed on the source tasks, cast above projection and both far above raw, predicts the ordering on an unseen target task.

What the factorial shows. Each single intervention executes its own partial program, LOAD alone at 0.7125 and ARITH alone at 0.9250, and together they execute the full program at 0.713 of the 0.775 available. A wrong LOAD value is not generic noise. It drives the full program implied by that wrong value on 57.5% of items, and the LOAD-only partial program on only 8.75%.

D.6 Production quota (ARITH + COMPARE)

Task. A production report gives two shift outputs and a quota, and the question is whether total production falls below, at or above the quota. Morning 47 and evening 38 against a quota of 82 gives 85, which is greater. The latent program is ARITH(a, b) $\rightarrow$ s followed by COMPARE(s, t).

Behavioral selection, frozen in code. We fixed four candidate families over the same numerical tuples in priority order: quota, capacity, workload and filtration. The selection rule was written before any scoring. Chain-of-thought had to reach 0.85, direct single-pass had to land between 0.40 and 0.85, every relation had to reach 0.15 under direct, the gap between CoT and direct had to reach 0.10, and the first qualifying family would win. The first generator leaked the relation through the arithmetic distribution, and we fixed it before any GPU scoring by matching the addend and sum distributions, making half the items carry, and using no distractors. Quota qualified first, with direct 0.4889 and CoT 1.0000, so the other families were never scored.

Runtime. The frontend reads a, b and t but not the relation. The first pass runs ARITH on a native ARITH-source recipient. The second runs COMPARE on a native COMPARE recipient whose own base relation differs from the target, with the ARITH result delivered through the frozen L8 cast. An

oracle arm installs the true sum through the same cast, and on calibration items it reaches 0.9333, with less at 0.96, equal at 0.84 and greater at 1.00, comfortably above the 0.80 runtime bar.

Result. 180 fresh items, 60 per relation, none dropped:

Table 32: Production-quota execution.

arm	rate
CoT	1.0000
oracle (true sum through the same cast)	0.9000
linked ARITH → COMPARE	**0.6611**
direct single pass	0.5222
wrong ARITH value → its own implied relation	0.7389
wrong ARITH value → original target	0.1833
shuffled	0.2000
raw ARITH state, random, no-op	0.0000

Three gates were preregistered. T1, composition, passes. T2, linked beats direct, passes with Δ = +0.1389 and a paired bootstrap LCB_{95} of +0.0556. T3, equivalence with CoT, fails at -0.3389. The verdict is LINKED_EXECUTION_CONFIRMED. The no-op arm returns the recipient's own base relation on every item, and ρ_{link} = 0.6611 / 0.9000 = 0.735. On the prespecified strict-order secondary the linked program reaches 0.9583, against 0.6333 for direct and 1.0000 for the oracle, but the pooled 0.6611 remains the headline. The loss concentrates on equality, 0.0667 linked against 0.700 for the oracle, which matches the ARITH-return equality loss of D.4.

D.7 CATALOG (LOAD + LOAD → COMPARE)

Task. A procurement catalog lists named entries with numerical attributes, and the question compares the values of two entries. Each item makes five neural calls. Each compared value is read as the LOAD transition between two queries on the byte-identical table, the target query minus a base query, which costs two LOAD calls per value. A single COMPARE call then receives both values in parallel, one per operand slot.

Shortcut audit. A fixed shortcut bar is invalid for a comparison task, because the maximum over many correlated leave-one-out rules sits near 0.43 under a permutation null at this sample size. We declare a shortcut only if a rule beats its own permutation-calibrated maximum-statistic null and is meaningfully above chance. In-sample pair lookups, which reach about 0.96 in sample, are rejected outright.

Frozen. The preregistration, sealed before opening, made joint execution the primary test. It kept a behavioral-advantage test that the pilot had already predicted would fail, since direct single pass on this surface runs between 0.88 and 0.97. The catalog task is therefore our parallel-composition test rather than a behavioral-advantage test.

Result. The 53 strict-order items are primary, and the 30 equality items are a secondary stress test. The oracle supplies both values as stated numerals through the same frozen casts.

Table 33: Catalog comparisons by return channel.

P1 channel	P2 channel	strict order	equality
stated	stated (oracle)	1.0000	0.8000
produced return	stated	1.0000	0.5333
stated	produced return	0.9811	0.5000
produced return	**produced return (linked)**	**0.9811**	0.4667

The verdict is LOAD_LOAD_COMPARE_EXECUTION_CONFIRMED, with ρ_{link} = 0.981 on the primary endpoint. The specificity controls behave as they should. Either produced channel alone

gives 0.0000 on target, raw uncast terminus states give 0.0000, and a wrong LOAD query drives its own implied relation at 1.0000 and the original target at 0.0000. Direct single pass is 0.9639 on all 83 items; it is not a matched comparison to the 53-item strict-order linked endpoint and is reported separately from that endpoint.

The joint interaction on the pooled rate, I_{joint} = RR − RS − SR + SS = +0.0844, is non-negative. Two produced returns in parallel cost no more than the sum of their individual costs, and the data rule out a large compounding penalty. They do not say why the two return errors interact favorably.

D.8 SENSOR CALIBRATION (LOAD → ARITH → COMPARE)

Task. A field-monitoring bulletin lists station readings, applies a positive calibration correction to one named station, and asks how the corrected reading compares to an action threshold:

Field monitoring bulletin / Dune Crossing recorded 46 / Kestrel Hill recorded 77 / Amber Fields recorded 58 / Ember Vale recorded 26 / Indigo Mesa recorded 35 / A calibration update requires that the Indigo Mesa reading be increased by 2. / The agency's action threshold is 36. / After the correction, is Indigo Mesa's reading less than, equal to, or greater than the action threshold? / Answer:

The latent program is LOAD(Indigo Mesa) = 35 → ARITH(35, 2) = 37 → COMPARE(37, 36) = greater. Skipping ARITH gives COMPARE(35, 36) = less, so the middle operation is required.

Construction. Corrections lie between 1 and 9, matching ARITH's frozen single-digit second operand, and they are always positive, since a negative correction would turn the operation from a sum into a difference. With a positive correction, a corrected value below the threshold implies a baseline below it too, so "less" can never be a skip-ARITH flip. The generator therefore forces every "greater" item to flip. Items come in matched triples, where one pair of correction and threshold produces all three relations by varying only the baseline. We audited surface and derived features against permutation-calibrated nulls. Two harness bugs were caught before scoring and fixed: a batch reader that returned the unintervened residual, and a two-operation ceiling built from a freshly rendered prompt. A payload-norm assertion and byte-aligned contexts now enforce both fixes. Because equality was already the known weak regime of ARITH → COMPARE, we preregistered less/greater as the primary endpoint and kept equality as a secondary stress test.

Result. 150 items, 100 strict-order and 50 equality, with none dropped. The provenance guard recorded 150 placements and 0 violations. Verdict THREE_OP_EXECUTION_CONFIRMED.

Table 34: Sensor-calibration execution by item class.

arm	all	less / greater	equal
oracle (COMPARE's native target state)	0.900	1.000	0.700
two-op ARITH → COMPARE suffix	0.633	0.900	0.100
LOAD → ARITH → COMPARE	**0.580**	**0.840**	0.060
direct single pass	0.333	0.490	0.020
CoT	1.000	1.000	1.000
wrong LOAD station → original target	0.293	0.440	0.000
raw uncast chain / matched random	0.313	0.470	0.000
no-op	0.313	0.470	0.000

The no-op arm executes the recipient's own base program at 1.0000. On the strict-order subset, its base relation differs from the target on 53% of items, so 0.470 is simply the rate at which the base answer happens to coincide with the target. Omitting ARITH altogether yields 0.333 on the balanced generator pool, with less at 1.000 and equal and greater at 0.000, which is chance. The chain cannot be reduced to LOAD → COMPARE.

Contrasts (strict order, item-clustered bootstrap):

Table 35: Sensor-chain counterfactual propagation.

quantity	value
wrong LOAD station → its own full downstream relation	0.990

The normalized incremental chain efficacy uses the definition of Section 3.3, with the no-op rate as the floor and the two-op suffix as the reference: $\rho_{\mathrm{fc,chain}} = (0.840 - 0.470)/(0.900 - 0.470) = 0.8605$. Normalizing against the two-op suffix isolates what the upstream LOAD contributes. Normalizing against COMPARE's oracle instead would give 0.698.

Direct and CoT. Direct and no-op are different baselines, with direct about 0.02 above no-op. All 150 items were joined on a semantic key and agreed on relation, margin and flip metadata. The linked program substantially rescues direct single-pass performance, but it does not outperform explicit chain-of-thought, which the model can do on its own and which reaches 1.000. The linked program also uses several forward passes and internal access, so the comparison is not compute-normalized.

D.9 Composition arithmetic

Parallel and serial composition obey different empirical error laws under the present instruments.

- **Parallel.** Two LOAD returns feeding COMPARE's two ports show no compounding penalty: $I_{\mathrm{joint}} = +0.0844$ on the pooled catalog rate (D.7).
- **Serial.** Adding LOAD upstream of the ARITH → COMPARE suffix costs -0.060 [-0.110, -0.019] (D.8). The product of the LOAD → ARITH return efficiency and the ARITH → COMPARE strict-order rate, $0.9794 \times 0.914 \approx 0.895$, overpredicts the chain's 0.8605. We recorded that independence product in advance as an expectation rather than a gate.

No single efficiency constant describes both. We do not claim lossless modularity, and we do not identify the mechanism of the serial penalty.

E SHARED INSTRUMENTS

This appendix defines the quantities, arms and procedures that Appendices B–D use repeatedly, so that they need not be restated there.

E.1 FIDELITY, ORACLES AND EFFICACY

Fidelity. F(a) is the fraction of held-out items on which arm a produces the behavior implied by the state it installs (Section 3.3). Behavior is the gate, and geometry and decodability appear only as explanation. Specificity tests distinguish the intended program from named wrong or partial programs where the task permits; other endpoints can have coincident labels, as in the sensor chain, and F is read against a named program, not as a bare correct or incorrect.

Oracle and ceiling. A native-state oracle installs the model's own target state through the same interface. Other protocols instead supply a true value through the same cast (quota), stated values (catalog), or the downstream suffix (sensor chain). The specified reference fidelity C is not a universal upper bound. An intervention is interpreted only when C clears a preregistered minimum. The rule exists because the first controlled composition test returned a low rate that turned out to be uninterpretable rather than negative. Its held-out state ceiling was as low as 0.06 (B.6, F).

Same-run anchors. Ceilings, floors and the primary intervention are recomputed in the same run. A historical ceiling is never used to normalize a later run, and a ceiling from one patch or readout instrument is never transferred to another. Where a target task needs a runtime ceiling, a calibration run establishes it before the evaluation items are opened (quota oracle 0.9333 against a 0.80 bar, D.6).

Efficacy. The main text uses the common form $\rho = (F - B)/(C - B)$ for a specified baseline B and reference C, with $C > B$. The table retains the historical names used in the experiment records; each names the floor and reference of its test.

Table 36: Historical efficacy names and their reference arms.

quantity	definition	used where
$\rho\,(B = 0)$	F(a) / C	tests whose untreated recipient scores 0 on the target (rover-routing, quota, catalog, pairwise edges). For catalog, which has no native target-state arm, C is the arm that delivers both values as stated numerals
ρ_{fc}	(F(a) − N) / (C − N), N the no-op rate	tests whose recipient runs a base program that coincides with the target at rate N > 0 (sensor chain, D.8)
η_{return}	(R − N) / (S − N), R produced return, S stated value	isolating the cost of reading a result out of a producer's state, relative to a stated numeral delivered through the same cast (D.4)
H	[F(CCG2) − F(FV)] / [F(native) − F(FV)]	natural two-operation composition when a state-independent continuation is strong (C.5)

For the sensor chain the ceiling C is the two-op ARITH → COMPARE suffix, because the question there is what the upstream LOAD adds. We also report the value against COMPARE's oracle (D.8). H is unclamped, its gate is H ≥ 0.50 together with a positive lower bound on the raw contrast, and a non-positive denominator returns HEADROOM_DEGENERATE. H measures the fraction of available behavioral headroom above the strongest continuation, not the fraction of the true mechanism recovered.

E.2 ARM DEFINITIONS

Table 37: Experimental arm definitions.

arm	definition
no-op	the recipient runs untouched under the same runtime, a causal control, not a measure of task competence
direct	the unmodified model on the actual target prompt, answering in one pass
CoT	the model generates worked intermediate steps freely before answering
oracle	the model's own target state installed through the same interface
stated	a numeral written in a prompt, read from its own span and delivered through the consumer's cast
produced return	the state an operation produces for its result, read out through the producer's return map and delivered through the consumer's cast
raw / uncast	the producer's state delivered without return map or cast, at matched norm
wrong value $\rightarrow$ target / $\rightarrow$ own	a deliberately different but valid producer result, scored against the original target and against the program that value implies
shuffled	the payload of a different item, delivered through the same path
matched random	a random displacement of the same norm. Where an off-direction control is needed, $\delta(\theta) = \cos\theta \cdot \delta + \sin\theta \cdot$ q with q $\perp \delta$ and $\|q\| = \|\delta\|$, so the norm is constant
token-bus	the norm-matched difference of input token embeddings, written at the same position, to test whether an effect is token substitution
cross-site / decoy	the same payload written at a position the consumer does not read
single port	in a two-argument call, only one argument delivered
fixed-vector continuation (FV)	the strongest state-independent competitor: a fixed displacement, per-mode or per-cell mean, in place of a state-dependent map
cast / projection routes	delivery through the consumer's read-and-bind cast, or projection onto its operand subspace

Wrong-value controls are scored against both their own implied program and the original target. Those outputs are separated where the task construction permits; the sensor-chain wrong-station intervention can retain the original relation. The no-op arm has to score 0 on target wherever the recipient and target programs differ, and that invariant caught an off-by-one indexing bug early in the project.

E.3 Capacity and rank

Rank is part of the interface, not a cosmetic hyperparameter. We choose it by causal efficacy on the source task under the current sample regime, never inheriting it from an earlier calibration and never choosing it by cosine similarity. Three episodes established the rule.

- **Description length is not writability.** In an early blind controlled study the description-length-optimal state had rank 4, and it was not reliably installable (0.303 against a no-op of 0.252, 2/4 folds). A rank chosen by causal efficacy on development data, 32, passed 4/4 withheld folds. The most compact predictive description and the causally writable state are different objects.
- **Inherited rank.** A natural SELECT map was fitted with rank inherited from an earlier numerical calibration where rank 32 had beaten rank 128 by less than 0.01. With only 60 fitting rows, ranks 32 and 128 are identical in sample space, so the tie broke low. The resulting predicted-reference handoff failed at 0.0513, while a diagnostic sweep reached 0.9833 at rank 128 with more tables (C.5).
- **Rank does not transfer across models.** Gemma's SELECT needed rank 128 (r32 0.0862, r128 0.8793), while Llama's best cell used rank 32 (C.7).

The state-dependent LOAD maps are ridge regressions ($\lambda = 1.0$) fitted in sample space on mean-centered states, with rank imposed by truncating the singular value decomposition of the full ridge solution. The frozen COMPARE interface has a fixed 20/40-dimensional capacity, chosen on COMPARE source data.

E.4 STATISTICAL PROCEDURE

Clustering. Where several recipient rows derive from one underlying item, we bootstrap underlying items, not rows. The LOAD → ARITH edge, for instance, has 100 rows over 74 items, and S8's categorical cell has 102 rows over 51 base tables. Bootstrap intervals use 10,000 draws unless stated otherwise. Paired contrasts, such as linked minus direct or chain minus suffix, are bootstrapped jointly over the same items. Where reported, per-arm intervals are 95% percentile-bootstrap intervals over items, with exact binomial intervals for cells where every item or no item succeeds.

Monte-Carlo tests. p_{MC} = (b + 1)/(m + 1), where b counts null replicates at least as extreme as the observation among m replicates. With m = 1,000 and b = 0 this gives 9.99×10^{-4}, the smallest value the test can report.

Preregistered versus descriptive. Each experiment names its primary endpoint and gates before its evaluation items are opened. Quantities computed afterwards are labelled descriptive and never promoted to a verdict. In this paper the descriptive quantities include the catalog interaction I_{joint} (D.7) and S8's decodability corroboration. The serial independence product 0.895 (D.9) was recorded in advance as an expectation but was not a gate, and the quota task's strict-order secondary was prespecified but does not replace the pooled headline.

Shortcut audits. A shortcut rule counts as a leak only if it exceeds its own permutation-calibrated maximum-statistic null and is meaningfully above chance. A fixed bar is invalid for this purpose, because the maximum over many correlated rules has a high null mean (≈ 0.43 at n ≈ 120 for a three-way task).

Guards. Before behavioral scoring, every intervention asserts a treatment norm above a numerical floor. We added that check after a dead-slot defect in which a diagnostic arm silently installed a zero vector (F). Interventions and caches are computed in the same forward pass. Result files are joined by semantic item keys rather than row indices, because the final runners shuffle item order. Before a gate is sealed, a feasibility script prints the attainable floor and ceiling, the sample size and clustering unit, and the direction of selection.

F NEGATIVE, ABANDONED AND CORRECTED RESULTS

This appendix is the project's ledger of results that failed, were abandoned, or were corrected after they were first recorded. Each entry gives what was believed, what broke it, and what stands now. We never rewrite a sealed verdict. Where a later experiment superseded a result, both are kept. Entries run chronologically within each branch.

F.1 CONTROLLED BRANCH

Table 38: Controlled experiments: negative and corrected results.

episode	believed	what broke it	what stands
first composition test	operators learned jointly with the state interface fail to compose	the state interface itself did not generalize to held-out compositions (fidelity 0.95 observed against 0.28 held out, with held-out ceilings as low as 0.06)	uninterpretable rather than negative. Replaced by operator-isolated CCG against a validated ceiling (80/80, B.6)
averaged order-reversal control	reversing order is one control	commuting pairs must succeed under reversal	reported separately: 0/6 non-commuting, 4/4 commuting
blind_h1	the description-length-optimal rank-4 state is the causal register	install 0.303 against no-op 0.252, 2/4 folds	sealed FALSE. The causally selected rank-32 state is a separate hypothesis, and blind discovery moved to a new substrate (blind_h2, B.8)
unseen-state retention of 119%	an interface retains more on unseen states than seen ones	the normalization used different null denominators for seen and unseen classes	rejected. The size-matched rerun gives $1.000 \rightarrow 0.966$
first actuator qualification	a shared rank-16 actuator qualifies, and orthogonal corruption helps	the orthogonal control inflated norm ($1.00 \rightarrow 1.80$), 43% of "wrong" states were target-equivalent, and an off-by-one gave the no-op arm a nonzero score	retracted. Constant-norm controls (E.2), and a new actuator sealed and qualified
write gain from argmax readback	a dense prototype write fully establishes the intermediate (1.0000)	the network's own consumer reads the requested state on only ≈ 0.88 of dense rows	withdrawn. The network's readback is now the primary write measure
mediation reported from the dense prototype	the first mediation table used the dense prototype to compute downstream gain	on BYPASS the dense prototype reaches 1.0000 while the localized write reaches 0.0000, so the first table would have called BYPASS mediated	corrected before reporting. Localized writes give MEDIATED 64/64 and BYPASS 0/64 (B.7)
"fully blind" recursive execution	the whole pipeline is blind	the mediator's support came from compiler labels	narrowed. A blind support finder later recovered the same eight coordinates and the same verdict (B.7)

F.2 NATURAL BRANCH

Table 39: Natural experiments: negative and corrected results.

episode	believed	what broke it	what stands
second natural family (non-commutative S_3 task)	a second task family closes the additive loophole	behavioral gate failed before any neural collection: two-step single-pass accuracy 0.17–0.38 across four models, while CoT reached 1.0000	abandoned before neural work, and not evidence about algebra knowledge
extremum SELECT	MAX/MIN selection is the natural SELECT task	behavioral admission failed (SELECT 0.7662, composed 0.4805)	closed without interface search. Positional selection used instead (C.1)
alternate prompt rendering	the blind factorization can be tested on a decoupled surface	the model failed admission on the alternate rendering (worst cells 0.28 and 0.00) and replacement was forbidden	surface invariance NOT_ ADJUDICATED (C.6)
first numerical LOAD boundary	LOAD recovered from a late boundary	the fixed-vector continuation already reached 0.9625 there	LOAD_LAW_NOT_ SEPARATED_FROM_ NULLS kept. A sealed successor on a fresh pool passes (C.4)
categorical predicted-reference handoff at rank 32	the handoff fails for categorical SELECT	inherited rank, since rank 128 reaches 0.9833 diagnostically	sealed CCG_ UNSUPPORTED kept for the frozen configuration. Rank rule adopted (E.3)
S7 absolute gate	composition must beat the continuation by +0.15	the continuation sat at 0.955, so the gate required 1.105	CCG2_NOT_ SEPARATED_FROM_ NULLS kept. The headroom estimand S8 was preregistered on a new pool (C.5)
Qwen categorical S8	H $\geq$ 0.50	H = 0.4146, with a significant contribution (LCB +0.0784)	CONTRIBUTION_NOT_ ADJUDICATED. Three proposed explanations, gradual answer formation, alignment losses and single-token names, were each retracted by a later control or analysis. Historical verdict retained; the alignment-certified successor passes with $\rho_{\mathrm{CCG}} = 0.7385$ (C.5)
cross-model scoring	task files are model-neutral	first-token IDs were Qwen's, and one script lacked the check	repaired to derive IDs from the live tokenizer before any Gemma causal result was accepted (C.7)
Llama at three shots	Llama cannot compose in one pass	an eight-shot successor passes (0.9583)	the three-shot result stays in the record, while the interpretation is retracted (C.7)
Gemma LOAD under S3b	Qwen's boundary rule ports	Gemma's generic continuation was already strong at the selected layer (gap +0.1312)	S3b failure reported. S3c was designed afterwards and evaluated on a fresh pool (C.7)
S3c pool statement	the S3c pool was untouched	part of it had been reserved, though never executed, by the blind study	erratum kept. A content-hash pool registry now enforces disjointness

Continued on next page

Table 39 (continued)

episode	believed	what broke it	what stands
payload-polymorphic LOAD	one LOAD executes all field types	per-field and cross-type holdouts: 0.0000 on categorical held-out fields and across types	MIXED_FIELD_LOAD_DOES_NOT_GENERALIZE; CROSS_TYPE_LOAD_DOES_NOT_TRANSFER in both directions (C.8)

F.3 CROSS-TASK BRANCH

Table 40: Cross-task linking: negative and corrected results.

episode	believed	what broke it	what stands
raw value interoperability	a LOAD value state can be installed directly as an ARITH operand	raw cross-source transplants recover 0.03–0.12	TYPED_BUT_INCOMPATIBLE. A consumer-defined cast rescues the first operand (ρ 0.706), which is what motivated calling conventions (D.2)
ARITH operand schedule	a bundled call writing the second operand at L34 is ARITH	ρ 0.448 (LCB 0.224)	ADD_BUNDLE_INCOMPLETE kept. The L36/L36 schedule was confirmed on a fresh pool (D.1)
rover-routing specificity criterion	a wrong LOAD value should rarely execute the LOAD-only partial program	on calibration items it often executed the wrong value's own full program instead	the original all-gates rule failed and is kept. Semantic criteria S1 and S2 were frozen before the evaluation items and passed (D.5), and the old criterion is retired
two-produced-operand arithmetic	a tree (a·b) + (c·d) can be linked	behavioral gate failed (0.8333 against 0.98, with each subtask at 1.0000)	stopped before neural collection
first LOAD → COMPARE arm	LOAD → COMPARE is weak (0.32)	the COMPARE reader was applied to LOAD's terminus at the wrong geometry	instrument-invalid. The LOAD return map was built for this edge (D.2)
"LOAD incompatible under the frozen interface"	LOAD cannot feed COMPARE	the verdict came from a clause requiring the cast to beat raw by +0.15, but raw stated spans matched native item by item	label kept but superseded: stated numerals interoperate generally (D.4)
early LOAD → ARITH near 1.0	an operation-output edge	code audit: the harness delivered the numeral stated in the prompt, not LOAD's produced result	reclassified as stated-value transport. The produced-return edge was then measured at 0.950 (D.4)
exact reproduction of the ARITH cast	the historical cast reproduces to 10^{-6}	it was fitted in-run and never serialized, with differences up to 3.2 $\times 10^{-5}$	ADD_CAST_NOT_REPRODUCED kept. A functional reconstruction passed its sealed signature gate and is serialized with hashes
first LOAD return-map run	the return map confirms	a token-string locator dropped 196 of 200 donors (n = 4)	void, retained as an invalid artifact. A character-offset locator verified 200/200 on a fresh pool

Continued on next page

Table 40 (continued)

episode	believed	what broke it	what stands
a ≈ 0.42 value-pair regime	a replicated plateau near 0.42	a diagnostic arm was a zero vector: a shared dead-slot defect	withdrawn from all tables. Non-degeneracy assertions added (E.4)
compact joint-binding law	a low-rank bilinear term explains ARITH's joint operand binding	gain +0.07 against a sealed +0.15 bar	NO_COMPACT_JOINT_BINDING_LAW
"COMPARE cannot do equality"	equality fails in the consumer	the quota oracle executes equality at 0.84	withdrawn. The loss sits in the ARITH-produced return (D.4)
quota: linked $\approx$ CoT	the linked program matches CoT	0.6611 against 1.0000	T3 failure kept (D.6)
catalog: linked $>$ direct	linked execution beats direct	direct single pass is already $\approx$ 0.96	NO_BEHAVIORAL_HEADROOM. Catalog reported as a composition test (D.7)
two computed sums $\rightarrow$ COMPARE	linking two ARITH results beats direct	pooled Δ +0.0167 [-0.0556, +0.0889], and a disjoint-tuple replication gives +0.0611 [-0.0167, +0.1389]	TWO_SUM_COMPOSITION_WITHOUT_ADVANTAGE: the two-sum program executes (0.5944, oracle 0.8444) but without a behavioral advantage
serial independence	chain efficacy $\approx$ product of edges (0.895)	measured 0.8605, with chain $-$ suffix -0.060 [-0.110, -0.019]	serial composition has a measurable additional cost, with the mechanism not identified (D.9)
same-pass latent linking	ARITH's result can reach COMPARE within one forward pass	ARITH's result forms near L36 and COMPARE's port closes by L12	a depth wall. Linked execution is staged (D.2)

G PROVENANCE

G.1 SEALING PROTOCOL

Every claim-bearing experiment follows the same order, stated once here.

- **Write the preregistration.** It names the primary endpoint, the arms, the gates and their thresholds, the item pools, and the verdict strings for each outcome, including the negative ones.
- **Certify the instrument.** Where an instrument is new, it is run on synthetic worlds with known answers (planted, null and adversarial) before it touches real states, and its thresholds are fixed from those worlds (B.8, C.2).
- **Freeze the objects.** Fitted maps are serialized and hashed as arrays, not as procedures, together with a fingerprint of the model's coordinate system. The code state is committed.
- **Predict and hash.** Where the test is prospective, predictions are computed and hashed before the targets exist or are opened. In the algebra-completion test (B.5) the held-out organisms are not compiled until the predictions are frozen. We compile the observed organisms, confirm that the targets are absent, infer them, hash the predictions, and only then compile the targets.
- **Run once.** The evaluation pool is opened and scored once. A pool touched by an invalid run is retired, not reused.
- **Adjudicate by the frozen rule.** The verdict string is the one the preregistration assigns. If a later experiment supersedes it, the later one gets its own registration and both are reported (F).

Blind experiments add an unblinding step: anonymous artifacts are committed before the truth map is opened, the adjudicator refuses to run before they exist, and canonical result paths refuse overwrite. Pools are checked against a registry keyed on content hashes before any run.

G.2 PREREGISTRATIONS AND FROZEN OBJECTS

Table 41: Preregistrations and frozen-object identifiers.

appendix	registration or frozen object	identifier	what it froze
B.5	algebra completion	predictions hashed before compilation	twelve held-out products
B.6	operator-isolated CCG	predictions hashed before opening	intermediates and finals for ten held-out compositions
B.8	blind_h1 hypothesis	439e1e124316829e	causal test at the description-length-selected rank (FALSE)
B.8	actuator qualification	3c7a7c050a85f9c1 → e131ad04 39fbc93a	the qualified write actuator
C.1	extremum admission	aab212a / 56d44095de1fd575	S0 closure
C.2	blind 2×4 recovery	1d518e4; surface amendment 29f7963; adjudicator 539abcf; pre-unblinding artifacts bb64a57; final f04d96e	τ_I, τ_Δ, verdict tiers, alternate-rendering rule
C.3	reference port	b14d059 / 8f3b036545aa108b	port liveness scan
C.4	SELECT, LOAD (S3b)	edc0b13, f5e9e44; c0c2e22	map cells, boundary rule
C.5	S7; S8	60f7afc; 9b719a0 (results ff69e40, bd9cbe9)	absolute gate; headroom estimand
C.5	qualified categorical successor	5b7aa4c6c9b702d2	alignment-certified family; unchanged composition gate

Continued on next page

Table 41 (continued)

appendix	registration or frozen object	identifier	what it froze
C.5	categorical handoff pool	c146765e817befad	replacement pool with nothing else changed
D.1	ARITH specification	53c11b6126f0e909	source task and confound audit
D.1	ARITH bundled call	d83b3b1b5e9ed96a	operand schedule
D.1	COMPARE operand port	1c71d9c8585a1685 / cba8ba6	port site L8 by frozen rule
D.2	COMPARE consumer interface	b29bd3ab0cbcbe86 / ddd15ba, c39ad08	read and bind maps, scales, normalization, capacity
D.2	OPERATION_FREEZE_V1	685aa7690f205183 / c69eb41, code 7348cab; ARITH return map 328ffee55c73b044	the operation library before any target task
D.2	LOAD return map	9d1f4c2f19707385	LOAD return interface
D.2	ARITH consumer cast	947f582ce51bd93f; arrays f039b bdf3235af95, 83728c022a36a73 3, 4b9c59dd0759987d, e53f7959 4a6714f0	functional reconstruction
D.4	LOAD → ARITH produced return	ddbba71e20dff2fc; result 96862d cae4c34fed	edge and gates
D.5	rover-routing delivery routes	predeclaration 2b3b4522b69332 41; result e74f6de62863a803	permitted producer layers and gains; first-operand cast (ARG1_BINDS_ONLY)
D.6	quota selection and gates	pilot 24b205c, selection 7702267, raw log 4b1f8db1c7dc 3c5f; result fccb1b6	family selection rule; T1–T3
D.7	catalog	ed4e27f4d392b0d6	primary execution test; behavioral test predicted to fail
D.8	sensor chain	result b22392203fb6cec2	construction, frontend boundary, endpoints

G.3 Integrity checks

The project maintains a manifest of sealed artifacts and frozen documents that is checked end to end, and an amendment ledger in which every change to a frozen document is recorded with its reason. The paper-number verifier regenerates each table from committed result files and compares it against the committed table instead of trusting exit status. It also normalizes Unicode minus signs before numeric comparison, re-derives pool hashes, and checks that every registered array exists and hashes to its claimed bytes.

G.4 Models and compute

Table 42: Models and experimental roles.

model	role
Tracr-compiled organisms	controlled experiments: sequence rearrangement, D_4 (3 layers, d_model 55), three-stage t1 (d_model 206), BYPASS (d_model 214), and blind-discovery substrates v1–v3
Qwen2.5-14B (base)	primary natural model, used for all natural and cross-task results unless stated, 48 layers
Gemma-4-12B (base)	cross-model replication of the SELECT → LOAD ladder (C.7), 48 layers, run through a qualified Hugging Face adapter
Llama-3.1-8B (base)	cross-model replication of the SELECT → LOAD ladder (C.7), 32 layers
Qwen2.5-7B	replication of raw stated-value transport (D.4), and earlier register-study replications
Gemma-2-9B	earlier register-study replication

All models are frozen, and Qwen state collection used fp32. The only trained objects are the small linear estimators described in each appendix. Experiments ran on single-GPU nodes of GH200 and H100 class. The numerical S8 run recorded torch 2.14.0 with CUDA 13.0, transformers 5.15.1 and TransformerLens 3.7.3.

H QUESTIONS WE EXPECT

1. Isn't the controlled arm just recovering a program you wrote? The program is known in advance, and that is the point of the controlled arm, which tests the procedure against ground truth. What the procedure is not given is the answer. Operation identities are hidden, and the recovered partition matches the six classes with a wide margin while head overlap misclassifies half the pairs (B.3). The algebra is identified after the fact as D_4 rather than Q_8 from an anonymous action bank, with the identity discovered rather than supplied (B.4). Twelve products are predicted before their organisms exist (B.5). The vocabulary transfers to an implementation compiled after it was frozen, within 131 bits of a free refit (B.8). And a network built to behave identically through a different internal route is rejected, with the mediator's support found blindly (B.7).

2. Is the D_4 table just memorized? No composition appears anywhere in the operator-isolated fitting data (B.6). A free 64-cell table is charged its full description cost and loses to the transferred operation vocabulary by 115,348 bits on a held-out implementation (B.8). Where algebraic support is removed, the procedure abstains rather than guesses: precision stays at 1.000 while coverage falls to 0.030 (B.5).

3. Does the blind natural recovery survive the factorial prompt surface? Not demonstrated. The primary rendering is factorial by design, and a surface baseline recovers the same assignment. The alternate rendering that would have decoupled surface from program failed behavioral admission, and replacing it was forbidden, so the abstract-structure verdict is NOT_ADJUDICATED (C.6). The paper uses the blind result for what it does establish, a uniquely identified selector $\times$ field organization that persists through depth and on unseen tables, and it assigns roles only from the causal experiments (C.3–C.5).

4. Is this function steering by another name? A function vector installs a fixed direction that triggers a behavior. The operations here are state-dependent transformations whose output depends on the recipient. A transplanted reference makes the recipient read its own table, at 1.000 for its own value and 0.000 for the donor's (C.3). A wrong LOAD value drives the whole downstream program that value implies, at 0.990 through two further operations (D.8). And each recovered map is gated against fixed-vector, per-mode and per-cell continuations fitted on the same data (C.4, C.5, D.1). Where a fixed continuation is strong enough to block separation, as in the first S7 gate, Gemma S3b and the Qwen categorical handoff, we report the failure and then measure composition by headroom above the continuation (C.5).

5. Is two-pass linking really the model composing? The claim is not single-pass composition. ARITH's result forms near L36 and COMPARE's port closes by L12, so the model's own schedule makes a same-pass handoff impossible (D.2). What is claimed is that operations recovered on separate tasks, joined only through frozen interfaces fitted on their own source tasks, execute unseen programs, and that counterfactual values propagate exactly through them. The decomposition into calls is specified externally, and we do not claim that the model decides it.

6. Does the frontend do the work? The target-local frontend reads the prompt's surface to decide which record to load and which quantities to pass on. It never sees the target answer, the intermediate values or the final relation. In the sensor task a provenance guard asserts this on every item, giving 150 placements and 0 violations (D.8). In the quota, catalog and sensor tasks the computation itself — retrieving the value, adding, comparing — is done by the frozen model's operations, and raw delivery that skips the calling convention fails (0.000 on every pairwise edge, D.4). In rover-routing the first-leg value is written directly in LOAD-output form, so that task tests ARITH's consumption and opcode control rather than retrieval (D.5).

7. Why does CoT beat the linked program? Chain-of-thought is available to the model and reaches 1.000 on the quota and sensor tasks. The linked program is not competing with it. It uses several forward passes, internal access and an external decomposition, so the comparison is not compute-normalized. Its purpose is to show that recovered operations execute a latent program the model does not complete in a single pass: 0.490 direct against 0.840 linked on the sensor task (D.8). The quota preregistration included a CoT-equivalence gate, and we report its failure (D.6).

8. Was S3c chosen so that Gemma would pass? S3c was designed after Gemma failed S3b, and we report S3b's failure as a failure. S3c was then sealed before its own pool was opened and evaluated

on fresh data. It is not a rule that always helps. On Qwen it selects the same boundary as S3b, and on Llama S3b gives the larger gap (C.7).

9. How was the Qwen categorical composition failure resolved? The original S8 run recovered $H = 0.4146$ and failed its frozen threshold; that verdict is retained. A prospectively alignment-certified successor passed with $\rho_{\mathrm{CCG}} = 0.7385$ under the unchanged gate. It increased fitting and evaluation support and selected a different SELECT site under the same frozen rule. Thus the new evidence establishes recoverability under the qualified instrument, without attributing the improvement to a single change (Appendix C.5).

10. What would falsify the calling-convention claim? Five observations would, and we tested for each. If a raw produced state executed without the cast, the cast would be doing no work. Raw states give 0.000. If a wrong producer value degraded the output generically instead of executing the program that value implies, the channel would not be carrying a value. Wrong values execute their own program at 0.930, 0.830, 0.739 and 0.990. If a consumer's cast ignored that consumer's causal structure, it would not be its own interface. COMPARE's cast reads tens at R^2 0.987 against units at 0.043. If the same payload kept its role when written at the other operand position, position would not be carrying the role. It executes the swapped comparison at 0.980. And if the convention did not survive a change of rendering, it would be a fit to one surface. Ours transfers at 0.89. It does not transfer across depth, which is why the site is part of the frozen object (D.2).

REPRODUCIBILITY STATEMENT

Every main-text claim is listed in the claim registry (A.2) with the appendix subsection that documents its design, frozen criteria, arms and results. The sealing protocol is stated once (G.1), and the preregistrations and frozen objects behind each claim are indexed with their identifiers (G.2). Arms and efficacy measures are defined in one place (E.1–E.2), and each recovered map is described with its fitting data and checkpoint. The compiled organisms are specified by their routing rules, operand routes, group actions and architectures (B.1–B.2), and their RASP sources will be released. The natural experiments use publicly available pretrained models (G.4), frozen throughout, and the only fitted objects are small linear estimators on hidden states. Code, task generators, frozen maps and result files will be released with the paper.