DOES THE WORKSPACE DO THE WORK? A CAUSAL PROTOCOL FOR VERIFYING GLOBAL-WORKSPACE COMPUTATION IN A LANGUAGE MODEL

Anonymous authors
Paper under double-blind review

ABSTRACT

Global Workspace Theory (GWT) is a leading theory of consciousness. It holds that information becomes conscious when it enters a limited-capacity workspace and is broadcast to specialized processes that use it. As AI is increasingly used to build and test such theories, an essential step is to determine whether a workspace built into a network is actually used or merely present. We introduce a causal protocol that answers this in stages. It asks whether the model depends on its workspace, whether receiving modules compute with the broadcast content, how that content acts, and whether it reaches several receivers at once. We apply it to a small language model whose modules can communicate only through a learned workspace. We confirm the results on four models trained under a protocol fixed in advance. Switching the broadcast off lowers accuracy by 41–52% on questions that need two modules' knowledge, while changing accuracy on single-module questions by at most 1.4 points. Dependence alone does not show use, so we then replace the broadcast with one taken from a different input. The receiver then combines the new content with its own memory, rather than copying the other input's answer, in 61–73% of cases, compared with 5–13% without intervention. Undoing only the resulting change in what the receiver retrieves restores the original output in 2,047 of 2,048 cases. This shows that the content acts through memory retrieval. A separately trained variant shows one broadcast serving three receivers. The protocol turns GWT's central claim into a testable property of trained models.

1 INTRODUCTION

Theories of consciousness increasingly make functional claims precise enough to be built into artificial systems. Indicator-based assessments check whether AI systems possess properties derived from these theories (Butlin et al., 2023; Chalmers, 2023). Interpretability work reports workspace-like representations in pretrained language models (Gurnee et al., 2026). AI also offers the theories a new experimental substrate that, unlike the brain, can be built to specification and modified through precise interventions.(Albantakis, 2025). All three research approaches rely on an assumption that is rarely tested: that a system built to contain a theory's component actually performs the computation the theory assigns to that component.

We call the distance between building a component and verifying its computation the verification gap. We study it for Global Workspace Theory (GWT; Baars, 1988; Dehaene et al., 1998; Dehaene and Changeux, 2011; Mashour et al., 2020), the most directly computational of the candidate conscious theories. GWT posits three components: specialized processors that work in parallel, a limited-capacity workspace for which they compete, and a global broadcast that makes the winning content available to all of them. On this view, what distinguishes consciously accessed information is not where it is stored. It is whether other processes can access and use it. GWT's central claim is thus itself a claim about use, and the verification gap falls exactly on it. Wiring a broadcast channel into a network is easy. Showing that training has taught the receivers to use what the channel carries is not. Our subject is this functional property, often called access consciousness (Block, 1995).

Standard checks do not close the gap. A natural check is ablation: if removing the channel harms performance, the channel is not being ignored. Ablation, however, can succeed merely because

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.

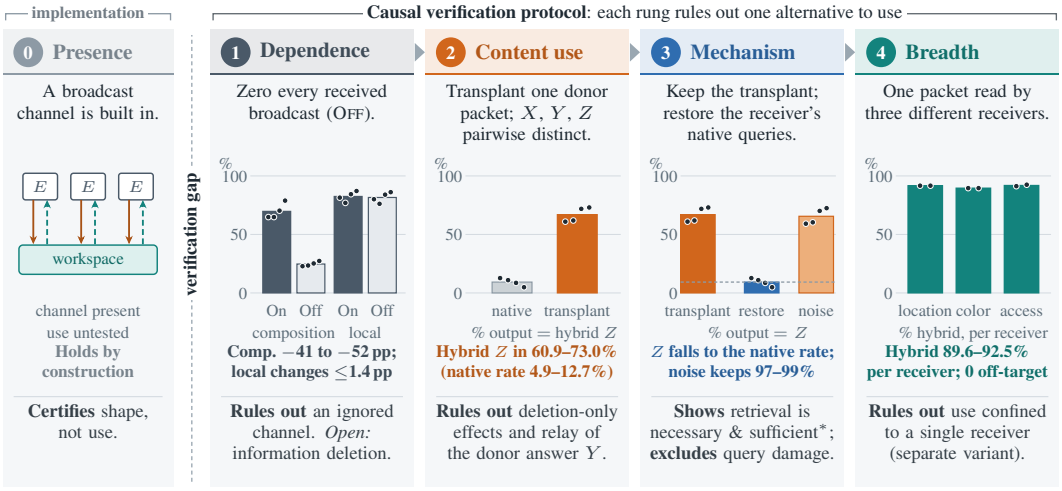

Figure 1: **From presence to use.** Building a broadcast establishes only its presence (rung 0). Each rung of the protocol applies a stricter intervention that rules out one alternative explanation (bottom row). In rungs 1–3, bars are means over the four confirmation models (A1, A2, B1, B2; 512 cases each), and dots are models. The dashed line in rung 3 marks how often the unperturbed model produces the hybrid answer Z by accident. Rung 4 uses the separately trained three-receiver variant (two seeds; cases in which X, Y, and Z are distinct). *Sufficiency holds at the output level; it comes from a supplementary query-only test on the same checkpoints and panels (Figure 4).

the channel carries information that no other pathway supplies. It thereby establishes the system's dependence on a route while revealing nothing about what the route conveys. A stronger check swaps in the broadcast computed from another input and asks whether the output changes accordingly. Yet even such a change may arise because the receiver passes on the donor's answer, rather than because it computes with the donor's content.

We propose a causal protocol that separates five increasingly strong claims (Figure 1): the broadcast is present, the system depends on it, receivers use its content, that content acts through a specific mechanism, and the use has the breadth the theory requires. We test each claim with an intervention that rules out the alternative left open by the claim below it. Two tools do most of the work. Three-value counterfactuals choose inputs on which three explanations predict three different outputs: ignoring the transplanted content, relaying another input's answer, and computing with the content. Native-query restoration and its complement, the query-only transplant, intervene on the receiver's memory queries. They test whether broadcast content acts by changing what the receiver retrieves. Restoration tests whether this route is necessary, and the query-only transplant tests whether it is sufficient.

A protocol is only as decisive as the system it is applied to. In a large pretrained model, every intervention on a candidate workspace leaves redundant pathways open. We therefore build a model organism instead (Elhage et al., 2022; Nanda et al., 2023). AFGW is a 6.9M-parameter autoregressive language model whose four recurrent modules can exchange information only through a single learned, competitive workspace. Two modules hold disjoint fact memories, so answering a question that combines their facts requires one module's knowledge to reach the other. We fix which memory each module can read and which module answers each question type. Everything else must be learned from answer tokens alone: what to broadcast, when, and how to use what is received.

We developed the tests on two models. We then froze the architecture, training recipe, intervention sites, and evaluation panels before training four new models on two new corpora. All four pass the first three rungs. Transplanting one broadcast packet makes the receiver produce the answer that requires combining the transplanted content with its own memory. This happens in 60.9–73.0% of 512 preselected cases per model, compared with 4.9–12.7% without intervention. Restoring only the receiver's native queries returns this rate exactly to its native level and recovers 2,047 of 2,048 native outputs. Conversely, installing only the redirected queries reproduces the transplant's output in 2,045 of 2,048 cases. A separately trained variant passes the fourth rung. Our contributions are:

- **The verification gap.** We separate five claims that are conflated when a theory-mandated component is built into a learned system. We also show that ablation passes by construction once the task is learned (Proposition 1).
- **A causal protocol.** A sequence of interventions progresses from presence to use, each designed to rule out a specific alternative explanation.(Section 3). The protocol needs only an identifiable broadcast site, a receiver-side read interface, and exact counterfactual targets.
- **A model organism.** AFGW's only channel between modules is a learned, competitive, recurrent broadcast, which makes the protocol's interventions exact (Section 2).
- **Evidence under confirmation discipline.** We use a frozen protocol, fresh corpora, and new initializations, with complete denominators and every exception reported. Under these conditions, the retrieval effect is positive in all 64 family–initialization combinations (Section 5).

2 AFGW: A Model Organism Built for Verification

AFGW implements four commitments of GWT: specialized processors, competition for limited shared access, global broadcast, and use of broadcast content by receivers. We chose the implementation so that each rung of the protocol can be applied exactly (Figure 2). Table 1 in Appendix C maps each commitment to its implementation and to the evidence for it.

Task. Each input describes a small world of four objects, four people, and four locations through two kinds of fact records. Binding records B map objects to people ("The holder of football is Jeff."). Location records S map people to locations ("Jeff is in the room."). Local questions query one relation. Composition questions need both. For example, "Where is the football?" has the answer $S(B(\text{football}))$, and reverse questions map a location back to an object. The two record types are stored in different modules' memories, so composition requires information to cross between modules.

Modules. Records are encoded independently, so no information crosses between them during encoding. AFGW has four recurrent modules. The binding reader E_1 reads only binding records, the location reader E_2 reads only location records, and E_0 and E_3 have no memory access. The question type determines which module answers. We call that module the receiver and denote it o; for "Where is the football?", the receiver is E_2.

Rounds. Each generated token is computed in four rounds, $r = 0, \ldots, 3$, starting from zero private states h_e and an empty workspace b. In each round, a fact-reading module e queries its own bank $\mathcal{B}_e$. The queries depend on its private state, the question features u_t and g, and the current broadcast:

$$q_e^r = Q_e(h_e^r, u_t, g, b^r), \qquad (C_e^r, \ell_e^r) = \text{Read}(q_e^r, \mathcal{B}_e). \tag{1}$$

The module then updates its state with a GRU-style update (Cho et al., 2014). It writes a packet p_e^r consisting of a control block and the lexical content ℓ_e^r of its reads. The packets compete for the workspace:

$$b^{r+1} = \sum_e \pi_e^r p_e^r, \tag{2}$$

where π^r is a learned softmax over the four modules and an empty NULL option. Every module receives the same b^{r+1}, and this broadcast enters every reader's next query. One module's knowledge can therefore change what another module retrieves next, which is the property our experiments test. All states reset at every token. The packet written in the last round ($r = 3$) is never read, so it serves as a built-in negative control.

Output and supervision. The vocabulary head sees only the question features and the receiver's final state and reads. It never sees the workspace or any other module. Decoding is greedy over the full vocabulary and must end with EOS. We fix memory access and which module answers each question type. Everything else is learned from final-answer tokens: what to broadcast, when, and how to use it. Intermediate entities, supporting records, and routing are never supervised. Appendix C gives the full specification, including the model's 6,891,784 parameters.

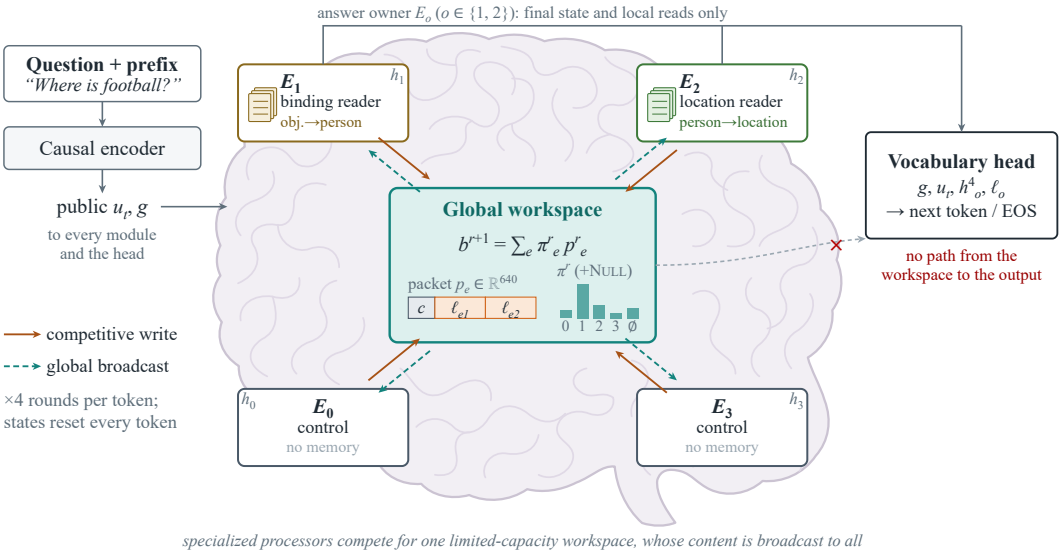

Figure 2: **AFGW computes each generated token in four rounds.** In each round, every module writes a packet: a control block c and two lexical blocks ℓ, which are empty for modules without memory. The packets compete with a NULL option, and the pooled message is broadcast to all modules before their next read. Numbered markers, colored as the rungs of Figure 1, show the design choices that make each rung identifiable: (1) the workspace is the only route between modules; (2) no path leads from the workspace to the output; (3) the broadcast enters each reader's next query, where the mechanism tests act. Routing bars are schematic.

Designed for identifiability. Three design choices, marked in Figure 2, make the rungs identifiable. (i) The workspace is the only route between modules, so switching it off is exact and its effect is provable. (ii) No path leads from the workspace to the output, so broadcast content can matter only through what a receiver does with it. (iii) The broadcast reaches a receiver by two routes: through its memory queries and directly through its private state. Which route training relies on is therefore an empirical question.

3 The Protocol: From Presence to Use

In AFGW, presence (rung 0) holds by construction; the protocol begins where an architectural assessment ends. The protocol requires three things: an identifiable broadcast site, a read interface on the receiver's side, and inputs whose counterfactual answers are known exactly. We present rungs 1–3 for AFGW with a running example (Figure 3); Appendix F gives formal definitions. Rung 4, breadth, applies the rung-2 transplant to a variant with several receivers (Section 5.4).

3.1 Rung 1: Dependence, and why it is not enough

The OFF intervention zeroes every received broadcast, in every round and at every generated token. It leaves memories, reads, state updates, and the output head unchanged.

Proposition 1 (OFF ceiling)**.** *Under OFF, the receiver's generated sequence is a function only of the question and the records in the receiver's own memory. Consequently, consider any two worlds that agree on the receiver's memory but give different answers to a composition question. OFF answers at most one of them correctly.*

The proof is an induction over rounds and tokens (Appendix F.1). Any model that learns to compose will therefore show a large OFF deficit. The deficit certifies that information flows through the workspace, but not what flows. More generally, when a channel is the only route for some information, removing it tells us about the route, not about the computation.

3.2 RUNG 2: CONTENT USE, VIA THREE-VALUE COUNTERFACTUALS

To test what the broadcast carries, we transplant it. In the recipient world of Figure 3, the football is held by Jeff, who is in the room. The receiver for "Where is the football?" is the location reader E_2, but only the binding reader E_1 knows who holds the football. We record E_1's packet in a donor world where Bill holds the football. We then substitute it for the recipient's packet at write $r = 2$ of the first answer token. Everything else evolves freely, including the competition for the workspace.

Three outputs are possible, and each points to a different explanation. Room, the recipient's own answer X, means the transplant was ignored. Kitchen, the donor's answer Y, means the packet relayed an answer. This is a live possibility, because by round 2 E_1's packet has received feedback from the donor's location reader. Hospital, where Bill is according to the recipient's own memory, means the receiver combined the transplanted content with its own knowledge. We call this the hybrid answer Z, because it is the correct answer in a world with the donor's bindings and the recipient's locations. If Y and Z were the same, a hybrid output could not separate computing from relaying. We therefore evaluate only cases in which X, Y, and Z are all distinct. Cases are selected symbolically from the worlds, never from model outputs.

Transplanted packets are the donor's own activations, so no correct answer is ever inserted into the model. Five controls accompany the transplant: replacing the packet with itself, a donor with the same facts in a different record order, a change to the memory that a local question does not need, packet noise matched to the transplant's norm, and a transplant at the never-read write $r = 3$.

3.3 RUNG 3: MECHANISM, VIA NECESSITY AND SUFFICIENCY OF REDIRECTED RETRIEVAL

Content use leaves open how the content acts. We hypothesize that it changes what the receiver retrieves from its own memory. The alternative is that the broadcast writes the answer into the receiver's state directly. Because AFGW's boundaries are enforced, we can enumerate every route (Figure 3, right). In the primary intervention, the transplant enters the computation only through b^3, which can reach the output by two routes. Route (a) runs through the receiver's round-3 queries and hence its reads, which feed the head and the final state. Route (b) runs directly into the receiver's final state. The other modules' round-3 updates only produce b^4, which nothing reads. We test the retrieval route with two complementary interventions on the receiver's queries. Together with the native and full-transplant runs, they form a 2×2 design over broadcast and queries (Figure 4a).

Native-query restoration tests necessity. It keeps the transplant but overwrites the receiver's round-3 queries with those recorded in the same model's unperturbed run on the same input; reading itself remains active. This severs route (a) and leaves route (b) open. If the hybrid answers vanish, redirected retrieval is necessary.

The query-only transplant tests sufficiency. It keeps the recipient running natively with its own broadcast, but installs the round-3 queries that the full transplant induced in that same recipient. This isolates route (a). If the transplant's output is reproduced, redirected retrieval is sufficient.

Restoration alone cannot show sufficiency, because the routes may interact. Consider a response $m(b, q) = bq$, with native values $(0, 0)$ and transplant values $(1, 1)$. Restoring q removes the whole effect, yet changing q alone produces none of it. The 2×2 design gives an exact per-case decomposition of the effect into three terms: a query-only change, a direct change at the native query, and their interaction (Appendix H.3). Two measurements complete the test. A matched-noise control perturbs the receiver's queries at the same sites by random displacements. Per case, round, and head, each displacement has the same norm as the one restoration applies. We also record read mass on the records needed for X and Z, to observe retrieval directly.

4 EXPERIMENTAL SETUP

Data and training. A family of worlds crosses three binding permutations, three location permutations, two record orders, six question types (four local and two composition), and four queried entities, giving 432 questions. Each model trains from its own initialization in two stages of 2,048 updates. The first stage uses local questions, with both readers seeing all records. The second stage

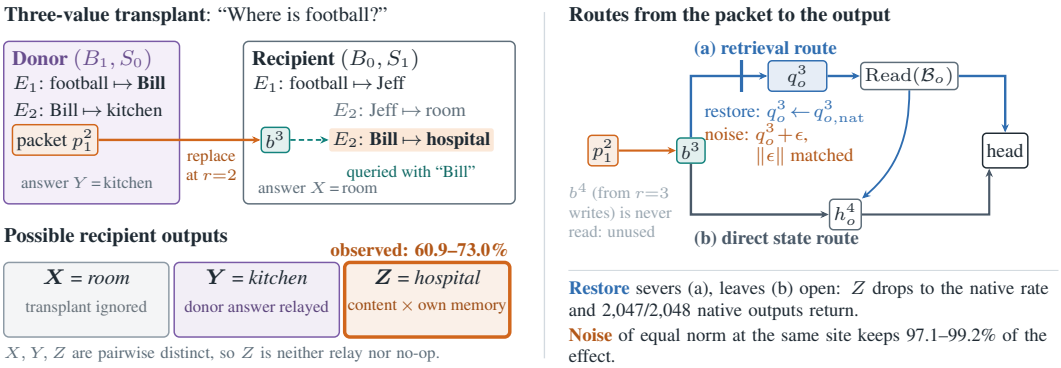

Figure 3: **Content and mechanism tests on a real case from the primary panel (Table 14).** Left: the donor's binding-reader packet from write $r = 2$ replaces the recipient's packet. An output of room (X) means the transplant was ignored, and kitchen (Y) means it relayed the donor's answer. Hospital (Z) means the receiver combined the transplanted content (Bill) with its own location memory. Right: the packet reaches the output only through b^3. It does so either via the receiver's round-3 queries and reads (route a) or directly via its final state (route b). Restoration severs route (a) and leaves route (b) open. Matched noise perturbs the same queries by the same amount.

uses mixed local and composition questions, with typed access restored. The loss is answer-token cross-entropy plus 0.1 times the EOS loss (Appendix D).

Discovery, then frozen confirmation. We developed the mechanism test on two discovery models and a 128-case panel that we inspected during development. We treat those results as exploratory. Before any new training, we fixed and hash-committed the architecture, curriculum, seeds, training endpoints, primary intervention site, and all evaluation panels. We then generated two new corpora, A and B, each with 8 training, 4 development, and 16 confirmation families. We trained two new initializations on each corpus (A1, A2, B1, B2). All four checkpoints existed before any evaluation on the confirmation split. No complete world is shared across splits. However, the grammar, the 24-name inventory, and some individual facts overlap, so confirmation tests recombination within one generating process. All confirmatory claims rest on these four models.

Panels and metrics. The primary panel has 512 symbolically selected recipients per model: 16 per composition direction in each of the 16 confirmation families. We transplant at write $r = 2$ of the first answer token, then apply restoration, noise, or nothing at the receiver's round-3 read. All three metrics use the full panel, including natively wrong cases, and exact match of the complete answer and EOS. The hybrid rate is the fraction of outputs equal to Z; the appendix tables call it target accuracy. Hybrid agreement is the fraction of outputs identical to the same model's native output on the hybrid world, errors included. Native recovery is the fraction of outputs identical to the unperturbed output. Secondary 128-case panels cover other writes and the controls, and separate 512-question panels test OFF (Appendix E).

5 Results

5.1 Rung 1: The trained model depends on its workspace

All four confirmation models fit all 3,456 training questions. On held-out worlds, composition accuracy is 64.02–75.48% and local accuracy is 79.10–86.31%. Deleting the broadcast reduces composition accuracy by 41.41–51.56 points but changes local accuracy by only 0.39–1.37 points (Figure 1, rung 1). The drop is not a net effect of offsetting changes. Per model, 243–277 composition answers flip from correct to wrong, against 13–31 in the other direction. As Proposition 1 predicts, the model depends on its workspace. The remaining question is what the workspace carries.

5.2 RUNG 2: RECEIVERS COMPUTE WITH BROADCAST CONTENT

Transplanting the binding reader's packet yields the hybrid answer Z in 60.9–73.0% of cases (312–374 of 512 per model). The unperturbed recipient produces Z by accident in only 4.9–12.7% of cases (Figure 1, rung 2). Because Y differs from Z by construction, these outputs are not relays of the donor's answer. The receiver has combined the transplanted person with its own location memory.

The transplant does more than produce the right symbol. The transplanted recipient reproduces the same model's native output on the hybrid world in 90.2–95.7% of cases, errors included. In other words, it behaves as though it had been given the hybrid world. This also explains most of the shortfall from 100%. At least 537 of the 677 target failures (79.3%, pooled descriptively) exactly reproduce the error that the model makes when natively given the hybrid world (Appendix H.1). The transplant thus inherits the model's own competence on the hybrid world. The missing hybrid answers mostly reflect that competence, not a failure of the transplanted content to take effect.

The effect is carried by content, in the expected part of the packet. On the discovery models, noise matched to the transplant's norm yields only 11–16 of 128 hybrid answers. Transplanting only the lexical blocks reproduces nearly the full effect (102/128 and 94/128), compared with 24/128 and 25/128 for the control block alone. In the confirmation models, self-replacement and a transplant at the never-read write $r = 3$ change no output. Earlier writes have smaller effects, consistent with the final read consuming content written late in the computation (Appendix E.1).

5.3 RUNG 3: CONTENT ACTS BY REDIRECTING THE RECEIVER'S RETRIEVAL

Redirected retrieval is necessary. Figure 4 summarizes the central finding: the answer follows the receiver's queries, not the broadcast. With the transplant still active, restoring the receiver's native queries reduces hybrid answers in every model. They fall to exactly the number that the unperturbed model produces by accident, a paired reduction of 48.24–68.16 points. Restoration also recovers the complete native output in 2,047 of 2,048 cases. The effect is clearest among recipients that the model answers correctly natively (332–404 per model). The transplant turns 206–304 of them into hybrid answers, and restoration leaves none in any model. A random query perturbation of the same norm, by contrast, retains 97.1–99.2% of the effect. Restoration therefore works because of what it restores, not because it disturbs the query. Route (b) stays open under restoration, so the broadcast cannot produce the answer change through the receiver's state alone.

Redirected retrieval is sufficient at the output level. After the confirmation study, we ran the query-only transplant on the same checkpoints and panels, under a protocol fixed before any new forward pass. The panels had already been observed, so we report the test as supplementary. With the native broadcast in place, installing only the transplant-induced queries gives a hybrid rate within one case of the full transplant's in every model. It also reproduces the full transplant's complete output in 2,045 of 2,048 cases. The full transplant changes the native output in 1,555 cases. In 1,551 of these, both endpoint equalities hold: the query-only output equals the full transplant's, and the restored output equals the native one. In the exact decomposition, the direct change at the native query is zero in every model. The interaction is nonzero in only 3 of 2,048 cases, and Appendix H.3 analyzes all four exceptions. We claim no more than output-level sufficiency. Query-only and full-transplant logits still differ, so the direct route and the interaction need not vanish at the logit level.

Retrieval shifts as predicted. Under both the full and the query-only transplant, the receiver's final-round read mass on the record needed for Z is 0.32–0.44. Restoration lowers it to 0.04–0.09; matched noise does not (Figure 4*b*). Mass on the native record moves the opposite way, from 0.05–0.10 under the transplant to 0.42–0.59 under restoration.

The effect is robust across families. The transplant-minus-restoration contrast is positive in all 64 family–initialization combinations (32 corpus-specific families × 2 initializations), with a minimum of 21.9 points. The noise-minus-restoration contrast is also positive in all 64, with a minimum of 15.6 points. Treating families as units, a one-sided sign test gives $p = 2^{-32} < 10^{-9}$. The discovery models showed the same pattern with somewhat larger effects (74.2–79.7%). This is expected for a panel inspected during development (Appendix E.1).

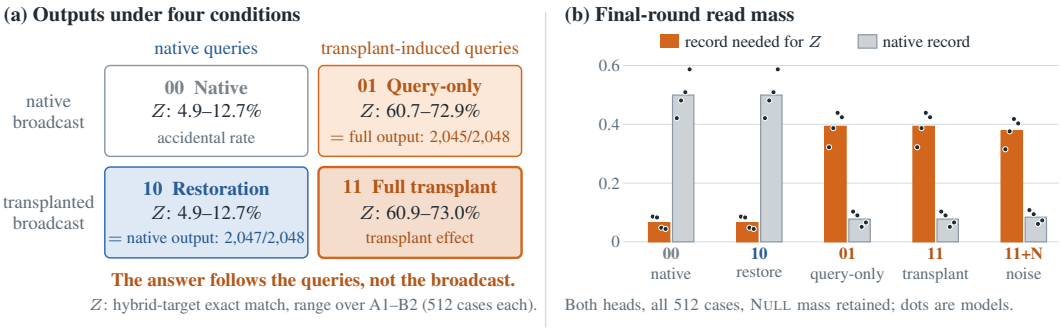

Figure 4: **The answer follows the receiver's queries, not the broadcast.** (a) The 2×2 design at the receiver's final read. It crosses the broadcast b^3 (native or transplanted) with the receiver's round-3 queries (native, or those that the transplant induced in the same recipient). Ranges give the hybrid rate over the four confirmation models (512 cases each). Agreements compare complete output sequences, including EOS, pooled over models. Cell 01 (query-only) is a supplementary test run after confirmation on the same checkpoints and panels. (b) Final-round read mass on the record needed for Z and on the native record (both heads, all cases, NULL mass retained; dots are models). Condition $11{+}N$ adds matched-norm query noise to the full transplant. Per-case decompositions and all exceptions are in Appendix H.3; the family-level analysis is in Appendix E.1.

5.4 Rung 4: One broadcast serves several receivers

GWT's broadcast is global: the same content should be available to many consumers. To test this, we trained a variant with one source module that reads object→person bindings and three receivers that map people to locations, colors, and access rights. One transplanted source packet therefore implies a different hybrid answer at each receiver. The variant has two initializations, trained with the same budget and answer-only loss. Its core rounds run once on the shared question, and each receiver then decodes from a private cache. It lies outside the frozen confirmation study and is evaluated on development families (Appendix G). On 512 joint episodes, a single transplanted source packet makes all three receivers produce their respective hybrid answers. This happens as often as the model answers all three natively (434/512 and 438/512), whereas deleting the broadcast leaves only 48/512 and 72/512 jointly correct. In cases where X, Y, and Z are distinct for a receiver, receiver-level hybrid accuracy is 89.58–92.50% (Figure 1, rung 4). The effect is also selective. Changing one receiver's memory, or cutting only one receiver's input, changes neither of the other receivers' outputs (zero changes in 1,024 comparisons per condition and seed).

6 Related Work

Consciousness science and AI. GWT and its neuronal variant are among the theories used to derive indicator properties for AI systems (Butlin et al., 2023). These properties concern access consciousness, the availability of information for reasoning, report, and control (Block, 1995). Philosophical analyses ask whether LLMs or LLM agents satisfy GWT's conditions (Chalmers, 2023; Goldstein and Kirk-Giannini, 2024). Other work links conscious-access functions to learning and generalization or formalizes workspace criteria (Bengio, 2017; Juliani et al., 2022; Kanai, 2026). We contribute a method for deciding when such criteria are met in a learned system, together with a model organism in which the method is decisive. This follows the case for toy models of consciousness theories (Albantakis, 2025).

Workspaces in AI systems. Some workspaces are scaffolded. Classical cognitive architectures (Shanahan, 2006; Franklin et al., 2014) and LLM agents built around a central buffer (Goldstein and Kirk-Giannini, 2024; Hu and Ying, 2025; Shang, 2026) pass text between model calls, so the workspace's use is fixed by design. Trained neural workspaces show that shared bottlenecks can aid coordination, generalization, and transfer (Goyal et al., 2021; 2022; VanRullen and Kanai, 2021; Devillers et al., 2024; Maytié et al., 2024; Dossa et al., 2024; Chateau-Laurent and VanRullen, 2025). They are evaluated mainly by performance, ablation, and transfer, which are checks that cannot establish use. Gurnee et al. (2026) discover verbalizable representations with functional hallmarks

of a global workspace in pretrained LLMs. Their analytic route retains scale and naturalness. Our constructive route trades these for an explicit, recurrent broadcast that is the only channel between modules, so interventions on it leave no redundant residual-stream routes open. The two routes approach the same question from opposite ends (Appendix B.1).

Causal analysis of neural networks. We build on activation patching and causal mediation (Vig et al., 2020; Meng et al., 2022; Zhang and Nanda, 2024), causal abstraction (Geiger et al., 2021; 2024), path patching (Wang et al., 2023; Goldowsky-Dill et al., 2023), and direct versus indirect effects (Pearl, 2001). Recent work warns that interpretability claims can outrun their evidence. Explanations may be non-identifiable (Méloux et al., 2025b), plausible findings arise even in random networks (Méloux et al., 2025a), interventions can leave the natural distribution (Grant et al., 2026), and counterfactual claims are hard to verify without ground truth (Joshi et al., 2026). The protocol is built around these concerns. Every rung targets a named alternative, transplanted packets are native activations, and symbolic worlds supply exact counterfactual targets. Pairwise-distinct targets separate a transferred answer from a transferred intermediate variable, and the main results are confirmed under a protocol fixed before training. Two-hop composition, a known difficulty for sequence models (Lake and Baroni, 2018; Press et al., 2023; Yang et al., 2024; Wang et al., 2024), serves here as a lever that forces one module's knowledge to be used by another.

7 Discussion and Conclusion

What the protocol establishes. Applied to AFGW, the protocol separates claims that architectural inspection and standard behavioral checks conflate. The trained model does more than contain or depend on a broadcast. Its receivers compute with the broadcast content, and they do so by letting it redirect their retrieval. This holds in all four models trained under a frozen protocol, and a separate variant shows one broadcast serving three receivers.

Why the result is not true by construction. The architecture makes workspace use possible; it does not make it happen. Dependence was guaranteed (Proposition 1), which is exactly why we do not rely on it. But nothing in the design forces receivers to combine transplanted content with their own memory rather than relay an answer. The direct route from the broadcast into the receiver's state is also open in every round. That the learned computation runs through retrieval is a property that training selected, and one the protocol could have refuted.

Implications. Indicator-based assessments typically ask whether a system has an architecture of a certain shape (Butlin et al., 2023). In our terms, they certify presence. Our results show that shape is not enough, and that the stronger claim of use can be tested. We suggest that future claims of workspace-like organization in AI systems, whether designed or discovered, be held to a higher standard. That standard is content-level, route-specific causal evidence, with controls matched to the intervention and confirmation under a fixed protocol. The gap is not specific to GWT: the other theories surveyed by Butlin et al. (2023) also yield indicators stated as functional roles of components. For machine learning, an explicit broadcast with enforced boundaries is interpretability by design, and model organisms with exact counterfactual ground truth can serve as testbeds for interpretability methods themselves (Méloux et al., 2025a; Joshi et al., 2026). Because the protocol needs only a broadcast site, a read interface, and exact counterfactual targets, a natural next step is to insert an AFGW-style workspace between specialist or retrieval components of a pretrained Transformer, with synthetic knowledge bases keeping counterfactuals exact, and to test further GWT properties such as ignition, capacity limits, maintenance, and report (Appendix B.2).

Conclusion. Building a theory-mandated component is only half the task. The other half is showing that the learned system uses it. We identified this verification gap and proposed a causal protocol that climbs from presence through dependence, content use, and mechanism to breadth. In a model organism built to make these distinctions causally identifiable, the protocol verifies that broadcast content is used, and that it is used through the receiver's retrieval. We offer the protocol, with AFGW as its reference instantiation, as a reusable standard for turning functional claims of consciousness science into falsifiable properties of trained systems.

AI USE STATEMENT

Generative AI tools were used for grammatical correction and language polishing of author-written text to improve clarity and readability.

REPRODUCIBILITY STATEMENT

Appendix C specifies the architecture, Appendix D the data and training, Appendix E the frozen confirmation protocol including all seeds and hash-committed plans, and Appendix F the intervention definitions and metrics.

REFERENCES

Larissa Albantakis. On the utility of toy models for theories of consciousness. arXiv:2508.00190, 2025. URL https://arxiv.org/abs/2508.00190. To appear in *The Scientific Study of Consciousness: Experimental and Theoretical Approaches* (Springer Nature).

Bernard J. Baars. *A Cognitive Theory of Consciousness*. Cambridge University Press, 1988.

Yoshua Bengio. The consciousness prior. arXiv:1709.08568, 2017. URL https://arxiv.org/abs/1709.08568.

Ned Block. On a confusion about a function of consciousness. *Behavioral and Brain Sciences*, 18(2): 227–247, 1995. doi: 10.1017/S0140525X00038188.

Patrick Butlin, Robert Long, Eric Elmoznino, Yoshua Bengio, Jonathan Birch, Axel Constant, George Deane, Stephen M. Fleming, Chris Frith, Xu Ji, Ryota Kanai, Colin Klein, Grace Lindsay, Matthias Michel, Liad Mudrik, Megan A. K. Peters, Eric Schwitzgebel, Jonathan Simon, and Rufin VanRullen. Consciousness in artificial intelligence: Insights from the science of consciousness. arXiv:2308.08708, 2023. URL https://arxiv.org/abs/2308.08708. Version 3.

David J. Chalmers. Could a large language model be conscious? arXiv:2303.07103, 2023. URL https://arxiv.org/abs/2303.07103.

Hugo Chateau-Laurent and Rufin VanRullen. Learning to chain operations by routing information through a global workspace. arXiv:2503.01906, 2025. URL https://arxiv.org/abs/2503.01906. Version 2.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of EMNLP*, pages 1724–1734, 2014. doi: 10.3115/v1/D14-1179.

Stanislas Dehaene and Jean-Pierre Changeux. Experimental and theoretical approaches to conscious processing. *Neuron*, 70(2):200–227, 2011. doi: 10.1016/j.neuron.2011.03.018.

Stanislas Dehaene, Michel Kerszberg, and Jean-Pierre Changeux. A neuronal model of a global workspace in effortful cognitive tasks. *Proceedings of the National Academy of Sciences*, 95(24): 14529–14534, 1998. doi: 10.1073/pnas.95.24.14529.

Benjamin Devillers, Léopold Maytié, and Rufin VanRullen. Semi-supervised multimodal representation learning through a global workspace. *IEEE Transactions on Neural Networks and Learning Systems*, 2024. URL https://arxiv.org/abs/2306.15711.

Rousslan Fernand Julien Dossa, Kai Arulkumaran, Arthur Juliani, Shuntaro Sasai, and Ryota Kanai. Design and evaluation of a global workspace agent embodied in a realistic multimodal environment. *Frontiers in Computational Neuroscience*, 18:1352685, 2024. doi: 10.3389/fncom.2024.1352685.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy models of superposition. Transformer Circuits Thread, 2022. URL https://transformer-circuits.pub/2022/toy_model/index.html.

Stan Franklin, Tamas Madl, Sidney D'Mello, and Javier Snaider. LIDA: A systems-level architecture for cognition, emotion, and learning. *IEEE Transactions on Autonomous Mental Development*, 6(1):19–41, 2014. URL https://ccrg.cs.memphis.edu/assets/papers/2013/franklin-ieee-tamd11.pdf.

Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. Causal abstractions of neural networks. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://arxiv.org/abs/2106.02997.

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*, volume 236, pages 160–187, 2024. URL https://proceedings.mlr.press/v236/geiger24a.html.

Nicholas Goldowsky-Dill, Chris MacLeod, Lucas Sato, and Aryaman Arora. Localizing model behavior with path patching. arXiv:2304.05969, 2023. URL https://arxiv.org/abs/2304.05969.

Simon Goldstein and Cameron Domenico Kirk-Giannini. A case for AI consciousness: Language agents and global workspace theory. arXiv:2410.11407, 2024. URL https://arxiv.org/abs/2410.11407.

Anirudh Goyal, Alex Lamb, Jordan Hoffmann, Shagun Sodhani, Sergey Levine, Yoshua Bengio, and Bernhard Schölkopf. Recurrent independent mechanisms. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/1909.10893.

Anirudh Goyal, Aniket Didolkar, Alex Lamb, Kartikeya Badola, Nan Rosemary Ke, Nasim Rahaman, Jonathan Binas, Charles Blundell, Michael Mozer, and Yoshua Bengio. Coordination among neural modules through a shared global workspace. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2103.01197.

Satchel Grant, Simon Jerome Han, Alexa R. Tartaglini, and Christopher Potts. Addressing divergent representations from causal interventions on neural networks. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2511.04638.

Wes Gurnee, Nicholas Sofroniew, Adam Pearce, Mateusz Piotrowski, Isaac Kauvar, Runjin Chen, Anna Soligo, Paul Bogdan, Euan Ong, Rowan Wang, Ben Thompson, David Abrahams, Subhash Kantamneni, Emmanuel Ameisen, Joshua Batson, and Jack Lindsey. Verbalizable representations form a global workspace in language models. arXiv:2607.15495, 2026. URL https://arxiv.org/abs/2607.15495. Preprint.

Pengbo Hu and Xiang Ying. Unified mind model: Reimagining autonomous agents in the LLM era. arXiv:2503.03459, 2025. URL https://arxiv.org/abs/2503.03459.

Shruti Joshi, Aaron Mueller, David Klindt, Wieland Brendel, Patrik Reizinger, and Dhanya Sridhar. Position: Causality is key for interpretability claims to generalise. arXiv:2602.16698, 2026. URL https://arxiv.org/abs/2602.16698.

Arthur Juliani, Kai Arulkumaran, Shuntaro Sasai, and Ryota Kanai. On the link between conscious function and general intelligence in humans and machines. *Transactions on Machine Learning Research*, 2022. URL https://openreview.net/forum?id=LTyqvLEv5b.

Ryota Kanai. A control-theoretic formulation of global workspace theory. arXiv:2608.15926, 2026. URL https://arxiv.org/abs/2608.15926. Preprint.

Brenden M. Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2873–2882, 2018. URL https://proceedings.mlr.press/v80/lake18a.html.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1711.05101.

George A. Mashour, Pieter Roelfsema, Jean-Pierre Changeux, and Stanislas Dehaene. Conscious processing and the global neuronal workspace hypothesis. *Neuron*, 105(5):776–798, 2020. doi: 10.1016/j.neuron.2020.01.026.

Léopold Maytié, Benjamin Devillers, Alexandre Arnold, and Rufin VanRullen. Zero-shot cross-modal transfer of reinforcement learning policies through a global workspace. *Reinforcement Learning Journal*, 2024. URL https://arxiv.org/abs/2403.04588.

Maxime Méloux, Giada Dirupo, François Portet, and Maxime Peyrard. The dead salmons of AI interpretability. arXiv:2512.18792, 2025a. URL https://arxiv.org/abs/2512.18792.

Maxime Méloux, François Portet, Silviu Maniu, and Maxime Peyrard. Everything, everywhere, all at once: Is mechanistic interpretability identifiable? In *International Conference on Learning Representations*, 2025b. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/51157713f1f1f572a05647e1aef1b1ed-Abstract-Conference.html.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, volume 35, pages 17359–17372, 2022. URL https://arxiv.org/abs/2202.05262.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2301.05217.

Judea Pearl. Direct and indirect effects. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, pages 411–420, 2001. URL https://ftp.cs.ucla.edu/pub/stat_ser/R273.pdf.

Ofir Press, Noah A. Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2108.12409.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5687–5711, 2023. doi: 10.18653/v1/2023.findings-emnlp.378. URL https://aclanthology.org/2023.findings-emnlp.378/.

Murray Shanahan. A cognitive architecture that combines internal simulation with a global workspace. *Consciousness and Cognition*, 15(2):433–449, 2006. doi: 10.1016/j.concog.2005.11.005.

Wenlong Shang. "theater of mind for LLMs: A cognitive architecture based on global workspace theory. arXiv:2604.08206, 2026. URL https://arxiv.org/abs/2604.08206.

Rufin VanRullen and Ryota Kanai. Deep learning and the global workspace theory. *Trends in Neurosciences*, 44(9):692–704, 2021. doi: 10.1016/j.tins.2021.04.005.

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, 2020. URL https://belinkov.com/assets/pdf/neurips2020.pdf.

Boshi Wang, Xiang Yue, Yu Su, and Huan Sun. Grokked transformers are implicit reasoners: A mechanistic journey to the edge of generalization. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://arxiv.org/abs/2405.15071.

Kevin Ro 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.

Sohee Yang, Elena Gribovskaya, Nora Kassner, Mor Geva, and Sebastian Riedel. Do large language models latently perform multi-hop reasoning? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10210–10229, 2024. doi: 10.18653/v1/2024.acl-long.550. URL https://aclanthology.org/2024.acl-long.550/.

Fred Zhang and Neel Nanda. Towards best practices of activation patching in language models: Metrics and methods. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2309.16042.

A Notation and Reader's Guide

The main text uses plain-language names for several quantities that the appendix tables label more technically. This guide maps one set of names to the other.

Worlds and answers. The recipient world (B_0, S_1) is the input on which the model is run and intervened on. The donor world (B_1, S_0) is the input from which the transplanted packet is taken. The hybrid world (B_1, S_1) combines the donor's bindings with the recipient's locations. X, Y, and Z are the correct answers in the recipient, donor, and hybrid worlds. The primary panel contains only cases in which X, Y, and Z are pairwise distinct.

Modules. The source is the module whose packet is transplanted: the binding reader E_1 for object→location questions and the location reader E_2 for location→object questions. The receiver is the module that produces the answer; the appendix calls it the output owner o.

Sites. Write r is the packet written in round r, which enters the broadcast b^{r+1}. Read r is the receiver's memory read in round r, whose queries depend on b^r. The primary site is write $r = 2$ followed by read $r = 3$, both on the first answer token.

Interventions. The appendix tables label the interventions as follows. OFF zeroes all received broadcasts. The full transplant is S, or condition 11. Native-query restoration is Q, or condition 10. The query-only transplant is condition 01. Matched query noise is N, or condition $11{+}N$. The native run is condition 00.

Metrics. The hybrid rate is exact match to Z followed by EOS; the appendix tables call it target accuracy, or "target". Hybrid agreement is identity with the same model's native output on the hybrid world, errors included. Native recovery is identity with the unperturbed output.

Models. The discovery models are seeds 104101 and 104102, evaluated on a 128-case panel inspected during development. The confirmation models, from study G4, are A1, A2, B1, and B2, trained on corpora A and B under the frozen protocol. The three-receiver variant uses seeds 106101 and 106102.

Routes (Figure 3, right). Route (a) runs from b^3 through the receiver's round-3 queries and reads to the output. Route (b) runs from b^3 directly into the receiver's final private state.

B Extended related work and discussion

This appendix gives the complete versions of the related-work and discussion material that the main text summarizes.

B.1 Extended related work

Consciousness science meets AI. GWT and its neuronal variant (Baars, 1988; Dehaene et al., 1998; Dehaene and Changeux, 2011; Mashour et al., 2020) are among several theories now used to derive functional criteria for assessing AI systems (Butlin et al., 2023). Following Block's (1995) distinction, these criteria concern access consciousness, that is, the availability of information for reasoning, report, and control. Philosophical analyses have asked whether LLMs or LLM-based agents meet GWT's conditions (Chalmers, 2023; Goldstein and Kirk-Giannini, 2024). The consciousness prior (Bengio, 2017) and Juliani et al. (2022) link conscious-access functions to learning and generalization, and Kanai (2026) recently proposed a control-theoretic criterion that identifies a workspace as a mediating subnetwork. Our contribution complements these analyses. It is methodological and empirical: we specify what evidence shows that a learned system uses a theory-mandated organization, and we build a model organism in which that evidence is decisive. This follows the broader case for toy models as a way to make theories of consciousness concrete and testable (Albantakis, 2025).

Three directions of prior work. Existing work approaches global workspaces from three directions, and each leaves the question of use open. Analyses of pretrained LLMs discover workspace-like structure after the fact, but cannot choose the architecture or its information boundaries, and standard Transformers lack the explicit recurrent broadcast that the theory describes. In language-agent scaffolds (Goldstein and Kirk-Giannini, 2024; Hu and Ying, 2025; Shang, 2026), the "workspace" is a text buffer outside the network, so whether its contents are used is determined by prompt design rather than by learned computation. Trainable neural workspaces (Goyal et al., 2022; VanRullen and

Kanai, 2021; Dossa et al., 2024; Chateau-Laurent and VanRullen, 2025) come closest, showing that shared communication bottlenecks can aid coordination, generalization, and transfer, but they are evaluated chiefly by task performance.

Scaffolded workspaces. Classical cognitive architectures implemented GWT-like cycles symbolically (Shanahan, 2006; Franklin et al., 2014), and their modern successors orchestrate pretrained LLMs around a central buffer. The Unified Mind Model builds an autonomous agent around GWT (Hu and Ying, 2025), Global Workspace Agents run an event-driven broadcast among heterogeneous LLM agents (Shang, 2026), and Goldstein and Kirk-Giannini (2024) argue that such language agents may satisfy GWT's conditions. In all of these systems, the workspace is external text passed between model calls. Its "use" is specified by design.

Trained neural workspaces. Goyal et al. (2022) replaced pairwise communication among neural modules with a limited-capacity shared workspace, reporting gains across tasks that include a next-word prediction experiment; related modular designs include Recurrent Independent Mechanisms (Goyal et al., 2021). The Global Latent Workspace (VanRullen and Kanai, 2021) has been realized as a shared multimodal latent space that supports semi-supervised learning and zero-shot policy transfer (Devillers et al., 2024; Maytié et al., 2024). Dossa et al. (2024) embodied a workspace agent in a multimodal environment, and Chateau-Laurent and VanRullen (2025) routed information through a workspace to chain operations, using correct routing in part of component pretraining. These works establish that workspace bottlenecks can be useful, but their evaluation centers on performance, ablation, and transfer. AFGW differs from them in three respects. It is an autoregressive language generator. Its information boundaries make the broadcast the only possible cross-module route, so the broadcast's necessity can be proven rather than assumed. And it comes with causal tests that ask what broadcast content does inside the receiver.

Discovered workspaces. Gurnee et al. (2026) use a Jacobian lens to identify a small set of verbalizable representations in pretrained LLMs that show functional hallmarks of a global workspace, a striking finding precisely because these models were never designed to have one. Their analytic route and our constructive route are complementary, but the constructive route offers three things that post hoc analysis of a monolithic network cannot. First, it brings the architecture closer to the one GWT actually describes: Gurnee et al. (2026) themselves note that the brain's workspace is closely associated with recurrent dynamics and interactions among brain regions that have no direct analog in the Transformer, whereas AFGW implements specialized modules, competition for access, and recurrent broadcast as explicit architectural components. Second, the workspace is a specified part of the computation rather than a subspace estimated by an analysis tool, so its identity does not depend on the validity of that tool. Third, because the broadcast is the only route between modules (Proposition 1), interventions on it are exact: switching it off, transplanting its content, or restoring a single downstream path changes precisely the information we intend to change, whereas in a model whose candidate workspace shares the residual stream with all other computation, the same manipulations may leave redundant routes open. The price of this control is scale and naturalness, which the discovered-workspace approach retains; we therefore see the two routes as approaching the same question from opposite ends.

Causal analysis of neural computation. Our interventions build on activation patching and causal mediation analysis (Vig et al., 2020; Meng et al., 2022; Zhang and Nanda, 2024), interchange interventions and causal abstraction (Geiger et al., 2021; 2024), path-level patching (Wang et al., 2023; Goldowsky-Dill et al., 2023), and the distinction between direct and indirect effects (Pearl, 2001). Recent work cautions that interpretability claims can outrun their evidence: candidate explanations may be non-identifiable (Méloux et al., 2025b), plausible-looking findings can arise even in randomly initialized networks (Méloux et al., 2025a), interventions can push representations off the model's natural distribution (Grant et al., 2026), and counterfactual claims are hard to verify without controlled supervision (Joshi et al., 2026). Our protocol is designed with these concerns in mind. Every rung is tested against an explicit alternative hypothesis, transplanted packets are the donor's native activations rather than synthesized vectors, symbolic worlds supply exact counterfactual targets, and the main results are confirmed under a protocol fixed before training. Two further features are specific to our setting. First, the three-value design guards against a known ambiguity of patching, namely that a patched component may transfer an answer rather than an intermediate variable. We avoid this by selecting cases in which the receiver's answer, the donor's answer, and the hybrid answer are pairwise distinct. Second, because the architecture enforces its information boundaries, we can

enumerate exactly all paths from the intervention point to the output (Section 3.3). This makes query restoration a sharp test of which route carries the effect.

Compositional reasoning. Our tasks are two-hop relational compositions. In this setting, neural sequence models struggle with systematic generalization (Lake and Baroni, 2018), and LLMs show a gap between knowing facts and composing them (Press et al., 2023; Yang et al., 2024; Wang et al., 2024). We use composition as a lever rather than a benchmark: it forces information held by one module to be used by another.

B.2 Extended discussion

Scope. GWT is our test case, but the verification gap arises whenever a theory-mandated component is built into a learned system. Within GWT, we test one functional property at the heart of conscious access: broadcast content is globally available for other processes to use. We ask whether that property can be built into a language model, acquired through learning, and verified causally, and we make no claim about subjective experience.

What the protocol connects. The protocol connects three levels that have so far been studied separately: an explicitly designed communication architecture derived from GWT, behavior acquired through training from answer tokens alone, and causal evidence that broadcast content steers what receivers compute. Within fixed weights, a transplanted broadcast redirects a receiver's memory retrieval, and removing only that redirection removes the behavioral effect; installing only the redirected queries, with the native broadcast in place, reproduces the effect at the output level.

Implications for consciousness science in AI. Indicator-based assessments typically ask whether a system has an architecture of a certain shape (Butlin et al., 2023); in our terms, they certify presence. Our results show that shape is not enough, and that the stronger claim of use can be tested. We therefore suggest that future claims of workspace-like organization in AI systems, whether designed or discovered, be held to the same standard. That standard requires content-level, route-specific causal evidence, controls matched to the intervention, and confirmation on fresh data under a fixed protocol. The verification gap is not specific to GWT: the other theories surveyed by Butlin et al. (2023) also yield indicators stated as functional roles of components, and implementing any of them opens the same distance between presence and use. This is the sense in which we see AI as an experimental substrate for consciousness science, and AFGW as a step toward AI with conscious-access-like organization: the step does not assert such properties, but makes them buildable and falsifiable.

Implications for machine learning. An explicit broadcast with enforced information boundaries is a form of interpretability by design. For any computation, we can state which module could have known what, and we can test which route carried the answer. Modular language models with such auditable communication may be useful wherever guarantees about information flow matter, such as compartmentalized knowledge, monitoring, and debugging. Model organisms with exact counterfactual ground truth may also serve as testbeds for interpretability methods themselves, whose conclusions are otherwise hard to validate (Méloux et al., 2025a; Joshi et al., 2026).

Toward scale. The protocol requires only an identifiable broadcast site and a receiver-side read interface, so it does not depend on our architecture or scale. A natural next step is to insert an AFGW-style workspace between specialist or retrieval components of a pretrained Transformer, where exact three-value counterfactuals remain constructible using synthetic knowledge bases. Beyond that, further GWT properties can be tested with the same logic, including nonlinear ignition, capacity limits, maintenance of workspace content across tokens (AFGW currently resets its workspace at every token), and report.

C Architecture specification

C.1 From theory to design: commitments and overview

This subsection gives the complete version of the architecture description summarized in Section 2. We use GWT to choose architectural commitments, and we choose their implementation so that each rung of the protocol in Section 3 can be applied exactly (Figure 2). We also explicitly separate what we supply by design from what training must discover (Table 1). We supply each module's memory

access and specify which module produces the answer for each question type. Everything else must be learned from final-answer supervision alone: what to communicate, when to communicate it, and how to use what is received. Intermediate entities, supporting records, and routing sequences are never supervised.

Table 1: GWT commitments, their implementation, and what our experiments test.

GWT function	Implementation in AFGW	What is supplied vs. established
Specialized, parallel processing	Four recurrent modules with private states. E_1 reads only binding records; E_2 reads only location records; E_0 and E_3 have no memory access.	Supplied: bank types and output ownership. Learned: the local tasks.
Limited-capacity shared access	One 640-dimensional packet per round, allocated by learned competition that includes a NULL option.	Established: the channel is used (Section 5.1). Not established: that competition or this width is necessary.
Global broadcast	The same message reaches every module before its next read and update.	Established: content-specific effects (Section 5.2); one broadcast serving three receivers (Section 5.4).
Content-dependent use by receivers	The received broadcast and the private state jointly form each receiver's memory queries.	Established: query restoration abolishes the content effect in all four confirmation models, and query-only transplantation reproduces it at the output level (Section 5.3).

Local encoding with enforced boundaries. The base model has 6,891,784 parameters, an 8,192-token vocabulary, and 256-dimensional token representations. Two causal-attention layers with ALiBi biases (Press et al., 2022) encode each fact record independently, while the question and the generated prefix form a separate segment. We denote the current representation of that segment by u_t, and we denote a fixed projection of the question anchor by g. Because each record is encoded in isolation, no information crosses between memory banks inside the encoder. Four recurrent modules receive the public features u_t and g, and the public question type determines the output-owning module $o \in \{1, 2\}$. Record types and output ownership are thus structural priors, and we state them as such.

One internal round. For each predicted token, the private states h_e and the workspace b start at zero, and four internal rounds $r = 0, 1, 2, 3$ follow. In each round, a fact-reading module e first forms two 128-dimensional queries and reads from its own bank $\mathcal{B}_e$ (Eq. 1). The contextual read C and the lexical read ℓ use the same joint record/token weights; their values come from contextual states and from normalized token embeddings, respectively. Record selection is sigmoid-gated and order-sensitive: it favors the last matching record, and any unallocated mass becomes an explicit NULL read (Section C.3 gives the exact reader). After a GRU-style update of its private state (Cho et al., 2014), each module writes a packet consisting of a 128-dimensional control block and two 256-dimensional lexical blocks, the latter being zero for non-readers. The packets then compete for the workspace (Eq. 2). Here the routing distribution π^r is a softmax over the four modules and NULL, whose packet is zero. Its query depends on the goal and the previous workspace control state, while the module keys depend on the updated private states. Crucially, every module receives the same b^{r+1}, and this broadcast enters every reader's next query. One module's content can therefore change what another module reads next, which is precisely the property our experiments test. The write at $r = 3$ is not used in any later step, so it serves as a built-in negative control.

Generation without a workspace shortcut. A tied-embedding vocabulary head receives only four inputs: g, the owner's final state h_o^4, the prefix representation u_t, and the owner's final lexical reads. It has no direct input from the workspace b, from any other module's state, or from an unrestricted read of the full memory. Decoding is greedy over the full vocabulary and must terminate with an EOS token, without any candidate-answer restriction or output repair. Record encodings may be cached, but private and workspace states reset at every token, so the four-round core reruns for each generated token. The multi-recipient extension has a different decoding schedule (Appendix G).

C.2 BASE-MODEL DIMENSIONS AND OWNERSHIP

Table 2 specifies the base model used for the main capability and query-path experiments. A four-entity story contains eight records; an eight-entity story contains sixteen. Modules E_1, E_2 have fact-reading interfaces, whereas E_0, E_3 can contribute control state but have zero lexical payload. The three-recipient extension changes the fact interfaces and the decoding schedule (Appendix G).

Table 2: Base architecture. The tokenizer and numerical checkpoint identifiers are supplied with the evidence package.

Quantity	Value	Quantity	Value
Trainable parameters	6,891,784	Vocabulary	8,192
Token/model dimension	256	Private-state dimension	192
Control dimension	128	Retrieval key dimension	128
Modules	4	Internal rounds per token	4
Reader heads per reader	2	Packet dimension	640
Local encoder layers	2	Attention heads per layer	4
Feed-forward dimension	512	Local attention window	32
Maximum records	16	Tokens per record	32
Maximum question tokens	64	Generation limit	32
Logit scale	$1/16$	Training arithmetic	FP32

The public grammar assigns the four local questions and two composition questions to their output owners. Person-to-location and location-to-person use E_2; person-to-object and object-to-person use E_1. Object-to-location uses E_2, while location-to-object uses E_1. This is a supplied task-type mapping.

C.3 INDEPENDENT LOCAL ENCODING AND COUPLED RETRIEVAL

Each record is flattened into an independent batch element before the local encoder. ALiBi positions restart within each segment. Contextual token states are $H_{av} \in \mathbb{R}^{256}$, for record a and token v. Learned pooling forms a record representation; learned maps form record keys k_a, token keys K_{av}, and contextual values V_{av}. A lexical value is instead a normalized raw embedding:

$$L_{av} = \sqrt{256}\,\frac{E(x_{av})}{\max(\|E(x_{av})\|_2, \epsilon)}. \tag{3}$$

The question/prefix is a separately encoded segment. Its last valid position provides u_t, while the fixed question anchor u_q gives $g = \tanh(W_g u_q)$.

Let $V_e(a) \in \{0, 1\}$ be the visibility mask. Each reader uses two queries and the following ordered record selector:

$$m^r_{eha} = V_e(a)\,\sigma\left((q^r_{eh})^\top k_a/\sqrt{128} + \beta_{eh}\right), \tag{4}$$

$$w^r_{eha} = m^r_{eha}\prod_{j>a}(1 - m^r_{ehj}), \qquad w^r_{eh,\text{NULL}} = \prod_a (1 - m^r_{eha}). \tag{5}$$

The last matching record is favored; the selector is not permutation invariant. A softmax over valid tokens within a visible record gives

$$\alpha^r_{ehav} = \mathrm{softmax}_{v\in\mathrm{valid}(a)}\left((q^r_{eh})^\top K_{av}/\sqrt{128}\right). \tag{6}$$

The same joint weights produce both outputs:

$$C^r_{eh} = \sum_{a,v} w^r_{eha}\alpha^r_{ehav}V_{av}, \qquad \ell^r_{eh} = \sum_{a,v} w^r_{eha}\alpha^r_{ehav}L_{av}. \tag{7}$$

Padding and invisible records have zero weight. The sum of record weights is $1 - w_{\text{NULL}}$, not necessarily one. A query intervention may change both the allocation among records and the total non-NULL mass; the results do not identify these as independent causes.

C.4 PRIVATE RECURRENCE, WRITES, AND THE VOCABULARY HEAD

The implemented query dependence can be written as

$$f_e^r = W_e^f[h_e^r; g; b^r], \qquad x_e^r = W_e^u[u_t; g], \tag{8}$$

$$q_e^r = \mathrm{reshape}_{2\times 128}\, W_e^q[h_e^r; f_e^r; u_t], \tag{9}$$

$$h_e^{r+1} = \mathrm{GRU}_e\left(\tanh(x_e^r + f_e^r + \rho_e^r), h_e^r\right), \tag{10}$$

where ρ_e^r is a projection of the two contextual reads for fact readers and zero for non-readers. Maps are module-specific, implemented in grouped form. For a reader, the control block c_e^r is a bounded projection of the contextual reads; for a non-reader it is a projection of the updated private state. Lexical blocks are zero for non-readers.

The write key is $k_e^r = \tanh(W_e^k h_e^{r+1})$. With $s^r = W_s[g; b_{\mathrm{ctrl}}^r]$, routing is

$$\pi^r = \mathrm{softmax}\left(\frac{(s^r)^\top k_0^r}{\sqrt{128}}, \ldots, \frac{(s^r)^\top k_3^r}{\sqrt{128}}, \eta(g, b_{\mathrm{ctrl}}^r)\right). \tag{11}$$

The final score belongs to NULL; its packet is zero. Source-packet interventions take place after the current write keys are computed, so they do not directly replace the current routing scores. They can alter future routing through recurrence.

For the public owner o, let a_t combine g and a projection of h_o^4. Vocabulary logits are

$$z_t = sE\ \mathrm{LN}\left[\phi(a_t) + \sigma(\lambda)W_u u_t + \frac{1}{2}\sum_{h=1}^{2}\tanh(\gamma_h(a_t))\ell_{oh}^{\mathrm{last}}\right], \quad s = 1/16. \tag{12}$$

The projection is tied to the vocabulary embedding. Only the owner's state and local lexical reads enter this head. The PAD embedding is constrained to zero. Teacher forcing supplies earlier answer tokens during training.

Algorithm 1 One native answer token in the base model

Require: Independently encoded records; question anchor; public owner o; generated prefix

1: Compute u_t, g; initialize $h_e^0 = 0$ and $b^0 = 0$
2: **for** $r = 0, 1, 2, 3$ **do**
3: Each module receives b^r; fact readers form queries and read only visible records
4: Compute coupled contextual/lexical reads and update private states
5: Construct module keys and control-plus-lexical packets
6: Compute competition with NULL and pool $b^{r+1} = \sum_e \pi_e^r p_e^r$
7: **end for**
8: Produce logits from the public features, owner's state, and owner's last lexical reads
9: Emit the next token; if generation continues, reset private/workspace states for the next token

D DATA, TRAINING, AND THE DISCOVERY–CONFIRMATION BOUNDARY

D.1 COMPLETE EXPERIMENTAL SETUP

This subsection gives the complete version of Section 4.

Relational worlds. A four-entity world contains four object–person bindings B and four person–location facts S, both bijections. There are six question types: four local types query B, S, or their inverses, and two composition types require $S \circ B$ (object → location) or $B^{-1} \circ S^{-1}$ (location → object). A family fixes the name inventories and fully crosses three binding permutations, three location permutations, two record orders, six question types, and four query sides, giving 432 questions per family.

Training. Each model starts from its own random initialization and trains in two stages. A local-task ancestor first trains for 2,048 updates with both readers seeing the full memory. A mixed branch then restores typed visibility and trains for another 2,048 updates on mixed local and composition batches, using a fresh AdamW optimizer (Loshchilov and Hutter, 2019). The effective batch size is 32, with

mixed batches split evenly between local and composition questions. The loss averages answer-token cross-entropy and adds 0.1 times the average EOS loss. Training one model takes about 22 minutes of measured step time, and all four confirmation models together take 1.46 hours (Appendix E).

Discovery, then frozen confirmation. We developed the query-restoration test on two models (seeds 104101 and 104102) trained on a fixed corpus and evaluated on a 128-case development panel that we inspected during development; we therefore report these results as discovery-stage evidence only. Before any new training, we fixed the architecture, the curriculum and learning-rate plans, the model and data seeds, the training endpoints, the primary intervention site, and the membership of every evaluation panel, and stored cryptographic hashes of all of them. This confirmation study (G4) generated two new corpora, A and B, each containing 8 training, 4 development, and 16 confirmation families (3,456, 1,728, and 6,912 questions, respectively). Each corpus trains two new initializations, giving the models A1, A2, B1, and B2. No complete fact world is shared across the six corpus/split combinations. The grammar, the 24-name inventory, and some individual fact edges do overlap, so the scope is recombination within the same generating process. All four final checkpoints existed before any neural evaluation on the confirmation split.

Panels. All 6,912 confirmation questions are evaluated natively for every model. The primary panel contains 512 symbolically eligible three-value recipients per model, namely 16 per composition direction in each of the 16 confirmation families. On it, we apply a transplant at write $r = 2$, followed by restoration or noise at the owner's $r = 3$ read. Secondary panels of 128 cases each cover earlier writes, the full write trajectory, and semantic controls, while separate OFF panels contain 512 composition and 512 local questions.

D.2 Paired relational worlds

For n entities, each family fixes inventories of n people, objects, and locations. Binding B and location S are bijections. Three permutations of each relation, two record orders, six question types, and all n query sides yield $3 \times 3 \times 2 \times 6 \times n = 108n$ cases per family. Four local types query a relation or its inverse; the two composition types query $S \circ B$ or its inverse. The public forms include `[bind] The holder of OBJECT is PERSON.` and `[state] PERSON is in the LOCATION.` Records contain static facts.

Eight training families and four families in each development/test split give 3,456/1,728/1,728 four-entity cases. The corresponding eight-entity counts are doubled. Semantic world duplication is checked by the generator. Entity strings include multi-token values; exact match includes the complete sequence.

The mechanism panel is separately identified as the discovery-stage four-entity mechanism development panel. Four families each contribute 32 eligible source/recipient combinations, equally divided overall between the two composition directions. Eligibility depends on symbolic targets X, Y, Z being distinct, not on model predictions. An additional 128 local questions support OFF and irrelevant-source controls. Case identifiers organize files and joins but are not embedded into the model. Truth records, support coordinates used for analysis, and pair maps are evaluation-only data.

D.3 Base-model curriculum and accounting

For each of seeds 104101 and 104102, a separately initialized model trains its own local-task ancestor for 2,048 updates. In this base-model curriculum both fact readers initially have full-bank visibility and private feedback. The shared mixed branch starts from that ancestor, restores typed visibility, and trains for another 2,048 updates with a fresh optimizer. The two initializations use the same generated training corpus and public batch-plan construction and they are independent initialization repetitions.

The effective batch is 32, accumulated from microbatches of eight. Mixed updates contain sixteen local cases and sixteen compositions, balanced across task types. The objective is

$$\mathcal{L} = \frac{1}{N_v} \sum_{t \in \mathcal{V}} -\log P(y_t \mid x, y_{<t}) + \frac{0.1}{N_e} \sum_{t \in \mathcal{E}} -\log P(y_t \mid x, y_{<t}), \tag{13}$$

with value and EOS denominators computed across the effective batch. Optimization uses AdamW (Loshchilov and Hutter, 2019), $(\beta_1, \beta_2) = (0.9, 0.95)$, $\epsilon = 10^{-8}$, weight decay 0.01, norm clipping

at one, and FP32 arithmetic. The mixed-stage peak learning rate is 10^{-4}, with sixteen warmup updates and cosine decay to 10^{-5}:

$$\mathrm{lr}(u) = \begin{cases} 10^{-4}u/16, & u \le 16, \\ 10^{-4}\left[0.1 + 0.9\,\dfrac{1+\cos(\pi(u-16)/2032)}{2}\right], & u > 16. \end{cases} \tag{14}$$

The local stage uses the recorded extended schedule with peak 3×10^{-4}: its first 256 updates preserve the earlier local schedule (cosine denominator 496), followed by continuous decay to 3×10^{-5}. The supplied plan-generation snapshots reference the original serialized rate plans; the local schedule should not be silently replaced with a newly chosen cosine schedule.

Measured training-step time across both stages was 1,238.93 and 1,243.99 seconds for the two base repetitions. Stage wall times including recorded overhead were 1,402.58 and 1,426.55 seconds, and peak allocated memory was 950,874,112 bytes. These are recorded implementation-specific measurements. The evidence manifest identifies the environments and checkpoints.

D.4 Discovery–confirmation boundary

Architectural choices, including lexical access, per-token resets, the curriculum, routing, and output scale, were developed on earlier two-entity studies (seed 59101) and on four-entity studies whose diagnostic splits we inspected repeatedly. The query-restoration test was identified on the 128-case development panel of seeds 104101 and 104102. We therefore treat every result from these models as discovery-stage evidence, and all confirmatory claims rest on G4 (Appendix E), for which every design and evaluation choice was fixed before training. Fixed endpoints and complete denominators throughout prevent selecting a checkpoint or a subset by its intervention effect.

E Frozen cross-corpus confirmation (G4)

E.1 Extended results by rung

This subsection gives the complete version of the results that Section 5 summarizes.

Rung 1: dependence. All four confirmation models fit all 3,456 training questions. On the held-out confirmation worlds, composition accuracy ranges from 64.02% to 75.48% and local accuracy from 79.10% to 86.31% (Table 3). Deleting the broadcast (OFF) reduces composition by 41.41–51.56 percentage points while changing local accuracy by only 0.39–1.37 points. The composition losses are not a net result of offsetting changes: 243–277 composition answers per model flip from correct to wrong, against only 13–31 that flip from wrong to correct. Generalization exits. The discovery-stage models reach 472/576 and 437/576 on their test worlds and 729/1,152 and 705/1,152 when tested at eight entities without retraining. By Proposition 1, the OFF deficit was expected once composition was learned. It confirms that information flows through the workspace, and the next question is what flows.

Table 3: Capability and broadcast dependence in the confirmation study. Full-confirmation composition uses 2,304 cases per model; each ON/OFF group uses its own fixed 512-case panel.

Run	Full-confirmation composition EM	Comp. ON	Comp. OFF	Local ON	Local OFF
A1	1,499/2,304 (65.06%)	332/512	117/512	417/512	410/512
A2	1,475/2,304 (64.02%)	332/512	120/512	394/512	390/512
B1	1,651/2,304 (71.66%)	360/512	129/512	432/512	430/512
B2	1,739/2,304 (75.48%)	404/512	140/512	446/512	441/512

Rung 2: content. Transplanting the binding reader's packet at write $r = 2$ yields the hybrid answer Z in 312, 317, 368, and 374 of 512 cases for A1, A2, B1, and B2, respectively (60.9–73.0%). The unperturbed recipient already outputs Z by accident in only 65, 56, 44, and 25 cases. These are not relays of the donor's answer, since Y differs from Z in every case by construction. Moreover, the transplant does something stronger than produce the right symbol: the transplanted recipient

reproduces the same model's native output on the hybrid world in 90.2–95.7% of cases (469, 487, 490, and 462 of 512), errors included. In effect, the receiver behaves as though it had been handed the hybrid world. Together, the two metrics also clarify what the remaining target failures are. Every correct agreement with the native hybrid output is also a correct target answer, so at least $a - t$ of the a agreements must reproduce a wrong native hybrid answer (Appendix H.1). Applied to our counts, at least 157, 170, 122, and 88 of the 200, 195, 144, and 138 target failures in A1, A2, B1, and B2 are exact reproductions of the error the same model makes when natively given the hybrid world. That is 63.8–87.2% of failures per model, or 537 of 677 (79.3%) pooled descriptively. Most of the shortfall from perfect target accuracy thus reflects the model's own competence on the hybrid world, which the transplant faithfully inherits, rather than a failure of the transplanted content to take effect. This is a lower bound derived from reported counts, and output agreement alone does not identify a unique internal representation.

The effect is carried by content rather than by perturbation, and it resides in the expected part of the packet. At the discovery stage, packet noise matched to the transplant's per-block norm yields only 11–16 of 128 hybrid answers. Transplanting only the two lexical blocks reproduces nearly the whole effect (102/128 and 94/128), whereas transplanting only the control block yields 24/128 and 25/128. In the confirmation study, self-replacement and the unconsumed $r = 3$ write leave every tested output unchanged. Transplants at earlier writes have smaller effects: 16–27/128 at $r = 0$ and 31–41/128 at $r = 1$, compared with 81–96/128 for the full trajectory $\{0, 1, 2\}$ (Figure 6). This pattern is consistent with the final read consuming content written late in the computation.

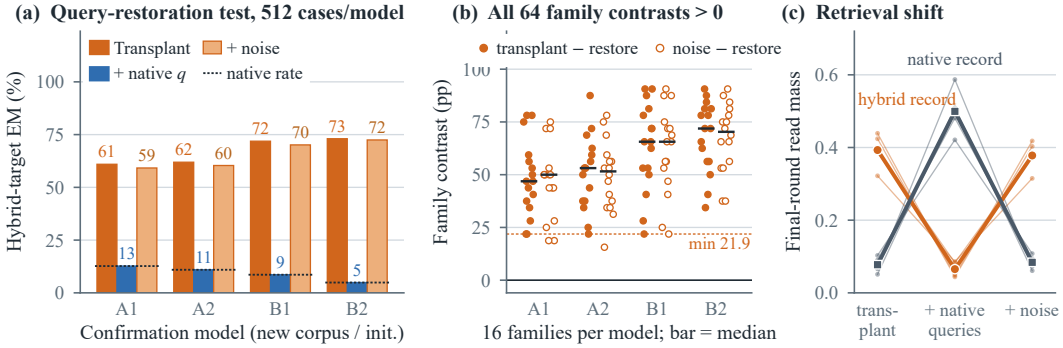

Figure 5: **Frozen confirmation of the retrieval route.** (a) The source packet is transplanted at write $r = 2$; the other two conditions additionally restore, or perturb with matched norm, the receiver's $r = 3$ queries. Each bar covers all 512 preselected cases per model, including native mistakes; numbers are percentages, and dotted lines show the native rate of accidental hybrid answers. (b) Family-level contrasts in hybrid-target exact match, one point per family (32 related recipients); bars mark medians and the dashed line the smallest transplant-minus-restoration contrast. (c) Final-round raw read mass on the record needed for the hybrid answer and on the native record, averaged over both heads and all 512 cases with NULL mass retained; thin lines are models, thick lines their mean.

Rung 3: restoration. Restoring the owner's native queries while keeping the transplant active reduces hybrid answers to 65, 56, 44, and 25 (Figure 5a), which are exactly the native accidental rates. The paired reductions are 48.24, 50.98, 63.28, and 68.16 percentage points. Restoration also recovers the complete native output sequence in 512, 512, 512, and 511 of 512 cases, or 2,047 of 2,048 overall. Because the base model resets its private and workspace states at every token and our interventions act only on the first answer token, recovering the first token implies recovering the entire deterministic continuation (Appendix H.2). We therefore read this figure as a decoding-level check of the first-token result, not as independent evidence that workspace content persists across tokens. The clearest view comes from the natively correct recipients (332–404 per model): among them, the transplant produces 206–304 hybrid answers, and restoration leaves zero in every model. By contrast, matched-norm query noise applied at the same sites retains 303, 309, 359, and 371 hybrid answers, that is, 97.1–99.2% of the transplant effect. Because restoration leaves route (b) open, the transplanted broadcast still enters the owner's final state directly. The near-total recovery of native outputs therefore shows that this direct route cannot produce the answer change on its own: redirected retrieval is necessary for the effect. Sufficiency is tested by the query-only condition (Appendix H.3).

Retrieval. Table 4 and Figure 5c show the corresponding change in reading mass. Transplantation moves the owner's final-round reading toward the record implied by the hybrid world, restoration reverses this shift, and noise does not. The mass on the native record moves in the opposite direction, from 0.05–0.10 under transplantation to 0.42–0.59 under restoration. All values average both heads over all 512 cases and keep NULL mass in the denominator.

Table 4: Final-round raw read mass on the record needed for the hybrid answer. The transplant remains active in all three columns.

Run	Transplant	+ Native queries	+ Query noise
A1	0.322	0.086	0.315
A2	0.387	0.083	0.376
B1	0.439	0.048	0.418
B2	0.424	0.044	0.403

Robustness. The effect does not come from a few favorable families. Transplant-minus-restoration is positive in all 64 family–initialization combinations (32 corpus-specific families, each with two initializations), with a smallest family-level contrast of 21.9 points (Figure 5b). Noise-minus-restoration is likewise positive in all 64, with a minimum of 15.6 points. Treating families as the unit, a one-sided sign test gives $p = 2^{-32} < 10^{-9}$. The single case in which restoration does not recover the native sequence is preserved in Appendix E.4. There, the native truth and output are *coat*, the donor truth is *cup*, and the hybrid target is *phone*; the transplant and noise conditions both produce *phone*, whereas restoration produces *notebook*.

From discovery to confirmation. On the discovery panel, transplantation gave 102/128 and 95/128 hybrid answers, restoration returned all 256 outputs to their native sequences, and noise retained 97/128 and 91/128. The confirmation study reproduces this pattern on a four-times-larger panel drawn from unseen corpora and new initializations. Effect sizes are somewhat smaller than at discovery (60.9–73.0% versus 74.2–79.7%), which is expected given that the discovery panel was inspected during development.

E.2 Registration, data construction, and training units

Before any G4 training, we fixed two data-generator seeds and four model seeds, the unchanged 6,891,784-parameter architecture, the two-stage curriculum and its learning-rate plans (copied elementwise from the discovery runs), the training endpoints, the primary intervention site ($r = 2$ write, $r = 3$ read), and the membership of all primary and secondary panels, and we stored hashes of all of these. No confirmation-model evaluation was used to change any of them, and development-split scores served only as diagnostics. The protocol is locally hash-committed rather than publicly preregistered. The one execution difference from the discovery runs is recorded: microbatches of 32 instead of 8, at the same effective batch size.

Corpus A uses generator seed 20260923401, with model seeds 20260923411 (A1) and 20260923412 (A2). Corpus B uses generator seed 20260923402, with model seeds 20260923421 (B1) and 20260923422 (B2). Each corpus contains 8 training, 4 development, and 16 confirmation families. With 432 questions per family, the split sizes are 3,456, 1,728, and 6,912. Each confirmation split contains 4,608 local and 2,304 composition questions. World signatures exclude record order and question identity; there is no identical complete fact world across the six corpus/split combinations. The grammar, 24-name inventory, and some atomic edges overlap. The data audit therefore supports new-world recombination under the same generating setup.

Each of the four lineages starts from its own random initialization, trains a full-information private-feedback LOCAL ancestor for 2,048 updates, and trains its typed-visibility SHARED MIXED branch for another 2,048 updates with a fresh optimizer. The discovery-stage learning rates are copied elementwise. The effective batch is 32, the mixed batch is half local and half composition, and the loss, optimizer, precision, and stopping rule are as specified in Appendix D. G4 executes microbatches of 32, whereas the discovery-stage runs used eight; this is recorded as an execution detail. Four lineages entail 16,384 optimizer updates, not eight independent models merely because each has two stages.

All four final MIXED checkpoints existed before neural evaluation on CONFIRM. Development scores were diagnostic only. The stored initial/final tensor digests and ancestor file hashes preserve each LOCAL-to-MIXED chain.

Table 5: Complete G4 endpoint capability. Every value is complete-answer exact match, including EOS. The full CONFIRM set is distinct from the selected 512-case intervention panel.

Run	Split	Local exact match	Composition exact match
A1	Train	2,304/2,304 (100.00%)	1,152/1,152 (100.00%)
A1	Dev	998/1,152 (86.63%)	421/576 (73.09%)
A1	Confirm	3,743/4,608 (81.23%)	1,499/2,304 (65.06%)
A2	Train	2,304/2,304 (100.00%)	1,152/1,152 (100.00%)
A2	Dev	999/1,152 (86.72%)	416/576 (72.22%)
A2	Confirm	3,645/4,608 (79.10%)	1,475/2,304 (64.02%)
B1	Train	2,304/2,304 (100.00%)	1,152/1,152 (100.00%)
B1	Dev	992/1,152 (86.11%)	422/576 (73.26%)
B1	Confirm	3,938/4,608 (85.46%)	1,651/2,304 (71.66%)
B2	Train	2,304/2,304 (100.00%)	1,152/1,152 (100.00%)
B2	Dev	1,013/1,152 (87.93%)	433/576 (75.17%)
B2	Confirm	3,977/4,608 (86.31%)	1,739/2,304 (75.48%)

E.3 Fixed panels and complete primary metrics

Before training, the primary panel selects 16 eligible cases per composition direction within each of the 16 confirmation families, totaling 512 recipients per model. Eligibility is defined by the symbolic $X/Y/Z$ distinction, not by any model output. The two initializations of a corpus use the same panel. The primary write is $r = 2$ and the intervened recipient read is $r = 3$, both on the first generated answer token. The separate OFF panels contain 512 composition and 512 local questions. Secondary write sets $\{0\}$, $\{1\}$, and $\{0, 1, 2\}$ and their query controls each retain 128 predetermined cases, as do the specified semantic controls.

The fixed G4 query-noise seed is 20260923451. A deterministic hash of the lineage, case, round, and head gives each random direction. Its magnitude equals the actual live-query-to-native-query displacement in the paired restoration execution. This uses a different seed from the earlier localization experiment but the same intervention definition. It is a same-site, same-norm control.

Table 6: G4 primary target counts. All columns except the last use 512 cases. "Native at Z" counts native recipient errors that already equal the hybrid target. S denotes source replacement, Q additionally restores native queries, and N additionally perturbs queries.

Run	Native at Z	S target	Q target	N target	$S - Q$ (pp)
A1	65/512	312/512	65/512	303/512	48.24
A2	56/512	317/512	56/512	309/512	50.98
B1	44/512	368/512	44/512	359/512	63.28
B2	25/512	374/512	25/512	371/512	68.16

Table 7: Different questions require different metrics. Source-target success measures correctness; source/hybrid-output agreement can reproduce a wrong native hybrid answer; query-restoration/native agreement measures recovery of a sequence. Each denominator is 512.

Run	Source target correct	Source / hybrid agreement	Query / native agreement
A1	312/512	469/512	512/512
A2	317/512	487/512	512/512
B1	368/512	490/512	512/512
B2	374/512	462/512	511/512

Table 8: Secondary native-correct analysis of the same primary panel. Eligibility requires the recipient to answer natively correctly. Coverage is explicit and does not replace the full 512-case denominator.

Run	Coverage	Source target	+ Native queries	+ Query noise
A1	332/512	207/332	0/332	201/332
A2	332/512	206/332	0/332	197/332
B1	360/512	266/360	0/360	259/360
B2	404/512	304/404	0/404	299/404

The family-level contrasts in Table 9 retain dependence among questions, layouts, and counterpart worlds. Both contrasts are positive in every family–initialization combination. There are 32 corpus-specific families, each evaluated using two trained initializations.

Table 9: G4 family-level differences in hybrid-target exact match (percentage points). Each entry is based on 32 related recipients. Within each run, $S - Q$ is source minus query restoration and $N - Q$ is matched query noise minus restoration. The same family number in corpora A and B does not identify the same inventory/worlds.

	A1		A2		B1		B2	
Family	$S-Q$	$N-Q$	$S-Q$	$N-Q$	$S-Q$	$N-Q$	$S-Q$	$N-Q$
f00	28.12	28.12	50.00	46.88	87.50	87.50	78.12	75.00
f01	40.62	43.75	37.50	37.50	65.62	68.75	71.88	68.75
f02	78.12	75.00	87.50	78.12	40.62	40.62	65.62	65.62
f03	21.88	18.75	53.12	50.00	53.12	53.12	62.50	65.62
f04	78.12	71.88	68.75	75.00	90.62	87.50	81.25	81.25
f05	46.88	50.00	37.50	34.38	21.88	21.88	40.62	37.50
f06	75.00	71.88	71.88	68.75	50.00	46.88	34.38	37.50
f07	50.00	50.00	53.12	53.12	90.62	90.62	84.38	84.38
f08	53.12	50.00	53.12	56.25	62.50	56.25	50.00	53.12
f09	59.38	50.00	62.50	59.38	81.25	75.00	90.62	90.62
f10	21.88	18.75	25.00	31.25	28.12	25.00	56.25	56.25
f11	37.50	43.75	59.38	53.12	65.62	65.62	78.12	75.00
f12	46.88	50.00	43.75	40.62	71.88	65.62	81.25	78.12
f13	43.75	43.75	56.25	56.25	71.88	71.88	87.50	87.50
f14	56.25	53.12	34.38	34.38	53.12	56.25	56.25	53.12
f15	34.38	25.00	21.88	15.62	78.12	71.88	71.88	71.88

Table 10: G4 final-round reading, averaged over both heads and all primary cases. Raw masses preserve NULL; the last column is the average per-read hybrid-record mass conditional on a non-NULL read, not the ratio of aggregated means. These are retrieval measurements.

Run	Condition	Native record	Hybrid record	NULL	Conditional hybrid
A1	Source	0.103	0.322	0.425	0.542
A1	+ Native queries	0.421	0.086	0.346	0.143
A1	+ Query noise	0.108	0.315	0.420	0.526
A2	Source	0.090	0.387	0.350	0.593
A2	+ Native queries	0.481	0.083	0.264	0.123
A2	+ Query noise	0.094	0.376	0.350	0.579
B1	Source	0.051	0.439	0.406	0.679
B1	+ Native queries	0.509	0.048	0.316	0.101
B1	+ Query noise	0.061	0.418	0.408	0.656
B2	Source	0.066	0.424	0.425	0.695
B2	+ Native queries	0.587	0.044	0.245	0.068
B2	+ Query noise	0.073	0.403	0.433	0.677

E.4 THE NATIVE-RECOVERY EXCEPTION

In B2, case `CONFIRM/f02/b0s2/o1/place_to_item/q2` is the sole primary restoration output that differs from the native sequence. Native truth and native output are `coat`; donor truth is `cup`; hybrid truth is `phone`. Source replacement and matched query noise both produce `phone`, but source replacement plus native-query restoration produces `notebook`. The total is 2,047/2,048 native sequence recoveries.

E.5 BROADCAST DELETION

Table 11: G4 OFF transitions. Each task panel has 512 cases. Small net local changes can hide individual correct-to-wrong and wrong-to-correct transitions.

Run	Task	ON correct	OFF correct	Correct→wrong	Wrong→correct	Unchanged
A1	Composite	332/512	117/512	243/512	28/512	172/512
A1	Local	417/512	410/512	23/512	16/512	472/512
A2	Composite	332/512	120/512	243/512	31/512	180/512
A2	Local	394/512	390/512	22/512	18/512	468/512
B1	Composite	360/512	129/512	244/512	13/512	210/512
B1	Local	432/512	430/512	14/512	12/512	481/512
B2	Composite	404/512	140/512	277/512	13/512	179/512
B2	Local	446/512	441/512	14/512	9/512	482/512

Table 12: All fixed G4 secondary write and control conditions. Every cell has denominator 128. Targets are the registered hybrid Z, including accidental-target scoring for native, self, and unconsumed-write rows; same-facts and irrelevant-source controls instead use the unchanged native truth. Their agreement reference is correspondingly the native recipient. Low self-control target counts therefore do not imply failure to preserve the native output.

Run	Condition	Target correct	Native unchanged	Hybrid agreement
A1	Native secondary	17/128	128/128	26/128
A1	Source $r = 0$	20/128	116/128	28/128
A1	$r = 0$ + native queries	17/128	128/128	26/128
A1	$r = 0$ + query noise	19/128	114/128	29/128
A1	Source $r = 1$	31/128	90/128	45/128
A1	$r = 1$ + native queries	17/128	128/128	26/128
A1	$r = 1$ + query noise	34/128	89/128	45/128
A1	Source trajectory	88/128	26/128	128/128
A1	Trajectory + native queries	17/128	128/128	26/128
A1	Trajectory + query noise	87/128	26/128	115/128
A1	Same facts, donor order	82/128	96/128	96/128
A1	Irrelevant source (local)	93/128	125/128	125/128
A1	Self-query restoration	17/128	128/128	26/128
A1	Source self-replacement	17/128	128/128	26/128
A1	Unconsumed write $r = 3$	17/128	128/128	26/128
A2	Native secondary	14/128	128/128	33/128
A2	Source $r = 0$	27/128	108/128	44/128
A2	$r = 0$ + native queries	14/128	128/128	33/128
A2	$r = 0$ + query noise	25/128	109/128	45/128
A2	Source $r = 1$	36/128	82/128	64/128
A2	$r = 1$ + native queries	14/128	128/128	33/128
A2	$r = 1$ + query noise	39/128	79/128	63/128
A2	Source trajectory	81/128	33/128	128/128
A2	Trajectory + native queries	14/128	128/128	33/128
A2	Trajectory + query noise	83/128	33/128	123/128
A2	Same facts, donor order	76/128	103/128	103/128
A2	Irrelevant source (local)	93/128	128/128	128/128
A2	Self-query restoration	14/128	128/128	33/128

Continued on next page

Table 12 continued

Run	Condition	Target correct	Native unchanged	Hybrid agreement
A2	Source self-replacement	14/128	128/128	33/128
A2	Unconsumed write $r = 3$	14/128	128/128	33/128
B1	Native secondary	11/128	128/128	24/128
B1	Source $r = 0$	18/128	114/128	33/128
B1	$r = 0$ + native queries	11/128	128/128	24/128
B1	$r = 0$ + query noise	20/128	113/128	32/128
B1	Source $r = 1$	39/128	84/128	55/128
B1	$r = 1$ + native queries	11/128	128/128	24/128
B1	$r = 1$ + query noise	42/128	79/128	59/128
B1	Source trajectory	96/128	24/128	128/128
B1	Trajectory + native queries	11/128	128/128	24/128
B1	Trajectory + query noise	92/128	23/128	116/128
B1	Same facts, donor order	95/128	109/128	109/128
B1	Irrelevant source (local)	99/128	127/128	127/128
B1	Self-query restoration	11/128	128/128	24/128
B1	Source self-replacement	11/128	128/128	24/128
B1	Unconsumed write $r = 3$	11/128	128/128	24/128
B2	Native secondary	7/128	128/128	15/128
B2	Source $r = 0$	16/128	111/128	25/128
B2	$r = 0$ + native queries	7/128	128/128	15/128
B2	$r = 0$ + query noise	17/128	110/128	25/128
B2	Source $r = 1$	41/128	77/128	57/128
B2	$r = 1$ + native queries	7/128	128/128	15/128
B2	$r = 1$ + query noise	40/128	76/128	54/128
B2	Source trajectory	96/128	14/128	126/128
B2	Trajectory + native queries	7/128	128/128	15/128
B2	Trajectory + query noise	96/128	13/128	116/128
B2	Same facts, donor order	100/128	115/128	115/128
B2	Irrelevant source (local)	111/128	127/128	127/128
B2	Self-query restoration	7/128	128/128	15/128
B2	Source self-replacement	7/128	128/128	15/128
B2	Unconsumed write $r = 3$	7/128	128/128	15/128

Figure 6 summarizes the write-site conditions of Table 12 and compares the primary contrast at discovery and confirmation.

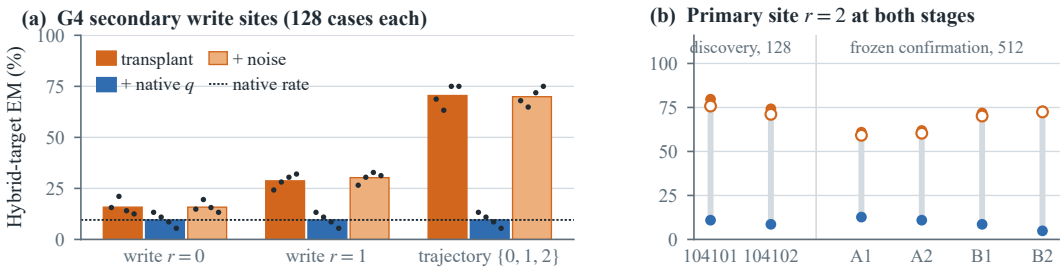

Figure 6: **Where the transplanted content acts.** (a) G4 secondary panels (128 cases per model): transplants at earlier writes have smaller effects than the full trajectory, and native-query restoration returns every site to the native rate (dotted). Bars are means over A1–B2; dots are models. (b) The primary $r = 2$ contrast on the inspected discovery panel (128 cases per seed) and in the frozen confirmation (512 cases per model). Filled orange: transplant; hollow: + matched query noise; blue: + native queries.

E.6 Learning curves, resource accounting, and recorded recovery

Figure 7 plots the predetermined composition subset of the mixed-stage training curves. The final endpoint metrics in Table 5 evaluate the entire training and confirmation sets.

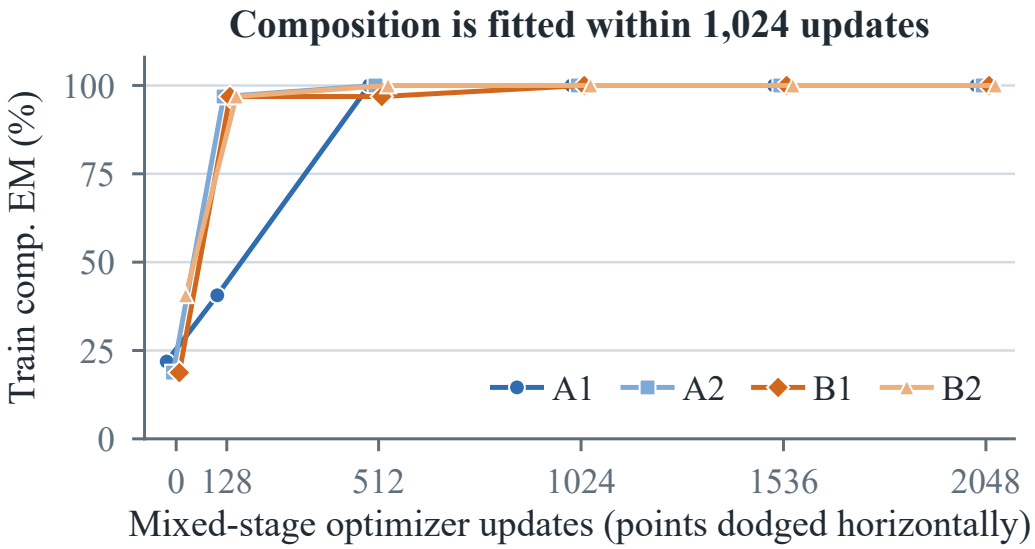

Figure 7: **G4 training** Mixed-stage composition on fixed training-curve subsets. Every run still completes all 2,048 updates; points are dodged horizontally so that coinciding runs remain visible.

Table 13: G4 measured training-step time and peak PyTorch memory across LOCAL and MIXED. Each run has 4,096 optimizer updates and 6,891,784 parameters. Memory is allocated/reserved tensor memory

Run	Training minutes	Peak allocated (MiB)	Peak reserved (MiB)
A1	21.85	905.5	1032.0
A2	21.47	905.5	1032.0
B1	21.96	907.8	1056.0
B2	22.06	907.8	1056.0

The training-time total is 5,240.48 seconds (1.456 hours). Runtime records separately report evaluation, generated episodes, and wall time. The producing pipeline also reports an isolated-process inference replay using four final checkpoints and 32 preselected cases per checkpoint. All 128 native/source/restoration/noise sequence comparisons per checkpoint agree; this is checkpoint-level inference validation.

F Intervention definitions and metrics

F.1 Extended protocol description

This subsection gives the parts of Section 3 that the main text abbreviates.

Proof sketch of Proposition 1. With $b^r \equiv 0$, the owner's query at round r depends only on (h_o^r, u_t, g). Its reads depend only on that query and on the owner's bank, whose records are encoded independently of every other record. Its update depends only on h_o^r, u_t, g, and those reads. By induction from $h_o^0 = 0$, both h_o^4 and the owner's final lexical reads are functions of $(u_t, g, \mathcal{B}_o)$, and the head uses only these quantities. Since u_t is computed from the question and the previously generated tokens, induction over tokens extends the claim to the entire output sequence. Once the model learns to compose, a large OFF deficit is therefore guaranteed. OFF certifies that the trained model routes information through the workspace.

A three-value case. Table 14 shows the real case from the primary panel that Figure 3 traces through the model. Reverse-direction questions exchange the binding and location roles.

Table 14: A three-value case from the primary panel: "Where is football?" Only the relevant binding and two location facts are shown. Entries are symbolic truths.

World	Binding	Location facts	Truth
Recipient (B_0, S_1)	football $\mapsto$ Jeff	Jeff $\mapsto$ room; Bill $\mapsto$ hospital	X: room
Donor (B_1, S_0)	football $\mapsto$ Bill	Jeff $\mapsto$ hospital; Bill $\mapsto$ kitchen	Y: kitchen
Hybrid (B_1, S_1)	football $\mapsto$ Bill	Jeff $\mapsto$ room; Bill $\mapsto$ hospital	Z: hospital

Content controls. Transplanted packets are the donor's actual native activations, so no evaluation code ever inserts a correct entity. Current routing scores are not overwritten, and the downstream dynamics evolve freely. We complement the transplant with five controls: self-replacement; a donor that holds the same facts in a different record order; an irrelevant-source change that alters only the bank a local question does not need; blockwise, norm-matched packet noise; and a transplant at the unconsumed write $r = 3$.

Route analysis. The value of the restoration test comes from the model's enforced boundaries, which let us enumerate the relevant routes exactly. In the primary intervention (write $r = 2$, first answer token), the transplanted packet enters the computation only through b^3, from which it can reach the output through exactly two routes. Along route (a), b^3 changes the owner's $r = 3$ queries and hence its reads, which enter both the output head (through the lexical reads) and the owner's final state update. Along route (b), b^3 bypasses retrieval and enters the owner's final state update h_o^4 directly, from which it feeds the head. The other modules' $r = 3$ updates only produce b^4, which nothing consumes. Restoration severs route (a) while leaving route (b) intact, which makes the test asymmetric in an informative way. If the direct route could carry the transplanted content into the answer on its own, the effect would survive restoration; if the effect disappears, retrieval is necessary for it. Restoration alone cannot, however, establish that redirected retrieval is sufficient, because the two routes could interact (Appendix H.3).

Limits of the mechanism test. Because contextual and lexical reads share queries, the test does not separate them. Equal Euclidean norm controls the size of a query perturbation but not its effect on record scores, which depends on the perturbation's direction relative to the record keys, so we treat the noise condition as a magnitude control rather than as an exact counterpart to restoration. Because restoration blocks every influence converging on the query, it establishes that retrieval is necessary for the effect rather than decomposing the effect into natural indirect effects.

Metrics. We report three distinct quantities, each computed over the full preselected panel, including cases that the model answers wrongly without intervention. Target accuracy is exact match to Z followed by EOS. Hybrid agreement is the fraction of outputs identical to the same model's native output on the hybrid world (B_1, S_1), errors included; it therefore asks whether the receiver behaves as if it lived in the hybrid world. Native recovery is the fraction of outputs identical to the unperturbed recipient output. In all cases, exact match covers the complete multi-token sequence and EOS.

F.2 Worlds, packet sites, and exact targets

For a forward composition, write the baseline world (B_0, S_0), source donor (B_1, S_0), recipient (B_0, S_1), and hybrid (B_1, S_1). Let the recipient, donor, and hybrid truth values be X, Y, Z. The primary localization panel enforces pairwise distinctness. For reverse composition, exchange the source and recipient relation roles. The same symbolic interpreter recomputes every target from public records. Source packets are taken from native donor activations; an evaluation script never inserts the correct person or support record into the model.

For the base model, the packet hook selects the source module, token index zero, and one or more writes in $\{0, 1, 2\}$. It replaces the entire 640-dimensional packet after key construction and before pooling. Other packets and current routing scores are left live. Later model states evolve naturally. Once the first answer token is emitted, generation proceeds without this source hook. In contrast, OFF zeros the entire dynamic receiving tensor at every core round of every generated token, including EOS.

F.3 Native-query restoration and its noise control

Formally, native-query restoration overwrites the receiver's read queries at every round after the earliest intervened write, while reading remains live:

$$q^k_{oh} \leftarrow q^k_{oh,\mathrm{native}}, \qquad k > \min \mathcal{W}, \quad h \in \{1, 2\}, \tag{15}$$

where $\mathcal{W}$ is the set of intervened writes. For the primary site ($\mathcal{W} = \{2\}$), only the final read at $k = 3$ is restored.

For each recipient case, native read queries are first recorded from the same checkpoint and same recipient input. In the source-patched run, both owner queries at all rounds after the earliest patched write are restored to these recorded queries. For write $r = 2$, only the final read at $r = 3$ is restored. Queries are changed at the retrieval interface; private states, bank records, source packets, and the output head remain live. This differs from setting retrieval to zero or replacing selected record indices.

Let a^k_h be the query immediately before the restoration hook in the executed restoration branch and $q^k_{h,\mathrm{native}}$ its replacement. The matched-noise branch uses its own live query $\widetilde{q}^k_h$ and a seeded unit direction $\widehat{\epsilon}^k_h$:

$$q^k_{h,\mathrm{noise}} = \widetilde{q}^k_h + \|q^k_{h,\mathrm{native}} - a^k_h\|_2 \widehat{\epsilon}^k_h. \tag{16}$$

The displacement norm is measured separately for each example, round, and head from the actual restoration execution. Source replacement remains active in both branches. Zero reference displacements produce no added noise. The discovery-stage query-noise seed is 20260923421, combined deterministically with case/site identifiers. G4 instead fixes seed 20260923451 and includes the lineage in the hash (Appendix E).

This intervention changes both contextual and lexical reads through shared queries and joint weights. It does not separately identify either pathway. It also intercepts source effects that reach the query through private state. The analysis localizes a controlled dependence in this implementation.

F.4 Routes from the final consumed broadcast to the output

The value of this test comes from the model's enforced boundaries, which let us enumerate the relevant routes exactly. In the primary intervention (write $r = 2$, first answer token), the transplanted packet enters the computation only through b^3, from which it can reach the output through exactly two routes. Along route (a), b^3 changes the owner's $r = 3$ queries and hence its reads, which enter both the output head (through the lexical reads) and the owner's final state update. Along route (b), b^3

bypasses retrieval and enters the owner's final state update h_o^4 directly, from which it feeds the head. The other modules' $r = 3$ updates only produce b^4, which nothing consumes. Restoration severs route (a) while leaving route (b) intact, so the hypothesis makes a sharp prediction: if content acts through retrieval, restoration should abolish the transplant effect, whereas if the broadcast instead writes the answer into the owner's state, the effect should survive restoration.

F.5 Content and implementation controls

Self replacement tests identity under the same hooks. Same-facts/different-order donors preserve relational semantics but change the ordered reader's input. Irrelevant-source local controls preserve the facts needed by the local question while changing the other bank. All are evaluated with their own matched question.

Packet noise is distinct from query noise. It replaces a source perturbation with random directions separately in the control block $[0, 128)$, lexical head 1 $[128, 384)$, and lexical head 2 $[384, 640)$. Norms match the actual prepatch-to-donor displacement in the semantic branch. Three packet-noise seeds, 20260923271–20260923273, are retained. The three block-isolation conditions replace only control, one lexical block, or both lexical blocks; whole-packet and whole-trajectory conditions are reported alongside them.

F.6 Metrics and dependent observations

Let $\hat{y}_i^I$ be the full sequence after intervention I, z_i the symbolic hybrid target, $\hat{y}_i^N$ the native recipient output, and $\hat{y}_i^H$ the native hybrid-world output. We distinguish

$$A_{\text{target}}(I) = \frac{1}{N} \sum_i \mathbf{1}[\hat{y}_i^I = z_i \text{ followed by EOS}], \tag{17}$$

$$C_{\text{hybrid}}(I) = \frac{1}{N} \sum_i \mathbf{1}[\hat{y}_i^I = \hat{y}_i^H], \qquad U(I) = \frac{1}{N} \sum_i \mathbf{1}[\hat{y}_i^I = \hat{y}_i^N]. \tag{18}$$

Native correctness is evaluated against the native world's truth. A wrong native answer can already equal z_i, so target hits need not be directed changes. Conversely, perfect hybrid-output agreement can reproduce the model's wrong hybrid answer. Secondary native-correct subsets are labeled with coverage; no such subset replaces the primary denominator.

Read mass is measured on the owner-local record required by the native or hybrid interpretation. Raw record mass is reported together with NULL mass and conditional mass after dividing by total non-NULL mass for each read. A mean of these conditional quantities is not generally the ratio of two aggregated raw means.

We report initialization-level and family-level counts. A family contains shared names and many related worlds, questions, orders, and counterfactual directions. Existing grouped analyses are descriptive of the frozen panels. The discovery-stage and reuse studies each have two initializations on one corpus. G4 adds two new corpora with two initializations each.

G Three-receiver variant

G.1 Main results of the three-receiver variant

GWT's broadcast is meant to be globally available. To test this property directly, we trained a separate variant with one source module that reads object $\rightarrow$ person bindings and three receivers that read person $\rightarrow$ location, person $\rightarrow$ color, and person $\rightarrow$ access mappings. The four core rounds run once on the shared question, after which each receiver decodes its answer from a private cache. No receiver's answer tokens ever re-enter the shared prefix or reach another receiver, which departs from the base model's per-token recurrence. We trained two initializations (seeds 106101 and 106102) with the same budget and answer-only loss, and we evaluate on 512 joint episodes from development families, each of which requires three complete correct answers.

A single transplanted source computation yields all three counterfactual answers jointly, at the same rate as native joint accuracy (Figure 8). In the subgroups where X, Y, and Z are distinct for a given

Table 15: Three-receiver variant. The two initializations share a training corpus, not weights.

Seed	Native joint correct	Transplant: all three hybrid targets	OFF joint correct
106101	434/512	434/512	48/512
106102	438/512	438/512	72/512

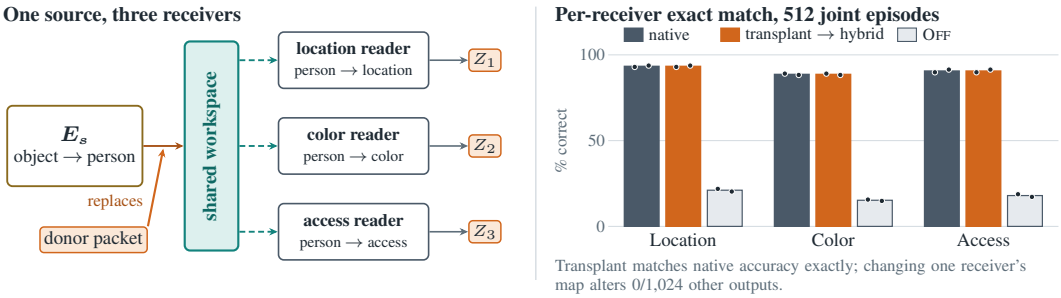

Figure 8: **One broadcast, several consumers.** Left: in the three-receiver variant, one source module reads object→person bindings and three receivers read person→location, color, and access maps, so a single transplanted source packet implies a different hybrid answer Z_k at each receiver. Right: per-receiver exact match on all 512 development episodes (bars: mean over seeds 106101 and 106102; dots: seeds). The transplant yields the hybrid target at exactly the native accuracy, whereas deleting the broadcast (OFF) collapses every receiver.

receiver, receiver-level hybrid accuracy is 89.58–92.50%, with coverage of 288, 384, and 320 of 512 for the three receivers. The effect is also selective: changing one receiver's mapping, or cutting only one receiver's input, changes neither of the other two receivers' outputs (zero changes in 1,024 comparisons per condition and seed). Therefore, the results show that the model learned to reuse shared information in this setting.

G.2 Architecture, decoding schedule, and training

The extension has one binding-source module and three recipients with location, color, and access mappings. Each module reads its own typed bank. Four internal rounds process one shared question/prefix, and each owner receives private cached answer features and lexical reads. Subsequent tokens update only that owner's text query and decode from its private cache. It is therefore a genuine same-computation reuse test, but not the base model's per-token recurrent schedule.

Each family uses four entities, two deranged choices for each of the four relation maps, two record orders, and four query sides. This gives $2^4 \times 2 \times 4 = 128$ joint episodes per family. Eight training families yield 1,024 joint episodes; four development families yield 512. Each joint episode requires three complete answers. Local questions are also generated for every individual bank. Seeds 106101 and 106102 each train a fresh local ancestor and a mixed branch. In this version, bank visibility stays local even during local-task training, unlike the base model's full-information local curriculum.

Local and mixed stages each use 2,048 updates, effective batch 32 and microbatch eight, the same recorded rate schedules, and final-answer losses. The mixed stage combines sixteen local questions with sixteen joint episodes. No correct intermediate entity or routing trajectory supervises the shared-prefix computation.

Source interventions replace the same frozen native donor packets at shared-prefix writes $0, 1, 2$. Mapping interventions change exactly one recipient's bank while keeping the source donor fixed. A single-recipient cut zeros its receiving message on all prefix rounds, not the other recipients' inputs. The private-decoder implementation was checked in the producer audit by changing one owner's generated prefix and testing that other owners were unaffected. The saved outputs are the basis of the offline manuscript audit.

Table 16: Shared three-recipient extension, all 512 development episodes. Each owner is evaluated in the same shared episode.

Seed	Owner	Native correct	Source target	Hybrid agreement	OFF correct
106101	1	476/512	476/512	512/512	112/512
106101	2	456/512	456/512	512/512	80/512
106101	3	460/512	460/512	512/512	96/512
106102	1	480/512	480/512	512/512	104/512
106102	2	452/512	452/512	512/512	76/512
106102	3	468/512	468/512	512/512	88/512

Table 17: Source-plus-recipient-remapping subgroups with pairwise distinct X, Y, Z. Coverage depends only on symbolic target values.

Seed	Owner	Eligible/all	Hybrid target correct	Hybrid-output agreement
106101	1	288/512	264/288	288/288
106101	2	384/512	344/384	384/384
106101	3	320/512	292/320	320/320
106102	1	288/512	264/288	288/288
106102	2	384/512	344/384	384/384
106102	3	320/512	296/320	320/320

G.3 PER-RECIPIENT AND THREE-VALUE RESULTS

For mapping-only and single-recipient-cut conditions, selectivity is measured relative to the native baseline. For source-plus-remapping it is measured relative to the source-patched reference. In each tested condition and seed, both non-target owners remain unchanged: zero changes among $2 \times 512 = 1,024$ comparisons. Same-facts order and other controls remain in the full exported summaries and per-case files.

Table 18: Shared-extension native joint correctness by family. Both seeds succeed completely on three families.

Seed	Family	All three correct
106101	f00	128/128
106101	f01	128/128
106101	f02	50/128
106101	f03	128/128
106102	f00	128/128
106102	f01	128/128
106102	f02	54/128
106102	f03	128/128

H Analytical properties of the intervention design and the query-only test

H.1 Fidelity versus target accuracy

Let $\widehat{y}_T$ be the complete transplanted output, $\widehat{y}_H$ the model's native output on the hybrid world, and Z the correct hybrid sequence including EOS. Set $e_T = \mathbb{P}(\widehat{y}_T \neq Z)$, $e_H = \mathbb{P}(\widehat{y}_H \neq Z)$, and $\delta = \mathbb{P}(\widehat{y}_T \neq \widehat{y}_H)$.

Proposition 2 (Fidelity–competence bound and finite-count consequences)**.** *For any joint distribution, or the empirical distribution on any finite panel,*

$$|e_T - e_H| \leq \delta. \tag{19}$$

For a panel of size n, let t count correct transplanted targets and let a count transplant/native-hybrid agreements. If f counts agreements that are wrong, then

$$f \geq \max(0, a - t). \tag{20}$$

Proof. The correctness indicators for $\widehat{y}_T$ and $\widehat{y}_H$ can differ only if the two sequences differ. Bound their absolute difference by $\mathbf{1}[\widehat{y}_T \neq \widehat{y}_H]$ and take expectations. Of the a agreements, at most t can be correct because every correct agreement is a correct transplant. Therefore at least $a - t$ agreements are wrong, and f is nonnegative. □

Table 7 reports (t, a) for four panels of size 512. Table 19 applies the bound to these existing counts; no new checkpoint execution is involved.

Table 19: Lower bounds on transplant failures that exactly reproduce the model's wrong native-hybrid output. Pooling is descriptive.

Run	Correct t	Agreement a	Failures $n - t$	Bound $a - t$	Failure fraction
A1	312	469	200	157	$157/200 = 78.50\%$
A2	317	487	195	170	$170/195 = 87.18\%$
B1	368	490	144	122	$122/144 = 84.72\%$
B2	374	462	138	88	$88/138 = 63.77\%$
Pooled	1,371	1,908	677	537	$537/677 = 79.32\%$

At least 537 of the 677 transplant target failures exactly reproduce the model's own wrong native-hybrid sequence. These are lower bounds, not exact unreported cross-tabulations. Output agreement does not identify a unique internal representation: distinct internal failures can produce equal outputs. Pooling is descriptive and does not make related examples or two initializations from the same family statistically independent.

H.2 First-token equivalence under per-token resets

Proposition 3 (First-token equivalence)**.** *Assume deterministic greedy decoding, the base model's per-token reset of all private and workspace states, unchanged input memories and persistent encodings or caches, and an intervention I that acts only during the first generated token (token 0), with no hooks active after token 0. Then*

$$\widehat{y}_0^I = \widehat{y}_0^{\text{native}} \quad \Longleftrightarrow \quad \widehat{\mathbf{y}}^I = \widehat{\mathbf{y}}^{\text{native}}. \tag{21}$$

Here $\widehat{\mathbf{y}}$ denotes the complete generated sequence, including EOS.

Proof. If first tokens agree, the next computation has the same public prefix, question, and unmodified memories and caches. Reset states are identical, and there are no remaining intervention hooks, so the next token is identical. Induction yields the same sequence and EOS behavior. The reverse implication is immediate. □

The 2,047/2,048 full native-sequence recoveries are therefore also first-token recoveries under these assumptions. They provide a decoding-level check, not independent evidence of maintaining workspace content across tokens. The proposition does not apply to the three-receiver variant (Appendix G), whose four shared rounds run once and feed private decoder caches, or to interventions on later tokens, altered persistent encodings or caches, or stochastic decoding.

H.3 Necessity, sufficiency, and interaction

Consider the base model's first answer token and final consumed write $r = 2$, followed by the owner's read at $r = 3$. Let $(b_{\text{native}}, q_{\text{native}})$ be the native recipient broadcast and queries and $(b_{\text{trans}}, q_{\text{trans}})$ the values produced by source transplantation in that same recipient. In particular, q_{trans} is not the donor's query. The two owner queries are considered together.

Four-condition design. Table 20 specifies the complementary interface experiment. We report its results at the end of this subsection.

Table 20: Four-condition comparison at the same final read interface.

Index	Condition	Broadcast	Retrieval queries
00	Native	b_{native}	q_{native}
11	Full source transplant	b_{trans}	q_{trans}
10	Query restoration	b_{trans}	q_{native}
01	Query-only transplant	b_{native}	q_{trans}

In conditions 01 and 10, override only the read interface. Do not overwrite the feedback input f, private state, memory records, token values, or output head. Both contextual and lexical retrieval remain live under the overridden queries.

Proposition 4 (Exact factorial decomposition)**.** *For any scalar response m_{ij} from these executions,*

$$\begin{aligned} m_{11} - m_{00} = (m_{01} - m_{00}) + (m_{10} - m_{00}) \\ + (m_{11} - m_{10} - m_{01} + m_{00}). \end{aligned} \tag{22}$$

The three terms are, respectively, query-only change, direct change at the native query, and interaction. Equality $m_{10} = m_{00}$ does not imply $m_{01} = m_{11}$ or zero interaction.

Proof. The identity follows by cancellation. For the non-implication, take $m(b, q) = bq$, native values $(0, 0)$, and transplant values $(1, 1)$. Then $m_{10} = m_{00} = m_{01} = 0$ and $m_{11} = 1$. Restoration removes the full effect, although query-only transplantation produces none of it and the interaction is one. □

Thus restoration tests whether redirected retrieval is necessary for the transplant effect; it does not establish retrieval's sufficiency or rule out a joint contribution from the direct route. On cases where the full transplant changes the measured response, equality of 01 and 11 together with equality of 10 and 00 supports endpoint necessity and sufficiency for that response. It does not prove the absence of direct-route effects under every possible broadcast, query, or input.

The off-diagonal conditions require an explicit interface intervention rather than a packet edit in general. At the fixed late site, the architecture's query equation has the form $q = q_* + Nf$, where q_* is fixed by the pre-read private state and public features. Holding the direct-feedback input at f_{native} fixes the natural query at q_{native}. Pairing it with $q_{\text{trans}} \neq q_{\text{native}}$ therefore requires breaking that interface equation. This controlled test is not an observational mediation estimate or a guarantee that the patched activation pair occurs on the training distribution.

Query-only experiment: setup and scope. We ran the query-only condition on the four confirmation checkpoints (A1, A2, B1, B2) and their unchanged 512-case primary panels. We did not retrain or modify the weights, architecture, tokenizer, task, or panels. In condition 01 the recipient runs natively, except that the owner's two retrieval queries at read round $r = 3$ of the first answer token are replaced by the queries recorded in the same recipient under condition 11. Non-owner queries,

rounds $r = 0$–2, and later tokens are never overridden. The protocol for this experiment was fixed before any new forward pass. The panels, however, had already been observed in the confirmation study, so the experiment is a supplementary interface test, not an independently sampled confirmation. The four initializations share two corpora, and related examples within a family are not independent, so pooled counts below are descriptive.

Implementation audit. For every case, conditions 01 and 00 had identical feedback inputs, feedback outputs, and pre-read private states, and so did conditions 10 and 11. The read weights and the contextual and lexical read results of 01 were identical to those of 11, and those of 10 were identical to those of 00. The three previously evaluated conditions reproduced all 6,144 saved confirmation sequences exactly, and the weight tensors hashed identically before and after the runs. A separate-process replay of all four conditions on the 32 preselected trace cases per model reproduced every sequence exactly, and floating-point quantities agreed within 2×10^{-5} absolute and relative tolerance. All four endpoint exceptions listed below were also reproduced in their original batches. They were selected after the results and serve only as a diagnostic.

Table 21: Four-condition hybrid-target counts (complete sequence and EOS) on the 512-case primary panels. "Agreement" is agreement with the same model's native output on the hybrid world, errors included (Table 7). The last column restricts the query-only condition to natively correct recipients; the full transplant gives 207, 206, 266, and 304 there.

Run	00	11	10	01	11 agreement	01 agreement	01, native-correct
A1	65/512	312/512	65/512	311/512	469/512	470/512	206/332
A2	56/512	317/512	56/512	317/512	487/512	487/512	206/332
B1	44/512	368/512	44/512	367/512	490/512	489/512	265/360
B2	25/512	374/512	25/512	373/512	462/512	461/512	303/404

Table 22: Per-case sequence comparisons. "Changed" counts cases in which the full transplant (11) changes the native output. "Both" requires $01 = 11$ and $10 = 00$. The last two columns give the maximum and the case mean of $\max |\Delta\text{logit}|$ between 01 and 11 on the first answer token.

Run	Changed	$01 = 11$ (all)	$01 = 11$ (on changed cases)	$10 = 00$ (on changed cases)	Both (on changed cases)	Max $\lvert\Delta\rvert$	Mean $\lvert\Delta\rvert$
A1	357	511/512	356/357	357/357	356/357	7.24	0.32
A2	377	512/512	377/377	377/377	377/377	2.65	0.20
B1	398	511/512	397/398	398/398	397/398	3.28	0.32
B2	423	511/512	422/423	422/423	421/423	3.88	0.41

Table 23: Exact per-case decomposition (Proposition 4), averaged over the 512 cases, in percentage points. Left: binary hybrid-target correctness, with the number of cases whose interaction is nonzero. Right: probability of the hybrid target's first token. This is not a full-answer probability, and when two answers share a first token it does not distinguish them.

	Hybrid-target exact match					First-token target probability			
Run	Total	Query-only	Direct	Interaction	Nonzero	Total	Query-only	Direct	Interaction
A1	48.24	48.05	0.00	0.20	1	47.94	47.91	0.006	0.026
A2	50.98	50.98	0.00	0.00	0	51.31	51.31	−0.008	0.009
B1	63.28	63.09	0.00	0.20	1	63.16	63.13	0.015	0.012
B2	68.16	67.97	0.00	0.20	1	68.27	68.14	0.006	0.128

Results. Query-only transplantation yields 311, 317, 367, and 373 hybrid targets out of 512, compared with 312, 317, 368, and 374 under the full transplant (Table 21; Figure 9). Its agreement with the native hybrid-world output, and its hybrid rate on natively correct recipients, differ from those of the full transplant by at most one case per model. The sequence-level comparison is more stringent (Table 22). Query-only and full-transplant sequences agree in 2,045 of 2,048 cases. Of the 1,555 cases in which the full transplant changes the native sequence, 1,552 satisfy $01 = 11$ and

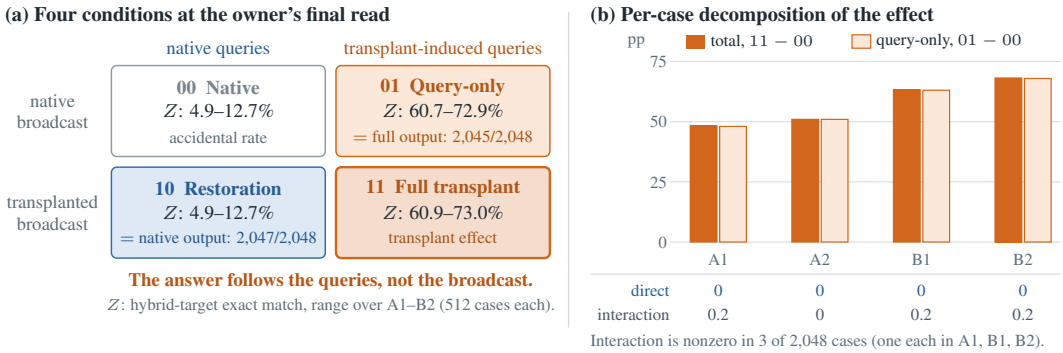

Figure 9: **Query-only test: four conditions and per-case decomposition.** (a) The four conditions at the owner's final read cross the broadcast b^3 (native or transplanted) with the owner's $r = 3$ queries (native, or those that the transplant induced in the same recipient). Ranges are hybrid-target exact match over A1–B2 (512 cases each); agreements compare complete output sequences, including EOS, pooled over models. (b) Per-case decomposition of the change in hybrid-target exact match, in percentage points (Proposition 4): the total effect ($11 - 00$) and the query-only change ($01 - 00$). The direct change at the native query ($10 - 00$) is zero in every model, and the interaction is nonzero in one case each in A1, B1, and B2.

1,551 satisfy both endpoint equalities. At the family level, query-only minus native hybrid-target accuracy is positive in all 64 family–initialization combinations, with a minimum of 18.8 points. All 32 sequences match the full transplant in 61 of these combinations. In the binary decomposition (Table 23), the direct change at the native query is exactly zero in every model. The interaction is 1/512 (0.20 points) in A1, B1, and B2 and zero in A2. In the first-token probability decomposition, the query-only change lies within 0.14 points of the total effect in every model, and the mean direct and interaction terms are at most 0.015 and 0.128 points in magnitude.

Exceptions. Four cases violate at least one endpoint equality, and all are retained. In three of them, one each in A1, B1, and B2, the full transplant produces the hybrid target and restoration recovers the native output, but query-only transplantation produces a third answer. In A1, it outputs *notebook* where the target is *football* and the native output *cup*. In B1, it outputs *Julie* where they are *room* and *kitchen*. In B2, it outputs *cup* where they are *book* and *telescope*. These are the three nonzero-interaction cases. The fourth is the restoration exception reported in Section 5.3 and Appendix E, in B2: there, restoration produces *notebook* instead of the native *coat*, whereas query-only transplantation reproduces the full transplant's *phone*. It is distinct from the other three.

Interpretation. On cases where the full transplant changes the native output, the joint equalities $01 = 11$ and $10 = 00$ support endpoint necessity and sufficiency of the redirected queries for that output. They hold in 1,551 of 1,555 such cases. The conclusion is output-level. Query-only and full-transplant first-token logits differ even when greedy outputs agree (Table 22). The results therefore do not establish zero direct-route effects or zero interaction at the logit level, under other broadcasts, queries, or inputs, or as natural indirect effects.

H.4 Order sensitivity

Fix a query. Records are encoded independently and within-record positions restart (Appendix C), so reordering records leaves their keys and token-weighted values unchanged at this fixed query. It changes the ordered selector. Write $m_a \in [0, 1]$ for a record gate and $\bar{V}_a$ for its token-weighted contextual or lexical value. Suppressing round, module, and head indices, the read is $R = \sum_a w_a \bar{V}_a$, with $w_a = m_a \prod_{j>a}(1 - m_j)$ and $w_{\text{NULL}} = \prod_j (1 - m_j)$.

Proposition 5 (Exact adjacent-swap effect)**.** *Suppose adjacent records a, b appear in that order, and define $T = \prod_{j \text{ after both}}(1 - m_j)$. Swapping just those two records changes the read by*

$$R_{\text{swap}} - R_{\text{original}} = T m_a m_b (\bar{V}_a - \bar{V}_b). \tag{23}$$

All other record weights and the NULL mass remain unchanged.

Proof. Before the swap, the two weights are $m_a(1-m_b)T$ and m_bT. Afterwards they are m_aT and $m_b(1-m_a)T$, respectively. Their changes are $+m_am_bT$ and $-m_am_bT$. Earlier records depend on the commutative product $(1-m_a)(1-m_b)$, and later records depend on neither gate. The NULL product is also unchanged. □

The effect is large when both records match the query, their values differ, and later records do not suppress both. The identity can be tested with the final query clamped; unrestricted input reordering can also change earlier recurrent states and the query itself.

Proposition 6 (Exact order average)**.** *For fixed gates and a uniformly random permutation of all n records,*

$$\mathbb{E}_{\text{order}} w_a = m_a \int_0^1 \prod_{j \neq a} (1 - t m_j)\, dt. \tag{24}$$

When all $n-1$ other record gates equal $\beta > 0$,

$$\mathbb{E}_{\text{order}} w_a = m_a \frac{1-(1-\beta)^n}{n\beta}. \tag{25}$$

For $\beta = 0$ the expectation is m_a.

Proof. Assign every record an independent continuous uniform priority to generate its order. Conditional on the target priority, let t be the probability that another record is after it. The after/before indicators are then independent and each factor has expectation $1 - tm_j$. The target priority makes t uniform on $[0,1]$, yielding the integral. Equal other-record gates reduce the product to $(1-t\beta)^{n-1}$; integration gives the stated expression. If $\beta = 0$, the product is one. □

Proposition 7 (Permutation-invariant reader implies permutation-invariant base model)**.** *Replace the ordered record selector by a softmax over visible record scores and a fixed-index NULL option:*

$$\widetilde{w}_a = \frac{e^{s_a}}{e^{s_{\text{NULL}}} + \sum_{j \text{ visible}} e^{s_j}}, \qquad \widetilde{w}_{\text{NULL}} = \frac{e^{s_{\text{NULL}}}}{e^{s_{\text{NULL}}} + \sum_{j \text{ visible}} e^{s_j}}. \tag{26}$$

Assume independent record encodings; no record-index features in any module; masks, keys, and values permuted with record identities; and all other operations following the stated architecture. Then permuting input records leaves all module reads, private states, broadcasts, and greedy generated tokens unchanged in exact arithmetic.

Proof. Initially the public features and zero private and workspace states are unchanged. At a fixed round, an unchanged query and a record permutation merely permute record scores, weights, and corresponding values. Their weighted sum and NULL mass are unchanged. The token softmax within each record is unchanged. Hence private updates, packets, routing scores, and broadcast are unchanged. Induct over rounds, then over generated tokens using identical prefixes and resets. □

This proposition guarantees invariance, not accuracy after retraining, easier optimization, or an equivalent learned strategy. In floating-point implementations, reduction-order effects require a numerical tolerance. Whether the content-dependent mechanism survives this change after retraining remains an empirical question.