WHAT DRIVES SUCCESSFUL LATENT COMMUNICATION? TRACING SENDER INFORMATION THROUGH LATENT WORKING MEMORY

Anonymous authors
Paper under double-blind review

ABSTRACT

Latent multi-agent systems can communicate through internal working memory rather than explicit text. In KV-cache relays, however, the transmitted working memory can contain both states produced while the sender processes its private input and states generated later during latent reasoning. Successful whole-relay communication therefore leaves a key attribution question unresolved: which source actually governs the receiver's sender-specific behavior? We study this behavioral provenance attribution problem using paired decision worlds with mutually exclusive sender-private requirements under the same public context. Our Causal Provenance Decomposition (CPD) first verifies sender-specific communication with native relays, then crosses the requirement source of sender-input states and sender-generated states while keeping both components available. Across Qwen3-4B and Llama-3.1-8B-Instruct on 150 paired worlds per Airline and Retail domain, receiver decisions under provenance conflict follow the sender-input source in approximately 63–85% of outputs, versus 1–10% for the sender-generated source. The resulting source-following margin is 52.7–84.0 percentage points and remains positive under conservative reassignment of all unmatched outputs. These results show that successful whole-relay communication does not imply comparable following of the generated-state donor under provenance conflict, motivating latent relays that transfer sender-private information without direct access to sender-input KV states.

1 INTRODUCTION

In a multi-agent system, one agent may observe user-specific information that another agent responsible for making the final decision cannot access directly. For example, a sender may know a private scheduling constraint that must be communicated to a receiver selecting a flight. Latent communication seeks to support such collaboration by transmitting internal representations across agent boundaries rather than relying only on explicit text (Zou et al., 2026; Du et al., 2026; Fu et al., 2026).

Successful latent communication shows that the transmitted state is useful, but not which provenance the receiver's sender-specific behavior follows. This ambiguity arises when the transmitted working memory contains both states produced while the sender processes its private input and states generated afterward during latent reasoning. The distinction is also relevant to relay designs that aim to transfer sender-private information through generated states without directly exposing sender-input KV states.

We study this problem in LatentMAS, whose shared KV working memory contains both input-context states and subsequently generated latent states (Zou et al., 2026). We refer to these two parts as **sender-input states** (P) and **sender-generated states** (L): P is produced while processing the sender's private input, whereas L is produced during the subsequent latent rollout. Both are transmitted together to the receiver as part of the same working memory.

We call the unresolved attribution target **behavioral provenance attribution**: determining which sender provenance the receiver's sender-specific behavior follows within a successful composite relay. This leads to the central question of our study:

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.

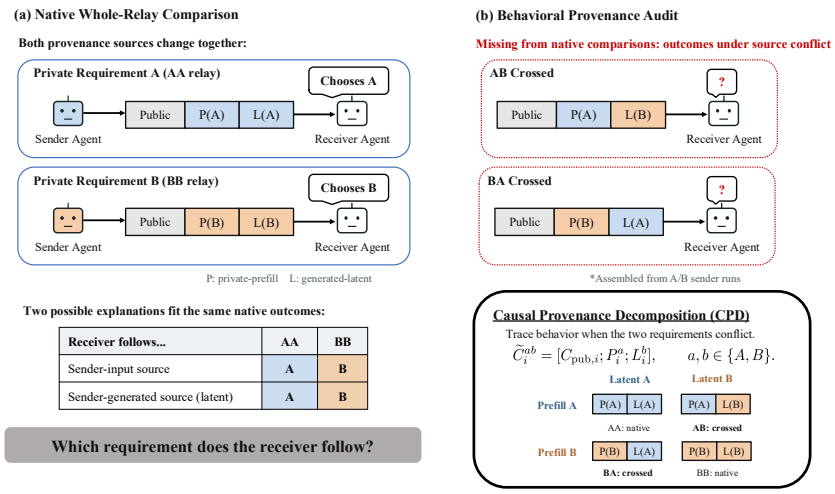

Figure 1: **Behavioral provenance attribution with CPD. (a)** Native relays (AA and BB) change the provenance of sender-input and sender-generated states together. The same sender-specific outcomes can therefore be consistent with following either source. **(b)** CPD adds crossed relays (AB and BA), assembled from separate sender runs, so that the two provenance sources correspond to conflicting private requirements while both components remain available. Receiver decisions can then be evaluated according to which donor's requirement they satisfy. P denotes sender-input private-prefill states, and L denotes sender-generated latent states. Public context, receiver inputs, component visibility, and cache geometry remain fixed. The invariant public prefix is omitted from the lower matrix. Outcomes in panel (a) are illustrative rather than empirical results.

In successful latent communication, which provenance within the latent working memory does receiver behavior follow: sender-input or sender-generated states?

Existing causal evaluations narrow this question but do not directly resolve it. Whole-relay audits establish sender-specific causal influence but vary sender-input and sender-generated provenance together. Zhang & Emu (2026) define component-restoration interventions, but do not report crossing sender-input and sender-generated sources under mutually exclusive sender-private requirements. SCIT intervenes on prompt-prefix and latent-tail states, but targets causal computation within a single model rather than inter-agent transmission (Ding et al., 2026). Most closely, Cheng et al. (2026, Appendix E.2) propose a position-partition factorial separating input and latent-generated cache positions, but do not execute the corresponding sender-input $\times$ sender-generated provenance analysis.

To study this question, we construct paired decision worlds with the same public context but mutually exclusive sender-private requirements, A and B. We first verify sender-specific communication using the native relays (P^A, L^A) and (P^B, L^B). We then construct provenance-conflict relays, (P^A, L^B) and (P^B, L^A), while keeping both state groups available and holding the receiver context, cache geometry, positions, and decoding controls fixed. The crossed conditions make the two provenance sources disagree and allow us to measure which private requirement the receiver's behavior follows. We refer to this audit protocol as **Causal Provenance Decomposition (CPD)**.

Figure 1 illustrates why native whole-relay comparisons leave behavioral provenance unresolved and how CPD addresses this ambiguity through crossed source assignments.

We evaluate LatentMAS with Qwen3-4B and Llama-3.1-8B-Instruct on 150 paired worlds in each of the Airline and Retail domains. Across all four model–domain settings, the native relays exhibit strong sender-specific effects, establishing that the complete working memory communicates the sender-private requirement. Under provenance conflict, receiver decisions follow the sender-input source in approximately 63–85% of outputs, compared with approximately 1–10% for the sender-generated source. The resulting source-following margins range from 52.7 to 84.0 percentage points.

Table 1: **Comparison of causal analyses most closely related to provenance attribution in latent communication.** Checkmarks indicate analyses that were executed and reported. A dash indicates that the criterion is not established by the cited report, rather than demonstrated absence. See Appendix A for definitions of the comparison criteria.

Work	Whole-State / Relay Behavioral Audit	Input / Generated Stage Intervention	Private-World $P \times L$ Provenance Crossing	P/L-Conflict Behavioral Provenance Following
A. Single-Agent Causal Intervention				
SCIT	✓	✓	×	×
B. Inter-Agent Communication Audit				
Wang et al.	—	—	×	×
Brito & Baquero	✓	✓	×	×
Zhang & Emu	✓	×	×	×
Cheng et al.	✓	✓	×	×
Ours	✓	✓	✓	✓

Definitions of the comparison criteria are provided in Appendix A.

Taken together, these results indicate that successful communication through the full working memory is not equally attributable to its two provenance sources. In the evaluated relays, the receiver's sender-specific behavior follows sender-input provenance far more strongly than the subsequently generated trajectory. This suggests a practical limitation for the evaluated training-free KV relay: without direct access to sender-input KV states, the same sender-specific behavior may not be preserved. Developing generated-state relays that can carry forward sender-private task information without exposing sender-input states is therefore an important direction for future work.

Contribution. Our main contribution is an empirical attribution of successful latent communication to its behavioral provenance:

- We show that whole-relay sender-specific success alone does not identify which provenance the receiver follows, and demonstrate in functioning inter-agent relays that, when sender-input and sender-generated states originate from conflicting private requirements, receiver decisions predominantly follow the sender-input provenance.

2 Related Work

2.1 Latent communication architectures

Latent communication replaces explicit text with internal representations that can be consumed directly by another model. KV-cache approaches transmit or reuse attention states: LatentMAS shares working memory containing both input-context and generated latent states (Zou et al., 2026), Cache-to-Cache learns projections between heterogeneous cache spaces (Fu et al., 2026), and KVCOMM aligns caches across differing prefix contexts for efficient reuse (Ye et al., 2025). Other methods communicate through hidden states or embeddings: Interlat learns a hidden-state interface (Du et al., 2026), StateBridge maps final-layer states into receiver inputs (Peng et al., 2026), and CIPHER communicates expected token embeddings (Pham et al., 2024).

These interfaces expose different portions of sender computation. Our analysis focuses on composite KV relays, such as LatentMAS, in which input-derived and subsequently generated states coexist in the transmitted working memory.

2.2 Causal communication audits

Recent work has moved beyond task success to test whether receiver behavior causally depends on sender-specific communication. Zhang and Emu compare current-example, other-example, absent, and self-generated messages, and define a component-restoration procedure for latent messages (Zhang & Emu, 2026). Cheng et al. audit full relayed caches using true, mismatched, zeroed, and

random alternatives in settings where relevant information is available only to the sender (Cheng et al., 2026).

Related inter-agent studies also establish behavioral dependence on hidden communication. Brito and Baquero analyze specialists with private evidence and manipulate transmitted latent states (Brito & Baquero, 2026), while Wang et al. causally intervene on hidden states and inter-agent KV-cache handoffs (Wang et al., 2026). Together, these works establish whole-message, sender-specific, and component-level causal effects, but do not report behavioral attribution between conflicting input-derived and generated provenance. Table 1 summarizes these distinctions.

2.3 Causal cache intervention

Cache interchange and activation patching provide established tools for localizing causal information within model states. SCIT performs source–recipient interventions on prompt-prefix, latent-tail, and full-cache states, directly comparing the causal roles of different cache regions (Ding et al., 2026). Brito and Baquero selectively manipulate generated latent-thought KV while retaining prompt-token KV (Brito & Baquero, 2026), and Wang et al. evaluate hidden-state and K/V-cache steering interventions (Wang et al., 2026).

Most closely, Cheng et al. (2026, Appendix E.2) propose an input-versus-generated position factorial as a natural follow-up to their full-cache audit, but do not execute the corresponding sender-input $\times$ sender-generated provenance analysis. Thus, cache substitution, stage-selective intervention, and factorial crossing are not themselves our novelty claim; our focus is the behavioral attribution of conflicting sender-input and sender-generated provenance within a composite inter-agent relay.

3 Composite-Relay Provenance Confounding

We consider paired decision worlds with a shared public context and two sender-private requirements, A and B. Each requirement admits a publicly valid decision, but no such decision satisfies both. Only the sender observes the selected requirement; the receiver's public input is identical across the pair. This construction lets us distinguish decisions satisfying one requirement from those satisfying the other without changing the receiver-visible task.

The sender transmits an invariant public prefix together with sender-input states P and sender-generated states L. Requirement A produces $[C_{\mathrm{pub}}; P^A; L^A]$, and requirement B produces $[C_{\mathrm{pub}}; P^B; L^B]$. We denote these native assignments by AA and BB, with the first letter indicating the source requirement of P and the second that of L. Because L is computed conditional on its originating input, replacing a native relay changes both sources together. We call this coupled experimental variation *composite-relay provenance confounding*; the term does not denote an unobserved statistical confounder.

Suppose the receiver satisfies A under AA and B under BB. Following either component's source would produce these same native outcomes, so even perfect native success cannot distinguish the two explanations. They predict different behavior when the sources disagree: under AB, input-following predicts an A-compatible decision, whereas generated-following predicts a B-compatible one. Our target is this *behavioral provenance attribution*: which donor's requirement receiver decisions follow under a controlled conflict between the two sources.

4 Causal Provenance Decomposition

4.1 Native and Crossed Relays

CPD partitions the cache by computational origin. Sender-input states P comprise the prefill segment beginning at the private input, including later fixed tokens whose states can depend on it. Sender-generated states L come from a fixed-length latent rollout conditioned on that prefill. The shared prefix contains only states verified to be invariant across the paired sender runs.

From those two runs, we assemble four relays:

$$\widetilde{C}^{ab} = [C_{\mathrm{pub}}; P^a; L^b], \qquad a, b \in \{A, B\}. \tag{1}$$

The AA and BB assignments reconstruct the native relays; AB and BA place their component sources in conflict. We copy keys and values together across all layers, retaining each generated segment from its donor rather than regenerating it under the substituted P. Public context, receiver inputs, component visibility, cache geometry, positions, and decoding controls remain fixed. Both components remain accessible, so this intervention varies source identity rather than availability. Reusing native sender runs, CPD adds two receiver evaluations per world (AB and BA) to the native AA/BB comparison.

Native communication. Before interpreting crossed behavior, we assess whether the native relay supports requirement-compliant, sender-specific decisions. Native constraint compliance rate (CCR) averages valid compliance for AA scored against A and BB scored against B. To distinguish aggregate success from source sensitivity, we also measure $\tau_R = \mu_{AA} - \mu_{BB}$, where μ_{ab} is the mean signed score: $+1$ for a valid A-compatible decision, -1 for a valid B-compatible decision, and zero otherwise. A positive contrast indicates that switching the native source from B to A shifts behavior toward A relative to B. This assessment does not filter worlds from the primary crossed analysis.

4.2 Behavioral Following under Provenance Conflict

The crossed assignments distinguish the two sources by associating them with mutually exclusive requirements. For AB, a valid decision satisfying A is input-following, and one satisfying B is generated-following; for BA, these assignments reverse. Remaining outputs are classified as *Other*, with invalid and valid-neither outcomes recorded separately.

Let F_P, F_L, and F_O denote the corresponding rates. With one output per assignment, they use all $2N$ crossed outputs from N worlds, weighting worlds and crossed directions equally and retaining invalid outputs. We compare the two following rates through

$$M = F_P - F_L, \tag{2}$$

where positive M indicates more frequent input-following than generated-following under conflict.

To assess whether unmatched outputs could reverse this ordering, we reassign all Other outputs to generated-following. This yields the conservative endpoint $B = F_P - (F_L + F_O) = 2F_P - 1$. Positive B means the ordering survives this reassignment. We report uncertainty for both quantities; B is a reassignment endpoint, not a statistical confidence bound. Rates are defined as proportions, while M and B are reported in percentage points.

4.3 Following in Native-Successful Worlds

To connect provenance following with successful communication within the same worlds, we additionally examine the subset S where both native outputs succeed: AA satisfies A, and BB satisfies B. Within S, n_P counts worlds where both crossed outputs follow the input source (AB satisfies A and BA satisfies B). Similarly, n_L counts worlds where both follow the generated source, with these requirements reversed.

We report the counts and rates $J_P = n_P/|S|$ and $J_L = n_L/|S|$; the remainder includes mixed following and cases with Other outputs. These are world-level joint rates within a subset selected by native outcomes, rather than the output-level rates F_P and F_L. The analysis is descriptive, with its selected denominator reported explicitly; the primary analysis retains all worlds.

4.4 Validity and Interpretation

Attribution requires that the intended relay substitutions are the only requirement-dependent changes to receiver inputs. We therefore check native-cache reconstruction, matched geometry, and invariant public states, and isolate receiver runs from additional text or metadata channels. Cache equality and receiver replay checks have separately reported coverage.

Because crossed components were not computed together, we also compare crossed and native invalid-output rates. This diagnostic can reveal generation breakdown, but valid generation alone does not establish semantic compatibility. CPD consequently measures behavior under specified

donor substitutions, not the information content of each component or a unique decomposition of the sender's computation. A lower generated-following rate does not establish that generated states exert no behavioral influence.

5 Experimental Setup

5.1 Paired Sender-Private Worlds

We construct paired decision tasks from the Airline and Retail domains of τ^2-Bench (Barres et al., 2025). Each public world has two mutually exclusive sender-private requirements, A and B, with nonempty, disjoint sets of compliant decisions. The receiver observes the same public context in both cases, while only the sender observes the private requirement. We retain database-grounded options and action-validity constraints and evaluate the specified decision, rather than the benchmark's full interactive workflow.

Both models, Qwen3-4B and Llama-3.1-8B-Instruct, use the same 150 worlds per domain: 87 Airline route clusters and 49 Retail product clusters. The sample combines the initial 100 and an additional 50 worlds per domain; the aggregate is not a separately confirmatory sample. Data construction is detailed in Appendix C.

5.2 Provenance Interventions

Native relays (P^A, L^A) and (P^B, L^B) test whether the complete relay transfers sender-private information. Crossed relays (P^A, L^B) and (P^B, L^A) retain both state groups and vary their source requirements, holding receiver context, cache geometry, positions, and decoding controls fixed. Availability interventions instead mask access to P or L while preserving their cache slots and positions.

Both models use 20 latent steps. Receiver decoding uses temperature 0.6, top-$p = 0.95$, top-$k = 20$, and at most 4,096 new tokens. Qwen uses latent-space realignment; Llama uses identity mapping with embedding-norm normalization. Appendix D specifies checkpoints and paired seeds; Appendix G provides sender and receiver prompts.

5.3 Evaluation

Each output is scored against both private requirements. Invalid outputs count as failures; valid decisions satisfying neither requirement contribute zero to both compliance rates. For crossed relays,

$$F_P = \tfrac{1}{2}\{\Pr(A \mid AB) + \Pr(B \mid BA)\}, \qquad F_L = \tfrac{1}{2}\{\Pr(B \mid AB) + \Pr(A \mid BA)\}.$$

Our primary contrast is $M = F_P - F_L$. Estimates weight worlds equally and retain all 150 worlds per setting. We report pointwise 95% percentile confidence intervals from 10,000 paired cluster-bootstrap resamples of whole route/product clusters. Appendix E gives the inference procedure and Appendix B defines the factorial source effects.

6 Results

6.1 Native Relays Successfully Transfer Sender-Private Information

The complete latent working memory produces sender-specific behavior in all four model–domain settings. Full-relay constraint compliance ranges from 63.7% to 84.7% (Table 2A), and the whole-relay effect τ_R ranges from 1.080 to 1.673, with 95% confidence intervals excluding zero in every setting (Table 2B). Thus, the attribution analysis starts from a functioning communication channel: changing the sender-private requirement changes the receiver's decision in the corresponding direction. These native comparisons establish the influence of the complete relay, but do not identify which component's provenance the receiver follows.

6.2 Availability Interventions Reveal an Asymmetry Between State Groups

Table 2A first examines which components of the working memory support requirement compliance. Retaining the public context and sender-input states while removing the generated segment yields

Table 2: **Provenance availability and source attribution.** Panel A reports constraint compliance under provenance-availability interventions. Panel B applies CPD. Both panels use the same 150 worlds per model and domain. Across both models, the private-prefill source effect closely tracks the whole-relay effect, while the generated-latent source effect is small.

		A. Provenance Availability — CCR (%)			
Model	**Domain**	**Full Relay**	**Public + Private**	**Public + Latent**	**Public Only**
Qwen	Airline	84.7	88.7	26.3	28.3
Qwen	Retail	67.0	75.0	18.3	23.0
Llama	Airline	70.7	67.0	28.0	32.3
Llama	Retail	63.7	66.0	24.3	27.7

		B. Provenance Source Attribution			
Model	**Domain**	**Whole Relay** τ_R	**Private Prefill** τ_P	**Generated Latent** τ_L	**Interaction** ψ
Qwen	Airline	1.673 [1.521, 1.814]	**1.677** **[1.557, 1.791]**	-0.003 [-0.068, 0.061]	-0.020 [-0.146, 0.095]
Qwen	Retail	1.260 [1.100, 1.409]	**1.247** **[1.095, 1.389]**	0.013 [-0.045, 0.072]	-0.053 [-0.215, 0.111]
Llama	Airline	1.320 [1.140, 1.489]	**1.317** **[1.158, 1.462]**	0.003 [-0.047, 0.052]	-0.033 [-0.117, 0.057]
Llama	Retail	1.080 [0.890, 1.257]	**1.067** **[0.879, 1.243]**	0.013 [-0.023, 0.050]	0.040 [-0.019, 0.106]

Notes. Panel A measures segment availability; Panel B measures source attribution by independently swapping the private-prefill and generated-latent sources. The outcome is $Y = \mathbf{1}[\text{A-compatible}] - \mathbf{1}[\text{B-compatible}]$. τ_R, τ_P, τ_L, and ψ denote the whole-relay, private-prefill, generated-latent, and interaction effects; $\tau_R = \tau_P + \tau_L$ (Appendix B). Brackets give 95% percentile cluster-bootstrap intervals (10,000 resamples; 87 Airline route clusters and 49 Retail product clusters). Both panels retain all 150 worlds per model and domain; invalid outputs count as failures ($Y = 0$). Qwen denotes Qwen3-4B and Llama denotes Llama-3.1-8B-Instruct.

CCRs of 66.0–88.7%, relatively close to those of the full relay. By contrast, retaining the public context and generated states while removing sender-input states yields CCRs of only 18.3–28.0%. Relative to the full relay, removing sender-input states reduces CCR by 39.3–58.3 percentage points; paired contrasts and confidence intervals are reported in Table 5.

This asymmetry motivates provenance attribution, but availability interventions change which components the receiver can access. They do not by themselves determine which private requirement the receiver follows when both components remain available. We therefore turn to CPD, which retains both state groups and independently crosses their requirement sources.

6.3 Receiver Behavior Follows Sender-Input Provenance

Table 2B reports the provenance effects from the four CPD cells. The sender-input effect τ_P closely tracks the whole-relay effect τ_R, while the signed average sender-generated effect τ_L is small, ranging from -0.003 to 0.013. Table 3 shows the behavior underlying these aggregate effects: AA and AB predominantly satisfy requirement A, whereas BA and BB predominantly satisfy requirement B. Thus, when the two provenance sources conflict, receiver decisions more often follow the sender-input donor. This direction holds separately in both AB and BA in every setting, so the observed asymmetry is not confined to one requirement or crossing direction.

Table 4 summarizes these same crossed outputs as source-following rates. Across the four settings,

$$F_P = 63.0\text{–}85.0\%, \qquad F_L = 1.0\text{–}10.3\%.$$

The resulting source-following margin is

$$M = 52.7\text{–}84.0$$

Table 3: **Raw 2×2 CPD cells underlying Table 2.** Each receiver output is scored separately against requirements A and B. Values are requirement-compliance rates (%) over the same 150 worlds per model–domain pair. Qwen denotes Qwen3-4B; Llama denotes Llama-3.1-8B-Instruct.

Model	Domain	Req.	Private A Latent A	Private A Latent B	Private B Latent A	Private B Latent B
Qwen	Airline	A	81.3	83.3	0.7	1.3
Qwen	Airline	B	0.7	1.3	86.7	88.0
Qwen	Retail	A	69.3	71.3	6.7	4.7
Qwen	Retail	B	3.3	4.0	62.7	64.7
Llama	Airline	A	68.7	69.3	4.0	4.7
Llama	Airline	B	4.7	4.0	70.0	72.7
Llama	Retail	A	67.3	66.0	10.7	11.3
Llama	Retail	B	8.0	10.0	60.0	60.0

Notes. P denotes the sender-private prompt-derived prefill segment and L the newly generated latent segment. The four columns are $AA = (P^A, L^A)$, $AB = (P^A, L^B)$, $BA = (P^B, L^A)$, and $BB = (P^B, L^B)$. Aggregate effects on Y are reported in Table 2. All values use the full 150-world samples with invalid generations retained as failures. Valid outputs satisfying neither requirement contribute zero to both rates.

Table 4: **Source following under provenance conflict.** Receivers follow the sender-input source more often than the sender-generated source. The conservative reassignment endpoint remains positive.

Model	Domain	Prefill F_P (%)	Latent F_L (%)	Other F_O (%)	Advantage M (pp)	Lower endpoint B (pp)
Qwen	Airline	85.0	1.0	14.0	84.0 [77.9, 89.6]	70.0 [58.7, 80.6]
Qwen	Retail	67.0	5.3	27.7	61.7 [53.2, 69.4]	34.0 [20.3, 47.1]
Llama	Airline	69.7	4.0	26.3	65.7 [58.2, 72.5]	39.3 [26.5, 51.3]
Llama	Retail	63.0	10.3	26.7	52.7 [43.1, 61.9]	26.0 [12.8, 39.3]

Notes. $F_P = \{\Pr(A \mid AB) + \Pr(B \mid BA)\}/2$ and $F_L = \{\Pr(B \mid AB) + \Pr(A \mid BA)\}/2$, where $\Pr(A \mid AB)$ denotes the fraction of A-compatible outputs in cell AB. $F_O = 1 - F_P - F_L$ includes both invalid and valid-neither outputs. $M = F_P - F_L$; $B = F_P - (F_L + F_O) = 2F_P - 1$ assigns every Other output to the latent source. The sensitivity endpoint B is distinct from its sampling CI. All 150 worlds per setting are retained; there are 87 route clusters in Airline and 49 product clusters in Retail. Brackets are pointwise 95% percentile CIs from 10,000 paired cluster-bootstrap resamples. All cells from a world remain paired, and whole route/product clusters are resampled. Rates use 300 crossed assignments per setting, with worlds paired for inference. Post-hoc cluster-CR1 one-sided tests of $M > 0$ and $B > 0$ all have Holm-adjusted $p < 0.001$ across eight comparisons. These rates summarize the crossed cells in Table 3; they are not independent experimental replications.

percentage points, with 95% confidence intervals entirely above zero in every setting. Assigning all unmatched outputs—including invalid and valid-neither outputs—to the sender-generated source leaves conservative endpoints B of 26.0–70.0 percentage points, again with confidence intervals entirely above zero. Here, B is the margin after this reassignment; its sampling uncertainty is reported separately by the confidence interval.

6.4 The Same Pattern Holds in Worlds Where Native Communication Succeeds

We also examine worlds in which both native relays succeed: AA satisfies A and BB satisfies B. This subset contains 70–112 of the 150 worlds per setting (Table 6). Within these same worlds, 80.0–93.5% follow sender-input provenance in both crossed directions, whereas only 0.0–1.4% follow sender-generated provenance in both directions.

This analysis is descriptive because the subset is selected using native outcomes, but it shows that native communication success and sender-input following occur within the same examples. The primary analysis above retains all 150 worlds per setting. Additional analyses of intervention validity and world-level output changes are reported in Tables 8 and 7; the token-derived KV control and exploratory carrier pilots are reported in Tables 9 and 10.

7 Discussion

Our results distinguish successful communication from behavioral provenance. The complete latent working memory exhibits sender-specific behavioral effects, but when sender-input and sender-generated states originate from different private requirements, receiver behavior follows sender-input provenance much more strongly.

This finding goes beyond whole-relay audits or availability tests. Those analyses can show that a relay works or that a state group matters, but they do not determine which provenance the receiver follows when both groups remain available and disagree.

The result also exposes a practical limitation of the evaluated training-free KV relay. Its sender-specific behavior remains strongly tied to direct access to sender-input KV states. This motivates studying whether future latent relays can preserve sender-private behavior through generated states without direct access to sender-input KV states.

8 Limitations

Our conclusions are limited to the evaluated LatentMAS configuration, two model families, and the Airline and Retail domains.

The crossed relays combine sender-input and sender-generated states that were not originally produced together. Our results therefore characterize behavior under these controlled provenance substitutions, not the semantic compatibility of the crossed caches.

Behavioral following also does not measure how much information each state group contains. Generated states can still affect individual outputs even when their average provenance-following effect is small. The token-derived KV control did not establish a positive post-prefill source effect or an increase over the native latent effect (Table 9); positive detection in that segment therefore remains unestablished.

9 Conclusion

We studied which provenance drives receiver behavior in successful latent communication. Across two models and two domains, native latent working memory exhibits sender-specific behavioral effects. However, when sender-input and sender-generated states originate from sender runs with opposing private requirements, receiver behavior follows sender-input provenance far more often: 63.0–85.0% versus 1.0–10.3%. The same pattern appears within worlds where both native communication directions succeed. Thus, successful whole-relay communication does not imply comparable following of the generated-state donor under provenance conflict.

These findings motivate latent relay designs that transfer sender-private information through generated states without requiring direct access to sender-input KV states.

AI use statement

Generative AI tools were used to aid or polish writing, support retrieval and discovery of related work, assist research ideation and execution, draft sections of the paper, and generate synthetic datasets. They were not used to prove mathematical claims. All AI-assisted outputs were reviewed by the authors, who take full responsibility for the final content of the paper.

Reproducibility statement

Appendices B–E specify estimands, world construction, checkpoint revisions, generation settings, and statistical procedures. Table 8 reports implementation-check coverage separately from behavioral outcomes. The main tables retain all evaluated worlds and can be recomputed from the paired receiver outputs without further model inference.

References

Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. τ^2-Bench: Evaluating conversational agents in a dual-control environment. *arXiv preprint arXiv:2506.07982*, 2025. doi: 10.48550/arXiv.2506.07982. URL https://arxiv.org/abs/2506.07982v1. Version 1.

Luís Brito and Carlos Baquero. When latent agents lie: KV-cache integrity in multi-agent LLM collaboration. *arXiv preprint arXiv:2606.28958*, 2026. doi: 10.48550/arXiv.2606.28958. URL https://arxiv.org/abs/2606.28958v1. Version 1.

Jiaming Cheng, Subhransu Das, and Rajiv Ramnath. When does latent communication pay? a causal audit of relayed KV caches in multi-agent LLMs. *arXiv preprint arXiv:2608.04893*, 2026. doi: 10.48550/arXiv.2608.04893. URL https://arxiv.org/abs/2608.04893v2. Version 2.

Yi Ding, Lijun Huang, and Menglin Yang. SCIT: Testing causal cache carriers in latent chain-of-thought models. *arXiv preprint arXiv:2608.27265*, 2026. URL https://arxiv.org/abs/2608.27265v1. Version 1.

Zhuoyun Du, Runze Wang, Huiyu Bai, Zouying Cao, Xiaoyong Zhu, Yu Cheng, Bo Zheng, Wei Chen, and Haochao Ying. Enabling agents to communicate entirely in latent space. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 27106–27129, San Diego, California, United States, July 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-long.1248. URL https://aclanthology.org/2026.acl-long.1248/.

Tianyu Fu, Zihan Min, Hanling Zhang, Jichao Yan, Guohao Dai, Wanli Ouyang, and Yu Wang. Cache-to-Cache: Direct semantic communication between large language models. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=LeatkxrBCi.

Yanwen Peng, Delvin Ce Zhang, Xi Wang, and Nikolaos Aletras. StateBridge: Training-free hidden-state alignment for latent communication in LLM multi-agent systems. In *Conference on Language Modeling*, 2026. URL https://openreview.net/forum?id=I0f7u57bJ5.

Chau Pham, Boyi Liu, Yingxiang Yang, Zhengyu Chen, Tianyi Liu, Jianbo Yuan, Bryan A. Plummer, Zhaoran Wang, and Hongxia Yang. Let models speak ciphers: Multiagent debate through embeddings. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=sehRvaIPQQ.

Chenxi Wang, Ruiyang Huang, Jiayan Sun, Lei Wei, and Yifan Wu. Out of sight, not out of mind: Unveiling latent attack in latent-based multi-agent systems. *arXiv preprint arXiv:2605.28214*, 2026. doi: 10.48550/arXiv.2605.28214. URL https://arxiv.org/abs/2605.28214v2. Version 2.

Hancheng Ye, Zhengqi Gao, Mingyuan Ma, Qinsi Wang, Yuzhe Fu, Ming-Yu Chung, Yueqian Lin, Zhijian Liu, Jianyi Zhang, Danyang Zhuo, and Yiran Chen. KV-COMM: Online cross-context KV-cache communication for efficient LLM-based multi-agent systems. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/1a074a28c3a6f2056562d00649ae6416-Abstract-Conference.html.

Huixiang Zhang and Mahzabeen Emu. Do latent channels actually communicate? a causal audit of latent multi-agent LLM communication. *arXiv preprint arXiv:2607.26773*, 2026. doi: 10.48550/arXiv.2607.26773. URL https://arxiv.org/abs/2607.26773v1. Version 1.

Jiaru Zou, Ruizhong Qiu, Gaotang Li, Xiyuan Yang, Katherine Tieu, Pan Lu, Ke Shen, Hanghang Tong, Yejin Choi, Jingrui He, James Zou, Mengdi Wang, and Ling Yang. Latent collaboration in multi-agent systems. In *International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=syG9I9ofd8.

A Details of the Causal-Analysis Comparison

Table 1 compares prior work along four dimensions chosen to distinguish our target estimand from related causal interventions on latent states. Because several prior studies perform interventions that partially overlap with these dimensions, we provide the precise definitions and qualifications below.

Whole-State / Relay Behavioral Audit. We mark this column when a study tests whether downstream behavior causally depends on the identity, replacement, removal, or manipulation of a complete designated internal state or transmitted inter-agent relay. This criterion does not require decomposition of that state into input-derived and subsequently generated components.

Input / Generated Stage Intervention. We mark this column when a study explicitly distinguishes input- or prompt-derived states from states subsequently produced during latent generation and intervenes on these regions separately. This criterion is broader than our provenance analysis: selective removal, replacement, patching, or manipulation of one region is sufficient.

Private-World $P \times L$ Provenance Crossing. This criterion is substantially narrower. We require both sender-input states P and sender-generated states L to remain available while their provenance is independently varied across mutually exclusive sender-private worlds. In our setting, the relevant crossed relays are (P^A, L^B) and (P^B, L^A), where private requirements A and B imply distinct behavioral targets. Component masking, corruption, or source–recipient patching alone does not satisfy this criterion.

P/L-Conflict Behavioral Provenance Following. We mark this column only when the evaluation measures which provenance-specific behavioral target the receiver follows when the provenance of P and L disagrees. Thus, observing that a perturbation changes task accuracy or receiver predictions is insufficient. The outcome must distinguish behavior consistent with the provenance of P from behavior consistent with the provenance of L.

A.1 Scope of Our Distinction

The distinction claimed by our comparison is therefore not the use of cache intervention, component restoration, source substitution, or factorial manipulation per se. Each has close precedents. Our target is the narrower causal estimand obtained by combining three properties: (i) information that is private to the sender, (ii) a composite inter-agent relay containing both sender-input and sender-generated states, and (iii) a conflict in which those two components originate from mutually exclusive private worlds.

CPD operationalizes this estimand by comparing native relays (P^A, L^A) and (P^B, L^B) with crossed relays (P^A, L^B) and (P^B, L^A), and then measuring whether receiver behavior follows the behavioral target associated with P or with L. Accordingly, our novelty claim concerns stage-specific behavioral provenance identification under an inter-agent private-information conflict, rather than the intervention primitive itself.

A.2 Validity checks

Reconstruction and information isolation. The assembled AA and BB caches must reproduce their native counterparts. Cache entries, geometry, and receiver replays are checked under matched controls, with coverage and numerical tolerances reported for each model–domain setting. Checks preserve layer structure, boundaries, positions, masks, and the invariant public prefix. Private text, donor labels, and experiment metadata must not create another receiver-visible path. Cache equality, first-step logit agreement, and full-output agreement are distinct checks; their coverage is reported separately.

Compatibility of crossed states. Every cell retains invalid outputs in its primary denominator. We distinguish A-compatible, B-compatible, valid-neither, and invalid outputs, and compare the average invalid rates of crossed and native cells. A large crossed-specific deterioration can indicate incompatibility. Conversely, a small observed gap or a confidence interval containing zero does not establish semantic compatibility or equivalence of invalid rates. Structural reconstruction, generation validity, and any predeclared performance gates are reported separately, so that successful cache assembly is not treated as a general validity guarantee.

Invalid-output sensitivity and equivalence. Under the primary scoring rule, invalid outputs receive zero. For a conservative sensitivity analysis, each invalid score may instead take any value in $[-1, 1]$, while valid scores remain fixed. For one scored outcome per world and cell, let F_i^{ab} indicate invalidity. The resulting sample bounds for the latent effect are

$$[\widehat{\tau}_L - b_L, \widehat{\tau}_L + b_L], \qquad b_L = \frac{1}{2N} \sum_{i,a,b} F_i^{ab}. \tag{3}$$

With repeated decoding, the corresponding within-world averages are used. These are sensitivity bounds, not confidence intervals. Equivalence under the observed scoring rule and invalid-robust population equivalence are different conclusions. The latter requires uncertainty-aware tests that the lower and upper endpoint means fall within the declared equivalence margin, with the applicable multiplicity correction. Selecting only worlds with four valid outputs changes the target population and cannot establish full-sample equivalence. A small observed $\widehat{\tau}_L$ therefore supports neither an exact-zero effect nor the absence of private information.

Alternative-carrier falsification controls. Separate StateBridge and CIPHER pilots test whether sender-specific decisions can arise through generated-state carriers without a direct prompt-KV handoff. Matched and opposite-requirement scoring, together with no-message controls, assess sender specificity. StateBridge uses token-conditioned hidden-state generation and vocabulary anchoring; CIPHER uses expected-embedding recurrence, with discrete diagnostics for message-boundary selection. These controls constrain a universal latent-failure explanation. Their small samples and different communication budgets do not identify why LatentMAS exhibits its observed pattern or establish carrier superiority. They are interpretation controls for the CPD finding, not additional provenance decompositions.

B Effect Definitions and the Role of Crossed Observations

For world i and cell ab, the signed score is

$$Y_i^{ab} = \mathbf{1}[A\text{-compatible}] - \mathbf{1}[B\text{-compatible}], \qquad \widehat{\mu}_{ab} = N^{-1} \sum_i Y_i^{ab}. \tag{4}$$

The requirement sets are disjoint. Invalid and valid-neither outputs receive zero, and remain separate categories in diagnostic summaries. With μ_{ab} denoting the corresponding mean, the source effects are

$$\tau_P = \tfrac{1}{2}[(\mu_{AA} - \mu_{BA}) + (\mu_{AB} - \mu_{BB})], \tag{5}$$

$$\tau_L = \tfrac{1}{2}[(\mu_{AA} - \mu_{AB}) + (\mu_{BA} - \mu_{BB})], \tag{6}$$

$$\psi = \mu_{AA} - \mu_{AB} - \mu_{BA} + \mu_{BB}, \tag{7}$$

$$\tau_R = \mu_{AA} - \mu_{BB} = \tau_P + \tau_L. \tag{8}$$

Each component effect averages a B-to-A donor substitution over the other component's two sources. The identity follows algebraically under equal weighting and does not assume additive receiver behavior. The component effects and τ_R range from -2 to 2 in signed-score units; ψ ranges from -4 to 4. They are not CCR percentages. Sample estimates substitute the empirical cell means, and the identity holds before rounding.

The direct following margin has the equivalent forms

$$M = F_P - F_L = \tfrac{1}{2}(\mu_{AB} - \mu_{BA}) = \tfrac{1}{2}(\tau_P - \tau_L). \tag{9}$$

Here M is a proportion; the tables multiply it by 100 to report percentage points. Native observations identify τ_R but leave $\mu_{AB} - \mu_{BA}$ undetermined. For example, mean-score patterns $(1, 1, -1, -1)$ and $(1, -1, 1, -1)$ in (AA, AB, BA, BB) order agree on both native cells but have opposite source-following behavior. CPD supplies the two crossed observations using the existing sender runs and two additional receiver evaluations per world. It changes the evaluation design, not the underlying communication algorithm.

These contrasts characterize responses to the specified donor substitutions. They do not uniquely decompose computation within a native sender run or quantify the information encoded in each segment.

C World Construction and Decision Scope

Both models evaluate the same 300 distinct public worlds: 150 Airline and 150 Retail. The four model–domain settings therefore do not represent 600 independent task worlds. Each domain combines the initial 100-world set and a subsequently fixed 50-world extension. The combined results are reported as a full-sample analysis, without treating the entire 150-world aggregate as a prespecified confirmatory cohort.

Worlds are derived from fixed copies of the τ^2-Bench databases and policies (Barres et al., 2025).[1] Airline worlds present flight choices under shared public booking constraints. Retail worlds are pre-confirmation single-item replacement decisions: the public order, target item, candidate product variants, prices, option values, and availability remain fixed. User authentication, final consent, and execution of a transaction are outside the evaluated decision.

The A/B requirements are newly constructed counterfactual constraints on these decisions. For example, Retail pairs can require different values of the same product option, reverse the priority of two option preferences, or contrast minimum price with minimum price subject to an option requirement. Every retained world has at least one publicly valid decision satisfying each requirement, and the two compliant sets are disjoint. All tied compliant decisions remain acceptable. Requirements, candidate validity, and compliant sets are determined from source data before inspecting receiver outcomes.

[1] Source repository: `https://github.com/sierra-research/tau2-bench`, commit `b7ea9074c1cba482b30687fecdb5c8425fd6f619`.

Only the sender receives the selected private requirement. The receiver receives the shared decision context and choices, plus the experimental relay. Answer sets, donor labels, and experimental metadata are not provided as an additional communication path. Task scores assess requirement compliance for this controlled decision, not the original benchmark's complete conversational workflow.

The 150 Airline worlds share 87 routes; the 150 Retail worlds share 49 products. Inference retains this clustering.

D Implementation and Generation Settings

We use Hugging Face inference with FP32 arithmetic and mathematical scaled-dot-product attention on NVIDIA H100 GPUs. The sender and receiver within each experiment use the same model; states are not exchanged between model families. The checkpoint revisions are:

- Qwen/Qwen3-4B: `1cfa9a7208912126459214e8b04321603b3df60c`.
- Meta-Llama-3.1-8B-Instruct: `0e9e39f249a16976918f6564b8830bc894c89659`.

Both models run 20 latent recurrence steps. Qwen enables the LatentMAS latent-space realignment and its thinking mode. Llama uses the identity-mapping path with embedding-norm normalization, its official chat template, and the fixed template date 26 July 2024. It does not receive Qwen-specific thinking markers. These settings were fixed before the corresponding main executions; model comparisons include these interface differences as well as differences in weights.

Receiver generation uses temperature 0.6, top-$p = 0.95$, top-$k = 20$, and a maximum of 4,096 new tokens. Each world/cell has one stored generation. The receiver seed is the integer represented by the first eight hexadecimal digits of SHA-256 of the UTF-8 world identifier; it is shared across that world's conditions. Sender runs use this seed plus one modulo 2^{32}, shared between the two requirements. The same random seed controls a paired run; it does not guarantee identical sampled token sequences across different caches.

Within each model/world pair, the A/B prompts match in total token length, private-segment start, and public-prefix token IDs. Finite semantically equivalent requirement formulations permit tokenizer-specific alignment without selecting on model success. There is no padding, truncation, or position rebasing in the native CPD construction. The public prefix is held fixed, both keys and values are copied at every layer, and a crossed latent segment is retained from its original donor execution. The receiver's public inputs and insertion boundary remain identical across the four cells.

Availability interventions preserve physical cache length and absolute positions while masking receiver access to the specified provenance segments. The labels Public+Private and Public+Latent in Table 2 mean access to C_{pub} together with P or L, respectively. Public Only retains only the invariant public segment. The receiver's public decision prompt is unchanged.

E Statistical Procedures

For a world-level statistic z_i, point estimates use $\widehat{\theta} = N^{-1} \sum_i z_i$. For following rates, each world's two crossed directions are averaged first. Availability contrasts likewise compare conditions within the same world and average the two requirement assignments. All cells remain paired throughout inference, and invalid outputs remain in the primary denominator.

We draw 10,000 bootstrap samples of G whole route/product clusters with replacement, retaining all worlds and conditions within every selected cluster. Each replicate divides the total of the selected world values by the selected world count, preserving world weighting when cluster sizes differ. Confidence intervals are the 2.5th and 97.5th percentiles, using seed 20260925 unless a pilot table specifies otherwise. These are pointwise intervals. They describe cluster resampling uncertainty in the stored paired runs, not variability estimated from repeated decoding seeds or a guarantee for arbitrary future task distributions.

Cluster-robust CR1 tests use an intercept-only model on world-level contrasts. With $s_g = \sum_{i \in g}(z_i - \widehat{\theta})$,

$$\widehat{\mathrm{se}}(\widehat{\theta})^2 = \frac{G}{G-1}\frac{\sum_g s_g^2}{N^2}. \tag{10}$$

Tests use a t reference distribution with $G-1$ degrees of freedom. The post-hoc following analysis applies Holm correction to the eight one-sided tests of $M > 0$ and $B > 0$ across four model–domain settings. All eight adjusted values are below 0.001. These tests and the pointwise intervals are separate summaries.

For the native-success subset, bootstrap draws resample the original 150-world clusters and recompute the selected numerator and denominator in each replicate. Selection uses native outcomes, so the resulting joint-following rates are descriptive. The token-derived control uses paired differences on its fixed 50-world subset; its two one-sided tests form a separate Holm family. Carrier-pilot intervals use their original seed and cluster units, as specified in their table.

F Additional Experimental Results

F.1 Availability Contrasts

Table 5 compares access conditions in the same worlds. These contrasts complement the crossed source effects; masking a segment and changing its donor are different interventions.

Table 5: **CCR effects of provenance availability.** The first four contrasts change access to one relay component while holding the other availability condition fixed; the fifth is their interaction. Values are percentage-point differences with cluster-bootstrap 95% confidence intervals over 150 worlds per model and domain.

Model	Contrast	Airline	Retail
Qwen	Full − Public+Private	-4.0 $[-8.5, 0.4]$	-8.0 $[-13.3, -2.9]$
	Full − Public+Latent	58.3 $[51.0, 65.7]$	48.7 $[41.3, 55.8]$
	Public+Private − Public	60.3 $[54.8, 65.7]$	52.0 $[45.8, 58.0]$
	Public+Latent − Public	-2.0 $[-7.4, 3.6]$	-4.7 $[-10.0, 1.0]$
	Availability interaction	-2.0 $[-9.0, 5.2]$	-3.3 $[-11.1, 4.1]$
Llama	Full − Public+Private	3.7 $[-2.9, 10.5]$	-2.3 $[-8.0, 3.0]$
	Full − Public+Latent	42.7 $[35.5, 49.7]$	39.3 $[32.5, 45.8]$
	Public+Private − Public	34.7 $[28.1, 40.7]$	38.3 $[31.9, 44.6]$
	Public+Latent − Public	-4.3 $[-9.7, 1.1]$	-3.3 $[-7.5, 1.0]$
	Availability interaction	8.0 $[0.4, 15.5]$	1.0 $[-6.1, 7.7]$

Notes. Positive values indicate higher CCR for the condition on the left. The interaction is $\text{Full} - \text{Public+Private} - \text{Public+Latent} + \text{Public}$. These contrasts measure segment availability, not the provenance source effects τ_P and τ_L. Both models use all 150 worlds per domain, with invalid outputs retained as failures. Intervals bootstrap paired world-level contrasts over whole route/product clusters; they are pointwise, not simultaneous intervals.

F.2 Following within Native-Successful Worlds

Table 6 connects native success and crossed behavior in the same examples. The full-sample conclusions do not depend on membership in this descriptive subset.

Table 6: **Source following within the same worlds that succeed natively in both directions.** Most selected worlds follow the sender-input source in both crossed cells.

Model	Domain	Native success $\lvert S\rvert/N$	Selected clusters	Both input J_P [95% CI]	Both generated J_L	Remaining patterns
Qwen	Airline	112/150	74	93/112 83.0% [74.0, 90.9]	0/112 0.0%	19/112
Qwen	Retail	70/150	37	56/70 80.0% [70.0, 89.0]	1/70 1.4%	13/70
Llama	Airline	85/150	52	71/85 83.5% [75.9, 90.9]	0/85 0.0%	14/85
Llama	Retail	77/150	40	72/77 93.5% [87.8, 98.6]	0/77 0.0%	5/77

Notes. S contains worlds where AA satisfies A and BB satisfies B. J_P is the fraction of S where AB satisfies A and BA satisfies B; J_L reverses those two targets. Following-pattern counts use $|S|$ as the denominator; percentages and CIs are in percent. Remaining patterns include mixed source following, valid-neither, and invalid outputs. The 95% CI for J_P resamples the original 150-world clusters and recomputes both selected numerator and denominator. This post-treatment subset is descriptive and does not replace the full-sample analysis. Zero observed J_L does not establish a zero population rate; degenerate zero-event bootstrap intervals are not shown.

F.3 Changes at Fixed Sender-Input Provenance

Table 7 reports individual-world changes when the generated-state donor changes at fixed prefill. Positive and negative signed changes can offset, and action changes need not change requirement compatibility. A small mean τ_L is therefore compatible with observable effects on individual outputs.

Table 7: **Observed changes when the latent source is swapped at fixed prefill.** A small signed average can coexist with changes in individual worlds.

A. Paired output changes and conditional effects

Model	Domain	Swap	**Category changed**	**Action changed both valid**	**Conditional contrast** [CI]
Qwen	Airline	$AA \to AB$	35/150 23.3%	44/123 35.8%	-0.013 [-0.100, 0.080]
Qwen	Airline	$BA \to BB$	26/150 17.3%	52/131 39.7%	0.007 [-0.082, 0.092]
Qwen	Retail	$AA \to AB$	49/150 32.7%	19/97 19.6%	-0.013 [-0.120, 0.099]
Qwen	Retail	$BA \to BB$	52/150 34.7%	12/84 14.3%	0.040 [-0.053, 0.132]
Llama	Airline	$AA \to AB$	21/150 14.0%	30/120 25.0%	-0.013 [-0.089, 0.064]
Llama	Airline	$BA \to BB$	17/150 11.3%	28/126 22.2%	0.020 [-0.037, 0.072]
Llama	Retail	$AA \to AB$	10/150 6.7%	6/121 5.0%	0.033 [-0.013, 0.090]
Llama	Retail	$BA \to BB$	10/150 6.7%	4/113 3.5%	-0.007 [-0.053, 0.034]

B. World-level signed latent contrasts

Model	Domain	$\ell_w > 0$	$\ell_w < 0$	$\ell_w = 0$	**Mean** $\lvert\ell_w\rvert$ [95% CI]
Qwen	Airline	23	23	104	0.177 [0.111, 0.246]
Qwen	Retail	25	25	100	0.227 [0.166, 0.295]
Llama	Airline	17	14	119	0.123 [0.086, 0.166]
Llama	Retail	9	7	134	0.067 [0.034, 0.104]

Notes. Categories are A-compatible, B-compatible, valid-neither, and invalid. Category-change counts retain all 150 worlds. Action-change denominators include only pairs where both outputs are valid; this column is descriptive and selected. Panel A contrasts are $d_A = Y_{AA} - Y_{AB}$ and $d_B = Y_{BA} - Y_{BB}$ for the corresponding swaps. Panel B uses $\ell_w = (d_{A,w} + d_{B,w})/2$, whose mean is τ_L. Counts sum to 150 per setting. Cells share the receiver seed within a world. These are observed changes in one paired decoding run, not full behavioral probabilities. All 150 worlds per setting are retained; there are 87 route clusters in Airline and 49 product clusters in Retail. Brackets are pointwise 95% percentile CIs from 10,000 paired cluster-bootstrap resamples. All cells from a world remain paired, and whole route/product clusters are resampled.

F.4 Intervention Checks and Output Validity

Table 8 distinguishes exhaustive cache and geometry checks from sampled effective-mask and replay checks. The crossed-minus-native invalidity gaps are small in point estimate, but this does not establish semantic compatibility or invalidity equivalence. No unmeasured replay coverage is inferred from cache identity alone.

Table 8: **Validity changes and implementation evidence.** Recorded structural checks pass; sampled replay coverage is reported separately.

A. Output invalidity

Model	Domain	Native invalid (%)	Crossed invalid (%)	Crossed − Native (pp)
Qwen	Airline	9.3 [3.5, 15.8]	8.7 [3.8, 14.0]	-0.7 [-5.2, 3.7]
Qwen	Retail	26.7 [20.5, 33.1]	25.7 [19.4, 32.3]	-1.0 [-4.9, 2.7]
Llama	Airline	12.7 [8.2, 17.4]	14.3 [10.5, 18.2]	1.7 [-1.9, 5.1]
Llama	Retail	19.3 [14.2, 24.7]	20.0 [14.5, 25.8]	0.7 [-1.7, 3.1]

B. Checks passed / evaluated

Check	Qwen Airline	Qwen Retail	Llama Airline	Llama Retail
Exact native diagonal cache/hash	300/300	300/300	300/300	300/300
Public-prefix identity (worlds)	150/150	150/150	150/150	150/150
Donor segment provenance (cells)	600/600	600/600	600/600	600/600
Public receiver inputs (worlds)	150/150	150/150	150/150	150/150
Position/length/input mask (worlds)	150/150	150/150	150/150	150/150
Full decode/effective mask (cells)	60/600	60/600	60/600	60/600
Measured first-logit equality	20/20	20/20	—	—
Measured complete-token replay	20/20	20/20	—	—

Notes. All 150 worlds per setting are retained; there are 87 route clusters in Airline and 49 product clusters in Retail. Brackets are pointwise 95% percentile CIs from 10,000 paired cluster-bootstrap resamples. All cells from a world remain paired, and whole route/product clusters are resampled. Native denotes AA/BB and crossed denotes AB/BA. Panel B is read from runtime receipts and rechecked against the stored raw-output and segment hashes. Cache identity uses exact equality; recorded first-logit comparisons use $\mathrm{atol} = \mathrm{rtol} = 0$, with maximum/mean absolute error 0. Each Qwen replay count covers 10 exploratory worlds (20 diagonals) in that domain; it does not cover the heldout50. — denotes no separately measured native-vs-reconstructed logit/full-output replay, not a failed test. Full-decode auditing covers 15 worlds per setting (60 cells). Structural agreement and small invalidity differences do not establish crossed-cache semantic compatibility.

F.5 TOKEN-DERIVED KV CONTROL

We replace the native generated segment L^b with T^b, constructed by teacher-forcing a short explicit donor requirement after the unchanged donor prefill. The same four source assignments are then evaluated as $[C_{\text{pub}}; P^a; T^b]$. The control text occupies 12–20 tokens before newline padding to 20 positions, contains no answer identifier or instruction to override the prefix, and is not separately supplied to the receiver. The original P segment is unchanged. This tests a token-derived carrier at the same cache positions; it is not a native continuous latent rollout.

The fixed sample is Qwen–Airline's 50-world extension, with 49 route clusters. All four cells and all invalid outputs are retained. Native results are recomputed on these exact same worlds. The primary comparison is the paired contrast $\Delta = \tau_T - \tau_L$, rather than the difference between two significance decisions. Table 9 reports $\Delta = 0.030$ with a 95% CI of $[-0.092, 0.157]$. Neither a positive control source effect nor an increase over the native latent effect is established. The control therefore does not supply positive evidence that this interface can detect post-prefill source following when the donor sources conflict.

Table 9: **Same-interface token-derived KV control.** The control did not establish a positive post-prefill source effect or a supported increase over the native latent effect.

Carrier	Matched CCR (%)	Post-prefill source effect	Crossed invalid (%)
Native latent L	84.0 [75.5, 91.8]	-0.040 [-0.135, 0.051]	4.0 [0.0, 9.2]
Token-derived KV T	92.0 [87.0, 96.9]	-0.010 [-0.082, 0.070]	2.0 [0.0, 4.9]
Control − native	8.0 [-1.0, 17.3]	0.030 [-0.092, 0.157]	-2.0 [-8.2, 2.9]

Notes. Qwen–Airline, the same 50 heldout worlds and 49 route clusters for both carriers. The original private prefill is reused unchanged. A short explicit rendering of the donor requirement is teacher-forced after that prefill and padded with newline tokens to the same 20 positions. No answer ID or priority instruction is added, and the receiver receives no separate control text. The effect is τ_L for native L and τ_T for control T; the last row directly estimates the paired difference $\Delta = \tau_T - \tau_L$. CCR/invalid differences in the last row are in pp. Brackets are pointwise 95% paired cluster-bootstrap CIs (10,000 resamples). All 50 fixed worlds and invalid outputs are retained. One-sided CR1 tests give $p = 0.6004$ for $\tau_T > 0$ and $p = 0.3216$ for $\Delta > 0$; both Holm-adjusted values are 0.6433. The descriptive quality guard passed, but the prespecified positive-detection criterion did not. This diagnostic carrier is not a new latent method or evidence of absent information in native latent states.

F.6 CONTRASTING CARRIER PILOTS

The StateBridge and CIPHER pilots use Qwen3-4B and the same ten Dev worlds per domain, separate from the main evaluation. StateBridge communicates generated token-conditioned hidden states; CIPHER uses expected-embedding recurrence. These interfaces omit the direct sender-input KV handoff. Text and token-suffix conditions provide additional references, and no-message evaluates the public-context decision alone. Table 10 gives both matched and opposite-requirement scores of stored outputs.

TSR is the fraction of publicly valid actions. CCR additionally requires the selected private requirement. CPF requires success for both A and B in the same world using their respective messages; its denominator is ten worlds, whereas TSR/CCR use twenty assignments. The counterfactual columns score those same outputs against the opposite target and do not add independent generations. These small pilots support sender-specific transfer in the tested alternative interfaces but are not full CPD replications or evidence of carrier superiority.

Table 10: **Sender-specific transfer in contrasting carrier pilots.** Results use the same 10 Dev worlds per domain, separate from the 150-world main evaluation.

Domain	Carrier	Matched TSR	Matched CCR [CI]	Matched CPF	Counterfactual CCR [CI]	Counterfactual CPF
Airline	StateBridge	90.0	90.0 [62.5, 100.0]	90.0	0.0 [0.0, 0.0]	0.0
Airline	CIPHER	100.0	100.0 [100.0, 100.0]	100.0	0.0 [0.0, 0.0]	0.0
Airline	CIPHER ≤ 64	95.0	95.0 [81.2, 100.0]	90.0	0.0 [0.0, 0.0]	0.0
Airline	Text	90.0	90.0 [62.5, 100.0]	90.0	0.0 [0.0, 0.0]	0.0
Airline	Token suffix ≤ 64	90.0	90.0 [62.5, 100.0]	90.0	0.0 [0.0, 0.0]	0.0
Airline	No message	100.0	50.0 [50.0, 50.0]	0.0	—	—
Retail	StateBridge	75.0	70.0 [50.0, 90.0]	50.0	5.0 [0.0, 15.0]	0.0
Retail	CIPHER	80.0	70.0 [45.0, 95.0]	60.0	5.0 [0.0, 15.0]	0.0
Retail	CIPHER ≤ 64	70.0	65.0 [35.0, 90.0]	60.0	5.0 [0.0, 15.0]	0.0
Retail	Text	75.0	70.0 [45.0, 95.0]	60.0	5.0 [0.0, 15.0]	0.0
Retail	Token suffix ≤ 64	70.0	60.0 [35.0, 80.0]	40.0	10.0 [0.0, 25.0]	0.0
Retail	No message	100.0	30.0 [15.0, 45.0]	0.0	—	—

Notes. All rates are percentages. Qwen3-4B is used. Airline has 7 route clusters; Retail has 10 product clusters. TSR is action validity, CCR target-requirement compliance, and CPF success for both target requirements in a world. TSR/CCR use 20 assignments; CPF uses 10 worlds. Invalid outputs remain failures. Counterfactual rates rescore each stored output against the opposite requirement; they are not additional receiver generations. No-message uses one output per world scored against both targets. Shared baseline outputs are reused, not independent replications. CCR intervals are the original pointwise 95% cluster-bootstrap CIs (10,000 resamples, seed 20260924). Degenerate bootstrap intervals at 0 or 100% do not establish population certainty. Different carrier interfaces and message lengths limit comparisons; these pilots do not validate native crossed-KV semantic compatibility.

G SENDER AND RECEIVER PROMPTS

This appendix records the prompts used for the main 150-world availability and CPD evaluations. The fixed instructions below are copied from the execution code. Uppercase braced fields are documentation placeholders for world-specific content; the executed prompts contain the actual values, not these placeholder strings. Display line wrapping is for readability. Each role receives one system message and one user message, rendered using the model-specific chat template.

Only the sender user message contains the selected private requirement. The receiver messages are identical for A and B and for all four CPD cells. The selected relay is supplied as a KV prefix, not appended to the receiver user message as private text. The sender's JSON-output instruction is retained in the latent-condition prefill; the native latent runtime then performs its fixed 20-step recurrence rather than decoding a JSON message.

The sender is instructed to preserve and relay a private requirement without selecting the downstream action. These tasks therefore test requirement transfer to a receiver with decision-relevant inventory, rather than requiring the sender to solve the downstream decision. The observed source-following asymmetry is conditional on this instruction and relay configuration.

G.1 AIRLINE SENDER

System message.

```
You are Agent 1, the upstream constraint holder in a two-agent workflow.

Your job is to communicate to Agent 2 the user-specific information that
     is necessary for Agent 2 to make the correct downstream decision.

Follow these rules:
1. Preserve the user's private constraint exactly in meaning. Do not
    weaken, strengthen, reinterpret, or replace it with your own
    preference.
2. Communicate all and only the decision-relevant private information
    needed by Agent 2.
3. Do not solve or execute the downstream task yourself.
4. Do not use tools or make environment changes.
5. Do not invent facts that are not present in the input.
6. If the private constraint changes, your communicated information
    should change accordingly.
7. Public task information may be used to understand what part of the
    private information is decision-relevant, but the private constraint
     must remain the authoritative user-specific requirement.

Your output will be passed to Agent 2 through the experiment's
    communication channel.
```

User message.

```
[PUBLIC TASK]
{PUBLIC_TASK}

[PUBLIC CONTEXT]
{PUBLIC_CUSTOMER_JSON}
Agent 2 has the current flight inventory.

[RELAY OUTPUT CONTRACT]
Relay the private constraint faithfully, including strict versus
    inclusive time boundaries and the order of optimization priorities.
    You do not have Agent 2's flight inventory: do not choose a flight
    or invent availability. Return one JSON object with a single field
    named private_constraint containing the complete requirement. Do not
     add account details or unrelated preferences.
```

```
[USER-PRIVATE CONSTRAINT]
{PRIVATE_REQUIREMENT}
```

The public task specifies the trip, permitted itineraries, and selection scope. The public customer JSON contains the shared profile fields. The sender is told that Agent 2 has the inventory but does not receive the inventory rows. The private requirement is last, after the relay contract. The private segment starts at the private marker and includes the remaining chat-template suffix.

G.2 AIRLINE RECEIVER

System message.

```
You are Agent 2, the downstream decision-making agent in a two-agent
    workflow.

You must complete the public task using:
- the public task and environment information available to you,
- the communication received from Agent 1,
- and the tools or actions available in the environment.

Follow these rules:
1. Treat Agent 1's communication as the only source of user-specific
    private constraints that are not present in your public context.
2. Preserve and follow any decision-relevant user constraint
    communicated by Agent 1.
3. Do not assume or invent private user preferences that were not
    communicated to you.
4. If multiple actions satisfy the public task, use the communicated
    user constraint to choose among them.
5. Complete the task according to the environment's policies and
    available tools.
6. Do not mention hidden evaluation labels, counterfactual variants, or
    benchmark metadata.
7. Make the final decision based on the actual task state and the
    received communication, not on a fixed default preference.

Your goal is both to complete the public task and to respect the user's
    communicated constraint.
```

User message.

```
[PUBLIC TASK]
{PUBLIC_TASK}

[PUBLIC CONTEXT]
{PUBLIC_CUSTOMER_JSON}

[AVAILABLE ENVIRONMENT / TOOL INFORMATION]
Read-only source inventory; all rows are available.
itinerary_id | origin | destination | departure_time | arrival_time |
    cabin | price | stops | available_seats
{INVENTORY_ROWS}

[AIRLINE POLICY]
{AIRLINE_POLICY}

[DECISION PROCEDURE]
Check the inventory before choosing. Use the communicated private
    requirement, not a default preference. Compare departure clock times
     numerically; an equality is not strictly earlier. First exclude
    every row violating a communicated airport or time restriction.
    Cabin ranking is business > economy > basic_economy. If highest
```

```
    cabin is requested, first keep only the highest available cabin,
    then compare prices within that cabin. If lowest fare is requested,
    compare prices across ALL allowed rows, including nonstop: saying
    one stop is acceptable does not require one stop. Copy the exact
    itinerary_id from an existing row; never combine flight IDs yourself
    . Verify the selected row's actual departure, origin, cabin and
    price against the requirement. Briefly state the checks BEFORE the
    selection. Output exactly one JSON object in this schema: {"checks
    ":"brief factual constraint and candidate comparison","itinerary_id
    ":"exact ID copied from the inventory"}. No code fences, other JSON
    objects or extra final decisions. Do not infer a private preference
    if no communication is present.
```

The inventory field contains every candidate row in the world's fixed order. The policy field contains the complete source Airline policy; it is not a summary. The public task and customer context match the sender's public fields.

G.3 RETAIL SENDER

System message.

```
You are Agent 1, holding the user's private requirement for one retail
    replacement selection. Communicate its complete meaning to Agent 2.
    Do not invent requirements or choose an item. You do not have the
    replacement inventory. Do not execute tools or transactions.
```

User message.

```
[SHARED SELECTION TASK]
{SHARED_TASK_JSON}
Agent 2 has the replacement inventory, prices and public selection rules
    .

[RELAY OUTPUT CONTRACT]
Return one JSON object with the field private_constraint containing the
    complete requirement. Preserve all required values, the ordering of
    preferences and price priorities. Do not include account details or
    choose an item.

[USER-PRIVATE CONSTRAINT]
{PRIVATE_REQUIREMENT}
```

The shared-task object has exactly the fields `old_item_id`, `order_id`, `order_status`, `product_name`, and `scope`. The executed JSON is serialized with sorted keys and no optional spaces. The sender does not receive the replacement inventory.

G.4 RETAIL RECEIVER

System message.

```
You are Agent 2. Select exactly one replacement variant from the
    supplied retail inventory. Use Agent 1's communication as the only
    source of the user's private requirement. Use the actual inventory
    and the public selection rules; do not invent an item or private
    preference.
```

User message.

```
[SELECTION CONTEXT]
{SELECTION_CONTEXT_JSON}
```

```
[COMPLETE PRODUCT INVENTORY]
{COMPLETE_VARIANTS_JSON}

[SELECTION RULES]
This is selection before user confirmation, not transaction execution.
    The customer and target order item have already been identified.
    Only the specified item is to change; all other order items remain
    unchanged. The order is eligible by its pending/delivered status.
    Select an available variant of the same product, with an item_id
    different from the old item. The listed non-gift payment method is
    available for any price difference; no private price cap is implied.
     All prices are integer USD cents. For an ordered preference,
    compare the first preference before the second; neither becomes a
    hard requirement unless stated as such. Break preference ties by
    lowest catalog price when requested. Accept any exact price tie. Do
    not infer uncommunicated preferences, including a desire to keep the
     old options. Briefly explain the checks, then return exactly one
    JSON object with checks, order_id, old_item_id and new_item_id. Copy
     IDs exactly from the supplied context. Do not book, pay, modify the
     order or request confirmation.
```

The selection-context object contains `old_item_id`, `old_options`, `old_price_cents`, `operation`, `order_id`, `order_status`, `payment_method_id`, `payment_source`, `product_id`, and `product_name`. Each inventory object contains `available`, `item_id`, `options`, and `price_cents`. The complete array includes unavailable variants and the old item; public rules determine which entries can be selected. Both objects and the array use the same canonical JSON serialization.