BEYOND RECOVERABILITY: UNDERSTANDING FAILURES IN INTERMEDIATE REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Intermediate representations are widely used to support reasoning in large language and multimodal models, yet they are typically evaluated by final-answer accuracy, which provides little insight into why a representation succeeds or fails. We introduce PADC, a diagnostic framework for localizing failures along the representation-to-answer pathway by separating Preservation, Access, Downstream Dependence, and Correct Computation. The framework is designed to distinguish whether failure arises because task-relevant information is not retained in the representation, cannot be reliably recovered by the model, fails to control downstream prediction, or is available and influential but still not transformed according to the task rule. Across controlled synthetic tasks and six natural-domain benchmarks, matched interventions reveal distinct signatures at these interfaces. Removing task-relevant structure produces strongly heterogeneous effects across tasks, indicating that relevance alone does not determine downstream necessity. In controlled relational reasoning, local task state can be recovered perfectly while composition over that state remains unreliable, separating state access from successful downstream integration. Most notably, held-out donor-state interchange induces systematic prediction shifts beyond matched sham interventions, yet the model rarely remains correct under both original and counterfactual states, showing that causal sensitivity to a recovered state is not sufficient for task-correct computation. The same qualitative separation persists across additional model families. Our results suggest that intermediate-representation quality should not be reduced to final accuracy, information retention, or recoverability alone; a useful reasoning representation must expose task-relevant state in a form that not only affects prediction, but supports the computation required to produce the correct answer.

1 INTRODUCTION

Intermediate representations have become a core component of inference in large language models and multimodal models (Chen et al., 2024b;a; Zhang et al., 2025). Rather than mapping questions directly to answers, modern models increasingly externalize intermediate computation as textual chains of thought (Wang & Zhou, 2024; Zhou et al., 2024; Sprague et al., 2025), diagrams (Wu et al., 2024; Man et al., 2025), sketches (Hu et al., 2024; Menon et al., 2024), tables (Chen et al., 2019), and other structured states (Xu et al., 2025; Shen et al., 2025; Wu et al., 2025). These representations have improved performance across mathematical reasoning, algorithmic tasks, and multimodal problem solving, motivating extensive comparison of alternative representational forms. Such comparisons are commonly based on final-answer accuracy, with higher-performing representations treated as more effective substrates for reasoning. Final accuracy, however, identifies an outcome rather than the source of that outcome. In particular, the same reference task state and reasoning content can be encoded in different representational forms while producing substantially different predictions. Because the underlying content to be reasoned over is held fixed, such differences point to variation in how the representation preserves, exposes, and supports computation over that content. Understanding intermediate representations therefore requires more than determining which form performs better; it requires identifying which part of the representation-to-answer process accounts for the observed difference.

We formalize this problem through four operational properties. *Preservation* asks whether task-relevant information remains encoded in the intermediate representation. *Access* asks whether the model can recover that information from the representation. *Downstream Dependence* asks

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.

whether controlled changes to the recovered state alter the model's prediction. *Correct Computation* asks whether the resulting prediction agrees with the task-defined answer. These properties define the **Preservation–Access–Downstream Dependence–Correct Computation (PADC)** framework, whose purpose is to localize failures that are indistinguishable under final-answer accuracy alone. Importantly, PADC characterizes externally testable properties of the representation-to-answer interface rather than assuming four distinct internal modules. A representation may therefore contain task-relevant information without making it reliably accessible, an accessible state may exert limited control over prediction, and a state that does control prediction may still fail to support the computation required by the task.

We study these distinctions through controlled interventions on textual and visual intermediate representations. Matched representation comparisons first establish that holding the reference reasoning content fixed does not determine downstream performance. Relevant-versus-control erasure measures the consequence of removing task-relevant structure, while answer-free recovery under controlled encodings evaluates whether the corresponding state is accessible. We then intervene directly on recovered states using matched donor-state interchange with sham controls to test whether state changes propagate to prediction, and evaluate original and counterfactual conditions against independently recomputed task answers to assess correct computation. Across controlled synthetic tasks, six natural-domain benchmarks, and additional model families, the resulting pattern is not explained by a single notion of representation quality. Task-relevant structure varies substantially in its downstream necessity; locally recoverable information can still fail to support relational integration; and, most importantly, recovered states can systematically control prediction while the resulting state-to-answer mapping remains unreliable. These results distinguish information availability from computational use: making task-relevant state explicit or recoverable is not sufficient unless the model can transform that state according to the computation required by the task. PADC therefore provides a diagnostic basis for evaluating intermediate representations by how they support computation, rather than by final performance alone.

2 Related Work

Intermediate reasoning representations. Intermediate representations have become a central interface for reasoning in large language and multimodal models. Existing representations differ in how intermediate computation is externalized. Textual reasoning traces, such as chain-of-thought, express intermediate computation through natural language (Wei et al., 2022; Kojima et al., 2022). Visual representations use diagrams, sketches, grounded regions, and external workspaces to expose intermediate structures (Wu et al., 2024; Hu et al., 2024; Menon et al., 2024; Man et al., 2025). Structured representations, including tables and symbolic states, provide explicit organization of relational or algorithmic information (Chen et al., 2019; Xu et al., 2025; Shen et al., 2025; Wu et al., 2025). These forms differ in how information is encoded and exposed to the model, with text favoring sequential reasoning and visual or structured formats making spatial, relational, or state-based information explicit.

Analyzing intermediate representations. Recent work has developed intervention-based and information-based approaches to understand how intermediate representations affect model behavior. Studies of reasoning traces manipulate chain-of-thought through truncation, perturbation, replacement, and unlearning-based interventions to examine whether intermediate states reflect or influence final predictions (Lanham et al., 2023; Paul et al., 2024; Tutek et al., 2025). Other approaches analyze intermediate information through attribution, sufficiency, necessity, and causal representation analyses (Jia et al., 2026). Similar intervention strategies have been applied to visual and structured representations, where controlled modifications of grounding, layouts, sketches, diagrams, and reasoning workspaces identify components that affect model behavior (Wu et al., 2024; Hu et al., 2024; Menon et al., 2024; Man et al., 2025; Zhou et al., 2026). Existing analyses primarily evaluate whether intermediate representations influence predictions, contain useful information, or faithfully reflect model computation. They provide limited separation of different failure sources along the representation-to-answer pathway. Failures may arise when task-relevant information is lost during representation, when preserved information cannot be recovered by the model, when recovered information does not influence prediction, or when state-dependent predictions still fail to produce the correct answer.

3 THE PADC FRAMEWORK

Intermediate representations can affect task performance in fundamentally different ways, yet final-answer accuracy alone cannot distinguish among them. A representation may fail to retain task-relevant information, the model may be unable to recover information that is present, recovered information may not influence the model's downstream prediction, or the model may still fail to perform the required task computation correctly. To disentangle these sources of success and failure, we propose **PADC**, a diagnostic framework that characterizes the representation-to-prediction process along four dimensions: **Preservation**, **Access**, **Downstream Dependence**, and **Correct Computation**. Figure 1 provides an overview of the framework. We next formalize the task setting and then define each diagnostic dimension along the path from a reference reasoning trajectory to the final prediction.

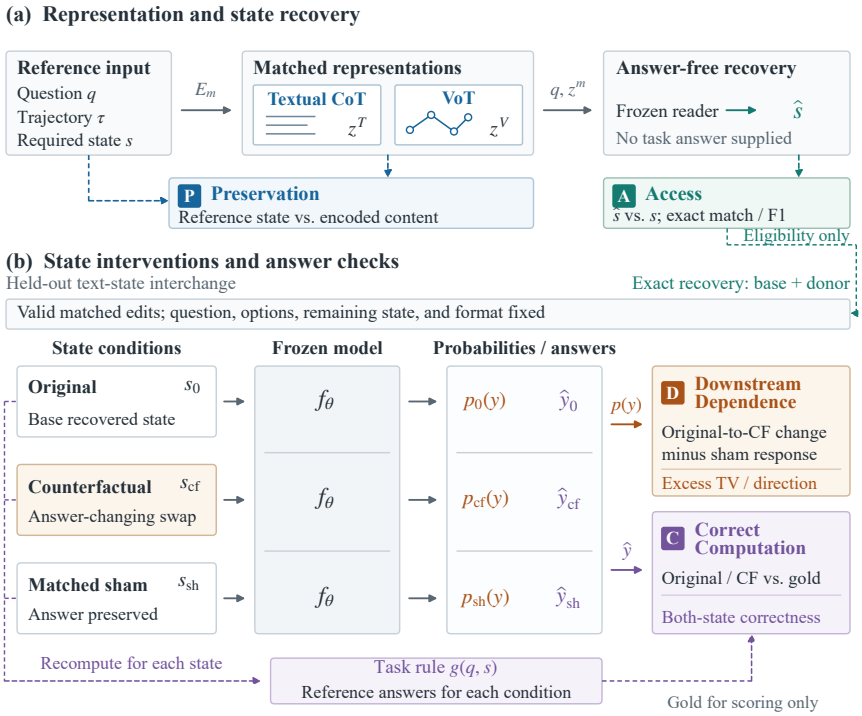

Figure 1: Overview of PADC. (a) Textual CoT and VoT encode the same reference trajectory. Preservation (P) checks whether the required state is retained, while Access (A) evaluates answer-free recovery against that state. (b) Held-out text-state interchange compares original, answer-changing counterfactual (CF), and answer-preserving sham conditions, requiring exact recovery of both base and donor states. Downstream Dependence (D) measures original-to-CF probability changes beyond sham using total variation (TV) and directional response. Correct Computation (C) evaluates whether both the original and CF predictions match independently recomputed task-rule answers, which are used only for scoring. The framework tests whether success at one interface is sufficient for success at the next.

3.1 TASK DEFINITION

For each sample i, let q_i denote the question, y_i the correct answer, and τ_i a reference reasoning trajectory containing the intermediate states associated with the task. The same trajectory is encoded into different intermediate representations:

$$z_i^m = E_m(\tau_i), \qquad m \in \{T, V\} \tag{1}$$

where T denotes Textual CoT and V denotes VoT. E_T renders the reference trajectory as text, whereas E_V renders it as a visual representation. Not all information in τ_i is required to determine the answer. We denote the answer-relevant part of the trajectory by the *task-relevant state* s_i:

$$s_i = P_{\kappa(i)}(q_i, \tau_i), \tag{2}$$

Table 1: Mapping from PADC components to the corresponding empirical questions and diagnostic experiments.

PADC Component	Empirical Question	Diagnostic Experiment
Preservation	Is Task-Relevant Information Preserved?	Matched Relevant-vs.-Irrelevant Erasure
Access	Can the Model Recover the Preserved State?	Answer-Free State Recovery and Relation-Order Probe
Downstream Dependence	Does the Recovered State Control Prediction?	Matched Donor-State Interchange with Sham Control
Correct Computation	Does the Model Use the State Correctly?	Paired Original–Counterfactual Correctness

where $\kappa(i)$ denotes the task associated with sample i, and $P_{\kappa(i)}$ is the corresponding state-extraction rule. The correct answer is determined by a task-specific computation over the question and the task-relevant state:

$$y_i = g_{\kappa(i)}(q_i, s_i), \tag{3}$$

where $g_{\kappa(i)}$ denotes the answer rule for task $\kappa(i)$. For example, consider a Breadth-First Search (BFS) task that asks for the next frontier. The task-relevant state s_i consists of the current frontier, graph connectivity, and visited-node status. The corresponding answer rule $g_{\kappa(i)}$ expands the current frontier along the graph, excludes previously visited nodes, and returns the next frontier. This formulation allows us to distinguish failures arising at different points between the reference trajectory and the final prediction. PADC operationalizes this distinction through four diagnostic dimensions—*Preservation*, *Access*, *Downstream Dependence*, and *Correct Computation*—which we define next.

3.2 Preservation, Access, Downstream Dependence, and Correct Computation

We organize PADC as four sequential diagnostic stages along the path from a reference reasoning trajectory to the final answer. These stages describe externally testable interfaces rather than distinct internal modules of the model. The first stage is *Preservation*, which asks whether the task-relevant state survives the transformation from the reference trajectory τ_i to the supplied representation z_i^m. The second stage is *Access*, which asks whether the model can recover that state from the representation, yielding a recovered state $\widehat{s}_{i,m}$. The third stage is *Downstream Dependence*, which asks whether changes in the recovered state induce changes in the model's prediction $\widehat{y}_{i,m}$. The final stage is *Correct Computation*, which asks whether the resulting prediction agrees with the task-defined answer y_i. We therefore analyze the following sequence:

$$\tau_i \xrightarrow{\text{Preservation}} z_i^m \xrightarrow{\text{Access}} \widehat{s}_{i,m} \xrightarrow{\text{Downstream Dependence}} \widehat{y}_{i,m} \xrightarrow{\text{Correct Computation}} y_i.$$

For the BFS example, Preservation tests whether the current frontier, graph connectivity, and visited-node status are retained in the representation. Access tests whether the model can recover these quantities. Downstream Dependence tests whether changing the recovered BFS state changes the predicted next frontier. Correct Computation tests whether that prediction matches the next frontier defined by the BFS update rule.

4 Experiment

We organize the experiments around the four PADC questions. A matched Textual CoT–VoT comparison first establishes that the same canonical reasoning trajectory can yield different downstream outcomes. Controlled interventions then test Preservation, Access, Downstream Dependence, and Correct Computation, followed by robustness checks across natural domains and model families. Table 1 summarizes the corresponding empirical questions and diagnostic experiments.

The following subsections describe the datasets and models, experimental design, and evaluation measures.

4.1 Datasets and Models

Datasets. Table 2 summarizes the datasets used across our experiments. We organize them into three complementary evaluation cohorts. The *primary matched-representation cohort* contains 1,401

Table 2: Overview of the primary datasets and evaluation cohorts.

Dataset	Task type	Samples
Primary matched-representation cohort		
ProofWriter	Deductive rule reasoning	600
ProntoQA	Multi-step deductive reasoning	300
CLRS Insertion Sort	Algorithmic state tracking	167
CLRS Binary Search	Algorithmic state tracking	167
CLRS BFS	Graph algorithm state tracking	167
Controlled diagnostic cohort		
Controlled self-describing states	Controlled graph-state reasoning	3,072
GraphInstruct (visual-access subset)	Graph-structure access	1,024
SyntheticMultiChannelGraph	Multi-relation graph reasoning	512
Natural-domain cohort		
Diagram-MMU	Scientific diagram reasoning	7,146
FlowchartQA	Flowchart reasoning	803
GSM-Symbolic	Mathematical reasoning	5,000
GeoQA3	Geometry reasoning	4,998
MapEval	Geospatial reasoning	1,000
TabFact	Table-based fact verification	12,839

instances from ProofWriter (Tafjord et al., 2021), ProntoQA (Saparov & He, 2022), and three CLRS tasks (Veličković et al., 2022): Insertion Sort, Binary Search, and BFS. For each instance, Textual CoT and VoT are derived from the same canonical reasoning trajectory, allowing us to compare representation formats while holding the reference reasoning content fixed. The *controlled diagnostic cohort* contains datasets designed to isolate specific properties of the PADC pipeline. Controlled self-describing states support state-level diagnostic analyses, the GraphInstruct visual-access subset probes recovery of graph structure from visual representations, and SyntheticMultiChannelGraph supports controlled multi-relation interventions. The *natural-domain cohort* contains six benchmarks spanning scientific diagrams, flowcharts, mathematical reasoning, geometry, maps, and tables: Diagram-MMU (Bo et al., 2026), FlowchartQA (Tannert et al., 2023), GSM-Symbolic (Mirzadeh et al., 2025), GeoQA3 (Chen et al., 2021), MapEval (Dihan et al., 2025), and TabFact (Chen et al., 2019). The full-scale natural-domain evaluation contains 31,786 instances. Complete sampling procedures and cohort-construction details are provided in Appendix B.1.

Models. We use Qwen2.5-VL-7B-Instruct (Bai et al., 2025a) as the primary model throughout the main experiments. To examine whether the observed PADC patterns are specific to a single model family, we additionally evaluate InternVL3.5-8B and Qwen3-VL-8B-Instruct (Wang et al., 2025; Bai et al., 2025b) in cross-model validation experiments. Qwen2.5-7B-Instruct (Yang et al., 2025) is used for symbolic-only diagnostics that do not require visual input. All models are evaluated in frozen form using their native input processors and greedy decoding, with no parameter updates during evaluation.

4.2 Diagnostic Experimental Design

We operationalize PADC through controlled interventions that target different stages of the representation-to-prediction pathway.

Preservation and Access. Preservation is probed by selectively removing task-relevant structures and measuring the resulting change in support for the correct answer relative to matched controls. Because a small erasure effect may also reflect redundancy or alternative reasoning paths, we complement this analysis with direct state-recovery tests. For Access, we ask the model to reconstruct predefined task-relevant states without access to the gold answer, and further vary the presentation between readable and opaque encodings while holding the underlying state fixed. Relation-order controls additionally distinguish local recovery from relational integration.

Downstream Dependence. We test whether the recovered state actually influences prediction using matched state interchange. The original task state is replaced by a donor state that changes the target relation while preserving the task format and answer space. A matched sham intervention controls for changes introduced by the interchange procedure itself. Prediction shifts beyond the sham condition provide evidence of Downstream Dependence.

Table 3: Task accuracy on the shared 1,401-instance cohort. Textual CoT and VoT are derived from the same canonical reasoning trajectory.

Task	n	Textual CoT	VoT	Repaired VoT	Repaired Extracted State
ProofWriter	600	0.562	0.657	0.340	0.330
ProntoQA	300	0.307	0.140	0.043	0.783
Insertion Sort	167	0.000	0.000	0.000	0.952
Binary Search	167	0.880	0.018	0.054	1.000
BFS	167	0.737	0.335	0.000	1.000

Correct Computation. To test whether the model correctly maps an available state to the task-defined answer, we evaluate paired original and counterfactual states and require correct predictions under both. We further use instruction and answer-interface controls to distinguish failures of task computation from failures caused by prompting or output delivery.

4.3 Evaluation Metrics

We use three complementary measures to evaluate different points along the PADC pathway: task performance, answer log-probability, and state recoverability.

Task performance. We retain each benchmark's canonical evaluator and report accuracy or exact match as appropriate. This measures whether the model ultimately produces the task-defined correct answer. The output interface is held fixed across the primary Textual CoT and VoT comparison.

Answer log-probability. To measure graded changes in downstream prediction, we compute the change in log-probability assigned to the gold answer under an intermediate condition c:

$$\mathrm{AIG}_i(c) = \log_2 p_\theta(y_i \mid q_i, c) - \log_2 p_\theta(y_i \mid q_i). \tag{4}$$

Positive values indicate increased support for the correct answer relative to the question-only baseline. For multi-token answers, we sum teacher-forced token log-probabilities.

State recoverability. We measure whether the model can reconstruct predefined fields of the task-relevant state s_i from the intermediate representation z_i^m without access to the gold answer. We report field-level F1 and exact recovery against the canonical task state. Together, these metrics capture progressively downstream aspects of PADC: recoverability measures access to task-relevant state, log-probability captures its effect on prediction, and task performance evaluates final correctness. Additional diagnostic measures and statistical procedures are reported in Appendix A.

5 Results

5.1 Same Reasoning, Unexpected Outcomes

A graph-structured task does not necessarily benefit from a visual reasoning representation. Breadth-First Search provides a particularly revealing case. Its computation depends explicitly on graph connectivity, the current frontier, and visited-node status, making a visual representation a natural candidate for exposing the relevant relational structure. If representation utility primarily followed task–representation compatibility, we would therefore expect VoT to perform at least comparably to Textual CoT on BFS. Instead, Textual CoT achieves an accuracy of 0.737, substantially higher than the 0.335 achieved by VoT, despite both being derived from the same canonical reasoning trajectory. This reversal is not isolated. Across the shared 1,401-instance cohort, the direction of the representation effect varies substantially by task. On ProofWriter, VoT achieves an accuracy of 0.657, exceeding the 0.562 of Textual CoT. On ProntoQA, Textual CoT achieves 0.307, compared with 0.140 for VoT. The gap is even larger on Binary Search, where Textual CoT reaches 0.880 while VoT achieves only 0.018. On Insertion Sort, both representations achieve zero accuracy. Fixing the underlying reasoning trajectory therefore determines neither the magnitude nor the direction of the representation effect. These results show that an intuitive match between task structure and representation form is insufficient to explain downstream performance.

In particular, explicitly visualizing relational structure does not guarantee that the model can exploit it effectively. Final-answer accuracy alone cannot distinguish whether such failures arise because

Table 4: Point estimates of matched task-relevance effects by synthetic relation channel and natural-domain dataset. Natural-domain results use the full-scale rerun with 2,000 source-cluster bootstrap replicates; synthetic results are unchanged.

Study: Synthetic		Study: Natural	
Channel or dataset	Matched effect M (bits, 95% CI)	Channel or dataset	Matched effect M (bits, 95% CI)
Adjacency	$0.660\ [-0.493, 1.840]$	Diagram-MMU	$0.643\ [0.608, 0.678]$
Direction	$0.744\ [0.296, 1.358]$	FlowchartQA	$2.236\ [2.015, 2.461]$
Value	$0.374\ [-0.015, 0.817]$	GSM-Symbolic	$0.131\ [-0.045, 0.302]$
Group	$1.336\ [0.644, 1.946]$	GeoQA3	$0.027\ [0.015, 0.037]$
		MapEval	$2.777\ [2.528, 3.012]$
		TabFact	$2.795\ [2.743, 2.845]$

Table 5: Effect of relation order on local readout and downstream path judgment across 512 paired test instances. Differences are ordered minus permuted, in percentage points (pp), with post-hoc, unadjusted 95% bootstrap intervals.

Measure	Ordered (%)	Permuted (%)	Difference (pp)
Target-edge readout	100.00	100.00	0.00
Target-path judgment	58.01	47.46	10.55 $[4.69, 16.41]$
Other-path judgment	57.42	45.90	11.52 $[5.66, 17.58]$

task-relevant information is not preserved, cannot be recovered, fails to influence prediction, or is followed by an incorrect task computation.

5.2 Is Task-Relevant Information Preserved?

Task-relevant structures are not uniformly necessary under erasure. A natural expectation is that removing information known to be relevant to the task should substantially reduce answer support. Table 4 shows that this is not always the case. In the synthetic setting, Group and Direction relations produce matched effects of 1.336 and 0.744 bits, while Adjacency and Value yield weaker effects of 0.660 and 0.374 bits. The discrepancy is even larger across natural-domain tasks. Removing task-relevant structures reduces answer support by 2.795 bits on TabFact, 2.777 bits on MapEval, and 2.236 bits on FlowchartQA, but by only 0.131 bits on GSM-Symbolic and 0.027 bits on GeoQA3.

All erased structures are task-relevant by construction, yet their downstream necessity differs sharply. A relevant structure can have little effect under removal when the remaining representation contains redundant evidence, supports an alternative reasoning path, or allows the missing information to be reconstructed. Task relevance therefore does not imply erasure necessity. This distinction also limits what can be inferred from the intervention. A large matched-erasure effect provides evidence that the removed information is retained and consequential for the current prediction. A small effect, however, does not imply failed Preservation, because the information may still be present but replaceable by other evidence. Erasure alone therefore cannot establish whether the model can recover the retained task-relevant state, motivating a separate test of *Access*.

5.3 Can the Model Recover the Preserved State?

Perfect local recovery does not guarantee successful relational integration. We construct a relation-composition task in which each instance contains several pairwise relations and asks whether a target path relation follows from them. The same set of relations is presented either in an order aligned with the target path or in a permuted order. We separately evaluate whether the model can recover an individual target relation and whether it can combine the relations to make the correct path judgment. Table 5 shows that, across 512 paired test instances, target-relation readout is perfect under both presentation orders. Path judgments, however, remain sensitive to relation order. Target-path accuracy is 58.01% with ordered relations and 47.46% with permuted relations, a difference of 10.55 percentage points with a 95% bootstrap interval of $[4.69, 16.41]$. The other-path judgment exhibits a similar gap of 11.52 percentage points. The model can therefore recover the relevant local relation while still failing to combine the available relations reliably.

Table 6: Orthogonal Access probe on 1,536 paired items. The underlying state is held fixed while only its presentation encoding changes.

Encoding	Readable	Opaque v1	Opaque v2	Sham
Samples	1,536	1,536	1,536	1,536
Answer-free readout	0.6875	0.3701	0.3688	0.3695

Table 7: Recovery, Downstream Dependence, and Correct Computation. Recovery confirmation uses 1,221 eligible graphs; held-out interchange uses 845 pairs from a disjoint cohort. Estimates and 95% bootstrap intervals are on the probability scale. Both-state correctness requires correct original and counterfactual answers.

Analysis	Metric	Estimate	95% CI
Recovery confirmation (1,221 graphs)	Original-state accuracy	0.5160	0.4945–0.5381
	Counterfactual accuracy	0.3773	0.3567–0.3972
	Both states correct	0.0747	0.0630–0.0866
Held-out interchange (845 graph pairs)	Excess TV over sham	0.0296	0.0245–0.0347
	Excess directional response	0.0661	0.0554–0.0772
	Both states correct	0.0678	0.0556–0.0805

Recovery diagnostics identify failures before recovery, but not after it. We evaluate this distinction on 900 fresh graphs using frozen diagnostic rules, with 300 instances constructed for each failure type (Appendix B.2, Table 9). The diagnostics correctly identify 283 of 300 Preservation failures and 285 of 300 Access failures, corresponding to identification rates of 94.3% and 95.0%. In contrast, only 40 of 300 post-recovery failures are identified, while 260 remain unresolved. This result shows that recovery-based diagnostics are effective when the failure concerns whether task-relevant information is preserved or recovered. Once recovery has succeeded, however, the same diagnostics provide little evidence about what happens downstream. Successful state recovery therefore does not establish that the recovered state influences the model's prediction. We further verify the Access manipulation with an orthogonal encoding control. Holding the underlying state fixed, answer-free recovery reaches 0.6875 under a readable encoding but remains near 0.37 under two opaque encodings and the sham condition. This confirms that the probe is sensitive to the accessibility of the representation rather than to changes in the underlying state.

Access therefore identifies whether task-relevant information can be recovered, but not whether that information governs the subsequent prediction.

5.4 Does the Recovered State Control Prediction?

Downstream Dependence asks whether the model's prediction is causally sensitive to the task-relevant state. We evaluate this with matched state interchange. For each paired instance, the original state is replaced with a donor state that changes the target relation while preserving the task format and answer space. A matched sham condition controls for effects introduced by the interchange procedure itself.

State interchange induces systematic prediction shifts. Table 7 reports results on 845 held-out matched pairs. Relative to the sham condition, donor-state interchange produces an excess total-variation shift of 0.0296, with a 95% CI of $[0.0245, 0.0347]$, and an excess directional response of 0.0661, with a 95% CI of $[0.0554, 0.0772]$. Both effects exceed the prespecified dependence margins. The model's prediction therefore responds systematically to controlled changes in the task-relevant state, beyond changes attributable to the interchange procedure. This establishes Downstream Dependence, but does not imply that the resulting prediction is task-correct.

5.5 Does the Model Use the State Correctly?

Correct Computation asks whether the model maps the supplied task-relevant state to the answer required by the task rule. We evaluate this with paired original and counterfactual states, requiring the model to answer both correctly.

Table 8: Cross-model replication on the same source cohorts, with model-specific donor pairs. Entries are means and paired 95% bootstrap intervals. P is the relevant-minus-semantic-control NLL contrast; A is the key-minus-opaque exact-recovery gain in percentage points (pp); D measures Downstream Dependence and C measures Correct Computation.

Outcome	InternVL3.5-8B	Qwen3-VL-8B
P: Relevant-erasure effect (bits)	4.278 (4.219–4.336)	6.959 (6.809–7.112)
A: Key-rescue recovery gain (pp)	1.630 (0.910–2.410)	5.340 (4.230–6.510)
D: Excess TV over sham	0.031 (0.019–0.044)	0.126 (0.088–0.168)
D: Excess directional response	0.074 (0.048–0.103)	0.260 (0.182–0.344)
C: Both states correct	0.026 (0.008–0.046)	0.160 (0.116–0.208)

State dependence rarely translates into correctness under both states. Table 7 reports low both-state correctness in both evaluation cohorts. Among 1,221 recovery-confirmed instances, the model answers both the original and counterfactual states correctly in only 7.47% of cases, with a 95% CI of $[0.0630, 0.0866]$. The held-out interchange cohort yields a similar rate of 6.78%, with a 95% CI of $[0.0556, 0.0805]$. The same table also shows that this low paired correctness coexists with measurable Downstream Dependence. The model responds systematically to changes in the task-relevant state, yet this response rarely produces the task-correct answer under both states. State dependence is therefore insufficient evidence of correct state-to-answer computation.

Table 7 further reports single-state accuracies of 0.5160 for the original state and 0.3773 for the counterfactual state in the recovery-confirmed cohort. These higher marginal accuracies indicate that the model can answer individual states correctly, while the much lower paired correctness shows that correct behavior is not stable across controlled state changes. A complementary prompt-based diagnostic on ProntoQA is reported in Appendix C.3.

Operation prompts improve ordinary accuracy relative to a general reasoning instruction, but paired counterfactual correctness remains only 1/30 for Overview and 2/30 for Scratchpad. Procedural guidance can therefore improve marginal accuracy without establishing reliable state-to-answer computation.

5.6 Robustness Across Domains and Models

The preceding experiments establish PADC dissociations across natural-domain tasks. We further evaluate whether these patterns persist across model families.

The P/A/D/C separation persists across models. Table 8 reports results for InternVL3.5-8B and Qwen3-VL-8B-Instruct. In both models, relevant erasure exceeds matched controls, readable state interfaces improve answer-free recovery, and donor-state interchange induces prediction shifts beyond sham interventions, while both-state correctness remains low. These results indicate that the separation among Preservation, Access, Downstream Dependence, and Correct Computation is not specific to the primary model, although effect magnitudes vary across tasks and model families.

6 Conclusion

We introduced PADC to separate Preservation, Access, Downstream Dependence, and Correct Computation in intermediate reasoning representations. Across controlled synthetic tasks, natural-domain benchmarks, and multiple model families, these properties are empirically dissociable: task-relevant information can remain encoded without being reliably recoverable, locally accessible information can fail to support downstream integration, and states that causally influence prediction can still yield incorrect task outcomes. These findings show that intermediate-representation quality cannot be reduced to modality, information presence, or final accuracy alone. What matters is whether task-relevant information survives encoding, becomes accessible to the model, governs prediction, and is transformed according to the task rule. The broader lesson is that making reasoning explicit is not sufficient; an effective intermediate representation must make the right state computationally usable in the right way.

AI USE STATEMENT

AI tools were used to assist with language revision, LaTeX editing, and literature retrieval and discovery. All experimental claims, numerical values, and citations were independently checked against the project artifacts or primary publication records. The authors remain fully responsible for the study design, analysis, interpretation, and final manuscript.

REFERENCES

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025a. doi: 10.48550/arXiv.2502.13923. URL https://arxiv.org/abs/2502.13923.

Shuai Bai et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025b. doi: 10.48550/arXiv.2511.21631. URL https://arxiv.org/abs/2511.21631.

Weihao Bo, Shan Zhang, Yanpeng Sun, Jie Liu, Yongke Yao, Jinhao Du, Wei He, Kai Zou, Zechao Li, and Jingdong Wang. Diagram-MMU: A multi-modal benchmark for scientific diagrams. *arXiv preprint arXiv:2608.12262*, 2026. doi: 10.48550/arXiv.2608.12262. URL https://arxiv.org/abs/2608.12262.

Jiaqi Chen, Jianheng Tang, Jinghui Qin, Xiaodan Liang, Lingbo Liu, Eric P. Xing, and Liang Lin. GeoQA: A geometric question answering benchmark towards multimodal numerical reasoning. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 513–523, 2021. doi: 10.18653/v1/2021.findings-acl.46. URL https://aclanthology.org/2021.findings-acl.46/.

Qiguang Chen, Libo Qin, Jin Zhang, Zhi Chen, Xiao Xu, and Wanxiang Che. M^3CoT: A novel benchmark for multi-domain multi-step multi-modal chain-of-thought. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pp. 8199–8221. Association for Computational Linguistics, 2024a. doi: 10.18653/v1/2024.acl-long.446. URL https://aclanthology.org/2024.acl-long.446/.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. Tabfact: A large-scale dataset for table-based fact verification. *arXiv preprint arXiv:1909.02164*, 2019.

Yangyi Chen, Karan Sikka, Michael Cogswell, Heng Ji, and Ajay Divakaran. Measuring and improving chain-of-thought reasoning in vision-language models. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 192–210, Mexico City, Mexico, June 2024b. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.11. URL https://aclanthology.org/2024.naacl-long.11/.

Mahir Labib Dihan, Md Tanvir Hassan, Md Tanvir Parvez, Md Hasebul Hasan, Md Almash Alam, Muhammad Aamir Cheema, Mohammed Eunus Ali, and Md Rizwan Parvez. MapEval: A map-based evaluation of geo-spatial reasoning in foundation models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 13774–13813. PMLR, 2025. URL https://proceedings.mlr.press/v267/dihan25a.html.

Yushi Hu, Weijia Shi, Xingyu Fu, Dan Roth, Mari Ostendorf, Luke Zettlemoyer, Noah A. Smith, and Ranjay Krishna. Visual sketchpad: Sketching as a visual chain of thought for multimodal language models. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-4423. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/fb82011040977c7712409fbdb5456647-Abstract-Conference.html.

Jinghan Jia, Joe Benton, and Eric Easley. Faithfulness as information flow: Evaluating and training faithful chain-of-thought reasoning. *arXiv preprint arXiv:2605.24286*, 2026. URL https://arxiv.org/abs/2605.24286.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in Neural Information Processing Systems*, 35: 22199–22213, 2022.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, et al. Measuring faithfulness in chain-of-thought reasoning, july 2023. *URL http://arxiv. org/abs/2307.13702*, 2023.

Yunze Man, De-An Huang, Guilin Liu, Shiwei Sheng, Shilong Liu, Liang-Yan Gui, Jan Kautz, Yu-Xiong Wang, and Zhiding Yu. Argus: Vision-centric reasoning with grounded chain-of-thought. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14268–14280, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Man_Argus_Vision-Centric_Reasoning_with_Grounded_Chain-of-Thought_CVPR_2025_paper.html.

Sachit Menon, Richard Zemel, and Carl Vondrick. Whiteboard-of-thought: Thinking step-by-step across modalities. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 20016–20031. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.1117. URL https://aclanthology.org/2024.emnlp-main.1117/.

Iman Mirzadeh, Keivan Alizadeh-Vahid, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models. In *International Conference on Learning Representations*, volume 2025, pp. 94743–94765, 2025.

Debjit Paul, Robert West, Antoine Bosselut, and Boi Faltings. Making reasoning matter: Measuring and improving faithfulness of chain-of-thought reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 15012–15032. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-emnlp.882.

Abulhair Saparov and He He. Language models are greedy reasoners: A systematic formal analysis of chain-of-thought. *arXiv preprint arXiv:2210.01240*, 2022.

Zhenyi Shen, Hanqi Yan, Linhai Zhang, Zhanghao Hu, Yali Du, and Yulan He. CODI: Compressing chain-of-thought into continuous space via self-distillation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 677–693. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.36. URL https://aclanthology.org/2025.emnlp-main.36/.

Zayne Sprague, Fangcong Yin, Juan Diego Rodriguez, Dongwei Jiang, Manya Wadhwa, Prasann Singhal, Xinyu Zhao, Xi Ye, Kyle Mahowald, and Greg Durrett. To CoT or not to CoT? chain-of-thought helps mainly on math and symbolic reasoning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/ead542f13a38179d1b55b88610f959a1-Abstract-Conference.html.

Oyvind Tafjord, Bhavana Dalvi, and Peter Clark. ProofWriter: Generating implications, proofs, and abductive statements over natural language. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 3621–3634, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.317. URL https://aclanthology.org/2021.findings-acl.317/.

Simon Tannert, Marcelo G. Feighelstein, Jasmina Bogojeska, Joseph Shtok, Assaf Arbelle, Peter W. J. Staar, Anika Schumann, Jonas Kuhn, and Leonid Karlinsky. FlowchartQA: The first large-scale benchmark for reasoning over flowcharts. In *Proceedings of the 1st Workshop on Linguistic Insights from and for Multimodal Language Processing*, pp. 34–46, 2023. URL https://aclanthology.org/2023.limo-1.5/.

Martin Tutek, Fateme Hashemi Chaleshtori, Ana Marasović, and Yonatan Belinkov. Measuring chain of thought faithfulness by unlearning reasoning steps. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 9946–9971, 2025.

Petar Veličković, Adrià Puigdomènech Badia, David Budden, Razvan Pascanu, Andrea Banino, Misha Dashevskiy, Raia Hadsell, and Charles Blundell. The CLRS algorithmic reasoning benchmark. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 22084–22102. PMLR, 2022. URL https://proceedings.mlr.press/v162/velickovic22a.html.

Weiyun Wang et al. InternVL3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025. doi: 10.48550/arXiv.2508.18265. URL https://arxiv.org/abs/2508.18265.

Xuezhi Wang and Denny Zhou. Chain-of-thought reasoning without prompting. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-2123. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/7a8e7fd295aa04eac4b470ae27f8785c-Abstract-Conference.html.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837, 2022. doi: 10.52202/068431-1800. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html.

Haoyi Wu, Zhihao Teng, and Kewei Tu. Parallel continuous chain-of-thought with jacobi iteration. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 914–926. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.47. URL https://aclanthology.org/2025.emnlp-main.47/.

Wenshan Wu, Shaoguang Mao, Yadong Zhang, Yan Xia, Li Dong, Lei Cui, and Furu Wei. Mind's eye of LLMs: Visualization-of-thought elicits spatial reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-2866. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/a45296e83b19f656392e0130d9e53cb1-Abstract-Conference.html.

Yige Xu, Xu Guo, Zhiwei Zeng, and Chunyan Miao. SoftCoT: Soft chain-of-thought for efficient reasoning with LLMs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, pp. 23336–23351. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.1137. URL https://aclanthology.org/2025.acl-long.1137/.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2025. doi: 10.48550/arXiv.2412.15115. URL https://arxiv.org/abs/2412.15115.

Ruohong Zhang, Bowen Zhang, Yanghao Li, Haotian Zhang, Zhiqing Sun, Zhe Gan, Yinfei Yang, Ruoming Pang, and Yiming Yang. Improve vision language model chain-of-thought reasoning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, pp. 1631–1662. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.82. URL https://aclanthology.org/2025.acl-long.82/.

Yiyang Zhou, Haoqin Tu, Zijun Wang, Zeyu Wang, Niklas Muennighoff, Fan Nie, Chaorui Deng, Shen Yan, Haoqi Fan, Yejin Choi, James Zou, Cihang Xie, Huaxiu Yao, and Qinghao Ye. When visualizing is the first step to reasoning: Mira, a benchmark for visual chain-of-thought. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26154–26164, 2026.

Zhanke Zhou, Rong Tao, Jianing Zhu, Yiwen Luo, Zengmao Wang, and Bo Han. Can language models perform robust reasoning in chain-of-thought prompting with noisy rationales? In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-3936. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/dfaa29ed28dfa175bcc5e2a54aa199f8-Abstract-Conference.html.

A Additional Evaluation Metrics

A.1 Within-format representation efficiency

To characterize how efficiently a representation supports the correct answer within its own native format, we define the representation information density

$$\mathrm{RID}_i^m = \frac{\mathrm{AIG}_i(z_i^m)}{N_i^m}, \tag{5}$$

where N_i^m denotes the native size of representation m for sample i: text tokens for Textual CoT and visual tokens for VoT.

RID measures answer log-probability gain per native representation unit. Because text and visual tokenizations are not commensurate, RID is used only for comparisons within a representation format, such as budget or compression analyses. We do not interpret RID as a compute-normalized measure for comparing Textual CoT and VoT directly.

A.2 Instance specificity

A representation may improve answer probability without necessarily carrying information specific to the current instance. To test whether the observed utility depends on instance-specific content, we compare each representation with a representation drawn from another sample of the same task.

Let $\pi(i)$ denote a matched sample index from the same task. We define

$$\mathrm{IS}_m = \mathbb{E}_i \left[\mathrm{AIG}_i(z_i^m) - \mathrm{AIG}_i(z_{\pi(i)}^m) \right]. \tag{6}$$

The question, answer interface, and task are held fixed; only the intermediate representation is replaced. A positive IS_m therefore indicates that the representation provides more support for the correct answer when it belongs to the current instance than when it is replaced by a task-matched representation from another instance. We regard instance specificity as supported when the paired confidence interval lies entirely above zero.

A.3 Task-specific evaluation details

For ProofWriter and ProntoQA, final-answer performance is evaluated against the corresponding task labels. The CLRS-derived experiments use task-specific targets: the final predecessor array for Insertion Sort, the stored terminal low pointer for Binary Search, and the final parent array for BFS. The next-frontier example in Section 3 illustrates the framework; it is not the output target of the BFS cohort in Table 3. Structured outputs are serialized for model interaction and evaluated with the experiment-specific parsers. Strict schema matching and semantic array rescoring are distinct evaluation rules, as detailed in Appendix C.2. An unchanged schema across representation conditions does not by itself exclude representation-dependent formatting failures.

For the historical Binary Search entries in Table 3, accuracy denotes agreement with these stored terminal-pointer labels; this differs from correctness for the literal archived question. The later repair decomposition uses an explicit terminal-low-pointer question in every arm (Appendix C.1); the historical comparison and the corrected decomposition therefore have different question contracts.

Across the primary Textual CoT and VoT conditions, the output schema, field order, delimiters, and evaluator are held fixed within each task. Output procedures nevertheless differ across experiments. Free-form generation and answer-free state recovery use deterministic greedy decoding. Candidate-scoring experiments instead evaluate fixed candidate answers by their model likelihoods, with the resulting scores or normalized probabilities used for prediction and intervention contrasts. Candidate scoring is not grammar-constrained open generation. The answer-interface audit and unified factorial archive identify their scoring procedures explicitly below.

B Dataset Construction and Statistical Procedures

B.1 Dataset construction and filtering

ProofWriter uses 600 balanced examples from the official open-world depth-5 test set (Tafjord et al., 2021). The ProntoQA holdout contains 300 examples from the generated Easy source, with 150

True and 150 False labels and 100 examples at each depth of 2, 3, and 4. Its sealed cohort manifest records seed 314159 and six separate smoke examples. The CLRS cohorts use the official length-16 training-generation protocol with seed 42 and contain 167 examples per task (Veličković et al., 2022). The sampling balances the first output. BFS removes padded hint states and zero-transition cases. The full GraphInstruct diagnostic pool contains 3,200 balanced examples from 16 tasks at two graph sizes; the visual-access analysis uses a fixed 1,024-example subset. The synthetic confirmation contains 512 held-out examples covering adjacency, direction, value, and group relations. The full-scale natural-domain rerun contains 31,786 records, with per-dataset counts in Table 2. MapEval includes 400 Visual, 300 Textual, and 300 offline supplied-context API records; Textual and API reuse identical local question/context records. GeoQA3 includes all three processed splits. This expanded rerun overlaps the earlier cohorts and is not an untouched holdout. The full-scale records contain 11,064 target-source groups within domains. Appendix F.1 reports the full-scale heterogeneity and redundancy analyses, including sensitivity to shared target/distractor sources and duplicated MapEval API records.

Experiment-specific cohorts. The 512-instance synthetic relation-erasure confirmation, the 1,536-instance auxiliary factorial intervention (Appendix E.1), and the 900-instance diagnostic validation below are different experiment cohorts, despite sharing the SyntheticMultiChannelGraph task family. Table 2 summarizes the primary datasets; the experiment-specific counts here distinguish the auxiliary cohorts used to test the PADC stages. Cross-model Preservation uses 16,384 weighted-path diagrams per model, whereas cross-model Access uses 1,536 reversible-code graphs (Appendix H). Recovery confirmation starts from 1,536 graphs and retains 1,221 eligible recovered graphs; the fresh donor-interchange study starts from 3,000 graphs and forms 845 pairs (Appendix D). Counts of graphs, pairs, and evaluation records refer to different units. Reuse within the primary 1,401-instance cohort and between the earlier recovery analyses is retained and is not treated as independent replication.

B.2 Construction and rules for the 900-item diagnostic validation

This text-based recovered-state validation uses Qwen2.5-VL-7B-Instruct on 900 fresh graphs, with 300 graphs per injected construction. Each construction contains 100 examples from each of three tasks: outgoing degree, directed reachability, and maximum incident value. Within each task, 50 graphs have six nodes and 50 have eight. Task families were selected from earlier held-out results using own-state accuracy below 0.5, access accuracy at least 0.6, excess TV above 0.01, and excess directional response above 0.02. New test responses were not used to select examples, tasks, or thresholds. Every retained graph admits a solver-verified answer-changing substitution and an answer-preserving sham with the same signed edit magnitude.

The Preservation construction masks one answer-changing relation value. Readable reformatting retains the missing value, whereas oracle repair restores the complete state. The Access construction retains both the reversibly encoded state and the complete decoding key; readable repair decodes this information without adding semantic content. This differs from a hidden-key intervention, where revealing a missing key supplies additional information. The post-recovery construction supplies a complete readable state on the selected task families. It does not guarantee that every response exhibits a downstream failure; correct responses and failed readouts remain in its denominator.

The fixed rules are applied in priority order. A Preservation signature requires failed base and readable exact recovery but successful oracle exact recovery. An Access signature requires failed base exact recovery and successful readable exact recovery. The post-recovery signature requires successful base exact recovery, field recovery at least 0.75, excess TV above 0.01, excess directional response above 0.02, and task accuracy at most 0.5 averaged over option rotations. All remaining cases are unresolved. The rules use measured diagnostic features, not construction labels or task identifiers, and were frozen before inference. Rescue in the first two rules denotes answer-free state recovery, not final-answer correctness.

Table 9 reports the validation outcomes summarized in Section 5.3. Of the 900 items, 292 remain unresolved: 17 in the Preservation construction, 15 in Access, and 260 in the post-recovery construction. The identification rates measure agreement with injected construction labels, not recovery of mutually exclusive internal failure causes. The experiment supports separation of the tested Preservation and Access signatures but does not establish reliable three-way diagnosis. Intervals for diagnostic accuracy and macro-F1 use 10,000 bootstrap replicates stratified by injected construction, counting unresolved predictions as errors.

Table 9: Frozen diagnostic rules on 900 fresh graphs. Each controlled construction contains 300 instances. Unresolved cases count as diagnostic errors; construction labels do not guarantee that the corresponding behavioral signature is realized.

Construction	Identified	Identification rate	Unresolved
Preservation	283 / 300	94.3%	17
Access	285 / 300	95.0%	15
Post-recovery failure	40 / 300	13.3%	260

Table 10: Primitive-state repair controls on the 1,401-instance cohort. Accuracy is reported by task; external execution supplies additional computed results and is not an access-only operation.

Task	Primitive state	Format only	Selection only	Execution appended
ProofWriter	0.265	0.252	0.385	0.612
ProntoQA	0.500	0.503	0.500	0.523
Insertion Sort	0.000	0.000	0.000	0.964
Binary Search	0.449	0.635	1.000	0.856
BFS	0.000	0.000	0.000	1.000

B.3 Statistical reporting for task-level contrasts

Statistical procedures are specific to each analysis. The archived task-level accuracy-intervention analysis records paired sample sizes, exact McNemar tests, and unadjusted and Benjamini–Hochberg-adjusted p-values. These adjustments do not apply to the exploratory E4 intact-order and ProntoQA prompt contrasts, whose intervals are post-hoc and unadjusted. The existing-data task-level accuracy and answer-log-probability reanalysis uses 10,000 paired sample-bootstrap replicates, keeping the two conditions for each sample together; it is a post-hoc analysis rather than a new preregistered experiment.

Natural-domain per-dataset erasure intervals use 2,000 source-cluster bootstrap replicates. Full-scale aggregate intervals use 10,000 replicates, either conditioning on the six observed domains or additionally resampling domains; these inferential targets are distinguished in Appendix F.1. The auxiliary factorial contrasts use 2,000 paired item-bootstrap replicates. E4 intact-order contrasts use 10,000 state-bootstrap replicates. Recovery-confirmation analyses average edits and rotations within each graph, whereas donor-interchange analyses average rotations within each pair before 10,000 pair-bootstrap replicates. The diagnostic-validation bootstrap is stratified by injected construction as specified above. There is no single task-cluster bootstrap rule shared by all synthetic experiments.

A positive point estimate whose interval includes zero is treated as uncertain. Equivalence additionally requires a prespecified effect-size bound; it cannot be inferred merely from a nonsignificant difference. The visual-control bounds and multiplicity treatment are specified in Appendix G.1.

C Repair and Recovery Diagnostics

C.1 State-repair results

Table 3 reports paired task scores for the complete repair pipeline, subject to the historical Binary Search question–label limitation described above, and Table 14 gives the corresponding answer-information changes. Repair can alter the fields exposed to the answer model, their selection, and the computational support supplied with them. It therefore cannot be interpreted as a pure readability intervention.

A separate decomposition reused the same primitive extraction within each task and compared formatting, field selection, and external execution (Table 10). Execution appended externally computed task results. Insertion Sort and BFS remained at zero under formatting or selection alone, but reached 0.964 and 1.000 after execution. Binary Search used an explicit terminal-low-pointer question consistently across these arms, resolving the archived question/label contract mismatch. These are post-hoc controls on a previously evaluated cohort, not a new untouched confirmation experiment.

C.2 READER AND ENCODING CONTROLS

The archived answer-interface control uses the canonical own state, a donor state, or a null payload for each of 167 BFS and 167 Insertion Sort instances. The own state includes the final result. The condition named constrained decoding scores four complete arrays, including the gold answer, rather than enforcing an array grammar during open generation. Its distractor constructor can produce arrays semantically identical to the gold answer but with different whitespace. Table 11 separates the archived score from a post-hoc semantic check of a unique complete array. For JSON-object generation, this check accepts an array embedded in a response even when the original object parser fails; it is a relaxed evaluation, not a rerun or an improvement in the generated answer. Multiple-choice generation is retained in the archive but is not pooled with these array-based checks because its original evaluator parses an option letter.

Table 11: Answer-interface audit with the own canonical payload. Entries are correct counts out of 167 per task. Rescoring compares a unique complete array with the target; the input already contains the final state.

Task	Interface	Archived score	Array rescoring
BFS	Candidate-array scoring	144	164
Insertion Sort	Candidate-array scoring	166	167
BFS	JSON-object generation	0	123
Insertion Sort	JSON-object generation	0	103

Repair minimality is a deterministic reanalysis of saved outputs. Table 12 retains the archived scorer and each operation's applicability rule. Filtering selects an answer-like span, formatting normalizes whitespace and punctuation, symbolic normalization changes numeric punctuation, and reordering sorts array values. The last two are not generally semantics-preserving: deleting commas can turn `[1,2]` into `[12]`, and sorting a predecessor or parent array changes the node-to-value mapping. The full transformation combines these operations. Its apparent accuracy increase from 27.41% to 27.47% reflects a denominator change with the same correct count. The archived gain statistic counts newly correct cases without subtracting newly incorrect cases, so we do not report it as a paired net gain. These results describe the implemented postprocessing only.

Table 12: Archived repair-minimality output processing on 1,401 saved answers. Correct counts and accuracies use applicable outputs only. Semantic changes in some operations prevent an interpretation as isolated, meaning-preserving repair.

Operation	Applicable	Correct	Accuracy (%)
No repair	1,401	384	27.41
Filtering only	1,401	384	27.41
Formatting only	1,401	384	27.41
Symbolic normalization only	1,401	384	27.41
Reordering only	174	0	0.00
Full transformation	1,398	384	27.47

The nominal full-factorial archive requires a separate provenance check. The intended 36-cell script uses the same candidate scorer for every output-interface label, sorts words for canonicalization, reverses characters for shuffling, and adds only a prefix for the wrong-task condition. It also partitions both samples and cells by shard. The files stored under that experiment name contain 1,536 relation-probe records without the intended factor assignments and an eight-arm summary that exactly matches a separate offline output-processing analysis. That summary records 384/1,401 correct outputs only when preservation, visibility, and parser handoff are all enabled, and zero otherwise. Its readout rate is a nonempty-string check. These artifacts cannot establish execution or effects of the intended 36-cell design.

The separate unified factorial run evaluates all 16 implemented conditions on each of 1,536 states, producing 24,576 records. Its factors are complete versus degraded state, raw versus canonical presentation, matched versus character-reversed state, and three instruction variants; the reversed-state conditions use only the constrained-label instruction. Degradation retains the first half of whitespace-separated tokens. In this implementation, the canonical label triggers no presentation

transformation, and direct checks confirm identical raw/canonical prompts and predictions for every matched condition pair. All three instruction variants score the same candidate labels rather than generate free-form responses. Table 13 reports one copy of the duplicated presentation conditions. The results characterize truncation, character reversal, and instruction sensitivity under candidate scoring; they do not identify separate presentation or decoding effects. This experiment is also distinct from the external interface intervention in Appendix E.1.

Table 13: Descriptive outcomes from the unified factorial archive. Each cell has 1,536 states; raw and canonical presentation conditions are identical and shown once. All instruction variants use candidate scoring.

State condition	Instruction	Complete (%)	Degraded (%)
Matched	Constrained label	36.39	36.07
Matched	Explanation	35.22	35.16
Matched	Answer only	38.67	35.81
Character reversed	Constrained label	34.77	34.70

The multi-reader probe uses the same 200 examples for Qwen and InternVL readers. Raw versus access-only accuracy is 0.240 versus 0.275 for Qwen and 0.260 versus 0.260 for InternVL. The access-only reader ablation uses 600 examples and varies only presentation operations. Raw accuracy is 0.428; whitespace-only, newline-only, delimiter-only, chunking-only, and full access-only conditions obtain 0.435, 0.378, 0.420, 0.435, and 0.353, respectively. The semantic-invariance audit covers all 600 examples. These controls are therefore interpreted as evidence about reader-interface dependence, rather than as a universal benefit of formatting repair.

The orthogonal readable-versus-opaque probe uses 1,536 paired items and four encodings of the same state. Answer-free readout is 0.6875 for the readable encoding, 0.3701 and 0.3688 for two opaque encodings, and 0.3695 for the sham encoding. Semantic hashes, token lengths, and lossless decode round trips match across encodings. The probe therefore quantifies a presentation-dependent access contrast at the reader interface; it does not establish that the underlying state is absent from the opaque strings.

Table 14 reports the complete state-repair comparison. The repaired path improves ProntoQA and all three CLRS tasks, but reduces usable information on ProofWriter. On a 30-example fixed BFS smoke test, the repaired path reaches exact state recovery and matches the ground-truth-state condition; this smoke test is not used as the primary estimate. On a 30-pair ProntoQA counterfactual smoke test, the original and repaired direct-visual conditions both obtain 0.10 accuracy; this small test is reported only as a diagnostic.

C.3 ProntoQA instruction and counterfactual diagnostics

The archived ProntoQA instruction study evaluates 300 examples per representation under no additional instruction, a general reasoning instruction, and an operation prompt. Table 15 reports ordinary task accuracy. Gains are paired accuracy differences between the operation and general instructions, expressed in percentage points. Their 95% percentile intervals use 10,000 paired sample-bootstrap resamples and are post-hoc and unadjusted. Operation prompts increase ordinary-answer accuracy from 17.33% to 25.00% for Overview and from 13.00% to 20.33% for Scratchpad.

A separate evaluation within this instruction study contains 30 original–counterfactual pairs per representation. All 30 original examples also occur in the corresponding 300-example ordinary-accuracy cohort, so these pairs do not constitute an independent replication. Under the operation prompt, both-state correctness is 1/30 for Overview and 2/30 for Scratchpad. These counts require correct predictions for both the original and answer-changing counterfactual states and are distinct from the marginal accuracies in Table 15. They support the limited conclusion in Section 5.5 that better ordinary accuracy under procedural guidance does not establish reliable correctness across state changes. This instruction study is distinct from the repaired-path counterfactual smoke test described in Appendix C.2.

Table 14: A repaired extracted-state path improves four of five tasks. The exception is ProofWriter, where the repaired path reduces answer information.

Task	VoT Acc.	Repaired-state Acc.	ΔAIG (bits, 95% CI)
ProofWriter	0.657	0.330	-2.145 $[-2.406, -1.882]$
ProntoQA	0.140	0.783	$+2.314$ $[2.152, 2.475]$
Insertion Sort	0.000	0.952	$+63.020$ $[61.950, 64.065]$
Binary Search	0.018	1.000	$+6.576$ $[6.184, 6.976]$
BFS	0.335	1.000	$+3.715$ $[3.291, 4.170]$

Table 15: ProntoQA instruction diagnostics on 300 examples per representation. Accuracy is in percent; gains compare the operation prompt with the general instruction, in percentage points (pp). Intervals are post-hoc and unadjusted.

Representation	No instruction	General	Operation	Gain (pp)
Overview	6.67	17.33	25.00	7.67 $[4.67, 11.00]$
Scratchpad	4.33	13.00	20.33	7.33 $[4.33, 10.67]$

D Held-out State Interchange

D.1 Recovery-confirmation cohort

The recovery-confirmation cohort contains 1,536 fresh graphs. Qwen2.5-VL-7B-Instruct recovers 1,453 task states exactly, and 1,221 of these admit a solver-valid answer-changing counterfactual. Original-state accuracy is 0.5160 (95% CI $[0.4945, 0.5381]$), counterfactual-state accuracy is 0.3773 ($[0.3567, 0.3972]$), and both-state correctness is 0.0747 ($[0.0630, 0.0866]$). These measurements establish a post-recovery performance gap but do not, by themselves, identify Downstream Dependence.

D.2 Fresh held-out donor interchange

The held-out interchange study uses 3,000 fresh graphs disjoint from the formal and recovery-confirmation cohorts. Exact state recovery succeeds on 2,848 graphs. Pairing requires both endpoints to be exactly recovered, mutually graph-disjoint, and semantically valid under an answer-changing single-relation substitution; 845 pairs satisfy these criteria. Pair selection uses only recovery and semantic-validity information, never downstream responses.

Each counterfactual replaces one task-relevant relation while holding the question, answer options, remaining state, and serialization template fixed. The matched sham preserves the answer while matching the direction and magnitude of the edit. The counterfactual answer is recomputed from the base question and hybrid state rather than inherited from the donor. Option rotations are averaged within pair before 10,000 pair-bootstrap resamples.

The held-out cohort was evaluated in two fixed batches of 1,000 and 2,000 source graphs. The second batch was added after the first result; all initial pairs were retained and no cross-batch rematching was performed. The earlier 167-pair analysis is reported separately in Appendix I.2 and is not part of this held-out sample. The final total of 3,000 graphs was therefore not fixed before the initial result. The pooled intervals below summarize the two collected batches under the pair-bootstrap procedure; they are not adjusted for the decision to extend the study.

The first batch contains 249 pairs and gives excess TV 0.0332 (95% CI $[0.0239, 0.0434]$), excess directional response 0.0739 ($[0.0540, 0.0954]$), and both-state correctness 0.0622 ($[0.0412, 0.0853]$). The additional batch contains 596 pairs and gives 0.0281 ($[0.0222, 0.0343]$), 0.0629 ($[0.0506, 0.0762]$), and 0.0701 ($[0.0549, 0.0856]$), respectively. Each batch separately has lower interval bounds above the specified dependence margins, while both-state correctness remains low.

For candidate-probability vectors, total variation is half the L_1 distance. Excess TV subtracts the matched-sham response from the answer-changing interchange response. Excess directional response is defined analogously for the probability margin of the counterfactual answer over the original answer. The pooled estimates are

$$\Delta\mathrm{TV} = 0.0296\ [0.0245, 0.0347], \qquad \Delta\mathrm{Dir} = 0.0661\ [0.0554, 0.0772].$$

Table 16: Auxiliary factorial interface intervention on 1,536 paired graphs. P controls preservation, A controls readability, and I controls external state injection. Readout is independent answer-free probe accuracy, not full-state exact recovery.

Preservation P	Access A	Injection I	Accuracy	Readout
$-$	$-$	$-$	0.471	0.379
$-$	$-$	$+$	0.471	0.379
$-$	$+$	$-$	0.471	0.625
$-$	$+$	$+$	0.467	0.625
$+$	$-$	$-$	0.471	0.378
$+$	$-$	$+$	0.469	0.378
$+$	$+$	$-$	0.471	0.847
$+$	$+$	$+$	0.626	0.847

Both exceed the prespecified dependence margins of 0.01 and 0.02. Both-state correctness is 0.0678 ($[0.0556, 0.0805]$). These estimates identify Downstream Dependence at the externally supplied state–answer interface; they do not identify an internal neural variable or circuit.

E Preservation and Access Controls

E.1 Auxiliary factorial interface intervention

This auxiliary interface study paired the same question, candidate set, and canonical state across eight interface arms for each of 1,536 SyntheticMultiChannelGraph items. $P-$ corrupted the target-state panel, $A-$ used a reversible opaque encoding, and $I-$ blocked state injection into answer computation while retaining the artifact for the independent reader probe. The eight arms produced 12,288 evaluations. Candidate scoring guaranteed structural parseability; construction flags were intervention audits rather than independent behavioral measurements.

Table 16 reports all arms. At $P + A+$, enabling injection increased accuracy by 0.1556 (95% CI $[0.1270, 0.1836]$). At $P + I-$, enabling readable access increased probe readout by 0.4694 ($[0.4518, 0.4873]$); at $A + I-$, enabling preservation increased readout by 0.2217 ($[0.2064, 0.2363]$). Intervals use 2,000 paired bootstrap resamples over item IDs. The interventions operate at an external interface and can affect multiple functional stages. In particular, the P-channel audit did not pass every non-target-pixel and lexical-leakage check, so the factorial does not identify independent internal circuits or perfectly isolated preservation effects.

F Natural-Domain Erasure and Redundancy Analysis

F.1 Matched erasure effects and domain heterogeneity

Table 4 and Figure 2 report the relation-erasure contrasts from the 31,786-record natural-domain rerun alongside the unchanged synthetic results. All point estimates are positive, but only Direction and Group among synthetic channels, and all natural domains except GSM-Symbolic, have intervals excluding zero. Per-dataset natural-domain intervals use 2,000 source-cluster bootstrap replicates. The sham-minus-irrelevant contrasts are -0.016, -0.115, 0.037, -0.009, -0.338, and -0.012 bits for Diagram-MMU, FlowchartQA, GSM-Symbolic, GeoQA3, MapEval, and TabFact, respectively. Excluding the duplicate API branch leaves 700 MapEval records and a matched effect of 2.752 bits ($[2.535, 2.973]$). The equal-domain natural mean is 1.435 bits, compared with a row-weighted mean of 1.442 bits. With 10,000 source-cluster bootstrap replicates, the equal-domain interval is $[1.372, 1.499]$ for the six fixed observed domains; additionally resampling domains gives the wider hierarchical interval $[0.513, 2.346]$.

Shared entities complicate domain-level uncertainty: the full-scale records contain 11,064 target-source groups, and target/distractor reuse connects observations within each domain. A post-hoc sensitivity analysis uses shared-entity dyadic variance estimates, assuming that errors for observations with no shared target or distractor source are uncorrelated. REML with modified Hartung–Knapp inference gives a random-effects mean of 1.432 bits ($[0.055, 2.809]$), with $Q = 11{,}515.812$ (5 degrees of freedom), $I^2 = 99.96\%$, and $\tau^2 = 1.716$ bits^2. Excluding the duplicate API branch gives 1.428

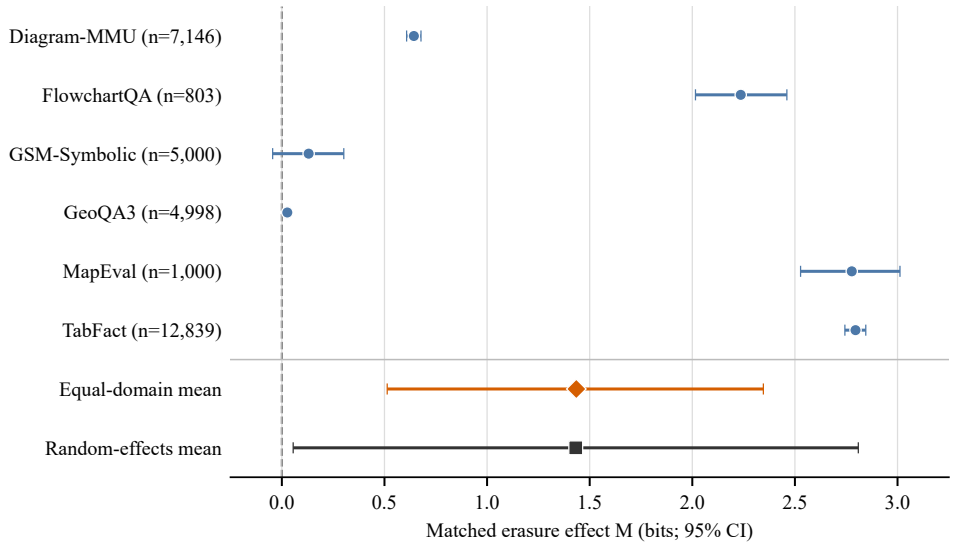

Figure 2: Matched relation-erasure effects vary across six natural domains in the full-scale rerun ($n = 31{,}786$). Circles show per-dataset means with 95% source-cluster bootstrap intervals (2,000 replicates). The diamond shows the equal-domain mean with a hierarchical domain/source bootstrap interval (10,000 replicates); the square shows the REML mean with a modified Hartung–Knapp interval using shared-entity dyadic variances. These summaries have different inferential targets and assumptions.

bits ($[0.057, 2.800]$) and $I^2 = 99.96\%$. The random-effects mean targets a broader domain-level population under additional assumptions, rather than only the six observed domains. Although its interval excludes zero, the exploratory new-domain prediction interval is $[-2.497, 5.361]$, and four of six leave-one-domain-out mean intervals include zero. Moreover, I^2 measures heterogeneity relative to sampling error and can increase as within-domain precision improves. These results therefore do not establish a uniformly positive effect in new domains.

F.2 Natural-domain redundancy analysis

All 31,786 natural-domain records are annotated with three zero-to-one structural proxies: Relation Necessity, Localizability, and Redundancy. Redundancy measures potential answer-relevant evidence that remains outside the erased task-state panel, including evidence in the surviving question and answer choices. Rules are frozen before joining per-instance erasure effects and do not use answer labels, but aggregate results were already known; this is a post-hoc exploratory analysis, not a preregistered test. The full-scale rules remove the obsolete 32-line cutoff from TabFact's localizability proxy. For MapEval's 600 Textual/API records, relation and locality proxies use question keywords; a zero Redundancy score means that no shortcut is detected, not that surviving evidence is proven absent. These new-branch labels have low confidence, and none of the full-scale proxy labels has independent human validation.

The primary regression is equal-row OLS with all three proxies, using $M_i = \mathrm{NLL}_i(\mathrm{relevant}) - \mathrm{NLL}_i(\mathrm{irrelevant})$ as the outcome. Intervals use target-source-cluster robust covariance with a finite-sample correction and $t(G-1)$ critical values, where $G = 11{,}064$. Relation Necessity has coefficient -0.335 (95% CI $[-0.515, -0.155]$), Localizability -0.124 ($[-0.252, 0.004]$), and Redundancy -2.149 ($[-2.265, -2.033]$). The pooled model explains 5.95% of per-instance effect variance. Adding domain indicators reduces the Redundancy coefficient to -0.455 ($[-0.530, -0.379]$). The fixed-effects model has $R^2 = 0.1665$, but the structural proxies add only 1.20 percentage points over the domain-only model and explain 1.42% of its remaining within-domain variation.

The association is sensitive to the weighting and dependence assumptions. Clustering over the six domains with $t(5)$ critical values widens the pooled Redundancy interval to $[-5.430, 1.132]$; the domain-fixed-effects interval is $[-0.811, -0.099]$. Giving each domain equal total weight yields coefficients of -2.251 ($[-2.530, -1.972]$) without domain indicators and -0.483 ($[-0.647, -0.318]$) with them, using target-source-cluster intervals. For the latter equal-domain fixed-effects model, the domain-cluster interval is $[-1.019, 0.054]$. Shared target/distractor dyadic sensitivity intervals for

Table 17: No general visual-carrier gain is confirmed in the broad control study, and its tested layout effects fall within the prespecified ± 0.15-bit equivalence bounds. These contrasts differ from E4's path-card ordering intervention. The access manipulation succeeds, but it does not produce the predicted demand–access law.

Contrast	Effect (bits)	95% CI	Decision
Visual carrier without layout	-0.087	$[-0.185, 0.022]$	Inconclusive
Text layout	0.055	$[-0.023, 0.141]$	Equivalent to zero
Image layout	0.007	$[-0.049, 0.067]$	Equivalent to zero
Carrier–layout interaction	-0.048	$[-0.119, 0.022]$	Equivalent to zero
Easy − Medium access probe	0.046	$[0.022, 0.072]$	Expected direction
Medium − Hard access probe	0.286	$[0.248, 0.326]$	Expected direction
Demand × access interaction	-0.259	$[-0.516, -0.043]$	Opposite direction

the equal-row models are $[-2.282, -2.016]$ and $[-0.529, -0.380]$, respectively. The larger sample size therefore does not remove the uncertainty associated with having only six domains.

Excluding MapEval's duplicate API branch leaves 31,486 records and gives Redundancy coefficients of -2.124 ($[-2.231, -2.017]$) and, with domain indicators, -0.465 ($[-0.532, -0.398]$). Excluding both new MapEval context branches leaves 31,186 records and gives -2.098 ($[-2.202, -1.994]$) and -0.472 ($[-0.536, -0.408]$), respectively. Applying the unchanged legacy proxy rules to all records also retains the negative association (-2.155 without and -0.472 with domain indicators). A post-freeze audit finds that the label-free inverse of the GSM-Symbolic option generator matches all 5,000 reference answers; this establishes an available construction shortcut, not its use by the model. Overall, the analysis supports an association between alternative evidence and erasure sensitivity, but neither a causal mechanism nor a sufficient explanation of cross-domain heterogeneity.

G Carrier and Layout Controls

G.1 Visual carrier and layout controls

The E4 experiment complements the broad controls with a specific relation-order intervention. Each of 512 states contains two three-edge paths. Edge cards explicitly encode endpoints and OPEN/CLOSED status; permutation changes their order while keeping the target card's position fixed. The archive contains 6,144 records across ordered/permuted, intact/deleted, and local-readout/path-query conditions. In the intact conditions, target-edge readout is 512/512 in both layouts; target-path correctness is 297/512 versus 243/512, and other-path correctness is 294/512 versus 235/512. Table 5 reports the post-hoc paired contrasts, with 10,000 state-level bootstrap resamples. The probe covers a target edge rather than all path endpoints and relations.

The predeclared E4 analysis instead examines relevance-by-deletion contrasts and their interaction with layout. In the ordered layout, deletion lowers target-path accuracy by 7.23 percentage points and other-path accuracy by 17.38 points. The resulting accuracy interaction is -0.1016 (95% CI $[-0.1816, -0.0234]$); the permuted-layout interaction is -0.0098 ($[-0.0430, 0.0215]$). The corresponding NLL interactions are 0.2103 bits ($[0.0114, 0.4084]$) and -0.3768 bits ($[-0.6061, -0.1487]$). The layout interaction is therefore -0.0918 on accuracy ($[-0.1680, -0.0156]$) and 0.5870 bits on NLL ($[0.4182, 0.7468]$). These outcomes do not support a uniform relevance-specific deletion effect across measures. The exploratory intact-order result is consequently reported separately from the predeclared mechanism test, and neither is a general claim about all visual layouts.

Table 17 reports the broad visual controls. The first four intervals are familywise adjusted. The registered equivalence rule requires the multiplicity-controlled interval to lie wholly within $[-0.15, +0.15]$ bits; "equivalent" therefore means within this practical tolerance, not an exactly zero effect. The last three intervals are the registered 95% intervals for the access study.

H Cross-Model Replication

InternVL3.5-8B and Qwen3-VL-8B-Instruct replicate the same directional P/A/D/C contrasts on the corresponding source cohorts. Preservation uses the same 16,384 weighted-path diagrams and compares task-relevant erasure with semantically matched irrelevant erasure. Access uses 1,536

Table 18: Cross-model replication of the P/A/D/C contrasts. Entries are means with paired 95% bootstrap intervals. The donor-interchange rows use model-specific exactly recovered subsets and should not be interpreted as direct effect-size rankings between models.

Outcome	InternVL3.5-8B	Qwen3-VL-8B-Instruct
P: Relevant-erasure effect (bits)	4.278 (4.219–4.336)	6.959 (6.809–7.112)
A: Key-rescue recovery gain (pp)	1.63 (0.91–2.41)	5.34 (4.23–6.51)
D: Excess TV over sham	0.031 (0.019–0.044)	0.126 (0.088–0.168)
D: Excess directional response	0.074 (0.048–0.103)	0.260 (0.182–0.344)
C: Both states correct	0.026 (0.008–0.046)	0.160 (0.116–0.208)

Table 19: Oracle-state downstream-dependence profiling on the full 1,401-sample cohort. The full-completeness profile supports aggregate sensitivity to instance-specific state information, with heterogeneous task-level accuracy effects; the 100-sample run is reported only as a smoke test.

Task	Matched accuracy	Shuffled accuracy	NLL separation (bits)
BFS	0.000	0.000	1.422
Binary Search	1.000	0.054	12.661
Insertion Sort	0.000	0.000	0.891
ProntoQA	0.647	0.500	1.915
ProofWriter	0.738	0.333	3.484
Aggregate	–	–	$3.687\ [3.427, 3.952]$

reversible-code graphs and measures the gain from revealing the inverse key while holding the encoded payload fixed. Downstream Dependence is evaluated on each model's own exactly recovered subset, producing 124 disjoint donor pairs for InternVL3.5-8B and 172 for Qwen3-VL-8B-Instruct.

Relevant erasure produces larger losses than the matched irrelevant control in both models, and revealing the inverse key improves exact recovery. Donor interchange also produces positive excess TV and directional response. Across the four co-primary Downstream Dependence tests, Bonferroni-adjusted one-sided 98.75% lower bounds exceed 0.01 for excess TV and 0.02 for excess directional response. Both-state correctness remains low.

The replicated result is structural rather than quantitative. Preservation, Access, and Downstream Dependence remain separable from Correct Computation across the tested models, but the effect magnitudes differ. Because donor interchange is conditioned on each model's own exactly recovered subset, the D estimates characterize different model–state interfaces and are not directly comparable as model rankings.

I Additional and Earlier Analyses

I.1 Oracle-state downstream-dependence profiling

The dedicated matched-versus-shuffled oracle-state profile evaluates whether the answer distribution is sensitive to the current instance's structured state when recovery errors are removed. The primary condition uses the full 1,401-sample cohort with complete state information (2,802 JSONL records, one matched and one shuffled record per sample). The 100-sample matched-versus-shuffled run is retained as a smoke test only. The 50%, 75%, partial-state, and wrong-task conditions use the same 1,401-sample cohort as robustness analyses. The aggregate NLL separation is 3.687 bits with a 95% confidence interval of $[3.427, 3.952]$; the matched-minus-shuffled accuracy delta is 0.318 with interval $[0.281, 0.353]$. Both preregistered gates pass. Table 19 reports the task-level breakdown.

I.2 Earlier recovery-cohort interchange analysis

The recovery-confirmation cohort contained 1,536 new graphs, of which 1,453 were exactly recovered by a reader given the target-panel text and public task metadata. The primary population comprised the 1,221 exactly recovered graphs with a solver-valid answer-changing counterfactual. Edits and option rotations were averaged within each graph before graph-level aggregation. At the original option

Table 20: Large-cohort robustness sweeps for oracle-state completeness and controls. All conditions use the 1,401-sample cohort.

Condition	Samples	Accuracy delta	NLL separation (bits)
Completeness 50%	1,401	0.121 $[0.084, 0.157]$	1.571 $[1.393, 1.752]$
Completeness 75%	1,401	0.168 $[0.132, 0.203]$	1.999 $[1.799, 2.201]$
Completeness 100%	1,401	0.318 $[0.281, 0.353]$	3.687 $[3.427, 3.952]$
Partial state	1,401	0.121 $[0.084, 0.157]$	1.571 $[1.393, 1.752]$
Wrong-task state	1,401	0.087 $[0.056, 0.118]$	0.530 $[0.355, 0.707]$

order, 606 of these 1,221 graphs were answered incorrectly despite exact upstream recovery. This observation diagnoses recovery-followed-by-answer failure; it does not establish absent downstream dependence or complete reading in the answer-stage forward pass.

Counterfactual changes produced TV response 0.0877 ($[0.0799, 0.0954]$), directional response 0.1239 ($[0.1088, 0.1388]$), and semantic prediction-flip rate 0.1202 ($[0.1066, 0.1334]$). Related-field reading and full-field copy discrimination reached only 0.7725 and 0.8048, respectively, whereas oracle-result handoff reached 0.9517. Neither the registered near-zero-response conditions nor the reliable-reading conditions passed. The prespecified subgroup passing all reading probes contained 150 graphs; its TV response was 0.1291 ($[0.0976, 0.1639]$) and both-state correctness was 0.1508 ($[0.1017, 0.2025]$). Accordingly, poor task accuracy cannot be interpreted as evidence that recovered information was unused.

Two distinct sham controls must be distinguished. The serialization-only sham had TV response 0.0829 ($[0.0776, 0.0880]$). The separately reported 0.0545 excess ($[0.0469, 0.0622]$) compared counterfactual response with an operator-matched sham at the original option order only; it is not counterfactual TV minus serialization-only-sham TV. The 0.0877 counterfactual estimate instead averages all option rotations. The operator sham matched edit type but not necessarily the magnitude of the numeric change.

The subsequent conditional-interchange experiment formed 167 non-reused base–donor pairs from the same 1,453 correctly recovered graphs. Of the 334 paired graphs, 317 also belonged to the 1,221-graph counterfactual analysis; this protocol therefore provides an additional intervention on the recovery cohort, not an independent sample replication. A single high-level relation variable was substituted while the question, options, remaining state, and dense JSON template were fixed. Answer-changing and answer-preserving substitutions matched the direction and magnitude of the numeric edit. The counterfactual answer was computed from the base question and hybrid state, rather than copied from the donor's original answer. TV is half the L_1 distance between candidate-probability vectors. Directional response is the change in the probability margin between the counterfactual and original correct answers; excess response subtracts the corresponding sham effect. Option rotations were averaged within each pair before 10,000 pair-bootstrap resamples.

Counterfactual and sham TV were 0.0440 and 0.0230, respectively, while same-value substitution produced zero response. The excess-TV and excess-direction lower bounds exceeded the registered thresholds of 0.01 and 0.02. Counterfactual interchange accuracy was 0.4476 ($[0.3862, 0.5090]$), and downstream field reading was 0.6003 ($[0.5793, 0.6228]$). These estimates are conditional on exact upstream recovery and valid matched substitutions. They are neither population-wide effects over all source graphs nor identification of aligned neural variables. The earlier recovered-state v3 run with parser rejection and gold-state fallback is excluded.