

Exact journal delivery after a lossy handoff

DiadkoShmek · Engineering follow-up · 23 October 2026

An explicit transport repair completed **8/8** fully traced optimal plans, compared with **0/8** through model-written prefix-truncated notes. The repaired workflow used 44.1% fewer recorded input tokens over the same eight source instances. This is a small, developer-controlled engineering result for a conventional lossless transport, not evidence of a reading effect or general agent improvement.

Codex substantially developed the adapter, protocol, execution and analysis under the author's direction. No external peer review or independent replication is claimed. The original negative reading-transfer results remain unchanged.

1. The failure that selected this repair

The [reading-transfer study](#) often preserved semantic warnings while losing exact claim IDs and support links after truncating the handoff. Its post-hoc diagnosis found receivers that could name an optimal work set but could not provide its required evidence. A useful repair must therefore deliver those records before asking the next model to use them.

The method supplies a design requirement: stored history must remain accessible at the decision that depends on it. Here that requirement becomes an executable transport contract. The actual encoding uses ordinary zlib and Base85; neither algorithm nor lossless journal transport is claimed as a new invention.

2. What changed

The freeform workflow uses a fresh ordinary sender to write a note and a fresh receiver to finish the task from its actual 3200-byte UTF-8 prefix. Truncation is disclosed. The sender has the same pre-event source history and the original 12000-byte whole-note safety bound.

The repaired workflow has no sender model. The host serializes the complete canonical journal, compresses it, and transfers an envelope plus its external SHA-256 descriptor within the same 3200-byte bound. The descriptor's 64 ASCII bytes count toward that bound. A new Python process reads the actual wire file, checks the external pin, enforces a hard decompression bound and replays the unchanged reference runtime. Its restored history then enters the ordinary receiver prompt. Every repaired prompt is byte-equal to the corresponding full-history prompt; no post-event oracle or ready answer is substituted.

The encoder receives no future withdrawal, work budget, inventory values, gold plan or model reply. It preserves the entire supplied history or raises HOLD. There is no fallback retrieval, sidecar journal or alternative codec selection. The decoded prompt is longer than the transport; its full native model usage is included below. This is a workflow intervention, not a fixed-input comparison of model reasoning ability.

3. Registration, task and coverage

The [protocol](#), adapter, unchanged generator/scorer, materialized source cases and gold, analysis and schedule were locally committed before model inference at `ff80b89998d0a9e007cdc371d1bc679d64fab66c`. The [freeze](#) binds their hashes. This is a developer-controlled precommitment, not independent preregistration.

The original input commit remains a historical Git object. Its incremental source-history bundle was withdrawn from current downloads for author privacy. Exact source inspection

requires a full historical clone containing `ff80b89998d0a9e007cdc371d1bc679d64fab66c` and its base `f592059bb6cbcf7df06cfd579cba007305157ec9`. Recorded scientific hashes and results are unchanged; historical commit IDs are not an independent timestamp authority or additional model trial.

Four new generator seeds each have two counterfactual histories: eight source instances, sixteen episodes, one repetition per condition. These are new instances of the **already inspected planning class**, not independent new task families. Tasks require scoped current claims, branch-specific orientation, revision and transitive support, a later source retirement, work prerequisites and budgeted value optimization. Exhaustive and recursive solvers agree; both are developer-authored and share the same task law.

Targets requested GPT-6.1 Sol/high through the existing isolated authenticated Codex CLI adapter. Every target had tools disabled. There were 24 captured calls, 24 terminal responses, no target errors and no unexpected tools. No failed outcome was replaced. One repaired receiver ran per case; freeform used one sender and one receiver per case. Wall duration was 588.354 seconds with at most two episodes concurrent. The common Codex harness remains; requested route identity is not independent provider attestation.

4. Predeclared primary and observed results

The primary requires an actually submitted optimal plan, all prerequisite and cost constraints, minimum cost among value ties, and the exact active evidence closure for every selected item. A correct explanation with an empty plan does not pass. Only a fully checked proposed plan is written and read back; no real release work executes.

Outcome	Freeform note	Exact journal transport
Source instances / episodes per condition	8	8
Fully traced optimal completion, primary	0/8	8/8
Optimal submitted work set, ignoring missing citations	5/8	8/8
Failed stages / target errors	0 / 0	0 / 0
Captured model calls	16	8
Retained transport bytes, including external pin where used	3200	3018-3034
Native input tokens, whole workflow	385,323	215,343
Cached input tokens, subset of input	26,496	11,776
Native output tokens	39,081	4,870

Paired primary outcomes: eight repaired wins, zero losses, zero ties. Each of the four source clusters has two repaired completions and zero freeform completions. No population significance or universal reliability estimate is inferred from four clusters.

The freeform condition's zero primary score does **not** mean it never found an optimal work selection: five submitted work sets were optimal. None supplied the complete required citation closure. This distinction is part of the task contract, not a retrospective reinterpretation of failure.

Input volume fell by 44.1% and output by 87.5% across the equal eight-case cohorts. Removing the sender stage is an explicit part of the intervention. The repaired input per fully completed case was 26,917.875 native tokens; the freeform ratio is undefined because its completion denominator is zero. Token and cache counters are not a monetary invoice or an energy measurement.

5. What the result supports

For these small repetitive journals, a complete lossless carrier fit where a freeform prefix omitted required records. Delivering those records through the host enabled the ordinary receiver to finish every registered case. The repair also removed eight model-written sender calls. This is a concrete operational effect of preserving accessible evidence, measured as completed work.

No book reading occurred in either new condition. The experiment cannot establish that reading the ontology improves performance, or that ontological terminology is necessary for the repair. A generic transport delivering the same restored history would produce the same prompt. The result does not change the earlier reading study's scores, sample or interpretation.

6. Boundaries and practical use

Compression is effective on this repeated schema. Incompressible or larger history can exceed the cap; the implementation then raises HOLD before inference. The decoder caps raw output at 256 KiB and enforces the existing runtime's finite operation limits. This is not a million-record memory system.

The external pin binds bytes relative to a caller's trusted descriptor; it does not authenticate authors or prevent coherent replacement of both descriptor and payload. Compression provides no confidentiality. A snapshot does not prove that the live journal remains current; live admission still needs its current head and authorized action. Deployment in House services, sustained natural-task benefit, voice, identity and subjective experience were not measured.

The small sample, single known generator, developer-authored oracle and common model family limit generalization. Wider natural-task evaluation and an information-equivalent generic transport control would be separate work.

7. Generic-file control and codec capacity

A separate post-cohort offline replay wrote and read the full canonical history as an ordinary file. For all eight original sources, the file and decoded wire produced identical canonical data and byte-identical receiver prompts. This control made zero model calls; it establishes input parity, not a new 8/8 model score for the ordinary file. A shared file already provides the information. The uncompressed file plus its digest takes 15,687 bytes, outside this particular 3200-byte channel. The codec result does not establish superiority when that larger or shared storage is available.

Eight added repetitive 512-character logs still fit for every source. Eight equally long deterministic variable logs exceeded capacity for every source; the adapter returned HOLD without inference or fallback. Ordinary files retained the full history in both cases. This finite replay demonstrates a dependence on compressibility and the need for an explicit storage contract. It is not a new held-out model cohort. [Control code and every result.](#)

8. A usable repository-check continuation

The separate [practical command](#) runs four actual offline gates on this repository, records their process exits and input fingerprints, and restores the evidence in a fresh process. Its retained run checked 79 files and correctly rejected stale evidence after a controlled modification of the transport source in a temporary copy. The real checkout is unchanged.

This command explicitly requires a shared permitted checkout and a 16,384-byte transport contract; its retained run used 6090 bytes including the pin. It is not the 3200-byte model experiment. It makes no model calls and does not establish natural-task LLM benefit or independent execution attestation. It offers a concrete use of source-bound evidence rather than a new performance claim.

9. Reproduce and inspect

- [Adapter and use](#) - [Registered protocol](#) - [Materialized cases and oracle](#) - [Every episode](#) - [Derived results](#) - [Raw final replies and native metadata](#) - [Actual transferred and restored artifacts](#) - [Offline verifier](#)

```
python3 -m unittest discover -s research/handoff_repair -q
python3 -O research/verify_handoff_repair.py
python3 -m unittest discover -s research -p test_handoff_repair_integrity.py -q
```

These commands use no model or network. The verifier reconstructs actual sender notes, prefix delivery, exact wire and decoded history, receiver prompts, native completion/usage consistency, every score and each written proposed-plan file. Adversarial checks reject changed delivery, a changed score, a coherently rehashed prompt, a terminal flip and inconsistent native usage, including under Python optimization. Internal consistency is not independent authentication.