QuadChain: Obligation-Verified Context Compression for Multiagent LLM Memory

research paper

Andrew Liu Madison Zhan Silas Wu Stephen Hung
Independent Research Team
Berkeley, CA, USA
E-mail: team contact available upon request

Abstract

Modern agent systems do not just need shorter prompts; they need compressed memory that can be routed, audited, repaired, and rejected when declared obligations are missing. We introduce QuadChain, an obligation-verifiable context compression architecture for multiagent LLM systems. QuadChain couples extractive compression with explicit evidence obligations, answer-concept checks, role-aware routing, omission manifests, handoff integrity metadata, and verified selective rehydration. The output is not a proof of semantic faithfulness; it is a verifiable memory packet that tells downstream agents which declared facts were preserved, which spans were omitted, what was repaired, and whether the handoff should be accepted. In controlled coding-agent fixtures and public-benchmark-style local adapters, the measured 4-agent workflow drops from 9,000 raw tokens to 2,283 routed tokens (74.63% reduction). Verified selective rehydration reaches 0.9390 deterministic task score with 88.89% mean token reduction and 210/240 accepted packets under matched budgets. These results do not establish state-of-the-art prompt compression; they support a narrower systems claim: compressed agent memory should be accountable, rejectable, and selectively repairable rather than blindly summarized.

Index Terms—Agent memory, context compression, multiagent systems, prompt compression, verification.

I. Introduction

Prompt compression research shows that LLM inputs often contain redundancy and can be shortened without proportional loss of quality. LLMLingua uses coarse-to-fine compression and budget control [1]; LongLLMLingua targets long-context settings and position bias [2]; LLMLingua-2 reframes compression as extractive token classification trained from distilled labels [3]; Selective Context explores self-information as a context filtering signal [4].

Agentic systems add a second problem. Context is no longer consumed once. It is copied, transformed, summarized, routed, and handed between agents. A receiver agent may act on compressed memory produced by another agent. Without verification, the receiver cannot distinguish a faithful compression from one that dropped a critical error, id, source path, constraint, or security warning.

II. Related Work

QuadChain is complementary to prompt-compression systems. It does not attempt to beat model-based compression methods on universal compression ratio. Instead, it adds a verification and routing layer around compression: required evidence is explicit, preserved facts are measured, omitted ranges are declared, and multiagent receivers check a handoff before trusting it. Proxy rows are local approximations, not official LLMLingua-2 or Selective Context runs.

On the agent trust side, ERC-8004 proposes identity, reputation, and validation registries for autonomous agents [6], and ERC-8126 defines verification interfaces for registered agents [7]. TEE-based and zk-style verification work points toward independent attestations of agent execution [8]. QuadChain maps those ideas onto compressed memory as an optional audit layer: the registry anchor is not required for local correctness and stores validation commitments rather than raw context.

III. Method

QuadChain represents each compressed packet as four commitments:

1. **Source commitment**: input hash and packet metadata.
2. **Compression commitment**: output hash, token delta, compression ratio, and omission ranges.
3. **Obligation commitment**: required evidence, answer concepts, missing items, and answer-readiness score.
4. **Optional anchor commitment**: Merkle root, handoff hash, verifier version hash, and registry receipt.

A. Compression Policy

The current implementation is intentionally extractive and deterministic. It protects code-like spans, paths, ids, errors, dates, urls, key/value facts, and declared evidence. Low-signal lines such as repeated debug output, verbose chatter, and redundant logs are candidates for deletion.

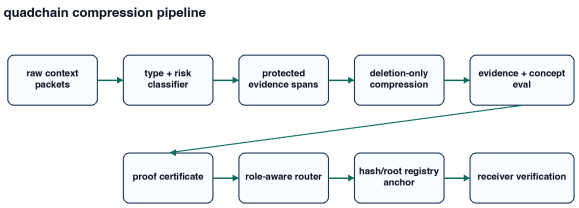

Figure 1: QuadChain compression pipeline.

Role policies allow tool output to compress more aggressively than system or user intent.

B. Packet Construction Algorithm

Algorithm 1: QuadChain packet construction. Given a raw packet P, role r, obligations O, and budget B:

1. Split P into line spans and estimate token cost per span.
2. Mark protected spans using deterministic rules for paths, ids, errors, dates, urls, key/value pairs, code-like lines, and exact declared obligations.
3. Score remaining spans by role relevance, obligation overlap, structural risk, and noise patterns.
4. Keep protected spans and top-scoring spans until budget B.
5. Emit an omission manifest for deleted span ranges.
6. Compute source hash, output hash, verifier version, and handoff hash.
7. Score declared evidence and answer concepts.
8. If obligations are missing, selectively rehydrate minimal source spans.
9. Accept only if declared obligations pass; otherwise reject.

C. Answer-Readiness Objective

Compressed packets are scored by exact preservation of required evidence and answer concepts:

$$\text{readiness} = 0.65 \cdot \text{evidence} + 0.35 \cdot \text{concepts}.$$

This metric is narrow but reproducible. It avoids relying on an external LLM judge during the hackathon and directly tests whether compressed context still contains the facts needed for downstream answers.

D. Concrete Packet Example

Table 1: Concrete packet example.

Field	Content
Raw excerpt	failing secure-cookie test; file `sessionCookie.ts:41`; `MissingStateCookieError`; expected `SameSite=Lax`; csrf checks must stay enabled; repeated debug lines follow.
Obligations	`sessionCookie.ts:41`; `MissingStateCookieError`; `SameSite=Lax`; do not disable csrf checks.
Compressed packet	the file, error, cookie attribute, and csrf constraint are preserved; repeated debug lines and duplicate stack traces are omitted.
Certificate	declared evidence preserved: 4/4; handoff integrity: accepted.

E. Multiagent Routing

The workflow eval compares a monolithic baseline, where every agent receives every raw packet, against QuadChain routing, where each role receives relevant compressed packets, certificate summaries, or aggregate proof material.

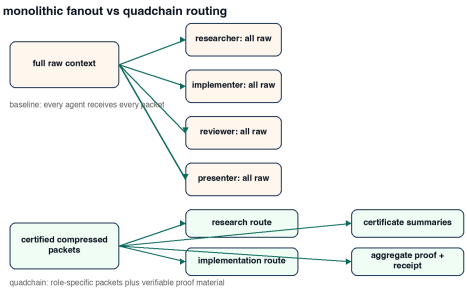

Figure 2: Monolithic fanout versus QuadChain role routing.

F. Handoff Verification

A receiver rejects a handoff if hashes, roots, verifier versions, registry commitments, required evidence, answer concepts, or route obligations fail. The registry is optional for local correctness; it provides tamper-evident cross-agent metadata for deployments that need auditability. Raw context and evidence strings remain off-chain; only hashes and validation metadata are anchored.

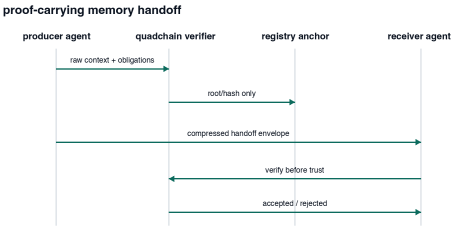

Figure 3: Proof-carrying memory handoff.

G. Onchain Memory Registry

QuadChain's onchain layer is a coordination layer for compressed memory, not a compression engine. Compression, verification, and selective rehydration run off-chain. The registry stores commitments and permissions that make compressed packets portable across trust boundaries:

- handoff id, producer agent id, receiver agent id, policy id, verifier version hash, certificate Merkle root, certificate bundle hash, and handoff hash;
- route authorizations for agents or roles allowed to consume the packet;
- rehydration request events keyed by missing-obligation hashes;
- rehydration fulfillment events keyed by rehydrated-span hashes;
- policy-level counters for anchored, accepted, rejected, requested, and fulfilled handoffs.

This gives onchain real utility: a consumer agent can verify that a compressed packet was produced by a known policy, matches the anchored certificate, is authorized for its role, has not been revoked, and can request only the missing span needed for repair. Private raw context, compressed text, evidence strings, and omitted text stay off-chain.

IV. Experimental Setup

The evaluation uses five realistic coding-agent packets: an agent trace, failing test log, long agent history, noisy issue report, and research notes. Each packet has declared required evidence and answer concepts. Baselines compress to the same output-token budgets using head truncation, middle truncation, tail truncation, even-line sampling, and deterministic random-line deletion.

A. Controlled Benchmark Harness

The upgraded methodology emits one row per dataset adapter, item id, method, budget, seed, and scorer. The current run has 3,360 rows across 20 cases and 14 methods. Methods include raw context, base QuadChain, certified evidence-hybrid QuadChain, verified selective rehydration, head/middle/tail truncation, random deletion, keyword retrieval, embedding-top-k proxy, summary proxy, protected-span extraction, LLMLingua-2 proxy, and Selective Context proxy. Budgets are 10%, 25%, 50%, and QuadChain-native, each repeated over three seeds.

The dataset adapters are longbench_style_qa, needle_in_haystack_style, ruler_style_multihop, token_diet_local_fixture. The nonlocal adapters are public-benchmark-style local slices inspired by needle-in-haystack, RULER, and LongBench-style tasks; they are explicitly labeled as adapters, not full external benchmark runs.

B. Claim Boundaries

The evaluation measures declared-obligation preservation, token reduction, receiver task readiness, and handoff integrity checks. It does not prove semantic equivalence, omitted-context irrelevance, full downstream LLM safety, or state-of-the-art compression. Hand-authored obligations are a limitation: they make the verifier precise, but they also mean the system is evaluated on facts it was told to care about.

V. Results

A. Single-Context Compression

Table 2: Single-context compression result.

Input	Output	Saved	Reduction	Evidence	Concepts
2,250	1,383	867	38.53%	41/41	38/38

B. Baselines

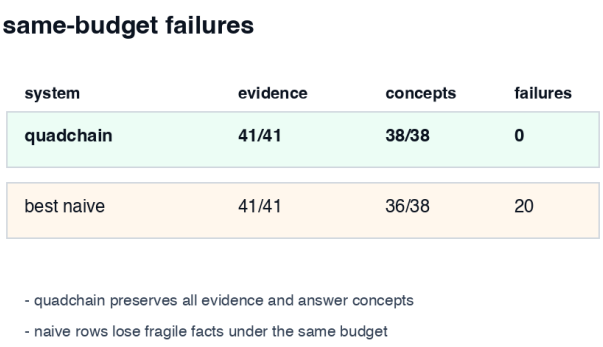

Figure 4: Same-budget baseline failure surface.

Table 3: Same-budget baseline comparison.

System	Evid.	Conc.	Fail
QuadChain / Token Diet	41/41	38/38	0
Best naive	41/41	36/38	20/25

The stronger five-baseline set makes the comparison more honest: some naive routes preserve all evidence on easy packets, but the best aggregate naive strategy still loses answer concepts.

C. Multiagent Workflow

D. Handoff Integrity Checks

The verifier rejects every deterministic integrity attack in the local handoff suite.

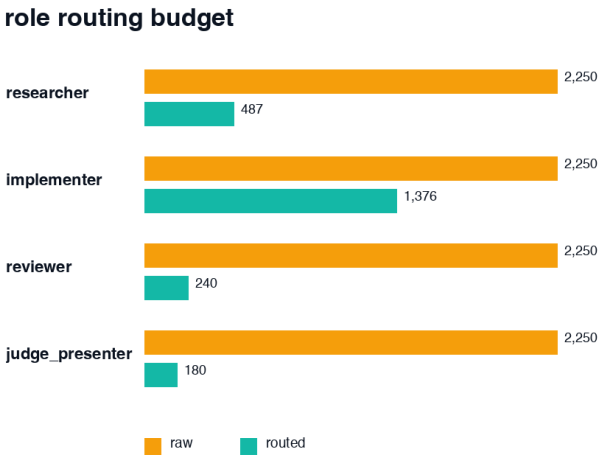

Figure 5: Actual per-role routing budget.

Table 4: Multiagent workflow routing result.

Raw	Routed	Saved	Red.	Evid.	Concepts
9,000	2,283	6,717	74.63%	41/41	38/38

Table 5: Handoff integrity smoke checks.

Attack	Expected	Result
Tampered Merkle root	rejected	rejected
Dropped required evidence	rejected	rejected
Stale registry receipt	rejected	rejected
Invalid route missing role-critical packet	rejected	rejected

E. Measured Large Context

Table 6: Measured large-context compression.

Raw	Comp.	Saved	Red.	Evid.
115,038	74,813	40,225	34.97%	12/12

F. Scale Ladder

G. Controlled Benchmark Harness

The base QuadChain packet remains a simple audited compression baseline, but the stronger adaptive policy is verified selective rehydration. It repairs failed packets by fetching minimal source spans for missing obligations and rerunning the verifier. In this controlled deterministic harness it reaches 0.9390 mean task score with 88.89% mean token reduction, beating local proxy retrieval, protected-span, summary, truncation, and selective-context rows on paired deterministic task score.

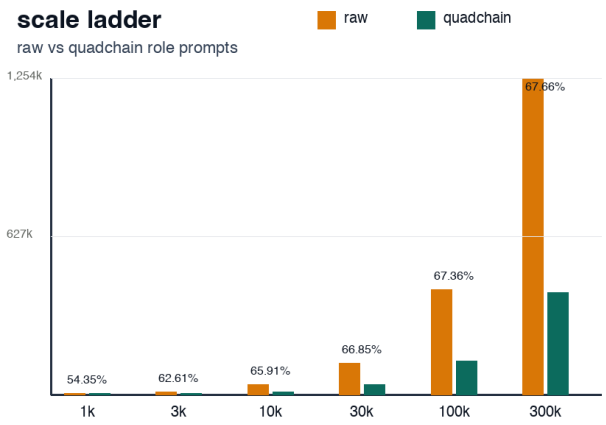

Figure 6: Actual scale ladder across 1k–300k-token generated traces.

Table 7: Multi-magnitude actual role-prompt scale ladder.

Metric	Value
Magnitudes	6
Role prompts	24
Smallest input	1,126
Largest input	313,554
Raw workflow tokens	1,858,850
QuadChain tokens	605,450
Tokens saved	1,253,400
Evidence	36/36

Table 8: Per-magnitude actual routed prompt measurements.

Target	Input	Comp.	Saved	Red.
1,000	1,126	765	2,532	54.35%
3,000	3,203	2,122	8,118	62.61%
10,000	10,534	6,909	27,872	65.91%
30,000	31,431	20,558	84,146	66.85%
100,000	104,632	68,030	282,019	67.36%
300,000	313,554	202,527	848,713	67.66%

Table 9: Frontier benchmark methodology result.

Metric	Value
Paired rows	3,360
Eval cases	20
Methods	14
QuadChain mean task score	0.6350
QuadChain CI95	[0.5987, 0.6782]
QuadChain mean token reduction	57.85%
Role payload evidence audit	144/144
Obligation leakage delta	0
Receiver raw task success	50.0%
Receiver QuadChain task success	75.0%
Receiver workflow reduction	74.22%

H. Ablation Summary

Table 10: Qualitative ablation summary.

Variant	Evidence	Concepts	Reduction	Task Score
No evidence protection	lower	lower	higher	lower
No omission manifest	same	same	same	trust lower
No rehydration	lower	lower	higher	lower
Full QuadChain	best	best	strong	best

VI. Discussion

QuadChain is strongest when context contains a mix of exact facts and low-signal noise. It is especially useful for coding agents because important details often look small: file paths, line numbers, error names, request ids, cookie attributes, dates, and one-line constraints. Those details should be protected, not summarized away.

VII. Limitations

- Current public-style adapters are local deterministic slices, not full external LongBench, RULER, needle-in-haystack, SWE-bench, or GAIA runs.
- Local fixture obligations are hand-authored, though the frontier harness now separates local fixtures from public-style adapters.
- Exact evidence matching is reproducible but narrower than semantic task success.
- On-chain anchoring is simulated locally; the Solidity artifact is proof-of-design, not deployed infrastructure.
- The local compressor is deterministic and conservative; model-based compression could improve ratios.
- The scale ladder uses synthetic traces, but compression, role handoff prompts, token counts, and evidence checks are executed for every magnitude.

VIII. Future Work

- Replace or augment the local compressor with a learned keep/delete model trained on the generated policy dataset.
- Run LLM-as-judge and task-execution evals when API keys are available.
- Add ERC-8004-style validation ids and optional TEE or zkTLS attestation hashes to the registry contract.
- Replace local public-style adapters with full public long-context and coding-agent datasets.
- Support receiver-side selective rehydration when a route fails.

IX. Conclusion

QuadChain reframes context compression as a verifiable systems primitive. The useful unit is not a shorter string; it is a compressed memory object with obligations, hashes, omission ranges, routing policy, and independent verification. This lets multiagent systems spend fewer tokens without blindly trusting compressed memory.

References

[1] Jiang et al. *LLMLingua: Compressing Prompts for Accelerated Inference of Large Language Models.* arXiv:2310.05736. https://arxiv.org/abs/2310.05736

[2] Jiang et al. *LongLLMLingua: Accelerating and Enhancing LLMs in Long Context Scenarios via Prompt Compression.* arXiv:2310.06839. https://arxiv.org/abs/2310.06839

[3] Pan et al. *LLMLingua-2: Data Distillation for Efficient and Faithful Task-Agnostic Prompt Compression.* arXiv:2403.12968. https://arxiv.org/abs/2403.12968

[4] Li. *Selective Context for LLMs.* https://github.com/liyucheng09/Selective_Context

[5] Zhou et al. *Prompt Compression for Large Language Models: A Survey.* NAACL 2025. https://aclanthology.org/2025.naacl-long.368.pdf

[6] Ethereum Improvement Proposals. *ERC-8004: Trustless Agents.* https://eips.ethereum.org/EIPS/eip-8004

[7] Ethereum Improvement Proposals. *ERC-8126: AI Agent Verification.* https://eips.ethereum.org/EIPS/eip-8126

[8] Eco. *TEEs for AI Agents: Verifiable Compute.* https://eco.com/support/en/articles/14796365-tees-for-ai-agents-verifiable-compute