Northeastern University

Transactive Cognitive Memory for Multi-Agent AI and Distributed Systems

Sheetal Naik

Master of Science in Data Analytics Engineering

Research Report

Principal Adviser: Dr. Mohammad Dehghani

Boston, Massachusetts
August 28, 2025

Abstract

Transactive memory is a socio-cognitive mechanism wherein members of a group remember "who knows what" and coordinate accordingly. This report operationalizes that idea for multi-agent AI and evaluates a *Transactive Cognitive Memory* (TCM) framework with three specialized roles (Planner, Researcher, Verifier) and four memory backends (Isolated, Shared, Selective, and TCM). A Beta–Bernoulli trust model with Thompson Sampling performs probability-matching delegation among agents.

Across three iterations (five tasks each), success is the fraction of tasks with a "SUPPORTED" verifier verdict. Iteration 1 shows an early advantage for Isolated memory (40%). Iteration 2 exhibits uniform failure (likely stricter verification and/or noisier retrieval; hypothesis pending rubric/version diffs). Iteration 3 attains 100% across all backends. Critically, in Iteration 3 the TCM backend achieved a **Delegation Rate of 66.7%** (others 0%), demonstrating superior specialization efficiency at equal accuracy and equal memory reads. Related work motivates minimally disruptive memory, category-bounded storage, and privacy-preserving evaluation (Zulfikar et al. 2024; Kirmayr et al. 2025; PIN AI Team et al. 2025).

1 Introduction

Long-horizon assistance with large language models (LLMs) requires memory that is useful to the task at hand and governed by clear, auditable policies. Prior research shows that memory-augmented assistants can minimize disruption while providing timely recall (Zulfikar et al. 2024). Practice emphasizes structured, bounded memories to preserve transparency and privacy (Kirmayr et al. 2025; PIN AI Team et al. 2025).

This report studies a **Transactive Cognitive Memory (TCM)** framework that emulates the human phenomenon of "who knows what" within a team. Specialized agents access distinct memory scopes; a probabilistic trust model learns which agent is most competent for a given query. Delegation uses Thompson Sampling over Beta posteriors, enabling rapid adaptation with sparse feedback.

2 Related Work

Minimal-disruption interfaces. Memoro demonstrates a wearable memory assistant with two interaction modes—*Query* and *Queryless*—and reports reduced disruption through contextual suggestions (Zulfikar et al. 2024). A compliant excerpt:

> *"Memoro uses a large language model (LLM) to infer the user's memory needs in a*

conversational context"

(p. 1).

Category-bounded memories. CarMem proposes long-term memory restricted to predefined categories to improve transparency and control (Kirmayr et al. 2025). Short quote:

"a long-term memory system for voice assistants, structured around predefined categories"

.

Privacy-preserving evaluation. The GOD model advocates on-device training and assessment within TEEs and notes it

"safeguards user data while applying reinforcement and imitation learning"

(PIN AI Team et al. 2025).

Delegation via probability matching. Thompson Sampling balances exploration and exploitation in sequential decisions (Russo et al. 2018). The tutorial states,

"Thompson sampling is an algorithm for online decision problems"

. The original 1933 paper argues there is

"no objection to the use of data, however meagre, as a guide to action"

(p. 285) (Thompson 1933). An industry overview underscores practical demand for persistent assistant memory (RoX818 2025).

3 System Design and Rationale

3.1 Roles and Backends

Planner decomposes a query into steps and selects a strategy. **Researcher** retrieves or generates evidence, consulting memory as needed. **Verifier** evaluates whether the answer is supported and emits a verdict in {SUPPORTED, UNCERTAIN}.

Four backends are compared:

- **Isolated:** per-role stores; no cross-role reads (high precision, limited reuse).
- **Shared:** a single global store (high reuse, risk of noise).
- **Selective:** global store with role-/task-aware filters (source, recency, topic).
- **TCM:** selective reads with trust-weighted routing via Thompson Sampling.

3.2 Why Thompson Sampling + Beta

Each agent i receives binary feedback: success if SUPPORTED, failure otherwise. Let (α_i, β_i) parameterize a Beta prior over competence $\theta_i \in [0, 1]$. After an event with outcome $y \in \{0, 1\}$:

$$\alpha_i \leftarrow \alpha_i + y, \qquad \beta_i \leftarrow \beta_i + (1 - y).$$

The posterior remains $\text{Beta}(\alpha_i, \beta_i)$; a simple trust score is $t_i = \frac{\alpha_i}{\alpha_i+\beta_i}$. Thompson Sampling draws $\tilde{\theta}_i \sim \text{Beta}(\alpha_i, \beta_i)$ for each agent and delegates to $\arg\max_i \tilde{\theta}_i$ (Russo et al. 2018; Thompson 1933). Conjugacy yields constant-time updates and fast learning from few observations.

3.3 Why the Distributed Use Case

Distributed settings combine *cloud* researchers (broad retrieval, heavy synthesis), *edge* planners (low-latency constraints), and *human* verifiers (safety-critical judgments). Specialization reduces bandwidth, improves fault isolation, and aligns with privacy-preserving evaluation (PIN AI Team et al. 2025).

3.4 Delegation Algorithm

Algorithm 1: TCM Delegation with Beta–Bernoulli Trust

Input: Query q, agents $\mathcal{A}$ with (α_i, β_i), backend B

for *each agent* $i \in \mathcal{A}$ **do**
  Sample $\tilde{\theta}_i \sim \text{Beta}(\alpha_i, \beta_i)$

$i^\star \leftarrow \arg\max_i \tilde{\theta}_i$

Planner produces plan π; Researcher executes π using B; Verifier returns $v \in \{\text{SUPPORTED}, \text{UNCERTAIN}\}$

if $v =$ *SUPPORTED* **then**
  $\alpha_{i^\star} \leftarrow \alpha_{i^\star} + 1$

else
  $\beta_{i^\star} \leftarrow \beta_{i^\star} + 1$

Output: Answer, citations, verdict, updated trust

4 Implementation

A core framework provides role templates, Beta trust utilities, and a lightweight UI (Python). A distributed "TCM Lab" instantiation offers Planner/Researcher/Verifier roles, the four backends, and a dashboard for evaluation. Public repositories exist for artifact review; links are omitted in this manuscript per submission requirements.

Sketch of trust update (Python):

```python
from dataclasses import dataclass
import numpy as np

@dataclass
class BetaTrust:
    alpha: float = 1.0
    beta: float  = 1.0
    def sample(self):
        return np.random.beta(self.alpha, self.beta)
    def update(self, success: bool):
        if success: self.alpha += 1
        else:       self.beta  += 1
```

5 Experimental Setup

Five tasks per iteration: (1) design a recommendation system, (2) research transformer architectures, (3) verify "GPT uses attention mechanisms", (4) plan a chatbot implementation, (5) analyze computer vision techniques. For each backend, the Verifier emits a verdict; **Success Rate** is:

$$\text{SuccessRate} = \frac{\#\,\text{SUPPORTED}}{5} \times 100\%.$$

Time, total memory entries, and memories accessed are logged.

6 Evaluation Objectives & Metrics

The primary objective of TCM is **learned specialization via delegation**. Accordingly:

- **Primary KPI: Delegation Rate** = delegated interactions / total interactions in the iteration log.[1]
- **Secondary: Memory Reads per Supported Verdict (MRSV)** = (memories accessed) / #SUPPORTED.
- **Tertiary: Latency** = wall-clock time per backend run.

[1] Computed from the Iteration 3 logs provided for this study.

When all backends reach 100% success, Delegation Rate distinguishes systems by specialization efficiency; MRSV and time are reported for completeness.

7 Results

Table 1. Iteration 1: Backend comparison (5 tasks).

Backend	Success	Time	Entries	Notes
Isolated	40.00%	0.00s	15	Best success
Shared	0.00%	0.00s	15	
Selective	0.00%	0.00s	15	
TCM	0.00%	0.00s	15	

Table 2. Iteration 2: Backend comparison (5 tasks).

Backend	Success	Time	Memories	Entries	Delegation Rate
Isolated	0.0%	7.2s	1	12	0%
Shared	0.0%	7.9s	2	12	0%
Selective	0.0%	8.8s	1	12	0%
TCM	0.0%	8.3s	1	12	0%

Table 3. Iteration 3: Backend comparison (5 tasks).

Backend	Success	Time	Memories	Entries	Delegation Rate	MRSV
Isolated	100.0%	6.8s	1	12	0%	1/5
Shared	100.0%	5.8s	1	12	0%	1/5
Selective	100.0%	7.0s	2	12	0%	2/5
TCM	100.0%	6.3s	1	12	**66.7%**	1/5

8 Discussion

Early performance. With cold-start priors and immature memory hygiene, a global store can amplify irrelevant retrievals; isolation raises precision, matching table 1.

Uniform failure in Iteration 2. All backends scored 0% in table 2. A likely cause is stricter verification and/or noisier retrieval; confirming this would require rubric/version diffs (treated here as a hypothesis).

Why TCM is the best case in Iteration 3. With equal accuracy (100%) and equal memory reads (1) across three backends, the distinguishing goal of TCM—*learned specialization*—is captured by Delegation Rate. TCM achieved **66.7%** delegation while other backends delegated 0% (table 3). This indicates that TCM not only solved the tasks but also *routed work to the most competent agent* in two-thirds of interactions.[2] Although Shared recorded a slightly lower latency (5.8s vs. 6.3s), the primary objective here is specialization via delegation; under that objective, **TCM is the best case**.

Implications. Delegation enables scalable teamwork: as task diversity grows, probability-matching routing should increasingly exploit specialized competence while continuing to explore uncertain options.

9 Threats to Validity

Small task sets inflate variance. Inconsistent verifier criteria and memory initialization reduce comparability across iterations. Fixing seeds, prompts, scoring rubrics, and expanding the stratified task suite would improve power and stability.

10 Ethical Considerations

Memory systems carry privacy risks; retention should be transparent and user-controlled. For human-in-the-loop verification, disclosure and consent are required. Category-bounded or queryless modes may reduce over-collection but must remain auditable (Kirmayr et al. 2025; Zulfikar et al. 2024; PIN AI Team et al. 2025).

11 Conclusion

A transactive cognitive memory framework has been presented, unifying agent specialization with Bayesian trust and Thompson Sampling. Empirical comparisons illustrate the dynamics of isolation vs. sharing vs. selective routing. In Iteration 3, **TCM achieved the highest Delegation Rate (66.7%)** at equal accuracy and equal memory reads, making it the *best case* for specialization efficiency. Future work includes contextual Thompson Sampling, multi-hop delegation, and richer evaluator labels.

[2]Delegation Rate computed as delegated interactions / total interactions in Iteration 3 logs.

Provenance Notes

Key claims in §§Related Work and Methods are supported by (Zulfikar et al. 2024; Kirmayr et al. 2025; PIN AI Team et al. 2025; Russo et al. 2018; Thompson 1933). Quotes were kept ≤25 words.

References

Kirmayr, J., L. Stappen, P. Schneider, F. Matthes, and E. André (2025). "CarMem: Enhancing Long-Term Memory in LLM Voice Assistants through Category-Bounding". In: *arXiv preprint arXiv:2501.09645*. URL: https://arxiv.org/abs/2501.09645.

PIN AI Team, B. Sun, G. Guo, R. Peng, B. Zhang, S. Wang, L. Florescu, X. Wang, D. Crapis, and B. Wu (2025). "GOD model: Privacy Preserved AI School for Personal Assistant". In: *arXiv preprint arXiv:2502.18527*. URL: https://arxiv.org/abs/2502.18527.

RoX818 (June 2025). *Memory-Augmented AI Assistants Actually Remember You*. Accessed 2025-08. URL: https://aicompetence.org/memory-augmented-ai-assistants-remember-you/.

Russo, D., B. Van Roy, A. Kazerouni, I. Osband, and Z. Wen (2018). "A Tutorial on Thompson Sampling". In: *Foundations and Trends in Machine Learning* 11.1, pp. 1–96. DOI: 10.1561/2200000070. URL: https://arxiv.org/abs/1707.02038.

Thompson, W. R. (1933). "On the Likelihood That One Unknown Probability Exceeds Another in View of the Evidence of Two Samples". In: *Biometrika* 25, pp. 285–294. URL: https://gwern.net/doc/statistics/decision/1933-thompson.pdf.

Zulfikar, W., S. Chan, and P. Maes (2024). "Memoro: Using Large Language Models to Realize a Concise Interface for Real-Time Memory Augmentation". In: *CHI '24: Proceedings of the CHI Conference on Human Factors in Computing Systems*. DOI: 10.1145/3613904.3642450. URL: https://arxiv.org/abs/2403.02135.