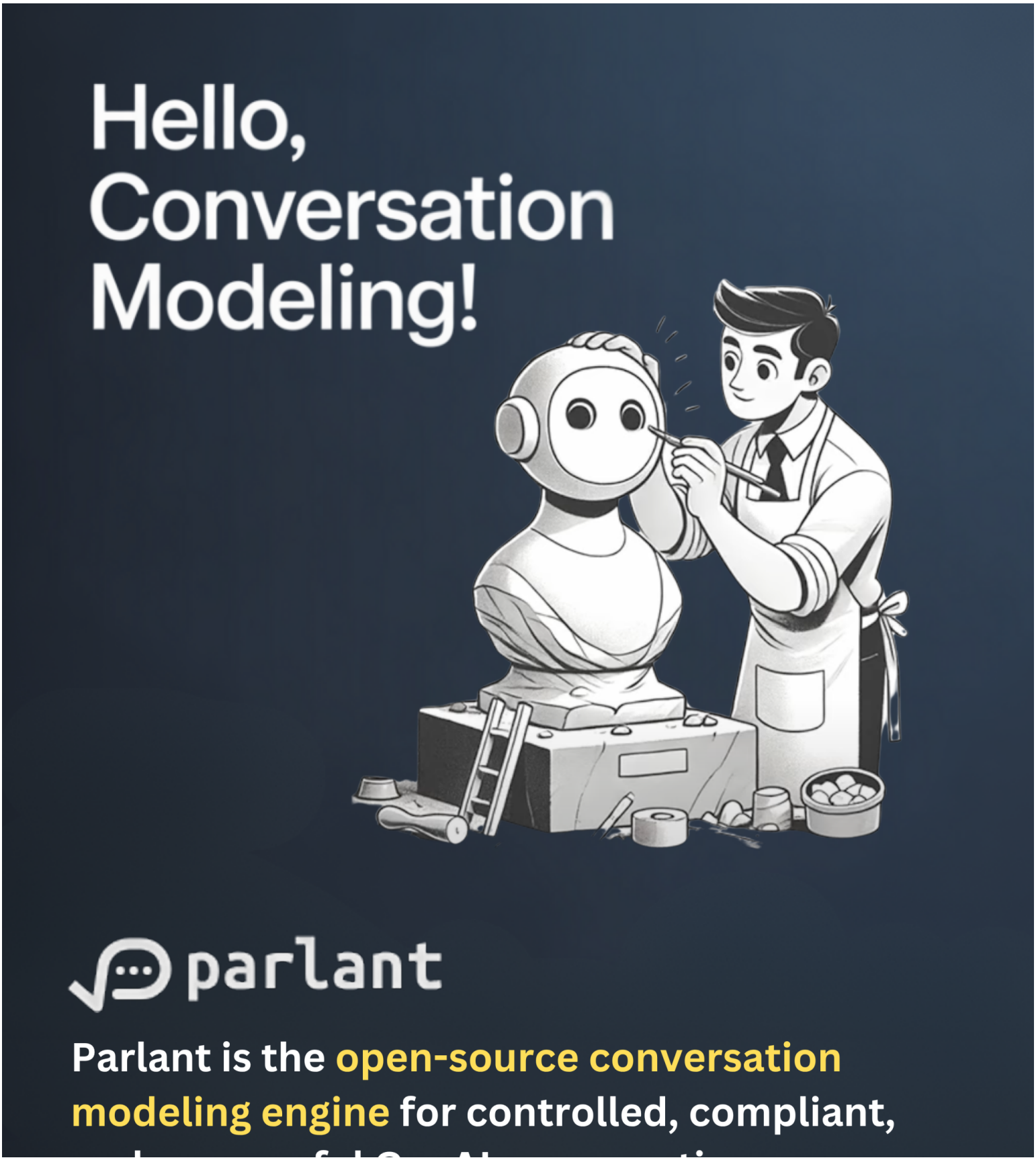

Hello,
Conversation
Modeling!
parlant
Parlant is the open-source conversation
modeling engine for controlled, compliant,

# MemOS: A Memory-Centric Operating System for Evolving and Adaptive Large Language Models

By **Sana Hassan** - June 14, 2025

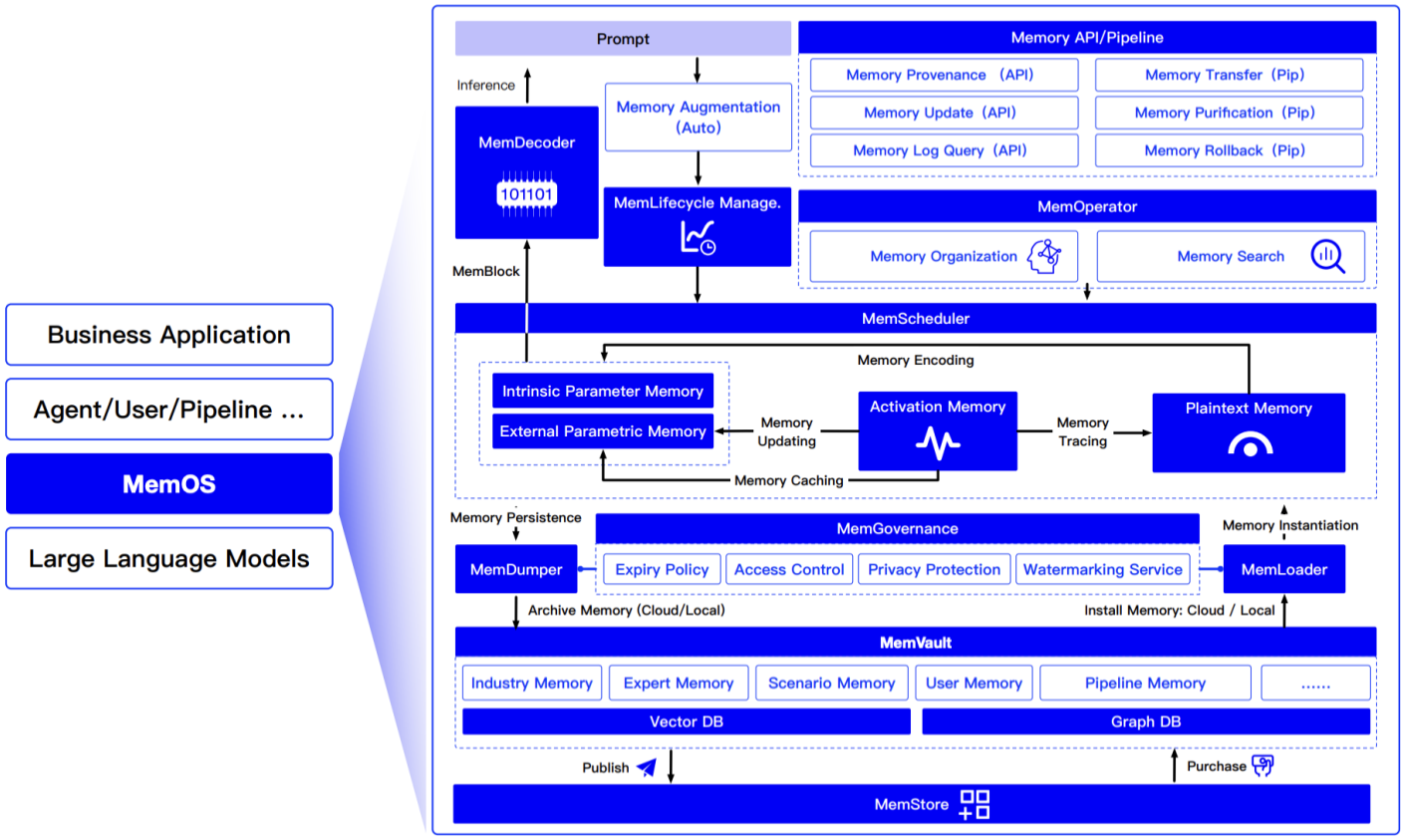



**Figure 5** Overview of the MemOS architecture: showing the end-to-end memory lifecycle from user input to API parsing, scheduling, activation, governance, and evolution—unified via MemCube.

LLMs are increasingly seen as key to achieving Artificial General Intelligence (AGI), but they face major limitations in how they handle memory. Most LLMs rely on fixed knowledge stored in their weights and short-lived context during use, making it hard to retain or update information over

time. Techniques like RAG attempt to incorporate external knowledge but lack structured memory management. This leads to problems such as forgetting past conversations, poor adaptability, and isolated memory across platforms. Fundamentally, today's LLMs don't treat memory as a manageable, persistent, or sharable system, limiting their real-world usefulness.

To address the limitations of memory in current LLMs, researchers from MemTensor (Shanghai) Technology Co., Ltd., Shanghai Jiao Tong University, Renmin University of China, and the Research Institute of China Telecom have developed MemO. This memory operating system makes memory a first-class resource in language models. At its core is MemCube, a unified memory abstraction that manages parametric, activation, and plaintext memory. MemOS enables structured, traceable, and cross-task memory handling, allowing models to adapt continuously, internalize user preferences, and maintain behavioral consistency. This shift transforms LLMs from passive generators into evolving systems capable of long-term learning and cross-platform coordination.


🎤 Talks • Demos • Networking • Certificate

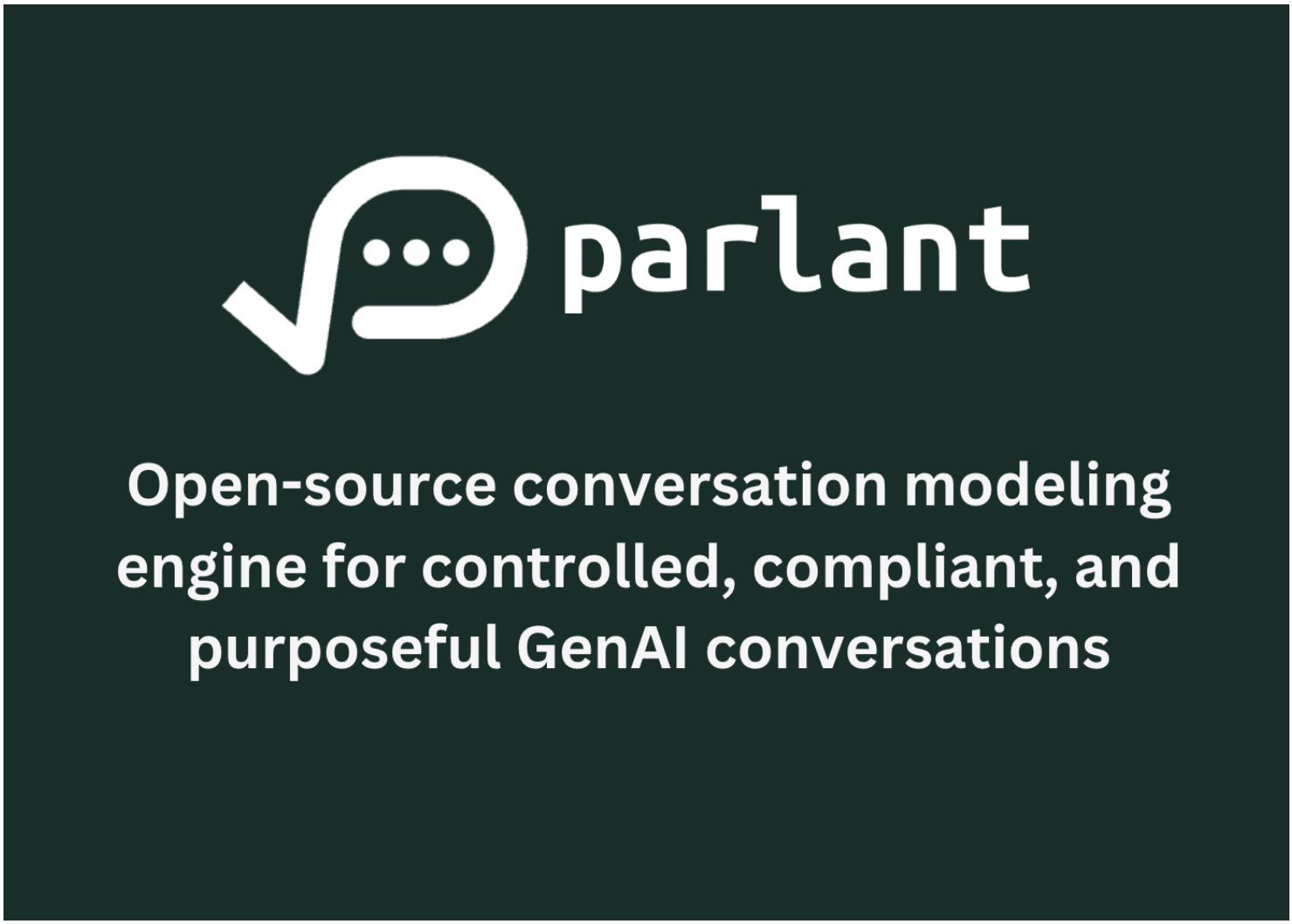
FREE REGISTRATION

---

As AI systems grow more complex—handling multiple tasks, roles, and data types—language models must evolve beyond understanding text to also retaining memory and learning continuously. Current LLMs lack structured memory management, which limits their ability to adapt and grow over time. MemOS, a new system that treats memory as a core, schedulable resource. It enables long-term learning through structured storage, version control, and unified

memory access. Unlike traditional training, MemOS supports a continuous “memory training” paradigm that blurs the line between learning and inference. It also emphasizes governance, ensuring traceability, access control, and safe use in evolving AI systems.

MemOS is a memory-centric operating system for language models that treats memory not just as stored data but as an active, evolving component of the model’s cognition. It organizes memory into three distinct types: Parametric Memory (knowledge baked into model weights via pretraining or fine-tuning), Activation Memory (temporary internal states, such as KV caches and attention patterns, used during inference), and Plaintext Memory (editable, retrievable external data, like documents or prompts). These memory types interact within a unified framework called the MemoryCube (MemCube), which encapsulates both content and metadata, allowing dynamic scheduling, versioning, access control, and transformation across types. This structured system enables LLMs to adapt, recall relevant information, and efficiently evolve their capabilities, transforming them into more than just static generators.




At the core of MemOS is a three-layer architecture: the Interface Layer handles user inputs and parses them into memory-related tasks; the Operation Layer manages the scheduling, organization, and evolution of different types of memory; and the Infrastructure Layer ensures safe storage, access governance, and cross-agent collaboration. All interactions within the system are mediated through MemCubes, allowing traceable, policy-driven memory operations. Through modules like MemScheduler, MemLifecycle, and MemGovernance, MemOS maintains a continuous and adaptive memory loop—from the moment a user sends a prompt, to memory injection during reasoning, to storing useful data for future use. This design not only enhances the model's responsiveness and personalization but also ensures that memory remains structured, secure, and reusable.

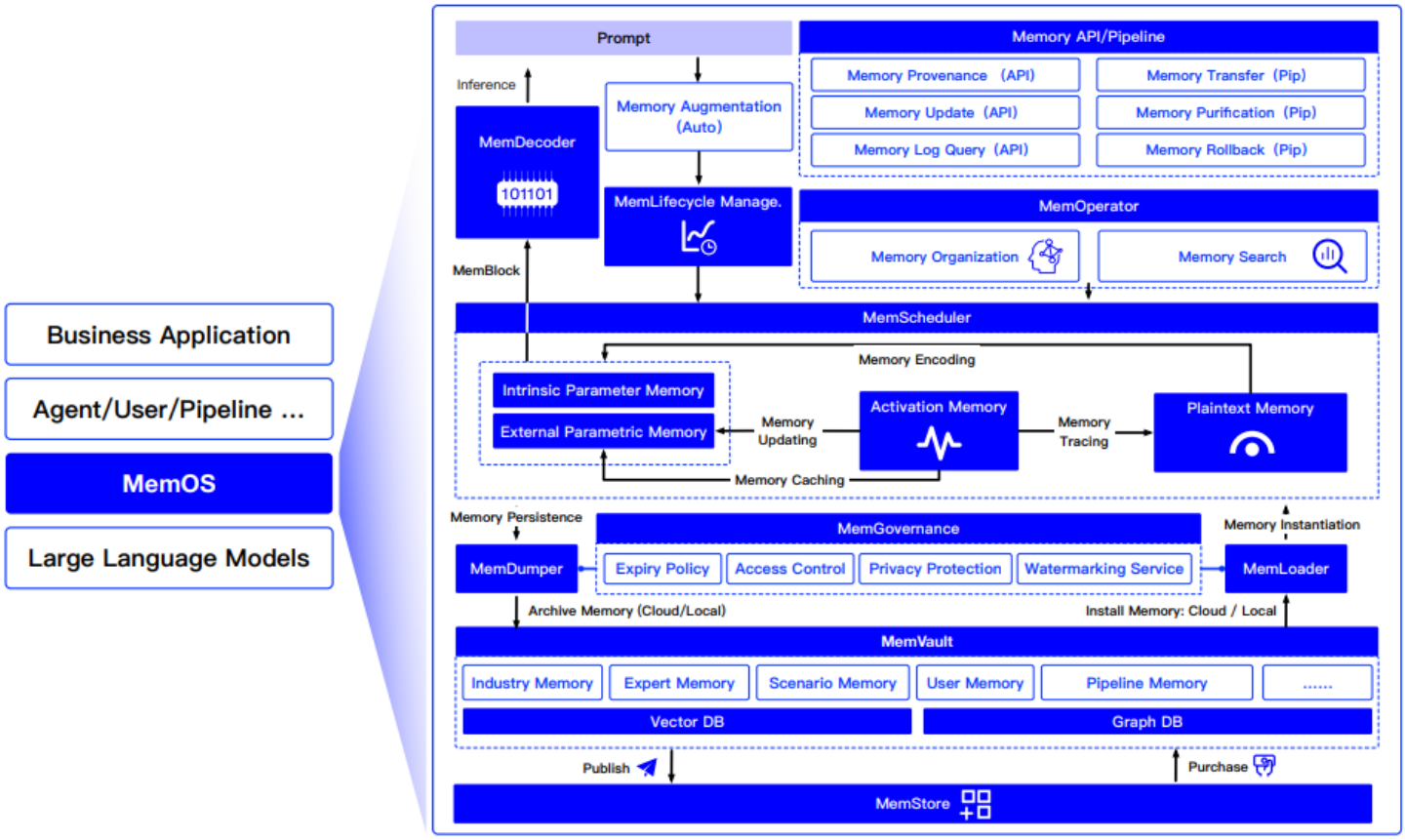



**Figure 5** Overview of the MemOS architecture: showing the end-to-end memory lifecycle from user input to API parsing, scheduling, activation, governance, and evolution—unified via MemCube.

In conclusion, MemOS is a memory operating system designed to make memory a central, manageable component in LLMs. Unlike traditional models that depend mostly on static model weights and short-term runtime states, MemOS introduces a unified framework for handling parametric, activation, and plaintext memory. At its core is MemCube, a standardized memory unit that supports structured storage, lifecycle management, and task-aware memory augmentation. The system enables more coherent reasoning, adaptability, and cross-agent collaboration. Future goals include enabling memory sharing across models, self-evolving memory blocks, and building a decentralized memory marketplace to support continual learning and intelligent evolution.

---

Check out the **Paper**. All credit for this research goes to the researchers of this project. Also, feel free to follow us on **Twitter** and don't forget to join our **100k+ ML SubReddit** and Subscribe to **our Newsletter**.

## Sana Hassan

+ posts

Sana Hassan, a consulting intern at Marktechpost and dual-degree student at IIT Madras, is passionate about applying technology and AI to address real-world challenges. With a keen interest in solving practical problems, he brings a fresh perspective to the intersection of AI and real-life solutions.

Previous article

**Sakana AI Introduces Text-to-LoRA (T2L): A Hypernetwork that Generates Task-Specific LLM Adapters (LoRAs) based on a Text Description of the Task**

Next article

**Internal Coherence Maximization (ICM): A Label-Free, Unsupervised Training Framework for LLMs**

## RELATED ARTICLES

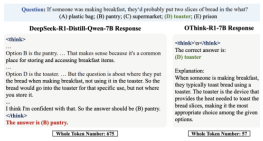


Figure 2: Comparison of a generated responses on CommonsenseQA.

### OThink-R1: A Dual-Mode Reasoning Framework to Cut Redundant Computation in LLMs

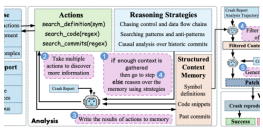
Sana Hassan - June 14, 2025


### Building AI-Powered Applications Using the Plan → Files → Code Workflow...

Sana Hassan - June 14, 2025


### AI-Generated Ad Created with Google's Veo3 Airs During NBA Finals, Slashing...

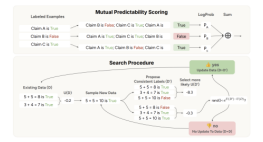
Jean-marc Mommessin - June 14, 2025



*Code Researcher* conducts deep research over code in three phases: (1) Stu
se and crash report as input, the Analysis phase performs multi-step reason
patterns, and commit history of code. It gathers context in a structured memo
phase filters the contents of the memory to keep relevant context and generat
idation phase uses external tools to validate the patch.

### Microsoft AI Introduces Code Researcher: A Deep Research Agent for Large...

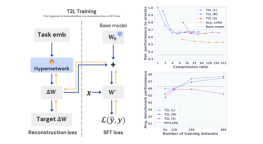
Asif Razzaq - June 14, 2025



gure 2: ICM optimizes labels for logical consistency and mutual predictability. **Top**: an illustrati
ample of mutual predictability scoring. **Bottom**: the searching process for labeling a new exampl

### Internal Coherence Maximization (ICM): A Label-Free, Unsupervised Training Framework for LLMs

Sajjad Ansari - June 14, 2025



re 1: **Left:** Conceptual overview of T2L's training routine. Given a set of task description embeddings, we train a
pernetwork to generate LoRA adaptation matrices (ΔW) for various tasks. The weights of T2L are either optimized
distill pre-trained LoRA weights or via multi-task supervised fine-tuning on downstream tasks. **Right, Top:** Relative
formance to the oracles on training SNI tasks with varying compression ratios. **Right, Bottom:** Zero-shot LoRA
neration performance on 10 benchmark tasks. As we increase the number of pre-training datasets, the performance of
1 increases for 3 different T2L architectures.

### Sakana AI Introduces Text-to-LoRA (T2L): A Hypernetwork that Generates Task-Specific LLM...

Asif Razzaq - June 13, 2025

Marktechpost AI

Watch on

## ABOUT US

Marktechpost is a California-based AI News Platform providing easy-to-consume, byte size updates in machine learning, deep learning, and data science research.

Contact us: Asif@marktechpost.com

## FOLLOW US

through our affiliates/referrals via product promotion in the articles