

What is LLM memory leakage?

LLM memory leakage is when an AI assistant with long-term memory brings stored details into conversations where they do not belong, such as a health detail surfacing in a work request. A related failure, memory-driven sycophancy, is when stored beliefs push the model to agree with the user instead of answering objectively. PersistBench measured both across 18 models and found median failure rates of 53% for cross-domain leakage and 97% for sycophancy.

HOW IT WORKS

01 How it happens and how PersistBench tests it

Memory is usually added to the model's context automatically, without regard to whether it is relevant. The model then decides what to use, and models are poor at leaving out something they have been shown.

In Inspect, PersistBench runs three tasks: `persistbench_cross_domain` and `persistbench_sycophancy` (200 samples each), plus `persistbench_beneficial_memory` (100 samples), which checks that the model still uses memory well when it should. The two failure tasks generate three answers per sample and keep the worst, because one inappropriate use of memory is enough to cause harm. A judge model scores each answer, and a score of 3 or more out of 5 counts as a failure.

FROM PRIVACY BUG TO ATTACK SURFACE

If an attacker can write to memory, through a poisoned document or a crafted chat, the same mechanism carries their instructions into every future session. That is memory poisoning, which AgentThreatBench tests.

02 What the results tell you

Across 18 frontier and open-source models, the median failure rate was 53% for cross-domain leakage and 97% for sycophancy. In plain terms, most models leaked irrelevant memory about half the time and almost always let stored beliefs shape their answer. For a business, that means personal or confidential context saved in one setting can surface in another, including in front of someone else using a shared account or a shared agent.

SOURCES

- [1] PersistBench: When Should Long-Term Memories Be Forgotten by LLMs?
- [2] PersistBench in Inspect Evals (task docs)
- [3] OWASP Top 10 for LLM Applications (2025)
- [4] OWASP Top 10 for Agentic Applications (2026)

Test your AI application before an attacker does.

securelayer7.net/learn/ai-security/llm-memory-leakage

[Open online](https://securelayer7.net/learn/ai-security/llm-memory-leakage)