# LanceDB Pro

# memory-lancedb-pro

Enhanced LanceDB memory plugin for OpenClaw — community reference guide

## Overview

**memory-lancedb-pro** is a community-developed, production-grade long-term memory plugin for OpenClaw. It replaces the built-in `memory-lancedb` plugin with a significantly more capable retrieval pipeline, designed for agents that need persistent, high-quality memory across sessions without manual tagging or configuration overhead.

The core problem it solves: standard OpenClaw agents have no memory between sessions. Every conversation starts from zero. memory-lancedb-pro automatically captures what matters from each session and retrieves relevant context in future ones.

Primary upstream repo: `CortexReach/memory-lancedb-pro`. Several community forks exist (win4r, McBorisson, fryeggs, kvc0769) with varying additions such as Volcengine multimodal embeddings or unified Claude Code/Claude Desktop support.

> **OpenClaw 2026.3+ compatibility:** The CortexReach fork has been updated to use `before_prompt_build` hooks, replacing the deprecated `before_agent_start` hook. If you are on 2026.3.24 or later, use this fork. Run `openclaw doctor --fix` after upgrading.

---

## Feature Comparison

| Feature | Built-in memory-lancedb | memory-lancedb-pro |
|---|---|---|
| Vector search | ✓ | ✓ |
| BM25 full-text search | ✗ | ✓ |
| Hybrid fusion (Vector + BM25) | ✗ | ✓ configurable weights |

| Feature | Built-in memory-lancedb | memory-lancedb-pro |
|---|---|---|
| Cross-encoder reranking | ✗ | ✓ Jina, SiliconFlow, Pinecone, etc. |
| Recency / time decay scoring | ✗ | ✓ |
| MMR diversity filtering | ✗ | ✓ |
| Multi-scope isolation | ✗ | ✓ global / agent / project / user |
| Smart LLM extraction | ✗ | ✓ optional, uses any OpenAI-compatible LLM |
| Management CLI | ✗ | ✓ list / search / stats / delete / export / import |
| Auto-capture on session end | ✓ basic | ✓ with deduplication, up to 3 per turn |
| Auto-recall before prompt | ✓ basic | ✓ adaptive — skips trivial/short queries |
| Noise filtering | ✗ | ✓ |
| Migration tool from built-in plugin | — | ✓ |

---

# Retrieval Pipeline

Queries pass through a multi-stage pipeline before results are injected into the agent prompt:

1. **Embed query** — using the configured OpenAI-compatible embedding provider
2. **Parallel search** — vector ANN search (cosine distance) + BM25 full-text search run simultaneously
3. **Hybrid fusion** — vector score used as base; BM25 hits receive a configurable weighted boost
4. **Rerank** — optional cross-encoder reranking via external API (60% cross-encoder score + 40% fused score)
5. **Lifecycle decay scoring** — recency boost, time decay, importance weight, length normalisation
6. **Filter** — hard minimum score, noise filter, MMR diversity deduplication
7. **Inject** — surviving memories injected as `<relevant-memories>` context block

If the reranker API fails, the pipeline degrades gracefully to cosine similarity reranking.

---

# Installation

# 1. Clone into your OpenClaw workspace

```
cd ~/.openclaw/workspace
git clone https://github.com/CortexReach/memory-lancedb-pro.git plugins/memory-lancedb-pro
cd plugins/memory-lancedb-pro
npm install
```

> **Common mistake:** Cloning the repo somewhere other than your workspace and then using a relative path in `plugins.load.paths`. Relative paths are resolved from the workspace root. Use an absolute path if cloning elsewhere.

# 2. Disable the built-in memory plugin

Only one memory plugin can be active at a time. If you previously used `memory-lancedb`, disable it before enabling this plugin.

# 3. Add to openclaw.json

```
{
  "plugins": {
    "load": {
      "paths": ["plugins/memory-lancedb-pro"]
    },
    "entries": {
      "memory-lancedb-pro": {
        "enabled": true,
        "config": {
          "embedding": {
            "apiKey": "${JINA_API_KEY}",
            "model": "jina-embeddings-v5-text-small",
            "baseURL": "https://api.jina.ai/v1",
            "dimensions": 1024,
            "taskQuery": "retrieval.query",
            "taskPassage": "retrieval.passage",
            "normalized": true
          }
        }
```

```
    }
   },
   "slots": {
    "memory": "memory-lancedb-pro"
   }
  }
}
```

Config changes require a gateway restart. With config watch enabled (default), this happens automatically.

---

# Key Configuration Options

| Option | Default | Notes |
|---|---|---|
| `autoCapture` | true | Capture memories at session end |
| `autoRecall` | true | Inject memories before prompt build |
| `smartExtraction` | true | Use LLM to classify memories instead of regex |
| `extractMinMessages` | 3 | Minimum messages before extraction runs |
| `captureAssistant` | true | Set false to only capture user messages |
| `retrieval.mode` | hybrid | `vector`, `bm25`, or `hybrid` |
| `retrieval.vectorWeight` | 0.7 | Weight for vector scores in hybrid fusion |
| `retrieval.bm25Weight` | 0.3 | Weight for BM25 scores in hybrid fusion |
| `rerank.enabled` | false | Enable cross-encoder reranking |
| `rerank.candidatePoolSize` | 12 | Candidates passed to reranker |
| `rerank.minScore` | 0.6 | Soft minimum score post-rerank |
| `rerank.hardMinScore` | 0.62 | Hard cutoff — below this is always dropped |
| `sessionMemory.enabled` | true | Store session summaries on `/new` |
| `autoRecall.minPromptLength` | 15 (EN) / 6 (CJK) | Skip recall for very short queries |

---

# Management CLI

The plugin ships with a CLI for direct memory management:

```
openclaw memory-pro list              # list stored memories
openclaw memory-pro search <query>    # semantic/keyword search
openclaw memory-pro stats             # storage stats
openclaw memory-pro delete <id>       # delete a specific memory
openclaw memory-pro export            # export all memories
openclaw memory-pro import <file>     # import memories
```

---

# Agent Tool Definitions

When loaded, the plugin registers these tools for the agent to use directly:

- `memory_recall` — retrieve relevant memories for a query
- `memory_store` — explicitly store a memory
- `memory_forget` — delete a memory by ID or query
- `memory_update` — update an existing memory

Plus additional management tools exposed via the CLI commands above.

---

# Multi-Scope Isolation

Memories can be scoped to control access between agents and users:

- global — shared across all agents
- agent:<id> — isolated to a specific agent
- project:<id> — shared within a project
- user:<id> — per-user isolation (useful for multi-user bots)
- custom:<name> — arbitrary named scope

---

# Telegram Setup

If running OpenClaw with Telegram, the easiest way to configure the plugin is via the bot directly. Send the following to your main bot:

> Help me connect this memory plugin with the most user-friendly configuration:
>
> https://github.com/CortexReach/memory-lancedb-pro
>
> Requirements:
>
> 1. Set it as the only active memory plugin
>
> 2. Use Jina for embedding and reranker
>
> 3. Use gpt-4o-mini for the smart-extraction LLM
>
> ... (continue with your preferences)

---

# Important Notes

> **jiti cache:** After modifying any `.ts` file in the plugin, you must clear the jiti cache before restarting the gateway, or OpenClaw will load stale compiled code:
> `rm -rf /tmp/jiti/ && openclaw gateway restart`

> **Memory quality guidelines:** Never store raw conversation summaries, large blobs, or duplicates. Prefer structured, atomic facts with keywords. On any tool failure or repeated error, call `memory_recall` with relevant keywords before retrying — the fix may already be stored.

> **Spaced repetition:** Frequently recalled memories decay more slowly, similar to spaced-repetition learning systems.

---

# Notable Community Forks

| Fork | Notable additions |
|---|---|
| `CortexReach/memory-lancedb-pro` | Primary upstream. Updated for OpenClaw 2026.3+ hook architecture. |
| `win4r/memory-lancedb-pro` | Widely referenced in docs; standard feature set. |
| `fryeggs/memory-lancedb-pro` | Unified edition — extends to Claude Code, Codex CLI, and Claude Desktop via shared LanceDB backend. |
| `kvc0769/memory-lancedb-pro` | Adds Volcengine multimodal embedding support. |
| `McBorisson/memory-lancedb-pro` | Uses RRF fusion (vs. weighted boost in other forks); includes JSONL distillation pipeline. |

Generated March 2026. Sources: CortexReach/memory-lancedb-pro, openclaw/openclaw docs, LanceDB blog.

Revision #1
Created 27 March 2026 10:12:30 by Conor
Updated 27 March 2026 10:13:51 by Conor