

Cortex: What If Your LLM Was a Wiki Compiler?

Automating the Karpathy Line of Compounding Knowledge

Manual wiki maintenance is brutal. takes longer to synthesize, cross-reference, and index than to read the original source.



Cortex automates the compounding effect.

Every new source doesn't just add one article.
It actively:

Updates 5 existing articles

Creates 2 new cross-links

Surfaces 1 hidden pattern

Reframing the Mental Model

You aren't programming with the LLM. You are programming the LLM.



Cortex contains zero LLM client code. No API keys, no prompt chains.

It is a **thin infrastructure** layer—an MCP server that reads a Markdown schema acting as the compiler specification.



Obsidian = The IDE

Where you browse, read, and explore.



The LLM = The Programmer

Reads sources, follows the schema, and writes wiki articles.

The Wiki = The Codebase

Grows, gets continuously refactored, and compounds in value.

The Compilation Pipeline

From Raw Input to Connected Knowledge

Drop a URL into inbox/. A shell hook nudges Claude
Code: “[Cortex] files waiting to be processed.”

Step 1: Ingest

Reads the URL via Jina Reader.
Generates frontmatter and saves
immutably to sources/.

Step 2: Compile

The LLM reads the schema
instructions. Decides which articles to
create or update. Writes synthesis
with `[[wikilinks]]` and source citations.

Step 3: Index

Updates the master index and
logs the operation. The Obsidian
vault updates in real-time.



Separation of Evidence & Synthesis

Immutable Sources vs. Mutable Wiki



Immutable Sources

- State: Fixed and permanent.
- Function: A definitive record of what was read.
- Rule: Once ingested, it never changes. If a source is outdated, you update the synthesis, not the evidence.

Mutable Wiki

- State: Continuously evolving.
- Function: Active synthesis and cross-referencing.
- Rule: Constantly refactored. Summaries are refined and new [[wikilinks]] are added as new evidence arrives.

Radical Minimalism, Emergent Intelligence

The intelligence lives in the
schema, not the server.



No UI. No build system. No embedded databases. When state is just human-readable files in a git repo, you get version control, backup, and pure portability for free. Everything else is Obsidian.