

Graphify + Obsidian

Graphify + Obsidian — Dein zweites Gehirn in Minuten

Ein Befehl macht aus **jedem Ordner** einen navigierbaren Knowledge-Graph. Code, PDFs, Bilder, Markdown, Audio, Video — alles wird zu verbundenem Wissen. Und mit **Obsidian** siehst du dieses Wissen als lebendige Karte: jede Notiz ein Knoten, jede Verbindung ein Backlink, alles durchsuchbar. Keine Vektor-Datenbank, kein ständiges Neu-Einlesen.

Woher's kommt: Andrej Karpathy (OpenAI-Mitgründer) schrieb auf X, KI solle *persistente Knowledge-Graphs* bauen, statt jedes Mal RAG-Chunks neu zu holen. ~48 Stunden später stand Graphify auf GitHub. 🔗 **Repo:**
<https://github.com/safishamsi/graphify>

⚠️ **Wichtig beim Installieren:** Das Paket heißt `graphifyy` — mit **doppeltem y**.
Das einfache `graphify` ist ein anderes Paket.

1. Graphify installieren

```
pip install graphifyy      # doppeltes y!  
graphify install
```

2. Graph + Obsidian-Vault bauen (in Claude Code)

```
/graphify . --obsidian      # aktueller Ordner, mit Obsidian-Vault  
/graphify ./raw --obsidian  # bestimmter Ordner
```

Das `--obsidian`-Flag ist der Schlüssel: Graphify legt zusätzlich zum Graphen einen fertigen **Obsidian-Vault** an (Ordner `obsidian/`). Der erste Lauf kostet einmal Tokens, jede Abfrage danach läuft gegen die kompakte `graph.json` — laut Repo-Test **bis zu 71,5x weniger Tokens pro Abfrage** (52-Dateien-Set aus Karpathy-Repos + Paper + Bildern).

3. Obsidian einrichten (dein zweites Gehirn)

1. **Obsidian** gratis laden: [obsidian.md](#) (Mac / Windows / Linux / iOS / Android).
 2. In Obsidian: „**Open folder as vault**“ → den von Graphify erzeugten `obsidian/`-Ordner auswählen.
 3. Fertig. Jede Node ist jetzt eine Notiz, jede Beziehung ein **Backlink**:
 - **Graph View** (`Cmd/Strg + G`): dein komplettes Wissen als interaktive Karte — klick dich durch die Verbindungen.
 - **Backlinks-Panel** (rechts): siehst du sofort, was mit was zusammenhängt.
 - **Quick Switcher** (`Cmd/Strg + 0`) & Suche: alles sofort auffindbar.
 4. **Aktuell halten**: Bei jedem neuen `--obsidian`-Build spiegelt der Vault den aktuellen Graphen. Eigene, dauerhafte Notizen also in einem **separaten** Ordner/Vault führen — der generierte Vault ist die lebende Wissenskarte.
-

4. Was Graphify erzeugt

DATEI / ORDNER	WAS DRIN IST
<code>graph.json</code>	Der Graph: Entity-Nodes (Konzepte, Funktionen, Klassen, Datasets) + Relationship-Edges

DATEI / ORDNER	WAS DRIN IST
<code>graph.html</code>	Interaktive Visualisierung im Browser — klicken, suchen, filtern
<code>obsidian/</code>	Obsidian-Vault mit Backlinks (siehe Schritt 3)
<code>wiki/</code>	Wikipedia-Style Wissensbasis
<code>GRAPH_REPORT.md</code>	Kern-Konzepte + überraschende Verbindungen + vorgeschlagene Fragen

5. Wie es Wissen liest

- **Code** → tree-sitter AST + Call-Graph (lokal, ohne LLM, 25 Sprachen)
- **Docs / PDFs** → LLM-Konzeptextraktion + Beziehungs-Inferenz (inkl. Citation-Mining)
- **Bilder** → Claude Vision (Architektur-Diagramme, Whiteboard-Fotos, Screenshots)
- **Video / Audio** → lokale Whisper-Transkription

6. Fakt vs. Vermutung — der entscheidende Punkt

Jede Verbindung ist getaggt:

- **EXTRACTED** — direkt in der Quelle gefunden (Fakt)
- **INFERRED** — erschlossen, mit Confidence-Score
- **AMBIGUOUS** — zur Prüfung markiert

Du weißt also immer, was belegt ist und was Vermutung. Dazu markiert Graphify **God-Nodes** (zentrale Konzepte mit den meisten Verbindungen) und **Surprising Connections** (unerwartete Cross-File-Links) — die tauchen in Obsidian direkt als Backlinks auf.

7. Der Graph pflegt sich selbst

```
graphify --watch           # Code sofort per AST, Docs lösen LLM-  
                           Reanalyse aus  
graphify hook install      # Auto-Rebuild nach JEDEM git commit  
graphify add https://arxiv.org/abs/XXXX # neues Paper in den bestehenden  
                           Graph
```

Ein SHA256-Cache verarbeitet nur **geänderte** Dateien neu — der Rest wird übersprungen.

8. Always-on (empfohlen)

```
graphify claude install
```

Danach liest Claude vor Architektur-/Wissensfragen automatisch die `GRAPH_REPORT.md` und navigiert per Graph-Struktur statt sich durch Dateien zu suchen. Funktioniert mit Claude Code, Codex u. a.

Für wen? Alle, die täglich mit AI-Coding-Tools arbeiten oder verstreute Notizen, Paper und Screenshots in ein strukturiertes zweites Gehirn verwandeln wollen. Obsidian- & Second-Brain-Leute: genau euer Ding.

Dein Wissen ist kein statisches Dokument mehr — es lebt und wächst mit dir.