

AI WORKFLOW

Claude Tokens sparen

3 Tools gegen ein explodierendes Nutzungslimit

68k ★

RTK

73k ★

Ponytail

58k ★

Graphify

Seit Fable zurück ist, brennt bei den meisten das Limit schneller durch als sonst.

Das liegt fast nie an der Aufgabe selbst, sondern daran, wie viel unnötigen Kontext Claude bei jedem Schritt mitliest. Diese drei Tools setzen genau da an.

DAS PROBLEM

Warum dein Limit so schnell brennt

Drei Stellen, an denen Claude unnötig Tokens verbrennt

01 Terminal-Output

Jeder Output, jede Fehlermeldung geht komplett an Claude weiter — dabei sind neunzig Prozent davon für die Aufgabe irrelevant.

02 Über-Engineering

Claude schreibt von sich aus oft mehr Code als nötig und baut Absicherungen für Fälle, die nie eintreten.

03 Wiederholte Suche

Claude durchsucht euren kompletten Code-Ordner bei fast jeder Anfrage neu, jedes einzelne Mal von vorne.

DIE LÖSUNG

3 Tools, die das lösen

Jedes Tool behebt genau eins der drei Probleme

RTK — Terminal-Filter

Filtert den Terminal-Output, bevor er bei Claude ankommt, sodass nur durchkommt was wirklich gebraucht wird. 68.000 Sterne auf GitHub.

github.com/rtk-ai/rtk

Ponytail — Der faule Senior-Dev

Bringt Claude dazu, nur den Code zu schreiben der wirklich nötig ist, ohne unnötige Absicherungen. 73.000 Sterne auf GitHub.

github.com/DietrichGebert/ponytail

Graphify — Code-Landkarte

Baut einmalig eine durchsuchbare Übersicht deines Code-Ordners, statt jedes Mal alles neu zu durchsuchen. 58.000 Sterne auf GitHub.

github.com/safishamsi/graphify

ÜBERSICHT

Alle drei im Vergleich

Auf einen Blick, was jedes Tool löst

Tool	Löst	Sterne
RTK	Zu viele Terminal-Tokens	68.000
Ponytail	Über-Engineerter Code	73.000
Graphify	Wiederholte Ordner-Suche	58.000

Direkt-Links:

github.com/rtk-ai/rtk

github.com/DietrichGebert/ponytail

github.com/safishamsi/graphify

MODELL-GUIDE

Wo Fable 5 am besten abschneidet

5 Bereiche mit den stärksten Ergebnissen laut Anthropic-Benchmarks

01 Große Code-Migrationen

80,3% auf SWE-bench Pro. Stripe hat damit eine Migration von 50 Millionen Zeilen Ruby-Code in nur einem Tag abgeschlossen.

02 Lange, selbstständige Agent-Sessions

Hält über mehrere Tage hinweg die Instruktionstreue bei komplexen Aufgaben — stark für Loops und Schedules, die lange laufen.

03 Finanzanalyse

Höchster Score aller Modelle auf Hebbias Finance Benchmark: liest Earnings Reports, 10-Ks und Proxy Statements und rechnet mit den Zahlen.

04 Dokumentenlastige Arbeit

Versteht Diagramme, Charts und Tabellen in PDFs — starke Verbesserung bei Aufgaben aus Recht, Finance und Architektur.

05 Komplexe Analytics-Aufgaben

Erstes Modell über 90% auf einem Analytics-Benchmark für lange, komplexe Auswertungen — 10 Punkte über Opus.

NÄCHSTER SCHRITT

Jetzt einrichten lassen

So kommst du in wenigen Minuten zu weniger Tokenverbrauch

01 Link öffnen

Öffne das GitHub-Repo des Tools, das dein größtes Problem löst — Terminal, Code-Umfang oder Suchzeit.
Direktlinks auf Seite 3.

02 An Claude schicken

Schick den Link an Claude Code und bitte darum, das Tool für dein Projekt einzurichten.

03 Testen

Lauf eine gewohnte Aufgabe einmal mit und einmal ohne das Tool und vergleiche den Tokenverbrauch.

Du willst mehr solche KI-Workflows für dein Business?

Folge @tim_huck_ auf Instagram für tägliche KI-Tools und Strategien, die dir einen unfairen Vorteil geben.

→ @tim_huck_ auf Instagram

→ www.shortmedia.agency