

CLAUDE CODE

Die saubere Übergabe

Warum Claude in langen Sitzungen schlechter wird und wie du ihn mit einer einzigen Datei wieder auf Anfang bringst

18 von 18

Modelle betroffen

20 bis 50 %

weniger Treffer

1 Datei

löst das Problem

Das Problem hat einen Namen, es heißt Context Rot.

Chroma hat achtzehn große Modelle darauf getestet und alle achtzehn wurden schlechter, je mehr im Fenster stand. Bei der Suche nach einer bestimmten Information fiel die Trefferquote zwischen zehntausend und hunderttausend Token um zwanzig bis fünfzig Prozent.

01 Warum Zusammenfassen das nicht löst

Auf Seite 3 steht, was bei einer Zusammenfassung wirklich passiert und was dabei ungefiltert mitwandert.

02 Die Übergabe und die zwei Prompts

Auf Seite 4 und 5 steht der Aufbau der Datei und der Prompt zum Kopieren, einmal zum Rausschreiben und einmal zum Wiederaufnehmen.

03 Vier Befehle, die kaum jemand kennt

Auf Seite 6 steht, wann du welchen nimmst, also Fokus setzen, nur einen Teil zusammenfassen oder früher automatisch auslösen.

Quellen: trychroma.com/research/context-rot und die offizielle Claude-Code-Doku unter code.claude.com/docs, beides geprüft am 3. September 2026.

SCHRITT 1

Erst mal nachsehen

Bevor du irgendwas machst, schau dir an, was dein Fenster überhaupt frisst

IN CLAUDE CODE EINTIPPEN

```
/context
```

01 Was du da siehst

Ein farbiges Raster mit deinem aktuellen Verbrauch. Laut Doku zeigt es dir zusätzlich Vorschläge, welche Werkzeuge besonders viel Kontext ziehen, wo deine Speicherdateien aufgeblasen sind und wann es eng wird. Wenn du über dem Limit bist, steht direkt dabei, wie weit und welcher Befehl dir Platz schafft.

02 Was dich meistens überrascht

Bei den meisten geht der Platz nicht für die Unterhaltung drauf, sondern für Dateien, die einmal gelesen wurden, und für Werkzeuge aus angebundenen Diensten. Also nicht für das, worüber du gerade redest.

03 Die Regel dahinter

Alles, was im Fenster steht, wird bei jeder einzelnen Nachricht wieder mitgeschickt. Ein Ordner, den du vor zwei Stunden mal aufmachen lassen hast, kostet dich also immer noch bei jeder Antwort etwas, und er verschlechtert obendrein die Trefferquote.

Mach das einmal mitten in der Arbeit und nicht am Anfang.

Am Anfang ist das Fenster natürlich leer und du siehst nichts. Interessant wird es genau dann, wenn du das Gefühl hast, dass er anfängt Sachen zu vergessen.

DER EIGENTLICHE PUNKT

Was beim Zusammenfassen passiert

Manches wird frisch von der Festplatte geholt, der Rest wird eingedampft

DAS HIER	PASSIERT DAMIT
CLAUDE.md im Projekt	Wird frisch von der Festplatte nachgeladen
Automatisches Gedächtnis	Wird frisch von der Festplatte nachgeladen
Der Plan aus dem Planmodus	Wird frisch von der Festplatte nachgeladen
Gelesene und geänderte Dateien	Die fünf zuletzt geänderten werden neu gelesen
Aufgerufene Skills	Kommen zurück, gekürzt auf 5.000 Token je Skill
Die ganze Unterhaltung	Wird zu einer Zusammenfassung eingedampft

Da steht die Lösung eigentlich schon drin.

Was als Datei auf der Festplatte liegt, kommt sauber zurück. Was nur in der Unterhaltung stand, wird zusammengefasst, und dabei landet auch jede Sackgasse und jede falsche Annahme mit im Ergebnis. Deswegen bringt es so viel, den Stand einmal rauszuschreiben statt ihn im Chat liegen zu lassen.

01 Ein Sonderfall, den du kennen solltest

Eine Datei über 5.000 Token kommt nach dem Zusammenfassen nur noch als Pfad zurück und nicht mehr mit Inhalt. Sie steht dann als Referenz da. Bei großen Dateien musst du also damit rechnen, dass er den Inhalt nochmal lesen muss.

Quelle: code.claude.com/docs/en/context-window, Abschnitt What survives compaction, geprüft am 3. September 2026.

DIE DATEI

So sieht die Übergabe aus

Fünf Überschriften, mehr braucht das Ding nicht

HANDOFF.MD IM PROJEKTORDNER

```
# Übergabe · <Projekt> · <Datum>

## Stand
Was fertig ist und läuft, mit Dateipfad.

## Gescheitert
Was ich probiert habe und woran es lag.

## Offen
Was als Nächstes drankommt, in der Reihenfolge.

## Wichtige Dateien
Die fünf, die man kennen muss, je ein Satz.

## Entscheidungen
Was festgelegt ist und warum.
```

01 Warum Gescheitert die wichtigste Überschrift ist

Das ist der Teil, den eine normale Zusammenfassung nie sauber hinbekommt. Ohne den fängt die neue Sitzung fröhlich mit dem Weg an, der letzte Woche schon nicht funktioniert hat, und du sitzt eine Stunde daneben und wunderst dich.

02 Warum Entscheidungen mit reinmuss

Sonst wirft er dir Sachen wieder um, die ihr längst geklärt hattet. Also die Wahl einer Bibliothek, die Struktur der Ordner oder die Namensgebung. Ein Satz reicht, aber die Begründung muss dabeistehen.

03 Wo die Datei liegen sollte

Direkt im Projektordner als HANDOFF.md. Dann findet er sie sofort und du kannst sie auch selber schnell überfliegen. Wenn du an mehreren Sachen parallel arbeitest, nimm einen Unterordner und häng das Datum an den Dateinamen.

DIE ZWEI PROMPTS

Einmal raus, einmal rein

Den ersten schickst du vor dem Leeren, den zweiten danach

1 · VOR DEM LEEREN

Schreib mir eine Übergabe nach HANDOFF.md, damit ich in einer neuen Sitzung genau hier weitermachen kann. Halt dich an diese Gliederung und schreib nichts rein, was nicht passiert ist:

```
## Stand – was fertig ist und läuft, mit Dateipfad
## Gescheitert – was ich probiert habe und woran es lag,
  je ein Satz, damit ich es nicht nochmal probiere
## Offen – was als Nächstes drankommt, in der Reihenfolge
## Wichtige Dateien – die fünf wichtigsten, je ein Satz
## Entscheidungen – was festgelegt ist und warum
```

Keine Zusammenfassung unserer Unterhaltung, nur der Stand.

2 · NACH DEM LEEREN

Lies HANDOFF.md und sag mir in drei Sätzen, wo wir stehen. Fang danach mit dem ersten Punkt unter Offen an.

Der letzte Satz im ersten Prompt ist der entscheidende.

Ohne den bekommst du wieder eine Nacherzählung von eurem Chat, also genau das, was du eigentlich loswerden wolltest. Du willst den Zustand haben und nicht den Verlauf.

01 Dazwischen kommt das Leeren

Zwischen den beiden Prompts tippst du slash clear. Damit fängt die Unterhaltung bei null an, und du kannst ihr sogar einen Namen mitgeben, dann findest du die alte über slash resume wieder.

Quelle: code.claude.com/docs/en/commands, geprüft am 3. September 2026. Die beiden Prompts sind von mir und stehen so nicht in der Doku.

DIE BEFEHLE

Wann du **welchen** nimmst

Die Übergabe ist der eine Weg, es gibt aber noch drei andere

BEFEHL	WOFÜR
<code>/context</code>	Nachsehen, was das Fenster gerade frisst
<code>/compact mit Zusatz</code>	Zusammenfassen, aber du gibst den Fokus vor
<code>/rewind</code>	Nur einen Teil der Unterhaltung zusammenfassen
<code>/autocompact</code>	Festlegen, ab wann er von selbst zusammenfasst
<code>/clear</code>	Alles leeren, wenn ein neues Thema anfängt
<code>/resume</code>	Eine geleerte Unterhaltung wiederfinden

DIE ZWEI MIT ZUSATZ

```
# Fokus vorgeben, statt ihn raten zu lassen
/compact focus on the auth bug fix

# früher auslösen, hier bei 500.000 Token
/autocompact 500k
```

Wenn du mitten in einer Sache steckst, nimm den Fokus-Zusatz.

Das Beispiel oben steht genau so in der Anthropic-Doku. Die Zusammenfassung behält dann das, was du vorgibst, statt das, was die Automatik für wichtig hält. Für einen echten Themenwechsel ist trotzdem die Übergabe plus slash clear der sauberere Weg.

Quelle: code.claude.com/docs/en/commands und [/docs/en/context-window](https://code.claude.com/docs/en/context-window), geprüft am 3. September 2026.

NÄCHSTER SCHRITT

Direkt umsetzen

Einmal beim nächsten längeren Projekt durchziehen, dann sitzt es

01 Heute noch slash context eintippen

Nimm die Sitzung, in der du gerade steckst, und schau dir das Raster an. Meistens siehst du sofort, was da eigentlich Platz wegnimmt, und danach verstehst du auch, warum er langsam schlechter wird.

02 Den ersten Prompt einmal abschicken

Bevor du das nächste Mal Feierabend machst, lässt du dir die Übergabe schreiben. Lies sie einmal durch und streich raus, was nicht stimmt. Beim ersten Mal dauert das fünf Minuten, danach kaum noch.

03 Am nächsten Tag den zweiten schicken

Neue Sitzung aufmachen, den zweiten Prompt rein und schauen, ob er die drei Sätze richtig hinkommt. Wenn ja, hast du deine Übergabe. Wenn nicht, fehlt in der Datei etwas und du weißt genau was.

Du willst sehen, was bei Claude Code sonst noch geht?

Auf Instagram zeige ich, was neu ist und was ich davon behalte, sobald ich es selber getestet habe.

→ @timhuck.ai auf Instagram

→ www.shortmedia.agency

Alle Angaben geprüft am 3. September 2026. Claude Code ändert sich schnell, die Befehle können also dazukommen oder anders heißen. Im Zweifel einmal slash help eintippen.