Asking your own AI about your causal maps

12 Nov 2026

This page is for people who code their data in Causal Map and also have Claude, ChatGPT or another AI chat app open while they write. It explains what changes when the chat app can read your causal maps directly, how that differs from MapCat inside the app, and what to watch for. The steps for connecting are on the help page Causal Map in Claude, ChatGPT and other AI apps.

The idea

Most of the work of an evaluation report happens away from the analysis software: in a draft, a theory of change, a client's terms of reference, a chat with an AI about how to structure the findings. Getting evidence from a causal map into that conversation usually means copying a table or a screenshot across, so the AI reasons about the copy.

Instead, Causal Map can be one of the tools your chat app uses. You add it once as a connector and sign in with your Causal Map account. From then on the model in your chat can list your projects, count and rank factors, compare groups of sources, draw a map after a set of filters or from a bookmark you saved, and fetch the quotes behind any link, in the same conversation as the rest of your work. The standard that makes this possible is MCP, the Model Context Protocol, which lets an AI app call tools that other services provide.

Two things follow. The counts come from Causal Map rather than from the model's reading of a pasted table: the model asks, and Causal Map's own query tool and filter engine, the ones the app uses, do the counting. And every map it shows comes with a link that opens the same view in the app, so you can move from the conversation to the full interface whenever a question needs more than a chat can show.

What a session looks like

Suppose you are drafting the findings on school attendance and ask: "What do people in the endline project say leads to children staying in school, and do the women and men tell it differently?"

The connector carries the same guidance MapCat works from, so a well-behaved model first says what the question means in causal-map terms: which factor is the outcome, whether to count sources or citations, and how many steps back to trace. It then runs that, perhaps a path trace towards the outcome, split by a source group for gender, shows the map, and quotes a few of the links behind the strongest arrows.

Where two ways of counting could disagree, it uses both and says so. It ends by suggesting what to look at next.

In claude.ai and the Claude desktop app the map appears in the conversation, drawn by Causal Map itself rather than by the model. Clicking an arrow brings up the quotes behind it; clicking a factor brings up its links. In other apps you get the same answer with a link to open the map in Causal Map.

Perhaps more importantly, depending on your use case: you can just chat with your favourite AI about your project. It has all the data and can answer clever questions and stupid questions with equinimity.

How it differs from MapCat

MapCat lives inside the app. It sees what you have open, can change your project (always with a confirm step), runs AI coding, including Code Like Me, and builds report decks. It runs on Causal Map's own AI arrangements, including the region setting each project has.

The connector lives in your chat app. At the moment it only reads; it cannot code, edit or delete anything. It uses the model and the subscription you already have with that app, so it costs no Causal Map credits. Its strength is that you use it in the same conversation as everything else you are working on: the draft paragraph you are checking, the programme's theory of change, notes from another dataset.

So the choice is about where the work is. Coding, cleaning and building the project belong in the app with MapCat; writing with the evidence at hand belongs wherever you write.

Good uses

- Checking a sentence in a draft against the data: "Does the coding support the claim that the savings groups led to new businesses? How many sources say so, and what do they say?"
- Exploring a project someone has shared with you before opening the app at all.
- Comparing what different groups of respondents say leads to an outcome, with the quotes to hand for the write-up.
- Asking how a project was coded: what its coding prompt contains and which coding runs produced its links.

What to watch for

The model is still interpreting. It chooses the filters, and a plausible-looking chain can still answer a slightly different question from the one you meant. Read what it ran: it should name its filters in order, following the filter ordering rules, and the link it gives opens exactly that view. The usual cautions of answering questions with causal maps apply, among them that a ranking resting on a handful of sources is weak, and that a path from A to C stitched together from different people's stories is nobody's story (the transitivity trap). Before quoting a count in a report, check the coding behind it.

Some of the app's filters are not available through the connector, notably the ones that use AI (such as Soft Recode and the clustering filters) and Pathways. A bookmark that uses one comes back with its link to open in the app instead of a map.

Where your data goes

When the connector answers, the factor labels, quotes and counts it returns go to the AI company behind your chat app, under your agreement with that company. Causal Map's own AI arrangements, such as a project's AI region, do not cover that route. Before connecting a project, check what your consent forms and client agreements say about which AI services may see the data. If a project's data must stay in one region, or must not reach a particular provider, keep its analysis inside the app.

Getting started

Every Causal Map account with a confirmed email can connect, free accounts included. The address and the steps for Claude, Claude Code, Cursor and other apps are on the help page Causal Map in Claude, ChatGPT and other AI apps.

This page was drafted by the Causal Map team with the assistance of Claude, an AI model from Anthropic; the content and judgements are our own.