# 🌻 Tasks 2 & 3 – Extensions – Introduction

Chapter contents.

9 Oct 2025

## What this chapter is all about

> In the previous chapter we introduced **minimalist causal coding**:

- we code **links** between named factors (plain-language labels)
- we let **factor labels** do most of the work (e.g. `wealth` vs `poverty`)
- we start from a **links table** (one row per coded causal claim), rather than building a separate factor metadata structure

That approach has general applicability. This folder collects a set of **practical extensions** that make the approach more useful for answering common evaluation questions — mainly by simplifying, querying, and comparing the map.

All the extensions in this folder are implemented in the Causal Map app, but the ideas are tool-independent.

As you read:

- start with the practitioner sections ("what is this for / how do I interpret it?")
- where a page needs more formal material, it's placed at the bottom under **Formal notes (optional)**

# Formal notes (optional)

If you want deeper theory/background (not required for using the method):

- Minimalist coding for causal mapping
- A formalisation of causal mapping

PAGES IN THIS CHAPTER

### Combining links into bundles

> In most projects, the data contains many repeated causal claims with the **same cause and the same effect** (often across many sources). We call these **bundles** (or **co-terminal link bundles**).

### The factors table

> The factors table is a **summary view** of your map: it tells you which factors are most prominent in the current view of the data, and how that changes across groups.

### Simplification - factor and link frequency

> This extension is about **simplifying** a causal map by keeping only the **most frequently mentioned**:

### Exclude links based on group or other metadata

> This extension is about filtering links using **metadata** (information attached to links and/or their sources), not just factor labels.

### Focus or exclude factors

> This extension is about using **factor labels** to carve out a useful subgraph of your causal map.

### Collapsing factor labels and excluding brackets

> This extension is about **using factor labels to unify many "different-looking" factors into one**.

### Path tracing and source tracing

> Path tracing is for answering questions like:

### Hierarchical coding

> When you code lots of sources, you quickly end up with **too many near-duplicate causes and effects** (and therefore too many nodes on the map). Hierarchical factor labels let you keep the detail *and* produce a smaller “summary map” by “zooming out” to a higher level.

## Opposites

> Opposites coding is for when your data naturally contains paired factors like:

## Plain coding

> Causal mapping doesn’t usually deal with the kind of non-causal themes which are the focus of ordinary QDA (like in NVivo!). However sometimes it can be really useful to be able to simply note the presence of something without any causal connection.

## AI extensions – Clustering, Soft Recode, Magnetisation

> You have already coded your dataset, manually or using AI, and now you want to relabel.

## Social Network Analysis – SNA

> For this extension, we can simply tweak our coding rule to code sending/receiving relationships between social actors instead of causal links between factors or events.

## Reporting global network statistics

> We do not actually provide these map-level statistics yet, e.g. "how connected overall is this whole map"?