Working Papers

Chapter contents.

23 Dec 2025

This chapter is a new set of working papers about causal mapping.

Core papers (start here)

- Minimalist coding for causal mapping: the core coding stance (“barefoot” link coding), why it is useful, and where it breaks.
- A formalisation of causal mapping: companion spec—data structures + conservative rules for aggregation/query.
- Causal mapping as causal QDA: positioning for qualitative methods / CAQDAS audiences.

Practical extensions (operations on a links table)

- Magnetisation: soft recoding with “magnets” (standardise labels at scale without re-coding quotes).
- A simple measure of the goodness of fit of a causal theory to a text corpus: coverage-style diagnostics for ToC fit.
- Combining opposites, sentiment: opposites transforms, sentiment as an annotation layer, and “despite” link typing.
- Hierarchical coding: hierarchical labels (;) and zoom-style simplification.

Related notes / fragments / examples

- !!!Qualitative Split-Apply-Combine: small-Q framing; causal mapping as a SAC variant; where genAI fits.
- 250! causal mapping turns QDA on its head: a short argument/fragment (kept for reuse).

- Assessing change in (cognitive models of) systems over time: worked example of “clerk vs architect” (auto-extraction + magnet-style structuring).

PAGES IN THIS CHAPTER

Minimalist coding for causal mapping

To the best of our knowledge, all major approaches to causal mapping (@axelrodStructureDecisionCognitive1976, @edenAnalysisCauseMaps1992, @laukkanenComparativeCausalMapping2016, @mauleCognitiveMappingCausal2003) would most like code (1) as `amount eaten` --> `energy level`. And they would treat (2) pretty much the same.

A formalisation of causal mapping

Abstract

Combining opposites, sentiment

Instead we take a **piece-by-piece approach**:

Despite-claims

Narratives often contain claims of the form:

Causal mapping as causal QDA

Causal mapping is a well-established family of approaches in social science for representing “what influences what”, according to sources, as a network of claims. This paper presents causal mapping as an interesting variant of Qualitative Data Analysis (QDA) in which the primary act of coding is not “apply a theme”, but **code a causal link** (an ordered pair of cause/effect labels) grounded in a quote and source. The resulting list of causal links can then be queried (filtering, tracing paths, etc) to answer research questions. Qualitative judgement (what are the main cause/effect labels and how are they organised?) remains central while many of the other tasks become more reproducible, checkable, and scalable. We will demonstrate causal mapping using Causal Map (app.causalmap.app) which is free to use for public projects.

A simple measure of the goodness of fit of a causal theory to a text corpus

See also: [[000 Working Papers ((working-papers))]]; [[005 Minimalist coding for causal mapping ((minimalist))]]; [[900 Magnetisation]].

Magnetisation

Intended audience: people who have done open-ended (often in-vivo) causal coding and need to standardise factor vocabularies for readable maps/tables without destroying provenance.

Assessing change in (cognitive models of) systems over time

Lonely in London

This paper consists of a human-written wrapper (Abstract and Reflection) around a virtual paper which was entirely written by AI with no human interaction. No dedicated CAQDAS software was used; instead, the project was done inside Cursor, a generic workspace for editing text and code; its AI agent is able to create and edit memo (text) files including its own instructions. The authors gave the AI some interview texts on loneliness and a high-level instruction to develop and iteratively apply a thematic analysis methodology of its own choosing. The AI planned the workflow, carried it out, and produced the final paper, as described in the Auto-ethnographic Reflection. It kept notes of how its workplan evolved, with explicit links to the source texts. The initial instructions, all the intermediate memo files it created and the final paper are available in a public GitHub repository.