🌻 Causal mapping as causal QDA

📅 23 Dec 2025

Abstract

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.

This paper is written for QDA/CAQDAS users who want:

- a clear definition of causal coding as a qualitative method,
- an account of why it can be simpler to apply than broader forms of thematic coding,
- an introduction to using Causal Map,
- an introduction to how causal analysis can provide answers to useful questions,
- a comparison of causal mapping to neighbouring approaches (thematic analysis, qualitative content analysis, systems modelling).

Unique contribution (what this paper adds):

- It defines causal mapping as **link-coding with provenance** (a links table as the core qualitative product), and it is explicit about the semantics: **claims ≠ facts**.
- It frames analysis as an explicit **pipeline of operations** over the links table (filters/transforms → derived views), rather than as a single narrative synthesis step.
- It proposes a bounded role for LLMs as a **checkable extraction assistant** ("clerk") and locates the interpretive burden in human choices about transforms and synthesis ("architect").

See also:

- Minimalist coding for causal mapping (coding stance, extensions, limits)
- A formalisation of causal mapping (companion spec)
- Combining opposites, sentiment (opposites/sentiment/despite as extensions)
- Magnetisation (soft recoding / magnets)
- A simple measure of the goodness of fit of a causal theory to a text corpus (coverage-style fit diagnostics)
- Working Papers (hub page for the working-paper set)

1. What is "causal QDA"?

In ordinary qualitative coding, a code typically denotes a **concept/theme**. In causal QDA, a coded unit denotes a **causal claim** expressed in the text:

- a **cause (influence factor)** label,
- an **effect (consequence factor)** label,
- a **verbatim quote** that provides evidence for the claim,
- and a **source identifier** (so we can maintain provenance).

One coding act yields an ordered pair: `Cause -> Effect`. A dataset of such acts yields a **links table**. The set of factor labels is derived from the endpoints of links; factors “exist” here primarily as participants in claims.

This paper treats that links table as the core qualitative product: a structured repository of **evidence-with-provenance** that can be inspected and re-analysed.

In evaluation practice, closely related “causal-claims-first” ways of working with narrative evidence already appear in the wild (e.g. causal mapping as described on BetterEvaluation (n.d.), and QuIP-style narrative causal-map visualisations (n.d.)).

1.1 A tiny worked example (what one coding act looks like)

If an interviewee says:

> “The floods destroyed our crops.”

then a causal-coding act records a link such as:

- `Floods -> Crops destroyed`

along with the quote and its source id. In this framing, the link *is* the code. A set of such links can be immediately visualised as a network (or analysed as a table) with off-the-shelf tools.

2. Why causal coding is often easier (and more checkable) than “find the themes”

The instruction “find the main themes” is (legitimately) open-ended: it depends on theoretical stance, positionality, research question, and the analyst’s preferred granularity. That openness is often a feature of “Big-Q” qualitative work; but it makes systematic comparison and scale difficult (e.g. thematic analysis (n.d.)).

This difference is also visible when people use generative AI. “List the main themes in this document” can be a useful time-saver, but it is massively sensitive to what one means by *theme* (and to the analyst’s implicit theory of what “matters”). You can narrow the prompt (“Identify the main kinds of relationship issues mentioned”), but at that point you are already moving from open generation towards a more constrained extraction task.

The causal coding task is narrower:

> Identify each passage where the text says that one thing influenced another, and record what influenced what.

This does not remove judgement (labels still matter; causal language can be ambiguous), but it reduces degrees of freedom at the point of coding. In practice, that usually improves:

- **traceability** (every link can be checked against a quote),
- **comparability** (multiple coders/teams can aim at the same target representation),
- **automation** (an AI can be constrained to do the low-level extraction task).

This is a “small-Q” move: it is not a claim that causal QDA replaces interpretive qualitative work, but that it is a useful, rigorous option when the research questions are themselves causal (which they often are in evaluation and applied social research).

2.1 Example: “themes” vs “links” on the same excerpt

Consider:

> "After the clinic started opening on Saturdays, I didn't have to miss work, so I could actually attend."

A "theme finding" pass might code: `Access`, `Clinic opening hours`, `Employment constraints`, `Attendance`.

A causal-coding pass would typically try to capture the explicit influence structure:

- `Saturday opening -> Not missing work`
- `Not missing work -> Attendance`

Both can be valuable, but the causal representation is immediately queryable as a mechanism (and can be checked line-by-line against quotes).

---

3. The output is not "just codes": it is a queryable qualitative model

Ordinary QDA typically culminates in a narrative account plus some supporting tables. Causal QDA yields a different primary object: a **network of causal claims**.

Once you have a links table, you automatically have a graph:

- nodes = factor labels
- directed edges = coded claims (often bundled by identical endpoints)

That graph is a qualitative model in a specific sense:

- it is a model of **what sources claim** influences what,
- not (by itself) a model of causal reality.

The payoff is that you can answer many questions *by querying this model* — without needing to ask an AI to produce a global synthesis, and without hiding methodological steps inside the analyst's head.

3.1 Example questions that become natural once you have a links table

- "What are the most frequently mentioned upstream influences on `Attendance`?"
- "How do the pathways into `Wellbeing` differ for younger vs older respondents?"
- "Which links are contested (both `X -> Y` and `X -> ~Y` appear), and by which subgroups?"

4. A transparent "library of operations" (filters + views)

Causal mapping analysis is often best described as a **pipeline**: pass the links table through a sequence of operations, then render a view (map/table).

At a high level these operations fall into:

- **Row selection**: restrict sources and/or links (contexts, evidence thresholds).
- **Topological selection**: retain only links on paths relevant to a question (e.g. mechanisms connecting X to Y).
- **Label transforms**: rewrite labels for summarisation (e.g. zoom/hierarchy; opposites).
- **Bundling + metrics**: compute `Citation_Count` / `Source_Count` etc.
- **Views**: map view, factors view, tables, exported summaries.

The crucial methodological point is not which software you use, but that the meaning of a derived map is always:

> “This view of the evidence, after these explicit transformations.”

This is what makes the method checkable and extensible: other researchers can replicate the same pipeline on the same links table, or change one step and see what changes.

4.1 Example: a concrete analysis pipeline (from a broad corpus to a specific view)

Suppose you want: “Mechanisms connecting `Training` to `Adoption`, but only for the younger respondents, and only where more than one source supports each link.”

One explicit pipeline could be:

- **Context selection**: keep only sources where `Age_Group = Younger`.
- **Evidence threshold**: keep only link bundles with `Source_Count >= 2`.
- **Path tracing**: retain links on paths from `Training` to `Adoption` (with an explicit path-length limit).
- **(Optional) zoom**: rewrite hierarchical labels to level 1 to produce a high-level view.
- **Render view**: show the resulting subgraph as a map, but keep click-through to the underlying quotes for each remaining link.

The point is not that this exact pipeline is “right”, but that it is explicit: the reader can see what was done, rerun it, and inspect what evidence is inside the view.

4.2 Causal QDA as “qualitative split-apply-combine” (and why this is a good place for AI)

A useful way to describe this workflow is as a qualitative variant of the **split-apply-combine** strategy: break a hard analytic problem into manageable pieces, operate on each piece consistently, then recombine into a coherent answer ({wickham 2011).

In causal QDA, the mapping is unusually clean:

- **Split (operationalise the research question)**: reduce the messy corpus to a repeatable micro-task: extract *each explicit causal claim* and record it as a link with provenance (`Cause`, `Effect`, `Source_ID`, `Quote`). This is the minimalist "barefoot" stance (see Minimalist coding for causal mapping).
- **Apply (a library of deterministic operations)**: run an explicit pipeline of filters/transforms/queries on the links table (context restriction, evidence thresholds, path tracing, zoom/hierarchy, opposites transforms, bundling, etc.). This is the "library of operations" described above.
- **Combine (synthesis and reporting)**: recombine outputs into an argument: a set of maps/tables plus a narrative interpretation, optionally including explicit fit diagnostics (e.g. theory vocabulary coverage; see A simple measure of the goodness of fit of a causal theory to a text corpus).

This framing also clarifies a practical division of labour when using LLMs:

- LLMs are best used as a **clerk** in the Split step (exhaustive extraction with quotes), because outputs are locally checkable and errors are local.
- Humans remain the **architect** in Apply/Combine: choosing transforms, deciding magnets/codebooks (see Magnetisation), and writing the interpretive account. A worked "architect vs clerk" example is in Assessing change in (cognitive models of) systems over time.

5. Reproducible ↔ emergent: where causal QDA sits

Many qualitative traditions sit towards the emergent end of a spectrum: questions, codes, and interpretations are refined through an iterative, interpretive process, and the final output is primarily narrative synthesis.

Causal QDA tends to sit further towards the reproducible end on some dimensions:

- a more explicitly constrained coding task (code causal claims),
- a structured intermediate product (links table + quotes),
- a documented analysis pipeline (filters/views) that others can rerun.

This does not mean causal QDA is “objective” or that it removes positionality. It means that a larger portion of the analysis chain becomes explicitly inspectable: anyone can trace from a map edge to the underlying quotes, and from a view to the sequence of operations that produced it.

6. AI in causal QDA: the “low-level assistant”, not the analyst

Generative AI can be used in at least two ways:

- **black-box synthesis**: ask the model for themes, a theory, or a causal map directly from a corpus.
- **constrained assistance**: ask the model to do the lowest-level, checkable work (extract candidate links + quotes), while humans control the pipeline and interpret results.

Our stance is the second. The reason is not moral; it is methodological:

- if an AI invents or “smooths” meaning during synthesis, it is hard to audit,
- whereas if an AI outputs a *list* of links each grounded in a quote, the output is directly checkable and errors are local.

Once you have the links table, most analysis steps can be deterministic and transparent (filters, transforms, bundling, path tracing), keeping the core interpretive burden where it belongs: on the human researcher.

6.1 Example: what “constrained” AI assistance looks like (and what it should output)

For a given transcript chunk, the AI can be instructed to output a list of candidate links, each with:

- the exact quote span
- a proposed `Cause` label
- a proposed `Effect` label
- a confidence/notes field (optional)

Example output items might look like:

- Quote: “We stopped going because the bus fare went up.” → `Bus fares increased -> Attendance decreased`
- Quote: “When the midwife explained it, I started washing my hands more.” → `Midwife training -> Hand washing`

These are still proposals (humans can edit labels and reject links), but each item is locally auditable.

7. Relationship to neighbouring QDA approaches

7.1 Thematic analysis / qualitative content analysis

Causal QDA is compatible with many QDA workflows: it can be used alongside thematic coding (e.g. thematic analysis (n.d.)) or qualitative content analysis (e.g. Mayring’s approach (n.d.)), or as a front-end that captures a causal layer of meaning that is often what evaluators ultimately need (drivers, barriers, mechanisms, pathways).

The key difference is that causal QDA stores **relations** (ordered pairs), not only categories. That enables subsequent reasoning and querying that is hard to do robustly if you only have unconnected theme tags.

7.2 “Post-coding” conversational analysis with AI

Some AI-assisted QDA approaches propose moving away from coding into a structured dialogue with an AI, using question lists and documented conversations as the main analytic trace (e.g. conversational analysis with AI) (Friese 2025). This is an interesting and plausible direction for some kinds of interpretive work.

Causal QDA differs in that it retains a strong “small-Q” intermediate representation: a quote-grounded links table plus deterministic analysis steps. The point is not to rule out hermeneutic/interpretive approaches, but to offer a complementary workflow that keeps intermediate claims explicit and machine-checkable.

8. Practical extensions (brief pointers)

Two extensions are particularly central in practice:

- **Hierarchical labels + zooming** (summarise without losing the ability to drill back to specific sublabels).
- **Soft recoding (magnetisation)** (standardise messy label vocabularies at scale).

Both are treated as explicit, auditable label transforms (they rewrite labels, not the underlying evidence):

- see Magnetisation
- and the “Extensions” section in Minimalist coding for causal mapping

8.1 Example: magnetisation as a label transform (not a re-interpretation)

Suppose your raw in-vivo factor labels include:

- `No money for transport`

- `Bus fare too high`
- `Transport costs`

Magnetisation can map these to a shared magnetic label such as `Transport cost barrier`, letting you aggregate evidence *without deleting the original wording*. The original link evidence remains traceable; you are rewriting labels for a particular view.

9. Limits and caveats (non-negotiable)

- **Claims ≠ facts**: a coded link is evidence that a source claims X influenced Y; it does not itself establish causal truth.
- **Frequency ≠ effect size**: citation/source counts measure evidence volume/breadth in the corpus, not causal magnitude in the world.
- **Transitivity is a payoff and a trap**: reasoning over paths requires explicit context handling (e.g. thread tracing within-source).
- **Causal QDA does not answer every research question**: some valuable meaning in texts is non-causal (identity work, norms, emotions, metaphors). Causal coding captures what is represented as making a difference and being affected; it is not the whole of qualitative inquiry.

9.1 Example: the transitivity trap (why "path tracing" needs constraints)

Source A says:

- `Training -> Knowledge`

Source B says:

- `Knowledge -> Adoption`

It is tempting to infer a "path" `Training -> Adoption`. But unless you impose (and report) constraints — e.g. thread tracing within-source, or only treating same-source chains as evidence for an indirect mechanism — you risk stitching together a mechanism that **no one actually claimed**.

10. Conclusion

If you want a QDA method that is:

- grounded in quotes and provenance,
- oriented to causal questions,
- able to produce a structured qualitative model that is queryable,
- and compatible with transparent, bounded use of AI,

then causal mapping can be understood as **causal QDA**: a serious, checkable member of the QDA family, and a pragmatic bridge between rich narrative evidence and reproducible analysis pipelines.

References

Friese (2025). *Conversational Analysis with AI - CA to the Power of AI: Rethinking Coding in Qualitative Analysis*. https://doi.org/10.2139/ssrn.5232579.

{wickham (2011). *The Split-Apply-Combine Strategy for Data Analysis*. https://doi.org/10.18637/jss.v040.i01.