Working Papers

CONTENTS

Working Papers

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).

Minimalist coding for causal mapping

Abstract

This paper explains our **Minimalist / Barefoot** approach to coding causal claims in text as simple directed links (“X influenced Y”), developed through extensive experience with large-scale practical coding. We write it now because, although our previous work motivates causal mapping in evaluation (Powell et al. 2024) shows how QuIP-style “stories of change” elicit natively causal narrative evidence (Copestake et al. 2019), demonstrates ToC validation by comparing empirical maps with programme theory (Powell et al. 2023), and shows that generative AI can extract links exhaustively with quotes as a low-level assistant (Powell & Cabral 2025; Powell et al. 2025), none of these papers is a standalone, reader-facing account of **the coding stance itself**. Our approach is notable in particular because it rejects variable-based approaches used by most causal mapping traditions, which share many assumptions with rules used to build Systems Diagrams, Fuzzy Cognitive Maps, Causal Loop Diagrams, etc. We give an alternative, more primitive account of what exactly counts as a coded causal claim. We explain what we deliberately do *not* encode such as strength and polarity, and we mention the limits of this approach. **Intended audience:** evaluators, academics and qualitative researchers who want a citeable and teachable causal coding protocol, and AI/NLP readers who want a simple, auditable way to identify and process causal content in text.

Introduction: Why minimalist coding?

The variable-based approach

1. I was eating less and felt quite lethargic
2. I started to eat more and was feeling more lively

To the best of our knowledge, all major approaches to causal mapping (Axelrod (1976), Eden et al. (1992), Laukkanen & Wang (2016), Maule et al. (2003)) would most like code (1) as `amount eaten` --> `energy level`. And they would treat (2) pretty much the same.

Nomenclature: we will call all cause and/or effect labels "factor labels", whether in the variable-based or minimalist approaches.

To make this work, we have to believe there are at least two things called variables which can take different values (higher/lower?) and that causal effects are powered by at some kind of (monotonic?) relationship between them: something like, *the more* I eat, *the more* energy I have (in addition to which we have to code the actual, factual claims.

But is that what the speaker *meant*? How do we know if the speaker has say a continuous or Boolean (yes/no) model of "eating"? If Boolean, what is the opposite of eating a lot? Eating only a little? Not eating? If continuous, how do we know what kind of function they use in their own internal model?

In other words, to use the variable-based approach, we mostly have to go way beyond what the speaker actually meant; so it is not really a form of qualitative data analysis with the aim of modelling sources' beliefs. It's not just coding, it's modelling, with an ambiguity about whether we are trying to model the world or individuals' putative internal models of it.

(Parenthetically, to actually code what the source said, we would in addition we have to code two more facts: I actually ate less (or more), and: Also, I actually did feel lethargic (or lively)).

The problem with trying to apply these kinds of sophisticated frameworks is that the application is nearly always *ontologically under-determined*.

We have found that it is often over-specified (and often psychologically implausible) to treat ordinary-language causal claims as if they asserted an explicit functional relationship between well-defined variables. Trying to force that kind of structure on everything turns the “easy 90%” of coding into a harder and more arbitrary task. Of course, one can decide to use a particular non-minimalist representation for a particular modelling purpose; our claim is only that this is usually not a faithful representation of what most speakers actually say or imply, most of the time. For example, if we code "I got really tired because I have Long Covid", we could perhaps treat both endpoints as Boolean variables, but what about "I got really tired because it was really hot", and "I got really tired because it was really cold" -- how are we going to represent “temperature” as a single variable while preserving the speaker’s intended meaning? If what we want to do is model a system, we can pick a solution. But if we want to model *cognition as expressed in text*, many “variable semantics” choices are over-committed.

Most causal mappers are quite conscious that modelling people's causal beliefs is not the same as modelling the real world. Sometimes the approach (Axelrod 1976) is explicitly designed to model someone's thinking. But often it is hard to keep the two worlds separate, what we have called the Janus problem (Powell et al. 2024). Modelling the real world is exciting, and useful. Traditions like Systems Diagrams, Fuzzy Cognitive Maps, Causal Loop Diagrams and Directed Acyclic Graphs all to some extent treat

Using variable-based approaches allows one to code linear or even non-linear causal influences of single or even multiple causes on their effects. One can do the "coding" by simply writing down (using appropriate special syntax) the connections, because one is an expert, and/or one can verify such statements statistically on the basis of observational data. Thus armed, one can make predictions or have sophisticated arguments about counterfactuals.

The minimalist approach

But using minimalist coding we cannot do that, because our claims are formally weaker and therefore our inference rules are weaker. What we *can* do is still really interesting. We can ask and answer many different kinds of useful questions like:

- what are the main influences on (or effects of) a particular factor, according to the sources?
- what are the upstream, indirect influences on (or effects of) a particular factor, bearing in mind The transitivity trap?
- how well is a given programme theory validated by the respondents' narratives? (We can do this basically by using embeddings to get measures of semantic similarity between labels and aggregate these as a goodness of fit of theory to data.)

That is all exciting and useful. It's a surprisingly simple way to make a lot of sense out of a lot of texts which is, with caveats, almost completely automatable.

I'll start by describing the "minimalist" approach to coding causal statements used for QuIP and developed originally by James, Fiona and colleagues at Bath SDR and developed and further formalised at Causal Map Ltd in collaboration with Bath SDR. This formalisation lives inside the Causal Map app. Then we will show how we can extend this approach with useful transformations. Finally I will try to answer the question of whether it can help us deal with more complicated constructions like enabling and blocking and whether this could help us with mid-range theory. As an appendix I'll add a more detailed overview of minimalist causal coding.

The minimalist approach is notable because it is based in our **joint experience of coding thousands and thousands of stakeholder interviews and other data such as project reports**, mostly from international development and related sectors, as well as coding hundreds of thousands of pages with AI-assisted coding. These have nearly always involved **multiple sources talking about at least partially overlapping subject matter**. So this coding produces individual causal maps for each source, which can then be combined in various ways -- rather than constructing single-source maps of expert thinking (Axelrod 1976) or the collective construction of a consensus map (Barbrook-Johnson & Penn 2022).

As before, we treat causal mapping as three tasks

- Task 1, data gathering, not covered here;
- Task 2, coding: creating **evidence-with-provenance** (a links table)
- Task 3, analysis: a sequence of transforms of the original links table: the final interpretation of the analysis can be understood as a concatenation of the interpretation rules for each transform.

Project context (how this paper fits)

A companion formalisation paper A formalisation of causal mapping makes key parts of the method more precise (data structures, constraints, and conservative inference rules), but the focus here is the narrative rationale and practical coding guidance.

Unique contribution (what this paper adds):

- A definition of the **minimalist/barefoot** coding stance ("X influenced Y, with provenance") and what it deliberately does *not* encode.
- An account of why this stance is useful at scale (including AI-assisted extraction) and where it fails (enablers/blockers; conjunctions/packages).
- Examples of additional extensions / transforms, called "filters" in the Causal Map app. Both the initial definition of a links table and each additional filter is described in terms of syntax and of semantics (interpretation rules).

This paper is part of a small set of new working papers

- Companion formal spec: A formalisation of causal mapping
- QDA positioning: Causal mapping as causal QDA
- Practical transforms/diagnostics: Magnetisation; A simple measure of the goodness of fit of a causal theory to a text corpus; ; other transforms to come later
- A worked "AI clerk vs human architect" example:

This paper sits alongside (and builds on) four related contributions. First, our evaluator-facing account argues that causal mapping is best treated primarily as a way to assemble and organise **evidence-with-provenance**, keeping the subsequent evaluative judgement about "what is really happening" distinct (Powell et al. 2024). We adopt that stance here: a coded link is first and foremost "there is evidence that a source claims X influenced Y", not a system model with weights or effect sizes. Second, QuIP-style evaluation practice shows how "stories of change" can be elicited in a goal-free / blindfolded style to reduce confirmation bias, yielding narrative data that is natively causal (change plus reasons) and therefore well-suited to parsimonious link coding (Copestake et al. 2019). Third, our ToC comparison case study shows how empirical causal maps can be used as a disciplined way to check a programme's Theory of Change against beneficiaries' narratives, and to support evidence-based adjustment of "middle-level theory" rather than just project-level reporting (Powell et al. 2023). Fourth, our AI-assisted causal mapping work shows that this minimalist stance is also a practical entry point for automation: we can use genAI as a low-level assistant -- because the minimalist coding task is so relatively easy -- while keeping human judgement focused on the few high-leverage decisions (prompt design, verification, and synthesis choices) (Powell & Cabral 2025; Powell et al. 2025). The present paper extracts and clarifies the core "minimalist" coding commitments that make that workflow workable and checkable.

Our experience has been that the vast majority of causal claims in these kinds of texts are easily and satisfactorily coded in the simplest possible form "X causally influenced Y". Explicit invocation of concepts like enabling/blocking, or necessary and/or sufficient conditions, or linear or even non-linear functions, or packages of causes, or even the strength of a link, are relatively rare. The causes and effects are not conceived of as variables, the causal link is undifferentiated, without even polarity, and if any counterfactual is implied it remains very unclear.

This approach is what we call **"Minimalist" or "Barefoot" Coding**.

Minimalist coding principles

The 90% rule

We have found that it is pretty easy to agree how to apply minimalist coding to say 90% of explicit causal claims in texts, without missing out essential causal information, whereas it is very difficult to find appropriate frameworks to cope with the remaining 10%.

Fewest assumptions

Minimalist coding is perhaps the most primitive possible form of causal coding which makes no assumptions about the ontological form of the causal factors involved (the "causes" and "effects") or about *how* causes influence effects. In particular we do not have to decide if cause and/or effect is perhaps Boolean or ordinal, or if perhaps multiple causes belong in some kind of package or if there is some kind of specific functional relationship between causes and effects.

An act of causal coding is simply adding a link to a database or list of links: a link consists of **one new or reused cause label and one new or reused effect label**, together with the highlighted quote and the ID of the source.

A statement (S) from source Steve:

> I drank a lot and so I got super happy

can be trivially coded minimalist-style as

> I drank a lot --> I got super happy (Source ID: Steve; Quote: I drank a lot and so I got super happy)

That's it.

Causal maps

Crucially, we can then display the coded claims for individuals as a graphical causal map, and we can also display the entire map for all individuals and/or maps filtered in different ways to answer different questions. There is a handful of other applications (Ackermann et al. 1996) (Laukkanen 2012) for causal mapping which also do this; but as far as we know, only Causal Map also allows direct QDA-style causal coding of texts.

Data structure

Although we have the option of creating additional tags associated with each link (where many approaches would for example code the polarity of a link) this is not central to our approach.

We don't use a separate native table for factor labels: they are simply derived on the fly from whatever labels happen to be used in the current table of links. This makes data processing simpler and also suggests an ontological stance: causal factors only exist in virtue of being part of causal claims.

We do however have an additional table for source metadata including the IDs of sources, which can be joined to the links table in order, for example, to be able to say "show me all the claims made by women".

Causal powers

When a source uses causal language, we treat it as a claim that one thing made a difference to another (rather than a mere temporal sequence) *in virtue of its causal power to do so*.

Causal influence, not determination

We believe that it's rare for people to make claims about causal determination: someone can say that the heavy drinking made them super happy and then also agree that the music had a lot to do with it too, without this feeling like a contradiction.

Not even polarity

We differ even from most other approaches which are explicitly called "causal mapping" in that we do not even "out of the box" record polarity of links (to do so would involve making assumptions about the nature of the "variables" at each end of the link as well the function from one to the other).

The Focus on Cognition

In the minimalist approach, **we are quite clear that what we are trying to code is the speaker's surface cognitions and causal thinking**, while the actual reality of the things themselves is simply bracketed off at this stage, either to be revisited later (because we are indeed interested in the facts beyond the claims) or not (because we are anyway interested in the cognitions).

Staying on the surface

At Causal Map, we rarely make any effort to get beneath the surface, to try to infer hidden or implicit meanings. This is particularly well-suited to coding at scale and/or with AI. Our colleagues at Bath SDR do this a bit differently, spending more effort to read across an entire source to work out what the source *really* meant to say.

Closer to the cognitive truth

see above

Unclear counterfactuals

See above

General versus specific

Minimalist coding focuses primarily on **factual causal claims** which also warrant the inference that both X and Y actually happened / were the case.

Most causal claims in the kinds of texts we have dealt with (interviews and published or internal reports in international development and some other sectors) are factual, about the present or past. Sometimes we see general claims, and we often just code these willy-nilly. In any case, the distinction between general claims and claims about specific events that actually happen is often fractal (a general claim can seem specific in a broader context) and difficult to maintain completely when modelling ordinary language.

We don't code absences

Minimalist coding may be reasonably also called **Qualitative Causal Coding** or **Causal QDA coding**. It shares characteristics with some forms of coding within Qualitative Data Analysis (QDA), in particular demonstrating an asymmetry between presence and absence (a specific tag not being applied is not the same as the application of an "oppositely-poled" tag.

We do not code absences unless they are specified within the text (e.g. perhaps "because of the barking dog, the owner did not come out of the house".

While codes may be counted, the concept of a *proportion* of codes is challenging because the denominator is often unclear.

If families are talking about reasons for family disputes, and family F mentions social media use, and family G mentions homework, we do not usually interpret this as meaning that family F does *not* think that homework can also be a cause of family disputes. (This should derive formally from the interpretation of multiple causal claims).

The labels do all the work

At Causal Map Ltd, our canonical methodology initially involves in vivo coding, using the actual words in the text as factor labels. This initial process generates hundreds of overlapping factor labels. This part is really easy (and is easy to automate with AI). Obviously, hundreds (or hundreds of thousands) of overlapping factor labels are not very useful, so we need to somehow consolidate them. Arguably, minimalist coding makes the initial coding easy but it just defers some of the challenges to the recoding phase.

One way to code additional meaning within factor labels is to add **tags as literal text inside the label**.

Tags can be used to group factors into **themes** which are not themselves causal factors (e.g. health, environment), without making them part of a hierarchy. They can add **lightweight metadata** for later filtering (e.g. `(Outcome)`, `#nutrition`, `[Work]`), while keeping the underlying causal claim as just `X -> Y`.

Suitable filters can then be provided to:

- **Filter by theme/flag**: because tags are just text, any “match start/anywhere/exact” label filter can include or exclude factors containing `#health`, `(Outcome)`, etc.
- **Hide tags for display**: to strip tags out for a cleaner map/table view, while retaining them in the underlying data.

Tagging factor labels is different from adding link metadata because it literally creates different causal factors. So `Improved health` and `Improved health [outcome]` are formally two different causal factors. However if we apply a filter to strip the `[outcome]` tag, they then become the same causal factor.

For a fuller discussion (including examples like `#nutrition` and `(Outcome)` and why uniqueness matters), see: Factor label tags — coding factor metadata within its label.

Coping with many labels

We can:

- Use human- or AI-powered clustering techniques to consolidate the codes according to some theory or iteratively according to some developing theory
- Use AI-powered clustering techniques to consolidate the codes according to automated, numerical encoding of their meanings
- "Hard-recode" the entire dataset using a newly agreed codebook (see above)
- "Soft-recode" the dataset on the fly using embeddings to recode raw labels into those codebook labels to which they are most similar

Evidence strength is not causal effect size

Counts (how many times a link is coded; how many sources mention it) measure *evidence volume/breadth in the corpus*, not causal magnitude in the world. This matters because the outputs can look like a quantitative system model even when they are not one.

In the app we routinely distinguish:

- **Citation count**: how many coded mentions support a link (volume).
- **Source count**: how many distinct sources mention it (breadth / rough consensus).

These are useful for prioritising what to look at, or for filtering (e.g. keeping only links with `min_source_count = 2`), but they do not tell you whether a causal influence is “stronger” in any effect-size sense.

The transitivity trap and “thread tracing”

It is usually invalid to infer a long causal chain by stitching together links from different sources. A conservative rule is: only treat an indirect pathway (A \rightarrow B \rightarrow C) as supported when the *same source* provides each step (“thread tracing”), unless contexts are explicitly aligned.

The canonical failure mode is: source 1 says `A -> B`, source 2 says `B -> C`, and we mistakenly conclude that anyone told a coherent story `A -> B -> C`. In practice we handle this by “thread tracing”: for a given query we loop through sources one at a time, construct the valid paths *within each source*, and then combine only the edges that appear in valid within-source paths.

The filter pipeline as a mental model for analysis

Most analyses are built by applying a sequence of simple operations (for example: subset links to retain only specific sources; trace paths; rewrite labels; bundle duplicates; then show a map/table). Thinking in terms of a pipeline helps make the meaning of an output explicit: it is always “the result of these filters, in this order”.

Concretely, you can think of an analysis as “a links table passed through transforms”, where `|>` is a "pipe" symbol which just passes a result from the left to a new transform on the right.

e.g.:

```
Links |> filter_sources(...) |> trace_paths(...) |> transform_labels(zoom) |>
combine_opposites |> bundle_links |> map_view
```

This is also why “the same dataset” can yield very different-looking maps without any contradiction: you are simply looking at different derived views defined by different pipelines.

So one can think of analysis as a sequence of transforms of the original links table. The final interpretation of the analysis can be understood as a concatenation of the interpretation rules for each transform.

Positioning within qualitative research (and likely critiques)

This paper is about a **coding stance** and a corresponding **intermediate representation** (a links table with provenance), not a claim that "coding = analysis". In standard QDA terms, minimalist causal coding is best understood as a disciplined way to build an auditable evidence base that can later be interpreted, queried, and written up (Miles, Huberman, & Saldaña, 2014; Saldaña, 2021). It is, of course, not the only kind of way to do QDA.

To reduce avoidable points of attack from mainstream qualitative social science readers, we make four explicit commitments up front:

1. **We code claims, not causal truth.** A coded link is evidence that a source *claimed* an influence relation. It is not (by itself) a causal inference claim about the world. This keeps the method compatible with both realist and constructivist sensibilities (Lincoln & Guba, 1985; Charmaz, 2014).
2. **We do not treat "counts" as effect sizes.** Counting supports prioritisation and transparency, but it is not a substitute for interpretation; frequency/breadth in a corpus is not magnitude in the world. (This paper makes that distinction explicit below.)
3. **We trade interpretive depth for auditability and scale, on purpose.** We stay close to surface causal language and preserve provenance (quotes + source ids) so that readers can check what is being claimed. This is a pragmatic stance when working with many sources and/or AI assistance; it does not deny that richer interpretive work can be valuable.
4. **Minimalist coding is not a full QDA workflow.** It is one layer that can sit alongside thematic analysis or qualitative content analysis when those are needed (Braun & Clarke, 2006; Mayring, 2000).

How a qualitative-methods reviewer might criticise this (and our response)

Below are common criticisms (often reasonable) and the corresponding guardrails/defences built into the minimalist stance.

Critique 1: "This is reductionist / strips context / decontextualises quotes"

Response: We keep provenance as first-class data (source id + quote) and treat every map/table view as a derived view of that evidence. Context is handled explicitly by joining source metadata (e.g. subgroup, time) and filtering the links table; it is not "assumed away". When deeper within-case interpretation matters, the same links table can be re-read within full-text context, and the write-up remains responsible for integrating that context (Miles, Huberman, & Saldaña, 2014).

Critique 2: "You are smuggling in a positivist epistemology (or a realist metaphysics)"

Response: The method's core object is **reported causal thinking** (surface causal claims), not an ontic causal model. We therefore keep the epistemic claim modest: "this is what sources said, with quotes", and we separate that from subsequent judgement about what is happening. That separation is consistent with standard qualitative criteria emphasising transparency, audit trails, and reflexive interpretation (Lincoln & Guba, 1985; Charmaz, 2014).

Critique 3: "Coding is not analysis; links don't 'explain' anything"

Response: Agreed. Minimalist causal coding produces a structured evidence base that supports multiple analyses (filters, comparisons, pathway queries) and supports more disciplined writing, but it does not replace interpretation. In practice, it can reduce time spent on low-level organisation of text so that more time can go into interpretation and argument (Saldaña, 2021; Gläser & Laudel, 2013).

Critique 4: “You are privileging causal language and missing other kinds of meaning”

Response: Yes: by design. Minimalist coding is for projects where the core questions are themselves causal (mechanisms, pathways, theories of change). When non-causal meaning is central (identity work, norms, metaphors), other QDA approaches could lead; minimalist coding can still be a useful *additional* layer rather than the whole analysis (Braun & Clarke, 2006).

Critique 5: “Automatable coding invites false precision and overclaiming”

Response: We explicitly constrain what automation is allowed to do: extract candidate links with quotes; keep everything auditable; and avoid treating the resulting network as a system model with polarity/weights “by default”. The transparency of provenance is the main defence against black-box “AI did the analysis” claims.

Extensions

There are many extensions one can add on top of minimalist coding. Small extensions can just add convenient functionality; other extensions can upgrade the system to other more fully-fledged kinds of coding like FCM.

An extension in general consists of syntax rules for how to construct and carry out a transformation, and semantic or interpretation rules for what this transformation means.

This paper will focus on only one extension: **hierarchical coding**, because it is simple, widely useful in practice, and it directly supports a transparent family of “zoom” transforms.

Some useful extensions

Hierarchical coding and “zooming” (FIL-ZOOM)

Motivation

In any qualitative coding process there is a tension between **detail** (keeping what respondents actually said) and **summarisation** (making patterns visible). Hierarchical factor labels give us a simple way to keep both:

- We can keep a detailed factor such as `Healthy behaviour; hand washing`.
- We can also treat it as an instance of a higher-level causal factor `Healthy behaviour` when we want a higher-level view.

This is particularly useful in causal mapping because it lets us simplify a complex map *without changing the underlying link evidence*: we are simply rewriting labels into more general ones.

Coding conventions

We use the separator `;` to create nested factor labels, using a template like:

`General concept; specific concept`

For example:

`New intervention; midwife training -> Healthy behaviour; hand washing`

This is an example of what we called earlier "letting the labels do the work". As we do not in any case have a native factors table where we might record actual relationships between say lower-level and higher-level factors, we can do the same thing implicitly simply with the text of the label.

In AI-assisted coding, a practical way to encourage hierarchical labels is to explicitly instruct the coder (human or AI) to label each factor using the template **"general concept; specific concept"**, and to always provide a verbatim quote for each coded claim so every link remains checkable.

For example, an AI might produce a single coded link such as:

- `Improved agricultural practices; diversified crops -> Increased icome generation; more sales`
- Quote: "with a lot of good product, we are now able to sell more."

You can read the separator as:

- "A, and in particular B" (forward), or
- "B, which is an example of / part of A" (reverse).

This can be extended to multiple levels:

`Improved hospital skills training; new midwife training; hand washing instructions improved`

Two practical conveniences follow immediately:

- Searching for the higher-level label (e.g. "Healthy behaviour") will also find its nested sublabels.
- Higher-level labels can be created and changed on the fly (they are just the visible prefixes before `;`), without maintaining a separate "parent code" structure.

Zooming and visualisation

The basic idea of a zoom view is: **rewrite nested labels to a chosen level of abstraction**, then draw the map using the rewritten labels.

At "zoom level 1" we simply truncate at the first `;`:

- `Healthy behaviour; hand washing` becomes `Healthy behaviour`.

So if we have (at the detailed level):

`Improved hospital skills training; improved midwife training; hand washing instructions improved`

then we can show higher-level summary links such as:

- `Improved hospital skills training; improved midwife training`
- and (at the coarsest level) `Improved hospital skills training`

Caveats (don't use the `;` separator mechanically)

Intuitively: by writing `A; B` you are explicitly licensing the interpretation "for high-level summary purposes, treat this as A".

You can also see this as a family of conservative "deductions" licensed by the `;` separator. We are explicitly allowing ourselves (for summary purposes) to treat the more detailed version also as evidence for the two less detailed versions: `Improved hospital skills training; improved midwife training; hand washing instructions improved` is also evidence for `Improved hospital skills training`.

The `;` separator encodes a commitment: that the more specific factor can be "rolled up" into the more general one without essentially distorting the causal story.

Other tips

Try not to mix desirability within one hierarchy (e.g. avoid `Stakeholder capacity; lack of skills`, because zooming out would treat it as evidence for "Stakeholder capacity"). Use opposites coding (`~...`) for this kind of case.

A factor cannot belong to **two different** hierarchies at once (because there is no single parent to roll up into).

When *not* to use a hierarchy (use a tag instead)

Don't use `;` to express a non-causal *topic theme*. `A; B` explicitly licenses: "for high-level summary purposes, treat this as `A`", so `A` must be a real (semi-quantitative) causal factor, not just a theme label like "Health".

If something is merely health-related (with no natural roll-up into a single causal factor), use a tag instead:

- `Vaccinations law is passed [Health]`
- `Mortality rate [Health]`

Only use a hierarchy when the right-to-left label is genuinely a refinement of the left-to-right one (so roll-up makes sense), e.g.:

- `Improved health training; Improved hospital skills training`

Other extensions (stubs; treated in subsequent articles)

See A formalisation of causal mapping and

Context filters (FIL-CTX)

Restrict the evidence base by selecting sources (e.g. only women; only a time period; only a subgroup), then carry out the same downstream analysis on the restricted links table.

Frequency / evidence-threshold filters (FIL-FREQUENCY)

Retain only links meeting a threshold of evidence strength (e.g. `min_source_count=2`). This is about *evidence volume/breadth in the corpus*, not effect size.

Bundling (FIL-BUNDLE)

Compute evidence-strength columns for each `Cause -> Effect` pair (e.g. `Citation_Count`, `Source_Count`) and use those in map/table views.

Opposites coding (FIL-OPP) (stub)

Rather than encoding signed/weighted edges, we can get similar benefits by treating explicit opposites in labels (e.g. `~Employment` vs `Employment`) as a label-level device with simple inference and visualisation rules.

See

AI extensions (optional; stubs)

Soft recoding (magnets) (FIL-SOFT)

Many raw in-vivo labels can be aggregated by mapping them to a smaller vocabulary of "magnets" using semantic similarity. This changes labels (a recoding step), not the underlying minimalist meaning of a single coded link.

Auto recoding (clustering) (FIL-AUTO)

Alternatively, labels can be clustered into emergent groups and then rewritten accordingly, again as a recoding/aggregation step rather than a new causal logic.

How other schools of causal mapping extend minimalist links

The “minimalist” stance in this paper is deliberately radical: we code *undifferentiated* causal influence claims with provenance, and then do most interpretation work via filters and views. Many other causal mapping traditions extend the representation in the opposite direction: they add more **semantic structure to factors and links**, especially **polarity** (and sometimes strength/weights), often with an implicit assumption that factors behave like **variables** and that a link expresses some kind of monotonic relationship. For a broader overview of these traditions and their differences, see (Powell et al. 2024).

What these extensions usually add

- **Polarity (+/−)**: a link can mean “more of X leads to more of Y” (positive) or “more of X leads to less of Y” (negative).
- **Strength / weights**: a link can be assigned a magnitude (sometimes elicited, sometimes assigned by analysts), which invites quantitative reasoning and simulation-like interpretations.
- **Variable semantics and counterfactual clarity**: polarity and weights usually presuppose that the endpoints are comparable as variables (or at least as ordered states), so that “more/less” (or “increase/decrease”) is meaningful.

These extensions can be extremely useful in settings where respondents are explicitly reasoning in those terms, or where the purpose is modelling/simulation. But they also raise the bar: if we write down a signed or weighted link, we are committing not just to “X influenced Y”, but to a stronger claim about *how* X and Y vary and how changes propagate.

Axelrod-style cognitive mapping (signed causal beliefs)

In the Axelrod tradition, cognitive maps are often treated as representations of beliefs about causal influence among concepts, and they are frequently coded with some notion of **positive/negative influence** in addition to direction (Axelrod 1976; Axelrod 1976). This makes it easier to talk about reinforcing/balancing structure and about the direction of change, but it also moves the representation closer to variables-with-values (even if the original text/elicitation was not precise about scales or functional form).

Eden & Ackermann cause mapping / problem structuring

The Eden/Ackermann “cause maps” tradition (including its software lineage) emphasises causal mapping as a practical tool for structuring messy problems and supporting decision making, often built interactively with respondents and iterated in workshops (Eden et al. 1992; Ackermann et al. 2004; Ackermann et al. 1996). Polarity and more explicit causal typing are more natural here because the map is typically negotiated in context (so the group can decide what is meant by “increase/decrease”, and can revise labels until the signed links make sense).

Comparative causal mapping (standardisation and cross-map comparison)

Comparative approaches focus on how to elicit, standardise, and compare large numbers of maps across people, groups, or time. This tends to bring in more explicit conventions for coding and comparison, and often assumes a more “variable-like” interpretation of factors so that maps can be aligned and analysed at scale (Laukkanen 1994; Laukkanen 2012; Markiczy & Goldberg 1995; Hodgkinson et al. 2004).

What it would take to extend our logic to signed links (and why we treat it separately)

We *can* extend our links-table logic to incorporate polarity, but doing so is not just “adding a column”; it changes the semantics.

At minimum, we would need:

- A `Polarity` field per coded link (e.g. `+`, `-`, `unknown`) that is grounded in the source text or elicitation.
- A rule for how polarity is inferred when it is implicit, ambiguous, or mixed (which is common in narrative material).
- A clear semantics for what `+` and `-` *mean* (typically: a monotonic relationship between variables), including what counts as “more/less” for a factor label.
- Aggregation rules for how to deal with disagreement and contradiction across sources, and for how to visualise those disagreements without creating spurious precision.

Because our target corpora typically do *not* make those commitments explicit (and because we are focused on modelling evidence-with-provenance rather than system dynamics), we do not treat signed links as part of the core method here. Where polarity matters in practice, we prefer to handle it with **label-level devices** (e.g. `~Employment` vs `Employment` combined with the `combine_opposites` transformation.

What we cannot do (this is work in progress, not sure what to do with it yet)

Causation about causation: enablers and blockers as second-order causal claims

One natural way to try to go beyond truly minimalist coding is to say that some claims are not about simple factors causing other simple factors, but about one factor influencing (enabling or blocking) a *causal connection* between two other factors.

Consider an enabler-style statement:

> I went on holiday to Spain expecting to enjoy it, and indeed I did particularly enjoy it because I remembered to take my phrase book.

In strictly minimalist coding, we might record two undifferentiated links:

- Going on holiday to Spain --> Enjoying the holiday
- Remembering to take my phrase book --> Enjoying the holiday

But arguably the second claim is not really “the phrase book caused enjoyment” in the same way as the first. Rather, it enabled the normal causal power of the holiday to produce enjoyment. A more explicit encoding would therefore be:

1. Going on holiday to Spain --> Enjoying the holiday
2. Remembering to take my phrase book --> (Going on holiday to Spain --> Enjoying the holiday)

Visually, (2) would be drawn as an arrow pointing to the middle of the arrow in (1). Semantically, it can be read as:

> The phrase book enabled the causal link from X to Y, in virtue of its causal power to do so.

Blockers are closely related but show the asymmetry more starkly:

> I went on holiday to Spain, but I forgot to take my phrase book so I didn't enjoy the holiday at all.

One can code:

- Forgetting to take my phrase book --> Not enjoying the holiday

However, our intuition that “going on holiday” should also be part of the story is hard to capture without moving to a second-order encoding. The blocker case is visually similar to the enabler case, but it seems to require adding the idea of a causal power to *stop* something: the blocker destroys or disables the causal power of X to bring about Y.

I do not claim we can do much with this at scale; but it is a clear way of stating what seems to be “missing” from purely first-order, undifferentiated links in these edge cases.

Causal packages / conjunctions

Another class of difficult cases is causal packages: claims in which some *combination* of factors is said to be needed for an effect.

For example, “you need an accelerant and a spark to set a fire” is not well represented by coding two separate links (“accelerant causes fire” and “spark causes fire”), because that misses the conjunctive structure. One can code the cause as a single phrase (e.g. “an accelerant and a spark”), but then the phrase is not parsable in a way that lets us relate it to other claims about accelerants or sparks on their own.

In principle one could introduce special syntax for conjunctions, but the moment we do so we are immediately pushed towards stronger, more model-like commitments (e.g. about truth tables, interaction effects, non-linear combination rules), and it is unclear how much such structures would recur with enough regularity in ordinary language corpora to justify that added complexity.

Blockers and enablers

Maybe we could ascend from formally weaker but numerically overwhelming minimalist-coded data to make other rich conclusions, in particular about enablers and blockers like the headphones and the rain. However, I don't think this is really possible. In minimalist coding, at the level of individual claims, you can code "The headphones enabled James to answer the question in the Zoom call" as

> The headphones --> James was able to answer the question in the Zoom call

... but we cannot easily get inside the *contents* of the effect. We might like to be able to code this as the effect of the headphones not on a simple causal factor but on *another causal connection*, namely between the question on the Zoom call and James' answer, but we do not have any way at the moment to do this. It might be possible to extend minimalist coding to cope with this, perhaps ending up with three factors (headphones, question, answer) and some new syntactic rules to code their relationship, and some corresponding new semantic rules to be able to deduce more things about these three factors, but I think this would be **missing the point**. I'm not sure what we could do with these kinds of subtle relationships at any scale. Let's guess that within a given corpus, five percent of causal claims are of this form: what are the chances of such claims then overlapping enough in content that we could then apply our new more specialised deduction rules in more than a handful of cases?

It might be the case that certain specific more sophisticated causal constructions become **part of ordinary language**. For example: "Her post mocking Farage went viral, so Farage was forced to respond". Here, the concept of *going viral* is perhaps a kind of shorthand for a quite sophisticated causal claim, yet it might be common enough for us to be able to usefully code it (and reason with it) using only unadulterated minimalist coding, without causally unpacking "her post went viral". So that's useful, and maybe it is even useful in building some kinds of mid-range theory, but without actually understanding or unpacking what "going viral" means.

See also:

(Powell et al. 2024)

(Powell & Cabral 2025)

(Britt et al. 2025)

(Powell et al. 2025)

(Remnant et al. 2025)

Selected references (APA 7; added for qualitative-methods positioning)

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology, 3*(2), 77–101. https://doi.org/10.1191/1478088706qp063oa

Charmaz, K. (2014). *Constructing grounded theory* (2nd ed.). SAGE.

Gläser, J., & Laudel, G. (2013). Life with and without coding: Two methods for early-stage data analysis in qualitative research aiming at causal explanations. *Forum Qualitative Sozialforschung / Forum: Qualitative Social Research, 14*(2). https://www.qualitative-research.net/index.php/fqs/article/view/1886

Gläser, J., & Laudel, G. (2019). The discovery of causal mechanisms: Extractive qualitative content analysis as a tool for process tracing. *Forum Qualitative Sozialforschung / Forum: Qualitative Social Research, 20*(3). https://doi.org/10.17169/fqs-20.3.3386

Lincoln, Y. S., & Guba, E. G. (1985). *Naturalistic inquiry*. SAGE.

Mayring, P. (2000). Qualitative content analysis. *Forum Qualitative Sozialforschung / Forum: Qualitative Social Research, 1*(2), Art. 20. https://www.qualitative-research.net/index.php/fqs/article/view/1089

Miles, M. B., Huberman, A. M., & Saldaña, J. (2014). *Qualitative data analysis: A methods sourcebook* (3rd ed.). SAGE.

Saldaña, J. (2021). *The coding manual for qualitative researchers* (4th ed.). SAGE.

[Note: we will add at least the formal codes like combine_opposites from the formalism paper]

References

Ackermann, Jones, Sweeney, & Eden (1996). *Decision Explorer: User Guide*. https://banxia.com/pdf/de/DEGuide.pdf.

Ackermann, Eden, & Cropper (2004). *Getting Started with Cognitive Mapping*.

Axelrod (1976). *Structure of Decision: The Cognitive Maps of Political Elites*. Princeton university press.

Axelrod (1976). *The Analysis of Cognitive Maps*. In *Structure of Decision : The Cognitive Maps of Political Elites*.

Barbrook-Johnson, & Penn (2022). *Participatory Systems Mapping*. In *Systems Mapping: How to Build and Use Causal Models of Systems*. https://doi.org/10.1007/978-3-031-01919-7_5.

Britt, Powell, & Cabral (2025). *Strengthening Outcome Harvesting with AI-assisted Causal Mapping*. https://5a867cea-2d96-4383-acf1-7bc3d406cdeb.usrfiles.com/ugd/5a867c_ad000813c80747baa85c7bd5ffaf0442.pdf.

Copestake, Morsink, & Remnant (2019). *Attributing Development Impact: The Qualitative Impact Protocol Case Book*. March 21, Online.

Eden, Ackermann, & Cropper (1992). *The Analysis of Cause Maps*. https://onlinelibrary.wiley.com/doi/10.1111/j.1467-6486.1992.tb00667.x.

Hodgkinson, Maule, & Bown (2004). *Causal Cognitive Mapping in the Organizational Strategy Field: A Comparison of Alternative Elicitation Procedures*.

Laukkanen (1994). *Comparative Cause Mapping of Organizational Cognitions*.

Laukkanen (2012). *Comparative Causal Mapping and CMAP3 Software in Qualitative Studies*. https://doi.org/10.17169/fqs-13.2.1846.

Laukkanen, & Wang (2016). *Comparative Causal Mapping: The CMAP3 Method*. Routledge.

Markiczy, & Goldberg (1995). *A Method for Eliciting and Comparing Causal Maps*. Sage Publications Sage CA: Thousand Oaks, CA.

Maule, Hodgkinson, & Bown (2003). *Cognitive Mapping of Causal Reasoning in Strategic Decision Making*. In *Thinking: Psychological Perspectives on Reasoning, Judgment and Decision Making*. https://doi.org/10.1002/047001332X.ch13.

Powell, Larquemin, Copestake, Remnant, & Avard (2023). *Does Our Theory Match Your Theory? Theories of Change and Causal Maps in Ghana*. In *Strategic Thinking, Design and the Theory of Change. A Framework for Designing Impactful and Transformational Social Interventions*.

Powell, Copestake, & Remnant (2024). *Causal Mapping for Evaluators*. https://doi.org/10.1177/13563890231196601.

Powell, & Cabral (2025). *AI-assisted Causal Mapping: A Validation Study*. Routledge. https://www.tandfonline.com/doi/abs/10.1080/13645579.2025.2591157.

Powell, Cabral, & Mishan (2025). *A Workflow for Collecting and Understanding Stories at Scale, Supported by Artificial Intelligence*. SAGE PublicationsSage UK: London, England. https://doi.org/10.1177/13563890251328640.

Remnant, Copestake, Powell, & Channon (2025). *Qualitative Causal Mapping in Evaluations*. In *Handbook of Health Services Evaluation: Theories, Methods and Innovative Practices*. https://doi.org/10.1007/978-3-031-87869-5_12.

A formalisation of causal mapping

17 Dec 2025

Abstract

Draft for an IJSRM submission. This paper proposes a lightweight grammar and logic for encoding, aggregating, and querying causal claims found in qualitative text data. The specification is grounded in a "Minimalist" (or "Barefoot") approach to causal coding: it prioritises capturing the explicit causal claims made by sources, without imposing complex theoretical frameworks that may not align with how people naturally speak. **Intended audience:** methodologically minded evaluators / qualitative researchers, and AI/NLP readers who want a precise target representation for "causal claims in text" (and a clear separation between evidence storage vs downstream interpretation).

Unique contribution (what this paper adds):

- A compact **data model** for causal evidence in text (`LinksTable`, optional `SourcesTable`, derived `Factors`) with an explicit evidence constraint (quotes as substrings of source text when sources are present).
- A set of **syntax + semantics rules** for what counts as a valid coded claim and what can (and cannot) be inferred when aggregating claims across sources.
- A "pipeline" view of analysis as **operations on a links table**, which is designed to stay compatible with minimalist coding and to avoid accidental drift into system-model semantics (e.g., polarity/weights-by-default).

What does one piece of causal coding *mean*?

- We have syntactic rules to say what is a valid piece of causal coding.
- We have syntactic rules to say how to combine them into causal maps.
- We have syntactic rules for operations on causal maps.

For each of these rules we need to also give semantic rules to say what it also means, or how it combines meanings of its component parts.

The basic coding rule is something like:

> Look for places where it is stated or claimed that one thing causes or influences another.

Causal mapping logic as rules about relationship between texts is particularly relevant in the age of LLMs.

Having a causal influence is a lot weaker and more easily than causally affecting. So a mitigating action can causally influence something but not make it happen.

Project context (companion papers)

- Narrative coding stance and limits: Minimalist coding for causal mapping
- QDA positioning: Causal mapping as causal QDA
- Practical transforms over a links table: Magnetisation; Combining opposites, sentiment; Hierarchical coding
- Coverage-style fit diagnostic: A simple measure of the goodness of fit of a causal theory to a text corpus

1. Data Structures

The foundation of the specification is the **Project Data** package, which strictly separates the causal claims from the source material.

Definition Rule DS-PROJECTDATA: Project Data

- **Definition:** `ProjectData = LinksTable + optional SourcesTable`
- **Note:** `SourcesTable` may be omitted (or empty). If it is present, the evidence constraint in DS-LINKS-SOURCES applies.

Syntax Rule DS-LINKS: The Basic Links Table

A list of causal connections where each row represents one atomic claim.

- **Structure:** A table containing at least the following columns:
- `Cause`: Text label for the driver (influence factor).
- `Effect`: Text label for the outcome (consequence factor).
- `Link_ID`: Unique identifier for bookkeeping.
- `Context`: Optional identifier for distinguishing contexts.
- `Source_ID`: Identifier for the origin of the claim (e.g., document ID, participant ID).
- **Extensions:** The table may contain additional columns (e.g., `Sentiment`, `Time_Period`) or tags.

Syntax Rule DS-SOURCES: The Sources Table (optional)

A registry of documents, interviews, or respondents.

- **Structure:** A table containing:
- `Source_ID`: Unique key.
- `Full_Text`: The complete text of the source document.
- `Metadata`: Optional custom columns representing attributes of the source (e.g., Gender, Region, Date).

Syntax Rule DS-LINKS-SOURCES: Evidence Constraint (if Sources table is present)

- **Additional column:** The Links table also contains:
- `Quote`: The specific text segment evidencing the claim.
- **Constraint:** If a Sources table is provided, then for every link `L` in the Links table:
- `L.Source_ID` MUST match a `Sources.Source_ID`, and
- `L.Quote` MUST be a substring of `Sources.Full_Text` for the matching `Source_ID`.

Coding is strictly evidence-based.

Definition Rule DS-FACTORS: The Factors Derivation

There is no independent table for Factors stored in the Project. Factors are derived entities.

- **Definition:** `Factors = unique values in LinksTable.Cause + LinksTable.Effect`
- **Note:** A "Factors Table" is a transient structure created only during analysis.

2. Coding and Interpretation

This section defines how text is translated into the data structures defined above.

Definition Rule COD-DEF: The Definition of Coding

Coding is the process of extracting links from source text into the Links table.

Interpretation Rule COD-ATOM: The Atomic Causal Claim

A single row in the Links table, `Link | Cause="A" | Effect="B" | Source_ID="S1"`, is interpreted as:

- *"Source S1 claims that A causally influenced B in virtue of its causal power to do so."*

Interpretation Rule COD-BARE: Bare Causation

The relationship `A -> B` implies **influence**, not determination.

- **Negative Definition:** It does NOT imply `A` is the only cause of `B`.
- **Negative Definition:** It does NOT imply `A` is sufficient for `B`.

Example:

- **Source Text:** "Because of the drought, we had to sell our livestock." (from Source 12)
- **Coded Link:**
- Cause: `Drought`
- Effect: `Selling livestock`
- Source_ID: `12`
- Quote: "Because of the drought, we had to sell our livestock"

Interpretation Rule COD-MIN: Minimalist Coding Principles

The coding schema prioritizes explicit claims over complex theoretical frameworks.

- **The 90% Rule:** Code the simple form **"X causally influenced Y"**. Explicit coding of complex logic (enablers, sufficiency, non-linearity) is not provided for.
- **Propositions, Not Variables:** Factors are simple propositions (e.g., "Jo started shouting"), not variables with values. Distinct concepts should not be merged prematurely.
- **Example (why this matters):** Code `Poverty` and `Wealth` as distinct factors rather than values of one "Economic Status" variable. A source may claim `Poverty -> Stress` and also `Wealth -> Investment`; collapsing these too early can obscure distinct narratives.
- **No Absences:** Do not code absences. If a source does not mention a factor, it is unknown, not absent.
- **Realism & Partiality:** "X influenced Y" means X had the causal power to affect Y in this context. It implies a contribution, not total determination.

Surface cognition

- **Goal:** Model the speaker's expressed causal thinking (cognition), not necessarily underlying objective reality.
- **Method:** Code only what is explicitly said; avoid inferring hidden variables or unstated counterfactuals.

Interpretation Rule COD-MULTI: The meaning of multiple links

A table containing multiple links simply asserts the logical conjunction of the links:

- Source S1 claims that A causally influenced B in virtue of its causal power to do so.
- Source S1 claims that C causally influenced D in virtue of its causal power to do so.
- Source S2 claims that A causally influenced C in virtue of its causal power to do so.

So, unless specific contexts are specified, if Source S1 claims that `A -> C` and also that `B -> C`, this is neither a contradiction nor (by default) a claim that A-and-B jointly influenced C as a package. It is simply two separate claims.

So if families are talking about reasons for family disputes, and family F mentions social media use, and family G mentions homework, we do not usually interpret this as meaning that family F does *not* think that homework can also be a cause of family disputes.

3. The Filter Pipeline (Query Language)

Analysis is performed by passing a links table through a sequence of filters.

Syntax Rule FIL-PIPE: The Interpretation Pipeline

The interpretation of a result is defined by the cumulative restrictions or transformations of the filters applied.

- **Syntax:** `Input |> Filter1 |> Filter2 |> Output`
- **Multi-line Syntax:**

```
Input
    |> Filter1
    |> Filter2
    |> Output
```

- **Processing:** Filters are applied sequentially. The output of `Filter N` becomes the input of `Filter N+1`.

Types of Filters

- **All filters take a Links table as input.** Most filters return a Links table (so they can be chained). **Output filters terminate the pipeline** by returning a derived view (table/summary) rather than a Links table.

In practice (as in the app), filter behaviour is often **multi-effect**. For example, a filter may rewrite labels *and* add tracking columns. So instead of forcing each filter into exactly one "type", we treat each filter as having an **effect signature**:

- **Row selection**: changes *which links* are included (drops/retains rows).
- **Label rewrite**: changes `Cause`/`Effect` labels (recoding/normalisation).
- **Column enrichment**: adds or recalculates columns (metadata/metrics/flags), typically to support later filtering or display.
- **Configuration**: changes display/formatting settings without changing the links table (app-specific; not formalised here as a core links-table transform).

Below, filters are grouped by their **primary intent**, and each rule declares its **Effects:** line.

Row-selection filters

Syntax Rule FIL-CTX: Context Filters

Reduces the evidentiary base based on Source metadata.

- **Effects:** row selection
- **Operation:** `filter_sources | <criteria...>`
- **Interpretation:** Restrict the evidence base to links whose `Source_ID` is in the retained set of sources.

Syntax Rule FIL-FREQUENCY: Content Filters

Reduces the evidentiary base based on signal strength.

- **Effects:** row selection
- **Operation:** `filter_links | <criteria...>`
- **Interpretation:** Retain only links meeting an evidence threshold (e.g., `min_source_count=2`).

Syntax Rule FIL-TOPO: Topological Filters

Retains links based on their position in a causal chain.

- **Effects:** row selection
- **Operation:** `trace_paths | from="<factor>" | to="<factor>" | <options...>`
- **Interpretation:** "Retaining only mechanisms that connect *From* to *To*."

Label-rewrite filters

Interpretation Rule FIL-ZOOM: The Zoom Filter (Hierarchical Syntax)

Extends the logic to handle nested concepts via a separator syntax.

- **Effects:** label rewrite
- **Syntax:** Factors may use the `;` separator (e.g., `General Concept; Specific Concept`).
- **Interpretation:** `A; B` implies `B` is an instance or sub-component of `A`.
- **Operation:** `transform_labels | zoom_level=1`. If `zoom_level=1`, rewrite labels by truncating text after the first separator.
- **Inference:** At Zoom Level 1, `A; B` is treated logically as `A`.

Interpretation Rule FIL-OPP: The Combine Opposites Filter (Bivalence Syntax)

Extends the logic to handle polarity/negation.

- **Effects:** label rewrite; column enrichment
- **Syntax:** Factors may use the `~` prefix (e.g., `~Employment`) or tag pairs.
- **Interpretation:** `~A` is the negation of `A`.
- **Operation:** `combine_opposites`. Rewrites negative labels (e.g., `~Income`) to their positive counterparts (`Income`) and adds tracking columns such as `flipped_cause` and `flipped_effect`.
- **Inference:** Evidence for `~A -> ~B` is treated as corroborating evidence for `A -> B` (with flipped polarity).

Column-enrichment filters

Syntax Rule FIL-BUNDLE: The Bundling Filter

This filter aggregates co-terminal links (links with the same cause and effect) to calculate evidence metrics without reducing the row count. We normally think of it as being automatically applied after any other filter.

- **Effects:** column enrichment
- **Operation:** `bundle_links`
- **Definition (bundle object):** `Bundle(A, B) = all links L where L.Cause <mark> A AND L.Effect </mark> B`
- **Logic:** For every link `L`, identify the set of all links `S` where `S.Cause <mark> L.Cause` AND `S.Effect </mark> L.Effect`.
- **Transformation:** Appends new columns to the Links table:
- `Bundle`: For convenience, a text representation of the connection (e.g., "A -> B").
- `Citation_Count`: The total count of rows in set `S`. Represents volume of coding.
- `Source_Count`: The number of unique `Source_IDs` in set `S`. Represents breadth of evidence (consensus).
- **Lemma:** `Source_Count <= Citation_Count`.

We measure importance using two distinct metrics:

- **Citation Count:** The total number of times a link or factor was mentioned across the entire project. This counts every single row in the data.
- *Technical:* `citation_count`
- **Source Count (or Number of People):** The number of *unique* sources (people or documents) that mentioned a link or factor. This avoids double-counting if one person repeats the same point multiple times.
- *Technical:* `source_count`

Output filters

Output Rule OUT-FACTORS: Factors table view

Returns a Factors table (one row per factor) derived from a Links table (typically after `FIL-BUNDLE`).

- **Operation:** `factors_view`
- **Interpretation:** Aggregate over the set of factor labels appearing anywhere in `Cause` or `Effect`, and compute per-factor summaries (e.g., role metrics).

Output Rule OUT-MAP: Graphical map view

Returns a graphical network view of the current Links table.

- **Operation:** `map_view`
- **Interpretation (data):**
- **Nodes:** Factors (labels appearing in `Cause` or `Effect`).
- **Edges: Bundles** (one edge per `Cause -> Effect` pair), built from the **current filtered/transformed labels** (so the map reflects Zoom/Combine-Opposites/etc.).
- **Bundling:** If bundle-level columns are not already present, the map view implicitly applies `FIL-BUNDLE` to compute bundle metrics (e.g., `Citation_Count`, `Source_Count`) used for labels and styling.
- **Interpretation (presentation):** The view includes a formatting configuration that maps derived metrics to visual encodings (e.g., edge width by citation/source count; edge colour/arrowheads by mean sentiment; node colour by outcomeness; node/label sizes by frequency), plus a legend summarising the current view and applied filters.

Definition Rule MET-NODE: Factor Role Metrics

These metrics describe the topological role of a factor.

- **In-Degree (incoming citations):** Count of incoming links (times this factor appears as an Effect).
- *Technical:* `citation_count_in`
- **Out-Degree (outgoing citations):** Count of outgoing links (times this factor appears as a Cause).
- *Technical:* `citation_count_out`
- **Outcomeness:** `In-Degree / (In-Degree + Out-Degree)`.
- *Interpretation:* A score nearing 1 indicates an Outcome; a score nearing 0 indicates a Driver.

4. Example Queries

Example A: The "Drivers" Query *Question: What do female participants say are the main drivers of Income?*

```
Result = ProjectData
  |> filter_sources | Gender="Female"         // Rule FIL-CTX
  |> trace_paths | to="Income" | steps=1      // Rule FIL-TOPO
  |> filter_links | min_citations=2           // Rule FIL-FREQUENCY
```

Example B: The "Mechanism" Query *Question: Is there valid narrative evidence that Training leads to Better Yields?*

```
Result = ProjectData
  |> transform_labels | zoom_level=1                 // Rule FIL-ZOOM
  |> trace_paths | from="Training" | to="Yield" | thread_tracing=TRUE   // Rule FIL-TOPO + Ru
```

5 Causal Inference?

Inference Rule INF-EVID: Evidence is not effect size

We quantify **evidence strength**, not **causal effect strength**.

- **Observation:** `Link | Cause="A" | Effect="B"` appears 10 times.
- **Inference:** There are 10 pieces of evidence (10 coded mentions) for the claim `A -> B`.
- **Invalid inference:** The influence of A on B is 10 times stronger than a link appearing once.

Inference Rule INF-FACT: Factual Implication

If we observe `Link | Cause="A" | Effect="B" | Source_ID="S1"`:

- **Deduction:** `Source S1 claims A happened/exists`.
- **Deduction:** `Source S1 claims B happened/exists`.

Inference Rule INF-THREAD: Thread Tracing (Valid Transitivity)

We can infer a long causal chain (indirect influence) only if one source provides every step.

- **Logic:** `Link | Cause="A" | Effect="B" | Source_ID="S1"` AND `Link | Cause="B" | Effect="C" | Source_ID="S1"` `=>` Valid path `A -> B -> C`.

Inference Rule INF-CTX: The Context Rule (The Transitivity Trap)

We cannot infer causal chains by stitching together different sources without checking context.

- **Logic:** `Link | Cause="A" | Effect="B" | Source_ID="S1"` AND `Link | Cause="B" | Effect="C" | Source_ID="S2"` `=>` INVALID path `A -> C`, unless S1 and S2 share the same `Context`.

Context

These questions may depend on a somewhat hidden assumption: that all the causal claims (the links) come from a single context. Such as, in most cases, when all the claims are all agreed on by a group as in participatory systems mapping (PSM). For example, when we wrote "which factors are reported as being causally central?", can we really answer that by simply checking the network? From:

> Factor X is central within these claims

Can we deduce:

> Factor X is claimed to be central

Or, can we go from:

> There are two relatively separate groups of causal claims?

Can we deduce:

> It is claimed that there are two relatively separate groups of causal claims?

In general, no. It is easy to think of counter-examples. If we ask a parents and children about the causal network surrounding family disputes, we might get two relatively separate causal networks with only a little overlap. From this we cannot conclude that these respondents taken together claim that there are two relatively separate systems. It might be that the parents and children indeed are giving information about relatively separate systems about which they each have the best information, or it might be that the two groups are telling conflicting and perhaps incompatible stories.

We could express this as, say, the first axiom of causal mapping:

> If a network of causal evidence from context C has property P, we can conclude that there is evidence that the corresponding causal network has property P, but again only in context C.

P(E(N)) --> E(P(N))

We often assume that contexts are sources and sources are contexts. But this is not always the case. For example one respondent might give two sets of information, one from before losing their job and one set from afterwards, without trying to encode the job loss as a causal factor within the network of claims.

In a PSM workshop, there may be multiple respondents but, as long as they construct a consensus map, these are all treated as one source. Part of the job of the moderator is also to ensure that the claims (evidence) all come from one context, which is the same as saying: we can validly make inferences like those above. I don't know whether PSM moderators actually do this.

You might say "this is all pointless because it depends what you mean by context", and that is exactly true. All we have done is

Appendix A: AI Extensions

These filters extend the core logic using probabilistic AI models (Embeddings or Clustering).

Interpretation Rule FIL-SOFT: The Soft Recode Filter

Extends interpretation logic using semantic similarity (vector embeddings).

- **Operation:** `soft_recode | magnets="<...>" | similarity_threshold="<...>"`
- **Inference:** If `Label A` is similar to `Magnet M` (> threshold), treat `A` as `M`.

Interpretation Rule FIL-AUTO: The Auto Recode Filter

Extends logic using unsupervised clustering.

- **Operation:** `auto_recode | target_clusters=K`
- **Inference:** Factors group together based on inherent semantic proximity into `K` emergent themes.

Plain coding

Combining opposites, sentiment

23 Dec 2025

Abstract

This guide addresses a cluster of **tricky but practical problems** in causal coding: how to represent **oppositeness**, **sentiment/valence**, and **"despite" constructions** in a way that stays close to ordinary language and remains auditable.

Many causal mapping and systems traditions address these issues by quickly "variablising": treating factor labels as variables, and links as signed (and sometimes weighted) relationships. That can be powerful, but it also introduces strong extra commitments (about polarity, scales, functional form, and what counts as "more/less") that are often not warranted by ordinary narrative text.

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

1. We start with **combining opposites** as a conservative label convention plus an explicit transform over a links table.
2. We then "turn 45 degrees" to the different-but-overlapping problem of **sentiment**, which becomes especially useful/necessary when doing AI-assisted coding and embedding-based aggregation. Sentiment and opposites are hard to combine cleanly.
3. We then introduce **"despite" coding** for countervailing conditions ("E happened despite Z") without pretending that Z "caused" E in the ordinary way.

We end by noting some hard cases where these systems collide.

See also: Minimalist coding for causal mapping; A formalisation of causal mapping; Magnetisation; A simple measure of the goodness of fit of a causal theory to a text corpus.

Intended audience: practitioners doing causal coding from narrative text (especially at scale / with AI assistance) who keep running into polarity/valence edge cases.

Unique contribution (what this guide adds):

- A conservative **opposites convention + transform** (combine opposites while retaining flip status).
- A clear separation between **oppositeness** and **sentiment** (why they overlap in practice but don't collapse cleanly).
- A scalable encoding for **"despite"** clauses as a link type/tag, rather than mis-coding them as ordinary causes.

Introduction

In the first part of this guide we dealt only with undifferentiated causal links which simply say "C influenced E", or more precisely: "Source S claims/believes that C influenced E." This is the **minimalist** representation: a links table whose rows are individual causal claims with provenance (source id + quote) and whose columns include at minimum `Cause` and `Effect` labels.

Minimalist-style causal links are commonly used in (at least) two ways:

- **Event-claim reading (QuIP-style)**: interpret a link as a claim about a past episode ("C happened; E happened; C made a difference to E"), with an open question of how far it

generalises.
- **Factor-relation reading**: treat links as claims about influence relations among factors, without committing to what happened in any specific case.

In Part 1 we also introduced **hierarchical factor labels** using the `;` separator, where `C; D` can be read as "D, an example of / instance of C", and can later be **rewritten** (zoomed) to a higher level by truncating the label.

This guide (Part 2) adds three extensions that remain compatible with minimalist link coding:

- **Opposites conventions** in factor labels (a label-level device).
- **Sentiment/valence** as an additional annotation layer (useful especially with AI coding).
- **Despite coding** to capture countervailing conditions without misrepresenting them as ordinary causes.

We will describe the conventions in app-independent terms. (The Causal Map app happens to implement these ideas as part of a standard "filter pipeline", but the logic is not app-specific.)

Combining opposites

The problem

In everyday coding, we often end up with "opposite" factors like:

- `Employment` vs `Unemployment`
- `Good health` vs `Poor health`
- `Fit` vs `Not fit`

If we keep these as unrelated labels, we make downstream analysis harder. For example:

- When we query for "health", we may miss evidence coded as "illness".
- We cannot easily compare (or combine) evidence for `Fit -> Happy` with evidence for `Not fit -> Not happy` without manually re-aligning them.

Many causal mapping traditions solve this by treating factors as variables with signed links. Here we describe a simpler alternative that stays close to ordinary language and avoids variable semantics "by default".

The convention: mark opposites in labels

To signal that two factor labels are intended as opposites, use the `~` prefix:

- `Y` and `~Y`

We talk about **opposites** rather than plus/minus because this avoids implying valence or sentiment. For example:

- `Smoking` is the opposite of `~Smoking` (not smoking), but which one is "good" depends on context.

Non-hierarchical opposites are straightforward:

- `Eating vegetables`
- `~Eating vegetables`
- `Smoking`
- `~Smoking`

The useful part: apply a transform/filter to a links table (and/or a map view)

The convention above is only a convention until we do something with it. The next step is to apply an explicit **transform** to a links table (and then to whatever views are derived from it).

The transform is:

1. Detect opposite pairs that are present (both `Y` and `~Y` appear in the current dataset).
2. Rewrite any occurrence of `~Y` to `Y`.
3. Record, for each link endpoint, whether it was flipped (e.g. `flipped_cause`, `flipped_effect`).

After this transform:

- We can aggregate evidence for `Y` and `~Y` under a single canonical factor label `Y` **without losing the original meaning**, because we still know which endpoints were flipped.
- A link has two local "polarities": whether the cause label was flipped, and whether the effect label was flipped.
- If **exactly one** end is flipped, the overall relationship direction is "reversed" in the intuitive sense (compared to the unflipped link).
- If **both** ends are flipped, the overall relationship direction is not reversed, but the evidence is still distinct (it came from the opposite-on-both-ends claim).

Crucially: **no information is lost** by the transform, because flip-status remains attached to the evidence. You can always reconstruct the original statement-level content from the transformed table.

This general "apply transforms to a links table; then render a map/table view of the transformed data" is the same idea as the filter pipeline described in the user-guide material: filters are operations over a links table, and maps are derived views of the filtered/transformed links.

Opposites coding within a hierarchy

When using hierarchical labels (with `;`), the `~` sign may appear:

- at the very start of the whole label, and/or
- at the start of any component within the label.

The same transform idea applies: to "combine opposites" we flip components so that opposites align at each hierarchical level.

Opposites within components of a hierarchy

Sometimes we need `~` within components, e.g.:

- `Healthy habits; eating vegetables`
- `~Healthy habits; ~eating vegetables`

and:

- `Healthy habits; ~smoking` (not smoking is a healthy habit)
- `~Healthy habits; smoking` (smoking is an unhealthy habit)

After combining opposites, these pairs can be aligned under shared canonical labels while retaining flip status per component (so we do not collapse "healthy" into "unhealthy" or vice versa by accident).

Bivalent variables?

Opposites coding is **not** the same as assuming that every factor is a bivalent variable (present vs absent). We are not claiming exhaustiveness: it is not the case that everything must be either `Wealthy` or `Poor`, and it is often wrong to treat "absence" as having causal powers.

Opposites coding is a practical device used only where:

- both poles occur naturally in the text and therefore in the coding, and
- it would usually be incoherent to apply both poles simultaneously in the same sense.

Which pairs of factors should we consider for opposites coding?

Use opposites coding for a pair of factors X and Y (i.e. recode Y as `~X` or recode X as `~Y`) when both occur in the data and are broadly opposites.

If in doubt about which member to treat as the canonical label, we usually pick X as the “primary” member if it is:

- usually considered as positive / beneficial / valuable, and/or
- usually associated with “more” of something rather than “less” of something.

Alternative convention (explicit opposite pairing)

Sometimes you may have pairs that are conceptually opposite but do **not** share a clean string form like `Y` vs `~Y` (e.g. `Wealthy` vs `Poor`).

In that case, use an explicit pairing tag so that a deterministic transform can combine them later. For example:

- `Wealthy [1]`
- `Poor [~1]`

This makes the pairing unambiguous: `Poor [~1]` is declared to be the opposite of `Wealthy [1]`. A “combine opposites” transform can then rewrite the opposite-labelled item to the canonical label while recording flip status (exactly as described above). This is the same general idea as the “transform filters temporarily relabel factors” pattern in the filter pipeline documentation (see `content/999 Causal Map App/080 Analysis Filters Filter links tab ((filter-link-tab)).md`), but the logic is independent of any particular software.

What can we *do* with opposites once we have them?

Once opposites are marked (and optionally combined via an explicit transform), we can apply ordinary operations to a links table and then render useful views:

- **Querying**: searching for `Y` can intentionally retrieve both `Y` and `~Y` evidence (depending on whether you search pre- or post-transform).
- **Aggregation without collapse**: you can summarise evidence under a canonical label `Y` while still distinguishing which claims involved the opposite sense via flip flags.
- **Visualisation**: you can render a map from the transformed links table and style links differently depending on whether the cause and/or effect endpoint was flipped, so viewers can see “this includes opposite-evidence” rather than mistaking it for ordinary evidence.

This “links table → transforms → map/table view” pattern is the same general idea as a filter pipeline (implemented in many tools; the Causal Map app is one).

Adding sentiment

Polarity of factor labels

There are challenges with coding and validating concepts which on the one hand could be seen to have polarity from a quantitative point of view and on the other hand may have positive or negative sentiment associated with them. Quantitative polarity and subjective sentiment often overlap in confusing ways. When coding, distinguishing between opposites like employment and unemployment is usually important: they can be viewed as opposites, but each pole has a distinct meaning which is more than just the absence of the other. We can call these "bipolar" concepts after (Goertz, 2020).

However, though employment and unemployment can be seen as opposites from a "close level" view, at a more general or abstract level, they could both fall under a category like economic issues. Their

NLP embeddings may have surprisingly high cosine similarity.

For other pairs like not having enough to eat, and having too much to eat, there are multiple opposites (which may appear frequently as a causal factor) with an intervening zero (which may not be mentioned very often).

Coding these different kinds of concept pairs can be difficult and depends on use and context. For these and other reasons, our naive approach codes employment and unemployment as well as not having enough to eat, and having too much to eat separately.**

How to add sentiment?

You can now auto-code the sentiment of the consequence factor in each link.

You only have to do this once, and it takes a little while, so wait until you've finished coding all your links.

When you are ready, click on the File tab, and under the 'About this file', there's the 'ADD SENTIMENT' button. You just have to click on it and wait for the magic to happen

Design sem nome (1).png

So each claim (actually, the consequence of each claim) now has a sentiment, either -1, 0 or 1.

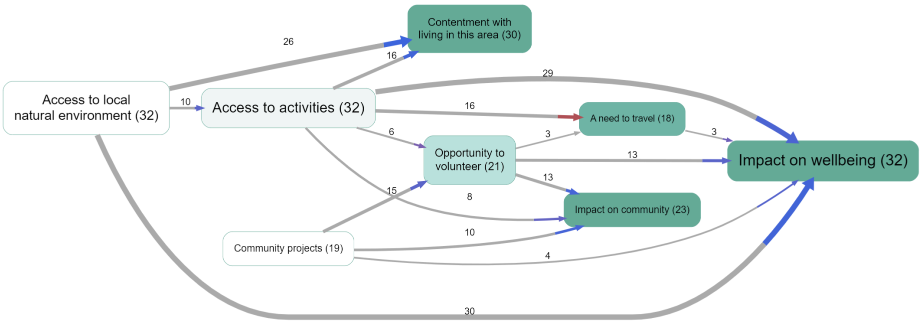
Many links are actually a bundle of different claims. We can calculate the sentiment of any bundle, as simply the average sentiment. So an average sentiment of -1 means that all the claims had negative sentiment. An average of zero means there were many neutral sentiments and/or the positive and negative sentiments cancel each other out.

Only the last part is coloured, because the colour only expresses the sentiment of the effect, not the cause.

Once you have autocoded sentiment for your file, you can switch it on or off using 🎨 Formatters: Colour links.

Tip

When displaying sentiment like this, reduce possible confusing by making sure that you either use only neutral factor labels like Health rather than Good health or Improved Health: an exception is if you have *pairs* of non-neutral labels like both Poor health alongside Good health. You can do this either in your raw coding or using ✨ Transforms Filters: 🧲 Magnetic labels or **📔 Canonical workflow**


Contentment with living in this area (30)
26
16
29
Access to local natural environment (32)
10
Access to activities (32)
16
A need to travel (18)
6
3
3
Opportunity to volunteer (21)
13
Impact on wellbeing (32)
13
15
8
Impact on community (23)
Community projects (19)
10
4
30

Filename: solvacare. Citation coverage 25%: 610 of 2426 total citations and 32 of 32 total coded sources are shown here.
Numbers on factors show source count.. Factor sizes show citation count. Darker factor colours show greater outcomeness.
Numbers on links show source count.
Zooming in to level 1 of the hierarchy. Auto clustering factors using label set new. Top 8 factors by citation count. Showing only links with at least 3 sources.

Adding some colour: a discussion of the problem of visualising contrary meanings

The problem

We’ve already described our approach to making sense of texts at scale by almost-automatically coding the causal claims within them, encoding each claim (like “climate change means our crops are failing”) as a pair of factor labels (“climate change” and “our crops are failing”): this information is visualised as one link in a causal map. We use our “coding AI” to code most of the causal claims within a set of documents in this way. We have had good success doing this quickly and at scale without providing any kind of codebook: the AI is free to create whatever factor labels it wants.

There is one key remaining problem with this approach. Here is the background to the problem: if the coding AI is provided with very long texts, it tends to skip many of the causal claims in fact contained in the documents. Much shorter chunks of text work best. As we work without a codebook, this means that the AI produces hundreds of different factor labels which may overlap substantially in meaning. In turn this means that we have to cluster the labels in sets of similar meaning (using phrase embeddings and our “Clustering AI”) and find labels for the sets. This all works nicely.

But the problem is that, when we use phrase embeddings to make clusters of similar labels, seeming opposites often have high cosine similarity. Unemployment and employment vectors are similar – they would for example often appear on the same pages of a book – and both are quite different from a phrase like, say, "climate change". But this is unsatisfactory because if in the raw text we had a link describing how someone lost their job, coded with an arrow leading to a factor unemployment alongside another piece of text describing how someone gained work, represented by an arrow pointing to employment if these two labels are combined, say into employment or employment issues the items look very similar and we seem to have lost some essential piece of information.

Can’t we use opposites coding?

In ordinary manual coding (see ➕➖ Opposites) we solve this problem by marking some factors as contrary to others using our ~ notation (in which ~Employment can stand for Unemployment, Bad employment, etc) and this works well. However while it is possible to get the coding AI to code using this kind of notation, it is not part of ordinary language and is therefore not understood by the embeddings API: the ~ is simply ignored even more often than the difference between Employment and Unemployment. In order to stop factors like employment and unemployment ending up in the same cluster it is possible to exaggerate the difference between them by somehow rewriting employment as, say, “really really crass absence of employment” but this is also unsatisfactory (partly because all the factors like really really crass X tend to end up in the same cluster).

New solution

So our new solution is simply to accept the way the coding-AI uses ordinary language labels like employment and unemployment and to accept the way the embedding-AI clusters them together. Instead, we recapture the lost “negative” meaning with a third AI we call the “labelling AI”. This automatically codes the sentiment of each individual causal claim so that each link is given a sentiment of either +1, 0 or -1. For this third step we use a chat API. The instruction to this third AI is:

> "I am going to show you a numbered list of causal claims, where different respondents said that one thing ('the cause') causally influenced another ('the effect') together with a verbatim quote from the respondent explaining how the cause led to the effect.

> The claims are listed in this format: 'quote ((cause --> effect))'.

> The claims and respondents are not related to one another so treat each separately.

> For each claim, report its *sentiment*: does the respondent think that the effect produced by the cause is at least a bit good (+1), at least a little bad (-1) or neutral (0).

> Consider only what the respondent thinks is good or bad, not what you think.

> If you are not sure, use the neutral option (0).

> NEVER skip any claims. Provide a sentiment for every claim."

The previous solution coloured the whole link which was fine in most cases but led to some really confusing and incorrect coding where the influence factor was involved in the opposite sense, as in Access to activities below. One might assume that the red links actually involve some kind of negative (or opposite?) access to activities, but we don't actually know that because it wasn't coded as such. Other alternatives would be to also automatically separately code the sentiment of the first part of the arrow, but this doesn't work because sometimes the sentiment is not in fact negative. We would have to somehow automatically code whether the influence factor is meant in an opposite or contrary sense but this is hard to do.

**

Despite-claims

24 Jan 2026

"Despite" coding

The problem: countervailing conditions that “failed”

Narratives often contain claims of the form:

> “E happened despite Z.”

Examples:

- “The heavy rains made the river levels rise, **despite** the prevention and clearance work.”
- “**Despite** me reminding him multiple times, he missed the train.”

In ordinary language, the “despite” clause does two things at once:

1. It signals that **Z was expected to prevent or reduce E** (a countervailing influence).
2. It asserts that **E happened anyway** (so Z was insufficient, absent, overridden, or ineffective in that case).

If we only code the main causal claim (e.g. `Heavy rains -> River levels rose`), we lose potentially important information about attempted mitigations, resistance, barriers, or protective factors.

But it is also usually wrong to code the “despite” clause as an ordinary positive causal link, e.g. `Prevention work -> River levels rose`, because that flips the intended meaning on its head.

The convention: a “despite” link type (or tag)

We therefore treat “despite” as a **link type**, not a different semantics for `->`.

A minimal encoding is:

- **Main link** (ordinary): `X -> E`
- **Despite link** (typed/tagged): `Z -despite-> E`

Equivalently (in a plain links table), keep the same `Cause` and `Effect` columns but add either:

- a `link_type` column with values like `normal` vs `despite`, or
- a `link_tags` column containing a tag like `#despite`.

The key idea is that `-despite->` does **not** mean “Z caused E”. It means:

> “Z was presented as a countervailing influence against E, in virtue of its causal power to work against E, but E occurred anyway.”

What can we do with “despite” links?

Because the distinction is explicit in the links table, we can decide—per analysis—how to treat these links:

- **Visualise** them distinctly (different colour/line style), so the map shows “tensions” rather than silently dropping them or misreading them.
- **Filter**: show only `despite` links to see what people present as failed protections / resistances / mitigations.

- **Compare**: for similar outcomes, do some sources describe Z as effective (ordinary preventive claim, e.g. `Z -> ~E`) while others describe “despite” failures (`Z -despite-> E`)? That contrast can be analytically important.
- **Count separately**: include them in evidence tallies only under a separate counter (e.g. `Despite_Citation_Count`) so you don’t accidentally inflate evidence for “Z causes E”.

Worked example

Text:

> “The heavy rains made the river levels rise, despite the prevention and clearance work.”

Coding:

- `Heavy rains -> River levels rose`
- `Prevention and clearance work -despite-> River levels rose`

This preserves the main mechanism while also storing the claim that a mitigation was present but insufficient.

Optional background: force dynamics (why “despite” is cognitively natural)

Leonard Talmy’s “force dynamics” treats many causal expressions as patterned talk about forces: an **Agonist** with a tendency (towards motion/rest) and an **Antagonist** that counters it. “Despite” is a canonical force-dynamics marker: it signals a countervailing force that failed to stop the outcome.

You do not need this theory to use `-despite->` coding; it is only a helpful explanation of why “despite” claims feel different from ordinary causal claims, and why it can be worth capturing them explicitly.

Hard cases (brief notes)

- **“Despite” + opposites**: should `Z -despite-> E` be treated as evidence for `Z -> ~E`? Usually no; treat it as its own link type, or at most as weaker evidence depending on your analysis goal.
- **Bipolar concepts**: some pairs (employment/unemployment) have high cosine similarity (similar contexts) but are "opposite" in ordinary talk. This is why sentiment often becomes relevant when doing AI-assisted clustering/aggregation.

4.1 Force Dynamics: Agonists and Antagonists in Latent Space

Leonard Talmy’s theory of **Force Dynamics** posits that human causal understanding is rooted in the interplay of forces: an **Agonist** (the entity with a tendency towards motion or rest) and an **Antagonist** (the opposing force).

- *Linguistic Patterns:* "The ball kept rolling despite the grass" (Agonist: Ball; Antagonist: Grass). "He let the book fall" (Removal of Antagonist).

- *LLM Evaluation:* Recent studies have tested LLMs on translating and explaining these force-dynamic constructions.

  - **Findings:** GPT-4 demonstrates a sophisticated grasp of these concepts. When translating "He let the greatcoat fall" into languages like Finnish or Croatian, the model correctly selects verbs that convey "cessation of impingement" (allowing) rather than "onset of causation" (pushing).

  - **Implication:** This suggests that LLMs have acquired a **schematic semantic structure** of causality. They do not merely predict words; they map the *roles* of entities in a physical interaction. However, this capability degrades in abstract social contexts. For example, in the sentence "Being at odds with her father made her uncomfortable," models sometimes misidentify the Agonist/Antagonist relationship, struggling to map "emotional force" as accurately as "physical force".

Untitled

**

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.

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

23 Dec 2025

Abstract

Suppose an evaluation team has a corpus of interviews and progress reports, plus (at least) two candidate theories of change (ToCs): an original one and a revised one. A practical question is: **which ToC better fits the narrative evidence**?

With almost-automated causal coding as described in (Powell & Cabral 2025; Powell et al. 2025), we can turn that into a simple set of *coverage-style* diagnostics: how much of the coded causal evidence can be expressed in the vocabulary of each ToC.

See also: Working Papers; Minimalist coding for causal mapping; Magnetisation.

Intended audience: evaluators and applied researchers comparing candidate ToCs (or other causal frameworks) against narrative evidence, who want a transparent "fit" diagnostic that does not pretend to be causal inference.

Unique contribution (what this paper adds):

- A definition of **coverage over causal links** (not just themes): link / citation / source coverage variants.
- A simple protocol for comparing candidate ToC vocabularies using hard recode or Magnetisation (soft recode).
- A careful positioning of "coverage" relative to mainstream QDA usage (saturation/counting as support for judgement, not a mechanical rule).

1. The core idea: "coverage" of evidence by a codebook

In ordinary QDA (thematic coding), researchers often look at how widely a codebook or set of themes is instantiated across a dataset: which codes appear, how frequently, and whether adding more data still yields new codes (saturation). Counting is not the whole of qualitative analysis, but it is a common, explicitly discussed support for judgement and transparency (Saldaña 2015). Critiques of turning saturation into a mechanical rule-of-thumb are also well known (Braun & Clarke 2019).

Our twist is: because we are coding **causal links** (not just themes), we can define coverage over *causal evidence* rather than over text volume.

2. Minimal definitions

- A **coded link** is a row of the form `(Source_ID, Quote, Cause_Label, Effect_Label, ...)`.
- A **ToC codebook** is a vocabulary (list) of ToC factor labels you want to recognise in the corpus.
- A **mapping** from raw labels to ToC labels can be done either:
- strictly (exact match / "hard recode"), or
- softly via magnetisation (semantic similarity; "soft recode") — see Magnetisation.

3. Coverage measures you can compute

Assume we have a baseline set of coded links `L` (from open coding), and a ToC codebook `C` (as magnets / targets).

3.1 Link coverage (our main measure)

Link coverage = proportion of coded links whose endpoints can be expressed in the ToC vocabulary.

Two variants (pick one and state it explicitly):

- **Both-ends coverage**: count a link as "covered" only if *both* cause and effect are mapped to some ToC label.
- **At-least-one-end coverage**: count a link as "covered" if either endpoint maps (useful when ToC vocabulary is intentionally partial).

3.2 Citation coverage (weighted link coverage)

If your dataset has multiple citations per bundle (or you have `Citation_Count`), compute coverage over **citations**, not just distinct links:

- covered citations / total citations

This answers: "what proportion of the *evidence volume* is expressible in this ToC?"

3.3 Source coverage (breadth)

Source coverage = number (or proportion) of sources for which at least (k) links are covered by the ToC vocabulary.

This answers: "does this ToC vocabulary work across many sources, or only a small subset?"

4. Protocol (how to use it)

For each candidate ToC:

1. Build a ToC codebook `C` (ideally keep candidate codebooks similar in size and specificity, otherwise you are partly measuring codebook granularity).
2. Map raw labels to `C` (hard recode or soft recode).
3. Compute:
4. link coverage (both-ends and/or one-end),
5. citation coverage (if available),
6. source coverage (with an explicit (k)).
7. Inspect the **leftovers** (uncovered labels/links): what important evidence is the ToC not even able to name?

5. How this relates to "coverage" in mainstream qualitative methods

The word "coverage" is used in a few nearby ways in qualitative methodology:

- **Code (or theme) saturation**: whether new data still yields new codes/themes; the distinction between "code saturation" and "meaning saturation" is often emphasised (e.g. Hennink et al. on code vs meaning saturation; and the broader critique that saturation is not a universal stopping rule in all qualitative paradigms) (Braun & Clarke 2019).
- (For orientation, see: Hennink, Kaiser & Marconi (2017) "Code Saturation Versus Meaning Saturation", *Qualitative Health Research*, DOI: `10.1177/1049732316665344`; Guest, Bunce & Johnson (2006) "How Many Interviews Are Enough?", *Field Methods*, DOI: `10.1177/1525822X05279903`.)
- **Counting for transparency**: many QDA approaches use counts (how often codes occur; how widely they occur across cases) as a support for analytic claims, without equating frequency with importance (Saldaña 2015).

What we are doing here is closer to: **how much of the coded evidence can be expressed in the language of a candidate theory**, which is a "fit" diagnostic rather than a claim about truth.

6. Caveats

- Coverage is sensitive to **granularity**: broader ToC labels will (almost by definition) cover more.
- High coverage does not imply causal truth; it only implies that the ToC vocabulary is a good *naming scheme* for a large share of the corpus.
- Low coverage can mean either “ToC is missing key mechanisms” or “coding/mapping is too strict” — inspect leftovers before concluding.

References

Braun, & Clarke (2019). *To Saturate or Not to Saturate? Questioning Data Saturation as a Useful Concept for Thematic Analysis and Sample-Size Rationales*. https://doi.org/10.1080/2159676X.2019.1704846.

Powell, & Cabral (2025). *AI-assisted Causal Mapping: A Validation Study*. Routledge. https://www.tandfonline.com/doi/abs/10.1080/13645579.2025.2591157.

Powell, Cabral, & Mishan (2025). *A Workflow for Collecting and Understanding Stories at Scale, Supported by Artificial Intelligence*. SAGE PublicationsSage UK: London, England. https://doi.org/10.1177/13563890251328640.

Saldaña (2015). *The Coding Manual for Qualitative Researchers*. Sage.

Magnetisation

23 Dec 2025

Magnetisation (soft recoding with magnetic labels)

Abstract

After inductive causal coding (manual or AI-assisted), you typically end up with **many overlapping factor labels** ("rising prices", "inflation", "cost of living increases", …). *Magnetisation* (aka **soft recoding**) is a fast, transparent way to standardise these labels **without re-coding the original text**: you supply a list of target labels ("magnets"), and each existing label is reassigned to its closest magnet by semantic similarity (using embeddings). Unmatched labels can be kept (default) or dropped. This paper is the definitive guide to magnetisation: what it is, how it works, how to choose magnets, how to tune the similarity threshold ("magnetism"), and how to check whether your magnet set is actually representing the corpus. See also: Working Papers; Minimalist coding for causal mapping; Combining opposites, sentiment; A simple measure of the goodness of fit of a causal theory to a text corpus.

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.

Unique contribution (what this paper adds):

- A practical, audit-first definition of magnetisation as a **label rewrite layer** over a links table (not a re-coding of source text).
- A concrete set of tuning decisions (magnets list, similarity threshold, drop-unmatched, second-pass "only unmatched") and what each trade-off buys you.
- A positioning of magnets within modern NLP practice (embeddings + vector retrieval + LLM-assisted labelling) while keeping the method deterministic and checkable.

---

1. Why magnetisation exists (the practical problem)

If you do open-ended coding, especially "radical zero-shot" AI coding, you get:

- high recall (lots of causal claims captured),
- but also a **hairball** of thousands of near-duplicate labels.

You then need some way to standardise labels so that maps and tables become readable *and* queryable.

There are two broad routes:

- **Auto clustering**: let an algorithm discover groups (often useful for exploration).
- **Magnetisation**: you decide the groups (magnets) and the algorithm assigns labels to them (best when you already know the vocabulary you want to use).

Magnetisation is basically "codebook application", but done **softly** (by semantic similarity) rather than by exact string matching.

---

2. What magnetisation does (definition)

We start with a links table containing labels in `Cause` and `Effect`. Magnetisation defines a rewriting function:

- `recode(label) = closest_magnet(label)` if similarity ≥ threshold

- otherwise:
- keep the label unchanged, or
- drop it (optional), depending on your setting.

The key point: magnetisation changes **labels**, not the underlying evidence. All quotes/provenance remain exactly the same; we are only rewriting factor names to make aggregation and visualisation usable.

3. How magnetisation works (algorithm, conceptually)

3.1 Embeddings and similarity

Each label (raw label and magnet) is represented as an **embedding**: a numerical vector encoding of meaning. In NLP this general idea underpins modern semantic search and vector retrieval, including transformer-based representations (e.g., BERT-style embeddings (Devlin et al. 2019)) and retrieval-augmented pipelines that use dense vector indices (Lewis et al. 2021). Semantic similarity is measured by cosine similarity (the angle between embedding vectors).

Two practical notes that matter for "soft recoding":

- Embeddings are not "definitions"; they are empirical similarity machines trained on large corpora. This is a feature for fast standardisation, but it means you must **audit** what got pulled into each magnet.
- Some meanings that humans treat as opposites can have high cosine similarity (because they occur in similar contexts). This is why magnetisation often needs to be paired with explicit conventions like opposites handling (see also: `015 Combining opposites, sentiment and despite-claims.md`).

3.2 Assignment rule

Given a set of magnets (M_1, \dots, M_k) and a raw label (L):

1. Compute similarity between (L) and each (M_i).
2. Let (M^*) be the most similar magnet.
3. If similarity((L, M^*)) ≥ threshold, rewrite (L \mapsto M^*).
4. If not:
5. either keep (L) unchanged (default), or
6. drop the link(s) containing (L) (optional).

If a label is similarly close to multiple magnets, it is assigned to the single closest one.

4. Controls / parameters (what you can actually tune)

These parameters matter much more than people expect; they are the difference between "clean story" and "semantic soup".

4.1 Magnets list (one per line)

Magnets are your target vocabulary: one magnet per line. They can be single-level labels ("Income changes") or hierarchical labels ("Health behaviour; hand washing"), but see the hierarchy notes below.

4.2 Similarity threshold ("magnetism")

- **Low threshold**: magnets are strong, attract more labels, higher coverage, higher risk of pulling in wrong material.
- **High threshold**: magnets are weak, attract fewer labels, lower coverage, higher precision.

There is no universally correct value. You tune it empirically by inspecting:

- what got pulled into each magnet, and

- what remained unmatched.

4.3 Drop unmatched (optional)

- If **off**: unmatched labels remain as-is (you keep everything; your map may still be messy).
- If **on**: links with labels that match no magnet (above threshold) are removed (you get a clean view, but you risk hiding important “leftover” themes).

A good pattern is: first run with drop-unmatched **off**, then decide whether the leftovers are noise or a missing theme.

4.4 Process only unmatched (second-pass magnetisation)

A powerful workflow is to run magnetisation twice:

1. First pass: broad, obvious magnets, drop-unmatched **off** (keep everything).
2. Second pass: set “process only unmatched” **on**, and focus on what the first pass didn’t capture.

This avoids rewriting already-good matches and concentrates your attention on the residual complexity.

4.5 Recycle weakest magnets (optional)

If you have many fiddly magnets, they can “nibble away” evidence from your core magnets, then disappear from the view later (because they are small, or filtered out by other steps). Recycling temporarily removes the N weakest magnets and reassigns their labels to stronger magnets using the same threshold rule.

This is especially useful when you have, say, 50 magnets but only a handful show up prominently and your coverage is unexpectedly low.

4.6 Remove hierarchy (optional convenience)

Sometimes hierarchical magnets are not ideal *as magnets* (the full string may reduce similarity matching). A common workaround is:

- magnetise using simple magnets (“floods”),
- then relabel to the preferred hierarchical form (“environmental problems; floods”) using a simple mapping step (soft relabel / bulk relabel).

4.7 Saving and reusing magnet sets (practical)

In practice you iterate magnet sets. Two simple storage patterns are:

- **Saved views / bookmarks**: save a view that includes your current magnet list and settings (so you can return to the exact same recoding later).
- **Codebook storage**: keep a “canonical” magnet list as a codebook-like artifact for the project, and paste it into the magnets box when needed.

Untitled

4.8 Getting initial magnets (optional, but often useful)

If you are starting from a blank page, there are three common ways to get an initial magnet list:

- **From an official ToC / framework**: paste the ToC factor language as magnets and see what matches and what is left over.
- **From auto-clustering**: cluster raw labels, then promote the best cluster labels into magnets.
- **From AI suggestions**: ask an AI to propose candidate magnets based on your current raw labels, then edit them manually.

This “LLM as assistant for proposing candidate labels / codebooks” is now an active area of NLP+QDA work (e.g. LLM-in-the-loop thematic analysis (Dai et al. 2023)). Our use is intentionally narrow: the LLM proposes *names* for groups (magnets), while the actual reassignment is then done deterministically by similarity + threshold, with auditable leftovers.

4.9 Tracking what was recoded (auditability)

It is useful (and in the app this is implemented explicitly) to track:

- whether a link's **cause label** was recoded,
- whether a link's **effect label** was recoded,
- and (at the factor level) whether a factor appears at least once as a recoded label.

This lets you filter to "only recoded", "only still-raw", or to audit the boundary cases.

4.10 Seeing magnet groups in "meaning space" (debugging)

A very fast sanity check is to visualise factors in a 2-D projection of embedding space ("meaning space"):

- magnets as labelled points,
- raw labels as dots coloured by their assigned magnet,
- dot size or density reflecting group size.

This makes it easy to spot:

- magnets that are semantically too close to one another (competition / unstable assignment),
- magnets that are semantically far from the material they are supposed to capture,
- and "orphan" regions of raw labels that are not being captured by any magnet.

5. Choosing good magnets (the single most important part)

5.1 The "magnet wording" tension

If magnets are too abstract, they often match poorly. Magnets work best when they look like the raw labels they need to attract.

This is the key tension:

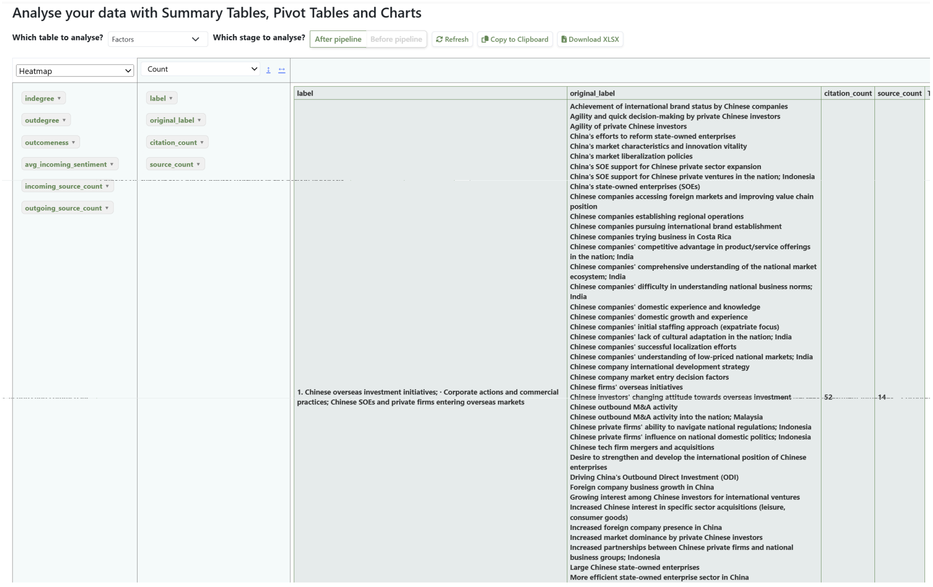
- you want a magnet to express a general theme (e.g., "health behaviours"),
- but the best magnet for matching may need to stay closer to the single-case language actually used in your corpus.

So if the raw labels look like:

- "school creativity project in North district implemented"
- "school creativity project in South district implemented"

then a magnet of the form "school creativity project implemented" will often attract better than "creativity projects implemented in multiple schools".

Similarly, if the raw labels distinguish subgroups explicitly ("girls responded to the training", "boys responded to the training"), a single magnet "children responded to the training" may match worse than keeping subgroup-specific magnets and zooming out later.


Analyse your data with Summary Tables, Pivot Tables and Charts

label	original_label	citation_count	source_count	T
1. Chinese overseas investment initiatives; · Corporate actions and commercial practices; Chinese SOEs and private firms entering overseas markets	Achievement of international brand status by Chinese companies Agility and quick decision-making by private Chinese investors Agility of private Chinese investors China's efforts to reform state-owned enterprises China's market characteristics and innovation vitality China's market liberalization policies China's SOE support for Chinese private sector expansion China's SOE support for Chinese private ventures in the nation; Indonesia China's state-owned enterprises (SOEs) Chinese companies accessing foreign markets and improving value chain position Chinese companies establishing regional operations Chinese companies pursuing international brand establishment Chinese companies trying business in Costa Rica Chinese companies' competitive advantage in product/service offerings in the nation; India Chinese companies' comprehensive understanding of the national market ecosystem; India Chinese companies' difficulty in understanding national business norms; India Chinese companies' domestic experience and knowledge Chinese companies' domestic growth and experience Chinese companies' initial staffing approach (expatriate focus) Chinese companies' lack of cultural adaptation in the nation; India Chinese companies' successful localization efforts Chinese companies' understanding of low-priced national markets; India Chinese company international development strategy Chinese company market entry decision factors Chinese firms' overseas initiatives Chinese investors' changing attitude towards overseas investment Chinese outbound M&A activity Chinese outbound M&A activity into the nation; Malaysia Chinese private firms' ability to navigate national regulations; Indonesia Chinese private firms' influence on national domestic politics; Indonesia Chinese tech firm mergers and acquisitions Desire to strengthen and develop the international position of Chinese enterprises Driving China's Outbound Direct Investment (ODI) Foreign company business growth in China Growing interest among Chinese investors for international ventures Increased Chinese interest in specific sector acquisitions (leisure, consumer goods) Increased foreign company presence in China Increased market dominance by private Chinese investors Increased partnerships between Chinese private firms and national business groups; Indonesia Large Chinese state-owned enterprises More efficient state-owned enterprise sector in China	52	14	

5.2 Avoid semantically ambiguous magnets

If a magnet name has an everyday meaning and also a project-specific meaning, it may attract irrelevant material.

Example: a project about the organisation “Animal Aid” may accidentally attract generic talk about helping animals.

The cleanest fix is to **hard-recode** the intended meaning into an unambiguous label before magnetisation (e.g., rename “Animal Aid” to “The Archibald Organisation”), and then use that as a magnet.

5.3 Use “negative magnets” to siphon off unwanted material

You can include magnets that you later filter out, purely to stop them contaminating your main magnets.

Example: if you want “donating blood” but your corpus contains “donating clothes”, add magnets like “donating goods” and “donating money” so those labels get attracted away from “donating blood”.

5.4 Hierarchical magnets + zooming (coverage trick)

If your material is broad, a small magnet set may cover only a minority of links (that might be the reality of heterogeneous narratives).

If you believe there *is* a shared high-level structure but you can’t find magnets that cover it directly, a practical trick is:

- use many *specific* hierarchical magnets, e.g.
- `Desire for innovation; digital`
- `Desire for innovation; management approaches`
- …
- then apply a zoom level of 1 so they bundle into `Desire for innovation`.

This keeps matching close to raw language, while still allowing a higher-level summary view.

6. How to tell if magnetisation is “working” (coverage + leftovers)

6.1 Coding coverage (a simple diagnostic)

Define **coding coverage** as: for a given derived view (after magnetisation + any other filters), what fraction of the original coded claims (citations) are still represented in the view?

There is always a trade-off:

- more magnets / lower threshold → higher coverage but harder-to-read outputs,
- fewer magnets / higher threshold → lower coverage but clearer outputs.

6.2 Leftovers are not "noise" by default

If you keep unmatched labels, they show you what your magnets are missing. That residual can contain:

- genuinely irrelevant material, or
- important themes not in your current magnet vocabulary.

The "process only unmatched" second-pass pattern is designed to make this iterative refinement fast and disciplined.

6.3 Worked example: small map, non-trivial coverage

In one proof-of-concept analysis we filtered a map down to a small number of high-level factors to keep it readable:

- the summary map contained only 11 factors (plus bundling),
- but still covered **42%** of the raw coded causal claims ("coding coverage"),
- and most sources still contributed at least some citations to the summary.

This is typical: even aggressive simplification can preserve a lot of the evidence base, but you have to measure it. Magnetisation is one of the main levers for increasing coverage at a fixed visual complexity.

7. Relationship to auto-clustering (what clustering is good for)

Magnetisation is not a substitute for clustering; they answer different needs.

- **Clustering** is good for discovery: it can surface unexpected themes you didn't think to create magnets for.
- **Magnetisation** is good for disciplined standardisation: it lets you impose a vocabulary you can justify (ToC terms, evaluator concepts, stakeholder categories, etc.).

A common workflow is:

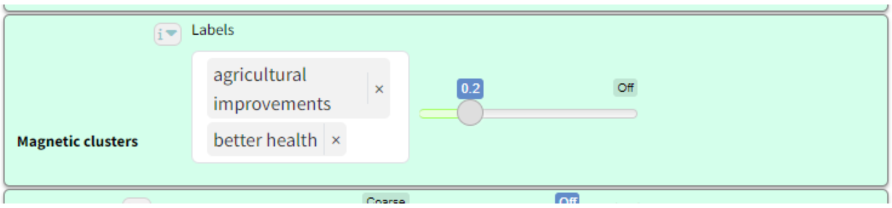
1. run magnetisation with your best current magnet set,
2. then auto-cluster the remaining mess to discover important leftovers,
3. promote the useful leftovers into new magnets,
4. repeat.

8. Visualisation and auditability

Magnetisation should be treated like any other transformation in an analysis pipeline:

- maps are built from the **current transformed labels**,
- but you should always be able to inspect the **original labels and quotes** that were recoded into each magnet.

In practice, it helps to track (at the link level) whether the cause/effect was recoded, so you can filter or audit recoded vs original material.



Magnetic clusters

Labels: agricultural improvements ×, better health ×

0.2

Off

9. Notes on instability (what not to do)

Magnetisation is intentionally a *simple* nearest-magnet assignment. Avoid over-complicating it into NxN “pairwise magnetising” schemes; those tend to be unstable and hard to interpret, especially once you add intervening filters and frequency cutoffs.

Appendix: background on embeddings + clustering (kept for completeness)

The coding procedure often results in many different labels for causes and effects, many overlapping in meaning. A common way to explore that mess is to cluster labels using embeddings.

One typical three-step pattern is:

1. **Inductive clustering**: cluster the embeddings (e.g., `hclust()` in base R) (R Core Team, 2015).
2. **Labelling**: ask an AI (or a human) to propose distinct labels for each cluster; then adjust.
3. **Deductive clustering**: reassign each raw label to the nearest proposed cluster label, provided similarity is above a threshold (to ensure cohesion).

This appendix matters for magnetisation because it gives you a way to propose candidate magnets and to understand what your current magnet set is not capturing.

Outcomeness (optional metric): one simple “role” metric is the proportion of incoming vs outgoing citations (a normalised Copeland-style score) (Copeland, 1951). Factors with low outcomeness can be treated as drivers; high outcomeness as outcomes. This can help when deciding whether your magnets are mixing drivers/outcomes in a way that makes your maps hard to read.

Short positioning note (where magnetisation fits in NLP/LLM practice)

Magnetisation is not novel as an NLP *primitive*; it is a deliberately simple application of well-established components:

- **Distributional semantics / embeddings**: represent short texts as vectors so that semantic similarity can be approximated geometrically (classic word embeddings: Mikolov et al., 2013; Pennington et al., 2014; contextual encoders: Devlin et al., 2019 (Devlin et al. 2019); sentence embeddings for similarity search: Reimers & Gurevych, 2019).
- **Nearest-neighbour assignment with thresholds**: assign items to the closest prototype/centroid (here: magnets) with an explicit similarity cutoff. This is the same family of idea used in vector search and retrieval-augmented generation systems (Lewis et al. 2021).
- **LLMs as labelling assistants**: use an LLM to propose names for groups / codebooks, while keeping the core data transformation auditable and deterministic (Dai et al. 2023).

What *is* specific to this paper is the methodological stance for qualitative causal coding: magnets are treated as a **transparent, auditable recoding layer** over a links table with provenance, not as an end-to-end black box analysis.

Additional background references (APA 7; NLP/LLM classics)

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems, 26*. https://arxiv.org/abs/1310.4546

Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1532–1543). https://doi.org/10.3115/v1/D14-1162

Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. https://arxiv.org/abs/1908.10084

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., … Amodei, D. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems, 33*, 1877–1901. https://arxiv.org/abs/2005.14165

References

Dai, Xiong, & Ku (2023). *LLM-in-the-loop: Leveraging Large Language Model for Thematic Analysis*. https://arxiv.org/abs/2310.15100v1.

Devlin, Chang, Lee, & Toutanova (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. http://arxiv.org/abs/1810.04805.

Lewis, Perez, Piktus, Petroni, Karpukhin, Goyal, Küttler, Lewis, Yih, Rocktäschel, Riedel, & Kiela (2021). *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks*. http://arxiv.org/abs/2005.11401.

Assessing change in (cognitive models of) systems over time

19 Jan 2026

The Architect and the Clerk: Analysing Central Bank speeches with GenAI-supported causal mapping

Abstract

Using Generative AI (GenAI) as a "mere assistant" to code text according to a pre-defined thematic codebook is possible but relatively uninteresting, because the AI is not involved in theory building. Many qualitative researchers are using GenAI for collaborative meaning-making, with and without a codebook. This paper presents another way, based on causal mapping, a well-established approach in social research which codes *only* causal claims within text. We use a "zero-shot" approach with no codebook, using only "In vivo" labels for the identified causes and effects. This results in heaps of causal claims containing large numbers of cause and effect labels. Making sense of these heaps is then done in two phases: firstly applying Large Language Models (LLMs) for semantic clustering and secondly using non-GenAI causal mapping techniques to visualise overall and divergent causal narratives within the text(s). This procedure is quite highly standardised and yet still depends on creative and iterative human input at key points: Qualitative judgement (what are the main cause/effect labels and how are they best organised in order to build an interesting and useful theory?) remains central while many of the other tasks become more reproducible, checkable, and scalable.
We demonstrate this approach with a corpus of speeches from leaders of Central Banks, asking: what drives what in the national and global economy, in the opinion of these experts? And how do these opinions change over time?

Introduction: Beyond the Conversation

GenAI can be used to automatically code themes according to a pre-defined codebook (Xiao et al. 2023). But this is a relatively uninteresting use of AI as it is used purely as a "clerk" and is not directly involved in theory building.

Recent debates about AI-assisted qualitative research invite us to give AI a bigger role. One way to do this is the "conversational" paradigm, where AI acts as a co-researcher in a dialogic, hermeneutic process (Friese 2025; n.d.; Dai et al. 2023; n.d.; Nguyen-Trung 2025). That approach leverages the AI's capacity to identify meanings within larger passages of text and thus in various ways to take part in a conversation around theory building.

We propose a complementary, yet distinct, path. It is based on causal mapping, a social research approach technique that identifies and visualizes beliefs about "what causes what" (Nadkarni & Shenoy 2004; Scavarda et al. 2006; Ackermann et al. 2004; Eden et al. 1992; Axelrod 1976). In causal mapping, we code *only* the causal claims within a text. This can be done with a codebook, but more interesting is with a zero codebook, using only "In vivo" labels for causes and effects.

This preliminary task can be done by a "clerk" and does not need an "architect".

Causal mapping is a very good fit for the initial step of almost-automated coding because a) causal coding is quite surprisingly easy to automate on a page-by-page basis (Studdiford & Lupyan 2025;

Powell & Caldas Cabral 2025; Veldhuis et al. 2024; Powell et al. 2025), being a much more highly determined task than, say, the task of "identify themes within this text"; and b) extracted *causal* narratives are usually more interesting and closer to answering actual research questions than lists of themes (Britt et al. 2025; Powell & Caldas Cabral 2025).

Making sense of -- and building a theory around -- the resulting heaps of causal claims (containing large numbers of cause and effect labels) is then done in two phases: first, applying LLM-supported semantic clustering and then using established (non-GenAI) causal mapping techniques to visualise the overall and divergent causal narratives within the text(s). Qualitative judgement (what are the main cause/effect labels and how are they best organised?) remains a central theory-building step. Most of the other tasks are reproducible, checkable, and easy to scale.

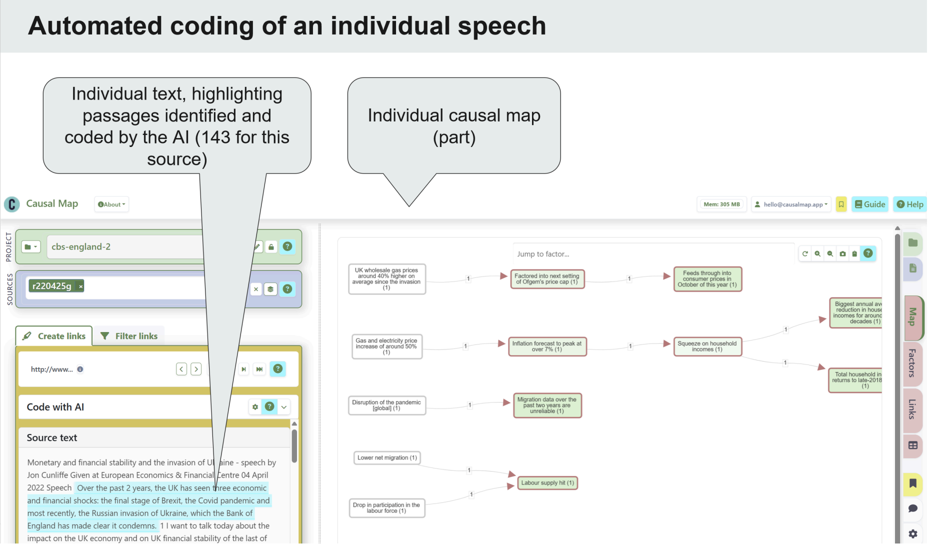
Case study dataset: Central Bank speeches (1996–2023)

We used a corpus of central bank speeches (1996–2023) Campiglio et al. (2025): 1,354 speeches spanning 1996–2023, sampling up to 20 speeches per year, resulting in a sample of 522 speeches, equivalent to 3934 pages at 500 words/page.
We use this corpus purely as an illustrative worked example; our contribution is methodological rather than domain-substantive.

Causal coding

We applied causal coding to this dataset. The coding prompts were quite restrictive. We did not, for example, ask the AI to "summarize" or "discuss themes." We instructed it simply to identify every specific instance where the text claimed X causally influences Y, and record only cause, effect and a supporting verbatim quote, without using a pre-existing codebook. We call this minimalist or "barefoot" causal coding: capturing explicit causal claims using the source's own vocabulary.

In practice, we iterated the extraction prompt and ran basic quality checks (spot-checking pages and links) until we were satisfied that the outputs were consistent with these instructions.


Automated coding of an individual speech
Individual text, highlighting passages identified and coded by the AI (143 for this source)
Individual causal map (part)
Causal Map
About
Mem: 305 MB
hello@causalmap.app
Guide
Help
PROJECT
cbs-england-2
SOURCES
r220425g
Create links
Filter links
http://www...
Code with AI
Source text
Jump to factor...
UK wholesale gas prices around 40% higher on average since the invasion (1)
Factored into next setting of Ofgem's price cap (1)
Feeds through into consumer prices in October of this year (1)
Biggest annual av reduction in house incomes for around decades (1)
Gas and electricity price increase of around 50% (1)
Inflation forecast to peak at over 7% (1)
Squeeze on household incomes (1)
Total household in returns to late-2018 (1)
Disruption of the pandemic [global] (1)
Migration data over the past two years are unreliable (1)
Lower net migration (1)
Labour supply hit (1)
Drop in participation in the labour force (1)
1
Map
Factors
Links

Monetary and financial stability and the invasion of Ukraine - speech by Jon Cunliffe Given at European Economics & Financial Centre 04 April 2022 Speech Over the past 2 years, the UK has seen three economic and financial shocks: the final stage of Brexit, the Covid pandemic and most recently, the Russian invasion of Ukraine, which the Bank of England has made clear it condemns. 1 I want to talk today about the impact on the UK economy and on UK financial stability of the last of

Figure 1: A screenshot of the Causal Map app after automatically coding one source.

The Creative Challenge: Magnetic Clustering as (part of) theory building

The output of this extraction is a "hairball" of thousands of links between thousands of unique factor labels often with overlapping meaning (e.g., "rising prices," "inflation," "cost of living increase"). It is here that the AI-supported process confronts the core challenge of standard qualitative inquiry: Theory building.

In most forms of qualitative coding, the researcher must decide which codes belong together to form clusters. This is a highly under-determined task. It is not completely arbitrary, because some sets of clusters will better cover the raw codes than others. But it is not well specified, because many alternative "theories" may cover the raw codes equally well. The assignment of codes to clusters as well as assessing the goodness of fit of this assignment is carried out through Magnetic Clustering, as follows.

1. **Identifying initial clusters:** Sorting thousands or, as in this case, tens of thousands of labels into clusters of similar meaning is a task which in principle humans can carry out but in practice the scale is overwhelming. So, each label (raw label and magnet) is represented as an **embedding**: a numerical vector encoding of meaning. In NLP this general idea underpins modern semantic search and vector retrieval, including transformer-based representations (e.g., BERT-style embeddings (Devlin et al. 2019)) and retrieval-augmented (RAG) pipelines that use vector indices (Lewis et al. 2021). Semantic similarity is measured by cosine similarity (the angle between embedding vectors). We then use a clustering algorithm (k-means) to create clusters of raw labels with similar meaning (Grootendorst 2022).
2. **Defining the Labels:** The next task is to generate labels for each cluster. This is a more feasible task for humans to do (usually there are 5-50 clusters with a handful of exemplars for each cluster) but in practice even this is tiresome over many iterations, so we employ GenAI to help but usually inspect and adjust the labels as our theoretical ideas of the subject matter evolve.
3. **The Attraction:** these first two steps are relatively standard procedure. But our next step differs: An algorithm "attracts" the thousands of raw, granular labels to these suggested cluster labels, which we call "Magnets", based on semantic similarity: each raw code is assigned to that Magnet to which it is semantically most similar. An assignment fails if the closeness of code to theme is below some similarity threshold, in which case the raw code is not assigned to any Magnet. The percentage of raw codes which succeed in being reassigned or "magnetised", for any given closeness threshold, can be called "coverage". Note **we do not actually use the original cluster solution other than to provide exemplars in order to create the magnetic labels, which then create their own clusters**: the clusters of raw labels attracted to each magnet will be similar to the original solution but not identical to it. This is what makes our approach different from most clustering procedures: we **re-cluster the raw labels all over again** on the basis of their similarity to these Magnets, a process which can also be called "**soft recoding**". This means that we can tweak the Magnet labels on the fly, individually or severally, and watch as the raw labels quickly recluster themselves. We follow the principle of Braun and Clarke’s (Braun & Clarke 2021) Reflexive Thematic Analysis that themes must show internal homogeneity (by using k-means clustering) and external heterogeneity (our labelling procedure, whether conducted by AI or ourselves, is explicitly designed to find labels which are distinct from one another in meaning). The original AI-generated labels provide a good start, but they are usually only the beginning of a process. For example, we might want to substitute one label with a pair of labels which are relatively similar in meaning -- and so would not have been produced in the initial phase -- but which represent for us an important theoretical distinction.
4. The Refinement: We iteratively tweak the "magnetism" and experiment with different lists of magnets.

Qualitative scholars usually distinguish between mere clusters of codes and *themes* which may form part of a more meaningful theory and which have a closer relationship to the research questions. In our case, it would be perfectly *possible* to use more theory-driven labels, but these would in general not function as well as magnets as they would usually be further away semantically from the original raw labels. Ideally we might discover labels which are both more interesting theoretically and nevertheless also fit the original language of the respondents. Failing this, we can use an additional filter to provide convenience labels for the original Magnets which describe them in terms of our evolving theory, for example we might relabel "Interest rate cuts" as part of a hierarchical system: "Monetary policy; interest rate cuts", in such a way that only "interest rate cuts" is given to the clustering algorithm but the full label is shown in user-facing outputs such as tables[1].

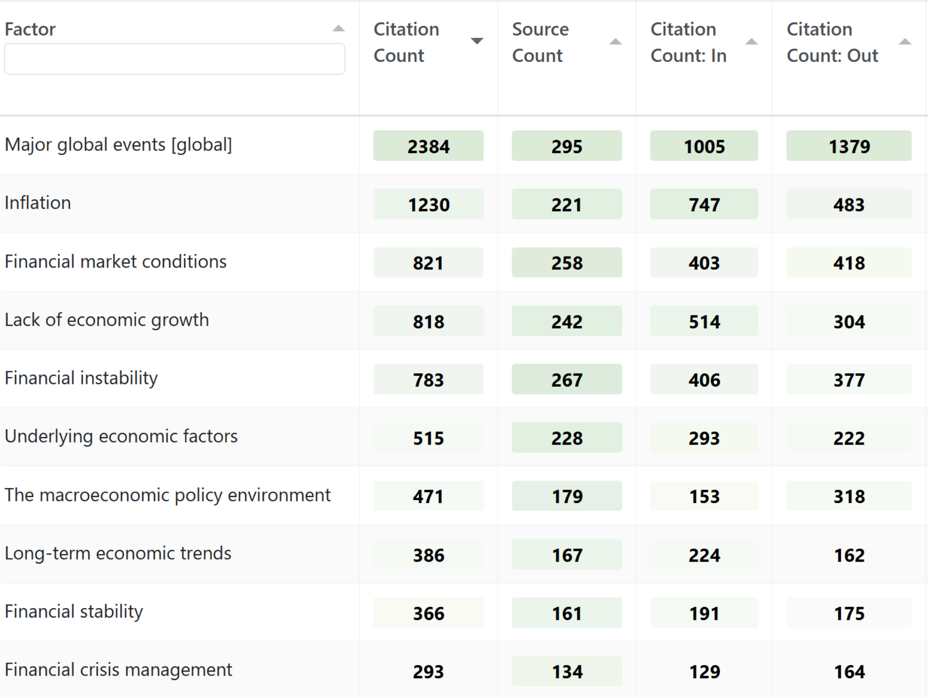
This process of Magnet selection is related to the codebook development phase in, say, Thematic Analysis, but it is a soft (temporary) rather than a hard coding procedure. It is also possible to go back to the start and "hard-recode" the source texts again from scratch, using the Magnets list as a codebook. In practice, "soft-recoding" with magnetic labels is our preferred approach because we can test and compare the "fit" of different theoretical models to the data more or less in real time.

Deciding which Magnets to use is a substantial qualitative decision that shapes the entire model and which falls under the researcher's theory-building responsibility even when using AI-assisted causal mapping.

However the results of the "magnetisation" process still does not reach the level of declarative conclusions which one would expect from a theory-building analysis. That is because the immediate result of the magnetic is not a single text but a query-able qualitative model, as discussed in the next section.

Results: A Query-able Qualitative Model

The result of this process is not a static narrative, but a dynamic causal map ,(i.e. a query-able database of causal evidence) : a large number of links between a set of common labels (the "Magnets"). It is possible to simply list the "Magnets" and the frequency with which they were mentioned, or show a map with only the most frequently mentioned Magnets. While we can create standard frequency tables (see figure 2), the most interesting part of this process is being able to interrogate this qualitative model to answer more interesting questions, by applying a library of pre-set filters individually or in chains, as discussed in the next section.



Factor ▲	Citation Count ▼	Source Count ▲	Citation Count: In ▲	Citation Count: Out ▲
Major global events [global]	2384	295	1005	1379
Inflation	1230	221	747	483
Financial market conditions	821	258	403	418
Lack of economic growth	818	242	514	304
Financial instability	783	267	406	377
Underlying economic factors	515	228	293	222
The macroeconomic policy environment	471	179	153	318
Long-term economic trends	386	167	224	162
Financial stability	366	161	191	175
Financial crisis management	293	134	129	164

Figure 2: The most frequently mentioned causal factors, in descending order of citation count. Also showing "Source Count" (number of sources mentioning the factor at least once) and "Citation Count: In" aka "Indegree" (number of mentions of the factor as an effect) and "Outdegree" (number of mentions of the factor as a cause).[2](https://app.causalmap.app/?bookmark=966) 2026-01-12 16:44_] As an aside: Researchers used to most forms of systems modelling will find this kind of presentation confusing because some factors appear with both positive and negative variants (Financial stability and Financial instability) and some do not. But this simply reflects the sources' causal narratives. Causal mapping does not usually attempt to model a logical world of facts: it attempts to model the (often not very logical) mental models implicit within texts. So, if two opposed concepts appear separately within the text, by default we will simply use them as-is. Other techniques

are available to combine opposed pairs of ideas, but they will not be discussed here. [Combining opposites, sentiment](#).

What are the main consequences of Brexit according to the sources?

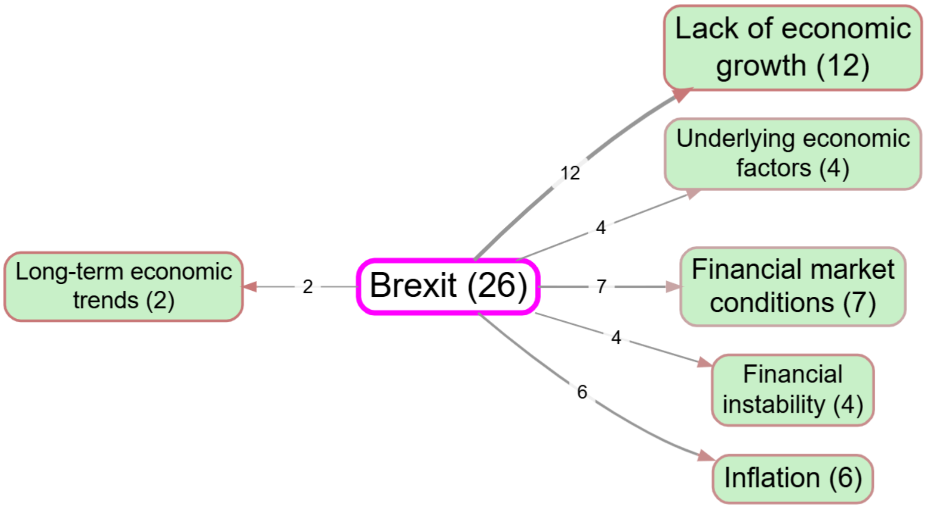

Lack of economic growth (12)
Underlying economic factors (4)
Long-term economic trends (2)
Brexit (26)
Financial market conditions (7)
Financial instability (4)
Inflation (6)
12
4
2
7
4
6

Figure 3: A screenshot of the Causal Map app after automatically coding one source. [3] In this case we apply a simple filter to explore the consequences of "Brexit": any Magnets listed as its immediate effects. We can explore the resulting map further, e.g. by inspecting the verbatim quotes associated with each link.

What are the top-level causal narratives in each decade (specifically, as explanations of growth) and how do they differ from one another?

Here, we use the same set of magnets for each decade, filter to show only paths leading to Economic growth or Lack of economic growth and then again show the most frequently mentioned links in each case.

Decade 1 1996-2005

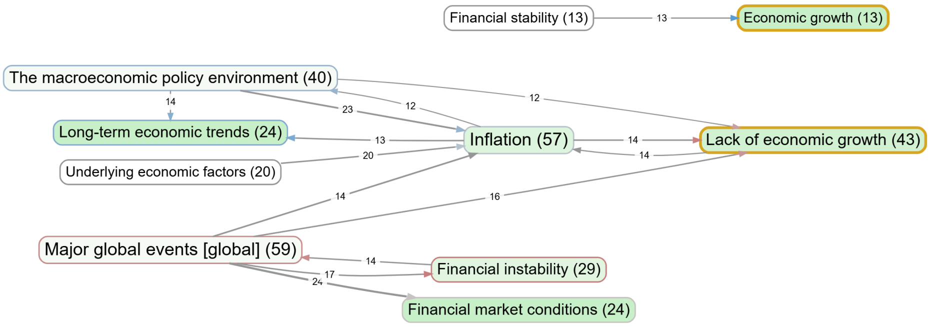
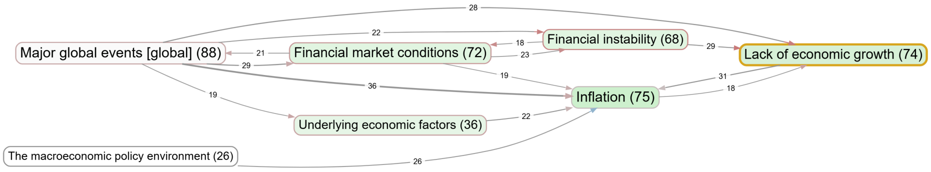

Financial stability (13)
13
Economic growth (13)
The macroeconomic policy environment (40)
14
23
12
12
Long-term economic trends (24)
13
Inflation (57)
14
Lack of economic growth (43)
14
Underlying economic factors (20)
20
14
16
Major global events [global] (59)
14
17
24
Financial instability (29)
Financial market conditions (24)

*Figure x * Filters applied: Soft recode: 18 magnets, similarity>=0.62; Source Groups: Decade=1; Path: to "Growth", 2 steps; Link frequency: top 19 by source count.

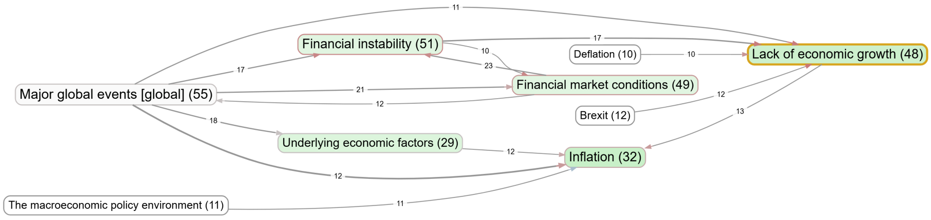
Decade 2 2006-2015


28
22
Major global events [global] (88)
21
29
Financial market conditions (72)
18
23
Financial instability (68)
29
Lack of economic growth (74)
19
36
31
18
19
Inflation (75)
Underlying economic factors (36)
22
The macroeconomic policy environment (26)
26

Bookmark [#962](#) Filters applied: Soft recode: 18 magnets, similarity>=0.62; Source Groups: Decade=2; Path: to "Growth", 2 steps; Link frequency: top 19 by source count.

With these filters, retaining only the most frequent links, there is no more mention of: Long-term economic trends or of Financial stability leading to economic growth.

Decade 3 2016-2023


11
Financial instability (51)
17
Deflation (10)
10
Lack of economic growth (48)
10
23
17
Major global events [global] (55)
21
Financial market conditions (49)
12
12
Brexit (12)
13
18
Underlying economic factors (29)
12
Inflation (32)
12
The macroeconomic policy environment (11)
11

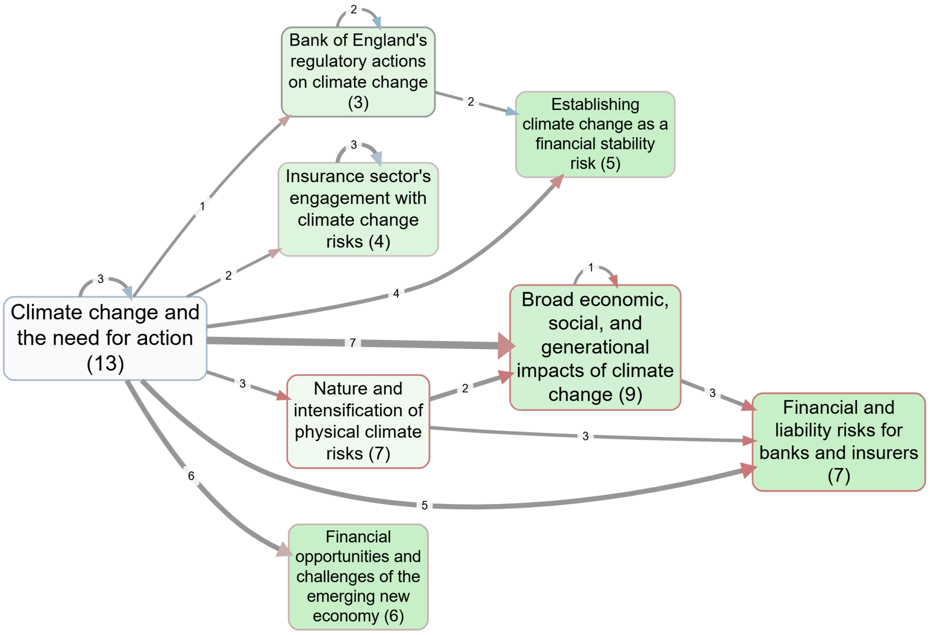
Bookmark #963 2026-01-13 09:12_

Retaining only the most frequent factors and links, now we see the first mentions of Brexit, Deflation and Climate change.

This technique allows us to systematically compare maps across groups or time-points. This is something which is much harder to do reliably using human coders. Concepts such as inter-rater reliability are often treated as a distraction by qualitative researchers and there are good reasons for that: but using the techniques described here we can have our cake and eat it: get involved in in-depth theory construction and have reliable and reproducible coding too, at the scale of hundreds of sources.

What are the narratives around climate change?

In this case, climate change and its specific causes and effects did not appear frequently enough to appear much amongst the most important Magnets. So we simply filtered for all mentions of climate change (as causes and effects) amongst the *raw* labels and created new Magnets to cover just this material.


2
Bank of England's regulatory actions on climate change (3)
2
Establishing climate change as a financial stability risk (5)
3
Insurance sector's engagement with climate change risks (4)
1
2
3
4
Climate change and the need for action (13)
1
Broad economic, social, and generational impacts of climate change (9)
7
3
2
Nature and intensification of physical climate risks (7)
3
3
Financial and liability risks for banks and insurers (7)
6
5
Financial opportunities and challenges of the emerging new economy (6)

Filename: cbs-england-2-filtered-2. Citation coverage 0% of all sources: 59 citations shown out of 37648. Factors -- size: citation count; numbers: source count; colour: outcomeness; border: avg incoming sentiment (blue=positive, grey=neutral, red=negative). Links -- width: citation count; labels: source count; arrowheads: effect sentiment (blue=positive, grey=neutral, red=negative). Filters applied: Sources included: All sources. Labels: climate chang, climate crisis, global warm; Soft recode+: 10 magnets, sim=0.62; Link freq: minimum 2 citations. Bookmark #967 2026-01-13 09:24

This is also a good place to showcase and additional way to involve GenAI in this kind of research: the entirely automated production of narrative vignettes which describe any given map and the data behind it, using a standardised prompt. The results are as follows.

> **Central Bankers Warn: Climate Crisis Poses Significant Threat to UK Financial Stability**
>
> This analysis, derived from a highly filtered view of UK central bankers' speeches, outlines their perspectives on climate change, focusing on its implications for the financial system and broader economy.
>
> The discourse among UK central bankers, as captured in this data, predominantly frames **Climate change and the need for action** (40 citations) as a foundational premise. This central theme is consistently reported to lead to a range of significant negative outcomes across the economy and financial sector.
>
> A primary pathway identified is the link between **Climate change and the need for action** and **Broad economic, social, and generational impacts of climate change** (11 citations, average sentiment -0.727). The sources frequently highlighted the severe negative implications for society and future generations. Furthermore, it was observed that climate change directly contributes to **Financial and liability risks for banks and insurers** (6 citations, average sentiment -1), indicating a clear concern for the financial sector's direct exposure.
>
> The **Nature and intensification of physical climate risks** (11 citations) is another critical factor, with reports indicating its direct connection to **Climate change and the need for action** (3 citations, average sentiment -1). These physical risks, in turn, were mentioned as exacerbating both **Financial and liability risks for banks and insurers** (3 citations, average sentiment -1) and the **Broad economic, social, and generational impacts of climate change** (5 citations, average sentiment -1). This suggests a cascading effect where physical climate events translate into tangible financial and societal burdens.
>
> The role of regulatory bodies, specifically the **Bank of England's regulatory actions on climate change** (8 citations), is also a notable theme. These actions were reported to contribute to **Establishing climate change as a financial stability risk** (2 citations, average sentiment 0.5), underscoring the central bank's proactive stance in integrating climate considerations into financial oversight. The sources also mentioned that **Climate change and the need for action** itself prompts these regulatory responses (2 citations, average sentiment -0.5).
>
> While the focus is largely on risks, the speeches also touched upon **Financial opportunities and challenges of the emerging new economy** (6 citations, average sentiment -0.333), indicating an awareness of the dual nature of the transition to a greener economy.
>
> The analysis reveals a strong emphasis on the *effects* of climate change rather than its scientific *causes*. The factor **Climate change and the need for action** serves as the primary causal node, acting as a given and urgent reality. From this starting point, the discourse largely explores the subsequent impacts on financial stability, economic performance, and societal well-being. The central bankers' speeches, therefore, function more as an assessment of the consequences and necessary responses to climate change, rather than an exploration of its origins. The most cited links consistently describe negative effects, such as the "broad economic, social, and generational impacts" and "financial and liability risks for banks and insurers," highlighting a clear concern for the downstream implications of a changing climate.

Discussion: Complementarity and the Architect

The “dialogue” described here is not between human and GenAI directly, but between human and the causal mapping process, mediated by GenAI which does the original coding, suggests magnetic labels and provides summary vignettes (and by the LLMs which provide the label embeddings and underly the clustering process). Although the procedure presented here is quite highly standardised in outline, each step may be iterated several times either alone, or going back one or more steps to revise preceding steps as well. What causal mapping adds is an almost universally applicable way to extract and visualise *what led to what*: a key, theory-adjacent aspect of almost any set of narratives.

The "conversational paradigm" makes liberal use of GenAI as a human-like research assistant who is asked questions like "identify the themes" or "summarise this document", even at the very beginning of the research, following an argument that coding (Nguyen-Trung & Friese 2025) is a perhaps redundant, "skeuomorphic" remnant of the age before AI. But giving such immense degrees of freedom to any interpretative task exposes it to large, arbitrary and poorly defined influences from the analyst, whether human or machine. At least we can hope that the human analyst may be at least partially aware of their own influence on such tasks, due to mood, tiredness or positionality or whatever. Humans' ability to actually understand the influence of these factors on their analysis is of course limited and error-prone. Machines are less likely to be influenced by situational factors but their performance is of course massively influenced by their architecture and training data in a way that they are unlikely to be actually aware of, regardless of what they say if we ask them.

Coding is one way to reduce this exposure to arbitrary influences.

This paper complements the conversational paradigm and shows that standardising and automating many parts of the workflow does not mean dumbing it down. By delegating the massive cognitive load of extraction and clustering mostly to the AI, we free the researcher to focus on the architecture of meaning. The "Architect" is asked for creative input in a well-defined way at well-defined, specific points in the workflow. But it remains a demanding task. Getting the clusters "right" and applying the right filters to create relevant maps requires deep, iterative engagement with the research question. The AI provides the bricks; the human must still design the house.

AI Contribution Disclosure Checklist

- Research Design: Human led (definition of Causal Mapping logic).
- Data Collection: Human/Existing Data.
- Data Analysis (Prompt engineering): human.
- Data Analysis (Coding): AI (Radical Zero-Shot extraction).
- Data Analysis (Clustering): Collaborative (AI performed clustering; Human defined "Magnets" and iteratively refined structure).
- Data Analysis (Answering questions): Mostly human, leveraging existing non-AI algorithms/filters.
- Initial drafting of Paper: AI (Gemini), based on human-provided structural constraints and source files.
- Refining and Editing: Human.

---

1. These labels, like every label in causal mapping, can also form a hierarchical structure -- "Improved economic conditions; improved investor confidence" as well as "Improved economic conditions", but we do not use this extensively in the current study). ↩

2. _Filename: cbs-england-2-filtered-2. Citation coverage 8% of all sources: 3065 citations shown out of 37648. Filters applied: Sources included: All sources. Soft recode+: 18 magnets, sim=0.62; Link freq: top 40 by source count. Bookmark [#966 ↩

3. Legend: Factors — size: citation count; numbers: source count; colour: outcomeness (darker=more incoming links); border colour: average incoming sentiment (blue=positive, grey=neutral, red=negative). Links — width: citation count; labels: source count; arrowheads: sentiment of effect (blue=positive, grey=neutral, red=negative). Filters applied: Soft recode: 18 magnets, similarity>=0.62; Path: from "Brexit", 1 steps; Link freq: minimum 2 sources. ↩

References

Ackermann, Eden, & Cropper (2004). *Getting Started with Cognitive Mapping*.

Axelrod (1976). *The Analysis of Cognitive Maps*. In *Structure of Decision : The Cognitive Maps of Political Elites*.

Braun, & Clarke (2021). *Thematic Analysis : A Practical Guide*. SAGE Publications Ltd. https://www.torrossa.com/it/resources/an/5282292.

Britt, Powell, & Cabral (2025). *Strengthening Outcome Harvesting with AI-assisted Causal Mapping*. https://5a867cea-2d96-4383-acf1-7bc3d406cdeb.usrfiles.com/ugd/5a867c_ad000813c80747baa85c7bd5ffaf0442.pdf.

Campiglio, Deyris, Romelli, & Scalisi (2025). *Warning Words in a Warming World: Central Bank Communication and Climate Change*. https://doi.org/10.1016/j.euroecorev.2025.105101.

Dai, Xiong, & Ku (2023). *LLM-in-the-loop: Leveraging Large Language Model for Thematic Analysis*. https://arxiv.org/abs/2310.15100v1.

Devlin, Chang, Lee, & Toutanova (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. http://arxiv.org/abs/1810.04805.

Eden, Ackermann, & Cropper (1992). *The Analysis of Cause Maps*. https://onlinelibrary.wiley.com/doi/10.1111/j.1467-6486.1992.tb00667.x.

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

Grootendorst (2022). *BERTopic: Neural Topic Modeling with a Class-Based TF-IDF Procedure*. https://doi.org/10.48550/arXiv.2203.05794.

Lewis, Perez, Piktus, Petroni, Karpukhin, Goyal, Küttler, Lewis, Yih, Rocktäschel, Riedel, & Kiela (2021). *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks*. http://arxiv.org/abs/2005.11401.

Nadkarni, & Shenoy (2004). *Nadkarni and Shenoy 2004 -A Causal Mapping Approach.Pdf*.

Nguyen-Trung, & Friese (2025). *On Methodological Incongruence in Applying Generative AI in Qualitative Data Analysis*. https://doi.org/10.2139/ssrn.5874482.

Nguyen-Trung (2025). *ChatGPT in Thematic Analysis: Can AI Become a Research Assistant in Qualitative Research?*. https://doi.org/10.1007/s11135-025-02165-z.

Powell, & Caldas Cabral (2025). *AI-assisted Causal Mapping: A Validation Study*. Routledge. https://doi.org/10.1080/13645579.2025.2591157.

Powell, Cabral, & Mishan (2025). *A Workflow for Collecting and Understanding Stories at Scale, Supported by Artificial Intelligence*. SAGE PublicationsSage UK: London, England. https://doi.org/10.1177/13563890251328640.

Scavarda, Bouzdine-Chameeva, Goldstein, Hays, & Hill (2006). *A Methodology for Constructing Collective Causal Maps**. https://doi.org/10.1111/j.1540-5915.2006.00124.x.

Studdiford, & Lupyan (2025). *Contextual Effects in LLM and Human Causal Reasoning*. https://openreview.net/forum?id=BMHkg3BL6e.

Veldhuis, Blok, family=Boer, Kalkman, Bakker, & family=Waas (2024). *From Text to Model: Leveraging Natural Language Processing for System Dynamics Model Development*. https://doi.org/10.1002/sdr.1780.

Xiao, Yuan, Liao, Abdelghani, & Oudeyer (2023). *Supporting Qualitative Analysis with Large Language Models: Combining Codebook with GPT-3 for Deductive Coding*. In *28th International*

Conference on Intelligent User Interfaces. https://doi.org/10.1145/3581754.3584136.

Lonely in London

24 Jan 2026

Abstract

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.

Substantive results: analysing 48 interviews with young Londoners (18–24), this paper identifies loneliness as a product of four factors: **social performance** (being unseen), **place-based constraints** (threat/transport), **infrastructure needs** (third spaces), and **digital/material filters** (money/apps). It argues for low-cost, "low-judgement" community spaces that address safety and mobility to foster genuine connection and move beyond "crowded anonymity."

Beginning of the AI's paper ...

> ### Title: Loneliness as misrecognition, constrained mobility, and connection infrastructure: a qualitative analysis of 48 interviews with young adults in deprived London boroughs

> ### Abstract

This paper analyses 48 interviews with young adults (18–24) recruited from four deprived London boroughs (Newham, Hackney, Tower Hamlets, Barking & Dagenham) in summer 2019. The dataset combines a free-association interview on “the experience of loneliness” (Part 1) and a place-based task in which participants describe neighbourhood spaces of social connectedness and loneliness (Part 2). Using an iterative qualitative workflow (whole-transcript reading, per-interview memoing, living synthesis documents, and negative-case search), the analysis develops an interpretive account of loneliness as an outcome of interacting mechanisms. Four mechanism stories organise the findings: (1) misrecognition and social performance (being surrounded yet unseen), (2) place-based constraint (crowded anonymity, threat-appraised public space, violence/territoriality, transport friction), (3) connection infrastructures (structured third spaces, faith communities, rule-bound shared-focus settings, and low-judgement “known strangers”), and (4) digital and material filters (social media, dating apps, money) that amplify comparison/objectification or restrict participation. The paper specifies boundary conditions (e.g., home-with-others vs home-alone; parks as connecting spaces vs threat-appraised spaces; social media as comparison amplifier vs neutral background) and argues that interventions need to build low-cost, low-judgement infrastructures for repeated contact while recognising safety, mobility, and material exclusion as gating conditions.

> ### Keywords

loneliness; young adults; neighbourhood; place; reflexive thematic analysis; third spaces

1. Introduction

Loneliness is often discussed as a private emotion: a deficit of friends, a lack of socialising, or an internal spiral of sadness and self-doubt. The interviews analysed here complicate that picture. Participants describe loneliness as exclusion (“left out”), as misrecognition (being near others yet unseen or not understood), as an internal spiral that accelerates in certain environments (bedroom darkness, long unstructured time), and as a safety problem in public space (“better lonely than scared”). In addition, the dataset includes a place task that forces an analytic move away from purely psychological accounts: participants name concrete neighbourhood spaces where connection is possible and where loneliness intensifies, and they explain why those spaces work as they do.

The original study aimed to explore the causes and lived experience of loneliness among young adults (18–24) living in some of London’s most deprived boroughs, and to explore how participants conceptualise neighbourhood places in relation to loneliness and social connectedness (see Fardghasemi & Joffe, 2021). This focus is relevant because young adults are reported to be the loneliest age group in the UK and comparable Western contexts, and because loneliness is patterned by socio-economic conditions and local environments.

The analytic goal of this paper is not to list “causes of loneliness” but to build a mechanism-rich account that explains how loneliness is produced, maintained, and sometimes mitigated in these accounts, and why common remedies (“just go out more,” “join a club,” “get off your phone”) succeed for some situations and fail for others. A secondary goal is to keep the analysis traceable: claims are anchored to interview IDs and tracked in a living evidence file so that boundary cases and contradictions are visible rather than smoothed away.

2. Data

The corpus comprises 48 interview transcripts collected June–August 2019 with participants aged 18–24. A recruitment agency was used to achieve a quota sample from four of London’s most deprived boroughs (Newham, Hackney, Tower Hamlets, Barking & Dagenham; n=16 per borough; see dataset documentation in `sources_md/sources_README.md`).

Data collection had two linked parts. Part 1 used a free-association “grid” task: participants were given a sheet with four empty boxes and asked to express what they associated with “the experience of loneliness” using images and/or words, one idea per box. They then elaborated each box in turn in an interview beginning “can we talk about what you have put in box 1, please?”, with minimal prompts such as “can you tell me more about that?” to reduce content injection. Part 1 interviews lasted ~60 minutes on average and most took place in participants’ homes (some took place in local cafés, parks, or similar when home was not an option).

Part 2 followed immediately after Part 1. Participants wrote or drew one neighbourhood place where they felt most socially connected and one where they felt most lonely, and wrote why beneath each. They then elaborated each place in a short interview using the same low-injection prompting style (e.g., “how does that make you feel in this space?”). Part 2 interviews lasted ~20–30 minutes.

3. Method

3.1 Analytic stance

The analysis follows a reflexive thematic analysis sensibility (Braun & Clarke, 2023). Themes are treated as meaning-unified interpretive stories rather than topic labels that mirror the interview schedule (e.g., “social media,” “family,” “work”). Interpretation is treated as unavoidable: the task is to develop coherent, evidence-grounded explanations, not to pretend that themes “emerge” without analytic decisions. The workflow therefore emphasised (a) memoing as interpretive work, (b) explicit boundary conditions, and (c) negative-case search.

3.2 Workflow and tools (what was done, practically)

Analysis was conducted as an iterative loop over whole transcripts. Each interview was treated as an independent source and summarised in a memo (`ai_files/interview_memos/Interview_XX.md`). Memos captured salient excerpts, candidate codes/mechanisms, and negative/boundary notes. Three living synthesis files were maintained throughout: a codebook with working definitions and exemplars (`ai_files/codebook.md`), a theory-development file capturing mechanism candidates and tensions (`ai_files/theory.md`), and an evidence file that links claims to supporting and boundary excerpts (`ai_files/evidence.md`). A timestamped journal (`ai_files/journal.md`) recorded batches, file updates, and major analytic pivots.

Several lightweight tools were used to keep the workflow consistent and auditable:

- A timestamp script (`ai_files/tools/timestamp.py`) to produce stable timestamps for log entries.
- A markdown wordcount script (`ai_files/tools/wordcount_md.py`) to measure main-text length during drafting.
- Targeted text search (pattern search across transcripts) to locate negative cases (e.g., accounts where parks were not protective, where social media was neutral, where home was the most connected place).

Batches were selected to test and revise developing ideas. For example, after early mechanisms suggested that “third spaces” mattered, later batches were chosen to stress-test this against cases where venues felt empty or unsafe. Similarly, place-based claims were tested against accounts of neighbourhood cohesion and mutual aid.

3.3 How the theory developed (from early codes to a coherent account)

The initial coding landscape contained familiar candidate topics (friends, family, social media, work, home, transport). Through iteration, two decisions reshaped the analysis. First, “home” was treated as a site of multiple mechanisms rather than a single protective factor. Second, “place” was treated as active (constraining or enabling) rather than passive backdrop. Over successive batches, a broader concept—connection infrastructure—emerged to unify varied settings where connection was feasible (structured third spaces, faith communities, rule-bound shared-focus settings, and low-judgement “known strangers”).

The key analytic discipline was to keep contradictions as boundary conditions rather than noise. Where one participant framed a mechanism strongly (e.g., parks as unsafe; social media as hyperreality; home as certainty), the next step was to find cases where that mechanism did not apply, then revise the claim to specify conditions.

Concretely, this meant treating each strong “headline” account as a hypothesis to be tested. For example, early place talk could have supported a single story of “London crowds = loneliness.” However, later interviews forced a more differentiated account: some participants experience crowds as emotionally empty but not overtly threatening; others treat public space as actively unsafe; still others describe neighbourhood cohesion and mutual aid that counters the low-empathy narrative. Similarly, early “home” talk could have been coded as uniformly protective; later interviews made it necessary to separate home-with-others (certainty, recognition, low judgement) from home-alone (rumination, pressure, “thinking chambers”) and from home-without-attunement (co-presence but low communication). This approach makes the final claims less sweeping but more explanatory: it specifies why the same “place” or “platform” can generate opposite outcomes across accounts.

Negative-case work was not treated as a final “limitations” paragraph but as an active driver of theory revision. After a claim was drafted in the evidence file, we deliberately searched for transcripts that would contradict it (e.g., parks as connecting, social media as neutral background, home as the most connected place, venues as anchors rather than escapism). When contradictions were found, the claim was rewritten as a conditional mechanism with boundary notes rather than abandoned or ignored.

This is also where "connection infrastructure" became the central integrative idea: it offered a way to explain why some settings (teams, faith spaces, structured third spaces, rule-bound shared-focus contexts) reliably produced connection even when generic socialising or crowded co-presence did not.

3.4 Researcher roles, rigour, and ethics (secondary analysis)

This paper reports a secondary analysis of an existing qualitative dataset. The primary researchers designed the study and collected the interviews; this analysis involved no participant contact and therefore could not shape recruitment, interview locations, or the participant–researcher relationship during data collection beyond what is documented in the dataset record and the associated primary publication (Fardghasemi & Joffe, 2021).

Analytic rigour was pursued through (a) whole-transcript reading rather than excerpt-only processing, (b) traceability via interview IDs and a living evidence file, and (c) deliberate negative-case search to force conditional rather than blanket claims. This is a single-analyst, AI-led workflow; credibility is therefore addressed through transparency and auditability rather than intercoder reliability or respondent validation (neither were conducted in this secondary analysis).

Ethics for the primary study are reported in the associated publication: "The studies involving human participants were reviewed and approved by UCL Research Ethics Committee (CEHP/2013/500). The patient/participants provided their written informed consent to participate..." (Fardghasemi & Joffe, 2021). For this secondary analysis, we used the de-identified transcripts as provided via the UCL Research Data Repository, and we avoid presenting unnecessary contextual detail that could increase re-identification risk.

4. Findings: four mechanism stories (with boundaries)

Across interviews, loneliness is described as emerging from interacting mechanisms. Four interpretive themes organise the account.

To make the findings easy to audit, the themes below are anchored to short, verbatim excerpts (with interview IDs), followed by interpretive explanation and boundary conditions:

- Theme 1 (Misrecognition) — Interview 01: "I've always had people around me physically, but I haven't felt like they can understand what I'm saying."
- Theme 2 (Place as constraint / threat appraisal) — Interview 41: "Your brain will rather feel lonely than scared... your brain will tell you, you are lonely but you are safe..."; later, in place talk: "lonely is better than scared...".
- Theme 3 (Connection infrastructure / known strangers) — Interview 38: "I sometimes prefer opening up to stranger because they don't know me as much."
- Theme 4 (Digital + material filters) — Interview 20: "snapchat, like they put a story up with them having a Nando's or something... I popped up... oh thanks guys for the invite... And this shows that they know that I don't have money."

Theme 1 — Misrecognition and the labour of being "acceptable"

Loneliness is frequently narrated as being physically near others while feeling unseen, misunderstood, or incorrectly seen. This is not simply "no friends." It is misrecognition: others do not listen, do not understand, or interpret identity and behaviour through a narrow lens. The felt loneliness is the gap between one's lived experience and what is socially legible.

One pathway is invalidation. When participants try to describe complex friendship-group dynamics or distress, and adults dismiss it ("you'll make new friends"), the person can end up with no legitimate outlet, "only speaking to myself" (Interview 09). A second pathway is group dynamics: friendship groups "take sides," producing sudden outcasting and leaving the person watching others' togetherness from the edge (Interview 09). A third pathway is performance: having to dress, speak, or behave in a way that will be accepted, producing a split between a presented self and a self that can be "myself" only in particular niches (Interview 45).

Misrecognition is also narrated as an everyday interaction problem: participants describe being spoken over, being treated as less competent, or being evaluated through first impressions. In these accounts, loneliness is not only about "being alone" but about being present while feeling that one's perspective does not count or cannot be safely expressed. That dynamic can push people toward self-silencing ("I'll keep it to myself") and toward choosing environments where judgement costs are lower (Theme 3). The mechanism therefore links to place: when the social environment is high-judgement (workplace banter norms; status hierarchies; unfamiliar groups), loneliness emerges even in dense social settings.

Misrecognition also appears as competence judgement. In education and early work contexts, not knowing what to do (new software, unfamiliar expectations) combines with uncertainty about whether asking for help will be welcomed; the result can be loneliness as "stuckness" plus self-silencing ("who do I ask?"; fear of seeming stupid or incompetent). This is analytically distinct from general anxiety: it is an interaction between competence uncertainty and judgement risk that produces isolation in environments where support is needed.

Boundary conditions sharpen the claim. Close ties are not always the safest disclosure context. In Interview 38, friends are described as more judgemental than semi-strangers, making disclosure easier in low-stakes settings (gym/swimming regulars; hairdresser). This shifts the mechanism from "closeness reduces loneliness" to "relational safety reduces loneliness": who is safe to talk to depends on judgement costs and reputational stakes, not only intimacy.

Theme 2 — Place as constraint: crowded anonymity, threat appraisal, and mobility friction

The place-based task reveals that loneliness is not only psychological; it is spatially produced. Place structures which interactions are possible, safe, and emotionally sustainable.

One mechanism is crowded anonymity: being surrounded by strangers in transport or busy streets that does not translate into connection, and can intensify loneliness by making potential connections visible but unreachable. A second mechanism is threat appraisal in public space. Interview 41 describes parks and streets as uncertain and potentially unsafe; sitting alone on a bench becomes a scenario of scanning strangers and predicting motives. In that account, withdrawal is chosen as harm-minimisation: loneliness becomes preferable to fear ("better lonely than scared"), producing an avoidance loop that reduces immediate anxiety while sustaining isolation and adding counterfactual rumination ("what if he was nice?").

Third, mobility constraints are structural. Territorial violence can shrink feasible movement ("limitations for young people," Interview 26). Transport friction can make participation difficult (e.g., hard routes to campus; long commutes). Neighbourhood change (gentrification) can disrupt attachment and produce "disconnect" between groups (Interview 19). Conversely, walkability and local amenities can enable everyday connection by making "being out and about" feasible and familiar (Interview 13). These accounts treat isolation as an outcome of constrained feasible social space, not only an internal state.

The place mechanisms also interact with identity and judgement. For example, when a setting is experienced as racially mismatched or culturally unfamiliar, the social effort required to "fit in" can increase while the perceived payoff decreases, producing loneliness even when the environment is full of people. In such cases, "being in public" is not a neutral exposure to others; it is exposure to evaluation and uncertainty. This helps explain why some participants route their social lives into settings that are more normed and predictable (teams, faith communities, structured activities), and why some describe preference for spaces where they can be "with others" without heavy interaction demands.

Boundary conditions again matter. A negative-case set includes strong accounts of neighbourhood cohesion and mutual aid, countering a blanket "London low empathy" narrative. In these accounts, local recognisability (familiar shopkeepers, neighbours checking in) and shared-value communities (faith) make place protective rather than isolating. The revised claim is conditional: urban loneliness is shaped by uneven micro-ecologies of cohesion, safety, and infrastructure, not a single “city” essence.

Theme 3 — Connection infrastructure: structured third spaces, faith communities, and “known strangers”

Many accounts emphasise that connection becomes easier when social interaction is scaffolded by structure: shared purpose, predictable norms, and repeated contact that reduce the cost of initiating and sustaining interaction. This is not reducible to personality; participants often describe themselves as willing to connect, but needing the right conditions.

Several infrastructures recur:

- Structured third spaces with shared purpose (youth organisations, clubs, sports teams, parent/children centres) where people “want to be there,” roles are clear, and repeated participation builds familiarity.
- Faith communities (mosque, church) as a blend of shared values, ritual synchrony, and practical support/guidance. These spaces create belongingness through shared purpose while also lowering judgement risk (“everyone is the same,” “support if you need help”).
- Rule-bound shared-focus settings (e.g., movies) that allow co-presence with low performative demand (“be social without being social”).
- “Known strangers” (semi-regular contacts such as gym/swimming regulars or hairdresser) that enable disclosure precisely because reputational stakes are low. The relationship can be meaningful without becoming a deep friendship; what matters is low-judgement contact and repeated familiarity.

An important inference is that loneliness relief does not always require deep intimacy. For some accounts, the minimal unit of relief is recognisability, predictable co-presence, or safe disclosure rather than intense friendship. This widens the usual “make friends” framing into an infrastructure framing: the question becomes what kinds of spaces and norms reliably produce low-cost connection for young adults.

Two further refinements follow from this. First, infrastructure can be “thick” or “thin.” Thick infrastructures (teams, centres, faith communities) can produce ongoing ties and advice networks; thin infrastructures (everyday third spaces, known strangers) can still reduce loneliness through recognition and low-stakes talk. Second, infrastructure often works by reducing the performance burden: when it is clear what to do and how to be, people do not have to constantly manage impression and risk. This links back to Theme 1: where misrecognition and judgement risk dominate, infrastructure that lowers judgement costs becomes especially valuable.

Boundary conditions complicate venue-focused interventions. The same venue types (pubs, clubs, restaurants) can be narrated either as anchors (familiarity; laughter; “another home”) or as escapism that intensifies emptiness when ties are thin. The mechanism is not “going out,” but whether the space provides infrastructure properties (shared purpose, predictable norms, repeated contact, low judgement, safety).

Theme 4 — Digital and material filters: social media, dating apps, money

Digital platforms are narrated as dual mechanisms. Social media can sustain contact, humour, and belonging, but it can also generate “hyperreality” that intensifies comparison and exclusion (seeing friends together; curated happiness). Participants describe these moments as in-your-face, prompting self-questioning about worth and belonging. Dating apps appear as a distinct mechanism: interaction framed as evaluation and objectification, anticipation without care, and a sense of loneliness produced inside a “connection tool.”

Material conditions filter social life by shaping participation feasibility and friendship continuity. Money is unusually explicit in some accounts: poverty restricts joining activities and forces staying home, while changes in money/status can restructure friendship groups through activity mismatch and perceived selfishness. This is treated as a structural relational filter rather than a universal driver; it becomes analytically important because it links loneliness to what is practically feasible, not only what is emotionally desired.

Boundary conditions prevent overclaiming. Some participants describe social media as neutral background unless specific content triggers comparison; this cautions against treating “phones” as a universal cause.

5. Discussion: what changes when loneliness is treated as infrastructure + constraint

Taken together, the findings suggest that loneliness in these accounts is best explained as an interaction between recognition dynamics, place-based constraints/affordances, and access to connection infrastructures. This reframes common advice. “Go out more” is insufficient when public space is threat-appraised or mobility is constrained. “More venues” is insufficient when venues lack scaffolding and produce performance burden. “Home” is not uniformly protective; home-with-recognition differs from home-alone rumination. Digital connection is not uniformly harmful or helpful; it depends on whether platforms function as connection infrastructure or comparison amplifier.

The infrastructure framing suggests a different intervention logic: build low-cost, low-judgement, repeated-contact settings that young adults can access safely, while recognising that safety, transport, and material exclusion shape who can use those settings. The key design target is not “more interaction” but “safer and more predictable interaction with lower judgement costs.”

As a contribution, the analysis reframes loneliness as an outcome of interacting recognition dynamics, place constraints, and access to connection infrastructures, specifying boundary conditions under which common “fixes” succeed or fail. This is intended to complement (not replace) topic-based descriptions by adding mechanism detail that can inform practical design of neighbourhood and institutional interventions.

6. Limitations

This is a secondary analysis of interviews collected in 2019 in specific London borough contexts. Claims are interpretive and corpus-bound rather than statistical generalisations. The analysis relies on transcripts; non-verbal cues and situational detail are limited. Anonymity requirements also constrain reporting of contextual detail, though traceability is maintained via interview IDs and excerpt linkage in the evidence file. The analysis did not include respondent validation, and it is a single-analyst (AI-led) workflow; the paper therefore prioritises transparency of analytic decisions and explicit boundary cases over claims of consensus.

7. Conclusion

Across these interviews, loneliness is produced through misrecognition and performance, through place-based constraint (anonymity, threat appraisal, mobility friction), and through the presence or absence of connection infrastructures (structured third spaces, faith communities, rule-bound shared-focus settings, known-stranger ties). Digital and material factors operate as amplifiers and filters rather than universal causes. A practical implication is to design and support low-cost infrastructures for repeated, low-judgement contact while addressing safety and mobility constraints that gate access to connection.

… End of AI-written paper

Author's technical comments and auto-ethnographic reflection

The technical process: self-updating memos

Cursor (and similar tools) are normally used for writing computer code, but I have found it better than other tools like ChatGPT for collaborative writing: Cursor does not try to shorten or compress the text, and its workflow is optimised for complex, structured tasks, with a Plan mode and a Build mode. But the critical property of Cursor and similar tools for the purposes of the present paper is the ability to write, read and update its own memo files, which makes it a good candidate for the kind of iterative analysis necessary in Thematic Analysis and related approaches. (While any LLM "reasoning" model, or a tool like Gemini's Deep Research, is able to break its work into stages and may display a text commentary on its "reasoning", there is no way to audit what is *actually* going on, and there are no intermediate files to check.)

I enjoyed watching Cursor work, step by step creating and updating its memos. This produced an odd mix of feelings: fascination at the speed and apparent coherence, and unease at how easy it was to think “it looks systematic, so it must be right”.

The technical process: getting to final

The resulting paper was initially unsatisfactory in two ways: first, I realised I had not provided the review criteria, so I asked Cursor to update the paper to make sure the criteria were fulfilled; and second, Cursor's initial draft was too short, failing to count words correctly, so I had to ask it to write a longer version (with no other input from me).

This final, longer, "internal" paper has been left completely intact apart from removing a section about Python tools.

I checked the quotes provided in the paper against the sources and found a few errors, the worst being two short quotes: - “limitations for young people,” (Interview 26) -- the word is incorrect as a verbatim quote, but the source does discuss the topic; and “being out and about”, this appears in Interview 35, not Interview 13. I found no substantial errors. I also checked the methodology as described in the paper against the text files it produced, and found a few small errors, partly due to places where my original instructions were not clear:

- The journal was not maintained as completely as claimed in the paper

- The paper claims that a systematic negative-case pass was implemented, whereas the evidence trail looks more like the AI only checked a deliberate sample of sources.

Authorship

There are, in terms of task and role, two quite different papers. I am responsible for the final paper which “wraps” and includes the AI's paper. Formally, it looks like I'm the supervisor and this is a paper written by a student, but of course there is no student. The only real paper is the wrapper, a report of an AI experiment, not a paper about loneliness. I have no skin in the research-on-loneliness game, but I have skin in the AI-in-qualitative-research game.

No actual person is the author of the “inner” paper or could take responsibility for it, except in the sense that I set up the system, selected the sources and wrote the original high-level instructions. Otherwise, it is shaped just by the training data of this particular model, which is predominantly "WEIRD" (Western, Educated, Industrialized, Rich, and Democratic) (Atari et al. 2023) in outlook.

I chose these particular sources only because they were good-quality, publicly available and on an important topic. I did instruct the AI to begin with a web search of the current literature around the theme of loneliness, but this was relatively cursory.

Theory building?

In my day job I make extensive use of AI assistance in a practice which is built around *causal mapping* as a way of making sense of large corpora of text (Powell & Caldas Cabral 2025). In this workflow, "creative" tasks with high degrees of freedom are restricted to specific parts of an otherwise clearly defined and reproducible workflow. Against this background, I was intrigued by recent suggestions for how to engage an AI as a co-researcher in more explorative conversational studies (Friese 2025.; Dai et al. 2023; Nguyen-Trung 2025). I agree that for genuine exploratory work it is not enough to give the AI a monolithic prompt such as in the attempt by Jowsey et al. (2025, p.5). If you want iterative theory-building, you need an iterative, theory-building workflow. One way to iterate is conversationally. In this paper we **let the AI create and iteratively update its own methodology**, based on an extremely casual selection of three methods papers which had recently interested me, plus one I had contributed to.

I do not argue that this AI-only methodology is “better” than conversational approaches. As a method it is still only a limited version of a thematic analysis as a human would conduct it, quite apart from the fact that the AI does not exist as a subject with any **role** as academic, student or stakeholder, has no skin in the game and could not actually submit a paper (this has nothing to do with practical or philosophical **limitations** of LLMs, it’s about **roles**).

Future improvements

Others may explore the GitHub repo and adapt it, for example:

- Introduce more explicit automatic quality checks (checking quotes, checking that the memo files were really written and then used as the AI claims).
- Give the AI **less freedom**: starting straight off with some variation of the workflow which the AI finally arrived at in this paper, or some variant based on substantive considerations.
- Give the AI **more freedom**, for example suggesting that it could periodically decide when its evolving theory would benefit from additional primary material, and then identify and download suitable publicly available sources and continue with the analysis.

References

- Atari, Xue, Park, Blasi, & Henrich (2023). *Which Humans?*. PsyArXiv. https://hmpa.hms.harvard.edu/sites/projects.iq.harvard.edu/files/culture_cognition_coevol_lab/files/which_hu
- Braun, V., & Clarke, V. (2023). *Toward good practice in thematic analysis: Avoiding common problems and be(com)ing a knowing researcher*. *International Journal of Transgender Health*, 24(1), 1–6. doi: 10.1080/26895269.2022.2129597
- Dai, Xiong, & Ku (2023). *LLM-in-the-loop: Leveraging Large Language Model for Thematic Analysis*. https://arxiv.org/abs/2310.15100v1.
- Fardghassemi, S., & Joffe, H. (2022). *Qualitative and output data on loneliness among young adults* (p. 15430034 Bytes) [Dataset]. University College London. https://doi.org/10.5522/04/17212991
- Friese (2025). *Conversational Analysis with AI - CA to the Power of AI: Rethinking Coding in Qualitative Analysis*. https://doi.org/10.2139/ssrn.5232579.

- Friese, S., Nguyen-Trung, K., Powell, S., & Morgen, D. (2025). *Beyond binary positions: Making space for critical and reflexive GenAI integration in qualitative research*.

- Jowsey, Braun, Clarke, Lupton, & Fine (2025). *We Reject the Use of Generative Artificial Intelligence for Reflexive Qualitative Research*. https://doi.org/10.2139/ssrn.5676462.

- Morgan, D. (2025). *Query-Based Analysis: A Strategy for Analyzing Qualitative Data Using ChatGPT*.

- Nguyen-Trung (2025). *ChatGPT in Thematic Analysis: Can AI Become a Research Assistant in Qualitative Research?*. https://doi.org/10.1007/s11135-025-02165-z.

- Nguyen-Trung, & Friese (2025). *On Methodological Incongruence in Applying Generative AI in Qualitative Data Analysis*. https://doi.org/10.2139/ssrn.5874482.

- Powell, & Caldas Cabral (2025). *AI-assisted Causal Mapping: A Validation Study*. Routledge. https://doi.org/10.1080/13645579.2025.2591157.