Coding context and mechanisms

22 Sep 2025

Summary

In evaluation work, people often make **context-dependent** causal claims:

> "When X holds, A leads to B."

If you code only "A → B", you may lose what makes the claim true *in this setting*.

This page is about simple, practitioner-friendly ways to keep that information while the causal map stays minimalist (links between factors):

- record context as **tags on links**, so you can filter and compare later (the first part of the page);
- or code it as one more causal factor, or as a tag in a factor label (the second part, which also covers mechanisms);
- and, for the rare claims where one factor enables or blocks the link between two others, see enablers and blockers.

Context as a tag on the link

One option is to treat context as **metadata** about a claim, not necessarily as another causal factor.

Example claim:

- "When enough oxygen is present, a spark will always cause a flame."

You might still code the causal link:

- `Spark → Flame`

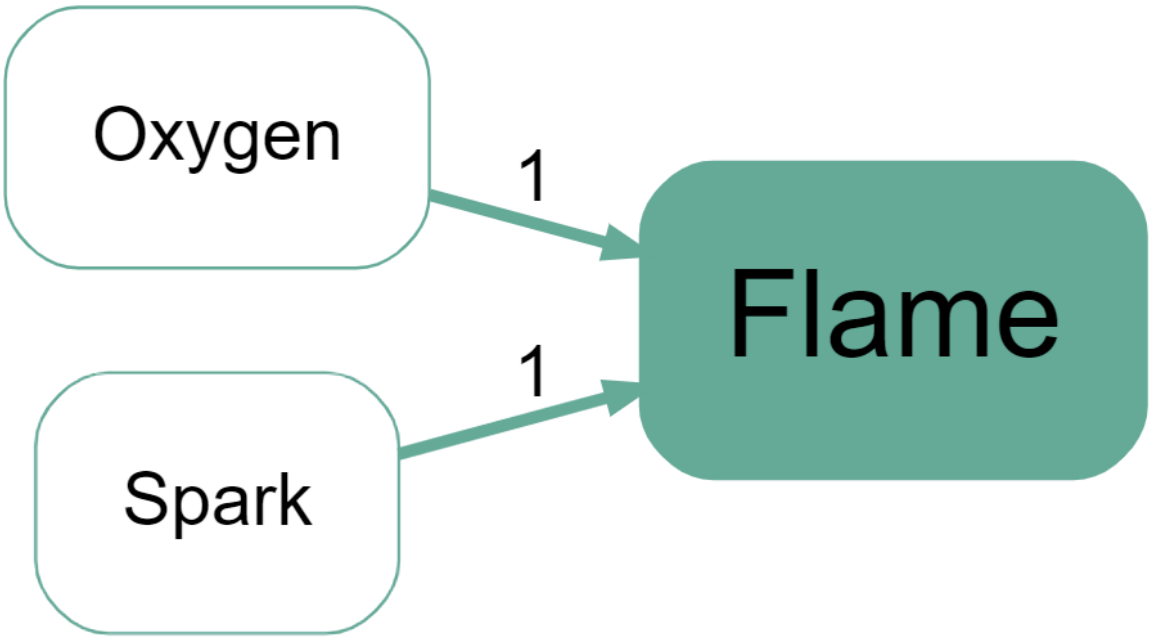
...but tag it with a context tag, e.g.:

- `Context: oxygen present`

That way you can later:

- show only links tagged with that context, or

- compare “with context” vs “without context” views.



Using context in the Causal Map app

You can encode context by adding **tags** to links (just like other link tags).

A simple convention that works well is to prefix them, e.g.:

- `Context: oxygen`
- `Context: rainy season`
- `Context: only for village X`



Then you can use the include/exclude tags filters to show only links with particular contexts (or to hide them).

Formal notes (optional)

One reason context feels different from “just another cause” is that we often do not have (or do not claim) information about what happens when the context is absent.

In the oxygen example, we know how Spark relates to Flame *given* Oxygen, but not what happens without Oxygen. This can be expressed as a partial truth table:

Oxygen	Spark	Flame
Yes	Yes	Yes
Yes	No	No
No	Yes	?
No	No	?

This is one motivation for treating “context” as enabling conditions or metadata rather than forcing it into the same causal-factor role as the other nodes.

Context and mechanism in minimalist coding

Evaluators will recognise two questions that any causal method has to answer. How do you code context? And what do you do about mechanisms? Both come up in realist evaluation and its Context-Mechanism-Outcome formula (Pawson & Tilley 1997). Minimalist coding answers both the same way. It gives them no special machinery and treats them as ordinary parts of the causal map. This is primitive and it flattens distinctions which can be important, but it is *relatively* easy to apply, and the results are not obviously worse than any more sophisticated approach.

Context as a factor, a link tag or a factor tag

Context is the standard worry. The same intervention has different effects in different places, and a good evaluation should say so. The realist instinct is to treat context as a separate kind of thing, a backdrop or a set of enabling conditions that decides whether a mechanism fires, distinct from the causes themselves.

One minimalist move is to code context as one more causal factor. If a source says the training worked because the community already trusted the facilitators, then `community already trusted the facilitators` is a factor, and it influenced the outcome, coded as an ordinary link. Nothing new is needed.

This is primitive. It drops the special status that context is often given, and it does not by itself mark the difference between a cause that triggers an effect and a condition that merely lets some other cause work. Where that difference really matters, it is the second-order problem of enablers and blockers, and we handle it the same way, usually in the write-up.

Coding context as a factor is not the only option. You can record it as link metadata, as described above (a column in the table of links, or in particular a dedicated link tag like `#afterCovid` or `#onlyInSevereCases`), or as a factor tag (`Trust [afterCovid]`). We use each of these where it fits.

There is no universally accepted way to encode context in realist evaluation, perhaps because there is no even half-way accepted operational agreement about how to distinguish triggers, contexts and mechanisms. Realist evaluation has argued for decades about what counts as context as opposed to mechanism or outcome. Yes, these concepts can be usefully applied in a given case or study. But there are different ways to apply them. That is a strength as well as a problem. But the problem remains that you can't just give a set of instructions on how to encode them and be sure that they will be applied in the same way by different assistant coders (human or AI). Given that, coding context as a plain factor is a reasonable default. It is simple, and it defers to the analysis phase the harder judgement about what role context is playing. Where a source states the condition explicitly ("when X holds, A leads to B"), the link tag described above keeps it without making that judgement.

Context and source

The idea of 'Context' also overlaps with the idea of 'Source'. *True in this context* is not so far from *True for this source*, and in our minimalist approach we have some surprisingly sophisticated ideas about dealing with what *True for this source* means when tracing causal influences upstream and downstream, see source tracing. In other words, to the extent that *Source* is for you a reasonable proxy for *Context*, then you can use the fact that all your source metadata is already available to you in your links dataset, for example you can add a filter to show only causal stories which are *true for these particular sources*.

The caution behind source tracing applies to any property of a whole network, and not only to paths. From "Factor X is central within these claims" can we deduce "Factor X is claimed to be central"? In general, no. If we ask parents and children about the causes of family disputes, we might get two relatively separate causal networks with only a little overlap. We cannot conclude that these respondents, taken together, claim that there are two relatively separate systems: parents and children may each be describing the part they know best, or the two groups may be telling conflicting and perhaps incompatible stories. We could express this as a rule:

> 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.

We often assume that contexts are sources and sources are contexts, but this is not always the case. One respondent might give two sets of information, one from before losing their job and one from afterwards, without encoding the job loss as a causal factor. And in a participatory mapping workshop there may be many respondents, but as long as they construct a consensus map they are treated as one source. See also Sources, cases and context.

Mechanisms

The CMO formula raises a sharper question that the realist literature has never really settled, even after a dedicated review of how realist evaluations use the term (Lemire et al. 2020): what is a mechanism? Definitions range over hidden generative processes, the reasoning and resources of actors, underlying social structures, and several other things, and a coder who has to decide which one applies before coding anything will not get far.

Minimalist coding avoids the definition. A mechanism is just a piece of the causal map. If context factors E and F lead to outcomes X and Y, possibly through intermediate factors M and N, then the mechanism is that part of the graph: the links from E and F through M and N to X and Y. There is nothing extra to encode and no separate object called a mechanism. You read it off the map by tracing paths, with the same caution about combining links across sources discussed under source tracing.

This too is primitive and flattening. It does not capture the generative, what-is-really-going-on-underneath sense that realists want from the word mechanism, and it will not satisfy a reader who thinks a mechanism must be more than the connections people described. But it works. It gives you a concrete, checkable object made of coded claims with quotes, where the realist account gives you a long-running argument about definitions. If a richer notion of mechanism is needed for a particular study, it can be built on top, the same way polarity can. The minimalist record does not block that, and it is a usable starting point in the meantime.

I sometimes use AI to help draft these pages, but the final text is ours.

References

Lemire, Kwako, Nielsen, Christie, Donaldson, & Leeuw (2020). *What Is This Thing Called a Mechanism? Findings from a Review of Realist Evaluations*. Causal Mechanism in Evaluation, 73--86. https://doi.org/10.1002/ev.20428.

Pawson, & Tilley (1997). *Realistic Evaluation*. Sage Publications Limited.