

The causalmodels package

Alex Markham
alex.markham@causal.dev

2026/08/22 v1.0.0

Quickstart

Loading the package with `\usepackage{causalmodels}` provides macros for common operators, relations, and edges. For example, some common operators include

$$\begin{array}{llll} \text{pa}(u) & \backslash\text{pa}\{u\} & \text{ch}(u) & \backslash\text{ch}\{u\} \\ \text{ne}(u) & \backslash\text{nei}\{u\} & \text{do}(X) & \backslash\text{doo}\{X\}, \end{array}$$

separation relations include

$$X \perp\!\!\!\perp Y \quad \backslash\text{indep}\{X\}\{Y\}\{\}, \quad X \not\perp\!\!\!\perp Y \mid Z \quad \backslash\text{nindep}\{X\}\{Y\}\{Z\},$$

and edge types include

$$\begin{array}{llll} u \text{---} v & u \backslash\text{undir} v & u \text{--}\text{>} v & u \backslash\text{rightdir} v \\ u \text{--}\text{<} v & u \backslash\text{leftdir} v & u \text{<--} v & u \backslash\text{bidir} v. \end{array}$$

See Section 2 for more macros and usage details.

Contents

1	Introduction	2
2	Usage	2
2.1	Graph operators	2
2.2	The do-operator	3
2.3	Separation relations	3
2.4	Edge marks	3
3	Implementation	4
3.1	Dependencies	4
3.2	Graph operators	5
3.3	The do-operator	5
3.4	Separation relations	5
3.5	Configuration: <code>\causalmodelsset</code>	7
3.6	The <code>\mkedge</code> helper	7
3.7	Edge marks	7
3.8	Diagram styles	8

1 Introduction

This package is a collection of macros for notation and symbols commonly used in the causal graphical models literature. On the one hand, these macros are relatively simply built from tools in `amsmath` and `PGF/TikZ`. On the other hand, I find myself redefining these macros almost every time I start a new `LATEX` document, and I regularly read new papers that (should) do the same, so I expect collecting them into a single `TEX` package to be useful nevertheless.

Besides typesetting common operators like $\text{pa}_G(u)$ properly using `amsmath`'s `\operatorname`—which is subtly different from using `\mathrm{pa}`—and more noticeably so from plain `$pa$`—this package makes three modest innovations: it uses `PGF/TikZ` to define (i) a well-behaved independence symbol ($\perp$) without the issues of other approaches¹ and (ii) a set of edges with uniform width when used inline (like $\rightarrow$ and $\circ\rightarrow$), and it (iii) provides two macros for simply and consistently defining new operators and edges.

If you find yourself regularly redefining a macro because it is not in this package—or modifying one that is part of this package—please send me an email² or open an issue/make a pull request³ describing it.

2 Usage

This section covers the package's four macro families—graph operators, the do-operator, separation relations, and edge marks—after loading it:

```
% \usepackage{causalmodels}
%
```

It requires `amsmath`, `tikz` (with the `arrows.meta` library), and `centernot`.

2.1 Graph operators

Each of the eleven graph operators below (e.g., `\pa`) shares one syntax,

$$\langle op \rangle[\langle graph \rangle][\{\langle vertex \rangle\}],$$

which is unary and prefix, takes a vertex (set) as its argument, plus an optional leading subscript (any graph label, e.g., `\pa[G]{u}`) to disambiguate between multiple graphs:

<code>pa(u)</code>	<code>\pa{u}</code>	<code>ch(u)</code>	<code>\ch{u}</code>	<code>an(u)</code>	<code>\an{u}</code>
<code>de(u)</code>	<code>\de{u}</code>	<code>nd(u)</code>	<code>\nd{u}</code>	<code>mb(u)</code>	<code>\mb{u}</code>
<code>ne(u)</code>	<code>\nei{u}</code>	<code>sp(u)</code>	<code>\spo{u}</code>	<code>dis(u)</code>	<code>\dis{u}</code>
<code>ant(u)</code>	<code>\ant{u}</code>	<code>cc(u)</code>	<code>\cc{u}</code>		

¹For example, approaches suggested at <https://tex.stackexchange.com/q/719657> tend to require using other fonts and can ignore scaling of surrounding math text; and the overlapping `\perp` approach discussed at <https://tex.stackexchange.com/q/79434> and in the comprehensive package documentation at <https://ctan.org/pkg/comprehensive>, with attribution to a post by Donald Arseneau on `comp.text.tex` in June 2000, produces a slightly thicker stroke weight where the symbols overlap.

²alex.markham@causal.dev

³<https://codeberg.org/alex-markham/causalatex>

In addition to the usual `\nei` (neighbors), `\pa` (parents), `\ch` (children), `\an` (ancestors), and `\de` (descendants), we have `\nd` (non-descendants) and `\mb` (Markov blanket) [3], `\spo` (spouse) and `\ant` (anterior) [7], `\dis` (district) [6], and `\cc` (chain component) [1].

For an operator not named above, `\newgraphop{⟨macro⟩}{⟨name⟩}` defines `⟨macro⟩` as a new operator, with the same syntax as above, that displays as `⟨name⟩`: `\newgraphop{\eg}{eg}` makes `\eg[G]{u}` available.

2.2 The do-operator

Pearl’s `do(·)` notation [4] is typeset with `\doo{⟨variable⟩}`, which is unary, prefix, and takes no optional argument:

$$\text{do}(X) \quad \text{\doo{X}}$$

2.3 Separation relations

Each of the four separation relations below (e.g., `\sep`) shares one syntax,

$$\langle relation \rangle [\langle sub \rangle] \{ \langle X \rangle \} \{ \langle Y \rangle \} \{ \langle Z \rangle \},$$

which is ternary and prefix, asserting that `⟨X⟩` and `⟨Y⟩` are (conditionally) separated given `⟨Z⟩` [3]. An optional leading subscript names the distribution, graph, or criterion it’s relative to, e.g., `\indep[P]{X}{Y}{Z}`, or using `m` or `\sigma` as `⟨sub⟩` in `\sep` to denote *m*-separation [7] or *σ*-separation [2]. Leaving `⟨Z⟩` empty (`{}`) gives the unconditional form, omitting the conditioning bar entirely.

$$\begin{array}{lll} X \perp\!\!\!\perp Y \mid Z & \text{\indep{X}{Y}{Z}} & u \perp v \mid w \quad \text{\sep{u}{v}{w}} \\ X \not\perp\!\!\!\perp Y \mid Z & \text{\nindep{X}{Y}{Z}} & u \not\perp v \mid w \quad \text{\nsep{u}{v}{w}} \end{array}$$

2.4 Edge marks

Edge marks (used across DAGs [3], CPDAGs [1], MPDAGs [5], ADMGs [6], ancestral graphs [7], and PAGs [8]) use `⟨u⟩ ⟨name⟩ ⟨v⟩`, which is binary, infix notation, built from four end marks at each end, so there are 4^2 different marks:

$$\begin{array}{lll} u \text{---} v & u \text{\tailtail} v & u \text{\rightarrow} v \quad u \text{\tailhead} v \\ u \text{\--}\text{\--} v & u \text{\tailcirc} v & u \text{\--}\ast v \quad u \text{\tailwild} v \\ u \text{\leftarrow} v & u \text{\headtail} v & u \text{\leftrightarrow} v \quad u \text{\headhead} v \\ u \text{\leftarrow}\text{\--} v & u \text{\headcirc} v & u \text{\leftarrow}\ast v \quad u \text{\headwild} v \\ u \text{\circ}\text{\--} v & u \text{\circtail} v & u \text{\circ}\text{\rightarrow} v \quad u \text{\circhead} v \\ u \text{\circ}\text{\--}\text{\--} v & u \text{\circcirc} v & u \text{\circ}\ast v \quad u \text{\circ}\text{\circ}\ast v \\ u \ast\text{\--} v & u \text{\wildtail} v & u \ast\text{\rightarrow} v \quad u \text{\wildhead} v \\ u \ast\text{\--}\text{\--} v & u \text{\wildcirc} v & u \ast\ast v \quad u \text{\wildwild} v \end{array}$$

Four of the sixteen also have short, colloquial aliases—either name draws the same edge:

$$\begin{array}{lll} u \text{---} v & u \text{\undir} v & \text{or } u \text{\tailtail} v \\ u \text{\rightarrow} v & u \text{\rightdir} v & \text{or } u \text{\tailhead} v \\ u \text{\leftarrow} v & u \text{\leftdir} v & \text{or } u \text{\headtail} v \\ u \text{\leftrightarrow} v & u \text{\bidir} v & \text{or } u \text{\headhead} v \end{array}$$

Every edge mark above also has a matching `causalmodels/⟨name⟩` TikZ style, for drawing edges between arbitrarily-positioned nodes in a real `tikzpicture` rather than typesetting a fixed-length inline glyph:

```
% \usetikzlibrary{positioning}
% \begin{tikzpicture}
%   \node (u) {$u$};
%   \node (v) [right=1.5cm of u] {$v$};
%   \draw[causalmodels/cirthead] (u) -- (v);
% \end{tikzpicture}
%
```

See Section 3.8 for the full set of style names.

Arrow length and width—for both inline glyphs and diagram styles—are configurable via `\causalmodelsset{arrow-length=⟨len⟩, arrow-width=⟨wid⟩}`. For an edge type not named above, `\mkedge{⟨arrows.meta spec⟩}` builds one directly, e.g., `\mkedge{->}` is exactly `\rightrightarrows`.

3 Implementation

This section documents how each macro from Section 2 is built. It also covers `\causalmodelsset`, plus `\newgraphop` and `\mkedge`, the helpers for custom operators and edge types.

```
1 (*package)
2 \NeedsTeXFormat{LaTeX2e}
3 \ProvidesPackage{causalmodels}%
4 [2026/08/22 v1.0.0 Notation macros for causal graphical models]
5 \end{package}
```

3.1 Dependencies

`amsmath` provides `\operatorname` and `\mathrel`, used throughout the graph operators, do-operator, and separation relations. `tikz` draws every edge glyph and diagram style; `arrows.meta` specifically supplies the circle and wildcard tip shapes (`\Circle[open]`, `\Rays[n=7]`) used by the edge marks (Section 3.7):

```
6 (*package)
7 \RequirePackage{amsmath}
8 \RequirePackage{tikz}
9 \usetikzlibrary{arrows.meta}
10 \end{package}
```

`\centernot` is needed for `\nindep`’s negation slash: plain `\not` sizes and positions its slash assuming a normal single-character-width symbol, and lands off-center on `\indep`’s wider hand-drawn glyph (Section 3.4). `\centernot` measures the actual width of its argument and centers accordingly:

```
11 (*package)
12 \RequirePackage{centernot}
13 \end{package}
```

3.2 Graph operators

`\pa` All eleven graph operators share `\causalmodels@graphop`, which takes an operator name, an optional leading subscript, and one required argument.

`\an` `\nei`, not `\ne`: `\ne` is already `amsmath`’s “not equal” ($\neq$)—only the callable name avoids the collision; it still typesets “ne” (`\operatorname{ne}`). Likewise

`\nd` `\spo`, not `\sp`: `\sp` is already plain `TeX`’s superscript primitive ($\sup$). It still typesets

`\mb` “sp” (`\operatorname{sp}`).

```
\nei 14 (*package)
\spo 15 \newcommand{\causalmodels@graphop}[3]{%
\dis 16 \def\causalmodels@graphopsub{#2}%
\ant 17 \operatorname{#1}\ifx\causalmodels@graphopsub\empty\else_{#2}\fi(#3)%
\cc 18 }
19 \newcommand{\pa}[2][]{\causalmodels@graphop{pa}{#1}{#2}}
20 \newcommand{\ch}[2][]{\causalmodels@graphop{ch}{#1}{#2}}
21 \newcommand{\an}[2][]{\causalmodels@graphop{an}{#1}{#2}}
22 \newcommand{\de}[2][]{\causalmodels@graphop{de}{#1}{#2}}
23 \newcommand{\nd}[2][]{\causalmodels@graphop{nd}{#1}{#2}}
24 \newcommand{\mb}[2][]{\causalmodels@graphop{mb}{#1}{#2}}
25 \newcommand{\nei}[2][]{\causalmodels@graphop{ne}{#1}{#2}}
26 \newcommand{\spo}[2][]{\causalmodels@graphop{sp}{#1}{#2}}
27 \newcommand{\dis}[2][]{\causalmodels@graphop{dis}{#1}{#2}}
28 \newcommand{\ant}[2][]{\causalmodels@graphop{ant}{#1}{#2}}
29 \newcommand{\cc}[2][]{\causalmodels@graphop{cc}{#1}{#2}}
30 \end{package}
```

`\newgraphop` `\newgraphop{<macro>}{<name>}` defines a new graph operator, for one this package doesn’t already name: `<macro>` is the macro it defines, and `<name>` is the string `\operatorname` displays. `\newgraphop{\eg}{eg}` makes `\eg[G]{u}` available, built the same way `\pa` is.

```
31 (*package)
32 \newcommand{\newgraphop}[2]{%
33 \newcommand{#1}[2][]{\causalmodels@graphop{#2}{##1}{##2}}%
34 }
35 \end{package}
```

3.3 The do-operator

Unlike the graph operators and separation relations, `\doo` needs no shared helper—it’s a single `\operatorname` call:

```
36 (*package)
37 \newcommand{\doo}[1]{\operatorname{do}(#1)}
38 \end{package}
```

3.4 Separation relations

`\indep` `\indep`, `\nindep`, `\sep`, and `\nsep` all share `\causalmodels@sepop`, which takes a glyph, an optional leading subscript, and three required arguments—the third left empty omits the “mid”.

`\nsep` `\indep`’s glyph is built the same way $\models$ relates to $\vdash$: $\vdash$ is a single vertical spine with one centered horizontal tick; $\models$ is the same spine with two horizontal ticks straddling that center. `\indep` applies that to $\perp$: its horizontal bar, with

the one centered vertical tick replaced by two ticks straddling the same center, at the same offset ratio (measured from T_EX's own $\vdash$ font outlines). The glyph is drawn directly from $\perp$'s own measured dimensions, remeasured on every use, so it tracks the surrounding font size, math style, and whatever math font is loaded.

```

39 (*package)
40 \newcommand{\causalmodels@sepop}[5]{%
41   \def\causalmodels@sepopsub{#2}%
42   \def\causalmodels@sepopz{#5}%
43   #3 \mathrel{#1\ifx\causalmodels@sepopsub\empty\else_{#2}\fi} #4%
44   \ifx\causalmodels@sepopz\empty\else\mid #5\fi
45 }
46 \newsavebox{\causalmodels@perpbox}
47 \newlength{\causalmodels@perpht}
48 \newlength{\causalmodels@perpwd}
49 \newlength{\causalmodels@indeptickoffset}
50 \newlength{\causalmodels@indepbarmid}
51 \newlength{\causalmodels@indeptick}
52 \newcommand{\causalmodels@indeptickoffsetratio}{0.1398}
53 \newcommand{\causalmodels@indepstrokewidth}{0.04em}
54 \newcommand{\causalmodels@indepstyled}[2]{%
55   \sbox{\causalmodels@perpbox}{\$#1\perp$}%
56   \settoheight{\causalmodels@perpht}{\usebox{\causalmodels@perpbox}}%
57   \settowidth{\causalmodels@perpwd}{\usebox{\causalmodels@perpbox}}%
58   \pgfmathsetlength{\causalmodels@indeptickoffset}%
59   {\causalmodels@indeptickoffsetratio*\causalmodels@perpwd}%
60   \pgfmathsetlength{\causalmodels@indepbarmid}{0.5\causalmodels@perpwd}%
61   \mathrel{\tikz[baseline=0pt]{%
62     \draw[line width=\causalmodels@indepstrokewidth]%
63       (0pt,0pt) -- (\causalmodels@perpwd,0pt);%
64     \pgfmathsetlength{\causalmodels@indeptick}%
65     {\causalmodels@indepbarmid-\causalmodels@indeptickoffset}%
66     \draw[line width=\causalmodels@indepstrokewidth]%
67       (\causalmodels@indeptick,0pt) --%
68       (\causalmodels@indeptick,\causalmodels@perpht);%
69     \pgfmathsetlength{\causalmodels@indeptick}%
70     {\causalmodels@indepbarmid+\causalmodels@indeptickoffset}%
71     \draw[line width=\causalmodels@indepstrokewidth]%
72       (\causalmodels@indeptick,0pt) --%
73       (\causalmodels@indeptick,\causalmodels@perpht);%
74   }}%
75 }
76 \newcommand{\causalmodels@indepglyph}%
77 {\mathpalette\causalmodels@indepstyled\relax}
78 \newcommand{\causalmodels@nindepglyph}%
79 {\centernot{\causalmodels@indepglyph}}
80 \newcommand{\causalmodels@nsepglyph}{\centernot{\perp}}
81 \newcommand{\indep}[4]{}%
82 {\causalmodels@sepop{\causalmodels@indepglyph}{#1}{#2}{#3}{#4}}
83 \newcommand{\nindep}[4]{}%
84 {\causalmodels@sepop{\causalmodels@nindepglyph}{#1}{#2}{#3}{#4}}
85 \newcommand{\sep}[4]{}{\causalmodels@sepop{\perp}{#1}{#2}{#3}{#4}}
86 \newcommand{\nsep}[4]{}%
87 {\causalmodels@sepop{\causalmodels@nsepglyph}{#1}{#2}{#3}{#4}}
88 (/package)

```

3.5 Configuration: `\causalmodelsset`

`\causalmodelsset` Its `arrow-length` and `arrow-width` keys resolve where the edge is actually drawn, not where `\causalmodelsset` is called, so they always match its surrounding font size:

$$u \longrightarrow v \longleftrightarrow w \circ \longrightarrow x$$

$$\mathcal{U} \longrightarrow \mathcal{V} \longleftrightarrow \mathcal{W} \circ \longrightarrow \mathcal{X}$$

```

89 (*package)
90 \newcommand{\causalmodels@arrowlength}{2.5ex}
91 \newcommand{\causalmodels@arrowwidth}{0.045em}
92 \pgfkeys{
93   /causalmodels/.cd,
94   arrow-length/.store in=\causalmodels@arrowlength,
95   arrow-width/.store in=\causalmodels@arrowwidth,
96 }
97 \newcommand{\causalmodelsset}[1]{\pgfkeys{/causalmodels/.cd,#1}}
98 \end{package}

```

3.6 The `\mkedge` helper

`\mkedge` Every edge mark below is built from `\mkedge{\langle arrows.meta spec \rangle}`: a short TikZ line with that tip decoration, wrapped as an inline math relation.

```

99 (*package)
100 \newcommand{\causalmodels@mkedge}[1]{%
101   \mathrel{\tikz[baseline=-0.5ex]{%
102     \draw[line width=\causalmodels@arrowwidth,#1]{%
103       (0,0) -- (\causalmodels@arrowlength,0);%
104     }}%
105 }
106 \newcommand{\mkedge}[1]{\ensuremath{\causalmodels@mkedge{#1}}}
107 \end{package}

```

`\causalmodels@defedge{name}{spec}` defines both the inline glyph and the matching `causalmodels/name` diagram style (Section 3.8) from one spec.

```

108 (*package)
109 \newcommand{\causalmodels@defedge}[2]{%
110   \expandafter\newcommand\csname #1\endcsname{\mkedge{#2}}%
111   \tikzset{causalmodels/#1/.style={#2,line width=\causalmodels@arrowwidth}}%
112 }
113 \end{package}

```

3.7 Edge marks

Every edge mark, regardless of how many marks decorate its ends, spans the same tip-to-tip width, so a plain `\undir` line and a doubly-decorated `\wildwild` line drawn to the same length look the same length, even though `\wildwild`'s decorations eat into more of that span. Every edge mark works in text and math mode.

Every edge is named compositionally, left-to-right, from a fixed vocabulary: **tail** (a plain line end), **head** (an arrowhead), **circ**, and **wild** (a wildcard mark, standing in for any of the other three).

For example, `\circhead` is circle-then-right-arrow and `\headcirc` is left-arrow-then-circle. Sixteen edges in total: every pairing of the four end marks (tail, head, circ, wild) at the two ends of the edge, $4^2 = 16$.

```

\headhead 114 (*package)
\tailcirc 115 \causalmodels@defedge{tailtail}{-}
\tailwild 116 \causalmodels@defedge{tailhead}{->}
\headcirc 117 \causalmodels@defedge{headtail}{<-}
\headwild 118 \causalmodels@defedge{headhead}{<->}
\circtail 119 \causalmodels@defedge{tailcirc}{-{Circle[open]}}
\circhead 120 \causalmodels@defedge{tailwild}{-{Rays[n=7]}}
\circcirc 121 \causalmodels@defedge{headcirc}{<-{Circle[open]}}
\circwild 122 \causalmodels@defedge{headwild}{<-{Rays[n=7]}}
\circwild 123 \causalmodels@defedge{circtail}{{{Circle[open]}}-}
\wildtail 124 \causalmodels@defedge{circhead}{{{Circle[open]}}->}
\wildhead 125 \causalmodels@defedge{circcirc}{{{Circle[open]}}-{Circle[open]}}
\wildcirc 126 \causalmodels@defedge{circwild}{{{Circle[open]}}-{Rays[n=7]}}
\wildwild 127 \causalmodels@defedge{wildtail}{{{Rays[n=7]}}-}
          128 \causalmodels@defedge{wildhead}{{{Rays[n=7]}}->}}
          129 \causalmodels@defedge{wildcirc}{{{Rays[n=7]}}-{Circle[open]}}
          130 \causalmodels@defedge{wildwild}{{{Rays[n=7]}}-{Rays[n=7]}}

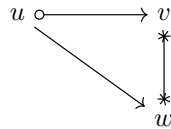
\undir    Four of the sixteen also get short, colloquial aliases ending in dir—exactly
\rightdir equivalent to the names above: \undir, \rightdir, \leftdir, \bidir.
\leftdir  131 \let\undir\tailtail
\bidir    132 \let\rightdir\tailhead
          133 \let\leftdir\headtail
          134 \let\bidir\headhead
          135 (/package)

```

3.8 Diagram styles

The edge marks above are complete, self-contained inline glyphs, not path styles, so they cannot be dropped into `\draw (u) - (v)`; inside a real `tikzpicture` to connect two arbitrarily-positioned nodes. The styles below, namespaced under `causalmodels/`, expose the same arrow-tip specs and configurable line width instead (`\causalmodelsset` affects diagrams and inline glyphs together, so a figure can match a paper’s inline notation exactly).

One style exists per edge mark above (including the four colloquial aliases), by the same name: `causalmodels/tailhead`, `causalmodels/circhead`, The sixteen non-alias styles are defined alongside their inline macros in Section 3.7, via `\causalmodels@defedge`; only the four alias styles below are defined directly here. Diagram styles deliberately don’t include arrow length—unlike the inline glyphs, diagram edges span whatever distance node positions dictate. Compare against the inline glyphs directly: $u \circ \rightarrow v$, $u \rightarrow w$, $v ** w$.



Alias styles work the same way.

```
136 (*package)
```

```

137 \tikzset{
138   causalmodels/undir/.style={causalmodels/tailtail},
139   causalmodels/rightdir/.style={causalmodels/tailhead},
140   causalmodels/leftdir/.style={causalmodels/headtail},
141   causalmodels/bidir/.style={causalmodels/headhead},
142 }
143 \end{package}
144 \endinput
145 \endinput
146 \end{package}

```

References

- [1] Steen A. Andersson, David Madigan, and Michael D. Perlman. A characterization of Markov equivalence classes for acyclic digraphs. *The Annals of Statistics*, 25(2):505–541, 1997. doi: [10.1214/aos/1031833662](https://doi.org/10.1214/aos/1031833662).
- [2] Patrick Forré and Joris M. Mooij. Markov properties for graphical models with cycles and latent variables. *arXiv preprint*, 2017. doi: [10.48550/arXiv.1710.08775](https://doi.org/10.48550/arXiv.1710.08775).
- [3] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988. URL [https://archive.org/details/probabilisticrea00pear](https://archive.org/details/probabilisticrea00pear/details/probabilisticrea00pear).
- [4] Judea Pearl. Causal diagrams for empirical research. *Biometrika*, 82(4):669–688, 1995. doi: [10.1093/biomet/82.4.669](https://doi.org/10.1093/biomet/82.4.669).
- [5] Emilija Perković. Identifying causal effects in maximally oriented partially directed acyclic graphs. In *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 530–539, 2020. URL <https://proceedings.mlr.press/v124/perkovic20a.html>.
- [6] Thomas S. Richardson. Markov properties for acyclic directed mixed graphs. *Scandinavian Journal of Statistics*, 30(1):145–157, 2003. doi: [10.1111/1467-9469.00323](https://doi.org/10.1111/1467-9469.00323).
- [7] Thomas S. Richardson and Peter Spirtes. Ancestral graph markov models. *The Annals of Statistics*, 30(4):962–1030, 2002. doi: [10.1214/aos/1031689015](https://doi.org/10.1214/aos/1031689015).
- [8] Peter Spirtes, Christopher Meek, and Thomas S. Richardson. An algorithm for causal inference in the presence of latent variables and selection bias. In Clark Glymour and Gregory F. Cooper, editors, *Computation, Causation, and Discovery*, pages 211–252. AAAI Press, Menlo Park, CA, 1999. doi: [10.7551/mitpress/2006.003.0009](https://doi.org/10.7551/mitpress/2006.003.0009).

Change History

v1.0.0
General: Initial release 1