6. DAGs

ISS5096 || ECI

Jaewon (“Jay-one”) Yoo

National Tsing Hua University

Outline

1. DAGs as a Causal Language
2. Identification on a DAG
3. A Practical Tool: `dagitty`

Where are we? Where are we going?

- **Last time**: moved from experiments to observational studies under *selection on observables*.
    - Regression under conditional unconfoundedness $\{Y_i(1), Y_i(0)\} \perp\!\!\!\perp D_i \mid \mathbf{X}_i$ + overlap.
    - Sensitivity analysis when unconfoundedness is suspect.
- **Key loose end**: what goes into **X**? The assumption says what **X** must *do*, not what it *contains*.
- **Today**, use **DAGs** to:
    - Derive **X** from causal structure (*backdoor*).
    - Identify the effect of D on Y even when no adequate **X** exists (*frontdoor*, *do-calculus*).
    - Automate the machinery with `dagitty`.
    - Beyond identification: what else a DAG buys us $\rightsquigarrow$ **6(b)**.

Sources: https://www.redbubble.com/i/sticker/D-Ya-Like-Dags-by-salamincheese/27407958.EJUG5

p.s.: from an old British-American comedy film starring Brad Pitt

1/ DAGs as a Causal Language

Recap: The Rubin Causal Model

So far this semester, we've defined causal effects as contrasts between potential outcomes:

$$\tau_i = Y_i(1) - Y_i(0), \quad \tau_{\text{ATE}} = \mathbb{E}[\tau_i], \quad \tau_{\text{ATT}} = \mathbb{E}[\tau_i \mid D_i = 1], \ldots$$

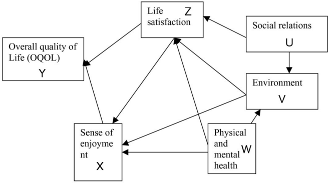
- A unit-level counterfactual framework (Neyman 1923, Rubin 1974).
- Identification: find assumptions (e.g., SOO) that let us recover τ from observed data.

- Today: an **alternative causal language**, directed acyclic graphs (DAGs), proposed by Judea Pearl.
- Same identification goal, different syntax. Complementary views.

An Alternative Causal Model: Causal Graphs

- Did social scientists not do causal inference before Rubin? **No!**
- The old paradigm: structural equation modeling and path analysis

1. Postulate a causal mechanism and draw a corresponding path diagram



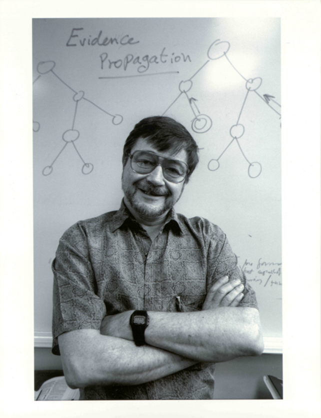
2. Translate it into a (typically linear) system of equations:

$$Y = \alpha_0 + \alpha_1 X + \alpha_2 Z + \varepsilon_\alpha$$
$$X = \beta_0 + \beta_1 Z + \beta_2 W + \beta_3 V + \varepsilon_\beta \quad \cdots$$

3. Estimate α, β, etc. typically assuming normality and exogeneity

- Heavily critiqued on several fronts (before Pearl's revival):
  - Strong distributional/functional form assumptions
  - No language to distinguish causation from association

Pearl's Attack



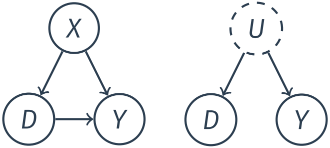
Judea Pearl (1936–) proposed a new causal inference framework based on nonparametric structural equation modeling (NPSEM)

- Originally a computer scientist
- Previous important work on artificial intelligence
- *Causality* (2000, Cambridge UP)
- Won the Turing Award in 2011 for his causal work

Pearl's framework builds on SEMs and revives it as a formal language of causality.

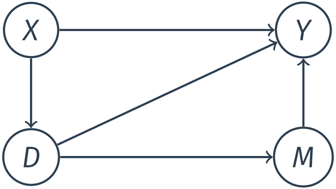
Directed Acyclic Graphs

- **Directed acyclic graphs** (DAGs) describe the causal structure of variables



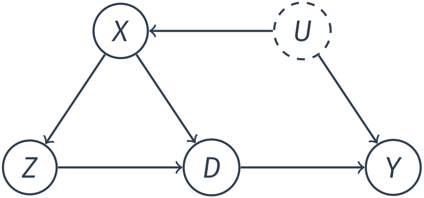
- **Nodes/vertices**: observed (solid) or unobserved (dashed) variables.
- **Edges**: arrows that encodes the presence or absence of a causal effect.
    - Arrow present = a direct causal effect: $Y_i(d) \neq Y_i(d')$ for some i and d.
    - Lack of an arrow = no causal effect: $Y_i(d) = Y_i(d')$ for all i and d.
    - Missing variables = no other common causes of any variables.
- **Directed**: each arrow implies a direction (causal ordering).
- **Acyclic**: no cycle: a variable cannot cause itself

DAG Terminologies



- **Path**: a sequence of edges that connect two nodes.
    - A **directed** or **causal** path is all in the same causal direction.
    - Non-causal path example: $D \leftarrow X \rightarrow Y$
- **Descendants**: nodes on a directed path away from some other node.
    - M is a descendant of D and X.
    - Ancestors is the reverse: X is an ancestor of M.
- **Parents**: immediate causes of a node.
    - D is the parent of Y and M.
    - **Children** are the reverse: M is a child of D.

DAGs to Distributions



$$Y = f_y(D, U, \varepsilon_y)$$

$$D = f_d(Z, X, \varepsilon_d)$$

$$X = f_x(U, \varepsilon_x)$$

$$Z = f_z(X, \varepsilon_z)$$

- **Markov factorization** (DAG = nonparametric SEM):

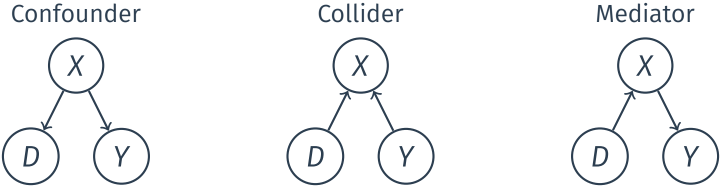
$$\mathbb{P}(X_1, \ldots, X_J) = \prod_{j=1}^{J} \mathbb{P}(X_j \,|\, \mathrm{pa}(X_j)), \quad \mathrm{pa}(X_j) = \text{parents of } X_j$$

- **Why this matters**: the factorization defines interventions.
  - $do(X_k = x_k)$: *exogenously set* $X_k = x_k$, so X_k no longer comes from its parents $\rightsquigarrow$ the factor $\mathbb{P}(X_k \mid \mathrm{pa}(X_k))$ **drops**.
  - Same operation graphically: **surgery** $G_{\overline{X_k}}$, delete arrows *into* X_k.
  - **Identification** = rewrite $\mathbb{P}(Y \mid do(D))$ via original factors; backdoor/frontdoor are the recipes.

D-Separation

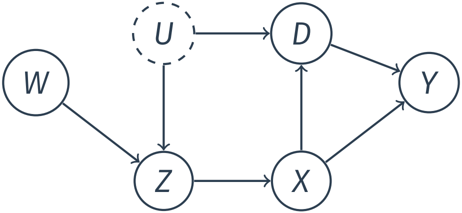
- Can we determine conditional independence from our causal DAG?
- Yes! To verify that $A \perp\!\!\!\perp B \mid C$ where each is a set of nodes:
  1. Find all paths from any vertex in A to any vertex in B.
  2. Check if each path is **blocked**.
  3. If all paths are blocked, then A is **d-separated** from B by C.
- A path is **blocked** conditional on C if:
  1. C includes a non-collider on that path **OR**
  2. Path includes a collider not in C and no descendant of any collider is in C.
- If A and B are d-separated, then we have $A \perp\!\!\!\perp B \mid C \rightsquigarrow$ if not, then d-connected.

Common Structures



- **Confounder** (fork): common cause. D, Y unconditionally dependent. Condition on X to remove confounding bias.
- **Collider** (inverted fork): common descendant. D, Y unconditionally independent. Do *not* condition on X, conditioning opens a spurious path. *Ex*: D = educ., Y = experience (indep. in population), X = hired (if either high); among hired, low D forces high $Y \rightsquigarrow$ correlation from nothing.
- **Mediator** (chain): on $D \to X \to Y$. D, Y dependent through X. Do *not* condition on X if the total $D \to Y$ effect is the target (blocks the indirect channel $\rightsquigarrow$ post-treatment bias).
- *Roles are **relative to a pair of nodes and a path**; the same node can play different roles for different pairs.*

D-Separation Example

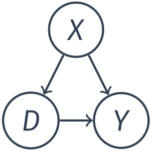


- Are W and Y marginally independent of each other?
    - Blocked: $W \rightarrow Z \leftarrow U \rightarrow D \rightarrow Y \quad W \rightarrow Z \leftarrow U \rightarrow D \leftarrow X \rightarrow Y$
    - Unblocked: $W \rightarrow Z \rightarrow X \rightarrow Y \quad W \rightarrow Z \rightarrow X \rightarrow D \rightarrow Y$
- Which variables should we condition on to make W and Y conditionally independent (d-separated)?
    - Block the unblocked paths without unblocking the blocked paths.
    - Conditioning on X would do this.
    - Conditioning on D and/or Z would unblock some of the blocked paths because they are colliders.

2/ Identification on a DAG

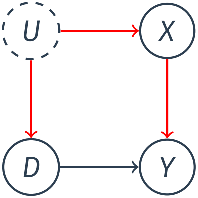
Backdoor Paths and Blocking Paths

- **Backdoor paths**: non-causal path from D to Y.
  - Would remain if we removed any arrows pointing out of D.
- Backdoor paths between D and Y $\rightsquigarrow$ common causes of D and Y:



- Here: backdoor path $D \leftarrow X \rightarrow Y$

Other Types of Confounding



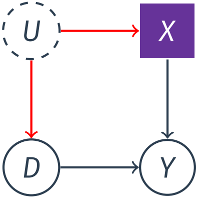
- D: job training. Y: getting a job. U: motivation (unobs.). X: applications sent.
- Big assumption: no arrow $U \longrightarrow Y$. Otherwise *no* observable would block $D \longleftarrow U \longrightarrow Y$; identification relies on X being the only route from U to Y.

Backdoor Criterion

$$(Y_i(1), Y_i(0)) \perp\!\!\!\perp D_i \mid \mathbf{X}_i$$

- Can we use a DAG to evaluate no unmeasured confounders?
- Holds if the **backdoor criterion** on a causal DAG is met:
  1. No vertex/node in **X** is a descendant of D (**no post-treatment bias**), and
  2. **X** blocks all backdoor paths from D to Y.
- The backdoor criterion is the graphical version of last week's conditional unconfoundedness: given the DAG, it *proves* the independence above; overlap is still an additional distributional assumption.
- What the criterion tells us, all at once:
  1. Whether there is confounding given this DAG,
  2. Whether it is possible to remove the confounding, and
  3. What variables to condition on to eliminate the confounding.

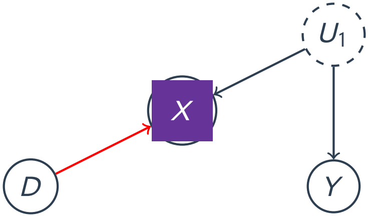
Other Types of Confounding *Redux*



- D: enrolling in a job training program.
- Y: landing a job!
- U: being motivated.
- X: number of job applications sent out.
- Big assumption here $\rightsquigarrow$ no arrow linking U to Y

- Conditioning on X blocks all backdoor paths.

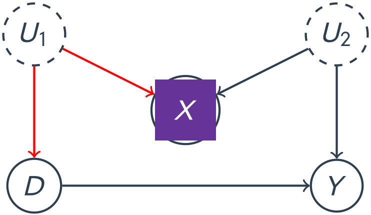
Why Not Condition on Descendents?



- No causal or statistical relationship between D and Y

- Conditioning on post-treatment variables opens up non-causal paths
    - $\rightsquigarrow$ Statistical relationship between D and Y conditional on X
    - But still no causal relationship $\rightsquigarrow$ selection bias.

M-Bias



- Without conditioning, the path $D \leftarrow U_1 \rightarrow X \leftarrow U_2 \rightarrow Y$ is **blocked by the collider** X.
- Conditioning on X opens it up, inducing confounding from nothing. (Hence **M-bias**, the graph's shape.)
- *Ex*: D = schooling, Y = earnings, U_1 = family background, U_2 = ability, X = test score.
- Controversial: contradicts "always control for pretreatment covariates."
  - Rubin: M-bias is a "mathematical curiosity".
  - Pearl: M-bias is a real threat; structure, not timing.

Good Controls, Bad Controls

Students often ask: “should I control for X?”, no one-size-fits-all answer.

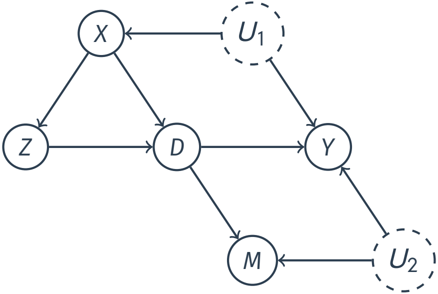
Cinelli, Forney, & Pearl (2022) organize the answers by structural role:

Type	Structural role	Effect of conditioning
Good	Confounder (common cause of D, Y)	Removes confounding bias
Neutral	Cause of Y only	Unbiased; improves precision
Neutral	Cause of D only	Unbiased; may reduce precision
Bad	Collider (common *effect* of D, Y)	Opens a spurious path (M-bias)
Bad	Mediator on $D \rightarrow Y$	Removes part of the causal effect
Bad	Descendant of Y	Induces selection bias

- The good / bad distinction is structural, *not* statistical, it depends on the DAG, not on what the data “looks like.”
- The backdoor criterion operationalizes this: it tells you *which* conditioning sets are good.

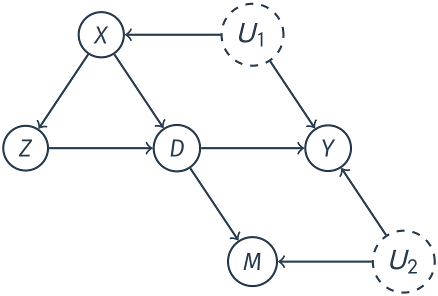
Usage Example of DAG

Assume that you've come out with a DAG based on your expertise:



- Suppose you want to identify a causal effect of D on Y.
    - In a nutshell, you want to block all the paths that yield statistical associations between D and Y.
    - Thus, you want to find a set of nodes **S** such that once we condition on **S**
        - **no unmeasured confounding** holds and
        - any descendant of D is not in S → no post-treatment bias.
- $\rightsquigarrow$ Use backdoor criterion!

Back Door Criterion Example



1. List all of the **backdoor paths** between D and Y.
2. List all the possible set of nodes **S** that you can condition on.
3. List all the **S** such that **blocks** all the backdoor paths.
4. Among those **S**, drop the sets which include a descend of D.

But Where Does the DAG Come From?

- Backdoor, frontdoor, and do-calculus all rely on a *correct* DAG. Where does it come from? Not from the data, from **theory and domain expertise**.
- Practical workflow:
  1. **Nodes**: D, Y, plus variables plausibly causing either (unobserved $\rightsquigarrow$ dashed).
  2. **Edges**: for each pair (A, B), “could A directly cause B?”. *Missing arrows are the strong assumptions.*
  3. **Direction**: temporal ordering, manipulability, natural laws.
  4. **Sanity check**: the DAG implies conditional independences, testable in data.
- Drawing is theoretical work: specifying the DAG “forces a clearer articulation of the research question and engagement with theoretical mechanisms” (Tafti & Shmueli 2025, *MISQ*).
- Uncertain DAG? **Draw competing scenarios** and compare identification across them, robustness through structure.

3/ A Practical Tool: `dagitty`

Why Automate Identification?

Manual backdoor analysis is fine on toy DAGs. On real research DAGs, it breaks down.

- **Scale**: 4–5 nodes is tractable by hand; 15+ nodes multiply combinatorially.
- **Correctness**: humans miss paths when colliders and chains interact $\leadsto$ biased effect.
- **Reproducibility**: a coded DAG is auditable, reviewers can verify end-to-end.
- **Robustness**: swap the DAG, re-run `adjustmentSets()`, see if identification survives.
- **Falsification**: a DAG implies conditional independences, testable in data via `dagitty`.

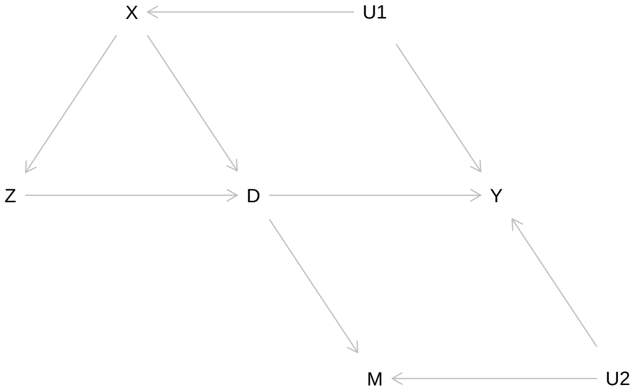
R Package: `dagitty`

- DAGitty: www.dagitty.net

```
> library(dagitty)
> g <- dagitty('dag {
      X [pos=”1,-1.5”]
      Y [pos=”4,0”]
      Z [pos=”0,0”]
      M [pos=”3,1.5”]
      D [pos=”2,0”]
      U1 [pos=”3,-1.5”]
      U2 [pos=”5,1.5”]
      X -> Z -> D -> Y
      X -> D -> M
      M <- U2 -> Y
      X <- U1 -> Y
  }')
> latents(g) <- c(”U1”, ”U2”)
```

R Package: `dagitty`

```
> plot(g) # Visualize the DAG
```



R Package: `dagitty`

- Access parent/ancestor nodes:

```
> parents(g, "D") ## [1] "X" "Z"
[1] "X" "Z"

> ancestors(g, "D") ## [1] "D" "Z" "X" "U1"
[1] "D"  "Z"  "X"  "U1"
```

- Or children/descendent nodes:

```
> children(g, "D") ## [1] "M" "Y"
[1] "M" "Y"

> descendants(g, "D") ## [1] "D" "Y" "M"
[1] "D" "Y" "M"
```

R Package: `dagitty`

- Identify paths using `paths()`:

```
> paths(g, "D", "Y")$paths
[1] "D -> M <- U2 -> Y"     "D -> Y"                 "D <- X <- U1 -> Y"
[4] "D <- Z <- X <- U1 -> Y"
```

- Extract causal path(s) by setting `directed = T`:

```
> paths(g, "D", "Y", directed = TRUE)$paths # only causal path(s)
[1] "D -> Y"
```

- Check whether two nodes are d-separated using `dseparated()`:

```
> dseparated(g, "Z", "D", c("X")) # because of Z -> D
[1] FALSE

> dseparated(g, "Z", "M", c("D"))
[1] TRUE
```

R Package: `dagitty`

- Access the set of nodes **S** to condition on for **no unmeasured confounding** to hold:

```
> adjustmentSets(g, "D", "Y", type="minimal")
{ X }
```

- Caveat: `adjustmentSets` may include unobserved variables which we cannot actually condition on.

```
> S = adjustmentSets(g, "D", "Y", type="all")

> S[!grepl("U1|U2", S)]
{ X }       # X typically preferred for simplicity and max statistical power, unless
{ X, Z }    # Z is a known strong confounder and you have sufficient data.
```

- Note that this implements a slightly more general criterion (sometimes it may contain descendants)

R Package: `dagitty`

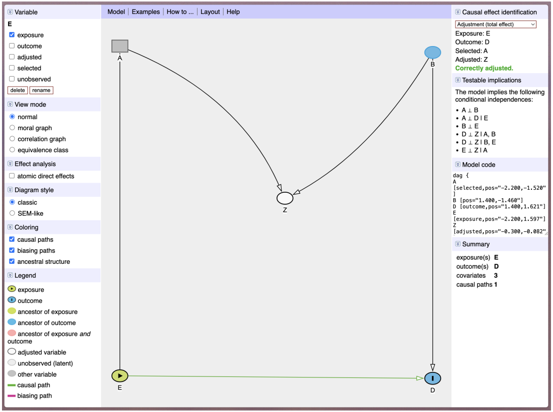
Full list of adjustment sets:

```
# S = adjustmentSets(g, "D", "Y", type="all")
> S
{ U1 }
{ U1, U2 }
{ M, U1, U2 }
{ X }
{ U1, X }
{ U2, X }
{ M, U2, X }
{ U1, U2, X }
{ M, U1, U2, X }
{ U1, Z }
{ U1, U2, Z }
{ M, U1, U2, Z }
{ X, Z }
{ U1, X, Z }
{ U2, X, Z }
{ M, U2, X, Z }
{ U1, U2, X, Z }
{ M, U1, U2, X, Z }
```

All satisfy the criterion, **which one do you pick?**

- First drop sets containing *unmeasured* variables $(U_1, U_2) \rightsquigarrow$ infeasible.
- **Minimal feasible** set $(\{X\})$: cheapest to collect, sufficient for identification.
- **Larger feasible** sets (e.g., $\{X, Z\}$): same bias, gain precision if the added variable is a *cause of* Y, like adding controls in regression.

DAGitty (`https://www.dagitty.net/`)



Variable

E

☑ exposure
☐ outcome
☐ adjusted
☐ selected
☐ unobserved

View mode

● normal
○ moral graph
○ correlation graph
○ equivalence class

Effect analysis

☐ atomic direct effects

Diagram style

● classic
○ SEM-like

Coloring

☑ causal paths
☑ biasing paths
☑ ancestral structure

Legend

- exposure
- outcome
- ancestor of exposure
- ancestor of outcome
- ancestor of exposure *and* outcome
- adjusted variable
- unobserved (latent)
- other variable
- causal path
- biasing path

Causal effect identification

Adjustment (total effect)

Exposure: E
Outcome: D
Selected: A
Adjusted: Z
Correctly adjusted.

Testable implications

The model implies the following conditional independences:

- $A \perp B$
- $A \perp D \mid E$
- $B \perp E$
- $D \perp Z \mid A, B$
- $D \perp Z \mid B, E$
- $E \perp Z \mid A$

Model code

```
dag {
A
[selected,pos="-2.200,-1.520"
]
B [pos="1.400,-1.460"]
D [outcome,pos="1.400,1.621"]
E
[exposure,pos="-2.200,1.597"]
Z
[adjusted,pos="-0.300,-0.082"
```

Summary

exposure(s)	**E**
outcome(s)	**D**
covariates	**3**
causal paths	**1**

- Draw DAGs interactively; valid adjustment sets highlighted live; exports R code.

Summing Up

- DAG is an **alternative causal language** to Rubin CM, same identification goal, different syntax.
- Identification on a DAG = eliminate the $do(\cdot)$ operator. **Backdoor** (block all backdoor paths with observed **S**) is our working tool today.
- "Should I control for X?" is a *structural* question, determined by the DAG, not by the data.
- Beyond $\widehat{\tau}$: a DAG encodes the whole causal model, not just $D \longrightarrow Y$, which is the topic of **6(b)**.
- Up next: **instrumental variables**, a third identification strategy that exploits an *upstream* exogenous variable, complementary to today's structures.

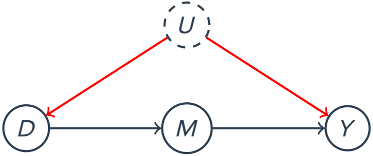
Onto the presentations & discussions!

Contact Information:
jaewon.yoo@iss.nthu.edu.tw
https://j1yoo.github.io/

清華大學
TSING HUA UNIVERSITY

What If Backdoor Fails?

- Backdoor criterion requires: some *observable* **S** that blocks all backdoor paths. What if no such **S** exists? E.g., a key confounder is unmeasurable.
- Pearl's insight: a fully *mediating* observable M can still rescue identification.



- Classic example (Pearl): D = smoking, M = tar deposits, Y = lung cancer, U = unmeasured genetic predisposition.
- No observable blocks $D \leftarrow U \rightarrow Y$. Yet $\mathbb{P}(Y \mid do(D))$ is still identified.

Frontdoor Criterion

- A variable (or set) M is a **frontdoor adjustment** for $D \longrightarrow Y$ if:
  1. M intercepts *all* directed paths from D to Y.
  2. No unblocked backdoor path from D to M.
  3. All backdoor paths from M to Y are blocked by D.
- Then the causal effect is identified:

$$\mathbb{P}(y \mid do(d)) \;=\; \sum_{m} \mathbb{P}(m \mid d) \sum_{d'} \mathbb{P}(y \mid m, d')\, \mathbb{P}(d')$$

- Intuition, decompose $D \longrightarrow Y$ via M:
  - $\mathbb{P}(m \mid d)$: $D \longrightarrow M$ has no backdoor $\rightsquigarrow$ identified directly.
  - $\sum_{d'} \mathbb{P}(y \mid m, d')\mathbb{P}(d')$: $M \longrightarrow Y$ adjusted by D, which blocks M's backdoors.
- Identification under unmeasured confounding, the power of an explicit DAG.
- *Note*: both backdoor and frontdoor are special cases of Pearl's 3-rule **do-calculus** (proven complete by Shpitser & Pearl 2006), a general-purpose identification engine, beyond today's scope.