Recovering causal graphs with adaptive interventions

Tea talk (10 August 2023)
Empirical Inference group @ MPI Intelligent Systems

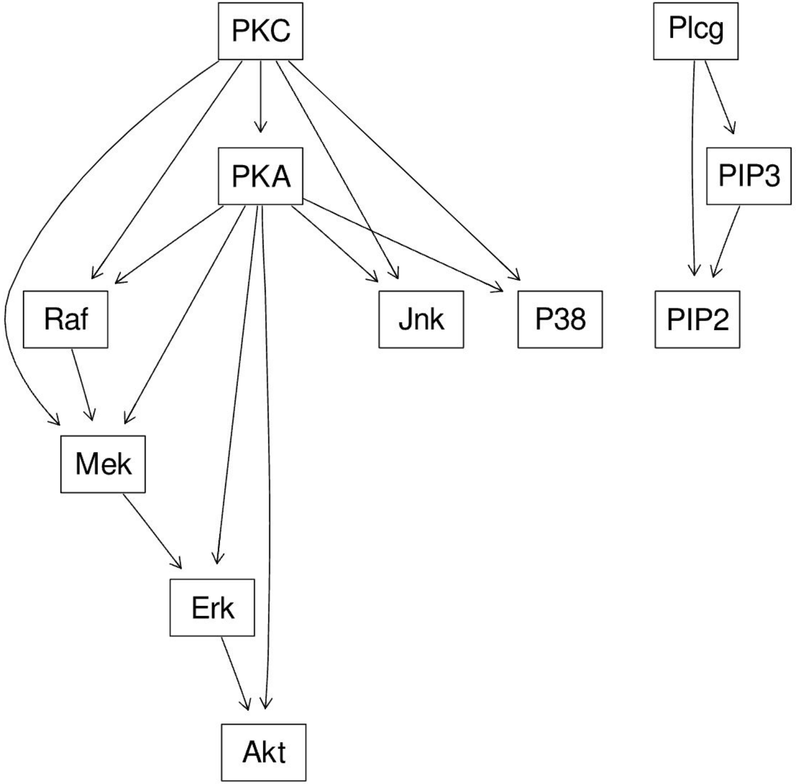

Davin
Choo


Arnab
Bhattacharyya


Themis
Gouleakis


Kirankumar
Shiragur

Suppose there is an underlying causal DAG G*



SACHS

Number of nodes: 11
Number of arcs: 17
Number of parameters: 178
Average Markov blanket size: 3.09
Average degree: 3.09
Maximum in-degree: 3

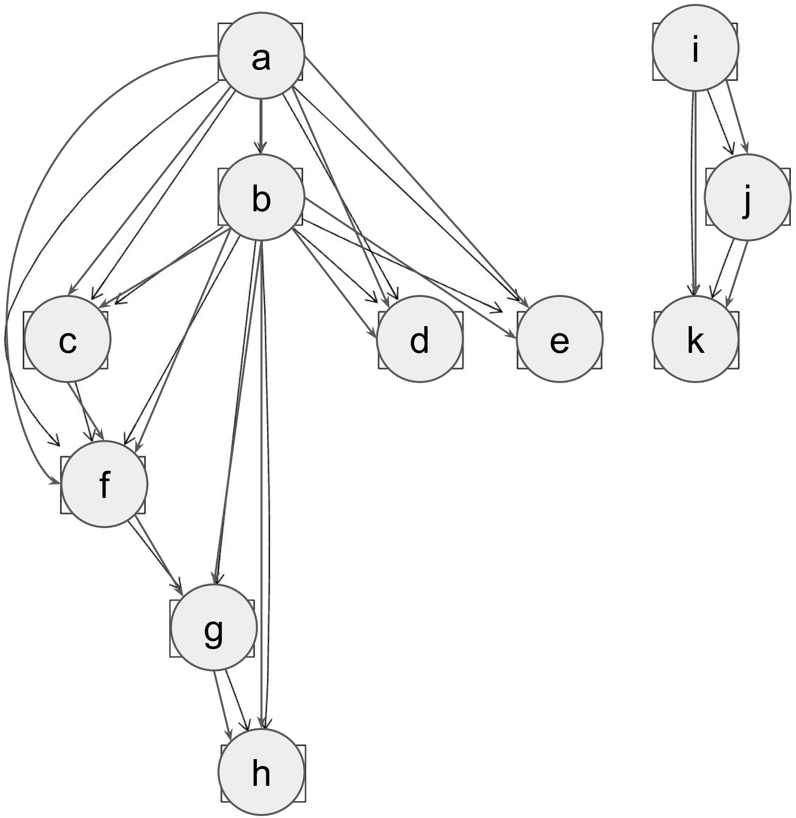
BIF (1.9kB)
DSC (1.9kB)
NET (1.7kB)
RDA (bn.fit) (2.4kB)
RDS (bn.fit) (2.4kB)

K. Sachs, O. Perez, D. Pe'er, D. A. Lauffenburger and G. P. Nolan. Causal Protein-Signaling Networks Derived from Multiparameter Single-Cell Data. Science, 308:523-529, 2005.

Assumption:
Causal sufficiency

Source: https://www.bnlearn.com/bnrepository/discrete-small.html#sachs

Suppose there is an underlying causal DAG G*



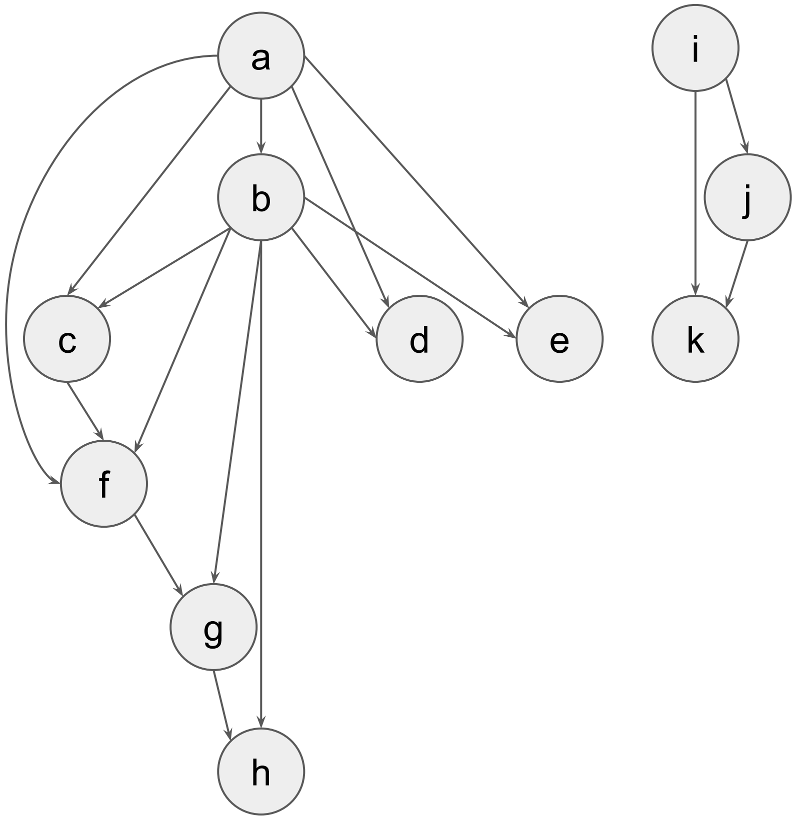
SACHS

Number of nodes: 11
Number of arcs: 17
Number of parameters: 178
Average Markov blanket size: 3.09
Average degree: 3.09
Maximum in-degree: 3

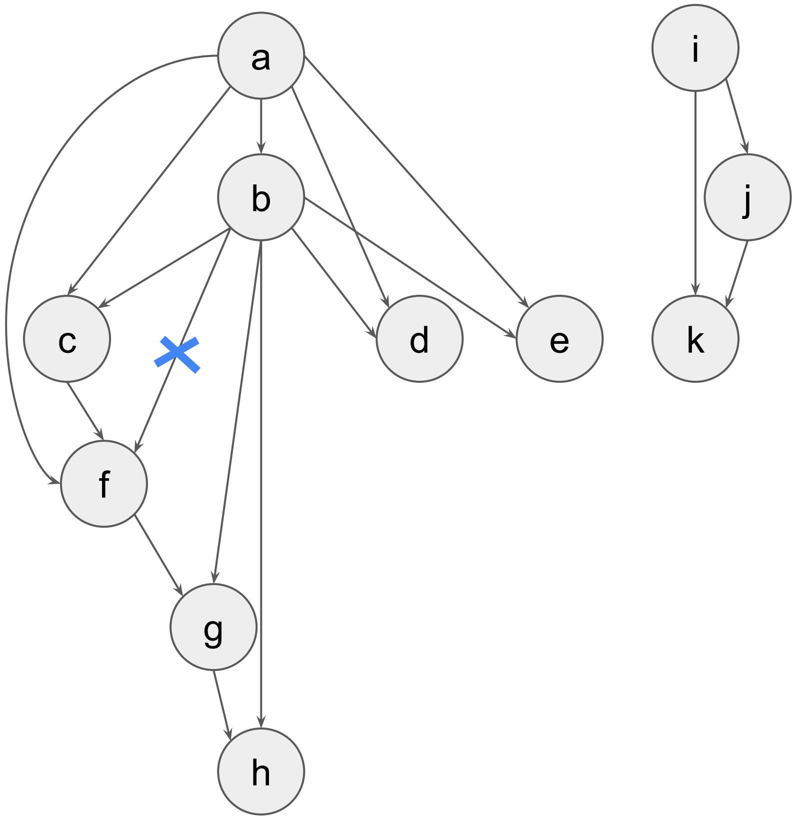
BIF (1.9kB)
DSC (1.9kB)
NET (1.7kB)
RDA (bn.fit) (2.4kB)
RDS (bn.fit) (2.4kB)

K. Sachs, O. Perez, D. Pe'er, D. A. Lauffenburger and G. P. Nolan. Causal Protein-Signaling Networks Derived from Multiparameter Single-Cell Data. Science, 308:523-529, 2005.

Suppose there is an underlying causal DAG G*

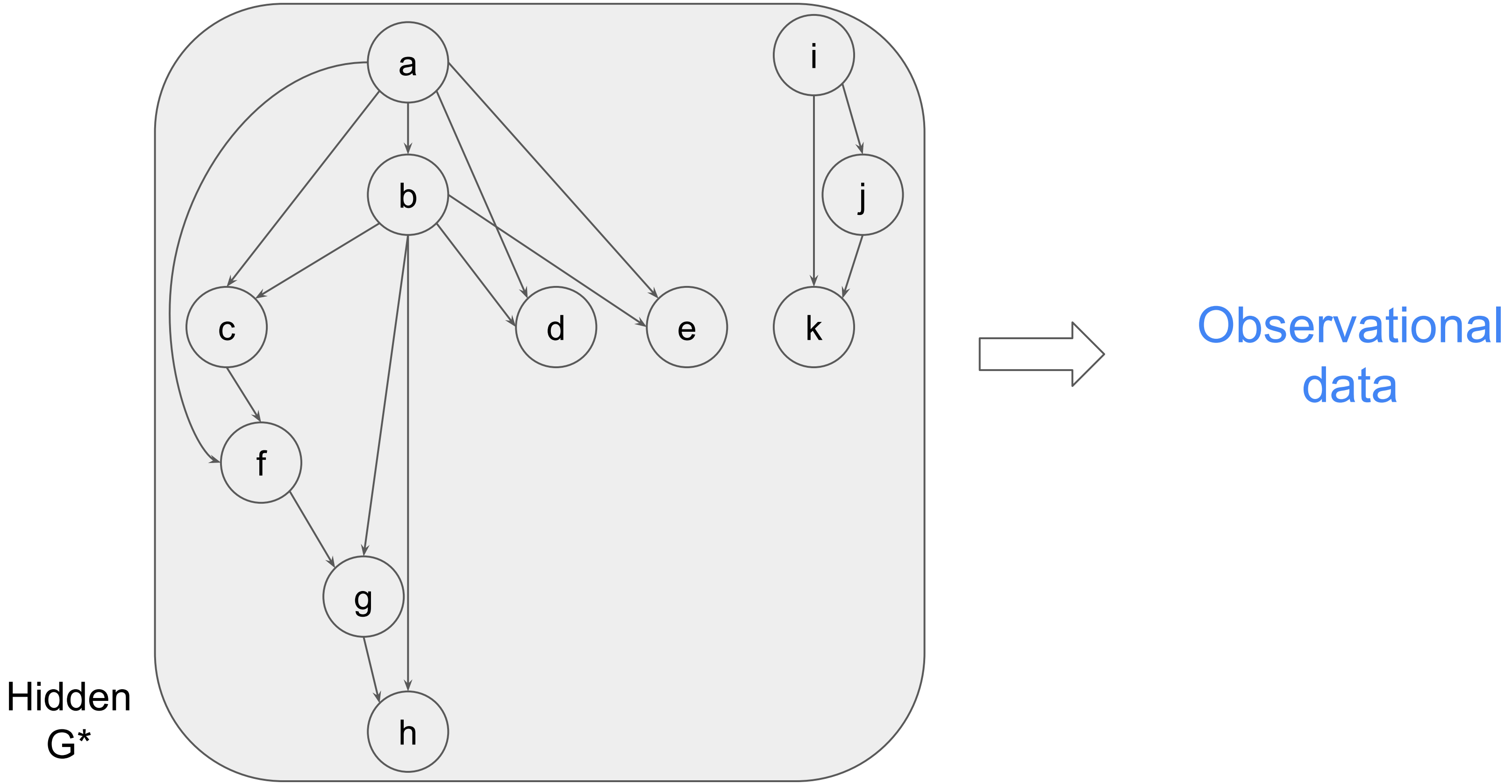


Suppose there is an underlying causal DAG G*

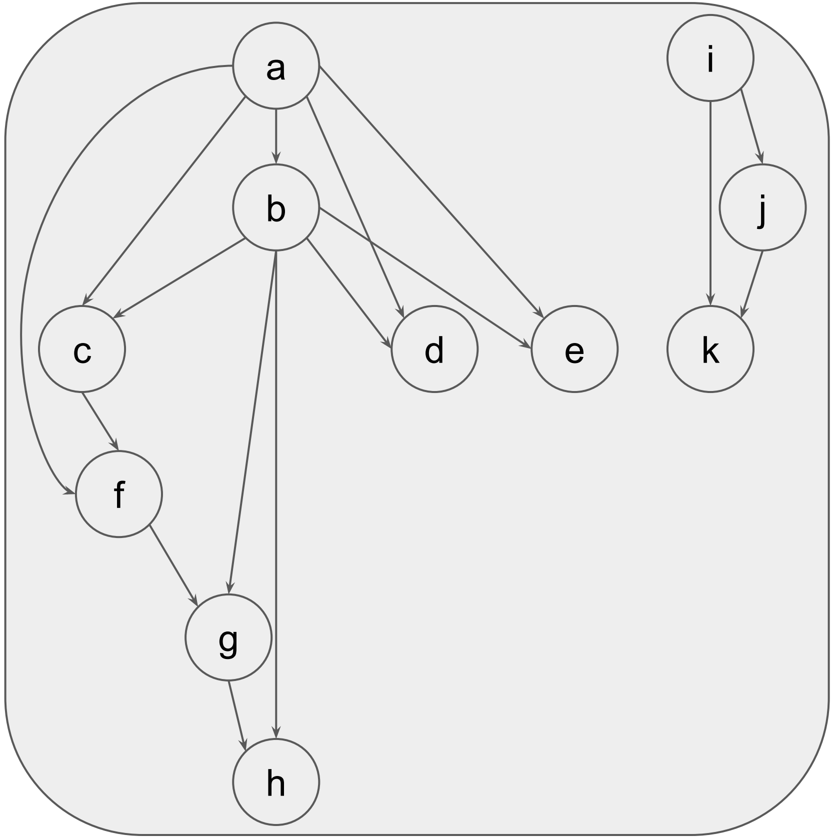


Let me modify G* slightly for this presentation

Goal: Recover DAG G* from data



Markov equivalence class [G*]

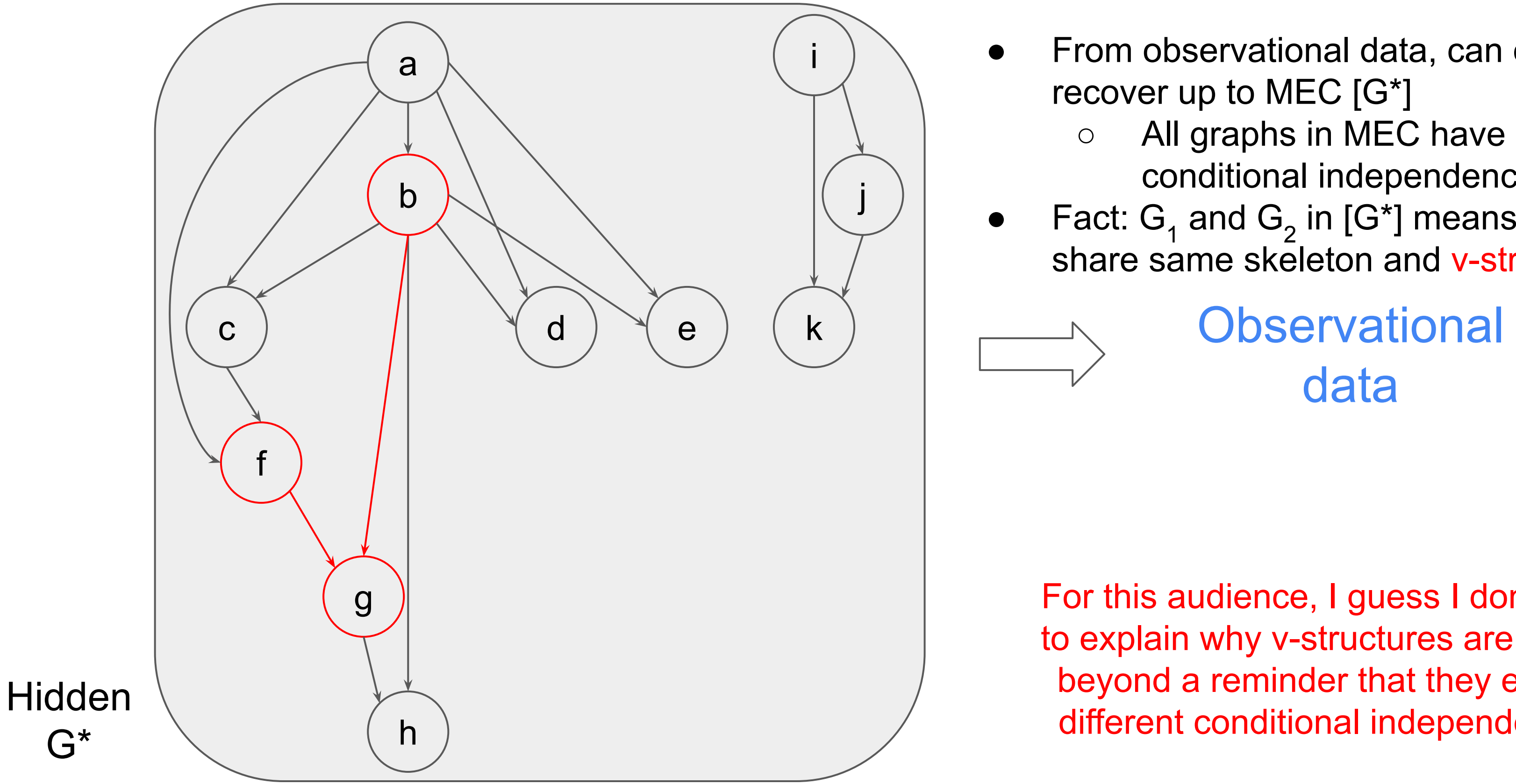


Hidden
G*

- From observational data, can only recover up to MEC [G*]
  - All graphs in MEC have same conditional independencies

Observational data

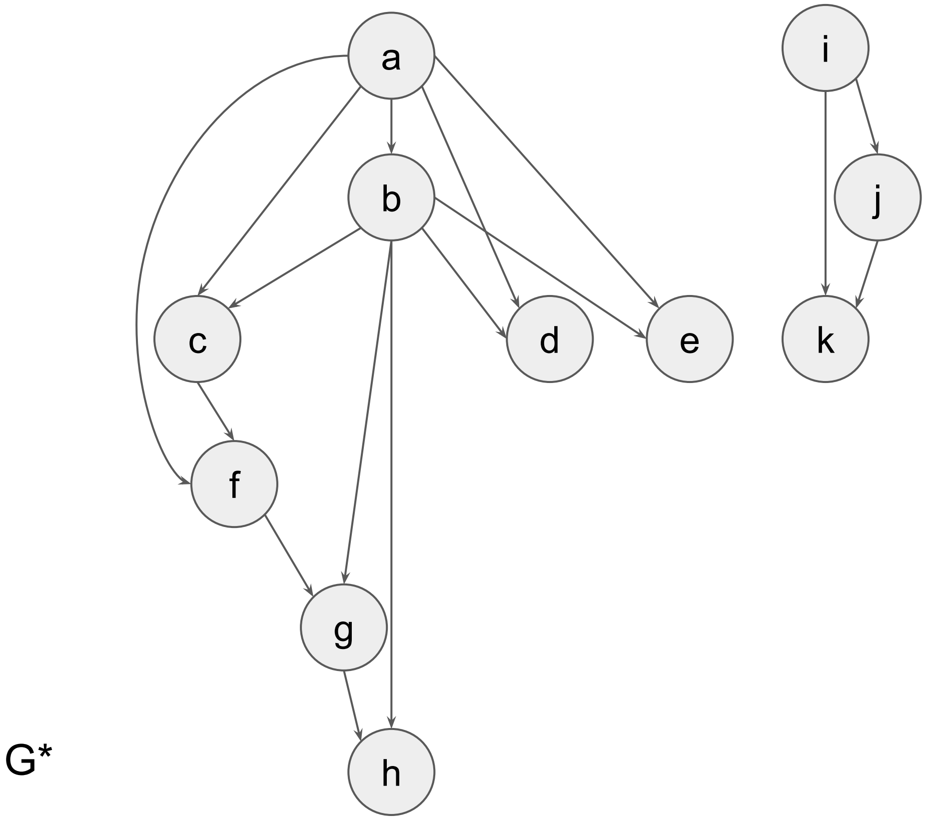
Markov equivalence class [G*]



- From observational data, can only recover up to MEC [G*]
    - All graphs in MEC have same conditional independencies
- Fact: G_1 and G_2 in [G*] means they share same skeleton and v-structures

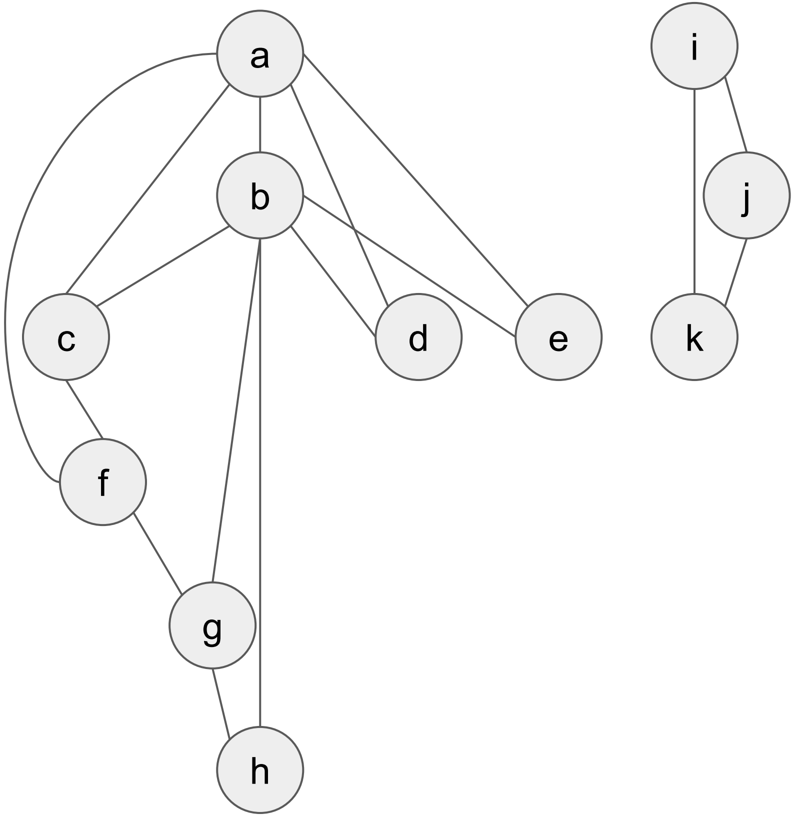
For this audience, I guess I don't need to explain why v-structures are special beyond a reminder that they encode different conditional independencies

Essential graph E(G*)



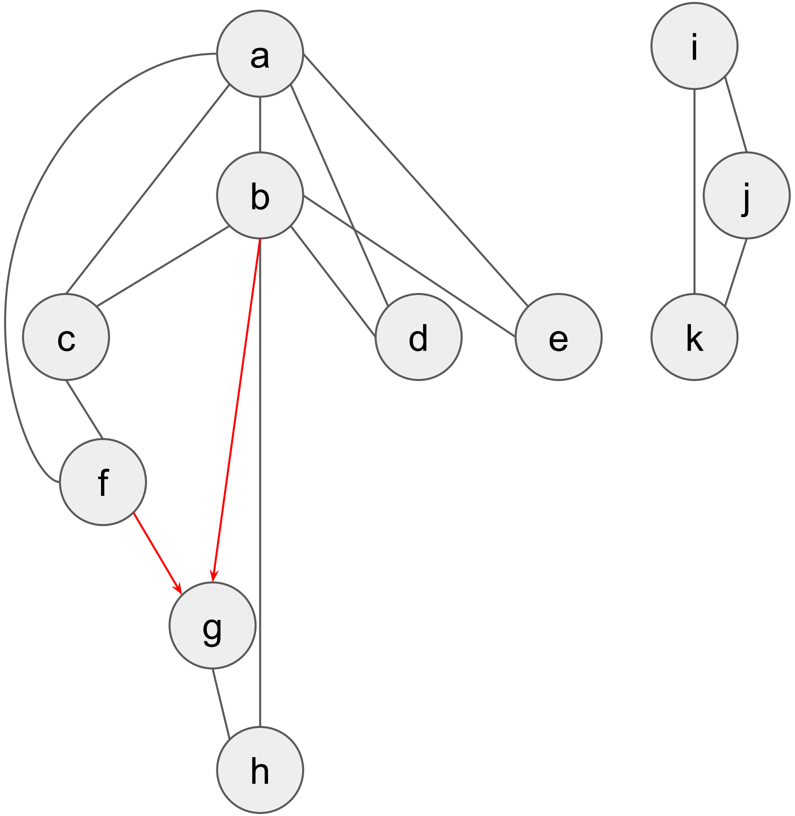
- From observational data, can only recover up to MEC [G*]
  - All graphs in MEC have same conditional independencies
- Fact: G_1 and G_2 in [G*] means they share same skeleton and v-structures
- Essential graph E(G*)
  - Graphical representation of [G*]
  - Partially oriented version of G*
- How to compute E(G*) from G*?

Essential graph E(G*)



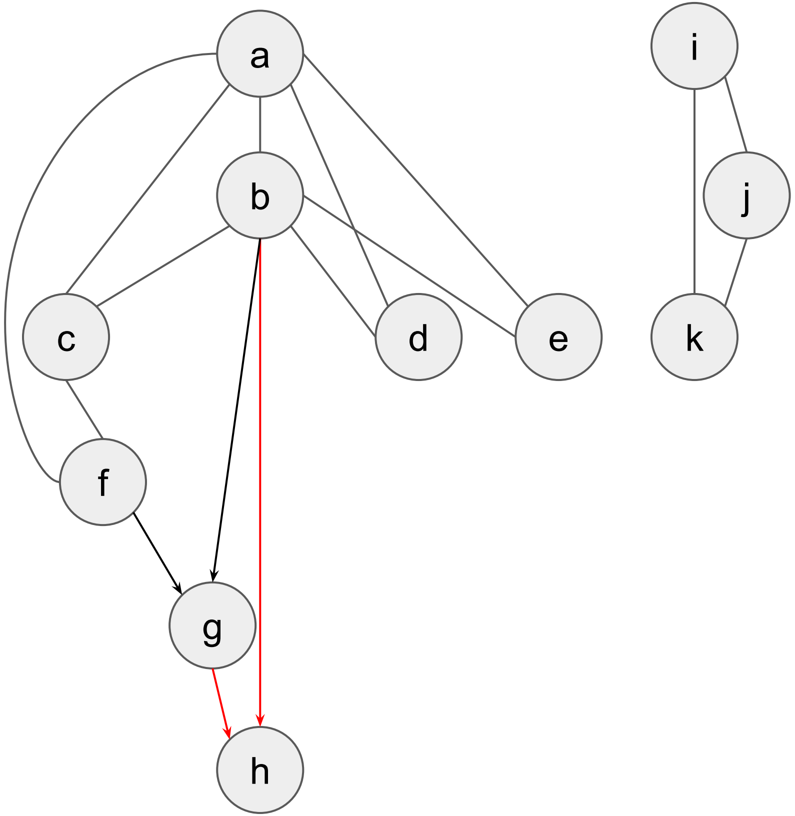
- From observational data, can only recover up to MEC [G*]
  - All graphs in MEC have same conditional independencies
- Fact: G_1 and G_2 in [G*] means they share same skeleton and v-structures
- Essential graph E(G*)
  - Graphical representation of [G*]
  - Partially oriented version of G*
- How to compute E(G*) from G*?
  - Start from skeleton of G*

Essential graph E(G*)



- From observational data, can only recover up to MEC [G*]
  - All graphs in MEC have same conditional independencies
- Fact: G_1 and G_2 in [G*] means they share same skeleton and v-structures
- Essential graph E(G*)
  - Graphical representation of [G*]
  - Partially oriented version of G*
- How to compute E(G*) from G*?
  - Start from skeleton of G*
  - Orient v-structures

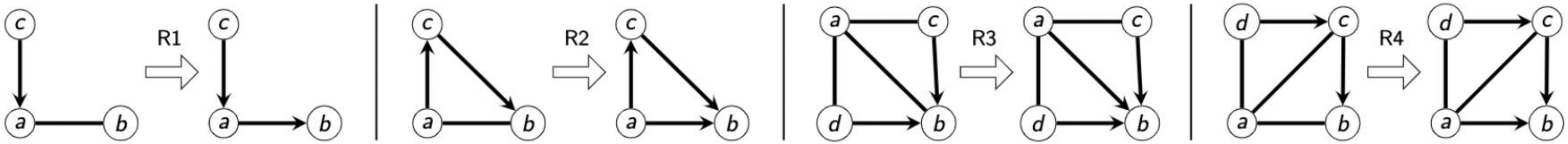
Essential graph E(G*)



E(G*)

- From observational data, can only recover up to MEC [G*]
  - All graphs in MEC have same conditional independencies
- Fact: G_1 and G_2 in [G*] means they share same skeleton and v-structures
- Essential graph E(G*)
  - Graphical representation of [G*]
  - Partially oriented version of G*
- How to compute E(G*) from G*?
  - Start from skeleton of G*
  - Orient v-structures
  - Apply Meek rules until fixed point

Meek rules [Meek 1995]

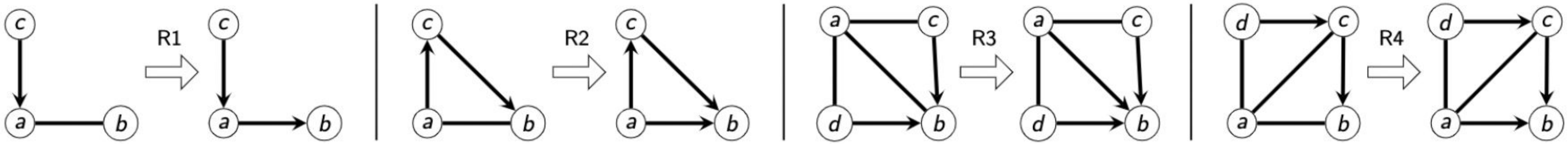


Will not wrongly orient arcs

Will not miss out on any orientations

- **Sound** and **complete** (with respect to arc orientations with acyclic completions)
- Converge in polynomial time [Wienöbst, Bannach, Liśkiewicz 2021]

Meek rules [Meek 1995]



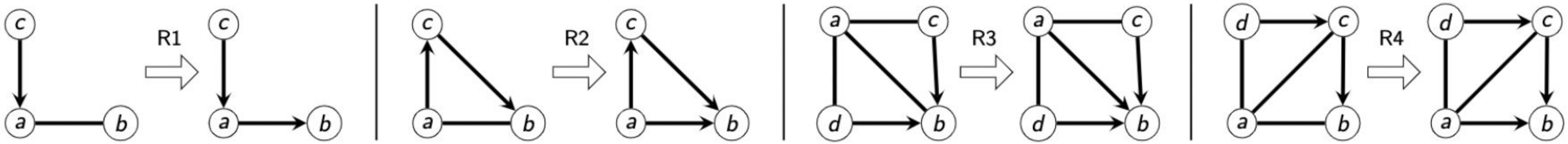
If b ← a, then new v-structure

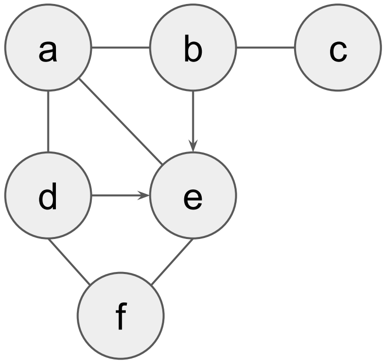
If b ← a, then cycle formed

If b ← a, then unoriented arcs would have been oriented in the same way in all DAGs within the MEC (via R2), i.e. they would not have been unoriented in the essential graph

- **Sound** and **complete** (with respect to arc orientations with acyclic completions)
- Converge in polynomial time [Wienöbst, Bannach, Liśkiewicz 2021]

Exercise: Getting a feel of Meek rules





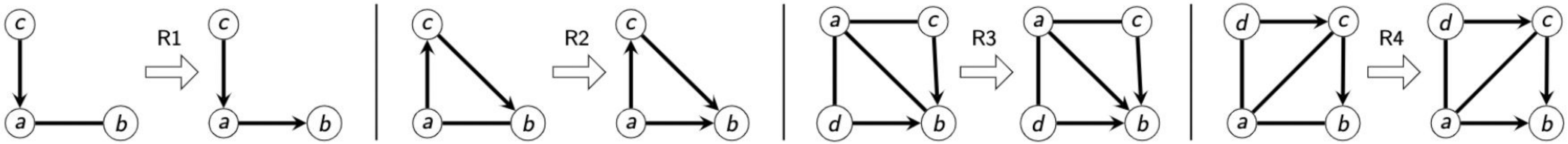
Suppose we are given this partially oriented graph…

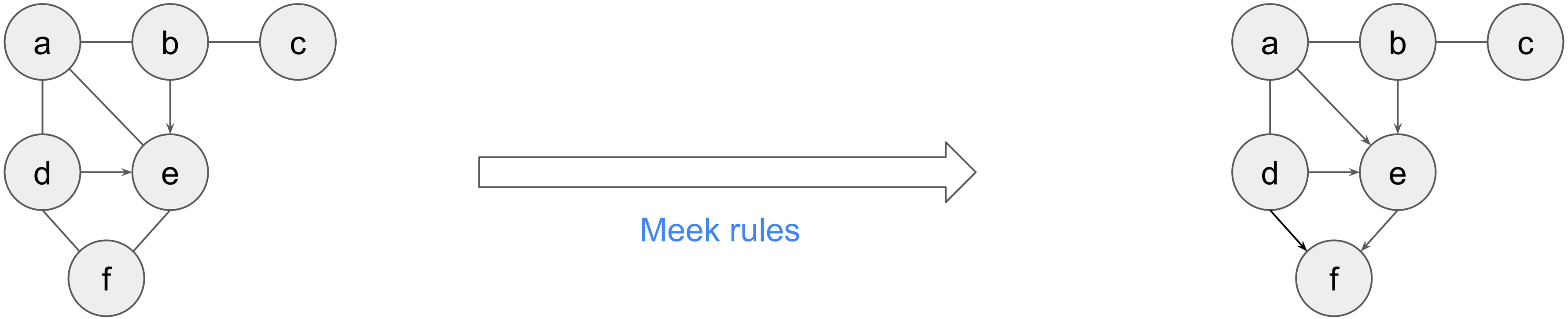
What additional arcs can we recover?

Quiz: How many unoriented edges remain?

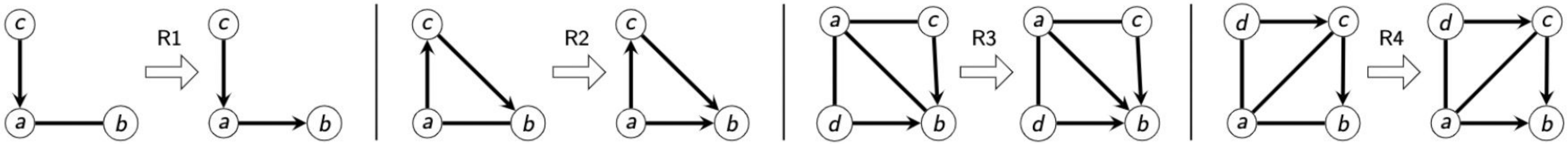
(A): 0
(B): 1
(C): 3
(D): 5

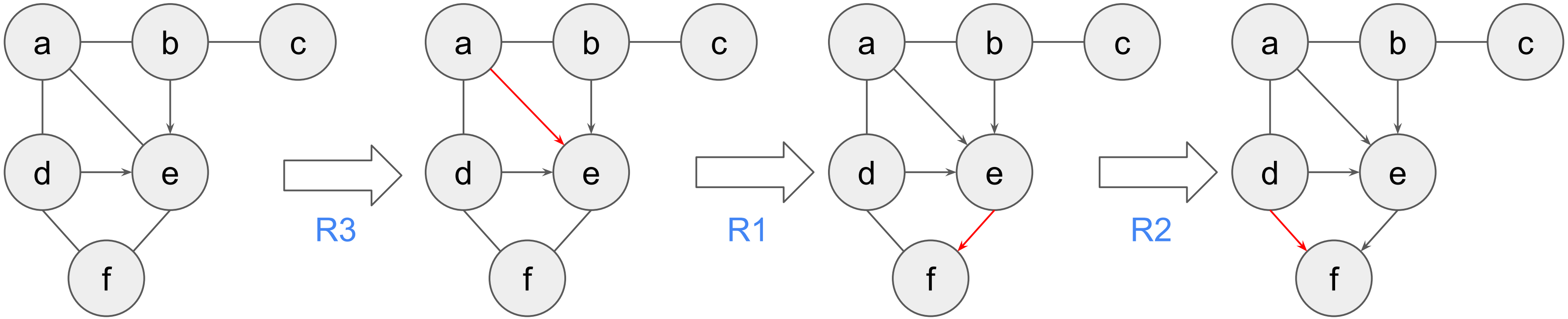
Exercise: Getting a feel of Meek rules





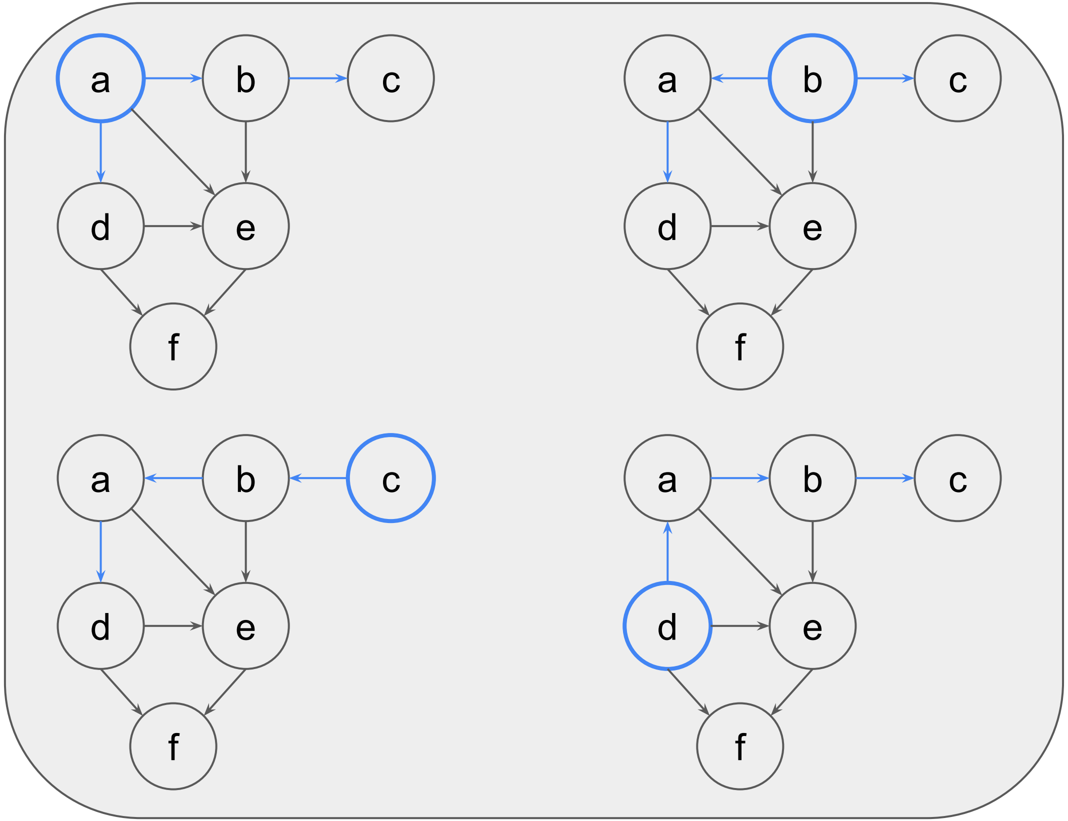
Exercise: Getting a feel of Meek rules





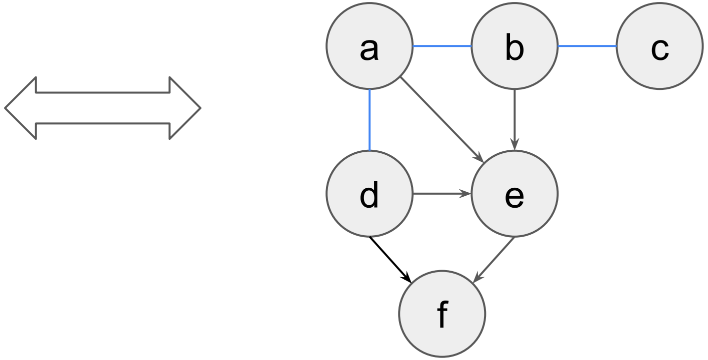
Note: Also okay to apply R1 first before R3. Ordering does not matter since Meek rules is complete!

E(G*) and the corresponding MEC [G*]



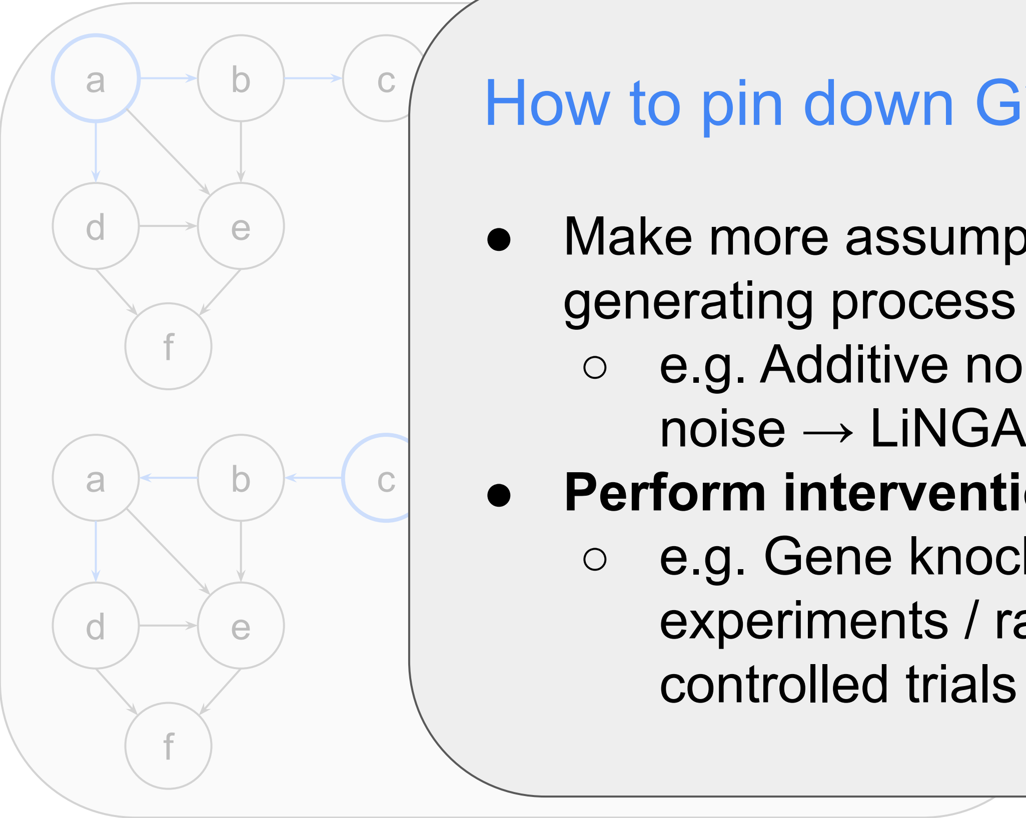
[G*]

Any of these 4 graphs could have been the true underlying causal graph G*



E(G*)

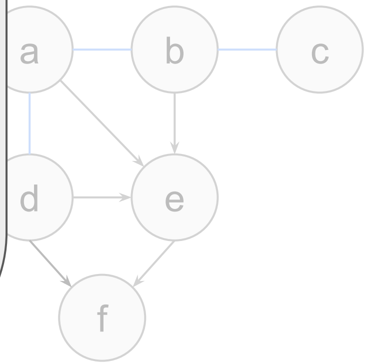
E(G*) and the corresponding MEC [G*]



How to pin down G* within [G*]?

- Make more assumptions on data generating process
  - e.g. Additive non-Gaussian noise → LiNGAM methods
- **Perform interventions**
  - e.g. Gene knockout experiments / randomized controlled trials

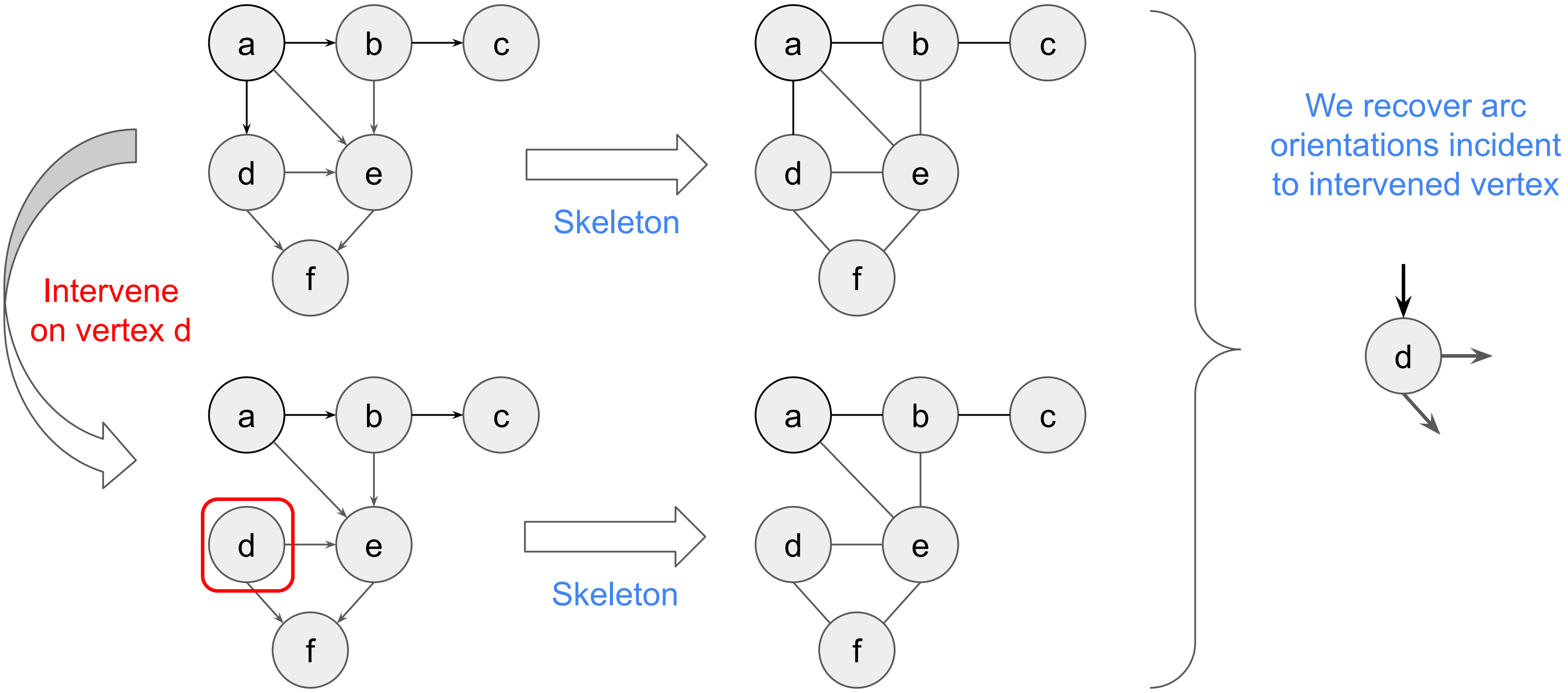
4 graphs could
en the true
ausal graph G*



[G*]

E(G*)

What do interventions buy us?



Caveats

- Assumptions ⟵ (Problem was still non-trivial and unresolved *despite* these assumptions)
  - Causal sufficiency
  - When we perform intervention on a vertex v, we recover arc orientations† incident to intervened vertex v (ignoring finite sample and computational concerns)
    - e.g. hard / perfect / do interventions then compare the skeletons
    - May also be possible with imperfect interventions while making other assumptions about the data generating process

- For this talk
  - Atomic / Single vertex interventions
  - Each vertex has the same intervention cost

- Objective and performance metric
  - Minimize number of interventions performed to recover G* from [G*]

† This is slightly different when we intervene on multiple vertices. We do not learn orientation of an edge {u,v} if we intervene on both at the same time.

Caveats

- Assum … ptions)
  - C
  - W … ations†
    in … ncerns)
    - … ns
    - … g other

We can abstract causal structure learning as a graph problem with specialized causal graph manipulation operations

- For this ta…
  - Atomic / Single vertex interventions
  - Each vertex has the same intervention cost
- Objective and performance metric
  - Minimize number of interventions performed to recover G* from [G*]

† This is slightly different when we intervene on multiple vertices. We do not learn orientation of an edge {u,v} if we intervene on both at the same time.

Before we proceed…
5Ws and 1H

https://knowyourmeme.com/memes/why-is-gamora
https://imgflip.com/memegenerator/184033944/Where-is-Gamora

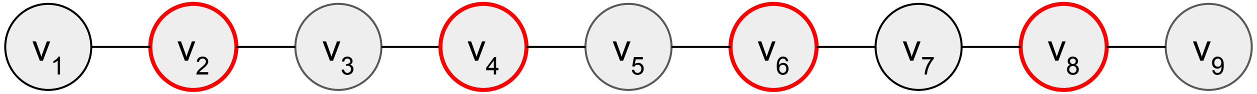
Non-adaptive interventions

- Given MEC [G*], decide on a **single fixed set of interventions** that recovers **any possible G*** within [G*]
- Graph-separating system† [Kocaoglu, Dimakis, Vishwanath 2017]
- For single interventions, this corresponds to a vertex cover

† Every unoriented arc {u,v} is "cut" by at least one intervention, i.e. there is some intervention J such that | J ∩ {u,v} | = 1.

Non-adaptive interventions

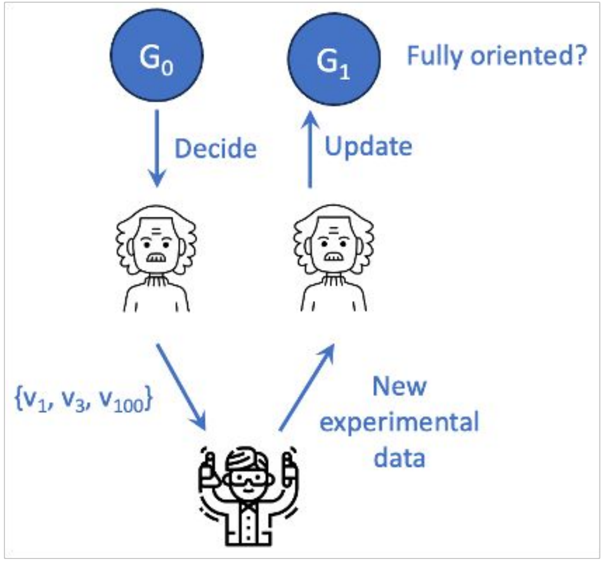
- Given MEC [G*], decide on a **single fixed set of interventions** that recovers **any possible G*** within [G*]
- Graph-separating system† [Kocaoglu, Dimakis, Vishwanath 2017]
- For single interventions, this corresponds to a vertex cover

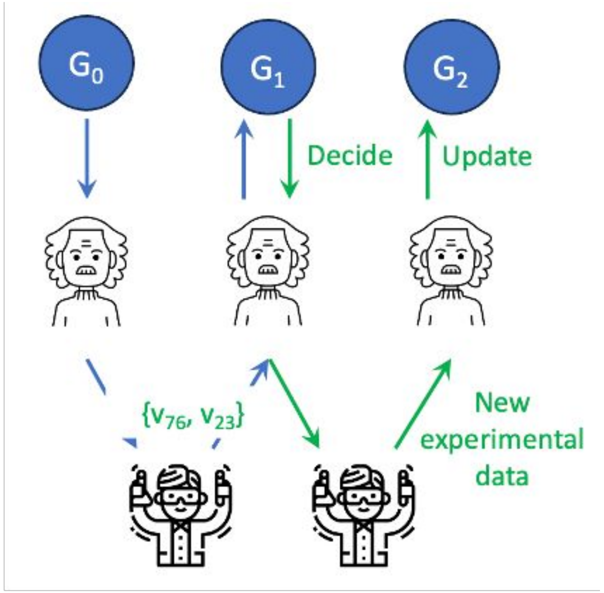


- Suppose the essential graph is an unoriented path on n = 9 nodes
- There are 9 possible DAGs in this MEC: Pick v_i as source and orient arcs away
- 4 non-adaptive interventions are necessary and sufficient

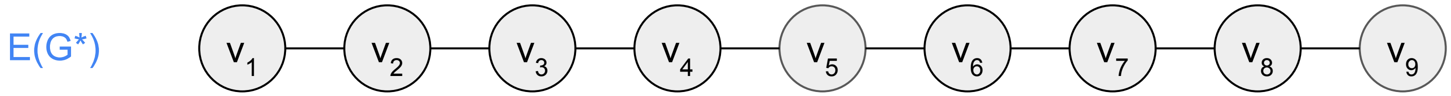
† Every unoriented arc {u,v} is "cut" by at least one intervention, i.e. there is some intervention J such that | J ∩ {u,v} | = 1.

Adaptive interventions



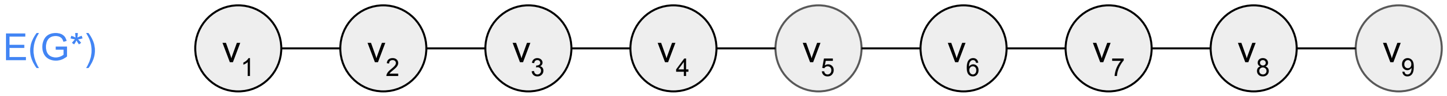


Power of adaptivity: Possibly exponential improvement!

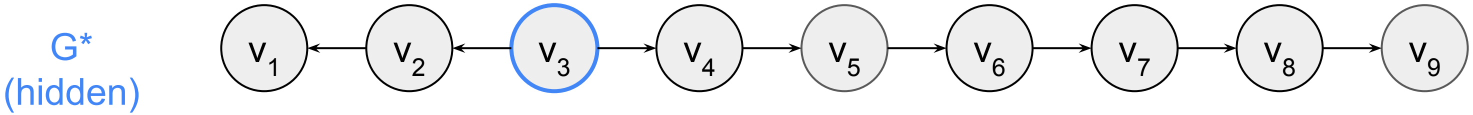


- Consider essential graph is a path on n nodes: **Θ(n) non-adaptive interventions**
- But we only need **Θ(log n) adaptive interventions** by simulating binary search!

Power of adaptivity: Possibly exponential improvement!



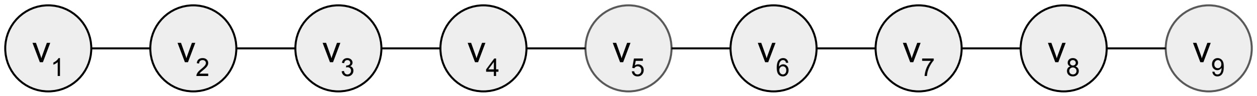
- Consider essential graph is a path on n nodes: **Θ(n) non-adaptive interventions**
- But we only need **Θ(log n) adaptive interventions** by simulating binary search!



Suppose this was G*

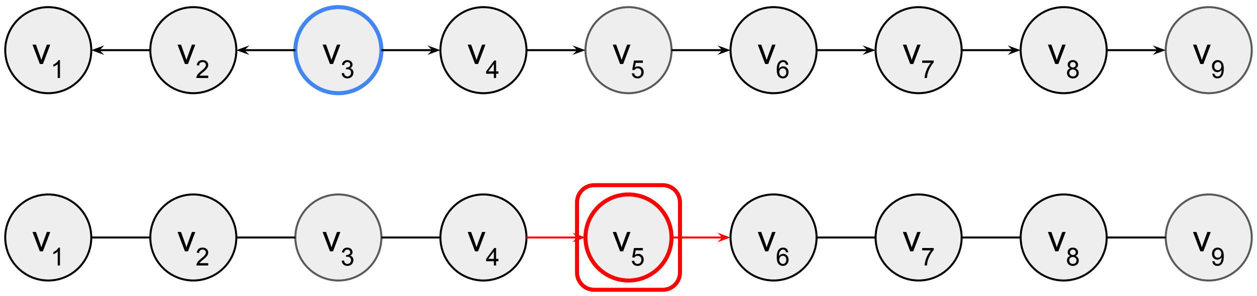
Power of adaptivity: Possibly exponential improvement!

E(G*)



- Consider essential graph is a path on n nodes: **Θ(n) non-adaptive interventions**
- But we only need **Θ(log n) adaptive interventions** by simulating binary search!

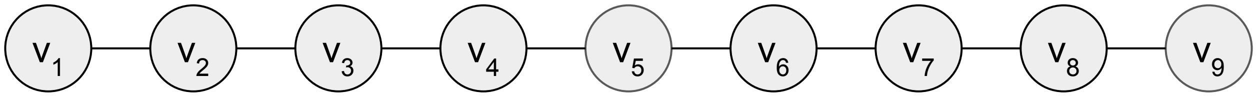
G*
(hidden)



Recover arc orientations incident to v_5

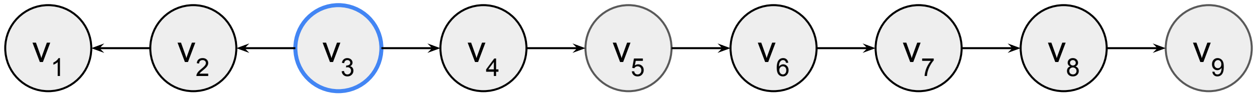
Power of adaptivity: Possibly exponential improvement!

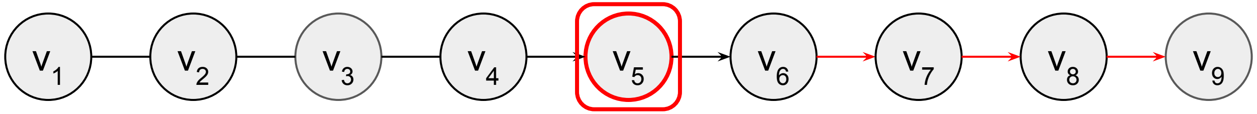
E(G*)



- Consider essential graph is a path on n nodes: **Θ(n) non-adaptive interventions**
- But we only need **Θ(log n) adaptive interventions** by simulating binary search!

G*
(hidden)

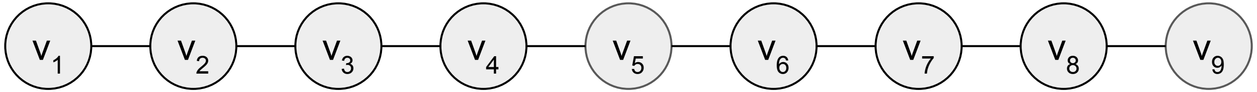




Apply Meek rules (in this case, R1)

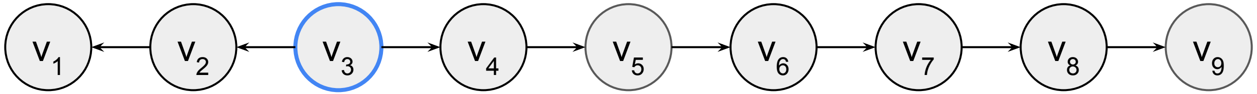
Power of adaptivity: Possibly exponential improvement!

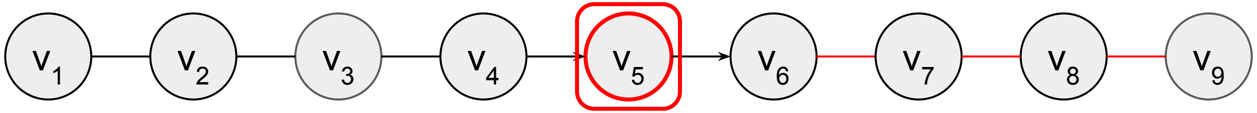
E(G*)



- Consider essential graph is a path on n nodes: **Θ(n) non-adaptive interventions**
- But we only need **Θ(log n) adaptive interventions** by simulating binary search!

G*
(hidden)





Recurse on unoriented v_1 - v_2 - v_3- v_4

How to measure performance?

- Since we recover arc orientations incident to intervened vertex, O(n) interventions always trivially suffice…
- But what if we **know** G* and tell someone else the best possible set of interventions to perform, in order to "verify"? What is the best we can hope for?
  - Clearly, the difficulty depends on structure of G*
  - Let us denote this "verification number" as v(G*)

How to measure performance?

- Since we recover arc orientations incident to intervened vertex, O(n) interventions always trivially suffice…
- But what if we **know** G* and tell someone else the best possible set of interventions to perform, in order to "verify"? What is the best we can hope for?
  - Clearly, the difficulty depends on structure of G*
  - Let us denote this "verification number" as $v(G^*)$
- What was known?†
  - If $E(G^*)$ is a clique on n vertices, $v(G^*) = \lfloor n/2 \rfloor$
  - If $E(G^*)$ is a tree on n vertices, $v(G^*) = 1$
    - Intervene on the source node, then apply Meek R1
  - Approximations and bounds to $v(G^*)$

    [Squires, Magliacane, Greenewald, Katz, Kocaoglu, Shanmugam 2020]
    [Porwal, Srivastava, Sinha 2022]

† Before our work

How to measure performance?

ns

What we can show

- Exact characterization of $\nu(G^*)$
- $O(\log n \cdot \nu(G^*))$ adaptive interventions always possible
- $\Omega(\log n \cdot \nu(G^*))$ is worst case necessary
- Along with many other extensions…

† Before our work

How to measure performance?

ns

What we can show

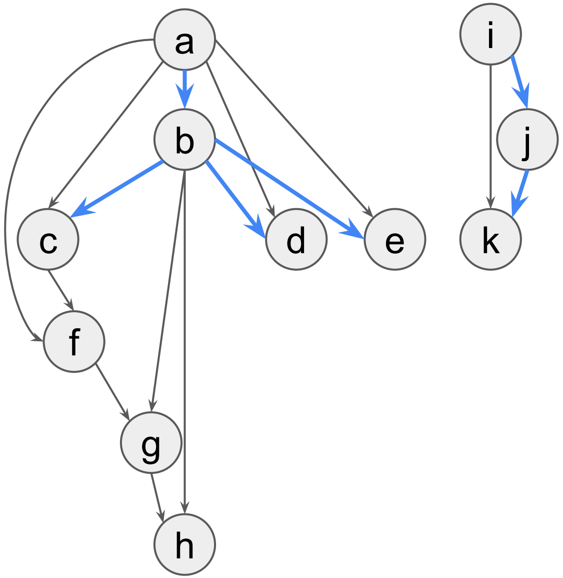
- **Exact characterization of v(G*)**
- **O(log n · v(G*)) adaptive interventions always possible**
- Ω(log n · v(G*)) is worst case necessary (n-node path)
- Along with many other extensions… (see ending slides)

† Before our work

Verification number v(G*) is the size of the minimum vertex cover of the covered edges of G*

To be precise, we showed that it is **necessary and sufficient** to intervene on at least one endpoint of every covered edge.

Verification number v(G*) is the size of the minimum vertex cover of the covered edges of G*

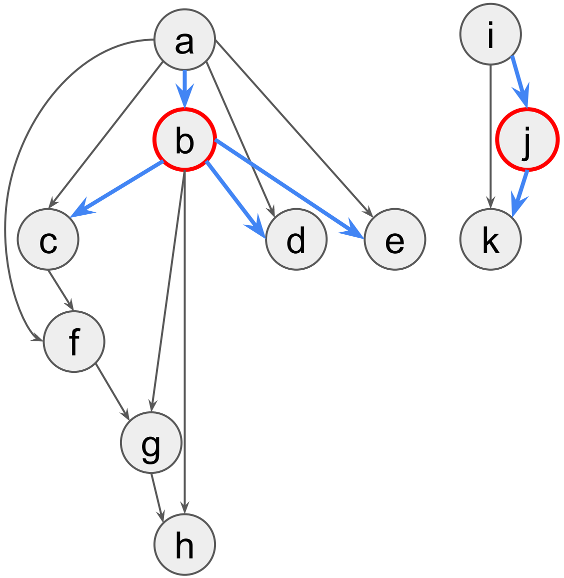


[Chickering 1995]

u → v is covered edge if
Pa(u) = Pa(v) \ {u}

i.e. u and v "share same parents"

Verification number v(G*) is the size of the minimum vertex cover of the covered edges of G*



- Minimum vertex cover is NP-hard to compute in general…
- What we can show:
  - Covered edges form a forest
  - So, we can use dynamic programming to compute v (G*) in linear time
  - Also works if vertices have different interventional costs

Appreciating prior results through our characterization†

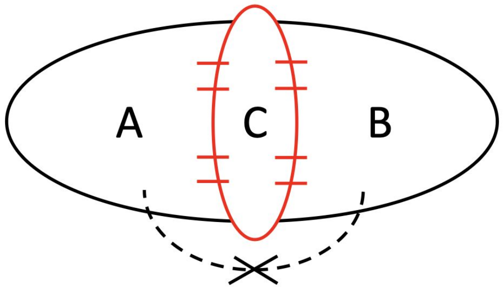
Verification number v(G*) is the size of the minimum vertex cover of the covered edges of G*

- If E(G*) is a clique on n vertices, $v(G^*) = \lfloor n/2 \rfloor$
  - Suppose clique topological ordering is $v_1, v_2, \ldots, v_n$
  - Then, covered edges are precisely $v_1 \to v_2, v_2 \to v_3, \ldots, v_{n-1} \to v_n$
- If E(G*) is a tree on n vertices, v(G*) = 1
  - Covered edges are precisely all edges incident to the root
- Non-adaptive interventions and graph separating systems
  - Two graphs are in the same MEC **if and only if** there is a sequence of covered edge reversals that transform between them [Chickering 1995]
  - Implication: Every unoriented edge in the essential graph is a covered edge for *some* DAG in the MEC, so non-adaptive interventions must cut all edges!

† Skip if no time

$O(\log n \cdot \nu(G^*))$ adaptive interventions always suffice

- Algorithm does not need to know $\nu(G^*)$, just the essential graph $E(G^*)$ as input
- Based on two ideas†:
  - Unoriented connected components are *chordal* graphs and information from one component does not help another [Hauser, Bühlmann 2012, 2014]
  - For any *chordal* graph $G = (V, E)$ on $|V| = n$ nodes, one can compute a *clique separator* C in polynomial time [Gilbert, Rose, Edenbrandt 1984]
    - That is, we can partition vertex set V into A, B, C such that: $|A|, |B| \leq n/2$; C is a clique; no edges between A and B



† I do not wish to define / introduce the notions of chordal graphs, chain components and interventional essential graphs, so let me be a little informal here :)

$O(\log n \cdot \nu(G^*))$ adaptive interventions always suffice

- Algorithm does not need to know $\nu(G^*)$, just the essential graph $E(G^*)$ as input
- Based on two ideas[†]:
  - Unoriented connected components are *chordal* graphs and information from one component does not help another [Hauser, Bühlmann 2012, 2014]
  - For any *chordal* graph $G = (V, E)$ on $|V| = n$ nodes, one can compute a *clique separator* C in polynomial time [Gilbert, Rose, Edenbrandt 1984]
    - That is, we can partition vertex set V into A, B, C such that: $|A|, |B| \leq n/2$; C is a clique; no edges between A and B
- Algo.: Find clique separators, intervene on vertices within one by one; Recurse
- Analysis
  - $O(\log n)$ rounds of recursion suffices
  - Incur $O(\nu(G^*))$ interventions per round (We proved a new stronger lower bound on $\nu(G^*)$; see [CSB22])

† I do not wish to define / introduce the notions of chordal graphs, chain components and interventional essential graphs, so let me be a little informal here :)

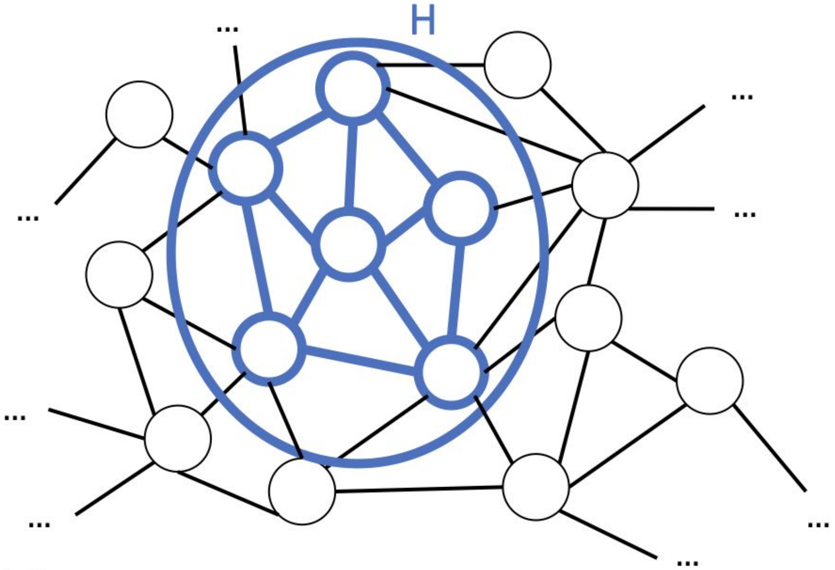
Some other related questions that we have also studied†

- Non-atomic / bounded size interventions
  - May intervene on more than 1 vertex in one intervention
- Vertices have varying interventional costs
  - It may be easier to enforce an intervention on diet (eat an apple a day) than exercise (run 10km every day) → w(diet = 1 apple) < w(exercise = run 10km)
  - Some vertices cannot be intervened, possibly due to ethics → $w(v) = \infty$

- Some motivating vignettes in the next few concluding slides:
  - What if we only care about a subgraph in the large causal graph? [CS23]
  - What if there are limited rounds of adaptivity? [CS23]
  - Can we make use of an imperfect expert knowledge to improve guarantees in a principled and provable fashion? [CGB23]

† See my webpage (davinchoo.com) for more details, or come talk to me! Some other follow-ups that we have studied are not shown here.

What if causal graph is HUGE?



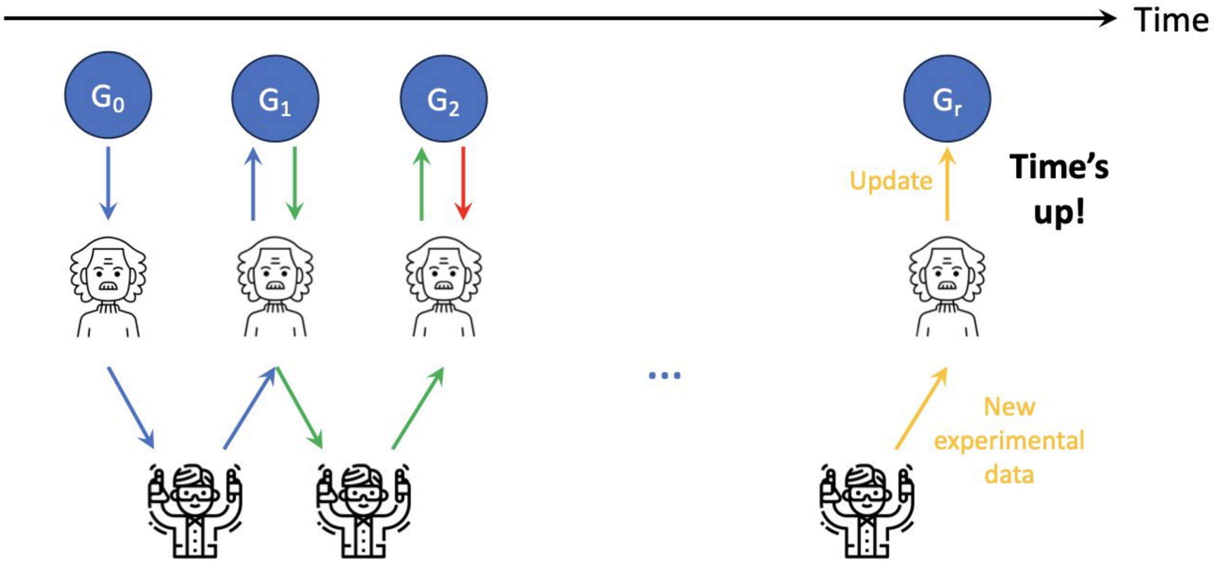
Local causal discovery:

Only care about a small subgraph of the larger graph

(Informal) Verification: Generalization of "DP on covered edge forest" [CS23]

(Informal) Search: $\mathcal{O}(\log|H| \cdot \nu(G^*))$ interventions suffices [CS23]

What if we have limited rounds of adaptivity?



Given a budget of r adaptive rounds, how to minimize number of interventions?

$\mathcal{O}\left(\min\{r, \log n\} \cdot n^{\frac{1}{\min\{r, \log n\}}} \cdot \nu(G^*)\right)$ interventions† suffice [CS23]

$r = 1$ ⟷ $r \in \mathcal{O}(\log n)$

$\mathcal{O}(n)$ — $\mathcal{O}\left(\log n \cdot \nu(G^*)\right)$

"Matches non-adaptive" — "Matches fully adaptive"

There are domain experts!

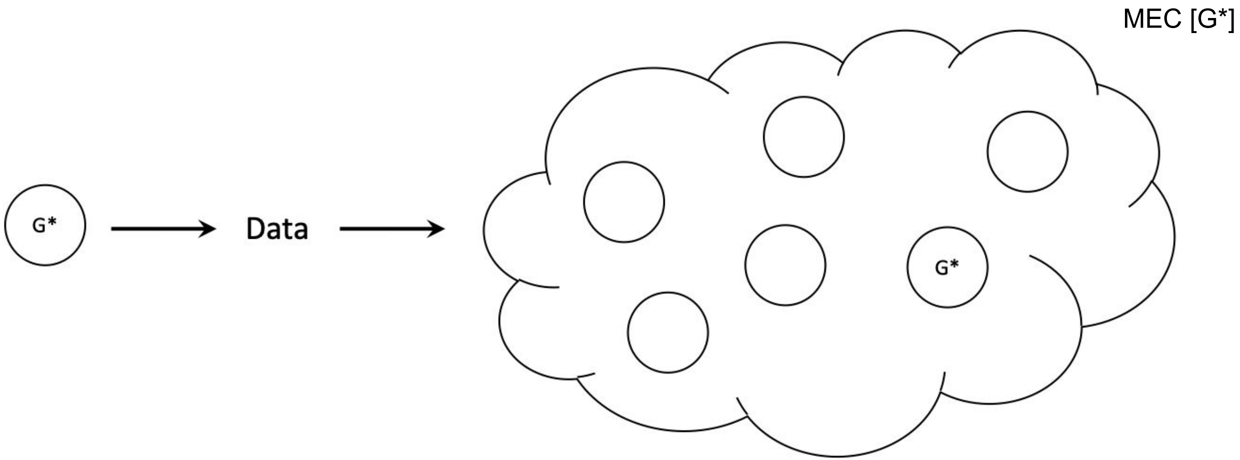
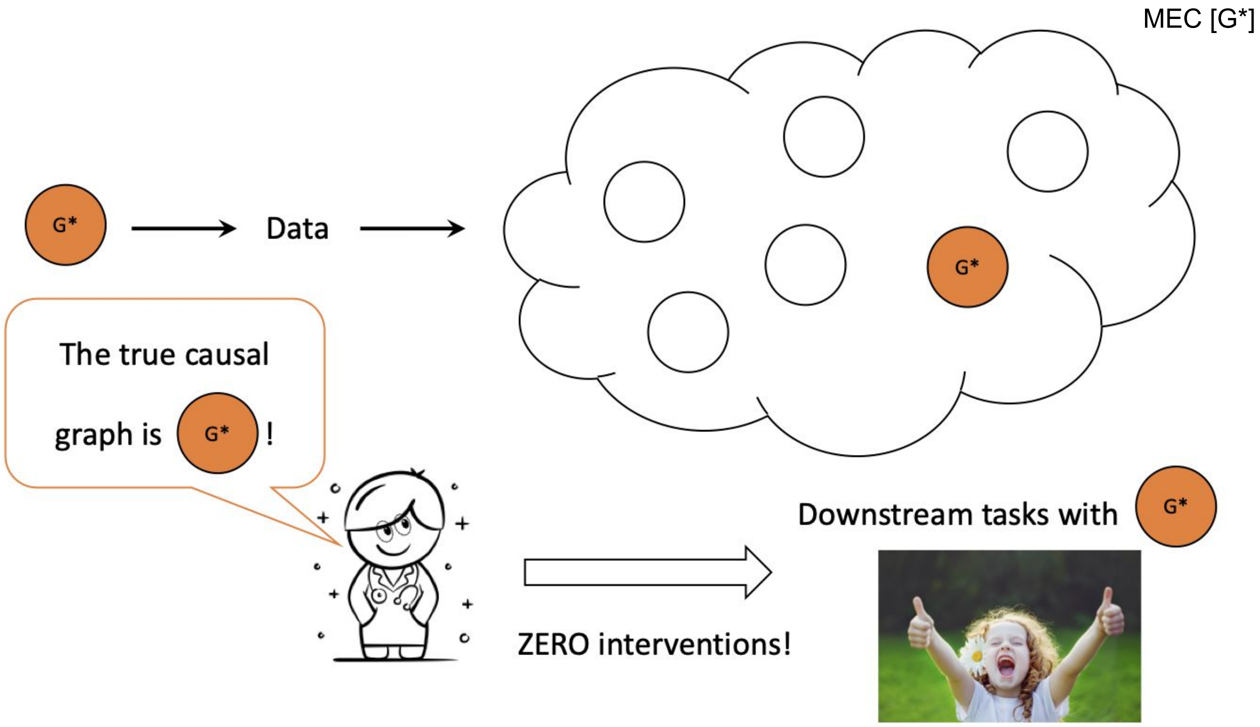


Image credit: https://thenounproject.com/icon/look-4968809/

There are domain experts!



But… experts can be wrong

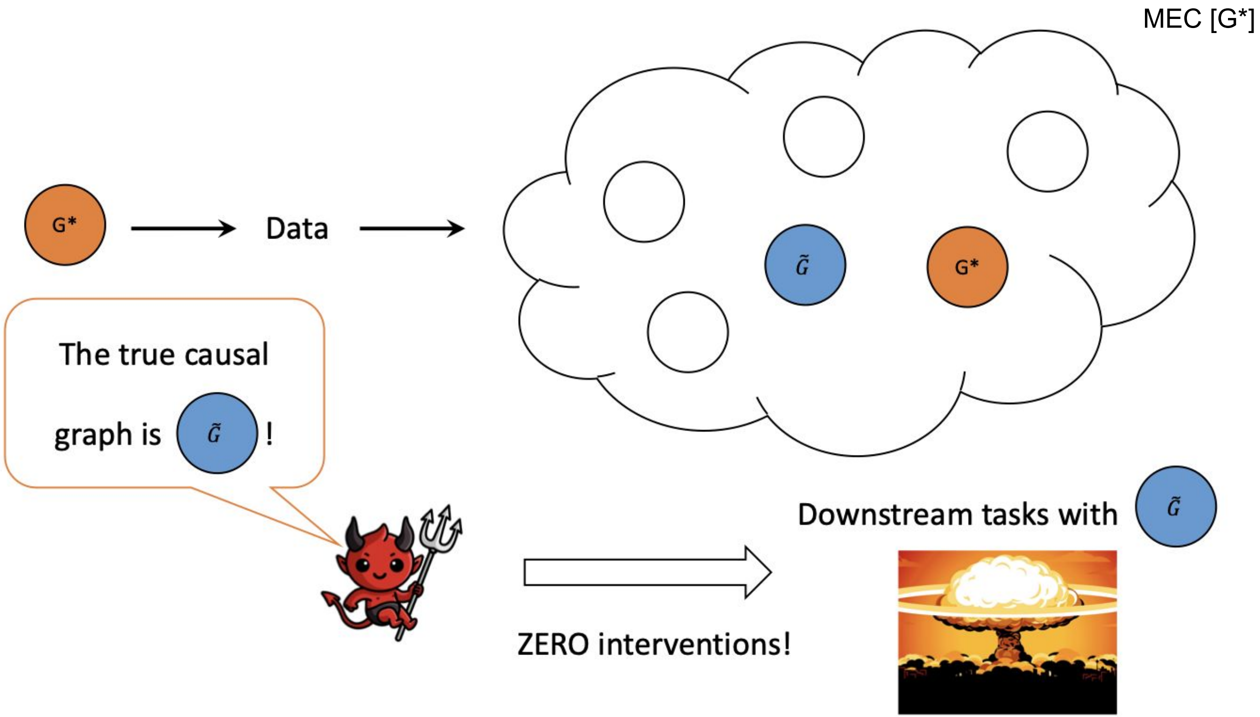
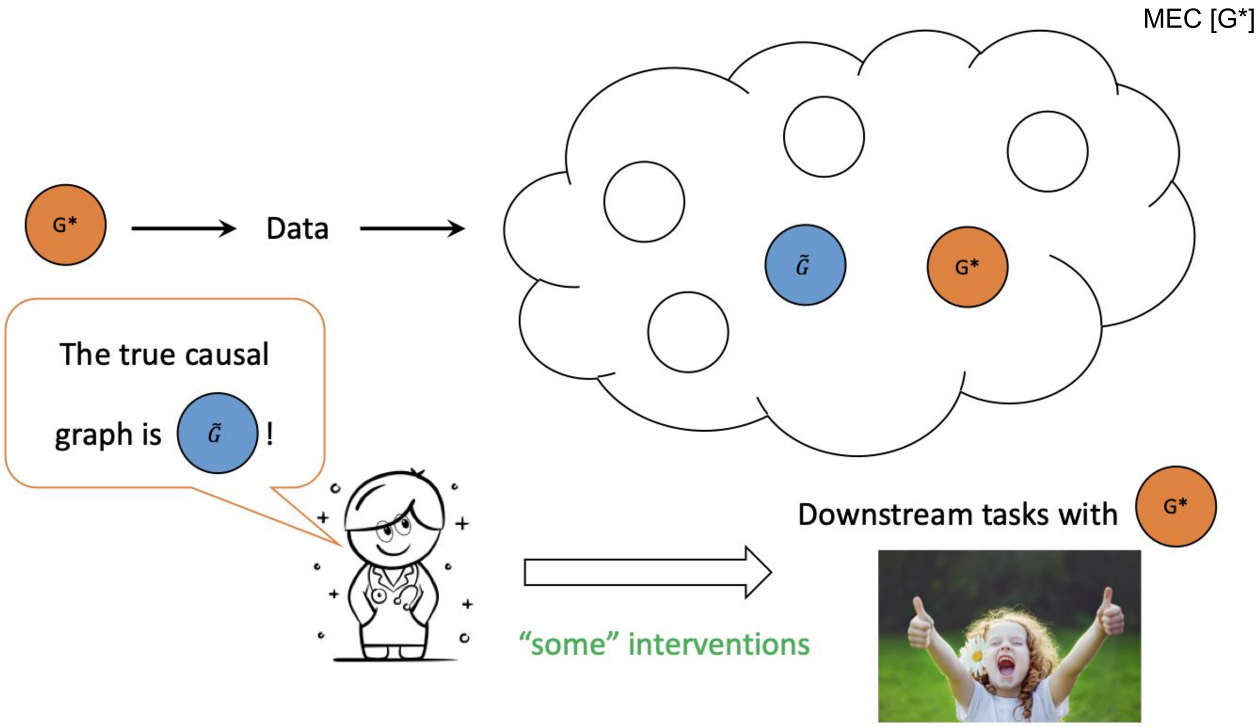


Image credit: https://dribbble.com/shots/3759014-Atomic-Illustrations/attachments/3759014-Atomic-Illustrations?mode=media

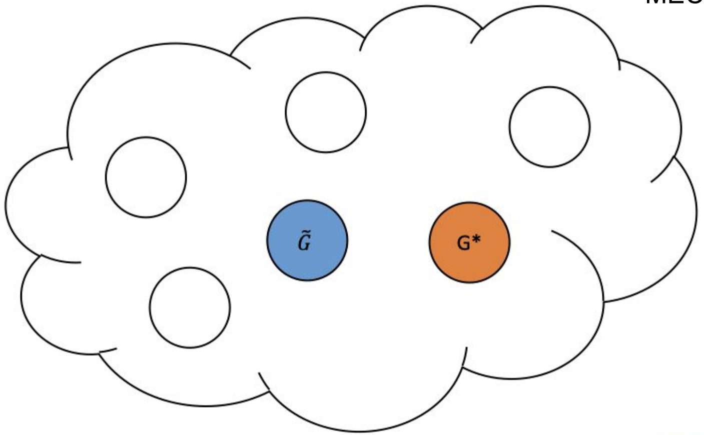
Searching with imperfect advice



Searching with imperfect advice

G* ⟶ Data ⟶

MEC [G*]



The true causal graph is $\tilde{G}$!

Quality of advice $\tilde{G}$

$0 \leq \psi(G^*, \tilde{G}) \leq n$

(good) (bad)

Downstream tasks with G*

"some" interventions

Advice search: $\mathcal{O}\left(\max\{1, \log \psi(G^*, \tilde{G})\} \cdot \nu(G^*)\right)$ interventions

[CGB23]