Adaptivity Complexity for Causal Graph Discovery (Supplementary Material)

Davin Choo[*1] **Kirankumar Shiragur**[*2]

[1] School of Computing, National University of Singapore
[2] Broad Institute of MIT and Harvard

A MEEK RULES

Meek rules are a set of 4 edge orientation rules that are sound and complete with respect to any given set of arcs that has a consistent DAG extension [Meek, 1995]. Given any edge orientation information, one can always repeatedly apply Meek rules till a fixed point to maximize the number of oriented arcs.

Definition 1 (Consistent extension)**.** A set of arcs is said to have a *consistent DAG extension* π for a graph G if there exists a permutation on the vertices such that (i) every edge $\{u, v\}$ in G is oriented $u \to v$ whenever $\pi(u) < \pi(v)$, (ii) there is no directed cycle, (iii) all the given arcs are present.

Definition 2 (The four Meek rules [Meek, 1995], see Fig. 1 for an illustration)**.**

R1 Edge $\{a, b\} \in E \setminus A$ is oriented as $a \to b$ if $\exists\, c \in V$ such that $c \to a$ and $c \not\sim b$.

R2 Edge $\{a, b\} \in E \setminus A$ is oriented as $a \to b$ if $\exists\, c \in V$ such that $a \to c \to b$.

R3 Edge $\{a, b\} \in E \setminus A$ is oriented as $a \to b$ if $\exists\, c, d \in V$ such that $d \sim a \sim c$, $d \to b \leftarrow c$, and $c \not\sim d$.

R4 Edge $\{a, b\} \in E \setminus A$ is oriented as $a \to b$ if $\exists\, c, d \in V$ such that $d \sim a \sim c$, $d \to c \to b$, and $b \not\sim d$.

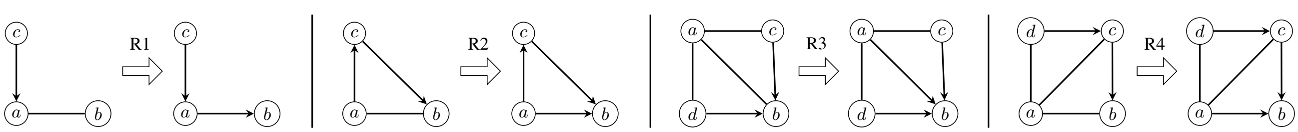


Figure 1: An illustration of the four Meek rules

There exists an algorithm [Wienöbst et al., 2021, Algorithm 2] that runs in $\mathcal{O}(d \cdot |E|)$ time and computes the closure under Meek rules, where d is the degeneracy of the graph skeleton[1].

B DEFERRED DETAILS

B.1 BASIC RESULTS

Lemma 3 (Equation 3.10 of [Graham et al., 1994])**.** *Let $f(x)$ be any continuous, monotonically increasing function with the property that x is an integer if $f(x)$ is an integer. Then, $\lceil f(x) \rceil = \lceil f(\lceil x \rceil) \rceil$.*

[*] Equal contribution

[1] A d-degenerate graph is an undirected graph in which every subgraph has a vertex of degree at most d. Note that the degeneracy of a graph is typically smaller than the maximum degree of the graph.

Accepted for the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023).

Lemma 17. *For positive integer n, and arbitrary real numbers m, x, we have* $\left\lceil \frac{\lceil \frac{x}{m} \rceil}{n} \right\rceil = \lceil \frac{x}{mn} \rceil$.

Proof. Apply Lemma 3 with the function as $f(x) = x/n$ on input as x/m. ☐

Lemma 4. *For $r \geq 2$, we have $\frac{r-1}{2} \cdot \left(\frac{2}{r}\right)^{\frac{1}{r-1}} \geq \frac{r}{4}$.*

Proof. Multiplying the left-hand side by $4/r$, we get

$$
\begin{aligned}
(r-1) \cdot \left(\frac{2}{r}\right)^{1+\frac{1}{r-1}} &\geq (r-1) \cdot \left(\frac{2}{r}\right) && \text{Since } r > 1 \\
&\geq 1 && \text{Since } r \geq 2
\end{aligned}
$$

Thus, the inequality holds. ☐

Lemma 5 (Theorem 12 of [Choo et al., 2022])**.** *For any causal DAG G^*, we have $\nu_k(G^*) \geq \lceil \frac{\nu_1(G^*)}{k} \rceil$.*

B.2 ALGORITHM FOR BOUNDED SIZE INTERVENTIONS

Algorithm 3 Adaptivity-sensitive search.

Input: Essential graph $\mathcal{E}(G^*)$, adaptivity round parameter $r \geq 1$, intervention size upper bound $k \geq 1$.
Output: A sequence of intervention sets $\mathcal{I}_1, \ldots, \mathcal{I}_r$ such that $\mathcal{E}_{\mathcal{I}_1,\ldots,\mathcal{I}_r}(G^*) = G^*$ and $|I| \leq k$ for any intervention in $I \in \mathcal{I}_i$ in intervention set $\mathcal{I}_i$, $1 \leq i \leq r$.

Initialize $L = \lceil n^{1/r} \rceil$.
for $i = 1, \ldots, r-1$ **do**
  Initialize $\mathcal{I}_i \leftarrow \emptyset$
  for chain comp. $H \in CC(\mathcal{E}_{\mathcal{I}_1,\ldots,\mathcal{I}_{i-1}}(G^*))$ **do**
    if H is a clique **then**
      Set $V' \leftarrow V(H)$.
    else
      Compute clique tree T_H of H.
      Compute L-balanced partitioning S of T_H via Algorithm 1.
      Let $V' \leftarrow \cup_{K_j \in S} V(K_j)$.
    end if
    Add output of Algorithm 4 on V' to $\mathcal{I}$.
  end for
  Intervene on all interventions in $\mathcal{I}_i$.
end for
Define $\mathcal{I}_r$ as output of Algorithm 4 on remaining relevant vertices and intervene on all interventions in $\mathcal{I}_r$.
return $\mathcal{I}_1, \ldots, \mathcal{I}_r$

Algorithm 4 Intervention subroutine.

Input: Set of vertices A, size upper bound $k \geq 1$.
Output: A k-separating system $B \subseteq 2^A$.

if $k = 1$ **then**
  Set $B \leftarrow A$.
else
  Define $k' = \min\{k, |A|/2\}$, $a = \lceil |A|/k' \rceil \geq 2$, and $\ell = \lceil \log_a n \rceil$.
  Compute labelling scheme of [Shanmugam et al., 2015, Lemma 1] on A with $(|A|, k', a)$.
  Set $B \leftarrow \{S_{x,y}\}_{x \in [\ell], y \in [a]}$, where $S_{x,y} \subseteq A$ is the subset of vertices whose x^{th} letter in the label is y.
end if
return B

C DEFERRED PROOFS

Theorem 2 (Atomic worst case)**.** *In the worst case, any r-adaptive algorithm needs to use at least $\Omega(\min\{r, \log n\} \cdot n^{1/\min\{r,\log n\}} \cdot \nu_1(G^*))$ atomic interventions against an adaptive adversary.*

Proof. Without loss of generality, we may assume $r \leq \log n$ and prove a lower bound of $\Omega(r \cdot n^{1/r} \cdot \nu_1(G^*))$.

Consider the case where the essential graph is a path on n nodes and the adversary can adaptively choose the source node as long as it is consistent with the arc directions revealed thus far. On a path essential graph, $\nu(G^*) = 1$.

Suppose $r = 1$. Then, by Theorem 13, we need to intervene on a G-separating system, which has size $\Omega(n)$. The claim follows since $\nu(G^*) = 1$.

Now, suppose $r \geq 2$. If currently have length ℓ segment and k interventions are performed, then there must be some segment of length at least $\ell/(k+1)$. Recurse on that. If the final round has length ℓ segment, need at least $\ell/2$ interventions because G-separating system on a segment of length ℓ has size at least $\ell/2$.

Suppose the algorithm intervenes on k_i vertices on the i-th round, for $1 \leq i \leq r$. where $k_i \geq 1$, so $k_i + 1 \leq 2k_i$ and so $1/(k_i+1) \geq 1/(2k_i)$.

Then, from the above discussion,

$$\begin{aligned} k_r &\geq \frac{1}{2} \cdot n \cdot \frac{1}{k_1+1} \cdot \frac{1}{k_2+1} \cdot \ldots \cdot \frac{1}{k_{r-1}+1} \\ &\geq \frac{1}{2^r} \cdot \frac{n}{k_1 \cdot k_2 \ldots \cdot k_{r-1}} \end{aligned}$$

So, the number of overall interventions used is

$$\begin{aligned} & k_1 + \ldots + k_r \\ \geq\ & k_1 + \ldots + k_{r-1} + \frac{1}{2^r} \cdot \frac{n}{k_1 \cdot k_2 \ldots \cdot k_{r-1}} \\ \geq\ & (r-1) \cdot \left(\prod_{i=1}^{r-1} k_i \right)^{\frac{1}{r-1}} + \frac{1}{2^r} \cdot \frac{n}{k_1 \cdot k_2 \ldots \cdot k_{r-1}} \end{aligned}$$

where the last inequality is the AM-GM inequality.

Let $x = k_1 \cdot k_2 \ldots \cdot k_{r-1}$. Then,

$$\sum_{i=1}^{r} k_i = k_1 + \ldots + k_r \geq (r-1) \cdot x^{\frac{1}{r-1}} + \frac{1}{2^r} \cdot \frac{n}{x}$$

Case 1: $\frac{1}{2^r} \cdot \frac{n}{x} \geq \frac{r}{4} \cdot n^{\frac{1}{r}}$

Then,

$$\sum_{i=1}^{r} k_i \geq \frac{1}{2^r} \cdot \frac{n}{x} \geq \frac{r}{4} \cdot n^{\frac{1}{r}} \in \Omega(r \cdot n^{\frac{1}{r}})$$

Thus, the claim holds as $\nu(G^*) = 1$.

Case 2: $\frac{1}{2^r} \cdot \frac{n}{x} < \frac{r}{4} \cdot n^{\frac{1}{r}}$

Then,

$$x > \frac{4 \cdot n^{1-1/r}}{2^r \cdot r} = \frac{2 \cdot n^{\frac{r-1}{r}}}{2^{r-1} \cdot r}$$

and Lemma 4 in Appendix B.1 tells us that

$$\begin{aligned} (r-1) \cdot x^{\frac{1}{r-1}} &> n^{\frac{1}{r}} \cdot \frac{r-1}{2} \cdot \left(\frac{2}{r}\right)^{\frac{1}{r-1}} \\ &\geq n^{\frac{1}{r}} \cdot \frac{r}{4} && \text{For } r \geq 2 \end{aligned}$$

So,

$$\sum_{i=1}^{r} k_i \geq (r-1) \cdot x^{\frac{1}{r-1}} \geq \frac{r}{4} \cdot n^{\frac{1}{r}} \in \Omega(r \cdot n^{\frac{1}{r}})$$

Thus, the claim holds as $\nu(G^*) = 1$. □

Theorem 3 (Bounded upper bound)**.** *Let $\mathcal{E}(G^*)$ be the observational essential graph of an underlying causal DAG G^* on n nodes. There is a polynomial time r-adaptive algorithm that uses $\mathcal{O}(\min\{r, \log n\} \cdot n^{1/\min\{r, \log n\}} \cdot \log k \cdot \nu_k(G^*))$ bounded sized interventions to recover G^* from $\mathcal{E}(G^*)$, where each intervention involves at most $k > 1$ vertices.*

Proof. We invoke Algorithm 3 with $k > 1$.

Number of interventions

The high level proof approach for is exactly the same as the proof of Theorem 1, except for how to compute intervention sets from the maximal clique vertices (obtained by "balanced partitioning" in the first $r-1$ rounds, within the while loop) and the from the remaining relevant vertices (in the final r-th round, outside the while loop).

In each iteration of the while-loop, we intervene on at most L cliques for each connected component. To orient the edges incident to these cliques we use the labelling scheme of Lemma 14 via Algorithm 4. So, the number of bounded size interventions we perform per round is

$$\mathcal{O}\left(L \cdot \log k \cdot \frac{\nu_1(G^*)}{k}\right)$$

By Lemma 5, we know that $\nu_k(G^*) \geq \lceil \frac{\nu_1(G^*)}{k} \rceil$. So, we can re-express the above bound as $\mathcal{O}\left(L \cdot \log k \cdot \nu_k(G^*)\right)$. Similarly, we use $\mathcal{O}\left(L \cdot \log k \cdot \nu_k(G^*)\right)$ bounded size interventions in the final round. Thus, over all r adaptive rounds, we use a total of

$$\mathcal{O}\left(r \cdot L \cdot \log k \cdot \nu_k(G^*)\right)$$

bounded size interventions. Substituting $L = \lceil n^{1/r} \rceil$ yields our desired bound.

Running time

Algorithm 3 only differs from Algorithm 2 by invoking Algorithm 4, which runs in polynomial time (see Lemma 14). Thus, Algorithm 3 runs in polynomial time. □

D EXPERIMENTS

The experiments are conducted on an Ubuntu server with two AMD EPYC 7532 CPU and 256GB DDR4 RAM. Our code and entire experimental setup is available at `https://github.com/cxjdavin/adaptivity-complexity-for-causal-graph-discovery`.

D.1 IMPLEMENTATION DETAILS

Checks to avoid redundant interventions The current implementation of [Choo et al., 2022]'s `separator` algorithm is actually n-adaptive because it performs "checks" before performing each intervention — if the vertices in the proposed intervention set S do *not* have any unoriented incident arcs, then the intervention set S will be skipped. One may think of such interventions as "redundant" since they do not yield any new information about the underlying causal graph. As such, we ran two versions of their algorithm: one without checks (i.e. $\mathcal{O}(\log n)$-adaptive) and one with checks (i.e. n-adaptive). Note that each check corresponds to an adaptivity round because an intervention within a batch of interventions may turn out to be redundant, but we will only know this after performing a check after some of the interventions within that batch have been executed.

Scaling our algorithm with checks Since $n^{\frac{1}{\log n}} = 2$, running Algorithm 2 (as it is) with adaptivity parameters $r \in \Omega(\log n)$ does not make much sense. As such, we define a checking budget $b = r - \lceil \log n \rceil$ and greedily perform up to b checks whilst executing Algorithm 2. This allows Algorithm 2 to scale naturally for $r \in \Omega(\log n)$.

Non-adaptive intervention round For the final round of interventions, let V' be the set of remaining relevant vertices. From our algorithm, we know that $|V'| \leq L$ but we may even intervene on less vertices in the final round. By [Kocaoglu et al., 2017], we only need to intervene on a graph-separating system of the subgraph $G[V']$. For atomic interventions, this exactly correspond to the minimum vertex cover of V'. To obtain this, we first compute the maximum independent set S of V' (which can be computed efficiently on chordal graphs [Gavril, 1972, Leung, 1984]), then only intervene on $V' \setminus S$.

Optimization before final round Note that we can always compute the intervention set $F \subseteq V$ which we *would* have intervened if $r = 1$. At any point in time of the algorithm, if F involves less vertices than the number of vertices required from the L-partitioning, then we simply treat the current adaptivity round as the final round, choose to intervene on F and use any remaining adaptive budget for performing checks.

D.2 SYNTHETIC GRAPHS

We use synthetic moral randomly generated graphs from earlier prior works [Choo et al., 2022, Squires et al., 2020, Choo and Shiragur, 2023]. For each of the graph classes and parameters, we generate 100 DAGs and plot the average with an error bar.

1. Erdős-Rényi styled graphs (used by [Squires et al., 2020, Choo et al., 2022])
   These graphs are parameterized by 2 parameters: number of nodes n and density ρ. Generate a random ordering σ over n vertices. Then, set the in-degree of the n^{th} vertex (i.e. last vertex in the ordering) in the order to be $X_n = \max\{1, \texttt{Binomial}(n-1, \rho)\}$, and sample X_n parents uniformly form the nodes earlier in the ordering. Finally, chordalize the graph by running the elimination algorithm of [Koller and Friedman, 2009] with elimination ordering equal to the reverse of σ.
   Parameters used: $n = \{10, 15, 20, \ldots, 95, 100\}$ and $\rho = 0.1$.
2. Tree-like graphs (used by [Squires et al., 2020, Choo et al., 2022])
   These graphs are parameterized by 4 parameters: number of nodes n, degree d, $e_{\min}$, and $e_{\max}$. First, generate a complete directed d-ary tree on n nodes. Then, add $\texttt{Uniform}(e_{\min}, e_{\max})$ edges to the tree. Finally, compute a topological order of the graph by DFS and triangulate the graph using that order. As the original definition of this graph class by [Squires et al., 2020] becomes very sparse as n grows, we tweaked the other parameters to scale accordingly by defining new parameters $d_{prop}, e_{\min,prop}, e_{\max,prop} \in [0, 1]$ as follows: $d = n \cdot d_{prop}$, $e_{\min} = n \cdot e_{\min,prop}$, and $e_{\max} = n \cdot e_{\max,prop}$.
   Parameters used: $n = \{100, 150, 200, \ldots, 450, 500\}$, $d_{prop} = 0.4$, $e_{\min,prop} = 0.2$, $e_{\max,prop} = 0.5$.
3. $G(n, p)$-union-tree (used by [Choo and Shiragur, 2023])
   These graphs are parameterized by 2 parameters: number of nodes n and edge probability p. An Erdős-Rényi $G(n, p)$ and a random tree T on n vertices are generated. Take the union of their edge sets, orient the edges in an acyclic fashion, then add arcs to remove v-structures.
   Parameters used: $n = \{10, 15, 20, \ldots, 95, 100\}$ and $p = 0.03$.

D.3 ALGORITHMS BENCHMARKED

While both the algorithm of [Choo et al., 2022] and Algorithm 2 have been implemented to take in a parameter k for bounded-size interventions, our experiments focused on the case of atomic interventions, i.e. $k = 1$.

`separator`: Algorithm of [Choo et al., 2022]. With checks, it allows for full adaptivity.

`separator_no_check`: `separator` but we remove checks that avoid redundant interventions, i.e. $\mathcal{O}(\log n)$ rounds of adaptivity.

`adaptive_r1`: Algorithm 2 with $r = 1$, i.e. non-adaptive

`adaptive_r2`: Algorithm 2 with $r = 2$

`adaptive_r3`: Algorithm 2 with $r = 3$

`adaptive_rlogn`: Algorithm 2 with $r = \log_2 n$

`adaptive_r2logn`: Algorithm 2 with $r = 2\log_2 n$. Can perform checks that avoid redundant interventions.

`adaptive_r3logn`: Algorithm 2 with $r = 3\log_2 n$. Can perform checks that avoid redundant interventions.

`adaptive_rn`: Algorithm 2 with $r = n$, i.e. full adaptivity allowed

D.4 EXPERIMENTAL RESULTS

As expected, we observe that higher rounds of adaptivity leads to lower number of interventions required. When $r \in \mathcal{O}(\log n)$, Algorithm 2 can match [Choo et al., 2022] with checks disabled. When $r = n$, Algorithm 2 can match [Choo et al., 2022] with its full adaptivity.

References

Davin Choo and Kirankumar Shiragur. Subset verification and search algorithms for causal DAGs. In *International Conference on Artificial Intelligence and Statistics*, 2023.

Davin Choo, Kirankumar Shiragur, and Arnab Bhattacharyya. Verification and search algorithms for causal DAGs. *Advances in Neural Information Processing Systems*, 35, 2022.

Fănică Gavril. Algorithms for minimum coloring, maximum clique, minimum covering by cliques, and maximum independent set of a chordal graph. *SIAM Journal on Computing*, 1(2):180–187, 1972.

Ronald L Graham, Donald E Knuth, and Oren Patashnik. Concrete mathematics: A foundation for computer science, 1994.

Murat Kocaoglu, Alex Dimakis, and Sriram Vishwanath. Cost-Optimal Learning of Causal Graphs. In *International Conference on Machine Learning*, pages 1875–1884. PMLR, 2017.

Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

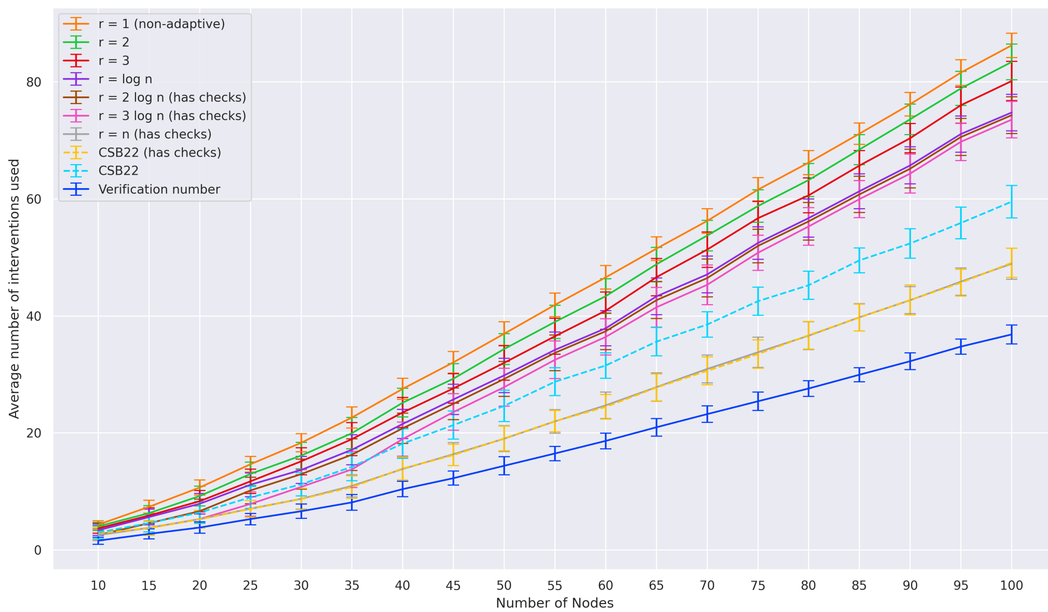
Joseph Y-T Leung. Fast algorithms for generating all maximal independent sets of interval, circular-arc and chordal graphs. *Journal of Algorithms*, 5(1):22–35, 1984.

Christopher Meek. Causal Inference and Causal Explanation with Background Knowledge. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, UAI'95, page 403–410, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc. ISBN 1558603859.

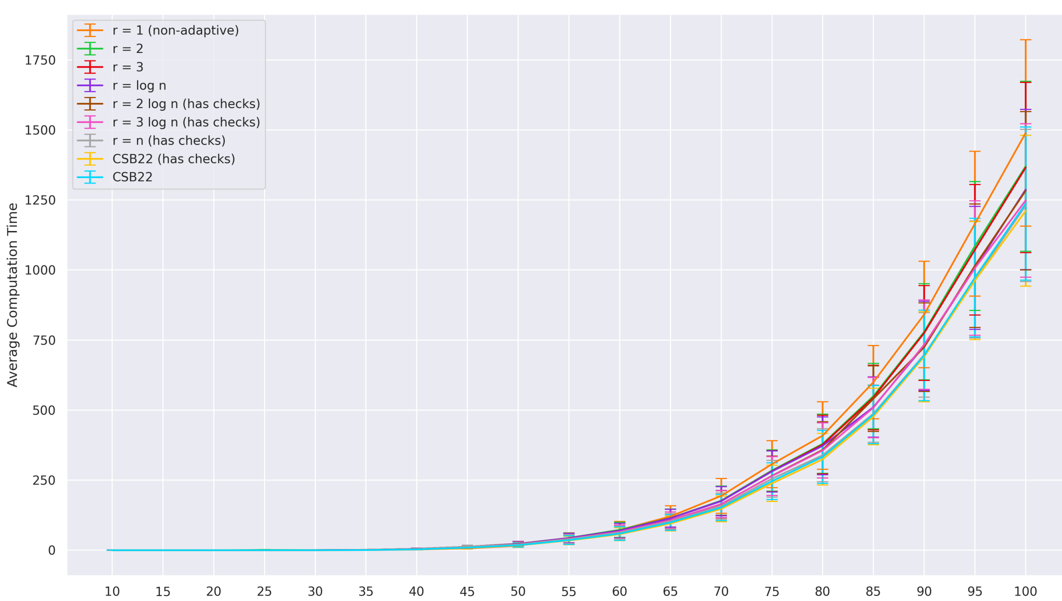
Karthikeyan Shanmugam, Murat Kocaoglu, Alexandros G. Dimakis, and Sriram Vishwanath. Learning Causal Graphs with Small Interventions. *Advances in Neural Information Processing Systems*, 28, 2015.

Chandler Squires, Sara Magliacane, Kristjan Greenewald, Dmitriy Katz, Murat Kocaoglu, and Karthikeyan Shanmugam. Active Structure Learning of Causal DAGs via Directed Clique Trees. *Advances in Neural Information Processing Systems*, 33:21500–21511, 2020.

Marcel Wienöbst, Max Bannach, and Maciej Liśkiewicz. Extendability of causal graphical models: Algorithms and computational complexity. In Cassio de Campos and Marloes H. Maathuis, editors, *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, pages 1248–1257. PMLR, 27–30 Jul 2021. URL `https://proceedings.mlr.press/v161/wienobst21a.html`.

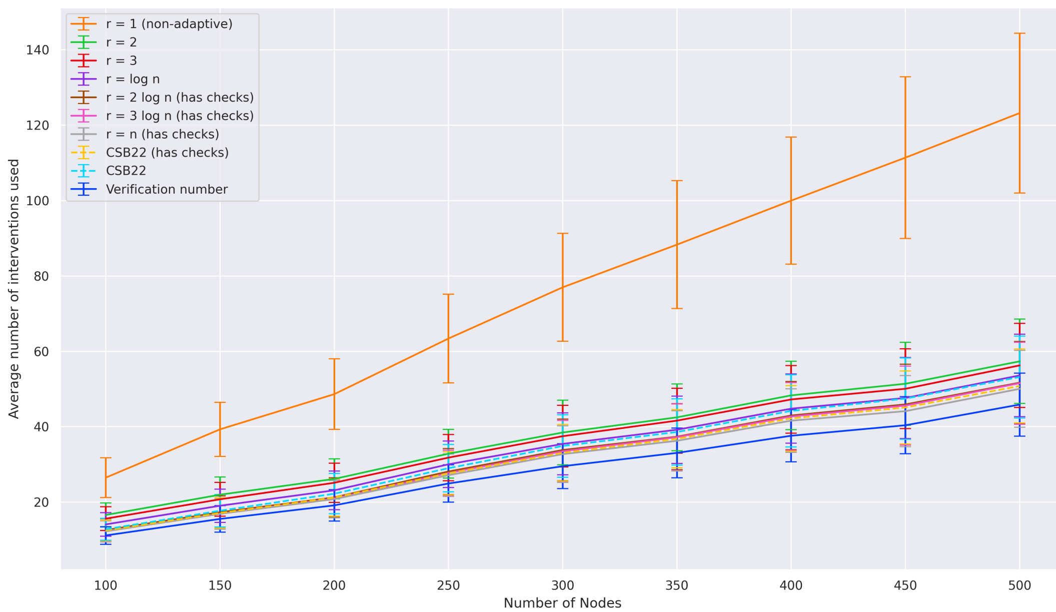


(a) Number of interventions

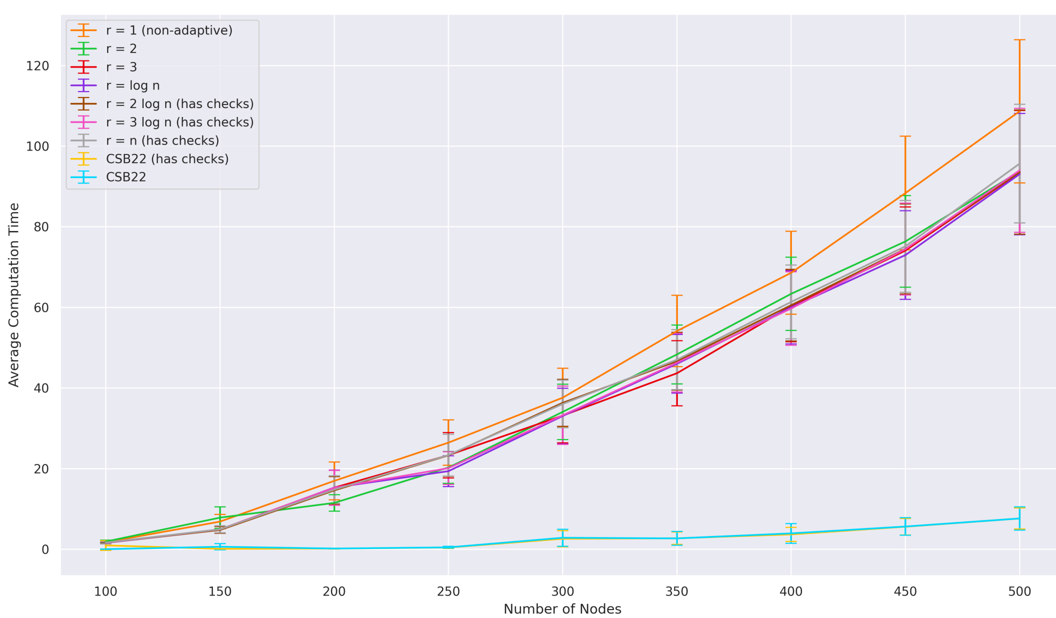


(b) Time

Figure 2: Experiment 1

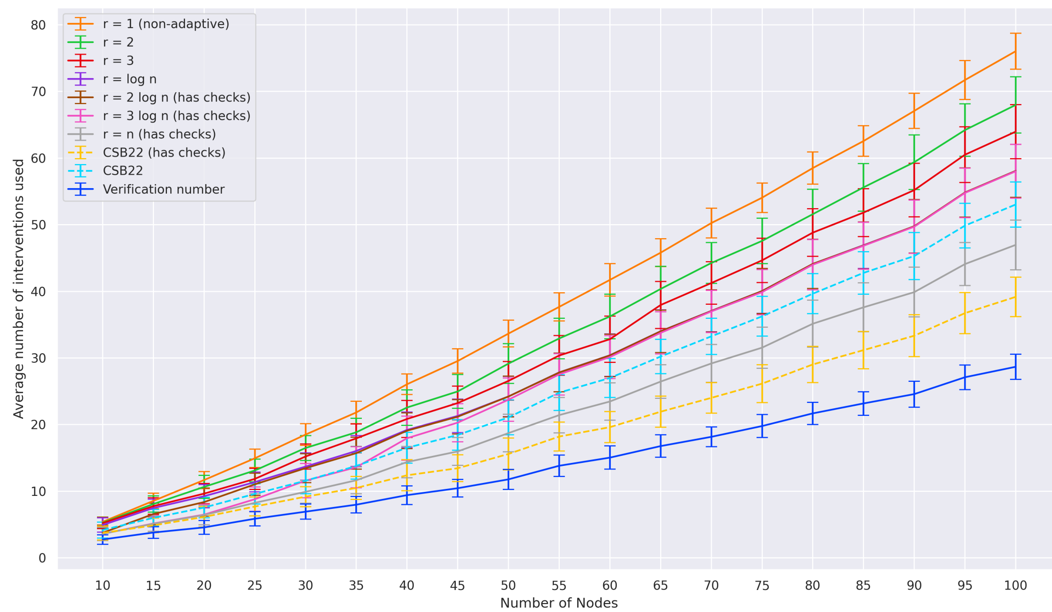


(a) Number of interventions

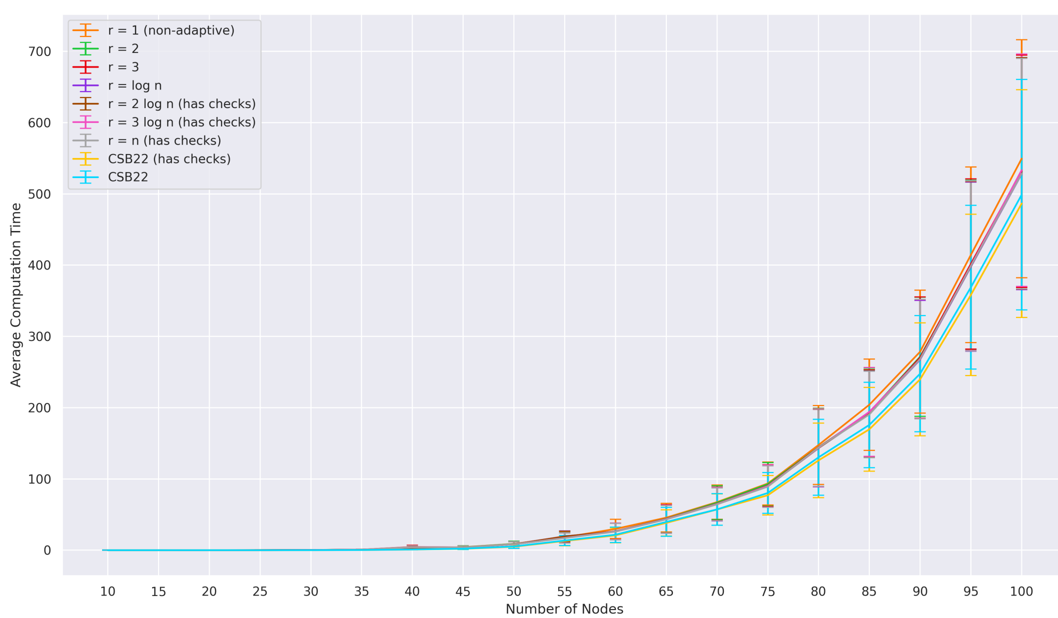


(b) Time

Figure 3: Experiment 2



(a) Number of interventions



(b) Time

Figure 4: Experiment 3