Causal Structure Learning With Momentum: Sampling Distributions Over Markov Equivalence Classes

Moritz Schauer smoritz@chalmers.se
Chalmers University of Technology and University of Gothenburg

Marcel Wienöbst m.wienoebst@uni-luebeck.de
Institute for Theoretical Computer Science, University of Lübeck

Editors: J.H.P. Kwisthout & S. Renooij

Abstract

In the context of inferring a Bayesian network structure (directed acyclic graph, DAG for short), we devise a non-reversible continuous time Markov chain, the "Causal Zig-Zag sampler", that targets a probability distribution over classes of observationally equivalent (Markov equivalent) DAGs. The classes are represented as completed partially directed acyclic graphs (CPDAGs). The non-reversible Markov chain relies on the operators used in Chickering's Greedy Equivalence Search (GES) and is endowed with a momentum variable, which improves mixing significantly as we show empirically. The possible target distributions include posterior distributions based on a prior over DAGs and a Markov equivalent likelihood. We offer an efficient implementation wherein we develop new algorithms for listing, counting, uniformly sampling, and applying possible moves of the GES operators, all of which significantly improve upon the state-of-the-art run-time.

Keywords: MCMC; Causal Discovery; Markov Equivalence Classes; DAGs.

1. Introduction

A Bayesian network is a probabilistic graphical model that represents a set of random variables and their conditional (in)dependencies using a directed acyclic graph (DAG). Graph and random variables are linked by the local Markov condition: variables are conditionally independent of their non-descendants given their parents, which induces a factorisation of the joint distribution via conditional distributions of variables given their parents (Lauritzen, 1996; Koller and Friedman, 2009). Typically, there are multiple such factorisations or multiple DAGs such that the local Markov condition holds.

Causal Bayesian networks, in which the edges in the DAG represent direct causal influences, provide a theory of how interventions change the joint distribution of latent and observable variables (Pearl, 2009; Peters et al., 2017). Here, one assumes the *causal* Markov condition that a variable conditional on its direct causes is independent of variables that are not directly or indirectly influenced by it. Therefore, even when assuming faithfulness, that all conditional independencies in the data are implied by the factorisation of the underlying causal DAG, observational data is generally insufficient to uniquely determine this graph. Instead the DAGs which cannot be told apart by observational data form a Markov equivalence class (MEC), that is an equivalence class of DAGs (Verma and Pearl, 1990; Heckerman et al., 1995), usually represented by a *completed partially directed graph* (CPDAG).[1] In Bayesian inference this manifests in marginal likelihoods that are the same

[1] Technical definitions are given in section 3.

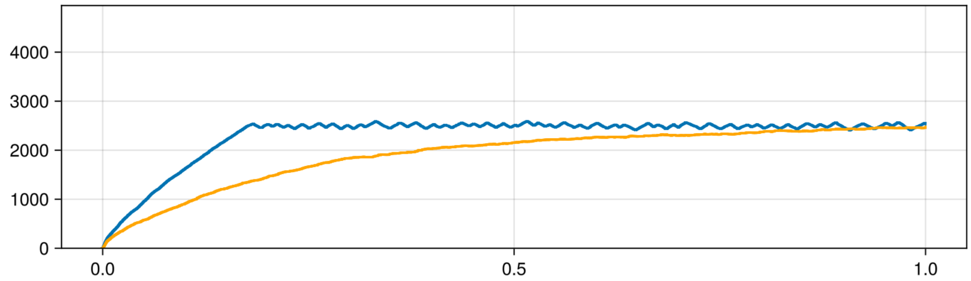

Figure 1: Continuous-time trace of the number of edges of the sampled graphs when targeting a uniform distribution on CPDAGs with 100 vertices for the non-reversible sampler proposed here in blue compared with the similar, but reversible, Zanella sampler in orange. The total time of 1 unit corresponds to 5 000 jumps. Our sampler reaches equilibrium considerably faster and mixes better.

for all members of the MEC. Bayesian inference starting from a prior distribution on the equivalence classes hence yields a posterior distribution over Markov equivalence classes.

Markovian Monte Carlo methods allow drawing samples from that posterior distribution. They work by constructing a stochastic process Z with a temporal Markov property,[2] that has the desired distribution as its equilibrium distribution, see Roberts and Rosenthal (2004) for a general account for Markov chains. The empirical distribution of samples taken from the process then approximates the usually intractable posterior distribution. This is classically done with discrete time Markov chains, but recently continuous time samplers have also become an active research area (Fearnhead et al., 2018).

In this work, the sampler is based on a stochastic process $Z = (Z_t)_{t \geq 0}$ taking values (γ, d) in a space of extended coordinates where $\gamma \in \mathcal{M}_n$ is a MEC on n variables and $d \in \{-1, +1\}$ is variable indicating a direction of movement corresponding to adding edges to γ if $d = +1$ and removing edges from γ if $d = -1$. This is analogous to Gustafson (1998) who adds a direction variable to a random walk on the integers in order to improve mixing by allowing for repeated moves in the same direction in contrast to choosing a random direction in every step. Also Hamiltonian Monte Carlo uses a momentum variable to balance random walk behaviour and systematic exploration (Neal, 1996).

The sampler relies on the operators introduced by Chickering (2002b) in the celebrated GES algorithm for estimating a single MEC. They allow to move between MECs, which have DAG members that differ only by a single edge deletion or insertion, thus providing a natural and efficient representation of this space. Moreover, they can be used to immediately obtain a reversible Markov chain, as for example recently explored by Zhou and Chang (2023) which propose a locally balanced Markov chain sampler in the sense of Zanella (2019) for the problem. Endowing them with momentum improves mixing and retains closeness to the GES approach, where there are two main phases: (i) the forward phase, during which edges are inserted and (ii) the backward phase, during which edges are deleted. Indeed, our algorithm, which we term *Causal Zig-Zag*, can be viewed as a generalisation of GES and it

[2] Not to be confused with the local Markov condition and the causal Markov assumption in the Bayesian network.

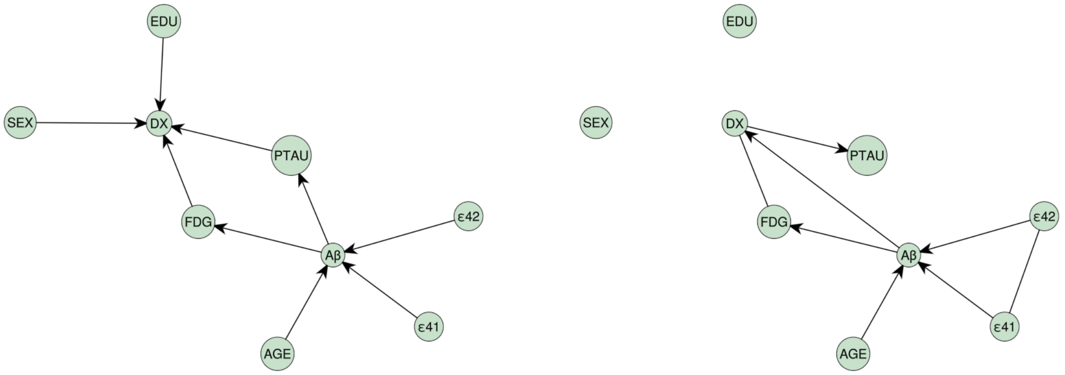

Figure 2: The expert assessment of the causal model on the left. On the right, the model with highest posterior probability 0.701, which coincides with the model found by GES. The two models with next highest posterior probabilities are shown in Figure 6 in the Appendix.

converges to it in the limit of increasing coldness given by a thermodynamic β as we show in section 6. Because GES itself provably recovers the MEC of the underlying true DAG in the limit of large sample size, this translates to Causal Zig-Zag, which is effective in finding high-posterior regions. More generally, we make the following contributions.

1. We present a sampler for Markov equivalence classes that is both non-reversible and locally balanced with application to Bayesian causal discovery and causal discovery with uncertainty quantification. Similar to the GES algorithm the sampler operates in alternating phases, one phase where edges are inserted and one phase where edges are removed. This makes the sampler non-reversible and improves mixing.

2. We base the sampler on new, efficient algorithms for listing, counting and applying possible moves in the space of MECs based on Chickering's Insert and Delete operators. These improvements go beyond the use cases in this work and also apply to the original GES and related algorithms.

3. We show the benefits and practicality of our approach empirically and make our implementation available in the software package CausalInference.jl.

As first illustration, we use our non-reversible sampler and a reversible counterpart to sample CPDAGs with 100 vertices uniformly. Both samplers start from the empty graph and continue for 5 000 steps. The samplers require no further choice of tuning parameters. Our sampler reaches equilibrium considerably faster, see figure 1. The time of reaching a large set such as, in this case, CPDAGs with 2400 to 2600 edges, from a single state (the empty graph) is informative about mixing times (Peres and Sousi, 2013).

As second illustration, we partly reproduce Shen et al. (2020). They consider data from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (`adni.loni.usc.edu`).[3] The variables extracted from the data are fludeoxyglucose PET (FDG), amyloid beta (Aβ), phosphorylated tau (PTAU), number of $\varepsilon 4$ alleles of apolipoprotein E; demographic

[3] See acknowledgements for more information.

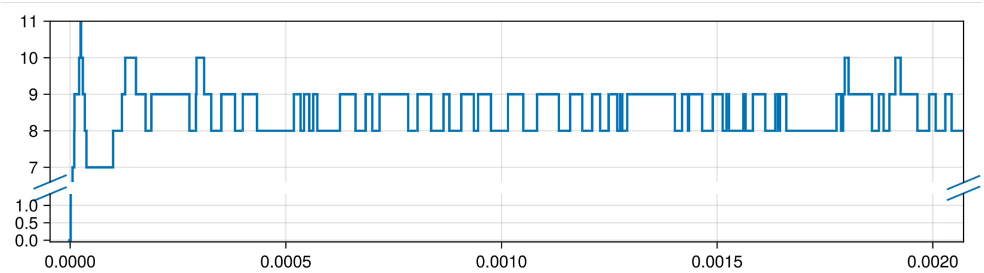

Figure 3: Continuous-time trace of the number of edges of the first sampled graphs for the ADNI data. At this time scale, the random time spend in each CPDAG is visible.

information: age, sex, years of education (EDU); and diagnosis on Alzheimer disease (DX). To account for possibly non-linear effects the number of $\varepsilon4$ alleles (0, 1, or 2) is dummy encoded ($\varepsilon42$, $\varepsilon41$), as it is done in Shen et al. (2020). We use our algorithm to sample CPDAGs proportional to their (exponentiated) BIC score with penalty 5.5 and run the sampler for 50 000 jumps starting from the empty graph. See section 4.1 in Chickering (2002b) for a discussion of the Bayesian Information Criterion (BIC) and its relationship to the marginal posterior. Our findings are shown in figures 2 and 3.

2. Related work

Bayesian methods for learning DAGs from observational data, which directly target the posterior probability over MECs, as we do in this work, are underrepresented in the literature with popular exact methods estimating the marginal posterior probability of every possible edge (Koivisto and Sood, 2004) and MCMC samplers focusing on the space of DAGs (Madigan et al., 1995; Giudici and Castelo, 2003; Grzegorczyk and Husmeier, 2008) or variable orderings (Friedman and Koller, 2000; Niinimäki et al., 2016; Kuipers and Moffa, 2017; Agrawal et al., 2018) being more widespread. Recently, differentiable formulations have been pursued and exploited by variational and MCMC methods (Lorch et al., 2021; Annadani et al., 2021; Cundy et al., 2021; Deleu et al., 2022; Annadani et al., 2023).

On the other hand, when aiming to estimate a single causal structure, classical algorithms such as PC (Spirtes et al., 2000) and GES (Chickering, 2002b) are at their core build on the notion of Markov equivalence. More generally, exploiting as well as analysing the space and properties of MECs has a long and fruitful history in the causal discovery and Bayesian network communities, beginning with Madigan et al. (1996), who pivoted the use of MCMC using the search space of MECs for Bayesian structure learning, and Gillispie and Perlman (2002), who initiated studies of the size distribution of MECs. Later these works were extended by Pena (2007); He et al. (2013), who again used MCMC to analyse, e.g, the average number of undirected edges in a CPDAG, focusing mainly on sparse graphs. Recently, Zhou and Chang (2023) showed that the GES operators by Chickering (2002a) have superior mixing properties compared to these earlier MCMC approaches. The sampler used by Zhou and Chang (2023) belongs to a class of discrete time locally balanced sampler in high dimensional spaces (Zanella, 2019). For the continuous time perspective, see (Power and Goldman, 2019).

3. Preliminaries

Graphs and notation. A partially directed graph, here short "graph", $G = (V, E)$ consists of a set of n vertices V and a set of m edges $E \subseteq V \times V$.[4] An undirected edge between vertices $x, y \in V$, denoted $x - y$, has both $(x, y) \in E$ and $(y, x) \in E$, and a directed edge $u \to v$ has $(x, y) \in E$ and $(y, x) \notin E$. Vertices linked by an edge (of any type) are *adjacent*, and vertices linked by undirected edges are *neighbours* of each other. We say that x is a *parent* of y if $x \to y$. We denote by $\mathrm{Pa}(x)$ and $\mathrm{Ne}(x)$ the set of parents and neighbors of x. A directed graph contains no undirected edges. A partially directed acyclic graph (PDAG) is a graph without directed cycles and a directed acyclic graph (DAG) is a *directed* graph with this property. We denote the space of DAGs over n vertices as $\mathcal{D}_n$. We let $\mathrm{U}(S)$ denote the uniform distribution on a set S. $\sqcup$ denotes the disjoint union of sets.

Markov equivalence classes. In case of a Bayesian network, the vertex set V is a set of random variables. A *v-structure* are vertices x, y, z such that $x \to y \leftarrow z$ and x, z are not adjacent. All DAGs on a vertex set V with the same set of v-structures and the same set of adjacencies are observationally equivalent or *Markov-equivalent* as shown by Verma and Pearl (1990) and form the Markov equivalence class (MEC). A CPDAG (completed PDAG) has $x \to y$, if $x \to y$ in each member of the equivalence class, and $x - y$, if there are DAGs G and G' in the MEC such that G contains $x \to y$ and G' contains $x \leftarrow y$. The CPDAG uniquely determines the MEC. We denote the space of CPDAGs or MECs as $\mathcal{M}_n$ and denote its elements by $\gamma, \eta, \dots \in \mathcal{M}_n$. A scoring function $\mathcal{D}_n \to [0, \infty)$ for DAGs is a *Markov equivalent score* if it assigns the same score to any DAG in the same MEC.

Markov jump process. Following Kallenberg (2002), a continuous time stochastic process $(Z_t)_{t \geq 0}$ on a countable state space S with almost surely right-continuous paths that are constant apart from isolated jumps with the temporal Markov property is a Markov jump process.[5]

In our case, the state space is the space of MECs $S = \mathcal{M}_n$ or the space of MECs extended by a direction or momentum, $S = \mathcal{M}_n \times \{+1, -1\}$, and an abstract notion of time inherent to the sampler, related but not identical to the run time of its implementation.

Denote the jump times of Z as $0 = \tau_0 < \tau_1 < \tau_2 < \dots$, these are random times τ where $Z_\tau \neq Z_{\tau-}$. The law of a Markov jump process can be described by

- the starting distribution $Z_0 \sim \nu$;
- the rate function $\Lambda \colon S \to [0, \infty)$ such that conditional on $Z_{\tau_i} = a$, $a \in S$, the time to the next jump $\tau_{i+1} - \tau_i$ is exponentially distributed with rate Λ depending on a;[6]
- a jump kernel, such that $Z_{\tau_{i+1}}$ has the conditional distribution κ_a given $Z_{\tau_i} = a$.

This entails by the Markov property that $\tau_1/\Lambda(Z_0)$, $(\tau_2 - \tau_1)/\Lambda(Z_{\tau_1})$, $\dots$ form an independent sequence of $\mathrm{Exp}(1)$ random variables and $Z_0, Z_{\tau_1}, Z_{\tau_2}, \dots$ an embedded discrete-time Markov chain where $P(Z_{\tau_i} = b \mid Z_{\tau_i} = a) = \kappa_a\{b\}$ with κ_a for $a \in S$ being a probability kernel $\sum_{b \in S} \kappa_a\{b\} = 1$ where $\kappa_a\{a\} = 0$ by construction.

[4] Excluding self-edges: $(x, x) \notin E$.

[5] We only consider time-homogeneous processes where $\mathrm{P}(Z_t = b \mid Z_s = a)$ only depends on $t - s$.

[6] So $\Lambda(a) = 1/(\mathrm{E}[\tau_{i+1} - \tau_i \mid Z_{\tau_i} = a])$.

We also define $\lambda(a \curvearrowright b) = \Lambda(a)\kappa_a\{b\}$ the specific rate of jumps from $a \in S$ to $b \in S$. Both total rate $\Lambda(a)$ and the jump kernel κ_a, $a \in S$, are determined by λ through $\Lambda(a) = \sum_{b \in S} \lambda(a \curvearrowright b)$ and $\kappa_a\{b\} = \frac{\lambda(a \curvearrowright b)}{\Lambda(a)},$, $b \in S$. This has intuitive meaning. As the minimum of independent exponential random variables with rates $\lambda(a \curvearrowright b_1), \dots, \lambda(a \curvearrowright b_k)$ is exponentially distributed with rate $\Lambda(a)$, one can either jump to a state drawn from κ_a after $\mathrm{Exp}(\Lambda(a))$ distributed time units, or chose the earliest jump to $b_1, \dots, b_k$ in the support of κ_a with jump times drawn each from (independent) distributions $\mathrm{Exp}(\lambda(a \curvearrowright b_1)), \dots, \mathrm{Exp}(\lambda(a \curvearrowright b_1))$.

A process has π as equilibrium distribution if $\sum_{a \in S} \mathrm{P}(Z_t \in B \mid Z_s = a)\pi\{a\} = \pi(B)$, where $t > s > 0, B \subset S$. A stronger requirement relevant for sampling is ergodicity, which for finite state spaces takes the form $\lim_{t\to\infty} \mathrm{P}\left(Z_t = b \mid Z_s = a\right) = \pi\{b\}$ for all $b, a \in S$ so that in the long run, states from Z can be used to approximate samples from π.

Operators for Markov equivalence classes. Chickering (2002a) defines two sets of operators on $\mathcal{M}_n$. The operator $\mathrm{Insert}(\gamma, x, y, T)$ inserts the edge $x \to y$ to the CPDAG γ and directs previously undirected edges $t - y$ to $t \to y$ for $t \in T$, such that vertices $t \in T$ become “tails” of a v-structure $t \to y \leftarrow x$. Here x and y are not adjacent and T are (undirected) neighbours of y that are not adjacent to x. The resulting PDAG is then completed[7] to a CPDAG γ' if possible, otherwise the insertion is not defined (invalid).

The operator $\mathrm{Delete}(\gamma', x, y, H)$ deletes an edge $x - y$ or $x \to y$ of the CPDAG γ' and directs previously undirected edges $x - h$ as $x \to h$ and $y - h$ as $y \to h$ for h in H such that vertices $h \in H$ become “heads” of new v-structures $x \to h \leftarrow y$. The resulting PDAG is then completed to a CPDAG γ if possible, otherwise the deletion is not defined (invalid).

We call a move or jump from MEC γ to MEC γ' *local* if there is a DAG $G \in \gamma$, which can be transformed to a DAG $G' \in \gamma'$ by a single edge insertion or deletion. Local moves are preferable for two reasons: Firstly, if a weight function w, for example the exponentiated BIC score, factorises over the DAGs,

$$w(G, \mathrm{Data}) = \prod_{x \in V} w(\mathrm{Pa}_G(x), x, \mathrm{Data}),$$

then changes in w can be computed efficiently by comparing local scores or local weights, see Chickering (2002a), corollaries 7 and 9.

Secondly, Theorems 15 and 17 of (Chickering, 2002a) give precise criteria for the validity of local moves. Denote by $\mathrm{NA}_x(y)$ the (undirected) neighbours of y that are adjacent to x. In short, $\mathrm{Insert}(\gamma, x, y, T)$ is a valid local move, if and only if (i) $\mathrm{NA}_x(y)$ and the elements of T form a clique and (ii) any path from y to x without a directed edge pointing towards y (such a path is called semi-directed) contains a vertex in $\mathrm{NA}_x(y) \cup T$. $\mathrm{Delete}(\gamma, x, y, H)$ is a valid local move, if and only if $H \subset \mathrm{NA}_x(y)$ and $\mathrm{NA}_x(y) \setminus H$ form a clique.

4. Random walks on Markov equivalence classes

The key for the construction of a Markov process on Markov equivalence classes is that the valid local Insert and Delete operators are mutual inverses.

[7] The *completion* of a PDAG refers to the CPDAG representation of the MEC with the same skeleton and v-structures as the PDAG. There are cases, when this CPDAG does not exist, namely when there are no DAGs with this skeleton and v-structures. A simple example is PDAG C_4, the cycle on four vertices.

Lemma 1 (Chickering (2002b); Zhou and Chang (2023)) *If $\gamma' = \operatorname{Insert}(\gamma, x, y, T)$, $x, y \in V$, $T \subset V$, $\gamma \in \mathcal{M}_n$ is a valid local move, then there is a unique set of undirected neighbours H of y that are adjacent to x in γ' such that $\gamma = \operatorname{Delete}(\gamma', x, y, H)$.*

Conversely if $\gamma = \operatorname{Delete}(\gamma', x, y, H)$ is a valid local move, then there is a unique set of undirected neighbours T of y that are not adjacent to x in γ such that $\gamma' = \operatorname{Insert}(\gamma, x, y, T)$.

There may be two operators going from γ to γ', which is precisely the case if the inserted or deleted edge is undirected and $\operatorname{Insert}(\gamma, x, y, T)$ equals $\operatorname{Insert}(\gamma, y, x, T)$ (same for Delete). Phrased differently, the number of operators turning γ into γ' is identical to the operators for the reverse direction from γ' to γ (Zhou and Chang, 2023).

Lemma 2 (Chickering (2002b); Zhou and Chang (2023)) *The edge inserted by a local* $\operatorname{Insert}(\gamma, x, y, T)$ *is undirected exactly if T is empty and* $\operatorname{Pa}(x) = \operatorname{Pa}(y)$.

We write $\gamma' \in \operatorname{Insert}(\gamma)$ and $\gamma \in \operatorname{Delete}(\gamma')$ to indicate that γ' can be obtained from γ by a valid *local* Insert operation and that γ can be obtained from γ' by a valid *local* Delete operation. For example this lemma entails, when declaring $\gamma, \eta \in \mathcal{M}_n$ (undirected) neighbours if $\eta \in \operatorname{Insert}(\gamma) \cup \operatorname{Delete}(\gamma)$, general algorithms to sample from undirected graphs such as a simple continuous time random walk on $S = \mathcal{M}_n$ with jump intensity

$$\lambda(\gamma \curvearrowright \eta) = \begin{cases} 1 & \text{if } \eta \in \operatorname{Insert}(\gamma) \sqcup \operatorname{Delete}(\gamma) \\ 0 & \text{otherwise.} \end{cases}$$

This process has $\mathrm{U}(\mathcal{M}_n)$ as stationary distribution. While this jump intensity is remarkably simple, practical implementation requires the efficient enumeration of valid Insert and Delete operators for example to determine the total rate $\Lambda(\gamma) = |\operatorname{Insert}(\gamma) \sqcup \operatorname{Delete}(\Gamma)|$, a topic we come back to in section 7. Here using lemma 2 allows to account for multiple moves yielding the same CPDAG η.

Alternatively, one can also move towards η with twice the rate if there are two operators from γ to η, as long as one then also moves back from η to γ with twice the rate. This leads to an easier implementation and thus we proceed this way in our code. Also the Zanella process (Power and Goldman, 2019), a generalisation of the simple continuous time random walk that can be used to sample from the a distribution π defined on $\mathcal{M}_n$, is now available.

Let π be a probability distribution on $\mathcal{M}_n$. Let $g\colon [0, \infty) \to [0, \infty)$ be a balancing function such as $\sqrt{t}$, $\min(1, t)$ or $t/(1+t)$ with the property $g(t) = tg(1/t)$. The Zanella process $(Z_t)_{t \geq 0}$ on $\mathcal{M}_n$ is defined by the intensity

$$\lambda(\gamma \curvearrowright \eta) = \begin{cases} g\left(\dfrac{\pi\{\eta\}}{\pi\{\gamma\}}\right) & \text{if } \eta \in \operatorname{Insert}(\gamma) \sqcup \operatorname{Delete}(\gamma) \\ 0 & \text{otherwise} \end{cases},$$

where $\gamma \in \mathcal{M}_n$.

Theorem 3 *Let the target probability π be strictly positive for all $\gamma \in \mathcal{M}_n$. Then Z is irreducible, π is the unique stationary distribution and*

$$\lim_{t \to \infty} \mathrm{P}\left(Z_t = \gamma \mid Z_s = a\right) = \pi\{\gamma\} \quad \textit{for all } \gamma, \eta \in \mathcal{M}_n.$$

The proof of this theorem goes along similar lines as the proof of Theorem 4 below, so we omit it.

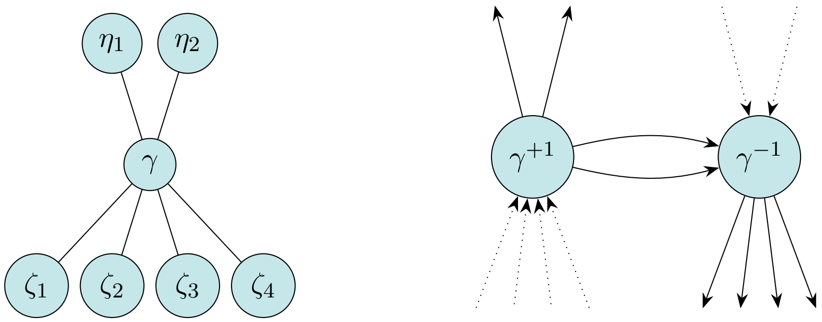

Figure 4: On the left, MEC γ with two neighbours η_1, η_2 in Insert(γ) and four neighbours $\zeta_1, \dots, \zeta_4$ in Delete(γ). The Zanella sampler for the uniform distribution on the space of MECs $\mathcal{M}_n$ will leave γ after an exponentially distributed time with total rate $\Lambda(\gamma) = 6$ towards one of the six neighbours drawn from $\kappa_\gamma = \mathrm{U}(\{\eta_1, \eta_2, \zeta_1, \zeta_2, \zeta_3, \zeta_4\})$. On the right, the situation is shown for the Zig-Zag sampler. To target a uniform distribution on $\mathcal{M}_n$, if $\gamma \in \mathcal{M}_n$ has 2 direct neighbours in Insert(γ) and 4 direct neighbours in Delete(γ), then move up from γ^{+1} with total rate 2, move from γ^{+1} to γ^{-1} with rate $2 = 4 - 2$ and down from γ^{-1} with total rate 4.

5. The Causal Zig-Zag sampler

We now define our sampler which can be thought of as Zanella process lifted by attaching a notion of direction. We baptise the non-reversible continuous-time sampler for Markov equivalence classes the "Causal Zig-Zag" motivated by the characteristic Zig-Zag pattern in the trace of the number of edges in the causal graph, see figure 1. Here, we exploit that Insert and Delete endow the space $\mathcal{M}_n$ with an intuitive interpretation of direction.

Let $S = \mathcal{M}_n \times \{-1, +1\}$. If $\gamma \in \mathcal{M}_n$, we denote the element $(\gamma, +1)$ by γ^{+1} and the element $(\gamma, -1)$ by γ^{-1} and write $\gamma^d = (\gamma, d)$ for $d \in \{-1, +1\}$. Again, choose a balancing function g and a target probability π on S and a Markov jump process Z as follows: For $\gamma \in \mathcal{M}_n$,

$$\lambda(\gamma^{+1} \curvearrowright \eta^{+1}) = \begin{cases} g\left(\dfrac{\pi\{\eta\}}{\pi\{\gamma\}}\right) & \text{if } \eta \in \mathrm{Insert}(\gamma) \\ 0 & \text{otherwise.} \end{cases}$$

$$\lambda(\eta^{-1} \curvearrowright \gamma^{-1}) = \begin{cases} g\left(\dfrac{\pi\{\gamma\}}{\pi\{\eta\}}\right) & \text{if } \gamma \in \mathrm{Delete}(\eta) \\ 0 & \text{otherwise.} \end{cases}$$

and for $\gamma \in \mathcal{M}_n$ and $d \in \{-1, +1\}$,

$$\lambda(\gamma^d \curvearrowright \gamma^{-d}) = \left(-\sum_\eta \lambda(\gamma^d \curvearrowright \eta^d) + \sum_\eta \lambda(\gamma^{-d} \curvearrowright \eta^{-d})\right)^+,$$

where $x^+ = \max(0, x)$ denotes the positive part. Note that λ can be computed if π is only known up to a multiplicative constant as typical for Bayesian applications. Figure 4 illustrates the neighboring states for the Zanella and Zig-Zag sampler.

Theorem 4 *Let the target probability $\pi\{\gamma\} > 0$ be strictly positive for all $\gamma \in \mathcal{M}_n$. Then Z is irreducible,*

$$\mathrm{P}(Z_t = b \mid Z_s = a) > 0$$

for all $a, b \in S$. The distribution $\tilde{\pi}$ on S with $\tilde{\pi}(\gamma^d) = \pi\{\gamma\}/2$ is the unique stationary distribution and

$$\lim_{t\to\infty} \mathrm{P}\left(Z_t = \gamma^d \mid Z_s = a\right) = \pi\{\gamma\}/2 \quad \textit{for all } a \in S,$$

where $\gamma \in \mathcal{M}_n, d \in \{+1, -1\}$.

Proof One first shows that any state γ^d communicates with $\mathbf{0}_n^{-1}$, where $\mathbf{0}$ denotes the empty graph. From this, the chain Z is irreducible (aperiodicity is not a concern for continuous time chains.) This part of the proof we give in the supplement (it bears some similarity to the consistency argument for the greedy equivalence search algorithm.)

It remains to show that $\tilde{\pi}$ is the stationary distribution of Z. This follows by applying proposition 9 in the supplement which gives general criteria for stationarity. We proceed by checking the three conditions of the proposition (equations (1), (2) and (3)).

Firstly, $\mathfrak{s}\colon S \to S$, $\mathfrak{s}(\gamma^d) = \gamma^{-d}$ is a bijection on S that is easily seen to be $\tilde{\pi}$-isometric (equation (1)).
Also, skew balance (equation (2)) holds: if $\eta \in \mathrm{Insert}(\gamma)$

$$\tilde{\pi}\{\gamma^{+1}\}\lambda(\gamma^{+1} \curvearrowright \eta^{+1}) = \frac{\pi\{\gamma\}}{2} g\left(\frac{\pi\{\eta\}}{\pi\{\gamma\}}\right)$$

$$= \frac{\pi\{\eta\}}{2} g\left(\frac{\pi\{\gamma\}}{\pi\{\eta\}}\right) = \tilde{\pi}\{\eta^{-1}\}\lambda(\eta^{-1} \curvearrowright \gamma^{-1})$$

using the balancing property of g. Else, if $\eta \notin \mathrm{Insert}(\gamma)$, also $\gamma \notin \mathrm{Delete}(\eta)$ and $\tilde{\pi}\{\gamma^{+1}\}\lambda(\gamma^{+1} \curvearrowright \eta^{+1}) = \tilde{\pi}\{\eta^{-1}\}\lambda(\eta^{-1} \curvearrowright \gamma^{-1}) = 0$.
Finally, we obtain the semi-local condition (equation (3)), $\Lambda(\gamma^{+1}) = \sum_{b\in S} \lambda(\gamma^{+1} \curvearrowright b) = \sum_{\eta\in\mathrm{Insert}(\gamma)} \lambda(\gamma^{+1} \curvearrowright \eta^{+1}) + \lambda(\gamma^{+1} \curvearrowright \gamma^{-1})$
$= \sum_{\eta\in\mathrm{Delete}(\gamma)} \lambda(\gamma^{-1} \curvearrowright \eta^{-1}) = \sum_{a\in S} \lambda(\gamma^{-1} \curvearrowright a)$
$= \Lambda(\mathfrak{s}(\gamma^{+1}))$. Thus the theorem is proved. ■

6. GES as limit of our sampler

It is interesting to note that when starting in the empty graph with the balancing function $g(x) = \sqrt{x}$ and target $\pi\{\gamma\} = \exp(\beta s(\gamma))$, where $\beta > 0$ is a coldness parameter and s is a Markov equivalent score, we recover the greedy equivalence search algorithm (GES) in the limit $\beta \to \infty$. In this limit, the Insert operator that improves the score the most is selected immediately with probability approaching 1 as long as there is such an edge addition which improves the score at all. This is because for $\eta \in \mathrm{Insert}(\gamma)$,

$$\kappa_{\gamma^{+1}}\{\eta^{+1}\} = \frac{\exp(\frac{1}{2}\beta(s(\eta) - s(\gamma))}{\sum\limits_{\zeta\in\mathrm{Insert}(\gamma)} \exp(\frac{1}{2}\beta(s(\zeta) - s(\gamma)))}$$

is a soft-max over the score improvements and the intensity $\Lambda(\gamma)$ approaches infinity. If no edge addition can improve the score anymore, the direction changes immediately if there is an edge removal that increases the score. In following second phase, again with probability approaching one, the Delete operator that improves the score the most is immediately selected with probability approaching 1 by same argument. This way the process reaches with probability approaching 1 in time approaching 0 the highest scoring model along the same trajectory as the GES with the same computational effort as a GES (when implemented with the same algorithmic improvements given below). This proves the following statement:

Theorem 5 *If started in the empty graph, with balancing function $g(x) = \sqrt{x}$, for all $t > 0$,*

$$\lim_{\beta \to \infty} \mathrm{P}(Z_t \in \{\gamma_\star^{+1}, \gamma_\star^{-1}\}) = 1,$$

where $\gamma_\star$ is the CPDAG found by a two-pass greedy equivalence search starting in the empty graph.

Moreover, for large β, with high probability Z visits the same models as the two-phase GES, with the same computational effort.

We refer to the thorough discussion in section 4 of Chickering (2002b). In particular, we conclude with the remark in section 4.3 that starting in the empty graph is an efficient way to converge towards the concentration of posterior mass in the large sample limit. Behaviour of piecewise deterministic processes under similar annealing schemes has been previously studied in Monmarché (2016).

7. Efficient algorithms for the underlying graph operations

Before stating our algorithmic results, it is necessary to revisit a basic problem in this area: computing a DAG in the MEC represented by a given CPDAG. It is well-known that this task can be solved in linear-time $O(n+m)$ for CPDAGs with n vertices and m edges relying on algorithms from the chordal graph literature (Chickering, 2002a). The key observation is that the directed edges of the CPDAG can be ignored and any acyclic and v-structure-free orientation of the undirected edges, will yield a DAG from the MEC. This task can be performed using, e.g., by the graph traversal algorithm *Maximum Cardinality Search*, MCS for short (Tarjan and Yannakakis, 1984), which, at each step, visits a vertex with the highest number of already visited neighbours. Appendix A.2 in (Chickering, 2002a) gives a good overview over this approach. More generally, the term *consistent extension* is used to describe a DAG with the same adjacencies and v-structures as a given (C)PDAG.

The computational task of applying one of the GES operators is fundamental, not only in the context of this work, but naturally also for GES itself and other score-based algorithms. Classically, the following approach is used, as described by Chickering (2002a): First, the operator is applied locally by inserting/deleting the edge and orienting edges incident to T, respectively H, yielding a PDAG. Second, for this PDAG, a consistent extension is computed. Third, the new CPDAG is directly computed from the consistent extension.

The first and third step can be performed in linear-time, however, the second step, when performed naively, needs time $O(n^3)$ (Dor and Tarsi, 1992; Wienöbst et al., 2021). We provide a linear-time algorithm for this problem by modifying the first and second step, building on ideas from Chickering (2002b) and Hauser and Bühlmann (2012):

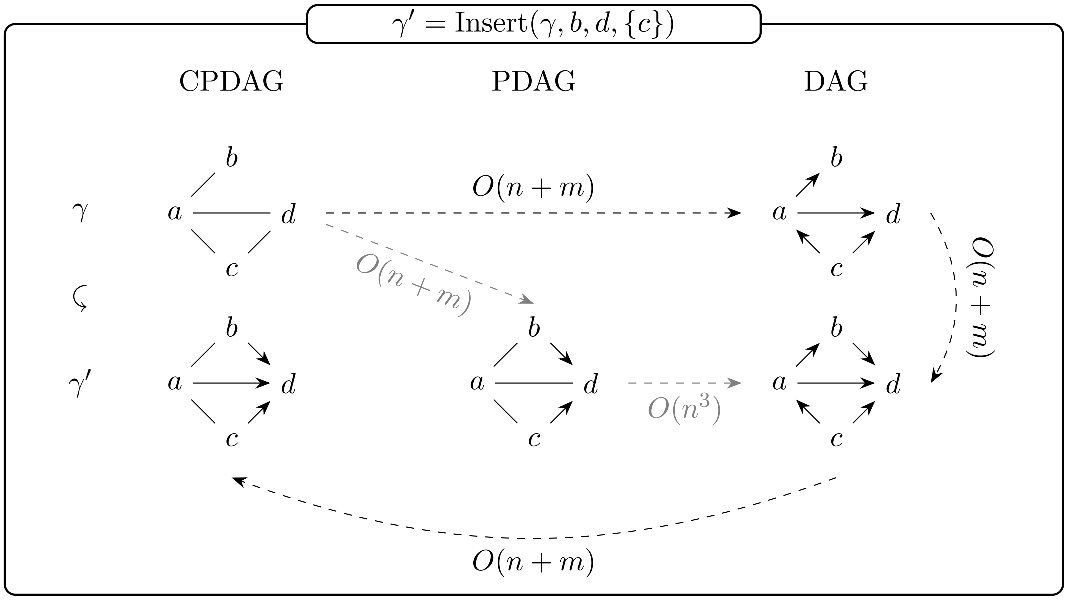

Figure 5: A schematic overview of the linear-time approach for applying a GES operator. Previous approaches add the inserted edge to the initial CPDAG, obtaining a PDAG associated with the new MEC γ'. However, going from this PDAG to the CPDAG, usually via a consistent DAG extension as intermediate step, necessitates time $O(n^3)$. In contrast, our approach finds a consistent DAG extension of the *initial* CPDAG in time $O(n+m)$, which has the property that applying the operator *directly* yields a DAG from γ'. Transforming this DAG into its CPDAG can be done in $O(n+m)$, as shown by Chickering (1995).

Theorem 6 *Let γ be a CPDAG. Applying a GES operator* $\text{Insert}(\gamma, x, y, T)$ *or* $\text{Delete}(\gamma, x, y, H)$ *to γ and obtaining γ' is possible in time $O(n+m)$.*

Proof By Theorem 15 and 17 in (Chickering, 2002b), any GES operator corresponds to a single edge insertion/deletion in a certain DAG in the MEC of γ. Our approach is as follows. First, compute a consistent extension of γ, which has the property that a single insertion/deletion yields a DAG from the new MEC represented by γ' in linear-time. Exploiting that γ is a CPDAG allows us to find this consistent extension G in linear-time using a modified MCS (described below). Then, the insertion/deletion can be performed in constant time to yield DAG G'. Afterwards, the "standard" third step of finding CPDAG γ' for DAG G' is applied (Chickering, 1995).

To perform the first step, we distinguish between the Insert and Delete operator. In case of the $\text{Insert}(\gamma, x, y, T)$, we perform an MCS which starts with visiting the vertices in T and $\text{NA}_x(y)$. As they form a clique, it is easy to see that this does not violate the properties of an MCS (the visit order is one which could be produced by a "standard" MCS). As discussed in the proof of Theorem 15 in (Chickering, 2002b) and Proposition 43 in (Hauser and Bühlmann, 2012) , this yields a DAG G with the desired property that inserting $x \to y$ gives $G' \in \gamma'$. For the $\text{Delete}(\gamma, x, y, H)$ operator, we proceed the same way only that vertices in $\text{NA}_x(y) \setminus H$ are visited first (afterwards x and y in this order). By the proof of Theorem 17 in (Chickering, 2002b), this gives a DAG $G' \in \gamma'$. ■

This time-complexity is asymptotically optimal, as there are graphs, for which $O(m)$ edges change after applying an operator.

In the framework described above, to obtain a *uniform* MCMC sampler of CPDAGs, it suffices to count the number of operators and to sample an operator with uniform probability. We derive the first polynomial-time algorithm for this task.[8]

Theorem 7 *Let γ be a CPDAG. The number of locally valid Insert and Delete operators can be computed in time $O(n^2 \cdot m)$. Sampling an operator uniformly is possible in the same time complexity.*

Sampling an operator in polynomial-time in this manner is only possible in the uniform case. When operators are weighted by their score, a different procedure is necessary.

There are multiple possible approaches to sample an operator proportional to an underlying local score, which may update after a move. In this work, we rely on the fact that, per move, usually only a few operator scores change. Hence, we use (i) caching of local scores to only recompute scores, which actually change. This is, as in the GES algorithm, crucial as the score computation can be the bottleneck of the algorithm (depending on sample size and the particular scoring procedure). Then, we (ii) efficiently list all operators one-by-one (without generating invalid operators), enabled by the insights from the previous section.

Corollary 8 *Let γ be a CPDAG with maximum number of neighbors d. The operators can be listed in time $O(n^2 \cdot m + |\mathrm{op}(\gamma)| \cdot d)$.*

Using this result and caching, the overall cost per move is in $O(n^2 \cdot m + |\mathrm{op}(\gamma)| \cdot d + |\mathrm{changed}(\gamma)| \cdot \mathrm{scoreeval})$, where scoreeval describes the time of a score evaluation. In our empirical studies, we find that the number of operators per pair of vertices is often constant (when the undirected edge degree is constant) and that the number of changed operators is usually very small, making the algorithmic improvements impactful.

8. Conclusions

We provide a novel continuous-time momentum-based MCMC sampler over the space of MECs based on the GES operators (Chickering, 2002b) and extended by a notion of direction. We show empirically that it can achieve favourable mixing time compared to earlier MCMC approaches and apply an efficient implementation of this sampler to the problem of observational causal discovery. In particular, our algorithmic improvements regarding the application of the GES operators, yielding linear-time for applying an operator and polynomial-time for counting the number of operators, go beyond this specific use case.

[8] The proof is provided in Appendix B in the supplement.

References

R. Agrawal, C. Uhler, and T. Broderick. Minimal I-MAP MCMC for scalable structure discovery in causal DAG models. In *International Conference on Machine Learning*, pages 89–98. PMLR, 2018.

S. A. Andersson, D. Madigan, and M. D. Perlman. A characterization of Markov equivalence classes for acyclic digraphs. *The Annals of Statistics*, 25(2):505–541, 1997.

Y. Annadani, J. Rothfuss, A. Lacoste, N. Scherrer, A. Goyal, Y. Bengio, and S. Bauer. Variational causal networks: Approximate Bayesian inference over causal structures. *arXiv preprint arXiv:2106.07635*, 2021.

Y. Annadani, N. Pawlowski, J. Jennings, S. Bauer, C. Zhang, and W. Gong. BayesDAG: Gradient-based posterior sampling for causal discovery. In *ICML 2023 Workshop on Structured Probabilistic Inference & Generative Modeling*, 2023.

D. M. Chickering. A transformational characterization of equivalent Bayesian network structures. In *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, pages 87–98, 1995.

D. M. Chickering. Learning equivalence classes of Bayesian-network structures. *Journal of Machine Learning Research*, 2:445–498, 2002a.

D. M. Chickering. Optimal structure identification with greedy search. *Journal of Machine Learning Research*, 3:507–554, 2002b.

C. Cundy, A. Grover, and S. Ermon. Bcd nets: Scalable variational approaches for Bayesian causal discovery. *Advances in Neural Information Processing Systems*, 34:7095–7110, 2021.

T. Deleu, A. Góis, C. Emezue, M. Rankawat, S. Lacoste-Julien, S. Bauer, and Y. Bengio. Bayesian structure learning with generative flow networks. In *Uncertainty in Artificial Intelligence*, pages 518–528. PMLR, 2022.

G. A. Dirac. On rigid circuit graphs. In *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, volume 25, pages 71–76. Springer, 1961.

D. Dor and M. Tarsi. A simple algorithm to construct a consistent extension of a partially oriented graph. *Technicial Report R-185, Cognitive Systems Laboratory, UCLA*, page 45, 1992.

P. Fearnhead, J. Bierkens, M. Pollock, and G. O. Roberts. Piecewise deterministic Markov processes for continuous-time monte carlo. *Statistical Science*, 33(3):386–412, 2018.

N. Friedman and D. Koller. Being Bayesian about network structure. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, pages 201–210, 2000.

S. B. Gillispie and M. D. Perlman. The size distribution for Markov equivalence classes of acyclic digraph models. *Artificial Intelligence*, 141(1-2):137–155, 2002.

P. Giudici and R. Castelo. Improving Markov chain monte carlo model search for data mining. *Machine learning*, 50:127–158, 2003.

M. Grzegorczyk and D. Husmeier. Improving the structure MCMC sampler for Bayesian networks by introducing a new edge reversal move. *Machine Learning*, 71(2-3):265–305, 2008.

P. Gustafson. *Statistics and Computing*, 8(4):357–364, 1998. doi: 10.1023/a:1008880707168. URL https://doi.org/10.1023/a:1008880707168.

A. Hauser and P. Bühlmann. Characterization and greedy learning of interventional Markov equivalence classes of directed acyclic graphs. *The Journal of Machine Learning Research*, 13(1):2409–2464, 2012.

Y. He, J. Jia, and B. Yu. Reversible MCMC on Markov equivalence classes of sparse directed acyclic graphs. *The Annals of Statistics*, 41(4):1742 – 1779, 2013. doi: 10.1214/13-AOS1125. URL https://doi.org/10.1214/13-AOS1125.

D. Heckerman, D. Geiger, and D. M. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine learning*, 20:197–243, 1995.

O. Kallenberg. *Foundations of modern probability*. Probability and its Applications (New York). Springer-Verlag, New York, second edition, 2002. ISBN 0-387-95313-2. doi: 10.1007/978-1-4757-4015-8.

M. Koivisto and K. Sood. Exact Bayesian structure discovery in Bayesian networks. *The Journal of Machine Learning Research*, 5:549–573, 2004.

D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

J. Kuipers and G. Moffa. Partition MCMC for inference on acyclic digraphs. *Journal of the American Statistical Association*, 112(517):282–299, 2017.

S. L. Lauritzen. *Graphical models*, volume 17. Clarendon Press, 1996.

L. Lorch, J. Rothfuss, B. Schölkopf, and A. Krause. Dibs: Differentiable Bayesian structure learning. *Advances in Neural Information Processing Systems*, 34:24111–24123, 2021.

D. Madigan, J. York, and D. Allard. Bayesian graphical models for discrete data. *International Statistical Review/Revue Internationale de Statistique*, pages 215–232, 1995.

D. Madigan, S. A. Andersson, M. D. Perlman, and C. T. Volinsky. Bayesian model averaging and model selection for Markov equivalence classes of acyclic digraphs. *Communications in Statistics–Theory and Methods*, 25(11):2493–2519, 1996.

P. Monmarché. Piecewise deterministic simulated annealing. *Latin American Journal of Probability and Mathematical Statistics*, 13(1):357, 2016. doi: 10.30757/alea.v13-15. URL https://doi.org/10.30757/alea.v13-15.

R. M. Neal. Monte carlo implementation. In *Bayesian Learning for Neural Networks*, pages 55–98. Springer New York, 1996. doi: 10.1007/978-1-4612-0745-0_3. URL https://doi.org/10.1007/978-1-4612-0745-0_3.

T. Niinimäki, P. Parviainen, and M. Koivisto. Structure discovery in Bayesian networks by sampling partial orders. *The Journal of Machine Learning Research*, 17(1):2002–2048, 2016.

J. Pearl. *Causality*. Cambridge university press, 2009.

J. M. Pena. Approximate counting of graphical models via MCMC. In *Artificial Intelligence and Statistics*, pages 355–362. PMLR, 2007.

Y. Peres and P. Sousi. Mixing times are hitting times of large sets. *Journal of Theoretical Probability*, 28(2):488–519, May 2013. doi: 10.1007/s10959-013-0497-9. URL https://doi.org/10.1007/s10959-013-0497-9.

J. Peters, D. Janzing, and B. Schölkopf. *Elements of causal inference: foundations and learning algorithms*. The MIT Press, 2017.

S. Power and J. V. Goldman. Accelerated sampling on discrete spaces with non-reversible markov processes, 2019. URL https://arxiv.org/abs/1912.04681.

G. O. Roberts and J. S. Rosenthal. General state space Markov chains and MCMC algorithms. *Probability Surveys*, 1(none), Jan. 2004. doi: 10.1214/154957804100000024. URL https://doi.org/10.1214/154957804100000024.

X. Shen, S. Ma, P. Vemuri, G. Simon, and the Alzheimer's Disease Neuroimaging Initiative. Challenges and opportunities with causal discovery algorithms: Application to alzheimer's pathophysiology. *Scientific Reports*, 10(1), Feb. 2020. doi: 10.1038/s41598-020-59669-x. URL https://doi.org/10.1038/s41598-020-59669-x.

P. Spirtes, C. N. Glymour, and R. Scheines. *Causation, prediction, and search*. MIT press, 2000.

R. E. Tarjan and M. Yannakakis. Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs. *SIAM Journal on computing*, 13(3):566–579, 1984.

T. Verma and J. Pearl. Equivalence and synthesis of causal models. In *Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence, UAI'90*, pages 255–270, 1990.

M. Wienöbst, M. Bannach, and M. Liśkiewicz. Extendability of causal graphical models: Algorithms and computational complexity. In *Uncertainty in Artificial Intelligence*, pages 1248–1257. PMLR, 2021.

G. Zanella. Informed proposals for local MCMC in discrete spaces. *Journal of the American Statistical Association*, 115(530):852–865, Apr. 2019. doi: 10.1080/01621459.2019.1585255. URL https://doi.org/10.1080/01621459.2019.1585255.

Q. Zhou and H. Chang. Complexity analysis of Bayesian learning of high-dimensional DAG models and their equivalence classes. *The Annals of Statistics*, 51(3):1058–1085, 2023.

Acknowledgements

Data used in preparation of this article were obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (adni.loni.usc.edu).

As such, the investigators within the ADNI contributed to the design and implementation of ADNI and/or provided data but did not participate in the writing of this article. A complete listing of ADNI investigators can be found at: http://adni.loni.usc.edu/wp-content/uploads/how_to_apply/ADNI_Acknowledgement_List.pdf.

ADNI data is de-identified and publicly available for download. All study participants provided written informed consent, and study protocols were approved by each local site's institutional review board.

Data collection and sharing for the ADNI project was funded by the Alzheimer's Disease Neuroimaging Initiative (ADNI) (National Institutes of Health Grant U01 AG024904) and DOD ADNI (Department of Defense award number W81XWH-12-2-0012). ADNI is funded by the National Institute on Aging, the National Institute of Biomedical Imaging and Bioengineering, and through generous contributions from the following: AbbVie, Alzheimer's Association; Alzheimer's Drug Discovery Foundation; Araclon Biotech; BioClinica, Inc.; Biogen; Bristol-Myers Squibb Company; CereSpir, Inc.; Cogstate; Eisai Inc.; Elan Pharmaceuticals, Inc.; Eli Lilly and Company; EuroImmun; F. Hoffmann-La Roche Ltd and its affiliated company Genentech, Inc.; Fujirebio; GE Healthcare; IXICO Ltd.; Janssen Alzheimer Immunotherapy Research & Development, LLC.; Johnson & Johnson Pharmaceutical Research & Development LLC.; Lumosity; Lundbeck; Merck & Co., Inc.; Meso Scale Diagnostics, LLC.; NeuroRx Research; Neurotrack Technologies; Novartis Pharmaceuticals Corporation; Pfizer Inc.; Piramal Imaging; Servier; Takeda Pharmaceutical Company; and Transition Therapeutics. The Canadian Institutes of Health Research is providing funds to support ADNI clinical sites in Canada. Private sector contributions are facilitated by the Foundation for the National Institutes of Health (www.fnih.org). The grantee organisation is the Northern California Institute for Research and Education, and the study is coordinated by the Alzheimer's Therapeutic Research Institute at the University of Southern California. ADNI data are disseminated by the Laboratory for Neuro Imaging at the University of Southern California.

Appendix A. Skew-balanced jump processes

Proposition 9 *If there is an bijection $\mathfrak{s}$ on S that is π-isometric:*

$$\pi\{a\} = \pi\{\mathfrak{s}(a)\}, \quad a \in S, \tag{1}$$

such that skew detailed balance

$$\pi\{a\}\lambda(a \curvearrowright b) = \pi\{\mathfrak{s}(b)\}\lambda(\mathfrak{s}(b) \curvearrowright \mathfrak{s}(a)) \quad a, b \in S \tag{2}$$

holds and such that the semi-local *condition*

$$\Lambda(a) = \Lambda(\mathfrak{s}(a)) \tag{3}$$

holds, then Z is π-stationary.

(3) typically requires that $\mathfrak{s}^n$ for some order $n = 1, 2, \ldots$ is the identity map. If $\mathfrak{s}$ is the identity ($n = 1$), then (3) and (1) hold automatically and (2) reduces to a detailed balance condition.

Also the case $n = 2$ is important. A map $\mathfrak{s}\colon S \to S$ is an *involution* if $\mathfrak{s} \circ \mathfrak{s}$ is the identity. For example, if $S = \mathcal{X} \times \{-1, 1\}$, then $\mathfrak{s}$ with $\mathfrak{s}((x, d)) = \mathfrak{s}((x, -d))$ for $(x, d) \in S$ is an involution. An involution is automatically an bijection. Importantly, (2) is trivial for $b = \mathfrak{s}(a)$, but turns into a linear constraint if designing samplers using $\mathfrak{s}$ with higher orders n.

Appendix B. Remaining proofs

A convenient criterium for stationary is as follows: If Z is stationary for π, then for bounded $f\colon S \to \mathbb{R}$

$$\sum_a \sum_b \lambda(a \curvearrowright b)(f(b) - f(a))\pi\{a\} = 0. \tag{4}$$

Conversely, if the preceding equation holds for all $f\colon S \to \mathbb{R}$ bounded, then Z is stationary.
]

Proof [Proof of proposition 9]

The proposition follows from (2) by $\sum_a \sum_b \lambda(a \curvearrowright b) f(b)\pi\{a\} = \sum_a \sum_b \lambda(\mathfrak{s}(b) \curvearrowright \mathfrak{s}(a)) f(b)\pi\{\mathfrak{s}(b)\}$ and with $z = \mathfrak{s}(a)$,

$$= \sum_b \sum_z \lambda(\mathfrak{s}(b) \curvearrowright z) f(b)\pi\{\mathfrak{s}(b)\}$$

and by the definition of the specific rate and its connection to the total

$$= \sum_b \Lambda(\mathfrak{s}(b)) f(b)\pi\{\mathfrak{s}(b)\} = \sum_a \sum_b \lambda(b \curvearrowright a) f(b)\pi\{b\}$$

$$= \sum_a \sum_b \lambda(a \curvearrowright b) f(a)\pi\{a\}.$$

In the last step we use that $\Lambda(\mathfrak{s}(b)) = \Lambda(b) = \sum_a \lambda(b \curvearrowright a)$ and $\pi\{\mathfrak{s}(b)\} = \pi\{b\}$. So we have established (4) for any summable f. ■

Proof [Supplement to the proof of Theorem 4.] Let $\gamma^d \in S$, where γ is not the graph with no edges $\mathbf{0}_n$ (assume that $n > 1$ so there is something to show.) We now prove $\mathrm{P}(Z_t = \mathbf{0}_n^{-1} \mid Z_s = \gamma^d) > 0, \gamma^d \in S$, $t > s$. We first find a state η^{-1} such that $\mathrm{P}(Z_{(t-s)/2} = \eta^{-1} \mid Z_s = \gamma^d) > 0, \gamma^d \in S$. If $d = -1$, one can take $\eta = \gamma$. Otherwise, if $d = +1$, though $\mathrm{Delete}(\gamma)$ is non-empty[9], it can still be that $\lambda(\gamma^{+1} \curvearrowright \gamma^{-1}) = 0$. But in that case, by construction, $\mathrm{Insert}(\gamma)$ is non-empty and $\lambda(\gamma^{+1} \curvearrowright \zeta^{+1}) > 0$ for some $\zeta \in \mathcal{M}_n$. Repeating that argument, at most $n(n-1)/2$ many jumps lead to a state $\eta^{-1} \in S$ and together, these jumps have positive probability to occur in a time interval of length $(t-s)/2$, so η^{-1} is that state we are looking for.

Now from η there is a sequence of at most $n(n-1)/2$ edge removal moves that reach $\mathbf{0}_n$. Together these jumps have again positive probability to occur in a time interval of length $(t-s)/2$. Therefore $\mathrm{P}(Z_t = \mathbf{0}_n^{-1} \mid Z_s = \gamma^d) \geq \mathrm{P}(Z_t = \mathbf{0}_n^{-1} \mid Z_{(t-s)/2} = \eta^{-1})\mathrm{P}(Z_{(t-s)/2} = \eta^{-1} \mid Z_s = \gamma^d) > 0$.

By repeating this argument, $\mathrm{P}(Z_t = \mathbf{0}_n^{-1} \mid Z_{(t-s)/2} = \gamma^{-d}) > 0$. Using skew balance to reverse the path from γ^{-d} to $\mathbf{0}_n^{-1}$ into a path from $\mathbf{0}_n^{+1}$ to γ^d, replacing edge inserts by edge deletions and vice versa, $\mathrm{P}(Z_t = \gamma^d \mid Z_{(t-s)/2} = \mathbf{0}_n^{+1}) > 0$.

Also $\mathrm{P}(Z_{(t-s)/2} = \mathbf{0}_n^{+1} \mid Z_s = \mathbf{0}_n^{-1}) > 0$ as $\lambda(\mathbf{0}_n^{-1} \curvearrowright \mathbf{0}_n^{+1}) > 0$ because there is no delete operator available, but one can insert an undirected edge to $\mathbf{0}_n$. We therefore have $\mathrm{P}(Z_t = \gamma^d \mid Z_s = \mathbf{0}_n^{-1}) \geq \mathrm{P}(Z_t = \gamma^d \mid Z_{(t-s)/2} = \mathbf{0}_n^{+1})\mathrm{P}(Z_{(t-s)/2} = \mathbf{0}_n^{+1} \mid Z_s = \mathbf{0}_n^{-1}) > 0$.

This is sufficient because S is finite. ■

Proof [Proof of Theorem 7] The task immediately reduces to counting the number of operators for each pair of vertices. We consider the $\mathrm{Delete}(\gamma, x, y, H)$ operator first. Here, the set of operators correspond to the subsets of $\mathrm{NA}_x(y)$, which form a clique. This further reduces to the problem of counting (and sampling) the number of cliques of a chordal graph, that is a graph without induced cycles of length ≥ 4, (Dirac, 1961) due to the fact that there can only be undirected edges between vertices in $\mathrm{NA}_x(y)$ (Lemma 3 (Chickering, 1995)) and that these undirected edges form a chordal graph in a CPDAG (Andersson et al., 1997). It is a basic fact that the number of cliques of a chordal graph γ is given by:

$$\prod_{u \in V} 2^{\mathrm{Pa}_D(u)} + 1,$$

where G is any consistent extension of γ due to the fact that all parents of u form a clique (else D would not be a consistent extension as it has additional v-structures). Each term in the product gives the number of cliques containing u as highest ordered vertex w.r.t. some fixed topological ordering of D. Evaluating this is clearly possible in $O(m)$ per pair x, y.

For the $\mathrm{Insert}(\gamma, x, y, T)$ operator, the set of operators is formed by subsets T of the undirected neighbours of y, which are nonadjacent with x, such that $\mathrm{NA}_x(y) \cup T$ is a clique and $\mathrm{NA}_x(y) \cup T$ blocks all paths from y to x without edge pointing towards y. The latter condition complicates the matter. It can be resolved as follows: Consider, for each neighbor of y, the set of vertices reachable via a path without edges pointing towards y (that is reachable via a semi-directed path $y - x \ldots$) not containing an undirected neighbour of y.

[9] γ has edges, so there is a DAG $G \in \gamma$ from which an edge can be removed to obtain some $G' \in \mathrm{Delete}(\gamma)$

This can be done independently of x taking overall time (for all y) $O(n^2m)$. If, under these constraints, x is reachable from a neighbour w of y, which is non-adjacent to x, then w has to be in T (else there is an open semi-directed path from y to x). After taking all such vertices w, none of the remaining vertices has an open semi-directed path to x. We show this by contradiction. Assume there exists z such that there is a semi-directed path from z to x not blocked by $\mathrm{NA}_x(y) \cup T$. There has to be a vertex a on this path, which is a neighbour of y else z would be in T, consider the one closest to x. Then, this vertex has an unblocked semi-directed path to x and hence is in T. This is a contradiction to the fact that the path is open given $\mathrm{NA}_x(y) \cup T$.

Hence, we can compute the set of vertices, which *must* be in T in overall time $O(n^2m)$, respectively $O(m)$ per pair x, y. Consequently, they need to form a clique with $\mathrm{NA}_x(y)$ (this can be checked in $O(m)$ as well). The remaining neighbours of y (non-adjacent with x), which are fully connected to $\mathrm{NA}_x(y)$ and the must-take vertices, may be part of T as long as they themselves form a clique. Hence, we arrive at the problem of counting the number of cliques in a chordal graph studied above, which can be solved in time $O(m)$.

It is easy to see that sampling can be performed in time $O(n^2m)$ (when performing counting as preprocessing) by first sampling a pair of vertices x, y with probability proportional to the number of locally valid operators and second sampling an operator for this set with uniform probability (which amounts to sampling a clique in a chordal graph). ■

Appendix C. Further models with high posterior probability for the ADNI database

In the main document, we only showed the DAG with highest posterior probability of 0.701 for the data from the ADNI database. In figure 6, we show the DAGs with the second- and third-highest posterior probabilities, which are 0.207 and 0.0049.

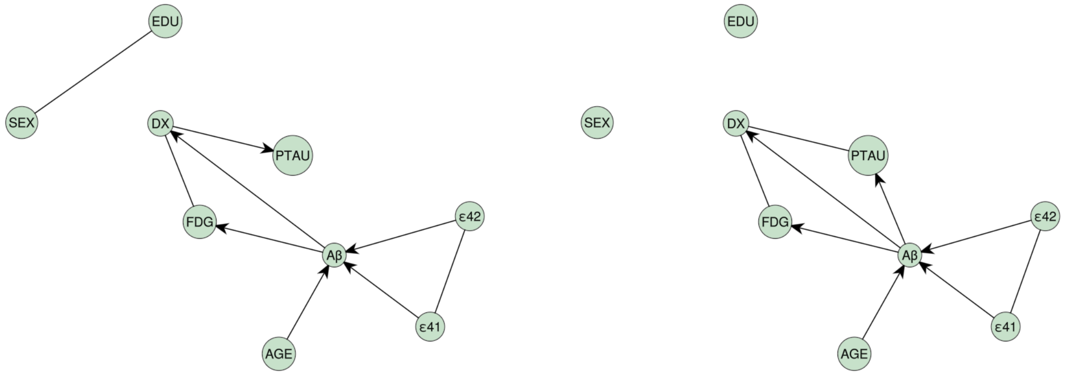

Figure 6: The two models with second- and third-highest posterior probabilities, namely 0.207 and 0.0049. This illustrates one particular use case of our sampler, namely uncertainty quantification in the situation where GES applies, such as uncertainty about the edge SEX to EDU.