Behaviormetrika (2018) 45:389–422
https://doi.org/10.1007/s41237-018-0051-2

INVITED PAPER

Reasoning with alternative acyclic directed mixed graphs

Jose M. Peña[1]

Received: 17 November 2017 / Accepted: 11 May 2018 / Published online: 22 May 2018

Abstract Acyclic directed mixed graphs (ADMGs) are the graphs used by Pearl (Causality: models, reasoning, and inference. Cambridge University Press, Cambridge, 2009) for causal effect identification. Recently, alternative acyclic directed mixed graphs (aADMGs) have been proposed by Peña (Proceedings of the 32nd conference on uncertainty in artificial intelligence, 577–586, 2016) for causal effect identification in domains with additive noise. Since the ADMG and the aADMG of the domain at hand may encode different model assumptions, it may be that the causal effect of interest is identifiable in one but not in the other. Causal effect identification in ADMGs is well understood. In this paper, we introduce a sound algorithm for identifying arbitrary causal effects from aADMGs. We show that the algorithm follows from a calculus similar to Pearl's do-calculus. Then, we turn our attention to Andersson–Madigan–Perlman chain graphs, which are a subclass of aADMGs, and propose a factorization for the positive discrete probability distributions that are Markovian with respect to these chain graphs. We also develop an algorithm to perform maximum likelihood estimation of the factors in the factorization.

Keywords Causality · Causal effect identification · Acyclic directed mixed graphs · Factorization · Maximum likelihood estimation

Communicated by Antti Hyttinen.

✉ Jose M. Peña
jose.m.pena@liu.se

[1] IDA, Linköping University, Linköping, Sweden

1 Introduction

Undirected graphs (UGs), bidirected graphs (BGs), and directed and acyclic graphs (DAGs) have extensively been studied as representations of independence models. DAGs have also been studied as representation of causal models, because they can model asymmetric relationships between random variables. DAGs and UGs (respectively BGs) have been extended into chain graphs (CGs), which are graphs with directed and undirected (respectively bidirected) edges but without semidirected cycles. Therefore, CGs can model both symmetric and asymmetric relationships between random variables. CGs with directed and undirected edges may represent different independence models depending on whether the Lauritzen–Wermuth–Frydenberg (LWF) or the Andersson–Madigan–Perlman (AMP) interpretation is considered (Lauritzen 1996; Andersson et al 2001). CGs with directed and bidirected edges have a unique interpretation, the so-called multivariate regression (MVR) interpretation (Cox and Wermuth 1996). MVR CGs have been extended by (i) relaxing the semidirected acyclity constraint so that only directed cycles are forbidden, and (ii) allowing up to two edges between any pair of nodes. The resulting models are called acyclic directed mixed graphs (ADMGs) (Richardson 2003). AMP CGs have also been extended similarly (Peña 2016). The resulting models are called alternative acyclic directed mixed graphs (aADMGs). It is worth mentioning that neither the original ADMGs nor any other family of mixed graphical models that we know of (e.g., summary graphs (Cox and Wermuth 1996), ancestral graphs (Richardson and Spirtes 2002), MC graphs (Koster 2002) or loopless mixed graphs (Sadeghi and Lauritzen 2014) subsume AMP CGs and hence aADMGs. To see it, we refer the reader to the works by (Richardson and Spirtes 2002, p. 1025) and (Sadeghi and Lauritzen 2014, Section 4.1).

In addition to represent independence models, some of the graphical models mentioned above have been used for causal effect identification, i.e., to determine if the causal effect of an intervention is identifiable from observational quantities. For instance, Pearl's approach to causal effect identification makes use of ADMGs to represent causal models over the observed variables (Pearl 2009). The directed edges represent causal relationships, whereas the bidirected edges represent confounding, i.e., a latent common cause. A key feature of Pearl's approach is that no assumption is made about the functional form of the causal relationships. That is, each variable A is an unconstrained function of its observed causes $Pa(A)$ and its unobserved causes U_A, i.e., $A = g(Pa(A), U_A)$. Without loss of generality, we can consider U_A as being unidimensional (Mooij et al 2016), Proposition 4). We do so. This U_A is sometimes called noise or error or residual. In this paper, we study causal effect identification under the assumption that $A = g(Pa(A)) + U_A$, also called additive noise model. This is a rather common assumption in causal discovery (Bühlmann et al 2014; Hoyer et al 2009; Mooij et al 2016; Peters et al 2014), mainly because it produces tractable models which are useful for gaining insight into the system under study. Note also that linear structural equation models, which have extensively been studied for causal effect identification (Pearl

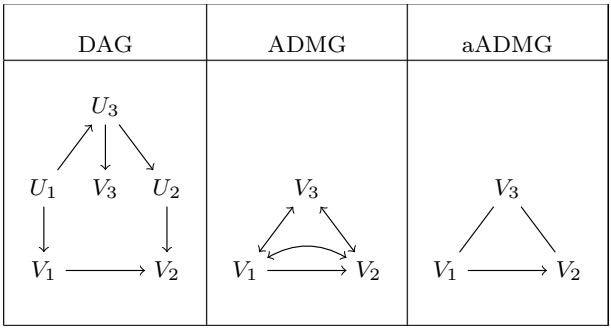

Fig. 1 Example where $f(v_2|\widehat{v_1})$ is identifiable from the aADMG but not from the ADMG

2009, Chapter 5), are additive noise models. As argued by Peña (2016), aADMGs are suitable for representing causal models with additive noise. The main difference between ADMGs and aADMGs is that an edge $A - B$ in an aADMG represents that the (unobserved) error variables associated with A and B are dependent and cannot be made conditionally independent given all of the other error variables, as opposed to a bidirected edge in an ADMG which represents marginal dependence due to confounding. Therefore, the ADMG and the aADMG of the domain at hand may encode different model assumptions. This may imply that one allows causal effect identification, whereas the other does not. We illustrate this with the example in Fig. 1, which is borrowed from Peña (2016). Peña and Bendtsen (2017) assign fictitious meanings to the variables in the example and provide additional examples. The ADMG and the aADMG in the figure represent the causal model over the observed variables represented by the DAG. The ADMG in the figure is derived from the DAG by keeping the directed edges between observed variables, and adding a bidirected edge between two observed variables A and B if and only if they have a confounder, i.e., the DAG has a subgraph of the form $A \leftarrow U_A \rightarrow \cdots \rightarrow U_B \rightarrow B$ or $A \leftarrow U_A \leftarrow \cdots \leftarrow U_C \rightarrow \cdots \rightarrow U_B \rightarrow B$ (Tian and Pearl 2002b, Sect. 5). The aADMG in the figure is derived from the DAG by keeping the directed edges between observed variables, and adding an undirected edge between two observed variables A and B if and only if the associated (unobserved) error variables are dependent and cannot be made conditionally independent given all of the other error variables, i.e., the DAG has a subgraph of the form $A \leftarrow U_A \rightarrow U_B \rightarrow B$ or $A \leftarrow U_A \rightarrow U_C \leftarrow U_B \rightarrow B$. Clearly, the causal effect on V_2 of intervening on V_1, denoted as the density function $f(v_2|\widehat{v_1})$, is not identifiable from the ADMG due to the confounding effect represented by the edge $V_1 \leftrightarrow V_2$ (Tian and Pearl 2002a, Theorem 4). However, $f(v_2|\widehat{v_1})$ is identifiable from the aADMG and is given by

$$f(v_2|\widehat{v_1}) = \int f(v_2|v_1, v_3) f(v_3)\, \mathrm{d}v_3. \tag{1}$$

To see it, note that the aADMG lacks the edge $V_1 - V_2$ which, as mentioned above, represents the assumption that U_1 and U_2 are independent given U_3 in the domain at hand. This implies that we can block all non-causal paths from V_1 to V_2 in the domain at hand by conditioning on V_3, since V_3 determines U_3 due to the additive noise assumption. Therefore, we can identify the desired causal effect by just adjusting for V_3. This is not possible in the ADMG because we cannot block all non-causal paths from V_1 to V_2 due to the confounding effect represented by $V_1 \leftrightarrow V_2$. This is an example where the assumptions encoded in the aADMG of the domain at hand allow causal effect identification, whereas the assumptions in the corresponding ADMG do not. However, the opposite can also happen: simply reverse the edge $U_3 \rightarrow U_2$ in Fig. 1. Then, the corresponding ADMG allows for causal effect identification, whereas the corresponding aADMG does not (Peña 2016). Therefore, ADMGs and aADMGs are not competing but complementary causal models. Their difference stems from using different families of graphical models to represent the independences between the error variables. Whereas ADMGs use BGs (also known as covariance graphs), aADMGs use UGs (also known as concentration graphs, Markov random networks, or Markov random fields). A missing edge in the former implies marginal independence, whereas a missing edge in the latter implies conditional independence given the rest of the nodes. In general, the BG and the UG over the error variables of the domain at hand are not Markov equivalent, i.e., they encode different assumptions about the domain at hand. That is why the ADMG of the domain at hand may allow causal effect identification, whereas the corresponding aADMG may not, or vice versa. Note that the ADMG and the aADMG share the same directed edges.

Causal effect identification in ADMGs is well understood (Pearl 2009; Shpitser and Pearl 2006; Tian and Pearl 2002a, b). The same is not true for aADMGs. As mentioned, aADMGs were proposed by Peña (2016), who mainly studied them as representation of statistical independence models. In particular, their global, local, and pairwise Markov properties were studied. Later, Peña and Bendtsen (2017) considered aADMGs for causal effect identification. Specifically, they presented a calculus similar to Pearl's *do*-calculus (Pearl 2009; Shpitser and Pearl 2006), and a decomposition of the density function represented by an aADMG that is similar to the Q-decomposition by Tian and Pearl (2002a, b). In this paper, we extend the decomposition to identify further causal effects. The result is a sound algorithm for causal effect identification in aADMGs. We also show that the algorithm follows from the calculus of interventions in Peña and Bendtsen (2017).

Then, we turn our attention to the use of aADMGs as representation of independence models. As mentioned, Peña (2016) describes Markov properties for aADMGs but no factorization property. We present a first attempt to fill in this gap by developing a factorization for the positive discrete probability distributions that are Markovian with respect to AMP CGs, which recall are a subclass of aADMGs. We also develop an algorithm to perform maximum likelihood estimation of the factors in the factorization. It is worth mentioning that a method for maximum likelihood estimation for Gaussian AMP CGs exists (Drton and Eichler 2006). It should also be mentioned that similar results exist for LWF and MVR CGs. Specifically, Lauritzen (1996) describes a factorization for the positive discrete probability distributions that are Markovian

with respect to LWF CGs, and makes use of the celebrated iterative proportional fitting (IPF) algorithm for maximum likelihood estimation of the factors in the factorization. The IPF algorithm guarantees convergence to a global maximum of the likelihood function under mild assumptions. Drton (2008, 2009) describes a factorization for the positive discrete probability distributions that are Markovian with respect to MVR CGs, as well as an algorithm for maximum likelihood estimation of the factors in the factorization. The algorithm, named iterative conditional fitting (ICF) algorithm, can be seen as being dual to the IPF algorithm. However, unlike the IPF algorithm, the ICF algorithm just guarantees convergence to a local maximum or saddle point of the likelihood function, but it has proven to perform well in practice.

The rest of the paper is organized as follows. Section 2 introduces some preliminaries, including a detailed account of aADMGs for causal modeling. Section 3 presents our novel algorithm for causal effect identification. It also proves that the algorithm is sound and it follows from a calculus of interventions. Section 4 presents our factorization for AMP CGs and the algorithm for maximum likelihood estimation. The paper ends with a discussion on follow-up questions worth investigating.

2 Preliminaries

Unless otherwise stated, all the graphs and density functions in this paper are defined over a finite set of continuous random variables V. We use uppercase letters to denote random variables and lowercase letters to denote their states. For the sake of readability, we use the elements of V to represent singletons, and sometimes we use juxtaposition to represent set union. An aADMG G is a graph with possibly directed and undirected edges but without directed cycles, i.e., $A \to \cdots \to A$ is forbidden. There may be up to two edges between any pair of nodes, but in that case the edges must be different and one of them must be undirected to avoid directed cycles. Edges between a node and itself are not allowed. A topological ordering of V with respect to G is an ordering such that if $A \to B$ is in G then $A < B$.

Given an aADMG G, the parents of a set $X \subseteq V$ in G are $Pa_G(X) = \{A | A \to B$ is in G with $B \in X\}$. The children of X in G are $Ch_G(X) = \{A | A \leftarrow B$ is in G with $B \in X\}$. The neighbours of X in G are $Ne_G(X) = \{A | A - B$ is in G with $B \in X\}$. The ancestors of X in G are $An_G(X) = \{A | A \to \cdots \to B$ is in G with $B \in X$ or $A \in X\}$. Moreover, X is called an ancestral set if $X = An_G(X)$. The descendants of X in G are $De_G(X) = \{A | A \leftarrow \cdots \leftarrow B$ is in G with $B \in X$ or $A \in X\}$. A route between two nodes V_1 and V_n on G is a sequence of (not necessarily distinct) edges $E_1, \ldots, E_{n-1}$ such that E_i links the nodes V_i and V_{i+1}. We do not distinguish between the sequences $E_1, \ldots, E_{n-1}$ and $E_{n-1}, \ldots, E_1$, i.e., they represent the same route. The route is called undirected if it only contains undirected edges. A component of G is a maximal set of nodes such that there is an undirected route in G between any pair of nodes in the set. The components of G are denoted as $C(G)$, whereas $Co_G(X)$ denotes the components to which the nodes in $X \subseteq V$ belong.[1] A set of nodes of G is complete if there exists an undirected

[1] In the original ADMGs, the components are usually called maximal C-components (Shpitser and Pearl 2006), and $Co_G(X)$ is called the district of X (Richardson 2003).

edge between every pair of nodes in the set. The complete sets of nodes of G are denoted as $\mathcal{K}(G)$. A clique of G is a maximal complete set of nodes. The cliques of G are denoted as $\mathcal{Q}(G)$. Given a set $W \subseteq V$, let G_W denote the subgraph of G induced by W, i.e., the aADMG over W that has all and only the edges in G whose both ends are in W. Similarly, let G^W denote the marginal aADMG over W, i.e., $A \to B$ is in G^W if and only if $A \to B$ is in G, whereas $A - B$ is in G^W if and only if $A - B$ is in G or $A - V_1 - \cdots - V_n - B$ is in G with $V_1, \ldots, V_n \notin W$.

A node C on a route in an aADMG G is said to be a collider on the route if $A \to C \leftarrow B$ or $A \to C - B$ is a subroute. Note that maybe $A = B$. Moreover, the route is said to be connecting given $Z \subseteq V$ when every collider on the route is in Z, and every non-collider on the route is outside Z. Let X, Y, and Z denote three disjoint subsets of V. When there is no route in G connecting a node in X and a node in Y given Z, we say that X is separated from Y given Z in G and denote it as $X \perp_G Y|Z$. The independence model represented by G is the set of separations $X \perp_G Y|Z$. Likewise, we denote by $X \perp_f Y|Z$ that X is independent of Y given Z in a density function f. We say that f satisfies the global Markov property or simply that it is Markovian with respect to G if $X \perp_G Y|Z$ implies that $X \perp_f Y|Z$ for all X, Y, and Z disjoint subsets of V.

Finally, we mention some properties that density functions satisfy as shown by, for instance, (Studený 2005, Chapter 2). For all X, Y, W, and Z disjoint subsets of V, every density function f satisfies the following four properties: symmetry $X \perp_f Y|Z \Rightarrow Y \perp_f X|Z$, decomposition $X \perp_f Y \cup W|Z \Rightarrow X \perp_f Y|Z$, weak union $X \perp_f Y \cup W|Z \Rightarrow X \perp_f Y|Z \cup W$, and contraction $X \perp_f Y|Z \cup W \wedge X \perp_f W|Z \Rightarrow X \perp_f Y \cup W|Z$. If f is positive, then it also satisfies the intersection property $X \perp_f Y|Z \cup W \wedge X \perp_f W|Z \cup Y \Rightarrow X \perp_f Y \cup W|Z$. Some (not yet characterized) probability distributions also satisfy the composition property $X \perp_f Y|Z \wedge X \perp_f W|Z \Rightarrow X \perp_f Y \cup W|Z$.

2.1 Causal interpretation of aADMGs

Let us assume momentarily that V is normally distributed. In this section, we show that an aADMG G can be interpreted as a system of structural equations with correlated errors. Specifically, the system includes an equation for each $A \in V$, which is of the form

$$A = \beta_A Pa_G(A) + U_A, \tag{2}$$

where U_A denotes the noise or error term. The error terms are represented implicitly in G. They can be represented explicitly by magnifying G into the aADMG G', as shown in Table 1. The magnification basically consists in adding the error nodes U_A to G and connect them appropriately. Figure 2 shows an example. Note that Eq. 2 implies that A is determined by $Pa_G(A) \cup U_A$ and U_A is determined by $A \cup Pa_G(A)$. Let U denote all the error nodes in G'. Formally, we say that $A \in V \cup U$ is determined by $Z \subseteq V \cup U$ when $A \in Z$ or A is a function of Z. We use $Dt(Z)$ to denote all

Table 1 Algorithm for magnifying an aADMG

	Input: An aADMG G.
	Output: The magnified aADMG G'.
1	Set $G' = G$
2	For each node A in G
3	Add the node U_A and the edge $U_A \rightarrow A$ to G'
4	For each edge $A - B$ in G
5	Replace $A - B$ with the edge $U_A - U_B$ in G'
6	Return G'

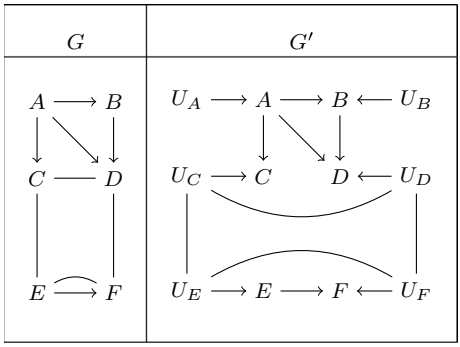

Fig. 2 Example of the magnification of an aADMG

the nodes that are determined by Z. From the point of view of the separations, that a node outside the conditioning set of a separation is determined by the conditioning set has the same effect as if the node was actually in the conditioning set. Bearing this in mind, it can be proven that, as desired, G and G' represent the same separations over V (Peña 2016, Theorem 9).

Finally, let $U \sim \mathcal{N}(0, \Lambda)$ such that $(\Lambda^{-1})_{U_A,U_B} = 0$ if $U_A - U_B$ is not in G'. Then, G can be interpreted as a system of structural equations with correlated errors as follows. For any $A \in V$

$$A = \sum_{B \in Pa_G(A)} \beta_{AB} B + U_A \tag{3}$$

and for any other $B \in V$

$$\text{covariance}(U_A, U_B) = \Lambda_{U_A,U_B}. \tag{4}$$

It can be proven that this causal interpretation of aADMGs works as intended: every density function $f(v)$ specified by Eqs. 3 and 4 is Gaussian and Markovian with respect to G (Peña 2016, Theorems 10 and 11).

A less formal but more intuitive interpretation of aADMGs is as follows. We can interpret the parents of each node in an aADMG as its observed causes. Its unobserved causes are summarized by an error node that is represented implicitly in the

Table 2 Algorithm for intervening on an aADMG.

	Input: An aADMG G and a set $X \subseteq V$.
	Output: The aADMG after intervening on X in G.
1	Delete from G all the edges $A \to B$ with $B \in X$
2	For each path $A - V_1 - \cdots - V_n - B$ in G with $A, B \notin X$ and $V_1, \ldots, V_n \in X$
3	Add the edge $A - B$ to G
4	Delete from G all the edges $A - B$ with $B \in X$
5	Return G

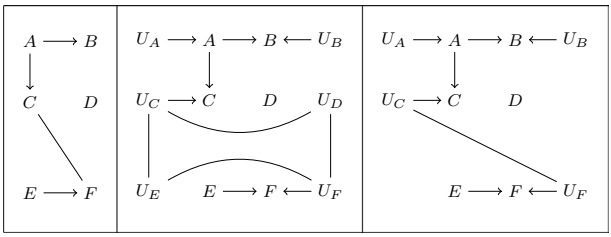

Fig. 3 Result of intervening on $\{D, E\}$ in the aADMG G in Fig. 2 (left), and on the magnified aADMG G' (center), and after marginalization of $\{U_D, U_E\}$ (right)

aADMG. We can interpret the undirected edges in the aADMG as the correlation relationships between the different error nodes. The causal structure is constrained to be a DAG, but the correlation structure can be any UG. This causal interpretation of aADMGs parallels that of the original ADMGs. There are, however, two main differences. First, the noise in ADMGs is not necessarily additive normal. Second, the correlation structure of the error nodes in ADMGs is represented by a covariance or bidirected graph. Therefore, whereas a missing edge between two error nodes in ADMGs represents marginal independence, in aADMGs it represents conditional independence given the rest of the error nodes. This means that the ADMG and the aADMG of the domain at hand may encode different assumptions, which may make a difference for causal effect identification, i.e., the effect may be identifiable in one model but not in the other. An example was provided in Sect. 1.

Given the above causal interpretation of an aADMG G, intervening on a set $X \subseteq V$ so as to change the natural causal mechanism of X amounts to modifying the right-hand side of the equations for the random variables in X. For simplicity, we only consider interventions that set variables to fixed values. Graphically, an intervention amounts to modifying G is shown in Table 2. Line 1 is shared with an intervention on an original ADMG. Lines 2–4 are best understood in terms of the magnified aADMG G': they correspond to marginalizing the error nodes associated with the nodes in X out of G'_U, the UG that represents the correlation structure of the error nodes. In other words, lines 2–4 replace G'_U with $(G'_U)^{U \setminus U_X}$, the marginal graph of G'_U over $U \setminus U_X$. This makes sense since U_X is no longer associated with X due to the intervention and, thus, we may want to marginalize it out because it is

unobserved. This is exactly what lines 2–4 imply. See Fig. 3 for an example. Note that the aADMG after the intervention and the magnified aADMG after the intervention represent the same separations over V (Peña 2016, Theorem 9). It can be proven that this definition of intervention works as intended: if $f(v)$ is specified by Eqs. 3 and 4, then $f(v\backslash x|\hat{x})$ is Markovian with respect to the aADMG resulting from intervening on X in G (Peña and Bendtsen 2017, Corollary 5).

It is worth mentioning that Eqs. 3 and 4 specify each node as a linear function of its parents with additive normal noise. The equations can be generalized to nonlinear or nonparametric functions as long as the noise remains additive, i.e., $A = g(Pa_G(A)) + U_A$ for all $A \in V$. The density function $f(u)$ can be any that is Markovian with respect to G'_U. That the noise is additive ensures that U_A is determined by $A \cup Pa_G(A)$, which is needed for Theorems 9 and 11 by Peña (2016) and Corollary 5 by Peña and Bendtsen (2017) to remain valid.[2] Hereinafter, we assume that $f(u)$ is positive. This is a rather common assumption in causal discovery (Peters et al 2017, Definition 7.3). For instance, it is justified when U is affected by measurement noise such that any value of U is possible. Moreover, note that if $f(u)$ is positive then $f(v)$ is also positive, which is desirable: it would be impossible to identify the effect of an intervention $\hat{x}$ if X never attains the value x in the observational regime (Pearl 2009, p. 78).

3 Causal effect identification in aADMGs

In this section, we present a novel sound algorithm for identifying arbitrary causal effects from aADMGs. The algorithm is based on a decomposition of $f(v)$. We also show that the algorithm follows from a calculus of interventions.

3.1 Identification by decomposition

Note that the system of structural equations corresponding to the causal model represented by an aADMG G induces a density function over $V = \{V_1, \ldots, V_n\}$, namely

$$f(v) = \int \left[\prod_i f(v_i|pa_G(V_i), u_i) \right] f(u)\, \mathrm{d}u. \tag{5}$$

Let $S_1, \ldots, S_k$ be a partition of V into components. Then $f(u)$ factorizes as

$$f(u) = \prod_j f(u_{S_j}) \tag{6}$$

[2] We believe that these results also hold under the post-nonlinear causal model assumption of Zhang and Hyvärinen (2009), i.e. $A = g_2(g_1(Pa_G(A)) + U_A)$ where g_2 is invertible. However, we do not explore this venue further in this paper.

because, as mentioned before, $f(u)$ is positive and Markovian with respect to G'_U. When a density function can be written as in Eqs. 5 and 6, we say that it factorizes according to G.

The density function induced by the post-interventional system of structural equations can be obtained from Eq. 5 by simply removing the terms for the variables intervened upon, that is

$$\begin{aligned} f(v\backslash x|\widehat{x}) &= \int \left[\prod_{V_i \in V\backslash X} f(v_i|pa_G(V_i), u_i) \right] f(u)\, \mathrm{d}u \\ &= \int \left[\prod_{V_i \in V\backslash X} f(v_i|pa_G(V_i), u_i) \right] f(u_{V\backslash X})\, \mathrm{d}u_{V\backslash X}. \end{aligned} \tag{7}$$

Moreover, we define the factor $q(c)$ with $C \subseteq V$ as follows:

$$q(c) = \int \left[\prod_{V_i \in C} f(v_i|pa_G(V_i), u_i) \right] f(u)\, \mathrm{d}u = \int \left[\prod_{V_i \in C} f(v_i|pa_G(V_i), u_i) \right] f(u_C)\, \mathrm{d}u_C.$$

Note from the two previous equations that $q(c) = f(c|\widehat{v\backslash c})$. Note also that $q(c)$ factorizes according to G^C. The next lemma shows that $q(c)$ is identifiable if C is a component in G. The proofs of the lemmas and theorems in this section can be found in Appendix A. Some proofs are adaptations of those by Tian and Pearl (2002a, b). We provide them for completeness.

Lemma 1 *Given an aADMG G,* let $S_1, \ldots, S_k$ *be a partition of V into components. Then,*

$$f(v) = \prod_j q(s_j)$$

and

$$q(s_j) = \prod_{V_i \in S_j} f(v_i|v^{(i-1)}),$$

where $V_1 < \cdots < V_n$ is a topological order of V with respect to G, and $V^{(i)} = \{V_1, \ldots, V_i\}$.

The following two lemmas show how certain factors are related. They will be instrumental later.

Lemma 2 *Given an aADMG G and two sets $E \subseteq C \subseteq V$ such that E is an ancestral set in G^C, then*

$$q(e) = \int q(c)\, d(c \backslash e).$$

Lemma 3 *Given an aADMG G, let $C_1, \ldots, C_k$ be a partition of a set $C \subseteq V$ into components in G^C. Then*

$$q(c) = \prod_j q(c_j)$$

and

$$q(c_j) = \prod_{V_i \in C_j} \frac{q(c^{(i)})}{q(c^{(i-1)})},$$

where $V_1 < \cdots < V_n$ is a topological order of C with respect to G^C, and $C^{(i)} = \{V_1, \ldots, V_i\}$. Moreover

$$q(c^{(i)}) = \int q(c)\, d(c \backslash c^{(i)}).$$

The previous lemmas can be generalized as follows. Let $A \subseteq V$ be an ancestral set in G, and let $B = V \backslash A$. Given $C \subseteq B$, we define the factor $q(c|a)$ as follows:

$$\begin{aligned} q(c|a) &= \int \left[\prod_{V_i \in C} f(v_i | pa_G(V_i), u_i) \right] f(u_B | u_A)\, \mathrm{d}u_B \\ &= \int \left[\prod_{V_i \in C} f(v_i | pa_G(V_i), u_i) \right] f(u_C | u_A)\, \mathrm{d}u_C. \end{aligned}$$

We now show that $q(c|a) = f(c|\widehat{b \backslash c}, a)$. Note that Eq. 7 implies that

$$\begin{aligned} & f(a, c | v \widehat{\backslash \{a, c\}}) \\ &= \left[\int \left[\prod_{V_i \in C} f(v_i | pa_G(V_i), u_i) \right] f(u_C | u_A)\, \mathrm{d}u_C \right] \prod_{V_i \in A} f(v_i | pa_G(V_i), u_i) f(u_A) \end{aligned}$$

because that A is an ancestral set in G implies that it determines U_A, and thus

$$f(a|\widehat{v\backslash\{a,c\}}) = \prod_{V_i \in A} f(v_i|pa_G(V_i), u_i) f(u_A)$$

by marginalization in the previous equation and recalling that A is an ancestral set in G which implies that no node in A has a parent in C. Then, combining the two previous equations implies that

$$f(c|\widehat{v\backslash\{a,c\}}, a) = \frac{f(a,c|\widehat{v\backslash\{a,c\}})}{f(a|\widehat{v\backslash\{a,c\}})} = \int \left[\prod_{V_i \in C} f(v_i|pa_G(V_i), u_i)\right] f(u_C|u_A)\,\mathrm{d}u_C$$
$$= q(c|a).$$

Note that $f(u_B|u_A)$ factorizes according to G'_{U_B} and, thus, $q(b|a)$ factorizes according to $H = G_B$. To see it, set $C = B$ in the previous equation. Then, $q(c|a)$ factorizes according to H^C.

Lemma 4 *Given an aADMG G and two disjoint sets $A, C \subseteq V$, then*

$$q(c|a) = \frac{q(a,c)}{\int q(a,c)\,dc}.$$

Moreover, if A is an ancestral set in $G^{A\cup C}$, then

$$q(c|a) = \frac{q(a,c)}{q(a)}.$$

The following three lemmas can be proven in much the same way as Lemmas 1–3.

Lemma 5 *Given an aADMG G and an ancestral set A in G, let $S_1, \ldots, S_k$ be a partition of $B = V\backslash A$ into components in $H = G_B$. Then*

$$f(b|a) = \prod_j q(s_j|a)$$

and

$$q(s_j|a) = \prod_{V_i \in S_j} f(v_i|v^{(i-1)}, a),$$

where $V_1 < \cdots < V_n$ is a topological order of B with respect to H, and $V^{(i)} = \{V_1, \ldots, V_i\}$.

Lemma 6 *Given an aADMG G, an ancestral set A in G, and two sets $E \subseteq C \subseteq V \backslash A$ such that E is an ancestral set in $(G_{V\backslash A})^C$, then*

$$q(e|a) = \int q(c|a)\, d(c \backslash e).$$

Lemma 7 *Given an aADMG G and an ancestral set A in G, let $B = V\backslash A$ and $H = G_B$. Also, let $C_1, \ldots, C_k$* be a partition of *$C \subseteq B$ into components in H^C. Then*

$$q(c|a) = \prod_j q(c_j|a)$$

and

$$q(c_j|a) = \prod_{V_i \in C_j} \frac{q(c^{(i)}|a)}{q(c^{(i-1)}|a)}$$

where $V_1 < \cdots < V_n$ is a topological order of C with respect to H^C, and $C^{(i)} = \{V_1, \ldots, V_i\}$. Moreover

$$q(c^{(i)}|a) = \int q(c|a)\, d(c \backslash c^{(i)}).$$

We are now in the position to introduce our sound algorithm for identifying an arbitrary causal effect $f(y|\widehat{x})$ from an aADMG G. Let X' be a maximal subset of X such that, for any $V_1 \in X'$, there is a path $V_1 \to \cdots \to V_n$ in G such that $V_n \in Y$ and $V_2, \ldots, V_n \notin X'$. Note that $f(y|\widehat{x}) = f(y|\widehat{x'})$. Hereinafter, we assume without loss of generality that $X' = X$. Let $B = De_G(X)$ and $A = V\backslash B$. Note that A is an ancestral set in G. Let $Y_1 = Y \cap A$ and $Y_2 = Y \cap B$. Then

$$\begin{aligned} f(y|\widehat{x}) &= \int f(y_2, a|\widehat{x})\, d(a \backslash y_1) \qquad = \int f(y_2|\widehat{x}, a) f(a|\widehat{x})\, d(a \backslash y_1) \\ &= \int f(y_2|\widehat{x}, a) f(a)\, d(a \backslash y_1), \end{aligned} \tag{8}$$

where the third equality follows from the fact that $A \cap De_G(X) = \emptyset$. Moreover

$$f(y_2|\widehat{x}, a) = \int f(b \backslash x|\widehat{x}, a)\, d(b \backslash \{x, y_2\}) = \int q(b \backslash x|a)\, d(b \backslash \{x, y_2\}).$$

Let $C = An_{(G_B)^{B\backslash X}}(Y_2)$. Then by Lemma 6

Table 3 Algorithm for causal effect identification from aADMGs.

	Input: An aADMG G and two disjoint sets $X, Y \subseteq V$.
	Output: An expression to compute $f(y\|\widehat{x})$ from $f(v)$ or FAIL.
1	Let $B = De_G(X)$ and $A = V \backslash B$
2	Let $Y_1 = Y \cap A$ and $Y_2 = Y \cap B$
3	Let $S_1, \ldots, S_k$ be a partition of B into components in G_B
4	Let $C = An_{(G_B)^{B\backslash X}}(Y_2)$
5	Let $C_1, \ldots, C_l$ be a partition of C into components in $(G_B)^C$
6	For each C_j such that $C_j \subseteq S_i$ do
7	Compute $q(s_i\|a)$ by Lemma 5
8	If C_j is an ancestral set in $(G_B)^{S_i}$ then
9	Compute $q(c_j\|a)$ from $q(s_i\|a)$ by Lemma 6
10	Else return FAIL
11	Return $\int [\int \prod_j q(c_j\|a)\, d(c\backslash y_2)] f(a)\, d(a\backslash y_1)$ by Lemma 7

$$f(y_2|\widehat{x}, a) = \int \int q(b\backslash x|a)\, d(b\backslash\{x, c\})\, d(c\backslash y_2) = \int q(c|a)\, d(c\backslash y_2).$$

Let $C_1, \ldots, C_l$ be a partition of C into components in $(G_B)^C$. Then by Lemma 7

$$f(y_2|\widehat{x}, a) = \int \prod_j q(c_j|a)\, d(c\backslash y_2). \tag{9}$$

Consequently, Eqs. 8 and 9 imply that $f(y|\widehat{x})$ is identifiable if $q(c_j|a)$ is identifiable for all j. Let $S_1, \ldots, S_k$ be a partition of B into components in G_B. Note that $C_j \subseteq S_i$ for some i, and recall that $q(s_i|a)$ is identifiable by Lemma 5, which implies that $q(c_j|a)$ is identifiable by Lemma 6 if C_j is an ancestral set in $(G_B)^{S_i}$. Table 3 summarizes the just described steps and the following theorem summarizes their correctness.

Theorem 1 *Given an aADMG G and two disjoint sets $X, Y \subseteq V$, if the algorithm in Table 3 returns an expression for $f(y|\widehat{x})$, then it is correct.*

Example 1 We run the algorithm in Table 3 to identify $f(v_2|\widehat{v_1})$ from the aADMG in Fig. 1. Then, $X = V_1$ and $Y = V_2$. Thus, $B = \{V_1, V_2\}$ and $A = V_3$ in line 1, and $Y_1 = \emptyset$ and $Y_2 = V_2$ in line 2. Then, $S_1 = V_1$ and $S_2 = V_2$ in line 3. Then, $C = V_2$ in line 4 and, thus, $C_1 = V_2$ in line 5. Note that $C_1 \subseteq S_2$ and, thus, $q(v_2|v_3) = f(v_2|v_1, v_3)$ by lines 6–9. Therefore, the algorithm returns $\int f(v_2|v_1, v_3) f(v_3)\, \mathrm{d}v_3$ which is the correct answer.

3.2 Identification by calculus

An alternative to the algorithm in Table 3 consists in repeatedly applying the rules below which, together with standard probability manipulations, aim to transform the causal effect of interest into an expression that only involves observational quantities. The rules are sound (Peña and Bendtsen 2017, Theorem 7). Given an aADMG G, let X, Y, Z, and W be disjoint subsets of V. The rules are as follows:

– Rule 1 (insertion/deletion of observations):

$$f(y|\hat{x},z,w) = f(y|\hat{x},w) \text{ if } Y \perp_{G_{\underrightarrow{X}}} Z|W,$$

where $G_{\underrightarrow{X}}$ denotes the graph obtained from G by deleting all directed edges in and out of X.

– Rule 2 (intervention/observation exchange):

$$f(y|\hat{x},\hat{z},w) = f(y|\hat{x},z,w) \text{ if } Y \perp_{G_{\underrightarrow{XZ}}} Z|W,$$

where $G_{\underrightarrow{XZ}}$ denotes the graph obtained from G by deleting all directed edges in and out of X and out of Z.

– Rule 3 (insertion/deletion of interventions):

$$f(y|\hat{x},\hat{z},w) = f(y|\hat{x},w) \text{ if } Y \perp_{G_{\underrightarrow{X}\overrightarrow{Z(W)}}} Z|W,$$

where $Z(W)$ denotes the nodes in Z that are not ancestors of W in $G_{\underrightarrow{X}}$, and $G_{\underrightarrow{X}\overrightarrow{Z(W)}}$ denotes the graph obtained from G by deleting all directed edges in and out of X and all undirected and directed edges into $Z(W)$.

We prove below that the algorithm in Table 3 actually follows from rules 1–3 and standard probability manipulations. To see it, note that all the steps in the algorithm involve standard probability manipulations except the application of Lemmas 5–7, which involve interventions. We prove below that these lemmas follow from rules 1–3. First, we prove that rule 1 is not really needed.

Lemma 8 *Rule 1 follows from rules 2 and 3.*

Lemma 9 *Lemmas 2 and 6 follow from rule 3.*

Lemma 10 *Lemmas 1, 3, 5, and 7 follow from rules 2 and 3.*

The following theorem summarizes the lemmas above.

Theorem 2 *Given an aADMG G and two disjoint sets $X, Y \subseteq V$, if the algorithm in Table 3 returns an expression for $f(y|\widehat{x})$, then it is correct. Moreover, the expression can also be obtained by repeated application of rules 2 and 3.*

4 Factorization property for discrete AMP CGs

In this section, we turn our attention to the use of aADMGs as representation of independence models. As mentioned, Peña (2016) describes Markov properties for aADMGs but no factorization property. We present a first attempt to fill in this gap by developing a factorization for the positive discrete probability distributions that are Markovian with respect to AMP CGs. As mentioned before, AMP CGs are a subclass of aADMGs with at most one edge between any pair of nodes and without semidirected cycles, i.e., $A \rightarrow B \multimap \cdots \multimap A$ is forbidden, where $\multimap$ stands for $\rightarrow$ or $-$. We show that the factorization property is equivalent to the existing Markov properties for AMP CGs. We also present an algorithm to perform maximum likelihood estimation of the factors in the factorization. Therefore, unless otherwise stated, all the graphs and probability distributions in this section are defined over a finite set of discrete random variables V. To be consistent with previous works on AMP CGs, we need to modify our definition of the set $De_G(X)$ in Sect. 2. In this section, the descendants of $X \subseteq V$ are $De_G(X) = \{A | A \circ\!\!- \cdots \circ\!\!- B$ with $B \in X$ or $A \in X\}$. The non-descendants of X are $Nd_G(X) = V \backslash De_G(X)$.

4.1 Markov properties

In Sect. 2, we introduced the global Markov property for aADMGs, which is a generalization of the global Markov property for AMP CGs developed by Andersson et al (2001) and Levitz et al (2001). Andersson et al also describe a block-recursive Markov property and show its equivalence to the global one. Specifically, a probability distribution p is Markovian with respect to an AMP CG G if and only if the following three properties hold for all $C \in \mathcal{C}(G)$ (Andersson et al 2001, Theorem 2):

- C1: $C \perp_p Nd_G(C) \backslash Co_G(Pa_G(C)) | Co_G(Pa_G(C))$.
- C2: $p(c|co_G(Pa_G(C)))$ is Markovian with respect to G_C.
- C3*: $D \perp_p Co_G(Pa_G(C)) \backslash Pa_G(D) | Pa_G(D)$ for all $D \subseteq C$.

The block-recursive property can be simplified. Specifically, C1, C2, and C3* hold if and only if the following two properties hold (Peña 2016, Theorem 6):

- C1*: $D \perp_p Nd_G(D) \backslash Pa_G(D) | Pa_G(D)$ for all $D \subseteq C$.
- C2*: $p(c|pa_G(C))$ is Markovian with respect to G_C.

Andersson et al also describe a local Markov property and show its equivalence to the global one under the assumption of the intersection and composition properties. Specifically, a probability distribution p satisfying the intersection and composition properties is Markovian with respect to an AMP CG G if and only if the following two properties hold for all $C \in \mathcal{C}(G)$ (Andersson et al 2001, Theorem 3):

- L1: $A \perp_p C \backslash (A \cup Ne_G(A)) | Nd_G(C) \cup Ne_G(A)$ for all $A \in C$.
- L2: $A \perp_p Nd_G(C) \backslash Pa_G(A) | Pa_G(A)$ for all $A \in C$.

The composition property assumption in the local property can be dropped. Specifically, a probability distribution p satisfying the intersection property is Markovian with respect to an AMP CG G if and only if the following two properties hold for all $C \in \mathcal{C}(G)$ (Peña 2016, Theorem 7):

- L1: $A \perp_p C \backslash (A \cup Ne_G(A)) | Nd_G(C) \cup Ne_G(A)$ for all $A \in C$.
- L2*: $A \perp_p Nd_G(C) \backslash Pa_G(A \cup S) | S \cup Pa_G(A \cup S)$ for all $A \in C$ and $S \subseteq C \backslash A$.

Andersson et al also describe a pairwise Markov property and show its equivalence to the global one under the assumption of the intersection and composition properties. Specifically, a probability distribution p satisfying the intersection and composition properties is Markovian with respect to an AMP CG G if and only if the following two properties hold for all $C \in \mathcal{C}(G)$ (Andersson et al 2001, Theorem 3]:

- P1: $A \perp_p B | Nd_G(C) \cup C \backslash (A \cup B)$ for all $A \in C$ and $B \in C \backslash (A \cup Ne_G(A))$.
- P2: $A \perp_p B | Nd_G(C) \backslash B$ for all $A \in C$ and $B \in Nd_G(C) \backslash Pa_G(A)$.

The composition property assumption in the pairwise property can be dropped. Specifically, a probability distribution p satisfying the intersection property is Markovian with respect to an AMP CG G if and only if the following two properties hold for all $C \in \mathcal{C}(G)$ (Peña 2016, Theorem 8):

- P1: $A \perp_p B | Nd_G(C) \cup C \backslash (A \cup B)$ for all $A \in C$ and $B \in C \backslash (A \cup Ne_G(A))$.
- P2*: $A \perp_p B | S \cup Nd_G(C) \backslash B$ for all $A \in C$, $S \subseteq C \backslash A$ and $B \in Nd_G(C) \backslash Pa_G(A \cup S)$.

4.2 Factorization

(Andersson et al 2001, p. 50) note that a probability distribution p that is Markovian with respect to an AMP CG G factorizes as

$$p(v) = \prod_{C \in \mathcal{C}(G)} p(c | co_G(Pa_G(C)))$$

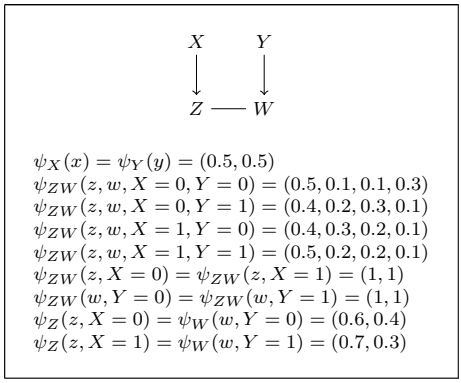

Fig. 4 Example of AMP CG factorization, where $\psi_{ZW}(z, w, X = 0, Y = 0) = (\alpha, \beta, \delta, \gamma)$ stands for $\psi_{ZW}(Z = 0, W = 0, X = 0, Y = 0) = \alpha, \psi_{ZW}(Z = 0, W = 1, X = 0, Y = 0) = \beta, \psi_{ZW}(Z = 1, W = 0, X = 0, Y = 0) = \delta$ and $\psi_{ZW}(Z = 1, W = 1, X = 0, Y = 0) = \gamma$

by C1. They also state that no further factorization of p appears to hold in general. We show in this section that this is not correct if p is positive. We start by proving an auxiliary result. The proofs of the lemmas and theorems in this section can be found in Appendix B.

Lemma 11 *Assume that p is positive and C1* holds. Then, C2* holds if and only if*

$$p(d|pa_G(C)) = \prod_{K \in \mathcal{K}(G^D)} \psi_D(k, pa_G(K)) \tag{10}$$

for all $D \subseteq C$, and where ψ_D are positive functions.

Remark 1 It is customary to think of the factors $\psi_D(k, pa_G(K))$ in Eq. 10 as arbitrary positive functions, whose product needs to be normalized to result in a probability distribution. Note, however, that Eq. 10 does not include any normalization constant. The reason is that the so-called canonical parameterization in Eq. 20 in Appendix B permits us to write any positive probability distribution as a product of factors that does not need subsequent normalization. One might think that this must be an advantage. However, the truth is that the cost of computing the normalization constant has been replaced by the cost of having to compute a large number of factors in Eq. 10. To see it, note that the size of $\mathcal{K}(G^D)$ is exponential in the size of the largest clique in G^D.

We can now introduce our necessary and sufficient factorization.

Theorem 3 *Let p be a positive probability distribution. Then, p is Markovian with respect to an AMP CG G if and only if*

$$p(v) = \prod_{C \in \mathcal{C}(G)} p(c|pa_G(C)) \tag{11}$$

with

$$p(d|pa_G(C)) = \prod_{K \in \mathcal{K}(G^D)} \psi_D(k, pa_G(K)) \tag{12}$$

for all $D \subseteq C$, and where ψ_D are positive functions.

Example 2 Figure 4 shows an example of the factorization in Theorem 3. All the random variables in the example are binary. Note that

$$\begin{aligned} p(z,w|x,y) &= \psi_{ZW}(z,w,x,y)\psi_{ZW}(z,x)\psi_{ZW}(w,y) = \psi_{ZW}(z,w,x,y) \\ p(z|x) &= \psi_Z(z,x) \\ p(w|y) &= \psi_W(w,y) \end{aligned}$$

and, thus, Eq. 12 actually imposes the constraints

$$\begin{aligned} \sum_w p(z,w|x,Y=0) &= \sum_w p(z,w|x,Y=1) = \psi_Z(z,x) \\ \sum_z p(z,w|X=0,y) &= \sum_z p(z,w|X=1,y) = \psi_W(w,y) \end{aligned}$$

or equivalently

$$\begin{aligned} \sum_w \psi_{ZW}(z,w,x,Y=0) &= \sum_w \psi_{ZW}(z,w,x,Y=1) = \psi_Z(z,x) \\ \sum_z \psi_{ZW}(z,w,X=0,y) &= \sum_z \psi_{ZW}(z,w,X=1,y) = \psi_W(w,y). \end{aligned}$$

For instance

$$\begin{aligned} &\psi_{ZW}(Z=0,W=0,X=0,Y=0) + \psi_{ZW}(Z=0,W=1,X=0,Y=0) \\ &= 0.5 + 0.1 = \psi_Z(Z=0,X=0) = 0.4 + 0.2 \\ &= \psi_{ZW}(Z=0,W=0,X=0,Y=1) + \psi_{ZW}(Z=0,W=1,X=0,Y=1). \end{aligned}$$

Remark 2 It follows from Theorem 3 and the proof of Lemma 11 that the positive probability distributions that are Markovian with respect to G can be parameterized by probabilities of the form $p(b, \overline{b}^* | pa_G(B), \overline{pa_G(B)}^*)$ for all $B \subseteq D$, $D \subseteq C$, and $C \in \mathcal{C}(G)$, where $\overline{b}^*$ and $\overline{pa_G(B)}^*$ denote the states that the variables in $D \backslash B$ and $Pa_G(D) \backslash Pa_G(B)$ take in d^* and $pa_G(D)^*$, which are arbitrary but fixed states of D and $Pa_G(D)$. Alternatively, we can parameterize the probability distributions by

factors of the form $\psi_D(k, pa_G(K))$ for all $K \in \mathcal{K}(G^D)$, $D \subseteq C$ and $C \in \mathcal{C}(G)$. Note that these parameters may be variation dependent or even functionally related, due to Eq. 12. In Example 2, for instance, the parameters $\psi_{ZW}(z, w, x, y)$ and $\psi_Z(z, x)$ are functionally related: setting the values for the former determines the values for the latter. That is why we avoid using the term "parameter" in the rest of this section, as some reserve this term for variation independent parameters. Instead, we use the term "factor". Although our factorization does not lead to a parametrization, it does bring some benefits: it induces a space efficient representation of the distribution at hand, and allows time efficient reasoning as well as data efficient estimation of the distribution.

In some cases, the following necessary and sufficient factorization may be more convenient.

Theorem 4 *Let p be a positive probability distribution. Then, p is Markovian with respect to an AMP CG G if and only if*

$$p(v) = \prod_{C \in \mathcal{C}(G)} p(c|pa_G(C)) \tag{13}$$

with

$$p(c|pa_G(C)) = \prod_{K \in \mathcal{K}(G_C)} \psi_C(k, pa_G(K)) \tag{14}$$

and

$$p(d|pa_G(C)) = p(d|pa_G(D)) \tag{15}$$

for all $D \subseteq C$, and where ψ_D are positive functions.

4.3 Factor estimation

Unfortunately, the factorization in Theorems 3 and 4 does not lead to an expression of the likelihood function that is easy to maximize, due to the constraints on the factors imposed by Eqs. 12 and 15 (recall Example 2 and Remark 2). To overcome this problem, we adapt the iterative conditional fitting (ICF) algorithm for maximum likelihood estimation in MVR CGs (Drton 2008; Drton and Richardson 2008). The algorithm just guarantees convergence to a local maximum or saddle point of the likelihood function, but it has proven to perform well in practice. Hereinafter, we focus on the factorization in Theorem 4 and drop the assumption that the product of factors in Eq. 14 is normalized. Specifically, we replace it with

Table 4 ICF algorithm for AMP CGs.

	Input: A sample from a positive probability distribution, and an AMP CG G.
	Output: Estimates of the factors in the factorization induced by G.
1	For each $C \in \mathcal{C}(G)$ do
2	Set $\varphi_C(k, pa_G(K))$ to arbitrary values for all $K \in \mathcal{Q}(G_C)$
3	Repeat until convergence
4	For each $K \in \mathcal{Q}(G_C)$ do
5	Solve a convex optimization problem to update $\varphi_C(k, pa_G(K))$ holding the rest of the factors fixed
6	Return $\varphi_C(k, pa_G(K))$ for all $C \in \mathcal{C}(G)$ and $K \in \mathcal{Q}(G_C)$

$$p(c|pa_G(C)) = \frac{\prod_{K \in \mathcal{Q}(G_C)} \varphi_C(k, pa_G(K))}{Z_C(pa_G(C))}, \tag{16}$$

where

$$Z_C(pa_G(C)) = \sum_c \prod_{K \in \mathcal{Q}(G_C)} \varphi_C(k, pa_G(K))$$

subject to the constraint that

$$Z_C(Pa_G(C) = \alpha) = Z_C(Pa_G(C) = \beta) \tag{17}$$

for all $\alpha \neq \beta$. The reason for this modification is given below.

The ICF algorithm for AMP CGs can be seen in Table 4. Due to Eq. 13, the log-likelihood function can be written as a sum of terms, each of which corresponds to the contribution of a component. We call these terms component log-likelihood functions. Specifically, the component log-likelihood function for $C \in \mathcal{C}(G)$ is

$$\begin{aligned} &\sum_{K \in \mathcal{Q}(G_C)} \sum_k \sum_{pa_G(K)} n(k, pa_G(K)) \log \varphi_C(k, pa_G(K)) \\ &- \sum_{pa_G(C)} n(pa_G(C)) \log Z_C(pa_G(C)), \end{aligned} \tag{18}$$

where $n(k, pa_G(K))$ is the number of instances in the available data where K and $Pa_G(K)$ take states k and $pa_G(K)$ simultaneously, and similarly for $n(pa_G(C))$. Since the factors corresponding to different components are variation independent, we can maximize the log-likelihood function by maximizing the component log-likelihood functions separately. The optimization problem to solve in line 5 of Table 4 consists in maximizing the component log-likelihood function for $C \in \mathcal{C}(G)$ holding all the factors but $\varphi_C(k, pa_G(K))$ fixed, and imposing the constraints due to Eqs. 15 and 17.

To see that this optimization problem is convex, note that the component log-likelihood function for $C \in \mathcal{C}(G)$ is concave since it is the log-likelihood function of a Markov network (Koller and Friedman 2009, Corollary 20.1), and the constraints are linear as follows. The constraints due to Eq. 17 are clearly linear in $\varphi_C(k, pa_G(K))$. The constraints due to Eq. 15 can be rephrased as

$$p(d|pa_G(D), Pa_G(C\backslash D) = \alpha) = p(d|pa_G(D), Pa_G(C\backslash D) = \beta) \tag{19}$$

for all $\alpha \neq \beta$, where

$$p(d|pa_G(C)) = \sum_{c\backslash d} p(c|pa_G(C)) = \frac{\sum_{c\backslash d} \prod_{K \in \mathcal{Q}(G_C)} \varphi_C(k, pa_G(K))}{Z_C(pa_G(C))}.$$

Note that the normalization constants on both sides of Eq. 19 coincide, due to the constraint in Eq. 17, and thus they can be removed. Therefore, the constraints due to Eq. 15 are also linear in $\varphi_C(k, pa_G(K))$. The optimization problem in line 5 is not only convex but also smooth, i.e., the objective function and the constraints have continuous derivatives. Thus, the problem can be solved via gradient ascent methods, for instance. Note that we cannot guarantee that the update in line 5 results in a proper probability distribution unless we normalize the product of factors, hence the change introduced in Eq. 16.

Example 3 We illustrate the ICF algorithm with the AMP CG in Fig. 4. Maximizing the component log-likelihood functions for X and Y results in the well-known closed-form estimates $\varphi_X(x) = n(x)/n$ and $\varphi_Y(y) = n(y)/n$, where n is the number of instances in the data available. To obtain the maximum likelihood estimates of $\varphi_{ZW}(z, w, x, y)$, we have to maximize the component log-likelihood function for ZW which, by Eq. 18, is

$$\sum_{z,w} \sum_{x,y} n(z, w, x, y) \log \varphi_{ZW}(z, w, x, y) - \sum_{x,y} n(x, y) \log \sum_{z,w} \varphi_{ZW}(z, w, x, y).$$

The maximization of the expression above is performed subject to the constraints due to Eqs. 15 and 17. The former constraints are

$$p(z|x, Y = 0) = p(z|x, Y = 1)$$
$$p(w|X = 0, y) = p(w|X = 1, y)$$

which are equivalent to

$$\sum_{w} p(z, w|x, Y = 0) = \sum_{w} p(z, w|x, Y = 1)$$
$$\sum_{z} p(z, w|X = 0, y) = \sum_{z} p(z, w|X = 1, y)$$

which are equivalent to

$$\sum_w \varphi_{ZW}(z, w, x, Y = 0) = \sum_w \varphi_{ZW}(z, w, x, Y = 1)$$
$$\sum_z \varphi_{ZW}(z, w, X = 0, y) = \sum_z \varphi_{ZW}(z, w, X = 1, y).$$

The constraints due to Eq. 17 are

$$\sum_{z,w} \varphi_{ZW}(z, w, X = 0, Y = 0) = \sum_{z,w} \varphi_{ZW}(z, w, X = 0, Y = 1)$$
$$\sum_{z,w} \varphi_{ZW}(z, w, X = 0, Y = 1) = \sum_{z,w} \varphi_{ZW}(z, w, X = 1, Y = 0)$$
$$\sum_{z,w} \varphi_{ZW}(z, w, X = 1, Y = 0) = \sum_{z,w} \varphi_{ZW}(z, w, X = 1, Y = 1).$$

Remark 3 As seen in Example 3, the factors corresponding to single-node components can be estimated in closed form. Therefore, instead of estimating the factors for the given AMP CG G, we may prefer to estimate the factors for an AMP CG G' that is Markov equivalent to G (i.e., it represents the same independence model) and has as many single-node components as possible. Luckily, G' can be obtained from G by repeatedly applying a so-called feasible split operation that, as the name suggests, splits a component of G in two (Sonntag and Peña 2015, Theorems 4 and 5).

5 Discussion

The ADMG and the aADMG of the domain at hand may encode different model assumptions, which may make a difference for causal effect identification, i.e., the effect of interest may be identifiable in one but not in the other. Causal effect identification in ADMGs is well understood. The same is not true for aADMGs. In this paper, we have shown that, as for the ADMGs, it is possible to develop a sound algorithm for identifying arbitrary causal effects from aADMGs. We have also shown that the algorithm follows from a calculus similar to Pearl's *do*-calculus. In the future, we would like to extend the algorithm in this paper so that it becomes complete for identifying arbitrary causal effects. We would also like to combine the original and alternative ADMGs into a family of mixed graphs with three types of edges, and develop a sound and complete algorithm for causal effect identification from them. Since using the right aADMG is crucial for causal effect identification, we are currently developing algorithms for learning aADMGs from observations. An assumption-free exact algorithm for learning aADMGs from observations and interventions has been proposed by Peña (2016). Given just observational data, the algorithm can only identify the true causal model up to Markov equivalence, because it is based on independence hypothesis tests. The algorithms that we are working on build on the

ideas by Hoyer et al (2009) (see also Peters et al 2017), who show how to exploit the nonlinearities in the data to identify the directions of the causal relationships.

In this paper, we have also considered the use of aADMGs as representation of independence models. Whereas Markov properties for aADMGs exist (Peña 2016), no factorization property exists as of today. In this paper, we have made a first attempt to fill in this gap by developing a factorization for the positive discrete probability distributions that are Markovian with respect to AMP CGs, which recall are a subclass of aADMGs. Unfortunately, finding the maximum likelihood estimates of the factors in the factorization is difficult in general. To solve this problem, we have adapted the iterative conditional fitting (ICF) algorithm (Drton 2008). However, the ICF algorithm only guarantees convergence to a local maximum or saddle point of the likelihood function. Therefore, the initial values of the factors are crucial to reach a good local maximum. Currently, our algorithm in Table 4 starts the search from arbitrary values. A more promising (albeit more computationally demanding) initialization may be running the iterative proportional fitting (IPF) procedure (Wainwright and Jordan 2008) to obtain estimates of the factors in Eqs. 13 and 14, i.e., without imposing the constraints due to Eq. 15. The constraints will be enforced by the subsequent run of the ICF algorithm. Table 5 shows the IPF algorithm. The multiplication and division in line 4 are elementwise. Moreover, p_e is the empirical probability distribution over V obtained from the given data, and p is the probability distribution over V due to the current estimates. Note that the component log-likelihood functions are variation independent, and each of them is the log-likelihood function of a Markov network. Since Markov networks are exponential families (Koller and Friedman 2009, Section 8.3), the IPF algorithm guarantees convergence to a global maximum (Wainwright and Jordan 2008, Section 6.1.2).

Note also that computing $p(k|pa_G(K))$ in line 4 of Table 5 requires inference. Fortunately, this can efficiently be performed by adapting the algorithm for inference in Bayesian and Markov networks developed by Lauritzen and Spiegelhalter (1988), and upon which most other inference algorithms build. Actually, the only step of the algorithm that needs to be adapted is the moralization step. Table 6 shows how

Table 5 IPF algorithm for initializing the ICF algorithm

	Input: A sample from a positive probability distribution, and an AMP CG G.
	Output: Initial estimates of the factors in the factorization induced by G.
1	For each $C \in \mathcal{C}(G)$ do
2	Set $\varphi_C(k, pa_G(K))$ to arbitrary values for all $K \in \mathcal{Q}(G_C)$
3	Repeat until convergence
4	Set $\varphi_C(k, pa_G(K)) = \varphi_C(k, pa_G(K)) \frac{p_e(k\|pa_G(K))}{p(k\|pa_G(K))}$ for all $K \in \mathcal{Q}(G_C)$
5	Return $\varphi_C(k, pa_G(K))$ for all $C \in \mathcal{C}(G)$ and $K \in \mathcal{Q}(G_C)$

Table 6 Moralization for the AMP CG in the IPF algorithm

	Input: An AMP CG G.
	Output: The moral graph of G.
1	Set $G^m = G$
2	For each $C \in \mathcal{C}(G)$ do
3	For each $K \in \mathcal{Q}(G_C)$ do
4	For each $X \in Pa_G(K)$ and $Y \in K$ do
5	Add the edge $X \to Y$ to G^m
6	For each $X, Y \in Pa_G(K)$ with $X \neq Y$ do
7	Add the edge $X - Y$ to G^m
8	Make all the edges in G^m undirected
9	Return G^m

to moralize the AMP CG G into an undirected graph G^m. Note that $K \cup Pa_G(K)$ is a complete subset of G^m. This guarantees that for every $K \in \mathcal{Q}(G_C)$ with $C \in \mathcal{C}(G)$, there will some clique in the triangulation of G^m that contains the set $K \cup Pa_G(K)$ and to which the factor $\varphi_C(k, pa_G(K))$ can be assigned. This is important in the subsequent steps of the inference algorithm. We plan to implement the ICF algorithm with both initializations, i.e., arbitrary values and the IPF algorithm, and report experimental results in a follow-up paper.

Appendix A: Proofs of the lemmas and theorems in Sect. 3

Proof of Lemma 1 To prove the first statement, note that

$$f(v) = \int \left[\prod_i f(v_i|pa_G(V_i), u_i)\right] f(u)\,\mathrm{d}u = \int \left[\prod_i f(v_i|pa_G(V_i), u_i)\right] \prod_j f(u_{S_j})\,\mathrm{d}u$$
$$= \prod_j \int \prod_{V_i \in S_j} f(v_i|pa_G(V_i), u_i) f(u_{S_j})\,\mathrm{d}u_{S_j} = \prod_j q(s_j),$$

where the second equality follows from Eq. (6).

We prove the second statement by induction over the number of variables in V. Clearly, the result holds when V contains a single variable. Assume as induction hypothesis that the result holds for up to n variables. When there are $n+1$ variables, these can be divided into components $S_1, \ldots, S_k, S'$ with factors $q(s_1), \ldots, q(s_k), q(s')$ such that $V_{n+1} \in S'$. As shown above

$$f(v) = q(s') \prod_j q(s_j).$$

Note also that $f(v^{(n)})$ factorizes according to $G^{V^{(n)}}$ and S_j is a component of $G^{V^{(n)}}$. Therefore

$$q(s_j) = \prod_{V_i \in S_j} f(v_i | v^{(i-1)})$$

by the induction hypothesis and the fact that $V_1 < \cdots < V_n$ is also a topological order of the nodes in $G^{V^{(n)}}$. Then, $q(s')$ is also identifiable and is given by

$$q(s') = \frac{f(v)}{\prod_j q(s_j)} = \frac{\prod_i f(v_i | v^{(i-1)})}{\prod_j q(s_j)} = \prod_{V_i \in S'} f(v_i | v^{(i-1)}).$$

Proof of Lemma 2

$$\begin{aligned}
&\int q(c)\, d(c \backslash e) \\
&= \int \int \left[\prod_{V_i \in E} f(v_i | pa_G(V_i), u_i) \prod_{V_i \in C \backslash E} f(v_i | pa_G(V_i), u_i) \right] f(u)\, \mathrm{d}u\, d(c \backslash e) \\
&= \int \left[\prod_{V_i \in E} f(v_i | pa_G(V_i), u_i) \int \prod_{V_i \in C \backslash E} f(v_i | pa_G(V_i), u_i)\, d(c \backslash e) \right] f(u)\, \mathrm{d}u \\
&= \int \left[\prod_{V_i \in E} f(v_i | pa_G(V_i), u_i) \right] f(u)\, \mathrm{d}u = q(e),
\end{aligned}$$

where the second equality follows from the fact that E is an ancestral set in G^C and, thus, no node in E has a parent in $C \backslash E$. The third equality is due to the fact that the integral over $c \backslash e$ equals 1. This may be easier to appreciate by performing the integral following an inverse topological order of the nodes in $C \backslash E$ with respect to G. □

Proof of Lemma 3 As mentioned above, $q(c)$ factorizes according to G^C. Therefore, the first statement can be proven in much the same way as the first statement in Lemma 1. The third statement follows from Lemma 2 since $C^{(i)}$ is an ancestral set in G^C.

We prove the second statement by induction over the number of variables in C. Clearly, the result holds when C contains a single variable. Assume as induction hypothesis that the result holds for up to n variables. When there are $n + 1$

variables, these can be divided into components $C_1, \ldots, C_k, C'$ with factors $q(c_1), \ldots, q(c_k), q(c')$ such that $V_{n+1} \in C'$. It follows from the first statement that:

$$q(c) = q(c') \prod_j q(c_j).$$

Note also that $q(c^{(n)})$ factorizes according to $G^{C^{(n)}}$ and C_j is a component of $G^{C^{(n)}}$. Therefore

$$q(c_j) = \prod_{V_i \in C_j} \frac{q(c^{(i)})}{q(c^{(i-1)})}$$

by the induction hypothesis and the fact that $V_1 < \cdots < V_n$ is also a topological order of the nodes in $G^{C^{(n)}}$. Then, $q(c')$ is given by

$$q(c') = \frac{q(c)}{\prod_j q(c_j)} = \frac{q(c^{(n+1)})}{\prod_j q(c_j)} = \frac{\prod_{i=1}^{n+1} \frac{q(c^{(i)})}{q(c^{(i-1)})}}{\prod_j q(c_j)} = \prod_{V_i \in C'} \frac{q(c^{(i)})}{q(c^{(i-1)})}.$$

□

Proof of Lemma 4 It suffices to note that

$$q(c|a) = f(c|\widehat{v \setminus \{a, c\}}, a) = \frac{f(a, c|\widehat{v \setminus \{a, c\}})}{f(a|\widehat{v \setminus \{a, c\}})} = \frac{q(a, c)}{\int q(a, c)\, dc}.$$

Moreover, if A is an ancestral set in $G^{A \cup C}$, then $\int q(a, c)\, dc = q(a)$ by Lemma 2. □

Proof of Lemma 8 (Huang and Valtorta 2006, Lemma 4) prove the same result for the original ADMGs. Their proof essentially applies to aADMGs too. Specifically, removing edges from an aADMG can only increase the separations represented by the aADMG. Then, if the antecedent of rule 1 is satisfied, so are the antecedents of rules 2 and 3. Then, we can replace the application of rule 1 with the application of rule 2 followed by the application of rule 3, i.e.,

$$f(y|\widehat{x}, z, w) = f(y|\widehat{x}, \widehat{z}, w) = f(y|\widehat{x}, w).$$

□

Proof of Lemma 9 We prove the result for Lemma 2. The proof for Lemma 6 is similar. First, note that

$$\int q(c)\, d(c\backslash e) = \int f(c|\widehat{v\backslash c})\, d(c\backslash e) = f(e|\widehat{v\backslash c}).$$

Moreover

$$q(e) = f(e|\widehat{v\backslash e}) = f(e|\widehat{v\backslash c}),$$

where the second equality follows from rule 3 since $E \perp_{G_{\underrightarrow{V\backslash CC\backslash E}}} C\backslash E|\emptyset$. To see that this separation holds, assume that there is a route ρ in $G_{\underrightarrow{V\backslash CC\backslash E}}$ between a node in E and a node in $C\backslash E$. Note that ρ cannot only contain nodes in C, because the nodes in $C\backslash E$ only have outgoing directed edges in $G_{\underrightarrow{V\backslash CC\backslash E}}$, which implies that E is not ancestral set in G^C, which contradicts the assumptions in Lemma 2. So, ρ must contain some node in $V\backslash C$. Note, however, that some node in $V\backslash C$ must be a collider in ρ because, in $G_{\underrightarrow{V\backslash CC\backslash E}}$, the nodes in $V\backslash C$ only have undirected edges whereas the nodes in $C\backslash E$ only have outgoing directed edges. Therefore, ρ is not connecting given $\emptyset$. □

Proof of Lemma 10 We prove the result for Lemma 1. The proofs for Lemmas 3, 5, and 7 are similar. Moreover, we only prove the first statement in Lemma 1, because the proof of the second statement provided in Lemma 1 only involves standard probability manipulations. Likewise, we do not need to prove the third statement of Lemmas 3 and 7 because, as shown in the proof of those lemmas, it follows from Lemma 2, which follows from rule 3 as shown in Lemma 9.

Let V be partitioned into components $S_1, \ldots, S_k$ for the aADMG G. Moreover, assume without loss of generality that if the edge $A \to B$ is in G, then $A \in S_i$ and $B \in S_j$ with $i \leq j$. Let $S_{<j} = \bigcup_{i<j} S_i$ and $S_{\leq j} = \bigcup_{i\leq j} S_i$. Note that

$$f(v) = \prod_j f(s_j|s_{<j}).$$

Moreover

$$f(s_j|s_{<j}) = f(s_j|\widehat{v\backslash s_{\leq j}}, s_{<j})$$

by rule 3 since $S_j \perp_{G_{\overrightarrow{V\backslash S_{\leq j}}}} V\backslash S_{\leq j} | S_{<j}$. To see that this separation holds, assume that there is a route ρ in $G_{\overrightarrow{V\backslash S_{\leq j}}}$ between a node in S_j and a node in $V\backslash S_{\leq j}$. Note that the nodes in $V\backslash S_{\leq j}$ only have outgoing directed edges in $G_{\overrightarrow{V\backslash S_{\leq j}}}$. Therefore, ρ implies that some node in $V\backslash S_{\leq j}$ is an ancestor in G of some node in $S_{\leq j}$, which contradicts our assumption above about the ordering of the components.

Finally, note that

$$f(s_j|\widehat{v\backslash s_{\leq j}}, s_{<j}) = f(s_j|\widehat{v\backslash s_j}) = q(s_j),$$

where the first equality follows from rule 2 because $S_j \perp_{G_{\overrightarrow{V\backslash S_{\leq j} S_{<j}}}} S_{<j} | \emptyset$. To see that this separation holds, assume that there is a route ρ in $G_{\overrightarrow{V\backslash S_{\leq j} S_{<j}}}$ between a node in S_j and a node in $S_{<j}$. Then, there exist two nodes $A \in S_j$ and $B \in S_{<j}$ that are adjacent in ρ or there exist two nodes $A' \in S_j$ and $B' \in V\backslash S_{\leq j}$ that are adjacent in ρ. However, either case implies a contradiction:

- $A - B$ contradicts that S_j is a component.
- $A \rightarrow B$ contradicts our assumption above about the ordering of the components.
- $A \leftarrow B$ contradicts that B has no outgoing directed edge in $G_{\overrightarrow{V\backslash S_{\leq j} S_{<j}}}$.
- $A' - B'$ contradicts that S_j is a component
- $A' \rightarrow B'$ and $A' \leftarrow B'$ contradict that B' only has undirected edges in $G_{\overrightarrow{V\backslash S_{\leq j} S_{<j}}}$.

Appendix B: Proofs of the lemmas and theorems in Sect. 4

Proof of Lemma 11 To prove the if part, it suffices to take $D = C$ and note that $G^D = G_C$. Then, C2* holds (Lauritzen 1996, Proposition 3.8). To prove the only if part, we adapt the proof of Theorem 3.9 by Lauritzen (1996) to prove that $p(d|pa_G(D))$ factorizes as indicated in Eq. 10. This implies the desired result by C1* and decomposition. Note that we cannot directly make use of Theorem 3.9 by Lauritzen (1996) because that would result in factors of the form $\psi_D(k, pa_G(C))$. So, we need to adapt the proof. Specifically, choose arbitrary but fixed states d^* and $pa_G(D)^*$ of D and $Pa_G(D)$. Given $B \subseteq D$, let $\overrightarrow{b}^*$ and $\overrightarrow{pa_G(B)}^*$ denote the states that the variables in $D\backslash B$ and $Pa_G(D)\backslash Pa_G(B)$ take in d^* and $pa_G(D)^*$. For all $B \subseteq D$, let

$$H_D(b, pa_G(B)) = \log p(b, \overrightarrow{b}^* | pa_G(B), \overrightarrow{pa_G(B)}^*).$$

Note that using the logarithm is warranted by the assumption of p being positive. For all $K \subseteq D$, let

$$\phi_D(k, pa_G(K)) = \sum_{B \subseteq K} (-1)^{|K \setminus B|} H_D(b, pa_G(B)), \tag{20}$$

where b and $pa_G(B)$ are the states that the variables B and $Pa_G(B)$ take in k and $pa_G(K)$. Now, we can apply the Möbius inversion (Lauritzen 1996, Lemma A.2) to obtain

$$\log p(d|pa_G(D)) = H_D(d, pa_G(D)) = \sum_{K \subseteq D} \phi_D(k, pa_G(K)),$$

where k and $pa_G(K)$ are the states that the variables K and $Pa_G(K)$ take in d and $pa_G(D)$. Then, it only remains to prove that $\phi_D(k, pa_G(K)) = 0$ whenever $K \notin \mathcal{K}(G^D)$. Consider two nodes S and T of K that are not adjacent in G^D. Then

$$\begin{aligned} \phi_D(k, pa_G(K)) = \sum_{B \subseteq K \setminus ST} (-1)^{|(K \setminus ST) \setminus B|} \big[& H_D(b, pa_G(B)) \\ - H_D(bs, pa_G(BS)) - H_D(bt, pa_G(BT)) + H_D(bst, pa_G(BST)) \big], & \end{aligned} \tag{21}$$

where b, bs, bt, and bst are the states that the variables B, BS, BT, and BST take in k, and likewise for $pa_G(B)$, $pa_G(BS)$, $pa_G(BT)$, and $pa_G(BST)$ with respect to $pa_G(K)$. Note that $S \perp_p T | (D \setminus ST) \cup Pa_G(C)$ by C2*, and $S \perp_p Pa_G(C) \setminus Pa_G(D) | (D \setminus ST) \cup Pa_G(D)$ by C1*, symmetry, decomposition, and weak union. Then, $S \perp_p T | (D \setminus ST) \cup Pa_G(D)$ by contraction and decomposition. This implies that

$$\begin{aligned} & H_D(bst, pa_G(BST)) - H_D(bs, pa_G(BS)) \\ & \quad = \log p(bst, \overline{bst}^* | pa_G(BST), \overline{pa_G(BST)}^*) \\ & \qquad - \log p(bs, \overline{bs}^* | pa_G(BS), \overline{pa_G(BS)}^*) \\ & \quad = \log p(s|b, \overline{bst}^*, pa_G(BST), \overline{pa_G(BST)}^*) \\ & \qquad + \log p(bt, \overline{bst}^* | pa_G(BST), \overline{pa_G(BST)}^*) \\ & \qquad - \log p(s|b, \overline{bst}^*, pa_G(BS), \overline{pa_G(BS)}^*) \\ & \qquad - \log p(b, \overline{bs}^* | pa_G(BS), \overline{pa_G(BS)}^*). \end{aligned}$$

Moreover, note that $S \perp_p Pa_G(T) \backslash Pa_G(D \setminus T) | (D \backslash ST) \cup Pa_G(D \backslash T)$ by C1*, symmetry, decomposition and weak union. This implies that

$$
\begin{aligned}
&H_D(bst, pa_G(BST)) - H_D(bs, pa_G(BS)) \\
&\quad = \log p(s|b, \overline{bst}^*, pa_G(BS), \overline{pa_G(BST)}^*) \\
&\qquad + \log p(bt, \overline{bst}^* | pa_G(BST), \overline{pa_G(BST)}^*) \\
&\qquad - \log p(s|b, \overline{bst}^*, pa_G(BS), \overline{pa_G(BST)}^*) \\
&\qquad - \log p(b, \overline{bs}^* | pa_G(BS), \overline{pa_G(BS)}^*) \\
&\quad = \log p(s^*|b, \overline{bst}^*, pa_G(BT), \overline{pa_G(BT)}^*) \\
&\qquad + \log p(bt, \overline{bst}^* | pa_G(BST), \overline{pa_G(BST)}^*) \\
&\qquad - \log p(s^*|b, \overline{bst}^*, pa_G(BT), \overline{pa_G(BT)}^*) \\
&\qquad - \log p(b, \overline{bs}^* | pa_G(BS), \overline{pa_G(BS)}^*).
\end{aligned}
$$

Note that the first and third terms of the right-hand side of the first equality above coincide. This warrants the changes made in the right-hand side of the second equality above. Moreover, $S \perp_p Pa_G(T) \backslash Pa_G(D \setminus T) | (D \backslash ST) \cup Pa_G(D \backslash T)$ also implies that

$$
\begin{aligned}
&H_D(bst, pa_G(BST)) - H_D(bs, pa_G(BS)) \\
&\quad = \log p(s^*|b, \overline{bst}^*, pa_G(BT), \overline{pa_G(BT)}^*) \\
&+ \log p(bt, \overline{bst}^* | pa_G(BST), \overline{pa_G(BST)}^*) \\
&- \log p(s^*|b, \overline{bst}^*, pa_G(B), \overline{pa_G(B)}^*) \\
&- \log p(b, \overline{bs}^* | pa_G(BS), \overline{pa_G(BS)}^*).
\end{aligned}
$$

Note also that $D \backslash S \perp_p Pa_G(S) \backslash Pa_G(D \setminus S) | Pa_G(D \backslash S)$ by C1* and decomposition. This implies that

$$
\begin{aligned}
&H_D(bst, pa_G(BST)) - H_D(bs, pa_G(BS)) \\
&\quad = \log p(s^*|b, \overline{bst}^*, pa_G(BT), \overline{pa_G(BT)}^*) \\
&\qquad + \log p(bt, \overline{bst}^* | pa_G(BT), \overline{pa_G(BT)}^*) \\
&\qquad - \log p(s^*|b, \overline{bst}^*, pa_G(B), \overline{pa_G(B)}^*) \\
&\qquad - \log p(b, \overline{bs}^* | pa_G(B), \overline{pa_G(B)}^*).
\end{aligned}
$$

Finally, as shown above $S \perp_p T | (D \backslash ST) \cup Pa_G(D)$, which implies that

$$
\begin{aligned}
& H_D(bst, pa_G(BST)) - H_D(bs, pa_G(BS)) \\
&= \log p(s^* | bt, \overline{bst}^*, pa_G(BT), \overline{pa_G(BT)}^*) \\
&\quad + \log p(bt, \overline{bst}^* | pa_G(BT), \overline{pa_G(BT)}^*) \\
&\quad - \log p(s^* | b, \overline{bs}^*, pa_G(B), \overline{pa_G(B)}^*) \\
&\quad - \log p(b, \overline{bs}^* | pa_G(B), \overline{pa_G(B)}^*) \\
&= \log p(bt, \overline{bt}^* | pa_G(BT), \overline{pa_G(BT)}^*) \\
&\quad - \log p(b, \overline{b}^* | pa_G(B), \overline{pa_G(B)}^*) \\
&= H_D(bt, pa_G(BT)) - H_D(b, pa_G(B)).
\end{aligned}
$$

Thus, all the terms in the square brackets in Eq. 21 add to zero, which implies that the entire sum is zero. □

Proof of Theorem 3 The only if part holds because C1* and decomposition imply Eq. 11, and Lemma 11 implies Eq. 12. To prove the if part, we prove that p satisfies C1* and C2*. Note that $Nd_G(C) = Nd_G(D)$. Note also that $p(d|pa_G(C))$ is a function of only D and $Pa_G(D)$ by Eq. 12. Then, Eq. 11 implies that

$$
\begin{aligned}
& p(d, nd_G(D)) && = p(d, nd_G(C)) \\
&= \left(\prod_{U \in C(G) : U \subseteq Nd_G(C)} p(u | pa_G(U)) \right) p(d | pa_G(C)) \\
&= g(nd_G(D)) h(d, pa_G(D))
\end{aligned}
$$

and thus C1* holds (Lauritzen 1996, Equation 3.6). Finally, C2* holds by Eq. 12 and Lemma 11.

Proof of Theorem 4 The only if part holds because C1* and decomposition imply Eqs. 13 and 15, and Lemma 11 implies Eq. 14. To prove the if part, we prove that p satisfies C1* and C2*. Note that $Nd_G(C) = Nd_G(D)$. This together with Eqs. 13 and 15 imply that

$$
\begin{aligned}
p(d, nd_G(D)) &= p(d, nd_G(C)) \\
&= \left(\prod_{U \in \mathcal{C}(G):\, U \subseteq Nd_G(C)} p(u|pa_G(U)) \right) p(d|pa_G(C)) \\
&= \left(\prod_{U \in \mathcal{C}(G):\, U \subseteq Nd_G(C)} p(u|pa_G(U)) \right) p(d|pa_G(D)) \\
&= g(nd_G(D))h(d, pa_G(D))
\end{aligned}
$$

and thus C1* holds (Lauritzen 1996, Equation 3.6). Finally, C2* holds by Eq. 14 (Lauritzen 1996, Proposition 3.8).

References

Andersson SA, Madigan D, Perlman MD (2001) Alternative Markov properties for chain graphs. Scand J Stat 28:33–85

Bühlmann P, Peters J, Ernest J (2014) CAM: causal additive models, high-dimensional order search and penalized regression. Ann Stat 42:2526–2556

Cox DR, Wermuth N (1996) Multivariate dependencies—models, analysis and interpretation. Chapman & Hall, London

Drton M (2008) Iterative conditional fitting for discrete chain graph models. In: Proceedings in computational statistics, pp 93–104

Drton M (2009) Discrete chain graph models. Bernoulli 15:736–753

Drton M, Eichler M (2006) Maximum likelihood estimation in gaussian chain graph models under the alternative markov property. Scand J Stat 33:247–257

Drton M, Richardson TS (2008) Binary models for marginal independence. J R Stat Soc B 70:287–309

Hoyer PO, Janzing D, Mooij J, Peters J, Schölkopf B (2009) Nonlinear causal discovery with additive noise models. Adv Neural Inf Process Syst 21:689–696

Huang Y, Valtorta M (2006) Pearl's Calculus of intervention is complete. In: Proceedings of the 22nd conference on uncertainty in artificial intelligence, pp 217–224

Koller D, Friedman N (2009) Probabilistic graphical models: principles and techniques. The MIT Press, USA

Koster JTA (2002) Marginalizing and conditioning in graphical models. Bernoulli 8:817–840

Lauritzen SL (1996) Graphical models. Oxford University Press, Oxford

Lauritzen SL, Spiegelhalter DJ (1988) Local computations with probabilities on graphical structures and their application to expert systems. J R Stat Soc B 50:157–224

Levitz M, Perlman MD, Madigan D (2001) Separation and completeness properties for AMP chain graph markov models. Ann Stat 29:1751–1784

Mooij JM, Peters J, Janzing D, Zscheischler J, Schölkopf B (2016) Distinguishing cause from effect using observational data: methods and benchmarks. J Mach Learn Res 17:1–102

Peña JM (2016) Alternative markov and causal properties for acyclic directed mixed graphs. In: Proceedings of the 32nd conference on uncertainty in artificial intelligence, pp 577–586

Peña JM, Bendtsen M (2017) Causal effect identification in acyclic directed mixed graphs and gated models. Int J Approx Reason 90:56–75

Pearl J (2009) Causality: models, reasoning, and inference. Cambridge University Press, Cambridge

Peters J, Mooij JM, Janzing D, Schölkopf B (2014) Causal discovery with continuous additive noise models. J Mach Learn Res 15:2009–2053

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

Richardson T (2003) Markov properties for acyclic directed mixed graphs. Scand J Stat 30:145–157
Richardson T, Spirtes P (2002) Ancestral graph markov models. Ann Stat 30:962–1030
Sadeghi K, Lauritzen SL (2014) Markov properties for mixed graphs. Bernoulli 20:676–696
Shpitser I, Pearl J (2006) Identification of conditional interventional distributions. In: Proceedings of the 22nd conference on uncertainty in artificial intelligence, pp 437–444
Sonntag D, Peña JM (2015) Chain graph interpretations and their relations revisited. Int J Approx Reason 58:39–56
Studený M (2005) Probabilistic conditional independence structures. Springer, New York
Tian J, Pearl J (2002a) A general identification condition for causal effects. In: Proceedings of the 18th national conference on artificial intelligence, pp 567–573
Tian J, Pearl J (2002b) On the identification of causal effects. Technical report R-290-L, Department of Computer Science, University of California, Los Angeles
Wainwright MJ, Jordan MI (2008) Graphical models, exponential families, and variational inference. Found Trends Mach Learn 1:1–305
Zhang K, Hyvärinen A (2009) On the identifiability of the post-nonlinear causal model. In: Proceedings of the 25th conference on uncertainty in artificial intelligence, pp 647–655