MARGINALIZATION AND CONDITIONING FOR LWF CHAIN GRAPHS

BY KAYVAN SADEGHI

Carnegie Mellon University

In this paper, we deal with the problem of marginalization over and conditioning on two disjoint subsets of the node set of chain graphs with LWF Markov property. For this purpose, we define the class of chain mixed graphs (CMGs) with three types of edges and, for this class, provide a separation criterion under which the class of CMGs is stable under marginalization and conditioning and contains the class of LWF chain graphs as its subclass. We provide a method for generating such graphs after marginalization and conditioning for a given CMG or a given LWF chain graph. We then define and study the class of anterical graphs, which is also stable under marginalization and conditioning and contains LWF CGs, but has a simpler structure than CMGs.

1. Introduction. In the area of graphical models, in which nodes are random variables and edges indicate some types of conditional dependencies, mixed graphs, which are graphs with different types of edges, have started to play an important role as they can deal with more complex independence structures that may arise in different statistical studies.

The first example of mixed graphs in the literature appeared in [9]. This was a chain graph (CG) with a specific interpretation of conditional independence, which is now generally known as the Lauritzen-Wermuth-Frydenberg or LWF interpretation. A formal interpretation, i.e. a Markov property, was later provided by [4]. This Markov property, together with other properties such as the factorization property was extensively discussed in [7]. By the term LWF CGs, one refers to the class of CGs with a specific independence structure that comes from the LWF Markov property.

It has become apparent that CGs with the LWF interpretation of independencies are important tools in capturing conditional independence structure of various probability distributions. For example, Studen y and Bouckaert [16] showed that for every CG, there exists a strictly positive discrete prob-

*Supported by grant #FA9550-12-1-0392 from the U.S. Air Force Office of Scientific Research (AFOSR) and the Defense Advanced Research Projects Agency (DARPA)

AMS 2000 subject classifications: Primary 62H99; secondary 62A99

Keywords and phrases: c-separation criterion, chain graph, independence model, LWF Markov property, m-separation, marginalization and conditioning, mixed graph
ability distribution that embodies exactly the independence statements displayed by the graph, and Peña [11] proved that almost all the regular Gaussian distributions that factorize with respect to a chain graph are faithful to it. This means that a Gaussian distribution chosen at random to factorize as specified by the LWF CG will have the independence structure of the graph and will satisfy no more independence constraints.

However, in the corresponding models to LWF CGs, when some variables are unobserved – also called latent or hidden – or when some variables are set to specific values, the implied independence structure, i.e. the corresponding independence structure after marginalization and conditioning respectively, is not well-understood.

The same problem for the well-known class of directed acyclic graphs (DAGs), which is a subclass of LWF CGs, has been a subject of study, and several classes of graphs have been defined in order to capture the marginal and conditional independence structure of DAGs. These include MC graphs [6], ancestral graphs [12], and summary graphs [17]; see also [13].

For LWF CGs, as it will be shown in this paper, one can capture the implied conditional independence structure by another LWF CG, but in general cannot capture the implied marginal independence structure by a CG. In this sense, CGs are stable under conditioning but not under marginalization.

Hence, in the case of marginalization, it is necessary to come up with a more complex class of graphs with a certain independence interpretation that captures the marginal independence structure of CGs, and in both cases of marginalization and conditioning, it is essential to provide methods by which the graphs that capture the marginal and conditional independence structure are generated. These are the main objectives of the current paper.

The structure of the paper is as follows: In the next section, we define mixed and chain graphs, and, for these classes of graphs, give graph theoretical definitions needed in this paper. In Section 3, we provide two equivalent ways for reading off independencies from a CG based on the LWF Markov property. In Section 4, we show that LWF CGs are stable under conditioning, and provide an algorithm for generating an LWF CG that captures the conditional independence structure of a given LWF CG. In Section 5, we define the class of chain mixed graphs with certain independence interpretation, and show that they capture the marginal independence structure of LWF CGs and that they are stable under marginalization, and provide an algorithm for generating such graphs after marginalization. In Section 6, we show that the class of CMGs is also stable under conditioning, provide the corresponding algorithm, and combine marginalization and conditioning for CMGs. In Section 7, we define the class of anteriorial graphs as a subclass
of CMGs, which also contains LWF CGs, and show that this class is stable under marginalization and conditioning. We also provide an algorithm for marginalization and conditioning for this class. In Section 8, we discuss the implications for probabilistic independence models that are faithful to LWF CGs.

2. Definitions for mixed graphs and chain graphs.

2.1. Basic graph theoretical definitions. A graph $G$ is a triple consisting of a node set or vertex set $V$, an edge set $E$, and a relation that with each edge associates two nodes (not necessarily distinct), called its endpoints. When nodes $i$ and $j$ are the endpoints of an edge, these are adjacent and we write $i \sim j$. We say the edge is between its two endpoints. We usually refer to a graph as an ordered pair $G = (V, E)$. Graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are called equal if $(V_1, E_1) = (V_2, E_2)$. In this case we write $G_1 = G_2$.

Notice that graphs that we use in this paper (and in general in the context of graphical models) are so-called labeled graphs, i.e. every node is considered a different object. Hence, for example, graph $i \rightarrow j \rightarrow k$ is not equal to $j \rightarrow i \rightarrow k$.

Here we introduce some basic graph theoretical definitions. A loop is an edge with the same endpoints. Multiple edges are edges with the same pair of endpoints. A simple graph has neither loops nor multiple edges. A complete graph is a simple graph with all pairs of nodes adjacent. A subgraph of a graph $G_1$ is graph $G_2$ such that $V(G_2) \subseteq V(G_1)$ and $E(G_2) \subseteq E(G_1)$ and the assignment of endpoints to edges in $G_2$ is the same as in $G_1$. An induced subgraph by a subset $A$ of the node set is a subgraph that contains all and only nodes in $A$ and all edges between two nodes in $A$.

A walk is a list $\langle i_0, e_1, i_1, \ldots, e_n, i_n \rangle$ of nodes and edges such that for $1 \leq m \leq n$, the edge $e_m$ has endpoints $i_{m-1}$ and $i_m$. A path is a walk with no repeated node or edge. A cycle is a walk with no repeated node or edge except $i_0 = i_n$. If the graph is simple then a path or a cycle can be determined uniquely by an ordered sequence of node sets. Throughout this paper, however, we use node sequences to describe paths and cycles even in graphs with multiple edges, but we assume that the edges of the path are all determined. It is usually apparent from the context or the type of the path which edge belongs to the path in multiple edges. We say a walk or a path is between the first and the last nodes of the list in $G$. We call the first and the last nodes endpoints of the walk or of the path. All other nodes are the inner nodes.
For a walk or path $\pi = \langle i_1, \ldots, i_n \rangle$, any subsequence $\langle i_k, i_{k+1}, \ldots, i_{k+p} \rangle$, $1 \leq k, k+p \leq n$, whose members appear consecutively in $\pi$, defines a subwalk or a subpath of $\pi$ respectively.

2.2. Some definitions for mixed graphs. A mixed graph is a graph containing three types of edges denoted by arrows, arcs (two-headed arrows), and lines (solid lines). Mixed graphs may have multiple edges of different types but do not have multiple edges of the same type. We do not distinguish between $i \rightarrow j$ and $j \rightarrow i$ or $i \rightarrow j$ and $j \rightarrow i$, but we do distinguish between $j \rightarrow i$ and $i \rightarrow j$. In this paper we are only considering mixed graphs that do not contain loops of any type. These constitute the class of loopless mixed graphs.

For mixed graphs, we say that $i$ is a neighbour of $j$ if these are endpoints of a line, and $i$ is a parent of $j$ and $j$ is a child of $i$ if there is an arrow from $i$ to $j$. We also define that $i$ is a spouse of $j$ if these are endpoints of an arc. We use the notations $\text{ne}(j)$, $\text{pa}(j)$, and $\text{sp}(j)$ for the set of all neighbours, parents, and spouses of $j$ respectively.

In the cases of $i \rightarrow j$ or $i \rightarrow j$ we say that there is an arrowhead pointing to (at) $j$.

A path $\langle i = i_0, i_1, \ldots, i_n = j \rangle$ is direction-preserving from $i$ to $j$ if all $i_ki_{k+1}$ edges are arrows pointing from $i_k$ to $i_{k+1}$. If there is a direction-preserving path from $j$ to $i$ then $j$ is an ancestor of $i$ and $i$ is a descendant of $j$. We denote the set of ancestors of $i$ by $\text{an}(i)$. Moreover, a cycle with the above property is called a direction-preserving cycle.

A path $\langle i = i_0, i_1, \ldots, i_n = j \rangle$ from $i$ to $j$ is a semi-direction-preserving path if it only consists of lines and arrows (it may contain only one type of edge), and every arrow $i_ki_{k+1}$ is pointing from $i_k$ to $i_{k+1}$. Thus a direction preserving path is a type of semi-direction-preserving path. We shall say that $i$ is anterior of $j$ if there is a semi-direction-preserving path from $i$ to $j$. We use the notation $\text{ant}(i)$ for the set of all anteriors of $i$. Notice that, since ancestral graphs have no arrowheads pointing to lines, our definition of anterior extends the notion of anterior used in [12] for ancestral graphs. Moreover, a cycle with the properties of semi-direction-preserving paths is called a semi-direction-preserving cycle.

A tripath is a path with three nodes. Note that [13] used the term V-configuration for such a path. However, here we follow [5] and most texts by letting a V-configuration be a tripath with non-adjacent endpoints.

In a mixed graph the inner node of three tripaths $i \rightarrow t \leftarrow j$, $i \leftarrow t \rightarrow j$, and $i \rightarrow t \rightarrow j$ is a collider (or a collider node) and the inner node of any other tripaths is a non-collider (or a non-collider node – also called a trans-
mitting node) on the tripath or more generally on any path of which the tripath is a subpath. We shall also say that the tripath itself with an inner collider or non-collider node is a collider or non-collider. We may speak of a collider or non-collider without mentioning the relevant tripath or path when this is apparent from context. Notice that a node may be a collider on one tripath and a non-collider on another.

A section of a walk in a mixed graph is a maximal subwalk that only consists of lines. Thus, any walk decomposes uniquely into sections (that are not necessarily edge-disjoint and may also be single nodes). As in any walk, we can also define the endpoints of a section. A section $\rho$ on a walk $\pi$ is called a collider section if one of the three following walks is a subwalk of $\pi$: $i \rightarrow \rho \rightarrow j$, $i \leftarrow \rho \leftarrow j$, and $i \rightarrow \rho \leftarrow j$. All other sections on $\pi$ are called non-collider sections. Similar to nodes, we may speak of a collider or non-collider sections without mentioning the relevant walk when this is apparent from context.

A trislide on a walk $\pi$ is a subpath $\langle i = i_0, i_1, \ldots, i_n = j \rangle$, where $ii_1$ and $i_{n-1}j$ are arrows or arcs and the subpath $\pi' = \langle i_1, \ldots, i_{n-1} \rangle$ is a section. Notice that the subpath $\pi'$ may be a single node, and therefore, by definition, tripaths that consist of arrows and arcs are considered trislides.

Three types of trislides $i \rightarrow o \ldots \rightarrow o \leftarrow j$, $i \leftarrow o \ldots \rightarrow o \leftarrow j$, and $i \leftarrow o \ldots \leftarrow o \leftarrow j$ are collider trislides and all other types of trislides are non-collider on any walk of which the trislide is defined.

Two walks $\pi_1$ and $\pi_2$ (including trislides, tripaths, or edges) between $i$ and $j$ are called endpoint-identical if there is an arrowhead pointing to $i$ in $\pi_1$ if and only if there is an arrowhead pointing to $i$ in $\pi_2$ and similarly for $j$. For example, the paths $i \rightarrow j$, $i \leftarrow k \leftarrow j$, and $i \rightarrow k \leftarrow l \leftarrow j$ are all endpoint-identical as they have an arrowhead pointing to $j$ but no arrowhead pointing to $i$ on the paths.

2.3. Chain graphs. Chain graphs (CGs) contain two types of edges (arrows and one symmetric type of edge, which can be lines or arcs); however, in this paper we are only interested in and work with CGs with arrows and full lines. Thus, in this paper, a CG is a graph consisting of lines and arrows that does not contain any semi-direction-preserving cycles with at least one arrow.

It is implied from the definition that CGs are characterized by having a node set that can be partitioned into disjoint subsets forming so-called chain components. These are connected subgraphs consisting only of undirected edges and are obtained by removing all arrows in the graph. All edges between nodes in the same chain component are full lines, and all
edges between different chain components are arrows. In addition, the chain
components can be ordered in such a way that all arrows point from a chain
with a higher number to one with a lower number.

For example, in Fig. 1(a) the graph is a chain graph with chain compo-
nents $\tau_1 = \{l, j, k\}$, $\tau_2 = \{h, q\}$, and $\tau_3 = \{p\}$, but in Fig. 1(b) the graph
is not a chain graph because of the existence of the $\langle h, k, q \rangle$ semi-direction-
preserving cycle.

![Fig 1](a) (b)

If one replaces every chain component with a single node, one obtains a
directed acyclic graph (DAG), a graph consisting exclusively of arrows and
without any direction-preserving cycles.

3. LWF Markov property for CGs. An independence model $J$ over
a set $V$ is a set of triples $\langle X, Y \mid Z \rangle$ (called independence statements), where
$X$, $Y$, and $Z$ are disjoint subsets of $V$ and $Z$ can be empty, and $\langle \emptyset, Y \mid Z \rangle$ and
$\langle X, \emptyset \mid Z \rangle$ are always included in $J$. The independence statement $\langle X, Y \mid Z \rangle$
is interpreted as “$X$ is independent of $Y$ given $Z$”. Notice that indepen-
dence models contain probabilistic independence models as a special case.
For further discussion on independence models, see [15].

A graph $G$ induces an independence model $J(G)$ by using a separation
criterion, which determines whether for three disjoint subsets $A$, $B$, and $C$
of the node set of $G$, $\langle A, B \mid C \rangle \in J(G)$. Such a criterion verifies whether $A$ is
separated from $B$ by $C$ in the sense that there are no paths of specific types
between $A$ and $B$ given $C$ in the graph. Such a separation is denoted by
$A \perp B \mid C$. It is clear that $J(G)$ satisfies the global Markov property, which
states that if $A \perp B \mid C$ in $G$ then $\langle A, B \mid C \rangle \in J$.

For CGs, at least four different separation criteria, i.e. four different types
of global Markov property have been discussed in the literature. Drton [3]
has classified them as (1) the LWF or block concentration Markov property,
(2) the AMP or concentration regression Markov property, as defined and
studied by [1], (3) a Markov property that is dual to the AMP Markov
property, and (4) the multivariate regression Markov property, as introduced by [2] and studied extensively recently; for example see [10; 18].

In this paper, we are interested in the LWF Markov property, and we introduce two equivalent separation criteria for this in this section. Henceforth, for the sake of brevity, by CGs we refer to CGs with the LWF Markov property.

The moralization criterion for CGs was defined in [4] and is a generalization of the moralization criterion for DAGs defined in [8]; see also [7]. The moral graph of a chain graph $G$, denoted by $(G)^m$ is a graph that consists only of full lines and that is generated from $G$ as follows: For every edge $ij$ in $G$ there is a line $ij$ in $(G)^m$. In addition if nodes $i$ and $j$ are parents of the same chain component in $G$ then there is the line $ij$ in $(G)^m$.

Now let $G_{\text{ant}}(A \cup B \cup C)$ be the induced subgraph of $G$ generated by $\text{ant}(A \cup B \cup C)$. The moralization criterion states that for $A$, $B$, and $C$, three disjoint subsets of the node set of $G$, if there are no paths between $A$ and $B$ in $(G_{\text{ant}}(A \cup B \cup C))^m$ whose inner nodes are outside $C$ then $A \perp_{\text{mor}} B \mid C$.

An equivalent criterion, called the c-separation criterion for CGs was defined in [16]. Here we present a simpler version of that criterion, presented in [14], with a different notation and wording:

A walk $\pi$ in a CG is a $c$-connecting walk given $C$ if every collider section of $\pi$ has a node in $C$ and all non-collider sections are outside $C$. We say that $A$ and $B$ are $c$-separated given $C$ if there are no $c$-connecting walks between $A$ and $B$ given $C$, and we use the notation $A \perp_c B \mid C$.

Notice that, as mentioned in [16], there is potentially an infinite number of walks, and therefore, this might not be an appropriate criterion for testing independencies. Although, in this paper, we only use this criterion in order to prove our theoretical results regarding marginalization and conditioning, and an infinite number of walks is not an issue for this purpose, in [14], it was shown that this criterion can also be implemented with an algorithm.

For example, in the graph of Fig. 2(a), the independence statement $j \perp h \mid l$ does not hold. This can be seen by looking at the moral graph $(G_{\text{ant}}(\{j,h,l\}))^m = (G_{\text{ant}}(\{j,h,k,q,l,r\}))^m$ in Fig. 2(b), and observing that the inner nodes of the path $\langle j, k, q, h \rangle$ are outside the conditioning set. The same conclusion can be made by looking at the walk $\langle j, k, l, r, q, h \rangle$, where the non-collider sections $k$ and $q$ are outside the conditioning set, but the inner node $l$ of the collider section $\langle l, r \rangle$ is in the conditioning set.

The equivalence of the moralization criterion and the original $c$-separation criterion was proven in Consequence 4.1 in [16]. The equivalence with the mentioned simplified criterion was proven in [14]. We use the notation $J_c(G)$ for the independence model induced from $G$ by the above criteria.
We first prove the following lemma, which provides an equivalent type of walk to $c$-connecting walks:

**Lemma 1.** There is a $c$-connecting walk between $i$ and $j$ given $C$ if and only if there is a walk between $i$ and $j$ whose sections are all paths, and on which nodes of every collider section are in $C \cup \text{ant}(C)$, and non-collider sections are outside $C$. In addition, these walks can be chosen to be endpoint-identical.

**Proof.** ($\Rightarrow$) Suppose that there is a $c$-connecting walk $\pi$ between $i$ and $j$ given $C$. Consider the shortest subpath $\rho_0$ of the section $\rho$ of $\pi$ between $k$ and $l$. If $\rho$ is collider then a node of $\rho$ is in $C$, and since all the nodes on $\rho$ (including those on $\rho_0$) are connected by lines, they are all in $C \cup \text{ant}(C)$. If $\rho$ is a non-collider then all the nodes on $\rho$ (including those on $\rho_0$) are outside $C$. Hence, by replacing all such $\rho$ by $\rho_0$ we obtain the desired walk.

($\Leftarrow$) Suppose that there is a walk $\pi$ between $i$ and $j$ whose sections are all paths and nodes of every collider section are in $C \cup \text{ant}(C)$, and non-collider sections are outside $C$. We keep all non-collider sections of $\pi$ intact. For a collider section $\rho$ between $k$ and $l$, if there is a node of $\rho$ in $C$, we keep it intact. Otherwise we replace $\rho$ with $\rho_4 = (k, \rho_1, \rho_2, c, \rho_2^\prime, \rho_3, l)$, where $\rho_1$ is a subpath of $\rho$ between $k$ and $h$, $\rho_2$ is a semi-direction-preserving path from $h$ to a member $c$ of $C$, $\rho_2^\prime$ is $\rho_2$ in the reverse direction, and $\rho_3$ is a subpath of $\rho$ between $h$ and $l$. It is easy to observe that $\rho_4$ is $c$-connecting given $C$. (If there is an arrow on $\rho_2$ then $\rho_4$ consists of non-collider sections containing $\rho_1$ and $\rho_3$, and a collider section containing $c$; otherwise $\rho_4$ is a collider section containing $c$.) Hence, by this replacement for all such $\rho$ on $\pi$, we obtain a $c$-connecting walk given $C$ between $i$ and $j$.

Finally, from the construction of walks that we have in both directions of the proof, it is seen that the walks are endpoint-identical. □
4. Stability of CGs under conditioning.  For a subset $C$ of $V$, the independence model after conditioning on $C$, denoted by $\alpha(J; \emptyset, C)$, is

$$\alpha(J; \emptyset, C) = \{\langle A, B | D \rangle : \langle A, B | D \cup C \rangle \in J \text{ and } (A \cup B \cup D) \cap C = \emptyset\}.$$ 

One can observe that $\alpha(J; \emptyset, C)$ is an independence model over $V \setminus C$.

We now present the definition of stability under conditioning as defined in [13]: Consider a family of graphs $\mathcal{T}$. If, for every graph $G = (V, E) \in \mathcal{T}$ and every disjoint subsets $C$ of $V$, there is a graph $H \in \mathcal{T}$ such that $J(H) = \alpha(J(G); \emptyset, C)$ then $\mathcal{T}$ is stable under conditioning. Notice that if the node set of such a graph $H$ is $N$ then $N = V \setminus C$.

Here in this section we prove that CGs are stable under conditioning. For this purpose, we provide an algorithm that, from a chain graph $G$ and after conditioning on $C$, generates a CG with the corresponding independence model after conditioning on $C$.

Algorithm 1. $\alpha_{CG}(G; \emptyset, C)$: (Generating a CG from a chain graph $G$ after conditioning on $C$)
Start from $G$.

1. Find all nodes in $C \cup \text{ant}(C)$ and call this set $S$.
2. Generate a line between the endpoints of collider trislides (including tripaths) with at least one inner node in $S$ if the line does not already exist.
3. Remove the arrowheads of all arrows pointing to members of $S$ (i.e. turn such arrows into lines).
4. Remove all nodes in $C$.

Notice that if one inner node of a trislide is in $S$ then so are all its inner nodes. In addition, step 2 of the algorithm is not applied repeatedly, i.e. the newly generated lines are not used for detecting new collider trislides. Fig. 3 illustrates how to apply Algorithm 1 step by step to a CG.

We consider Algorithm 1 to be a function, denoted by $\alpha_{CG}(G; \emptyset, C)$, from the set of CGs and a conditioning subset of their node set to the set of CGs. Notice that for every chain graph $G$, it holds that $\alpha_{CG}(G; \emptyset, \emptyset) = G$. We first show that $\alpha_{CG}(G; \emptyset, C)$ is a CG:

Proposition 1. Graphs generated by Algorithm 1 are CGs.

Proof. The generated graphs obviously contain only lines and arrows, thus it is enough to prove that they do not contain semi-direction-preserving cycles with an arrow. Suppose, for contradiction, that a generated graph does
contain a semi-direction-preserving cycle \( \pi \) with an arrow. If a line \( ij \) on \( \pi \) has been generated by step 3 then \( i, j \in S \) in \( G \) and, therefore, all nodes on \( \pi \) are in \( S \). This implies that there is no arrow on \( \pi \), a contradiction. If a line \( kl \) has been generated by step 2 then it is easy to see that both \( k, l \in S \), and again there is no arrow on \( \pi \), a contradiction. Therefore, all lines on \( \pi \) exist in the original graph, and no arrows are generated by the algorithm. Hence, \( \pi \) exists in the original graph, a contradiction.

The following theorem shows that \( \alpha_{CG}(\cdot; \emptyset, \cdot) \) is well-defined in the sense that, instead of directly generating a CG, we can split the nodes that we condition on into two parts, first generate the CG related to the first part, then from the generated CG, generate the desired CG related to the second part.
Theorem 1. For a chain graph $G$ and disjoint subsets $C$ and $C_1$ of its node set,

$$\alpha_{CG}(\alpha_{CG}(G; \emptyset, C); \emptyset, C_1) = \alpha_{CG}(G; \emptyset, C \cup C_1).$$

Proof. Since there are no arrows generated by Algorithm 1, it is enough to show that a line in $\alpha_{CG}(G; \emptyset, C \cup C_1)$ is a line in $\alpha_{CG}(\alpha_{CG}(G; \emptyset, C); \emptyset, C_1)$ and vice versa.

$(\Rightarrow)$ Suppose that there is a line between $i$ and $j$ in $\alpha_{CG}(G; \emptyset, C \cup C_1)$, and this line has been generated by step 3. We then have that $i, j \in \text{ant}(C \cup C_1)$ in $G$. Notice that $\text{ant}(C \cup C_1) = \text{ant}(C) \cup \text{ant}(C_1)$. If $i, j \in \text{ant}(C)$ in $G$ then the $ij$ arrow turns into a line in $\alpha_{CG}(G; \emptyset, C)$, and remains a line in $\alpha_{CG}(\alpha_{CG}(G; \emptyset, C); \emptyset, C_1)$. If $i, j \in \text{ant}(C_1) \setminus \text{ant}(C)$ in $G$ then since semi-direction-preserving paths in $G$ remain semi-direction-preserving in $\alpha_{CG}(G; \emptyset, C)$, $i, j \in C_1 \cup \text{ant}(C_1)$ in $\alpha_{CG}(G; \emptyset, C)$. Therefore, the $ij$ arrow turns into a line in $\alpha_{CG}(\alpha_{CG}(G; \emptyset, C); \emptyset, C_1)$.

Now suppose that the line has been generated by step 2. In this case, there is a collider trislide $\pi$ between $i$ and $j$ with inner nodes in $C \cup C_1 \cup \text{ant}(C \cup C_1)$ in $G$. If one inner node is in $C \cup \text{ant}(C)$ then there is a line in $\alpha_{CG}(G; \emptyset, C)$ between $i$ and $j$. If there are no inner nodes in $C \cup \text{ant}(C)$ then $\pi$ exists in $\alpha_{CG}(G; \emptyset, C)$ since $i$ and $j$ are not in $\text{ant}(C)$ and arrowheads remain intact. Therefore, a line is generated between $i$ and $j$ in $\alpha_{CG}(\alpha_{CG}(G; \emptyset, C); \emptyset, C_1)$.

$(\Leftarrow)$ Suppose that there is a line between $i$ and $j$ in $\alpha_{CG}(\alpha_{CG}(G; \emptyset, C); \emptyset, C_1)$. First, suppose that in Algorithm 1 applied to $\alpha_{CG}(G; \emptyset, C)$ for $C_1$, this line has been generated by step 3. It implies that $i, j \in \text{ant}(C_1)$ in $\alpha_{CG}(G; \emptyset, C)$. The semi-direction-preserving path from $i$ to a member of $C_1$ in $\alpha_{CG}(G; \emptyset, C)$ remains semi-direction-preserving in $G$ unless a line was generated by steps 2 or 3 of the algorithm. In these cases, it is easy to observe that $i \in \text{ant}(C)$ in $G$. Therefore, in either case, $i, j \in \text{ant}(C \cup C_1)$ in $G$. Hence, by step 3 the edge $ij$ is a line in $\alpha_{CG}(G; \emptyset, C \cup C_1)$.

Now we show that the line could not be generated by step 2. If it is then there is a collider trislide $\pi$ between $i$ and $j$ in $\alpha_{CG}(G; \emptyset, C)$ with inner nodes in $C_1 \cup \text{ant}(C_1)$. A line on $\pi$ cannot be generated by step 3 of the algorithm applied to $G$ since otherwise $i, j \in \text{ant}(C)$ and the arrows on $\pi$ would have turned into lines. It cannot be generated by step 2 either since again it is easy to see that $i, j \in \text{ant}(C)$.

Denote now by $\mathcal{CG}$ the set of all CGs. We first provide the following trivial statement:

Proposition 2. The map $\alpha_{CG}(\cdot; \emptyset, \cdot) : \mathcal{CG} \rightarrow \mathcal{CG}$ is surjective.
Proof. The result follows from the fact that $\alpha_{CG}(G; \emptyset, \emptyset) = G$. □

Here we introduce the core idea in proposing Algorithm 1: The modification applied by the function should generate a graph that induces the conditional independence model.

Theorem 2. For a chain graph $G$ and disjoint subsets $A$, $B$, $C$, and $C_1$ of its node set,

$$\langle A, B \mid C_1 \rangle \in \mathcal{J}_c(\alpha_{CG}(G; \emptyset, C)) \iff \langle A, B \mid C \cup C_1 \rangle \in \mathcal{J}_c(G).$$

Proof. We need to prove that $A \perp \!\!\!\!\perp B \mid C \cup C_1$ in $G$ $\iff$ $A \perp \!\!\!\!\perp B \mid C_1$ in $\alpha_{CG}(G; \emptyset, C)$.

($\Rightarrow$) Suppose that there is a $c$-connecting walk $\pi$ given $C \cup C_1$ between $i$ and $j$ in $G$. Notice that all non-collider sections of $\pi$ are outside $C$. Suppose that there is a collider section $\rho$ on $\pi$ with a node in $C$. By Lemma 1, there is a $kl$ line after applying step 2. By replacing the trislide by the $kl$ line, we show that the walk remains $c$-connecting: This is obviously true unless possibly the $kl$ line makes a larger section $\rho'$. Notice that no node on $\rho'$ is in $C \cup C_1$ since otherwise there would have been a non-collider section with an inner node in $C \cup C_1$ on $\pi$, and $\pi$ would not have been $c$-connecting given $C \cup C_1$. $\rho'$ is, therefore, a non-collider section outside $C \cup C_1$, and the claim is true.

By an inductive argument, after applying step 2 of the algorithm, we obtain a $c$-connecting walk given $C \cup C_1$ with no node in $C$. Therefore it is a $c$-connecting walk given $C_1$. After applying step 3 of the algorithm, if a collider section $\rho''$ (on $\langle r, \rho'', s \rangle$) turns into a non-collider section then nodes on $\rho''$ are in $\text{ant}(C)$ and, therefore, there is a collider trislide $\langle r, \rho_0', s \rangle$ before applying step 3, which generated an $rs$ line by step 2. This line can be used instead of the trislide. By induction, the result follows.

($\Leftarrow$) Suppose that there is a $c$-connecting walk $\pi$ given $C_1$ between $i$ and $j$ in $\alpha_{CG}(G; \emptyset, C)$. We prove that there is a $c$-connecting walk given $C \cup C_1$ between $i$ and $j$ in $G$: Let $G_2$ be $\alpha_{CG}(G; \emptyset, C)$ before applying step 4 of Algorithm 1, i.e. before removing the nodes in $C$ and let $G_1$ be the graph generated after step 2 of the algorithm.

The walk $\pi$ is obviously $c$-connecting given $C \cup C_1$ in $G_2$. Suppose that a line $kl$ on $\pi$ was an arrow from $k$ to $l$ before step 3 of the algorithm. This implies that $k \in \text{ant}(C)$ in $G$ and the section $\rho_0$ containing the $kl$ line in $G_2$ is a non-collider (even if there are other lines replaced by arrows on $\rho_0$). Hence, all nodes of $\rho_0$ are outside $C \cup C_1$. In addition, these are all in
ant($C$) in $G_1$ since by step 2 of the algorithm no lines or arrows are removed or replaced. Therefore, regardless of whether $\rho_0$ is a collider or a non-collider, by Lemma 1, there is a $c$-connecting walk between the endpoints of $\rho$ in $G_1$. By the same argument for all such $kl$ arrows, $\pi$ is $c$-connecting given $C \cup C_1$ in $G_1$.

For every line $sr$ on $\pi$ in $G_1$ generated from a collider trislide $s \rightarrow \rho_1 \leftarrow r$, by Lemma 1, it is clear that there is a $c$-connecting walk $\pi_0$ given $C \cup C_1$ between $s$ and $r$, which is endpoint-identical to the $sr$ line. Hence by replacing $sr$ by $\pi_0$ we obtain a $c$-connecting walk given $C \cup C_1$. By induction, the result follows.

We, therefore, have the following immediate corollary:

**Corollary 1.** The class of chain graphs, $\mathcal{CG}$, with the LWF Markov property is stable under conditioning.

5. **Stability of CGs under marginalization.** Similar to the conditioning case, for a subset $M$ of $V$, the independence model after marginalization over $M$, denoted by $\alpha(J; M, \emptyset)$, is defined by

$$\alpha(J; M, \emptyset) = \{(A, B | D) \in J : (A \cup B \cup D) \cap M = \emptyset\}.$$ 

One can observe that $\alpha(J; M, \emptyset)$ is an independence model over $V \setminus M$.

The definition of stability under marginalization is defined similarly to the conditioning case: For a family of graphs $\mathcal{T}$, if, for every graph $G = (V, E) \in \mathcal{T}$ and every disjoint subsets $C$ of $V$, there is a graph $H \in \mathcal{T}$ such that $J(H) = \alpha(J(G); M, \emptyset)$ then $\mathcal{T}$ is stable under marginalization. We see again that the node set of $H$ is $N = V \setminus M$.

Here we see that CGs are not closed under marginalization. For example, it can be shown that $G$ in Fig. 4 is a CG (in fact a DAG) whose induced marginal independence model cannot be represented by a CG. We leave the details as an exercise to the reader.

**Fig 4.** (a) A chain graph $G$, by which it can be shown that the class of CGs is not stable under marginalization. ($\emptyset \not\in M.$)

Hence, we define the class of chain mixed graphs (CMGs) to be the class of mixed graphs without semi-direction-preserving cycles with at least an arrow. Notice that we allow CMGs to have multiple edges consisting of arcs.
and arrows and arcs and lines. This is a generalization of chain graphs since if a CMG does not contain arcs then it is a chain graph.

For example, in Fig. 5(a) the graph is a CMG, but in Fig. 5(b) the graph is not a CMG because of the existence of the $\langle h, p, q \rangle$ semi-direction-preserving cycle.

![Fig 5](a) A CMG. (b) A mixed graph that is not a CMG.

We provide a $c$-separation criterion for CMGs, and using this, show that CMGs are closed under marginalization and contain chain graphs. For this purpose, we provide in this section an algorithm that, from a CMG (or a chain graph) $G$ and after marginalization over $M$, generates a CMG with the corresponding independence model after marginalization over $M$. We define a $c$-separation criterion for CMGs with exactly the same wordings as that of CGs: A walk $\pi$ in a CG is a $c$-connecting walk given $C$ if every collider section of $\pi$ has a node in $C$ and all non-collider sections are outside $C$. We say that $A$ and $B$ are $c$-separated given $C$ if there are no $c$-connecting walks between $A$ and $B$ given $C$, and we use the notation $A \perp_c B \mid C$.

However, notice that this is in fact a generalization of the $c$-separation criterion for CGs since, for CMGs, bidirected edges on $\pi$ may make a section collider.

We now provide an algorithm that, from a chain mixed graph $G$ and after marginalization over $M$, generates a CMG with the corresponding independence model after marginalization over $M$. Notice that this algorithm may indeed apply to a CG.

**Algorithm 2.** $\alpha_{CMG}(G; M, \emptyset)$: (Generating a CMG from a chain mixed graph $G$ after marginalization over $M$)

Start from $G$.

1. Generate an $ij$ edge as in Table 1, steps 8 and 9, between $i$ and $j$ on a collider trislide with an endpoint $i$ and an endpoint in $M$ if the edge of the same type does not already exist.
2. Generate an appropriate edge as in Table 1, steps 1 to 7, between the endpoints of every tripath with inner node in $M$ if the edge of the same type does not already exist. Apply this step until no other edge can be generated.

3. Remove all nodes in $M$.

|   |   |   |
|---|---|---|
| 1 | $i \leftarrow m \rightarrow j$ | generates | $i \leftarrow j$ |
| 2 | $i \leftarrow m \rightarrow j$ | generates | $i \leftarrow j$ |
| 3 | $i \rightarrow m \rightarrow j$ | generates | $i \rightarrow j$ |
| 4 | $i \leftarrow m \rightarrow j$ | generates | $i \leftarrow j$ |
| 5 | $i \leftarrow m \rightarrow j$ | generates | $i \leftarrow j$ |
| 6 | $i \rightarrow m \rightarrow j$ | generates | $i \rightarrow j$ |
| 7 | $i \leftarrow m \rightarrow j$ | generates | $i \leftarrow j$ |
| 8 | $m \rightarrow i \cdots \circ \rightarrow j$ | generates | $i \leftarrow j$ |
| 9 | $m \rightarrow i \cdots \circ \rightarrow j$ | generates | $i \leftarrow j$ |

Notice that the first seven cases, except cases 3 and 6, generate an endpoint-identical edge to the tripath.

Fig. 6 illustrates how to apply Algorithm 2 step by step to a CG. We consider Algorithm 2 a function denoted by $\alpha_{CMG}$. Notice that for every chain mixed graph $G$, it holds that $\alpha_{CMG}(G; \emptyset, \emptyset) = G$. We first show that $\alpha_{CMG}(G; M, \emptyset)$ is a CMG:

**Proposition 3.** Graphs generated by Algorithm 2 are CMGs.

**Proof.** The resulting graphs have obviously the three desired types of edges, thus it is enough to prove that there is no semi-direction-preserving cycle that contains an arrow in the graph. Suppose, for contradiction, that there exists such a cycle. It is easy to observe that by replacing a generated line or arrow with the generating tripaths (cases 1, 2, 6, and 7 of Table 1) or trislide (case 8), a semi-direction-preserving path remains semi-direction-preserving. Therefore, it is implied inductively that there is a semi-direction-preserving path in the original chain graph. This also contains an arrow.
since an arrow can be only replaced by a tripath or a trislide that contains an arrow. This is a contradiction.

We first provide a lemma that expresses the global behaviour of step 2 of Algorithm 2:

**Lemma 2.** Let $G$ be a CMG. There exists an edge between $i$ and $j$ in $\alpha_{CMG}(G; M, \emptyset)$ if and only if there exists a walk $\pi$ between $i$ and $j$ in the graph generated after applying step 1 of Algorithm 2 to $G$ whose inner sections are all non-collider and whose inner nodes are all in $M$. In addition, if $\pi$ consists of more than one edge then the type of the edge $ij$ determines the existence of arrowheads pointing to $i$ and $j$ on $\pi$ in accordance to Table 1, where $m$ is replaced by the subwalk of $\pi$ that excludes $i$ and $j$.

**Proof.** ($\Leftarrow$) Suppose that there exists a walk $\pi$ between $i$ and $j$ in the graph generated after applying step 1 of Algorithm 2 to $G$ whose inner sections are all non-collider and whose inner nodes are all in $M$. By Algorithm 2, for a section between $k$ and $l$, a line between $k$ and $l$ is generated, and then, for a tripath $(h, q, r)$ consisting of a line $hq$ with $q \in M$, the same edge as $qr$ is generated. Therefore, a path is generated between $i$ and $j$ whose inner nodes are in $M$, and on which there may be only lines adjacent to $i$ and $j$ and every section is a non-collider node. We first apply steps 1, 4, and 5 of Table 1, and obtain endpoint identical edges, and then apply the rest.
of the steps of the table, and obtain an edge whose type is in accordance to the table.

(⇒) Suppose that there is an edge between $i$ and $j$ in $\alpha_{CMG}(G; M, \emptyset)$. If this edge has been generated by step 2 of Algorithm 2 then it has been generated by one of the tripaths in Table 1 in an iteration of step 2. Each edge in the tripath may have now been generated by a tripath with the inner node in $M$. By an inductive argument, we imply that in the graph generated after applying step 1 of Algorithm 2 to $G$, there is a walk $\pi$ (because of possible self-intersections) between $i$ and $j$ whose inner nodes are in $M$. We show that there is no collider section on $\pi$: If, for contradiction, there is a collider section $\rho$ with endpoints $\langle k, \rho, l \rangle$ then it is easy to observe that, in some iteration of the algorithm, we obtain a collider tripath with endpoints $k$ and $l$, but no edge can be generated between $k$ and $l$ by the algorithm. Hence, there is no edge between $i$ and $j$ in $\alpha_{CMG}(G; M, \emptyset)$, a contradiction. The existence of arrowheads pointing to the endpoints of $\pi$ are in accordance to the table, since otherwise, by the same argument as the other direction of the proof, we see that we obtain a different type of edge. \hfill \square

The following theorem shows that $\alpha_{CMG}(:, :, \emptyset)$ is well-defined:

**Theorem 3.** For a chain mixed graph $G$ and disjoint subsets $M$ and $M_1$ of its node set,

$$\alpha_{CMG}(\alpha_{CMG}(G; M, \emptyset); M_1, \emptyset) = \alpha_{CMG}(G; M \cup M_1, \emptyset).$$

**Proof.** (⇒) Suppose that in $\alpha_{CMG}(\alpha_{CMG}(G; M, \emptyset); M_1, \emptyset)$, there is an edge between $i$ and $j$. We prove that there is the same edge in $\alpha_{CMG}(G; M \cup M_1, \emptyset)$. By Lemma 2, there exists a walk $\pi$ between $i$ and $j$ in the graph generated after applying step 1 of Algorithm 2 to $\alpha_{CMG}(G; M, \emptyset)$ whose inner sections are all non-collider, inner nodes are all in $M_1$, and endpoints are in accordance to Table 1. By replacing arrows and arcs on $\pi$ by paths appearing in cases 8 and 9 of Table 1 (the replacements that have occurred in step 1 of Algorithm 2), only the sections become larger, wherein the added nodes to the sections are not in $M_1$. Denote the new walk by $\pi_1$. Notice that on $\pi_1$, the neighbors of every node in $M_1$ that are not in $M_1$ are children of $M_1$.

For every edge of $\pi_1$, again by Lemma 2, there exists a walk between its endpoints in the graph generated after applying step 1 of Algorithm 2 to $G$ with the same mentioned properties, but with inner nodes in $M$. Notice that by Table 1, it is clear that by replacing edges by walks at this stage, all nodes, which are non-colliders on $\pi_1$, remain non-collider on the new
walk that consists of all newly generated paths. In addition, nodes that are children of $M_1$ become children of $M \cup M_1$. Again by replacing arrows and arcs on by paths in cases 8 and 9 of Table 1, sections become larger, wherein the added nodes to the sections are not in $M_1$. Denote this new walk in $G$ by $\pi_2$. It can be seen that the neighbors of every node in $M \cup M_1$ that are not in $M \cup M_1$ are children of $M \cup M_1$.

By applying step 1 of Algorithm 2 to $G$ for $M \cup M_1$, all subpaths of $\pi_2$ of form $k \rightarrow \rho$ or $k \leftarrow \rho$ where $\rho = o \ldots l$ is the maximal subpath outside $M \cup M_1$ can be replaced by the $kl$ arrow or arc respectively (since $l$ is a child of $M \cup M_1$). Therefore, in the graph before applying step 2 of the algorithm to $G$ there is a walk whose inner sections are all non-collider, inner nodes are all in $M \cup M_1$, and endpoints are in accordance to Table 1. This implies, by Lemma 2 that there is the same edge in $\alpha_{CMG}(G; M \cup M_1, \emptyset)$.

$\Leftarrow$ Suppose there is an edge between $i$ and $j$ in $\alpha_{CMG}(G; M \cup M_1, \emptyset)$. By Lemma 2, there exists a walk $\pi$ between $i$ and $j$ in the graph generated after applying step 1 of Algorithm 2 to $G$ whose inner sections are all non-collider, inner nodes are all in $M \cup M_1$, and endpoints are in accordance to Table 1. By replacing arrows and arcs on $\pi$ by paths appearing in cases 8 and 9 of Table 1 (the replacements that have occurred in step 1 of Algorithm 2), in $G$ only the sections become larger, wherein the added nodes to the sections are not in $M \cup M_1$. Denote the new walk by $\pi_1$. Notice that on $\pi_1$, the neighbors of every node in $M \cup M_1$ that are not in $M \cup M_1$ are children of $M \cup M_1$.

By applying step 1 of Algorithm 2 to $G$ for $M$, all subpaths of $\pi_1$ of form $k \rightarrow \rho$ or $k \leftarrow \rho$ where $\rho = o \ldots l$ is the maximal subpath where $l$ is a child of $M$ can be replaced by $kl$ arrows or lines respectively. Now the generated walk can be partitioned into subwalks with endpoints in $M_1$ or outside $M \cup M_1$ and all inner nodes in $M$ (there might be single edges in the partition). All these subwalks with lengths more than two satisfy the conditions of Lemma 2 for $M$. Hence, in $\alpha_{CMG}(G; M, \emptyset)$, there exist edges between the endpoints of the subwalks in accordance to Table 1. These edges form a walk, which is denoted by $\pi'$. Since there are no collider sections in $\pi$, it is easy to observe that, based on Table 1, there are no collider sections on $\pi'$. In addition, on $\pi'$, neighbors of every node in $M_1$ that are not in $M_1$ are children of $M_1$. Therefore, again by applying step 1 of the algorithm for $M_1$ and then applying Lemma 2, we obtain the same edge in $\alpha_{CMG}(\alpha_{CMG}(G; M, \emptyset); M_1, \emptyset)$.

The following result shows that all CMGs may be generated after marginalization for CGs. Denote by $CMG$ the set of all CMGs.
Proposition 4. The map $\alpha_{CMG}(\cdot;\cdot,\emptyset): CG \to CMG$ is surjective.

Proof. Consider an arbitrary chain mixed graph $H$. Define a chain graph $G$ from $H$ as follows: Keep all arrows and lines of $H$ in $G$ and replace arcs $ij$ with $i \xleftarrow{m} j$; and define a subset $M$ of the node set of $G$ as the set of all such $m$.

We first prove that $G$ is a CG. It only contains the two desired types of edges. In addition, it does not contain semi-direction-preserving cycles that contains an arrow since if, for contradiction, it does then it must contain the tripath $i \xleftarrow{m} j$, which is impossible.

In addition $\alpha_{CMG}(G;M,\emptyset) = H$. This is because the only type of tripath with inner node in $M$ is case 4 of Table 1; and these tripaths obviously turn into all arcs existing in $H$.

Here we prove the main result of this section:

Theorem 4. For a chain mixed graph $G$ and disjoint subsets $A$, $B$, $M$, and $C_1$ of its node set,

$$\langle A,B|C_1 \rangle \in J_c(\alpha_{CMG}(G;M,\emptyset)) \iff \langle A,B|C_1 \rangle \in J_c(G).$$

Proof. We need to prove that $A \perp_{c} B|C_1$ in $G \iff A \perp_{c} B|C_1$ in $\alpha_{CMG}(G;M,\emptyset)$.

($\Rightarrow$) Suppose that there is a $c$-connecting walk $\pi$ given $C_1$ between $i$ and $j$ in $G$. For every walk of form $i \xleftarrow{m} i \cdot \cdot \cdot \circ \xleftarrow{m} j$ or $i \xleftarrow{m} \cdot \cdot \cdot \circ \xleftarrow{m} j$ with an inner node in $C_1$, after applying step 1 of Algorithm 2, there is a generated $ij$ edge. We replace all these walks with the generated edge and call the resulting walk $\pi'$. It is easy to observe that $\pi'$ is $c$-connecting. Now consider all maximal subwalks of $\pi'$ whose inner sections are all non-collider, endpoints are not in $M$, and inner nodes are all in $M$. Notice that all nodes of $\pi'$ that are in $M$ are included in these subwalks since no collider section on $\pi'$ has all nodes in $M$. By Lemma 2, instead of these subwalks, there are edges in $\alpha_{CMG}(G;M,\emptyset)$ with endpoints in accordance to Table 1. By replacing all the subpaths with these edges, we obtain a walk $\pi''$ in $\alpha_{CMG}(G;M,\emptyset)$. The walk $\pi'$ is $c$-connecting given $C_1$ since every node that is an inner node of a collider or a non-collider section in $\pi'$ is an inner node of a collider or a non-collider section in $\pi'$. This again can be verified by looking into the first 7 cases of Table 1 and the fact that there is no collider section $\rho$ for which $m \xrightarrow{\rho}$.

($\Leftarrow$) Suppose that there is a $c$-connecting walk $\pi$ given $C_1$ between $i$ and $j$ in $\alpha_{CMG}(G;M,\emptyset)$. By Lemma 2, for every edge $kl$ on $\pi$, there is a walk $\pi'$
between \( k \) and \( l \) in the graph after applying step 1 of Algorithm 2 to \( G \) with the stated properties in the lemma. By replacing every edge on \( \pi \) by such \( \pi' \) in \( G \), we obtain a walk \( \pi'' \). We prove that \( \pi'' \) is \( c \)-connecting given \( C_1 \): Notice that \( \pi' \) is obviously \( c \)-connecting. Hence, it is enough to prove that, for a replaced edge \( kl \), if \( l \) is an inner node of a collider or a non-collider section, after the replacement, it remains an inner node of a collider or non-collider section respectively. This again can be verified by looking into the first 7 cases of Table 1.

Now by replacing a \( uv \) edge on \( \pi'' \) by a path in \( G \) by step 1 of the algorithm, where \( u \) is a child of \( M \), we obtain a larger walk. It is easy to observe that if \( u \) is on a collider section or a non-collider section \( \rho \) on \( \pi'' \) then it remains on a collider section or a non-collider section respectively. If \( \rho \) is non-collider, and since all inner nodes of the new path are outside \( C_1 \) we obviously obtain a \( c \)-connecting walk. The collider case is obvious since a node on \( \rho \) is in \( C \). By an inductive argument for all such \( uv \) edges, we obtain the result.

We, therefore, have the following immediate corollary:

**Corollary 2.** The class of chain mixed graphs, \( \text{CMG} \), with \( c \)-separation criterion is stable under marginalization.

### 6. Stability of CMGs under marginalization and conditioning.

**6.1. Stability of CMGs under conditioning.** In the previous section, we showed that the class of CMGs is stable under marginalization. In this section, we first show that the class of CMGs is also stable under conditioning, and provide a generalization of Algorithm 1 for conditioning for CMGs:

**Algorithm 3.** \( \alpha_{\text{CMG}}(G; \emptyset, C) \): (Generating a CMG from a chain mixed graph \( G \) after conditioning on \( C \))

\( \text{Start from } G. \)

1. Find all nodes in \( C \cup \text{ant}(C) \) and call this set \( S \).
2. For collider trislides illustrated in Table 2, steps 4 and 5, with an endpoint \( i \) and one endpoint in \( S \), generate an \( ij \) edge following the table if the edge does not already exist;
3. For collider trislides (including tripaths) illustrated in Table 2, steps 1-3, with at least one inner node in \( S \), generate an edge following the table; i.e. generate an endpoint-identical edge to the trislide between the endpoints of the trislide, if the edge does not already exist. Apply
this step repeatedly until no other edge can be generated, but do not use generated lines (to generate new sections).

4. Remove the arrowheads of all arrows and arcs pointing to members of $S$ repeatedly (i.e. turn such arrows into lines and such arcs into arrows).

5. Remove all nodes in $C$.

Notice that if an inner node of a trislide is in $S$ then all the inner nodes are in $S$.

Table 2

| Table 2 | Types of edges induced by trislides with inner node in $s \in S = C \cup \text{ant}(C)$. |
|---------|------------------------------------------------------------------|
| 1       | $i \leadsto s \cdots \leadsto j$ generates $i \leftarrow j$       |
| 2       | $i \rightleftharpoons s \cdots \leadsto j$ generates $i \leftarrow j$ |
| 3       | $i \rightleftharpoons s \cdots \leq j$ generates $i \leftarrow j$ |
| 4       | $s \rightleftharpoons i \cdots \rightleftharpoons j$ generates $i \leftarrow j$ |
| 5       | $s \rightleftharpoons i \cdots \rightleftharpoons j$ generates $i \leftarrow j$ |

Fig. 7 illustrates how to apply Algorithm 3 step by step to a CMG. We first follow the same procedure as in the previous section.

**Proposition 5.** Graphs generated by Algorithm 3 are CMGs.

**Proof.** Graphs generated by Algorithm 3 have the three desired types of edges. We prove that there is no semi-direction-preserving cycle with an arrow in a chain mixed graph $G$. Suppose, for contradiction, that a generated graph does contain a semi-direction-preserving cycle $\pi$ with an arrow. If a line $kl$ on $\pi$ has been generated by step 3 of the algorithm and case 1 of Table 2 or an arrow from $k$ to $l$ on $\pi$ has been generated by case 2 then it is easy to see that in the previous iteration of the algorithm $k \in S$. Inductively, this implies that $k \in S$ in $G$ unless possibly an arrow on the semi-direction-preserving path in an iteration has been generated by step 4. It is also easy to observe that in this case, $k \in \text{pa}(S)$ implies that $k \in \text{ant}(S)$ in the previous
Fig 7. (a) A chain mixed graph $G$, $\square \in C$. (b) The graph after applying step 1 of Algorithm 3, $\square \in S = C \cup \text{ant}(C)$. (c) The generated graph after applying step 2 (step 5 of Table 2). (d) The generated graph after applying step 3 (steps 2 and 3 of Table 2). (e) The generated graph after applying step 4. (f) The generated CMG from $G$.

iteration. Inductively again, $k \in S$ in $G$. This implies that there is no arrow on $\pi$, a contradiction.

If an arrow from $k$ to $l$ on $\pi$ has been generated by case 4 of the table then $k \in \text{ant}(l)$ in $G$ with an arrow existing in the direction-preserving path from $k$ to $l$. Applying this to all edges of $\pi$ implies that there is a semi-direction-preserving cycle with an arrow in $G$, a contradiction. If a line $ij$ or an arrow from $i$ to $j$ on $\pi$ has been generated by step 4 then $i \in S$ in $G$ and again, a contradiction.

**Lemma 3.** Let $G$ be a CMG. There exists an edge between $i$ and $j$ in the graph generated after step 3 of Algorithm 3 if and only if there exists an endpoint identical walk $\pi$ to the edge between $i$ and $j$ in the generated graph after step 2 whose inner sections are all collider and whose inner nodes are all in $C \cup \text{ant}(C)$.

**Proof.** ($\Leftarrow$) Suppose that there exists a walk $\pi$ between $i$ and $j$ in the generated graph after step 2 whose inner sections are all collider and whose
inner nodes are all in $C \cup \text{ant}(C)$. The walk $\pi$ can be partitioned into sections. For a section $\rho$ such that $\pi' = \langle kpl \rangle$, if $\rho$ is a path then by step 3 of the algorithm, an endpoint identical edge $kl$ is generated. If $\rho$ is not a path then consider the shortest subpath $\pi''$ of $\pi'$. This contains the edges containing $k$ and $l$ that are not lines, and all its other edges are lines; hence, $\pi''$ is a collider trislide. It is clear that all its inner nodes are also in $C \cup \text{ant}(C)$. Therefore, again by step 3, an endpoint-identical edge $kl$ is generated. Notice that $kl$ is either an arrow or an arc unless $k$ and $l$ are endpoints of $\pi$ since all sections of $\pi$ are collider. Therefore, the algorithm is applied to all such generated $kl$ edges. The fact that all sections of $\pi$ are collider also implies that the generated walk after replacing all such $kl$ edges contains only collider sections. Now, by an inductive argument, we obtain the result.

$(\Rightarrow)$ Suppose that there is an edge between $i$ and $j$ in the graph generated after step 3 of Algorithm 3. If this edge has been generated by step 3 of Algorithm 3 then it has been generated by one of the first three trislides in Table 2 in an iteration of step 3 of the algorithm. Each arrow or arc in the trislide may now have been generated by a trislide with inner nodes in $C \cup \text{ant}(C)$ (since no generated line can be used in the iterations). Since the trislides are endpoint-identical to the generated edge, it is implied that all sections remain collider. By an inductive argument, we imply that, in the graph after applying step 2 of the algorithm, there is a walk $\pi$ between $i$ and $j$ whose inner nodes are in $C \cup \text{ant}(C)$ and all sections are collider. The walk $\pi$ is also endpoint identical to the edge $ij$ since otherwise, by the same argument as in the other direction of the proof, we see that we obtain a different type of edge.

\[ \begin{align*}
\text{Theorem 5.} \quad &\text{For a chain mixed graph } G \text{ and disjoint subsets } C \text{ and } C_1 \text{ of its node set,} \\
\alpha_{\text{CMG}}(\alpha_{\text{CMG}}(G; \emptyset, C); \emptyset, C_1) &= \alpha_{\text{CMG}}(G; \emptyset, C \cup C_1). 
\end{align*} \]

\textbf{Proof.} Let $H = \alpha_{\text{CMG}}(G; \emptyset, C)$ and denote by $H_0$ the graph after applying steps 1 and 2 of Algorithm 3 to $H$, and by $H_1$ the graph after applying step 3 of Algorithm 3 for $C_1$ to $H$. In addition, denote by $G_0$ the graph generated after applying steps 1 and 2 of Algorithm 3 to $G$ for $C$, and by $G_1$ the graph generated after applying step 3 of Algorithm 3. Denote also by $K_0$ the graph generated after steps 1 and 2 of Algorithm 3 to $G$ for $C \cup C_1$ and by $K_1$ this graph generated after step 3 of the algorithm. In summary we have that

\[ \begin{align*}
\alpha_{\text{CMG}}(\alpha_{\text{CMG}}(G; \emptyset, C); \emptyset, C_1) &\leftarrow H_1 \leftarrow H_0 \leftarrow H \leftarrow G_1 \leftarrow G_0 \leftarrow G.
\end{align*} \]
\[ G \to K_0 \to K_1 \to \alpha_{\text{CMG}}(G; \emptyset, C \cup C_1). \]

We first prove that there is an \( ij \) edge in \( \alpha_{\text{CMG}}(\alpha_{\text{CMG}}(G; \emptyset, C); \emptyset, C_1) \) if and only if there is an \( ij \) edge in \( \alpha_{\text{CMG}}(G; \emptyset, C \cup C_1) \):

(\( \Rightarrow \)) Suppose that in \( \alpha_{\text{CMG}}(\alpha_{\text{CMG}}(G; \emptyset, C); \emptyset, C_1) \) there is an edge between \( i \) and \( j \). It is clear that there is an \( ij \) edge in \( H_1 \). Now, by Lemma 3, in \( H_0 \) there exists an endpoint-identical walk \( \pi \) between \( i \) and \( j \) to the edge \( ij \) in \( H_1 \) whose inner sections are all collider and inner nodes are all in \( C_1 \cup \text{ant}(C_1) \). Notice that this means that all edges on \( \pi \) are either lines or arcs except possibly those containing \( i \) and \( j \). By replacing arcs or arrows on \( \pi \) by paths \( \pi_0 \) with an endpoint \( k \) that is a spouse of a member of \( C_1 \cup \text{ant}(C_1) \), based on cases 4 and 5 of Table 2, the generated walk \( \pi' \) in \( H \) satisfies all the mentioned properties for \( \pi \) since \( \pi_0 \) remains in a collider section and \( k \in C_1 \cup \text{ant}(C_1) \) for \( k \neq i, j \). In addition, the endpoints of \( \pi' \) are not in \( \text{ant}(C) \) in \( G \) since otherwise the arrowheads of arcs would be removed by step 4 of the algorithm and \( \pi \) would contain non-collider sections.

For every line \( kl \) of \( \pi' \) in \( G_1 \), the line also exists in \( G \) since otherwise \( k \) and \( l \) are in \( \text{ant}(C) \), and the section containing \( kl \) in \( G_1 \) cannot remain collider. For every arc \( kl \) of \( \pi' \) in \( G_1 \), by Lemma 3, there exists an endpoint-identical walk to an arc between \( k \) and \( l \) in \( G_0 \) with the same mentioned properties, but with inner nodes in \( C \cup \text{ant}(C) \). It is clear that by replacing arcs by these walks this stage in order to obtain a larger walk \( \pi'' \), all sections on \( \pi'' \) remain collider. Therefore, in \( G_0 \), \( \pi'' \) is a walk whose inner sections are all collider, inner nodes are all in \( C \cup C_1 \cup \text{ant}(C \cup C_1) \). By the same argument as above by replacing an edge on \( \pi'' \) by cases 4 and 5 of Table 2 for \( C \), we obtain a walk \( \pi''' \) with the same properties as that of \( \pi'' \) in \( G \). Clearly by replacing paths in steps 4 and 5 of Table 2 for \( C \cup C_1 \) by arcs and arrows, we obtain a walk in \( K_0 \) with the same properties since collider sections on the walk remain collider. Therefore, by Lemma 3, there is an edge between \( i \) and \( j \) that is endpoint-identical to \( \pi''' \) in \( K_1 \).

(\( \Leftarrow \)) Suppose there is an edge between \( i \) and \( j \) in \( \alpha_{\text{CMG}}(G; \emptyset, C \cup C_1) \). Thus, there is an \( ij \) edge in \( K_1 \). By Lemma 3, there exists an endpoint-identical walk \( \pi \) between \( i \) and \( j \) in \( K_0 \) whose inner sections are all collider and inner nodes are all in \( C \cup C_1 \cup \text{ant}(C \cup C_1) \). By replacing arrows and arcs by the paths in steps 4 and 5 of Table 2, the generated walk \( \pi' \) still satisfies the properties for \( \pi \). Consider all subwalks on \( \pi' \) that consist of connected collider sections with endpoints in \( C_1 \cup \text{ant}(C_1) \setminus (C \cup \text{ant}(C)) \) and all inner nodes in \( C \cup \text{ant}(C) \). (Notice that if an inner node of a section is in \( C \cup \text{ant}(C) \) then so are all the inner nodes.) All these subwalks satisfy the conditions of Lemma 3 for \( C \) after applying step 2 of the algorithm for \( C \) by what we have shown in the other direction of the proof. Therefore,
in $H$, there exist edges between the endpoints of the subwalks. These edges are all arcs except possibly those including $i$ and $j$, and form a walk, which we call $\pi''$. Hence, there are no non-collider sections on $\pi''$. In addition, all its inner nodes are in $C_1 \cup \text{ant}(C_1)$. Since in $G$, none of these inner nodes is in $C \cup \text{ant}(C)$, after applying step 4 of the algorithm, the edges on $\pi'$ do not change. Therefore, again there is an endpoint-identical edge to $\pi''$ in $\alpha_{CMG}(\alpha_{CMG}(G; \emptyset, C); \emptyset, C_1)$.

We now prove that the $ij$ edge is the same in both graphs:

$(\Rightarrow)$ Suppose that there is no arrowhead at $i$ in $\alpha_{CMG}(\alpha_{CMG}(G; \emptyset, C); \emptyset, C_1)$. There is no arrowhead at $i$ on $ij$ in $H_1$ unless possibly $i \in \text{ant}(C_1)$ in $H$. If there is no arrowhead at $i$ on $ij$ in $H_1$ (and consequently $H_2$ and $H$) then there is no arrowhead at $i$ on the walk in $G_1$ unless possibly $i \in \text{ant}(C)$. Here we show that if $i$ is in $\text{ant}(C_1)$ in $H$ then it is also in $\text{ant}(C \cup C_1)$ in $G$:

For every line or arrow $kl$ (from $k$ to $l$) on the semi-direction-preserving path from $i$ to a member of $C_1$, either this line exists in $G$ or, (1) an arrowhead is removed by step 4 of the algorithm, and $i \in \text{ant}(C)$; (2) the edge has been generated by Lemma 3, and hence $k \in \text{ant}(C)$; (3) the edge has been generated by step 2 of the algorithm, and hence $k \in \text{ant}(l)$. By induction for nodes from $i$ to the member of $C_1$, we conclude that $i$ is either in $\text{ant}(C)$ or in $\text{ant}(C_1)$. Therefore, in $G$ there is an arrowhead at $i$ on $\pi'$ or one of the following occurs: either $i \in \text{ant}(C_1)$ in $H$, which implies $i \in \text{ant}(C \cup C_1)$ in $G$ or $i \in \text{ant}(C)$ in $G$. Either case implies that there is no arrowhead at $i$ on $ij$ in $\alpha_{CMG}(G; \emptyset, C \cup C_1)$.

$(\Leftarrow)$ If there is no arrowhead at $i$ on $ij$ in $\alpha_{CMG}(G; \emptyset, C \cup C_1)$ then either there is no arrowhead at $i$ on $ij$ or $i \in \text{ant}(C \cup C_1)$ in $G$. If there is no arrowhead at $i$ on $ij$ in $G$ then it is easy to show that there is no arrowhead at $i$ on $ij$ in $\alpha_{CMG}(\alpha_{CMG}(G; \emptyset, C); \emptyset, C_1)$. If $i \in \text{ant}(C)$ then the arrowhead is removed at $i$ in $H$. If $i \in \text{ant}(C_1) \setminus \text{ant}(C)$ in $G$ then $i \in \text{ant}(C_1)$ in $H$, which implies that the arrowhead at $i$ is removed in $\alpha_{CMG}(\alpha_{CMG}(G; \emptyset, C); \emptyset, C_1)$.

\textbf{Theorem 6.} For a chain mixed graph $G$ and disjoint subsets $A$, $B$, $C$, and $C_1$ of its node set,

\[ \langle A, B \mid C_1 \rangle \in J_e(\alpha_{CMG}(G; \emptyset, C)) \iff \langle A, B \mid C \cup C_1 \rangle \in J_e(G). \]

\textbf{Proof.} We prove that $A \perp_C B \mid C \cup C_1$ in $G$ if and only if $A \perp_C B \mid C_1$ in $\alpha_{CMG}(G; \emptyset, C)$.

$(\Rightarrow)$ Suppose that there is a $c$-connecting walk given $C \cup C_1$ between $i$ and $j$ in $G$. After applying step 2 of Algorithm 3, we obviously obtain another $c$-connecting walk given $C \cup C_1$ between $i$ and $j$. By Lemma 1, there is an
alternative $c$-connecting walk $\pi$, where all sections are paths and inner nodes of collider sections are in $\mathcal{C} \cup \mathcal{C}_1 \cup (\text{ant}(\mathcal{C}) \cup \text{ant}(\mathcal{C}_1))$. Consider all maximal subwalks of $\pi$ whose inner sections are all collider, endpoints are not in $\mathcal{C}$, and nodes of each section are in $\mathcal{C} \cup \text{ant}(\mathcal{C})$. In addition, all nodes of $\pi$ that are in $\mathcal{C}$ are included in these subwalks since no non-collider section on $\pi$ has a node in $\mathcal{C}$. By Lemma 3, instead of these subwalks, there are endpoint-identical edges after applying step 3 of Algorithm 3. By replacing all the subpaths with these edges, we obtain a walk $\pi'$. The walk $\pi'$ is $c$-connecting given $\mathcal{C}_1$ since generated edges on $\pi'$ are endpoint-identical to the subpaths on $\pi$ that have been replaced.

Now by step 4 of the algorithm, no collider sections turn into a non-collider one on $\pi'$ since if an arrowhead on a node $k$ is removed then $k \in \text{ant}(\mathcal{C})$ in $\mathcal{G}$ and so are all inner nodes of the section that contains $k$; hence, $k$ cannot be on $\pi'$ by how $\pi'$ is generated. Therefore, $\pi'$ is a $c$-connecting walk given $\mathcal{C}_1$ in $\alpha_{\text{CMG}}(\mathcal{G}; \emptyset, \mathcal{C})$

$\Leftrightarrow$ Suppose that there is a $c$-connecting walk given $\mathcal{C}_1$ between $i$ and $j$ in $\alpha_{\text{CMG}}(\mathcal{G}; \emptyset, \mathcal{C})$. By lemma 1, there is a $c$-connecting path $\pi$ whose sections are all paths and members of collider sections in $\mathcal{C}_1 \cup \text{ant}(\mathcal{C}_1)$. Before applying step 4 of Algorithm 3, a non-collider section $\tau$ on $\pi$ might be some adjacent collider sections if all its inner nodes are in $\text{ant}(\mathcal{C})$ in $\mathcal{G}$. Denote the semi-direction-preserving path between a node $h$ on $\tau$ and a member of $\mathcal{C}$ or a member of $\mathcal{C}_1$ by $\pi''$. Let $\mathcal{G}_0$ be the graph after applying step 2 on $\mathcal{G}$. By Lemma 3, for every edge $kl$ on $\tau$, there is an endpoint-identical walk $\pi'$ between $k$ and $l$ in $\mathcal{G}_0$ whose inner sections are all collider and inner nodes are all in $\mathcal{C} \cup \text{ant}(\mathcal{C})$. In the case that all inner nodes are in $\text{ant}(\mathcal{C}) \setminus \mathcal{C}$, denote the semi-direction-preserving path between a node $h'$ in $\tau$ and a member of $\mathcal{C}$ or a member of $\mathcal{C}_1$ by $\pi'''$.

We generate a walk $\pi_0$ in $\mathcal{G}_0$ as follows: (1) We first replace every $\tau$, as defined above, with $\pi' = (\tau_0, \pi'', \pi'''(r), \tau_1)$, where $\tau_0$ and $\tau_1$ are subwalks that partition $\tau$ and share $h$ and $\pi'''(r)$ is $\pi''$ in reverse direction; (2) we then replace every edge on $\pi$ with such $\pi'$ in the case that an inner node of $\pi'$ is in $\mathcal{C}$; (3) replace an edge on $\pi$ in the case that all inner nodes of $\pi'$ are in $\text{ant}(\mathcal{C}) \setminus \mathcal{C}$ with $\tau'' = (\pi'_0, \pi''', \pi'''(r), \pi'_1)$, where $\pi'_0$ and $\pi'_1$ are subwalks that partition $\pi'$ and share $h'$, and $\pi'''(r)$ is $\pi'''$ in reverse direction. We prove that $\pi_0$ is $c$-connecting given $\mathcal{C} \cup \mathcal{C}_1$: Notice that $\pi'$ is obviously $c$-connecting given $\mathcal{C} \cup \mathcal{C}_1$. The walk $\tau''$ is also $c$-connecting given $\mathcal{C} \cup \mathcal{C}_1$ since $\tau'$ consists of adjacent non-collider sections and only one collider section, and on $\tau''$, there is only one node in $\mathcal{C} \cup \mathcal{C}_1$, which is an inner node of the collider section. In addition, $\tau'$ is also $c$-connecting given $\mathcal{C} \cup \mathcal{C}_1$ in the same way. Finally, for a replaced edge $kl$, if $l$ is an inner node of a collider or a non-
collider section, after the replacement by the endpoint-identical $\pi'$ or $\tau''$, it remains an inner node of a collider or non-collider section respectively.

Now suppose that a $qr$ edge on $\pi_0$ has been generated by step 2 of Algorithm 3. If no inner node of $\varsigma_1 = r \cdots \circ \leftarrow q$ or of $\varsigma_2 = r \cdots \circ \leftrightarrow q$ (which are displayed in cases 4 and 5 of Table 2) is in $C_1$ then the replacement of the $qr$ edge with stated paths provides a $c$-connecting walk given $C \cup C_1$ in $G$. If there is an inner node of $\varsigma_1$ or $\varsigma_2$ in $C_1$ then by replacing the $qr$ edge with $r \leftarrow s \leftarrow \varsigma_1$ or $r \leftarrow s \leftrightarrow \varsigma_1$, where $s \in C \cup \text{ant}(C)$, we obtain a $c$-connecting walk given $C \cup C_1$ in $G$ (by using Lemma 1). This completes the proof.

\textbf{Corollary 3.} The class of chain mixed graphs, $\mathcal{CMG}$, with $c$-separation criterion is stable under conditioning.

6.2. Combination of marginalization and conditioning for CMGs. Corollaries 5 and 3 imply that $\mathcal{CMG}$ with $c$-separation criterion is stable under marginalization and conditioning, which formally holds when there is a graph $H \in \mathcal{CMG}$ such that $J_c(H) = \alpha(J_c(G); M, C)$, where

\[ \alpha(J; M, C) = \{ \langle A, B \mid D \rangle : \langle A, B \mid D \cup C \rangle \in J \text{ and } (A \cup B \cup D) \cap (M \cup C) = \emptyset \}. \]

We now deal with the case where there are both marginalization and conditioning subsets in a CMG. We first have the following important result, which illustrates that in order to both marginalize and condition, it does not matter whether we marginalize first by using Algorithm 2 and then condition by using Algorithm 3 or vice versa:

\textbf{Proposition 6.} For a chain mixed graph $G$ and two disjoint subsets $M$ and $C$ of its node set, it holds that

\[ \alpha_{\mathcal{CMG}}(\alpha_{\mathcal{CMG}}(G; M, \emptyset); \emptyset, C) = \alpha_{\mathcal{CMG}}(\alpha_{\mathcal{CMG}}(G; \emptyset, C); M, \emptyset). \]

\textbf{Proof.} We first prove that there is an $ij$ edge in $\alpha_{\mathcal{CMG}}(\alpha_{\mathcal{CMG}}(G; M, \emptyset); \emptyset, C)$ if and only if there is an $ij$ edge in $\alpha_{\mathcal{CMG}}(\alpha_{\mathcal{CMG}}(G; \emptyset, C); M, \emptyset)$. We then prove that these edges are of the same type.

$(\Rightarrow)$ Suppose that in $\alpha_{\mathcal{CMG}}(\alpha_{\mathcal{CMG}}(G; M, \emptyset); \emptyset, C)$ there is an edge between $i$ and $j$. There is also an $ij$ edge before applying step 4 of Algorithm 3. By lemma 3, there exists an endpoint-identical walk between $i$ and $j$ in the graph generated after applying step 2 of Algorithm 3 to $\alpha_{\mathcal{CMG}}(G; M, \emptyset)$ whose inner sections are all collider and inner nodes are all in $C \cup \text{ant}(C)$. It is easy to see that an endpoint-identical walk $\pi$ with the same properties exists in $\alpha_{\mathcal{CMG}}(G; M, \emptyset)$. For every edge $kl$ on $\pi$, by Lemma 2, there exists
a walk \( \pi' \) between \( k \) and \( l \) in \( G \) whose inner sections are all non-collider and inner nodes are all in \( M \). We denote the walk in \( G \) that consists of all such adjacent \( \pi' \) by \( \pi_0 \). It is easy to observe that all collider sections have nodes in \( C \cup \text{ant}(C) \) in \( G \). By considering the maximal subwalks of \( \pi_0 \) that contain only collider sections and in which all nodes are in \( C \cup \text{ant}(C) \), but endpoints are outside \( C \), and by applying step 2 of Algorithm 3 and using Lemma 3, we obtain a walk \( \pi_1 \) that only contains non-collider sections with inner nodes in \( M \). By replacing arrows and arcs on \( \pi \) by paths appearing in cases 8 and 9 of Table 1 (the replacements that have occurred in step 1 of Algorithm 2), in \( \alpha_{CMG}(G; \emptyset, C) \) only the sections become larger, wherein the added nodes to the sections are not in \( M \). Denote the new walk by \( \pi_0 \). Notice that on \( \pi_0 \), the neighbors of every node in \( M \) that are not in \( M \) are children of \( M \).

For every edge \( kl \) on \( \pi_0 \), by Lemma 3, there exists an endpoint-identical walk \( \pi' \) between \( k \) and \( l \) in the graph generated after applying step 2 of Algorithm 3 to \( \alpha_{CMG}(G; \emptyset, C) \) whose inner sections are all collider and inner nodes are all in \( C \cup \text{ant}(C) \). We denote the walk in this graph that consists of all such adjacent \( \pi' \) of \( \pi_0 \) by \( \pi_1 \). It is easy to observe that all non-collider sections on \( \pi_1 \) have all inner nodes outside \( C \), and all collider sections have inner nodes in \( C \cup \text{ant}(C) \). In \( G \), there are possible paths appearing in cases 4 and 5 of Table 2 instead of some arrows and arcs on \( \pi_1 \). By replacing these, we denote the walk by \( \pi_2 \). It still holds that all non-collider sections on \( \pi_2 \) have all inner nodes outside \( C \), and all collider sections have inner nodes in \( C \cup \text{ant}(C) \). By considering a non-collider section \( \rho \) that is a path on \( \pi_2 \) for which all nodes are outside \( M \) and an endpoint \( k \) is child of \( M \), by step 1 of Algorithm 2, instead of \( \langle \rho, l \rangle \), we obtain a \( k \) \( l \) edge according to cases 8 and 9 of Table 1. Denote the generated walk by \( \pi_3 \). By considering the maximal subwalks of \( \pi_3 \) that contain only non-collider sections and in which all nodes are in \( M \), but endpoints are outside \( M \), and by using Lemma 2, we obtain a walk \( \pi_4 \) in \( \alpha_{CMG}(G; M, \emptyset) \) that contains collider sections with nodes in \( C \cup \text{ant}(C) \) and non-collider sections outside \( C \). In addition, all non-collider sections have an endpoint that is a spouse of \( C \cup \text{ant}(C) \). Therefore, by step 2 of Algorithm 3, we obtain a walk \( \pi_5 \) where all sections are collider. Notice that it is easy to prove by induction that if a node is in the anterior set of \( C \) in \( G \).
then it remains an anterior of \( C \) in \( \alpha_{\text{CMG}}(G; M, \varnothing) \) and also after applying step 2 on \( \alpha_{\text{CMG}}(G; M, \varnothing) \). Hence, all inner nodes of \( \pi_5 \) are in \( C \cup \text{ant}(C) \) in \( \alpha_{\text{CMG}}(G; M, \varnothing) \). By Lemma 3, we conclude that there is an \( ij \) edge in the graph generated by applying step 3 of Algorithm 3 to \( \alpha_{\text{CMG}}(G; M, \varnothing) \). Thus, there will also be an \( ij \) edge in \( \alpha_{\text{CMG}}(\alpha_{\text{CMG}}(G; \varnothing, C); M, \varnothing) \).

We now prove that the \( ij \) edge is of the same type in both graphs. For every graph generated by a step of the algorithm, we discussed a walk between \( i \) and \( j \) in both directions of the proof above. We focus on the arrowhead pointing to \( i \) on these walks:

\( (\Rightarrow) \) If in \( \alpha_{\text{CMG}}(\alpha_{\text{CMG}}(G; M, \varnothing); \varnothing, C) \), there is no arrowhead pointing to \( i \) on the \( ij \) edge then there is no arrowhead pointing to \( i \) before applying step 4 of Algorithm 3 or \( i \in \text{ant}(C) \) in \( \alpha_{\text{CMG}}(G; M, \varnothing) \). This implies that after applying step 2 of Algorithm 3, if there is an arrowhead at \( i \) then \( i \in \text{ant}(C) \) in \( \alpha_{\text{CMG}}(G; M, \varnothing) \). In \( \alpha_{\text{CMG}}(G; M, \varnothing) \), by looking at cases 4 and 5 of Table 2, the same statement holds. By looking at the first seven cases of Table 1 and by Lemma 2, it is clear that in the graph after applying step 1 of Algorithm 2 to \( G \), if there was no arrowhead at \( i \) then there is no arrowhead at \( i \) at this stage, and moreover, an anterior remains an anterior. This is also the case for cases 8 and 9 of Table 1 and hence, in \( G \), if there is an arrowhead at \( i \) then \( i \in \text{ant}(C) \).

This implies that there is no arrowhead at \( i \) in \( \alpha_{\text{CMG}}(G; \varnothing, C) \). If by steps 1 or 2 of Algorithm 2 an arrowhead at \( i \) is generated then at some iteration of the algorithm, one of the steps 3, 6, 8, or 9 has been used. Denote the inner node of the tripath in cases 3 and 6 or the other endpoint of the section containing \( i \) in the paths in cases 8 or 9 by \( q \). There is an arrowhead at \( q \) and by what we just showed, in \( G \) there is also an arrowhead at \( q \) on the corresponding path between \( i \) and \( j \). If there is an arrowhead at \( i \) in \( G \) then it is implied that \( q \in \text{ant}(C) \) in \( G \) and therefore there cannot be an arrowhead at \( q \) in \( \alpha_{\text{CMG}}(G; \varnothing, C) \). This also implies that in \( G \) the same edge or path between \( i \) and \( q \) in \( \alpha_{\text{CMG}}(G; \varnothing, C) \) exists. (This can be seen as only cases 4 and 5 of Tables 2 can be applied to the path between \( i \) and \( q \) in \( G \)). Hence, in \( \alpha_{\text{CMG}}(G; M, \varnothing) \) there exists an arrowhead at \( i \), which is a contradiction. Therefore, there is no arrowhead at \( i \) after applying steps 1 or 2 of Algorithm 2 to \( \alpha_{\text{CMG}}(G; \varnothing, C) \). This implies that there is no arrowhead at \( i \) in \( \alpha_{\text{CMG}}(\alpha_{\text{CMG}}(G; \varnothing, C); M, \varnothing) \).

\( (\Leftarrow) \) If in \( \alpha_{\text{CMG}}(\alpha_{\text{CMG}}(G; \varnothing, C); M, \varnothing) \) there is no arrowhead pointing to \( i \) on the \( ij \) edge then there is no arrowhead pointing to \( i \) in \( \alpha_{\text{CMG}}(G; \varnothing, C) \). If there is an arrowhead at \( i \) before applying step 4 of Algorithm 3 then \( i \in \text{ant}(C) \) in \( G \). This implies that there is no arrowhead at \( i \) unless possibly \( i \in \text{ant}(C) \) in \( G \). Suppose now, for contradiction, that in \( G, i \not\in \text{ant}(C) \)
and in $\alpha_{CMG}(G;M,\emptyset)$ there is an arrowhead at $i$ on the $ip$ edge on the corresponding walk between $i$ and $j$. This arrowhead is associated with step 3, 6, 8, or 9 of Table 1. The fact that $i \notin \operatorname{ant}(C)$ in $G$ and $i$ is connected to a line imply that the same paths or tripaths as in the mentioned cases in Table 1 exist in $\alpha_{CMG}(G;\varnothing,C)$. This means that the $ip$ edge with an arrowhead at $i$ exists in $\alpha_{CMG}(\alpha_{CMG}(G;M,\emptyset);\varnothing,C)$, a contradiction. Therefore, in $\alpha_{CMG}(G;M,\emptyset)$ there is no arrowhead at $i$ unless possibly $i \in \operatorname{ant}(C)$ in $G$, which implies that $i \in \operatorname{ant}(C)$ in $\alpha_{CMG}(G;M,\emptyset)$. This implies that there is no arrowhead at $i$ on the $ij$ edge in $\alpha_{CMG}(\alpha_{CMG}(G;M,\emptyset);\varnothing,C)$.

Hence, we denote the function in Proposition 6 by $\alpha_{CMG}(G;M,C)$. We show that the corresponding algorithm (Algorithm 2 followed by Algorithm 3 or vice versa) is well-defined:

**Theorem 7.** For a chain mixed graph $G$ and disjoint subsets $M$, $M_1$, $C$, and $C_1$ of its node set,

$$\alpha_{CMG}(\alpha_{CMG}(G;M,C);M_1,C_1) = \alpha_{CMG}(G;M \cup M_1,C \cup C_1).$$

**Proof.** By definition and Proposition 6, Theorem 5, and Theorem 3, it is implied that

$$\alpha_{CMG}(\alpha_{CMG}(G;M,C);M_1,C_1) = \alpha_{CMG}(\alpha_{CMG}(\alpha_{CMG}(G;M,\emptyset);\varnothing,C);M_1,C_1)
= \alpha_{CMG}(\alpha_{CMG}(\alpha_{CMG}(\alpha_{CMG}(G;M,\emptyset);\varnothing,C);M_1,\emptyset);\varnothing,C_1)
= \alpha_{CMG}(\alpha_{CMG}(\alpha_{CMG}(\alpha_{CMG}(G;\varnothing,C);M,\emptyset);M_1,\emptyset);\varnothing,C_1)
= \alpha_{CMG}(\alpha_{CMG}(\alpha_{CMG}(\alpha_{CMG}(G;M \cup M_1,\emptyset);\varnothing,C);\varnothing,C_1)
= \alpha_{CMG}(\alpha_{CMG}(G;M \cup M_1,\emptyset);\varnothing,C \cup C_1) = \alpha_{CMG}(G;M \cup M_1,C \cup C_1).$$

We now have the main result, which illustrates that by applying Algorithm 2 followed by Algorithm 3 (or vice versa), we obtain the marginal and conditional independence model for a CMG (or a CG) after marginalization and conditioning.

**Theorem 8.** For a chain mixed graph $G$ and disjoint subsets $A$, $B$, $M$, $C$, and $C_1$ of its node set,

$$\langle A, B \mid C_1 \rangle \in J_c(\alpha_{CMG}(G;M,C)) \iff \langle A, B \mid C \cup C_1 \rangle \in J_c(G).$$
Proof. By definition and Proposition 6, Theorem 6, and Theorem 4, it is implied that
\[ \langle A, B \mid C_1 \rangle \in J_c(\alpha_{CMG}(G; M, C)) = J_c(\alpha_{CMG}(\alpha_{CMG}(G; M, \emptyset); \emptyset, C)) \iff \langle A, B \mid C \cup C_1 \rangle \in J_c(\alpha_{CMG}(G; M, \emptyset)) \iff \langle A, B \mid C \cup C_1 \rangle \in J_c(G). \]

\[ \Box \]

7. Anterial graphs. The definition of CMGs can be considered a generalization of the definition of summary graphs (SGs) by [17]: CMGs collapse to SGs when there are no arrowheads pointing to full lines. CMGs are also analogous to SGs in the sense that they capture the marginal and conditional models for CGs, and SGs capture the marginal and conditional models for DAGs; and CMGs exclude graphs with semi-direction-preserving cycles and SGs exclude graphs with direction-preserving cycles.

The class of ancestral graphs, defined by [12], captures the same independence modes as those of SGs, but has a simpler structure than SGs. In this section, we define the class of anterial graphs (AnGs), which can be thought of as a generalization of and analogous to ancestral graphs with the same relationship to CMGs as that of ancestral graphs to SGs.

An anterial graph is a mixed graph that does not contain semi-direction-preserving cycles that contain at least an arrow and does not contain arcs with one endpoint that is an anterior of the other endpoint. This implies that, unlike CMGs, AnGs are simple graphs. For example, in Fig. 8(a) the graph is an AnG, but in Fig. 8(b) the graph is not an AnG because of the existence of the arc \( kq \), where \( k \in \text{ant}(q) \) via the semi-direction preserving path \( (k, j, l, h, q) \) as well as the arc \( qp \), where \( q \in \text{ant}(p) \).

![Fig 8. (a) An AnG. (b) A CMG that is not an AnG.](image-url)

Here we show, from an anterial graph and after marginalization and conditioning, how to generate an anterial graph with the corresponding marginal and conditional independence model.
Algorithm 4. \( \alpha_{\text{AnG}}(G; M, C) \): (Generating an AnG from an anterial graph G)

Start from \( G \).

1. Apply Algorithm 2.
2. Apply Algorithm 3.
3. Generate respectively arrows from \( j \) to \( i \) or arcs between \( i \) and \( j \) for trisides \( j \xrightarrow{o} \cdots \xrightarrow{o} i \xleftarrow{k} \) or \( j \xleftarrow{o} \cdots \xleftarrow{o} i \xrightarrow{k} \) when \( k \in \text{ant}(i) \) if the arrow or the arc does not already exist.
4. Generate respectively an arrow from \( j \) to \( i \) or an arc between \( i \) and \( j \) for trisides \( j \xrightarrow{k_1} \cdots \xleftarrow{k_m} i \xrightarrow{k_1} \cdots \xleftarrow{k_m} i \) when there is an \( 1 \leq r \leq m \) such that \( k_r \in \text{ant}(i) \) if the arrow or the arc does not already exist. Continually apply this step until it is not possible to apply it further.
5. Remove the arc between \( j \) and \( i \) in the case that \( j \in \text{ant}(i) \), and replace it with an arrow from \( j \) to \( i \) if the arrow does not already exist; and remove the arc between \( j \) and \( i \) in the case that \( j \in \text{ant}(i) \) and \( i \in \text{ant}(j) \), and replace it with a line between \( i \) and \( j \) if the line does not already exist.

Notice that, as we will see, steps 3, 4, and 5 of Algorithm 4 generate, from the generated CMG after step 2, an AnG that captures the same independence model as that of the CMG. In addition, in step 4, one \( k_r \) being in \( \text{ant}(i) \) implies that all \( k_r, 1 \leq r \leq m \), are in \( \text{ant}(i) \). Fig. 9 illustrates how to apply these steps to a CMG.

Fig 9. (a) A chain mixed graph \( G \). (b) The graph after applying step 3 of Algorithm 4. (c) The graph after applying step 4. (d) The generated AnG after applying step 5.
We consider Algorithm 4 a function denoted by $\alpha_{\text{AnG}}$. Notice that for every anterial graph $G$, it holds that $\alpha_{\text{AnG}}(G; \emptyset, \emptyset) = G$. We again follow a parallel theory as that in the previous sections:

**Proposition 7.** Graphs generated by Algorithm 4 are AnGs.

**Proof.** By Propositions 3 and 5, we know that, after step 2 of Algorithm 4, we obtain a CMG. Steps 3 and 4 does not generate a semi-direction-preserving cycle with an arrow by generating an arrow from $j$ to $i$: This is because if, for contradiction, that is the case then $j \in \text{ant}(i)$ (for case 3) or $j \in \text{ant}(k)$ and $k \in \text{ant}(i)$ which imply that $j \in \text{ant}(i)$ (in case 4) in the generated graph after applying step 2. This is a contradiction since it means by induction that the semi-direction-preserving cycle with an arrow exists in this graph.

Step 5 obviously removes all arcs with one endpoint that is an anterior of the other endpoint. This step also does not generate semi-direction-preserving cycles with an arrow by replacing an arc $ij$ by an arrow from $j$ to $i$ or an $ij$ line: This is because if, for contradiction, that is the case then $j \in \text{ant}(i)$ in the generated graph after applying step 4, which is a contradiction since it means by induction that the semi-direction-preserving cycle with an arrow exists in this graph.

We first prove that Algorithm 4 does not need to be applied to an anterial graph, but it can be applied to a chain mixed graph. We denote the function corresponding to steps 3, 4, and 5 of Algorithm 4 by $\alpha_{\text{CMG,AnG}}$. Notice that $\alpha_{\text{AnG}}(G; M, C) = \alpha_{\text{CMG,AnG}}(\alpha_{\text{CMG}}(G; M, C))$. Denote also by a walk between $i$ and $j$ on which all sections are collider and every inner node is in $\{i, j\} \cup \text{ant}({i, j})$ a primitive inducing walk. This is a generalization of primitive inducing paths, defined in [12].

**Lemma 4.** Let $H$ be a chain mixed graph and $M$ and $C$ be two subsets of its node set. It holds that $\alpha_{\text{AnG}}(\alpha_{\text{CMG,AnG}}(H); M, C) = \alpha_{\text{AnG}}(H; M, C)$.

**Proof.** There are two differences between $H$ and $\alpha_{\text{CMG,AnG}}(H)$: (1) For an $ij$ arc such that $j \in \text{ant}(i)$ in $H$, the arrowhead at $j$ is removed on $ij$ in $\alpha_{\text{CMG,AnG}}(H)$; (2) for a primitive inducing walk $\pi$ between $i$ and $j$ in $H$, there is an endpoint-identical $ij$-edge in $\alpha_{\text{CMG,AnG}}(H)$. (Notice that the trislides in step 3 of Algorithm 4 yields a primitive inducing walk $\langle j, \ldots, i, k, i \rangle$.) Notice also that $\text{ant}(C)$ is the same in both graphs.

- After applying step 1 of Algorithm 4 (Algorithm 2), for each difference, the following occurs:
(1) This step of the algorithm may generate further differences for (1) if one of the two following situations occurs:
(i) There is a $kj$ edge with an arrowhead pointing to $j$ and $j \in M$. In this case in $\alpha_{CMG,AnG}(H)$ there is obviously an endpoint-identical edge to $\langle k, j, i \rangle$ between $k$ and $i$ by step 4 of Algorithm 4. This implies that this edge also exists after applying step 1. However, the same edge exists in $\alpha_{AnG}(H; M, C)$: For $l$, the node adjacent to $j$ on the semi-direction-preserving path from $j$ to $i$, it holds that $\langle l, j, i \rangle$ is non-collider with no arrowhead at $j$ on the $lj$ edge; hence, an $li$ edge with no arrowhead at $l$ is generated. A $kl$ edge with an arrowhead at $l$ is also generated because of the non-collider tripath $\langle k, j, l \rangle$. Now by step 4 of Algorithm 4, an endpoint-identical $ki$ edge to $\langle k, l, i \rangle$ is generated.
(ii) There is $k \succ \circ \ldots \prec i$ or $k \prec \succ \circ \ldots \succ i$, and $j \in M$. In this case in $\alpha_{CMG,AnG}(H)$ there is obviously an edge between $k$ and $i$ by step 3 of Algorithm 4. This implies that this edge also exists after applying step 1. However, the same edge exists in $\alpha_{AnG}(H; M, C)$: By defining $l$ as above, by step 1 of Algorithm 4, an $li$ arc is generated and $l \in \text{ant}(i)$; hence, by step 3 of Algorithm 4, the same type of $ki$ edge is generated.

(2) Suppose that there is an arrowhead at $j$ on the generated $ij$ edge in $\alpha_{MCG,AnG}(H)$. This step of the algorithm may generate further differences for (2) if one of the two following situations occurs:
(i) There is a $kj$-edge with no arrowhead pointing to $j$, and $j \in M$. In this case, by using the $\langle i, j, k \rangle$, existing in $\alpha_{CMG,AnG}(H)$, a $ki$-edge is generated. For $H$, by using $\langle h, j, k \rangle$, where $h$ is the node adjacent to $j$ on $\pi$, a $kh$-edge is generated, which establishes an endpoint-identical primitive inducing walk between $i$ and $k$ that generates an endpoint-identical $ik$ edge by step 4 of Algorithm 4 since $h \in \text{ant}(k)$.
(ii) There is $k \circ \ldots \circ j$ or $k \circ \ldots \circ j$, the $ij$ edge is an arrow from $i$ to $j$ in $\alpha_{MCG,AnG}(H)$, and $i \in M$. In this case, by using $\langle i, j, \ldots, k \rangle$ in $\alpha_{CMG,AnG}(H)$, an edge between $k$ and $j$ is generated. In $H$, notice that $h$, the adjacent node to $j$ on $\pi$, is in $\text{ant}(j)$. This is because the node $l$, the child of $i$ on $\pi$ is in $\text{ant}(j)$ since the graph is a CMG and it cannot be an anterior of $i$; thus if $h \in \text{ant}(i)$ then $h \in \text{ant}(j)$ via $l$. After marginalizing on $i$ by step 1, $h$ remains an anterior of $j$. Therefore, by step 3 of Algorithm 4, an edge between $k$ and $j$ is generated.

• After applying a part of step 2 of Algorithm 4 (steps 2 and 3 of Algorithm 3) the following occurs:
(1) This step of the algorithm may generate further differences for (1) if one of the two following situations occurs:
(i) There is a path $\pi'$ of form $j \cdot \cdots \cdot \cdot \cdot \prec k$ or $j \cdot \cdots \cdot \cdot \cdot \prec k$ with an inner node in $C \cup \text{ant}(C)$. In this case an endpoint-identical $ki$ edge is generated for $H$ by step 3 of Algorithm 3. However, such an edge already exists in $\alpha_{\text{CMG,AnG}}(H)$ since $\langle \pi', i \rangle$ in $H$ is primitive-inducing.
(ii) There is a path $\pi''$ of form $i \cdot \cdots \cdot \cdot \cdot \prec k$ or $i \cdot \cdots \cdot \cdot \cdot \prec k$ and $j$ in $C \cup \text{ant}(C)$. In this case a $ki$ edge is generated for $H$ by step 2 of Algorithm 3. However, such an edge already exists in $\alpha_{\text{CMG,AnG}}(H)$ by step 3 of Algorithm 4 using $\langle \pi'', j \rangle$.

(2) This step of the algorithm may generate further differences for (2) if one of the two following situations occurs:
(i) There is an arrowhead pointing to $j$ on $\pi$ and there is a path $\pi'$ of form $j \cdot \cdots \cdot \cdot \cdot \prec k$ or $j \cdot \cdots \cdot \cdot \cdot \prec k$ with an inner node in $C \cup \text{ant}(C)$. In this case an endpoint-identical $ki$ edge is generated for $\alpha_{\text{CMG,AnG}}(H)$. For $H$, by using $\langle h, j, \ldots, k \rangle$, where $h$ is the node adjacent to $j$ on $\pi$, a $kh$-edge is generated, which establishes an endpoint-identical primitive inducing walk between $i$ and $k$ that generates an endpoint-identical $ik$ edge by step 4 of Algorithm 4 since $h \in \text{ant}(k)$.
(ii) There is an arrowhead pointing to $j$ on $\pi$ and there is a path $\pi''$ of form $i \cdot \cdots \cdot \cdot \cdot \prec k$ or $i \cdot \cdots \cdot \cdot \cdot \prec k$ and $j$ in $C \cup \text{ant}(C)$. In this case a $ki$ edge is generated for $\alpha_{\text{CMG,AnG}}(H)$ by step 2 of Algorithm 3. For $H$, consider the node $l$ adjacent to $i$ on $\pi$. If $l \in \text{ant}(i)$ then by step 3 of Algorithm 4, a $ki$ edge is generated. If $l \in \text{ant}(i)$ then $l \in \text{ant}(C)$ and by step 2 of algorithm 3, we are done again.

• After applying step 4 of Algorithm 3 the following occurs:
(1) This step of the algorithm may generate further differences for (1) if $j \in C \cup \text{ant}(C)$. In this case the arrowhead at $j$ on the $ij$-arc in $H$ is removed, which, however, is already the case in $\alpha_{\text{CMG,AnG}}(H)$;
(2) This step of the algorithm may generate further differences for (2) if any of the nodes on $\pi$, say $l$, is in $C \cup \text{ant}(C)$. However, in $H$, by the previous step of the algorithm, the nodes that are adjacent to the endpoints of the section containing $l$ on $\pi$ have become adjacent, and establish a shorter primitive inducing walk between $i$ and $k$ (or $j$).

Hence, thus far, the differences between the two generated graphs are the same as the differences between $H$ and $\alpha_{\text{CMG,AnG}}(H)$. Therefore, by applying steps 3, 4, and 5 of Algorithm 4, the same graphs will be generated. \qed
Theorem 9. For an anterial graph $G$ and disjoint subsets $M$, $M_1$, $C$, and $C_1$ of its node set,

$$\alpha_{\text{AnG}}(\alpha_{\text{AnG}}(G; M, C); M_1, C_1) = \alpha_{\text{AnG}}(G; M \cup M_1, C \cup C_1).$$

Proof. Using Theorem 7 and Lemma 4, we have the following:

$$\alpha_{\text{AnG}}(\alpha_{\text{AnG}}(G; M, C); M_1, C_1) = \alpha_{\text{AnG}}(\alpha_{\text{CMG}}.\alpha_{\text{AnG}}(\alpha_{\text{CMG}}(G; M, C)); M_1, C_1) = \alpha_{\text{AnG}}(G; M \cup M_1, C \cup C_1).$$

Denote the set of all AnGs by $\text{ANG}$.

Proposition 8. The map $\alpha_{\text{AnG}} : \mathcal{CG} \to \mathcal{ANG}$ is surjective.

Proof. Consider an arbitrary anterial graph $H$. We define a chain graph $G$ from $H$ in the same way as in the proof of Proposition 4: Keep all arrows and lines of $H$ in $G$ and replace arcs $ij$ with $i \prec m \succ j$; and define a subset $M$ of the node set of $G$ as the set of all such $m$. By Proposition 4, we know that $G$ is a CG.

Here we prove that $\alpha_{\text{AnG}}(G; M, \emptyset) = H$. Again, by Proposition 4, we know that after step 1 of Algorithm 4, we obtain $H$. Step 2 does not change the graph since the conditioning set is empty. Steps 3, 4, and 5 do not change the graph either since $H$ is an AnG, and these steps do not change AnGs.

Theorem 10. For an anterial graph $G$ and disjoint subsets $A$, $B$, $M$, $C$, and $C_1$ of its node set,

$$\langle A, B \mid C_1 \rangle \in J_c(\alpha_{\text{AnG}}(G; M, C)) \iff \langle A, B \mid C_1 \rangle \in J_c(G).$$

Proof. By Theorem 8, it is enough to prove that $A \perp_c B \mid C_1$ in $\alpha_{\text{AnG}}(G; M, C)$ if and only if $A \perp_c B \mid C_1$ in $\alpha_{\text{CMG}}(G; M, C)$.

Since Steps 1 and 2 of Algorithm 4 generate $\alpha_{\text{CMG}}(G; M, C)$, we need to prove that there is a $c$-connecting walk in a chain mixed graph $H$ if and only if there is a $c$-connecting walk after applying steps 3, 4, and 5 of the algorithm to $H$.

($\Rightarrow$) Suppose that there is a $c$-connecting walk $\pi$ given $C_1$ between $i$ and $j$ in $H$. After applying steps 3 and 4, $\pi$ is intact. If an arc $kl$ is replaced by
an arrow from $k$ to $l$ or a $kl$ line in step 5 of the algorithm then we have the two following cases: 1) If $k$ is on a non-collider section on $\pi$ by using the $kl$ arrow or line instead of arc, one obtains a $c$-connecting walk. 2) Suppose that $k$ is an endpoint of a collider section $\rho$ and there is $\pi_1 = \langle h, \rho, l \rangle$ on $\pi$. By Lemma 1, one can assume that $\rho$ is a path. It is easy to show that before steps 3 and 4 of the algorithm $k \in \text{ant}(l)$. If $h \neq l$ then by step 4, there is an endpoint identical $hl$ edge to $\pi_1$. One can now use the $hl$ edge instead of $\pi_1$ to obtain a $c$-connecting walk. If $h = l$ then $\rho$ can be considered to be the single node $k$. Now if $h$ is on a non-collider section then we can easily skip $k$ to obtain a $c$-connecting path. If $h$ is an endpoint of a collider section $\rho'$ then from $\pi_2 = \langle q, \rho', k \rangle$ and by using step 3 of the algorithm, we obtain an endpoint-identical $qh$ edge, which can be replaced by $\pi_2$ to obtain a $c$-connecting path. This, by an inductive argument, implies the result.

$(\Leftarrow)$ Suppose that there is a $c$-connecting walk $\pi$ given $C_1$ between $i$ and $j$ in $\text{arg}(H; \emptyset, \emptyset)$, which is the graph $H$ after applying steps 3, 4, and 5 of Algorithm 4. If by step 5 of the algorithm an arc $kl$ is replaced by an arrow from $k$ to $l$ or a $kl$ line then we need to check the case where there is a subwalk $h \xrightarrow{o} \cdots \circ k$ or $h \xleftarrow{o} \cdots \circ k$ on $\pi$. In this case, by what we showed in the other direction of the proof, before applying step 5, $k \in \text{ant}(l)$, and therefore, there exists an endpoint-identical $hl$ edge, generated by step 3 or 4 of the algorithm, which can be used instead of $\langle h, \ldots, k, l \rangle$ to obtain a $c$-connecting walk. By an inductive argument, we imply that there is a $c$-connecting walk $\pi'$ between $i$ and $j$ before applying step 5.

Now suppose that an arrow or an arc $kl$ on $\pi'$ has been generated by step 4 of the algorithm. This implies that there exists an endpoint-identical collider trislide $\pi_0 = \langle k, k_1, \ldots, k_m, l \rangle$, where $k_r$ are in $\text{ant}(l)$; i.e. there is a semi-direction-reserving path $\tau_0$ from $k_r$ to $l$. Notice first that after applying step 4, antecursors do not change; thus $k_r$ is an anterior of $l$ before applying step 4. For an arbitrary fixed $r$, if $k_r \in C_1 \cup \text{ant}(C_1)$ then we replace $kl$ with $\pi_0$ and use Lemma 1 to obtain a $c$-connecting walk given $C_1$. If $k_r \notin C_1 \cup \text{ant}(C_1)$ then we replace $kl$ with $\pi_1 = \langle \pi_{kr}, \tau_0 \rangle$, where $\tau_{kr}$ is a subpath of $\tau_0$ with endpoints $k$ and $k_r$. Notice that in $\pi_1$, all sections are non-collider, and we know that all nodes are outside $C_1$. Therefore, $\pi_1$ is $c$-connecting given $C_1$ itself. In addition, $l$ is on a non-collider section on $\pi'$ since otherwise $k_r \in \text{ant}(C_1)$. Hence, after this replacement, $l$ remains on a non-collider section outside $C_1$, which implies that the new walk remains $c$-connecting. By induction now, there is a $c$-connecting walk $\pi''$ between $i$ and $j$ before step 4 of the algorithm.
Now suppose that an arrow or an arc $kl$ on $\pi''$ has been generated by step 3 of the algorithm. This implies that there exists an $\pi_2 = k \rightarrow o \cdots \rightarrow l$ or $\pi_2 = k \leftarrow o \cdots \leftarrow l$, where $h$ is a spouse of $l$ and $h \in \text{ant}(l)$, i.e. there is a semi-direction-reserving path $\tau_1$ from $h$ to $l$. Notice first that after applying step 3, anteriors do not change; thus $h$ is an anterior of $l$ before applying step 3. If $l$ is on a collider path on $\pi''$ or $l$ is a non-collider and all inner nodes of $\pi_2$ are outside $C_1$ then we replace $kl$ with $\pi_2$ to obtain a $c$-connecting walk given $C_1$. If $l$ is a non-collider and an inner node of $\pi_2$ is in $C_1$ then we replace $kl$ with $\langle k, \ldots, l, h, \tau_1 \rangle$ in the case that $h \in C_1$; by $\langle k, \ldots, l, h, \tau_{1q} \rangle$ in the case that $h \notin C_1$ and no node on $\tau_1$ is in $C_1$; and by $\langle k, \ldots, l, h, \tau_{1q}, \tau_{1q}^r, l \rangle$, where $\tau_{1q}$ is the subwalk of $\tau_1$ from $l$ to $q$ and $\tau_{1q}^r$ is $\tau_{1q}$ in the reverse direction, in the case that $h \notin C_1$ and $q \in \tau_1$ is in $C_1$. It is straightforward to check that these walks between $k$ and $l$ are all $c$-connecting given $C_1$. In addition, after the replacement, $l$ remains on a non-collider section. Hence the walk between $i$ and $j$ remains $c$-connecting given $C_1$. Again by induction, we obtain a $c$-connecting walk given $C_1$ between $i$ and $j$ before step 3 of the algorithm, which completes the proof.

Corollary 4. The class of anterial graphs, $\mathcal{ANG}$, with $c$-separation criterion is stable under marginalization and conditioning.

8. Probabilistic independence models for CMGs and AnGs. The most interesting independence models are induced by probability distributions. Consider a set $V$ and a collection of random variables $(X_\alpha)_{\alpha \in V}$ with joint density $f_V$. By letting $X_A = (X_\alpha)_{\alpha \in A}$ for each subset $A$ of $V$, we then use the short notation $A \perp \perp B \mid C$ for $X_A \perp \perp X_B \mid X_C$ and disjoint subsets $A$, $B$, and $C$ of $V$.

For a given independence model $J$, a probability distribution $P$ is called faithful with respect to $J$ if, for random vectors $X_A$, $X_B$, and $X_C$ with probability distribution $P$,

$$A \perp \perp B \mid C \text{ if and only if } \langle A, B \mid C \rangle \in J.$$ 

We say that $J$ is probabilistic if there is a distribution $P$ that is faithful to $J$.

From a given collection of random variables $(X_\alpha)_{\alpha \in V}$ with a probability distribution $P$, one can induce an independence model $J(P)$ by demanding

$$\text{if } A \perp \perp B \mid C \text{ then } \langle A, B \mid C \rangle \in J(P).$$

Notice that $J(P)$ is obviously probabilistic.
For a chain graph $G$, we say that a probability distribution with density $f$ factorizes with respect to $G$ if

$$f(x) = \prod_{\tau \in T} f(x_{\tau} | x_{pa(\tau)}) ,$$

where $T$ is the set of chain components of $G$; and

$$f(x_{\tau} | x_{pa(\tau)}) = \prod_a \phi_a(x),$$

where $a$ varies over all subsets of $\tau \cup pa(\tau)$ that are complete in the moral graph of the subgraph of $G$ induced by $\tau \cup pa(\tau)$, and $\phi_a(x)$ is a function that depends on $x$ through $x_a$ only; see [7] for more discussion.

Now let $\alpha(P; M, C)$ be the probability distribution obtained by usual probabilistic marginalization and conditioning for the probability distribution $P$. It is easy to show that if $P$ is faithful to $J$ then $\alpha(P; M, C)$ is faithful to the marginal and conditional independence model $\alpha(J; M, C)$; see Theorem 7.1 and Corollary 7.3 of [12].

It is also known that if $G$ is a CG then there is a regular Gaussian distribution that is faithful to it. In fact, almost all the regular Gaussian distributions that factorize with respect to a CG are faithful to it; see [11]. In other words, the independence model $J_c(G)$ is probabilistic.

Hence, it is implied that if $H$ is a CMG or an AnG then $J_c(H)$ is probabilistic; i.e. there is a distribution (in fact a Gaussian distribution) that is faithful to it. This is because we know that, for every CMG or anterial graph $H$, there is a chain graph $G$ and subsets of its node set $M$ and $C$ such that $\alpha_{CMG}(G; M, C) = H$ or $\alpha_{AnG}(G; M, C) = H$ (Propositions 2, 4, or 8); by what we discussed above there is a regular Gaussian distribution $N$ that is faithful to $J_c(G)$, and therefore $\alpha(N; M, C)$ is faithful to $\alpha(J_c(G); M, C) = J_c(H)$. One can obtain the same result for the strictly positive discrete probability distribution since there is such a distribution that is faithful to a given CG [16]. These results motivate the use of CMGs and AnGs.

In the Gaussian case, there exists a known parametrization for the regular Gaussian distributions that factorize with respect to a CG; see [19] and [11] for two slightly different but equivalent parametrizations.

The main obstacle in providing a parametrization for the classes of CMGs and AnGs is that they are not maximal in the sense that there are missing non-adjacent pairs of nodes to which there are no independence statements associated. This is clear since, for example, the class of ancestral graphs is a subclass of both MCGs and AnGs, and there exist non-maximal ancestral
graphs; see also Fig. 10, for an example of an AnG that is not ancestral and that induces no independence statement of form $j \perp \perp k \mid C$ for any choice of $C$. However, there is a method to generate, from non-maximal CMGs and AnGs, maximal CMGs and AnGs that induce the same independence models. We leave the details of this method for future work.

Fig 10. A non-maximal AnG.

In addition, as explained before, it is clear that MCGs act similarly to summary graphs in the problem of marginalization and conditioning for DAGs, and AnGs act similarly to ancestral graphs. There is a known parametrization for maximal ancestral graphs in the Gaussian case; see [12]. It seems that such a parametrization can be generalized for the class of maximal AnGs, which is again subject to further work.

Acknowledgements. The author is grateful to Steffen Lauritzen and Thomas Richardson for helpful discussions, and Nanny Wermuth for helpful discussions and comments.

References.
[1] Andersson, S. A., Madigan, D. and Perlman., M. D. (2001). Alternative Markov Properties for Chain Graphs. Scand. J. Stat. 28 33-85.
[2] Cox, D. R. and Wermuth, N. (1993). Linear dependencies represented by chain graphs (with discussion). Stat. Sci. 8 204–218; 247–277.
[3] Drton, M. (2009). Discrete chain graph models. Bernoulli 15 736–753.
[4] Frydenberg, M. (1990). The chain graph Markov property. Scand. J. Stat. 17 333–353.
[5] Kiiveri, H., Speed, T. P. and Carlin, J. B. (1984). Recursive causal models. J. Aust. Math. Soc., Ser. A 36 30–52.
[6] Koster, J. T. A. (2002). Marginalizing and conditioning in graphical models. Bernoulli 8 817–840.
[7] Lauritzen, S. L. (1996). Graphical Models. Clarendon Press, Oxford, United Kingdom.
[8] Lauritzen, S. L. and Spiegelhalter, D. J. (1988). Local computations with probabilities on graphical structures and their application to expert systems. J. Roy. Statist. Society B 50 157-224.
[9] Lauritzen, S. L. and Wermuth, N. (1989). Graphical models for association between variables, some of which are qualitative and some quantitative. Ann. Statist. 17 31–57.
[10] Marchetti, G. M. and Lupparelli, M. (2011). Chain graph models of multivariate regression type for categorical data. Bernoulli 17 827-844.
[11] PEÑA, J. M. (2011). Faithfulness in Chain Graphs: The Gaussian Case. In Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS 2011) 15 588-599. JMLR.org.

[12] RICHARDSON, T. S. and SPIRTES, P. (2002). Ancestral graph Markov models. Ann. Statist. 30 962–1030.

[13] SADEGHI, K. (2013). Stable mixed graphs. Bernoulli 19 2330-2358.

[14] STUDENY, M. (1998). Bayesian Networks from the Point of View of Chain Graphs. In UAI 496-503. Morgan Kaufmann, San Francisco, CA.

[15] STUDENY, M. (2005). Probabilistic Conditional Independence Structures. Springer-Verlag, London, United Kingdom.

[16] STUDENY, M. and BOUCKAERT, R. R. (1998). On chain graph models for description of conditional independence structures. Ann. Statist. 26 1434–1495.

[17] WERMUTH, N. (2011). Probability distributions with summary graph structure. Bernoulli 17 845–879.

[18] WERMUTH, N. and SADEGHI, K. (2012). Sequences of regressions and their independences. TEST 21 215-252 and 274-279.

[19] WERMUTH, N., WIEDENHECK, M. and COX, D. R. (2006). Partial inversion for linear systems and partial closure of independence graphs. BIT 46 883–901.