Physica A 596 (2022) 127097

Contents lists available at ScienceDirect

Physica A

journal homepage: www.elsevier.com/locate/physa

# Cycle analysis of Directed Acyclic Graphs

Vaiva Vasiliauskaite [a,*], Tim S. Evans [b], Paul Expert [c,d,e]

[a] Computational Social Science, ETH Zurich, Zurich, CH 8092, Switzerland
[b] Centre for Complexity Science, Theoretical Physics Group, Imperial College London, London, SW7 2AZ, UK
[c] Global Business School for Health, University College London, London, WC1E 6BT, UK
[d] World Research Hub Initiative, Tokyo Institute of Technology, Tokyo, 152-8550, Japan
[e] Department of Primary Care and Public Health, Imperial College London, SW7 2AZ, London, UK

ARTICLE INFO

*Article history:*
Received 20 September 2021
Received in revised form 15 February 2022
Available online 25 February 2022

*Keywords:*
Complex systems
Network theory
Data science
Statistics
Minimal Cycle Bases
Directed Acyclic Graphs
Transitive reduction

ABSTRACT

In this paper, we employ the decomposition of a directed network as an undirected graph plus its associated node meta-data to characterise the cyclic structure found in directed networks by finding a Minimal Cycle Basis of the undirected graph and augmenting its components with direction information. We show that only four classes of directed cycles exist, and that they can be fully distinguished by the organisation and number of source–sink node pairs and their antichain structure. We are particularly interested in Directed Acyclic Graphs and introduce a set of metrics that characterise the Minimal Cycle Basis using the Directed Acyclic Graphs meta-data information. In particular, we numerically show that transitive reduction stabilises the properties of Minimal Cycle Bases measured by the metrics we introduced while retaining key properties of the Directed Acyclic Graph. This makes the metrics a consistent characterisation of Directed Acyclic Graphs and the systems they represent. We measure the characteristics of the Minimal Cycle Bases of four models of transitively reduced Directed Acyclic Graphs and show that the metrics introduced are able to distinguish the models and are sensitive to their generating mechanisms.

© 2022 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

## 1. Introduction

Hierarchy is a landmark of complexity [1], and can take multiple forms, e.g. "order", "level", "control" or "inclusion" [2]. A unifying feature of all types of hierarchy is that they can be represented as a partially ordered set (poset) that represents the relationships between elements of the system. The term "partial" reflects the fact that not all pairs of elements need to be directly ordered. Directed Acyclic Graphs (DAGs) do not contain directed closed paths and can thus be used to represent the relationships between the elements of a poset. A key distinguishing feature between a poset and a DAG representation of a poset is that each relationship encoded in a poset may be represented by either an explicit edge in the DAG or implicitly by a path within the DAG. If every directed edge in the DAG defines a relationship in the equivalent poset, then we find from the transitivity condition of the poset that any nodes linked by a path in the DAG must also define a relationship in the poset, whether or not there is an edge representing a direct connection. A good example is a citation network where each node is a paper and directed edges represent citations. As citations are always from newer to older papers, a citation network is a DAG. We can look back in time from one chosen publication following paths in the citation network to see which earlier papers influenced it. However, most of these earlier papers will not be listed

* Corresponding author.
*E-mail address:* vvasiliau@ethz.ch (V. Vasiliauskaite).

https://doi.org/10.1016/j.physa.2022.127097
0378-4371/© 2022 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

in the bibliography of the chosen paper so most relationships are indirect, encoded only by paths of length two or more. These indirect path-only connections in the DAG are always explicit relationships in the corresponding citation poset.

DAGs are thus "doubly-complex" systems. The first level of complexity lies in the order relationship underlying the DAG, and the second comes from the "missing information", relationships in the poset which are not realised as explicit edges in the DAG. In this paper, we characterise the mesoscopic structures present in DAGs that are carved in the underlying poset by the unobserved transitive edges.

Mesoscopic structures are rich descriptors helping us to understand the organisation of complex systems, and the characterisation of connectivity patterns is one of the pillars of the study of complex networks. These structures are usually defined in terms of "over-connectivity" and the notion of more densely connected subsets of nodes is at the centre of the definition of many mesoscopic organisation patterns. Such structures can take the form of core–periphery [3,4], community [5–11], or clique [12–14]. Cliques, cores, and communities are important examples of emergent organisational patterns that can be interpreted as capturing beyond pairwise node organisation. The representation of higher-order interaction and their formalisation in terms of hypergraphs or simplicial complexes has recently seen a surge of interest [15,16].

While over-connectivity can be used to define mesoscopic or higher order structures, it is not ideal for DAGs. In this case, it is more appropriate to define structures by the *absence* of connectivity. The notion of ANTICHAINS—generalised "anti-paths"— which group together nodes that do not share direct connectivity, are relevant in DAGs, as they are composed of nodes at the same hierarchical level [17]. Another example are cycles: they can be described as a subset of nodes that have exactly two neighbours in their induced subgraph. A cycle can be seen as the boundary of a potential clique in a thresholded graph, or a hole in the fabric of a network. Cycles are also examples of structures that encode mesoscopic information as they form organised motifs.

In this paper we show that, despite their name, cycles do play a role in the analysis of DAGs and that directed trees are the only DAGs with no cycles. Our starting point is the following, albeit informal, definition of a network: a network is *a graph with something more*. This definition clearly highlights that graphs, being pure combinatorial objects, do not encode all the nuances and constraints of the complex systems that they represent. We formalise this definition by considering a DAG to be a combination of two elements: (1) an undirected graph and (2) the meta-data associated with its nodes and edges.

The idea at the centre of our work is to use meta-data to give interpretability and contextualisation to the Minimal Cycle Basis associated with the undirected graph underlying a DAG. We then introduce metrics characterising the properties and organisation of generalised directed cycles forming said basis. In particular, we use transitive reduction on DAGs and show that the metrics we introduce differentiate between four different DAGs generating models; two deterministic ones: the lattice and Russian Dolls DAGs and two random: Erdős–Rènyi and Price model DAGs. Transitive reduction on edges reveals the tree-ness of the graph underlying a DAG by removing non-essential cycles. We introduce a new compartment for "acyclic" network analysis in the rich topological data analysis toolbox.

This paper is organised as follows. In Section 2 we first formally describe Directed Acyclic Graphs and transitive reduction. Then, we define and describe the generalised directed cycles and cycle bases and conclude by discussing the desirable properties transitively reduced DAGs have with respect to cycles bases. In Sections 2.3 and 2.4, we discuss the procedures used to compute the transitively reduced minimum cycle basis. In Section 2.5 we introduce metrics for characterising cycle bases. And finally, in Section 3, we study the differences between Minimum Cycle Bases in several DAGs models, and show that our proposed metrics can differentiate and characterise network families.

## 2. Methods

In this section, we review the basic properties of Directed Acyclic Graphs (DAGs), define four classes of generalised cycles in DAGs and introduce Minimal Cycle Bases (MCB), and discuss the effect of Transitively Reduction on the MCB.

### 2.1. Graphs, hierarchy, and order

For many data sets, an undirected graph $\mathcal{G}$ is an excellent way to capture important relational information. The set $\mathcal{V}$ of $|\mathcal{V}| = N$ objects of interest form the nodes, and the existence of a relationship between a pair of nodes is encoded by an undirected edge, gives us the edge set $\mathcal{E}$ with $|\mathcal{E}| = E$ edges. For instance, the nodes could be documents and we connect nodes by an edge if one document cites another giving us an undirected network representation of what is known as a citation network.

We are interested in the cycles in our networks, which are simply paths between vertices that run along the edges in a graph, as long as they end where they started without any other vertex being visited twice. Formally, a path is a sequence of nodes in the network, $\{v_0, v_1, v_2, \ldots, v_\ell\}$, where every consecutive pair of nodes is an edge in the graph $(v_i, v_{i+1}) \in \mathcal{E}$[1] [19]. A cycle is then a path that starts and ends at the same vertex ($v_0 = v_\ell$, giving a closed path) but

[1] Note in some contexts, for example section 4.2.5 of [18], what we call a path is known as a 'walk' while our self-avoiding paths are what others call a 'path'.

otherwise does not intersect itself, therefore a cycle can be treated as a subgraph $\mathcal{C}$ in which all nodes have a degree of two.

However, in many cases we can go further as additional information in the meta-data gives a natural direction to the pairwise relationships. In our citation network example, the edge could point from the document whose bibliography contains the second document. The set of nodes in the directed graph representation of the data is the same $\mathcal{V}$. The edges in the directed case are also the same pairs of nodes as the edges in the undirected case but are now endowed with a direction, $\mathcal{E}_{\text{dir}}$. That is if $(v_i, v_j)$ is an edge from $v_i$ to $v_j$ in the directed case we know that $(v_i, v_j) = (v_j, v_i)$ is also an edge in the undirected representation. The paths in a directed graph must respect the direction of each edge so that in the formal definition $(v_i, v_{i+1})$ must be an edge, it is not sufficient for $(v_{i+1}, v_i)$ to be an edge of a path in directed graph $\mathcal{D}$. Closed directed paths are called directed cycles.

We are interested in a special case where there are no directed cycles in the directed network representation so that we have a Directed Acyclic Graph. This lack of directed cycles reflects an order implicit in our system, typically encoded in some additional information available in our data set. The simplest origin of such an order is a natural hierarchy. For instance, in some contexts it might be useful to direct the links in our citation network from high impact to lower impact papers (see journal ranking [20] or code prerequisites [17] for further examples). Another common form of this order is derived from time stamped nodes. For instance each publication has a publication date and we can only cite from newer to older documents. Thus in the directed graph representation of a citation network, the directed paths in the network will never form a directed cycle because that would require at least one document to cite a document published later in time.[2] A couple of points are worth noting. The meta-data may not always be able to order every pair of nodes. Equally, just because two nodes can be placed in a certain order by the meta-data does not mean that there must be an edge between that pair of nodes. In our DAG citation network, two papers can have the same publication date and papers do not cite all papers published before them.

An undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ can thus be turned into a directed graph $\mathcal{D} = (\mathcal{V}, \mathcal{E}_{\text{dir}})$ if its nodes are endowed with meta-data such that any edge can be assigned a direction based on the order between its nodes and form a directed edge set $\mathcal{E}_{\text{dir}}$. In this work, we do not allow multiple edges between nodes and only consider simple graphs with no self-loops. Using some meta-data, we consider a local order for each pair of nodes that are connected by an edge in $\mathcal{G}$. This order is a relation operator $\preceq$ that possesses reflexivity: $u \preceq u$, and antisymmetry: if $u \preceq v$ and $v \preceq u$ then $u = v$, while transitivity is implicit, since the relations concern only a pair of nodes. However, if the collection of local orders $\mathcal{O}$ applied to the set of all nodes $\mathcal{V}$ also possesses transitivity: where $u \preceq v$ and $v \preceq w$ implies $u \preceq w$ for all $u, v, w \in \mathcal{V}$, then the set of order relations $\mathcal{O}$ together with $\mathcal{V}$ form a partially ordered set — a POSET, while $\mathcal{O}$ itself is known as a partial order.

We define a function $F_{\text{dir}}$ that takes an undirected graph $\mathcal{G}$ and all available pairwise order relations between nodes in $\mathcal{O}$ and yields a directed graph $\mathcal{D}$:

$$F_{\text{dir}}(\mathcal{G}, \mathcal{O}) = \mathcal{D}. \tag{1}$$

$F_{\text{dir}}$ maps the nodeset onto itself, $F_{\text{dir}}(v \in \mathcal{V}, \mathcal{O}) \to v \in \mathcal{V}_{\text{dir}}$ for all nodes in $\mathcal{V}$, whereas for edges, $F_{\text{dir}}$ maps each undirected edge $(u, v) \in \mathcal{E}$ onto one directed edge in $\mathcal{E}_{\text{dir}}$, based on how the pairs of nodes are ordered in $\mathcal{O}$:

$$F_{\text{dir}}((u, v), \mathcal{O}) = \begin{cases} (u, v) \in \mathcal{E}_{\text{dir}} \text{ if } u \preceq v \text{ in } \mathcal{O} \\ (v, u) \in \mathcal{E}_{\text{dir}} \text{ if } u \succeq v \text{ in } \mathcal{O}. \end{cases} \tag{2}$$

In the case when $\mathcal{O}$ is a partial order, $F_{\text{dir}}$ maps $\mathcal{G}$ onto a Directed Acyclic Graph $\mathcal{D}$.

Furthermore, we define a function $F_{\text{undir}}$ that maps a directed graph $\mathcal{D}$ to its underlying undirected $\mathcal{G}$ by striping its edges from their direction, see Fig. 1(a) for an illustration of both functions. $F_{\text{dir}}$ can be applied to any graph or subgraph and thus to cycles. In the next section, we will study in detail the images of cycles obtained by $F_{\text{dir}}$ and show that they can be classified into four general classes.

In this paper, we will focus our attention on a subclass of DAGs: transitively reduced DAGs, or reduced DAGs for short. Transitive reduction is an operation that, when applied to a Directed Acyclic Graph, removes all edges representing transitive relationships and thus only keeps the edges that are essential to maintain connectivity. Specifically, if for an edge $(u, v)$ there is another, longer path which connects nodes $u$ and $v$, the edge $(u, v)$ is not essential and can be "reduced", that is such an edge is not present in the reduced DAG. This operation amounts to keeping the longest path(s) between every pair of nodes and thus the reduced DAG is a pruned version of a DAG that preserves nodes and the partial order structure of $\mathcal{O}$ [22] while keeping the least number of edges (see Fig. 1a). Moreover, transitive reduction of a DAG yields a unique reduced DAG, contrary to general directed graphs. A reduced DAG is thus a well-defined sparsified version of the original DAG. Transitive reduction has been applied to study citation networks, which form DAGs, to characterise the types of citations received by papers [23].

[2] Of course, in practice data is never perfect. In our example, documents can be produced in different versions at different times so the directed citation network may produce a few directed cycles, less than one percent in some examples [21]. However, the order encoded in the meta-data, here the time-stamp of the vertices, gives us several ways to produce a pure DAG representation.

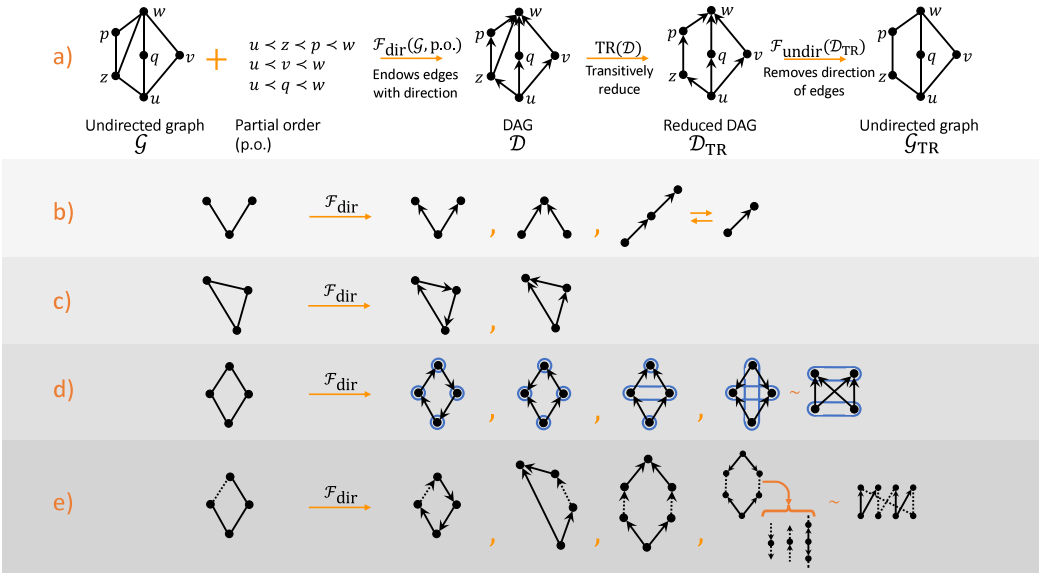


**Fig. 1.** (a) Illustration of the central idea of the paper, a directed network is a combination of an underlying graph and meta-data associated with directionality. The functions $F_{\text{dir}}$ and $F_{\text{undir}}$ map between the undirected and directed versions of a graph. Transitive reduction can be used to simplify DAG and their corresponding underlying graph. (b) The undirected wedge has three possible images: the source wedge, the sink wedge and the neutral wedge. The wedge contraction operation transforms a neutral wedge which has at most one non-neutral node as boundaries into an edge; the edge extension operation transforms an edge into a neutral wedge. (c) An undirected triangle can only be mapped onto two classes of generalised directed cycles: feedback and shortcut. (d) Size four undirected cycles can be mapped into four generalised cycles classes: feedback, shortcut, diamond and mixer. Antichains are shown in blue. Shortcut, diamond and mixer generalised cycles can be present in DAGs, but only diamonds and mixers can be present in transitively reduced DAGs. (e) Undirected cycles of any size larger than four can be mapped onto the same four generalised cycles classes as in (d). Each class is fully characterised by its antichain structure in its contracted form, see (d), and the relative positions and number of source and sink nodes pairs, see Table 1. Dashed line represent chains of nodes of arbitrary length. For feedback, shortcut and diamonds cycles, the dashed line can only be chains of neutral nodes pointing to a set direction. For mixers, dashed lines can be made up of neutral nodes, but may also contain additional source and sink wedges attached to neutral nodes in coherent direction.

### 2.2. Generalised directed cycles

Our definition of a graph $\mathcal{G}$ and its directed counterpart $\mathcal{D}$ means that for every cycle in the undirected graph, we can trace out the same set of vertices in the DAG, while retaining the sequence of directions of the edges. Since each cycle $\mathcal{C}$ is a proper subgraph of $\mathcal{G}$, we can apply $F_{\text{dir}}$ to obtain all possible directed cycle images. We will call these cycles *generalised directed cycles*, a term we prefer to *oriented cycles* [24,25] as it shows that they include cycles that are directed closed paths and also avoids confusion with the notion of orientation in algebraic topology. We will use the uniqueness of the undirected graph underlying each directed and directed acyclic graph to give new insights into the organisation of data with a hierarchy by considering generalised directed cycles and the information they contain about the mesoscopic organisation of directed graphs in general and DAGs in particular.

In this section, we will show that the directional organisation of generalised cycles of any size is well structured and can be organised into four classes. Each class has a natural interpretation in terms of information processing. We will build our argument by inspecting directed motifs of size 3, then 4 and show that no other category appears for larger cycles.

First, let us note that nodes in a generalised directed cycle can be of three types, based on the way the orientation of its edges: (i) a SOURCE NODE: both edges point outwards, (ii) a SINK NODE: both edges point inwards, (iii) a NEUTRAL NODE: one edge points inwards and the other outwards. These nodes naturally induce wedges (see Fig. 1(b)): (i) the SOURCE WEDGE ( ), (ii) the SINK WEDGE ( ), (iii) the NEUTRAL WEDGE ( ). Closing a wedge by adding a third edge produces the smallest possible cycles. Let us close each wedge by adding an edge, giving in total 6 possibilities. It is easy to see by inspection that out of these 6 possibilities, there are in reality only two distinct types: directed cycles, comprising only neutral nodes, and shortcut triangles comprising one node of each type, see Fig. 1(c). Moreover, one can be turned into the other using the EDGE REVERSAL operation that flips the direction of an edge.

We now define the EDGE-WEDGE EXTENSION operation that increases the size of a cycle by one by transforming an edge into a neutral wedge by adding a node ( ). First, let us consider the three node directed feedback cycle. By adding a neutral node via edge-wedge extension we obtain a cycle of the form ( ) which remains a feedback cycle, as all its nodes remain neutral. The edge reversal operation applied to a feedback cycle produces a shortcut cycle: reversing the

direction of any one edge in a four node feedback cycle produces a four node shortcut cycle: . This cycle can also be obtained from the shortcut triangle by extending the arc that does not directly connect the source and sink nodes. Although at the three node scale, there are no other directed cycle images that can be distinguished, two more classes of cycles emerge when considering cycles of four nodes. Starting from a four node shortcut cycle, reversing the direction of an edge adjacent to the first one reversed yields a new type of cycle, the diamond: . Finally, if we reverse the edge opposite to the first one instead of an adjacent one in the four nodes shortcut cycle, we obtain another type of cycle, the mixer: . All other edge reversals lead back to a cycle isomorphic to one these. We note that the directed and shortcut types do not exist in transitively reduced DAGs, while the two new types, the diamond and mixer do. We will show below that no new classes appear as the size of a cycle increases.

First let us characterise these four classes of generalised cycles using the number of source–sink node pairs and their antichain composition. We then show that no larger cycles can create a structure that falls outside this classification. The way we constructed each class directly tells us the number of source–sink nodes pairs: 0 for the directed cycles, 2 for the mixer and 1 both for the shortcut and diamond cycles. While the shortcut and diamonds cycles both have one pair, they differ by the connection of the nodes in the pair: direct in the shortcut cycle, and indirect in the diamond. We also observe that a diamond can be contracted into a shortcut. However, shortcuts are transitively reducible, while diamonds are not. The contraction of a diamond into a shortcut therefore fundamentally alters its properties, justifying the existence of two distinct classes. The differentiation between shortcuts and diamonds is made absolutely clear when considering the second property we use to organise the generalised directed cycles classes: antichains.

An antichain $\mathcal{A}$ is a mesoscopic structure that can be defined on an ordered set (and therefore on an ordered set that describes the reachability in a graph) and is a subset of elements of an order in which all elements are incomparable. To formally define an antichain, we first note that reachability relationship in any graph gives rise to a preorder, where $u \prec v$ in the preorder if and only if there is a path from $u$ to $v$ in the graph. Conversely, every preorder is the reachability relationship of a graph. However, many different graphs may have the same reachability preorder as each other. For example, reachability in a directed graph gives rise to an equivalence relation for nodes linked by paths in both directions and so the classes of this relation define the strongly connected components. It is easy to see that many topologically different undirected graphs with the same vertex set may have the same equivalence relation, based on the reachability criterion. In the same way, reachability in directed acyclic graphs gives rise to partially ordered sets, where a relation $u \prec v$, which indicates that a directed path from $u$ to $v$ exists, implies that $v \prec u$ cannot occur and that a directed path from $v$ to $u$ does not exist.

Therefore, reflecting upon the order imposed by the reachability criterion, an antichain is a subset of nodes in a graph, such that none of the nodes are pairwise connected with edges or paths:

$$\mathcal{A} = \{u, v \in \mathcal{V} | \forall u, v \in \mathcal{A} : d(u, v) = \infty, d(v, u) = \infty\}, \tag{3}$$

with $d(u, v)$ a length of the shortest path from $u$ to $v$.

We highlight two types of antichains: (i) a maximal antichain is not a proper subset of any other antichain, (ii) a unitary antichain is formed of a single node. The generalised cycles of length four can thus be characterised by their composition in terms of antichains, see Fig. 1(d): directed and shortcut cycles are comprised of four unitary antichains, diamond cycles of two unitary and one non-unitary antichains and the mixer of two non-unitary antichains. The characterisation of the four classes is summarised in Table 1.

All that remains for us to do is to show that these two properties, the antichain structure and node characterisation, are enough to uniquely classify generalised directed cycles of any size. To do so, we will show (i) that a cycle of any size can be grown starting from the class minimal size representatives, 3 for feedback cycles and shortcuts and 4 for diamonds and mixers, while preserving the class properties and (ii) that any cycle of arbitrary length larger than or equal to 4 can be contracted into one and only one of the four classes. From the edge direction manipulation in the directed cycles of size 4, we observe two key facts: first, reversing the direction of an edge bounded by two neutral nodes creates a pair of source/sink nodes and therefore reversing the direction of an edge bounded by a pair of source and sink nodes annihilates that pair and transforms both source and sink nodes into neutral nodes. Second, reversing the direction of an edge bounded by a source or sink node and a neutral node moves the source/sink node along the cycle. These imply that the edge-wedge extension operation preserves the number of source–sink pairs.

The edge-wedge extension operation can thus be used to grow these cycles types to an arbitrary size while conserving the number of source–sink pairs. We thus only need to show that no source–sink node arrangements other than those covered in Table 1 can appear in larger cycles. In a cycle of length four, it is not possible to have more than two source–sink pairs, but growing a mixer of size 4 twice with the edge extension operation can provide one edge with two neutral boundary nodes, which can be reversed to create a new source–sink nodes pair. In general, a cycle of length $L$ can have at most $L/2$ pairs of source–sink nodes. However, we show that any generalised cycle with more than two pairs of source–sink nodes are mixers as they can be contracted into two non-unitary antichains.

We define the EDGE-WEDGE CONTRACTION operation that reduces a neutral wedge into an edge ( ) by removing the neutral node if at most one of the wedge boundary node is a source or a sink, ensuring no source nor sink nodes disappear in the contraction process, thus preserving the number of source–sink pairs as well as their fundamental relative organisation, e.g. a diamond cannot be turned into a shortcut. Moreover, it is clear that iterating the contraction process on

**Table 1**
Each cycle class is uniquely characterised by their number of source and sink pairs combined with their maximal antichain composition. Cycles of any size can thus be attributed a unique class and role based on the characteristics listed in the table below. Cycle classes are here represented in their most contracted forms with the nodes at each end of the dashed edges identified. Directed and Shortcut cycles of any size can be contracted into cycles of size 3, while diamond can be contracted into cycles of 4 nodes and mixers into cycles of size twice the number of source and sink pairs. For Directed, Shortcut and Diamond cycles, dashed lines represent chains of neutral nodes of arbitrary length. For Mixer cycles, dashed lines can represent any combination of neutral nodes and source/sink pairs.

| Class | Number of source/sink pairs | Number of maximal antichains | Interpretation |
|---|---|---|---|
| Directed | 0 | $S$ unitary | Feedback loop |
| Shortcut | 1 (connected) | $S$ unitary | Transitively reducible |
| Diamond | 1 (disconnected) | 2 unitary, 1 or more non-unitary | Resilience of information transfer |
| Mixer | 2 or more | 2 or more non-unitary | Mixing of information |

any cycle with more than two pairs of source–sink nodes leads to a contracted form made of two non-unitary antichains, see Fig. 1e. We consider mixers of any size to fall into the same class as they perform the same function as the size four mixer that is mixing into one sink node information coming from two independent ancestors. This explicitly puts more emphasis on the antichain organisation than on the number of source/sink pairs. Further investigation is needed to discover whether the number of source/sink nodes warrants the introduction of sub-classes for the mixer class, as this might be relevant to the evolution of dynamical processes for example. Thus any generalised cycle can be contracted into a form that falls into one of the four classes we defined.

Each cycle class can be interpreted from a node information processing point of view that underlies our classification. Directed cycles can be seen as feedback/reinforcement of information loops. Shortcut cycles are transitively reducible, reflecting the redundancy of the reducible edge in the context of message modification. Since no nodes exist on the transitive edge, no new information is created and there is no information mixing at the sink wedge of such cycle and therefore removal of this edge leads to no loss of topological information, nor no loss of information processing. Notably, reducible edges also reflect a network's resilience, as they ensure that when longer paths are corrupted, some information is retained through transitive edges. Diamond cycles encode both resilience, as information set from one source has two paths to reach its destination, and diversity of information processing as the intermediary nodes might modify the information they receive. Finally mixers are providing information from two independent sources to each sink.

The characterisation of the four generalised directed cycle classes we define and their interpretation are summarised in Table 1. The illustrations of each class are of their minimal representation: directed and shortcut cycles can be maximally contracted into cycles of length 3, while diamond are of minimal size 4 and mixer cycles have a minimal size of twice their number of source and sink pairs, i.e. size 4 for the most minimal mixer. Directed cycles do not exist in DAGs by definition, and shortcut cycles are killed by transitive reduction, leaving only diamonds and mixer cycles in transitively reduced DAGs, emphasising that these two classes are the key organising structures in DAGs.

We conclude this section by noting that the definitions presented here generalise similar definitions introduced in the context of network motifs, i.e. cycles of a given length. The motifs introduced in [26] can be translated into our nomenclature: a "bi-parallel" motif is a diamond, a "bi-fan" is equivalent to the simplest contracted mixer of length 4, "feed-forward" loop is a triangle/shortcut, and a feed-back motif is a feed-back cycle. We emphasise that there is no notion of path-wedge contraction in motifs, as they are of fixed size: a cycle of a size larger that 4 would be a different motif, whereas we see cycles of all sizes as belonging to the same class, if, after path-wedge contraction, they share the same properties. Such motifs were also studied in DAGs in [27,28], where authors emphasised importance of strict triangles in acyclic graphs.

### 2.3. Minimum cycle bases in undirected graphs

The cycle space of an undirected graph is the vector space of the set of its Eulerian cycles endowed with the symmetric difference, or equivalently with the element-wise addition over $\mathfrak{F}_2$ when representing cycles with an edge-cycle incidence matrix. The result of this operation is another subgraph that consists of edges that appear an odd number of times in the cycles taken as arguments, see [29] for more details. It is well-known that the number of independent cycles of an undirected graph is the rank of the cycle-edge incidence matrix (6), the circuit rank [30], and is equal to:

$$d = E - N + n_c, \tag{4}$$

where $n_c$ is the number of connected components in the graph.

Except in cases where no pairs of cycles overlap, i.e. the cycles are independent, the cycle basis is in general not unique. A strategy to reduce the number of possible cycle bases is to impose constraints on its elements. A simple constraint is to require minimality of the representation of the cycle basis. A Minimum Cycle Basis (MCB) is defined as a cycle basis in which the total length of the cycles in the cycle basis is minimal. While the minimality criterion reduces the possible choice of cycles in the MCB, there is no guarantee that it is unique.

Many types of cycle bases exist, an extensive study of the hierarchy of such bases is given in [31]. In particular it is possible to define Minimal Directed Cycle Bases, see [32]. We nevertheless decided to consider the MCB of undirected cycles underlying DAGs and then characterise them with directionality. Our decision was motivated on the one hand by our general approach to consider a DAGs as a combination of an undirected graph and meta-data encoding directionality, and on the other hand by computational considerations: directed MCB algorithms typically run in $\mathcal{O}(E^3N)$, against $\mathcal{O}(E^2N)$ for MCB, and codes to compute MCB are readily available in standard graph packages, e.g. networkx, which is not the case for Directed MCB algorithms. In particular, we used de Pina's algorithm that finds one type of MCBs in undirected networks, called fundamental cycle bases. We review the algorithm in detail in Appendix A, while here we note that a cycle basis $\mathfrak{C}$ is fundamental if the following holds [33]:

$$\mathcal{C}_\alpha \not\subset \mathcal{C}_\beta \quad \forall \quad \mathcal{C}_\alpha, \mathcal{C}_\beta \in \mathfrak{C}. \tag{5}$$

To compute a fundamental MCB, one considers a minimum spanning tree $\mathcal{T}$ along with the set of edges $\mathcal{E}_B$ of $\mathcal{G}$ that are not present in $\mathcal{T}$ [34], where each fundamental cycle is a path in $\mathcal{T}$ whose endpoints are connected by one $e \in \mathcal{E}_B$.

### 2.4. Transitive reduction of DAGs and minimum cycle basis

In this paper, we are particularly interested in the properties of the MCBs of transitively reduced DAGs. The MCB of a network is not unique, but transitive reduction makes them defined well-enough that their characteristics are robust descriptors of reduced DAGs.

The procedure to obtain the directed cycle images of the MCB of a reduced DAG is illustrated in Fig. 2. We apply $F_{\text{undir}}$ to the transitively reduced DAG $\mathcal{D}_{\text{TR}}$ to obtain the underlying undirected graph $\mathcal{G}_{\text{TR}}$. We then compute an MCB of that graph and apply $F_{\text{dir}}$ to each cycle of the MCB to determine its properties.

Transitive reduction limits the types of generalised cycles that can be present in the cycle basis to diamond and mixers, as all shortcut cycles are removed as they represent transitive closure. This brings a natural interpretation of transitive reduction in terms of information processing on a DAG: transitive reduction removes all structures that do not modify information nor are essential for information transfer in the network. Transitive reduction therefore has a non-trivial effect on the dimension of cycle basis of the undirected graph underlying $\mathcal{D}$ and reduces its dimension following (4). We also note that although the MCB of a transitively reduced DAG will in general be composed of both diamonds and mixers, it is possible to modify an MCB finding algorithm, Horton's algorithm [30], to obtain an MCB which is composed of only diamonds: a Minimal Diamond Basis. The details of this variation of Horton's algorithm is presented in Appendix B.

MCB algorithms are sequential and stochastic in nature, and as MCBs are in general not unique, the MCB obtained will be run-dependent. Another effect of transitive reduction is the reduction of choices for the representative of cycles in the MCB as it greatly reduces the overlaps between cycles, and in some cases, the MCB is even unique, see Fig. 2 for the difference between the MCBs obtained when transitive reduction is employed before applying $F_{\text{undir}}$ and when it is not. To ensure that the MCBs found in reduced DAGs are stable and well-defined representatives of the underlying cycle space and therefore that their properties are true characterisations of the systems under study, we studied the statistics of the MCB statistics obtained for two transitively reduced DAGs model over $n$ runs. We considered four different types of networks of size $N = 300$: two Erdős–Rényi DAGs with edge probabilities $p = 0.1$ and $p = 0.8$, and two Price models with out-degree $m = 3$ and $m = 5$, see Section 3 for precise definitions of these models. We used four different Cycle Basis statistics whose definitions we give in the following subsection: the standard deviation in edge participation $E_p$, the leading eigenvalue $\lambda^{\mathsf{C}}_{\text{max}}$ of the cycle adjacency matrix $\mathbf{M}$ (7), the average cycle balance $\langle b \rangle$ from (9), and the average of cycle stretch $\langle s \rangle$, defined in (11). Fig. 3 shows the mean values and standard deviations of those statistics for each type of network considered. The results vary little for each type of DAG.

We now have a stable and consistent way to represent the cyclic structure underlying directed graphs in general and in particular transitively reduced DAGs. With this, we introduce a series of metrics that can be used to characterise the topological and geometrical properties of MCBs.

### 2.5. Cycle metrics

Defining and finding cycles in reduced DAGs provides a representation of mesoscopic and higher-order structures. We can now explicitly use the meta-data associated with DAGs to characterise cycle bases, and by extension characterise reduced DAGs themselves. We introduce simple, intuitive and yet insightful metrics to characterise cycle bases at three levels: cycle, cycle interactions and cycles embedding in the reduced DAG. Some of these measures can be computed for any cycle basis, while others are specific to DAGs. A summary of the metrics presented in this section can be found in Table 2.

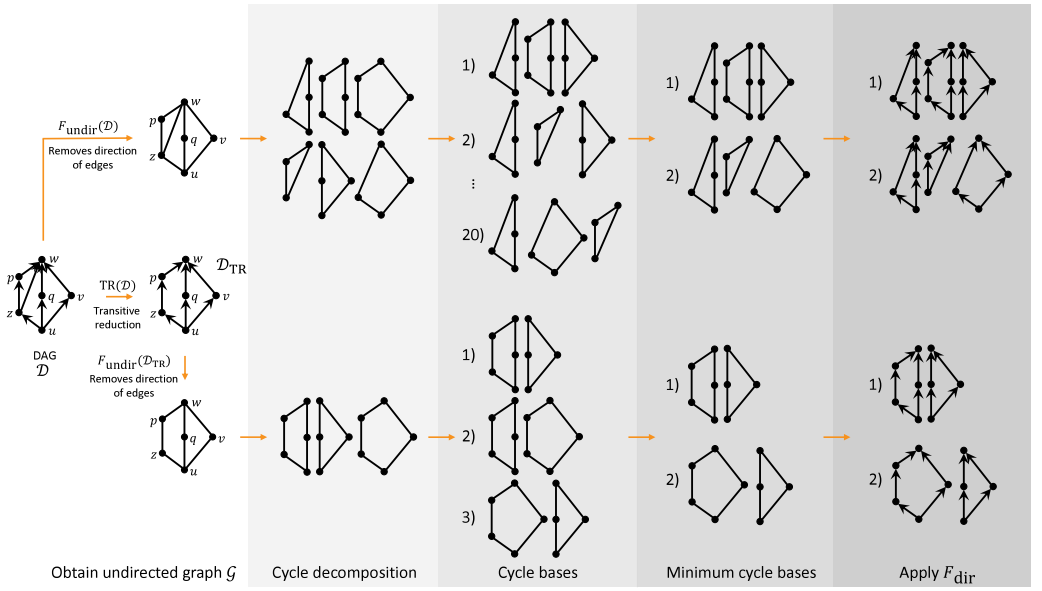


**Fig. 2.** This figure compares the two routes to obtain the directed image of a Minimal Cycle Basis (far right column) of a DAG (far left column). In this paper, we follow the bottom route, that Transitively Reduce the DAGs, with the effect of limiting the directed images of cycles to mixers and diamonds. In large networks, it also has the effect of significantly reducing the dimension of the MCB and stabilising it, making its properties measured by the metrics introduced in Section 2.5 appropriate descriptors to characterise the cyclic structure of DAGs.

*Cycle-edge incidence matrix.* A fundamental object is the undirected edge-cycle incidence vector $c$ for a cycle $\mathcal{C}$

$$c_i = \begin{cases} 1 \text{ if } e_i \in \mathcal{C}, \\ 0 \text{ otherwise.} \end{cases} \tag{6}$$

The information about all cycles can thus be represented by an $E \times d$ edge-cycle incidence matrix $\mathbf{C}$, with $E$ the number of edges and $d$ the number of cycles, see (4), that has the edge-cycle incidence vectors as its row. We can then define the matrix $\mathbf{M}$:

$$\mathbf{M} = \mathbf{C}\,\mathbf{C}^{\mathsf{T}} \tag{7}$$

with entries $M_{\alpha\beta} = |\{e \in \mathcal{E}_\alpha\} \cap \{e \in \mathcal{E}_\beta\}|$, where $\mathcal{E}_\alpha$ is the set of edges of the cycle $\mathcal{C}_\alpha$. Its off-diagonal entries measure the overlap between two cycles and its diagonal elements measure the size of a cycle. This matrix $\mathbf{M}$ can be interpreted in at least two ways: a cycle covariance matrix or a weighted cycle adjacency matrix with self loops. The spectral properties of the Laplacian matrix $\mathbf{L}^C = M - \text{diag}(\mathbf{M})$ are also relevant for characterising cycle interactions and the organisation of cycles: the dimension of the nullspace of $\mathbf{L}^C$, null($\mathbf{L}^C$) is equal to the number of cycle connected components. For a basis where there is no overlap between cycles, this is equal to the dimension of the basis.

*Characteristics of isolated cycles.* Let us introduce some basic cycle statistics. The only purely topological statistic we are going to introduce is the size $S_\alpha$ of a cycle $\mathcal{C}_\alpha$, which is equal to the number of edges, equivalently of nodes, that it comprises:

$$S_\alpha = M_{\alpha\alpha} = |\mathcal{C}_\alpha| = \sum_\beta \mathbf{C}_{\alpha,\beta} = \sum_\alpha c_\alpha. \tag{8}$$

In a Minimum Cycle Basis, the distribution of $S_i$, as well as its mean are minimal by definition. All other metrics are going to be defined using the meta-data associated with the DAGs.

The meta-data allows to study the paths (rather than walks) that constitute cycles. The length of a path $\ell$ is the number of edges between a source and a sink node. A cycle composed of paths that have equal length can be thought to be maximally "balanced", where BALANCE reflects on the variation in the lengths of paths in the cycle. For example the diamond cycles in Fig. 1 are maximally balanced. The balance of the paths contained in a cycle $\mathcal{C}_\alpha$, $b_\alpha$, can be defined using the coefficient of variation:

$$b_\alpha = \frac{\sigma_\alpha(\ell)}{\langle \ell \rangle_\alpha}, \tag{9}$$

where $\langle \ell \rangle_\alpha$ is the average length of the paths in $C_\alpha$ and $\sigma_\alpha(\ell)$ is their standard deviation.

**Table 2**
Measures which characterise an individual cycle (first block), cycle interaction with other cycles (second block), and the network as a whole (third bock) for transitively reduced DAGs.

| Measure | Notation | Explanation |
|---|---|---|
| Cycle size | $S_\alpha = M_{\alpha\alpha} = \lvert\mathcal{C}_\alpha\rvert = \sum_\alpha c_\alpha$ | Size of a cycle. |
| Cycle type | type = *diamond* if $n_{src} = n_{snk} = 1$, *mixer* otherwise. | Classifies cycles into three types, mixers, diamonds, and triangles. |
| Balance | $\frac{\sigma_\alpha(\ell)}{\langle\ell\rangle_\alpha}$ | Variation in lengths of the paths $l$ that make up a cycle. |
| Height | $(1/\lvert\mathcal{C}_\alpha\rvert)\sum_{u\in\mathcal{C}_\alpha} h_u$. | Average height of the nodes comprising a cycle. |
| Stretch | $\max(\{h_u \lvert u \in \mathcal{C}_\alpha\}) - \min(\{h_u \lvert u \in \mathcal{C}_\alpha\})$ | Maximal height difference of the nodes comprising a cycle. |
| Eigenvalues of **M** | $\lambda^C_\alpha$ | "Effective" cycle size: trade-off between the actual cycle size and the sizes of cycles it is adjacent to. |
| Statistics of cycles in MCB | $\langle\cdot\rangle, \sigma(\cdot), \cdot_{\max}$, where $\cdot = S, b, s, h$ | These statistics describe a characteristic cycle within a network; the maximum of characteristics defines an "extremal" cycle. |
| Number of cycle connected components | $\text{null}(\mathbf{L}^C)$ | Number of connected components of cycles. |
| Largest eigenvalue of **M** | $\lambda^C_{\max}$ | The largest effective cycle size. |
| Edge participation | $E_p = \frac{1}{E}\sum_{e\in\mathcal{E}} \lvert\{\mathcal{C}_\alpha \in \mathfrak{C} \lvert e \in \mathcal{E}_\alpha\}\rvert$ | Mean edge participation in cycles. If $E_p$ is small, a network is tree-like, whereas if it is large—it is lattice-like. |
| Variation in $E_p$ | $\sigma(E_p)$ | Variation in edge participation. If there is large variation $\sigma(E_p)$, it indicates that there are denser and less compact cycle regions. |

In a DAG, each node $v$ also has a height $h(v)$, which is defined as the length of the longest path from any source node to $v$ [17]. Heights of nodes can be used to localise cycles them with a "vertical" coordinate. We define the height of a cycle $\mathcal{C}_\alpha$ as the average height of its nodes:

$$h_\alpha = \frac{1}{\lvert\mathcal{C}_\alpha\rvert}\sum_{v\in\mathcal{C}_\alpha} h(v). \tag{10}$$

A cycle in a reduced DAG would always have $\sigma^2(h) > 0$, as they are formed of at least three distinct maximal antichains.

To characterise the relative localisation of a cycle, it is interesting not only to consider its average position in the DAG, but also to capture its spread. We define the stretch of cycle $\mathcal{C}_\alpha$ as the largest difference between the heights of any two nodes in $\mathcal{C}_\alpha$:

$$s_\alpha = \max(\{h_u \lvert u \in \mathcal{C}_\alpha\}) - \min(\{h_u \lvert u \in \mathcal{C}_\alpha\}). \tag{11}$$

If a cycle is treated as a subgraph, the stretch is then simply the maximal height of the cycle. The stretch cannot be smaller than one for mixers and two for diamonds and shortcuts. The largest height in a cycle always belongs to one of the sink nodes, whereas the smallest height always to one of the sources.

*Cycle interactions.* Cycles are adjacent if they share one or more edges. $M_{\alpha\beta}$ measures the overlap, i.e. the number of common edges, between two cycles $\mathcal{C}_\alpha$ and $\mathcal{C}_\beta$, and the diagonal entry $M_{\alpha\alpha} = S_\alpha$ the size of a cycle, as a cycle shares all edges with itself. The spectral properties and eigenvectors of **M** thus reveal the relative organisation of cycles in a DAG: the extent of cycle pairwise overlap/interaction. An in-depth interpretation and analysis of such covariance matrix is beyond the scope of this study, but we note that potentially interesting insights about their interconnectivity of cycles can be obtained from the structure of **M** and that results relating to the spectral properties of covariance matrices apply. In this study, we will look at one metric from the matrix spectra — the largest eigenvalue of **M**, $\lambda^C_{\max}$. A large maximal eigenvalue of a covariance matrix indicates strong variance direction in the data the matrix represents, meaning that a subspace spanned by few eigenvectors of the matrix can be used to describe the data [35].

Another pertinent indirect measure of cycle interactions is the average number of cycles per edge, $E_p$:

$$E_p = \frac{1}{E}\sum_{e\in\mathcal{E}} \lvert\{\mathcal{C}_\alpha \in \mathfrak{C} \lvert e \in \mathcal{E}_\alpha\}\rvert . \tag{12}$$

This quantity is directly related to the "treeness" of a DAG: $E_p = 0$ if and only if there are no cycles in a DAG, in which case the DAG is a tree. Let us qualitatively explain the structure of a DAG for important parameters ranges: for $0 < E_p < 1$, there are edges in the network which do not participate in any cycle and the DAG is still locally a tree and cycles are clustered in branches, like grapes. When $E_p \geq 1$ branches might still exist, but $\sigma(E_p)$ must be considered as well to determine the statistical treeness of a DAG as, on average, each edge participates in one cycle. When $E_p \sim 2$ and $\sigma(E_p) \sim 0$, we find a regime equivalent to a lattice DAG that we will discuss in Section 3.

*Characterisation of a cycle basis.* The statistics of the metrics defined above for individual cycles naturally describe properties of the global network structure. For instance, the standard deviation of the height of cycles indicates how

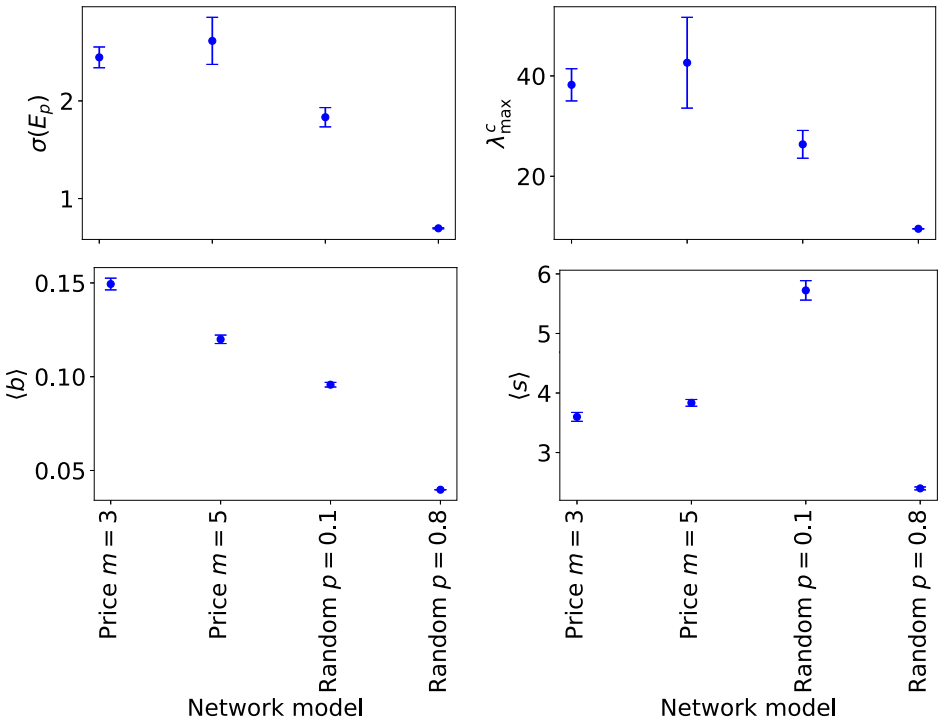


**Fig. 3.** Statistics for the standard deviation in the edge participation $E_p$ (12), the leading eigenvalue $\lambda^C_{\max}$ of the cycle adjacency matrix **M** (7), the average cycle balance $\langle b \rangle$ (9), and the average cycle stretch $\langle s \rangle$ (11), obtained from running De Pina's algorithm $n = 10$ times on four types of networks. The abscissa indicates a network model, the ordinate — obtained values of several statistics of MCBs. The four different models considered are: random DAGs of Section 3.3 with $p = 0.1$ and $p = 0.8$, and the Price model with $m = 3$ as well as $m = 5$. All networks have $N = 300$ nodes.

scattered or lumped together within a network cycles are, larger value indicating that cycles span all heights. On the other hand, a largely stretched-out cycle can have a height close to half of the largest height $h_{\max}/2$, and yet this cycle is arguably different from a cycle whose all nodes have heights close to the average height of a cycle — the latter cycle is much less stretched-out. Combinations of metrics can distinguish such differences between cycles organisation, as we show for two random DAG models in Section 3.

Many properties of a cycle basis are also encoded in **M**. The leading eigenvalue of **M** contains information about the effective size of the largest cycle component, or in variance terms captures most of the interconnectedness of cycles. Intuition can be gained by observing that when two cycles share one or more edges, their "corresponding" eigenvalues shift: one eigenvalue increases, while the other is reduced, the magnitude of the shift corresponding to extent of the overlap. Thus a large eigenvalue of **M** can indicate that it represent a small, highly interconnected cycle, or a large independent cycle: the localisation of the corresponding eigenvector deciding which case it is.

## 3. Transitively reduced DAG network models are characterised by different cycle statistics

We use the transitively reduced versions of four network models to show the usefulness and explanatory power of the metrics described in Section 2.5. To determine the directed images of the MCB underlying the reduced DAG, we follow the procedure detailed in 2.4 and illustrated in Fig. 2. Unless otherwise stated, we considered networks with $N = 500$ nodes, and for stochastic network models, we generated $n = 20$ realisations for each parameter value and computed one MCB for each realisation, see Section 2.3 for a justification to use a single MCB as a representative of the ensemble of MCBs. An example of each type of network is given in Fig. 4. The code used to generate the results reported here can be found on Zenodo [36].

### 3.1. Lattice model

The first network model we discuss is a finite lattice graph turned into a DAG by giving each edge a direction. Specifically, to build our lattice DAG we assign a coordinate $(x_i, y_i)$ to each node $u_i$, where $x_i, y_i$ are non-negative integers less than some fixed size parameter $L$. We then have edges from $u_i$ to two nodes $(x_i + 1, y_i)$ and $(x_i, y_i + 1)$ provided these are allowed coordinates. See Fig. 4 for an illustration. This model is naturally transitively reduced.

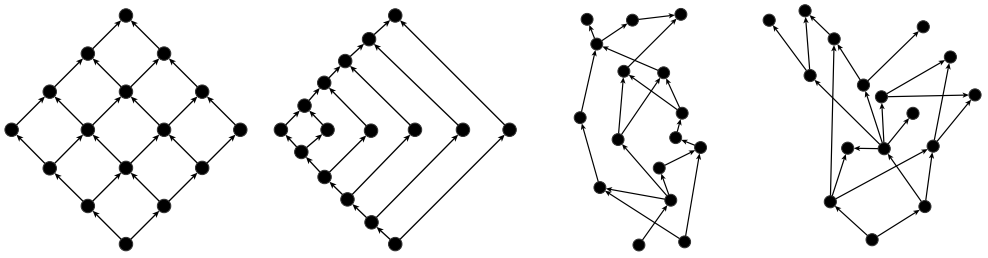

**Fig. 4.** Illustrations of the four network models considered. From left to right: lattice ($L = 4$), Russian doll ($k = 6$), Erdős–Rènyi DAG ($p = 0.5$), Price model ($m = 4, \delta = 0.8$). All networks are transitively reduced, so the cycles seen are those considered in Section 3. In all cases, the direction encoded in the DAGs is such that all arrows point up the page.

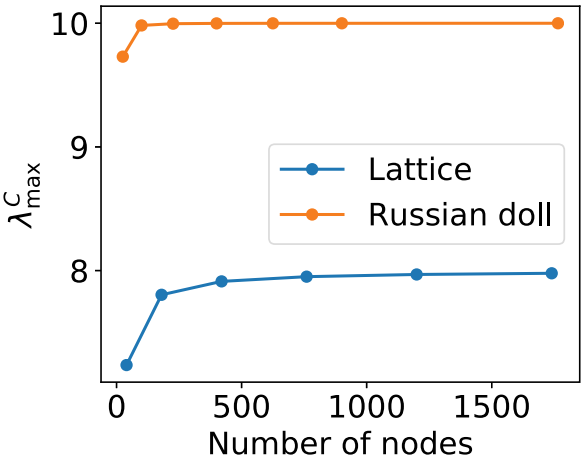


**Fig. 5.** $\lambda^C_{\max}$ as a function of a number of nodes in lattice DAGs (blue), and Russian doll DAGs (orange). Analytical results overlap with numerical results.

The MCB is unique and its features are analytically tractable. Clearly, for our lattice DAG all cycles are diamonds of the same size $C_\alpha = \langle C \rangle = 4$ so there is no variance, $\sigma(C) = 0$, and so our balance statistic (9) of each cycle is also zero, $b_\alpha = 0\ \forall \alpha$. The mean stretch of cycles (11) is always 2. By symmetry, the mean cycle height is half the total height of the lattice DAG, $\langle h \rangle = \ell_{\max}/2$ where $\ell_{\max} = 2(L-1)$. The variation in the height of the cycles, $\sigma(h)$ increases with a network size, since in a lattice all cycles are evenly distributed across a network. The mean edge participation $\lim_{N\to\infty}\langle E_p \rangle = 2$. It is also clear that $\text{null}(\mathbf{L}^C) = 1$.

The eigenvalues of $\mathbf{M}$ for this lattice model can be solved exactly because of its symmetry, see Appendix C. We can show that

$$\lambda_\alpha = 4 + 2\cos\left(\frac{\pi m}{L}\right) + 2\cos\left(\frac{\pi n}{L}\right), \tag{13}$$

with $\alpha = (m-1) + (n-1)(L-1), \quad m, n \in \{1, \ldots, (L-1)\}$.

Therefore the largest eigenvalue of $\mathbf{M}$ converges $\lim_{N\to\infty} \lambda^C_{\max} = 8$ for a large lattice. Fig. 5 shows that $\lambda^C_{\max}$ does approach the infinite lattice value 8 rapidly.

### 3.2. Russian doll

The Russian doll DAG $R_d$ has $N = (1 + 3d)$ nodes, $\{u_0, u_1, \ldots, u_{3d}\}$, and $4d$ edges. We start from $R_0$ which is the trivial DAG with one node and no edges. To grow the Russian Doll DAG $R_d$ we take the DAG $R_{d-1}$ and add the three nodes $u_{3d-2}$, $u_{3d-1}$ and $u_{3d}$. Four new directed edges as also included: $(u_{3d-2}, u_{3d-5})$, $(u_{3d-2}, u_{3d-1})$, $(u_{3d-1}, u_{3d})$, and $(u_{3d-3}, u_{3d})$. This ensures that the Russian doll DAGs $R_d$ are transitively reduced by construction. This construction shows that $u_{3d-2}$ is the source and $u_{3d}$ is the sink node for $R_d$. Since the new edge $(u_{3d-2}, u_{3d-5})$ links the new source node in $R_d$ to the source node in the previous DAG $R_{d-1}$, we see this edge is part of the longest path in $R_d$. Likewise, the new edge from the sink node $u_{3d-3}$ of $R_{d=1}$ to the new sink node $u_{3d}$ of $R_d$ is also part of the longest path in $R_d$. Thus the height of the Russian doll DAG grows by two for each step of this iterations giving us a network height of $2d$ for $R_d$.

Note also how this iterative construction shows that we are adding one more cycle to those already in $R_{d-1}$. For a minimal cycle basis, we can define this to be the cycle $C_d$ formed by the four edges added when creating $R_d$ and two

of the edges added a step earlier when creating $R_{d-1}$, namely $(u_{3d-5}, u_{3d-4})$ and $(u_{3d-4}, u_{3d-3})$. The $d$ cycles defined by this iterative process form the unique Minimal Cycle Basis. The cycles in the MCB of the Russian doll DAG only contains diamond cycles and the properties of the MCB are analytically tractable.

All the cycles in this MCB, except one, have size 6. The 'first' cycle, the one equivalent to $R_1$ containing $\{u_0, u_1, u_2, u_3\}$, has size 4. Thus $\lim_{N\to\infty}\langle S\rangle = 6$, $\lim_{N\to\infty}\sigma(S) = 0$. Each cycle is a diamond with balance (9) $b = 1/2$ except for the first cycle, thus asymptotically, $\langle b\rangle = 1/2$. The cycles are arranged symmetrically so the height of all cycles $h_\alpha$ (10) is identical and equal to half the height of the whole DAG, so $h_\alpha = d$.

However, the stretch (11) will increase with $N$. To see this, consider the extra cycle added when constructing $R_d$ from $R_{d-1}$ discussed above. As this contains the sink and source nodes for $R_d$ the stretch of this new cycle in the MCB is equal to $2d$, the height of $R_d$. By induction we see that the $d$ cycles in this MCB have heights equal to the positive even numbers up to $2d$ so the mean value is simply $(1 + d)$ and grows linearly as the size of the DAG.

Let us move from the statistical features of the Russian doll MCB to its spectral properties. First, it is clear that $\text{null}(\mathbf{L}^C) = 1$. The $\mathbf{M}$ cycles overlap matrix of (7) is a simple tridiagonal matrix for which the eigenvectors and eigenvalues can be found. In particular, the eigenvalues are given by

$$\lambda_\alpha = 6 + 4\cos\left(\frac{2\pi\alpha}{(2d+1)}\right) \tag{14}$$

with $\alpha = 1, 2\ldots, d$, as Fig. 5 shows, see also Appendix D. The largest eigenvalue has the limiting value of $\lim_{d\to\infty}\lambda_1 = \lambda^C_{\max} = 10$. To intuitively understand the origin of this value, we first observe that all but one cycle are of size 6, giving the cycles size contribution. Moreover, the largest eigenvalue of $\mathbf{M}$ gets a contribution of 2 from each overlapping cycles as they share 2 edges in the MCB. Since each cycle, besides the first and last, have two neighbours, we obtain $10 = 6+2+2$.

### 3.3. Erdős–Rényi DAG

The Erdős–Rényi network model [37] is probably the simplest random network model: each pair of nodes are connected with a probability $p$ to give a simple graph. We obtain an Erdős–Rényi DAG similarly to an Erdős–Rényi graph. We assign each node $u$ an unique integer $i_u$, and add a directed edge between each pair of nodes $(u, v)$ with probability $p$, with the direction of each edge going from the node with the smaller-valued integer to the larger-valued index, $i_u < i_v$. In what follows, we will call the transitive reduction of these random DAGs "reduced Erdős–Rényi DAGs".

The statistics of the minimal cycle basis of reduced Erdős–Rényi DAGs are richer than the previous simple models. Given the edge density parameter $p$ it is straightforward to estimate the expected number of cycles of a regular undirected random graph using (4). We note that in Erdős–Rényi DAGs, transitive reduction has the following suppression effect: the larger $p$ is, the more edges will be reduced by transitive reduction: the density of reduced Erdős–Rényi DAGs is monotonically decreasing with $p$, and consequently, there are more cycles at low $p$ than there are at high $p$, as is clear from Fig. 6. For instance, at $p = 1$, we have a complete DAG which after transitive reduction is reduced to a single path that follows the order prescribed by the $i_u$, with one pair of source and sink nodes, all other nodes being neutral. Therefore the reduced graph has the lowest possible density and no cycles.

We can further explain the behaviour we see in the cycle statistics of reduced Erdős–Rényi DAGs. Every cycle found at low $p$ in the reduced Erdős–Rényi DAG is a subgraph of a possible clique in the original undirected Erdős–Rényi graph, the clique which shares the same nodes as the cycle. As $p$ rises, the number of cliques and the size of typical cliques in the Erdős–Rényi graph is increasing so it becomes more likely that the cycle found at low $p$ has become a subgraph of a clique in the Erdős–Rényi graph. However, transitive reduction means any clique is reduced to a single path with no loops. So we can understand the fall in the number of cycles in the reduced Erdős–Rényi DAGs seen in Fig. 6.

In the large $p$ regime, we see that the number of cliques, and their size, become so large that cycles are segregated in the reduced Erdős–Rényi DAG, as indicated by the behaviour of $\text{null}(\mathbf{L}^C)$ in Fig. 7. Since a clique of any size is always transitively reduced to a path element, it does not contain any cycles, and thus does not contribute to the circuit rank of the graph. If a network initially contains many large cliques, we are left with little space where cycles can "form". We expect cycle sizes and the number of edges shared between cycles to reduce as $p$ increases in the reduced DAG.

The largest eigenvalue $\lambda^C_{\max}$ follows that same pattern as $E_p$: large for small $p$ and decrease linearly with $p$, see Fig. 7 (top left). On the contrary, the average cycle size $\langle S_i\rangle$ decreases with $p$, see Fig. 8 (top centre). Taken together, this indicates that the high value of $\lambda^C_{\max}$ is driven by the interconnectedness of the cycles. This is supported when the full eigenspectrum is considered, see Fig. 7. For large $p$ the eigenvalues follow a "flat" distribution, i.e. there is no isolated large eigenvalue, whereas for small $p$ the distribution is steeper, with one significantly dominant eigenvalue. This can be understood if we consider the cycle overlap matrix $\mathbf{M}$ as the adjacency matrix of some effective cycle graph. For small $p$, the matrix is dense with high value entries. If used to describe a broadcast process inherent in the definition of eigenvalue centrality, the process will spread fast, one highly dominant eigenvalue, and will reach equilibrium quickly due to the large gap between dominant and other eigenvalues. For large $p$, while the original network is dense, after transitive reduction we are left with a relatively small and empty cycle matrix $\mathbf{M}$ with mostly low value entries. The Perron–Frobenius theorem tells us that the bound on the largest eigenvalue goes down with increasing $p$.

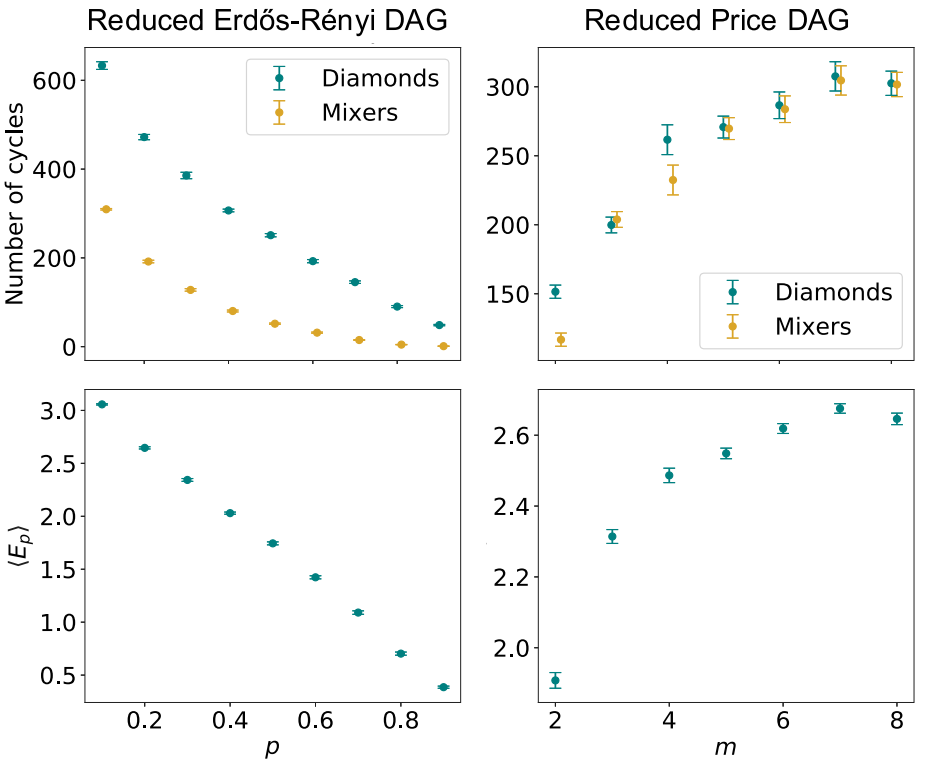


**Fig. 6.** Number of cycles of mixers and diamonds in transitively reduced Erdős–Rènyi DAGs and transitively reduced Price model (top) and the corresponding average edge participation $E_p$ (bottom). We considered networks with $N = 500$ nodes, generated $n = 20$ realisations for each parameter value and computed one MCB for each realisation, and $\delta = m(1+m)$ for the Price Model.

### 3.4. Price model

The Price model is one of the oldest models of citation networks [38] and naturally defines a DAG with a fat-tailed distribution for the number of papers with a given citation count.

The Price model is a growing network model and nodes are labelled with a discrete time $t$ which represents the step at which a node was introduced to a network. The network growth starts from a seed network $\mathcal{D}(t)$. The graph $\mathcal{D}(t+1)$ is obtained by first adding one new vertex, labelled with $(t+1)$. This new vertex is connected to $m$ vertices $\{v\}$ chosen with probability $\Pi(t, v)$ from the set of vertices in $\mathcal{D}(t)$. Our convention is that the edge runs from node $v$ to node $(t+1)$.

In this model, $\Pi(t, v)$ is composed of two terms. With probability $\delta$, the node $(t+1)$ is connected to a vertex $v$ chosen with a probability proportional to the number of edges $k^{\text{out}}(t, v)$ leaving $v$ at the time $t$. Price called this CUMULATIVE ADVANTAGE and, after normalisation, we have that the probability of choosing $v$ is $k^{\text{out}}(t, v)/E(t)$. The second process (random attachment) happens with probability $(1-\delta)$ and in this case we choose the source vertex $v$ uniformly at random from the set of vertices in $\mathcal{D}(t)$, i.e. with probability $1/N(t)$. So the probability of connecting the vertex $(t+1)$ to an existing vertex $v$ is $\Pi(t, v)$ where

$$\Pi(t, v) = \delta \frac{k^{\text{out}}(t, v)}{E(t)} + (1-\delta)\frac{1}{N(t)} \text{ if } t \geq v \geq 1\,. \tag{15}$$

For simplicity, we always chose $\delta$ such that $\delta = m/(1+m)$, the choice originally made by Price.[3] The result of this choice is that for larger values of $m$ the amount of preferential attachment increases, and the amount of random attachment decreases. For an in-depth discussion of Price model see [39]. Let us now consider the minimal cycle bases of the DAGs obtained from this Price model. Our first remark is about the effect of transitive reduction. The number of edges after transitive reduction, $E_{\text{TR}}$, increases with $m$ in the Price model, as Fig. 6 (bottom right) shows. Since edge participation increases with $m$, together with the decrease of the average cycle sizes, we can conclude that networks tend to have an increasing number of cycles with increasing $m$. It is important to note that even for small $m$, $E_p > 1$ indicates that, on

[3] The undirected version of this model with $\delta = \frac{1}{2}$ is the original Barabási–Albert model, see [19] for a discussion.

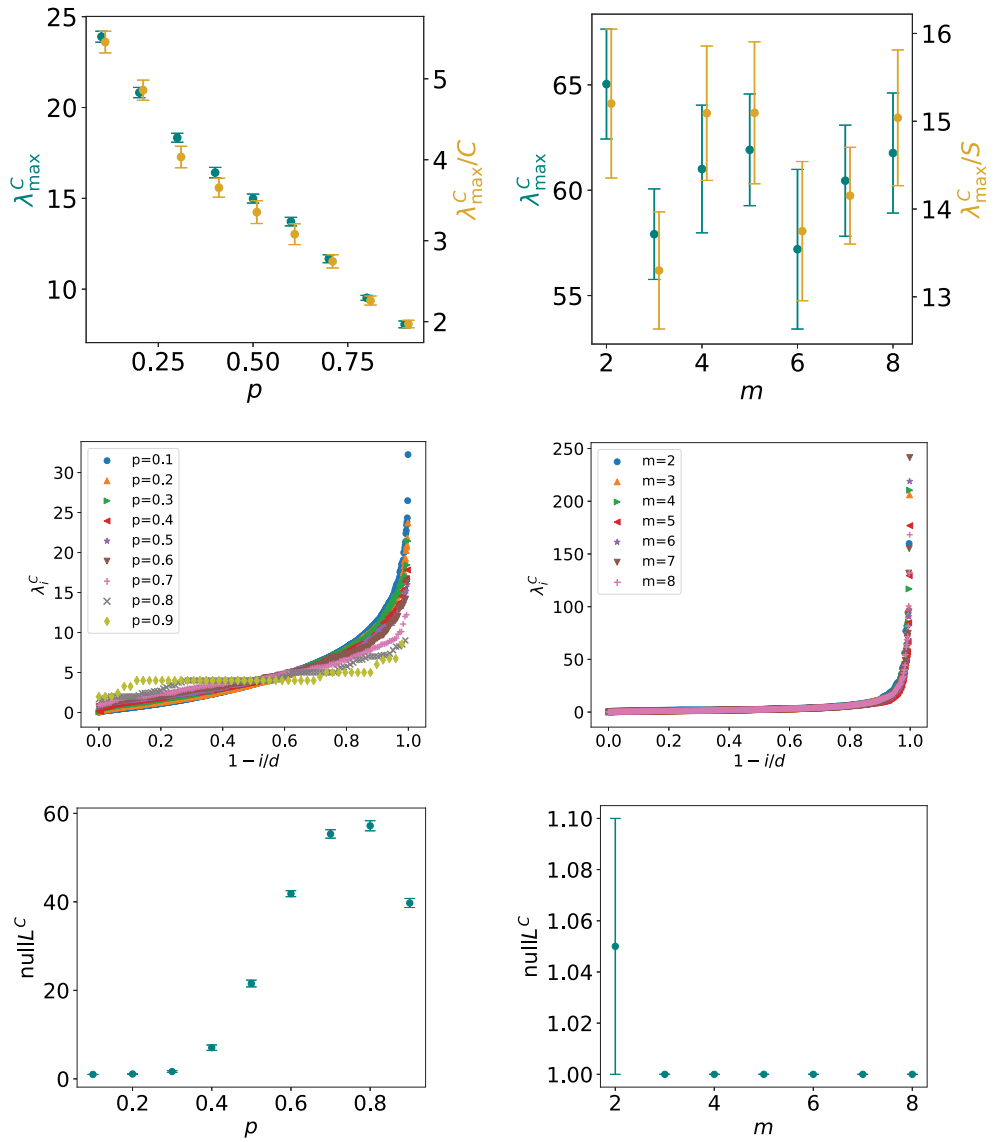


**Fig. 7.** Spectral properties: dominant and normalised dominant eigenvalue, full spectrum for $\mathbf{M}^C$ and size of null$\mathbf{L}^C$ - of matrices derived from the Minimal Cycle Bases of the reduced Erdős–Rènyi DAG model (left) and reduced Price model (right). We considered networks with $N = 500$ nodes, generated $n = 20$ realisations for each parameter value and computed one MCB for each realisation, and $\delta = m(1 + m)$ for the Price model.

average, each edge participates in more that one cycle. Although on average each edge participates in at least one cycle, there are edges which are "more active" than others, as the average value of $E_p$ shows on the bottom right figure of Fig. 6. Thus transitively reduced Price model creates networks that are "more lattice-like" as the density parameter $m$ increases.

Interestingly, the eigenvalues of $\mathbf{M}$ for the reduced Price model DAGs are independent of $m$, contrary to the eigenvalues of the reduced Erdős–Rènyi DAGs which depend on $p$. The largest eigenvalue $\lambda^C_{max}$ also does not fluctuate significantly with $m$. One plausible explanation of this phenomenon is that a large value of $\lambda^C_{max}$ can be a result of either strong interconnectedness of a cycle with other cycles, or its large size. In Fig. 6 we saw that $E_p$ increases with $m$, whereas $\langle C \rangle$ decreases, indicating that cycles tend to be more interconnected, but smaller with increasing $m$. Thus the quasi-stationarity of $\lambda^C_{max}$ indicates that a balance between edge participation $E_p$ and average cycle size $\langle S_i \rangle$ must exist. In Fig. 10 we consider $\lambda^C_{max}$ of Price DAGs with fixed $m$ and varied parameter $c$ which relates to the amount of random attachment of edges. Here we see that as randomness is increased in Price DAGs, the largest eigenvalue $\lambda^C_{max}$ decreases, indicating that the quasi-stationarity between the size and cycle interconnectedness is broken. Since $\langle S \rangle$ varies by a small amount as $c$ is

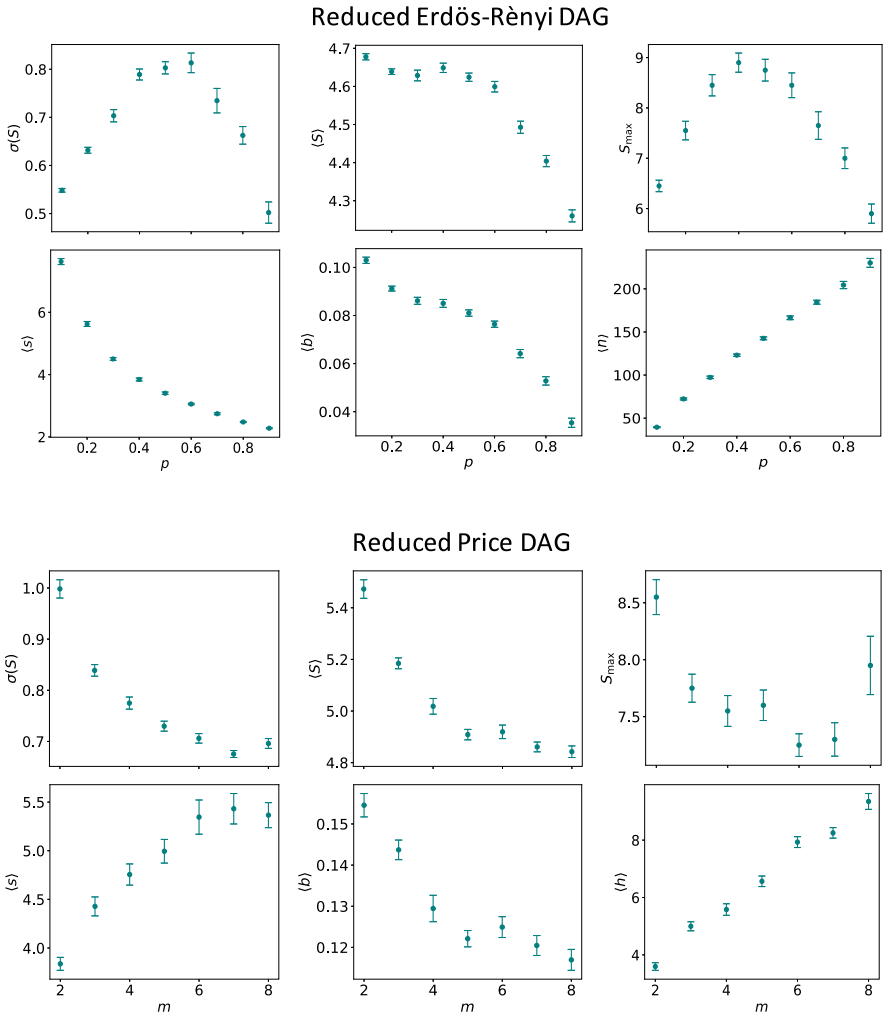


**Fig. 8.** Collected cycle basis statistics — $\sigma(S)$, $\langle S\rangle$, $S_{\max}$, $\langle s\rangle$, $\langle b\rangle$ and $\langle h\rangle$, see Section 2.5 for the definitions — for transitively reduced Erdős–Rènyi DAGs (top panel) and transitively reduced Price model (bottom panel). We considered networks with $N = 500$ nodes, and for stochastic network models, we generated $n = 20$ realisations for each parameter value and computed one MCB for each realisation.

varied, and $\langle h\rangle$ increases, it means that random attachment allows for cycles to occur in more varied areas of a network thereby suppressing $\lambda^{C}_{\max}$.

Finally, we see that in the reduced Price model DAGs, unlike the reduced Erdős–Rènyi DAGs, diamonds are as prevalent as mixers, see Fig. 6.

### 3.5. Comparing DAG models

We now turn to the question of the utility of the metrics to differentiate the MCBs for the DAG models and therefore the model themselves, and particularly their added values with respect to purely topological metrics. To make comparisons fair, the networks produced should have, at least approximately, similar properties. We split our comparison into two groups: deterministic and random DAG models. We match deterministic models on the number of cycles and the random models on the density of edges post transitive reduction, which is a natural choice when trying to understand the effect of the generating mechanisms on the properties and organisation of the MCBs.

The Lattice and Russian doll models have a lot of similarities but are also clearly differentiated by key metrics. While some difference are size independent and fixed features of the models, e.g. $\lambda^{C}_{\max}$ and balance, other are size dependent and the difference will increase with the respective size of the DAGs, e.g. the stretch and height. We note that some of these differences depend on the choice of the root node in the Russian doll model.

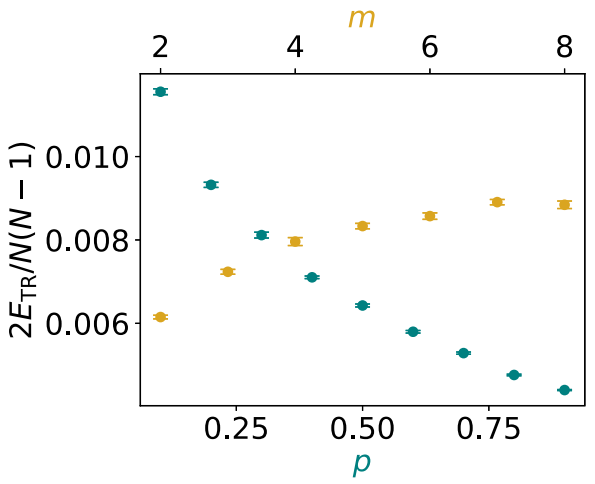


**Fig. 9.** Comparison of the density of reduced DAGs from the Price model (green symbols, top abscissa) and Erdős–Rènyi DAGs (yellow symbols, bottom abscissa), values averaged over $n = 10$ realisations for networks of size $N = 500$ and $\delta = m(1 + m)$ for the Price model. The densities statistically coincide at approximately $p = 0.3$ for Erdős–Rènyi DAGs, and $m = 4$ for Price DAG. We compare MCB statistics for these parameter values in Table 3.

Let us now turn to the random models. To obtain transitively reduced DAGs with similar densities, we first scan through the parameter values $p$ and $m$ to determine a target density value post transitive reduction, see Fig. 9. As the density is computed post transitive reduction, it is a random variable, hence the error bars and the impossibility of matching exactly the densities of the transitively reduced DAGs. The value of the metrics obtained for 1 realisation of each network models at their target density are presented in Table 3. First we remark that the standard errors of the mean are always small comparatively to the value of the average, showing that the MCBs obtained have well-defined characteristics. The MCB of two random models also share similarities with metrics having overlapping standard errors of the mean, such as the maximum cycles size, the stretch. Average cycle size and edge participation are also fairly close. transitively reduced Price DAGs tend to have a balance mixture of diamonds and mixers, while transitively reduced Erdős–Rènyi DAGs have a clear majority of diamonds. Mixers being more prevalent in transitively reduced Price DAGs can be understood from the generating mechanism, as a more intricate structure is due to the attachment constraint. Another very significant difference due to the preferential attachment of the Price model is the much higher interconnectedness of cycles, reflected by a much higher value for the leading eigenvalue of **M**.

The MCB thus reflects specificities of DAG models and the metrics we introduced naturally capture them. As expected, while the metrics are correlated, none is enough to capture the complexity of the organisation of MCBs when considered in isolation to the others. We also conjecture that their usefulness generalises to other DAG models, as well as to real world DAGs, thus providing objects capable not only of characterising DAGs but also differentiating them. Our results also show that pure topological metrics, i.e. not using any meta-data information, are not enough to differentiate the different models and that localising the topological features using meta-data is necessary to uncover differences, similarly to what is done in [15] with persistent homology.

## 4. Conclusion

In this paper, we defined four general classes of directed cycles by considering the underlying undirected graph and augmenting it with the meta-data associated with edge directionality. We then presented a principled way to define, characterise and interpret cycles and Minimal Cycle Bases in Directed Acyclic Graphs (DAGs) by computing an MCB on the underlying undirected graph and augmenting it with the meta-data associated with DAGs. We then focused on transitively reduced DAGs.

We simplified the representation of a DAG via transitive reduction to consider the minimal amount of information needed to retain causal connectivity with the aim to reduce the variance in the composition of the MCB and thus a stable characterisation of network models once enriched with the meta-data. The simplification also gives the potential to only consider specific types of cycle bases based purely on diamonds. Moreover, we have shown numerically that this simplification does not reduce the discriminatory power of the metrics we introduced, as we were able to clearly differentiate and characterise DAG models. Moreover, the comparison between the MCB for the reduced Erdős–Rènyi DAG

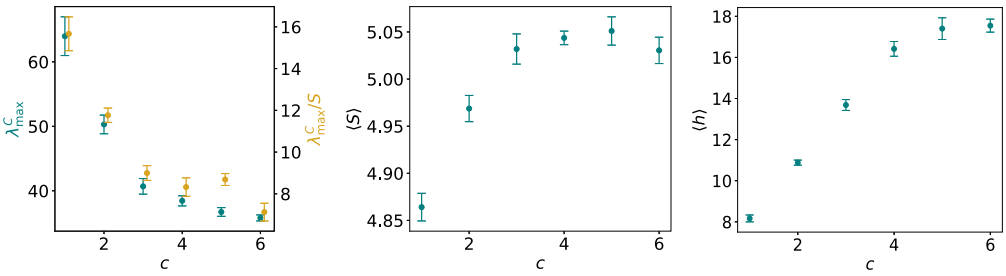


**Fig. 10.** From left to right: dominant and normalised dominant eigenvalue, average cycle size, average cycle height of MCBs of the Price model DAGs generated with fixed $m = 7$ and varied $c = m(1-\delta)$. Here large $c$ values relate to networks in which many edges are attached due to random attachment as opposed to preferential attachment.

**Table 3**
Comparison of the MCB characteristics between the Lattice, Russian doll, the transitively reduced Erdős–Rènyi DAG and transitively reduced Price model; with respective parameters $p = 0.3$ and $m = 3$ for the random models, see Fig. 9. Values for the two random models are averaged over the cycles comprising the MCB. Errors are standard error of the mean over $n$ realisations.

| Measure | Lattice | Russian doll | Random $p = 0.3$ | Price $m = 4$ |
|---|---|---|---|---|
| $\sigma(C)$ | 0 | 0 | $0.7 \pm 0.0(1)$ | $0.8 \pm 0.01$ |
| $\langle C \rangle$ | 4 | 6 | $4.6 \pm 0.0(1)$ | $5.0 \pm 0.04$ |
| $C_{\max}$ | 4 | 6 | $8.5 \pm 0.2$ | $7.6 \pm 0.1$ |
| $\langle s \rangle$ | 2 | $2 + d/2$ | $4.5 \pm 0.04$ | $4.8 \pm 0.1$ |
| $\langle b \rangle$ | 0 | 1/3 | $0.086 \pm 0.0$ | $0.13 \pm 0.00(3)$ |
| $\langle h \rangle$ | $d$ | $\frac{5d}{6} + \frac{1}{3} - \frac{1}{6d}$ | $97.3 \pm 1.4$ | $5.6 \pm 0.2$ |
| #Diamonds | $d$ | $d$ | $385.6 \pm 7.2$ | $261.7 \pm 10.8$ |
| #Mixers | 0 | 0 | $127.9 \pm 2.8$ | $232.5 \pm 8.4$ |
| $E_p$ | 2 | 2 | $2.3 \pm 0.01$ | $2.5 \pm 0.0(2)$ |
| $\lambda^C_{\max}$ | 8 | 10 | $18.3 \pm 0.2$ | $61 \pm 3$ |
| $\lambda^C_{\max}/C$ | 2 | 1.66 | $4.0 \pm 0.1$ | $15 \pm 0.8$ |
| null($\mathbf{L}^C$) | 1 | 1 | $1.65 \pm 0.2$ | $1 \pm 0$ |

and the reduced Price model DAG, and also between the lattice and Russian doll models, clearly shows that topological features are not enough to pinpoint differences, particularly the ones linked to the generating mechanisms, between systems and localising the topological features of cycles.

Some of the metrics we defined are not *reduced* DAG-specific and can be used to characterise any cycle basis, and can thus be applied to any directed network. The measures which are DAG-specific, i.e. that require the notion of order, height, antichain, and longest path, may still be the most interesting: they can literally be used as "coordinates" of cycle, allowing their geometrical embedding and localisation.

## CRediT authorship contribution statement

**Vaiva Vasiliauskaite:** Conceptualization, Methodology, Software, Formal analysis, Investigation, Writing – original draft, Writing – review & editing, Visualization, Funding acquisition. **Tim S. Evans:** Formal analysis, Investigation, Writing – original draft, Writing – review & editing. **Paul Expert:** Conceptualization, Methodology, Formal analysis, Investigation, Writing – original draft, Writing – review & editing, Visualization, Supervision.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgements

This work is supported by the European Union – Horizon 2020 Program under the scheme "INFRAIA-01-2018-2019 – Integrating Activities for Advanced Communities", Grant Agreement n.871042, "SoBigData++: European Integrated Infrastructure for Social Mining and Big Data Analytics" (http://www.sobigdata.eu). Paul Expert is partially supported by NIHR Imperial BRC (grant number NIHR-BRC-P68711).

## Contributions

All authors developed the theoretical concepts. VV and PE designed the experiments discussed in the paper. VV created the software used and performed the data analysis. All authors wrote the manuscript. PE supervised the work. All authors read and approved the final manuscript.

## Appendix A. De Pina's algorithm to obtain a fundamental MCB

De Pina's algorithm (see [40,41]) finds FUNDAMENTAL CYCLE BASES—cycle bases that are computed from a minimum spanning tree of a graph [34]. To obtain a fundamental MCB, a spanning tree of a graph $\mathcal{T}$ is considered along with the set of edges $\mathcal{E}_B$ of $\mathcal{G}$ that are not present in $\mathcal{T}$. A fundamental cycle is defined as a cycle that consists of a path in $\mathcal{T}$ whose endpoints are connected by one $e \in \mathcal{E}_B$. A cycle basis $\mathfrak{C}$ is fundamental if the following holds [33]:

$$\mathcal{C}_\alpha \not\subset \mathcal{C}_\beta \quad \forall \quad \mathcal{C}_\alpha, \mathcal{C}_\beta \in \mathfrak{C} \tag{A.1}$$

De Pina's algorithm begins by initialising a set of support vectors $S_i = \{e_i\}$, one for each edge in $\mathcal{E}_B$, i.e. there is a non-zero entry $e_i$, otherwise the vector is zero. Two procedures are then iterated until a set of $d$ linearly independent cycles of size is obtained.

An iteration $i$ begins with computing a set of vectors $C_\alpha$, each an equivalent representation of an $i$th cycle that is constructed by considering an edge $e_i$ (this step is explained in the following paragraph). From this set of cycles, the shortest cycle $C_\alpha$ for which $\langle C_\alpha, S_i\rangle = 1$ holds is added to the cycle basis. Then each vector $S_j, j > i$ is updated, such that $S_j \to S_j + S_i$ if $\langle C_\alpha, S_j\rangle = 1$. This step ensures that the set of $\{S_j, \ldots, S_d\}$ is orthogonal to $\{C_1, \ldots, C_\alpha\}$ ($\alpha = j - 1$), $C_\alpha$ is orthogonal to the set $\{C_1, \ldots, C_{\alpha-1}\}$.

To find $C_\alpha$, a new graph $\mathcal{G}_\alpha$ is constructed. It contains two copies of each node, $v^+, v^-$ for all $v \in \mathcal{V}$. The resulting network can be thought of as a multilayer network, where nodes are placed in three layers: nodes with positive sign ($v^+$), nodes with negative sign ($v^-$), and nodes that are neutral ($v$). To add edges to $\mathcal{E}_\alpha$, the edgeset of $\mathcal{C}_\alpha$ and therefore the representation of a vector $C_\alpha$, for each edge $e_i = (u, v) \in \mathcal{E}$ we check the following: if $S_i \neq 1$, then add edges $(u^+, v^+)$ and $(u^-, v^-)$ to the edge set of $\mathcal{G}_\alpha$. If $S_i = 1$, then add edges $(u^+, v^-)$ and $(u^-, v^+)$ to the edge set of $\mathcal{G}_\alpha$. Within each layer, we have edges $e_i$ which have $S_i = 0$. Between the layers we have edges $e_j$ which have $S_j = 1$. Any $v^+$ to $v^-$ path in $\mathcal{G}_\alpha$ corresponds to a cycle in $\mathcal{G}$ once positive/negative edges in $\mathcal{G}_\alpha$ are matched to their neutral counterparts in $\mathcal{G}$. If an edge $e \in \mathcal{G}$ occurs multiple times we include it if the number of occurrences of $e$ modulo 2 is 1. In order to find the shortest cycle, we need to find the shortest path from $v^+$ to $v^-$ for all $v \in \mathcal{V}$. There can be more than one such path. The shortest of those paths (and therefore the shortest cycle in $\mathcal{G}$) for which $\langle C_\alpha, S_i\rangle = 1$ is then appended to the current. This concludes an iteration of De Pina's algorithm. The algorithm finishes after $d$ such iterations.

## Appendix B. Description of Horton's algorithm for minimum diamond basis

Horton's algorithm [30] is the first proposed polynomial time algorithm for finding a fundamental MCB. The algorithm first produces $\mathcal{O}(EN)$ of fundamental cycles $\mathcal{H}$, called HORTON CYCLES. Horton showed that $\mathcal{H}$ contains at least one fundamental cycle basis, and the goal of the algorithm is to sift the Horton set to find it.

To find the shortest paths in each tree, Dijkstra's shortest path search algorithm can be used. Then, Gaussian elimination is performed to find a Minimum Cycle Basis from $\mathcal{H}$. A variation of Horton's algorithm was proposed in [42], where authors suggested to replace Gaussian elimination by an iterative construction of a cycle basis. To find an MCB in this way, one uses support vectors in order to retrieve MCB from $\mathcal{H}$. These support vectors are described in description of De Pina's algorithm.

It is possible to adapt Horton's algorithm to DAGs to obtain a cycle basis composed solely of diamonds. To proceed, suppose that we have a directed acyclic graph with a single global source node, a root. Such DAG can be reduced to an arborescence, a directed tree which has one source node that has paths to every other node [12]. If we consider the directionality of edges while building $\mathcal{H}$, all directed paths must originate at a node with a smaller height and end at a node with larger height. In other words, for each $v$ and edge $(u, v)$ a Horton cycle with nodes $SP(v, u), SP(v, w), \{u, v\}$ can be constructed. Such a cycle is always a diamond. There is also no need for building $N$ arborescences (composed of $v$ as a source node as well as its descendants), as each such arborescence is contained within the arborescence grown from the global root. Since we build a single tree, we reduce the size of $\mathcal{H}$ from $\mathcal{O}(NE)$ to $\mathcal{O}(E)$, minimising both temporal and memory demands of the algorithm. Of course, such cycle basis must not be minimum, as it is not able to detect mixers. We will also achieve a heuristic speedup at node level, as the paths are searched for only in its descendant list which is (heuristically) smaller than $N$. Let us call this algorithm for minimum diamond basis (MDB) detection.

## Appendix C. Lattice model spectral properties

Here we give more details on the spectral properties of the Lattice model of Section 3.1. Consider a network which is equivalent to a square lattice of $L$ by $L$ nodes. The MCB is unique and can be visualised as an $(L-1)\times(L-1)$ square 'cycle' lattice (in fact the dual of the original lattice). In this lattice each cycle is a lattice point and each cycle has one edge in common with the cycle represented by the nearest neighbour nodes in this cycle lattice. The dimension of the basis is $(L-1)^2$. We find this we find the $\mathbf{M}$ for $L=5$ is of the form

$$\begin{array}{cccccccccccccccc}
4 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 4 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 4 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 4 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 4 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 4 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 4 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & 4 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 4 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 4 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 4 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 4 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 4 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 4 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 4 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 4
\end{array} \tag{C.1}$$

If we consider the entry $M_{\alpha\beta}$ with indices $\alpha,\beta\in\{0,1,\ldots,(L-2)^2-1\}$ then we can think of label $\alpha$ as representing the cycle placed at $(x,y)$ in the cycle lattice where $x,y\in\{1,2,\ldots,(L-1)\}$ and $\alpha=(x-1)+(y-1)*(L-1)$ so $x,y\in\{1,\ldots,(L-1)\}$. The nodes of the original lattice are then at $(X,Y)$ where $X,Y\in\{1,2,\ldots,L\}$ with the cycle $(x,y)$ running between nodes of the original network at $(X,Y)=\{(x,y),(x+1,y),(x,y+1),(x+1,y+1)\}$. If we write the eigenvectors as $v_\alpha\equiv v_{x,y}$ then the eigenvalue equation is

$$\lambda v_{x,y}=4v_{x,y}+v_{x+1,y}+v_{x-1,y}+v_{x,y+1}+v_{x,y-1}\,, \tag{C.2}$$

with $v_{0,y}=v_{x,0}=v_{L,y}=v_{x,L}=0$. This can be solved by assuming the eigenvectors take the form $v_{x,y}=\sin(\pi mx/L)\sin(\pi ny/L)$ which gives the eigenvalues as

$$\lambda^{(\omega)}=4+2\cos\left(\frac{\pi m}{L}\right)+2\cos\left(\frac{\pi n}{L}\right)\,, \tag{C.3}$$

with $\omega=(m-1)+(n-1)(L-1)$, $\quad m,n\in\{1,\ldots,(L-1)\}$. Note that the solution shows that the constant 4 comes from the number of edges each cycles shared with itself, i.e. the size of each cycle, while each of the four cos factors comes from one edge shared with each neighbouring cycle. Once again we notice in the large system limit, $L\to\infty$, the largest eigenvalue is 8 (the Peron–Frobenius theorem tells us the largest eigenvalue is between five and eight inclusive) with 4 coming from the size of the cycle, and the remaining contribution of four is the number of edges shared with other cycles.

For example, for the square lattice with $L=5$, 4 shows numerical results to three decimal places for the eigenvalues and eigenvectors for cycle overlap matrix $\mathbf{M}$.

Note that the index on the eigenvectors in Table C.1, $\phi$, is assigned in some unspecified order by the numerical routine but is in a one-to-one map with the $\omega$ labels used in (C.3). The eigenvalues are in the row $\lambda^{(\omega)}$ while $\mathbf{e}_\alpha^{(\omega)}$ gives the $\alpha$th entry of the eigenvector $\omega$, that is

$$\lambda^{(\omega)}\mathbf{e}_\alpha^{(\omega)}=\sum_\alpha M_{\alpha\beta}\mathbf{e}_\beta^{(\omega)}\,. \tag{C.4}$$

The symmetry group, $D_4$, ensures that some eigenvalues are repeated in pairs and in that case the eigenvectors given are just two examples from a two-dimensional subspace. Accidental degeneracy is also present, e.g. in the eigenvectors 12 to 15 (see Table C.1).

## Appendix D. Russian doll model spectral properties

Here we give more details on the spectral properties of the Russian Doll model of Section 3.2. Suppose we label the cycles so that $C_d$ is the largest cycle in the MCB (minimal cycle basis) of $R_d$ described in Section 3.2. The matrix $\mathbf{M}$ of (7) describing cycle overlap is a symmetric tridiagonal matrix and here it has a 6 on the diagonal, 2 on the leading (non-zero) off-diagonals, with the exception that $M_{00}=4$ as the first cycle, $C_1$, the only cycle in $R_1$, is the only one of size 4. That is

**Table C.1**
Numerical results for eigenvalues of a square lattice with $L = 5$.

| $\phi$ | 0 | 5 | 4 | 2 | 6 | 9 | 12 | 13 | 15 | 14 | 10 | 11 | 3 | 8 | 7 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\lambda^{(\phi)}$ | 7.236 | 6.236 | 6.236 | 5.236 | 5 | 5 | 4 | 4 | 4 | 4 | 3 | 3 | 2.764 | 1.764 | 1.764 | 0.764 |
| $\mathbf{e}_0^{(\phi)}$ | 0.138 | 0.002 | 0.316 | 0.362 | −0.316 | −0.033 | −0.548 | 0.008 | −0.055 | −0.008 | −0.316 | −0.010 | 0.362 | −0.005 | 0.316 | 0.138 |
| $\mathbf{e}_1^{(\phi)}$ | 0.224 | −0.155 | 0.354 | 0.224 | −0.158 | −0.368 | −0 | −0.119 | 0.297 | 0.356 | 0.158 | 0.358 | −0.224 | 0.164 | −0.354 | −0.224 |
| $\mathbf{e}_2^{(\phi)}$ | 0.224 | −0.352 | 0.158 | −0.224 | −0.158 | −0.368 | 0.183 | 0.079 | −0.282 | 0.244 | −0.158 | −0.358 | −0.224 | −0.356 | 0.158 | 0.224 |
| $\mathbf{e}_3^{(\phi)}$ | 0.138 | −0.316 | 0 | −0.362 | −0.316 | −0.033 | 0 | −0.412 | −0.195 | −0.240 | 0.316 | 0.010 | 0.362 | 0.316 | 0 | −0.138 |
| $\mathbf{e}_4^{(\phi)}$ | 0.224 | 0.161 | 0.354 | 0.224 | −0.158 | 0.335 | 0 | 0.119 | −0.297 | −0.356 | 0.158 | −0.348 | −0.224 | −0.152 | −0.354 | −0.224 |
| $\mathbf{e}_5^{(\phi)}$ | 0.362 | 0.002 | 0.316 | 0.138 | 0.316 | 0.033 | 0.365 | −0.087 | 0.337 | −0.236 | 0.316 | 0.010 | 0.138 | −0.005 | 0.316 | 0.362 |
| $\mathbf{e}_6^{(\phi)}$ | 0.362 | −0.316 | 0 | −0.138 | 0.316 | 0.033 | 0 | 0.531 | −0.102 | −0.117 | −0.316 | −0.010 | 0.138 | 0.316 | 0 | −0.362 |
| $\mathbf{e}_7^{(\phi)}$ | 0.224 | −0.355 | −0.158 | −0.224 | −0.158 | 0.335 | −0.183 | −0.079 | 0.282 | −0.244 | −0.158 | 0.348 | −0.224 | −0.351 | −0.158 | 0.224 |
| $\mathbf{e}_8^{(\phi)}$ | 0.224 | 0.355 | 0.158 | −0.224 | −0.158 | 0.335 | 0.183 | 0.079 | −0.282 | 0.244 | −0.158 | 0.348 | −0.224 | 0.351 | 0.158 | 0.224 |
| $\mathbf{e}_9^{(\phi)}$ | 0.362 | 0.316 | 0 | −0.138 | 0.316 | 0.033 | −0 | −0.531 | 0.102 | 0.117 | −0.316 | −0.010 | 0.138 | −0.316 | −0 | −0.362 |
| $\mathbf{e}_{10}^{(\phi)}$ | 0.362 | −0.002 | −0.316 | 0.138 | 0.316 | 0.033 | −0.365 | 0.087 | −0.337 | 0.236 | 0.316 | 0.010 | 0.138 | 0.005 | −0.316 | 0.362 |
| $\mathbf{e}_{11}^{(\phi)}$ | 0.224 | −0.161 | −0.354 | 0.224 | −0.158 | 0.335 | −0 | −0.119 | 0.297 | 0.356 | 0.158 | −0.348 | −0.224 | 0.152 | 0.354 | −0.224 |
| $\mathbf{e}_{12}^{(\phi)}$ | 0.138 | 0.316 | 0 | −0.362 | −0.316 | −0.033 | 0 | 0.412 | 0.195 | 0.240 | 0.316 | 0.010 | 0.362 | −0.316 | −0 | −0.138 |
| $\mathbf{e}_{13}^{(\phi)}$ | 0.224 | 0.352 | −0.158 | −0.224 | −0.158 | −0.368 | −0.183 | −0.079 | 0.282 | −0.244 | −0.158 | −0.358 | −0.224 | 0.356 | −0.158 | 0.224 |
| $\mathbf{e}_{14}^{(\phi)}$ | 0.224 | 0.155 | −0.354 | 0.224 | −0.158 | −0.368 | 0 | 0.119 | −0.297 | −0.356 | 0.158 | 0.358 | −0.224 | −0.164 | 0.354 | −0.224 |
| $\mathbf{e}_{15}^{(\phi)}$ | 0.138 | −0.002 | −0.316 | 0.362 | −0.316 | −0.033 | 0.548 | −0.008 | 0.055 | 0.008 | −0.316 | −0.010 | 0.362 | 0.005 | −0.316 | 0.138 |

general we have a matrix of the form

$$\mathbf{M} = b\mathbf{I} + \begin{pmatrix} -c & a & 0 & 0 & \cdots & 0 & 0 & 0 \\ a & 0 & a & 0 & \cdots & 0 & 0 & 0 \\ 0 & a & 0 & a & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & a & 0 & a \\ 0 & 0 & 0 & 0 & \cdots & 0 & a & 0 \end{pmatrix} \tag{D.1}$$

with $a = 2$, $b = 6$, $c = 2$ in our case. Following [43] we find the eigenvalues satisfy

$$\lambda^{(\omega)} = 6 + 4\cos\left(\frac{2\pi\omega}{(2d+1)}\right), \quad \omega \in \{1, \ldots, d\}. \tag{D.2}$$

where the 6 comes from the size of the majority of cycles.

For example, $d = 4$ we have

$$\mathbf{M} = \begin{pmatrix} 4 & 2 & 0 & 0 \\ 2 & 6 & 2 & 0 \\ 0 & 2 & 6 & 2 \\ 0 & 0 & 2 & 6 \end{pmatrix} \tag{D.3}$$

and we find approximately that the eigenvalues and eigenvectors are

| $\omega$ | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| $\lambda$ | 9.064 | 6.695 | 4.000 | 2.241 |
| $\mathbf{e}_0^{(\omega)}$ | 0.228 | −0.429 | 0.577 | 0.657 |
| $\mathbf{e}_1^{(\omega)}$ | 0.577 | −0.577 | 0.000 | −0.577 |
| $\mathbf{e}_2^{(\omega)}$ | 0.657 | 0.228 | −0.577 | 0.429 |
| $\mathbf{e}_3^{(\omega)}$ | 0.429 | 0.657 | 0.577 | −0.228 |

where $9.064 \approx 6 + 4\cos(2\pi/9)$, $6.695 \approx 6 + 4\cos(4\pi/9)$, $4 = 6 + 4\cos(6\pi/9)$, and $6.695 \approx 6 + 4\cos(8\pi/9)$.

## References


[1] H.A. Simon, The architecture of complexity, in: Facets of Systems Science, Springer, 1991, pp. 457–476.
[2] D. Lane, Hierarchy, complexity, society, in: D. Pumain (Ed.), Hierarchy in Natural and Social Sciences, Springer, Netherlands, Dordrecht, 2006, pp. 81–119.
[3] S. Kojaku, N. Masuda, Core–periphery structure requires something else in the network, New J. Phys. 20 (4) (2018) 043012.
[4] D.S. Bassett, N.F. Wymbs, M.P. Rombach, M.A. Porter, P.J. Mucha, S.T. Grafton, Task-based core–periphery organization of human brain dynamics, PLoS Comput. Biol. 9 (9) (2013) e1003171.
[5] M. Girvan, M.E. Newman, Community structure in social and biological networks, Proc. Natl. Acad. Sci. USA 99 (12) (2002) 7821–7826.
[6] S. Fortunato, Community detection in graphs, Phys. Rep. 486 (3) (2010) 75–174, http://dx.doi.org/10.1016/j.physrep.2009.11.002.
[7] X. Zhang, Z. Ma, Z. Zhang, Q. Sun, J. Yan, A review of community detection algorithms based on modularity optimization, J. Phys. Conf. Ser. 1069 (2018) 012123.
[8] M.A. Javed, M.S. Younis, S. Latif, J. Qadir, A. Baig, Community detection in networks: A multidisciplinary review, J. Netw. Comput. Appl. 108 (2018) 87–111.
[9] Z. Yang, R. Algesheimer, C.J. Tessone, A comparative analysis of community detection algorithms on artificial networks, Sci. Rep. 6 (1) (2016) 1–18.
[10] J. Leskovec, K.J. Lang, M. Mahoney, Empirical comparison of algorithms for network community detection, in: Proceedings of the 19th International Conference on World Wide Web, 2010, pp. 631–640.
[11] A. Lancichinetti, S. Fortunato, Community detection algorithms: A comparative analysis, Phys. Rev. E 80 (5) (2009) 056117.
[12] M. Coscia, Using arborescences to estimate hierarchicalness in directed complex networks, PLoS One 13 (1) (2018) e0190825, http://dx.doi.org/10.1371/journal.pone.0190825.
[13] G. Palla, I. Derényi, I. Farkas, T. Vicsek, Uncovering the overlapping community structure of complex networks in nature and society, Nature 435 (7043) (2005) 814–818.
[14] T.S. Evans, Clique graphs and overlapping communities, J. Stat. Mech. Theory Exp. 2010 (12) (2010) P12037.
[15] G. Petri, P. Expert, F.E. Turkheimer, R.L. Carhart-Harris, D. Nutt, P.J. Hellyer, F. Vaccarino, Homological scaffolds of brain functional networks, J. R. Soc. Interface 11 (101) (2014) 20140873.
[16] F. Battiston, G. Cencetti, I. Iacopini, V. Latora, M. Lucas, A. Patania, J.-G. Young, G. Petri, Networks beyond pairwise interactions: Structure and dynamics, Phys. Rep. 874 (2020) 1–92.
[17] V. Vasiliauskaite, T.S. Evans, Making communities show respect for order, Appl. Netw. Sci. 5 (1) (2020) 1–24.
[18] S. Wasserman, K. Faust, Social Network Analysis: Methods and Applications (Structural Analysis in the Social Sciences), Cambridge University Press, 1994.
[19] M. Newman, Networks: An Introduction, Oxford University Press, 2010.
[20] T. Loach, T. Evans, Ranking journals using altmetrics, in: A.A. Salah, Y. Tonta, A.A.A. Salah, C. Sugimoto, U. Al (Eds.), Proceedings of ISSI 2015 Istanbul: 15th International Society of Scientometrics and Informetrics Conference, Istanbul, Turkey, 29 June to 3 July, 2015, 2015, pp. 89–94.
[21] J. Clough, T. Evans, What is the dimension of citation space? Physica A 448 (2016) 235–247, http://dx.doi.org/10.1016/j.physa.2015.12.053.

[22] A.V. Aho, M.R. Garey, J.D. Ullman, The transitive reduction of a directed graph, SIAM J. Comput. 1 (2) (1972) 131–137, http://dx.doi.org/10.1137/0201008.
[23] J.R. Clough, J. Gollings, T.V. Loach, T.S. Evans, Transitive reduction of citation networks, J. Complex Netw. 3 (2) (2015) 189–203.
[24] T. Kavitha, K. Mehlhorn, Algorithms to compute minimum cycle basis in directed graphs, Theory Comput. Syst. 40 (4) (2007) 485–505, http://dx.doi.org/10.1007/s00224-006-1319-6.
[25] C. Liebchen, R. Rizzi, A greedy approach to compute a minimum cycle basis of a directed graph, Inform. Process. Lett. 94 (3) (2005) 107–112.
[26] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, U. Alon, Network motifs: Simple building blocks of complex networks, Science 298 (5594) (2002) 824–827.
[27] C. Carstens, Motifs in directed acyclic networks, in: 2013 International Conference on Signal-Image Technology & Internet-Based Systems, IEEE, 2013, pp. 605–611.
[28] Z.-X. Wu, P. Holme, Modeling scientific-citation patterns and other triangle-rich acyclic networks, Phys. Rev. E 80 (3) (2009) http://dx.doi.org/10.1103/physreve.80.037101, arXiv:0908.2615v1.
[29] R. Diestel, Graph theory: Springer graduate text GTM 173, in: Springer Graduate Texts in Mathematics (GTM), Springer, New York, 2012.
[30] J.D. Horton, A polynomial-time algorithm to find the shortest cycle basis of a graph, SIAM J. Comput. 16 (2) (1987) 358–366, http://dx.doi.org/10.1137/0216026.
[31] T. Kavitha, C. Liebchen, K. Mehlhorn, D. Michail, R. Rizzi, T. Ueckerdt, K.A. Zweig, Cycle bases in graphs characterization, algorithms, complexity, and applications, Comp. Sci. Rev. 3 (4) (2009) 199–243, http://dx.doi.org/10.1016/j.cosrev.2009.08.001.
[32] R. Hariharan, T. Kavitha, K. Mehlhorn, Faster algorithms for minimum cycle basis in directed graphs, SIAM J. Comput. 38 (4) (2008) 1430–1447, http://dx.doi.org/10.1137/060670730.
[33] M.M. Sysło, On cycle bases of a graph, Networks 9 (2) (1979) 123–132.
[34] C. Liebchen, R. Rizzi, Classes of cycle bases, Discrete Appl. Math. 155 (3) (2007) 337–355, http://dx.doi.org/10.1016/j.dam.2006.06.007.
[35] J. Friedman, T. Hastie, R. Tibshirani, et al., The Elements of Statistical Learning, vol. 1, Springer Series in Statistics, New York, 2001.
[36] V. Vasiliauskaite, Algorithms for studying cycle bases in transitively reduced DAGs, Zenodo (2022) http://dx.doi.org/10.5281/ZENODO.5993845.
[37] P. Erdős, A. Rényi, On the evolution of random graphs, Publ. Math. Inst. Hung. Acad. Sci. 5 (1) (1960) 17–60.
[38] D.J. d. S. Price, The scientific foundations of science policy, Nature 206 (4981) (1965) 233–238.
[39] T.S. Evans, L. Calmon, V. Vasiliauskaite, The longest path in the price model, Sci. Rep. 10 (1) (2020) 1–9.
[40] T. Kavitha, K. Mehlhorn, D. Michail, K.E. Paluch, An $\mathcal{O}(m^2n)$ algorithm for minimum cycle basis of graphs, Algorithmica 52 (3) (2007) 333–349, http://dx.doi.org/10.1007/s00453-007-9064-z.
[41] J.C. de Pina, Applications of Shortest Path Methods (Ph.D. thesis), University of Amsterdam, 1995.
[42] K. Mehlhorn, D. Michail, Implementing minimum cycle basis algorithms, in: S.E. Nikoletseas (Ed.), Experimental and Efficient Algorithms, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005, pp. 32–43.
[43] S. Kouachi, Eigenvalues and eigenvectors of tridiagonal matrices, Electron. J. Linear Algebra 15 (2006).