Relationships among generalized positive feedback loops determine possible community outcomes in plant-pollinator interaction networks

Fatemeh Sadat Fatemi Nasrollahi,[1,*] Jorge Gómez Tejeda Zañudo,[2] Colin Campbell [3] and Réka Albert [1,4,†]

[1]*Department of Physics, The Pennsylvania State University, University Park, Pennsylvania 16802, USA*
[2]*Broad Institute of MIT and Harvard, Cambridge, Massachusetts 02142, USA*
[3]*University of Mount Union, Alliance, Ohio 44601, USA*
[4]*Department of Biology, The Pennsylvania State University, University Park, Pennsylvania 16801, USA*

Attractors in Boolean network models representing complex systems such as ecological communities correspond to long-term outcomes (e.g., stable communities) in such systems. As a result, identifying efficient methods to find and characterize these attractors allows for a better understanding of the diversity of possible outcomes. Here we analyze networks that model mutualistic communities of plant and pollinator species governed by Boolean threshold functions. We propose a novel attractor identification method based on generalized positive feedback loops and their functional relationships in such networks. We show that these relationships determine the mechanisms by which groups of stable positive feedback loops collectively trap the system in specific regions of the state space and lead to attractors. Put into the ecological context, we show how survival units—small groups of species in which species can maintain a specific survival state—and their relationships determine the final community outcomes in plant-pollinator networks. We find a remarkable diversity of community outcomes: up to an average of 43 attractors possible for networks with 100 species. This diversity is due to the multiplicity of survival units (up to 34) and stable subcommunities (up to 14). The timing of species influx or outflux does not affect the number of attractors, but it may influence their basins of attraction.

I. INTRODUCTION

The interactions between the components of complex systems give rise to emergent, collective behaviors; elucidating this relationship has been of great interest to physicists [1–4]. Representation as a network and network-based dynamic modeling are useful tools to understand these emergent properties in such systems [2,3,5]. Ecological systems are composed of many interlinked species that interact directly or through their shared environment. Stable community outcomes are key emergent ecosystem properties. The resilience of such communities, i.e., their propensity to return to their previous state after a stress or perturbation, is of key importance to species survival and has been studied widely in the context of network science (e.g., Refs. [6–9]).

There are various types of interactions among species in ecological systems: competition for common resources, predator-prey interactions, and mutualistic interactions (pollination and seed dispersal). Network science methods have been successfully implemented to characterize the systems involving such interactions. Studies range from the statistical analysis of food webs [10–13], to network models of communities in which species have mutualistic and competing interactions [14], and the use of multilayer networks to incorporate several types of interactions simultaneously [15,16].

Here we focus on plant-pollinator mutualistic interactions that sustain plant communities. Due to a global loss of pollinator species, the composition of plant-pollinator communities is being threatened continuously across the world [17–20]. Since the majority of food crops need pollination to survive [21], the decline in pollination may cause a loss of biodiversity in crop species [22]. The study of plant-pollinator interaction networks and possible stable community outcomes also contributes to the general goal of maintaining species richness and conservation of biodiversity in the ecosystem [23–28].

A plant-pollinator community can be described by a bipartite network in which plants and pollinators are represented by two different types of nodes and their interactions are represented by edges [29]. The structure of these networks has been analyzed using metrics such as connectance (the fraction of all possible links that are realized in a network), degree distribution (the probability distribution of the number of interactions per species), nestedness (which shows how much the interactions among more specialized species are a subset of the interactions among generalist species), centralization (the extent to which a network has central species), and modularity (the extent to which the interactions among species can be organized into modules) [30–35].

The assembly of a new plant-pollinator community was previously modeled by Campbell *et al.* [26] using two key assumptions: First, there is an influx of species from a regional species pool, simulated as recurring invasion attempts; second, species persist due to their interactions with each other. As a result, the regional source pool determines a network

_____
*fxf32@psu.edu
†rza1@psu.edu

of mutualistic interactions that does not necessarily represent a stable community, but a subset of the species may be able to form and sustain a stable community outcome. These possible stable community outcomes represent the attractors (long-term behaviors) of the model. The properties of such possible stable communities in this model have been studied extensively. For instance, the effect of global and local species loss [24,36], invasion of new species [25], and transient species [27] on final community composition have been characterized. The results of this model have been supported by experimental evidence, for instance the potential beneficial effect of the introduction of a generalist plant into an existing plant-pollinator community has been experimentally verified in Russo *et al.* [37].

Here we aim to complement these studies by asking a question that was not a focus of previous analyses: How many alternative stable communities can form from the same regional species pool? Put in technical terms, we aim to determine and understand the repertoire of attractors of mutualistic networks. Previous studies have emphasized the necessity of positive feedback loops for the formation of alternative stable communities [38–40]. Here we focus on the connection between the the stable positive feedback loops and the emergent dynamics and stable community outcomes. We show that these stable positive feedback loops determine survival units of species, whose compositions shape the repertoire of final community outcomes.

In this work we first propose a method to simplify the plant-pollinator interaction networks obtained from the dynamical model of Campbell *et al.* [26], and we show that the loss of fidelity in this simplification is minimal. Then we characterize the attractors of these networks, and examine the relationship between the number of attractors and the number of independent self-sustaining feedback loops in the networks. Last, we show that the relationships between the stable positive feedback loops of these networks determine the number of possible attractors. Understanding these relationships allows for developing control and management strategies for mutualistic networks.

II. BACKGROUND AND METHODS

A. Properties of the Campbell *et al.* model

In the model of Campbell *et al.* [26], plant and pollinator species are represented by two different types of nodes. An interaction between a plant species and a pollinator species means that the pollinators of this species habitually visit the plants. As this interaction affects the survival of both species involved, it is represented as two directed edges of opposite direction. The expectation is that the interaction is mutually beneficial: the plant provides nectar for the pollinator, and the pollinator helps the plant reproduce. This is not always the case as there can be a mismatch between the length of the pollinator's proboscis (l_{po}) and the plant's nectar depth (l_{pl}):

(1) If $l_{pl} > l_{po}$, then the plant is pollinated, but the pollinator cannot take sustenance from it. The time and effort spent on visiting the plant is wasted. This type of interaction is represented as a positive edge toward the plant and a negative edge toward the pollinator.

(2) If $l_{pl} < l_{po}$, then the pollinator feeds from the plant but the plant is not pollinated. This type of interaction is represented as a negative edge toward the plant and a positive edge toward the pollinator.

Due to the scarcity of information on the structure and dynamics of real plant-pollinator systems, Campbell *et al.* use ensembles of prototypical models informed by distributions published in the literature, aiming to uncover general properties of stable communities. There are 36 ensembles (from networks of size 10 with 5 plants and 5 pollinators to size 100 with 50 plants and 50 pollinators). Each ensemble consists of 1000 networks that have the same source pool size and composition, while the degree and characteristic length of each node is different. As commonly observed in real ecological systems, the probability of each plant (pollinator) interacting with n pollinators (plants) follows an exponentially cutoff power law with properties that are taken from the literature [41]. The plants' nectar depths and pollinators' proboscis lengths are taken from skew normal distributions reported in Stang *et al.* [42]. As a consequence of the mismatch informed by these distributions, only about 7.7% of the interactions are mutually beneficial, and the rest are beneficial in one direction and detrimental in the other direction.

In this model it is assumed that each species i is either present with the state $\sigma_i(t) = 1$, meaning that its population abundance is above a threshold value, or absent with the state $\sigma_i(t) = 0$, meaning that its population abundance is below a threshold value at time t. We will also use the more generic terms "active" for $\sigma_i(t) = 1$ and "inactive" for $\sigma_i(t) = 0$. The state of the whole network at a given time t is denoted by the vector $[\sigma_1(t), \sigma_2(t), \ldots, \sigma_N(t)]$, which corresponds to a unique community composition. The model uses synchronous update, meaning that the state of every node is recalculated at multiples of a time step. The state of each node at time $t + 1$ is determined by the state of its regulators at time t via an update function f. Specifically,

$$\sigma_i(t + 1) = f_i(\sigma_j(t)) = H\Big[\sum_j W(j, i)\sigma_j(t) \Big], \qquad (1)$$

where $\sigma_i(t)$ is the state of node i at time t, $\{\sigma_j\}$ are the states of the regulators of node i at time t, and $W(j, i)$ is the weight of the interaction from node j to node i. The update function expresses that for each species to be able to survive at time $t + 1$, the sum of the weights of its active (present) regulators at time t should be larger than zero. We denote this with a step function $H(x)$, which is equal to 1 if $x > 0$ and zero otherwise. In the work of Campbell *et al.*, a weight of -1 is assigned to negative interactions and the weight of positive interactions is set to 4 (see Ref. [26] for more details). This asymmetry reflects that negative interactions are relatively minor in severity. Pollinators that spend time unsuccessfully attempting to feed do not lose their opportunity to feed on other plants, and plants that provide nectar without achieving pollination can be pollinated by other pollinators in a community. The threshold function in Eq. (1) resembles the Hamiltonian of the 2D Ising model in the absence of a magnetic field $H(\sigma) = \sum_j J_{ij}\sigma_i\sigma_j$, in which J_{ij} denotes the interaction strength between a pair of spins (i, j), and σ_k are discrete variables that take binary values. One important distinction of plant-
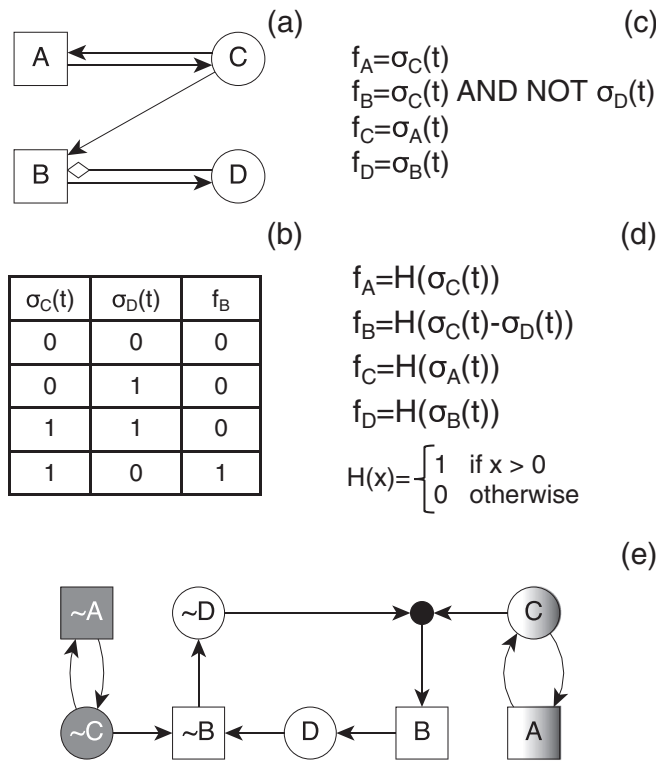
FIG. 1. Illustration of the expanded network that incorporates the update functions, and identification of stable motifs. Panel (a) shows a bipartite network, in which a regular edge represents a beneficial (positive) effect on the target node, while the an edge with a diamond arrowhead represents a harmful (negative) effect. Panel (b) shows the update function of node B in a truth table format. The update function expresses that the simultaneous activity of node C and inactivity of node D results in the activation of node B. Nodes A, C, and D have only one positive regulator and their future state is solely determined by the current state of their positive regulator. Panels (c) and (d) show the same update functions using Boolean operators and threshold functions, respectively. Panel (e) shows the expanded network of the network shown in panel (a). For each node in panel (a), there are two virtual nodes in panel (e) that represent the two possible states of that node: the active state is labeled with the node name, and the inactive state is labeled with the node name prefixed by $\sim$. The composite node that corresponds to the "AND" operator in the functions of node B is shown with a black circle. There are two stable motifs in this network, highlighted with gray (A=0 and C=0) and gray gradient (A=1 and C=1). Locking-in of the gray stable motif yields the system's convergence to a point attractor in which all nodes are inactive. Locking-in of the gray gradient stable motif yields an attractor in which A and C are active, B and D oscillate.

pollinator networks is that certain J_{ij}s are positive (+4 for beneficial interactions) and others negative (-1 for harmful interactions).

B. Attractors, stable motifs, and stochastic updating

In general, Boolean update functions can be specified by truth tables [Fig. 1(b)], Boolean operators [Fig. 1(c)], or threshold functions [Fig. 1(d)]. Every temporal trajectory in a Boolean model reaches a set of states in which it is trapped. This set of states is referred to as the attractor of the Boolean

model. Attractors that consist of a single system state are called point attractors, fixed points, or steady states. In a point attractor the output of each node's update function is the same as the node's state. Attractors that consist of multiple states depend on the manner of update of the nodes. Synchronous update (where all nodes are updated at the same time) leads to deterministic limit cycles, while stochastic update (e.g., updating a randomly selected node at each time) leads to so-called complex attractors.

Since each node i in the interaction network has two states, there are 2^N states for any Boolean model that has N nodes. The exponential dependence of the size of the state space on the number of nodes makes the systematic mapping of the state space computationally expensive. To resolve this problem, we use a novel attractor-finding method that relies on the construction of an expanded network, which integrates the structure of the interaction network and the Boolean functions governing the dynamics [43,44]. The analysis of the expanded network allows for identification of stable subnetworks called stable motifs [45,46]. Stable motifs are the smallest positive circuits (generalized positive feedback loops) that can sustain a specific state regardless of the state of the rest of the nodes outside the stable motif. In plant-pollinator networks each stable motif can be interpreted as the smallest group of species that maintain a specific survival state. Thus, it corresponds to a survival unit, which can either be a subcommunity (if the stable motif expresses the presence of some species) or the simultaneous extinction of all species in the group. Locking-in of a stable motif (i.e., achieving and then maintaining the associated state) confines the system trajectories into a trap space. A trap space is a region of the state space in which (i) a subset of the nodes have a fixed state, and (ii) if the system enters the trap space, it cannot escape [47]. Point attractors are another important type of trap space, in which all the nodes have a fixed state.

The recently introduced most permissive Boolean network represents the most general update framework: By adding two intermediate states, called "increasing" and "decreasing," and instantaneous transitions out of these states, this framework is equivalent to allowing all possible update orders (including partial synchrony among nodes) in the original Booolean network [48]. A major advantage of this framework is that all of its attractors are minimal trap spaces, i.e., trap spaces that do not include a different trap space. Clearly, point attractors are minimal trap spaces, as the state of every node is fixed in a point attractor. An innovation of the most permissive Boolean network is the assurance that attractors made up by multiple states are minimal trap spaces as well, and that a minimal trap space cannot contain more than one attractor [48]. In this work we will implicitly use the most permissive Boolean network framework. We do not explicitly include the two intermediate states, but allow all possible update orders, as these need not to be explicitly invoked to characterize the trap spaces and attractors. Ecologically this means that we make no assumptions about the simultaneity of species influx or outflux. Our analysis does not need to follow the dynamics of the community in time, but rather considers the way in which the successive locking-in of stable motifs restricts the trajectory of the system, and confines it to trap spaces of decreasing size.

Once there are no more stable motifs to lock in, the system reaches a minimal trap space, with the maximum possible number of nodes that acquire a stationary state as a result of locking-in of stable motifs. This minimal trap space is guaranteed to contain a single attractor [48]. If the states of all nodes are determined (1 or 0) within the trap space, then the trap space corresponds to a point attractor. If the states of a set of nodes remain undetermined, then these nodes oscillate and the trap space corresponds to a complex attractor. Since stable motif and trap space based attractor-finding methods do not require the full search of state space, they simplify the attractor-finding task considerably and are more efficient [45,46,49,50].

The expanded network expresses the Boolean update functions of each node in disjunctive prime form. For each node in the interaction network, two virtual nodes are defined in the expanded network which represent the two possible states the node can take: The active state is labeled with the node name and the inactive state is labeled with the node name prefixed by ∼. Also, "AND" operators among two or more node states in an update function are represented using composite nodes. An edge from a virtual node to a composite node means that the node state described by the virtual node is a necessary condition for the node state described by the virtual node that the composite node points to. An edge from one virtual node to another means that the node state described by the first one is a sufficient condition for the node state described by the second one. A stable motif is a strongly connected subgraph (SCS) of the expanded network, meaning that for any pair of virtual nodes there is at least one path in both directions. There are three additional requirements: (i) it is consistent, meaning that it cannot contain a virtual node and its negation; (ii) it is composite-closed, meaning that if it includes a composite node, it also includes all of the input nodes to that composite node as well; (iii) it is a minimal subgraph, meaning that it does not have any subsets that are strongly connected and satisfy the previous two conditions [see Fig. 1(e)]. To find the stable motifs we used the Java implementation of Refs. [45,46] available at Ref. [51] and the python package PyBoolNet [52] available at Ref. [53]. All the executions in this study were ran on a Macintosh desktop PC with 32 GB memory and a 3.2 GHz 8-core Xeon processor.

C. The current state of the art in computing the number of attractors

At present there are no tools that determine the number of attractors of most permissive Boolean networks. The closest applications use general asynchronous update, in which a randomly selected node is updated at each time. The python package PyBoolNet, available at Ref. [53], has several attractor-finding methods. The fastest of these methods uses an answer set programming based algorithm to identify minimal trap spaces [49,52] and approximate each (general asynchronous) attractor with one of the minimal trap spaces. We used the function `trap_spaces()` to find the minimal trap spaces for plant-pollinator networks.

The stable motif based attractor-finding method introduced in Refs. [45,46] follows an iterative approach: After the identification of the stable motifs of a system, one is chosen to

lock-in, and the system is then reduced by tracing the downstream effects of the stable motif in stabilizing other nodes. Next, the reduced system's stable motifs are identified; one is then selected to lock in and the process repeats until there are no remaining stable motifs. The process is repeated for each stable motif and each reduced system until all successions of stable motifs are explored. Each succession ends when there is no other stable motifs to lock in and the system reaches a minimal trap space. This minimal trap space is a point attractor if the system reduced completely at the end of the process. Otherwise, it is assumed to contain at least one complex attractor in which a set of nodes of the final reduced system oscillate (This method assumes general asynchronous update, in which it is possible that more than one complex attractor resides in a minimal trap space). The successive steps in finding the minimal trap spaces is summarized in a succession diagram, in which each succession starts with one of the stable motifs of the original system, followed by the stable motifs of the reduced systems and ends with a minimal trap space. We will refer to this algorithm as "SM analysis 2013."

A new and improved algorithm for stable motif succession diagram-based attractor-finding was recently developed [50]. In addition to the implementation improvements, this algorithm also identifies motif-avoiding complex attractors that reside outside of minimal trap spaces. These complex attractors are rare but possible under asynchronous update, and not possible in the most permissive Boolean network framework. We will refer to this algorithm as "SM analysis 2021."

The method introduced in this work is similar to these previous methods in that it determines minimal trap spaces based on stable motifs (and their weaker counterpart called conditionally stable motif). Our method differs from the stable motif succession-based identification in that we identify the groups of stable motifs that determine each attractor without the necessity to follow all successions of stable motif locking-ins. We compared the performance of the four methods on ensembles of plant-pollinator network models and report the results in the Sec. III G. When comparing to the SM analysis 2021 method we disable the search for motif-avoiding attractors to maintain consistency with the other three methods.

D. Logical domain of influence

To determine the influence of a fixed node state or of a locked-in stable motif on the state of the rest of the components in a system, we use the concept of logical domain of influence (LDOI) [54]. The LDOI of a node state $\sigma_i(t) = \sigma_i$, denoted $\mathrm{LDOI}(\sigma_i)$, is determined by a percolation process on the expanded network [54]. We start from the corresponding virtual node in the expanded network and follow its outgoing edges. If we reach another virtual node, then we add it to the LDOI, because an edge from one virtual node to another represents a sufficient relationship in the expanded network. However, if we reach a composite node, then we only add it to the LDOI if all the virtual nodes with incoming edges to that composite node are already in the LDOI. This search continues until no other new nodes can be added to the LDOI. For instance, in Fig. 1(e), the LDOI of the node state $\sigma_A = 0$ is $\mathrm{LDOI}(\sigma_A = 0) \equiv \mathrm{LDOI}(\sim A) = \{\sim C, \sim B, \sim D\}$,

while the LDOI of the node state $\sigma_A = 1$ is LDOI($\sigma_A = 1$) $\equiv$ LDOI(A) = {C}. The LDOI of a node state set (which can be a stable motif) is determined similarly with the difference of having multiple starting points in the expanded network. For instance in Fig. 1(e), LDOI($\sigma_D = 0, \sigma_C = 1$) $\equiv$ LDOI($\sim$D, C) = {C&$\sim$D, A, B}. After reaching B in this example, we reach D which is in conflict with the node state $\sim$D that is the starting point. Since the LDOI algorithm [54] assumes that the starting point node state is maintained, a contradictory state cannot be in the LDOI; the percolation terminates prior to adding D. In this study, we used the python implementation of this algorithm which is available at Ref. [55].

III. RESULTS

Our objective, finding the stable motifs and attractors of the Boolean threshold models governing the assembly of plant-pollinator communities, requires (i) transformation of the threshold functions of the Campbell *et al.* model to disjunctive prime form, (ii) identification of stable motifs and conditionally stable motifs, and (iii) identifying groups of (conditionally) stable motifs that lead to distinct attractors. Each of these steps necessitated developing efficient methods and algorithms. Each part of the analysis allows insights into the composition of plant-pollinator communities. In this section we alternate between presenting each novel methodology, in the order of their application in the analysis, and the result of their application.

The transformation and simplification of the Boolean threshold functions and the results of this preprocessing step are described in Secs. III A to III C. The construction of the expanded network and identification of the stable motifs and conditionally stable motifs are discussed in Sec. III D. In Secs. III E to III G we determine the relationships among (conditionally) stable motifs, and show that grouping them based on these relationships and enumerating the effective number of mutual exclusivities between these groups correctly identifies the number of attractors. In the final Sec. III H we present the statistical properties of the identified stable community outcomes (attractors of the Boolean models) and discuss the self-sustaining groups of species that contribute to the final stable community outcome.

A. Conversion of threshold functions to disjunctive prime form and finding the prime implicants of the threshold functions

In order to construct the expanded network of plant-pollinator networks described by the Campbell *et al.* model, the threshold functions described in Sec. II A need to be converted to the disjunctive prime form, i.e., the complete sum of prime implicants or Blake Canonical form. We propose a method to find the prime implicants of threshold functions defined as Eq. (1). Our method is equivalent with the MINTRUE procedure of [56], with the difference that our method applies directly to threshold functions that include negative weights. An equivalent formulation of the threshold function in Eq. (1) is

$$f_i(\sigma_{P_k}(t), \sigma_{N_l}(t)) = H\Big[\sum_{P_k} 4\sigma_{P_k}(t) - \sum_{N_l} \sigma_{N_l}(t)\Big], \quad (2)$$

where $\{P_k\}$ are the positive regulators of node i, and $\{N_l\}$ are its negative regulators. Note that the coefficient of 4 for the weight of positive interactions and -1 for the weight of the negative interactions are assigned to be consistent with Campbell *et al.* model, and in principle they can be altered. For each prime implicant of the threshold function f, the minimal condition in Eq. (2) should hold, that is, $H(x) = 1$. Assume N_p is the total number of positive regulators, N_n is the total number of negative regulators, $p_a > 0$ is the number of active positive regulators, n_a is the number of active negative regulators, and n_i is the number of inactive negative regulators. Then the minimal condition becomes

$$4p_a - n_a = 1.$$

Substituting $N_n = n_a + n_i$ in this equation yields

$$n_i = N_n - 4p_a + 1, \quad (3)$$

which is the minimum number of inactive negative regulators in each prime implicant that has a number p_a active positive regulators. Starting from $p_a = 1$ and increasing to $p_a = N_p$, we are able to build all the prime implicants of the threshold functions. Since n_i should be non negative, this process terminates after the first case of having $n_i \leqslant 0$. An example of applying this method is given in Appendix A.

B. Not all species and interactions can establish in the community

Conversion from the threshold functions to the disjunctive prime form of Boolean functions causes the loss of certain edges. There are two mechanisms:

(1) If a node does not receive any positive influences in the network (which represents the regional species pool), then the species is not able to establish. An example of such a node is pl_1 in Fig. 2. All other species whose sole positive influence is from this species are also unable to persist. These species are highlighted with gray in Fig. 2(c).

(2) If a node has at least one positive regulator and three or fewer negative regulators, then its future state is determined by the current state of its positive regulators [$n_i \leqslant 0$ in Eq. (3)]. An example of such node is pl_2 in Fig. 2. The truth table in panel (b) shows that regardless of the state of its three negative regulators, pl_2 is active if po_2 is active, and the negative edges are dropped in the in disjunctive prime form, as seen in Fig. 2(c).

Implementing the conversion from threshold functions to disjunctive normal Boolean functions causes an average of 92.2% of all the negative edges and 63.98% of the positive edges to be dropped. We note that all the mutually beneficial interactions, which are shown as double-positive edges in the network model, are preserved in the conversion. Figure 3 shows the average percentage of the species in the network (species pool) that could not establish. The percentage of these species decreases from 78.7% in the smallest networks to 57.9% in the largest networks. This decreasing tendency is expected since increasing the species pool gives rise to an increased possibility of finding beneficial interactions for each species. Overall, the networks to be analyzed are reduced considerably, as on average 66% of the species in each species pool are found to not be able to establish.
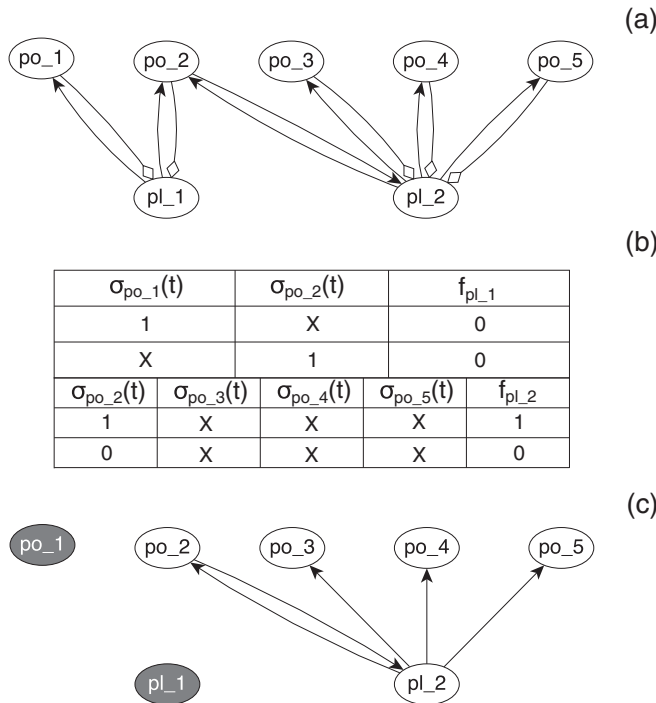
FIG. 2. A small plant-pollinator network depicting the mechanisms of edge loss when converting the threshold functions of the Campbell *et al.* model to disjunctive prime form. Panel (a) shows the bipartite network; plants are denoted with "pl" and pollinators are denoted with "po." Edges with a regular arrowhead show positive influence, while edges with diamond arrowheads show negative influence on the target species. Panel (b) shows the truth tables for pl_1 and pl_2. The "X" in table entries means that the state of that particular species can be either 0 or 1. In this model, pl_1 becomes extinct because neither po_1 nor po_2 are able to pollinate it; this leads to the extinction of po_1. The future state of pl_2 depends only on the current state of po_2, because the weight of this positive influence is larger than the cumulative weight of the negative influences from po_3, po_4, and po_5. These observations allow us to simplify the network, as shown in panel (c), reducing computational complexity. The species that were not able to persist are highlighted with gray.

C. Simplification of the Boolean functions

The disjunctive prime form of a Boolean threshold function can become extensive due to combinatorial explosion. This problem is acute for larger network sizes, where the number of variables in the update functions is larger than 5. To limit the combinatorial complexity, we propose regulator sparsification during the conversion of the threshold functions. We develop a simplification method that reduces the negative regulators by selecting a subset of such regulators of each node randomly. We minimize the reduction in fidelity by having roughly the same probability of $H(x) = 1$ for each node after simplification. The intuitive meaning of this sampling is that the remaining negative regulator(s) of each node are representative of the corresponding species' aggregated costs and wasted opportunities due to interactions with mismatched partners. We assume that any remaining negative regulator can out-compete the positive regulator(s); in other words, the aggregated costs can overcome the effect of positive interac-
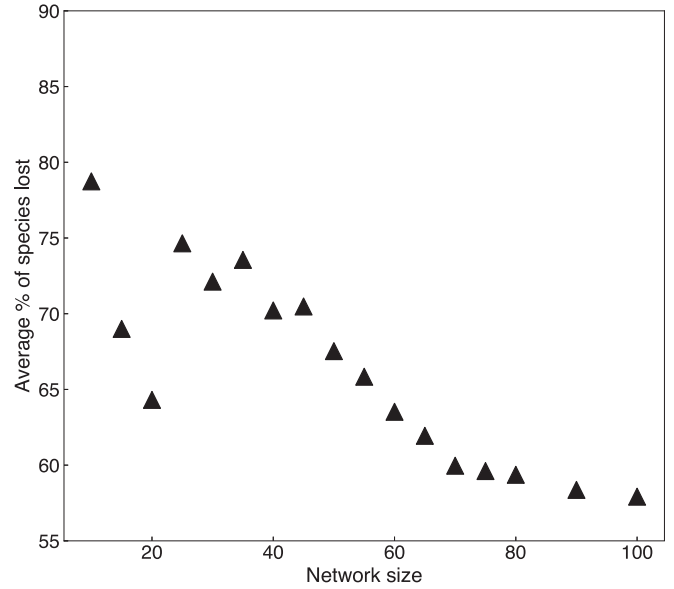

FIG. 3. Average percentage of the species in the original species pool that could not establish in the community for each network size. Examples of such species are depicted in Fig. 2.

tions. This assumption yields an inhibitor-dominant Boolean function.

Assuming that node i has N_p positive regulators and N_n negative regulators, we hypothesize that the Boolean function in disjunctive prime form can be written as

$$f_i = [\sigma_{P_1}(t) \text{ OR} \ldots \text{OR } \sigma_{P_{N_p}}(t)]$$
$$\text{AND NOT } \sigma_{N_1}(t) \ldots \text{ AND NOT } \sigma_{N_x}(t), \quad (4)$$

in which $\{P_k\}$ are the positive regulators, $\{N_l\}$ are randomly sampled negative regulators, and x is the number of negative regulators we keep. We use the conservation of probability of $H(x) = 1$ to calculate x. The probability of $H(x) = 1$ in the original threshold function is

$$p_t = \frac{\sum_{p_a=1}^{N_p} \binom{N_p}{p_a} * \left[\sum_{n_a=0}^{4p_a-1} \binom{N_n}{n_a} \right]}{2^{(N_p+N_n)}}.$$

The probability of having $H(x) = 1$ for the simplified Boolean function is

$$p_b = \frac{2^{N_p} - 1}{2^{(N_p+x)}}.$$

Conservation of probability ($p_t \approx p_b$) requires that

$$\frac{\sum_{p_a=1}^{N_p} \binom{N_p}{p_a} * \left[\sum_{n_a=0}^{4p_a-1} \binom{N_n}{n_a} \right]}{2^{(N_p+N_n)}} \approx \frac{2^{N_p} - 1}{2^{(N_p+x)}}. \quad (5)$$

Equation (5) can be solved to find x, the number of randomly selected negative regulators to be kept in the Boolean function. Considering that $\sum_{p_a=1}^{N_p} \binom{N_p}{p_a} = 2^{N_p} - 1$, one can see that the numerator of the left hand side is always greater than the numerator of the right hand side, which guarantees that $x < N_n$ (for further information see Appendix B 1). The solution to the equation is not necessarily an integer, and hence a choice of direction for the inequality between p_b and p_t is needed.

(1) $p_b \geqslant p_t$: This choice introduces a slight bias toward 1 for p_b, meaning that less number of negative regulators are kept in the Boolean function. In this case, the stable communities will have more species and be more diverse.

(2) $p_b \leqslant p_t$: This choice introduces a slight bias toward 0 for p_b, meaning that more number of negative regulators are kept in the Boolean function. In this case, the stable communities will have less species and be less diverse.

We compared the attractors of simplified Boolean models using each direction of the inequality with the exact Boolean models, as described in Appendix B 1. We found the attractors of both simplified Boolean models show an average of 95% agreement with the attractors of the exact Boolean model. We chose $p_b \geqslant p_t$, which gives slightly more diversity and more rich stable community outcomes. This choice yields an equation one can solve to find the number x for each number of positive and negative regulators. As further validation of the simplification, described in Appendix B 2, we found a 98.8% agreement of the attractors of the simplified and exact Boolean models under perturbations.

1. Summary statistics of the simplified networks

Implementing the proposed simplification of the Boolean functions causes an additional loss of 7.6% (on average) of the original number of negative edges. Altogether, the vast majority (an average of 99.8%) of the original negative edges were dropped in the conversion from threshold functions. The node pairs that are connected by at least one edge fall into three categories:

(1) Nodes connected by a positive feedback loop (mutually beneficial interaction). The edges are positive on both ends. After simplification, on average 31.84% of the connected node pairs are connected by this type of edge pattern.

(2) Nodes connected by a single positive edge: This interaction is beneficial for one party and detrimental for the other; the negative edge was dropped. After simplification an average of 67.76% of the node pairs are connected by this type of edge.

(3) Nodes connected by a negative feedback loop: These are also interactions beneficial for one party and detrimental for the other, and the negative edge remains. After simplification, an average of 0.4% of the node pairs are connected by this type of edge pattern. The negative edges in this category are the only negative edges that are present in the models.

As in the simplified models a very small fraction of the edges is negative, we expect that a majority of the models do not have any negative edges. Indeed, as shown in Fig. S1 in the Supplemental Material [57], we find that 92% of the networks do not contain any negative edges, which significantly simplifies the analysis of such models.

D. Finding the conditionally stable motifs (CSMs) and their supports

Having the update functions in disjunctive prime form, stable motif analysis can be implemented to find the survival units of plant-pollinator network models. A key generalization of the concept of stable motif (SM) is the concept of conditionally stable motif (CSM). Intuitively, a CSM becomes a SM if a specific condition or set of conditions is satisfied. CSMs are identified in the expanded network as strongly connected subgraphs (SCSs) that are not composite-closed (i.e., they contain at least one composite node without containing all of its inputs) but satisfy the consistency criterion (i.e., the set of their virtual nodes and composite node inputs does not contain both a virtual node and its negation) [58]. The set of external inputs of composite nodes inside of the SCS makes up the condition of the CSM. If during the dynamics of the system the locking-in of a SM fixes the external input node(s) of the composite node(s) inside the CSM, then the CSM becomes a SM. For example, in Fig. 4(b) the virtual nodes $\sim$po_2, $\sim$pl_2, $\sim$po_3 and $\sim$pl_3 and two composite nodes form a CSM whose condition is $\sim$pl_1. The locking-in of the SM formed by the virtual nodes $\sim$po_1 and $\sim$pl_1 satisfies the condition of this CSM. We call the group of SMs whose locking-in satisfies the condition of a CSM, the support of the CSM, and the CSM that has a support, a supported CSM. Ecologically, this represents a situation in which the presence or absence of specific species (e.g., a survival unit) enables another survival unit to sustain a survival or extinction state. The interpretation of the previous example is that the absence of pl_1 results in a situation in which all the species in the survival unit of pl_2, po_2, pl_3, and po_3 become extinct in at least one of the stable community outcomes. We developed an algorithm to find the CSMs and their supports, as described in Secs. III D 1 and III D 2.

1. Identification of CSMs by merging consistent cycles

In the expanded network representation, every minimal cycle that satisfies the consistency criterion is called a consistent cycle. Each consistent cycle corresponds to a positive feedback loop in the plant-pollinator interaction network (see Fig. 4). Each consistent cycle that contains composite node(s) is a CSM. However, larger CSMs that contain multiple cycles may also exist. We are especially interested in larger CSMs that have a smaller set of conditions than their constituent cycles because the cycles satisfy each other's conditions. We propose an efficient way to identify such larger CSMs by adapting the concept of cycle graph initially introduced in Ref. [58]. The nodes of the cycle graph are the consistent cycles of the expanded network. A directed edge in the cycle graph indicates that (i) a virtual node in the source cycle satisfies one of the conditions of the target cycle and (ii) the source and target cycles and their conditions are all consistent with each other. For example, in Fig. 4(b), the virtual node $\sim$po_2 is a member of the pink cycle and the condition of the green cycle; thus, the pink cycle satisfies the condition of the green cycle. This relationship is shown in Fig. 4(c) as an edge that starts at the node of the cycle graph that corresponds to the pink cycle and ends at the node that corresponds to the green cycle.

Mutual partial condition satisfaction between two cycles C_1 and C_2 implies that in the expanded network there is a path between any virtual node in C_1 to any virtual node in C_2 and vice versa; meaning that the union of these two cycles is an SCS in the expanded network. For instance, in Fig. 4, the pink and green cycles mutually satisfy (one of) each other's conditions ($\sim$po_2 and $\sim$po_3, respectively). The union of the two cycles is an SCS in the expanded network. Moreover,
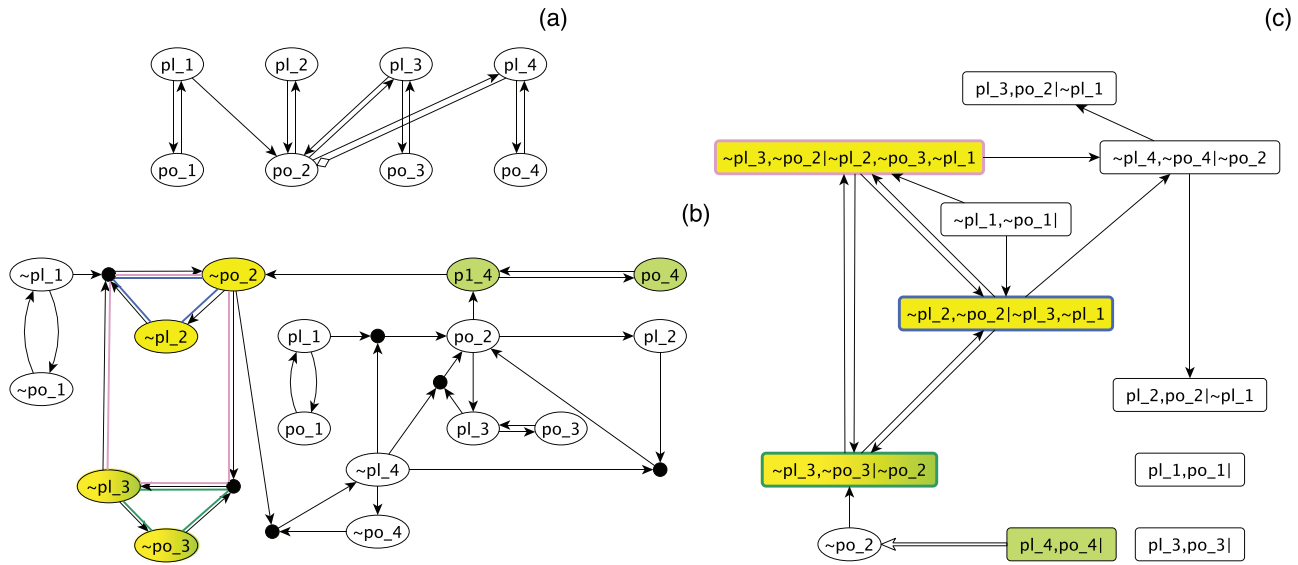
FIG. 4. Finding the CSMs from the consistent cycles. Panel (a) shows an interaction network that has 4 plants and 4 pollinators after simplification. Panel (b) indicates the corresponding expanded network. There are four SMs, each identifiable as a cycle between a virtual node associated to a pollinator and a virtual node associated to a plant. In addition to the SMs, there are several consistent cycles in this expanded network, three of which are highlighted with blue, pink and green lines alongside the edges. Panel (c) shows the cycle graph that corresponds to the expanded network in panel (b). The nodes of this graph (rectangular symbols) represent consistent cycles in the expanded network, and are highlighted with the same colors. The label of each node consists of two parts separated by a vertical line: the first part is the node states in that consistent cycle, and the second part is the conditions of that consistent cycle. Consistent cycles without any conditions are SMs. A directed edge in the cycle graph shows that one or more virtual node of the source cycle satisfies one or more condition of the target cycle. The hollow edge from the light green SM to the virtual node ∼po_2 (which is only added for illustration purposes) represents that ∼po_2 is in the LDOI of the light green SM. The blue, pink and green cycles partially satisfy each others' conditions and make up an SCC in the cycle graph. They also form a consistent SCC in the expanded network in panel (b); as a result, they can be merged to form the yellow CSM, whose condition is ∼pl_1. The green cycle has the condition ∼po_2, which has two ways of being satisfied: one through the yellow CSM, and one through the light green SM. This is because po_2 can become extinct due to the extinction of pl_2, or as a consequence of the inhibitory edge from pl_4. As a result, this cycle not only participates in the yellow CSM but also qualifies as a separate supported CSM.

this SCS is consistent and composite-open, thus it is also a CSM. More generally, a strongly connected subgraph (SCS) in the cycle graph indicates a group of cycles that (partially) satisfy each others' conditions and are pairwise consistent. If all the cycles in the SCS are consistent, then it corresponds to a (C)SM.

We identify CSMs by merging the cycles that form SCSs in the cycle graph. In particular, we identify maximal strongly connected subgraphs (in other words, strongly connected components) in the cycle graph. Each strongly connected component (SCC), when consistent, is a SM if it is composite-closed or a CSM if it is composite-open. As can be seen in Fig. 4(c), the blue, pink and green cycles form an SCC in the cycle graph, and hence are part of an SCC in the expanded network in Fig. 4(b) as well. This SCC highlighted with yellow is consistent and composite-open, thus it is a CSM, with condition (∼pl_1). The virtual node ∼pl_1 is a member of a SM, thus this SM serves as the support of the yellow CSM.

In a network that contains negative edges, it is possible that an SCC of the cycle graph is not consistent and thus it is not a CSM. In this case, we find the pair(s) of cycles in the SCC that are not consistent with each other and connect them with an edge that represents inconsistency. For each node of the SCC that has an inconsistency relationship, we exclude the nodes that it is inconsistent with and determine the SCC

formed by the remaining nodes. Each of these sub-SCCs of the original SCC is kept and recorded as a separate CSM. Due to the scarcity of negative feedback edges in the networks after simplification, there was a single case of inconsistent SCC in the ensemble of 36 000 networks.

Within a consistent SCC in the cycle graph (which is either a SM or a CSM), there may be smaller SCSs or individual cycles whose condition can be satisfied outside the consistent SCC. We describe the identification of CSMs resulting from such SCSs in Sec. III D 3. Finally, we find the single consistent cycles that did not participate in any SCCs. These sole cycles are also separate CSMs and recorded as new entities. An example of such cycles is the consistent cycle consisting of node states ∼pl_4 and ∼po_4 in Fig. 4(c). While this cycle is not participating in the SCC, it is a separate CSM.

2. Finding the support(s) of each CSM

The condition(s) of each CSM can be satisfied directly or indirectly, i.e., through the logical domain of influence (LDOI) of a virtual node (see Sec. II D). There are three specific ways:

(1) The condition can be a virtual node inside a SM or CSM.

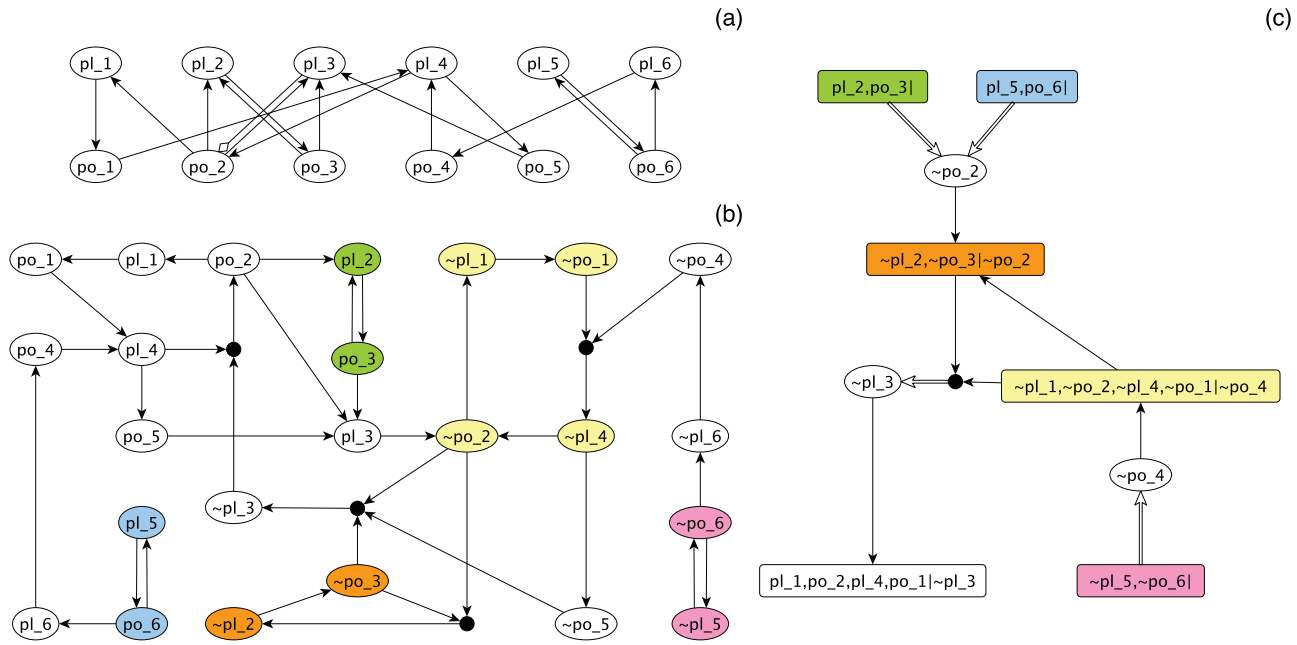(2) The condition can be a virtual node in the LDOI of a SM or CSM.

FIG. 5. Finding the supports of a CSM. Panel (a) shows an interaction network consisting of six plants and six pollinators after simplification. Panel (b) shows the expanded network and panel (c) shows the cycle graph. The orange cycle in panels (b) and (c) is a CSM with the condition ~po_2. This condition can be satisfied using two different paths: The first one is through the LDOI of the blue SM from po_6 to pl_6, po_4, pl_4, po_5, pl_3, and ~po_2. The second is through the yellow CSM, whose condition ~po_4 is satisfied by the LDOI of the pink SM that consists of ~pl_5 and ~po_6. Both the blue SM and the union of the pink SM and yellow CSM are valid supports of the orange cycle. They are mutually exclusive in the state of pl_5 and po_6. We note that the SM made up by pl_2 and po_3 is not a support of the orange cycle: It satisfies its condition but it contradicts its node states. For a similar reason the cycle in the bottom left of panel (c) has no support.

(3) The condition can be a virtual node in the LDOI of a combination of SMs or SMs and CSMs.

Let us consider the situation that we want to find the supports of one particular CSM in an expanded network whose SMs are known. We start with the single SMs in set $M = \{M_1, M_2, \dots\}$ and their LDOIs. If the conditions of the CSM are a subset of $M_i \cup \text{LDOI}(M_i)$ and the node states in the CSM and its conditions are consistent with the node states in $M_i \cup \text{LDOI}(M_i)$, then M_i is a support for the CSM. This continues until all single-motif supports of the CSM are found. For example, in Fig. 5, the blue SM can satisfy the conditions of the orange CSM and is consistent with it, and hence is its single-motif support. Note that the green SM, although satisfies the condition of the orange SM, is not consistent with it, and hence is not a support. Next, combinatorial supports of each CSM are found. Combinations of (C)SMs that are consistent with each other, can satisfy all the conditions of the CSM, and do not include previously found supports are found and added to the list of supports of the CSM. The union of each of these supports and the CSM can be a (member of a) support for another CSM. For example, in Fig. 5, another way of satisfying the condition of the orange CSM, ~po_2, is through the union of the pink SM and the yellow CSM. As a result, this union is a combinatorial support of the orange CSM.

Following the same procedure, all supports of all CSMs are found. It is of particular importance to find all the supports for each CSM, because two supports of the same CSM can be mutually exclusive (i.e., inconsistent with each other because they stabilize the same node(s) to opposite states). Mutually exclusive supports of the same CSM result in two separate trap spaces and consequently two different attractors (see Fig. 5). Since we are interested in finding all the attractors, we need to find all the supports of each CSM. This task is simpler in the majority of plant-pollinator networks that do not have negative edges, as in these networks it is not possible that a CSM has mutually exclusive supports.

3. Identifying intermediate supported CSMs within a consistent SCC

A consistent SCC in the cycle graph has multiple smaller SCSs or individual cycles inside, each of which is what we call an intermediate CSM. The support of these intermediate CSMs can be the same as the support of the SCC, but this is not always the case. We developed an algorithm to find the intermediate CSMs who have their own support(s).

Consider a consistent SCC in the cycle graph. By construction, at least one of the conditions of each cycle in this SCC is satisfied by a virtual node in another cycle of the SCC. This condition may also be satisfied by a SM. We first determine the LDOI of all SMs M_i in the network, and form the union $\mathcal{L} = \cup \text{LDOI}(M_i)$ in which i goes over all indices of SMs in the set M. If the intersection of $\mathcal{L}$ and the virtual nodes inside the consistent SCC is nonempty, then one or more of the virtual nodes in the LDOI of a SM may also serve as condition(s) of intermediate SCSs inside the SCC. As a result, intermediate CSMs inside the SCC might be independently supported by

SMs; we need to identify these SMs. We consider S_i, defined as the intersection of LDOI(M_i) and of the virtual nodes inside the SCC. Each virtual node from each set S_i may be a condition of one or multiple CSMs we want to identify. These CSMs do not contain members of S_i as virtual nodes; instead the members of each S_i are inputs to the composite node(s) of CSMs. This restricts the cycles that participate in the intermediate CSMs we want to identify to cycles that do not intersect S_i. We identify the consistent SCSs and single cycles among these cycles. If there exists a consistent SCS or cycle such that (i) it is supported and (ii) the support is different than the support of the consistent SCC, then it is kept as a separate CSM. For example, in the expanded network in Fig. 4(b) the union of SM LDOIs (the set $\mathcal{L}$) consists of the virtual node ∼po_2, which is also a virtual node of the SCC highlighted with yellow. As a result, there might be smaller CSMs within the SCC that are supported by the SMs from the outside. Eliminating the blue and pink cycles, both of which contain ∼po_2, leaves the green cycle. As the condition of the green cycle is ∼po_2, and its support (the light green SM) is different than the support of the yellow SCC (the SM made up of ∼pl_1 and ∼po_1), the green cycle is recorded as a separate supported CSM. This search continues until all S_is are separately explored and the relevant CSMs are identified.

4. Regularities of the CSM and support structure of plant-pollinator networks

The vast majority of the networks in the plant-pollinator model ensemble do not contain negative edges (see Fig. S1 in the Supplemental Material [57]). In these networks all regulators have a positive effect and the activity of any individual regulator is able to sustain the activity of a target species. In the following we will refer to a (C)SM made up by virtual nodes that express the inactive state of the constituent species as "inactive (C)SM." We determined that in networks with no negative edges all CSMs are inactive, and each of them has a single (possibly combinatorial) support, which also consists of inactive (C)SMs. Because of the unique nature of the support in these networks (i) it is not necessary to search for smaller CSMs within maximal CSMs and (ii) the support identification process can be stopped after the first hit. We derive and illustrate these properties in Appendix D 1.

Analysis of the ensemble of all networks that contain only positive edges after simplification confirms this result. The 30 431 networks in this ensemble collectively have 30 068 CSMs, each of which has a single support. All of the CSMs and their supports express the absence of the constituent species. In summary, in plant-pollinator networks that lack negative edges, the mutualistic interactions lead to stable subcommunities (active SMs), irreversible extinction of certain groups of interdependent species (inactive SMs), and conditional extinction of other groups of interdependent species (inactive CSMs).

As described in Sec. III C 1, following the conversion and simplification of the regulatory functions only 0.4% of node pairs in our ensemble are connected with a negative edge. These negative edges form specific patterns in the interaction network as well as the expanded network. Any node that has an incident negative edge must have at least one incident

positive edge; otherwise, it would have been eliminated during co version from threshold functions to disjunctive prime form. Inhibition is combined with activation via "AND NOT." Furthermore, each negative edge must appear together with a positive edge in the opposite direction. These regularities have strong consequences, as we derive in Appendix D 2.

In these systems it is not possible to have mutual condition satisfaction between a cycle that expresses the activity of the constituent species and a cycle that expresses the inactivity of the constituent species. Networks that contain negative edges may have supported active or mixed-state CSMs. However, the conditions leading to such CSMs did not occur in our ensemble (see Appendix D 2 for details). Finally, in our networks the only situation in which an intermediate CSM has its own support is if it has an active SM in its support, while the cycle graph SCC that contains the intermediate CSM only has inactive SM(s) as support. Only 3.5% of the ensemble of 36 000 networks can have CSMs with more than one support.

E. The number of (C)SMs as well as their relationships determine the number of attractors

As attractors are the minimal possible trap spaces, a first hypothesis could be that the number of SMs and CSMs determines the number of attractors. The ecological intrepretation of this hypothesis is that the number of community outcomes is determined by the number of survival units in the plant-pollinator network. To test this hypothesis, we determined the number of minimal trap spaces with PyBoolNet (as described in Sec. II C). The number of minimal trap spaces is equal to the number of attractors of most permissive Boolean networks. In Fig. 6 we plot the number of minimal trap spaces (attractors) against the number of (C)SMs. This figure indicates that the majority of networks have less than 10 CSMs and attractors. The data point closest to the origin has a single SM and a single attractor. Networks in this category have a single positive feedback loop whose states correspond to one SM and one unsupported CSM, leading to only one attractor. The most frequent result among small networks is two SMs (arising from the two opposite states of the same positive feedback loop) and two attractors. Other highly occurring data points are (4,4), mainly coming from two independent positive feedback loops, and (6,8), arising from three independent positive feedback loops.

This figure shows that there is a large range of the number of attractors possible for the same number of (C)SMs. For example, in the case of 15 (C)SMs the number of attractors can be anywhere between 2 and 21. Due to the high degree of scatter seen in this figure, we conclude that there is no one-to-one relationship among the number of (C)SMs and the number of attractors. To better understand the mapping from motifs to attractors, we identify the functional relationships among SMs and CSMs. We find that there are three types of relationships: dependence, mutual exclusivity and logical determination.

(1) Dependence: If motif M_2 is a CSM and has a condition that can be satisfied by motif M_1, then we say that motif M_2 is dependent on motif M_1.

(2) Logical determination: Motif M_1 logically determines motif M_2 if $M_2 \subset$ LDOI(M_1).
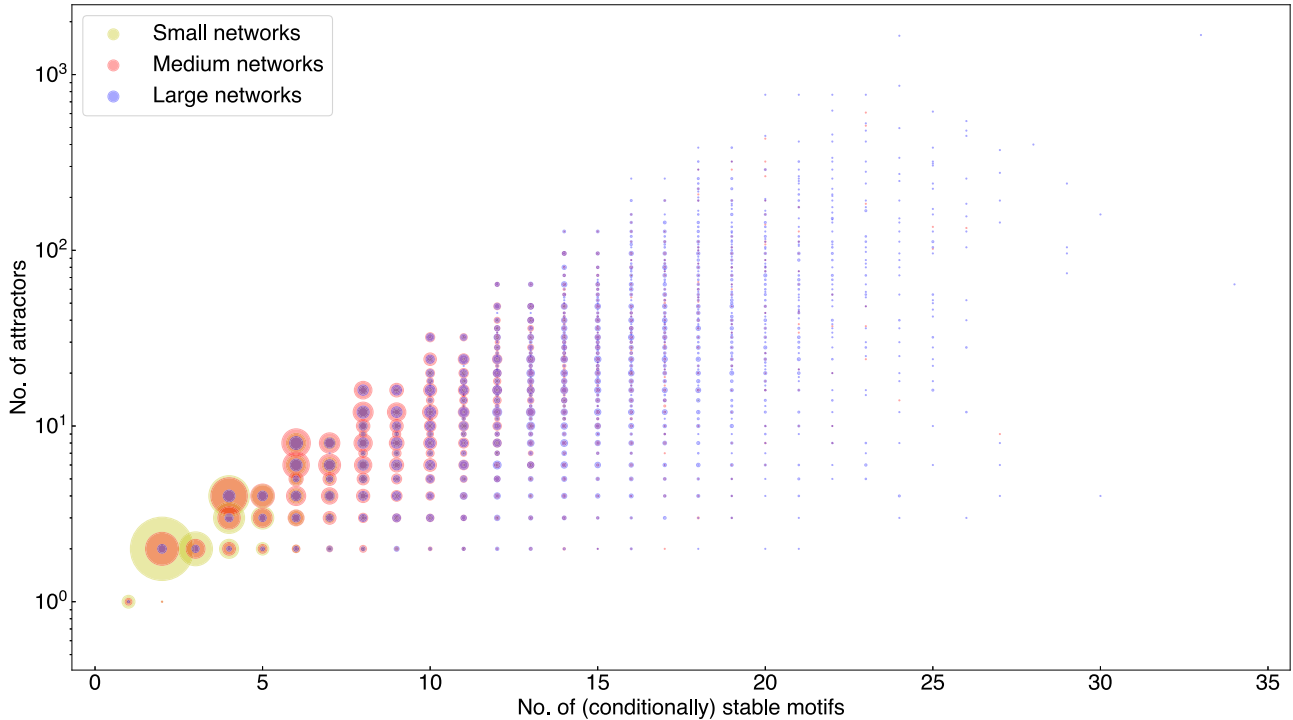
FIG. 6. Scatter plot of the number of attractors against the number of (conditionally) stable motifs. The 36 batches of networks were categorized by size into three groups: small networks (10–35 nodes), medium networks (40–65 nodes), and large networks (70–100 nodes). The area of the data points represents the relative occurrence of each data point.

(3) Mutual exclusivity: Two motifs M_1 and M_2 are mutually exclusive if they share at least one node that is in different states in the two motifs.

These relationships are the underlying mechanisms by which the (C)SMs successively stabilize, confine the system to specific smaller trap spaces in each step, and finally result in attractors. Dependence enforces all motif successions to follow the necessity of having a support of a CSM lock in (achieve and then maintain the corresponding state) earlier within the same succession. For example, in Fig. 5(c), the pink SM satisfies the condition of the yellow CSM, and thus the yellow CSM is dependent on the pink SM to stabilize. As a result, it is not possible to have a motif succession sequence that contains the yellow CSM but not the pink SM. If two (or more) motifs are mutually in the LDOI of each other, then it means they can lock-in in arbitrary order and will trap the system in the same region of state space. Mutual exclusivity enforces that there is no motif succession that contains two motifs that share at least one node but in different states. For example, in Fig. 5, the pink and the blue SMs consist of the same nodes but in opposite states. A succession sequence that would contain both of these SMs would be self-contradictory and hence is not possible. These two SMs correspond to contradictory regions of the state space; they anchor motif successions that lead to at least two separate attractors. This suggests that the most dominant relationship in determining the number of attractors is the mutual exclusivity: distinct attractors result from mutually exclusive motif successions. In the next section we propose a method in which the functional relationships are used to construct the allowed successions that are mutually exclusive and hence lead to distinct attractors.

F. Finding the number of attractors using the functional relationships between SMs and CSMs

All attractors of the networks in our ensemble have a nonempty set of nodes that have a stationary state. The nodes in these sets and their stationary states are determined by the successive locking-in of (C)SMs within the minimal trap space; the stabilized nodes include the nodes of these motifs and their LDOI. The SM succession diagram based attractor-finding process (described in Sec. II C) requires repeated SM based network reduction, which becomes time consuming for large networks. Also, the SM succession diagram discovers and exhaustively lists all the motif successions (often consisting of the same motifs but in different order) that lead to the same minimal trap space. The identification of equivalent motif permutations significantly increases the time necessary for building the succession diagram and for the attractor-finding process.

Here we propose a method that eliminates the need for identifying all the sequences in which SMs can lock in. We determine CSMs and their supports as described in Sec. III D and identify the compatible motif combinations based on the relationships discussed in Sec. III E. We then find the mutually exclusive motif combinations that successfully lead to distinct attractors. A key pillar of this method is the observation that the locking-in of a CSM necessitates the prior locking-in of its support, or if it has multiple supports, one of its supports. Thus, we define a motif group as a CSM merged with one of its supports. If the CSM has more than one support, then there will be separate motif groups, each of which consists of one of the supports and the CSM itself. For instance, in Fig. 7(d), M4
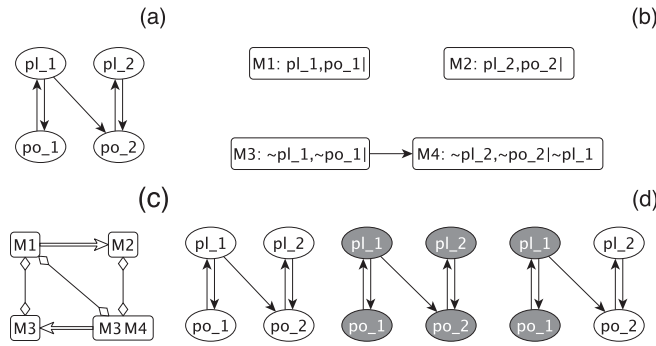
FIG. 7. The functional relationships among SMs and motif groups and construction of the allowed successions to find the number of attractors. Panel (a) shows an example subnetwork consisting of 2 plants and 2 pollinators. Panel (b) shows the cycle graph for the interaction network in (a). Three consistent cycles M1, M2, and M3 are SMs, while the last consistent cycle M4 is a CSM with the condition of ~pl_1, which is satisfied by M3. Thus, motifs M3 and M4 are merged into the motif group M3M4. Panel (c) illustrates the network of functional relationships among the two SMs M1 and M2 and the motif group M3M4. M2 is in the LDOI of M1, and by definition M3 is in the LDOI of the motif group M3M4. These relationships are shown by hollow arrows from M1 to M2 and from M3M4 to M3. M3M4 and M1, M3 and M1, as well as M3M4 and M2 are mutually exclusive (they have shared nodes but in different states), which is represented as diamond arrowheads between these nodes. Following the construction of this network, the LDOI of each SM and maximal motif group is calculated to determine if any of them lead to a point attractor. As a result of stabilization of M1 or M3M4, all the nodes in the interaction network stabilize, and since they are mutually exclusive, each leads to a separate attractor. The first and second figures from left in panel (d) represent the attractors resulting from stabilization of M1 and M3M4 respectively. The nodes that are in gray (white) represent the extinction (persistence) of the corresponding species in that specific attractor. In one of these attractors all species persist and in the other they all become extinct. There also exists a combination of motifs, {M2, M3}, which stabilizes all nodes and excludes both previously found attractors, and hence it leads to a new attractor. The last figure in panel (d) shows this attractor in which half the species persist and the other half become extinct.

is a CSM whose support is M3. M4 is merged with its support to create the merged motif group M3M4.

This merging step implements the first functional relationship, dependence. The other two functional relationships, logical determination and mutual exclusivity, are then determined between all SMs and all the newly constructed motif groups. The definition of logical determination and mutual exclusivity applies the same way to motif groups as to single motifs. By definition, there is a logical determination edge from each motif group to every constituent SM or more elementary motif group.

The functional relationships among SMs and motif groups are illustrated in the form of a network in Fig. 7(c). In this network, each node represents either a motif group (a merger of a CSM with one of its supports) or a SM, and each edge represents a specific type of functional relationship. Logical determination is shown by hollow arrows and mutual exclusivity is shown by diamond arrowheads. For instance, in Fig. 7(c),

M2 is in the LDOI of M1, meaning that if M1 stabilizes, M2 automatically stabilizes as well. This is shown by a hollow arrow from M1 to M2. Also, M3M4 and M1 or M3 and M1 or M3M4 and M2 have shared nodes but in different states, which results in diamond arrowheads between these nodes.

Having constructed this network, we calculate the LDOI of each SM and motif group. This step is to find the SMs and motif groups that stabilize the whole network and lead to point attractors; in other words if all the nodes in the interaction network attain a fixed state as a result of stabilization of a SM or motif group and its LDOI, then it leads to a point attractor. The only motif groups that might stabilize the whole network are the maximal motif groups, i.e., motif groups that are not subgroups of other motif groups. As a result, SMs and maximal motif groups are analyzed to find the ones that lead to immediate point attractors. Each point attractor that differs from all previously recorded point attractors is to be recorded as a new distinct attractor. In the example of Fig. 7, M3M4 and M1 do not leave any nodes nonstationary in the interaction network and as a result, each leads to a point attractor. Since they are also mutually exclusive, they lead to two separate attractors.

Next we find all the combinations of SMs and motif groups that alone do not lead to point attractors, yet together they successfully stabilize the nodes in the interaction network and lead to an attractor. The search can be targeted by these considerations:

(1) The SMs and motif groups in these combinations must be consistent.

(2) Combinations of two SMs or motif groups that have a LDOI relationship do not need to be considered, as the effectiveness of that combination in trapping the system would not surpass the effectiveness of the source of the LDOI relationship. The same argument holds for SMs and motif groups that have a mutual LDOI relationship.

(3) As a specific application of the previous point, combination of a motif group with any of its subsets does not need to be considered.

In the example of Fig. 7, the only allowed combination is {M2, M3}. As M2 is mutually exclusive with M3M4, and M3 is mutually exclusive with M1, this combination leads to a distinct new attractor. Overall, this example network has three attractors.

It should be noted that it is possible that a SM or motif group alone cannot stabilize all the nodes in the interaction network, but it also cannot participate in a consistent combination. Such a motif leads to a complex attractor, in which the nodes outside of the LDOI of the motif oscillate. If the motif is mutually exclusive with all previously found attractors, then it is added to the list of attractors and counted as a separate one. Examples of such motifs can be seen in Fig. 1(e), in which the gray gradient stable motif cannot stabilize the nodes B and D and leaves them nonstationary. This motif cannot be grouped with the other SM (in gray), because they are mutually exclusive. The gray gradient motif leads to a complex attractor in which A=C=1 and B and D oscillate, as seen in the expanded network.

As a further illustration, in Fig. S2 in the Supplemental Material [57] we go through all the steps of this attractor identification method for the example network of Fig. 5.

TABLE I. Summary of the completion percentage of attractor identification by four methods: consistent groups of SMs (suggested in this work), PyBoolNet, SM analyses 2013 and 2021. For network sizes of 40 and 55, 100 networks were randomly selected and were analyzed using all four methods. For network sizes of 70 and 100, 50 networks were randomly selected and the same 50 networks were analyzed using all four methods. For each network size there are four rows, each showing the cumulative completion percentages for one of the methods within the specified time limits.

Network size	Method	Completion % 30 s	Completion % 1 min	Completion % 3 min	Completion % 15 min	Completion % 30 min
40	Consistent Groups of SMs	100	100	100	100	100
40	PyBoolNet trap spaces	100	100	100	100	100
40	SM analysis 2021	71	75	79	86	91
40	SM analysis 2013	51	65	78	92	95
55	Consistent Groups of SMs	100	100	100	100	100
55	PyBoolNet trap spaces	100	100	100	100	100
55	SM analysis 2021	44	52	54	61	65
55	SM analysis 2013	16	25	38	61	67
70	Consistent Groups of SMs	100	100	100	100	100
70	PyBoolNet trap spaces	100	100	100	100	100
70	SM analysis 2021	20	26	32	46	50
70	SM analysis 2013	4	8	14	22	34
100	Consistent Groups of SMs	92	94	98	100	100
100	PyBoolNet trap spaces	100	100	100	100	100
100	SM analysis 2021	0	2	10	12	18
100	SM analysis 2013	0	0	2	6	8

G. Performance

There are various features and optimization steps that we implemented in each part of the attractor-finding process that minimize the computational time of finding the attractors of each network:

(1) We include a precheck of whether the network has negative edges. As described in Appendix D, in networks without negative edges each CSM has a single support, which is confined to one half of the expanded network. In these networks it is not necessary to search for smaller CSMs within maximal CSMs and the support identification process can be stopped after the first hit.

(2) We use the specific features of the plant-pollinator networks in our ensembles (described in Appendix D) to identify additional networks in which it is not necessary to search for smaller CSMs within maximal CSMs.

(3) At the stage when the CSMs in the system have just been identified and before starting the search for the support of the CSMs, we preorder the CSMs based on the likelihood that their support will be combinatorial. If the condition of CSM_2 is part of CSM_1, then it is likely that the support of CSM_2 will be combinatorial, thus it is analyzed later than CSM_1. For example, in Fig. 5, the condition of the orange CSM is contained in the yellow CSM, thus the orange CSM is analyzed after the yellow CSM. This consideration ensures that all combinatorial supports of each CSM are found and reduces the computational time of finding the supports of CSMs such as the orange CSM.

(4) Realizing the importance of functional relationships between the SMs and motif groups, the functional relationships are included in the algorithm to find the supports of each CSM. While finding combinatorial supports of each CSM, we only search through the combinations that include SMs and motif groups that are consistent and are not in the LDOI of each other.

The python implementation of our method is available at Ref. [59]. We first did an exploratory analysis to compare the computational time of the attractor-finding process using our method, minimal trap space identification with PyBoolNet (described in Sec. II C), the SM analysis 2013 (Java implementation of the SM succession diagram based attractor identification method) [45,46], and the SM analysis 2021 (python implementation of the SM succession diagram based attractor identification method) [50]. We used four ensembles with different number of nodes (40, 55, 70, and 100). The PyBoolNet trap space identification and our method were able to finish the analysis of all 1000 networks in each ensemble. Both SM analyses 2013 and 2021 took several hours for certain networks. To better compare the methods, we set time limits after which the analysis of a network should be abandoned. We also restricted the analysis to a randomly selected sample of 100 networks (for networks with 40 and 55 nodes) or a randomly selected sample of 50 networks (for networks with 70 and 100 nodes). The summary of time limits and the percentage of completed networks within that time limit using each method is shown in Table I. Analyzing the whole ensemble of 1000 networks, the average runtime per network with our method ranged from 0.1 s (for 40 node networks) to 151 s (for 100 node networks). The average runtime per network with PyBoolNet ranged from 0.1 s (for 40 nodes) to 1.3 s (for 100 nodes). This analysis shows that our method is faster than the other two SM methods, while the minimal trap space identification of PyBoolNet is the fastest. All methods gave identical attractors.

Next we used our method to find the number of attractors for all the network models in this study, 36 000 networks with the size range of 10 to 100. We verified that for each network the number of attractors using our method was the same as the number of minimal trap spaces we found using PyBoolNet. Furthermore, we found that the number of attractors was preserved whether single or multiple nodes were
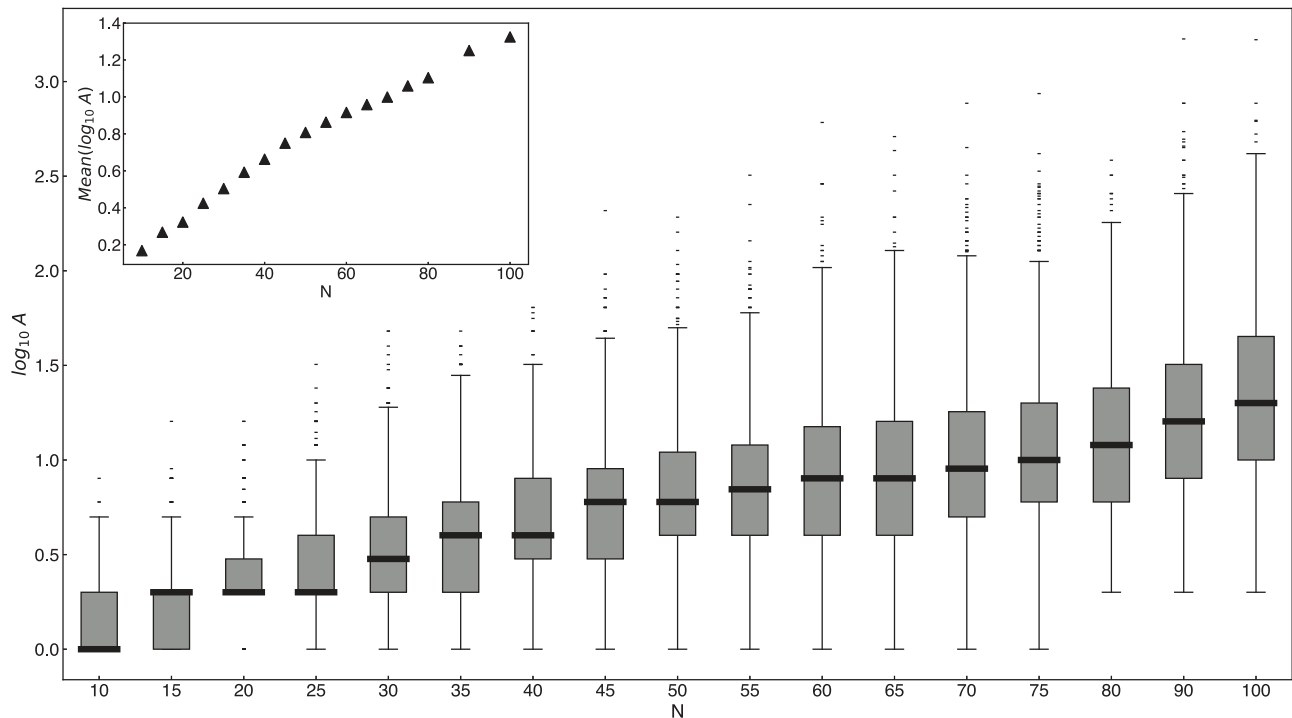
FIG. 8. Box plot of the 10 based logarithm of the number of attractors (A) as a function of network size (N). The inset on the top left represents the average of the 10 based logarithm of the number of attractors as a function of network size.

updated at the same time; see Appendix C for more details. The ecological interpretation of this result is that the number of community outcomes does not depend on the degree of synchronicity of species influx and outflux. We conclude that our method predicts the number of attractors correctly by counting the mutually exclusive consistent groups of SMs and CSMs. While our method is less efficient than minimal trap space identification by PyBoolNet, it has the benefit of identifying the motif groups (survival units) that lead to each attractor (community outcome).

H. Properties of attractors in plant-pollinator interaction networks

The 36 000 networks in the ensemble have altogether 384 282 attractors. The vast majority (99%) of these attractors are point attractors and thus represent community outcomes in which all species maintain a specific survival state. Only 4047 of the attractors are complex; in these attractors a small fraction (14% on average) of the species oscillate between being present and absent. These 4047 complex attractors belong to 1014 networks, which is 2.8% of all networks.

Noting that the distribution of the number of attractors for networks of the same size is approximately log-normal, we indicate the box plot of the 10 based logarithm of the number of attractors as a function of network size in Fig. 8. In this figure it is observed that as the network size increases, the number of attractors increases as well. This is expected: When the number of species increase in a potential species pool, there is more chance for each species to find interaction partners that can help it persist in the ecosystem.

Turning to the composition of the communities represented by attractors, in Fig. 9 we indicate the percentage of species present in attractors, averaged over networks of the same size. Only a small percentage of the species oscillate (less than 2% of the species that had a chance to establish). Up to 48% of the species that had a chance to establish did ultimately establish. This indicates that although only an average of 7.7% of the interaction pairs are mutually beneficial in the original networks, they ensure the formation of stable communities that encompass a sizable percentage of the original pool (up to 20%). This finding is in line with the results of Campbell *et al.* [26].

Our analysis allows for a deeper understanding of these results and connects the number of possible community outcomes with the number of stable subcommunities. We estimate the latter by determining the number of active SMs that are not mutually in the LDOI of each other. Two or more active SMs that have mutual LDOI relationships with each other lock in the same subset of nodes to the same state; we interpret these as the same subcommunity. If n stable subcommunities were independent of each other, then one would expect that the number of community outcomes scales as 2^n. We find networks that accord with this scaling, but also networks in which the number of community outcomes is less than this expectation (see Fig. 10). In this figure, we plotted the number of community outcomes (attractors) versus the number of independent subcommunities (active SMs that are not mutually in the LDOI of each other). Networks with 2^n community outcomes represent the majority when there are up to 6 subcommunities, are a minority for 7 and 8 subcommunities, and are not represented for nine or more subcommunities. A contributing factor to the numbers lower
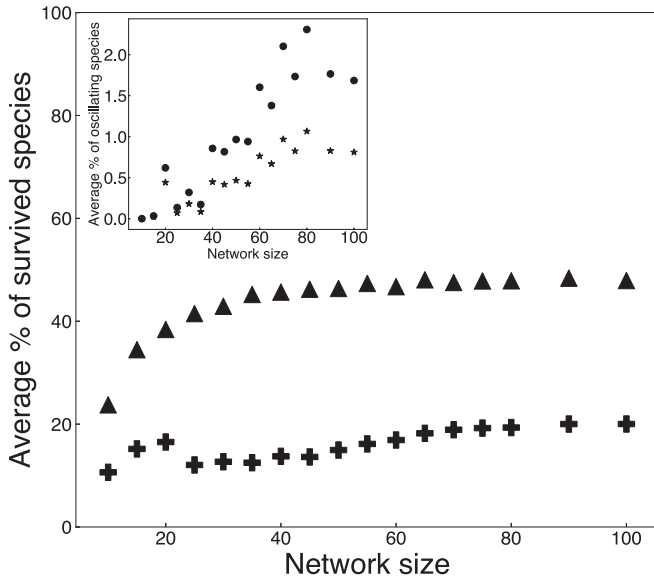
FIG. 9. Average percentage of species present in the attractors (community outcomes) as a function of network size. The percentages are shown relative to the size of the original species pool (cross marker) and relative to the number species in the species pool that had a chance to establish (triangle marker). The inset represents the average percentage of species that oscillate in the attractors as a function of network size. These percentages too were calculated relative to the size of the original species pool (star marker) and relative to the number of species in the species pool that had a chance to establish (circular marker). As the network size increases, on average more species manage to survive or oscillate in the attractors, but the rate of increase slows (saturates).
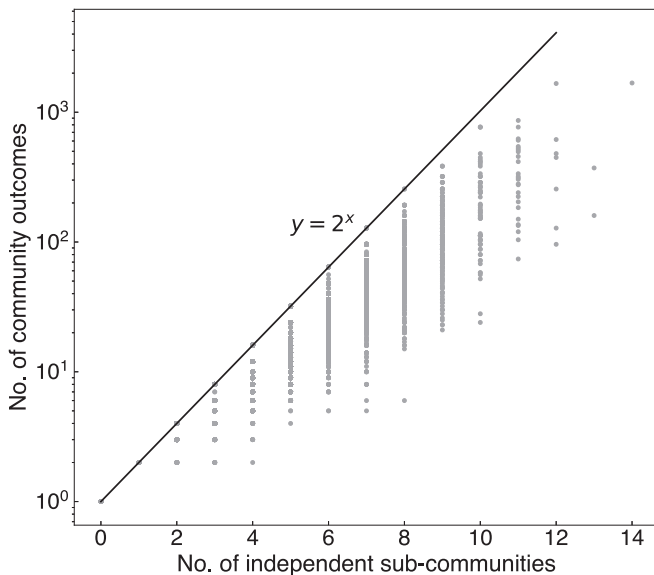

FIG. 10. Scatter plot of the number of possible community outcomes (attractors) versus the number of independent subcommunities (active SMs that are not mutually in the LDOI of each other) in each network. Each symbol represents a distinct pair of (x, y) values. The solid line shows the function $y = 2^x$, which corresponds to the number of community outcomes in case of subcommunities that do not affect each other's survival states in any way. The actual number of community outcomes is less than or equal to this expectation.

than 2^n is the existence of subcommunities whose survival sustains each other in an asymmetrical way (i.e., via a one sided logical determination relationships among the corresponding active SMs). In such cases, not every survival unit can stabilize independently, and hence there are less than 2^n possible community outcomes.

IV. DISCUSSION

The development and analysis of models that capture salient behavior of ecological systems such as plant-pollinator communities without fine grain details are crucial to practical efforts for maintaining pollinator species [60]. We contribute to this endeavour by deeper analysis of the dynamic Boolean network model of mutualistic plant-pollinator community assembly developed in Ref. [26]. Here, we described a method to identify and characterize stable communities of plants and pollinators via identifying smaller groups of interacting species, whose survival reaches a stable state and then determining the relationships among these smaller groups of species. We showed how these relationships contribute to the eventual formation of distinct stable community outcomes. By analyzing ensembles of networks generated by a well-established model for plant-pollinator community assembly [26], we found a remarkable community diversity: up to an average number of 43 community outcomes possible for the largest regional species pools, each community preserving an average of 20% of the original species pool.

Our analysis relies on the expanded network representation, which represents the causal relationships between the variables of a dynamical model. Construction of the expanded network from the original threshold functions in the Campbell *et al.* model required the identification of the prime implicants of the Boolean threshold functions, which resulted in 92.2% of the negative edges being dropped from the Boolean models. We also proposed and implemented a simplification method that samples the remaining negative edges while preserving the survival probability $H(x) = 1$, which we found preserves most attractors. This simplification considerably decreases the complexity of the regulatory functions. For the example of a node that has three positive regulators and five negative regulators presented in Appendix A, the Boolean function in disjunctive prime form has 33 prime implicants, while the simplified function only has three prime implicants, each corresponding to a positive regulator. After the conversion and simplification almost all (99.8%) of the negative edges are eliminated while 100% of the double positive edges are preserved. The vast majority of the edges represent single (67%) or mutual (31%) positive interactions, in agreement with the mutualistic nature of the plant-pollinator interactions. One of the advantages of this simplification method is that it takes the dynamics of the system and the threshold function (1) into consideration. Other sparsification methods such as weighted bipartite matching rely solely on the network topology. The network selected by bipartite matching would contain only one incoming edge, which is positive, to each target node. This removes all the negative edges, 14% of single positive edges (on average), and 30% of double positive edges, and hence a smaller number of positive interactions remain. As these positive interactions are important for the stability of

plant-pollinator community outcomes, we expect that the network obtained by bipartite matching would have lesser species richness and fewer community compositions (attractors).

We propose a method of attractor identification that builds on and improves previous methods to link stable motifs and conditionally stable motifs to attractors. Both the stable motif succession diagram based attractor-finding process [45,46,50] and the method described here rely on the identification of successions of SMs and CSMs; however, in this work instead of building the succession diagram through stable motif-based network reduction, we detect the functional relationships among motifs and identify mutually exclusive combinations of stable motifs and motif groups. Construction of motif groups using the functional relationships eliminates a large part of the combinatorial explosion observed in stable motif succession diagrams. Indeed, due to the abundance of mutually positive interactions the plant-pollinator networks have a large number of (conditionally) stable motifs; up to 35 in the largest networks (see Fig. 6). While both the SM analyses 2013 and 2021 spent a considerable amount of time tackling the combinatorial effect of exploring all motif successions and permutations, we addressed this complexity by the use of LDOI relationships among stable motifs and motif groups. We took advantage of this relationship to simplify the combinations of stable motifs and motif groups and reduce the computational time considerably.

The plant-pollinator interaction networks we analyzed in this study have properties that significantly simplify the attractor-finding process. The majority of these networks (92% of 36 000 networks) do not contain negative edges, which results in disjoint expanded networks that are simpler to analyze. Being bipartite with power law degree distribution is another property that restricts the number of edges in these networks, and hence decreases the complexity considerably. In addition, the Boolean functions follow the same template, restricting the possible (conditionally) stable motif patterns. This helps in focusing the search in the cycle graph and expanded network.

The same features that simplify our analysis also allow the preservation of the number of attractors for a very broad class of update schemes (including single node and multi-node stochastic update). The ecological interpretation of this result is that the number of community outcomes does not depend on the timing of species influx or outflux, but rather on the causal relationships among species and survival units. Timing is expected to influence the trajectories that lead to each attractor. For example, in cases where two mutually exclusive stable motifs (survival units) exist, timing may influence which of these stable motifs locks in. At that point the other stable motif, and all community outcomes that contain it, are no longer attainable.

While PyBoolNet can quickly identify attractors in these networks, it does not uncover stable or conditionally stable survival units and the relationships between them. Our method allows for directly interpretable insights into the survival units that make up each community outcome and their relationships. We demonstrate that these relationships play a key role in explaining the diversity of community outcomes. Another advantage of our method is that it provides information about the trajectories the system takes toward a community outcome, not only in its initial formation but also in response to perturbations in the form of species extinction, invasion, interaction loss or gain.

The information gained from stable motif analysis supports ecological findings from field observations in plant-pollinator communities. For instance, Vilà *et al.* conducted a meta-analysis of the observed impacts of invasion of a single species on established communities. This study reported that alien plants caused an average of 43.5% decrease in native plant species abundance and 17.3% decrease in native animal species abundance [61]. While there is not enough information to precisely model the real plant-pollinator networks, our study allows qualitative insights into the observed decrease in the species abundance. Certain invasive species mimic other species to attract pollinators but do not provide any food for them, thereby adding negative interactions to the system. Our analysis illustrates that as the number of negative interactions increases, there is a lower chance for the establishment of active stable motifs. As a result, species richness and abundance decrease due to detrimental invasive species.

Invasive species can also have a beneficial effect on the community. Russo *et al.* observed that the introduction of a highly pollinator-attracting plant to an established community led to a 302% increase in the abundance of native pollinator species as well as a 35% increase in the pollinator species richness [62]. Such trends are verified in a previous simulation study performed on the Campbell *et al.* model [25] and can be explained via our analysis: as the number of positive interactions increase, there is a higher chance for having active stable motifs, thus stable communities with more present species form.

Regarding species extinction, Biella *et al.* observed that when removing a single generalist plant, an extinction cascade follows [63]. This finding is in line with our results: the absence of generalist species can be a driver of an inactive stable motif; if the driver is activated the whole stable motif locks in. The stabilization of an inactive stable motif is equivalent to an extinction cascade in the community.

Our method also allows the identification of attractor control strategies. Based on the knowledge of the (C)SMs stabilizing and participating in each trajectory leading to a particular attractor, one can determine the minimal set of node states such that if they are fixed, it is guaranteed that the system reaches the desired attractor [45,46]. This information can also form the basis of strategies to destabilize an undesired attractor [64]. In the ecological landscape, this knowledge can be used to propose targeted interventions for conservation purposes.

We expect that the methods described here will effectively uncover the attractor repertoire of Boolean models of information spreading in social networks (such as the Voter model), which do not contain negative influences [5]. Previous methods to identify the attractors of the Voter model include simulation and minimizing the Hamiltonian, which is similar to the Ising model describing a ferromagnet in the absence of an external field. The threshold function (1) can also be thought of as the Hamiltonian of the Ising model; however, one should note that biological and social systems are out-of-equilibrium systems, so their attractors are not necessarily minima of a Hamiltonian energy function. The search for

suitable energy functions is an active area of research, see for example the review article [65]. As a result not all the attractors of such systems can be found using the minimization of the Hamiltonian; this has been observed in the Voter model as well as in plant-pollinator networks.

When our method is applied to general ensembles of Boolean networks, it is likely that new combinatorial bottlenecks will be encountered. For example, in systems whose negative edges are not coupled to positive edges there will be a larger diversity in the composition of conditionally stable motifs, both in terms of virtual nodes and in terms of cycles. We expect that the number of motif groups will be larger than in the plant-pollinator ensembles studied here. Nevertheless, this increase will not be drastic, as the number of stable motifs determines the support of each CSM, and most Boolean networks have fewer stable motifs than our ensembles. The application of our method to general Boolean networks will be a topic of future work.

APPENDIX A: AN EXAMPLE ILLUSTRATING THE CONVERSION OF THRESHOLD FUNCTIONS TO DISJUNCTIVE PRIME FORM

We illustrate the conversion of threshold functions to disjunctive prime form with an example of a node A that has 3 positive regulators $\{P_1, P_2, P_3\}$ and 5 negative regulators $\{N_1, \ldots, N_5\}$. For simplicity of notation we will not explicitly indicate the time dependence of the node states and thus we will use σ_i instead of $\sigma_i(t)$. For $p_a = 1$, the minimum number of inactive negative regulators is $n_i = 2$, hence two negative regulators should be inactive for $H(x) = 1$. As a result, the prime implicants are

$$\sigma_{P_1} \text{ AND } (\text{NOT } \sigma_{N_1}) \text{ AND } (\text{NOT } \sigma_{N_2}),$$

$$\sigma_{P_1} \text{ AND } (\text{NOT } \sigma_{N_1}) \text{ AND } (\text{NOT } \sigma_{N_3}),$$

$$\vdots$$

$$\sigma_{P_1} \text{ AND } (\text{NOT } \sigma_{N_4}) \text{ AND } (\text{NOT } \sigma_{N_5}).$$

As the number of cases of having 2 inactive regulators is $\binom{5}{2} = 10$ and there are 3 positive regulators, there are 30 prime implicants for $p_a = 1$. For $p_a = 2$, the minimum number of inactive negative regulators is $n_i < 0$, meaning that 2 active positive regulators can overcome all negative regulators. This leads to the additional prime implicants

$$\sigma_{P_1} \text{ AND } \sigma_{P_2},$$

$$\sigma_{P_1} \text{ AND } \sigma_{P_3},$$

$$\sigma_{P_2} \text{ AND } \sigma_{P_3}.$$

Increasing the number of active positive regulators to $p_a = 3$ increases the overall weight of the positive regulators even further. The implicant (σ_{P_1} AND σ_{P_2} AND σ_{P_3}) is not a prime implicant since it can be covered by the more general prime implicants of $p_a = 2$ case, which are (σ_{P_1} AND σ_{P_2}), (σ_{P_1} AND σ_{P_3}), and (σ_{P_2} AND σ_{P_3}); thus the process terminates at $p_a = 2$. The disjunction of the constructed prime implicants results in the complete Boolean update function of node i:

$$
\begin{aligned}
f_i = &\left(\sigma_{P_1} \text{ OR } \sigma_{P_2} \text{ OR } \sigma_{P_3}\right) \\
&\text{AND } \big[\left(\text{NOT } \sigma_{N_1} \text{ AND NOT } \sigma_{N_2}\right) \\
&\text{OR} \ldots \\
&\text{OR } \left(\text{NOT } \sigma_{N_4} \text{ AND NOT } \sigma_{N_5}\right)\big] \\
&\text{OR } \left(\sigma_{P_1} \text{ AND } \sigma_{P_2}\right) \\
&\text{OR } \left(\sigma_{P_1} \text{ AND } \sigma_{P_3}\right) \\
&\text{OR} \left(\sigma_{P_2} \text{ AND } \sigma_{P_3}\right).
\end{aligned}
\tag{A1}
$$

APPENDIX B

1. Deciding the direction of the inequality in Equation (5)

To determine the proper direction for Eq. (5) in Sec. III C, we compared the attractors of both cases of the simplified Boolean models (i.e., both inequality directions) with the attractors of the exact Boolean model which is generated using the prime implicants discussed in Sec. III A. As the closeness of the regulatory functions can be evaluated for any update method, we use synchronous update, in which all the nodes are updated simultaneously, due to its simplicity. We used four different initial conditions:

(1) IC 100%: All of the species in the species pool are present at $t = 0$.

(2) IC 75%: 75% of the species are selected randomly and assumed to be present at $t = 0$.

(3) IC 50%: 50% of the species are selected randomly and assumed to be present at $t = 0$.

(4) IC 25%: 25% of the species are selected randomly and assumed to be present at $t = 0$.

There are two types of nodes in each attractor corresponding to synchronous update: nodes that acquire a fixed state and nodes whose state changes in a cyclic manner. We considered that an attractor of the simplified model agrees with an attractor of the exact model if (i) the sets of nodes that acquire a fixed state and the sets of nodes that oscillate are the same, and (ii) the state of each fixed node is the same. This comparison was performed for the ensemble with the largest networks (50 plants and 50 pollinators) and an ensemble of smaller networks (20 plants and 30 pollinators) and the summary is shown in Table II.

As it can be seen in Table II, the attractors of both simplified Boolean models show good agreement with the attractors of the exact Boolean model. We choose the first case, $p_b \geqslant p_t$, which gives slightly more diversity and more rich stable communities. Subsequently, simplifying Eq. (5) yields the

TABLE II. Average percent agreement between the attractors of the exact Boolean model and the attractors of the simplified Boolean models for two ensembles of networks (50-plant-50-pollinator and 20-plant-30-pollinator networks). Each ensemble has 1000 networks that the averages are calculated over.

Initial condition	50 plants 50 pollinators	
	$p_b \leqslant p_t$	$p_b \geqslant p_t$
IC 100%	90.6	92.6
IC 75%	91.1	92.5
IC 50%	91.2	92.5
IC 25%	93	95.2
Initial condition	20 plants 30 pollinators	
	$p_b \leqslant p_t$	$p_b \geqslant p_t$
IC 100%	97.8	95.2
IC 75%	98	95.7
IC 50%	98.4	96.5
IC 25%	99	98.2

condition

$$x \leqslant N_n - \frac{\ln\left(\frac{\sum_{p_a=1}^{N_p}\binom{N_p}{p_a}*\left[\sum_{n_a=0}^{4p_a-1}\binom{N_n}{n_a}\right]}{2^{N_p}-1}\right)}{\ln 2}, \quad \text{(B1)}$$

where x is the largest integer number that makes this inequality hold true. Fig. 11 shows the number of negative regulators we keep (x) as a function of total number of negative regulators (N_n) for 4 different number of positive regulators.
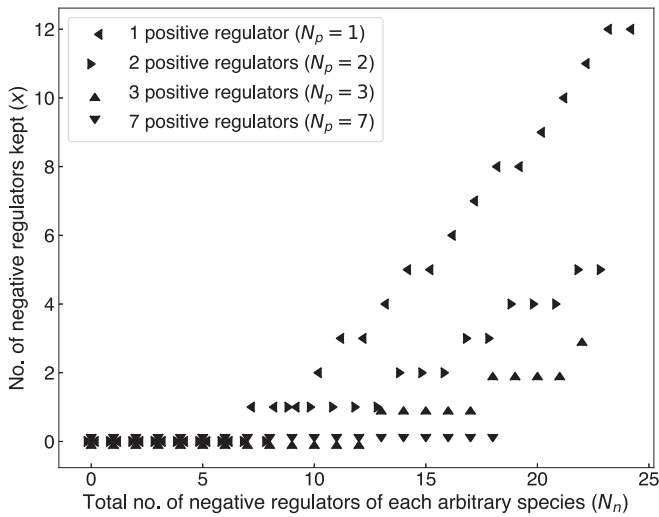

FIG. 11. The number of negative regulators of each arbitrary species that are kept after the simplification according to Eq. (B1). The figure show examples of the number of negative regulators kept in the simplified model versus the total number of negative regulators in the threshold model, for four values of the number of positive regulators. According to the in-degree distributions in these networks, the maximum number of regulators a species can have is 25. As the number of positive regulators increases, the number of negative regulators they can overcome increases as well. Seven positive regulators can overcome 18 negative regulators, thus in this case there is no need to ensure the inactive state of any of the negative regulators.

TABLE III. Average percent agreement between the synchronous attractors of the exact and simplified Boolean models in the case of perturbations for the set of networks that have 50 plants and 50 pollinators. This average was calculated over an ensemble of 1000 networks, and IC 25% was used as the initial condition. The first column shows the percentage of the nodes (w_p) that were selected at random and fixed to the active state in both models. The second column shows the percent difference in the attractor of the exact model caused by the perturbation. The third column shows the percent agreement between the attractors of the exact and simplified models when they both undergo the same perturbation. When 50% or more of the nodes were perturbed, the attractors of the two models became identical.

w_p	Effect of perturbation	Average % agreement
10	19.0	96.7
30	25.8	99.6
50	27.1	100

According to this figure a node that has 3 positive regulators and 5 negative regulators (as presented in Appendix A) would have a simplified Boolean function that does not have any negative regulators.

2. Validation of the simplified networks

To further investigate the accuracy of the simplified model in capturing the properties of attractors, the attractors of the simplified and exact models are compared under synchronous update and in the case of perturbations. We used the ensemble of networks that have 50 plants and 50 pollinators. Since according to Fig. 3 about 58% of the species are inactive in all attractors, we decided to set and maintain in the active state 10–50% of the nodes (selected randomly). Also, given that the simplified Boolean function results in a slight tendency toward $H(x) = 1$ for each node, the initial condition IC 25%, which can potentially cause a large discrepancy in the attractors, was used. Table III shows that the attractors of the simplified model capture the attractors of the exact model to a great extent (98.8%).

APPENDIX C: THE ATTRACTORS ARE THE SAME UNDER GENERAL ASYNCHRONOUS UPDATE

Under the update schedule that results from most permissive Boolean networks (i.e., the update schedule that permits simultaneous updates of any subset of nodes) the attractors coincide with the minimal trap spaces. The software PyBool-Net uses general asynchronous update, which does not allow simultaneous updates. While the point attractors are preserved under all update schemes, there are differences in the complex attractors: Under general asynchronous update there can be multiple complex attractors in the same trap space and there can be complex attractors that lie outside of minimal trap spaces. We were interested whether the update scheme affects the number of attractors in our ensembles of plant-pollinator networks.

Following the identification of minimal trap spaces with PyBoolNet, we implemented three criteria provided in the
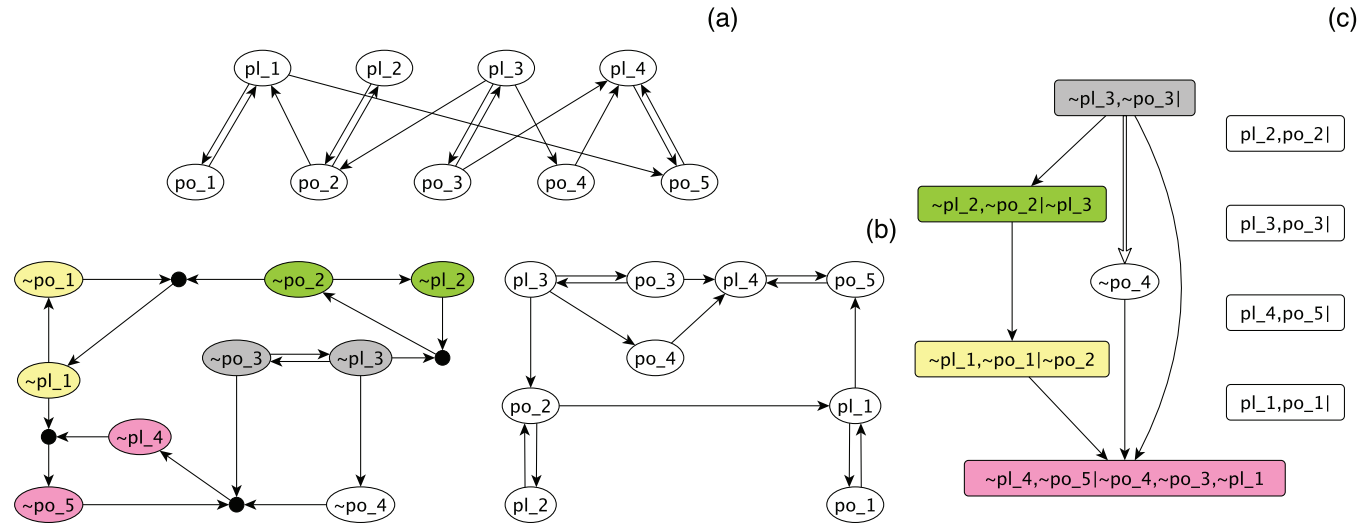
FIG. 12. An example of an interaction network without negative edges. Panel (a) shows the interaction network consisting of 4 plants and 5 pollinators. The sole case of a source SCC is the positive feedback loop made up of pl_3 and po_3. As a result, only this SCC leads to a SM representing the inactive states of species (to an inactive SM). The rest of the SCCs lead to SMs representing active states of species or CSMs representing the inactive states of species. Panel (b) shows the disjoint expanded network. The 4 inactive SMs and CSMs are highlighted with color: The SM consisting of ∼pl_3 and ∼po_3 is highlighted with gray, the CSM consisting of ∼pl_2 and ∼po_2 is highlighted with green, the CSM consisting of ∼pl_1 and ∼po_1 is highlighted with yellow, and the CSM consisting of ∼pl_4 and ∼po_5 is highlighted with pink. Panel (c) shows the cycle graph with the cycles highlighted with the same colors. The green and yellow CSMs each have a single condition. The condition of the green CSM is a virtual node within the gray SM, and the condition of the yellow CSM is a virtual node within the green CSM. As a result, the sole support of the green CSM is the gray SM, and the sole combinatorial support of the yellow CSM is the union of the gray SM and green CSM. The pink CSM has multiple conditions that are successfully satisfied by the union of the gray SM, green CSM, and yellow CSM. This union is also the only support of the pink CSM.

package to determine the relationship of the attractors (under general asynchronous update) and minimal trap spaces: "univocality," which is satisfied if there is exactly one attractor within each minimal trap space, "completeness," which is satisfied if there are no attractors outside the identified minimal trap spaces, and "faithfulness," which is satisfied when the nonstationary variables within each minimal trap space oscillate in all the attractors contained in it [66]. The number of minimal trap spaces is equal to the number of attractors if the first two checks are true, i.e., if there is no attractor outside the minimal trap spaces and each trap space contains exactly one attractor; otherwise, it is a lower bound. If all three checks are satisfied, then the attractors of the system are equivalent to the identified minimal trap spaces, i.e., they have the same nodes with stationary states and same nodes with oscillatory states.

We found that all three checks are satisfied in all 36 000 networks in our ensemble. This indicates that the number of attractors equals the number of minimal trap spaces under general asynchronous update as well. The ecological interpretation of this result is that the number of community outcomes is preserved whether species enter or leave the community individually or in groups.

APPENDIX D: FEATURES OF CSMS AND THEIR SUPPORTS IN PLANT-POLLINATOR NETWORKS

1. Networks without negative edges

The vast majority of the networks in the plant-pollinator models do not contain negative edges (see Fig. S1 in the Supplemental Material [57]). In these networks all regulators

have a positive effect and multiple regulators are connected by the "OR" operator in the regulatory function of each node. This means that the activity of any individual regulator is able to sustain the activity of a target species. In this subsection we determine the properties of CSMs and their supports in these networks that do not contain negative edges.

For networks with no negative edges in the interaction network, the expanded network does not have any paths that connect a virtual node that expresses the active state of a species with a virtual node that expresses the inactive state of another species. As a consequence, the expanded network is disconnected. The weakly connected components (i.e., maximal subgraphs that are connected if we disregard the directions of the edges) of the expanded network of these networks fall into two categories: connected components that consist of virtual nodes representing active states, and connected components of virtual nodes representing inactive states [see Fig. 12(b)].

Due to the sufficiency expressed by the "OR" operator, in these networks any two-node positive feedback loop can maintain the active state of its nodes, and yields a SM. Conversely, each CSM, which by definition must contain a composite node (AND gate), cannot fall in the component made up by virtual nodes representing active states. Each CSM must fall within a component made up by virtual nodes representing inactive states, and its support(s) must be within the same component.

A network that does not contain negative edges has a SM made up by virtual nodes representing inactive states if and only if the network has a strongly connected component of

positive or double-positive edges that contains the incoming edges of each of its nodes. A technical term for such strongly connected component is source SCC. In other words, if a node has an incoming edge that is not part of an SCC, then the virtual node representing the inactive state of that node cannot be part of a SM. This is because the inactivity of that node depends on an "AND" gate that unites the inactive states of all its regulators. Because inactive SMs are based on source SCCs, and source SCCs cannot intersect or interact, it is not possible that a virtual node is contained in multiple inactive SMs in plant-pollinator networks that lack negative edges.

There are only a few possibilities for the support of an inactive CSM. We first consider inactive CSMs that have a single condition. The support could be a SM that contains the condition of the focal CSM (i.e., the CSM in question). The support could be a SM that contains a node that is connected to the condition of the focal CSM by a linear chain of positive edges. An example of such a chain is po_6, pl_6, and po_4 in Fig. 5; through this chain the pink SM forms the support of the yellow CSM. The condition of the focal CSM could be contained in another inactive CSM; in this case the focal CSM has a combinatorial support. An example of such a situation is the yellow CSM in Fig. 12: its support is made up by the combination of the gray SM and green CSM. Finally, if the focal CSM has multiple conditions, its support must be a combination of the previously described ways of satisfying a single condition. Such an example is the pink CSM in Fig. 12: its three conditions are satisfied by the union of the gray SM (both directly and via ∼po_4) and the green and yellow CSMs. Because of the impossibility of overlaps among inactive SMs and CSMs, in each of these situations there is a single support.

2. Networks that contain negative edges

As shown in Fig. S1 in the Supplemental Material [57], following the conversion and simplification of the regulatory functions only 8% of all networks have negative edges. Importantly, each negative edge appears together with a positive edge in the opposite direction. Furthermore, any node that has an incident negative edge must have at least one incident positive edge; otherwise, it would have been eliminated during model simplification. In the previous section we described the simplifications to the CSM identification process in networks that do not contain any negative edges. Here we focus on the networks that do have negative edges and show that even in these networks simplifications are possible due to the specific patterns these negative edges form.

In a general Boolean model it is possible that a cycle formed by virtual nodes corresponding to the activity of the constituent species and a cycle formed by virtual nodes corresponding to inactivity of the constituent species mutually satisfy each other's conditions. This would require mutual inhibition between two positive feedback loops. However, due to the obligate pairing of negative edges with positive edges in the opposite direction in plant-pollinator interaction networks, such mutual inhibition must be combined with mutual activation. As illustrated in Fig. 13(c), this network leads to a SM formed by the inactive state of all four nodes, and two active CSMs, each of which has an empty support. We conclude that
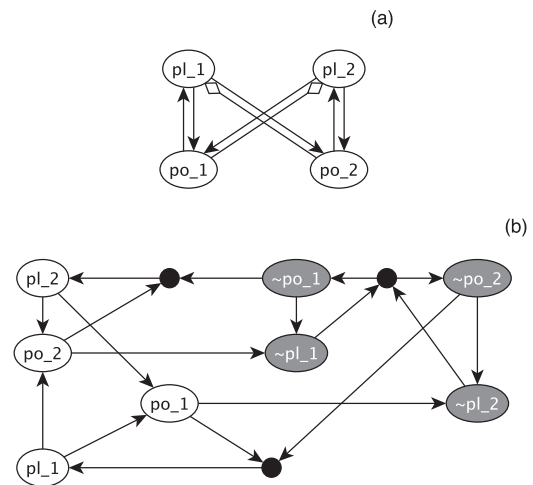

FIG. 13. Mutual inhibition and activation between two positive feedback loops leads to unsupported CSMs in plant-pollinator networks. Panel (a) shows the subnetwork after simplification. Note that a negative edge always comes paired with a positive edge in the opposite direction, and the species that have a negative regulator must also have a positive regulator. As a consequence, the two positive feedback loops mutually activate and inhibit each other at the same time. The expanded network in (b) shows that there is an inactive SM (highlighted with gray) and two active CSMs. These two active CSMs are not supported, since the gray SM, which can satisfy their conditions, is not consistent with them.

in these systems it is not possible to have mutual condition satisfaction between a cycle that expresses the activity of the constituent species and a cycle that expresses the inactivity of the constituent species.

CSMs made up by virtual nodes expressing the active state of the corresponding species (active CSMs) are possible if a positive feedback loop of two nodes intersects a negative feedback loop of two nodes. The obligate coupling of a negative edge with a positive edge in the other direction generally causes the active CSM to lack support. The simplest case of a supported active CSM is shown in In Fig. 14(a). This configuration is unlikely, as it requires that two nodes have seven incoming negative edges each, and network simplification results in these nodes forming two chained negative feedback loops. We did not observe any such or similar configurations in our ensemble of 36 000 networks. All the active CSMs in the ensemble have an empty support.

In general, a positive feedback loop of more than two nodes could contain an even number of negative edges, and the corresponding consistent cycle in the expanded network could contain a mixture of virtual nodes that represent the active state of a species and virtual nodes that represent the inactive state of a different species. Figure 14(c) indicates the simplest subnetwork that can yield a mixed-state cycle that is a CSM with a nonempty support.

Next, we identify the patterns related to the support of CSMs. Consider an inactive CSM (a CSM made up by virtual nodes that express inactive states). Applying Boolean negation to the specific Boolean functions in our ensembles, one can realize that the "AND" gates in the virtual nodes expressing inactivity arise from the "OR" gates of positive regulators.
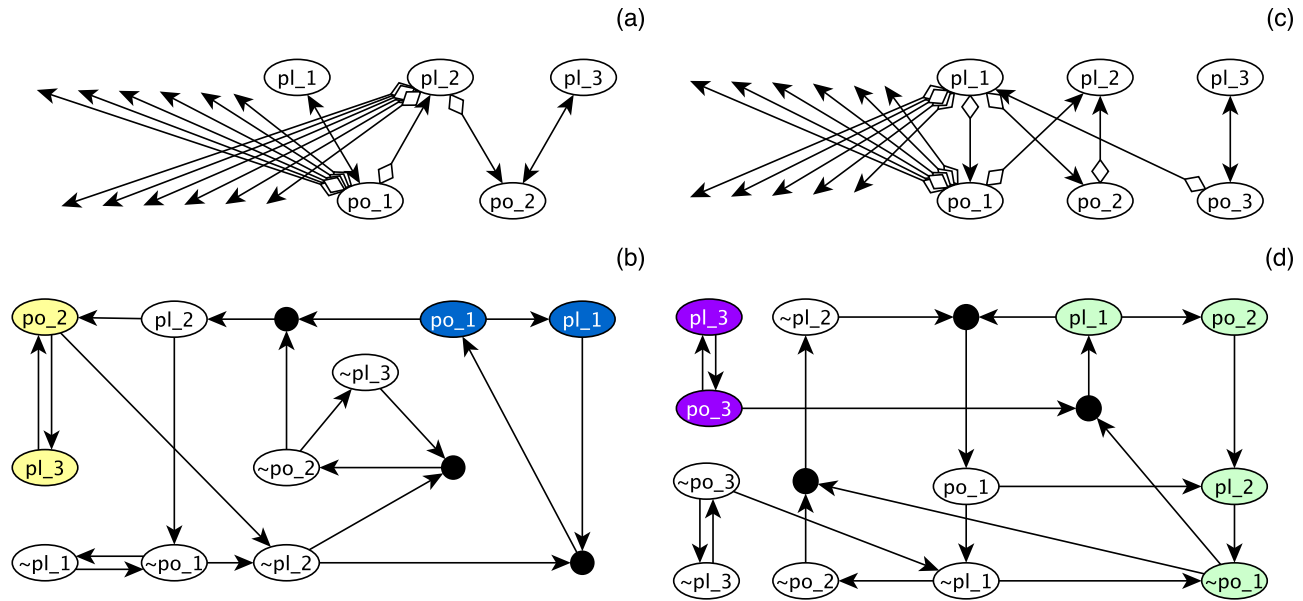
FIG. 14. The simplest subnetworks that can yield supported active or mixed-state CSMs. Panel (a) shows a subnetwork that has 9 plants and 8 pollinators. Nodes pl_2 and po_1 each have 1 positive and 7 negative incoming edges. For each of these nodes the starting point of 6 incoming negative edges can be arbitrary nodes outside of the nodes with labels. Only one of the negative incoming edges is kept during network simplification. A single case among 49 possible choices, namely, keeping the negative edges from po_2 to pl_2 and from pl_2 to po_1 results in a network consisting of pl_1, pl_2, pl_3, po_1, and po_2 that is capable of forming a supported active CSM. Panel (b) shows the corresponding expanded network. The yellow SM consisting of po_2 and pl_3 is the support of the blue active CSM. Panel (c) shows a subnetwork that has the same number of plants and pollinators. Keeping the negative edges from po_1 to pl_1 and from pl_2 to po_1 results in a simplified network that has a mixed-state CSM, highlighted with green in the expanded network of panel (d). The purple SM consisting of pl_3 and po_3 is the support of this mixed state CSM. This case has the same probability of 1 in 49 choices.

As a consequence, each condition of this CSM must be a virtual node that expresses the inactivity of a species. One can verify this property by inspecting the conditions of the CSMs in Figs. 4, 5, and 12. The CSM can have a support that is a SM made up by virtual nodes that express active states; we refer to such a SM as "active SM." There are two requirements for an active SM to be (a part of) the support of an inactive CSM: (i) the condition of the inactive CSM needs to be in the LDOI of the active SM, and (ii) the active SM and its LDOI must be consistent with the inactive CSM and all of its conditions. The blue SM in Fig. 5 satisfies both of these conditions and is a support of the orange CSM. The green SM only satisfies the first condition but not the second, thus is not a support of the orange CSM.

The two requirements indicated above are also relevant to the identification of intermediate CSMs, subsets of an SCC that have their own different support (see Sec. III D 3). Due to the properties described earlier in this section, the only situation in which an intermediate SCS needs to be kept is if the corresponding intermediate CSM has an active SM in its support, while the SCC only has inactive SM(s) as support. Thus, we can make the search for intermediate SCSs of an SCC more specific by determining the intersection of $\mathcal{L}_A$ (the union of the LDOIs of active SMs) and the virtual nodes of the SCC. If $\mathcal{L}_A$ does not contain any members of the SCC, then there is no need to search for intermediate SCSs.

There exist networks in our ensemble that contain negative edges yet none of their CSMs can be supported by an active SM as the two conditions are not satisfied. In these networks all CSMs have a single support. The only networks that can have CSM(s) with multiple supports are the networks in which active SMs can support at least one of the CSMs. In such networks, CSMs can either have supports from both active and inactive SMs and CSMs, or from multiple active SMs. In the first case the supports can be mutually exclusive as in Fig. 5, and in the second case at least one of them is in the LDOI of the other(s). We found that only 3.5% of the ensemble of 36 000 networks can have CSMs with more than one support.

[1] C. R. Shalizi, Methods and techniques of complex systems science: An overview, in *Complex Systems Science in Biomedicine*, Topics in Biomedical Engineering International Book Series, edited by T. S. Deisboeck and J. Y. Kresh (Springer, Boston, MA, 2006), pp. 33–114.

[2] H. Sayama, *Introduction to the Modeling and Analysis of Complex Systems* (Open SUNY Textbooks, New York, 2015).

[3] N. Boccara, *Modeling Complex Systems* (Springer Science & Business Media, Berlin, 2010).

[4] D. Garcia, A. Garas, and F. Schweitzer, Modeling collective emotions in online social systems, in *Collective Emotions* (Oxford University Press, Oxford, UK, 2014), p. 389.

[5] C. Castellano, S. Fortunato, and V. Loreto, Statistical physics of social dynamics, Rev. Mod. Phys. **81**, 591 (2009).

[6] J. S. Clark, D. Bell, C. Chu, B. Courbaud, M. Dietze, M. Hersh, J. HilleRisLambers, I. Ibáñez, S. LaDeau, S. McMahon *et al.*, High-dimensional coexistence based on individual variation: A synthesis of evidence, Ecological Monographs **80**, 569 (2010).

[7] G. Barabás, M. J. Michalska-Smith, and S. Allesina, The effect of intra-and interspecific competition on coexistence in multispecies communities, Am. Nat. **188**, E1 (2016).

[8] J. M. Levine, J. Bascompte, P. B. Adler, and S. Allesina, Beyond pairwise mechanisms of species coexistence in complex communities, Nature (London) **546**, 56 (2017).

[9] J. Grilli, G. Barabás, M. J. Michalska-Smith, and S. Allesina, Higher-order interactions stabilize dynamics in competitive network models, Nature (London) **548**, 210 (2017).

[10] J. M. Montoya and R. V. Solé, Small world patterns in food webs, J. Theor. Biol. **214**, 405 (2002).

[11] R. V. Sole and M. Montoya, Complexity and fragility in ecological networks, Proc. R. Soc. London B **268**, 2039 (2001).

[12] J. Camacho, R. Guimerà, and L. A. Nunes Amaral, Robust Patterns in Food Web Structure, Phys. Rev. Lett. **88**, 228102 (2002).

[13] J. A. Dunne, R. J. Williams, and N. D. Martinez, Network structure and biodiversity loss in food webs: robustness increases with connectance, Ecol. Lett. **5**, 558 (2002).

[14] U. Bastolla, M. A. Fortuna, A. Pascual-García, A. Ferrera, B. Luque, and J. Bascompte, The architecture of mutualistic networks minimizes competition and increases biodiversity, Nature (London) **458**, 1018 (2009).

[15] S. Kéfi, V. Miele, E. A. Wieters, S. A. Navarrete, and E. L. Berlow, How structured is the entangled bank? the surprisingly simple organization of multiplex ecological networks leads to increased persistence and resilience, PLoS Biol. **14**, e1002527 (2016).

[16] S. Pilosof, M. A. Porter, M. Pascual, and S. Kéfi, The multilayer nature of ecological networks, Nat. Ecology Evol. **1**, 0101 (2017).

[17] J. C. Grixti, L. T. Wong, S. A. Cameron, and C. Favret, Decline of bumble bees (bombus) in the north american midwest, Biol. Conserv. **142**, 75 (2009).

[18] J. C. Biesmeijer, S. P. Roberts, M. Reemer, R. Ohlemüller, M. Edwards, T. Peeters, A. Schaffers, S. G. Potts, R. Kleukers, C. Thomas *et al.*, Parallel declines in pollinators and insect-pollinated plants in britain and the netherlands, Science **313**, 351 (2006).

[19] D. Goulson, G. C. Lye, and B. Darvill, Decline and conservation of bumble bees, Annu. Rev. Entomol. **53**, 191 (2008).

[20] L. A. Burkle, J. C. Marlin, and T. M. Knight, Plant-pollinator interactions over 120 years: Loss of species, co-occurrence, and function, Science **339**, 1611 (2013).

[21] A.-M. Klein, I. Steffan-Dewenter, and T. Tscharntke, Fruit set of highland coffee increases with the diversity of pollinating bees, Proc. R. Soc. London B **270**, 955 (2003).

[22] S. G. Potts, J. C. Biesmeijer, C. Kremen, P. Neumann, O. Schweiger, and W. E. Kunin, Global pollinator declines: Trends, impacts, and drivers, Trends Ecol. Evol. **25**, 345 (2010).

[23] L. A. Burkle and R. Alarcón, The future of plant–pollinator diversity: Understanding interaction networks across time, space, and global change, Am. J. Bot. **98**, 528 (2011).

[24] C. Campbell, S. Yang, K. Shea, and R. Albert, Topology of plant-pollinator networks that are vulnerable to collapse from species extinction, Phys. Rev. E **86**, 021924 (2012).

[25] C. Campbell, S. Yang, R. Albert, and K. Shea, Plant–pollinator community network response to species invasion depends on both invader and community characteristics, Oikos **124**, 406 (2015).

[26] C. Campbell, S. Yang, R. Albert, and K. Shea, A network model for plant–pollinator community assembly, Proc. Natl. Acad. Sci. USA **108**, 197 (2011).

[27] C. Campbell, K. Shea, S. Yang, and R. Albert, Motif profile dynamics and transient species in a boolean model of mutualistic ecological communities, J. Complex Netw. **4**, 127 (2016).

[28] C. N. Kaiser-Bunbury and N. Blüthgen, Integrating network ecology with applied conservation: a synthesis and guide to implementation, AoB Plants **7**, plv076 (2015).

[29] J. Memmott, The structure of a plant-pollinator food web, Ecol. Lett. **2**, 276 (1999).

[30] P. Jordano, Patterns of mutualistic interactions in pollination and seed dispersal: Connectance, dependence asymmetries, and coevolution, Am. Nat. **129**, 657 (1987).

[31] P. Jordano, J. Bascompte, and J. M. Olesen, The ecological consequences of complex topology and nested structure in pollination webs, *Plant-Pollinator Interactions: From Specialization to Generalization*, edited by N. M. Waser and J. Ollerton (University of Chicago Press, 2006), pp. 173–199.

[32] J. Bascompte, P. Jordano, C. J. Melián, and J. M. Olesen, The nested assembly of plant–animal mutualistic networks, Proc. Natl. Acad. Sci. USA **100**, 9383 (2003).

[33] J. M. Olesen, J. Bascompte, Y. L. Dupont, and P. Jordano, The modularity of pollination networks, Proc. Natl. Acad. Sci. USA **104**, 19891 (2007).

[34] M. A. Fortuna, D. B. Stouffer, J. M. Olesen, P. Jordano, D. Mouillot, B. R. Krasnov, R. Poulin, and J. Bascompte, Nestedness versus modularity in ecological networks: Two sides of the same coin? J. Anim. Ecol. **79**, 811 (2010).

[35] P. P. Staniczenko, J. C. Kopp, and S. Allesina, The ghost of nestedness in ecological networks, Nat. Commun. **4**, 1391 (2013).

[36] T. LaBar, C. Campbell, S. Yang, R. Albert, and K. Shea, Global versus local extinction in a network model of plant–pollinator communities, Theoret. Ecol. **6**, 495 (2013).

[37] L. Russo, R. Albert, C. Campbell, and K. Shea, Experimental species introduction shapes network interactions in a plant-pollinator community, Biol. Invasions **21**, 3505 (2019).

[38] I. Noy-Meir, Stability of grazing systems: an application of predator-prey graphs, J. Ecol. **63**, 459 (1975).

[39] S. Kéfi, M. Holmgren, and M. Scheffer, When can positive interactions cause alternative stable states in ecosystems? Function. Ecol. **30**, 88 (2016).

[40] R. M. May, Thresholds and breakpoints in ecosystems with a multiplicity of stable states, Nature (London) **269**, 471 (1977).

[41] T. Fukami, Community assembly along a species pool gradient: Implications for multiple-scale patterns of species diversity, Pop. Ecol. **46**, 137 (2004).

[42] M. Stang, P. G. Klinkhamer, N. M. Waser, I. Stang, and E. van der Meijden, Size-specific interaction patterns and size matching in a plant–pollinator interaction web, Ann. Botany **103**, 1459 (2009).

[43] R. Albert and H. G. Othmer, The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in *Drosophila melanogaster*, J. Theor. Biol. **223**, 1 (2003).

[44] R.-S. Wang and R. Albert, Elementary signaling modes predict the essentiality of signal transduction network components, BMC Syst. Biol. **5**, 44 (2011).

[45] J. G. Zañudo and R. Albert, An effective network reduction approach to find the dynamical repertoire of discrete dynamic networks, Chaos **23**, 025111 (2013).

[46] J. G. Zanudo and R. Albert, Cell fate reprogramming by control of intracellular network dynamics, PLoS Comput. Biol. **11**, e1004193 (2015).

[47] H. Siebert, Analysis of discrete bioregulatory networks using symbolic steady states, Bull. Math. Biol. **73**, 873 (2011).

[48] L. Paulevé, J. Kolčák, T. Chatain, and S. Haar, Reconciling qualitative, abstract, and scalable modeling of biological networks, Nat. Commun. **11**, 4256 (2020).

[49] H. Klarner, A. Bockmayr, and H. Siebert, Computing maximal and minimal trap spaces of boolean networks, Natur. Comput. **14**, 535 (2015).

[50] J. C. Rozum, J. G. T. Zañudo, X. Gan, D. Deritei, and R. Albert, Parity and time reversal elucidate both decision-making in empirical models and attractor scaling in critical Boolean networks, Sci. Adv. **7**, eabf8124 (2021).

[51] See https://github.com/jgtz/StableMotifs.

[52] H. Klarner, A. Streck, and H. Siebert, PyBoolNet: A python package for the generation, analysis and visualization of Boolean networks, Bioinformatics **33**, 770 (2017).

[53] See https://github.com/hklarner/PyBoolNet.

[54] G. Yang, J. Gómez Tejeda Zañudo, and R. Albert, Target control in logical models using the domain of influence of nodes, Front. Physiol. **9**, 454 (2018).

[55] See https://github.com/yanggangthu/BooleanDOI.

[56] Y. Crama and P. L. Hammer, *Boolean Functions: Theory, Algorithms, and Applications* (Cambridge University Press, Cambridge, MA, 2011).

[57] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevE.104.054304 for the details on the statistics of the negative edges in the networks and complete steps of the attractor identification method.

[58] D. Deritei, J. Rozum, E. R. Regan, and R. Albert, A feedback loop of conditionally stable circuits drives the cell cycle from checkpoint to checkpoint, Sci. Rep. **9**, 16430 (2019).

[59] See https://github.com/FatemehFN/consistent_groups_of_sms.

[60] T. Lander, Network modelling, citizen science and targeted interventions to predict, monitor and reverse bee decline, Plants, People, Planet **2**, 111 (2020).

[61] M. Vilà, J. L. Espinar, M. Hejda, P. E. Hulme, V. Jarošík, J. L. Maron, J. Pergl, U. Schaffner, Y. Sun, and P. Pyšek, Ecological impacts of invasive alien plants: A meta-analysis of their effects on species, communities and ecosystems, Ecol. Lett. **14**, 702 (2011).

[62] L. Russo, C. Nichol, and K. Shea, Pollinator floral provisioning by a plant invader: Quantifying beneficial effects of detrimental species, Divers. Distrib. **22**, 189 (2016).

[63] P. Biella, A. Akter, J. Ollerton, A. Nielsen, and J. Klecka, An empirical attack tolerance test alters the structure and species richness of plant–pollinator networks, Funct. Ecol. **34**, 2246 (2020).

[64] C. Campbell and R. Albert, Edgetic perturbations to eliminate fixed-point attractors in Boolean regulatory networks, Chaos **29**, 023130 (2019).

[65] J. Wang, Landscape and flux theory of nonequilibrium dynamical systems with application to biology, Adv. Phys. **64**, 1 (2015).

[66] H. Klarner and H. Siebert, Approximating attractors of Boolean networks by iterative CTL model checking, Front. Bioengineer. Biotechnol. **3**, 130 (2015).