

Fast Nonparametric Inference of Network Backbones for Weighted Graph Sparsification

Alec Kirkley^{✉*}

*Institute of Data Science, University of Hong Kong, Hong Kong;
Department of Urban Planning and Design, University of Hong Kong, Hong Kong;
and Urban Systems Institute, University of Hong Kong, Hong Kong*



(Received 14 October 2024; accepted 13 June 2025; published 11 July 2025)

Network backbones provide useful sparse representations of weighted networks by keeping only their most important links, permitting a range of computational speedups and simplifying network visualizations. A key limitation of existing network backbone methods is that they either require the specification of a free parameter (e.g., significance level) that determines the number of edges to keep in the backbone or impose specific restrictions on the topology of the backbone (e.g., that it is a spanning tree). Here we develop a completely nonparametric framework for inferring the backbone of a weighted network that overcomes these limitations and automatically selects the optimal set of edges to retain using the minimum description length principle. We develop objective functions for global and local network backbone methods which evaluate the importance of an edge in the context of the whole network and individual node neighborhoods, respectively, and are generalizable to any weight distribution under Bayesian model specifications that fix the average edge weight either exactly or in expectation. We then construct an efficient and provably optimal greedy algorithm to identify the backbone minimizing our objectives, whose run-time complexity is log-linear in the number of edges. We demonstrate our methods by comparing them with existing methods in a range of tasks on real and synthetic networks, finding that both the global and local backbone methods can preserve network connectivity, weight heterogeneity, and spreading dynamics while removing a substantial fraction of edges.

DOI: [10.1103/4pg6-mtmt](https://doi.org/10.1103/4pg6-mtmt)

Subject Areas: Complex Systems

I. INTRODUCTION

In a range of practical applications including the simulation of epidemics or information cascades [1,2] and the visualization of large networks [3], it is helpful to alleviate computational burden by reducing the number of edges in a network while preserving key properties of interest for computations. This task is known as *graph sparsification* [4], and often involves identifying a *network backbone* consisting of a network subgraph in which important characteristics such as degree and strength distributions [5] or shortest paths [6] are preserved. As there are many possible objectives for sparsification, there have been many proposed methods for identifying backbones in networks [7,8]. Some of these methods can be used to sparsify unweighted graphs, but they are most often applied to weighted graphs in which edges with higher weight are more likely to be kept in the network backbone when the

weight signals edge importance with respect to the network structure or dynamics.

A number of existing methods for graph sparsification are stochastic, aiming to preserve properties such as network spectra with high probability when sampling edges based on their structural properties [9–11]. However, most network backbone methods are deterministic in nature and prune the network down to a single subgraph for analyses. Network backbone methods can generally be categorized as either *global* or *local* in nature [8]. Global backbone methods consider the whole structure of the network when considering whether or not to remove an edge, while local backbone methods consider only the neighborhood surrounding the node(s) at one or both ends of the edge in question.

The simplest method for weighted network backbone is a global thresholding procedure in which all edges with weight less than some predefined threshold are dropped from the network. Another straightforward global backbone method is to compute the minimum (maximum) spanning tree with respect to weights transformed to represent some notion of distance (similarity) between nodes [12]. Spanning trees have many useful applications but are naturally acyclic so cannot possibly preserve important network properties such as transitivity, clustering, or path lengths. The high salience skeleton introduced

*Contact author: alec.w.kirkley@gmail.com

Published by the American Physical Society under the terms of the [Creative Commons Attribution 4.0 International](https://creativecommons.org/licenses/by/4.0/) license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

in Ref. [6] is a global backbone method that aims to identify edges critical for routing along shortest paths in the network. This method has the appealing properties of being principled and relatively insensitive to its only user-chosen threshold, but has a computational burden that is at least quadratic in the number of nodes in practice and often does not preserve the global connectivity of the graph (as we will see in Sec. III B). Percolation thresholding, which imposes the maximum global weight threshold that allows the network to be a single connected component [13,14], provides an alternative global backbone method which alleviates this issue at the cost of often requiring a large fraction of edges to be retained in order to ensure all nodes are connected. Other global backbone methods tend to suffer from the same issues of scalability and/or placing tight restrictions on the backbone topology, with scalability a particular issue for methods that require the computation of shortest paths [15,16] or community structure [17].

Local backbone methods provide a scalable alternative to global methods by considering edges based only on nearby network structure, with the trade-off that they may fail to preserve large-scale properties. One of the most popular methods for local network backbone is the disparity filter [5], which drops edges if their weight falls below a prespecified significance level under a suitable null model relative to the other edges in a node's neighborhood. The Polya filter [18] generalizes the disparity filter to a whole family of Polya urn null models, allowing for greater flexibility in modeling weight heterogeneity. A number of other statistical methods have also been considered that aim to remove edges that are deemed insignificant under a suitable null model [19–22]. All of these methods have the requirement of a user-defined threshold or significance level that directly determines the number of edges present in the final backbone. While this approach can be advantageous for identifying multiple scales at which a network exhibits significant structural patterns [5], it also leaves arguably the most critical backbone choice (the number of edges to retain) up to the user and does not have a natural mechanism for selecting the single backbone representation to be used in an application.

Properly regularized Bayesian methods for *network reconstruction* have circumvented the need for fixing free parameters when inferring statistically significant network structure [23–25]. These methods are statistically principled and robust, and are in part aimed at addressing concerns that backbone and thresholding methods—by enforcing the removal of edges—may impose unwanted biases on the structure of the pruned networks and remove important connectivity information [26]. Despite their range of benefits, network reconstruction methods are usually not suitable for network backbone because they often *increase* the number of edges in a network when there is strong evidence for edges being omitted during the measurement process. They also tend to be computationally

expensive due to the need to search over edge configurations involving node pairs that do not have any observed edge, the number of which scales quadratically with the network size. There are fast approximations aimed to alleviate this issue and provide better asymptotic run-time scaling [27], although the resulting scaling is still worse than that of many backbone methods due to the inherently higher complexity of the reconstruction problem. Network backbone methods' comparatively low complexity and prioritization of sparsity in the final network representation are often advantageous compared to network reconstruction techniques when considering visualization and downstream tasks with high computational overhead [8].

There are also important classes of methods aimed at sparsifying networks so as to preserve or destroy one particular structural or dynamical property. For example, spectral sparsification [4] aims to sparsify a network while preserving the spectral properties of a particular graph operator such as the graph Laplacian or graph adjacency matrix. And network dismantling [28,29] aims to identify the minimal set of nodes or edges to remove in order to destroy any extensive connected components. While these methods are more effective than general network backbones for graph sparsification in specific problem contexts, network backbones are useful representations when a variety of exploratory and simulation-based analyses are performed on the same network, and the goal is to preserve connectivity, weight heterogeneity, and other structural and dynamical properties while reducing the dataset size for simpler manipulation and computation. As such, evaluation of network backbones is intrinsically multifaceted [8].

Here we develop fast, completely parameter-free methods for inferring global and local backbones of a weighted network based on the minimum description length (MDL) principle [30]. We adapt our framework to Bayesian model specifications in which total edge weight is fixed in expectation (“canonical” models, informally) and exactly (a “microcanonical” model), proving that simple greedy approaches can find the exact optimum in a subset of these models. Minimizing over these objectives provides the backbone that best compresses the edge weight structure in the network, while automatically selecting for the number of edges to retain in the backbone. Our methods provide flexible, principled, and interpretable solutions for network backbone that easily scale to large networks. We test our methods on a range of real and synthetic network data, finding that they compare favorably against existing methods in a number of dimensions while eliminating the need for tuning any parameters. Finally, we study an example application in network epidemic spreading, finding that our methods allow for the accurate estimation of the associated percolation transition on networks while substantially reducing the required run-time for the calculation.

II. METHODS

We start by deriving a general global backbone objective through maximum *a posteriori* (MAP) estimation in a Bayesian generative model of the network weights in which the total network weight is only fixed in expectation. We informally call this model a *canonical* model to emphasize that the exact total weight is not known ahead of time. (A truly “canonical” model is more restrictive as it requires a *maximum entropy* distribution with weight fixed in expectation, which in the present case is true only for the geometric and exponential distributions for integer-valued and real-valued weights, respectively.) We apply this generative model to the level of node neighborhoods to develop a local backbone model.

We then discuss the approximate equivalence between our canonical Bayesian generative model in the case of geometric weights and a simpler *microcanonical* Bayesian formulation that fixes the total network weight to an observed value. (Here we focus only on a truly “microcanonical” model that is uniform over all positive integer edge weight configurations with fixed total sums in the backbone and nonbackbone.) Finally, we demonstrate that our microcanonical objective permits a fast, exact greedy procedure for identifying the MDL-optimal backbones, and extend our proof of greedy optimality to a set of common weight distributions under the canonical model formulation.

Our methods are motivated by the MDL principle, which states that the optimal model for a dataset is the model that allows us to transmit the dataset using the fewest bits [30]. By considering both the information to transmit a dataset given a model *and* the information to transmit the model itself, the MDL principle provides a flexible framework for penalizing overly complex models to reduce the risk of overfitting. The MDL principle has been employed in the network context for a number of methods in the past, for the tasks of community detection [31–33], graph comparison [34,35], graph summarization [36,37], and for extracting structural regularities in dynamic networks and hypergraphs [38–41]. In the backbone context, the MDL principle allows us to naturally promote sparsity in the inferred backbone while encouraging the backbone to carry the most significant weights in the network. There are many possible weight distributions for our canonical backbone models, each resulting in a different MDL formulation, and so technically we are considering not one but *many* possible MDL formulations of the backbone problem.

A. Canonical models for global and local backbone

Our problem considers an input network $G = \{(i, j, w_{ij})\}$, which we represent as a set of E weighted tuples $e = (i, j, w_{ij})$. The tuple $e = (i, j, w_{ij})$ indicates an edge from a node i to a node j with weight $w_{ij} \geq 0$ that

represents the strength of interaction among i and j . The network G must be weighted, as the backbone methods we present find backbones that allow for compression of the edge weights in G . The graph G may have self-edges or multiedges—multiedges can be combined into a single edge weight—and G may be directed or undirected. In the undirected case we simply consider both (i, j, w_{ij}) and (j, i, w_{ij}) to be in G , with the edge (i, j, w_{ij}) appearing in an undirected backbone if either (i, j, w_{ij}) or (j, i, w_{ij}) is retained in the directed backbone (as done in Ref. [5]). Bipartite networks can be accommodated by our framework with no additional modifications. We will let $W = \sum_{e \in G} w_e$ denote the total edge weight in the weighted graph G . The goal of our network backbone method is to extract a subset $G^{(b)} \subseteq G$ of weighted edges that best compress the overall weight distribution in G . We call the subset $G^{(b)}$ the *backbone* of G , which has $E^{(b)} \leq E$ edges with a total weight of $W^{(b)} \leq W$.

To derive a global backbone objective, we can consider a generative process in which we start with an unweighted network G_u , and for each edge $e \in G_u$ we classify e as being either a backbone edge or nonbackbone edge by assigning it an indicator variable:

$$b_e = \begin{cases} 1 & e \text{ is a backbone edge} \\ 0 & e \text{ is not a backbone edge.} \end{cases} \quad (1)$$

This results in a binary vector $\mathbf{b}$ indexed by edge pairs $e = 1, \dots, E$ that partitions the set of edges into a backbone and nonbackbone. We then use a mixture model for the edge weights in which the weight w_e is generated according to the distribution $P(w_e | \theta_{b_e})$, where θ_{b_e} is a set of parameters for the distribution P that depend on whether or not the edge e is in the backbone. We therefore have two independent sets of model parameters θ_1 and θ_0 for the backbone and nonbackbone edges, respectively. This mixture model imposes a distinction between the weight statistics of the backbone and the nonbackbone edges, which is the essential feature utilized by the majority of weighted network backbone methods [5,13,18–22].

For the canonical model, we generate the backbone assignments $\mathbf{b} \sim \text{Bernoulli}(\pi_b)$, where $\pi_b \in [0, 1]$ is the probability of an edge being assigned to the backbone. This gives us a prior distribution over the backbone $\mathbf{b}$ of

$$P(\mathbf{b} | \pi_b) = \prod_{e \in G_u} \pi_b^{b_e} (1 - \pi_b)^{1-b_e}. \quad (2)$$

The likelihood of an observed weighted network G and a backbone assignment $\mathbf{b}$ given the model parameters and the unweighted network G_u can then be written as

$$\begin{aligned}
P(G, \mathbf{b} | \boldsymbol{\theta}_1, \boldsymbol{\theta}_0, \pi_b) &= P(G | \boldsymbol{\theta}_1, \boldsymbol{\theta}_0, \mathbf{b}) P(\mathbf{b} | \pi_b) \\
&= \prod_{e \in G_u} [\pi_b P(w_e | \boldsymbol{\theta}_1)]^{b_e} [(1 - \pi_b) P(w_e | \boldsymbol{\theta}_0)]^{1-b_e} \\
&= \pi_b^{E(b)} (1 - \pi_b)^{E-E(b)} \\
&\quad \times \prod_{e \in G^{(b)}(\mathbf{b})} P(w_e | \boldsymbol{\theta}_1) \prod_{e \in G \setminus G^{(b)}(\mathbf{b})} P(w_e | \boldsymbol{\theta}_0), \quad (3)
\end{aligned}$$

where $G^{(b)}(\mathbf{b}) \subseteq G$ is the subset of weighted backbone edges associated with the assignment $\mathbf{b}$. The above likelihood is (as implied in the model formulation) conditional on knowing the unweighted network G_u , but we remove explicit dependence on G_u in the notation for brevity.

Putting priors $P(\boldsymbol{\theta}_1)$, $P(\boldsymbol{\theta}_0)$, and $P(\pi_b)$ on the model parameters that are conjugate to the likelihoods $\prod_e P(w_e | \boldsymbol{\theta}_1)$, $\prod_e P(w_e | \boldsymbol{\theta}_0)$, and $P(\mathbf{b} | \pi_b)$, respectively, we can marginalize over $\{\boldsymbol{\theta}_1, \boldsymbol{\theta}_0, \pi_b\}$ to form the joint distribution for the canonical model,

$$\begin{aligned}
P_C^{(\text{global})}(G, \mathbf{b}) &= \int P(G, \mathbf{b} | \boldsymbol{\theta}_1, \boldsymbol{\theta}_0, \pi_b) P(\boldsymbol{\theta}_1) P(\boldsymbol{\theta}_0) \\
&\quad \times P(\pi_b) d\boldsymbol{\theta}_1 d\boldsymbol{\theta}_0 d\pi_b \\
&= B(E^{(b)} + 1, E - E^{(b)} + 1) \\
&\quad \times \int P(\boldsymbol{\theta}_1) \prod_{e \in G^{(b)}} P(w_e | \boldsymbol{\theta}_1) d\boldsymbol{\theta}_1 \\
&\quad \times \int P(\boldsymbol{\theta}_0) \prod_{e \in G \setminus G^{(b)}} P(w_e | \boldsymbol{\theta}_0) d\boldsymbol{\theta}_0, \quad (4)
\end{aligned}$$

where

$$B(x, y) = \frac{(x-1)!(y-1)!}{(x+y-1)!} \quad (5)$$

is the Beta function.

MAP estimation with this model then aims to solve the following optimization problem to identify the optimal backbone assignments $\hat{\mathbf{b}}$:

$$\hat{\mathbf{b}} = \operatorname{argmax}_{\mathbf{b} \in \{0,1\}^E} \{P(\mathbf{b} | G)\}. \quad (6)$$

This can be equivalently framed in terms of the *description length* $\mathcal{L}_C^{(\text{global})}(G^{(b)})$ of the canonical global backbone model, which we can parametrize with just the weighted backbone $G^{(b)}$ as

$$\begin{aligned}
\mathcal{L}_C^{(\text{global})}(G^{(b)}) &= -\log P_C^{(\text{global})}(G, \mathbf{b}) \\
&= \log(E+1) + \log \binom{E}{E^{(b)}} \\
&\quad - \log \left[\int P(\boldsymbol{\theta}_1) \prod_{e \in G^{(b)}} P(w_e | \boldsymbol{\theta}_1) d\boldsymbol{\theta}_1 \right] \\
&\quad - \log \left[\int P(\boldsymbol{\theta}_0) \prod_{e \in G \setminus G^{(b)}} P(w_e | \boldsymbol{\theta}_0) d\boldsymbol{\theta}_0 \right]. \quad (7)
\end{aligned}$$

We use the notation $\log(x) \equiv \log_2(x)$ for brevity, so that the description length is in units of bits. Equation (7) tells us, for an optimal encoding over networks and backbones under our assumptions about the weight distribution, how much information is required to transmit a specific network G and backbone partition $\mathbf{b}$ [42].

Solving Eq. (6) is equivalent to minimizing the description length of Eq. (7), giving the MDL objective,

$$\hat{G}^{(b)} = \operatorname{argmin}_{G^{(b)} \subseteq G} \{\mathcal{L}_C^{(\text{global})}(G^{(b)})\}, \quad (8)$$

where $\hat{G}^{(b)}$ is the MDL-optimal backbone. We can see that Eq. (7) has two contributions. The first two terms penalize the complexity of the backbone $G^{(b)}$ by encouraging the backbone size $E^{(b)}$ to be small, for $E^{(b)} \leq E/2$. These terms also encourage the backbone sizes $E^{(b)}$ to be large, for $E^{(b)} > E/2$ —the backbone $G^{(b)}$ is equally as compressive as its complement $G \setminus G^{(b)}$ since specifying the backbone edges is equivalent to specifying the nonbackbone edges given that the unweighted graph G_u is known. (This “bit flip symmetry” under the interchange $0 \Leftrightarrow 1$ is intrinsic to information theoretic methods such as mutual information measures [43].) However, as the backbones $G^{(b)}$ with sizes in the regime $E^{(b)} \geq E/2$ are nearly as dense as the original graphs, they do not provide meaningful sparsification, so we will focus on the regime $E^{(b)} \leq E/2$ —it is arguably not meaningful to suggest that more than half of the edges in G have statistically significant weights, regardless of the null model one chooses. The final two terms in Eq. (7) encourage the weight distributions in the backbone and nonbackbone to be distinct and for the model P to provide an effective summary of their variabilities.

We can generalize our canonical Bayesian model to the neighborhood level for the edge weights in each node neighborhood $\partial_i = \{(i, j, w_{ij})\}$ for all nodes j toward which i has a directed or undirected edge. We denote the strength s_i of each node i as its total incident weight $s_i = \sum_{e \in \partial_i} w_e$ and the degree of node i with $k_i = |\partial_i|$. In this model, each node i has its own probability $\pi_b(i)$ that one of its adjacent edges $e \in \partial_i$ is in the backbone $G^{(b)}$. Similarly, each node i has its own model parameters $\boldsymbol{\theta}_1(i)$ and $\boldsymbol{\theta}_0(i)$ specific to its neighborhood’s edge weights.

This parametrization allows the network neighborhoods ∂_i to have different weight distributions, which in turn allows the local backboning procedure to identify statistically significant weights in each neighborhood separately as is done in other local backboning methods [5].

We can then compute the joint distribution $P_C^{(\text{local})}(G, \mathbf{b})$ for this canonical local backboning model by integrating out the parameters $\{\pi_b(i), \theta_1(i), \theta_0(i)\}$ for each node neighborhood ∂_i and taking a product over the independent neighborhoods, assuming a directed network where (i, j) and (j, i) are treated independently. The result is equivalent to assuming N independent copies of the global backboning method, one for each neighborhood ∂_i , which each have their own associated backbone $\partial_i^{(b)}$ with total weight $s_i^{(b)}$ and number of edges $|\partial_i^{(b)}| = k_i^{(b)}$. The resulting joint distribution is given by

$$\begin{aligned} P_C^{(\text{local})}(G, \mathbf{b}) &= \prod_{i=1}^N P_C^{(\text{global})}(\partial_i, \mathbf{b}_{\partial_i}) \\ &= \prod_{i=1}^N B(k_i^{(b)} + 1, k_i - k_i^{(b)} + 1) \\ &\quad \times \prod_{i=1}^N \int P[\theta_1(i)] \prod_{e \in \partial_i^{(b)}} P[w_e | \theta_1(i)] d\theta_1(i) \\ &\quad \times \prod_{i=1}^N \int P[\theta_0(i)] \prod_{e \in \partial_i \setminus \partial_i^{(b)}} P[w_e | \theta_0(i)] d\theta_0(i), \end{aligned} \quad (9)$$

where $\mathbf{b}_{\partial_i} = \{b_e : e \in \partial_i\}$ is the backbone partition of the edges in the neighborhood ∂_i . The description length of interest is then

$$\mathcal{L}_C^{(\text{local})}(G^{(b)}) = -\log P_C^{(\text{local})}(G, \mathbf{b}) = \sum_{i=1}^N \mathcal{L}_C^{(\text{neig})}(\partial_i^{(b)}), \quad (10)$$

where

$$\begin{aligned} \mathcal{L}_C^{(\text{neig})}(\partial_i^{(b)}) &= \log(k_i + 1) + \log \binom{k_i}{k_i^{(b)}} \\ &\quad - \log \left[\int P[\theta_1(i)] \prod_{e \in \partial_i^{(b)}} P[w_e | \theta_1(i)] d\theta_1(i) \right] \\ &\quad - \log \left[\int P[\theta_0(i)] \prod_{e \in \partial_i \setminus \partial_i^{(b)}} P[w_e | \theta_0(i)] d\theta_0(i) \right]. \end{aligned} \quad (11)$$

Similarly to the global case, the local objective has bit flip symmetry, but in this case at the neighborhood level.

We therefore focus on the regime $k_i^{(b)} \leq k_i/2$, to promote sparse backbones—under the same logic as before, one cannot say that more than half of the edges in a neighborhood have statistically significant weights relative to the rest of the neighborhood, under any reasonable null model that treats nodes independently.

Equations (7) and (10) provide flexible objectives to solve for the backbone $G^{(b)}$ of a weighted network G by assessing global and local edge weight heterogeneity, respectively. These canonical model objectives allow for any underlying assumption about the weight distributions within and outside of the backbone $G^{(b)}$. They are adaptable to either discrete or continuous weights, and provide a means of model selection among different mixtures of underlying weights through the comparison of the description length values under different distributional assumptions for P .

B. Microcanonical models for global and local backboning

An alternative Bayesian formulation of network backbones can be done in a microcanonical framework in which we fix the network weight W rather than observing W as a result of generating the network edge weights. This allows us to calibrate the model so that the weight generative process being fit to the observed data G will always generate the correct total weight W . This has the benefit of not requiring the marginalization over any free parameters. Additionally, fixing statistics of interest exactly rather than in expectation (and vice versa) can permit easier analytical treatment as well as result in qualitatively distinct statistical behaviors in network ensembles [44], making microcanonical models worth exploring in their own right.

For the microcanonical generative model of network backbones, we assume that W is known in addition to the unweighted graph G_u . We first draw the number of backbone edges $E^{(b)}$ uniformly at random from the interval $[0, E]$, giving a prior of $P(E^{(b)}) = (E + 1)^{-1}$. Similarly, we draw the total backbone weight $W^{(b)}$ uniformly at random. Given that we have positive integer weights, we know that $W^{(b)} \geq E^{(b)}$, and $W^{(b)} \leq W - (E - E^{(b)})$, since each edge must have weight at least 1. Therefore, given $E^{(b)}$, $W^{(b)}$ is then drawn uniformly at random from the interval $[E^{(b)}, W - (E - E^{(b)})]$, resulting in the prior $P(W^{(b)} | E^{(b)}) = (W - E + 1)^{-1}$. We then draw the backbone assignments $\mathbf{b}$ uniformly at random from the set of all binary partitions with $\sum_{e=1}^E b_e = E^{(b)}$, of which there are $\binom{E}{E^{(b)}}$, giving a prior of $P(\mathbf{b} | E^{(b)}) = \binom{E}{E^{(b)}}^{-1}$. Finally, given the specification of the backbone edges and total backbone weight, we draw the weights of all edges in the backbone and nonbackbone uniformly at random given that the $E^{(b)}$ edges in the backbone must have weights that sum to $W^{(b)}$ and the $E - E^{(b)}$ edges in the nonbackbone must have

weights that sum to $W - W^{(b)}$. The number of configurations of k positive integers that sum to n is called the number of *compositions* of the integer n into k parts, and is given by $\binom{n-1}{k-1}$, so the probability of a particular configuration of the backbone and nonbackbone weights given their sums is $P(G|\mathbf{b}, W^{(b)}, E^{(b)}) = \binom{W^{(b)}-1}{E^{(b)}-1} \binom{W-W^{(b)}-1}{E-E^{(b)}-1}^{-1}$.

Putting this all together, the full joint distribution of the microcanonical model can be written as

$$\begin{aligned} P_M^{(\text{global})}(G, \mathbf{b}) &= P(E^{(b)}) \times P(W^{(b)}|E^{(b)}) \times P(\mathbf{b}|E^{(b)}) \\ &\quad \times P(G|\mathbf{b}, W^{(b)}, E^{(b)}) \\ &= (E+1)^{-1} \times (W-E+1)^{-1} \times \binom{E}{E^{(b)}}^{-1} \\ &\quad \times \binom{W^{(b)}-1}{E^{(b)}-1}^{-1} \binom{W-W^{(b)}-1}{E-E^{(b)}-1}^{-1}, \end{aligned} \quad (12)$$

which has description length

$$\begin{aligned} \mathcal{L}_M^{(\text{global})}(G^{(b)}) &= -\log P_M^{(\text{global})}(G, \mathbf{b}) \\ &= \log(E+1) + \log(W-E+1) + \log \binom{E}{E^{(b)}} \\ &\quad + \log \binom{W^{(b)}-1}{E^{(b)}-1} + \log \binom{W-W^{(b)}-1}{E-E^{(b)}-1}. \end{aligned} \quad (13)$$

As before, this objective can be minimized over backbones $G^{(b)}$ to find the MDL-optimal backbone under the microcanonical global model.

We can also formulate a microcanonical local backbone model in a similar way, drawing $\{k_i^{(b)}, s_i^{(b)}, \mathbf{b}_{\partial_i}, \partial_i^{(b)}, \partial_i \setminus \partial_i^{(b)}\}$ using hierarchical uniform priors analogous to those for $\{E^{(b)}, W^{(b)}, \mathbf{b}, G^{(b)}, G \setminus G^{(b)}\}$, respectively, to form each neighborhood-level term $P_M^{(\text{global})}(\partial_i^{(b)}, \mathbf{b}_{\partial_i}|s_i)$ in an independent factorization analogous to Eq. (9). The only additional consideration we must make for the microcanonical model is that, since s_i is assumed known for all $i = 1, \dots, N$, we need to put a uniform prior on $\mathbf{s} = \{s_1, \dots, s_N\}$ given the known total network weight W . By adding this prior, we can compare the global and local microcanonical description lengths directly, since both models assume the same prior knowledge (G_u and W). Since each edge has weight at least 1, node i must have a strength that satisfies $s_i \geq k_i$. In other words, a uniform distribution of strengths $\mathbf{s}$ will consider all possible ways to distribute the *excess weight* $W - E$ onto the N nodes, allowing for some nodes to receive no excess weight (in other words, $s_i = k_i$). A uniform prior on $\mathbf{s}$ is then equivalent to generating a random composition of $W - E$ as N non-negative integers, allowing the integers to potentially be zero, of which there are $\binom{N+W-E-1}{W-E}$ possible combinations.

The final prior for $\mathbf{s}$ is then $P(\mathbf{s}) = \binom{N+W-E-1}{W-E}^{-1}$. This prior is a global constant that does not impact the inferred backbones, but is technically necessary for a direct comparison of the global and local description length values for model comparison.

Including the prior on the strengths $P(\mathbf{s})$, we have a description length for the local microcanonical model of

$$\begin{aligned} \mathcal{L}_M^{(\text{local})}(G^{(b)}) &= -\log P(\mathbf{s}) - \log P_M^{(\text{local})}(G^{(b)}, \mathbf{b}|\mathbf{s}) \\ &= -\log P(\mathbf{s}) - \log \left(\prod_{i=1}^N P_M^{(\text{global})}(\partial_i^{(b)}, \mathbf{b}_{\partial_i}|s_i) \right) \\ &= \log \binom{N+W-E-1}{W-E} + \sum_{i=1}^N \mathcal{L}_M^{(\text{neig})}(\partial_i^{(b)}), \end{aligned} \quad (14)$$

where

$$\begin{aligned} \mathcal{L}_M^{(\text{neig})}(\partial_i^{(b)}) &= \log(k_i + 1) + \log(s_i - k_i + 1) + \log \binom{k_i}{k_i^{(b)}} \\ &\quad + \log \binom{s_i^{(b)}-1}{k_i^{(b)}-1} + \log \binom{s_i - s_i^{(b)}-1}{k_i - k_i^{(b)}-1}. \end{aligned} \quad (15)$$

One can show (see Appendix A) that the microcanonical global and neighborhood-level description lengths of Eqs. (13) and (15) are asymptotically equivalent to the canonical description lengths of Eqs. (7) and (11), respectively, when weights follow a geometric distribution. The full canonical local description length of Eq. (10) with geometric weights is equivalent to the full microcanonical local description length of Eq. (14) only up to the term $\log \binom{N+W-E-1}{W-E}$ associated with the prior $P(\mathbf{s})$, which is a global constant that does not affect the inferred backbones.

One of the major benefits of using MDL-based learning objectives is that they naturally provide a simple criterion for model selection—the model providing the lowest description length at the end of the fitting process is the best model for the data [30]. We can then determine whether a global or local backbone, as well as which weight model, provides the most parsimonious description of the observed network G , by fitting the objectives in Eqs. (7), (10), (13), and (14) then comparing the resulting description lengths. The common downside of MDL objectives is that they can be challenging to minimize, often requiring sophisticated Markov chain Monte Carlo methods [45]. However, we show in the next section that the microcanonical description lengths in Eqs. (13) and (14) can be minimized exactly using a fast greedy algorithm. We can extend this proof of optimality to a set of common weight distributions under the canonical formulations in Eqs. (7) and (10).

C. Optimization

All four of the MDL backbone objectives of Secs. II A and II B—Eqs. (7), (10), (13), and (14)—must be minimized over sets of backbone edges $G^{(b)} \subseteq G$ to identify an MDL-optimal backbone under each description length formulation. With naive optimization by enumeration of all possible backbones $G^{(b)}$, this would require 2^E evaluations of a given objective—one for each possible backbone—which is intractable for all but the smallest networks. We show here that for the microcanonical objectives of Eqs. (13) and (14), as well as for a wide variety of weight distributions for the canonical models of Eqs. (7) and (10), one can find the MDL-optimal backbone using simple greedy procedures requiring only $O(E \log E)$ operations—the bottleneck consisting of sorting the set of edges G by weight at the beginning of the algorithm.

We can analyze the microcanonical objectives first, then generalize our results to the canonical objectives. Consider a greedy procedure in which the backbone $G^{(b)}$ starts empty and we add edges $e \in G$ to $G^{(b)}$ in decreasing order of weight w_e , stopping the procedure when $E^{(b)} = E/2$ edges have been added to the backbone. (As discussed in Sec. II A, given the bit flip symmetry of the description length objectives with respect to $\mathbf{b}$, the regime of interest for backbone is $E^{(b)} \leq E/2$.) We then scan over the past solution candidates and identify the one with the lowest description length to find the optimal backbone. Details of this procedure are shown in Algorithm 1. During this algorithm, at all steps we must necessarily have that

$$\frac{W^{(b)}}{E^{(b)}} \geq \frac{W}{E} \geq \frac{W - W^{(b)}}{E - E^{(b)}}. \quad (16)$$

In other words, the average edge weight in the backbone $G^{(b)}$ is at least as high as the average edge weight in the total network G , which is in turn at least as high as the average edge weight in the nonbackbone $G \setminus G^{(b)}$.

We can now consider the change in description length,

$$\begin{aligned} \Delta \mathcal{L}_M^{(\text{global})}(W^{(b)} + 1) &= \mathcal{L}_M^{(\text{global})}(G_+^{(b)}) - \mathcal{L}_M^{(\text{global})}(G^{(b)}) \\ &= \log \frac{W^{(b)}}{W^{(b)} - E^{(b)} + 1} \frac{(W - W^{(b)}) - (E - E^{(b)})}{W - W^{(b)} - 1}, \end{aligned} \quad (17)$$

where $E_+^{(b)} = E^{(b)} \geq 1$ and $W_+^{(b)} = W^{(b)} + 1$ for the backbone $G_+^{(b)}$. Equation (17) gives the change in description length induced by an increase in the total backbone weight $W^{(b)}$ for a fixed number of backbone edges $E^{(b)}$. Rearranging the argument of the logarithm, we can see that $\Delta \mathcal{L}_M^{(\text{global})}(W^{(b)} + 1) < 0$ —in other words, the description length is decreasing in $W^{(b)}$ —when

Algorithm 1. Global backbone [Eqs. (7) and (13)].

Input: Graph $G = \{(e = (i, j), w_e = w_{ij})\}_{e=1}^E$
Output: Global backbone $G^{(b)} \subseteq G$
Initialization:
 -Sort edges G by decreasing edge weight w_e ;
Algorithm:
 -Set $W^{(b)} = 0$, $G^{(b)} = []$, $\mathcal{L}_S = []$;
 -Compute $\mathcal{L} = \mathcal{L}^{(\text{global})}(W^{(b)} = 0, E^{(b)} = 0)$ using Eq. (7) or Eq. (13) (whichever is being minimized) and append to $\mathcal{L}_S$;
for $E^{(b)} \leftarrow 1$ **to** $\lfloor E/2 \rfloor$ **do**
 $i, j, w_{ij} \leftarrow G[E^{(b)} - 1]$;
 $W^{(b)} \leftarrow W^{(b)} + w_{ij}$;
 Append (i, j, w_{ij}) to $G^{(b)}$;
 $\mathcal{L} \leftarrow \mathcal{L}^{(\text{global})}(W^{(b)}, E^{(b)})$;
 Append $\mathcal{L}$ to $\mathcal{L}_S$;
end
 -Set $G^{(b)} \leftarrow G^{(b)}[:\text{argmin}(\mathcal{L}_S)]$;
return $G^{(b)}$;

$$W^{(b)} > (E^{(b)} - 1) \frac{W - 1}{E - 2}. \quad (18)$$

Now, using the inequality of Eq. (16) and $1 \leq E^{(b)} \leq E/2$, we have that for the greedy procedure in Algorithm 1

$$\begin{aligned} W^{(b)} &= (E^{(b)} - 1) \frac{W^{(b)}}{E^{(b)}} \frac{E^{(b)}}{E^{(b)} - 1} \\ &\geq (E^{(b)} - 1) \frac{W}{E} \frac{E}{E - 2} \\ &> (E^{(b)} - 1) \frac{W - 1}{E - 2}, \end{aligned} \quad (19)$$

which is precisely the condition required in Eq. (18) for the description length $\mathcal{L}_M^{(\text{global})}$ to be decreasing in the backbone weight $W^{(b)}$ for fixed $E^{(b)}$. In other words, the best backbone $G^{(b)}$ for any number of edges $E^{(b)} \in [1, E/2]$ is the one with the highest total weight $W^{(b)}$, and so Algorithm 1 will always be able to find an optimal solution for the backbone $G^{(b)}$. (The other optimal solution has backbone $G \setminus G^{(b)}$, due to the bit flip symmetry.) Note that this condition still allows for the backbone $G^{(b)}$ to potentially be empty—the description length may be minimized at $E^{(b)} = 0$, if the weights are sufficiently homogeneous. Using a similar argument, we can also show that the canonical description length of Eq. (7) is minimized with the greedy procedure in Algorithm 1 for common weight distributions including the Poisson, geometric, and exponential distributions, which all fall under the natural exponential family (NEF) of distributions [46] (see Appendix D).

Mapping the global variables $\{E^{(b)}, W^{(b)}, \mathbf{b}, G^{(b)}, G \setminus G^{(b)}\}$ to the neighborhood level variables

$\{k_i^{(b)}, s_i^{(b)}, \mathbf{b}_{\partial_i}, \partial_i^{(b)}, \partial_i \setminus \partial_i^{(b)}\}$ as before, we have that the best neighborhood-level backbone $\partial_i^{(b)}$ for any node degree $k_i^{(b)} \in [1, k_i/2]$ is the one with the highest total weight $s_i^{(b)}$, and so the application of the greedy procedure in Algorithm 1 to each neighborhood ∂_i separately gives the MDL-optimal backbone $G^{(b)} = \cup_{i=1}^N \partial_i^{(b)}$ for the local objectives of Eqs. (10) and (14). Details of this procedure are shown in Algorithm 2.

The computational complexity of both Algorithms 1 and 2 is $O(E \log E)$, with the bottleneck being the sorting of G during initialization. Both algorithms are therefore only slightly slower in practice than constructing G in the first place. These algorithms are also completely nonparametric, with no required threshold for determining the edges to retain in the backbone, as the optimal edge set is determined automatically through parsimony and the MDL principle.

The final description lengths one is interested in comparing in practice are

- (i) $\mathcal{L}_C^{(\text{global})}(\hat{G}^{(b)})$ [minimum of Eq. (7)]
- (ii) $\mathcal{L}_C^{(\text{local})}(\hat{G}^{(b)})$ [minimum of Eq. (10)]
- (iii) $\mathcal{L}_M^{(\text{global})}(\hat{G}^{(b)})$ [minimum of Eq. (13)]
- (iv) $\mathcal{L}_M^{(\text{local})}(\hat{G}^{(b)})$ [minimum of Eq. (14)]

where $\hat{G}^{(b)}$ is the backbone minimizing the corresponding objective for each expression. The objective giving the lowest description length provides the most parsimonious summary of the data G and should be the preferred backbone model.

It is also useful to construct an *inverse compression ratio* to determine the extent to which we can compress a

network using its MDL-optimal backbones relative to a naive encoding where we do not use any backbone (equivalent mathematically to setting $G^{(b)} = \{\}$ as the empty graph). For the experiments in Sec. III we will use the microcanonical objectives of Eqs. (13) and (14). (These are, as shown in Appendix A, asymptotically equivalent to the canonical backbone models with geometric weights.) We will compare the levels of compression for the two methods using the inverse compression ratio,

$$\eta^{(\text{global/local})} = \frac{\mathcal{L}_M^{(\text{global/local})}(\hat{G}^{(b)})}{\max[\mathcal{L}_M^{(\text{global})}(\{\}), \mathcal{L}_M^{(\text{local})}(\{\})]}, \quad (20)$$

where global/local indicates the model of interest. Equation (20) is a useful ratio in practice for the following reasons.

- (1) It is normalized in $[0, 1]$ even for very small networks where $\mathcal{L}_M^{(\text{global})}(\{\})$ and $\mathcal{L}_M^{(\text{local})}(\{\})$ may differ considerably.
- (2) It is proportional to the final description length values, so that $\eta^{(\text{global})} > \eta^{(\text{local})}$ only when the global method compresses better than the local method in absolute terms (and vice versa for $\eta^{(\text{global})} < \eta^{(\text{local})}$).

Figure 1 shows a diagram of both the global and local backbone methods for a small synthetic example network, along with the inverse compression ratios for the microcanonical models. The code implementing these algorithms can be found in the PANINIPy package for nonparametric network inference [47].

III. RESULTS

A. Synthetic backbone reconstruction

As a first set of experiments to test our backbone methods, we examine the ability of these methods to reconstruct planted backbone structure in synthetic data simulated from the canonical generative models described in Sec. II A. In the experiments we first generate an unweighted, directed graph with a fixed number of out edges k for each node i . The focus of the MDL backbone methods is on the edge weight distributions, and degree fluctuations simply provide noise in the reconstruction results so degrees are set to be uniform across nodes. We then generate the weights for this network using the global and local canonical generative schemes described in Sec. II A, with the priors $P(\pi_b)$ and $P(\theta_0)$ (for the global generative model) or $P[\theta_0(i)]$ (for the local generative model) uniform over the range $[0, 1]$ and $P(w_e|\theta_{b_e})$ or $P[w_e|\theta_{b_e}(i)]$ set to a geometric distribution. As a measure of how “noisy” the backbone is—in other words, how similar the backbone weights and nonbackbone weights are—we fix a parameter $\gamma = \theta_1/\theta_0$ for each simulation which is equivalent to the ratio of the mean nonbackbone edge weight and mean backbone edge weight. We set $\theta_1 = \gamma\theta_0$ for each simulation to fix the ratio of the mean weight, with

Algorithm 2. Local backbone [Eqs. (10) and (14)].

Input: -Graph $G = \{(e = (i, j), w_e = w_{ij})\}_{e=1}^E$;
 -directed (boolean)

Output: Local backbone $G^{(b)} \subseteq G$

Initialization:
 -If directed is False, duplicate edges in G to include both directions (i, j) and (j, i) ;
 -Sort edges G by decreasing edge weight w_e ;
 -Iterate through G and construct adjacency list a such that $a[i] = \partial_i$. $a[i]$ is sorted by weight since G was sorted;

Algorithm:
 -Set $G^{(b)} = \{\}$;
for $i \leftarrow 1$ **to** N **do**
 Compute $\partial_i^{(b)}$ by applying Algorithm 1 to $a[i]$;
 -Ignore initialization step;
 -Use associated global model (C/M) for $\mathcal{L}$;
 Append $\partial_i^{(b)}$ to $G^{(b)}$;
end
 -If directed is False, sort node indices in each edge $(i, j, w_{ij}) \in G^{(b)}$ and remove duplicates to convert back to undirected graph;
return $G^{(b)}$;

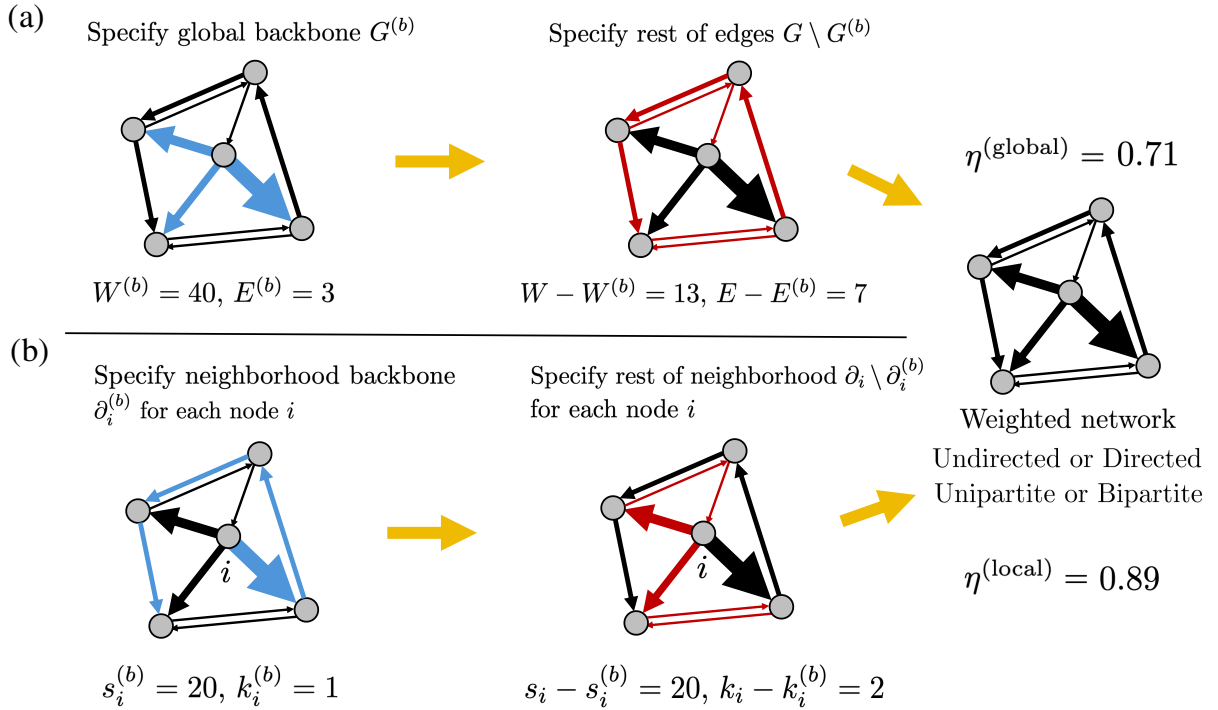


FIG. 1. Global and local MDL backbones for a small example network. (a) Schematic of the global backbone objectives [Eqs. (7) and (13)], with the total weight and number of edges indicated for the backbone $G^{(b)}$ (left) and remaining edges (right) of a network G . For the canonical objective [Eq. (7)], edge weights w_e are sampled according to $w_e \sim P(\cdot | \theta_{b_e})$, where $b_e \in \{0, 1\}$ is an indicator variable for whether or not e is in the backbone. Meanwhile, for the microcanonical objective [Eq. (14)], edge weights are distributed uniformly in the backbone and nonbackbone given the constraints imposed by $E^{(b)}, W^{(b)}$, which are sampled uniformly given the total weight W and number of edges E in the network. (b) Schematic of the local backbone transmission objectives [Eqs. (10) and (14)], applied to out neighborhoods ∂_i for each node i . The total strength $s_i^{(b)}$ and degree $k_i^{(b)}$ are indicated for the backbone neighborhood $\partial_i^{(b)}$ around an example node i . For both panels, backbone edges are highlighted in blue, while nonbackbone edges are highlighted in red. The local network backbone $G^{(b)}$ is the union $G^{(b)} = \cup_{i=1}^N \partial_i^{(b)}$ of the neighborhood backbones. For this example network, the global method provides a more compressive backbone, giving an inverse compression ratio [Eq. (20)] of $\eta^{(\text{global})} = 0.71$ versus $\eta^{(\text{local})} = 0.89$ for the local backbone.

the exact mean weights $1/\theta_1$ and $1/\theta_0$ (and consequently the weight variances) varying for each trial based on the sampled value $\theta_0 \sim \text{Uniform}([0, 1])$.

We test the reconstruction and compression capability for the global and local MDL backbone methods in Sec. II B using two measures. The first is the Jaccard similarity index between the planted backbone $G_{\text{planted}}^{(b)}$ generated from the model and the inferred backbone $G_{\text{MDL}}^{(b)}$ for each method. Since the networks G are represented as edge sets, the Jaccard similarity index between a network G_1 and a network G_2 can be computed as

$$\text{Jaccard similarity}(G_1, G_2) = \frac{|G_1 \cap G_2|}{|G_1 \cup G_2|}. \quad (21)$$

The Jaccard similarity in this case tells us how well the global and local MDL backbone algorithms can recover planted structure from their own (approximate canonical) generative models, and falls in the range $[0, 1]$ with 0

indicating an inferred empty or complete graph and 1 indicating perfect recovery. There are many alternative suitable measures for computing the graph similarity such as the network mutual information [43], but in practice they give the same qualitative trends as the Jaccard similarity for this example. The second measure we use to evaluate the methods is the inverse compression ratio η [Eq. (20)], which tells us how well the methods compress the synthetic network data.

Figure 2 shows the results of our reconstruction experiments, which were run for random directed regular graphs with $N = 100$ nodes, degree $k \in \{5, 20, 100\}$ —allowing self-loops—and varying noise levels $\gamma \in [0, 1]$. Both the global (left-hand column of panels) and local (right-hand column of panels) generative models were used, to determine the extent to which each backbone method is robust under model misspecification. We can see from Fig. 2(a) that, as expected, the reconstruction performance becomes worse as we increase the noise level γ . At $\gamma \approx 10^{-3}$ we observe near perfect backbone recovery for both methods

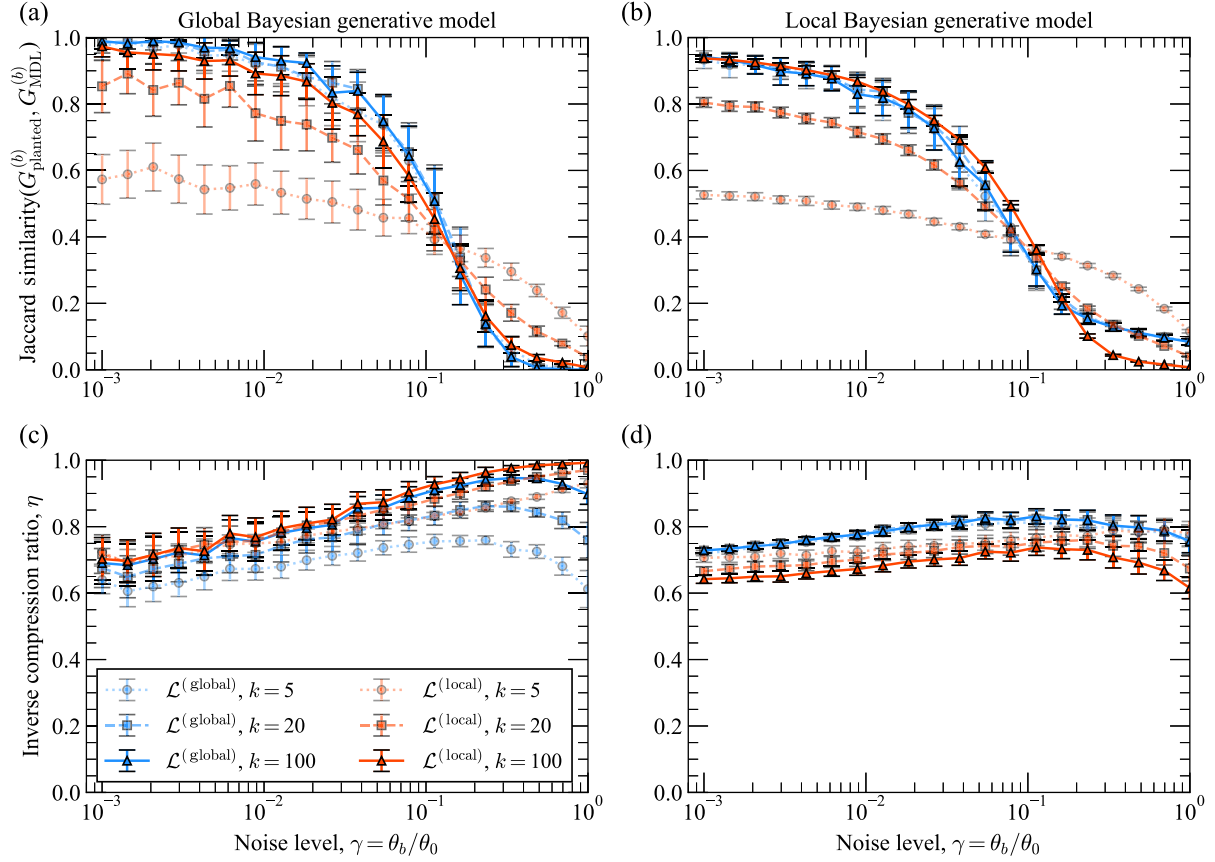


FIG. 2. Reconstruction of planted backbone structure in synthetic network data. (a) Network Jaccard similarity [Eq. (21)] between the planted and inferred backbones $G_{\text{planted}}^{(b)}$ and $G_{\text{MDL}}^{(b)}$, using the microcanonical global and local MDL objectives in Eqs. (13) and (14), respectively. The difficulty of the reconstruction task is varied by changing the noise parameter $\gamma = \theta_1/\theta_0$ —the ratio of the means for geometric weight distributions in the nonbackbone and backbone—as well as the degree k of the nodes in the underlying random regular directed graph with $N = 100$ nodes. The synthetic networks in this panel are drawn from the global canonical model in Eq. (4) with geometric weights. (b) Network Jaccard similarity between the planted and inferred backbones, for networks generated with the local canonical model of Eq. (9) with geometric weights. (c) Inverse compression ratio [Eq. (20)] for the same set of experiments as in (a). (d) Inverse compression ratio for the same experiments as in (b). Markers indicate averages over 100 simulations and error bars represent one standard error in the mean.

in the high degree regime, but at $\gamma \approx 1$ we see that neither inferred backbone has any significant shared structure with the planted backbone. We can see that the global method is insensitive to degree k while the local method is sensitive to k , since lower k will result in greater weight fluctuations within the node neighborhoods but less significant fluctuations at the global level since $E = 500$ even for the lowest degree value ($k = 5$) studied. We also see that, as expected, the global MDL method has better performance in the low noise regime than the local MDL method, since it is (approximately) the Bayes-optimal algorithm. However, for higher levels of noise γ we see that the local backbone method achieves better performance, which improves for smaller degrees k . This is because in this regime the global backbone method tends to infer very sparse backbones, but the local method can still find backbones with a moderate level of connectivity due to considering

within-neighborhood weight fluctuations rather than global weight fluctuations.

In Fig. 2(b) we show the reconstruction performance for the same set of experiments but for the local generative model, which allows the model parameters $\pi_b(i)$, $\theta_0(i)$, $\theta_1(i)$ to vary for each node neighborhood ∂_i . This causes fluctuations in the weight distributions across nodes, making it more challenging in general to infer the correct backbone structure. Indeed, we can observe in Fig. 2(b) that both methods exhibit a modest drop in reconstruction performance, with the local MDL method now outperforming the global MDL method in the low noise regime for high degrees. Neither method is capable of perfect reconstruction for low noise levels in this more challenging task, but both can still recover the planted backbone with reasonable accuracy.

We plot the inverse compression ratio [Eq. (20)] versus the noise level γ in Figs. 2(c) and 2(d) for the global and

local generative models, respectively. We see that for the global generative model the global MDL method outperforms the local MDL method, and for the local generative model the local MDL method outperforms the global MDL method. This is consistent with the global and local methods being approximately Bayes optimal for the global and local generative models, respectively, due to the asymptotic equivalence discussed in Appendix A. In both cases substantial compression is possible using both methods.

In Appendix B, we demonstrate numerically the optimality of the greedy algorithm in similar reconstruction tests on much smaller networks where exact enumeration over backbones is possible, to support the calculations in Sec. II C.

The results in Fig. 2 demonstrate the capability for these methods to identify meaningful backbone structure and provide considerable compression of network data in the presence of noise and model misspecification. In the next section we compare the MDL methods against existing methods on synthetic data with various levels of planted global and local homogeneity in the edge weights.

B. Comparison on synthetic networks

The MDL methods we propose in this paper are fully nonparametric, in contrast to many existing methods which require the specification of the desired number of edges $E^{(b)}$ for the backbone or require a significance level at which to retain edges under some null model for the weights [5,8,18,22]. This makes direct comparison with existing methods challenging, since it is unclear how to choose free parameters for methods that require them.

We therefore select a few popular existing methods for global and local backbone for comparison, with parameters fixed to common values. The first method we use for comparison is the disparity filter [5], which is a principled inference-based method for local backbone that selects edges below a prespecified significance level α within each node neighborhood for the backbone. The null model used in the disparity filter is one in which the weight assignment is uniform conditioned on the node degree and strength. For more direct comparison with our own local method, for one baseline we fix $E^{(b)} = |G_{\text{local}}^{(b)}|$ as the number of edges to retain in the disparity filter and set α to obtain the desired number of edges. For the other disparity filter baseline, we set the p -value threshold to the common value of $\alpha = 0.05$. We constrain the method to focus on out neighborhoods for these synthetic examples.

For the first global backbone baseline, we use the high saliency skeleton [6], which computes the saliency of a link based on its occurrence frequency in the shortest path trees rooted at each node, giving a succinct global view of edge importance. It is observed that the saliency distribution of links in real networks is highly bimodal, so the skeleton is not sensitive to the specific choice of the saliency cutoff for

the backbone, which is chosen to be the center of the saliency distribution (0.5). The same methodology is applied here for the baseline. Because of its computation of the shortest path tree for each node, the high saliency skeleton can become too computationally expensive for large networks, so for $N > 10000$ we randomly sample the shortest path trees for 10000 nodes to estimate the link saliency. For the other global backbone baseline, we use the weighted percolation threshold method [13,14], which consists of adding the edges to the network in decreasing order of weight until the network forms a fully connected component. In Fig. 3 we plot the results of all six measures applied to the workplace contacts dataset of Ref. [48], obtained using the procedures described in Sec. III C.

The synthetic networks used for the comparison experiments were generated using the following procedure. First, as in Sec. III A, an unweighted network of $N = 1000$ nodes is generated as a k -regular random directed graph with the specified out degree $k = 50$. Given an input parameter W for the total edge weight, all edges are then assigned weight 1 and the excess weight $W - Nk$ is distributed across the nodes' out neighborhoods using a symmetric Dirichlet-multinomial distribution with concentration parameter h_{str} . This allows us to tune the level of homogeneity in the out strengths $\{s_i\}$ across nodes with a single parameter h_{str} : $h_{\text{str}} \rightarrow 0$ places all the excess edge weight in a single out neighborhood ∂_i to give a highly heterogeneous strength distribution, and $h_{\text{str}} \rightarrow \infty$ distributes all the weight equally across nodes to give a perfectly homogeneous strength distribution. We then distribute the total excess edge weight $s_i - k$ among the out edges in the neighborhood ∂_i using another symmetric Dirichlet-multinomial distribution, this time with concentration parameter h_{neig} . $h_{\text{neig}} \rightarrow 0$ places all the excess edge weight on a single edge w_{ij} to give a highly heterogeneous weight distribution in each neighborhood, and $h_{\text{neig}} \rightarrow \infty$ distributes all the weight equally across edges to give a perfectly homogeneous weight distribution within the neighborhoods. The synthetic networks therefore depend on the four parameters $\{N, W, h_{\text{str}}, h_{\text{neig}}\}$, which we vary in our experiments.

We compare backbone methods using multiple indicators, similar to the methodology used in Ref. [8]. We compare the fraction of edges $E^{(b)}/E$ and total weight $W^{(b)}/W$ appearing in the backbone to get a sense of how sparse the inferred backbones are with each method. We also compute the inverse compression ratio [Eq. (20)], using the global description length [Eq. (13)] for the global methods (global MDL, high saliency skeleton, percolation) and the local description length [Eq. (14)] for the local methods (local MDL, disparity filters) to see how well each method compresses the data. To examine the extent to which the backbones maintain global connectivity, we compute the fraction of nodes $N^{(b)}/N$ with nonzero degree in the backbone and the relative *reachability* R , defined by

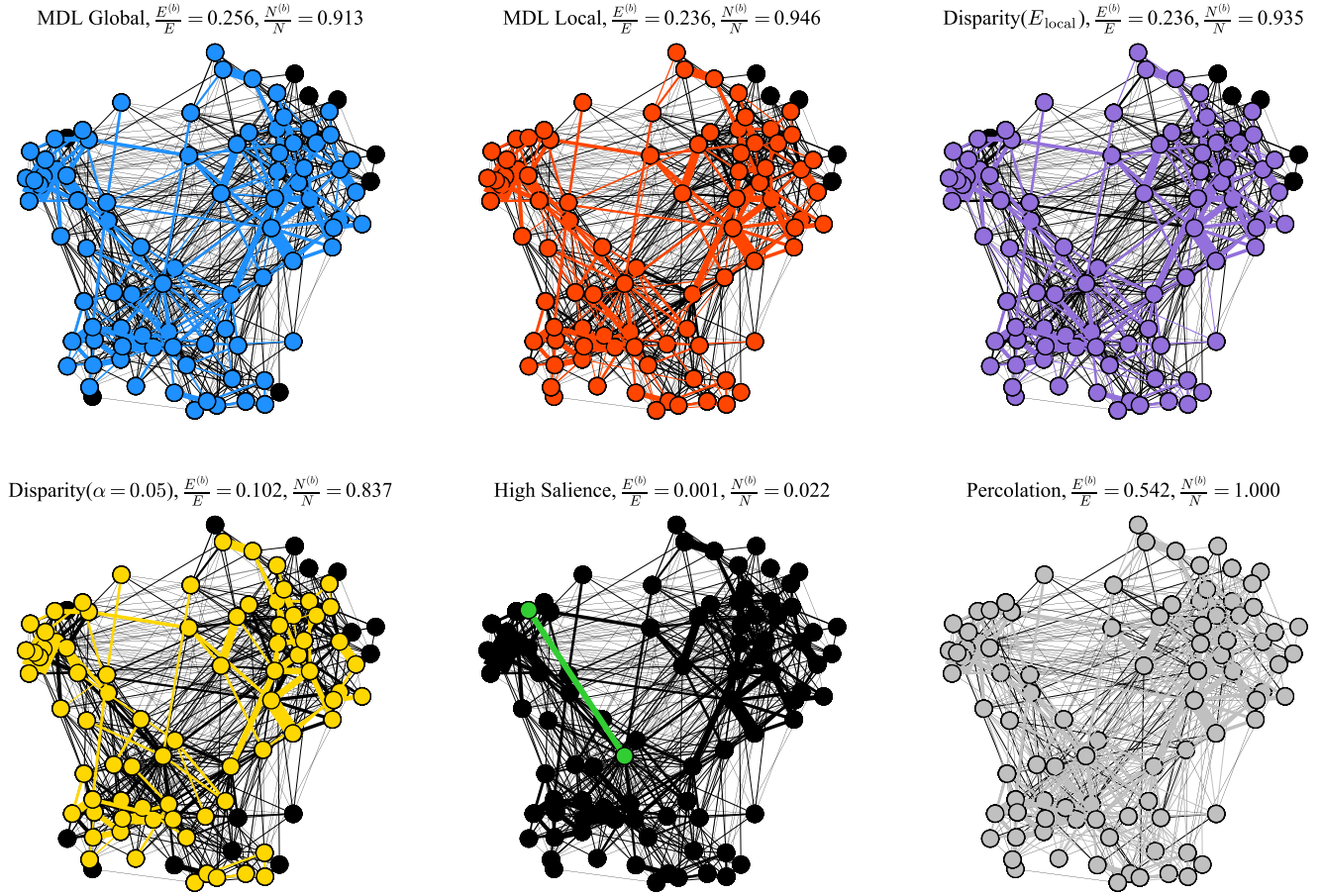


FIG. 3. Six backboning methods applied to a network of workplace contacts. All six methods used for the tests in Sec. III were applied to the workplace contacts network of Ref. [48], obtained from the Netzschleuder repository [49] along with the other networks described in Sec. III C. Edges in the backbone are highlighted in color, while all other edges are black. Edges are scaled proportionally to weight. The fraction of edges $E^{(b)}/E$ and nodes $N^{(b)}/N$ retained in the backbone are listed alongside each backboning method above the corresponding plot. The backbone densities and node connectivity (in terms of number of edges retained and number of nonisolated nodes in the backbone) observed for this network are fairly consistent with those observed for other real networks. The percolation backbone has the highest density; the disparity filter with $\alpha = 0.05$ and the high saliency skeleton have the lowest density at the cost of the lowest connectivity; and the three methods of the top row (global MDL, local MDL, and disparity filter with E equal to the local MDL method) retain most of the node connectivity with lower edge densities.

$$R(G^{(b)}) = \frac{\text{no. pairs } i, j \text{ with directed path } i \rightarrow j \text{ in } G^{(b)}}{\text{no. pairs } i, j \text{ with directed path } i \rightarrow j \text{ in } G}. \quad (22)$$

Since $R(G^{(b)})$ can become computationally prohibitive for large networks, for $N > 10\,000$ we randomly sample a subgraph of 10 000 nodes to estimate the reachability. We also determine the extent to which the strength distribution has been preserved across nodes by computing the Hellinger distance between the strengths $\{s_i\}_{i=1}^N$ in the original network G and the strengths $\{s_i^{(b)}\}_{i=1}^N$ in each backbone $G^{(b)}$ using

$$D_{\text{str}}(G, G^{(b)}) = \sqrt{\frac{1}{2} \sum_{i=1}^N (\sqrt{p_i} - \sqrt{q_i})^2}, \quad (23)$$

where $p_i = s_i/W$ and $q_i = s_i^{(b)}/W^{(b)}$.

In Fig. 4 we plot the fraction of edges (top row) and weight (bottom row) retained in the different backbones of the different backboning methods versus the synthetic model parameters. Unless otherwise specified as an independent variable, in all simulations we set $N = 1000$, $W/N = 1000$, $k = 50$, and $h_{\text{str}} = h_{\text{neig}} = 0.1$. There are a few major trends we can observe here. The first is that when $h_{\text{neig}} = 0.1$ is held constant at a moderate value (first

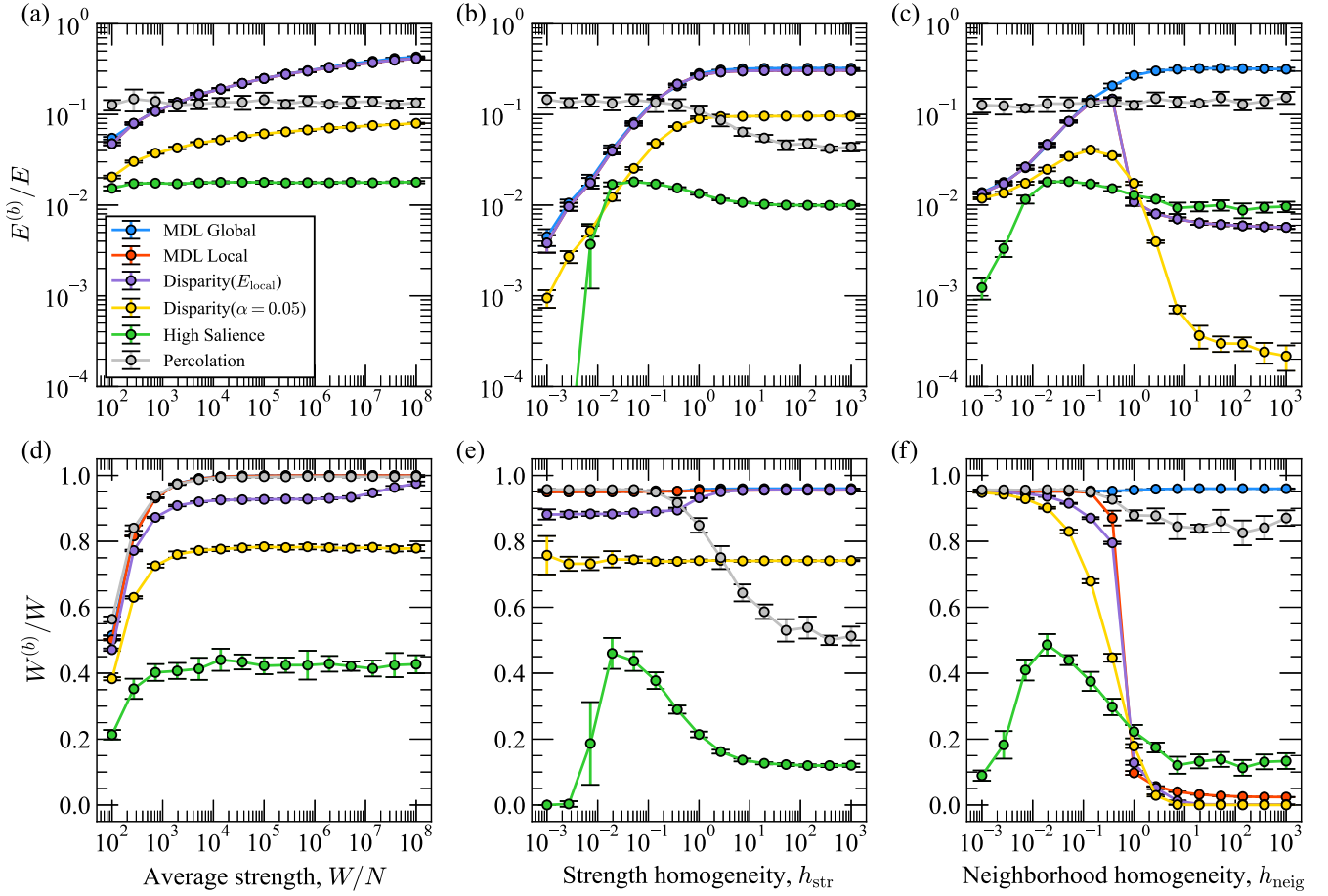


FIG. 4. Concentration of weight along different backbones. Top row: fraction of edges retained in the backbone, $E^{(b)}/E$, versus the (a) average node strength W/N , (b) level of homogeneity h_{str} in the node strength distribution, and (c) level of homogeneity h_{neig} in the weights within each node neighborhood. Bottom row: panels (d)–(f) plot the fraction of weight retained in the backbone, $W^{(b)}/W$, against the same parameters. Markers indicate averages over 10 simulations from the synthetic network model described in Sec. III B, and error bars represent one standard error in the mean.

two columns in Fig. 4), both MDL methods consistently retain a similar number of edges (an average of around 20% of all edges), which is the highest among all methods for $W/N \gtrsim 10^3$. We can also see that the high saliency skeleton consistently retains the fewest edges (an average of around 2% of all edges) in these cases, while the disparity filter with an $\alpha = 0.05$ significance level and the percolation threshold find an intermediate number of backbone edges (an average of around 8% and 12% of all edges, respectively). We also observe that the local MDL method consistently retains a higher total backbone weight $W^{(b)}$ for the same number of edges $E^{(b)}$ as the disparity filter when the latter is set to have the same edge count. The three local backboning methods are highly sensitive to variations in the heterogeneity of weights within the node neighborhoods h_{neig} [Figs. 4(c) and 4(f)], while the global MDL method and the high saliency skeleton are less sensitive to these fluctuations and the percolation method is highly insensitive to the fluctuations. The local backboning

methods all have a sudden drop in the number of edges and total weight they retain at $h_{\text{neig}} \approx 1$, which is the point at which the within-neighborhood variance in the edge weights drops significantly according to this Dirichlet concentration parameter. This high level of weight homogeneity within neighborhoods leaves little statistical evidence to retain any edges in the backbone, since no subset of weights are significantly higher than the rest.

While the size of the backbones is an important factor for efficient downstream graph computations, it is also important that the sparsified networks retain their global connectivity. In the top row of Fig. 5 we plot the fraction of nonisolated nodes in the backbone (i.e., nodes with degree 0) versus the three model parameters. We can observe that all backbones except the high saliency skeleton have nearly zero isolated nodes for a large portion of the parameter space in Figs. 5(a) and 5(b), with higher levels of node isolation for very low levels of strength homogeneity $h_{\text{str}} \lesssim 10^{-2}$ due to sparser backbones across the board. In Fig. 5(c)

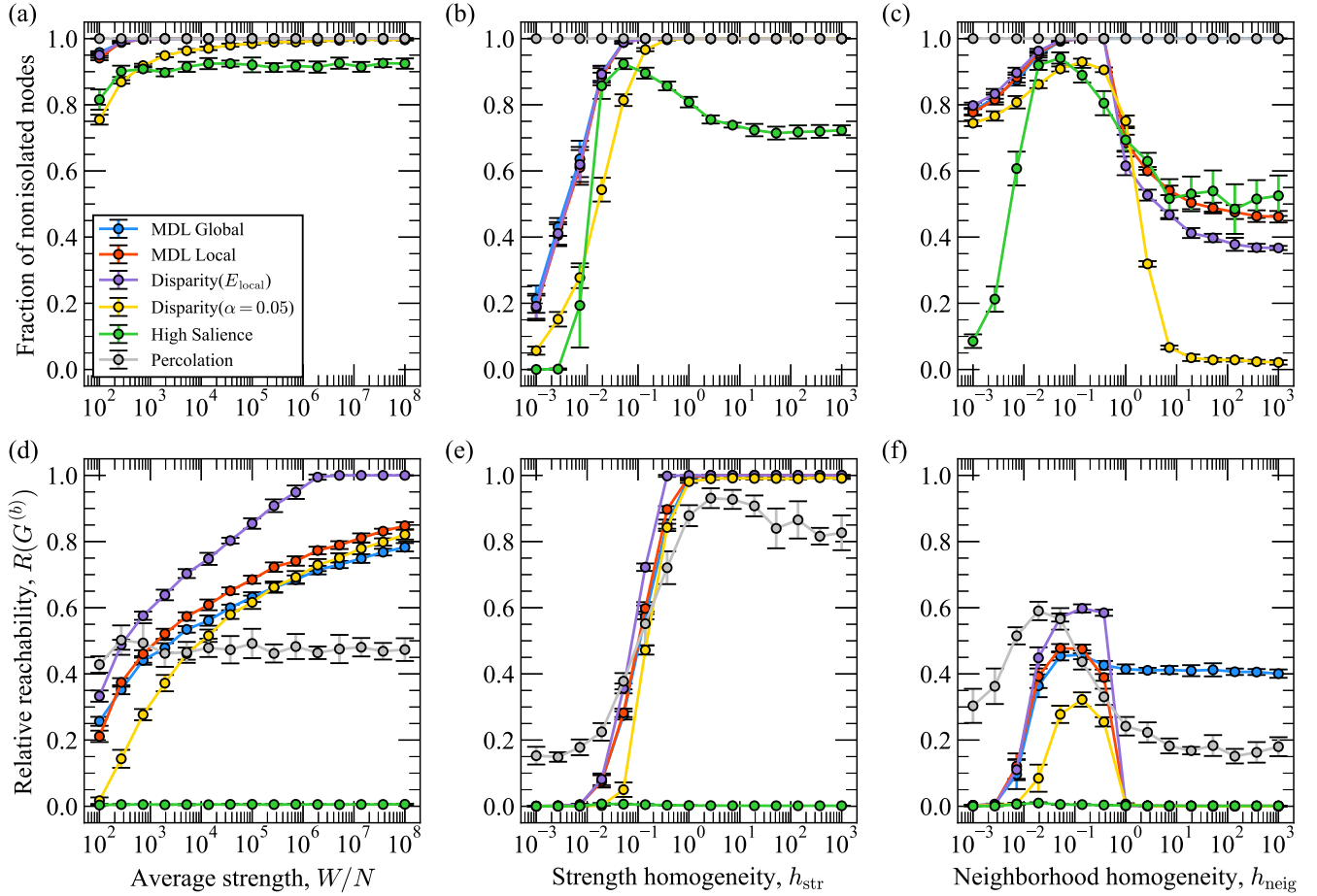


FIG. 5. Global connectivity of different backbones. Top row: fraction of nonisolated nodes retained in the backbone versus the (a) average node strength W/N , (b) level of homogeneity h_{str} in the node strength distribution, and (c) level of homogeneity h_{neig} in the weights within each node neighborhood. Bottom row: panels (d)–(f) plot the relative reachability [Eq. (22)] against the same parameters. Markers indicate averages over 10 simulations from the synthetic network model described in Sec. III B, and error bars represent one standard error in the mean.

we again see strong fluctuations for the local backbone methods—particularly for the disparity filter at $\alpha = 0.05$ —with the MDL method again performing the best out of the local methods and the global MDL method outperforming the high saliency skeleton in terms of node connectivity. The percolation-based method, by definition, has zero isolated nodes for all parameter settings, at the cost of consistently high fractions of edges being retained in this homogeneous weight regime [Fig. 4(c)].

The bottom row of Fig. 5 shows similar connectivity patterns across methods but this time with respect to reachability [Eq. (22)]. Here the disparity filter has a slight edge over the local MDL method in most experiments for the same number of edges and consistently worse reachability for $\alpha = 0.05$. The high saliency skeleton has exactly zero reachability in all experiments, suggesting that although this method highlights important edges for global routing it may tend to construct backbones that are too sparse to maintain global connectivity in the network. The

percolation backbone, despite enforcing a single weakly connected component, does not fully retain directed connectivity as required by $R(G^{(b)})$, scoring lower than other methods in most tests. All methods except the global MDL and percolation methods have poor connectivity in the regime of high neighborhood weight homogeneity, becoming too sparse to have a high reachability value when there is little signal to distinguish the neighborhood backbones.

In Fig. 10 in Appendix C, we plot a few additional metrics against the synthetic model parameters, to understand how compressive each method is and how well each preserves the strength distribution of the original network. We also plot the sizes of the inferred backbones versus the number of nodes in the original network in the same appendix (Fig. 11), finding a very weak relationship with this model parameter. Finally, we show the run-time of some different backbone methods versus the network size in Fig. 12. The slightly superlinear scaling of the MDL run times is consistent with the theoretical scaling of

$O(E \log E)$ discussed in Sec. II C. We also observe a substantially higher computational complexity for the high salience skeleton due to the computation of the shortest path trees.

C. Comparison on real networks

We now compare the six backbone algorithms from Sec. III B on a large corpus of real network datasets coming from different application domains. We initially collected 69 networks from the Netzschleuder repository [49] by retrieving all weighted networks with edge weights $w_{ij} \geq 1$ and less than 10^7 edges. Non-integer-valued weights w_{ij} were rounded to the nearest integer for analyses since the microcanonical models were used. As the methods in Sec. III B are all adaptable to both directed and undirected networks, we analyzed the 32 directed networks in the corpus using both edge directions separately (in edges and out edges) and analyzed the remaining 37 undirected networks with both edge directions simultaneously as described in Sec. II A. The networks in the final corpus represent a diverse array of domains, and (using the classification provided by Ref. [49]) in total we inferred 54 social network backbones, 11 informational network backbones, 9 biological network backbones, 7 technological network backbones, 6 transportation network backbones, and 5 economic network backbones using each of the six methods. The networks ranged in size from $E = 91$ to $E = 5\,743\,258$ with total weight ranging from $W = 282$ to $W = 2\,541\,576\,441$ and average degree ranging from $\langle k \rangle = 1.1$ to $\langle k \rangle = 238.3$.

Our first experiment with this real-world network corpus examines the extent to which the backbones differ between the global and local MDL backbone procedures described in Sec. II B. In Fig. 6(a) we plot the fraction of the original edges retained in the local MDL backbone versus the fraction retained in the global MDL backbone for each of the backbones examined. We only include examples for which both methods found a nonempty backbone, noting the number of violating examples in the axis labels. We can see that the global MDL method tends to find empty backbones in a large fraction of the examples studied (25/92), indicating that this method is frequently not able to compress real network structure any better than a naive edge transmission. On the other hand, only two networks failed to be compressed by the more flexible local MDL method. When both methods give nontrivial backbones, we can see that they return a similar number of edges, with the local MDL method returning more edges in many cases but neither method exceeding $E^{(b)}/E \approx 0.35$. In Fig. 6(b) we plot the inverse compression ratios [Eq. (20)] for each method, which indicates that when the global MDL method does compress it does so quite well, even compressing roughly $\approx 90\%$ of the information in the network relative to a naive edge transmission in some cases. We see that the global method compresses better than the local method in many cases, demonstrating that a global threshold can in fact provide an effective backbone for many real networks. Neither panel indicates any clear differentiation among the backbone structure of networks based on domain.

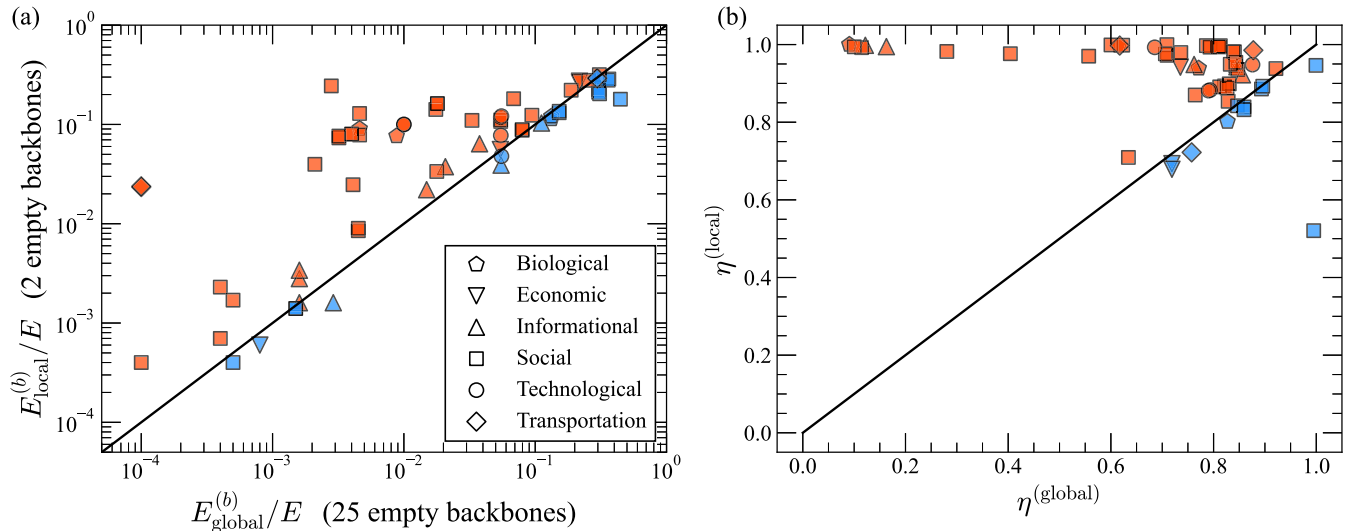


FIG. 6. Global and local backbones across a real-world network corpus. Global and local MDL backbones were inferred using the objectives of Eqs. (13) and (14), respectively, for the set of real networks described in Sec. III C. We plot (a) the fraction of initial edges retained in the local MDL backbone and global MDL backbone and (b) the inverse compression ratio [Eq. (20)] for each backbone examined. We only include examples for which both the global and local MDL methods inferred a nonempty backbone, noting the empty backbones in the axis labels. In both panels we plot the line $y = x$ for reference and color the examples for which the global (local) backbone returned a higher value with blue (red). We also indicate with different markers the domain of each example network.

In the next experiment we compare all six backbone methods of Sec. III B with respect to the same metrics—specifically, the fraction of edges retained in the backbone, the fraction of nodes with degree zero in the backbone, the Hellinger distance [Eq. (23)] between the strength distributions of the backbone and original network, the fraction of weight retained in the backbone $m = W^{(b)}/W$, and the relative reachability [Eq. (22)] between the backbone and original graph. We omit analysis of the inverse compression ratio since trivially the global and local MDL methods will always obtain better compression than their global and local counterparts, and the direct comparison of the two MDL methods is done in Fig. 6. Figures 7(a)–7(e) show the results of these experiments through pairwise comparisons among all pairs of backbone methods for each of the six measures. We can see that in the real networks the methods

are roughly ordered like percolation $>$ MDL local $\approx$ disparity(E_{local}) $>$ MDL global $>$ disparity($\alpha = 0.05$) $>$ high salience when considering the number of edges, the total weight, the fraction of isolated nodes, and the relative reachability of the inferred backbones. Meanwhile, the reverse pattern is observed for the Hellinger distance. MDL local has a slight edge over disparity(E_{local}) with respect to the fraction of isolated nodes and the reachability, while disparity(E_{local}) has a slight edge for the Hellinger distance. The percolation backbone performs the best overall in these tests but at the cost of retaining a substantial fraction of edges (an average of 84%).

In Fig. 7(f) we plot the Jaccard similarity [Eq. (21)] between the backbones generated by each pair of methods, averaged over all networks in the corpus. We observe very low levels of similarity among all pairs of backbones except

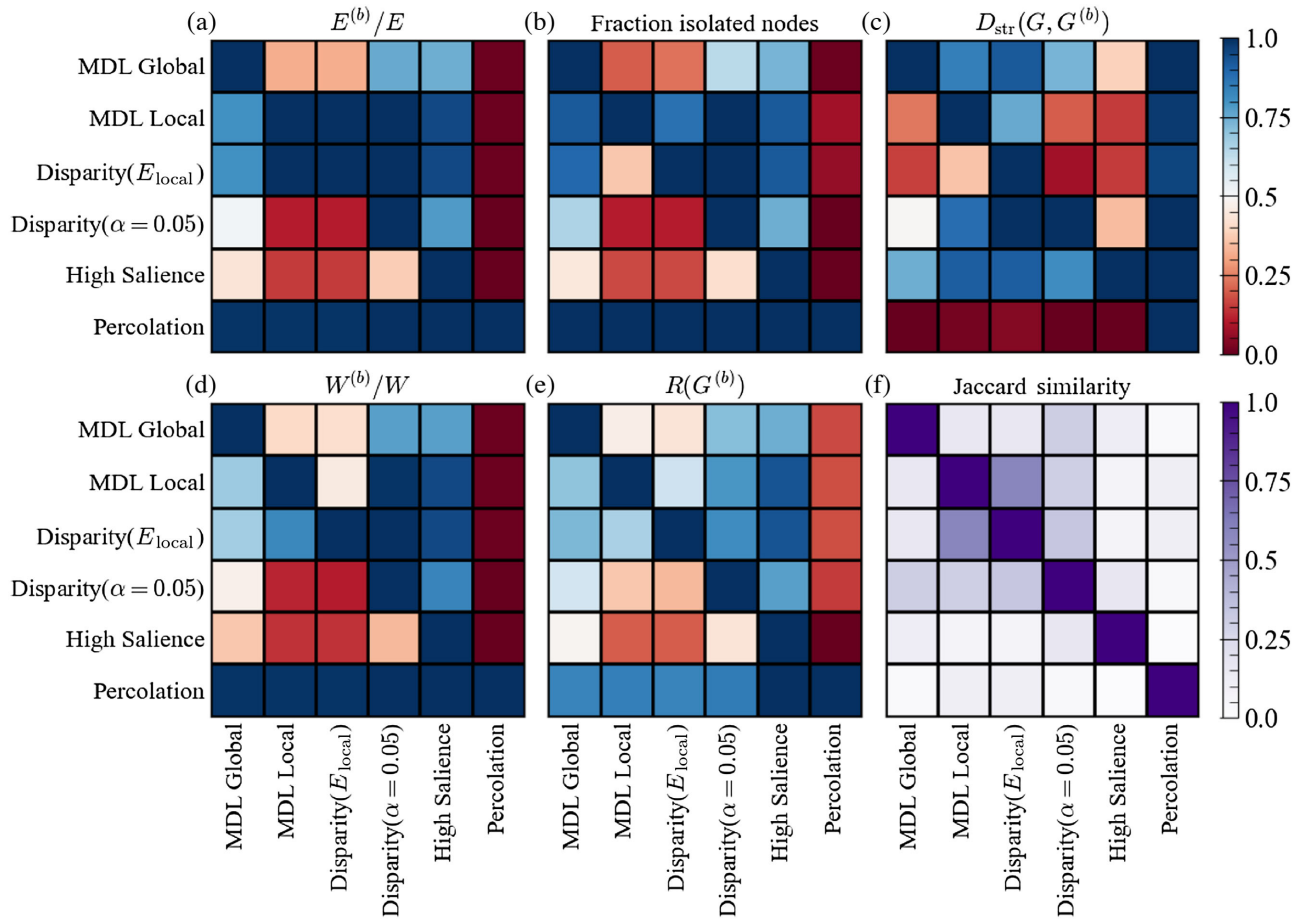


FIG. 7. Comparison of backbone metrics on real networks. The six backbone methods were compared using the same metrics as in Sec. III B on the real networks in the corpus described in Sec. III C. In each of the first five panels we plot a matrix $\mathbf{B}(m)$ for a metric m of interest such that $B_{ij}(m)$ is the fraction of real networks in the corpus for which $m_i \geq m_j$. High values $B_{ij}(m)$ indicate that the metric m was higher for backbone method i than backbone method j in a large portion of real network instances. We plot the matrix $\mathbf{B}(m)$ for (a) the fraction of edges retained in the backbone, $m = E^{(b)}/E$, (b) the fraction of nodes with degree zero in the backbone, (c) the Hellinger distance [Eq. (23)] between the strength distributions of the backbone and original network, (d) the fraction of weight retained in the backbone, $m = W^{(b)}/W$, and (e) the relative reachability [Eq. (22)] between the backbone and original graph. In (f) we plot the average network Jaccard similarity [Eq. (21)] between the backbones generated by each pair of methods, with a different color scale to indicate the different interpretation of the heat map.

for those generated by the MDL local and disparity (E_{local}) methods, which have a moderately high average overlap of roughly 0.5.

Overall, the results in Secs. III B and III C indicate that the local MDL method and the disparity filter set to the same number of backbone edges as the local MDL method tend to best preserve the original network structure in many cases. Meanwhile, the global MDL method often finds empty backbones and does not compress relative to a naive transmission scheme, but in the instances that it does compress it performs well across the metrics while retaining fewer edges and providing even better compression than the local method. Setting the disparity filter to the typical significance level of $\alpha = 0.05$ tends to be too conservative in some real networks, suggesting that this parameter needs to be carefully tuned in practice for different applications. The high salience skeleton is quite conservative in its backbone estimation, as it highlights only a few links that are the most significant for global network routing. And, finally, the percolation backbone tends to preserve connectivity and weight heterogeneity but does not provide substantial sparsification, retaining a majority of the edges in most networks.

D. Comparison of spreading dynamics on network backbones

As an example downstream application of our backbone methods, we examine their performance against baselines for sparsifying networks while retaining dynamical properties relevant for spreading and percolation calculations. Specifically, we adapt a standard method for estimating the epidemic threshold from long-term transmission probabilities among each pair of nodes [50] to the weighted network case, and study how accurate and computationally expensive each backbone is for this calculation on an example network.

In the original model discussed in Ref. [50], there is a single parameter ϕ characterizing the probability that a node i has strong enough contact with any other node j for the disease to get passed from $j \rightarrow i$ if j ends up with the disease. This is analogous to the occupation probability in a bond percolation process. In our weighted network adaptation, we consider the parameter $\phi(w_{ij})$ characterizing this probability to be a monotonic function of the weight w_{ij} of the edge between nodes i and j . (For simplicity, we restrict our analyses to undirected networks.) This mechanism corresponds to i and j having a greater connectivity and potential to transmit the disease to each other when they have a higher weight edge between them. Specifically, we consider a process in which the weight w_{ij} can be interpreted as the frequency of contact between i and j , and that there is a probability p for the disease to spread independently from each contact, so that $\phi(w_{ij}) = 1 - (1 - p)^{w_{ij}}$.

By varying $p \in [0, 1]$, we can find a percolation transition in this model in which an extensive giant component forms and an epidemic can spread to the entire network. We can compute this threshold using a method analogous to that of Ref. [51] in which a constant occupation probability is considered. Letting $u_{i \rightarrow j}$ be the probability that node i is not connected to giant component through node j —in other words, the probability that j does not contribute to the epidemic through passing the disease to node i —we have

$$u_{i \rightarrow j} = 1 - \phi(w_{ij}) + \phi(w_{ij}) \prod_{k \in \mathcal{N}_{j \setminus i}} u_{j \rightarrow k}, \quad (24)$$

where $\mathcal{N}_{j \setminus i}$ is the set of nodes attached to j other than node i . The probability that i is part of the giant cluster (e.g., epidemic cluster) is then given by

$$S_i = 1 - \prod_{j \in \mathcal{N}_i} u_{i \rightarrow j}, \quad (25)$$

where $\mathcal{N}_i$ is the set of nodes connected to i , and the fractional size S of the giant cluster is

$$S = \frac{1}{N} \sum_{i=1}^N S_i = 1 - \frac{1}{N} \sum_{i=1}^N \prod_{j \in \mathcal{N}_i} u_{i \rightarrow j}. \quad (26)$$

Technically, the above calculation is only exact on trees—otherwise, we cannot assume the independence of j 's neighbors to write $\prod_{k \in \mathcal{N}_{j \setminus i}} u_{j \rightarrow k}$. However, it often gives a good approximation in practice for treelike sparse networks. (If one is interested in more complex approximations accounting for dependence among node neighbors [52,53], sparsifying the networks ahead of time through backbone methods may be necessary, for tractability.)

Equations (24) and (25) define a system of message passing equations that can be used to solve for the size of the percolating epidemic cluster of any network G as a function of the per-contact transmission probability p . To do this, the probabilities $u_{i \rightarrow j}$ are initialized to random values, and Eq. (24) is iterated until convergence, at which point S_i can be computed for each node i using Eq. (25).

Figure 8 shows the results of simulating this message passing system on the reality mining network of student contacts collected from mobile phone data in Ref. [54], for p ranging from 10^{-6} to 1. The system was simulated on the full network (black curve) as well as the backbones constructed using the six methods of Sec. III B. To compare these curves, we compute the average absolute error $\langle |S - S_0| \rangle$ between the values S associated with each backbone method and the full network's values S_0 . We can observe that the percolation-based backbone method performs the best with respect to this metric, although we will see that it does not provide a substantial computational speedup since it retains a large fraction of the edges.

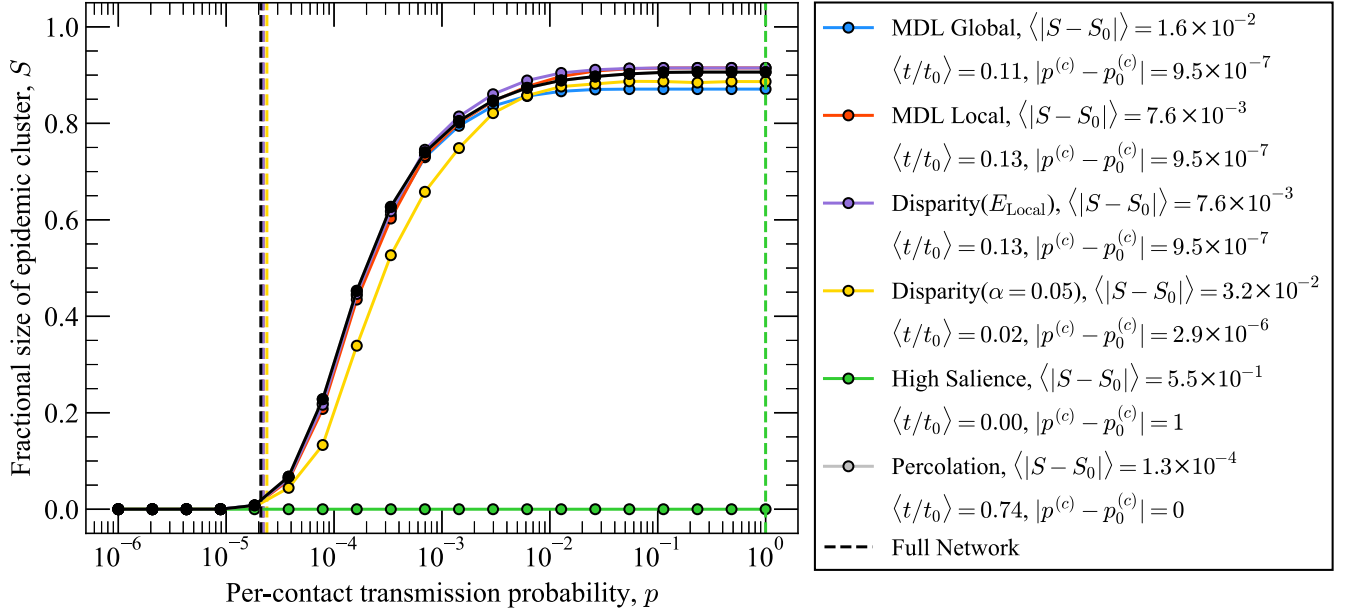


FIG. 8. Speeding up epidemic calculations with network backbones. Fractional size of the epidemic cluster [Eq. (26)] versus the per-contact transmission probability p for the message passing system of Eqs. (24) and (25). The percolation threshold estimated using Eq. (27) for each network backbone and the full network (black) are shown with dotted vertical lines. The average error in S , run-time savings, and error in the percolation threshold (as described in Sec. III D) are shown in the right-hand panel.

The local MDL method and the disparity filter with $E = E_{\text{local}}$ perform similarly under this measure, still providing a good approximation of the full network's giant component size at each p , while the global MDL method performs slightly worse, deviating from the desired curve for high values of p . The disparity filter with $\alpha = 0.05$ performs worse yet, with noticeably large deviations as the giant component grows in size, and the high salience skeleton—with very little retained in the backbone—performs the worst out of the six methods for this task.

We can notice that at a critical probability $p^{(c)} \in [0, 1]$, the system transitions from a fragmented state with many small connected components ($S \approx 0$) to having a single extensive giant connected component ($S \approx 1$). There is a trivial fixed point in the message passing system [Eq. (24)] of $u_{i \rightarrow j} = 1$ —in other words, when no one is in percolating cluster ($S = 0$). We can therefore find the percolation transition by expanding in $u_{i \rightarrow j} = 1 - \epsilon_{i \rightarrow j}$ for small $\epsilon_{i \rightarrow j}$ and identifying at which value of p this fixed point is no longer stable. When the trivial fixed point is stable, we have no giant component, and when it is unstable, we have a giant component. (A similar calculation is performed in Ref. [51] for constant occupation probability ϕ .)

Plugging in $u_{i \rightarrow j} = 1 - \epsilon_{i \rightarrow j}$ to Eq. (24) and expanding for small $\epsilon_{i \rightarrow j}$ gives

$$\epsilon_{i \rightarrow j} = \phi(w_{ij}) \sum_{k \in \partial_{j \setminus i}} \epsilon_{j \rightarrow k} = \sum_{i \rightarrow j, l \rightarrow k} B_{i \rightarrow j, l \rightarrow k}^{(\phi)} \epsilon_{l \rightarrow k}, \quad (27)$$

where $B_{i \rightarrow j, l \rightarrow k}^{(\phi)} = \phi(w_{ij}) \delta_{jl} (1 - \delta_{ik})$ are entries—indexed by edges $i \rightarrow j$ and $l \rightarrow k$ —in a modified version of the

nonbacktracking (or Hashimoto) matrix that account for the transmission probabilities. The system will percolate when the above fixed point is unstable, i.e., when $\mathbf{B}^{(\phi)}$ has a leading eigenvalue of magnitude greater than 1. We can then identify the value $p^{(c)}$ numerically without the need for extensive message passing simulations by evaluating the leading eigenvalue of $\mathbf{B}^{(\phi)}$ at some starting guess p and running a binary search over p until the leading eigenvalue of $\mathbf{B}^{(\phi)}$ has magnitude close to 1. The value at which this terminates will be an estimate of $p^{(c)}$.

In Fig. 8 we can examine how well each backboning method preserves the percolation threshold $p^{(c)}$ of the full network and how much run-time it saves. A good balance of these two factors is ideal in practice, as we desire accuracy while saving computational cost. (For the particular example network studied here, we do not face any computational bottleneck. But this serves as a proof of concept for application to much larger networks.) We measure the run-time savings $\langle t/t_0 \rangle$ as the fractional reduction relative to the run-time t_0 for the full network, averaged over all evaluations of the binary search for $p^{(c)}$. We then measure the error in the percolation threshold as $|p^{(c)} - p_0^{(c)}|$, where $p_0^{(c)}$ is the threshold for the full network.

We can see that, while the percolation-based backbone has the most accurate estimation of $p^{(c)}$, it fails to reduce the run-time of the computation substantially, requiring on average 74% of the run-time of the full network calculation to compute the leading eigenvalues. On the opposite extreme, the leading eigenvalue calculations on the high

saliency skeleton require almost no run-time, but are completely inaccurate and estimate $p^{(c)} = 1$. The disparity filter with $\alpha = 0.05$ has the next highest error in $p^{(c)}$, though it provides run times much lower than other methods (other than the high saliency skeleton). The global MDL, local MDL, and disparity filter set to $E = E_{\text{local}}$ find a good middle ground, allowing for a roughly $10\times$ speedup in the calculation while approximating the percolation transition to within 6 decimal places.

These results together indicate that the MDL backbones are capable of preserving not only structural but also dynamical properties of networks, despite not explicitly involving dynamics in their formulation.

IV. CONCLUSION

In this paper we develop a completely nonparametric framework for inferring the backbone of a weighted network which utilizes the MDL principle to promote sparsity. Our method is adapted to infer both global and local network backbones, and is generalizable to any mixture of weight distributions over the backbone and nonbackbone edges using a flexible Bayesian generative model. We develop fast exact optimization schemes for our global and local MDL backbone objectives that are log-linear in the number of edges, allowing these principled methods to easily scale to networks with millions of edges. We compare our method with existing methods in a range of tasks on synthetic and real network data, showing that the proposed MDL methods are capable of substantially sparsifying a wide variety of networks while retaining meaningful global and local structural characteristics.

There are a number of potential avenues for future work extending our methods. One critical limitation of the proposed method is that it does not apply to unweighted graphs, since the weights are used for compression to identify a meaningful backbone. Extending the MDL framework for backbone unweighted graphs is therefore an important avenue for future work. Additionally, as mentioned in Sec. II A, our method is generalizable for

inferring global and local network backbones under any mixture of weight distributions in the backbone and non-backbone, with a subset of these permitting exact minimization with Algorithms 1 and 2. Here we study a microcanonical model asymptotically equivalent to a canonical model with geometric weights, but in future work it is important to examine other distributions which may provide better compression of networks with different types of weight heterogeneity. One could also fully utilize the posterior distribution over backbones in this Bayesian framework to sparsify networks in a stochastic manner, similarly to some existing sparsification techniques [4]. Both of these extensions may incur a substantial additional computational burden for optimization and sampling compared to the fast greedy optimization performed here, so present new challenges for practical implementation. One could also extend our framework to sparsify hypergraphs or other higher order networks with weight metadata. Finally, it may be possible to extend our method to infer sparse functional network backbones by exploiting regularities in dynamical information such as node or edge states.

ACKNOWLEDGMENTS

The author acknowledges support from the HKU-100 Start Up Fund.

APPENDIX A: ASYMPTOTIC EQUIVALENCE BETWEEN MICROCANONICAL MODEL AND CANONICAL MODEL WITH GEOMETRICALLY DISTRIBUTED WEIGHTS

In this appendix we demonstrate the approximate equivalence between the microcanonical model of Sec. II B and the canonical model of Sec. II A when weights are distributed according to a geometric distribution. Letting $P(w|\theta) = \theta(1-\theta)^{w-1}$ be a geometric distribution and $P(\theta) = 1$ a uniform prior on $[0, 1]$ (which is conjugate to $P(w|\theta)$), we have that Eq. (7) can be written as

$$\begin{aligned}
\mathcal{L}_C^{(\text{global})}(G^{(b)}) &= \log(E+1) + \log\left(\frac{E}{E^{(b)}}\right) - \log \int \prod_{e \in G^{(b)}} \theta_1(1-\theta_1)^{w_e-1} d\theta_1 - \log \int \prod_{e \in G \setminus G^{(b)}} \theta_0(1-\theta_0)^{w_e-1} d\theta_0 \\
&= \log(E+1) + \log\left(\frac{E}{E^{(b)}}\right) - \log \int \theta_1^{E^{(b)}} (1-\theta_1)^{W^{(b)}-E^{(b)}} d\theta_1 - \log \int \theta_0^{E-E^{(b)}} (1-\theta_0)^{(W-W^{(b)})-(E-E^{(b)})} d\theta_0 \\
&= \log(E+1) + \log\left(\frac{E}{E^{(b)}}\right) - \log B(E^{(b)}+1, \\
&\quad W^{(b)}-E^{(b)}+1) - \log B[E-E^{(b)}+1, (W-W^{(b)})-(E-E^{(b)})+1] \\
&= \log(E+1) + \log(W^{(b)}+1) + \log(W-W^{(b)}+1) + \log\left(\frac{E}{E^{(b)}}\right) + \log\left(\frac{W^{(b)}}{E^{(b)}}\right) + \log\left(\frac{W-W^{(b)}}{E-E^{(b)}}\right) \\
&= \mathcal{L}_M^{(\text{global})}(G^{(b)}) + \delta,
\end{aligned} \tag{A1}$$

where

$$\delta = \log \frac{(W^{(b)} + 1)(W - W^{(b)} + 1)(W^{(b)})(W - W^{(b)})}{(W - E + 1)(E^{(b)})(E - E^{(b)})} \leq \log \frac{(W/2 + 1)^2(W/2)^2}{(W - E + 1)(E - 1)}, \quad (\text{A2})$$

for $E^{(b)} \in [1, E/2]$. For $E^{(b)} = 0$, we evaluate only the second integral since there are no edges in the backbone $G^{(b)}$, and the resulting expressions differ by $\delta = \{[(W + 1)W]/(W - E + 1)E\}$. Regardless, the correction scales like $\delta \sim O(\log W + \log E)$, and so we have

$$\mathcal{L}_C^{(\text{global})}(G^{(b)}) = \mathcal{L}_M^{(\text{global})}(G^{(b)}) + O(\log W + \log E), \quad (\text{A3})$$

and the canonical and microcanonical expressions are asymptotically equal for $W, E \gg 1$.

Similarly, the microcanonical neighborhood-level description length of Eq. (11) is equivalent to the neighborhood-level description length of Eq. (15) up to a correction of $O(\log s_i + \log k_i)$, making these objectives asymptotically equivalent for $s_i, k_i \gg 1$. However, the full canonical local description length of Eq. (10) is only equivalent to the corresponding microcanonical expression in Eq. (14) up to $-\log P(s) = \log \binom{N+W-E-1}{W-E}$ in addition to a correction of $O(N(\log k + \log s))$ which is of a similar size. Despite this discrepancy in the network-level description lengths for the local backboning methods, since the backboning is done for each neighborhood ∂_i separately the microcanonical model is easily capable of inferring planted backbone

structure generated from the canonical model (as seen in Fig. 2).

APPENDIX B: GREEDY OPTIMALITY IN NUMERICAL EXPERIMENTS

Here we plot the results of reconstruction experiments using the same generative model as in Sec. III A, except with small enough networks ($N = 6$) to allow for exhaustive enumeration over all backbones $G^{(b)} \subseteq G$ to verify the optimality of Algorithms 1 and 2. Figure 9 shows the compression ratio [Eq. (20)], obtained using both the greedy method (x axis) and exact enumeration over possible backbones (y axis), for all experimental trials shown in Fig. 2. We find that the greedy algorithm

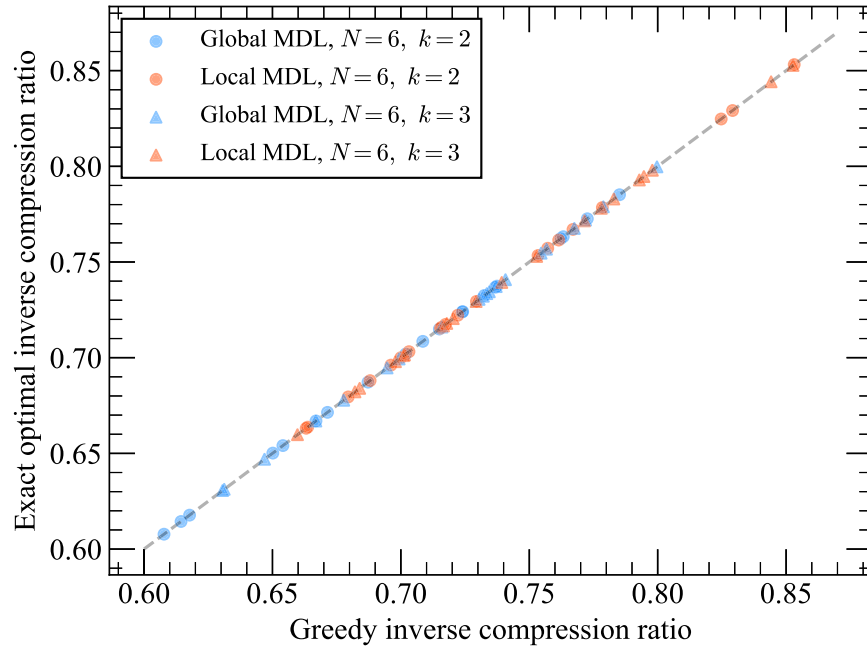


FIG. 9. Optimality of greedy algorithm. Inverse compression ratios for all reconstruction experiments in Fig. 2, performed on networks with $N = 6$ and $k \in \{2, 3\}$. The results obtained from the greedy methods of Algorithms 1 and 2 are shown on the x axis, while the results obtained through exhaustive enumeration over possible backbones are shown on the y axis. We find that the greedy algorithms give identical results to exact enumeration, as expected, for all cases studied.

identifies an identical description length to exact enumeration in all cases up to machine precision.

APPENDIX C: ADDITIONAL COMPARISONS ON SYNTHETIC NETWORKS

In this appendix we provide additional tests to compare the different bonboning methods with the synthetic network model of Sec. III B.

In the top row of Fig. 10, we can see that (as expected) the global MDL method is the most compressive among the global methods and the local MDL method is the most compressive among the local methods. Although the local methods tend to be more compressive in general, for high strength homogeneity [Fig. 10(b)] we find a slight compressibility advantage for the global method, suggesting that for networks with simple weight distributions one may favor the global MDL method over the local MDL method due to the additional model complexity of the latter. All methods exhibit worse compression as the weights become

more homogeneous, since there is little statistical evidence for a separate backbone of high weight edges. In the bottom row we plot the Hellinger distance between the node strength distributions in the full network and the backbone [Eq. (23)], which shows similar performance for the global MDL method, local MDL method, and disparity filter with E_{local} , while the disparity filter at $\alpha = 0.05$ has slightly worse performance (e.g., a higher discrepancy in the strength distributions). In Fig. 10(f), however, we see that the MDL methods are much more robust to neighborhood weight heterogeneity fluctuations than the other methods, maintaining relatively similar strength distributions in the inferred backbones as the original networks.

Figure 11 shows the sizes of the inferred backbones versus the number of nodes in the original network, suggesting a very weak relationship with this model parameter, and Fig. 12 plots the run times of a few methods for comparison, the slightly superlinear scaling of the MDL run times consistent with the $O(E \log E)$ scaling discussed in Sec. II C.

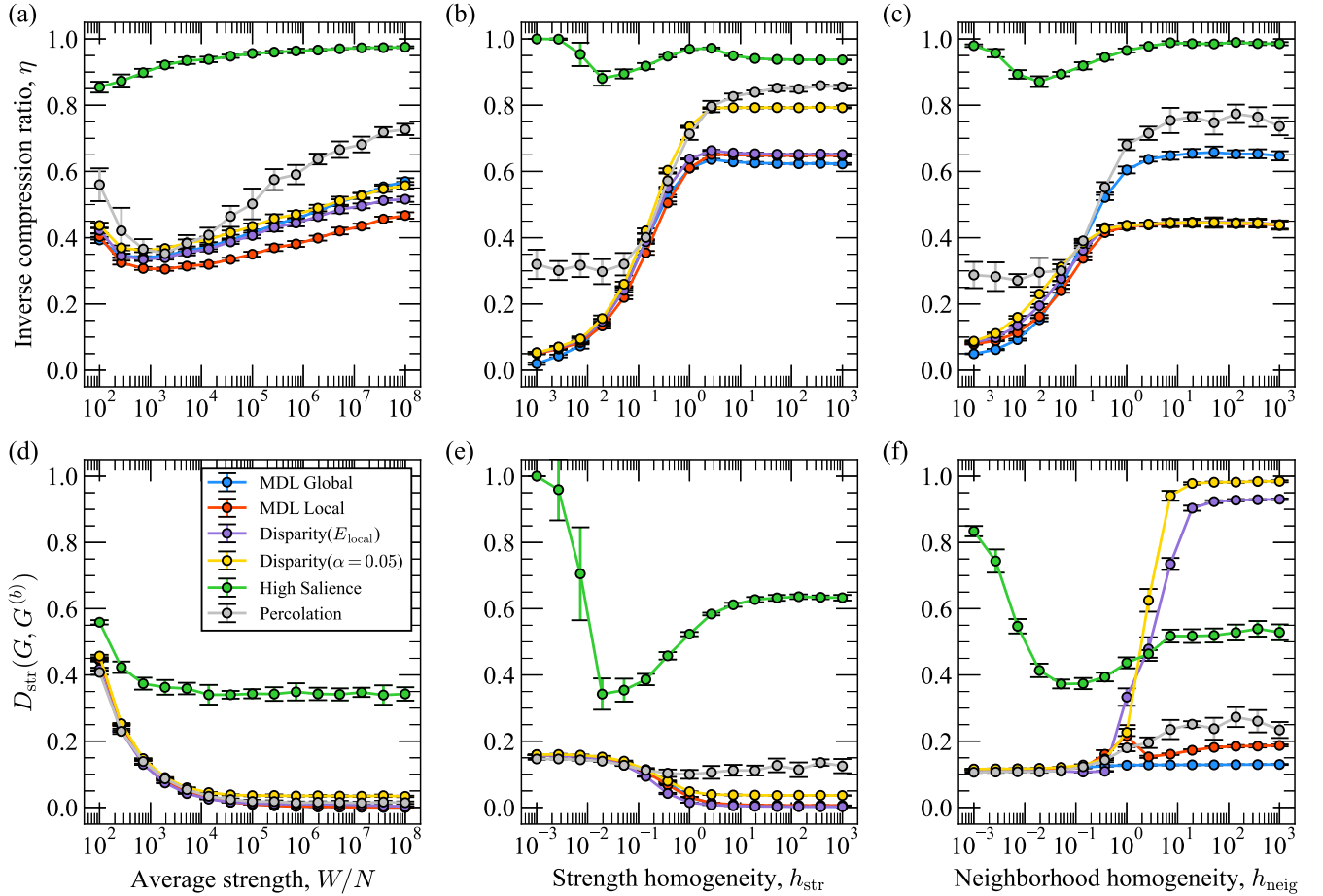


FIG. 10. Compression and node strength discrepancies among different backbones. Top row: inverse compression ratio [Eq. (20)] versus the (a) average node strength W/N , (b) level of homogeneity h_{str} in the node strength distribution, and (c) level of homogeneity h_{neig} in the weights within each node neighborhood. Bottom row: panels (d)–(f) plot the Hellinger distance [Eq. (23)] between the original strength distribution and the backbone strength distribution against the same parameters. Markers indicate averages over 10 simulations from the synthetic network model described in Sec. III B, and error bars represent one standard error in the mean.

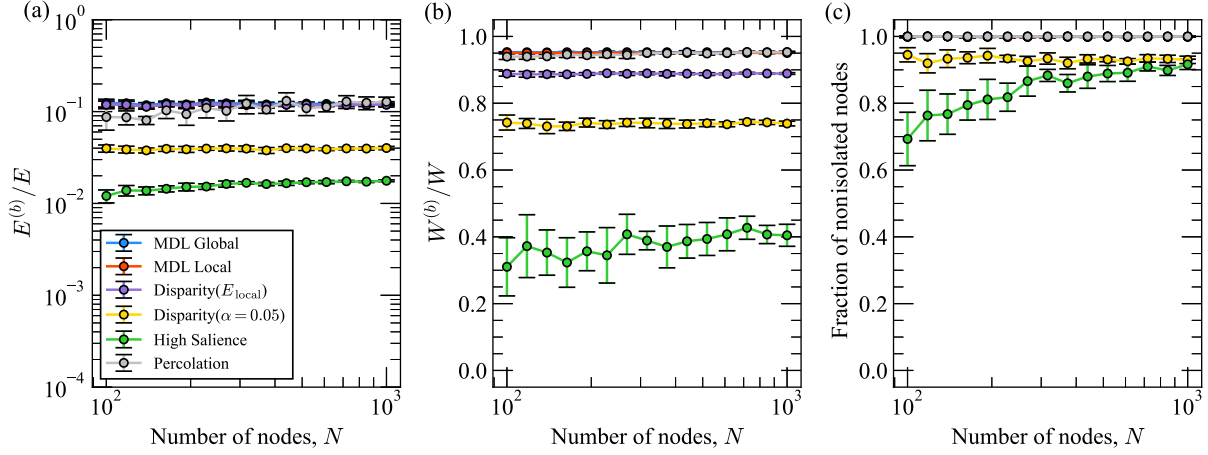


FIG. 11. Sizes of backbones for different network sizes. (a) Fraction of edges retained in the backbone, (b) fraction of total weight retained in the backbone, and (c) fraction of nonisolated nodes in the backbone versus the number of nodes in the synthetic networks generated with the model in Sec. III B. Markers indicate averages over 10 simulations from the synthetic network model described in Sec. III B, and error bars represent one standard error in the mean.

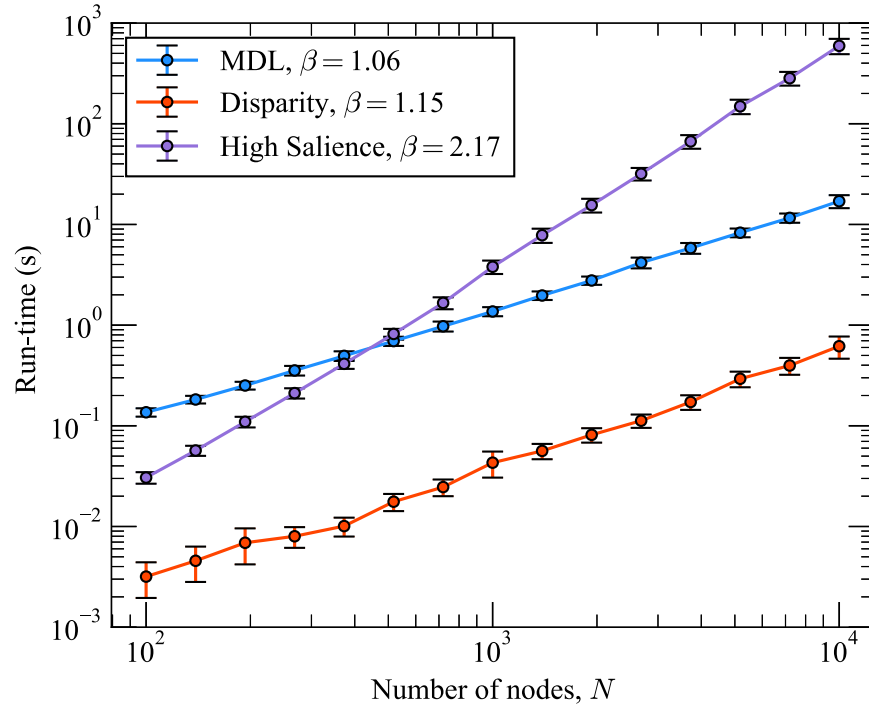


FIG. 12. Run-time scaling of different backboning methods. Run-time per backbone evaluation (in seconds) versus the number of nodes N for the synthetic network experiments in Sec. III B. The global and local MDL backbones are computed together, as well as the disparity filter backbones. Best-fit slope values β for linear fits of the form $\log(\text{run-time}) = \beta \log(N) + C$ are shown alongside each method in the legend. Markers indicate averages over 10 simulations from the synthetic network model described in Sec. III B, and error bars represent one standard error in the mean.

APPENDIX D: PROOF OF GREEDY ALGORITHM OPTIMALITY FOR CANONICAL BACKBONE MODELS

Here we prove the optimality of the greedy algorithm in Algorithm 1 (and consequently Algorithm 2, when applied

to individual neighborhoods) for a set of weight distributions in what is called the natural exponential family (NEF) [46], parametrized by

$$P(w|\theta) = h(w)e^{\theta w - A(\theta)}, \quad (\text{D1})$$

where $h(w)$ is some function of only the weight and $A(\theta) = \log \int h(w)e^{\theta w}$ is the log normalizing constant (or log partition function). Because of the linearity in the exponent, in NEF distributions the edge weights in the backbone and nonbackbone interact only with the model

parameters θ through their sum $W^{(b)}$ in the model likelihood, allowing us to analyze the effect of changing $W^{(b)}$ with fixed $E^{(b)}$.

Using Eq. (7), we have, up to constants independent of $W^{(b)}$,

$$\mathcal{L}_C^{(\text{global})} \propto -\log \left[\int P(\theta_1) \prod_{e \in G^{(b)}} P(w_e | \theta_1) d\theta_1 \int P(\theta_0) \prod_{e \in G \setminus G^{(b)}} P(w_e | \theta_0) d\theta_0 \right]. \quad (\text{D2})$$

Using the arguments from Sec. II C, if we have

$$\Delta \mathcal{L}_C^{(\text{global})}(W^{(b)} + 1) < 0 \quad (\text{integer weights}), \quad (\text{D3})$$

$$\frac{\partial \mathcal{L}_C^{(\text{global})}}{\partial W^{(b)}} < 0 \quad (\text{continuous weights}), \quad (\text{D4})$$

given the greedy conditions,

$$E^{(b)} \leq \tilde{E}^{(b)}, \quad (\text{D5})$$

$$\frac{W^{(b)}}{E^{(b)}} \geq \frac{W}{E} \geq \frac{\tilde{W}^{(b)}}{\tilde{E}^{(b)}}, \quad (\text{D6})$$

then the greedy algorithms of Algorithms 1 and 2 are optimal. We have used the notation $\tilde{W}^{(b)} = W - W^{(b)}$ and $\tilde{E}^{(b)} = E - E^{(b)}$ for brevity.

For the Poisson distribution with a maximum entropy exponential prior with scale parameter λ , we have that

$$\mathcal{L}_C^{(\text{global})} \propto \log \frac{(E^{(b)} + \lambda)^{W^{(b)}+1} (\tilde{E}^{(b)} + \lambda)^{\tilde{W}^{(b)}+1}}{W^{(b)}! \tilde{W}^{(b)}!}. \quad (\text{D7})$$

Using the same argument as in Sec. II C, we have

$$\Delta \mathcal{L}_C^{(\text{global})}(W^{(b)} + 1) = \log \frac{(E^{(b)} + \lambda)}{(W^{(b)} + 1)} \frac{(\tilde{W}^{(b)} + 1)}{(\tilde{E}^{(b)} + \lambda)}. \quad (\text{D8})$$

This is negative for $[(W^{(b)} + 1)/(E^{(b)} + \lambda)] > [(\tilde{W}^{(b)} + 1)/(\tilde{E}^{(b)} + \lambda)]$, which is asymptotically equivalent to the greedy condition $(W^{(b)}/E^{(b)}) > (\tilde{W}^{(b)}/\tilde{E}^{(b)})$.

For the geometric distribution, we showed in Appendix A that the description length under a maximum entropy uniform prior is given by

$$\mathcal{L}_C^{(\text{global})} = \mathcal{L}_M^{(\text{global})}(G^{(b)}) + \log \frac{(W^{(b)} + 1)(\tilde{W}^{(b)} + 1)(W^{(b)})(\tilde{W}^{(b)})}{(W - E + 1)(E^{(b)})(E - E^{(b)})}, \quad (\text{D9})$$

which gives

$$\Delta \mathcal{L}_C^{(\text{global})}(W^{(b)} + 1) = \log \frac{\tilde{W}^{(b)} - \tilde{E}^{(b)}}{W^{(b)} - E^{(b)} + 1} \frac{W^{(b)} + 2}{\tilde{W}^{(b)} + 1}, \quad (\text{D10})$$

where $\tilde{E}^{(b)} = E - E^{(b)}$ and $\tilde{W}^{(b)} = W - W^{(b)}$. This is negative for $[(W^{(b)} + 2)/(E^{(b)} + 1)] > [(\tilde{W}^{(b)} + 1)/(\tilde{E}^{(b)} + 1)]$, which is again asymptotically equivalent to the greedy condition $(W^{(b)}/E^{(b)}) > (\tilde{W}^{(b)}/\tilde{E}^{(b)})$.

For the exponential distribution with a maximum entropy exponential prior with scale parameter λ , we have that

$$\mathcal{L}_C^{(\text{global})} \propto \log \frac{(W^{(b)} + \lambda)^{E^{(b)}+1} (\tilde{W}^{(b)} + \lambda)^{\tilde{E}^{(b)}+1}}{E^{(b)}! \tilde{E}^{(b)}!}, \quad (\text{D11})$$

which gives

$$\frac{\partial \mathcal{L}_C^{(\text{global})}}{\partial W^{(b)}} = \frac{E^{(b)} + 1}{W^{(b)} + \lambda} - \frac{\tilde{E}^{(b)} + 1}{\tilde{W}^{(b)} + \lambda}. \quad (\text{D12})$$

This is negative for $[(W^{(b)} + \lambda)/(E^{(b)} + 1)] > [(\tilde{W}^{(b)} + \lambda)/(\tilde{E}^{(b)} + 1)]$, which is asymptotically equivalent to the greedy condition $(W^{(b)}/E^{(b)}) > (\tilde{W}^{(b)}/\tilde{E}^{(b)})$.

We thus have that Algorithms 1 and 2 are asymptotically optimal for the canonical backboning model with Poisson, geometric, and exponential distributions $P(w|\theta)$, archetypal examples of NEF distributions.

-
- [1] A. Mercier, S. Scarpino, and C. Moore, *Effective resistance against pandemics: Mobility network sparsification for high-fidelity epidemic simulations*, *PLoS Comput. Biol.* **18**, e1010650 (2022).
 - [2] M. Mathioudakis, F. Bonchi, C. Castillo, A. Gionis, and A. Ukkonen, *Sparsification of influence networks*, in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (ACM, New York City, 2011), p. 529.
 - [3] M. Imre, J. Tao, Y. Wang, Z. Zhao, Z. Feng, and C. Wang, *Spectrum-preserving sparsification for visualization of big graphs*, *Comput. Graph.* **87**, 89 (2020).

- [4] D. A. Spielman and N. Srivastava, *Graph sparsification by effective resistances*, in *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing* (ACM, New York City, 2008), pp. 563–568.
- [5] M. Á. Serrano, M. Boguná, and A. Vespignani, *Extracting the multiscale backbone of complex weighted networks*, *Proc. Natl. Acad. Sci. U.S.A.* **106**, 6483 (2009).
- [6] D. Grady, C. Thiemann, and D. Brockmann, *Robust classification of salient links in complex networks*, *Nat. Commun.* **3**, 864 (2012).
- [7] Z. P. Neal, *Backbone: An R package to extract network backbones*, *PLoS One* **17**, e0269137 (2022).
- [8] A. Yassin, A. Haidar, H. Cherifi, H. Seba, and O. Togni, *An evaluation tool for backbone extraction techniques in weighted complex networks*, *Sci. Rep.* **13**, 17000 (2023).
- [9] D. A. Spielman and S.-H. Teng, *Spectral sparsification of graphs*, *SIAM J. Comput.* **40**, 981 (2011).
- [10] Z. Su, Y. Liu, J. Kurths, and H. Meyerhenke, *Generic network sparsification via degree- and subgraph-based edge sampling*, *Inf. Sci.* **679**, 121096 (2024).
- [11] E. John and I. Safro, *Single- and multi-level network sparsification by algebraic distance*, *J. Complex Netw.* **5**, 352 (2017).
- [12] Z. Wu, L. A. Braunstein, S. Havlin, and H. E. Stanley, *Transport in weighted networks: Partition into superhighways and roads*, *Phys. Rev. Lett.* **96**, 148702 (2006).
- [13] G. Li, L. A. Braunstein, S. V. Buldyrev, S. Havlin, and H. E. Stanley, *Transport and percolation theory in weighted networks*, *Phys. Rev. E* **75**, 045103(R) (2007).
- [14] A. Haimovici, E. Tagliazucchi, P. Balenzuela, and D. R. Chialvo, *Brain organization into resting state networks emerges at criticality on a model of the human connectome*, *Phys. Rev. Lett.* **110**, 178101 (2013).
- [15] R. J. Zhang, H. E. Stanley, and F. Y. Ye, *Extracting h-backbone as a core structure in weighted networks*, *Sci. Rep.* **8**, 14356 (2018).
- [16] T. Simas, R. B. Correia, and L. M. Rocha, *The distance backbone of complex networks*, *J. Complex Netw.* **9**, cnab021 (2021).
- [17] S. Rajeh, M. Savonnet, E. Leclercq, and H. Cherifi, *Modularity-based backbone extraction in weighted complex networks*, in *Proceedings of the International Conference on Network Science* (Springer, New York, 2022), pp. 67–79.
- [18] R. Marcaccioli and G. Livan, *A Pólya urn approach to information filtering in complex networks*, *Nat. Commun.* **10**, 745 (2019).
- [19] M. Tumminello, S. Micciche, F. Lillo, J. Piilo, and R. N. Mantegna, *Statistically validated networks in bipartite complex systems*, *PLoS One* **6**, e17994 (2011).
- [20] N. Dianati, *Unwinding the hairball graph: Pruning algorithms for weighted complex networks*, *Phys. Rev. E* **93**, 012304 (2016).
- [21] G. Casiraghi, V. Nanumyan, I. Scholtes, and F. Schweitzer, *From relational data to graphs: Inferring significant links using generalized hypergeometric ensembles*, in *Proceedings of the International Conference on Social Informatics* (Springer, New York, 2017), pp. 111–120.
- [22] N. J. Foti, J. M. Hughes, and D. N. Rockmore, *Nonparametric sparsification of complex multiscale networks*, *PLoS One* **6**, e16431 (2011).
- [23] M. E. Newman, *Network structure from rich but noisy data*, *Nat. Phys.* **14**, 542 (2018).
- [24] L. Peel, T. P. Peixoto, and M. De Domenico, *Statistical inference links data and theory in network science*, *Nat. Commun.* **13**, 6794 (2022).
- [25] T. P. Peixoto, *Network reconstruction via the minimum description length principle*, *Phys. Rev. X* **15**, 011065 (2025).
- [26] G. T. Cantwell, Y. Liu, B. F. Maier, A. C. Schwarze, C. A. Serván, J. Snyder, and G. St-Onge, *Thresholding normally distributed data creates complex networks*, *Phys. Rev. E* **101**, 062302 (2020).
- [27] T. P. Peixoto, *Scalable network reconstruction in subquadratic time*, *arXiv:2401.01404*.
- [28] A. Braunstein, L. Dall'Asta, G. Semerjian, and L. Zdeborová, *Network dismantling*, *Proc. Natl. Acad. Sci. U.S.A.* **113**, 12368 (2016).
- [29] L. Zdeborová, P. Zhang, and H.-J. Zhou, *Fast and simple decycling and dismantling of networks*, *Sci. Rep.* **6**, 37954 (2016).
- [30] J. Rissanen, *Modeling by shortest data description*, *Automatica* **14**, 465 (1978).
- [31] T. P. Peixoto, *Bayesian stochastic blockmodeling*, *Advances in Network Clustering and Blockmodeling* (John Wiley & Sons, Hoboken, NJ, 2019), pp. 289–332.
- [32] A. Kirkley, *Spatial regionalization based on optimal information compression*, *Commun. Phys.* **5**, 249 (2022).
- [33] T. P. Peixoto and A. Kirkley, *Implicit models, latent compression, intrinsic biases, and cheap lunches in community detection*, *Phys. Rev. E* **108**, 024309 (2023).
- [34] C. Coupette and J. Vreeken, *Graph similarity description: How are these graphs similar?*, in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (ACM, New York City, 2021), pp. 185–195.
- [35] C. Coupette, S. Dalleiger, and J. Vreeken, *Differentially describing groups of graphs*, in *Proceedings of the AAAI Conference on Artificial Intelligence* (AAAI, Washington, DC, 2022), Vol. 36, pp. 3959–3967.
- [36] Y. Liu, T. Safavi, A. Dighe, and D. Koutra, *Graph summarization methods and applications: A survey*, *ACM Comput. Surv.* **51**, 1 (2018).
- [37] A. Kirkley, A. Rojas, M. Rosvall, and J.-G. Young, *Compressing network populations with modal networks reveal structural diversity*, *Commun. Phys.* **6**, 148 (2023).
- [38] N. Shah, D. Koutra, T. Zou, B. Gallagher, and C. Faloutsos, *Timecrunch: Interpretable dynamic graph summarization*, in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (ACM, New York City, 2015), pp. 1055–1064.
- [39] J.-G. Young, G. Petri, and T. P. Peixoto, *Hypergraph reconstruction from network data*, *Commun. Phys.* **4**, 135 (2021).
- [40] A. Kirkley, *Identifying hubs in directed networks*, *Phys. Rev. E* **109**, 034310 (2024).
- [41] A. Kirkley, *Inference of dynamic hypergraph representations in temporal interaction data*, *Phys. Rev. E* **109**, 054306 (2024).
- [42] D. J. MacKay, *Information Theory, Inference and Learning Algorithms* (Cambridge University Press, Cambridge, England, 2003).

- [43] H. Felipe, F. Battiston, and A. Kirkley, *Network mutual information measures for graph similarity*, *Commun. Phys.* **7**, 335 (2024).
- [44] I. Voitalov, P. Van der Hoorn, M. Kitsak, F. Papadopoulos, and D. Krioukov, *Weighted hypersoft configuration model*, *Phys. Rev. Res.* **2**, 043157 (2020).
- [45] T. P. Peixoto, *Nonparametric Bayesian inference of the microcanonical stochastic block model*, *Phys. Rev. E* **95**, 012317 (2017).
- [46] C. N. Morris, *Natural exponential families with quadratic variance functions*, *Ann. Stat.* **10**, 65 (1982).
- [47] A. Kirkley and B. He, *Paninipy: Package of algorithms for nonparametric inference with networks in Python*, *J. Open Source Software* **9**, 7312 (2024).
- [48] M. Génois, C. L. Vestergaard, J. Fournet, A. Panisson, I. Bonmarin, and A. Barrat, *Data on face-to-face contacts in an office building suggest a low-cost vaccination strategy based on community linkers*, *Network Sci.* **3**, 326 (2015).
- [49] T. P. Peixoto, *The Netzscheuler network catalogue and repository*, 2020, <https://networks.skewed.de>.
- [50] M. Newman, *Networks* (Oxford University Press, New York, 2018).
- [51] B. Karrer, M. E. J. Newman, and L. Zdeborová, *Percolation on sparse networks*, *Phys. Rev. Lett.* **113**, 208702 (2014).
- [52] J. S. Yedidia, W. T. Freeman, Y. Weiss *et al.*, *Understanding belief propagation and its generalizations*, *Explor. Artif. Intell. New Millennium* **8**, 0018 (2003).
- [53] A. Kirkley, G. T. Cantwell, and M. Newman, *Belief propagation for networks with loops*, *Sci. Adv.* **7**, eabf1211 (2021).
- [54] N. Eagle and A. Pentland, *Reality mining: Sensing complex social systems*, *Pers. Ubiquitous Comput.* **10**, 255 (2006).