QUANTIFYING HUMAN PRIORS OVER ABSTRACT RELATIONAL STRUCTURES

Gecia Bravo-Hermsdorff

A DISSERTATION

PRESENTED TO THE FACULTY

OF PRINCETON UNIVERSITY

IN CANDIDACY FOR THE DEGREE

OF DOCTOR OF PHILOSOPHY

RECOMMENDED FOR ACCEPTANCE

BY THE PRINCETON NEUROSCIENCE INSTITUTE

ADVISOR: YAEL NIV

JANUARY 2020

Abstract

Some new tasks are trivial to learn, while others are essentially impossible; what determines how easy it is to learn the structure of a given task? Similar to how our priors about visual scenes demonstrably color our perception of the world, our priors about the structure of tasks shape our learning, decision-making, and generalization abilities. Drawing inspiration from the insights afforded to neuroscience by the characterization of visual priors, in this dissertation, we strive to quantify priors over abstract structures. In particular, we focus on graphs: the structure of interactions.

In Chapter 3, we describe the natural analogue of cumulants (e.g., mean, (co)variance, skew, kurtosis) for graphs, building a hierarchical description based on local correlations between an increasing number of connections. This provides a principled framework for quantifying the propensity of a network to display arbitrary substructures, and allows one to meaningfully compare networks of different sizes and edge densities.

In Chapter 4, we analyze graph structure globally, via the dynamics of diffusion, providing an algorithm that reduces a graph while preserving its large-scale structure. Our framework analytically unifies two areas of research, namely graph sparsification (removing edges) and graph coarsening (merging nodes), and is competitive with current state-of-the-art algorithms.

From a neuroscience perspective, we develop a novel method for quantifying human priors over graphs (Chapters 2 and 3). In Chapter 5, we apply this method to graphical representations of social and navigation tasks: two domains that have been relevant (over evolutionary timescales) to our everyday life. We find that the resulting priors exhibit non-trivial graphical structure. While some features appear to be general, such as the preferred amount of sparsity as a function of graph size, other features appear to be domain-specific, such as the preference for triadic closure in social interactions.

Through the development of principled methods for analyzing network structure and the use of an analytically tractable model of human learning on graphs, this work provides useful tools that could cross-pollinate with other active areas of research, such as artificial intelligence.

Acknowledgements

We are like dwarfs on the shoulders of giants,

so that we can see more than they, and things at a greater distance,

not by virtue of any sharpness on sight on our part, or any physical distinction,

but because we are carried high and raised up by their giant size.

—— Bernard De Chartres, *circa 1130*

I am truly grateful for how much I have learned during my time in graduate school. Academics aside, Princeton has also gifted me with truly amazing friends and collaborators.

During my first year, I met Lee M. Gunderson (my strange loop and now husband). Since then, we have been stuck together like glue(ons, QCD). Personally, he is my joyful origin to return to when the stochastic walk of life takes me too far. Academically, without him, I would not be the scientist I am today. Not only do two chapters in this thesis consist of our academic projects together, but his intelligence, creativity and ability to teach math and science now permeate the way I think. I am beyond grateful for his deep love, companionship, adventurous personality, and that he has decided to share all of this with me (also that he got as "nerd-sniped" by graphs as I did!).

The neuroscience department has a great community, from which I have benefited both academically and personally. I would like to thank our (unfairly awesome) first-year neuroscience cohort — Matt Panichello, Alex Piet, Diana Liao, Alex Song, Kim Stachenfeld, Luis Piloto, and Sam Ritter — you all are amazing people; it was a genuine pleasure to progress in both academia and life with you. I also would like to thank some mentors in particular. Carlos Brody, for his incredible support, particularly at the beginning and the end of this journey: without the delicious cheese surprises, I wouldn't have made it through the (mandatory for international students) english class, and without his guidance in the final phases, I wouldn't have managed to defend now. Matt Botvinick and the members of

the his lab, who provided me with an awesome time during my first year. Conversations with Matt were always great; he is incredibly knowledgeable and discussions with him are always inspirational. Tom Griffths, for allowing me to freely participate in his lab meetings, for being on my thesis committee, and for always having his doors open. Also, a much deserved thank you to the members of Niv lab for stimulating conversations and company. In particular, Stephanie Chan, for valuable guidance on how to navigate graduate school; Nina Rouhani, for boosting my mental and physical health (hot pilates, self-esteem, hair, and love); and Val Felso, for embodying the adage "when the going gets tough, the tough get going".

I also would like to single out several other dear friends in the neuroscience department that made the time here so much more fun and fulfilling (the reasons listed are in no way exhaustive). Matt Panichello, for his calm ear. Alex Piet, for developing our dance moves. Diana Liao and Alex Song, for ridiculously awesome massages. Kim Stachenfeld, for her unique kimness and empowerment of our fly team. Talmo Pereira, for his programmatic wizardry and "judicious" knowledge on how to go "full Brazilian". Olga Lositsky, for being the cutest person on earth, with a personality to match. Ben Deverett, for enjoayable car rides to retreat. And Luis Piloto, for his puns. The help of the administrative staff has made everything so much easier, especially for someone with my organizational skills. In particular, I would thank two graduate program administrators: Dawn Tindal, who is dearly missed, and Alex Michaud, who provided me with an unreasonable amount of support when the days were rough.

I also would like to thank the following people in the neuroscience department for memorable academic and/or personal interactions: Sam McDougle, Sama Ahmed, Jamal Willians, Angela Radulescu, Sarah Dubrow, Hessam Akhlaghpour, Ryan Ly, Flora Bouchacourt, Rolando Massis, Nicole Drummond, Yayoi Teramoto, Nathan Parker, Ashley Linder, Jonathan Berliner, Bas van Opheusden, Ahmed El Hady, Mingbo Cai,

Yeon Soon Shin, Lili Cai, Sam Lewallen, Pavlos Kollias, Abby Novick, Kara Enz, Sebastian Musslick, Laura Bustamante, Arthur Prat-Carrabin, Sarah Aghdam, Kevin Miller, Alex Riordan, Leenoy Meshulam, Kevin Lloyd, Mingyu Song, DJ Strouse, Daniel Wilson, Sam Zorowitz, Daniel Bennett, Anne Mennen, Victoria Ritvo, Aaron Bornstein, Ida Momennejad, Yael Niv, Orren Karniol-Tambour, Rachit Dubey, Cristina Domnisoru, Wouter Kool, Ed Clayton, Alex Lewis, Bob Wilson, Amitai Shenhav, Chris Baldassano, Rava Rivera, Ilana Witten, and Jonathan Pillow.

I also benefited a lot from classes, seminars, colleagues, and friendships in other departments, particularly the mathematics and computer sciences departments. First, I would like to thank some mentors in particular. Bernard Chazelle, for taking the time to be one of my readers and on my exam committee, for insightful discussions, for his course on natural algorithms, and for providing a role model for doing fun and rigorous science. Michael Aizenman, for altruistically sharing his wisdom on topics as complex as the Cauchy–Riemann equations, and as simple as the inverse relationship between one's ability to derive and the distance between their nose and the chalkboard (it really made me feel empowered to know that even he has this problem!). Emmanuel Abbe, for giving an amazing course on the theory of random graphs, which fundamentally shaped my research interests. Alan Sly, for helpful discussions and giving two courses that continued to feed this interest. Han Liu, who taught an excellent (and demanding) course on theory of statistical learning, which forged an incredibly valuable study group comprised of many smart and fun people: Nina Gnedin, Chase Perlen, Alex Beatson, Niranjani Prasad, and Kayla McCue. Gillat Kol and Mark McConnell for their enthusiastic courses on two fascinating topics: computational complexity and abstract algebra, respectively.

I have also enjoyed friendship and associated intellectual discussions with many others at Princeton. In particular, the mathematical discussions with Matt de Courcy-Ireland and Daniel Vitek are always enlightening. I would also like to thank: Bianca Dumitrascu,

Santiago Cuellar, Florian Sprung, Farzan Beroz, Shai Chester, Genna Gliner, Mochi Liu, Jordan Ash, Matheus Ferreira, Mattias Fitzpatrick, Katie and Andrew Wolf, Amit Halevi, Fernando Rossine, and Maryam Bahrani. An extra special thank you goes to our closest friend in Princeton, Matt Hernandez, who is basically family, for the intellectually stimulating discussions and all the rest.

On the topic of family acquired during my time in Princeton, I would like to thank Ashlyn Maria Bravo Gundermsdorff, our canine offspring. Her unconditional love and keen perceptual abilities continue to make me a better and more fulfilled person. And the family I gained from Lee: I lovingly thank my parents-in-law Brenda and Mark Gunderson for their love and support. Also, a special thank you to Megan Ihle, Alyssa Ihle, and Emily Gunderson for so patiently and dedicatedly doing my experiments.

I also would like to thank other people outside Princeton that were important during graduate school. In particular, I was lucky enough to have close to Princeton two very special friends from my home town in Brazil. Aurélio Amaral, for being one of my best friends since childhood, knowing me so well, and loving me anyway. Livia Camargo, for also being our family here, for giving us balance, fun, love and for all the rest. I want to specifically thank two dear friends I made during this period. Andra Mihali, for always having her door open, both physically and emotionally. Joe Olson, for giving much needed encouragement (and playlists) in these final two months. I also want to thank Yotta Theodoni, for her unique Greek hospitality. My dear friend, Bruno Pace, for non-stop crazy inspiring conversations about science and art since tempos imemorias. Claire Bradly, for a super fun visit. And the amazing people from the Minds, Brains, and Machine summer course for such an intellectually stimulating and fun time. In particular, Angela Bruno, Leo Casarsa, Leyla Isik, Daniel Estandian, Ning Leow, Kelsey Allen, Olivier Hénaff, Grace Cho, Gemma Noguera, Tomaso Poggio, Gabriel Kreiman, and Josh Tenenbaum.

I would like to thank two mentors that were instrumental for me to get to where I am. Read Montague, who gave me the opportunity to work in his lab doing research in neuroeconomics. Emmanuel Duppoux, my master's advisor, who was always very encouraging. And I also thank all the members of the emotional and social cognition lab at Catech for the amazing internship experience that got me excited to come to the US. In particular, my dear friends, Alma Gharib, Sean Brady, Ronnie Bryan, and Rosemary Rohde, for always providing me with much hospitality and love.

I also would like to thank these dear friends who gave me a family away from home in Paris: Lucille Lecoutre, Marine Guillanton, Thomas Andrillon and Chiara Varazzani, Max Magne, and Sophie Thomain (who I and the entire world lost way too early). I also thank my dear friends from childhood who continue to believe and support me during both good and bad times: Marina Croce, Camila Leal, Alice Sicuro, Thais Tavares, Maria Clara Pessoa, Marina Cortez, and Anna Carolina Estefan. I also include Annette Bordage Bessa, my french teacher in middle and high school, for opening my ears to the world. Finally, I would like to thank my family for raising me with so much love and always being supportive of my plans, no matter how unusual they were for them. Extra special thank you to my dad, Jorge Hermsdorff, and to my uncle-dad, Andre Schwarzer. And, last but definitely not least, my mom, Lucia Santiago Bravo, without her support and dogmatic belief in my abilities, I would likely not have even left Brazil.

To my strange loop, Lee M. Gunderson.

Contents

Abstract iii

Acknowledgements v

0 Thesis Roadmap **1**

1 Motivation **4**

1.1 Prelude 4

1.2 A Bayesian brain 5

1.3 Our brains rely on efficient priors 6

1.3.1 The success of the efficient coding hypothesis 6

1.3.2 Our priors over the structure of tasks shape our ability to learn them 8

1.4 Towards quantifying our priors over the structure of tasks 10

1.4.1 Formulating a tractable problem:

from general tasks to social and navigation "task graphs" 10

1.4.2 Overview of our approach 11

2 The Theory of MCMC with People **13**

2.1 Overview 13

2.2 Iterated learning as Markov Chain Monte Carlo with People (MCMCP) . . 14

2.2.1 From drawings to probabilities 14

2.2.2 MCMCP: a Bayesian model of iterated learning 16
2.3 MCMCP on graphs 19
2.3.1 Our experimental algorithm 20
2.3.2 Rate of convergence to the prior 21
2.4 Resource constrained MCMCP 24
2.4.1 Exploiting the Bayesian assumption 25
2.4.2 A combinatorial explosion 27
2.4.3 Practical advantages of our hierarchical parameterization 29
2.5 Methods 31
2.5.1 Mixing time 31

3 Introducing Graph Cumulants:
A Principled Framework for Quantifying Network Structure 33

3.1 Abstract 33
3.2 Motivation 34
3.3 Graph moments 36
3.4 Graph cumulants 38
3.5 Unbiased estimators of graph cumulants: Inference from a single network . 40
3.6 A natural hierarchical family of network models 41
3.7 Quantifying the importance of network substructures 44
3.7.1 Scaled graph cumulants 44
3.8 Graph cumulants for networks with additional properties 46
3.8.1 Directed edges 46
3.8.2 Node attributes 47
3.8.3 Edge weights 49
3.9 Discussion 52

3.10 Derivations and Methods 53
3.10.1 Efficiently computing graph moments 53
3.10.2 Python module for computing graph cumulants 54
3.10.3 Local graph cumulants 55
3.10.4 Graph cumulants are additive 57
3.10.5 Unbiased graph cumulants 60
3.10.6 Statistical inference without constructing an explicit null model . . 65
3.10.7 Fitting ERGMs using unbiased graph cumulants 66
3.10.8 A geometric understanding of the degeneracy problem of ERGMs . 69
3.10.9 Clustering analysis 75

4 Probing Network Global Structure: Spectral Graph Reduction 78

4.1 Prologue 78
4.2 Abstract 79
4.3 Motivation 79
4.4 Why the Laplacian pseudoinverse 82
4.5 Our graph reduction framework 84
4.5.1 Reducing edges and nodes 84
4.5.2 Preserving the Laplacian pseudoinverse 85
4.5.3 Minimizing the error 86
4.5.4 A cost function for spectral graph reduction 89
4.5.5 Node-weighted Laplacian 90
4.6 Our graph reduction algorithm 91
4.7 Experimental results 94
4.7.1 Hyperbolic interlude 94
4.7.2 Comparison with spectral graph sparsification 96

4.7.3 Comparison with graph coarsening algorithms 97
4.8 Conclusion 98
4.9 Derivations and Methods 99
4.9.1 Derivation of the optimal probabilistic action to an edge 99
4.9.2 Lifting the matrices of a contracted graph 101
4.9.3 Proof of the relationship between the hyperbolic distance and σ-spectral approximation 104
4.9.4 Number of edges acted upon per iteration can be $\mathcal{O}(|V|)$ 106
4.9.5 Efficiently computing m_e 108
4.9.6 Comparison of graph reduction methods using typical similarity measures 111

5 The Structure of Human Priors over Navigation and Social Tasks 114
5.1 Preamble 114
5.2 Main findings 115
5.2.1 Priors favor sparsity 115
5.2.2 Priors are more adaptive than iid connections 116
5.2.3 A regime change in the preferred density of connections 117
5.2.4 Priors favor more "egalitarian" configurations 118
5.2.5 Priors over social interactions favor triangles 120
5.2.6 Priors have non-trivial domain-dependent graphical structure 121
5.2.7 Higher order substructures reveal differences between domains . . . 123
5.3 Discussion 125
5.4 Experimental procedure 126
5.4.1 Experimental design 126
5.4.2 Exclusion criteria 128

5.4.3 Scalability . . . 129
5.4.4 Anecdotal interlude . . . 129
5.4.5 Experimental considerations . . . 130
5.5 Modeling . . . 131
5.5.1 Model selection . . . 132
5.6 Future directions . . . 132
5.6.1 Euler experiment . . . 133
5.6.2 Community detection experiment . . . 133
5.6.3 Analysis of these new experiments . . . 133
5.6.4 Comparison with real networks . . . 134
5.6.5 Cultural effects . . . 134

A Appendix for Graph Cumulants (Chapter 3) 135

A.1 Spectral motivation . . . 135
A.1.1 Simple graphs . . . 135
A.1.2 Generalizations . . . 138
A.2 Variance of unbiased graph cumulants . . . 140
A.3 Formulas for graph moments and graph cumulants . . . 143
A.3.1 Simple graphs . . . 144
A.3.2 Directed graphs . . . 146

B Appendix for Spectral Graph Reduction (Chapter 4) 150

B.1 Perturbations to eigenvalues of the Laplacian pseudoinverse . . . 150
B.2 Applications to graph visualization . . . 152

Bibliography 154

Chapter 0

Thesis Roadmap

The White Rabbit put on his spectacles.

"Where shall I begin, please your Majesty?" he asked.

"Begin at the beginning," the King said gravely,

"and go on till you come to the end: then stop."

—— Lewis Carroll,

Alice's Adventures in Wonderland

While not particularly linear, the work presented in this thesis can be brought into focus through the (curved) lens of striving to quantify our priors over abstract structures. In particular, we focus on graphs: the structure of interactions. The organization is as follows.

We begin at the beginning, with Chapter 1 discussing the questions about human learning, decision-making, and generalization abilities that initially motivated this thesis. In particular, we highlight how our representation of a new task (ie,[1] our prior over the structure of tasks) sharply constraints how fast and efficiently (if at all) we can solve it.

[1]The author agrees with the sentiment of the footnote on page xv of [53], viz, omitting superfluous full stops to obtain a more efficient compression of, eg: *videlicet*, *exempli gratia*, etc.

In Chapter 2, we describe the theory of Markov Chain Monte Carlo with People (MCMCP), a model of a recursive process whereby an agent learns from data generated by the previous agent. This framework has been used to construct experiments to estimate human priors across a variety of domains. Here, we consider it in a novel context: obtaining priors over graphs. We show that priors can be more efficiently recovered by leveraging the assumptions of the MCMCP framework and directly fitting a Bayesian model to the aggregated data. This is in contrast with the standard MCMCP approach of considering the data to be samples from the prior, once the chain has (hopefully) converged.

In Chapter 3, we describe the natural analogue of cumulants (eg, mean, (co)variance, skew, kurtosis) for graphs, building a hierarchical description of a network based on local correlations between an increasing number of connections. The prescription is general, seamlessly incorporating additional information often present in real networks, such as directed edges, node attributes, and edge weights. Our framework provides a principled method for quantifying the propensity of a network to display arbitrary substructures, and allows one to meaningfully compare networks of different sizes and edge densities. Moreover, it gives rise to a natural hierarchical family of maximum entropy models for networks. This family enjoys all the theoretical advantages associated with exponential random graph models (ERGMs), without suffering from their common practical pitfall known as the "degeneracy problem" (wherein typical samples from the model are remarkably different from the network they intend to model).

In Chapter 4, we analyze graph structure from an alternative perspective, via the global dynamics of diffusion throughout the graph, providing an algorithm that reduces a graph while preserving its large-scale structure. Using the graph Laplacian pseudoinverse $\boldsymbol{L}^{\dagger}$, our framework analytically unifies two active areas of research, namely graph sparsification (removing edges) and graph coarsening (merging nodes). We compare our algorithm with

several existing methods using real networks, and demonstrate that it more accurately preserves the large-scale structure.

In Chapter 5, we use the methods from Chapter 2 and the theory from Chapter 3 to quantify human priors over graphical representations of social and navigation tasks. As tasks in these domains have been relevant to our everyday life over evolutionary timescales, our brains have likely adapted to solve them. We find that the recovered priors have non-trivial graphical structure, requiring null models that constrain higher-order correlations. Some features appear to be general (such as the preferred amount of sparsity as a function of graph size), while other features appear to be domain-dependent (such as the preference for triadic closure in social interactions). Finally, after discussing future directions, we come to the end, then stop.

Chapter 1

Motivation

Solving a problem simply means representing it so as to make the solution transparent.

—— Herbert A. Simon,

The Sciences of the Artificial

1.1 Prelude

Our lives are punctuated by a multitude of seemingly disparate new tasks (eg, navigating a new place, interacting with new people, starting a new paragraph) that we are able to perform with relative ease. Still, if we consider all possible tasks we could be faced with, we would not (at least initially) be good at most of them (eg, the Stroop task [217, 60], predicting the financial market, writing a thesis). This simple observation leads to a fundamental, yet largely unanswered, question in cognitive neuroscience: are there essential structural properties that unite the tasks that our brains are "naturally" good at solving, and if so, what are they?

1.2 A Bayesian brain

As scarcity and chance are essential facts of life, effectively handling uncertainty is key to any organism's survival. The correct expression for the probability of a hypothesis given some new data is known as the Bayes' rule [22]:

$$p(\text{hypothesis} = h|\text{data} = d) = \frac{p(\text{data} = d|\text{hypothesis} = h)p(\text{hypothesis} = h)}{\sum_{h \in \text{hypotheses}} p(\text{data} = d|\text{hypothesis} = h)p(\text{hypothesis} = h)}, \tag{1.1}$$

Abstractly, we think of an organism as having a fixed[1] hypothesis space $\mathcal{H}$, containing every hypothesis h the organism could ever consider. Likewise, the world provides a fixed data space $\mathcal{D}$, containing every datum d that the organism could ever encounter. The organism's "belief"[2] about how likely a given hypothesis h is prior to observing any data is known as the *prior*, denoted as $p(\text{hypothesis} = h)$. After the organism observes new data d, they compute how likely each hypothesis h is to have "generated" these data, known as the *posterior*, $p(\text{hypothesis} = h|\text{data} = d)$. They compute this posterior using Bayes' rule (equation (1.1)), which requires their previous knowledge (ie, their prior) and their "model" of the data (known as the *likelihood*, $p(\text{data} = d|\text{hypothesis} = h)$). Naturally, the organism leverages their accumulated knowledge by using their updated belief about the hypothesis space (ie, their posterior) as their prior for a new situation; indeed, "today's posterior is tomorrow's prior" [156].

[1] We remark that there exist principled ways of incorporating a hypothesis space that grows as the organism learns, viz, nonparametric Bayesian statistics. However, such a discussion would lead us astray.

[2] If we completely abstract the meaning of the random variables h and d, Bayes rule (equation (1.1)) is simply proper counting. However, endowing h and d with meaning leads to many discussions of semantics. Indeed, when we say "prior", other humans have their own priors about what we mean by priors. Thus, we felt it necessary to briefly clarify our priors about "prior". We say the organism "believes", "infers", etc; these are just convenient terms, not a statement of volition on the part of the organism. Moreover, while we assume Bayesian agents in our analysis, we make no commitment about the neural implementation of generating the priors, representing them, or computing the posterior.

An extensive body of research suggests that many organisms approximate equation (1.1) for a wide variety of inference problems, ranging from basic sensory processes to higher-order cognitive functions (eg, [138, 75, 220, 69, 43, 33, 89, 178, 192, 73]). Importantly, the results of such computations rest crucially on the choice of a prior.

1.3 Our brains rely on efficient priors

The importance of priors cannot be understated. Indeed, many aspects of our perception and cognition can be understood through their quantification.

1.3.1 The success of the efficient coding hypothesis

A particularly successful story of how priors can help inform neuroscience begins with the investigation of the "efficient coding hypothesis". It states that sensory systems have evolved to maximize information transmitted to the brain [15], implying that priors over sensory inputs (here, the neural representations of new stimuli) have adapted to effectively encode the relevant statistics of the environment. This framework has contributed to major advances in our understanding of the neural code [192]. For example, the mammalian cochlea and auditory nerve fibers have properties that allow for efficient representation of the acoustic structure of speech and a wide range of other natural sounds [154, 166]. Moreover, higher-order computations such as visual attention [178] and working memory [164, 35] display analogous properties.

The visual system enjoys the most thorough investigation of the efficient coding hypothesis and is (not coincidentally) arguably the most well-understood sensory system (see Figure 1.1). For example, many properties of mammalian visual cells, such as sensitivity to orientation and spatial frequency, maximize the transmission of information, conditioned

on the statistics of natural images [81, 205, 206, 192]. Moreover, visual priors literally color our perception. Our color percepts can be described in terms of hue (the sensation of the relative redness, greenness, or blueness[3]), saturation (the perceived difference from neutral gray), and brightness (the apparent intensity). While computers frequently characterize colors using these attributes as a set of basis functions, the representation used by our retina is not beholden to such convenient interpretations. Instead, the visual system exploits statistical covariations present in natural scenes, directly influencing our perception of color [157]. This principle of efficient coding also explains several well-known visual illusions [110, 111] (Figure 1.1b), evidentiating a general and important computational tradeoff in biology: priors cannot be both exhaustive and efficient. That is, by efficiently distinguishing relevant visual information, our visual priors render us blind to insignificant differences. Might there be similar "illusions" with respect to our priors over the structure of tasks?

[3]If you are fortunate enough to be tetrachromatic, feel free to add a fourth color of your choice.

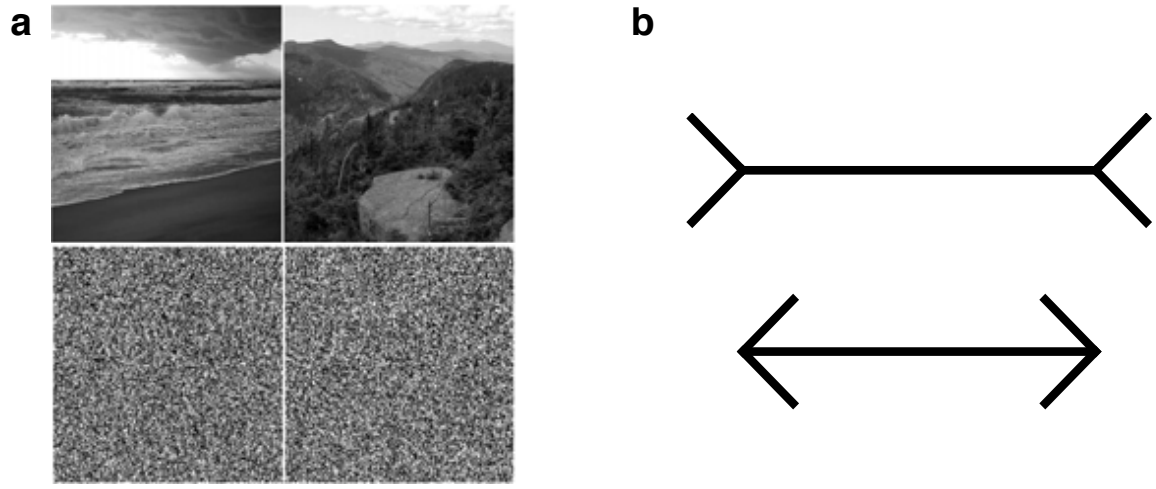


Figure 1.1: **Our visual priors color our perception.** In visual neuroscience, quantifying the statistics of natural images and interpreting them through the lens of the "efficient coding hypothesis" [15] has led to major advances in our understanding of the visual system, providing explanations for a wide range of phenomena. As such, visual priors are a key component of our understanding of the visual system. For example, in Figure **a)**, it is rather trivial for a human with proper vision to distinguish between the top right and top left images. However, this task is much more difficult for the bottom two images, even though their difference at the pixel level is rather similar to the difference between the top two images. This illustrates how our visual priors help us to distinguish scenes that are relevant to us, at the expense of rendering us blind to unnatural differences. Indeed, many visual illusions can be understood as exploiting these priors. For example, Figure **b)**, displays the Müller-Lyer illusion, in which the top line looks convincingly longer than the bottom one, even though their length is the same. A study by Howe & Purves [110] provided a quantitative explanation of this effect. They analyzed a large collection of 3D natural images, measuring the length of straight lines, conditioned on the structures at their endpoints. They found that lines with outward pointing structures (such as the top line) were on average longer in 3D space than those terminated by inward pointing structures.

1.3.2 Our priors over the structure of tasks shape our ability to learn them

The efficient coding hypothesis naturally extends to the domains of learning, decision-making and planning [34, 200, 46]. Here, we focus on the importance of priors over the structure

of tasks, ie, the brain's representation of new tasks. The processes involved in building such priors are more subtle [36]: an organism can (and should) act so as to "sample" tasks that maximize some sort of expected reward; the learning process that updates these priors is not well-understood; and the computations performed by the organism rest recursively on their priors over the structure of tasks. However, the basic principle still holds: these priors should efficiently encode useful invariants shared among tasks that are relevant to the organism (a process known in the literature as "learning to learn").

A property of high-dimensional spaces that is particularly irritating for effective learning (known as "curse of dimensionality" [25]) is that the inclusion of a few more dimensions in the representation of a task requires one to collect exponentially more data to learn said task (as volume grows exponentially in the number of dimensions). This suggests that priors over the structure of tasks should be compact, filtering out redundancies.

However, there is no free lunch [122]; reduced representations also constrain the set of tasks that one can efficiently solve. Therefore, these priors cannot be both exhaustive and efficient. For example, suppose that our brains *were* to represent every possible dimension of a task (eg, when learning when to cross a street, one would have to consider the number of trees, the weather, the color of the cars, etc). Given our finite lifetime, the amount of data we might hope to sample is truly a small parameter compared to the exponential number of combinations we would need to learn about; thus, we would never be able to generalize.

Indeed, as the prior used in a given task sharply constrains how quickly and efficiently (if at all) this task can be solved, such a prior should leverage on the useful structure of tasks that are relevant to the organism [34, 19, 200].

1.4 Towards quantifying our priors over the structure of tasks

1.4.1 Formulating a tractable problem: from general tasks to social and navigation "task graphs"

The covert nature of mental processes renders the quantification of any prior a challenging task. Due to its abstract nature, quantifying priors over tasks is indeed a formidable goal, if not only because the question "what is a task?" is arguably too open-ended. Thus, to make progress towards this goal, instead of attempting to codify the abstract structure of an arbitrary task, in this thesis, we focus on tasks in two specific domains: social and navigation (see Figure 1.2). Our motivation is twofold.

First, from a neuroscience perspective, these tasks have been relevant to our everyday life over evolutionary timescales. Thus, our brains have likely adapted to solve them. Indeed, there is much of evidence supporting this hypothesis (see Chapter 5 for further discussion). For example, the hippocampus appears to encode a spatial map of the environment [29, 163, 76, 176]. Likewise, there appear to be brain regions specialized in the processing of social information and theory of mind (ie, the modeling of others' mental states) [190, 189, 73, 3].

Second, from a methodological perspective, these tasks have a natural representation as graphs, which serves to make the space of tasks both discrete and finite (see Chapter 2) and allows us to use graph theoretical tools to quantify their structure (see Chapter 3).

Kaliningrad, Russia

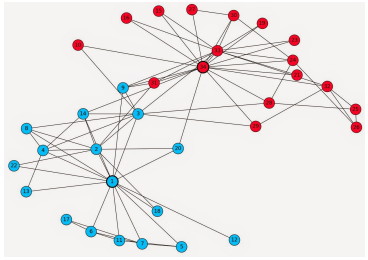
"⟶"

The seven bridges of Königsberg

facebook

facebook

"⟶"



Zachary karate club

Figure 1.2: **Formulating a tractable problem: from arbitrary tasks to "task graphs".** Quantifying human priors over tasks is a formidable goal, if not only for the reason that even the question "what is a task?" is arguably too open-ended. To make progress towards this goal, we focus on two specific domains: social and navigation. The top and bottom mappings refer to two abstractions that played a vital role in the development of network science. *Top*: The graph representing Euler's analysis of the seven bridges of Königsberg [78], which laid the foundations of graph theory. *Bottom*: The Zachary's karate club network [236], a small social network that has served as a popular benchmark for work in community detection. The quotes around the arrows are meant to clarify that (as any reasonable person) we are *not* arguing that the massively complicated space of social and navigation tasks can be faithfully reduced to the small graphs in the experiments in this thesis. However, in analogy with these examples, we *do* argue that starting with smaller, simpler examples is a sensible approach to eventually understand our priors over more complex tasks.

1.4.2 Overview of our approach

In brief, our framework for quantifying these priors relies on four key ingredients:

1. First, we abstract the notion of tasks, representing them as graphs.

2. Second, we collected human data in social and navigation experiments, using a method similar in spirit to Markov Chain Monte Carlo with People (MCMCP).

3. Third, we fit the aggregated data to a Bayesian model that leverages the MCMCP assumptions.

4. Lastly, to quantify these priors and make comparisons, we developed a novel natural hierarchical parameterization of distributions over graphs.

Chapter 2

The Theory of MCMC with People

Bob does have grounds for complaint, however:

the protocol tells him that he is communicating with Alice (who is honest)

but it does not cover the structured proof language Isar.

—— Confused Markov Chain,

A Markov chain trained on the bible and programing books [113]

2.1 Overview

In this chapter, we study some aspects of the theory of Markov Chain Monte Carlo with People (MCMCP) [93], a technique that has been used to estimate human priors for a variety of domains [224, 149, 235, 48, 135, 153, 234, 233]. Here we study it in a novel context, namely to obtain priors over graphs.

We show that the priors can be more accurately recovered by leveraging the assumptions of the MCMCP framework and directly fitting this model to the data.[1] This is in contrast with the standard MCMCP approach of recording data once the chain has (hopefully)

[1]This is somewhat ironic given the amount of effort that the author and Talmo Pereira (her collaborator in the construction of the experimental platform) poured into developing an online experimental platform that smoothly allocates participants to the appropriate chains in real-time.

converged. In order for our approach to scale to higher number of nodes (and for the results to be interpretable), we propose a novel hierarchical parameterization of probability distributions over graphs. We will explain the motivation, construction, and theoretical framework underlying this parameterization in great detail in Chapter 3. In this chapter, we demonstrate some benefits of this parameterization that are relevant for the analysis of participants' data (Chapter 5), namely that it allows for: more accurate reconstruction of the prior (*in silico* data), better generalization (*in sapiens* data), and effective recovery of meaningful summary statistics (*in data*).

We start our discussion with one of the first uses of iterated learning/MCMCP in psychology, followed by a detailed explanation of the MCMCP model and its assumptions.

2.2 Iterated learning as Markov Chain Monte Carlo with People (MCMCP)

Iterated learning refers to the process whereby an agent learns from data generated by another agent, who themselves learned it the same way, and so on (see Figure 2.2). It is an important and highly researched psychological phenomenon [136, 108, 167, 219, 94]. Under some assumptions (which we shall soon discuss), iterated learning can be viewed as a Markov Chain Monte Carlo algorithm (instantiated by the agents) that has as its stationary distribution the agents' shared prior over the relevant space.

2.2.1 From drawings to probabilities

Bartlett [17] was one of the first people (in 1932) to use iterated learning to study a psychological process (memory). The result of the experiment is displayed in Figure 2.1. While the results were evocative (hieroglyphic owls transmuted into sketches of cats through

iterated learning!), the experiment was lacking in quantitative analysis. We now describe how the framework proposed by Griffiths & Kalish [93] fills this gap.

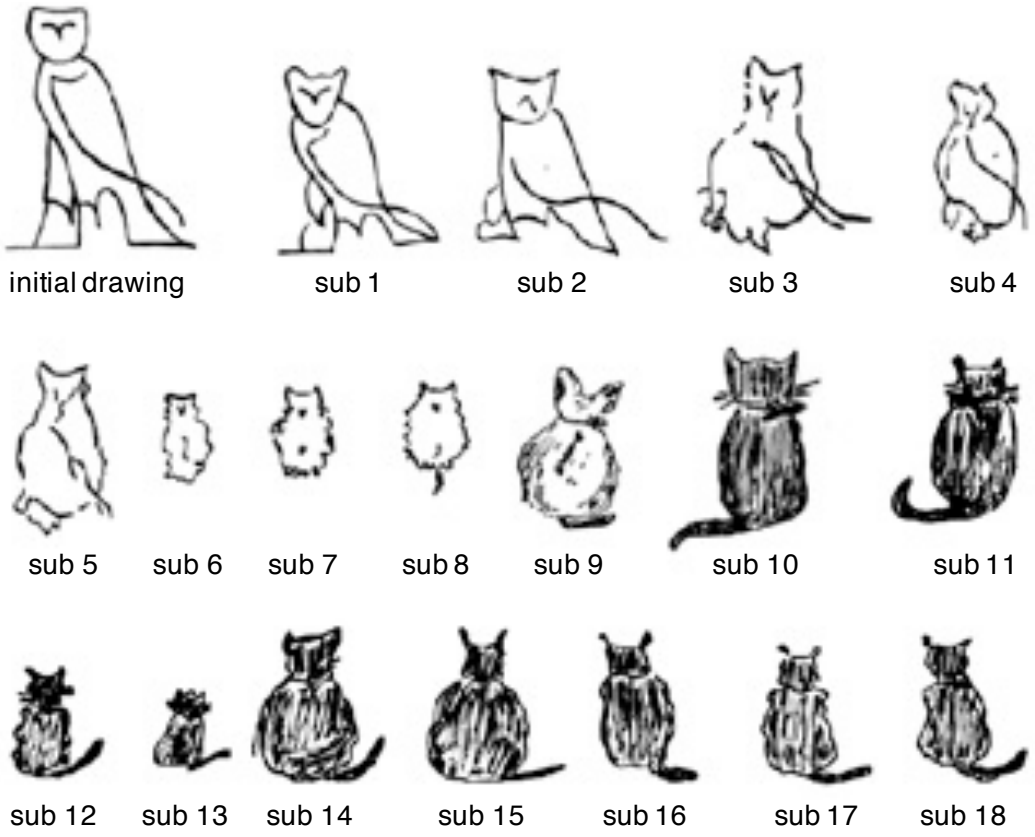


Figure 2.1: **Avian hieroglyphs beget feline sketches, or so it appears.** Shown here are the results from one of the first psychological experiments using iterated learning, conducted by Bartlett in 1932 [17]. He presented an abstract depiction of an owl ("initial drawing") to the first subject for one minute, then asked them to draw it from memory. The first subject's drawing was then shown to the next subject for the same amount of time, who then proceeded to draw it back from memory, and so on (each drawing is labelled by its artist). Exploiting the noisy transmission of information, the owl quickly evolved into a cat! The scientific challenge is to produce meaningful quantitative statements from such a result. We argue that to meet this challenge would require meaningful summary statistics of human priors over the space of such drawings. Unfortunately, thus far, this remains a daunting task. In this thesis, we make a first step in this direction by developing principled summary statistics of priors over the space of small graphs, considered as abstractions of social and navigation tasks.

2.2.2 MCMCP: a Bayesian model of iterated learning

As illustrated in Figure 2.2, the MCMCP model can be seen as a formalization of the "telephone game" [112].

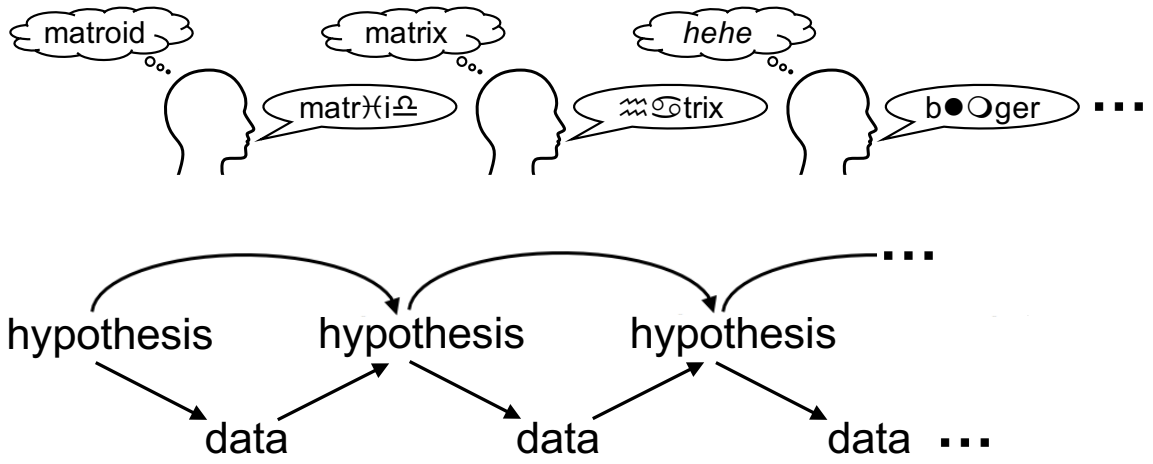


Figure 2.2: **Schema of Markov Chain Monte Carlo with People (MCMCP).**
Top: The MCMCP model is effectively a formalization of the globally famous "telephone game" [112]. One person starts the game (ie, the chain), with some initial condition (here, a word). This person noisily transmits this information to the next person (here, by whispering the word in their ear). That person does their best to understand this noisy data, and forms a hypothesis about its content (here, what the previous person said to them). Then, they whisper it into the next person's ear, and the chain continues as a back-and-forth between hypothesis and (noisy) data. Usually, the final word has nothing to do with the initial conditions, instead converging on something relevant to that group of people (eg, if you have ever played this game with young children, you find that the chain often converges to emotionally charged words in their lexicon, such as the one suggestively illustrated in the right-most speech bubble).
Bottom: This back-and-forth between hypothesis and data forms a Markov chain (MC) over the space of hypotheses (when marginalizing over the data). Under some assumptions, this MC converges to the participants' shared prior over the hypothesis space. The most relevant assumptions are that all participants: share the same prior (ie, they assign the same probabilities to each hypothesis in the hypothesis space), use Bayes rule to sample from their posterior to provide the data to next participant, and share the same likelihood function $p(d|h)$ (ie, the data are all produced in the same known way). Thus, the final "samples" (or data points) from the experiment can be treated as samples from the participants' shared prior. Indeed, this approach has been employed to quantify human priors in a diversity of contexts (eg, [224, 48, 235, 149]).

Let $\mathcal{D}$ denote the space of all data that the participants might observe (eg, in Figure 2.2, $\mathcal{D}$ is the space of all sounds a participant could make while whispering a word, and d is one such utterance). In this setup, the *amount of data transmitted at each iteration is fixed*[2] (eg, in Figure 2.2, a single whisper). Let $\mathcal{H}$ denote the space of all hypotheses that the participants might have about what the datum actually was (eg, in Figure 2.2, $\mathcal{H}$ is the space of all words (in the participants' relevant lexicon) and h is one such word). For simplicity, we consider both $\mathcal{D}$ and $\mathcal{H}$ to be discrete and finite (with size $|\mathcal{D}|$ and $|\mathcal{H}|$, respectively).

We assume that the participants are *identical Bayesian agents*, sharing the same beliefs and behavior. In particular, they are all assumed to have the *same shared prior* over hypotheses $p(h)$ and the *same* correct model for data generation, ie, *likelihood function* $p(d|h)$ (eg, in Figure 2.2, given a specific hypothesis $h \in \mathcal{H}$, all participants have the same probability of generating a given utterance $d \in \mathcal{D}$).

At iteration t, upon observing the datum d_{t-1}, the Bayesian participant computes the probability distribution over $\mathcal{H}$ given this d_{t-1} (ie, the posterior, $p(h|d_{t-1})$):

$$p(h|d) = \frac{p(d|h)p(h)}{\sum_{h \in \mathcal{H}} p(d|h)p(h)} = \frac{p(d|h)p(h)}{p(d)}. \tag{2.1}$$

We also assume that the participant chooses a hypothesis h_t by *sampling iid from their posterior*, and generate data d_t for the next participant in the chain using $p(d|h_t)$.

This back-and-forth between the data observed by the participants and their resulting hypotheses can be marginalized over the data to specify transition probabilities for a Markov chain (MC) over the space of hypotheses $\mathcal{H}$. As we have assumed that the participants share the same fixed decision rule, the transition probabilities do not change across iterations. Hence, we have a time homogeneous MC, characterized by the following transition matrix

[2] If the amount of data transmitted increases over time, then self-sustained learning can occur (as opposed to always converging to the participants' shared prior), see Chazelle & Wang [54, 55] for a mathematical analysis of this case.

$\boldsymbol{T}$, with entries:

$$\boldsymbol{T}_{ij} = p(h_{t+1} = i | h_t = j) = \sum_{k \in \mathcal{D}} p(h_{t+1} = i | d_{t+1} = k) p(d_{t+1} = k | h_t = j), \qquad (2.2)$$

where $p(h_{t+1} = i | h_t = j)$ is the probability that the participant chooses the hypothesis i after observing data generated from hypothesis j.

The transition matrix $\boldsymbol{T}$ encodes all information about the MC over the hypothesis space defined by the MCMCP model. Hence, we can study its dynamics and asymptotic behavior by simply using linear algebra. If $\vec{b}_0$ is a row probability vector (with $|\mathcal{H}|$ entries) encoding our initial knowledge about the hypothesis space, then $\vec{b}_1 = \boldsymbol{T}\vec{b}_0$ encodes this knowledge at iteration 1. Likewise, for iteration t, $\vec{b}_t = \boldsymbol{T}^t\vec{b}_0$.

In order to understand the asymptotic behavior of the model, we want to know if there is a probability distribution $\vec{\pi}$ that, given an arbitrary large number of iterations t, the chain always equilibrates to it, no matter the initial condition $\vec{b}_0$, ie,

$$\lim_{t \to \infty} \boldsymbol{T}^t \vec{b}_0 = \vec{\pi} \qquad \forall \vec{b}_0. \qquad (2.3)$$

Such a distribution $\vec{\pi}$ is called the stationary distribution of the MC with the transition matrix $\boldsymbol{T}$.

A necessary requirement for the probability vector $\vec{\pi}$ to be the stationary distribution of this MC, is that it must be an eigenvector of $\boldsymbol{T}$ with associated eigenvalue 1 (ie, the eigenvector associated with the largest eigenvalue, since $\boldsymbol{T}$ is a stochastic matrix). Moreover, $\boldsymbol{T}$ must have only a single eigenvalue with value 1, so as for $\vec{\pi}$ to be unique. This requirement translates to one more assumption in the MCMCP model, namely, that the MC is ergodic. That is, the chain is irreducible (ie, any state/hypothesis can be reached from any other state/hypothesis with a non-zero probability in a finite number of iterations, a fair assumption

given humans' non-zero probability of doing something strange/unexpected) and aperiodic (not an issue for our MCMCP over graphs, as there is a nonzero probability of forming the hypothesis that generated the observed data).

Taking all the assumptions together we can now easily verify the main result of the MCMCP framework, namely that the stationary distribution of $\boldsymbol{T}$ is equal to participants' shared prior. Representing the shared prior as a vector with entry i equal to $p(h = i)$ and substituting in the definition of $\boldsymbol{T}$ (equation 2.2), we have

$$\begin{aligned}
p(h_{t+1} = i) &= \sum_{j \in \mathcal{H}} \boldsymbol{T}_{ij} p(h_t = j) \\
&= \sum_{j \in \mathcal{H}} \sum_{k \in \mathcal{D}} p(h_{t+1} = i | d_{t+1} = k) p(d_{t+1} = d_k | h_t = j) p(h_t = j) \\
&= \sum_{k \in \mathcal{D}} p(h_{t+1} = i | d_{t+1} = k) \sum_{j \in \mathcal{H}} p(d_{t+1} = k | h_t = j) p(h_t = j) \\
&= \sum_{k \in \mathcal{D}} p(h_{t+1} = i | d_{t+1} = k) p(d_{t+1} = k) \\
&= \sum_{k \in \mathcal{D}} \frac{p(d_{t+1} = k | h_t = i) p(h_t = i)}{p(d_{t+1} = k)} p(d_{t+1} = k) \\
&= \sum_{k \in \mathcal{D}} p(d_{t+1} = k | h_{t+1} = i) p(h_t = i) \\
&= p(h_t = i).
\end{aligned}$$

Thus, the prior $p(h)$ satisfies the condition 2.3, and is therefore the stationary distribution $\vec{\pi}$ that we seek.

2.3 MCMCP on graphs

We now explain our algorithm for generating MCMCP experiments on graphs (Figure 2.3), relating it with the explanation of the MCMCP model from the previous section.

2.3.1 Our experimental algorithm

Figure 2.3 illustrates our algorithm for generating MCMCP experiments on graphs. (See Chapter 5 for a detailed explanation of the cover stories. Also, see this link for a video with a demonstration of our gamified online experimental platform, which allows participants to interactively draw the graphs.)

For a given chain, the graphs are with a fixed number of nodes (we ran experiments with 4, 5, 6, 7, 8, 10, 12, and 15 nodes, see Chapter 5), and we only consider simple graphs (ie, unweighted, undirected, networks with no self-loops or multiple edges). The data d are the pairwise relations shown to the participants (ie, a set of pairs of nodes and the presence or absence of an edge between each pair), which we refer to as "partial graphs", and $\mathcal{D}$ is the set of all possible partial graphs. The hypothesis space $\mathcal{H}$ corresponds to all simple graphs with n nodes, and h is the graph returned by the participant.

The assumption that participants share the same correct likelihood function implies that they know the algorithm used to produce the experiment. Here, this means that they believe that the partial graphs are generated by randomly erasing a fraction of the relations from some underlying graph. This is clearly articulated in our experiments.

The transition matrix $\boldsymbol{T}$ of our MC over the hypothesis space (ie, all unique/nonisomorphic simple graphs with n nodes) has entries:

$$\boldsymbol{T}_{ij} = p(g_j|g_i) = \sum\nolimits_k p(g_j|d_k)p(d_k|g_i) \tag{2.4}$$

where $p(d_k|g_i)$ is the probability of seeing partial graph d_k by randomly obscuring sr relations of the graph g_i (where s is the fraction of relations obscured, and $r = \binom{n}{2}$ is the total number of pairwise relations), and $p(g_j|d_k)$ is given by Bayes rule using a fixed prior.

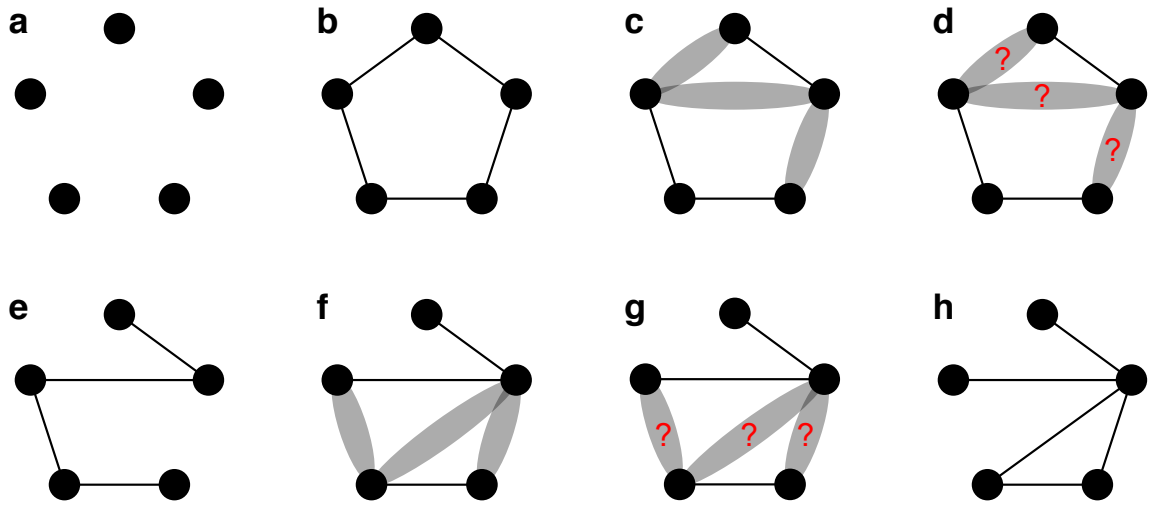


Figure 2.3: **Schema of our algorithm for generating MCMCP experiment on graphs.**
a) For a given chain, we restrict attention to a fixed number of nodes (here 5).
Our experimental algorithm consists of the following steps:
b) First, we *initialize* the chain by creating an underlying "task graph" for the first participant (eg, the nodes representing students and edges representing friendships, see Chapter 5).
c) Then, we *obscure* a fraction s of this graph's pairwise relations at random (here $s = \frac{3}{10}$), and show the remaining pairwise relations (the "partial graph") to the participant.
d) Then, we ask this participant to *infer* the obscured relations based on this partial information (eg, whether the pairs of students in gray ellipses are friends or not).
e) Then, based on the response of this participant, we *update* the task graph.
Then, we repeat this procedure: **f)** *obscure* relations; **g)** participant *infers* the obscured relations; **h)** *update* the task graph.
We refer to this sequence — **f)** obscure, **g)** infer, **h)** update — as one iteration.

2.3.2 Rate of convergence to the prior

A natural question to ask is: how long does it take for a given chain to converge to the prior? For small number of nodes, it is possible to enumerate all nonisomorphic graphs and explicitly construct the transition matrix (equation 2.4) for a given choice of prior over these graphs. The question of how fast the chain converges to the prior can then be answered exactly using linear algebra (see Section 2.5.1).

Figure 2.4 displays the asymptotic mixing time (defined as in Section 2.5.1) for chains over 6 nodes (156 nonisomorphic graphs) using different priors and fraction of relations obscured. As expected, the mixing time is highly dependent on both the overall shape of the prior, as well as the fraction of relations obscured. In fact, the mixing time can vary many orders of magnitude depending on these parameters. This poses potential difficulties in using MCMCP to recover priors: we do not know the shape of the true prior, collecting data with humans is costly, and obscuring too many relations could result in an experiment that is too underconstrained for participants to adequately engage and give their true prior.

In the special case of a simple prior not sensitive graphical structure and given by an Erdős–Rényi distribution $\text{ER}_{n,p}$ (ie, every potential edge is included independently with probability p), we have the exact expression for the asymptotic mixing time, namely:

$$\tau_m|_{\text{ER}} = -\frac{1}{\log(1-s)}, \tag{2.5}$$

where s is the fraction of relations obscured at each iteration.

Thus, as illustrated in Figure 2.5, for an $\text{ER}_{n,p}$ prior the asymptotic mixing time only depends on the fraction of relations obscured (but not on the number of nodes) and is relatively fast.

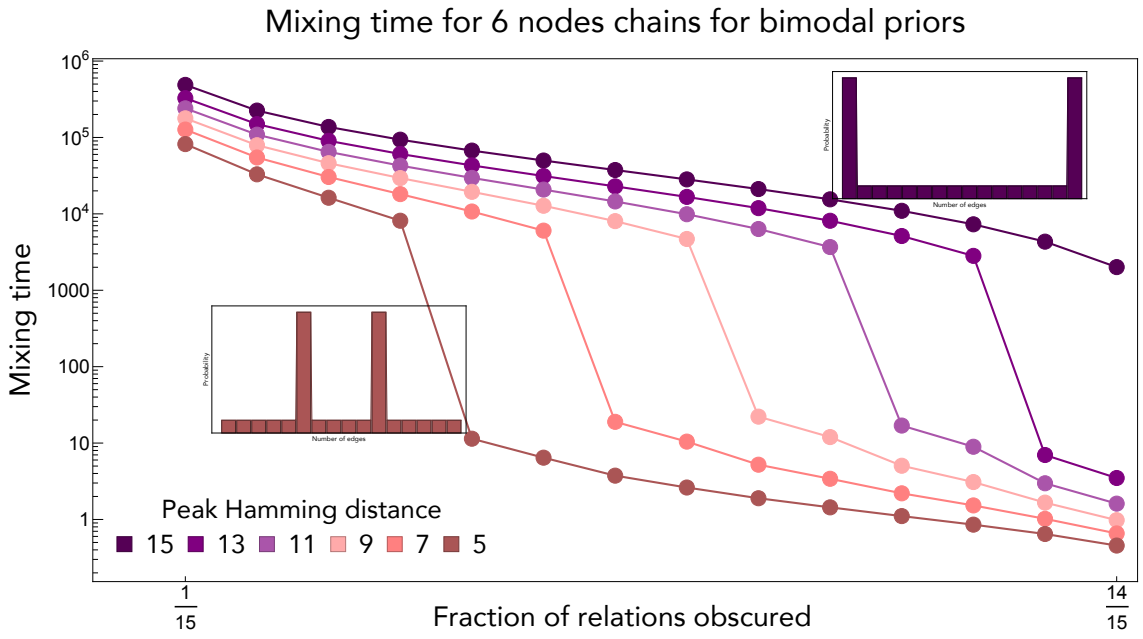


Figure 2.4: **Convergence rate to the prior is highly influenced by the shape of the prior and the amount of information shown at each iteration.** We computed the asymptotic mixing times τ_m of Markov chains over graphs (as defined in Figure 2.3) with 6 nodes (156 nonisomorphic graphs) for a range of multimodal priors and of fractions of relations obscured s. We parametrized the priors with probabilities given by the number of edges in the graph (here, 0 to 15). In particular, we gave 50% of the probability to graphs separated by x edges (the "peak Hamming distance") and distributed the rest of the probability uniformly to the other graphs. For example, for a peak Hamming distance of 5, we gave 25%, distributed equally, between all graphs with 5 edges, another 25% between all graphs with 10 edges, and distributed the remaining 50% equally between all others. As expected, the higher the fraction of relations obscured, the faster the mixing time (lower vertical axis). Conversely, the larger the distance (in terms of number of edges) between the peaks of the prior, the slower the mixing time. Varying these parameters leads the mixing time to vary many orders of magnitude.

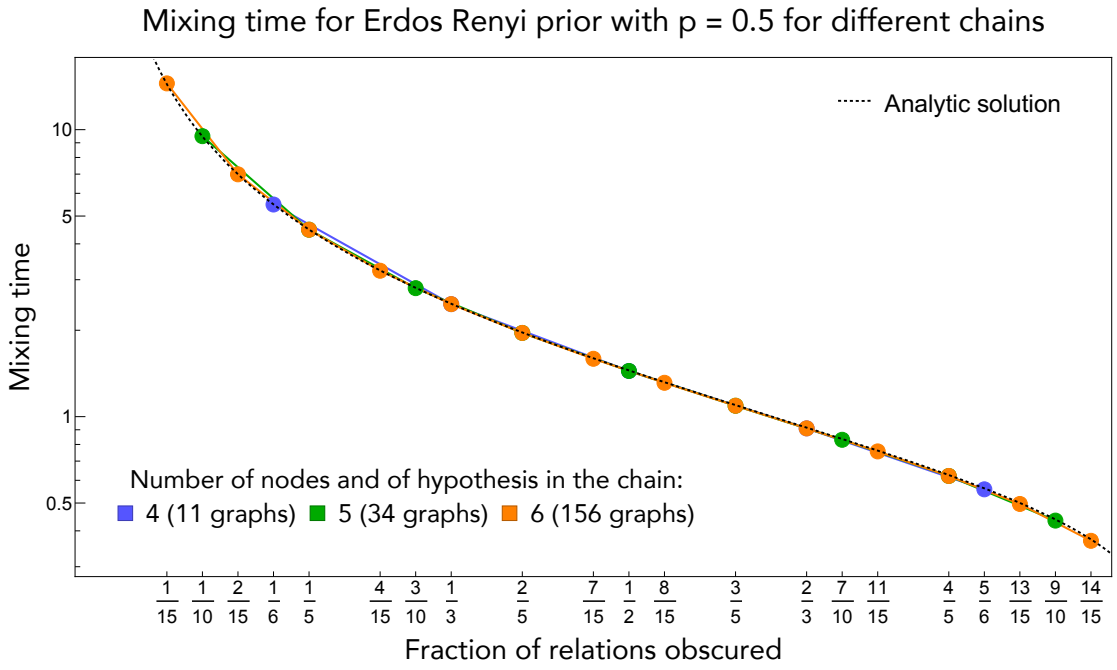


Figure 2.5: **Chains converge quickly for simple priors.** Displayed are the asymptotic mixing times τ_m of Markov chains over graphs with 4 nodes (11 nonisomorphic graphs), 5 nodes (34 nonisomorphic graphs) and 6 nodes (156 nonisomorphic graphs), for a variety of fraction of relations obscured s, and using Erdős–Rényi distributions $\mathrm{ER}_{n,1/2}$ as priors, where $\mathrm{ER}_{n,1/2}$ is not sensitive to any graphical structure (the probability of every edge is given by an independent coin flip). In this case, chains converge relatively quickly to the prior, with convergence times depending only on the fraction of relations obscured, independent of the number of nodes. In fact, there is a closed form solution for τ_m when the prior is $\mathrm{ER}_{n,p}$ (dotted black curve, equation (2.5)).

2.4 Resource constrained MCMCP

In standard MCMC, one uses samples (here the answers by the participants) generated by the algorithm to reconstruct the target (stationary) distribution (their prior). This process wastes much of the data for two reasons. First, one must discard the initial samples until the chain has (hopefully[3]) converged to its stationary distribution, the so called "burn-in" period [186]. Second, as the samples are correlated, one has fewer effective samples. While this

[3] As determining convergence can be non-trivial in certain cases, especially when the state space is large.

might not always be a problem (eg, when samples are generated using an efficient computer), in MCMCP the primary bottleneck is due to the use of human participants.

2.4.1 Exploiting the Bayesian assumption

Fortunately, in our case, we can use the experimental data more efficiently by leveraging the additional structure provided by the Bayesian assumption. Specifically, we propose to recover participants' prior by fitting the MCMCP model directly to their collective choices, as opposed to using the observed graph frequency (once the chain has (hopefully) converged) as a proxy of the prior as is done in standard MCMCP. In our model, the unknowns are the probabilities that the prior gives to each of the nonisomorphic graphs on the relevant number of nodes.

To compare these methods, we simulated our MCMCP experiment over graphs, assuming identical ideal Bayesian agents, respecting all assumptions of the MCMCP model (Figure 2.3). As illustrated in Figure 2.6, we find that the fitting method recovers the prior more accurately than the standard MCMC sampling method (especially in the case of constrained chain length).[4] In addition, our method does not require estimation of the mixing time, which, as shown in Figure 2.4, can vary dramatically depending on the prior, number of nodes, and fraction of relations obscured.

[4]We use KL divergence as a measure of distance to the prior (as opposed to, eg, total variation distance), as it is more sensitive to relative differences in probabilities. However, the results are similar when using other measures.

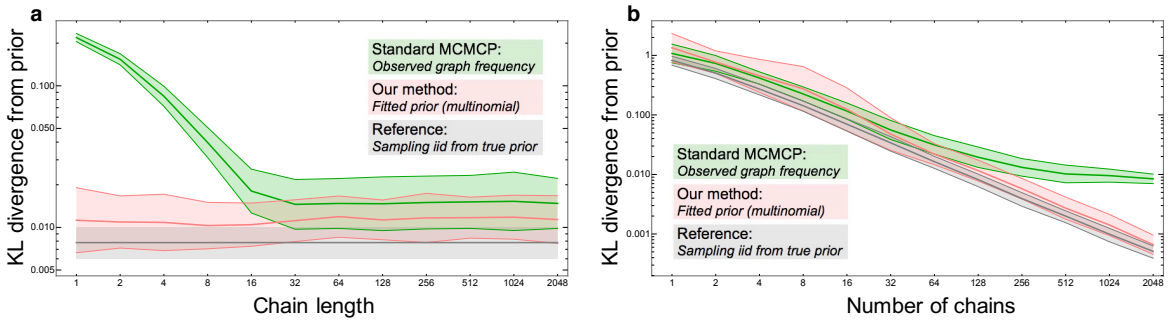


Figure 2.6: **The prior can be more accurately recovered by leveraging the assumptions of MCMCP and fitting this Bayesian model to the aggregate data.**
We simulated the response of identical ideal Bayesian agents (respecting all assumptions of the MCMCP model) playing our experiment on graphs with 5 nodes and with half of the relations being obscured at each iteration. The agents' shared prior was chosen to give an asymptotic mixing time of $\tau_m \sim 13$ iterations. For each simulation, we fit the resulting data to the MCMCP model with a multinomial prior using maximum likelihood estimation. We then computed the logarithm of the KL divergence from the agents' true prior to: the fitted prior (**red**); the observed frequency of graphs (**green**); and (as a reference) the distribution obtained by sampling iid from the true prior the same number of times (**gray**). Shading denotes ± 1 standard deviation about the mean for 64 simulations for a given choice of parameters.
a) For all simulations, we fixed the total number of data points (where each data point corresponds to the response of a single agent) across all chains to 2048, but varied the length of the chains that generated these data. Note that using the observed frequency of graphs is doomed to fail when the chain length is shorter than the mixing time τ_m, as there is not enough time for any chain to approach the prior. Moreover, fitting the data to recover the prior does better than using the observed frequency even when the chain length is much longer than the mixing time.
b) For all simulations, we fixed the chain length to 16, but varied the total number of chains. As the number of data points increases, fitting the prior continues to improve, while using the observed graph frequency asymptotes to some finite error. This asymptote is mainly due to the contribution from graphs in the beginning of the chains. While theoretically one could simply remove these initial samples, in practice one does not know precisely when the chain has sufficiently converged. Thus, our approach has the additional advantage of not having to estimate the mixing time, and appears to perform equally well regardless of chain length.

In Figure 2.7, we remove the issue of the burn-in period by initializing the simulated chains with a graph sampled from the underlying prior of the simulated agents. Even with the burn-in period removed, neighboring samples in a chain are correlated. This results

in an effective decrease in the number of samples obtained from such an MCMCP chain (asymptotically, one can approximate the effective sample size by dividing the total number of data points by the mixing time, although more precise estimates exist [115]). When the chain is longer than the mixing time, our method of fitting the aggregated data to the MCMCP model to recover the prior again outperforms the traditional MCMCP approach.

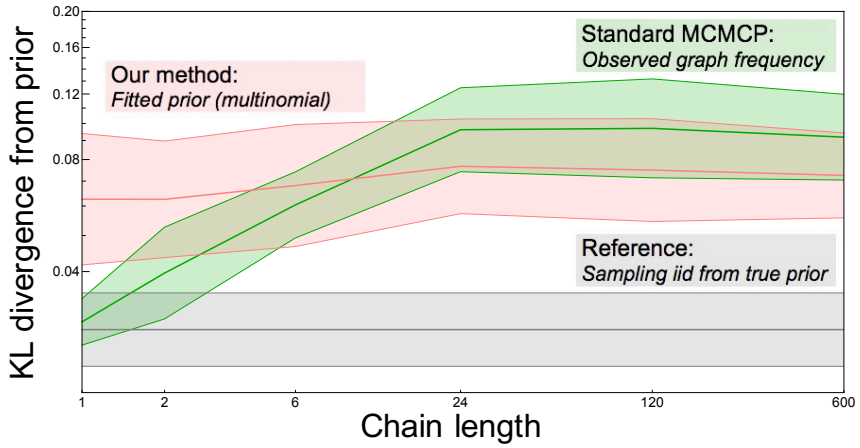


Figure 2.7: **Our fitting method outperforms the standard MCMCP sampling approach, even when the "burn-in" period is eliminated.** We generated synthetic data using the same specification as in Figure 2.6. However, we start with artificially pre-converged chains, by initializing the chains using a graph sampled from the true prior. Each simulation of the model has chains of a given length, with the number of chains chosen to provide 600 total data points. Clearly, when the chain length is 1, the perfect initialization renders the standard approach equivalent to sampling iid from the true prior. However, as the chain length increases, correlations between neighboring samples result in a decrease in the effective sample size, and the error when using the standard MCMCP sampling approach increases. When the chain length is $\gtrsim \tau_m$, our method of recovering the prior by directly fitting the MCMCP model to the data outperforms the standard approach.

2.4.2 A combinatorial explosion

The elephant in the room is that the results so far are predicated on the assumption that one can fit the entire prior using a multinomial distribution over the space of all nonisomorphic graphs with n nodes. The motivation for using MCMCP in the first place is that exhaustively

enumerating the hypothesis space is computationally intractable. Indeed, the dimension of the space of priors over graphs grows super-exponentially in the number of nodes (see Figure 2.8). Moreover, the number of samples will typically be vastly smaller than this dimension, leading to a highly underdetermined system.

This elephant also has a calf; even if we could completely and accurately determine such priors over graphs, the parameterization offers little in terms of interpretable summary statistics.

nodes:	unique graphs:	unique representations:
3	4	8
4	11	64
5	34	1024
6	156	32768
7	1044	2097152
8	12346	268435456
9	274668	68719476736
10	12005168	35184372088832
11	1018997864	36028797018963968
12	165091172592	73786976294838206464
13	50502031367952	30223145490365729367654 4
14	29054155657235488	24758800785707605497982484 48
15	31426485969804308768	40564819207303340847894502572032

Figure 2.8: **A combinatorial explosion.** The *middle* column corresponds to the number of unique (ie, nonisomorphic) simple graphs with n (*left* column) nodes, and the *right* column corresponds to the number of distinct representations (ie, the number of distinct $n \times n$ binary symmetric traceless matrices). The number of nonisomorphic graphs grows super-exponentially in the number of nodes [114], and it quickly becomes infeasible to fit the priors with perfect resolution. In this thesis, we propose a novel and principled hierarchical parameterization of such priors over graphs (see Chapter 3 for the parametrization and Chapter 5 for the application to human priors).

In the next chapter, we describe in detail a novel hierarchical parameterization of distributions over graphs that solves both of these pachyderm-related issues. In short, our parameterization allows for modeling of priors with increasing levels of structural complexity. The prior complexity is indexed by the model "order", with the first order corresponding to $\text{ER}_{n,p}$, the highest order corresponding to a full multinomial (ie, the probability of each

nonisomorphic graph can be independently specified), and the remaining orders interpolating smoothly and meaningfully between these two extremes.

2.4.3 Practical advantages of our hierarchical parameterization

We conclude this chapter by demonstrating some practical advantages of this hierarchical parameterization relevant to our analysis of participants' data in Chapter 5. In particular, we obtain: more accurate recovery of the prior in simulated data, where the ground truth is known (Figure 2.9); improved generalization of real data (Figure 2.10); and effective recovery of meaningful summary statistics (Figure 2.11).

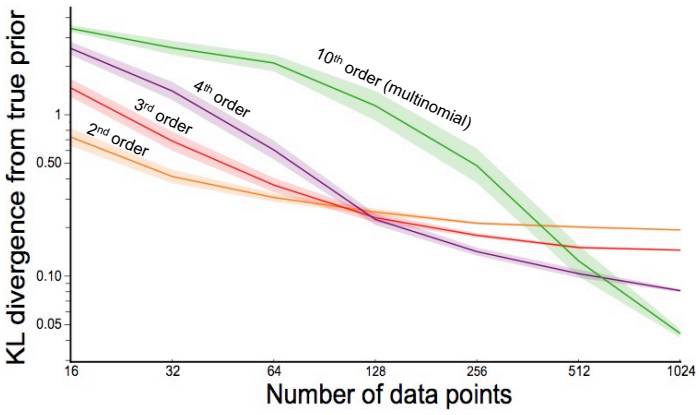


Figure 2.9: **Our hierarchical parametrization of distribution over graphs allows for more accurate recovery of the prior.** We generated synthetic data using the same specification as in Figure 2.6 with a chain length of 1. We fit the MCMCP model to the simulated data using our hierarchical parameterization of the prior for several choices of order (higher orders correspond to more structured/complex priors). Shading corresponds to ± 1 standard error about the mean for 64 runs of this simulation. When the data are limited, using a lower order parametrization recovers the prior more accurately. As the quantity of data increases, the ordering incrementally inverts until the model with highest complexity (ie, a full multinomial) does best. However, as the number of parameters in such a model is super-exponential in the number of nodes (and engaged human attention is expensive and difficult to obtain), the optimal order will typically be intermediate.

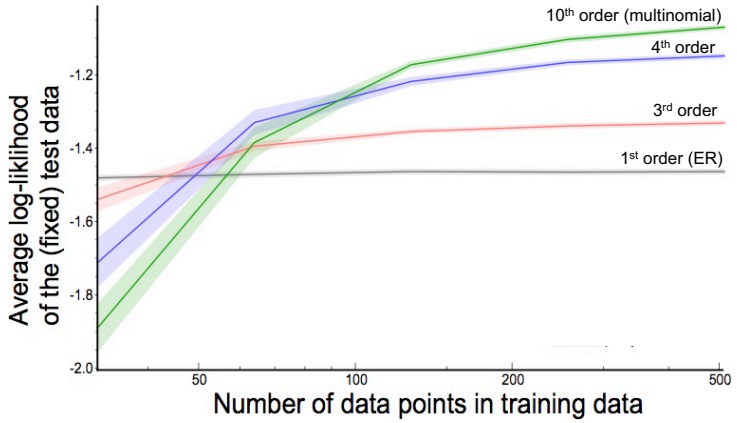


Figure 2.10: **Generalization of real data are improved when using our hierarchical parameterization.** We used 1210 data points from participants doing our experiment on 5 nodes. We randomly partitioned the data into test (698 data points) and training data, and fit the MCMCP model to the test data using our hierarchical parameterization of the prior for several choices of order (higher orders correspond to more structured/complex priors). Shading corresponds to ± 1 standard error about the mean for 64 repetitions of this process. In accord with the bias-variance tradeoff, when the data are limited, using a lower order parametrization to model the prior results in better generalization (higher log-likelihood of the unseen data). However, as the number of data points increases, higher order parameterizations do increasingly better. Again, in practice, the optimal order is typically intermediate.

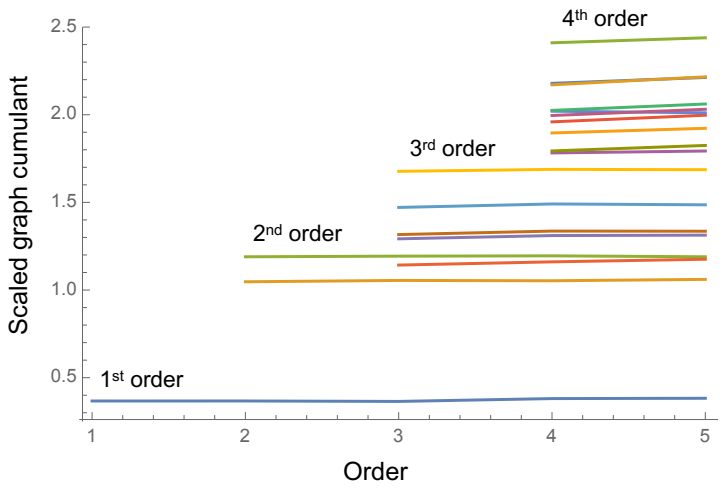


Figure 2.11: **Our hierarchical parameterization of the prior effectively recovers meaningful summary statistics.** We simulated identical ideal Bayesian agents with a realistic prior playing our experiment for graphs with 6 nodes. The prior was obtained by fitting a (fifth-order, see Chapter 3) model to the data of actual participants. The horizontal axis corresponds to the order of the fit and the vertical axis to relevant summary statistics (ie, the scaled graph cumulants, see Chapter 3), where fits of lower order have fewer relevant statistics. Fits of higher order have summary statistics that agree with those from lower orders. This desirable property will prove to be useful when analyzing participants' data (Chapter 5).

2.5 Methods

2.5.1 Mixing time

The rate of convergence to the prior depends on the magnitude of the second-largest eigenvalue λ_2 of the transition matrix $\boldsymbol{T}$, with the convergence becoming slower as λ_2 increases towards 1. Decomposing the solution in terms of eigenvectors yields $\vec{p}(t) = \sum_i c_i \vec{v}_i \lambda_i^t$. To determine the rate of convergence, we are interested in the longest exponential decay, which is determined by λ_2. We define the asymptotic mixing time τ_m as how many iterations (when the number of iterations t is large) it takes for the distance

between $p(t)$ and the stationary distribution to decrease by a factor of e. Here, the distance is measured in terms of total variation (TV) distance (although nearly any norm will do), where the total variation d_{TV} distance between two probability vectors $\vec{b}_t$ and $\vec{\pi}$ on the same space $\mathcal{H}$ measures the maximal error when approximating $\vec{\pi}$ by $\vec{b}_t$ to predict an event h, ie,

$$d_{TV}(\vec{b}_t, \vec{\pi}) = \|\vec{b}_t - \vec{\pi}\| = \sup_{h \subset \mathcal{H}} |\vec{b}_t(h) - \vec{\pi}(h)|. \tag{2.6}$$

Hence, d_{TV} is between 0 and 1. For the asymptotic mixing time τ_m, we have

$$\lambda_2^t = e^{-t/\tau_m} \qquad \Longrightarrow \quad \tau_m = |\ln(\lambda_2)|^{-1}, \tag{2.7}$$

as $0 < \lambda_2 < 1$ under our ergodicity assumption.

Chapter 3

Introducing Graph Cumulants: A Principled Framework for Quantifying Network Structure

Simplicity does not precede complexity, but follows it.

—— Alan Perlis,

Epigrams on Programming

This work is a collaboration with Lee M. Gunderson
and is part of a co-authored manuscript under preparation.

3.1 Abstract

In an increasingly interconnected world, it becomes imperative to understand and summarize the structure of these networks. However, this task is nontrivial; proposed summary statistics are as diverse as the networks they describe, and a standardized hierarchy has not yet

been established. In contrast, distributions of vector-valued random variables admit such a description in terms of their cumulants (eg, mean, (co)variance, skew, kurtosis). Here, we describe the natural analogue of cumulants for networks. This notion is general, seamlessly incorporating additional information often present in real networks, such as directed edges, node attributes, and edge weights. The proposed framework provides a principled method for quantifying the propensity of a network to display arbitrary substructures (such as cliques to measure clustering), and allows one to meaningfully compare networks of different sizes and edge densities. Moreover, it gives rise to a natural hierarchical family of maximum entropy models for networks. This family enjoys all the theoretical advantages associated with exponential random graph models (ERGMs), without suffering from their common practical pitfall known as the "degeneracy problem".

3.2 Motivation

The power of natural science relies on the ability to find abstract representations of complex systems and describe them with appropriate summary statistics. For example, using the pluripotent language of graph theory, the field of network science distills a variety of systems into the entities comprising them (the nodes) and their pairwise interactions (the edges), ie, networks/graphs. Remarkably many systems, which at the surface appear to be completely unrelated, naturally admit such a description [160]: physical (eg, electrical circuits [12], the cosmic web [64], Feymann diagrams [79]), biological (eg, brains [162], food webs [148], protein interactions [9]), social (eg, friendships [229], affiliations [31], collaborations [20]), and technological (eg, transportation networks [23], financial transactions [21], the internet [98]).

As the field of network science developed, several recurring themes were noticed [59, 14, 172]. Among the most well-known are: sparsity [174] (only a small fraction of

all possible connections exist); high clustering [173] (tightly connected groups of nodes); scale-free degree distributions [13] (most nodes have few connections and a few nodes have many); and the "small-world" phenomenon [230] (most pairs of nodes are only a few connections away from each other). As a consequence, real networks are often analyzed with measures tailored to capture these properties [77, 92, 72] and models are often chosen to mimic some set of them [230, 13, 63, 45]. However, network statistics are often intertwined [71, 179], for example, decreasing sparsity tends to promote both clustering and small-world properties. Moreover, many models, with different mechanisms [199, 107, 30, 129, 239, 45, 137, 49, 129], can reproduce similar sets of these "universal" characteristics. It is not currently clear how to compare between different network models (nor their associated statistics) within a single principled framework. A standardized hierarchy of descriptive network statistics and associated null models is needed.

To this end, we draw inspiration from the classical notion of cumulants of a random variable: a hierarchical sequence of summary statistics that efficiently encodes the underlying distribution [222, 196, 91]. While other hierarchical descriptions exist, cumulants are the natural choice due to their many desirable features [196, 91, 223, 97, 39]. For example, they are directly relevant to the fields of statistical and particle physics [128, 144, 165]. In particular, their unique additive nature when applied to sums of independent random variables [196] is integral to foundational results in probability and statistics, such as the central limit theorem and its generalizations [97, 91]. Moreover, the lower-order cumulants have intuitive interpretations. The first two orders, the mean and variance, correspond to the distribution's center of mass and its spread around it, and are taught in nearly every introductory statistics course [96]. The next two orders have likewise been given unique names (skew and kurtosis), and are useful to describe data that deviate from normality, appearing in a variety of applications, such as finance [95], economics [226] and psychology [28]. Indeed, cumulants are essentially universally used by the statistics community.

By generalizing the combinatorial derivation of cumulants [210, 211, 128], we derive the analogue for networks, which we refer to as *graph cumulants*. We first show how they give rise to a natural hierarchical family of maximum entropy null models for networks (Section 3.6). We then demonstrate their usefulness as intuitive statistical measures of the propensity of a network to display substructures of interest, allowing for systematic and meaningful comparisons between networks (even with different sizes or edge densities, Section 3.7). Finally, we describe their natural extension to networks with additional properties, applying this formalism to real networks containing directed edges, node attributes, and edge weights (Section 3.8). We also provide a publicly available code that computes these graph cumulants.[1] For ease of exposition, we first introduce our framework for the case of simple graphs, ie, unweighted, undirected networks with no self-loops or multiple edges.

3.3 Graph moments

The notion of cumulants of a real-valued random variable is more easily describable by first considering its moments (ie, the expected value of its powers). Real networks may be viewed as graph-valued random variables, and graph cumulants are likewise better understood by first considering graph moments.

To motivate our definition of graph moments, consider a network G with n nodes, and the most elementary measurement one can take of it, ie, that which provides the smallest nonzero amount of information. This measurement is a binary query, which yields 1 if an edge exists between a random pair of nodes in G, and 0 otherwise[2]. At first order, we

[1] The code will be made available when this work is officially published in a peer-reviewed journal.

[2] One might consider measuring a random node instead. However, the presence of a node *per se* does not give any new information. On the other hand, querying the degree of a node yields an integer that provides information about all of its edges, thus containing more information than the binary result from a query of a single edge.

consider repeated observations of a *single* such measurement. We define the first-order graph moment $\mu_{1/}$ (the "mean") as the expected value of this quantity, ie, the counts of edges in G normalized by $\binom{n}{2}$, the maximum possible number of connections between these nodes (ie, the counts of edges in the complete graph with n nodes). Hence, the first-order graph moment of a network G is equal to its edge density.

Expanding upon this notion, we define the second-order graph moments. We again consider a binary query, but now about the state of *two* potential connections. This query yields 1 if G contains edges between both pairs of nodes, and 0 otherwise. We now must distinguish between two cases: when the two edges share a node, thereby forming a wedge ($\mu_{2\wedge}$); and when they do not share any node ($\mu_{2/\!/}$). Each case is associated with a different second-order graph moment, which we analogously define as the counts of the associated subgraph in G, normalized by the counts of this subgraph in the complete graph with n nodes, $\frac{n!}{2(n-3)!}$ and $\frac{n!}{8(n-4)!}$, respectively.

Likewise, we define an r^{th}-order graph moment for each of the different configurations that r distinct edges can form. Again, μ_{rg} is defined as the counts of the subgraph g in G, normalized by the counts of this subgraph in the complete graph with n nodes. Hence, there is an r^{th}-order moment for each of the nonisomorphic subgraphs with exactly r edges (see Figure 3.1 for the subgraphs associated with graph moments up to third order).

The set of graph moments of a network G (or of a distribution over networks, when G is a graph-valued random variable) up to order r induces a hierarchy of descriptions of G with increasing precision. In Appendix A.1, we provide a spectral motivation for this hierarchy of graph moments.[3]

The scalability of our framework is determined by the computational complexity associated with counting the relevant connected subgraphs, as these imply the counts

[3]This description provided the initial motivation for this project. It was discovered when studying the theory of MCMCP over graphs (Chapter 2) and attempting to meaningfully describe the results from Chapter 5.

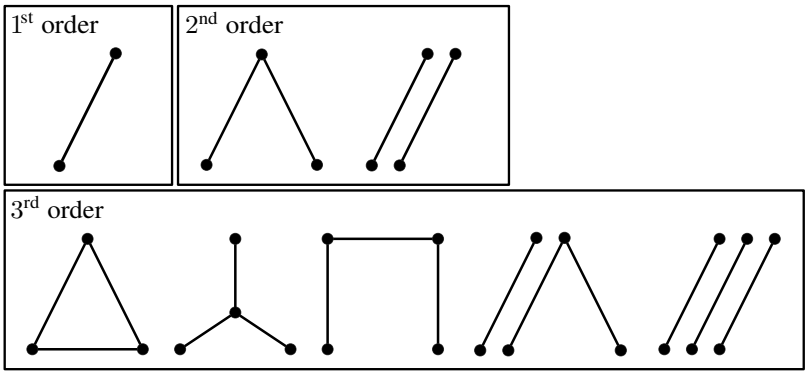


Figure 3.1: **Subgraphs associated with graph moments**. The graph moment μ_{rg} of a network G is defined as the counts of the subgraph g (with r edges), normalized by the counts of this subgraph if connections were present between all pairs of nodes in G. Displayed here are the subgraphs g associated with the graph moments up to third order for simple graphs. The full set of r^{th}-order moments contains all subgraphs with exactly r edges, including disconnected subgraphs.

of the disconnected subgraphs (see Section 3.10.1 for details). This is an important fact, as the counts of the disconnected subgraphs are generally orders of magnitude larger. Moreover, we can leverage numerous methods for efficiently counting connected subgraphs [68, 85, 1].

3.4 Graph cumulants

As the density of smaller subgraphs (eg, edges) increases, the appearance of larger subgraphs will clearly also tend to increase. Hence, we would like a measure that captures the difference between the observed value of a given graph moment and that which would be expected due to graph moments of lower order, so as to quantify the propensity for a specific network substructure (ie, subgraphs). Cumulants are the natural quantities with this desired property. For example, the variance quantifies the intuitive notion of the "spread" of a distribution (regardless of its mean), while the skew and the kurtosis reflect, respectively, the asymmetry

and contribution of large deviations to this spread. We now generalize this notion to obtain the graph cumulants.

While often defined via the cumulant generating function [97], cumulants have an equivalent combinatorial description. At order r, it involves the partitions of a set of r elements [210, 128]:

$$\mu_r = \sum_{\pi \in P_r} \prod_{b \in \pi} \kappa_{|b|}, \tag{3.1}$$

where μ_r is the r^{th} moment, κ_r is the r^{th} cumulant, P_r is the set of all partitions of a set of r elements, π is one of these partitions, and b is a subset of a partition π.

When generalizing this procedure to graph moments, the partitioning P_r of the edges must respect their connectivity (see Figure 3.2 for an example). These expressions can be inverted to yield the graph cumulants in terms of graph moments (summarized in Appendix A.3 and provided in our code).

Figure 3.2: **To expand a graph moment in terms of graph cumulants, enumerate the unique partitions of the edges in the associated subgraph**. Displayed here is the third-order graph moment associated with the 3-line subgraph ($\mu_{3\sqcap}$) expanded in terms of graph cumulants. The first term in the expression ($\kappa_{3\sqcap}$) corresponds to "partitioning" the 3-line graph into a single set of three edges, which inherits the connectivity of the original 3-line. The last term ($\kappa_{1/}^3$) corresponds to partitioning it into three sets, each with a single edge. The remaining terms ($\kappa_2\kappa_1$) correspond to partitioning the 3-line into a set with one edge and a set with two edges. This can be done in three different ways: in two cases (the two $\kappa_{2\wedge}\kappa_{1/}$ terms), the set with two edges has them sharing a node; and in one case (the $\kappa_{2/\!/}\kappa_{1/}$ term), the set with two edges has them not sharing any node.

Essentially, the defining feature of cumulants is their additive nature when summing independent random variables [196, 91, 66]. In Section 3.10.4, we show that the graph

cumulants of independent graph-valued random variables have the same additive property for a natural notion of summing graphs.

3.5 Unbiased estimators of graph cumulants: Inference from a single network

Thus far, we did not make a distinction between the graph moments of an observed network G and those of the distribution $\mathcal{G}$ from which this network was sampled. This is because, in a sense, they are the same: $\langle \mu_{rg}(G) \rangle = \mu_{rg}(\mathcal{G})$ (where the angled brackets $\langle \cdot \rangle$ denote expectation with respect to the distribution $\mathcal{G}$), a property known as "inherited on the average" [223].

However, for cumulants, this distinction is important. Cumulants of a distribution are defined by first computing the moments of the distribution, then converting them to cumulants (as opposed to computing the cumulants of the individual samples then taking the expectation of those quantities). Due to the fact that the cumulants are nonlinear functions of the moments, they are in general not preserved in expectation, ie, they are not inherited on the average [84]. For example, consider a sample of n observations from a distribution over the real numbers. The variance of these observations gives an estimate whose expectation is less than the variance of their underlying distribution, and one should multiply it by the well-known correction factor of $\frac{n}{n-1}$. Indeed, just as neglecting the factor of $\frac{n}{n-1}$ for the sample variance yields a biased result, if one simply uses the expressions given in Appendix A.3, then $\langle \kappa_{rg}(G) \rangle \neq \kappa_{rg}(\mathcal{G})$.

For higher-order cumulants, the generalizations of this finite sample size correction factor are known as the k-statistics [84]. They provide minimum-variance unbiased estimates of the cumulants of the true underlying distribution using a finite number of observations

[83, 117, 133, 132, 131]. In most applications [14, 172], one wishes to estimate a probability distribution over networks after observing only a single network. Thus, we desire analogous unbiased estimators for the graph cumulants of the underlying distribution from which the single observed network was sampled. In Section 3.10.5, we describe a procedure to obtain these unbiased estimators for graph cumulants, as well as their variance (Appendix A.2). In particular, we derive the exact expressions for the unbiased estimators of graph cumulants up to third order, as well as the variance for first order.

Deriving the expressions for the unbiased graph cumulants and their variance is a promising avenue for research (see Section 3.10.6) This would allow for principled statistical tests of the proclivity (or aversiveness) for arbitrary substructures without requiring one to explicitly construct a null model and sample from it (a procedure that is in general quite computationally expensive, and is one of the main obstacles when analyzing real networks [44, 179, 194]).

Moreover, we now discuss how these unbiased graph cumulants can be used to construct a principled hierarchical family of network models (see Sections 3.10.7 and 3.10.8 for more details).

3.6 A natural hierarchical family of network models

A ubiquitous problem, arising in many forms, is that of estimating a probability distribution based on partial knowledge [43, 218, 192, 110, 142, 59, 153]. Often, one has a set of properties that the desired distribution should have, but the problem is typically highly unconstrained. The maximum entropy principle [119, 120] provides a natural prescription: of the distributions that satisfy these constraints, choose the one that assumes the least amount of additional information, ie, that which maximizes the entropy. For example, when

modeling real-valued data, one often uses a normal distribution, the maximum entropy distribution (over the reals) with constrained mean and variance.

The analogous distributions for networks are called exponential[4] random graph models (ERGMs). These models are used to analyze a wide variety of real networks [44, 203, 216, 50, 82, 141, 229, 106, 142]. While it is possible to use any set of constraints in an ERGM, a common choice is to enforce that its expected counts of edges and of other context-dependent substructures (such as the wedge or triangle for social networks [194]) are equal to their counts in the observed network.

Unfortunately, ERGMs of this type often result in pathological distributions, exhibiting strong multimodality, such that typical samples have properties far from those of the observed network they were intended to model. For example, the sampled networks are often either very dense or very sparse. This phenomenon is known in the community as the "degeneracy problem" [52]. Much effort has gone into understanding this obstacle [125, 124, 123, 52], and while some approximate remedies have been proposed [109, 204, 208], a principled and systematic way to alleviate degeneracy has thus far remained elusive.

Based on our framework, we propose a hierarchical set of constraints which specify a family of ERGMs that is immune to the degeneracy problem (see Figure 3.3 and Section 3.10.8 for details). Specifically, our procedure for selecting an ERGM of order r' based on a single observed network is as follows (see Section 3.10.7 for details): *1)* Use the observed network to compute the unbiased estimators of *all* graph cumulants up to some order r' (the expressions are provided in Section 3.10.5 and included in our code); *2)* Convert these unbiased graph cumulants into the desired graph moments of the ERGM using the standard formulas (equation (3.34), the expressions are also provided in Appendix A.3 and included in our code); and *3)* Fit an ERGM that has these moments in expectation.

[4] In fact, any probability distribution in the exponential family is a maximum entropy distribution for a given choice of base measure and set of constraints. These models have been extensively studied and are widely used due to their numerous theoretical and practical advantages [7].

To the best of our knowledge, currently implemented ERGMs do not include all the subgraph counts (or equivalently moments) up to some order r'. This is perhaps due to the fact that not all of these subgraphs are deemed important to model the observed network, or because disconnected subgraphs are not usually thought of as substructures (or "motifs" [26]).

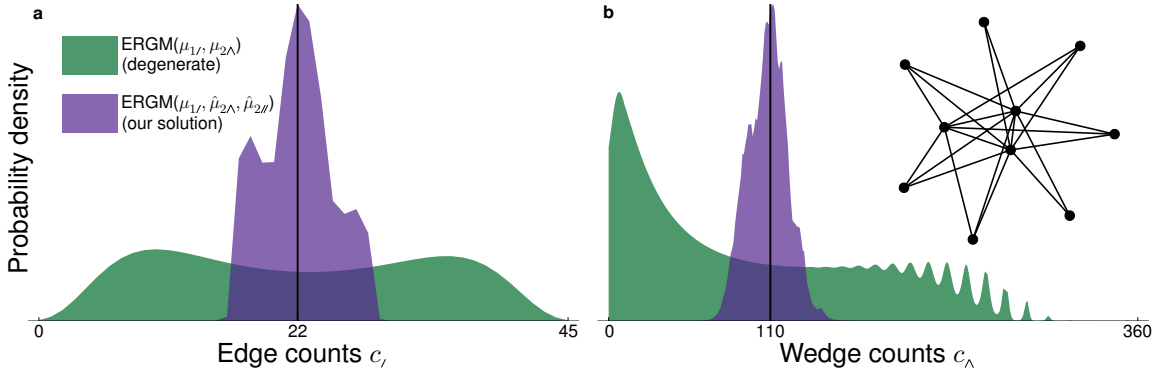


Figure 3.3: **Our proposed hierarchical family of ERGMs does not suffer from the "degeneracy problem"**. We computed the exact distributions over simple graphs on 10 nodes resulting from fitting two ERGMs to the same observed graph (shown on the right), and plot the resulting distributions of edge counts (**a**) and wedge (ie, 2-star) counts (**b**). To aid in visualization, we display continuous versions of the true discrete distributions. Green corresponds to a network model frequently used in the literature: ERGM$(\mu_{1/}, \mu_{2\wedge})$, the maximum entropy distribution with constrained expected counts of edges and wedges. Purple corresponds to our analogous second-order model: ERGM$(\mu_{1/}, \hat{\mu}_{2\wedge}, \hat{\mu}_{2/\!/})$, which constrains the expected counts of edges, as well as the (unbiased) counts of wedges and two edges that do not share any node. We fit the model using the procedure described in the Section 3.10.7, with unbiasing parameter $\eta = \frac{1}{11}$. Black lines denote the counts in the observed network that these distributions are intended to model. In sharp contrast to our proposed ERGM$(\hat{\mu}_{1/}, \hat{\mu}_{2\wedge}, \hat{\mu}_{2/\!/})$, the currently used ERGM$(\mu_{1/}, \mu_{2\wedge})$ can result in a distribution whose typical samples are notably different than the observed network. This is reflected in the fact that the green distributions have maxima far from their means. This undesirable behavior, known as the "degeneracy problem", tends to become even more pronounced for larger networks. In Section 3.10.8, we explain why this occurs and why our proposed family of ERGMs does not suffer from this problem.

3.7 Quantifying the importance of network substructures

3.7.1 Scaled graph cumulants

When considering different real-valued random variables, their cumulants are often scaled to yield dimensionless quantities that allow for interpretable comparisons. For example, the relative error is defined as the standard deviation divided by the mean ($\kappa_2^{1/2}/\kappa_1$). Likewise, we define the r^{th}-order scaled cumulant associated with the subgraph g (with r edges) as $\tilde{\kappa}_{rg} \equiv \kappa_{rg}/\kappa_{1/}^r$. While we will perform averaging and other computations using $\tilde{\kappa}_{rg}$, we will report the "signed" r^{th} root of the final quantity, ie, the real number with magnitude equal to $\left|\tilde{\kappa}_{rg}\right|^{1/r}$ and with the same sign as $\tilde{\kappa}_{rg}$. These scaled cumulants allow for a principled comparison of the propensity of different networks to exhibit a particular substructure, even when these networks have different sizes and/or edge densities.

To illustrate this point, we consider clustering, one of the hallmark features of many real networks [229, 230]. This notion is frequently understood as the prevalence of triadic closure (ie, triangles), and is often quantified by the clustering coefficient C, defined as the probability that two neighbors of the same node are themselves connected. While this quantity is easily expressed within our formalism as $C = \mu_{3\Delta}/\mu_{2\wedge}$, it is neither a cumulant nor dimensionless. We propose that the scaled triangle cumulant $\tilde{\kappa}_{3\Delta}$ is a more appropriate measure of clustering in networks, as demonstrated in Figure 3.4 (see Figure 3.11 for the natural extension to clustering in bipartite networks).

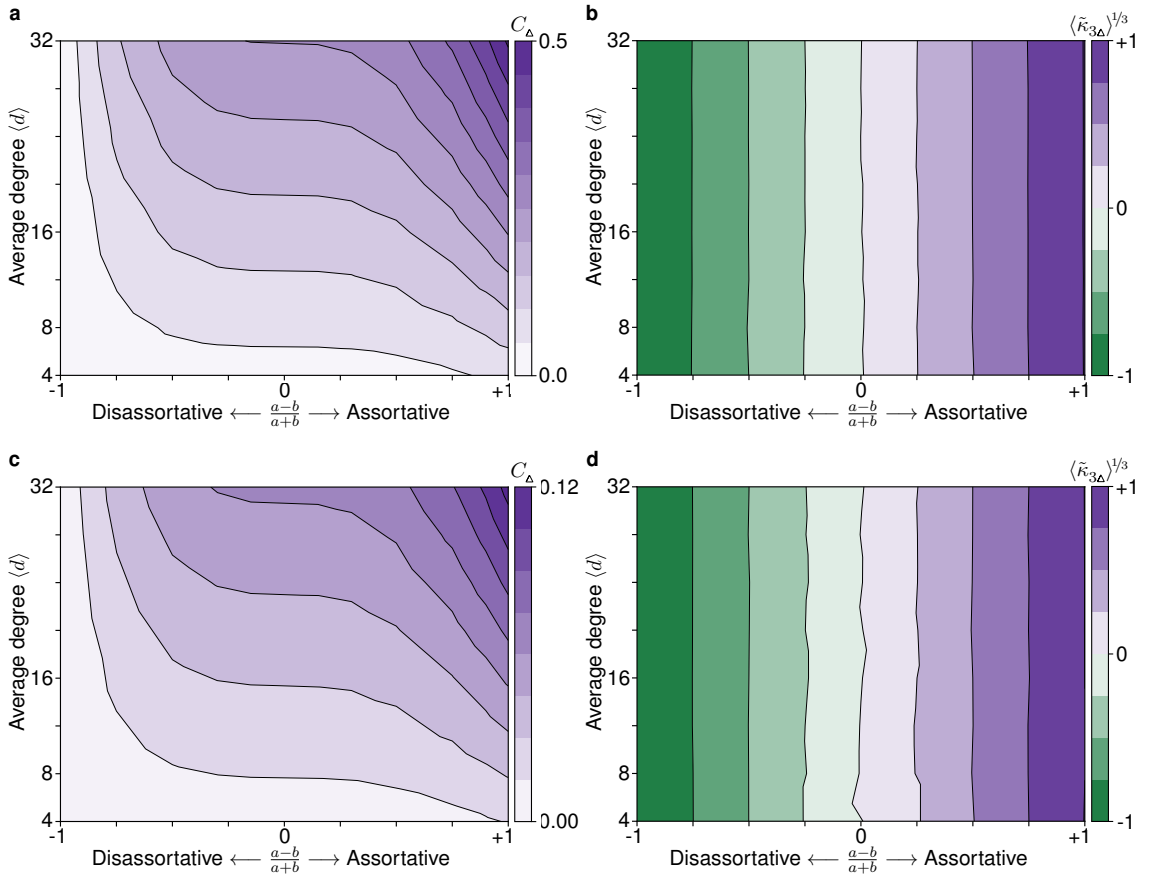


Figure 3.4: **The scaled triangle cumulant provides a principled measure of triadic closure that is invariant to network size and edge density**. We generated the symmetric stochastic block model on n nodes with 2 communities, SSBM$(n, 2, a, b)$, for a range of a and b, for 128 (*top* plots) and 512 (*bottom* plots) nodes. The horizontal axis indicates the level of assortativity: 0 corresponds to Erdős–Rényi (ER) graphs (no community structure), $+1$ to two disjoint ER graphs (only connections within communities), and -1 to random bipartite graphs (only connections between communities). The vertical axis indicates the edge density, in terms of average degree. **a,c)** Global clustering coefficient $C_\triangle$. **b,d)** Scaled triangle cumulant $\tilde{\kappa}_{3\triangle}$. For each set of parameters, we compute $C_\triangle$ and $\tilde{\kappa}_{3\triangle}$ for each instance of the model, and display the (signed) third root of the average value. While both measures increase with assortativity, the clustering coefficient also increases with average degree and decreases with increasing number of nodes. In addition, the scaled triangle cumulant has a natural interpretation: a value of 0 indicates that the number of triangles is precisely what is expected from the lower-order graph cumulants (manifestly true for the ER model), and a value of -1 indicates that there are no triangles in the network (as is the case in bipartite networks).

3.8 Graph cumulants for networks with additional properties

While it is possible to treat most networks as undirected, unweighted graphs, real networks frequently contain more information. Our formalism naturally incorporates many such augmentations. In general, the subgraphs associated with moments of order r are all the unique (including disconnected) graphs with r edges, endowed with the additional properties. The conversion to graph cumulants likewise respects this additional structure. We now discuss the specific cases of directed edges, node attributes, and edge weights (see Appendix A.1 for the case of hypergraphical networks), and illustrate their ability to quantify substructures using real networks. In Appendix A.3, we provide expressions for computing some of these augmented graph moments and graph cumulants. For clarity, we consider each of these properties individually. Combining these properties is relatively straightforward.

3.8.1 Directed edges

When analyzing a directed network, one must have graph moments that incorporate the orientation of the edges. While one still simply considers the number of directed edges at first order (as edge orientation only carries meaning when considered with respect to some other structure), there are now five second-order moments. The wedge configuration (two edges sharing one node) is now associated with three moments: one with both edges oriented towards the central node, one with both edges oriented away from it, and one with an edge towards and the other away. The relative orientation of two edges that do not share any node cannot be determined, and therefore is still associated with a single moment. Finally, the configuration of two reciprocal edges (ie, two nodes that are connected by edges in both directions) is associated with the fifth second-order moment. The appropriate normalization

is with respect to the counts in the complete directed graph, ie, that which has every pair of nodes connected by edges in both directions. Figure 3.5 illustrates the additional structured revealed by incorporating the directed structure of protein interaction networks.

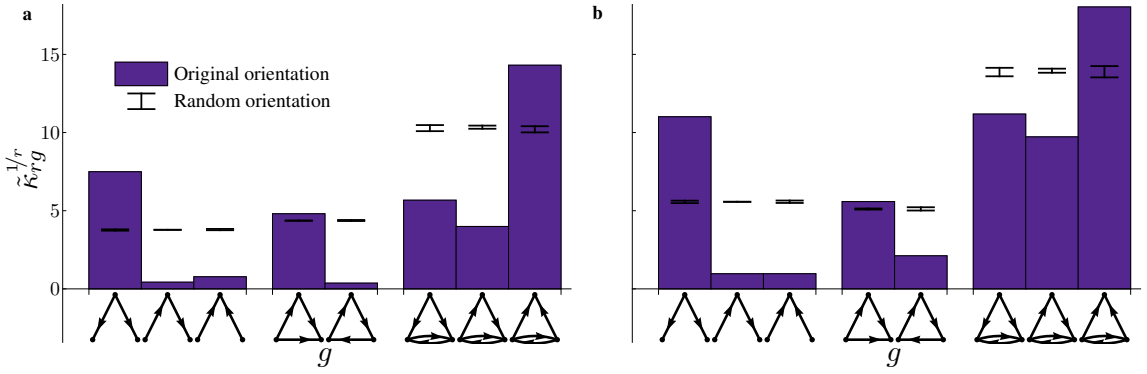


Figure 3.5: **The graph cumulant formalism naturally incorporates directed structure**. We use the scaled cumulants of directed networks to analyze transcriptional regulatory networks of protein interactions [27] of: **a**) yeast (4441 nodes and 12873 edges), and **b**) humans (3197 nodes and 6896 edges). As the number of directed subgraphs rapidly increases as a function of order r, we only display the scaled cumulants for substructures that are believed to be particularly important in such networks, namely, all the orientations associated with the $\tilde{\kappa}_{2\wedge}$, $\tilde{\kappa}_{3\triangle}$, and $\tilde{\kappa}_{4\diamondsuit}$. Error bars denote the mean and one standard deviation for the same network with randomized orientations. Despite the marked phenotypical differences between these two species, their regulatory networks display notable similarities. In particular, within the triangular structures ($\tilde{\kappa}_{3\triangle}$), the feedforward (or transitive) structures ($\tilde{\kappa}_{3\not\triangle}$) are significantly more prevalent than the cyclic structures ($\tilde{\kappa}_{3\not\triangle}$). Interestingly, within the wedge structures ($\tilde{\kappa}_{2\wedge}$), there is a higher prevalence of a central protein regulating many others ($\tilde{\kappa}_{2\not\wedge}$). While proteins that regulate each other display a propensity to both regulate the same other protein ($\tilde{\kappa}_{4\not\diamondsuit}$). Hence, for a given protein that interacts with many others, if the other proteins do not interact with each other, then the given protein has a propensity to regulate both of them. However, if the other proteins do regulate each other, then the propensity is for those proteins to regulate the given protein.

3.8.2 Node attributes

Often, the nodes of a network may be categorized via their attributes (eg, demographics for social networks). The graph moments of such networks are defined by endowing the

subgraphs with same attributes. For example, consider the case of a network in which every node has one of two possible "flavors": "charm" and "strange". There are now three subgraphs associated with the first-order moments: an edge between two "charm" nodes, an edge between two "strange" nodes, and an edge between one of each. To compute the moments, we normalize by their counts in the complete graph on the same set of nodes: here, $\binom{n_{\text{ch}}}{2}$, $\binom{n_{\text{str}}}{2}$, and $n_{\text{ch}}n_{\text{str}}$, respectively. Figure 3.6, we analyze a (binary) gendered network of primary school students [215], illustrating how incorporating node attributes can elucidate the correlations between node types and their connectivity patterns.

Bipartite networks (see Figure 3.11) are a common special case of networks with node attributes [170], where the nodes are assigned to a binary category and edges can only occur between nodes in different categories (eg, authors and publications [37], plants and pollinators [47]). The computations are essentially the same; the only differences are that unrealizable subgraphs are not considered, and the appropriate normalization is with respect to their counts in the complete bipartite graph $K_{n_{\text{ch}},n_{\text{str}}}$. For example, now there is only one first-order moment (an edge connecting a "charm" to a "strange"), and there are two second-order wedge moments: a "charm" node connected to two "strange" nodes, and a "strange" node connected to two "charm" nodes.

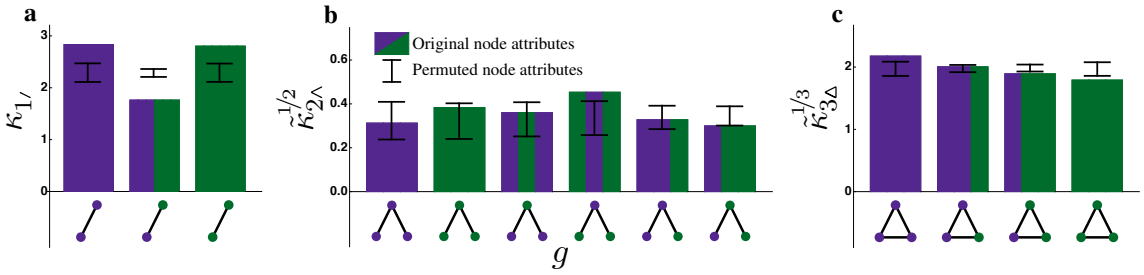


Figure 3.6: **Including node attributes reveals additional structure**. We use the scaled graph cumulants of weighted networks with a binary node attribute to analyze a social network (222 nodes and 5364 edges) of interactions between primary school students during one school day [215], with edge weights proportional to the number of interactions between pairs of students and with node attributes corresponding to their (binary) sex. Purple nodes indicate female students, green nodes indicate male students, and error bars denote mean and one standard deviation of the same network with the sex of the students randomly permuted. **a)** The three first-order cumulants. **b)** The six second-order scaled wedge cumulants. **c)** The four third-order scaled triangle cumulants. The first-order graph cumulants (ie, the density of edges between nodes of the indicated type) reveal a symmetric preference for homophily between the sexes, an effect well-documented in the social science literature. As all second-order scaled wedge cumulants are positive, we can infer a preference for hubs (ie, nodes with degree notably larger than the average). Likewise, as all third-order scaled triangle cumulants are notably positive, we can infer a preference for triadic closure, an effect also seen in many social networks [229, 194]. While there does not appear to be much of a difference between the sexes at second order, the third-order scaled cumulants suggest that triadic closure is more prevalent when more participants are female.

3.8.3 Edge weights

To motivate the definition of graph moments for weighted networks, consider an unweighted network as equivalent to a complete weighted network with edge weight equal to 1 if there is an edge, and 0 if there is not. If the subgraphs are counted with multiplicity equal to the product of the edge weights that comprise them, the moments of this weighted complete network are equal to those of the original unweighted network. Indeed, this equivalence naturally arises when considering the notion of summing graph-valued random

variables (see Section 3.10.4), and remains consistent for real-valued edge weights [160]. The normalization is the same as in the unweighted case, ie, divide the (weighted) count of the relevant subgraph by the counts of this subgraph in the unweighted complete network with the same number of nodes. Hence, weighted networks may have moments greater than one. In Figure 3.7, we analyze a weighted network of social interactions [116], illustrating how the incorporation of edge weights can increase the signal and even change the resulting interpretations.

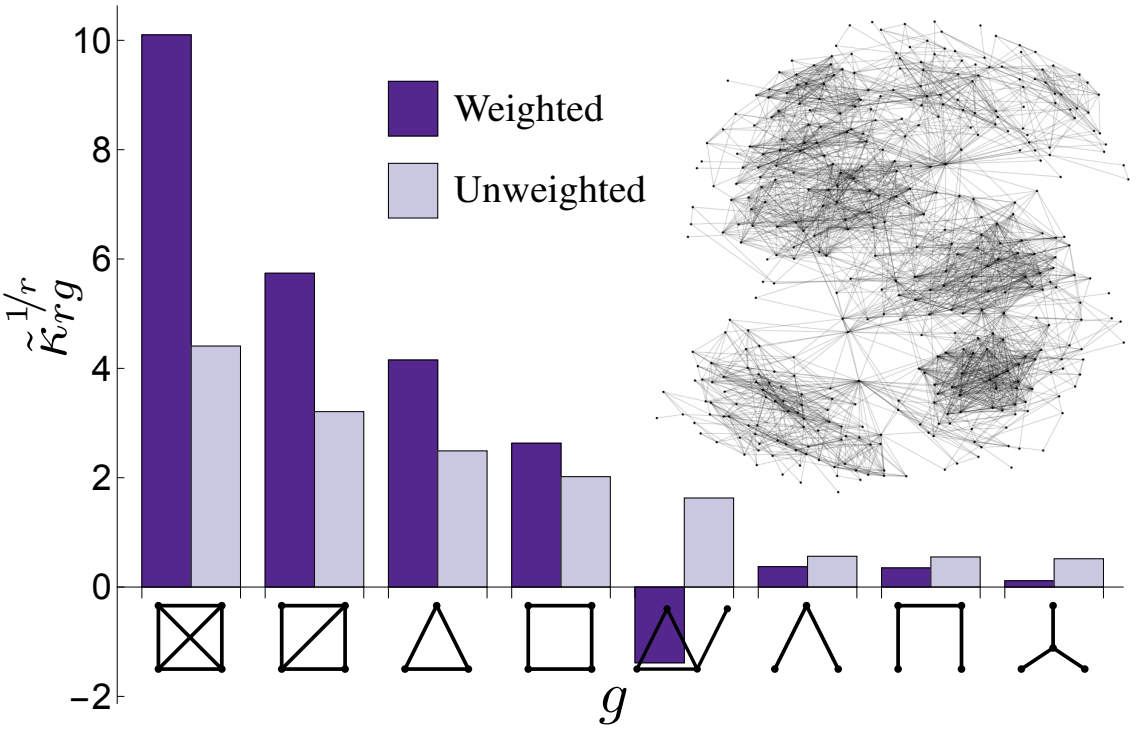


Figure 3.7: **Incorporating edge weights can increase the signal and possibly change the resulting interpretations.** We use the scaled graph cumulants for weighted networks to analyze a social network (410 nodes and 2765 edges) of face-to-face interactions during a 12-hour exhibition on infectious diseases [116], with edge weights proportional to the total time a pair of people spent interacting. The vertical axis corresponds to the value of scaled graph cumulants for the weighted network (in dark purple) and for the same network but unweighted (in light purple). The scaled cumulants associated with clustering (ie, $\tilde{\kappa}_{3\triangle}$ and $\tilde{\kappa}_{6\boxtimes}$) increase when using the true edge weights. Conversely, many of the others become smaller, in particular, $\tilde{\kappa}_{2\wedge}$ and $\tilde{\kappa}_{3\curlywedge}$, both of which are associated with power-law properties of the degree distribution. The most notable deviation occurs for $\tilde{\kappa}_{4\triangle\!\!\!/}$; it is positive in the unweighted network, but negative in the weighted network. This negative cumulant could be interpreted as an anticorrelation between triadic closure and people who interact with many others, not unreasonable for such an exhibition: the hosts are likely to talk to many different people, whereas groups of visitors tend to interact amongst themselves.

3.9 Discussion

In this work, we generalize the classical notion of cumulants to networks, thus providing a principled hierarchical framework within which different network models and their associated statistics may be systematically compared. Moreover, we provide a procedure for deriving unbiased estimators of these graph cumulants (analogous to the k-statistics [84, 83, 117, 133, 132, 131]). We also show how the formalism naturally extends to networks with additional structure, giving specific examples for directed edges, node attributes, and edge weights (Figures 3.5, 3.6, and 3.7).

We illustrate the utility of this framework by considering two major challenges in network science, namely, the generation of null models of networks [194, 179], and the quantification of the propensity of a network for various substructures [237, 26]. In particular, we describe a hierarchical family of maximum entropy network models that controls the "degeneracy" problem (Figure 3.3), providing a principled prescription for obtaining distributions that are clustered around the values they intend to model. As a concrete example, we show how the scaled graph cumulants can be used as an interpretable measure of clustering for both simple (Figure 3.4) and bipartite (Figure 3.11) networks, even when the networks have different sizes and edge densities. Aside from their mathematical interest, graph cumulants could aid in a wide range of other practical applications. We now highlight two such applications that we believe could be particularly impactful. Another application, namely modeling human priors over graphical representations of naturalistic tasks, is provided in Chapter 5.

First, as previously mentioned, the expressions for the unbiased estimators and their variance can be leveraged to perform statistical tests to measure the significance of arbitrary substructures without the need for constructing and sampling from a network null model. Deriving and implementing these expressions offers a principled, systematic, and highly efficient way to analyze networks with arbitrary metadata (see Section 3.10.6).

Second, the field of artificial intelligence has recently devoted much attention on devising principled ways to efficiently utilize graph-structured data [38, 41, 40, 88, 104, 134, 228, 100, 182, 151, 139], and local graph cumulants (see Section 3.10.3) offer a promising tool for this endeavor. For example, for node classification problems [99, 134, 228, 100], one could use the local graph cumulants of each node as the feature vectors for a clustering algorithm. Moreover, one could then recursively use the class assigned to the each node to compute the new local graph cumulants (augmented with these node attributes, i.e., the classes) as new feature vectors for the clustering algorithm.

The introduction of graph cumulants opens a wide avenue of research, and has the potential for addressing major issues in network science and related fields, such as machine learning on graph-structured data.

3.10 Derivations and Methods

3.10.1 Efficiently computing graph moments

To provide an intuition as to why disconnected subgraph counts are derivable from connected subgraph counts, consider the case of second-order moments for simple graphs. From first order, one has the counts of edges in the network, $c_{/} = \binom{n}{2}\mu_{1/}$. Now, consider all unordered pairs of distinct edges. Each pair corresponds to a single second-order count: either of a wedge, or of two edges that do not share a node, so $\binom{c_{/}}{2} = c_{\wedge} + c_{/\!/}$. Hence, the count of two edges that do not share any node $c_{/\!/}$ is directly derivable from the count of edges $c_{/}$ and the count of wedges $c_{\wedge}$.

A similar argument applies to all orders. For instance, at third order, one obtains relations for the disconnected subgraphs (ie, three edges that do not share any node, and a wedge and an edge that do not share any node) by considering all pairs of an edge and a wedge that do

not share an edge, as well as all triplets of distinct edges. This leads to the expressions:

$$c_{\wedge}(c_{/}-2) = c_{/\!/\wedge} + 3c_{\triangle} + 3c_{\lambda} + 2c_{\sqcap}$$
$$\binom{c_{/}}{3} = c_{/\!/\!/} + c_{\triangle} + c_{\lambda} + c_{\sqcap} + c_{/\!/\wedge}$$

We have algorithmically derived the expressions up to sixth order and incorporated them in our code.

We now discuss the scalability of counting the instances of a (connected) subgraph g with n' nodes in a network G with m edges and n nodes. The complexity of a naïve enumeration of the $\frac{n!}{(n-n')!}$ potential node mappings scales as $\mathcal{O}(n^{n'})$. However, there exist notably more efficient algorithms for certain subgraphs (such as triangles, stars, and cliques), especially when G has particular properties, such as sparsity [26, 70]. For example, the worst-case computational time complexity for counting n'-cliques is known to be at most $\mathcal{O}(n' m^{n'/2})$ time [57]. The counts of the r-stars are precisely proportional to the r^{th} factorial moments of the degree distribution, and thus can be computed as quickly as the degree distribution, namely in $\mathcal{O}(m)$ time. Moreover, some of these algorithms can be substantially accelerated through parallel computation and approximate values can be obtained by stochastic methods. Asymptotics aside, from a pragmatic perspective, Pinar et al. [183] showed that exact counting of all connected subgraphs with up to 5 nodes can be done for networks with tens of millions of edges in minutes on a commodity machine.

3.10.2 Python module for computing graph cumulants

We provide a python module that computes the graph moments and graph cumulants (as well as their unbiased counterparts). We use the python package igraph [67] to count instances of subgraphs in a network.

In conjunction with the symbolic computation available in Mathematica [231], we automatically derived the expressions for the counts of disconnected subgraphs in terms of the connected counts (see Section 3.10.1), as well as the expressions for converting graph moments to graph cumulants (see Appendix A.3).

Our python code contains the expressions to obtain graph cumulants up to: sixth order for undirected unweighted networks, fifth order for undirected weighted networks (plus a single K_4 at sixth order), fifth order for directed unweighted networks (plus a single K_3 at sixth order), subgraphs of $K_{2,2}$ for bipartite networks, and subgraphs of K_3 for networks with binary node attributes.

3.10.3 Local graph cumulants

Graph moments and graph cumulants are essentially properties of the network as a whole. However, for tasks such as node classification [99, 134, 228, 100] and link prediction [155, 207, 6, 155], one often desires quantities that depend only on the local neighborhood (eg, the local clustering coefficient of the node [173] as compared to the global clustering coefficient of the network). Here, we describe how to derive local graph moments and cumulants associated with a node, providing the expressions necessary to compute the local triangle cumulant.

We define the local graph moments of a node as the density of rooted subgraphs with this node as its root. For example, for simple graphs, there are now two first-order moments. The first is associated with an edge between the root node and another node, and is therefore the degree of the rooted node, normalized by its degree in a complete graph on the same number of nodes:

$$\mu_{1\curlywedge} = \frac{c_{\curlywedge}}{n-1},$$

where the star denotes the rooted node. The second first-order moment is associated with an edge between two nodes, neither of which are the root node. Thus, it is the counts of edges that do not use this node, normalized by this count in the complete graph on the same number of nodes:

$$\mu_{1\overset{\star}{-}} = \frac{c_{\overset{\star}{-}}}{\binom{n-1}{2}}.$$

Likewise, one has:

$$\mu_{2\overset{\star}{\wedge}} = \frac{c_{\overset{\star}{\wedge}}}{\binom{n-1}{2}},$$
$$\mu_{2\overset{\star}{\angle}} = \frac{c_{\overset{\star}{\angle}}}{2\binom{n-1}{2}},$$
$$\mu_{3\overset{\star}{\triangle}} = \frac{c_{\overset{\star}{\triangle}}}{\binom{n-1}{2}}.$$

The definition of local graph cumulants follows the same procedure as before, now taking care to incorporate the presence of this rooted node, ie,

$$\kappa_{3\overset{\star}{\triangle}} = \mu_{3\overset{\star}{\triangle}} - \mu_{2\overset{\star}{\wedge}}\mu_{1\overset{\star}{-}} - 2\mu_{2\overset{\star}{\angle}}\mu_{1\overset{\star}{/}} + 2\mu_{1\overset{\star}{/}}^2\mu_{1\overset{\star}{-}}. \tag{3.2}$$

The scaled local triangle cumulant is likewise defined as

$$\tilde{\kappa}_{3\overset{\star}{\triangle}} = \frac{\kappa_{3\overset{\star}{\triangle}}}{\mu_{1\overset{\star}{/}}^2\mu_{1\overset{\star}{-}}}. \tag{3.3}$$

A similar procedure can be used to obtain local graph cumulants associated to the edges, where instead of designating a rooted node, one designates an edge. This could be particularly useful for link prediction problems.

3.10.4 Graph cumulants are additive

Essentially, the defining property of cumulants is their additive nature when summing independent random variables [196, 91, 66] (eg, $\text{Var}(X+Y) = \text{Var}(X) + \text{Var}(Y)$ when X and Y are independent). In this section, we first define a natural notion of "summing" graph-valued random variables $\mathcal{G}_1, \mathcal{G}_2, \ldots$ (all with the same number of nodes), then we show that the graph cumulants of these distributions sum when the samples from them are independent.

As with real-valued random variables, we consider summing the random variable $\mathcal{G}$ and not a particular instantiation G. For ease of explanation, consider a single network G to be the random variable $\mathcal{G}$ with a delta function as its probability distribution (we refer to such a distribution as a network). As the distributive law holds, our treatment generalizes to arbitrary distributions.

There are a variety of operations that compose two networks, such as the disjoint union and a variety of graph products [175]. Here, we consider the sum of two networks (with the same number of nodes) $\mathcal{G}_1 + \mathcal{G}_2$ to be at the level of their adjacency matrices, defined by simply adding the entries component-wise. In general, the same network can be represented by many adjacency matrices, thus we must assign equal probability to each of these possibilities for the operation to be well-defined. In particular, if a network $\mathcal{G}$ with n nodes is representable by an adjacency matrix $A = \{a_{ij}\}$, then one distributes the probability associated to this network uniformly over all matrices $A' = \{a_{\pi(i)\pi(j)}\}$ for all permutations π of $\{1, \ldots, n\}$. Thus, when summing two networks, one considers all the ways that their sets of representative adjacency matrices could sum. The result is a new graph-valued random variable over weighted networks.

As this summation yields a distribution over weighted networks, to show that their graph cumulants sum, we must generalize the notion of subgraph density to this weighted networks

(itself a useful extension). Previous works have considered several ways to generalize counts of substructures of unweighted networks (eg, triangles to measure clustering) to the weighted case [177, 16, 201, 8]. Within our framework, the consistent prescription is to treat a weighted edge as a collection of multiple edges that sum to its weight. Thus, when counting subgraphs, one should consider each instance with multiplicity equal to the product of the edge weights that compose it [160] (as opposed to considering each instance of a subgraph equally). The normalization to graph moments is the same as before, ie, the counts of the subgraphs in the unweighted complete network (thus, the moments may now be greater than one). Likewise, the conversion from graph moments to graph cumulants remains identical (and the expressions are included in our code).

With the definitions for summing graph-valued random variables and how to compute moments and cumulants on the (weighted) output, we can now state the main result of this section: *For two independent graph-valued random variables over n nodes, $\mathcal{G}_1$ and $\mathcal{G}_2$, the graph cumulants of their sum is the sum of their cumulants, ie,*

$$\kappa_{rg}(\mathcal{G}_1 + \mathcal{G}_2) = \kappa_{rg}(\mathcal{G}_1) + \kappa_{rg}(\mathcal{G}_2),$$

which by induction holds for summing any number of independent graph-valued random variables.

To demonstrate how to verify this property, we consider the specific cases of $\kappa_{1/}$, $\kappa_{2\wedge}$, and $\kappa_{3\Delta}$. Clearly, for the first moment:

$$\mu_{1/}(\mathcal{G}_1 + \mathcal{G}_2) = \mu_{1/}(\mathcal{G}_1) + \mu_{1/}(\mathcal{G}_2), \tag{3.4}$$

as edge weights simply sum and the normalization remains the same.

For $\mu_{2\wedge}$, one must consider the 2^2 ways in which a wedge could be formed: both edges from $\mathcal{G}_1$, giving $\mu_{2\wedge}(\mathcal{G}_1)$; both edges from $\mathcal{G}_2$, giving $\mu_{2\wedge}(\mathcal{G}_2)$; a "left" edge from $\mathcal{G}_1$ and a "right" edge from $\mathcal{G}_2$, giving $\mu_{1/}(\mathcal{G}_1)\mu_{1/}(\mathcal{G}_2)$; and a "left" edge from $\mathcal{G}_2$ and a "right" edge from $\mathcal{G}_1$, giving $\mu_{1/}(\mathcal{G}_2)\mu_{1/}(\mathcal{G}_1)$. Thus,

$$\mu_{2\wedge}(\mathcal{G}_1 + \mathcal{G}_2) = \mu_{2\wedge}(\mathcal{G}_1) + \mu_{2\wedge}(\mathcal{G}_2) + 2\mu_{1/}(\mathcal{G}_1)\mu_{1/}(\mathcal{G}_2) \tag{3.5}$$

Substituting (3.4) and (3.5) into the expression for $\kappa_{2\wedge}$, we have

$$\begin{aligned}
\kappa_{2\wedge}(\mathcal{G}_1 + \mathcal{G}_2) &= \mu_{2\wedge}(\mathcal{G}_1 + \mathcal{G}_2) - \mu_{1/}(\mathcal{G}_1 + \mathcal{G}_2)^2 \\
&= \big(\mu_{2\wedge}(\mathcal{G}_1) + \mu_{2\wedge}(\mathcal{G}_2) + 2\mu_{1/}(\mathcal{G}_1)\mu_{1/}(\mathcal{G}_2)\big) - \big(\mu_{1/}(\mathcal{G}_1) + \mu_{1/}(\mathcal{G}_2)\big)^2 \\
&= \mu_{2\wedge}(\mathcal{G}_1) + \mu_{2\wedge}(\mathcal{G}_2) - \mu_{1/}^2(\mathcal{G}_1) - \mu_{1/}^2(\mathcal{G}_2) \\
&= \kappa_{2\wedge}(\mathcal{G}_1) + \kappa_{2\wedge}(\mathcal{G}_2),
\end{aligned}$$

as desired.

Likewise, for $\mu_{3\Delta}$, one must again consider the 2^3 ways in which a triangle could be formed: all edges from $\mathcal{G}_1$, giving $\mu_{3\Delta}(\mathcal{G}_1)$; all edges from $\mathcal{G}_2$, giving $\mu_{3\Delta}(\mathcal{G}_2)$; a wedge from $\mathcal{G}_1$ and an edge from $\mathcal{G}_2$ (occurring for three configurations), giving $3\mu_{2\wedge}(\mathcal{G}_1)\mu_{1/}(\mathcal{G}_2)$; and a wedge from $\mathcal{G}_2$ and an edge from $\mathcal{G}_1$ (again occurring for three configurations), giving $3\mu_{2\wedge}(\mathcal{G}_2)\mu_{1/}(\mathcal{G}_1)$. Thus,

$$\mu_{3\Delta}(\mathcal{G}_1 + \mathcal{G}_2) = \mu_{3\Delta}(\mathcal{G}_1) + \mu_{3\Delta}(\mathcal{G}_2) + 3\mu_{2\wedge}(\mathcal{G}_1)\mu_{1/}(\mathcal{G}_2) + 3\mu_{2\wedge}(\mathcal{G}_2)\mu_{1/}(\mathcal{G}_1). \tag{3.6}$$

Substituting (3.4), (3.5) and (3.6) into the expression for $\kappa_{3\Delta}$, we again find that

$$\kappa_{3\Delta}(\mathcal{G}_1 + \mathcal{G}_2) = \kappa_{3\Delta}(\mathcal{G}_1) + \kappa_{3\Delta}(\mathcal{G}_2),$$

as desired.

3.10.5 Unbiased graph cumulants

In this section, we discuss the desired properties of unbiased estimators $\hat{\kappa}_{rg}$ of graph cumulants, focusing our discussion on simple graphs. We describe the procedure for deriving these $\hat{\kappa}_{rg}$, using the two second-order cumulants as an example. We also provide the resulting expressions up to third order.

Given a single network observation G, we desire estimators $\hat{\kappa}_{rg}(G)$ for the graph cumulants of the *distribution* $\mathcal{G}$ that generated G. More specifically, suppose one generates a maximum entropy distribution with graph cumulants equal to these $\hat{\kappa}_{rg}(G)$ up to some choice of order r'. Consider sampling single networks from this distribution and computing their $\hat{\kappa}_{rg}$. We require that, for $r \leq r'$, the expectations $\langle \hat{\kappa}_{rg} \rangle$ be equal to the cumulants κ_{rg} of the distribution itself.

In order to derive these $\hat{\kappa}_{rg}$, consider a large graph G_N with N nodes (the "population" network), where $N \gg n$. Randomly select a set of n nodes from G_N, and observe the induced subgraph G (the "sample" network). In the spirit of k-statistics [84], we seek expressions that are invariant under this subsampling (ie, $\langle \hat{\kappa}_{rg}(G) \rangle = \hat{\kappa}_{rg}(G_N)$), and have the appropriate limit (ie, $\hat{\kappa}_{rg}(G_N) \to \kappa_{rg}(G_N)$ as $N \to \infty$). The resulting infinite graph G_∞ is essentially equivalent to the distribution $\mathcal{G}$ over graphs on n nodes, ie, $\kappa_{rg}(G_\infty) = \kappa_{rg}(\mathcal{G})$.

We now describe our procedure applied to the two second-order graph cumulants. We consider expressions of the form

$$\hat{\kappa}_{2g} = A(n)\mu_{2\wedge} + B(n)\mu_{1/}^2 + C(n)\mu_{1/} \tag{3.7}$$

and determine how the functions $A(n)$, $B(n)$, and $C(n)$ should change when removing a single random node. This provides a recursion relation, and determines these functions up

to a few constants, which are then chosen so as to agree with κ_{rg} as $n \to \infty$. Note that it is not necessary to include a term with $\mu_{2/\!/}$ in this expression, as it can be expressed as a function of the other terms (see Section 3.10.1).

We now determine the expectation of each of the terms ($\mu_{2\wedge}$, $\mu_{1/}^2$, and $\mu_{1/}$) when removing a single random node from G. The easiest term is $\mu_{1/}$. Let $c_{/}$ be the counts of edges in the original graph on n nodes (as usual), d_i be the degree of node i, and $c'_{/}$ as the counts in the graph with this single node removed. Clearly, $c'_{/} = c_{/} - d_i$, so $\langle c'_{/} \rangle = c_{/} - \langle d_i \rangle$. As $\langle d_i \rangle = \frac{2c_{/}}{n}$, we obtain $\langle c'_{/} \rangle = (1 - \frac{2}{n})c_{/}$. Dividing by the edge counts in the corresponding complete graph, we obtain $\langle \mu'_{1/} \rangle = \frac{\binom{n}{2}}{\binom{n-1}{2}}(1 - \frac{2}{n})\mu_{1/} = \mu_{1/}$. In fact, this is general: all graph moments are preserved in expectation under subsampling of the nodes,

$$\langle \mu'_{rg} \rangle = \mu_{rg}. \tag{3.8}$$

Products of moments, however, are not necessarily preserved in expectation. Indeed, $\langle c'^2_{/} \rangle = \langle (c_{/} - d_i)^2 \rangle = c^2_{/} - 2c_{/}\langle d_i \rangle + \langle d_i^2 \rangle$. Fortunately, $\langle d_i^2 \rangle$ is easily expressed in terms of the wedge moment:

$$\begin{aligned} c_{\wedge} &= \sum_i \binom{d_i}{2} \\ &= \frac{n}{2}\left(\langle d_i^2 \rangle - \langle d_i \rangle\right), \end{aligned}$$

thus, $\langle d_i^2 \rangle = \frac{2}{n}c_{\wedge} + \langle d_i \rangle$ and $\langle c'^2_{/} \rangle = \frac{2}{n}c_{\wedge} + (1 - \frac{4}{n})c^2_{/} + \frac{2}{n}c_{/}$. Dividing by the appropriate counts in the corresponding complete graph yields

$$\langle \mu'^2_{1/} \rangle = \frac{4}{(n-1)(n-2)}\mu_{2\wedge} + \frac{n(n-4)}{(n-2)^2}\mu^2_{1/} + \frac{4}{(n-1)(n-2)^2}\mu_{1/}.$$

Therefore, the resulting recursion relations for $A(n)$, $B(n)$, and $C(n)$ are:

$$A(n) = A(n-1) + \frac{4}{(n-1)(n-2)} B(n-1) \tag{3.9}$$

$$B(n) = \frac{n(n-4)}{(n-2)^2} B(n-1) \tag{3.10}$$

$$C(n) = C(n-1) + \frac{4}{(n-1)(n-2)^2} B(n-1). \tag{3.11}$$

The solution to equation (3.10) is

$$B(n) = b \frac{n(n-1)}{(n-2)(n-3)}. \tag{3.12}$$

Substituting equation (3.12) into equations (3.9) and (3.11) yields

$$A(n) = b \frac{4(n-4)}{(n-3)} + a, \tag{3.13}$$

$$C(n) = b \frac{(n-1)(n-4)}{(n-2)(n-3)} + c. \tag{3.14}$$

To obtain $\hat{\kappa}_{2\wedge}$, we require that the $n \to \infty$ limit gives the values corresponding to $\kappa_{2\wedge}$, viz,

$$A_\wedge \to 1, \quad B_\wedge \to -1, \quad C_\wedge \to 0.$$

Thus, the relevant constants are $a_\wedge = 5$, $b_\wedge = -1$, $c_\wedge = 1$, and $\hat{\kappa}_{2\wedge}$ is given by

$$\hat{\kappa}_{2\wedge} = \frac{n+1}{n-3} \mu_{2\wedge} - \frac{n(n-1)}{(n-2)(n-3)} \mu_{1\prime}^2 + \frac{2}{(n-2)(n-3)} \mu_{1\prime}. \tag{3.15}$$

Recalling that $c_\wedge + c_{/\!/} = \binom{c_\prime}{2}$, we repeat the same procedure to obtain $\hat{\kappa}_{2/\!/}$. This case is a bit less straightforward (although the result, $\hat{\kappa}_{2/\!/} = 0$, is comparatively simpler). As $n \to \infty$,

the number of pairs of edges in the network grows as $\binom{c_{/}}{2} = \frac{n^4}{8}\mu_{1/}^2$, while the number of wedges is bounded by $c_{\wedge} \leq \frac{n^3}{2}$. Therefore, the number of pairs of edges that do not share any node is $c_{/\!/} = \frac{n^4}{8}\mu_{1/}^2(1 - \mathcal{O}(\frac{1}{n}))$. Thus, $\mu_{2/\!/} = \mu_{1/}^2 + \mathcal{O}(\frac{1}{n})$, and $\kappa_{2/\!/} = \mathcal{O}(\frac{1}{n}) \to 0$. Hence, the (unique) coherent solution is $a_{/\!/} = b_{/\!/} = c_{/\!/} = 0$, thus $\hat{\kappa}_{2/\!/} = 0$. Indeed, the same result holds for graphons (the natural limit of a sequence of dense graphs of increasing size [32, 161]), where it can be shown [160] that $\mu_{2/\!/} \xrightarrow[n\to\infty]{} \mu_{1/}^2$, and therefore $\kappa_{2/\!/} \xrightarrow[n\to\infty]{} 0$. More generally, $\hat{\kappa}_{rg} = 0$ for all disconnected subgraphs g.

To summarize, as with real-value random variable, the unbiased estimator of the first-order cumulant is simply the first-order cumulant itself:

$$\hat{\kappa}_{1/} = \mu_{1/}. \tag{3.16}$$

The unbiased estimators of the second-order cumulants are as follows:

$$\hat{\kappa}_{2\wedge} = \frac{n+1}{n-3}\mu_{2\wedge} - \frac{n(n-1)}{(n-2)(n-3)}\mu_{1/}^2 + \frac{2}{(n-2)(n-3)}\mu_{1/}, \tag{3.17}$$

$$\hat{\kappa}_{2/\!/} = 0. \tag{3.18}$$

And the unbiased estimators of the third-order cumulants are:

$$\hat{\kappa}_{3\triangle} = \alpha_{1/}\mu_{1/} + \alpha_{1/1/}\mu_{1/}^2 + \alpha_{2\wedge}\mu_{2\wedge} + \alpha_{1/1/1/}\mu_{1/}^3 + \alpha_{2\wedge 1/}\mu_{2\wedge}\mu_{1/}$$
$$+ (1+\alpha_{3\triangle})\mu_{3\triangle} + \alpha_{3\curlywedge}\mu_{3\curlywedge} + \alpha_{3\sqcap}\mu_{3\sqcap}, \tag{3.19}$$

$$\hat{\kappa}_{3\curlywedge} = \alpha_{1/}\mu_{1/} + \alpha_{1/1/}\mu_{1/}^2 + \alpha_{2\wedge}\mu_{2\wedge} + \alpha_{1/1/1/}\mu_{1/}^3 + \alpha_{2\wedge 1/}\mu_{2\wedge}\mu_{1/}$$
$$+ \alpha_{3\triangle}\mu_{3\triangle} + (1+\alpha_{3\curlywedge})\mu_{3\curlywedge} + \alpha_{3\sqcap}\mu_{3\sqcap}, \tag{3.20}$$

$$\hat{\kappa}_{3\sqcap} = \alpha_{1/}\mu_{1/} + \alpha_{1/1/}\mu_{1/}^2 + \alpha_{2\wedge}\mu_{2\wedge} + \alpha_{1/1/1/}\mu_{1/}^3 + \alpha_{2\wedge 1/}\mu_{2\wedge}\mu_{1/}$$
$$+ \alpha_{3\triangle}\mu_{3\triangle} + \alpha_{3\curlywedge}\mu_{3\curlywedge} + (1+\alpha_{3\sqcap})\mu_{3\sqcap}, \tag{3.21}$$

$$\hat{\kappa}_{3/\!\!\wedge} = 0, \tag{3.22}$$

$$\hat{\kappa}_{3/\!\!/} = 0, \tag{3.23}$$

where

$$\alpha_{1/} = \frac{16}{(n-2)(n-3)(n-4)(n-5)}, \tag{3.24}$$

$$\alpha_{1/1/} = \frac{-12n(n-1)}{(n-2)(n-3)(n-4)(n-5)}, \tag{3.25}$$

$$\alpha_{2\wedge} = \frac{12(n+3)}{(n-3)(n-4)(n-5)}, \tag{3.26}$$

$$\alpha_{1/1/1/} = \frac{2n^2(n-1)^2}{(n-2)(n-3)(n-4)(n-5)}, \tag{3.27}$$

$$\alpha_{2\wedge 1/} = \frac{-3(n+3)n(n-1)}{(n-3)(n-4)(n-5)}, \tag{3.28}$$

$$\alpha_{3\triangle} = \frac{6n+2}{(n-3)(n-4)(n-5)}, \tag{3.29}$$

$$\alpha_{3\curlywedge} = \frac{6n+2}{(n-4)(n-5)}, \tag{3.30}$$

$$\alpha_{3\sqcap} = \frac{12(n-1)}{(n-4)(n-5)}. \tag{3.31}$$

In short, as the unbiased cumulants are polynomials in the moments, we consider the expected changes in the relevant graph moments of a network G and their products when removing a single random node from it. By equating these expressions, we obtain a recursion relation for the coefficients of the polynomials. We solve these recursion relations using Mathematica, fixing the undetermined constants such that the resulting expression agrees in the $n \to \infty$ limit.

3.10.6 Statistical inference without constructing an explicit null model

One can apply a similar procedure to the square of the unbiased cumulants. This results in a recursion relation for their variance (see Appendix A.2 for details).

The unbiased graph cumulants and their variance can be used to test whether the observed network has a significant proclivity for a substructure g with r edges. In particular, measure the moments of the observed network up to order $2r$, and use these to compute the unbiased cumulant $\hat{\kappa}_{rg}$, as well as its variance $\text{Var}(\hat{\kappa}_{rg})$. Suppose that $\hat{\kappa}_{rg} > 0$ (potentially indicating a proclivity for substructure g). To determine the statistical significance of this assessment, compute the (squared) Z-score associated with the null hypothesis that $\hat{\kappa}_{rg} < 0$: $Z_{rg}^2 = \frac{\hat{\kappa}_{rg}^2}{\text{Var}(\hat{\kappa}_{rg})}$. Likewise, if $\hat{\kappa}_{rg} < 0$ (potentially indicating an aversiveness for substructure g), the squared Z-score expression is the same. If $Z_{rg}^2 \lesssim 1$, one can conclude that the observed network is ambivalent to substructure g. Conversely, if $Z_{rg}^2 \gtrsim 1$, one can conclude that the observed network has a proclivity (or aversiveness) for substructure g, depending on the sign of $\hat{\kappa}_{rg}$. This procedure can also be used to measure the similarity between two networks, by applying a two-sample t-test to the pair of unbiased cumulants associated to each particular substructure.

The standard conversion from a Z-score to a p-value tacitly assumes normality, which does not hold in general. However, our procedure to obtain unbiased cumulants and their

variance can also be used to obtain higher-order cumulants of the distributions of unbiased cumulants (although the derivations become incredibly tedious). Future work on these unbiased cumulants (and the cumulants of their distributions) could lead to principled statistical tests of the proclivity (or aversiveness) for arbitrary substructures. Such statistical tests are advantageous, as they do not require one to explicitly construct a null model and sample from it (a procedure that is in general quite computationally expensive).

3.10.7 Fitting ERGMs using unbiased graph cumulants

We now describe how to infer a model from our proposed hierarchical family of ERGMs using a *single* observed network G, specializing our discussion to simple graphs. In particular, we consider ERGMs that have constraints on their expected counts of all subgraphs (or, equivalently, their associated moments) up to some order r'. These distributions have the following form:

$$p(G) = \frac{1}{Z}\text{ER}_{n,1/2}(G)\ \exp\left(\sum_g \beta_g c_g(G)\right), \tag{3.32}$$

$$Z = \sum_{G'\in\Omega}\left[\text{ER}_{n,1/2}(G')\ \exp\left(\sum_g \beta_g c_g(G')\right)\right], \tag{3.33}$$

where Z is the normalization constant (or partition function); $\text{ER}_{n,p}$ is an Erdős–Rényi model[5]; β_g are the parameters (Lagrange multipliers) associated with subgraph g; $c_g(G)$ are the counts of subgraph g in G; and Ω is the space of all simple graphs on n nodes.

While an ERGM is specified by the desired expectation of statistics of interest (in this case, subgraph counts/moments), the parameters needed to compute the distribution are the β_g, and in general must be solved for numerically. Moreover, the partition function

[5]The distribution over simple graphs with n nodes, in which the each edge occurs independently with probability p. In particular, $\text{ER}_{n,1/2}$ is the uniform (ie, maximum entropy) distribution over all $n \times n$ binary symmetric traceless matrices, and thus proportional to $\frac{n!}{\text{Aut}(G)}$.

Z often cannot be exactly computed, as the space of unique graphs over n nodes grows super-exponentially with n (eg, on 10 nodes there are 12005168 [114]). However, there exists a large body of literature on MCMC and variational techniques to efficiently approximate this Z [141, 208].

Aside from the prescription to use *all* subgraph counts up to order r' (including disconnected subgraphs), our proposed hierarchical family of ERGMs includes the following three-step recipe:

1. Use the observed network to compute the unbiased estimators $\hat{\kappa}_{rg}$ of the graph cumulants (up to some order r') of the underlying ERGM distribution (the expressions are provided in Section 3.10.5 and included in our code).

2. Convert these unbiased cumulants into the target moments of the distribution $\hat{\mu}_{rg}$ using the standard formulas to convert cumulants to moments, eg,

$$\begin{aligned}
\hat{\mu}_{2\wedge} &= \hat{\kappa}_{2\wedge} + \hat{\kappa}_{1/}^2, \\
\hat{\mu}_{3\Delta} &= \hat{\kappa}_{3\Delta} + 3\hat{\kappa}_{2\wedge}\hat{\kappa}_{1/} + \hat{\kappa}_{1/}^3, \\
\hat{\mu}_{4\times} &= \hat{\kappa}_{4\times} + 4\hat{\kappa}_{3\curlywedge}\hat{\kappa}_{1/} + 3\hat{\kappa}_{2\wedge}^2 + 6\hat{\kappa}_{2\wedge}\hat{\kappa}_{1/}^2 + \hat{\kappa}_{1/}^4.
\end{aligned}$$

   The general form is given by:

$$\mu_{rg} = \sum_{\pi \in P_r} \prod_{b \in \pi} \kappa_{|b|\, g_b}, \tag{3.34}$$

   where P_r is the set of partitions of the r edges, π is one of those partitions, b is a subset of a partition, and g_b is the subgraph formed by the edges in this subset (the expressions are provided in Appendix A.3 and are also included in our code).

3. Fit an ERGM that has these unbiased moments $\hat{\mu}_{rg}$ in expectation.

The use of these unbiased cumulants is a notable departure from the typical choice of constraints for ERGMs, as we are now selecting a distribution with expected subgraph counts different than those of the observed network (and are in general, not realizable by any single network). Nevertheless, the ERGM distribution induced by this choice generates samples that are appropriately clustered about the observed network (see Section 3.10.8 for a detailed intuition).

Partially unbiasing a distribution

In some cases, it may not be appropriate to assume that the nodes of the observed network were sampled randomly from a very large underlying network (the "population"). This is particularly relevant when the observed network is small, or when it could feasibly represent a significant fraction of the system of interest. For these cases, we introduce an adjustable unbiasing parameter $\eta \in [0, 1]$, where 1 corresponds to the aforementioned "fully" unbiased case, and 0 corresponds to using the original moments μ_{rg} of the observed network ("no unbiasing"). Essentially, instead of assuming that the underlying network is infinite, this parameter controls its size (N) relative to that of the observed network (n), defined as $\eta = 1 - \frac{n}{N}$.

The procedure is essentially the same as before, with a modified second step. In particular, one first computes the unbiased estimates of the cumulants using the moments μ of the observed network (step 1 of our procedure):

$$\mu \xrightarrow[\text{eq. (3.16)–(3.23) with } n = n]{} \hat{\kappa} \tag{3.35}$$

Then one uses these $\hat{\kappa}$ in the inverse expressions, now using $n = N$:

$$\hat{\kappa} \xrightarrow[\text{eq. (3.37)–(3.39) with } n = N]{} \hat{\mu} \tag{3.36}$$

where these inverse expressions (up to second order) are

$$\hat{\mu}_{1/} = \hat{\kappa}_{1/}, \tag{3.37}$$

$$\hat{\mu}_{2\wedge} = \frac{n-3}{n+1}\hat{\kappa}_{2\wedge} + \frac{n(n-1)}{(n+1)(n-2)}\hat{\kappa}_{1/}^2 - \frac{2}{(n+1)(n-2)}\hat{\kappa}_{1/}, \tag{3.38}$$

$$\hat{\mu}_{2/\!/} = \frac{1}{\#_{/\!/}}\left(\binom{\#_{/}\hat{\mu}_{1/}}{2} - \#_{2\wedge}\hat{\mu}_{2\wedge}\right), \tag{3.39}$$

where $\#_g$ is the count of subgraph g in the complete network with n nodes (eg, the denominators of equations (A.10)–(A.21) in Appendix A.3).

Finally, one fits an ERGM that has expected moments $\hat{\mu}_{rg}$ (step 3 of our procedure).

The limit $\eta \to 1$ corresponds to $N \to \infty$, and agrees with the standard expressions provided in Appendix A.3. The limit $\eta \to 0$ corresponds to $N = n$ and generate an ERGM that is more narrowly concentrated on the observed network G (eg, if one fits an second-order ERGM, it will give zero probability to networks that do not have the same number of edges as G).

3.10.8 A geometric understanding of the degeneracy problem of ERGMs

Degeneracy is essentially the appearance of large-scale multimodality in the ERGM distribution; despite the fact that the expectation of subgraph counts of samples from this distribution is equal to those of the observed network, typical samples from it have counts vastly different from this average.

Essentially, this arises due to the shape of the base distribution (ie, $\text{ER}_{n,1/2}$) as a function of the statistics with constrained expected values [109] (here the subgraph counts, or equivalently, their graph moments). Recall that the ERGM distribution is given by equation (3.32), ie,

$$p(G) \propto \text{ER}_{n,1/2}(G)\ \exp\left(\sum_g \beta_g c_g(G)\right). \tag{3.40}$$

Projecting this distribution to the space of counts (ie, summing the probability of all networks with the same relevant counts), and taking its logarithm yields:

$$\ln p(\vec{c}) = \ln\big(\text{ER}_{n,1/2}(\vec{c})\big) + \vec{\beta}\cdot\vec{c}, \tag{3.41}$$

where $\vec{c}$ is the vector of relevant subgraphs counts (ie, those whose expectations are constrained), $\vec{\beta}$ is the vector of their associated parameters, and we have dropped the constant term.

Hence, to understand the behavior of $p(G)$, it is geometrically instructive to look at the shape of $\ln\big(\text{ER}_{n,1/2}\big)$ as a function of $\vec{c}$ (see Figures 3.8 and 3.9).

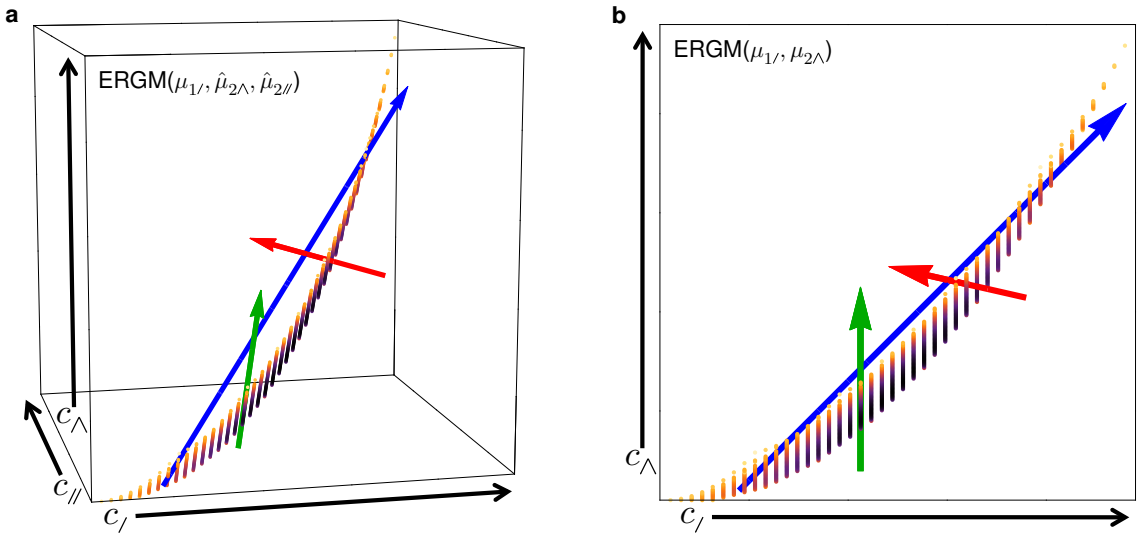


Figure 3.8: **Using all subgraph counts up to second order allows for greater control over the resulting distribution**. **a)** When represented as a function of $(\mu_/, \mu_\wedge, \mu_{//})$, the density of $\text{ER}_{10,1/2}$ occupies a 2D submanifold with extrinsic curvature. Three orthogonal directions can independently control the edge counts (blue arrow), the wedge counts (green arrow), and the spread in the edge counts (red arrow). **b)** In contrast, when representing the distribution as a function of $(\mu_/, \mu_\wedge)$, one can only independently control two of these quantities. When the counts of edges and wedges are the constraints, as in $\text{ERGM}(\mu_/, \mu_\wedge)$, the spread in the edge counts cannot be controlled, and can result in "degenerate" distributions with significant bimodality.

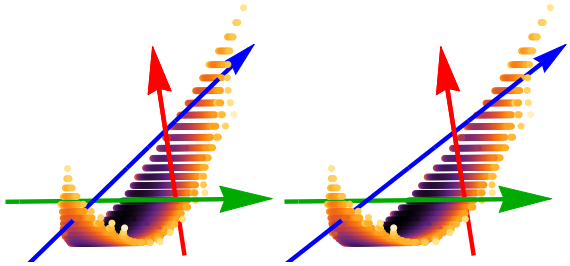


Figure 3.9: **A stereographic image of the (logged) base distribution in 3D space** $(\mu_{/}, \mu_{\wedge}, \mu_{/\!/})$. Displayed is the log of the base distribution $\ln(\mathrm{ER}_{10,1/2}(\mu_{/}, \mu_{\wedge}, \mu_{/\!/}))$ embedded in the space of all subgraph counts up to second order, ie, edges, wedges, and two edges that do not share any node. To fully appreciate the stereographic effect, print the image using either US letter size or standard A4 paper. Begin with the paper close to your eyes, and allow your left eye to focus on the left image, while your right eye focuses on the right image. Slowly move the paper away from your eyes, while maintaining focus on the middle of the "three" images, until it is approximately 20–40cm away. The "middle" image should be more apparent than those on either side, and its upper right should appear farther than its bottom left. We assume an average pupillary distance of 63 mm, and remark that the parallax has been decreased by a factor of $\sim$2 to make it easier to overlay the two images. Dark indicates high probability, and light indicates low. See also the animation here.

To provide intuition about this phenomenon and our proposed solution, here we give attention to the commonly used (and easily visualizable) 2D model, denoted by ERGM$(\mu_{/}, \mu_{\wedge})$, which constrains the expected counts of edges and wedges in the distribution to be equal to those in the observed network. For comparison, we consider our second-order model, which additionally constrains the expected counts of pairs of edges that do not share any node. We will discuss both the case when the expectation of these three subgraph counts are constrained to be those of the observed network, denoted by ERGM$(\mu_{/}, \mu_{\wedge}, \mu_{/\!/})$, and when they are constrained to be the equal to their unbiased values, denoted by ERGM$(\hat{\mu}_{/}, \hat{\mu}_{\wedge}, \hat{\mu}_{/\!/})$.

Consider the convex hull formed by all points representing realizable subgraph counts for a single network in their respective 2D (for ERGM$(\mu_{/}, \mu_{\wedge})$) or 3D (for ERGM$(\mu_{/}, \mu_{\wedge}, \mu_{/\!/})$

or ERGM($\hat{\mu}_{/}, \hat{\mu}_{\wedge}, \hat{\mu}_{/\!/}$)) spaces (Figure 3.8). Any choice of target expected counts that is within the convex hull may be realized by some ERGM in the corresponding class. However, some choices of target values for the expected counts *require* degenerate distributions (even without maximizing the entropy). For example, consider the constraints $\langle c_{/} \rangle = \frac{1}{2}\#_{/}$, $\langle c_{\wedge} \rangle = \frac{1}{2}\#_{\wedge}$. Indeed, the *only* distribution that satisfies these constraints is an equal mixture of the empty and complete networks (in a sense, the most "degenerate" distribution possible). This motivates a restriction on the choices of target values, a natural choice being that the target expected counts are those of the actual observed network!

Even with this restriction, ERGM($\mu_{/}, \mu_{\wedge}$) does not always concentrate around its target values. This happens in particular when one chooses a network that lies along a concave boundary of the support of $\ln(\text{ER}_{n,1/2}(c_{/}, c_{\wedge}))$. This is due to the effect of the linear term in equation (3.41), which must be chosen to "push" the distribution in Figure 3.8b in a direction perpendicular to the blue arrow (towards the upper left). While this indeed increases the number of wedges, it is inevitably coupled with a motion of the probability density toward the "tips" of the distribution. Thus, the number of wedges and the *spread* in the number of edges cannot be independently controlled.

In contrast, ERGM($\mu_{/}, \mu_{\wedge}, \mu_{/\!/}$) has an additional degree of freedom for $\vec{\beta}$. Thus, it is able to independently control the expected number of edges and wedges as well as the spread in the number of edges. However, if one chooses the triplet of target expected counts to be those from the observed network, the resulting distribution necessarily concentrates to a single number of edges. This is essentially due to the fact that all such triplets lie on the boundary of the convex hull. For a fixed number of edges, the relationship between the second-order moments is linear: $\#_{\wedge}\mu_{2\wedge} + \#_{/\!/}\mu_{2/\!/} = C$. Additionally, the relationship between the counts of edges and this invariant sum C has a curvature that does not change sign: $C = \frac{1}{2}(\#_{/}^2\mu_{1/}^2 - \#_{/}\mu_{1/})$. Thus, for any observed network, all distributions with expected counts that match this network must only have support along this linear direction (ie, the set

of triplets with the same invariant sum C), and therefore have the same number of edges as the observed network. Thus, we have "solved" the degeneracy problem by essentially specifying the number of edges. However, given the many sources of noise in real-world network data, this solution is likely unreasonable for many applications.

In order to spread the number of edges, we would like our triplet of target expected counts to be slightly in the interior of the convex hull, in the direction of the red arrow in Figures 3.8 and 3.9. This may seem to be an unusual choice, as such target counts are not realizable by any single network. However, this is indeed quite natural: even the $\mathrm{ER}_{n,p}$ distributions have triplets of expected counts that lie in this direction. The unbiased graph cumulants derived in Section 3.10.5 provide a natural and consistent prescription for obtaining such modified triplets of target expected counts.

We remark that not all observed networks result in realizable distributions (those with modified triplets of target expected counts that lie outside the convex hull). However, this occurs for networks that are unlikely to be observed when subsampling nodes from a large network, such as regular graphs (ie, all nodes have exactly the same number of edges) or nearly-regular graphs. From a pragmatic perspective, this is unlikely to be a problem as real networks tend to not have such properties. In particular, as shown in Figure 3.10 the fraction of networks that result in such unrealizable triplets of expected counts decreases as the number of nodes increases. Moreover, if one does observe a network for which this is the case, the issue is often alleviated by use of an intermediate choice of the unbiasing parameter η (see Section 3.10.7).

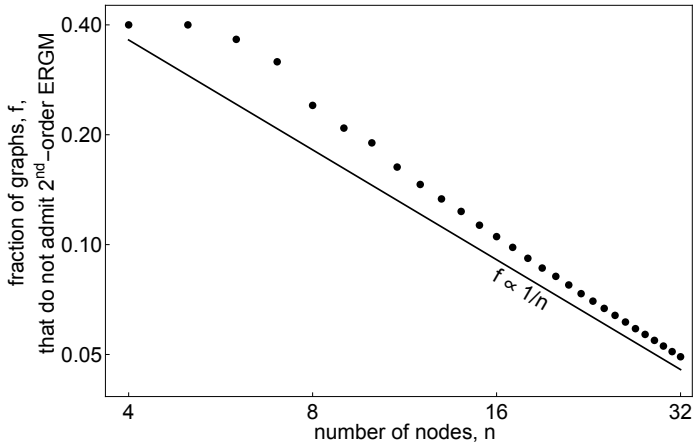


Figure 3.10: **The fraction of "problematic" networks vanishes as $n \to \infty$.** We approximate the space of possible triplets of subgraph counts $(c_/, c_\wedge, c_{/\!/})$ for networks with 4 to 32 nodes. For a given number of edges $e \in [0, \binom{n}{2}]$, we evaluate $\min(c_\wedge)$ and $\max(c_\wedge)$ over all networks with n nodes and e edges. We then include all integer-valued triplets $(c_/, c_\wedge, \binom{c_/}{2} - c_\wedge)$ within this range. For each triplet $(c_/, c_\wedge, c_{/\!/})$, we compute its fully unbiased ($\eta = 1$) moments, and convert to unbiased counts. We compute the convex hull of the triplets of original counts, and report the fraction of triplets of unbiased counts that lie outside of this region. Such triplets cannot be the expected moments of any distribution over networks on this number of nodes. However, their corresponding networks are typically extremal in the sense that some subgraph counts (such as $c_\wedge$) are maximized or minimized given some lower-order conditions (such as fixed $c_/$). The prevalence of such networks decreases as the number of nodes increases, and is unlikely to occur in practice. However, if one encounters such a situation, it can be remedied by reducing the unbiasing parameter η (such as, in Figure 3.3).

3.10.9 Clustering analysis

As our formalism is able to incorporate additional information, such as edge weights, directed edges, or node attributes, it provides a principled notion of clustering for any combination of such cases, an active domain of research that has arguably not reached a consensus. For instance, while several measures of clustering in weighted networks have been proposed [177, 16, 201, 8] (using, eg, maximum/minimum/averaging procedures), our framework provides a self-consistent prescription: count subgraphs with a weight equal to

the product of the edges that comprise them [160]. Likewise, for directed networks, the two third-order scaled triangle cumulants provide two measures for clustering: one with cyclic orientation and one with transitive.

We illustrate the flexibility of our framework using simple graphs (Figure 3.4) and bipartite networks (Figure 3.11). For bipartite networks, the extensions are somewhat less straightforward, as now triangles are excluded. Indeed, several notions of clustering have been proposed [181, 237, 42]. In general, these involve measuring the appearance of 4-cycles or 6-cycles in the network, similarly compared to the number of incomplete cycles. In Figure 3.11, we compare our scaled graph cumulant of the 4-cycle subgraph $\tilde{\kappa}_{4\square}$ with the clustering coefficient for the 4-cycle proposed by [193], expressed in our framework as $C_{\square} = \mu_{4\square}/\mu_{3\sqcap}$ (analogous to the standard clustering coefficient). We show that our measure is again more directly sensitive to the propensity for clustering, without the unnecessary scaling with the overall edge density.

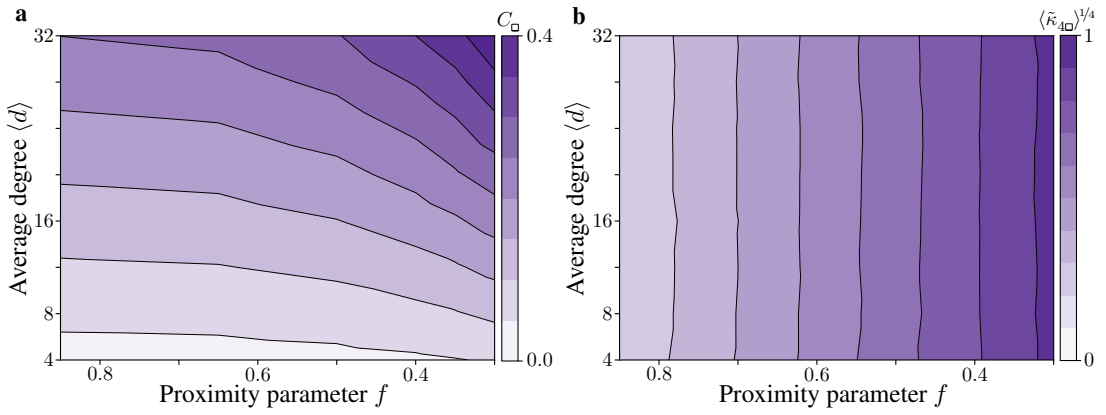


Figure 3.11: **A principled measure of clustering in bipartite networks.** We simulated a bipartite geometric graph model with two parameters, f and $\langle d \rangle$, which determine the propensity for clustering (horizontal axis) and the edge density (vertical axis), respectively. The model first divides the nodes into two groups and randomly places them on the unit sphere. Each node may only connect to nodes from the other group, and only when they are within a certain radius, such that the surface area it contains is a fraction f of the entire unit sphere. Among these possible connections, a specified number of them are randomly chosen, so as to match the desired average degree $\langle d \rangle$. **a)** Global bipartite clustering coefficient $C_\square$ [193]. **b)** Scaled square cumulant $\tilde{\kappa}_{4\square}$. For each set of parameters, we computed $C_\square$ and $\tilde{\kappa}_{4\square}$ for each instance. We display $C_\square$ (left) and the (signed) fourth root of the average value $\tilde{\kappa}_{4\square}$ (right). Clustering is promoted when the proximity parameter f is small (right side of the plots). While both clustering measures increase for decreasing f, the bipartite clustering coefficient also notably increases with average degree, whereas the scaled square cumulant is insensitive to such changes in edge density. In addition, as f approaches 1 (the fully random bipartite limit), $\tilde{\kappa}_{4\square}$ approaches 0, indicating no propensity for clustering.

Chapter 4

Probing Global Structure: Spectral Graph Reduction

Do not miss the forest for the trees.

—— Conventional Wisdom

4.1 Prologue

In the previous chapter, we characterized network structure from a local perspective, building a hierarchical description based on the correlations between an increasing number of connections. In this chapter, we attack this question from a dual perspective [160]. Essentially, we seek reduced graphs in which the dynamics of diffusion processes are similar to that of the original graph, thereby preserving its global structure. In what follows, we formalize this goal and explain the work resulting from this project.

This work was done with Lee M. Gunderson and is published as a co-authored paper in Advances in Neural Information Processing Systems (NeurIPS), 2019, under the title: "A Unifying Framework for Spectrum-Preserving Graph Sparsification and Coarsening"[38].

4.2 Abstract

How might one "reduce" a graph? That is, generate a smaller graph that preserves the global structure at the expense of discarding local details? There has been extensive work on both graph sparsification (removing edges) and graph coarsening (merging nodes, often by edge contraction); however, these operations are currently treated separately. Interestingly, for a planar graph, edge deletion corresponds to edge contraction in its planar dual (and more generally, for a graphical matroid and its dual). Moreover, with respect to the dynamics induced by the graph Laplacian (eg, diffusion), deletion and contraction are physical manifestations of two reciprocal limits: edge weights of 0 and ∞, respectively. In this work, we provide a unifying framework that captures both of these operations, allowing one to simultaneously sparsify and coarsen a graph while preserving its large-scale structure. The limit of infinite edge weight is rarely considered, as many classical notions of graph similarity diverge. However, its algebraic, geometric, and physical interpretations are reflected in the Laplacian pseudoinverse $\boldsymbol{L}^\dagger$, which remains finite in this limit. Motivated by this insight, we provide a probabilistic algorithm that reduces graphs while preserving $\boldsymbol{L}^\dagger$, using an unbiased procedure that minimizes its variance. We compare our algorithm with several existing sparsification and coarsening algorithms using real-world datasets, and demonstrate that it more accurately preserves the large-scale structure.

4.3 Motivation

Many complex structures and phenomena are naturally described as graphs (eg, brains, social networks, the internet, etc). Indeed, graph-structured data are becoming increasingly relevant to the field of machine learning [40, 41, 102]. These graphs are frequently massive, easily surpassing our working memory, and often the computer's relevant cache [18]. It

is therefore essential to obtain smaller approximate graphs to allow for more efficient computation.

Graphs are defined by a set of nodes V and a set of edges $E \subseteq V \times V$ between them, and are often represented as an adjacency matrix $\boldsymbol{A}$ with size $|V| \times |V|$ and density $\propto |E|$. Reducing either of these quantities is advantageous: graph "coarsening" focuses on the former, aggregating nodes while respecting the overall structure, and graph "sparsification" on the latter, preferentially retaining the important edges.

Spectral graph sparsification has revolutionized the field of numerical linear algebra and is used, eg, in algorithms for solving linear systems with symmetric diagonally dominant matrices in nearly-linear time [213, 62] (in contrast to the fastest known algorithm for solving general linear systems, taking $\mathcal{O}(n^\omega)$-time, where $\omega \approx 2.373$ is the matrix multiplication exponent [150]).

Graph coarsening appears in many computer science and machine learning applications, eg: as primitives for graph partitioning [198] and visualization algorithms[1] [101]; as layers in graph convolution networks [41, 207]; for dimensionality reduction and hierarchical representation of graph-structured data [147, 56]; and to speed up regularized least square problems on graphs [105], which arise in a variety of problems such as ranking [171] and distributed synchronization of clocks [209].

A variety of algorithms, with different objectives, have been proposed for both sparsification and coarsening. However, a frequently recurring theme is to consider the graph Laplacian $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$, where $\boldsymbol{D}$ is the diagonal matrix of node degrees. Indeed, it appears in a wide range of applications, eg: its spectral properties can be leveraged for graph clustering [80]; it can be used to efficiently solve min-cut/max-flow problems [58]; and for undirected, positively weighted graphs (the focus of this paper), it induces a natural

[1] For animated examples using our graph reduction algorithm, see the following link: youtube.com/playlist?list=PLmfiQcz2q6d3sZutLri4ZAIDLqM_4K1p-.

quadratic form, which can be used, eg, to smoothly interpolate functions over the nodes [146].

Work on spectral graph sparsification focuses on preserving the Laplacian quadratic form $\vec{x}^\top \boldsymbol{L}\vec{x}$, a popular measure of spectral similarity suggested by Spielman & Teng [213]. A key result in this field is that any dense graph can be sparsified to $\mathcal{O}(|V|\log|V|)$ edges in nearly linear time using a simple probabilistic algorithm [212]: start with an empty graph, include edges from the original graph with probability proportional to their effective resistance, and appropriately reweight those edges so as to preserve $\vec{x}^\top \boldsymbol{L}\vec{x}$ within a reasonable factor.

In contrast to the firm theoretical footing of spectral sparsification, work on graph coarsening has not reached a similar maturity; while a variety of spectral coarsening schemes have been recently proposed, algorithms frequently rely on heuristics, and there is arguably no consensus. Eg: Jin & Jaja [121] use k eigenvectors of the Laplacian as feature vectors to perform k-means clustering of the nodes; Purohit et al. [185] aim to minimize the change in the largest eigenvalue of the adjacency matrix; and Loukas & Vandergheynst [159] focuses on a "restricted" Laplacian quadratic form.

Although recent work has combined sparsification and coarsening [238], they used separate algorithmic primitives, essentially analyzing the serial composition of the above algorithms. The primary contribution of this work is to provide a unifying probabilistic framework that allows one to simultaneously sparsify and coarsen a graph while preserving its global structure by using a *single* cost function that preserves the Laplacian pseudoinverse $\boldsymbol{L}^\dagger$.

Corollary contributions include: **1)** Identifying the limit of infinite edge weight with edge contraction, highlighting how its algebraic, geometric, and physical interpretations are reflected in $\boldsymbol{L}^\dagger$, which remains finite in this limit (Section 4.4); **2)** Offering a way to quantitatively compare the effects of edge deletion and edge contraction (Section 4.4 and 4.5); **3)** Providing a probabilistic algorithm that reduces graphs while preserving $\boldsymbol{L}^\dagger$, using an

unbiased procedure that minimizes its variance (Sections 4.5 and 4.6); **4)** Proposing a more sensitive measure of spectral similarity of graphs, inspired by the Poincaré half-plane model of hyperbolic space (Sections 4.7.3 and 4.9.3); and **5)** Comparing our algorithm with several existing sparsification and coarsening algorithms using synthetic and real-world datasets, demonstrating that it more accurately preserves the large-scale structure (Sections 4.7 and 4.9.6).

4.4 Why the Laplacian pseudoinverse

Many computations over graphs involve solving $\boldsymbol{L}\vec{x} = \vec{b}$ for $\vec{x}$ [221]. Thus, the algebraically relevant operator is arguably the Laplacian pseudoinverse $\boldsymbol{L}^\dagger$. In fact, its connection with random walks has been used to derive useful measures of distances on graphs, such as the well-known effective resistance [51], and the recently proposed resistance perturbation distance [169]. Moreover, taking the pseudoinverse of $\boldsymbol{L}$ leaves its eigenvectors unchanged, but inverts the nontrivial eigenvalues. Thus, as the largest eigenpairs of $\boldsymbol{L}^\dagger$ are associated with global structure, preserving its action will preferentially maintain the overall "shape" of the graph (see Appendix B.1 for details). For instance, the Fielder vector [80] (associated with the "algebraic connectivity" of a graph) will be preferentially preserved. We now discuss in further detail why $\boldsymbol{L}^\dagger$ is well-suited for both graph sparsification and coarsening.

Attention is often restricted to undirected, positively weighted graphs [61]. These graphs have many convenient properties, eg, their Laplacians are positive semidefinite ($\vec{x}^\top \boldsymbol{L}\vec{x} \geq 0$) and have a well-understood kernel and cokernel ($\boldsymbol{L}\vec{1} = \vec{1}^\top \boldsymbol{L} = \vec{0}$). The edge weights are defined as a mapping $W\colon E \to \mathbb{R}_{>0}$. When the weights represent connection strength, it is generally understood that $w_e \to 0$ is equivalent to removing edge e. However, the closure of the positive reals has a reciprocal limit, namely $w_e \to +\infty$.

This limit is rarely considered, as many classical notions of graph similarity diverge. This includes the standard notion of spectral similarity, where $\widetilde{G}$ is a σ-spectral approximation of G if it preserves the Laplacian quadratic form $\vec{x}^{\mathsf{T}}\boldsymbol{L}_{G}\vec{x}$ to within a factor of σ for all vectors $\vec{x} \in \mathbb{R}^{|V_G|}$ [213]. Clearly, this limit yields a graph that does not approximate the original for any choice of σ: any $\vec{x}$ with different values for the two nodes joined by the edge with infinite weight now yields an infinite quadratic form. This suggests considering only vectors that have the same value for these two nodes, essentially contracting them into a single "supernode". Algebraically, this interpretation is reflected in $\boldsymbol{L}^{\dagger}$, which remains finite in this limit: the pair of rows (and columns) corresponding to the contracted nodes become identical (see Section 4.9.2).

Physically, consider the behavior of the heat equation $\partial_t\vec{x} + \boldsymbol{L}\vec{x} = \vec{0}$: as $w_e \to +\infty$, the values on the two nodes immediately equilibrate between themselves, and remain tethered for the rest of the evolution.[2] Geometrically, the reciprocal limits of $w_e \to 0$ and $w_e \to +\infty$ have dual interpretations: consider a planar graph and its planar dual; edge deletion in one graph corresponds to contraction in the other, and vice versa. This naturally extends to nonplanar graphs via their graphical matroids and their duals [180].

Finally, while the Laplacian operator is frequently considered in the graph sparsification and coarsening literature, its pseudoinverse also has many important applications in the field of machine learning [188], eg: online learning over graphs [104]; similarity prediction of network data [88]; determining important nodes [227]; providing a measure of network robustness to multiple failures [187]; extending principal component analysis to graphs [197]; and collaborative recommendation systems [184]. Hence, graph reduction algorithms that preserve $\boldsymbol{L}^{\dagger}$ would be useful to the machine learning community.

[2] In the spirit of another common analogy (edge weights as conductances of a network of resistors), breaking a resistor is equivalent to deleting that edge, while contraction amounts to completely soldering over it.

4.5 Our graph reduction framework

We now describe our framework for constructing probabilistic algorithms that generate a reduced graph $\widetilde{G}$ from an initial graph G, motivated by the following desiderata: **1)** Reduce the number of edges/nodes (Section 4.5.1); **2)** Preserve $\boldsymbol{L}^\dagger$ in expectation (Section 4.5.2); and **3)** Minimize the change in $\boldsymbol{L}^\dagger$ (Section 4.5.3).

We first define these goals more formally. Then, in Section 4.5.4, we combine these requirements to define our cost function and derive the optimal probabilistic action (ie, deletion, contraction, or reweight) to perform to an edge.

4.5.1 Reducing edges and nodes

Depending on the application, it might be more important to reduce the number of nodes (eg, coarsening a sparse network) or the number of edges (eg, sparsifying a dense network). Let r be the number of prioritized items reduced during a particular iteration. When those items are nodes, then $r = 0$ for a deletion, and $r = 1$ for a contraction. When those items are edges, then $r = 1$ for a deletion, however $r > 1$ for a contraction is possible: if the contracted edge forms a triangle in the original graph, then the other two edges will become parallel in the reduced graph (see Figure 4.6 in Section 4.9.2). With respect to the Laplacian, this is equivalent to a single edge with weight given by the sum of these now parallel edges. Thus, when edge reduction is prioritized, a contraction will have $r = 1 + \tau_e$, where τ_e is the number of triangles in the original graph G in which the contracted edge e participates.

Note that, even when node reduction is prioritized, the number of edges will also necessarily decrease. Conversely, when edge reduction is prioritized, contraction of an edge is also possible, thereby reducing the number of nodes as well. For the case of simultaneously sparsifying *and* coarsening a graph, we choose to prioritize edge reduction, although nodes could also be a sensible choice.

4.5.2 Preserving the Laplacian pseudoinverse

Consider perturbing the weight of a single edge $e = (v_1, v_2)$ by Δw. The change in the Laplacian is

$$\boldsymbol{L}_{\widetilde{G}} - \boldsymbol{L}_{G} = \Delta w \vec{b}_e \vec{b}_e^{\top}, \tag{4.1}$$

where $\boldsymbol{L}_{\widetilde{G}}$ and $\boldsymbol{L}_{G}$ are the perturbed and original Laplacians, respectively, and $\vec{b}_e$ is the (arbitrarily) signed incidence (column) vector associated with edge e, with entries

$$(b_e)_i = \begin{cases} +1 & i = v_1 \\ -1 & i = v_2 \\ 0 & \text{otherwise.} \end{cases} \tag{4.2}$$

The change in $\boldsymbol{L}^{\dagger}$ is given by the Woodbury matrix identity[3] [232]:

$$\boldsymbol{L}_{\widetilde{G}}^{\dagger} - \boldsymbol{L}_{G}^{\dagger} = -\frac{\Delta w}{1 + \Delta w \vec{b}_e^{\top} \boldsymbol{L}_{G}^{\dagger} \vec{b}_e} \boldsymbol{L}_{G}^{\dagger} \vec{b}_e \vec{b}_e^{\top} \boldsymbol{L}_{G}^{\dagger}. \tag{4.3}$$

Note that this change can be expressed as a matrix that depends *only* on the choice of edge e, multiplied by a scalar term that depends (nonlinearly) on the change to its weight:

$$\boldsymbol{\Delta L}^{\dagger} = \underbrace{f\left(\tfrac{\Delta w}{w_e}, w_e \Omega_e\right)}_{\text{nonlinear scalar}} \times \underbrace{\boldsymbol{M}_e}_{\text{constant matrix}}, \tag{4.4}$$

[3] This expression is only officially applicable when the initial and final matrices are full-rank; additional care must be taken when they are not. However, for the case of changing the edge weights of a graph Laplacian, the original formula remains unchanged [191, 168] (so long as the graph remains connected), provided one uses the definitions in Section 4.5.5 (see also Sections 4.9.2 and 4.9.5).

where

$$f = -\frac{\frac{\Delta w}{w_e}}{1 + \frac{\Delta w}{w_e} w_e \Omega_e}, \tag{4.5}$$

$$\boldsymbol{M}_e = w_e \boldsymbol{L}_G^\dagger \vec{b}_e \vec{b}_e^\top \boldsymbol{L}_G^\dagger, \tag{4.6}$$

$$\Omega_e = \vec{b}_e^\top \boldsymbol{L}_G^\dagger \vec{b}_e. \tag{4.7}$$

Hence, if the probabilistic reweight of this edge is chosen such that $\mathbb{E}[f] = 0$, then we have $\mathbb{E}[\boldsymbol{L}_{\tilde{G}}^\dagger] = \boldsymbol{L}_G^\dagger$, as desired. Importantly, f remains finite in the following relevant limits:

$$\begin{aligned} \text{deletion:} \quad & \tfrac{\Delta w}{w_e} \to -1, \qquad f \to (1 - w_e \Omega_e)^{-1} \\ \text{contraction:} \quad & \tfrac{\Delta w}{w_e} \to +\infty, \qquad f \to -(w_e \Omega_e)^{-1}. \end{aligned} \tag{4.8}$$

Note that f diverges when considering deletion of an edge with $w_e \Omega_e = 1$ (ie, an edge cut). Indeed, such an action would disconnect the graph and invalidate the use of equation (4.3) (see footnote 3). However, this possibility is precluded by the requirement that $\mathbb{E}[f] = 0$.

4.5.3 Minimizing the error

Minimizing the magnitude of $\boldsymbol{\Delta L}^\dagger$ requires a choice of matrix norm, which we take to be the sum of the squares of its entries (ie, the square of the Frobenius norm). Our motivation is twofold. First, the algebraically convenient fact that the Frobenius norm of a rank one matrix has a simple form, viz,

$$m_e \equiv \|\boldsymbol{M}_e\|_{\mathrm{F}} = w_e \vec{b}_e^\top \boldsymbol{L}_G^\dagger \boldsymbol{L}_G^\dagger \vec{b}_e. \tag{4.9}$$

Second, the square of this norm behaves as a variance; to the extent that the $\boldsymbol{M}_e$ associated to different edges can be treated as (entrywise) uncorrelated one can decompose multiple perturbations as follows:

$$\mathbb{E}\left[\left\|\sum \boldsymbol{\Delta L}^{\dagger}\right\|_{\mathrm{F}}^{2}\right] \approx \sum \mathbb{E}\left[\left\|\boldsymbol{\Delta L}^{\dagger}\right\|_{\mathrm{F}}^{2}\right], \tag{4.10}$$

which allows the single-edge results from Section 4.5.4 to be iteratively applied to our reduction algorithm, which has multiple reductions (Section 4.6).

In Figure 4.1, we empirically validate this assumption for networks with a variety of structures. In fact, the true error is statistically equal to or less than the estimated error. Thus, the estimated error may be used by `StopCriterion` in Algorithm 1.

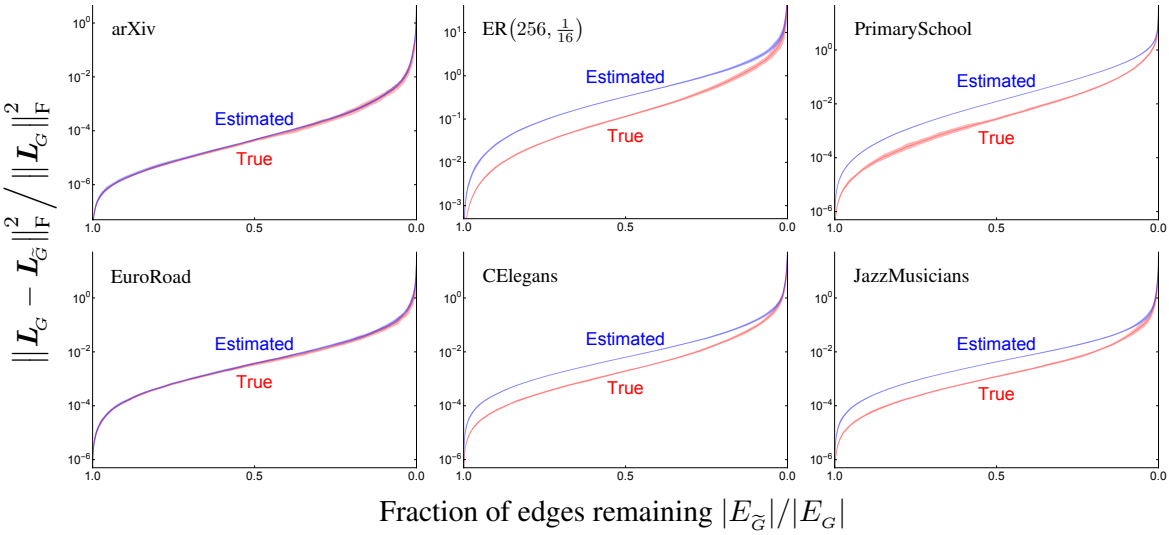


Figure 4.1: **The approximation of uncorrelated changes to $\boldsymbol{L}^\dagger$ is nearly exact or a conservative estimate**. We test the validity of equation (4.10) using a variety of datasets: *Top left*: a triangular mesh of the text "arXiv" (902 nodes and 2203 edges); *Top middle*: an Erdős–Rényi model (256 nodes and $p = 1/16$); *Top right*: a weighted social network of face-to-face interactions between primary school students, with initial edge weights proportional to the number of interactions between pairs of students (236 nodes and 5899 edges) from [215]; *Bottom left*: a transportation network of European cities and roads between them (1039 nodes and 1305 edges) from [225]; *Bottom middle*: the *C. elegans* posterior nervous system connectome (269 nodes and 2902 edges) from [118]; and *Bottom right*: a collaboration network of Jazz musicians (198 nodes and 2742 edges) from [90]. We applied Algorithm 1, prioritizing edge reduction (allowing for deletion, contraction, and reweighting), and setting $q = 1/16$ and $d = 1/4$. We recorded the estimated error and the true error in $\boldsymbol{L}^\dagger$ as a function of amount of reduction. Shading denotes one standard deviation about the mean for 32 runs of the algorithm. In general, the estimated error serves as an approximate upper bound of the true error in $\boldsymbol{L}^\dagger$ (although it is nearly exact for graphs with a geometric quality). The validity of the approximation allows one to use a bound on the estimated error as a `StopCriterion` in Algorithm 1 (Section 4.6).

For subtleties associated with edge contraction (see Section 4.9.5, in particular equation (4.38)).

4.5.4 A cost function for spectral graph reduction

Combining the discussed desiderata, we choose to minimize the following cost function:

$$\mathcal{C} = \mathbb{E}\left[\left\|\boldsymbol{\Delta L}^{\dagger}\right\|_{\mathrm{F}}^{2}\right] - \beta^{2}\mathbb{E}[r]\,, \tag{4.11}$$

subject to

$$\mathbb{E}\left[\boldsymbol{\Delta L}^{\dagger}\right] = \boldsymbol{0}, \tag{4.12}$$

where the parameter β controls the tradeoff between number of prioritized items reduced r and error incurred in $\boldsymbol{L}^{\dagger}$. This cost function naturally arises when minimizing the expected squared error for a given expected amount of reduction (or equivalently maximizing the expected number of reductions for a given expected squared error).

We desire to minimize this cost function over all possible reduced graphs. As, when reducing multiple edges, $\mathbb{E}[r]$ is additive and the expected squared error is empirically additive, we are able to decompose this objective into a sequence of minimizations applied to individual edges. Thus, minimization of this cost function for each edge acted upon can be seen as a probabilistic greedy algorithm for minimizing the cost function for the final reduced graph.

Here, we describe the analytic solution for the optimal action (ie, probabilistically choosing to delete, contract, or reweight) to be applied to a single edge. We provide the solution in Figure 4.2, and a detailed derivation in Section 4.9.1.

For a given edge e, the values of m_e, $w_e\Omega_e$, and τ_e are fixed, and minimizing the cost function (4.11) (given (4.12)) results in a piecewise solution with three regimes, depending on the value of β: **1)** When $\beta < \beta_1(m_e, w_e\Omega_e, \tau_e) = \min(\beta_{1d}, \beta_{1c})$, β is small compared with the error that would be incurred by acting on this edge, thus it should not be changed; **2)** When $\beta > \beta_2(m_e, w_e\Omega_e, \tau_e)$, β is large for this edge, and the optimal solution is to

probabilistically delete or contract this edge ($p_d + p_c = 1$; no reweight is required); and **3)** In the intermediate case ($\beta_1 < \beta < \beta_2$), there are two possibilities, depending on the edge and the choice of prioritized items: if $\beta_{1d} < \beta_{1c}$, the edge is either deleted or reweighted, and if $\beta_{1c} < \beta_{1d}$, the edge is either contracted or reweighted.

$\beta < \beta_1$	$p_d = 0, \quad p_c = 0, \quad \frac{\Delta w}{w_e} = 0$
$\beta_1 < \beta < \beta_2$, $\beta_{1d} < \beta_{1c}$	$p_d = 1 - \frac{m_e}{(1-w_e\Omega_e)\beta}, \quad p_c = 0,$ $\frac{\Delta w}{w_e} = \left(1 - \frac{p_d}{1-w_e\Omega_e}\right)^{-1} - 1$
$\beta_1 < \beta < \beta_2$, $\beta_{1c} < \beta_{1d}$	$p_d = 0, \quad p_c = 1 - \frac{m_e}{w_e\Omega_e\beta\sqrt{1+\tau_e}},$ $\frac{\Delta w}{w_e} = -\frac{p_c}{w_e\Omega_e}$
$\beta > \beta_2$	$p_d = 1 - w_e\Omega_e, \quad p_c = w_e\Omega_e$

	prioritizing edges	prioritizing nodes
β_{1d}	$\frac{m_e}{1-w_e\Omega_e}$	∞
β_{1c}	$\frac{m_e}{w_e\Omega_e}\frac{1}{\sqrt{1+\tau_e}}$	$\frac{m_e}{w_e\Omega_e}$
β_2	$\frac{m_e}{w_e\Omega_e(1-w_e\Omega_e)}\frac{1}{1+\sqrt{1+\tau_e}}$	$\frac{m_e}{w_e\Omega_e(1-w_e\Omega_e)}$

Figure 4.2: *Left*: **Minimizing $\mathcal{C}$ for a single edge** e. There are three regimes for the solution, depending on the value of β. When node reduction is prioritized, set $\tau_e = 0$. *Right*: **Values of β dividing the three regimes**. Note that when edge reduction is prioritized, the number of triangles enters the expressions, and when node reduction is prioritized, there is no deletion in the intermediate regime. However, for either choice, both deletion and contraction can have finite probability, and the algorithm does not exclusively reduce one or the other. Thus, when simultaneously sparsifying and coarsening a graph, the prioritized items may be chosen to be either edges or nodes. We remark that the values of β_{1d}, β_{1c}, and β_2 might be of independent interest as measures of edge importance for analyzing connections in real-world networks.

4.5.5 Node-weighted Laplacian

When nodes are merged, one often represents the connectivity of the resulting graph $\widetilde{G}$ by a matrix of smaller size. To properly compare the spectral properties of $\widetilde{G}$ with those of the original graph G, one must keep track of the number of original nodes that comprise these "supernodes" and assign them proportional weights. The appropriate reduced Laplacian $\boldsymbol{L}_{\widetilde{G}}$ (of size $|V_{\widetilde{G}}| \times |V_{\widetilde{G}}|$) is then $\boldsymbol{W}_n^{-1}\boldsymbol{B}^\top\boldsymbol{W}_e\boldsymbol{B}$, where the $\boldsymbol{W}$ are the diagonal matrices of the

node weights[4] and the edge weights of $\widetilde{G}$, respectively, and $\boldsymbol{B}$ is its signed incidence matrix with columns given by (4.2).

Moreover, one must be careful to choose the appropriate pseudoinverse of $\boldsymbol{L}_{\widetilde{G}}$, which is given by

$$\boldsymbol{L}_{\widetilde{G}}^{\dagger} = \left(\boldsymbol{L}_{\widetilde{G}} + \boldsymbol{J}\right)^{-1} - \boldsymbol{J}, \tag{4.13}$$

$$\boldsymbol{J} = \frac{1}{\vec{1}^{\top}\vec{w}_n}\vec{1}\vec{w}_n^{\top}, \tag{4.14}$$

where $\vec{w}_n \in \mathbb{R}_{>0}^{|V_G|}$ is the vector of node weights. Note that $\boldsymbol{L}_{\widetilde{G}}^{\dagger}\boldsymbol{L}_{\widetilde{G}} = \boldsymbol{L}_{\widetilde{G}}\boldsymbol{L}_{\widetilde{G}}^{\dagger} = \boldsymbol{I} - \boldsymbol{J}$, the appropriate node-weighted projection matrix.

To compare the action of the original and reduced Laplacians on a vector $\vec{x} \in \mathbb{R}^{|V_G|}$ over the nodes of the original graph, one must "lift" $\boldsymbol{L}_{\widetilde{G}}$ to operate on the same space as $\boldsymbol{L}_{G}$. We thus define the mapping from original to coarsened nodes as a $|V_{\widetilde{G}}| \times |V_G|$ matrix $\boldsymbol{C}$, with entries

$$c_{ij} = \begin{cases} 1 & \text{node } j \text{ in supernode } i \\ 0 & \text{otherwise.} \end{cases} \tag{4.15}$$

The appropriate lifted Laplacian is $\boldsymbol{L}_{\widetilde{G},l} = \boldsymbol{C}^{\top}\boldsymbol{L}_{\widetilde{G}}\boldsymbol{W}_n^{-1}\boldsymbol{C}$. Likewise, the lifted Laplacian pseudoinverse is $\boldsymbol{L}_{\widetilde{G},l}^{\dagger} = \boldsymbol{C}^{\top}\boldsymbol{L}_{\widetilde{G}}^{\dagger}\boldsymbol{W}_n^{-1}\boldsymbol{C}$ (see Section 4.9.2 for a detailed rationale of these definitions).

4.6 Our graph reduction algorithm

Using this framework, we now describe our graph reduction algorithm. Similar to many graph coarsening methods [103, 195], we obtain the reduced graph by acting on the initial

[4] $\boldsymbol{W}_n$ is often referred to as the "mass matrix" [140]. We note that the use of the random walk matrix $\boldsymbol{D}^{-1}\boldsymbol{L}$ can be seen as using the node degrees as a surrogate for the node weights.

graph (as opposed to adding edges to an empty graph, as is frequently done in sparsification [145, 152]).

Care must be taken, however, as simultaneous deletions/contractions may result in undesirable behavior. Eg, while any edge that is itself a cut-set will never be deleted (as $w_e\Omega_e = 1$), a collection of edges that together make a cut-set might all have finite deletion probability. Hence, if multiple edges are simultaneously deleted, the graph could become disconnected. In addition, the single-edge analysis could underestimate the change in $\boldsymbol{L}^\dagger$ associated with simultaneous contractions. Eg, consider two highly-connected nodes that are each the center of a different community, and a third auxiliary node that happens to be connected to both: contracting the auxiliary node into either of the other two would be sensible, but performing both contractions would merge the two communities.

Algorithm 1 describes our graph reduction scheme. Its inputs are: G, the original graph; q, the fraction of sampled edges to act upon per iteration; d, the minimum expected decrease in prioritized items per edge acted upon; and `StopCriterion`, a user-defined function. With these inputs, we implicitly select β. Let $\beta_{\star e}$ be the minimum β such that $\mathbb{E}[r] \geq d$ for edge e. For each iteration, we compute $\beta_{\star e}$ for all sampled edges, and choose a β such that a fraction q of them have $\beta_{\star e} < \beta$. We then apply the corresponding probabilistic actions to these edges. The appropriate choice of `StopCriterion` depends on the application. Eg, if one desires to bound the accuracy of an algorithm that uses graph reduction as a primitive, limiting the Frobenius error in $\boldsymbol{L}^\dagger$ is a sensible choice (it is trivial to keep a running total of the estimated error, see Section 4.5.3). On the other hand, if one would like the reduced graph to be no larger than a certain size, then one can simply continue reducing until this point. While both criteria may also be implicitly implemented via an upper bound on β, the relationship is nontrivial and depends on the structure of the graph.

The aforementioned problems associated with simultaneous deletions/contractions can be eliminated by taking a conservative approach: acting on only a single edge per iteration.

Algorithm 1 `ReduceGraph`

Inputs: graph G, fraction of sampled edges to act upon q,
minimum $\mathbb{E}[r]$ per edge acted upon d, and a `StopCriterion`
Initialize $\widetilde{G}_0 \leftarrow G$, $t \leftarrow 0$, *stop* $\leftarrow$ False
while not (*stop*) **do**
  Sample an independent edge set
  for (edge e) **in** (sampled edges) **do**
    Compute Ω_e, m_e (see equations (4.7) and (4.9))
    Evaluate $\beta_{\star e}$, according to d (see Tables in Figure 4.2)
  end for
  Choose β such that a fraction q of the sampled edges
  (those with the lowest $\beta_{\star e}$) are acted upon
  Probabilistically choose to reweight, delete, or contract these edges
  Perform reweights and deletions to $\widetilde{G}_t$
  Perform contractions to $\widetilde{G}_t$
  $\widetilde{G}_{t+1} \leftarrow \widetilde{G}_t,\ t \leftarrow t+1$
  stop $\leftarrow$ `StopCriterion`$(\widetilde{G}_t)$
end while
return reduced graph $\widetilde{G}_t$

However, this results in an algorithm that does not scale favorably for large graphs. A more scalable solution involves carefully sampling the candidate set of edges. In particular, we are able to significantly ameliorate these issues by sampling the candidate edges such that they do not have any nodes in common (ie, the sampled edges form an independent edge set). Not only does this eliminate the possibility of "accidental" contractions, but, empirically, it also suppresses the occurrence of graph disconnections (the small fraction that become disconnected are restarted). At each iteration, our algorithm finds a random maximal independent edge set in $\mathcal{O}(|V|)$ time using a simple greedy algorithm.[5] In practice, the size of such a set scales as $\mathcal{O}(|V|)$ (although it is easy to find families for which this scaling does not hold, eg, star graphs). Our algorithm then computes the Ω_e and m_e of these sampled edges, and acts on the fraction q with the lowest $\beta_{\star e}$.

[5] Specifically, randomly permute the nodes, and sequentially pair them with a random available neighbor (if there is one). The obtained set contains at least half as many edges as the maximum matching [10].

The main computational bottleneck of our algorithm is computing Ω_e and m_e (equation (4.9)). However, we can draw on the work of [212], which describes a method for efficiently computing ε-approximate values of Ω_e for all edges, requiring $\widetilde{\mathcal{O}}(|E| \log |V|/\epsilon^2)$ time. With minimal changes, this procedure can also be used to compute approximate values of m_e with similar efficiency (in Section 4.9.5, we discuss the details of how to efficiently compute approximations of m_e). As we must compute these quantities for each iteration, we multiply the running time by the expected number of iterations, $\mathcal{O}(|E|/qd|V|)$. Empirically, we find that one is able to set $q \sim 1/16$ and $d \sim 1/4$ with minimal loss in reduction quality (see Section 4.9.4). Thus, we expect that our algorithm could have a running time of $\widetilde{\mathcal{O}}(\langle k \rangle |E|)$, where $\langle k \rangle$ is the average degree. However, in the following results, we have used a naive implementation: computing $\boldsymbol{L}^\dagger$ at the onset, and updating it using the Woodbury matrix identity.

4.7 Experimental results

In this section, we empirically validate our framework and compare it with existing algorithms. We consider two cases of our general framework, namely graph sparsification (excluding regimes involving edge contraction), and graph coarsening (prioritizing reduction of nodes). In addition, as graph reduction is often used in graph visualization, we generated videos of our algorithm simultaneously sparsifying and coarsening several real-world datasets (see footnote 1 and Appendix B.2).

4.7.1 Hyperbolic interlude

When comparing a graph G with its reduced approximation $\widetilde{G}$, it is natural to consider how relevant linear operators treat the same input vector. If the vector $\boldsymbol{L}_{\widetilde{G},l}\vec{x}$ is aligned with $\boldsymbol{L}_G\vec{x}$, the fractional error in the quadratic form $\vec{x}^\top \boldsymbol{L}\vec{x}$ is a natural quantity to consider, as it

corresponds to the relative change in the magnitude of these vectors. However, it is not so clear how to compare output vectors that have an angular difference. Here, we describe a natural extension of this notion of fractional error, which draws intuition from the Poincaré half-plane model of hyperbolic geometry. In particular, we choose the boundary of the half-plane to be perpendicular to $\vec{x}$ and compute the geodesic distance between $\boldsymbol{L}_{G}\vec{x}$ and $\boldsymbol{L}_{\widetilde{G},l}\vec{x}$, viz,

$$d_{\vec{x}}(\boldsymbol{L}_0, \boldsymbol{L}_1) \stackrel{\text{def}}{=} \operatorname{arccosh}\left(1 + \frac{\left\|(\boldsymbol{L}_0 - \boldsymbol{L}_1)\vec{x}\right\|_2^2 \left\|\vec{x}\right\|_2^2}{2\left(\vec{x}^\top \boldsymbol{L}_0 \vec{x}\right)\left(\vec{x}^\top \boldsymbol{L}_1 \vec{x}\right)}\right), \tag{4.16}$$

where $\boldsymbol{L}_0$ and $\boldsymbol{L}_1$ are positive definite matrices (for now).

We define the hyperbolic distance between these matrices as

$$d_h(\boldsymbol{L}_0, \boldsymbol{L}_1) \stackrel{\text{def}}{=} \sup_{\vec{x}} d_{\vec{x}}(\boldsymbol{L}_0, \boldsymbol{L}_1)\,. \tag{4.17}$$

This dimensionless quantity inherits the following standard desirable features of a distance: symmetry and non-negativity, $d_h(\boldsymbol{L}_0, \boldsymbol{L}_1) = d_h(\boldsymbol{L}_1, \boldsymbol{L}_0) \geq 0$; identity of indiscernibles, $d_h(\boldsymbol{L}_0, \boldsymbol{L}_1) = 0 \iff \boldsymbol{L}_0 = \boldsymbol{L}_1$; and subadditivity, $d_h(\boldsymbol{L}_0, \boldsymbol{L}_2) \leq d_h(\boldsymbol{L}_0, \boldsymbol{L}_1) + d_h(\boldsymbol{L}_1, \boldsymbol{L}_2)$. In addition, we note that $d_h(c\boldsymbol{L}_0, c\boldsymbol{L}_1) = d_h(\boldsymbol{L}_0, \boldsymbol{L}_1)\ \forall c \in \mathbb{R}\backslash\{0\}$, emphasizing its interpretation as a fractional error.

This notion naturally extends to (positive semidefinite) graph Laplacians if one considers only vectors $\vec{x}$ that are orthogonal to their kernels (ie, require that $\vec{1}^\top \vec{x} = 0$ when taking the supremum in (4.17)). With this modification, the connection with the spectral graph sparsification can be stated as follows:

Theorem 1. *If $d_h\left(\boldsymbol{L}_G, \boldsymbol{L}_{\widetilde{G}}\right) \leq \ln(\sigma)$, then $\widetilde{G}$ is a σ-spectral approximation of G.*

Here, the notion of σ-spectral approximation is the same as in Spielman & Teng [213] (see Section 4.4), and thus is restricted to sparsification only. The proof is provided in Section 4.9.3.

As $d_{\vec{x}}$ is analogous to the ratio of quadratic forms with $\vec{x}$, d_h is likewise analogous to the notion of a σ-spectral approximation. Moreover, as $d_{\vec{x}}$ and d_h also consider angular differences between $\boldsymbol{L}_G\vec{x}$ and $\boldsymbol{L}_{\tilde{G},l}\vec{x}$, they serve as more sensitive measures of graph similarity.

In the following sections, we compare our algorithm with other graph reduction methods using $d_{\vec{x}}$, where we choose $\vec{x}$ to be eigenvectors of the original graph Laplacian. In Section 4.9.6, we replicate our results using more standard measures (eg, quadratic forms and eigenvalues).

4.7.2 Comparison with spectral graph sparsification

Figure 4.3 compares our algorithm (prioritizing edge reduction, and excluding the possibility of contraction) with the standard spectral sparsification algorithm of Spielman & Srivastava [212] using three real-world datasets. We choose to compare with this particular sparsification method because it directly aims to optimally preserve the Laplacian. To the best of our knowledge, other sparsification methods either do not explicitly preserve properties associated with the Laplacian [202, 4], or share the same spirit as Spielman & Srivastava's algorithm [86] (often considering other settings, such as distributed [143] or streaming [127] computation). The results in Figure 4.3 show that our algorithm better preserves $\boldsymbol{L}^\dagger$ and preferentially preserves its action on eigenvectors associated with global structure.

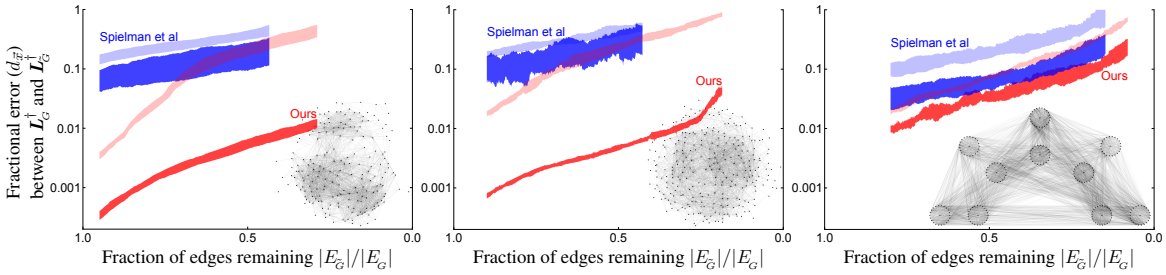


Figure 4.3: **Our sparsification algorithm preferentially preserves global structure**. We applied our algorithm without contraction (**Ours**) and compare with that of Spielman & Srivastava [212] (**Spielman et al**) using three datasets: *Left*: a collaboration network of Jazz musicians (198 nodes and 2742 edges) from [90]; *Middle*: the *C. elegans* posterior nervous system connectome (269 nodes and 2902 edges) from [118]; and *Right*: a weighted social network of face-to-face interactions between primary school students, with initial edge weights proportional to the number of interactions between pairs of students (236 nodes and 5899 edges) from [215]. For the two algorithms, we compute the hyperbolic distance $d_{\vec{x}}$ (fractional error) between $\boldsymbol{L}_G^\dagger \vec{x}$ and $\boldsymbol{L}_{\tilde{G}}^\dagger \vec{x}$ at different levels of sparsification for two choices of $\vec{x}$: the smallest non-trivial eigenvector of the original Laplacian (*dark shading*), which is associated with global structure; and the median eigenvector (*light shading*). Shading denotes one standard deviation about the mean for 16 runs of the algorithms. The curves end at the minimum edge density for which the sparsified graph is connected.

4.7.3 Comparison with graph coarsening algorithms

Figure 4.4 compares our algorithm (prioritizing node reduction) with several existing coarsening algorithms using three more real-world datasets. In order to make a fair comparison with these existing methods, after contracting their prescribed groups of nodes, we appropriately lift the resulting reduced $\boldsymbol{L}_{\tilde{G}}^\dagger$ (see Section 4.9.2). We find that our algorithm more accurately preserves global structure.

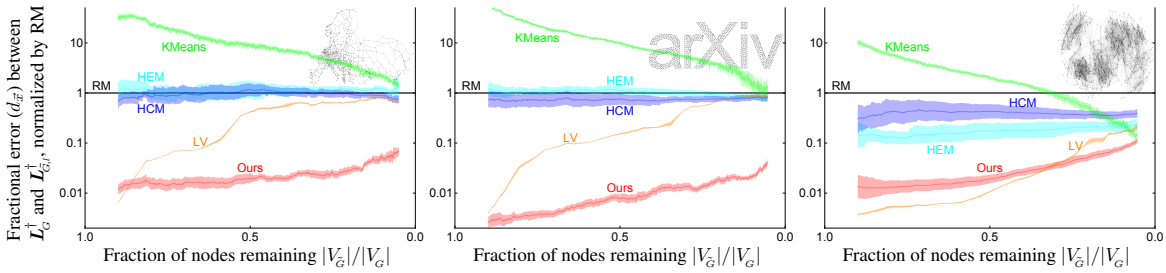


Figure 4.4: **Our algorithm preserves global structure more accurately than other coarsening algorithms**. We compare our algorithm (prioritizing node reduction) (**Ours**) to several existing coarsening algorithms: two classical methods for graph coarsening (heavy-edge matching (**HEM**) [130] and heavy-clique matching (**HCM**) [130]), and two recently proposed spectral coarsening algorithms (local variation by Loukas [158] (**LV**) and the k-means method by Jin & Jaja [121] (**KMeans**)). We ran the comparisons using three datasets: *Left*: a transportation network of European cities and roads between them (1039 nodes and 1305 edges) from [225]; *Middle*: a triangular mesh of the text "arXiv" (902 nodes and 2203 edges); and *Right*: a weighted social network of face-to-face interactions during an exhibition on infectious diseases, with initial edge weights proportional to the number of interactions between pairs of people (410 nodes and 2765 edges) from [116]. For all algorithms considered, we compute the hyperbolic distance $d_{\vec{x}}$ (fractional error) between $\boldsymbol{L}_G^\dagger \vec{x}$ and $\boldsymbol{L}_{\tilde{G},l}^\dagger \vec{x}$, where $\vec{x}$ is the smallest non-trivial eigenvector of the original Laplacian (associated with global structure). To provide a baseline, we plot their mean fractional error normalized by that obtained by random matching (**RM**) [130] for the same level of coarsening. Shading denotes one standard deviation about the mean for 16 runs of the algorithms.

4.8 Conclusion

In this work, we unify spectral graph sparsification and coarsening through the use of a single cost function that preserves the Laplacian pseudoinverse $\boldsymbol{L}^\dagger$. We describe a probabilistic algorithm for graph reduction that employs edge deletion, contraction, and reweighting to keep $\mathbb{E}\left[\boldsymbol{L}_{\tilde{G}}^\dagger\right] = \boldsymbol{L}_G^\dagger$, and uses a new measure of edge importance ($\beta_\star$) to minimize its variance. Using synthetic and real-world datasets, we demonstrate that our algorithm more accurately preserves global structure compared to existing algorithms. We hope that our framework (or

some perturbation of it) will serve as a useful tool for graph algorithms, numerical linear algebra, and machine learning.

4.9 Derivations and Methods

4.9.1 Derivation of the optimal probabilistic action to an edge

As discussed in Section 4.5.4, we seek to minimize:

$$\mathcal{C} = \mathbb{E}\left[\left\|\boldsymbol{\Delta L}^{\dagger}\right\|_{\mathrm{F}}^{2}\right] - \beta^{2}\mathbb{E}[r]\,, \tag{4.18}$$

subject to

$$\mathbb{E}\left[\boldsymbol{\Delta L}^{\dagger}\right] = \boldsymbol{0}. \tag{4.19}$$

When reducing multiple edges, $\mathbb{E}[r]$ is additive and $\mathbb{E}\left[\left\|\boldsymbol{\Delta L}^{\dagger}\right\|_{\mathrm{F}}^{2}\right]$ is approximately additive (see Section 4.5.3). Thus, we partition this minimization into a sequence of subproblems, treating each perturbation to an edge individually.

Recall that

$$\boldsymbol{\Delta L}^{\dagger} = \underbrace{f\left(\tfrac{\Delta w}{w_e}, w_e\Omega_e\right)}_{\text{nonlinear scalar}} \times \underbrace{\boldsymbol{M}_e}_{\text{constant matrix}}, \qquad \text{where} \qquad f = -\frac{\frac{\Delta w}{w_e}}{1 + \frac{\Delta w}{w_e} w_e \Omega_e}.$$

We now derive the optimal probability of deleting (p_d), contracting (p_c), or reweighting ($1 - p_d - p_c$) a given edge e, along with the change to its weight (Δw) in the case of the latter.

The constraint (4.19) requires that this reweight satisfies

$$\frac{p_d}{1-w_e\Omega_e} - \frac{p_c}{w_e\Omega_e} + (1-p_d-p_c)\,\mathbb{E}[f|\text{reweight}] = 0, \tag{4.20}$$

where we have used the following limits:

$$\begin{aligned} \text{deletion:} \quad & \tfrac{\Delta w}{w_e} \to -1, \qquad f \to (1-w_e\Omega_e)^{-1} \\ \text{contraction:} \quad & \tfrac{\Delta w}{w_e} \to +\infty, \qquad f \to -(w_e\Omega_e)^{-1}. \end{aligned} \tag{4.21}$$

Likewise, the cost function (4.18) for acting on the edge e becomes:

$$\mathcal{C} = \left(\frac{p_d}{(1-w_e\Omega_e)^2} + \frac{p_c}{(w_e\Omega_e)^2} + (1-p_d-p_c)\,\mathbb{E}\big[f^2|\text{reweight}\big] \right) m_e^2 - \beta^2\,(r_d p_d + r_c p_c), \tag{4.22}$$

where r_d and r_c are the number of prioritized items that would be removed by a deletion or contraction, respectively.

For a fixed p_d and p_c, $\mathbb{E}[f|\text{reweight}]$ is fixed by equation (4.20). As $\frac{\partial^2 f}{\partial \Delta w^2} > 0$ everywhere, the inequality $\mathbb{E}[f^2|\text{reweight}] \geq \mathbb{E}[f|\text{reweight}]^2$ becomes an equality under minimization of (4.22).

Thus, if an edge is to be reweighted, it will be changed by the unique Δw satisfying

$$\frac{p_d}{1-w_e\Omega_e} - \frac{p_c}{w_e\Omega_e} - (1-p_d-p_c)\frac{\frac{\Delta w}{w_e}}{1+\frac{\Delta w}{w_e}w_e\Omega_e} = 0. \tag{4.23}$$

Clearly, the space of allowed solutions lies within the simplex $\mathcal{S}$: $0 \leq p_d$, $0 \leq p_c$, $p_d + p_c \leq 1$. The additional constraint $-1 \leq \frac{\Delta w}{w_e} \leq \infty$ further implies that $p_c \leq w_e\Omega_e$ and $p_d \leq 1 - w_e\Omega_e$. Hence, we substitute (4.23) into (4.22), and minimize it over this domain (given m_e, $w_e\Omega_e$,

τ_e, and β). After some careful elementary calculus, we obtain the solution provided in Figure 4.2.

4.9.2 Lifting the matrices of a contracted graph

Here, we provide a detailed rationale for the definitions given in Section 4.5.5, namely, the choice of $\boldsymbol{L}_{\widetilde{G}}$ and $\boldsymbol{L}_{\widetilde{G}}^{\dagger}$, and how to "lift" these matrices to the original dimension $|V_G| \times |V_G|$ when edges have been contracted.

Recall the following definitions:

$$\boldsymbol{L}_{\widetilde{G}} = \boldsymbol{W}_n^{-1} \boldsymbol{B}^\top \boldsymbol{W}_e \boldsymbol{B}, \tag{4.24}$$

$$\boldsymbol{L}_{\widetilde{G}}^{\dagger} = \left(\boldsymbol{L}_{\widetilde{G}} + \boldsymbol{J}\right)^{-1} - \boldsymbol{J}, \tag{4.25}$$

$$\boldsymbol{L}_{\widetilde{G},l} = \boldsymbol{C}^\top \boldsymbol{L}_{\widetilde{G}} \boldsymbol{W}_n^{-1} \boldsymbol{C}, \tag{4.26}$$

$$\boldsymbol{L}_{\widetilde{G},l}^{\dagger} = \boldsymbol{C}^\top \boldsymbol{L}_{\widetilde{G}}^{\dagger} \boldsymbol{W}_n^{-1} \boldsymbol{C}, \tag{4.27}$$

where

$$\boldsymbol{J} = \frac{1}{\vec{1}^\top \vec{w}_n} \vec{1} \vec{w}_n^\top, \tag{4.28}$$

$$\boldsymbol{C} = \{c_{ij}\} = \begin{cases} 1 & \text{node } j \text{ in supernode } i \\ 0 & \text{otherwise.} \end{cases} \tag{4.29}$$

The above definitions ensure that the lifted $\boldsymbol{L}_{\widetilde{G},l}^{\dagger}$ of the contracted graph is identical to the $w_e \to \infty$ limit of the original $\boldsymbol{L}_G^{\dagger}$.

To illustrate the consistency of these definitions, we consider a concrete example: the line graph with 3 edges, where the center edge is to be contracted (Figure 4.5). Let the center edge have weight $w_e \gg 1$, while the other two have a fixed weight of 1.

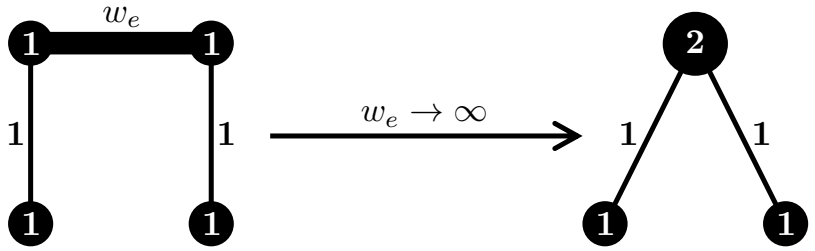


Figure 4.5: **Contracting the center edge of a line graph**. *Left:* Original graph G with large weight w_e on the center edge. *Right:* Reduced graph $\widetilde{G}$ obtained by contracting this edge ($w_e \to \infty$). Note that the weight of the contracted nodes sum to give the weight of the resulting supernode in the reduced graph.

For the original graph G, the Laplacian and its pseudoinverse are

$$\boldsymbol{L}_G = \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 1+w_e & -w_e & 0 \\ 0 & -w_e & 1+w_e & -1 \\ -0 & 0 & -1 & 1 \end{pmatrix}, \quad \boldsymbol{L}_G^\dagger = \frac{1}{8}\begin{pmatrix} 5+\frac{2}{w_e} & -1+\frac{2}{w_e} & -1-\frac{2}{w_e} & -3-\frac{2}{w_e} \\ -1+\frac{2}{w_e} & 1+\frac{2}{w_e} & 1-\frac{2}{w_e} & -1-\frac{2}{w_e} \\ -1-\frac{2}{w_e} & 1-\frac{2}{w_e} & 1+\frac{2}{w_e} & -1+\frac{2}{w_e} \\ -3-\frac{2}{w_e} & -1-\frac{2}{w_e} & -1+\frac{2}{w_e} & 5+\frac{2}{w_e} \end{pmatrix}.$$

For the contracted graph $\widetilde{G}$, we have

$$\boldsymbol{W}_n = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \boldsymbol{J} = \begin{pmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{pmatrix}, \quad \boldsymbol{C} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Thus, the reduced Laplacian and its pseudoinverse are

$$\boldsymbol{L}_{\widetilde{G}} = \begin{pmatrix} 1 & -1 & 0 \\ -\frac{1}{2} & 1 & -\frac{1}{2} \\ 0 & -1 & 1 \end{pmatrix}, \quad \boldsymbol{L}_{\widetilde{G}}^\dagger = \frac{1}{8}\begin{pmatrix} 5 & -2 & -3 \\ -1 & 2 & -1 \\ -3 & -2 & 5 \end{pmatrix}.$$

When lifted to the original dimensions $|V_G| \times |V_G|$, these become

$$\boldsymbol{L}_{\widetilde{G},l} = \begin{pmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} & 0 \\ -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} \\ 0 & -\frac{1}{2} & -\frac{1}{2} & 1 \end{pmatrix}, \quad \boldsymbol{L}^{\dagger}_{\widetilde{G},l} = \frac{1}{8} \begin{pmatrix} 5 & -1 & -1 & -3 \\ -1 & 1 & 1 & -1 \\ -1 & 1 & 1 & -1 \\ -3 & -1 & -1 & 5 \end{pmatrix}.$$

Note that the lifted $\boldsymbol{L}^{\dagger}_{\widetilde{G},l}$ is equal to the $w_e \to \infty$ limit of the original $\boldsymbol{L}^{\dagger}_{G}$, as desired. In contrast, the original $\boldsymbol{L}_{G}$ diverges, while the lifted $\boldsymbol{L}_{\widetilde{G},l}$ averages the rows and columns of the merged nodes. Moreover, regardless of whether node weights are included in the definitions, using the standard Moore–Penrose pseudoinverse of the reduced Laplacian will yield a lifted pseudoinverse that is not equivalent to the original in the $w_e \to \infty$ limit.

Additionally, we remark that, while contraction always requires the summing of node weights, it can also lead to the summing of edge weights (when the contracted edge participates in any triangle in the original graph, see Figure 4.6).

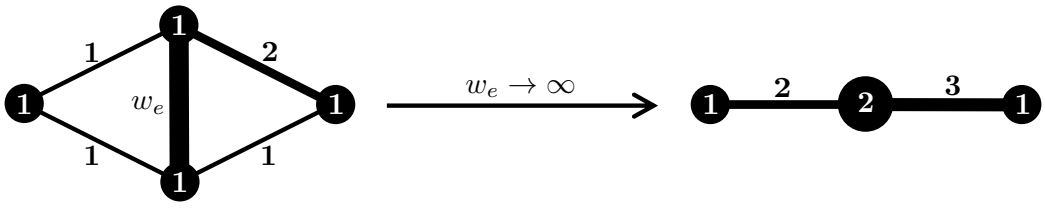


Figure 4.6: **Contracting an edge that participates in triangles**. *Left:* Original graph G containing an edge with large weight w_e that participates in two triangles. *Right:* Reduced graph $\widetilde{G}$ obtained by contracting this edge ($w_e \to \infty$). Note that the two non-contracted edges in each triangle form a single edge in the reduced graph with weight equal to their sum.

4.9.3 Proof of the relationship between the hyperbolic distance and σ-spectral approximation

In this section, we prove Theorem 1 from Section 4.7.1:

Theorem 1. *If $d_h\big(\boldsymbol{L}_{G}, \boldsymbol{L}_{\widetilde{G}}\big) \leq \ln(\sigma)$, then $\widetilde{G}$ is a σ-spectral approximation of G.*

Proof. Let G be the original graph and $\widetilde{G}$ its sparse approximation (no contraction/removing of nodes). Recall the relevant definitions:

$\widetilde{G}$ is a σ-spectral approximation of G [213] if

$$\frac{1}{\sigma}\vec{x}^{\top}\boldsymbol{L}_{G}\vec{x} \leq \vec{x}^{\top}\boldsymbol{L}_{\widetilde{G}}\vec{x} \leq \sigma\vec{x}^{\top}\boldsymbol{L}_{G}\vec{x}, \quad \forall\vec{x} \in \mathbb{R}^{|V_G|}. \tag{4.30}$$

We propose to instead measure the hyperbolic distance between the resulting $\boldsymbol{L}_{G}\vec{x}$ and $\boldsymbol{L}_{\widetilde{G}}\vec{x}$, namely

$$d_h\big(\boldsymbol{L}_{G}, \boldsymbol{L}_{\widetilde{G}}\big) \stackrel{\text{def}}{=} \sup_{\vec{x}\perp\vec{1}} \left\{ \operatorname{arccosh}\left(1 + \frac{\big\|(\boldsymbol{L}_{G} - \boldsymbol{L}_{\widetilde{G}})\vec{x}\big\|_2^2 \, \big\|\vec{x}\big\|_2^2}{2\big(\vec{x}^{\top}\boldsymbol{L}_{G}\vec{x}\big)\big(\vec{x}^{\top}\boldsymbol{L}_{\widetilde{G}}\vec{x}\big)}\right)\right\}, \tag{4.31}$$

where $\boldsymbol{L}_{G}$ and $\boldsymbol{L}_{\widetilde{G}}$ are the Laplacians of G and $\widetilde{G}$, respectively, and $\vec{x}$ is perpendicular to their kernels.

Consider the result of a Laplacian acting on such a vector $\vec{x}$, and decompose the output as a component parallel to $\vec{x}$ with magnitude $\ell_{\parallel}$ and a component $\vec{\ell}_{\perp}$ perpendicular to $\vec{x}$:

$$\boldsymbol{L}_{G}\vec{x} = \ell_{\parallel}\frac{\vec{x}}{\big\|\vec{x}\big\|_2} + \vec{\ell}_{\perp}, \qquad \boldsymbol{L}_{\widetilde{G}}\vec{x} = \widetilde{\ell}_{\parallel}\frac{\vec{x}}{\big\|\vec{x}\big\|_2} + \widetilde{\vec{\ell}}_{\perp}. \tag{4.32}$$

Hence,

$$\left\|(\boldsymbol{L}_G - \boldsymbol{L}_{\widetilde{G}})\vec{x}\right\|_2^2 = (\ell_\parallel - \widetilde{\ell}_\parallel)^2 + \left\|\vec{\ell}_\perp - \vec{\widetilde{\ell}}_\perp\right\|_2^2, \tag{4.33}$$

$$\vec{x}^\top \boldsymbol{L}_G \vec{x} = \ell_\parallel \left\|\vec{x}\right\|_2, \qquad \vec{x}^\top \boldsymbol{L}_{\widetilde{G}} \vec{x} = \widetilde{\ell}_\parallel \left\|\vec{x}\right\|_2. \tag{4.34}$$

Let $z = \widetilde{\ell}_\parallel / \ell_\parallel$. Substituting (4.34) into (4.30), we see that $\widetilde{G}$ is a σ-spectral approximation of G if

$$\max\left\{\frac{\widetilde{\ell}_\parallel}{\ell_\parallel}, \frac{\ell_\parallel}{\widetilde{\ell}_\parallel}\right\} = \max\left\{z, \frac{1}{z}\right\} \leq \sigma. \tag{4.35}$$

Now, substituting 4.33 into (4.16), we obtain:

$$\begin{aligned} d_{\vec{x}}\big(\boldsymbol{L}_G, \boldsymbol{L}_{\widetilde{G}}\big) &= \operatorname{arccosh}\left(1 + \frac{\left((\ell_\parallel - \widetilde{\ell}_\parallel)^2 + \left\|\vec{\ell}_\perp - \vec{\widetilde{\ell}}_\perp\right\|_2^2\right)\left\|\vec{x}\right\|_2^2}{2\ell_\parallel \left\|\vec{x}\right\|_2 \widetilde{\ell}_\parallel \left\|\vec{x}\right\|_2}\right) \\ &\geq \operatorname{arccosh}\left(1 + \frac{\widetilde{\ell}_\parallel^2 - 2\widetilde{\ell}_\parallel \ell_\parallel + \ell_\parallel^2}{2\ell_\parallel \widetilde{\ell}_\parallel}\right) \\ &\geq \operatorname{arccosh}\left(\frac{1}{2}\left(z + \frac{1}{z}\right)\right). \end{aligned}$$

Using the identity $\operatorname{arccosh}(x) = \ln\left(x + \sqrt{x^2 - 1}\right)$,

$$\begin{aligned} d_{\vec{x}}\big(\boldsymbol{L}_G, \boldsymbol{L}_{\widetilde{G}}\big) &\geq \ln\left(\frac{1}{2}\left(z + \frac{1}{z}\right) + \sqrt{\frac{1}{4}\left(z + \frac{1}{z}\right)^2 - 1}\right) \\ &\geq \ln\left(\frac{1}{2}\left(z + \frac{1}{z}\right) + \frac{1}{2}\left|z - \frac{1}{z}\right|\right) \\ &\geq \left|\ln(z)\right|. \end{aligned}$$

Thus, if $d_{\vec{x}}\big(\boldsymbol{L}_G, \boldsymbol{L}_{\widetilde{G}}\big) \leq \ln(\sigma)\ \forall \vec{x} \perp \vec{1}$, then $\widetilde{G}$ is a σ-spectral approximation of G, as desired. □

4.9.4 Number of edges acted upon per iteration can be $\mathcal{O}(|V|)$

In this section, we study the effect of varying the parameter q, the fraction of sampled edges acted upon, using real-world datasets from different domains (Figure 4.7).

For each iteration of our algorithm, we sample a random independent edge set and act on the fraction q with the lowest $\beta_{\star e}$ (see Section 4.6). We find that the resulting error asymptotes around $q \sim 1/16$. We expect that by combining this sampling method with existing algorithmic primitives (eg, [212], see Section 4.9.5), our algorithm could achieve a running time of $\widetilde{\mathcal{O}}(\langle k \rangle |E|)$, where $\langle k \rangle$ is the average degree (see Section 4.6). This would allow it to be used in large-scale applications of graph reduction.

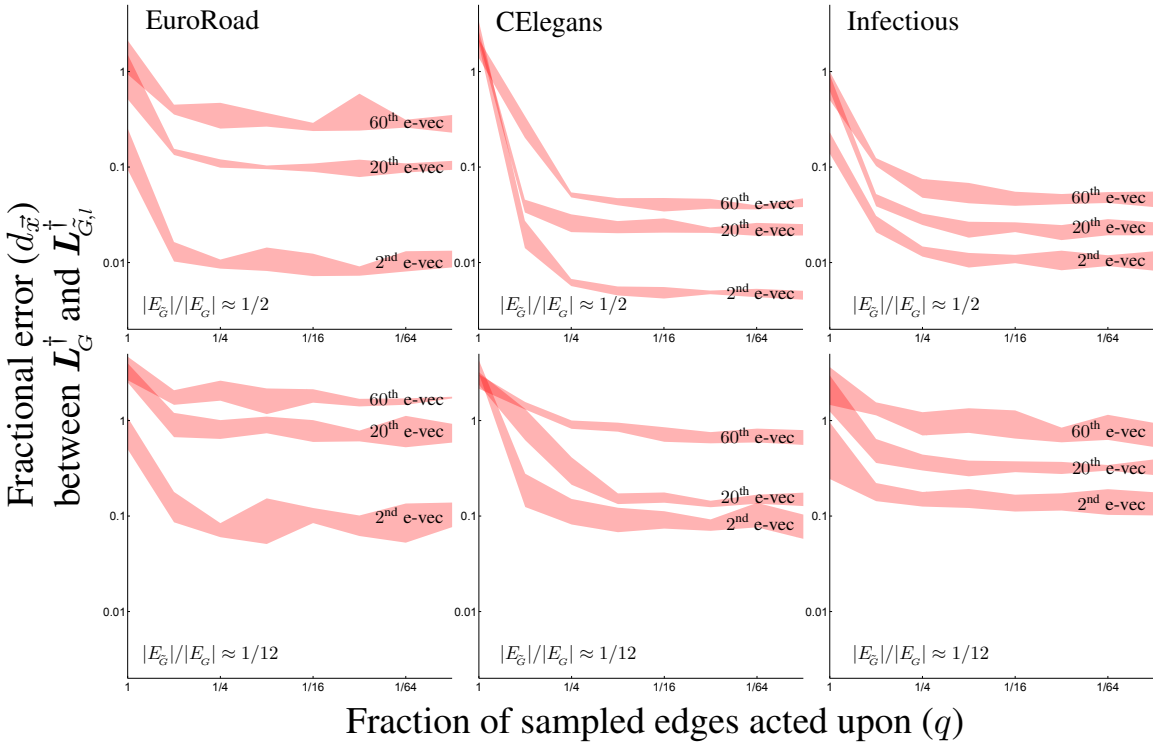


Figure 4.7: **Number of sampled edges acted upon per iteration can be** $\mathcal{O}(|V|)$. We study the effect of varying q, the fraction of the sampled edges that are acted upon per iteration, using three datasets: *Left*: a transportation network of European cities and roads between them (1039 nodes and 1305 edges) from [225]; *Middle*: the *C. elegans* posterior nervous system connectome (269 nodes and 2902 edges) from [118]; and *Right*: a weighted social network of face-to-face interactions during an exhibition on infectious diseases, with initial edge weights proportional to the number of interactions between pairs of people (410 nodes and 2765 edges) from [116]. We prioritize edge reduction (allowing for deletion, contraction, and reweighting). At each iteration, the algorithm randomly samples a maximal independent edge set, and chooses β such that a fraction q of these edges (with the lowest $\beta_{\star e}$) are acted upon. For each run, we compute the hyperbolic distance $d_{\vec{x}}$ (fractional error) between $\boldsymbol{L}_G^\dagger \vec{x}$ and $\boldsymbol{L}_{\tilde{G},l}^\dagger \vec{x}$, where $\vec{x}$ is one of three eigenvectors of the original Laplacian. *Top* plots display the results when the graph has $1/2$ of its original number of edges, and *bottom* plots when it has $1/12$. Shading denotes one standard deviation about the mean for 8 runs of the algorithm for a given value of q. Note that a significant fraction ($q \sim 1/16$) of the sampled edges can be reduced each iteration without sacrificing much in terms of accuracy. As, empirically, the size of the independent edge sets are typically $\mathcal{O}(|V|)$, the number of edges acted upon per iteration can likewise be $\mathcal{O}(|V|)$.

4.9.5 Efficiently computing m_e

As discussed in Section 4.6, the main computational bottleneck of our algorithm is computing Ω_e and m_e. For Ω_e, we can draw on the work of [212], which describes a method for efficiently computing ε-approximate values of Ω_e for all edges, requiring $\widetilde{\mathcal{O}}(|E| \log |V|/\epsilon^2)$ time. In this section, we describe an analogous procedure to efficiently compute the m_e.

Recall that the reduced Laplacian is:

$$\boldsymbol{L}_{\widetilde{G}} = \boldsymbol{W}_n^{-1} \boldsymbol{B}^\top \boldsymbol{W}_e \boldsymbol{B},$$

hence, the quantity $\widehat{\boldsymbol{L}_{\widetilde{G}}} \stackrel{\text{def}}{=} \boldsymbol{W}_n^{1/2} \boldsymbol{L}_{\widetilde{G}} \boldsymbol{W}_n^{-1/2}$ is clearly symmetric.

Less obvious is the fact that $\widehat{\boldsymbol{L}_{\widetilde{G}}^\dagger} \stackrel{\text{def}}{=} \boldsymbol{W}_n^{1/2} \boldsymbol{L}_{\widetilde{G}}^\dagger \boldsymbol{W}_n^{-1/2}$ is also symmetric. This can be seen by noting that $\boldsymbol{W}_n^{1/2} \boldsymbol{J} \boldsymbol{W}_n^{-1/2}$ is symmetric, and using the definition of the inverse (equation (4.25)):

$$\begin{aligned}
\widehat{\boldsymbol{L}_{\widetilde{G}}^\dagger} &= \boldsymbol{W}_n^{1/2} \boldsymbol{L}_{\widetilde{G}}^\dagger \boldsymbol{W}_n^{-1/2} \\
&= \boldsymbol{W}_n^{1/2} \left(\left(\boldsymbol{L}_{\widetilde{G}} + \boldsymbol{J} \right)^{-1} - \boldsymbol{J} \right) \boldsymbol{W}_n^{-1/2} \\
&= \boldsymbol{W}_n^{1/2} \left(\boldsymbol{L}_{\widetilde{G}} + \boldsymbol{J} \right)^{-1} \boldsymbol{W}_n^{-1/2} - \boldsymbol{W}_n^{1/2} \boldsymbol{J} \boldsymbol{W}_n^{-1/2} \\
&= \left(\boldsymbol{W}_n^{1/2} \boldsymbol{L}_{\widetilde{G}} \boldsymbol{W}_n^{-1/2} + \boldsymbol{W}_n^{1/2} \boldsymbol{J} \boldsymbol{W}_n^{-1/2} \right)^{-1} - \boldsymbol{W}_n^{1/2} \boldsymbol{J} \boldsymbol{W}_n^{-1/2}.
\end{aligned}$$

We also remark that $\widehat{\boldsymbol{L}_{\widetilde{G}}^\dagger}$ is indeed the pseudoinverse of $\widehat{\boldsymbol{L}_{\widetilde{G}}}$:

$$\widehat{\boldsymbol{L}_{\widetilde{G}}^\dagger} \widehat{\boldsymbol{L}_{\widetilde{G}}} = \widehat{\boldsymbol{L}_{\widetilde{G}}} \widehat{\boldsymbol{L}_{\widetilde{G}}^\dagger} = \boldsymbol{I} - \boldsymbol{W}_n^{1/2} \boldsymbol{J} \boldsymbol{W}_n^{-1/2}$$

The change to the reduced Laplacian $\boldsymbol{L}_{\widetilde{G}}$ is given by

$$\boldsymbol{\Delta L}_{\widetilde{G}} = \boldsymbol{W}_n^{-1} \vec{b}_e \Delta w_e \vec{b}_e^\top$$

Thus, by the Woodbury matrix identity, the change to its inverse is

$$\boldsymbol{\Delta L}_{\widetilde{G}}^{\dagger} = f w_e \boldsymbol{L}_{\widetilde{G}}^{\dagger} \boldsymbol{W}_n^{-1} \vec{b}_e \vec{b}_e^{\top} \boldsymbol{L}_{\widetilde{G}}^{\dagger}$$

where f is given by equation (4.5).

Lifting this change back to the original dimension via equation (4.27) gives

$$\boldsymbol{\Delta L}_{\widetilde{G},l}^{\dagger} = f w_e \boldsymbol{C}^{\top} \boldsymbol{L}_{\widetilde{G}}^{\dagger} \boldsymbol{W}_n^{-1} \vec{b}_e \vec{b}_e^{\top} \boldsymbol{L}_{\widetilde{G}}^{\dagger} \boldsymbol{W}_n^{-1} \boldsymbol{C}$$

In particular, as $\boldsymbol{L}_{\widetilde{G}}^{\dagger} \boldsymbol{W}_n^{-1}$ is symmetric, $\boldsymbol{\Delta L}_{\widetilde{G},l}^{\dagger}$ is also symmetric, thus we can write the Frobenius norm as

$$\left\| \boldsymbol{\Delta L}_{\widetilde{G},l}^{\dagger} \right\|_F = f w_e \vec{b}_e^{\top} \boldsymbol{L}_{\widetilde{G}}^{\dagger} \boldsymbol{W}_n^{-1} \boldsymbol{C}\, \boldsymbol{C}^{\top} \boldsymbol{L}_{\widetilde{G}}^{\dagger} \boldsymbol{W}_n^{-1} \vec{b}_e \tag{4.36}$$

$$= f m_e \tag{4.37}$$

Note that the definition of m_e provided in Section 4.5.3 (equation (4.9)) applies to the case of unit node weights, and the general expression is given by

$$m_e = w_e \vec{b}_e^{\top} \boldsymbol{L}_{\widetilde{G}}^{\dagger} \boldsymbol{L}_{\widetilde{G}}^{\dagger} \boldsymbol{W}_n^{-1} \vec{b}_e, \tag{4.38}$$

where we have used $\boldsymbol{C}\,\boldsymbol{C}^{\top} = \boldsymbol{W}_n$.

Thus, we can express m_e in terms of $\widehat{\boldsymbol{L}_{\widetilde{G}}^{\dagger}}$:

$$\begin{aligned} m_e &= w_e \vec{b}_e^{\top} \boldsymbol{W}_n^{-1/2} \widehat{\boldsymbol{L}_{\widetilde{G}}^{\dagger}} \widehat{\boldsymbol{L}_{\widetilde{G}}^{\dagger}} \boldsymbol{W}_n^{-1/2} \vec{b}_e \\ &= w_e \left\| \widehat{\boldsymbol{L}_{\widetilde{G}}^{\dagger}} \boldsymbol{W}_n^{-1/2} \vec{b}_e \right\|_2^2. \end{aligned}$$

We can now use the Johnson–Lindenstrauss lemma to build a structure from which one can efficiently compute approximations of m_e. Let $\boldsymbol{Q}$ be a random projection matrix of size $k \times n$, where $k = \mathcal{O}(\log n/\varepsilon^2)$, then one can compute ε-approximations of m_e as follows:

$$m_e \approx w_e \left\| \boldsymbol{Q} \widehat{\boldsymbol{L}_{\tilde{G}}^{\dagger}} \boldsymbol{W}_n^{-1/2} \vec{b}_e \right\|_2^2 .$$

Let $\boldsymbol{Z} = \boldsymbol{Q}\widehat{\boldsymbol{L}_{\tilde{G}}^{\dagger}}$, and denote the i^{th} rows of $\boldsymbol{Q}$ and $\boldsymbol{Z}$ by $\vec{q}_i$ and $\vec{z}_i$, respectively. Then, one can make k calls to an efficient algebraic multigrid solver (we used the *pyamg* package [24]) to obtain approximate solutions to $\widehat{\boldsymbol{L}_{\tilde{G}}}\vec{z}_i = \vec{q}_i$ for the k rows of $\boldsymbol{Z}$. An approximation to the m_e of any edge can now be computed by taking the difference between the columns of $\boldsymbol{Z}\,\boldsymbol{W}_n^{-1/2}$ corresponding to the two nodes jointed by this edge, and taking the squared 2-norm of the result.

Constructing the projection matrix

Care must be taken in constructing the projection matrix $\boldsymbol{Q}$. In particular, its rows must be orthogonal to the null space of $\widehat{\boldsymbol{L}_{\tilde{G}}}$, namely $\vec{w}_n^{1/2}$. In addition, the columns must be nearly unit length. To this end, we initialize $\boldsymbol{Q}$ as a random matrix with entries $\{1/\sqrt{k}, -1/\sqrt{k}\}$ with equal probability and iterate the following steps:

1. For each column, scale its values such that it has unit length
2. For each row, subtract its weighted mean $\vec{q}_i^{\top}\vec{w}_n^{1/2}/\vec{1}^{\top}\vec{w}_n^{1/2}$

We iterate this procedure until the columns have nearly unit lengths, to within a factor sufficiently smaller than ε.

As a proof of concept, in Figure 4.8, we show the approximate m_e as a function of their exact values.

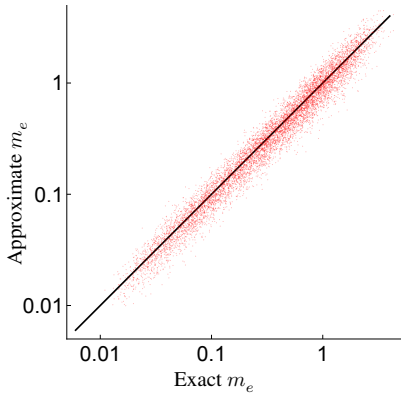


Figure 4.8: **Efficient approximation of m_e.** As a proof of concept, we compare the approximation of m_e (computed using the procedure described in this section) with their exact values. Here, we consider a 64×64 torus graph (4096 nodes and 8192 edges), where the edge weights are randomly distributed as $\exp(U(-2, 2))$, where $U(a, b)$ is the uniform distribution. To calculate the approximate m_e, we project from 4096 to 33 dimensions, resulting in approximations that are typically within a factor of 1.27 of the exact value.

4.9.6 Comparison of graph reduction methods using typical similarity measures

Our proposed hyperbolic distance is not usually used as a measure of similarity. Hence, in this section, we show that other more commonly used measures yield similar results when comparing graph reduction algorithms.

Sparsification

Figure 4.9 compares our algorithm (prioritizing edge reduction, and excluding the possibility of contraction) with the spectral sparsification algorithm of [212] using a stochastic block model (SBM) with four distinct communities. We choose a highly associative SBM due to the clear separation between the eigenvectors associated with global structure (ie, the

communities) and the bulk of the spectrum. Note that these algorithms have different objectives (preserving $\boldsymbol{L}^\dagger$ and $\boldsymbol{L}$, respectively), and both accomplish their desired goal.

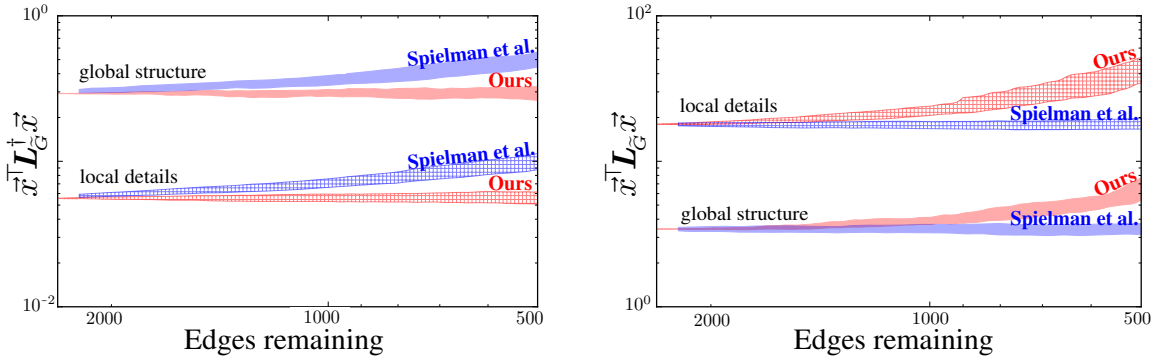


Figure 4.9: **Our sparsification algorithm preferentially preserves global structure**. We compare our algorithm without contraction (in **red**) with that of Spielman & Srivastava [212] (in **blue**) using a symmetric stochastic block model (256 nodes, 4 communities, and intra- and inter-community connection probabilities of 2^{-2} and 2^{-6}, respectively). We ran both algorithms 16 times on the same initial graph. For each eigenvector of the original Laplacian, we compute the mean and standard deviation of its quadratic forms (with $\boldsymbol{L}_{\widetilde{G}}$ and with $\boldsymbol{L}_{\widetilde{G}}^\dagger$) as a function of edges remaining. We divide the eigenvectors into two groups: the 3 nontrivial eigenvectors ("global structure") and the remaining eigenvectors ("local details"), and compute the average mean and average standard deviation for each group. Shading denotes one (average) standard deviation about the (average) mean. *Left*: Laplacian pseudoinverse quadratic form. *Right*: Standard Laplacian quadratic form. Note that the upward bias of the "reciprocal" quadratic form is expected for both algorithms (as $\mathbb{E}[X] \leq 1/\mathbb{E}[1/X]$ for any random variable $X > 0$).

Coarsening

Figure 4.10 replicates the results of Figure 4.4, but uses the Laplacian pseudoinverse quadratic form to measure the reduction quality instead of our proposed hyperbolic distance.

Figure 4.11 compares our method with that of Loukas [158], using the average relative error of the k lowest non-trivial eigenvalues of the Laplacian (ie, $\frac{1}{k}\sum_{i=2}^{k+1} \left|\widetilde{\lambda}_i - \lambda_i\right| / \lambda_i$) to measure the reduction quality.

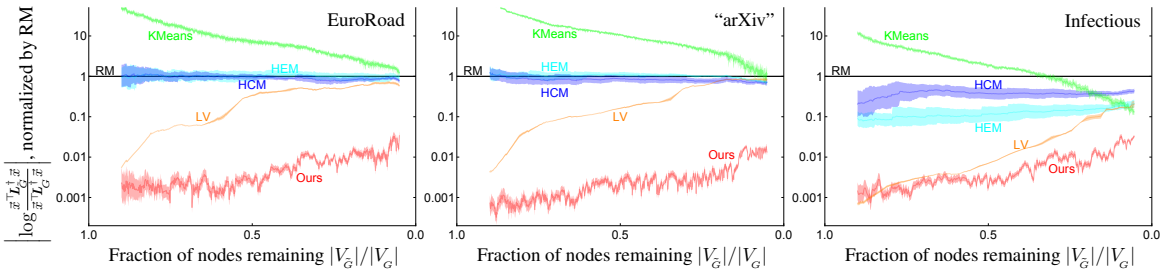


Figure 4.10: **Our coarsening algorithm performs even better when using the quadratic form with $\boldsymbol{L}^\dagger$**. Here we replicate the experiments in Figure 4.4. However, instead of using our proposed hyperbolic distance, we consider the logarithm of the fractional change in the Laplacian pseudoinverse quadratic form for $\vec{x}$ the lowest non-trivial eigenvector of the original Laplacian: $\left|\log\left(\vec{x}^\top \boldsymbol{L}_{\tilde{G}}^\dagger \vec{x} \,/\, \vec{x}^\top \boldsymbol{L}_G^\dagger \vec{x}\right)\right|$. As before, for each algorithm, we plot the mean of this quantity normalized by that obtained by random matching (**RM**). Shading denotes one standard deviation about the mean for 16 runs of the algorithms. The results are remarkably similar to those obtained using our proposed hyperbolic distance (Figure 4.4). The most notable deviation is that our algorithm appears to perform *better* when compared using this quadratic form.

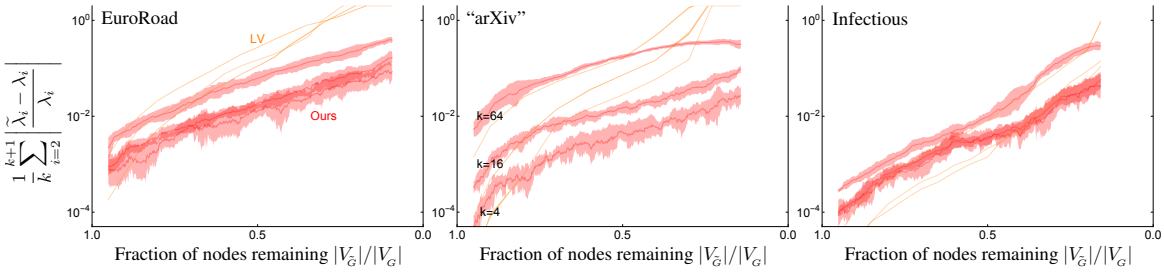


Figure 4.11: **Our algorithm preferentially preserves the lower portion of the Laplacian spectrum**. We compare our coarsening algorithm (**Ours**) with that of Loukas [158] (**LV**) using the same three datasets as in Figure 4.4. We use the relative error in the k lowest non-trivial eigenvalues of the Laplacian: $\frac{1}{k}\sum_{i=2}^{k+1}\left|\widetilde{\lambda}_i - \lambda_i\right|/\lambda_i$, a measure of spectral similarity considered in [158]. Shading denotes one standard deviation about the mean for 8 runs of the algorithms. Note that our algorithm performs considerably better when applied to graphs with a geometric quality.

Chapter 5

The Structure of Human Priors over Navigation and Social Tasks

The nice thing about having a brain is that one can learn,
that ignorance can be supplanted by knowledge,
and that small bits of knowledge can gradually pile up into substantial heaps.

—— Douglas R. Hofstadter,
Le Ton beau de Marot: In Praise of the Music of Language

5.1 Preamble

In this chapter, we quantify human priors over the structure of social and navigation tasks, using the aforementioned methods and theory.

In brief (see Sections 5.4 and 5.5 for a detailed description of the methods), we ran experiments in MTurk using our online platform that allows participants to draw the graphs (see here for a demonstration video). The basic structure of the experiment is as described in Chapter 2, Figure 2.3: the participant observes pairs of relations ("partial graphs") and has to infer the obscured relations. There were four cover stories in total, two in each domain.

For the social domain, the participants had to infer friendships (edges) between pairs of students (nodes) in a classroom, or infer friendships (edges) between pairs of coworkers (nodes) in a workplace. For the navigation domain, the participants had to infer borders (edges) separating pairs of neighborhoods (nodes) in a city, or infer trails (edges) connecting pairs of nature sites (nodes) in a nature park.

We then recover participants' priors by fitting the Bayesian MCMCP model to their data (as in Chapter 2), parametrizing the prior using our hierarchy of distributions over graphs (Chapter 3).

5.2 Main findings

5.2.1 Priors favor sparsity

At lowest order (see Chapter 3), we find that participants tend to favor sparsity as the number of nodes n increases (see Figure 5.1). However, the average number of connections *per node* appears to be a slowly increasing function of n, and is remarkably consistent across all conditions.

This leads to some interesting questions. From the perspective of human cognition, it is well-known that there is a limited amount of information ($\sim 2^5$ bits [126]) that can be held in working memory [11, 65, 164]. This suggests a constraint of bounded degree in human priors over graphs, in accord with our results for a small numbers of nodes. Moreover, the average degree of a planar graph is necessarily less than 6 [5]. And while place and grid cells appear to be adapted to such planar configurations [29], they have also been shown to encode structure in non-spatial domains [214, 76]. Could such a (planar) graphical constraint exist in the hippocampus due to inherent Euclidean proclivities? Regardless, one might posit a

regime change, where the modality of reasoning about graphical structure switches when the number of potential relations exceeds some relevant constraints.

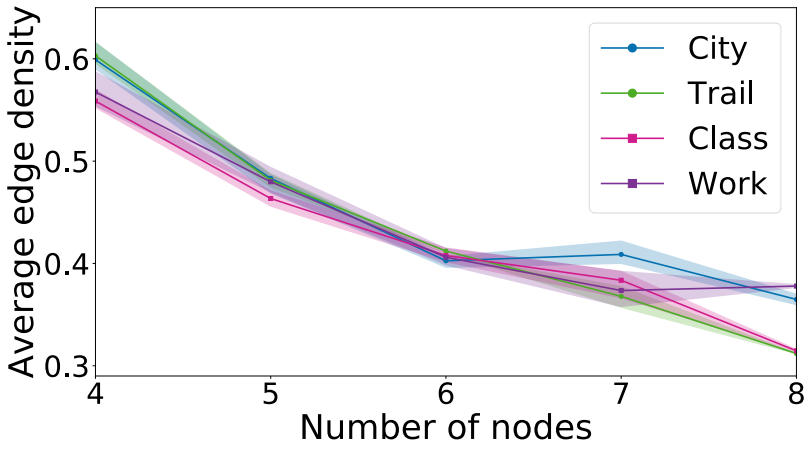


Figure 5.1: **Priors favor sparsity.** Displayed is the average fraction of connections as a function of number of nodes for the fitted priors (using our hierarchical parameterization) for each of the four cover stories: neighborhoods in a city (in **blue**), trails in a nature park (in **green**), friendships in a classroom (in **pink**), and friendships in a workplace (in **purple**). Solid curves denote the average edge density of the priors fitted to the participants' data. To compute the error, we simulated ideal Bayesian MCMCP agents responding to the same data (ie, the partial graphs) seen by the participants, using as a prior that which was fitted to the participants' data. We then fit these simulated data using the same model to obtain a simulated prior. Shading denotes one standard deviation about the average edge density of these simulated priors for 32 repetitions of this process. The dominant trend is that the edge density decreases as a function of the number of nodes. Nevertheless, the number of connections *per node* appears to be a slowly increasing function of the size of the graph. This trend is remarkably similar across different conditions, and could be a general feature of humans' priors over connections.

5.2.2 Priors are more adaptive than iid connections

To further investigate this trend over the number of connections, we now analyze in detail the structure of priors over the number of edges only, without considering their relative positions (ie, neglecting the graphical structure). In Figure 5.2, we display the second and third scaled cumulants of these priors as a function of the number of nodes. Even though

participants have a preference for sparsity, their priors are "adaptive", in the sense that if they indeed observe many connections, they are able to reflect this in their response. In contrast, an agent that treats the connections as completely independent would (by definition) not take this information into consideration in their response (and these scaled cumulants would be zero, as opposed to the consistently positive values observed).

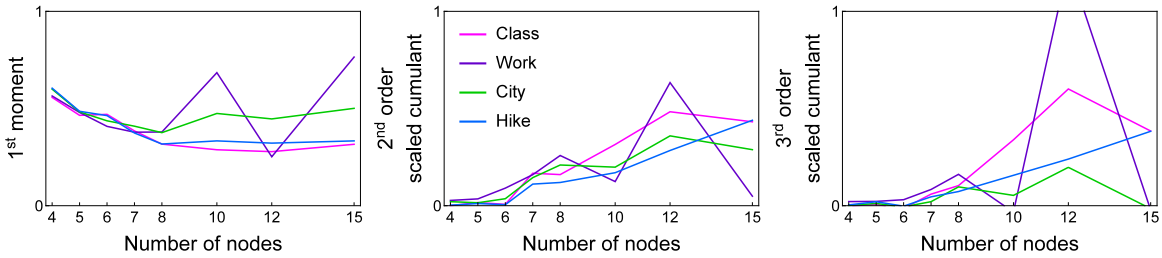


Figure 5.2: **Priors are more adaptive than simply considering iid connections.** Here, we consider priors that are only sensitive to the number of edges in the graph, thereby removing all graphical structure. For each cover story and each number of nodes, we fit the data to the Bayesian MCMCP model with a sixth-order prior over the space of number of edges (see Section 5.5 for details). Displayed are the first moment, along with the appropriately scaled higher-order cumulants. These are the factorial cumulants (the appropriate choice for a distribution over a set of positive integers), and have the same interpretation as the classical cumulants [74]. Namely, the second-order scaled cumulant quantifies the relative spread in the distribution, giving zero if the distribution is Binomial. Likewise, the third-order scaled cumulant quantifies the analogue of skew, or asymmetry, again giving zero if the distribution is Binomial. In particular, the fact that the second-order scaled cumulant is consistently positive suggests that the priors are, in a sense, more "adaptive", in that they more easily encompass graphs with different edge densities.

5.2.3 A regime change in the preferred density of connections

In Figure 5.3, we display these priors distributions over edge density. There appears to be a transition between priors that are unimodal in the number of edges to a symmetric bimodal mixture of sparse and dense distributions. In particular, this suggests that the prior may consistent of essentially two populations. This hypothesis could be tested by fitting a hierarchical Bayesian model consisting of a mixture of multiple populations.

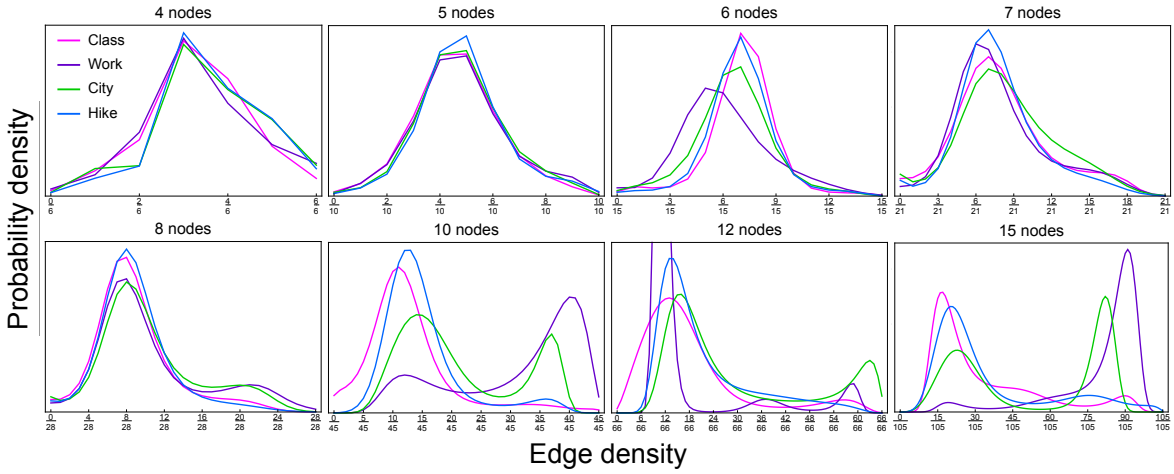


Figure 5.3: **The appearance of bimodality in priors over edge density.** We use the same methodology as in Figure 5.2, and display here the resulting distribution over edge density. For lower numbers of nodes ($\lesssim$5), all cover stories appear to coalesce to a single distribution over edges with an edge density near $1/2$. As the number of nodes increases ($\sim$5–8), the priors seem to favor sparsity. Once such a sparse distribution and its complement are sufficiently well-separated in edge density, two nearly symmetric "populations" emerge.

5.2.4 Priors favor more "egalitarian" configurations

We now investigate the sensitivity of the prior to graphical substructures (see Chapter 3). Considering the second-order cumulants (Figure 5.4), we find that the scaled cumulant associated with two edges that do not share any node is consistently higher than the scaled cumulant associated with the wedge (ie, 2-star). This effect is particularly noticeable for graphs with fewer nodes, and indicates a preference for graphs with degree distributions that are more uniform ("egalitarian"), as opposed to having hierarchical/scale-free properties ("the rich get richer"). However, this difference decreases as the number of nodes increases, suggestively appearing to switch at around eight nodes, in accord with the hypothesis of a qualitative change in the way people represent graphs of different sizes.

For the social domain, this could be interpreted as a belief that people in smaller groups tend to have similar numbers of friends. This is in sharp contrast with the scale-free properties frequently observed in large social networks (eg, facebook, twitter). A cute hypothesis is that this disparity between peoples' priors over small social networks and the properties of large online social networks might explain the somewhat common feeling of social isolation/inadequacy despite such a connected world (although the increase in the preference for wedges with increasing network size suggests the possibility that such structures are indeed reflected in our priors over larger social networks).

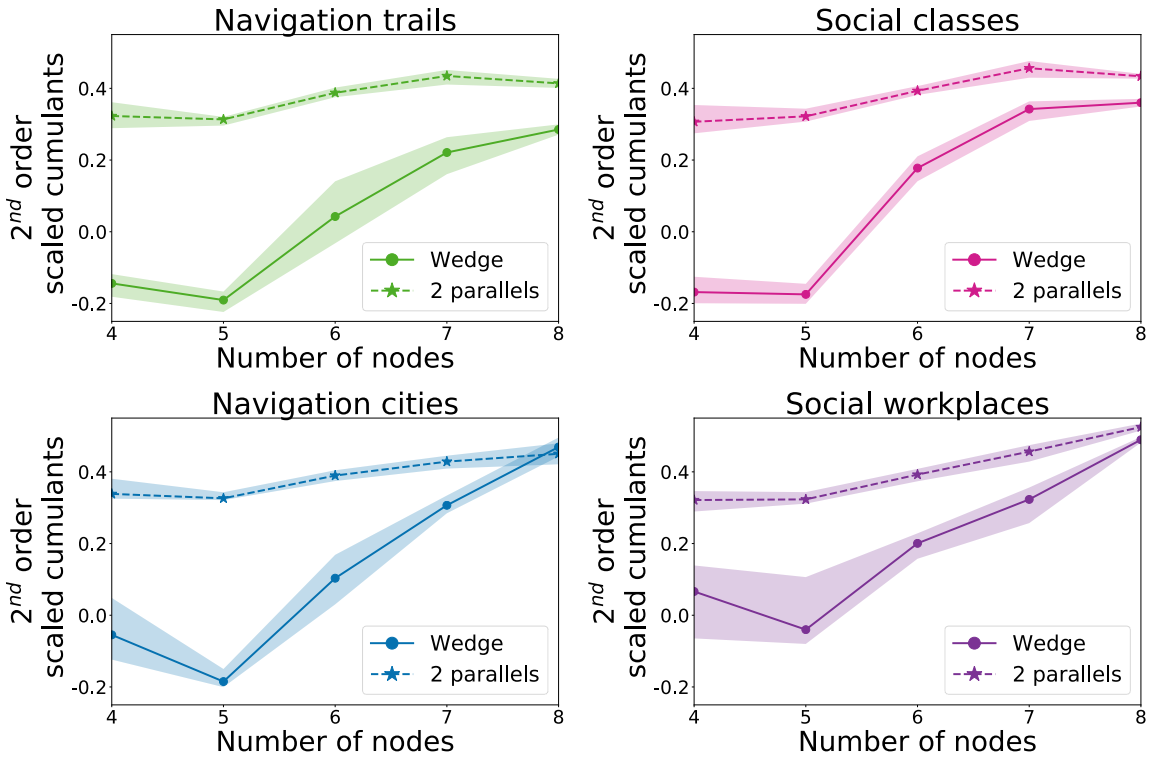


Figure 5.4: **Priors favor more "egalitarian" configurations, specially for small number of nodes.** The methodology to recover the priors and obtain the error bars is the same as in Figure 5.1. Solid curves correspond to the second-order scaled cumulant associated with two edges that share a node (forming a wedge), which measures the preference for graphs in which a small number of nodes participate in many connections (while many nodes have only a few connections). Dashed curves correspond to the second-order scaled cumulant associated with two edges that do not share any node, which measures the preference for graphs in which nodes have similar numbers of connections. The fact that the scaled wedge cumulant is consistently lower indicates that people's priors favor distributing connections more uniformly than would be expected from random placement. While this trend holds for small graphs, the trend suggests that there might be a reversal for larger graphs.

5.2.5 Priors over social interactions favor triangles

At third order, we find a striking preference for triangles in the social domain (as measured by the scaled triangle cumulant), as compared to the navigation domain (Figure 5.5). This is

in accord with the well-known fact that social networks tend to exhibit triadic closure [229] (ie, one's friends tend to be friends with each other).

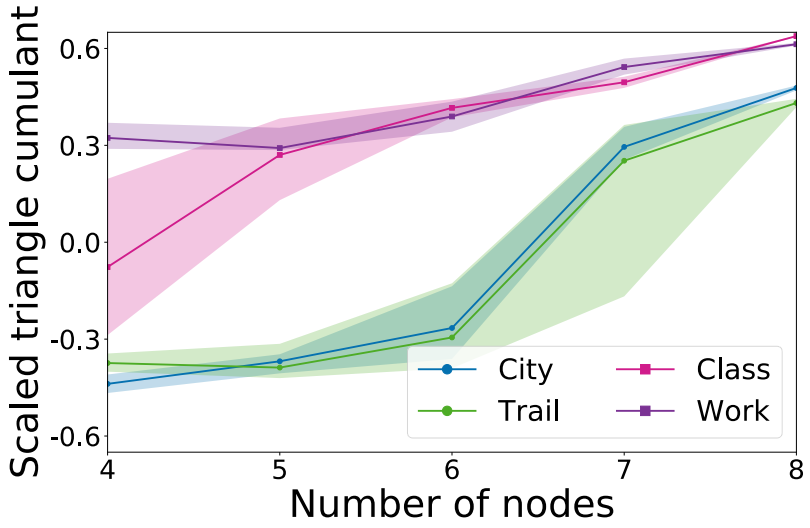


Figure 5.5: **Priors over social interactions favor triangles.** The methodology to recover the priors and obtain the error bars is the same as in Figure 5.1. The vertical axis corresponds to the scaled triangle cumulant (a more principled measure of triadic closure than the traditionally used clustering coefficient, see Figure 3.4 in Chapter 3). The scaled triangle cumulant for the cover stories in the social domain are clearly above those in the navigation domain. Thus, the social priors appear to reflect the well-known fact that social networks tend to exhibit triadic closure [229].

5.2.6 Priors have non-trivial domain-dependent graphical structure

In Figure 5.6, we study the generalization of the prior within and between domains, finding that priors more accurately describe other cover stories within the same domain than between domains.

We remark that if the diagonal were less than one, this would indicate that we do not have enough data, as the quadrupling of data is more important than the specialization to that particular cover story. In fact, it was quite vindicating to see the diagonal values rise as we ran more experiments.

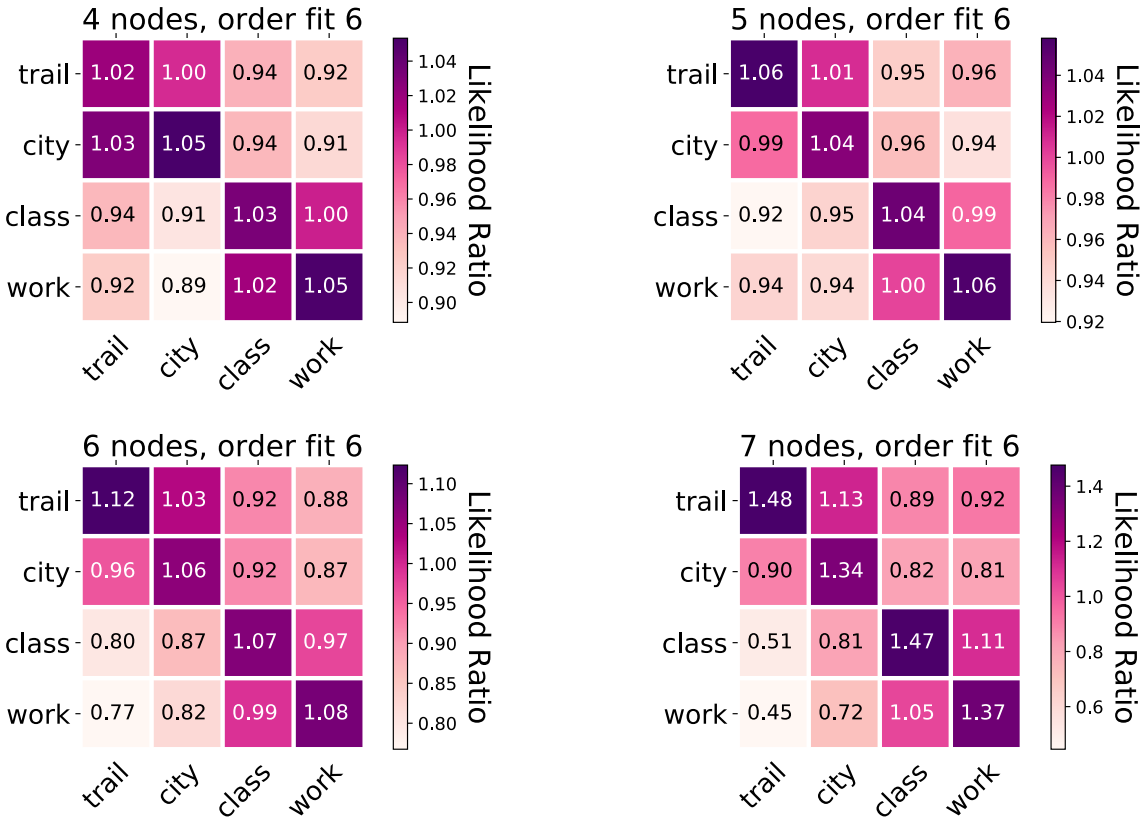


Figure 5.6: **Priors have non-trivial domain-dependent graphical structure.** For each of the 4×4 squares, we randomly partitioned the data from each cover story into a training set (80%) and a test set (20%). We fit a model (on the specified number of nodes and model order) to the training data from the cover stories listed in the row, and measured the log-likelihood of the test data in the column, given this model. We normalized this quantity by the log-likelihood of the column test data given a "non-specialized model" (a Bayesian MCMCP model fit to the aggregated training data over all cover stories), reporting the mean of 50 repetitions of this procedure. Hence, when a cell entry equals 1, the corresponding specialized model (from the row cover story) explains the corresponding data (from the column cover story) equally as well as the non-specialized model; when it is larger than 1, the specialized model explains the data better than the non-specialized model; and when it is smaller than 1, the opposite is true. For each square, the number is color-coded to aid in visualization, with darker colors corresponding to a better fit of the specialized model. If you take off your glasses, there is a convincing 2×2 block diagonal structure, reflective of the fact that priors from one cover story generalize better to cover stories in the same domain.

5.2.7 Higher order substructures reveal differences between domains

In Figure 5.7, we study how the order of the fitted prior affects generalization within and between domains. Recall from Chapter 3 that the order of the fit corresponds to its structural complexity, eg, first order constrains only the edge density, while sixth order constrains all subgraphs (ie, substructures) with 6 edges, thus corresponding to a full multinomial fit for graphs with four nodes. We find that lower fits do not differentiate between domains, whereas higher order do. This is reasonable, as one would expect more specific models to be able to better distinguish different domains.

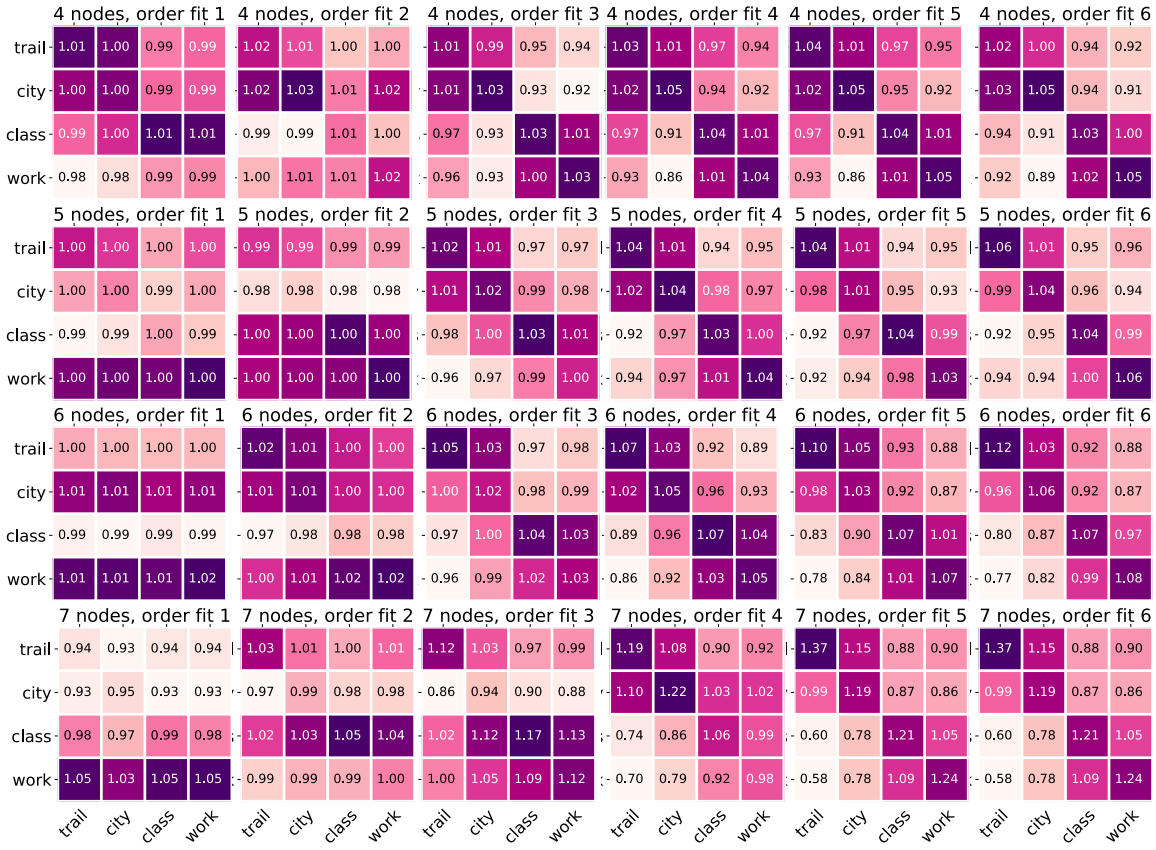


Figure 5.7: **Higher-order substructures reveal differences between domains.** Each 4×4 square is generated the same way as in Figure 5.6. The order of the fit increases from 1 (lowest complexity) on the left to 6 (higher complexity) on the right, and number of nodes varies vertically. Again, this image is best viewed sans spectacles. The specific numbers are not as important as the overall trend from left to right: for lower-order fits, specialization between domains is not particularly apparent; whereas higher orders offer sufficient resolution to distill the 2×2 block structure. This is in agreement with our previous results, where we find that lower-order substructures cannot differentiate between the cover stories (first-order statistics in Figure 5.1 and second-order in Figure 5.4). However, at third order, we find a difference between the social and navigation domains (Figure 5.5). Moreover, in model selection (see Section 5.5.1), higher-order fits (5 and 6) perform better than lower-order fits (for all cover stories up to 7 nodes), indicating that the priors are sensitive to higher-order substructures that differentiate between domains.

5.3 Discussion

In this work, we characterized human priors over the graphical structure of several social and navigation tasks, using a novel method to model distributions over graphs. We find that the priors have non-trivial graphical structure, requiring higher-order correlations between connections to describe. The sensitivity of the prior to lower-order substructures, such as edge density and correlations between pairs of edges, is quite similar between all cover stories. On the other hand, the sensitivity of the prior to higher-order substructures, such as triangles, appears to be more domain specific. In view of the hippocampus' well-understood role in encoding the structure of navigation tasks and its (albeit less well-understood) role in encoding structure in other tasks [214], further studies into how priors vary across domains could prove insightful in understanding the encoding of information in this region.

One particularly striking result requires no mention of the graphical structure, concerning only priors over the distribution of the edge density. For graphs smaller than $\sim$7–8 nodes (21–28 edges), the distribution in edge density is generally peaked, with a preference for sparsity that increases with number of nodes. The sparsity is to be somewhat expected, as many real networks are indeed sparse. However, representations of nearly-complete dense graphs can be sparsely represented via the graph complement. Indeed, the appearance (at $\sim$10 nodes) of a second peak at large edge density when the sparse peak becomes well-separated from its complement is suggestive. An intriguing possibility is that such a change might be indicative of an underlying difference in how our brain processes relational information.

One aspect of these experiments that warrants mentioning is the duality between the two navigation cover stories. In particular, both were suggestively planar, however, the connectivity was of two different forms. The trails are rather analogous to 1-vectors, while the boundaries between neighborhoods are analogous to 1-forms; while a "large" trail

implies a large separation between the two sites, a "large" boundary between neighborhoods implies the opposite. While one might expect that this dual representation would result in different priors, it is remarkable that these navigation cover stories led to priors that are at least as similar as those between the two social cover stories.

5.4 Experimental procedure

5.4.1 Experimental design

We (the author and Talmo Pereira) have developed an online experimental platform that smoothly allocates participants to the appropriate experimental chains in real time and gamified the experiment (see here for a demonstration video). The structure of the experiments was the same for all four cover stories.

The experiment begins with an introduction about the particular cover story and poses several questions to the participant to ensure their understanding. After a video demonstration of the interactive platform, each trial proceeds as follows: the shown relations appear on the top of the screen, the participant is presented with an interface where they can move the nodes and add edges. Once they submit their response, a question appears about which node they thought was the most important (asked in many different ways, to foster engagement).

To incentivize participants to give their true prior, they are told that there is an underlying truth, and that they are rewarded by correctly guessing the relations obscured. They were also told that the nodes were randomly sampled from a large underlying graph, and at each trial, they were asked their estimated size of this underlying graph.

Each participant performed 16 trials during the course of the experiment, consisting of 2 graphs for each number of nodes n, and with a varying number of relations shown $\#_{\text{shown}}$,

There were 6 different options for the precise sequences of trials a given participant was assigned to:

- $(n, \#_{\text{shown}})$: $(4,2)$, $(4,4)$, $(5,3)$, $(5,7)$, $(6,5)$, $(6,7)$, $(7,6)$, $(7,9)$, $(8,7)$, $(8,12)$, $(10,8)$, $(10,19)$, $(12,8)$, $(12,28)$, $(15,20)$, $(15,40)$;
- $(n, \#_{\text{shown}})$: $(4,4)$, $(4,2)$, $(5,7)$, $(5,3)$, $(6,7)$, $(6,5)$, $(7,9)$, $(7,6)$, $(8,12)$, $(8,7)$, $(10,19)$, $(10,8)$, $(12,28)$, $(12,8)$, $(15,40)$, $(15,20)$;
- $(n, \#_{\text{shown}})$: $(4,5)$, $(5,5)$, $(6,7)$, $(7,9)$, $(8,12)$, $(10,19)$, $(12,28)$, $(15,10)$, $(4,3)$, $(5,1)$, $(6,5)$, $(7,6)$, $(8,7)$, $(10,8)$, $(12,15)$, $(15,10)$;
- $(n, \#_{\text{shown}})$: $(4,3)$, $(5,1)$, $(6,5)$, $(7,6)$, $(8,7)$, $(10,8)$, $(12,15)$, $(15,10)$, $(4,5)$, $(5,5)$, $(6,7)$, $(7,9)$, $(8,12)$, $(10,19)$, $(12,28)$, $(15,10)$;
- $(n, \#_{\text{shown}})$: $(4,3)$, $(5,9)$, $(6,7)$, $(7,9)$, $(8,12)$, $(10,19)$, $(12,15)$, $(15,40)$, $(15,10)$, $(12,8)$, $(10,8)$, $(8,7)$, $(7,6)$, $(6,5)$, $(5,5)$, $(4,1)$;
- $(n, \#_{\text{shown}})$: $(4,1)$, $(5,5)$, $(6,5)$, $(7,6)$, $(8,7)$, $(10,8)$, $(12,8)$, $(15,10)$, $(15,40)$, $(12,15)$, $(10,19)$, $(8,12)$, $(7,9)$, $(6,7)$, $(5,9)$, $(4,3)$.

Participants were randomly assigned to one of these sequences.

There were two social cover stories:

1. **Class**: participants were asked to infer the friendships (relations) between students (nodes) in classroom.
2. **Work**: participants were asked to infer the friendships (relations) between coworkers (nodes) in a workplace.

And two navigation cover stories:

1. **Trail**: participants were asked to infer the trails (relations) between nature sites (nodes) in a nature park.

2. **City**: participants were asked to infer the borders (relations) between neighborhoods (nodes) in a city.

5.4.2 Exclusion criteria

As data collected online are often contaminated with participants that are not appropriately engaged in the experiment, we implemented a systematic method for excluding such data. In particular, we exclude all trials that met any of the following:

1. the participant took less than 3 seconds per shown relation to submit their answer;

2. the participant moved fewer than $\left\lceil \frac{n}{4} \right\rceil - 1$ nodes; or

3. if $\#_{\text{obs}} > 5$ and $f_{\text{add}} n > 1$

where $\#_{\text{obs}}$ is the number of relations obscured, f_{add} is the fraction of obscured relations that the participant chose to be edges in their answer, and n is the number of nodes. Moreover, if a given participant had fewer than 4 valid trials, we excluded all trials from this participant. These exclusion heuristics were judiciously chosen after observing the distribution of participants' responses.

The total number of participants and trials for each cover story (before the exclusion criteria were applied) are as follows:

- social (students and friendships) — 443 participants, 5340 trials

- social (coworkers and friendships) — 342 participants, 4317 trials

- navigation (nature sites and trails) — 359 participants, 4163 trials

- navigation (neighborhoods and borders) — 347 participants, 4013 trials

After the exclusion criteria were applied, the total number of participants and trials for each cover story are as follows:

- social (students and friendships) — 362 participants, 4795 trials
- social (coworkers and friendships) — 269 participants, 3675 trials
- navigation (nature sites and trails) — 299 participants, 3823 trials
- navigation (neighborhoods and borders) — 289 participants, 3569 trials

5.4.3 Scalability

To handle the combinatorial explosion inherent with an increasing number of nodes, we employed a method of subsampling graphs from $\mathrm{ER}_{n,1/2}$ with appropriate weights. This allowed us to fit distributions over graphs with 8 nodes (eg, in some cases, we obscured 21 relations, resulting in 2^{21} possible ways to complete the graph, necessitating such a method).

We remark that there were some difficulties in fitting distributions over graphs with 10 or more nodes (as opposed to the results presented over edges). Despite our best efforts in the fitting process, the distributions seemed to become concentrated on the complete graph. We are working on ways to make this fitting procedure more robust. However, we did manage to obtain reasonable priors in certain cases: see the video here for an animation of a simulation of a Markov Chain with a realistic prior over graphs with 12 nodes (derived from the cover story of friendships between students).

5.4.4 Anecdotal interlude

During Thanksgiving in 2018, we had the opportunity of observing two ideal participants: my two cousins-in-law (14 years and 17 years). They took over two and a half hours

(without a break) to do one of our experiments (as opposed to the average of $\sim$25 minutes for MTurk participants). They also provided valuable feedback. In particular, for the smaller graphs, their imagination was highly resonant with the relevant context. However, when interacting with the larger graphs ($\geq$10 nodes), they "just started making pretty designs". This might help explain some of the issues associated with fitting models to these data; if people become exhausted with the task of carefully determining all obscured relations, it is not unreasonable that they might choose a "simple" solution, the canonical ones being all connections or no connections (offering an arguably more boring explanation of the results in Figure 5.3).

5.4.5 Experimental considerations

A variety of pilot experiments provided valuable insights into how to make an engaging and intuitive experiment. For example, we had originally started without a visual interface for manipulating the graphs, so the participants had to hold the connections in their head and respond to a series of yes/no questions. We also added an extra question after each graph to make it more engaging (see Section 5.4.1). These and other improvements were incorporated into the final experiments (resulting in several MTurk workers sending personal emails about how fun the experiments were, and overall positive feedback in the post-experiment questionnaire).

An inherent limitation is the open-ended nature of our experiment; there is no ground truth, hence it is impossible to truly know whether participants were engaged in the task. To mitigate this, we did our best to select the data corresponding to engaged participants using a variety of measures (see Section 5.4.1).

5.5 Modeling

We fit the participants' collective data by maximizing the log-likelihood of the graphs the participants' submitted, given the Bayesian MCMCP model with a prior parametrized by equation (3.32) in Chapter 3. We used Newton's method for this optmization (which we implemented in python).

Specifically, the log-likelihood of the data is given by:

$$\mathcal{L}(\vec{\beta}) = \log \prod_t \frac{\Big(\big(\mathrm{ER}_{1/2}(G)\big)\big|_t\Big) \exp\Big(\vec{\beta}\cdot\vec{\mu}(G)\Big)}{\sum\limits_{G'\in\Omega}\Big(\big(\mathrm{ER}_{1/2}(G')\big)\big|_t\Big) \exp\Big(\vec{\beta}\cdot\vec{\mu}(G')\Big)} \tag{5.1}$$

where t is the index of the trial, G is the completed graph the participant chose (G' loops over all possible graphs Ω), $\vec{\beta}$ is the vector of parameters to be fitted, $\vec{\mu}(G)$ is the vector of moments of the graph G, and $\big(\mathrm{ER}_{1/2}(G)\big)\big|_t$ is the uniform distribution over the ways that the "partial graph" at trial t could be completed (ie, $\big(\mathrm{ER}_{1/2}(G)\big)\big|_t$ results from using a fair coin flip for all the relations that were not specified at trial t).

The gradient ($\vec{\nabla}\mathcal{L}$) entries are given by:

$$\frac{\partial\mathcal{L}}{\partial\beta_i} = \sum_t \left[\mu_i(G) - \frac{\sum\limits_{G'\in\Omega}\mu_i(G')\Big(\big(\mathrm{ER}_{1/2}(G')\big)\big|_t\Big) \exp\Big(\vec{\beta}\cdot\vec{\mu}(G')\Big)}{\sum\limits_{G'\in\Omega}\Big(\big(\mathrm{ER}_{1/2}(G')\big)\big|_t\Big) \exp\Big(\vec{\beta}\cdot\vec{\mu}(G')\Big)}\right]. \tag{5.2}$$

The Hessian ($\vec{\nabla}\vec{\nabla}\mathcal{L}$) entries are given by:

$$\frac{\partial^2 \mathcal{L}}{\partial \beta_i \partial \beta_j} = \sum_t \left[\frac{\left(\sum_{G' \in \Omega} \mu_i(G') \left(\left(\mathrm{ER}_{1/2}(G') \right) \big|_t \right) \exp\left(\vec{\beta} \cdot \vec{\mu}(G') \right) \right) \left(\sum_{G' \in \Omega} \mu_j(G') \left(\left(\mathrm{ER}_{1/2}(G') \right) \big|_t \right) \exp\left(\vec{\beta} \cdot \vec{\mu}(G') \right) \right)}{\left(\sum_{G' \in \Omega} \left(\left(\mathrm{ER}_{1/2}(G') \right) \big|_t \right) \exp\left(\vec{\beta} \cdot \vec{\mu}(G') \right) \right)^2} - \frac{\sum_{G' \in \Omega} \mu_i(G') \mu_j(G') \left(\left(\mathrm{ER}_{1/2}(G') \right) \big|_t \right) \exp\left(\vec{\beta} \cdot \vec{\mu}(G') \right)}{\sum_{G' \in \Omega} \left(\left(\mathrm{ER}_{1/2}(G') \right) \big|_t \right) \exp\left(\vec{\beta} \cdot \vec{\mu}(G') \right)} \right]. \tag{5.3}$$

The Newton iteration:

$$\vec{\beta}^{n+1} = \vec{\beta}^n - \left(\vec{\nabla}\vec{\nabla}\mathcal{L} \right)^{-1} \cdot \vec{\nabla}\mathcal{L}, \tag{5.4}$$

is iterated until machine precision.

5.5.1 Model selection

For each number of nodes and each condition, we selected the order of the model by cross-validation. However, one of the advantages of our parameterization is that fitting the data with a lower-order model accurately recovers lower-order cumulants, even when the data are generated by a prior of higher-order (Figure 2.11 in Chapter 2).

5.6 Future directions

An important follow-up question to the work presented in this chapter, which we have already started to address, is: *does our ability to reason about such graphs reflect our priors about their structure?* To address this question, we ask participants to solve computational

problems on small graphs. In particular, we have implemented and collected data from two experiments, inspired by canonical problems in the field of network science.

5.6.1 Euler experiment

For the domain of spatial navigation, the experiment is inspired by the "Seven Bridges of Königsberg Problem", whose solution was given by Euler in 1736 [78], and sparked the foundations of graph theory. Essentially, the goal is to find the shortest path that traverses every connection (trail) on the map at least once (where the initial and final locations are arbitrary). The video of our experiment can be found here.

5.6.2 Community detection experiment

For the domain of social interaction, the experiment is inspired by the community detection problem, which led to the explosive development of spectral graph theory [2]. Essentially, we asked participants to separate a friendship network into two groups of given sizes while breaking as few friendships as possible. The video of our experiment can be found here.

5.6.3 Analysis of these new experiments

These canonical problems have optimal solutions, which are computationally tractable for the size of the graphs used in these experiments. This provides well-defined measures of the relative accuracy of the participants' solutions. Our hypothesis is that people are able to more easily find optimal solutions for graphs that occur more frequently in their prior. To avoid obvious confounds, we compare groups of graphs with similar relevant graph theoretical properties.

5.6.4 Comparison with real networks

Another promising future direction is to compare the priors we quantified with the actual structure of analogous real networks, employing the framework presented in Chapter 3. For the social domain, there are several datasets available online (such as the ones used in Chapters 3 and 4). However, for the domain of spatial navigation, ideally one would encode trail maps of nature parks and neighborhood maps of cities into graphs.

5.6.5 Cultural effects

One of the author's closest friends, Livia Camargo Tavares Souza, does linguistics fieldwork on two aboriginal tribes in the Amazon rainforest in Brazil: the Yawanawa and the Xinane. Among the many fascinating cultural differences, their complex kinship is particularly interesting (and potentially relevant for priors over the social domain): while parallel cousins are forbidden from marriage, marriage between cross cousins is considered ideal. Moreover, the Xinane do not have a written language, and, until a few years ago, had not been contacted. The authorization to conduct such an experiment is in place, and we are working on a design that is both cross-cultural and in their native languages (both from the Pano family). Such an endeavor could offer insights into the generality of these priors.

Appendix A

Appendix for Graph Cumulants (Chapter 3)

A.1 Spectral motivation

In this section, we describe a spectral motivation for graph moments and their generalizations. We again begin with the simplest case, ie, simple graphs, then we generalize this spectral framework.

A.1.1 Simple graphs

Consider the problem of parameterizing a distribution over simple graphs with n nodes. We will represent such networks by ordered binary vectors of length $\binom{n}{2}$, where each entry represents an unordered pair of nodes, with 1 indicating that these two nodes are connected by an edge and 0 that they are not. Let $\mathcal{X}$ be the space of all such vectors. In general, the same graph can be represented by multiple vectors, and distributions over these graphs must give the same probability to all vectors that represent the same graph.

When parametrizing a distribution, it is often desirable that the distribution be "smooth", in the sense that similar graphs are assigned relatively similar probabilities. As our notion of similarity, we consider the "graph edit distance" [87], defined as the minimum number of edge changes (ie, additions or deletions) needed to transform one graph into the other. For example, the wedge (ie, 2-star) is distance 1 from the single edge and from the triangle, and distance 2 from the empty graph. Despite the fact that they have the same number of edges, the 3-star is distance 2 from the triangle, as one edge needs to be removed and another edge added (in a different position).

One common method for parameterizing smooth functions is via a Fourier representation. For example, a low-pass filter is equivalent to giving preference to the low frequency (ie, long wavelength) terms, where the location of the cutoff determines the smoothness of the output. This motivated us to consider an appropriate Laplacian that gives a similarly smooth parameterization over the space of networks.

To this end, we define a (weighted, directed) "edit graph" H_n, where its nodes represent distinct (ie, non-isomorphic) networks on n nodes. A directed edge from one node to another appears whenever the network it represents can be transformed into the other by adding or removing an edge at a single location. The weight of the edge is given by the number of locations that could be altered to effect this transformation (see Figure A.1 for the case of networks with 4 nodes). The Laplacian of this edit graph is defined as: $L_H = D_{\text{out}} - A_H^\top$, where D_{out} is the diagonal matrix of out-degrees, and $A_H = \{a_{ij}\}$ is the (asymmetric) adjacency matrix with entries a_{ij} equal to the weight of the transition from i to j.

The lowest eigenvalue of L_H is 0, and is associated with a left eigenvector that assigns the same value to each distinct network and a right eigenvector that is uniform over all representations of the networks (ie, over all binary vectors of length $\binom{n}{2}$), thus corresponding to the $\text{ER}_{n,1/2}$ distribution. The remainder of the spectrum contains additional structure.

Its support is the set of positive integers up to and including $\binom{n}{2}$, and each integer has a predictable degeneracy: the multiplicity of an eigenvalue $\lambda = r$ is equal to the number of distinct subgraphs with exactly r edges. This is not just a combinatorial coincidence; the span of left eigenvectors with eigenvalue up to and including r is precisely the span of subgraph counts up to order r. For example, a particular linear combination of the eigenvectors with eigenvalue $\lambda \leq 2$ gives a vector that is precisely the counts of wedges in the different networks, another combination gives the counts of pairs of edges that do not share an edge, and another gives the counts of edges.

The structure of this spectrum gives rise to a hierarchical parameterization of distributions over networks that is equivalent to our proposed family of hierarchical ERGMs, namely

$$p(G) \propto v_{d,0}(G) \ \exp\left(\sum_{i=1}^{r}\sum_{j=1}^{r_i} \beta_{i,j} v_{l,i,j}(G)\right), \tag{A.1}$$

where $v_{d,0}$ is the right eigenvector of L_H with eigenvalue 0 (ie, the $\text{ER}_{n,1/2}$ base distribution); $v_{l,i}$ is the set of left eigenvectors of L_H with eigenvalue i, and $v_{l,i,j}$ is one such vector; r_i is the number of eigenvectors with eigenvalue i; and $\beta_{i,j}$ are the coefficients to be fitted.

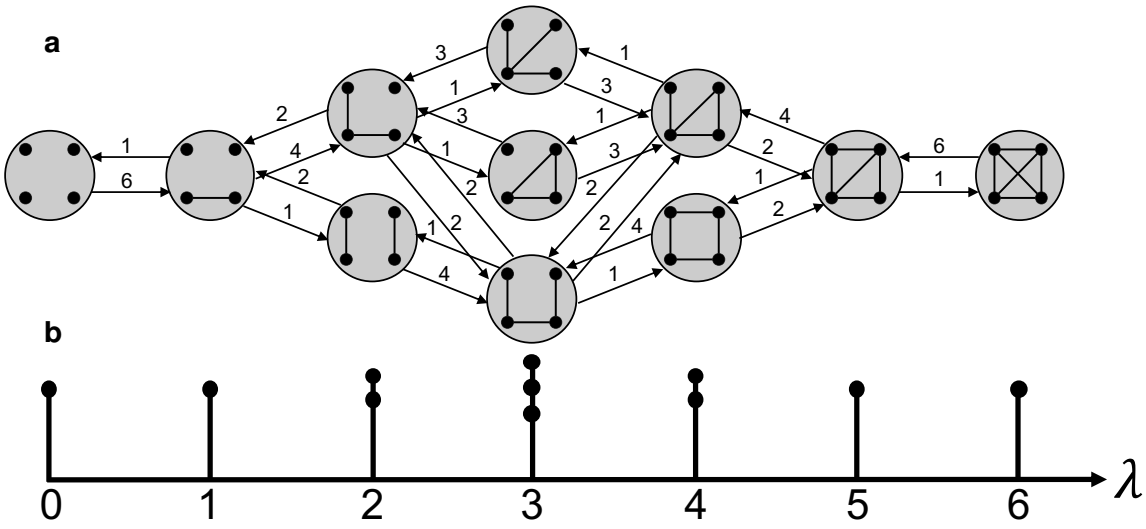


Figure A.1: **Spectral motivation for graph moments. a)** Directed weighted "edit graph" H_4 associated with networks with 4 nodes. The large nodes represent each of the 11 distinct (non-isormophic) networks. A directed edge from node u to node w indicates that network u can be transformed into network w by adding/removing an edge at a single location, where the edge weight is equal to the number of locations that could effect this transformation. Note that every node has an out-degree of 6, corresponding to the $\binom{4}{2}$ possible edge locations. **b)** Schematic of the spectrum of the Laplacian of H_4. The eigenvalues are integers ranging from 0 to $\binom{4}{2}$, and are degenerate with multiplicity equal to the number of distinct subgraphs (on at most 4 nodes) with that number of edges. This correspondence has a combinatorial interpretation: the span of left eigenvectors with eigenvalue $\lambda \leq r$ is precisely the span of subgraph counts up to order r.

A.1.2 Generalizations

We now generalize the concept of moments and cumulants for distributions over a set $\mathcal{X} \equiv \mathcal{A}^\ell$, ie, the vectors of length $\ell \in \mathbb{N}$ over the alphabet $\mathcal{A} = \{a_1, a_2, \ldots\}$, invariant with respect to a group $\mathbb{G}$ acting on this set $\mathcal{X}$. The action of $\mathbb{G}$ induces an equivalence relation on $\mathcal{X}$: $x \sim y \Leftrightarrow \exists g \in \mathbb{G} \mid x = g \circ y$, partitioning it into orbits. The distribution over $\mathcal{X}$ is then characterized by assigning a measure to each of these orbits. For example, for the case of simple graphs, the group $\mathbb{G}$ is S_n, acting by permuting the n nodes of a graph. The

set $\mathcal{X}$ consists of ordered binary vectors of length $\ell = \binom{n}{2}$, where each entry represents an unordered pair of nodes, with 1 indicating that these two nodes are connected by an edge and 0 that they are not. Distinct (ie, nonisomorphic) graphs are in different orbits, and all the elements in a given orbit correspond to the same graph, with the probability associated to that orbit distributed uniformly over all of its elements.

With this framework, we can construct the weighted directed "edit graph" described in the previous section for an arbitrary set $\mathcal{X}$ and a group $\mathbb{G}$ acting upon it. We can then use the spectrum of the Laplacian of the resulting edit graph to obtain the number of moments at each order. Again, the nodes of the edit graph are the orbits of $\mathcal{X}$, and a directed edge from one orbit to another appears whenever an element in the former can be transformed into one in the latter by changing just one of its entries. The weight of the edge is given by the number of distinct entries that could be altered to effect this transformation.

This abstraction allows one to apply this formalism to a variety of situations, and naturally encompasses the generalizations previously presented in this paper. For example, for unweighted directed networks with no self-loops, $\mathcal{X}$ is the set of all ordered binary vectors of length $\ell = 2\binom{n}{2}$, where each entry represents an *ordered* pair of nodes, with 1 indicating that there is an edge from the first node to the second and 0 that there is not. The group $\mathbb{G}$ is again S_n. As another example, consider the case of undirected unweighted bipartite networks, ie, every node has one of two possible "flavors" ("charm" and "strange"), and edges can occur only between nodes of different flavors. The set $\mathcal{X}$ consists of all ordered binary vectors of length $\ell = n_{\text{ch}} n_{\text{str}}$, where each entry represents a different unordered pair of nodes with different flavors, and 1 indicates that these two nodes are connected by an edge and 0 that they are not. The group $\mathbb{G}$ allows for permutations of nodes with the same flavor, namely $S_{n_{\text{ch}}} \times S_{n_{\text{str}}}$.

We now illustrate the versatility of this formalism by describing an additional generalization, namely, k-uniform hypergraphs, ie, a network with hyperedges that connect

k nodes. As in the graph case, the group $\mathbb{G}$ is the symmetric group S_n acting by permuting the n nodes. The set $\mathcal{X}$ consists of all ordered binary vectors of length $\ell = \binom{n}{k}$, where each entry represents an unordered set of k nodes, and a 1 indicates the presence of a hyperedge between them and 0 its absence. The orbits of this group action again partition the elements of $\mathcal{X}$ into the isomorphism classes of the hypergraphs they represent. The eigenvalues of the corresponding edit graph follow a similar pattern: associated to the eigenvalue of 0 is a right eigenvector that is uniform over distinct hypergraphs, and a left eigenvector that is uniform over all elements of $\mathcal{X}$. Likewise, for the remaining left eigenvectors, there is one eigenvector with associated eigenvalue of 1 that is linear in the number of hyperedges. At second order (associated eigenvalue 2), there are now *k* eigenvalues (for $n \geq 2k$), corresponding to the k ways that two hyperedges can relate (sharing any number from 0 to $k-1$ nodes).

A.2 Variance of unbiased graph cumulants

One can use a procedure similar to that for obtaining the unbiased graph cumulants (see Chapter 3 Section 3.10.5), now applied to the square of the unbiased cumulants. This results in a recursion relation for the variance of the unbiased cumulant. We illustrate this by outlining the derivation of the variance of the first-order unbiased cumulant $\hat{\kappa}_{1/}$, where $\hat{\kappa}_{1/} = \mu_{1/}$.

In general, the expressions for $\text{Var}(\hat{\kappa}_{rg})$ require moments up to order $2r$, a property analogous to that for real-valued random variables [83, 131]. Thus, for first order, we consider expressions of the form

$$\text{Var}(\mu_{1/}) = A(n)\mu_{2\wedge} + B(n)\mu_{1/}^2 + C(n)\mu_{1/}. \tag{A.2}$$

and again determine how the functions $A(n)$, $B(n)$, and $C(n)$ should change when removing a single random node. This provides a recursion relation, and determines these functions up to a few constants, which are now chosen to give zero variance as $n \to \infty$.

Assume that we started with an initial graph on $N \gg n$ nodes, and have subsampled it down to n nodes, thereby accruing a variance $\text{Var}(\mu_{1/})$ in the first moment. We consider the additional variance obtained by randomly removing another node . The change in the first moment when removing a node i is:

$$\begin{aligned}\mu_{1/} \to \mu'_{1/} &= \frac{1}{\binom{n-1}{2}}\left(\binom{n}{2}\mu_{1/} - d_i\right) \\ &= \mu_{1/} + \frac{2}{n-2}\mu_{1/} - \frac{2}{(n-1)(n-2)}d_i \end{aligned} \tag{A.3}$$

Thus, $\text{Var}(\mu'_{1/}) = \text{Var}(\mu_{1/}) + \Delta\text{Var}$, where

$$\begin{aligned}\Delta\text{Var} &= \left\langle \left(\frac{2}{n-2}\mu_{1/} - \frac{2}{(n-1)(n-2)}d_i\right)^2 \right\rangle_{n\leftarrow N} \\ &= \frac{4}{(n-2)^2}\langle \mu_{1/}^2 \rangle_{n\leftarrow N} - \frac{8}{(n-1)(n-2)^2}\langle \mu_{1/} d_i \rangle_{n\leftarrow N} \\ &\quad + \frac{4}{(n-1)^2(n-2)^2}\langle d_i^2 \rangle_{n\leftarrow N}, \end{aligned} \tag{A.4}$$

where the angle brackets $\langle\cdot\rangle_{n\leftarrow N}$ denote the expectation over graphs with n nodes obtained by randomly subsampling from the original N nodes.

Starting with the last term, recall from Chapter 3 Section 3.10.5 that $\langle d_i \rangle = \frac{2c_{/}}{n}$ and $\langle d_i^2 \rangle = \frac{2c_{\wedge}}{n} + \langle d_i \rangle$, where angle brackets without a subscript denotes expectation with respect

to a single graph. Thus, we have

$$\begin{aligned}\langle d_i^2 \rangle_{n \leftarrow N} &= \left\langle \frac{2c_{\wedge}}{n} + \frac{2c_{/}}{n} \right\rangle_{n \leftarrow N} \\ &= (n-1)(n-2)\langle \mu_{2\wedge} \rangle_{n \leftarrow N} + (n-1)\langle \mu_{1/} \rangle_{n \leftarrow N} \\ &= (n-1)(n-2)\hat{\mu}_{2\wedge} + (n-1)\hat{\mu}_{1/}, \end{aligned} \tag{A.5}$$

where $\hat{\mu}_{rg}$ denote the moments of the graph with N nodes as $N \to \infty$, as moments are preserved in expectation under subsampling of nodes.

For the middle term, as the nodes are removed randomly,

$$\begin{aligned}\langle \mu_{1/} d_i \rangle_{n \leftarrow N} &= \left\langle \mu_{1/} \frac{2\binom{n}{2}\mu_{1/}}{n} \right\rangle_{n \leftarrow N} \\ &= (n-1)\langle \mu_{1/}^2 \rangle_{n \leftarrow N}, \end{aligned} \tag{A.6}$$

and thus partially cancels with the first term, giving $-\frac{4}{(n-2)^2}\langle \mu_{1/}^2 \rangle_{n \leftarrow N}$.

Computing this first term requires the inclusion of the current amount of variance:

$$\begin{aligned}\langle \mu_{1/}^2 \rangle_{n \leftarrow N} &= \langle \mu_{1/} \rangle_{n \leftarrow N}^2 + \text{Var}(\mu_{1/}) \\ &= \hat{\mu}_{1/}^2 + \text{Var}(\mu_{1/}). \end{aligned} \tag{A.7}$$

Substituting these into the expression for $\text{Var}(\mu'_{1/})$ yields a recursion relation:

$$\begin{aligned}
\text{Var}(\mu'_{1/}) &= \text{Var}(\mu_{1/}) - \frac{4}{(n-2)^2}\langle \mu^2_{1/}\rangle_{n\leftarrow N} + \frac{4}{(n-1)^2(n-2)^2}\langle d_i^2\rangle_{n\leftarrow N} \\
&= \text{Var}(\mu_{1/}) - \frac{4}{(n-2)^2}\left(\hat{\mu}^2_{1/} + \text{Var}(\mu_{1/})\right) \\
&\quad + \frac{4}{(n-1)^2(n-2)^2}\left((n-1)(n-2)\hat{\mu}_{2\wedge} + (n-1)\hat{\mu}_{1/}\right) \\
&= \text{Var}(\mu_{1/}) - \frac{4}{(n-2)^2}\left(\hat{\mu}^2_{1/} + \text{Var}(\mu_{1/})\right) \\
&\quad + \frac{4}{(n-1)(n-2)}\hat{\mu}_{2\wedge} + \frac{4}{(n-1)(n-2)^2}\hat{\mu}_{1/} \\
&= \left(1 - \frac{4}{(n-2)^2}\right)\text{Var}(\mu_{1/}) - \frac{4}{(n-2)^2}\hat{\mu}^2_{1/} \\
&\quad + \frac{4}{(n-1)(n-2)}\hat{\mu}_{2\wedge} + \frac{4}{(n-1)(n-2)^2}\hat{\mu}_{1/}. \qquad \text{(A.8)}
\end{aligned}$$

The solution to this recursion relation (along with the condition that $\text{Var}(\mu_{1/}) \to 0$ as $n \to \infty$) gives

$$\begin{aligned}
\text{Var}(\hat{\kappa}_{1/}) &= \frac{1}{\binom{n}{2}}\Big(\hat{\mu}_{1/} + (3-2n)\hat{\mu}^2_{1/} + 2(n-2)\hat{\mu}_{2\wedge}\Big) \\
&= \frac{1}{\binom{n}{2}}\Big(\mu_{1/}(1-\mu_{1/}) + 2(n-2)\hat{\kappa}_{2\wedge}\Big). \qquad \text{(A.9)}
\end{aligned}$$

A.3 Formulas for graph moments and graph cumulants

Here, we present expressions for computing some relevant graph moments of a network G (with n nodes) using only the counts c_g of the *connected* subgraphs g in G, and their conversion to graph cumulants in the $n \to \infty$ limit. While we do not display all expressions here, we have automated their derivation to arbitrary order, and the results up to sixth order are included explicitly in the code associated with this paper.

A.3.1 Simple graphs

For the simplest type of network (ie, undirected, unweighted, with no self-loops or multiple edges), we enumerate here the expressions for all third-order graph cumulants, as well as those that are necessary for computing the (sixth-order) cumulant associated with the complete graph in four nodes.

These graph moments are given by:

$$\mu_{1/} = \frac{c_{/}}{\binom{n}{2}} \tag{A.10}$$

$$\mu_{2\wedge} = \frac{c_{\wedge}}{3\binom{n}{3}} \tag{A.11}$$

$$\mu_{2//} = \frac{c_{//}}{3\binom{n}{4}} = \frac{\binom{c_{/}}{2} - c_{\wedge}}{3\binom{n}{4}} \tag{A.12}$$

$$\mu_{3\triangle} = \frac{c_{\triangle}}{\binom{n}{3}} \tag{A.13}$$

$$\mu_{3\curlywedge} = \frac{c_{\curlywedge}}{4\binom{n}{4}} \tag{A.14}$$

$$\mu_{3\sqcap} = \frac{c_{\sqcap}}{12\binom{n}{4}} \tag{A.15}$$

$$\mu_{3/\wedge} = \frac{c_{/\wedge}}{30\binom{n}{5}} = \frac{c_{\wedge}(c_{/} - 2) - 3c_{\triangle} - 3c_{\curlywedge} - 2c_{\sqcap}}{30\binom{n}{5}} \tag{A.16}$$

$$\mu_{3///} = \frac{c_{///}}{15\binom{n}{6}} = \frac{\binom{c_{/}}{3} - c_{\triangle} - c_{\curlywedge} - c_{\sqcap} - c_{/\wedge}}{15\binom{n}{6}} \tag{A.17}$$

$$\mu_{4\triangle\!\!/} = \frac{c_{\triangle\!\!/}}{12\binom{n}{4}} \tag{A.18}$$

$$\mu_{4\square} = \frac{c_{\square}}{3\binom{n}{4}} \tag{A.19}$$

$$\mu_{5\boxslash} = \frac{c_{\boxslash}}{6\binom{n}{4}} \tag{A.20}$$

$$\mu_{6\boxtimes} = \frac{c_{\boxtimes}}{\binom{n}{4}} \tag{A.21}$$

The corresponding graph cumulants (in the $n \to \infty$ limit) are given by:

$$\kappa_{1/} = \mu_{1/} \tag{A.22}$$

$$\kappa_{2\wedge} = \mu_{2\wedge} - \mu_{1/}^2 \tag{A.23}$$

$$\kappa_{2//} = \mu_{2//} - \mu_{1/}^2 \tag{A.24}$$

$$\kappa_{3\triangle} = \mu_{3\triangle} - 3\mu_{2\wedge}\mu_{1/} + 2\mu_{1/}^3 \tag{A.25}$$

$$\kappa_{3\curlywedge} = \mu_{3\curlywedge} - 3\mu_{2\wedge}\mu_{1/} + 2\mu_{1/}^3 \tag{A.26}$$

$$\kappa_{3\sqcap} = \mu_{3\sqcap} - 2\mu_{2\wedge}\mu_{1/} - \mu_{2//}\mu_{1/} + 2\mu_{1/}^3 \tag{A.27}$$

$$\kappa_{3/\wedge} = \mu_{3/\wedge} - \mu_{2\wedge}\mu_{1/} - 2\mu_{2//}\mu_{1/} + 2\mu_{1/}^3 \tag{A.28}$$

$$\kappa_{3///} = \mu_{3///} - 3\mu_{2//}\mu_{1/} + 2\mu_{1/}^3 \tag{A.29}$$

$$\begin{aligned}\kappa_{4\triangle/} = {} & \mu_{4\triangle/} - \mu_{3\triangle}\mu_{1/} - \mu_{3\curlywedge}\mu_{1/} - 2\mu_{3\sqcap}\mu_{1/} \\ & - 2\mu_{2\wedge}^2 - \mu_{2\wedge}\mu_{2//} + 10\mu_{2\wedge}\mu_{1/}^2 + 2\mu_{2//}\mu_{1/}^2 - 6\mu_{1/}^4\end{aligned} \tag{A.30}$$

$$\kappa_{4\square} = \mu_{4\square} - 4\mu_{3\sqcap}\mu_{1/} - 2\mu_{2\wedge}^2 - \mu_{2//}^2 + 8\mu_{2\wedge}\mu_{1/}^2 + 4\mu_{2//}\mu_{1/}^2 - 6\mu_{1/}^4 \tag{A.31}$$

$$\begin{aligned}\kappa_{5\boxslash} = {} & \mu_{5\boxslash} - 4\mu_{4\triangle/}\mu_{1/} - \mu_{4\square}\mu_{1/} - 2\mu_{3\triangle}\mu_{2\wedge} - 2\mu_{3\curlywedge}\mu_{2\wedge} - 4\mu_{3\sqcap}\mu_{2\wedge} - 2\mu_{3\sqcap}\mu_{2//} \\ & + 4\mu_{3\triangle}\mu_{1/}^2 + 4\mu_{3\curlywedge}\mu_{1/}^2 + 8\mu_{3\sqcap}\mu_{1/}^2 + 4\mu_{3\sqcap}\mu_{1/}^2 \\ & + 20\mu_{2\wedge}^2\mu_{1/} + 8\mu_{2\wedge}\mu_{2//}\mu_{1/} + 2\mu_{2//}^2\mu_{1/} \\ & - 48\mu_{2\wedge}\mu_{1/}^3 - 12\mu_{2//}\mu_{1/}^3 + 24\mu_{1/}^5\end{aligned} \tag{A.32}$$

$$\begin{aligned}\kappa_{6\boxtimes} = \mu_{6\boxtimes} &- 6\mu_{5\boxslash}\mu_{1/} - 12\mu_{4\triangle/}\mu_{2\wedge} - 3\mu_{4\square}\mu_{2//} + 24\mu_{4\triangle/}\mu_{1/}^2 + 6\mu_{4\square}\mu_{1/}^2 \\ &- 4\mu_{3\triangle}\mu_{3\curlywedge} - 6\mu_{3\sqcap}^2 \\ &+ 24\mu_{3\triangle}\mu_{2\wedge}\mu_{1/} + 24\mu_{3\curlywedge}\mu_{2\wedge}\mu_{1/} + 48\mu_{3\sqcap}\mu_{2\wedge}\mu_{1/} + 24\mu_{3\sqcap}\mu_{2//}\mu_{1/} \\ &- 24\mu_{3\triangle}\mu_{1/}^3 - 24\mu_{3\curlywedge}\mu_{1/}^3 - 72\mu_{3\sqcap}\mu_{1/}^3 \\ &+ 12\mu_{2\wedge}^3 + 15\mu_{2\wedge}^2\mu_{2//} + 3\mu_{2//}^3 \\ &- 153\mu_{2\wedge}^2\mu_{1/}^2 - 90\mu_{2\wedge}\mu_{2//}\mu_{1/}^2 - 27\mu_{2//}^2\mu_{1/}^2 \\ &+ 288\mu_{2\wedge}\mu_{1/}^4 + 72\mu_{2//}\mu_{1/}^4 - 120\mu_{1/}^6 \end{aligned} \tag{A.33}$$

A.3.2 Directed graphs

We now enumerate the expressions necessary for computing the graph cumulants of all directed subgraphs on three nodes, including the sixth-order graph cumulant associated with the complete directed triad.

These directed graph moments are given by:

$$\mu_{1\nearrow} = \frac{c_{1\nearrow}}{2\binom{n}{2}} \tag{A.34}$$

$$\mu_{2\nearrow\!\!\nearrow} = \frac{c_{2\nearrow\!\!\nearrow}}{3\binom{n}{3}} \tag{A.35}$$

$$\mu_{2\nearrow\!\!\rightarrow} = \frac{c_{2\nearrow\!\!\rightarrow}}{3\binom{n}{3}} \tag{A.36}$$

$$\mu_{2\nearrow\!\!\leftarrow} = \frac{c_{2\nearrow\!\!\leftarrow}}{6\binom{n}{3}} \tag{A.37}$$

$$\mu_{2\not\rightarrow} = \frac{c_{2\not\rightarrow}}{\binom{n}{2}} \tag{A.38}$$

$$\mu_{3\not\nearrow\!\!\nearrow} = \frac{c_{3\not\nearrow\!\!\nearrow}}{6\binom{n}{3}} \tag{A.39}$$

$$\mu_{3\not\nearrow\!\!\rightarrow} = \frac{c_{3\not\nearrow\!\!\rightarrow}}{6\binom{n}{3}} \tag{A.40}$$

$$\mu_{3\triangle} = \frac{c_{3\triangle}}{6\binom{n}{3}} \tag{A.41}$$

$$\mu_{3\triangle} = \frac{c_{3\triangle}}{2\binom{n}{3}} \tag{A.42}$$

$$\mu_{4\triangle} = \frac{c_{4\triangle}}{3\binom{n}{3}} \tag{A.43}$$

$$\mu_{4\triangle} = \frac{c_{4\triangle}}{3\binom{n}{3}} \tag{A.44}$$

$$\mu_{4\triangle} = \frac{c_{4\triangle}}{6\binom{n}{3}} \tag{A.45}$$

$$\mu_{4\not\nearrow\!\!\nearrow} = \frac{c_{4\not\nearrow\!\!\nearrow}}{3\binom{n}{3}} \tag{A.46}$$

$$\mu_{5\not\triangle} = \frac{c_{5\not\triangle}}{6\binom{n}{3}} \tag{A.47}$$

$$\mu_{6\not\triangle} = \frac{c_{6\not\triangle}}{\binom{n}{3}} \tag{A.48}$$

The corresponding directed graph cumulants (in the $n \to \infty$ limit) are given by:

$$\kappa_{1,} = \mu_{1,} \tag{A.49}$$

$$\kappa_{2,} = \mu_{2,} - \mu_{1,}^2 \tag{A.50}$$

$$\kappa_{2,} = \mu_{2,} - \mu_{1,}^2 \tag{A.51}$$

$$\kappa_{2,} = \mu_{2,} - \mu_{1,}^2 \tag{A.52}$$

$$\kappa_{2,} = \mu_{2,} - \mu_{1,}^2 \tag{A.53}$$

$$\kappa_{3,} = \mu_{3,} - \mu_{2,}\mu_{1,} - \mu_{2,}\mu_{1,} - \mu_{2,}\mu_{1,} + 2\mu_{1,}^3 \tag{A.54}$$

$$\kappa_{3,} = \mu_{3,} - \mu_{2,}\mu_{1,} - \mu_{2,}\mu_{1,} - \mu_{2,}\mu_{1,} + 2\mu_{1,}^3 \tag{A.55}$$

$$\kappa_{3,} = \mu_{3,} - \mu_{2,}\mu_{1,} - \mu_{2,}\mu_{1,} - \mu_{2,}\mu_{1,} + 2\mu_{1,}^3 \tag{A.56}$$

$$\kappa_{3,} = \mu_{3,} - 3\mu_{2,}\mu_{1,} + 2\mu_{1,}^3 \tag{A.57}$$

$$\begin{aligned}\kappa_{4,} = {} & \mu_{4,} - 2\mu_{3,}\mu_{1,} - 2\mu_{3,}\mu_{1,} - \mu_{2,}^2 - \mu_{2,}^2 - \mu_{2,}\mu_{2,} \\ & + 2\mu_{2,}\mu_{1,}^2 + 4\mu_{2,}\mu_{1,}^2 + 4\mu_{2,}\mu_{1,}^2 + 2\mu_{2,}\mu_{1,}^2 - 6\mu_{1,}^4\end{aligned} \tag{A.58}$$

$$\begin{aligned}\kappa_{4,} = {} & \mu_{4,} - 2\mu_{3,}\mu_{1,} - 2\mu_{3,}\mu_{1,} - \mu_{2,}^2 - \mu_{2,}^2 - \mu_{2,}\mu_{2,} \\ & + 4\mu_{2,}\mu_{1,}^2 + 2\mu_{2,}\mu_{1,}^2 + 4\mu_{2,}\mu_{1,}^2 + 2\mu_{2,}\mu_{1,}^2 - 6\mu_{1,}^4\end{aligned} \tag{A.59}$$

$$\begin{aligned}\kappa_{4,} = {} & \mu_{4,} - \mu_{3,}\mu_{1,} - \mu_{3,}\mu_{1,} - \mu_{3,}\mu_{1,} - \mu_{2,}\mu_{2,} - \mu_{2,}\mu_{2,} - \mu_{2,}\mu_{2,} \\ & + 2\mu_{2,}\mu_{1,}^2 + 2\mu_{2,}\mu_{1,}^2 + 6\mu_{2,}\mu_{1,}^2 + 2\mu_{2,}\mu_{1,}^2 - 6\mu_{1,}^4\end{aligned} \tag{A.60}$$

$$\begin{aligned}\kappa_{4,} = {} & \mu_{4,} - 2\mu_{3,}\mu_{1,} - 2\mu_{3,}\mu_{1,} - \mu_{2,}\mu_{2,} - \mu_{2,}^2 - \mu_{2,}^2 \\ & + 2\mu_{2,}\mu_{1,}^2 + 2\mu_{2,}\mu_{1,}^2 + 4\mu_{2,}\mu_{1,}^2 + 4\mu_{2,}\mu_{1,}^2 - 6\mu_{1,}^4\end{aligned} \tag{A.61}$$

$$\kappa_{5\clubsuit} = \mu_{5\clubsuit} - \mu_{4\spadesuit}\mu_{1\prime} - \mu_{4\spadesuit}\mu_{1\prime} - 2\mu_{4\spadesuit}\mu_{1\prime} - \mu_{4\heartsuit}\mu_{1\prime} -$$
$$- \mu_{3\heartsuit}\mu_{2\wedge} - \mu_{3\heartsuit}\mu_{2\wedge} - \mu_{3\heartsuit}\mu_{2\theta} - \mu_{3\heartsuit}\mu_{2\wedge} - \mu_{3\heartsuit}\mu_{2\wedge}$$
$$- \mu_{3\heartsuit}\mu_{2\theta} - \mu_{3\spadesuit}\mu_{2\wedge} - \mu_{3\spadesuit}\mu_{2\wedge} - \mu_{3\spadesuit}\mu_{2\wedge}$$
$$+ 6\mu_{3\heartsuit}\mu_{1\prime}^2 + 6\mu_{3\heartsuit}\mu_{1\prime}^2 + 9\mu_{3\spadesuit}\mu_{1\prime}^2 + 2\mu_{3\spadesuit}\mu_{1\prime}^2$$
$$+ 2\mu_{2\wedge}^2 + 2\mu_{2\wedge}^2 + 6\mu_{2\wedge}^2 + 2\mu_{2\theta}^2 + 2\mu_{2\wedge}\mu_{2\wedge}$$
$$+ 4\mu_{2\wedge}\mu_{2\wedge} + 4\mu_{2\wedge}\mu_{2\wedge} + 2\mu_{2\wedge}\mu_{2\theta} + 2\mu_{2\wedge}\mu_{2\theta} + 4\mu_{2\wedge}\mu_{2\theta}$$
$$- 12\mu_{2\wedge}\mu_{1\prime}^3 - 12\mu_{2\wedge}\mu_{1\prime}^3 - 24\mu_{2\wedge}\mu_{1\prime}^3 - 12\mu_{2\theta}\mu_{1\prime}^3 + 24\mu_{1\prime}^5 \quad \text{(A.62)}$$

$$\kappa_{6\clubsuit} = \mu_{6\clubsuit} - 6\mu_{5\clubsuit}\mu_{1\prime} - 3\mu_{4\spadesuit}\mu_{2\wedge} - 3\mu_{4\spadesuit}\mu_{2\wedge} - 6\mu_{4\spadesuit}\mu_{2\wedge} - 3\mu_{4\heartsuit}\mu_{2\theta}$$
$$+ 6\mu_{4\spadesuit}\mu_{1\prime}^2 + 6\mu_{4\spadesuit}\mu_{1\prime}^2 + 12\mu_{4\spadesuit}\mu_{1\prime}^2 + 6\mu_{4\heartsuit}\mu_{1\prime}^2 - 3\mu_{3\heartsuit}^2 - 3\mu_{3\heartsuit}^2 - 3\mu_{3\spadesuit}^2 - \mu_{3\spadesuit}^2$$
$$+ 12\mu_{3\heartsuit}\mu_{2\wedge}\mu_{1\prime} + 12\mu_{3\heartsuit}\mu_{2\wedge}\mu_{1\prime} + 12\mu_{3\heartsuit}\mu_{2\theta}\mu_{1\prime} + 12\mu_{3\heartsuit}\mu_{2\wedge}\mu_{1\prime} + 12\mu_{3\heartsuit}\mu_{2\wedge}\mu_{1\prime}$$
$$+ 12\mu_{3\heartsuit}\mu_{2\theta}\mu_{1\prime} + 12\mu_{3\spadesuit}\mu_{2\wedge}\mu_{1\prime} + 12\mu_{3\spadesuit}\mu_{2\wedge}\mu_{1\prime} + 12\mu_{3\spadesuit}\mu_{2\wedge}\mu_{1\prime} + 12\mu_{3\spadesuit}\mu_{2\wedge}\mu_{1\prime}$$
$$- 36\mu_{3\heartsuit}\mu_{1\prime}^3 - 36\mu_{3\heartsuit}\mu_{1\prime}^3 - 36\mu_{3\spadesuit}\mu_{1\prime}^3 - 12\mu_{3\spadesuit}\mu_{1\prime}^3$$
$$+ 2\mu_{2\wedge}^3 + 6\mu_{2\wedge}\mu_{2\wedge}\mu_{2\theta} + 6\mu_{2\wedge}\mu_{2\wedge}^2 + 2\mu_{2\wedge}^3 + 6\mu_{2\wedge}\mu_{2\wedge}^2 + 6\mu_{2\wedge}^2\mu_{2\theta} + 2\mu_{2\theta}^3$$
$$- 18\mu_{2\wedge}^2\mu_{1\prime}^2 - 18\mu_{2\wedge}\mu_{2\wedge}\mu_{1\prime}^2 - 36\mu_{2\wedge}\mu_{2\wedge}\mu_{1\prime}^2 - 18\mu_{2\wedge}\mu_{2\theta}\mu_{1\prime}^2 - 18\mu_{2\wedge}^2\mu_{1\prime}^2$$
$$- 36\mu_{2\wedge}\mu_{2\wedge}\mu_{1\prime}^2 - 18\mu_{2\wedge}\mu_{2\theta}\mu_{1\prime}^2 - 54\mu_{2\wedge}^2\mu_{1\prime}^2 - 36\mu_{2\wedge}\mu_{2\theta}\mu_{1\prime}^2 - 18\mu_{2\theta}^2\mu_{1\prime}^2$$
$$+ 60\mu_{2\wedge}\mu_{1\prime}^4 + 60\mu_{2\wedge}\mu_{1\prime}^4 + 120\mu_{2\wedge}\mu_{1\prime}^4 + 60\mu_{2\theta}\mu_{1\prime}^4 - 120\mu_{1\prime}^6 \quad \text{(A.63)}$$

Appendix B

Appendix for Spectral Graph Reduction (Chapter 4)

B.1 Perturbations to eigenvalues of the Laplacian pseudoinverse

We first provide the lowest order change in the eigenvalues of $\boldsymbol{L}_G^\dagger$. Then, we show how it relates to the Frobenius norm of the perturbation, explicitly relating it to our graph reduction algorithm.

Consider an inverse Laplacian $\boldsymbol{L}^\dagger$, which has an eigenvector $\vec{x}$ (without loss of generality, assume $\|\vec{x}\|_2 = 1$) with associated eigenvalue λ. If we perturb $\boldsymbol{L}^\dagger$ by $\varepsilon\boldsymbol{\Delta L}^\dagger$, we can solve for the first-order corrections to this "eigenpair" as follows:

$$(\boldsymbol{L}^\dagger + \varepsilon\boldsymbol{\Delta L}^\dagger)(\vec{x} + \varepsilon\Delta\vec{x}) = (\lambda + \varepsilon\Delta\lambda)(\vec{x} + \varepsilon\Delta\vec{x})$$

$$(\boldsymbol{L}^\dagger - \lambda)\Delta\vec{x} = (\Delta\lambda - \boldsymbol{\Delta L}^\dagger)\vec{x} + \mathcal{O}(\varepsilon),$$

where we have used $\boldsymbol{L}^\dagger \vec{x} = \lambda \vec{x}$.

Taking the inner product with $\vec{x}$ gives

$$\begin{aligned}
\vec{x}^\top (\boldsymbol{L}^\dagger - \lambda) \Delta\vec{x} &= \vec{x}^\top (\Delta\lambda - \boldsymbol{\Delta L}^\dagger) \vec{x} \\
\Delta\vec{x}^\top (\boldsymbol{L}^\dagger - \lambda) \vec{x} &= \Delta\lambda \vec{x}^\top \vec{x} - \vec{x}^\top \boldsymbol{L}^\dagger \vec{x} \\
0 &= \Delta\lambda - \vec{x}^\top \boldsymbol{L}^\dagger \vec{x},
\end{aligned}$$

where we have used the symmetry of $\boldsymbol{L}^\dagger$.

This provides the first-order correction to the eigenvalues of $\boldsymbol{L}^\dagger + \varepsilon \boldsymbol{\Delta L}^\dagger$:

$$\Delta\lambda = x^\top \boldsymbol{\Delta L}^\dagger \vec{x}. \tag{B.1}$$

The correction in (B.1) is controlled by the operator norm of $\boldsymbol{\Delta L}^\dagger$,

$$\Delta\lambda = \vec{x}^\top \boldsymbol{\Delta L}^\dagger \vec{x} \leq \sup_{\|\vec{x}\|_2 = 1} \left\| \boldsymbol{\Delta L}^\dagger \vec{x} \right\|_2 = \left\| \boldsymbol{\Delta L}^\dagger \right\|_{\text{op}}.$$

Thus, bounding the first-order correction to the eigenvalues,

$$|\Delta\lambda| \leq \left\| \boldsymbol{\Delta L}^\dagger \right\|_{\text{op}}. \tag{B.2}$$

As the operator norm is bounded by the Frobenius norm (by the Cauchy–Schwarz inequality), the estimated error (ie, $\sum \mathbb{E}\left[\left\| \boldsymbol{\Delta L}^\dagger \right\|_{\text{F}}^2 \right]$, equation (4.1)) provides a conservative bound for the change in the eigenvalues of the resulting reduced graph.

Moreover, as the bound is the same for all eigenvalues of the perturbed $\boldsymbol{L}^\dagger$, the *relative* error is more tightly bounded for its largest eigenvalues (those associated with large-scale structure).

B.2 Applications to graph visualization

Data visualization is an important (and aesthetically pleasing) application of graph reduction. As such, we generated videos of our algorithm reducing several real-world datasets. Figure B.1 displays several stages of our algorithm applied to a temporal social network. A video of this reduction can be found here; an application to an airport network (a case with both geometric and scale-free aspects) can be found here; an application to the European road network can be found here, and a reduction of a "hierarchical meta-graph" can be found here.[1]

[1] Explicit urls for the non-hyperlinked:
youtube.com/watch?v=qqLJclVUML8; youtube.com/watch?v=tXUr6RBRaEI;
youtube.com/watch?v=UVhT0y4Uae0; and youtube.com/watch?v=i3u4kkxMK40.

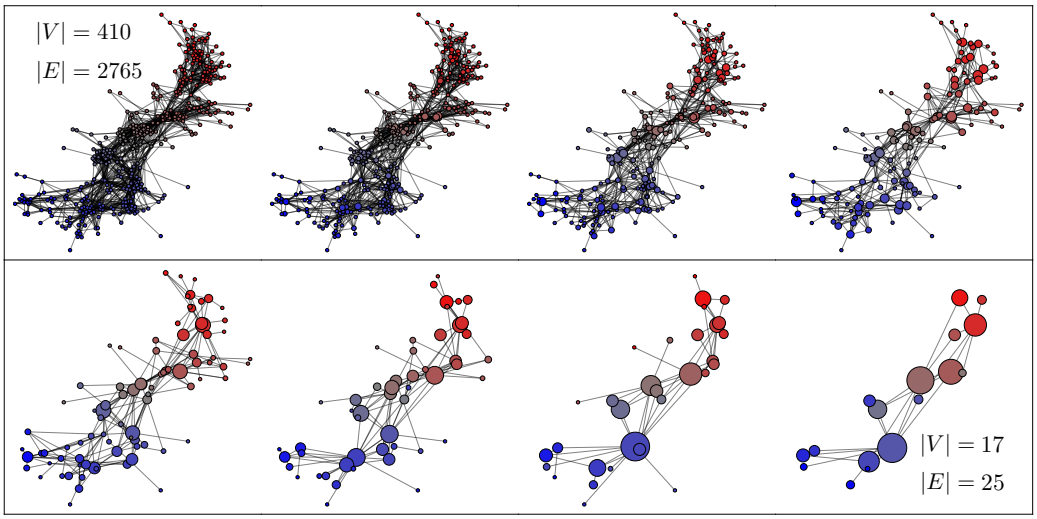


Figure B.1: **Visualization of our graph reduction algorithm preserving global structure**. We applied our algorithm (prioritizing edge reduction, and allowing for deletion, contraction, and reweighting) to a weighted social network of face-to-face interactions during an exhibition on infectious diseases, with initial edge weights proportional to the number of interactions between pairs of people (410 nodes and 2765 edges) from [116]. Node color indicates the lowest nontrivial eigenvector of the reduced Laplacian, which in this case is aligned with the temporal direction. This graph displays a notable amount of hierarchical clustering (owing to its social nature), which is reflected in the reduced graphs. Eg, our algorithm begins by collapsing small, tightly-knit clusters of several people into one "supernode", corresponding to groups of people who visited the exhibition together. A video of this reduction can be found here.

Bibliography

[1] *Counting Subgraphs via Homomorphisms* (2009), Springer Berlin Heidelberg.

[2] ABBE, E. Community detection and stochastic block models: Recent developments. *Journal of Machine Learning Research 18*, 177 (2018), 1–86.

[3] ADOLPHS, R. Social cognition and the human brain. *Trends in cognitive sciences 3*, 12 (1999), 469–479.

[4] AHN, K. J., GUHA, S., AND MCGREGOR, A. Graph sketches: sparsification, spanners, and subgraphs. In *Proceedings of the 31st ACM SIGMOD-SIGACT-SIGAI symposium on Principles of Database Systems* (2012), ACM, pp. 5–14.

[5] AIGNER, M., ZIEGLER, G. M., HOFMANN, K. H., AND ERDOS, P. *Proofs from the Book*, vol. 274. Springer, 2010.

[6] AL HASAN, M., AND ZAKI, M. J. A survey of link prediction in social networks. In *Social network data analytics*. Springer, 2011, pp. 243–275.

[7] AMARI, S.-I. *Information geometry and its applications*, vol. 194. Springer, 2016.

[8] ANTONIOU, I. E., AND TSOMPA, E. T. Statistical analysis of weighted networks. *Discrete Dynamics in Nature and Society 2008* (2008), 1–16.

[9] ATHANASIOS, A., CHARALAMPOS, V., VASILEIOS, T., ET AL. Protein-protein interaction (ppi) network: Recent advances in drug discovery. *Current drug metabolism 18*, 1 (2017), 5–10.

[10] AUSIELLO, G., CRESCENZI, P., GAMBOSI, G., KANN, V., MARCHETTI-SPACCAMELA, A., AND PROTASI, M. *Complexity and Approximation: Combinatorial Optimization Problems and their Approximability Properties*. Springer Science & Business Media, 2012.

[11] BADDELEY, A. The magical number seven: Still magic after all these years?

[12] BAEZ, J. C., AND FONG, B. A compositional framework for passive linear networks. *arXiv preprint arXiv:1504.05625* (2015).

[13] BARABÁSI, A.-L., AND ALBERT, R. Emergence of scaling in random networks. *science 286*, 5439 (1999), 509–512.

[14] BARABÁSI, A.-L., ET AL. *Network science*. Cambridge university press, 2016.

[15] BARLOW, H. B., ET AL. Possible principles underlying the transformation of sensory messages. *Sensory communication 1* (1961), 217–234.

[16] BARRAT, A., BARTHÉLEMY, M., PASTOR-SATORRAS, R., AND VESPIGNANI, A. The architecture of complex weighted networks. *Proc. Natl Acad. Sci. USA 101*, 11 (2004), 3747–3752.

[17] BARTLETT, F. C., BARTLETT, F. C., AND KINTSCH, W. *Remembering: A study in experimental and social psychology*, vol. 14. Cambridge University Press, 1932.

[18] BATSON, J., SPIELMAN, D. A., SRIVASTAVA, N., AND TENG, S.-H. Spectral sparsification of graphs: Theory and algorithms. *Communications of the ACM 56*, 8 (2013), 87–94.

[19] BATTAGLIA, P. W., HAMRICK, J. B., BAPST, V., SANCHEZ-GONZALEZ, A., ZAMBALDI, V., MALINOWSKI, M., TACCHETTI, A., RAPOSO, D., SANTORO, A., FAULKNER, R., ET AL. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261* (2018).

[20] BATTISTON, F., MUSCIOTTO, F., WANG, D., BARABÁSI, A.-L., SZELL, M., AND SINATRA, R. Taking census of physics. *Nature Reviews Physics 1*, 1 (2019), 89.

[21] BATTISTON, S., GLATTFELDER, J. B., GARLASCHELLI, D., LILLO, F., AND CALDARELLI, G. The structure of financial networks. In *Network Science*. Springer, 2010, pp. 131–163.

[22] BAYES, T. An essay towards solving a problem in the doctrine of chances. *Philosophical Transactions 53* (1763), 370–418.

[23] BELL, M. G., AND IIDA, Y. *Transportation network analysis*. 1997.

[24] BELL, W., OLSON, L., AND SCHRODER, J. Pyamg: Algebraic multigrid solvers in python v3. 0, 2015. *URL http://www. pyamg. org. Release 3* (2015).

[25] BELLMAN, R. The theory of dynamic programming. *Bull. Amer. Math. Soc. 60*, 6 (11 1954), 503–515.

[26] BENSON, A. R., GLEICH, D. F., AND LESKOVEC, J. Higher-order organization of complex networks. *Science 353*, 6295 (2016), 163–166.

[27] BHARDWAJ, N., YAN, K.-K., AND GERSTEIN, M. B. Analysis of diverse regulatory networks in a hierarchical context shows consistent tendencies for collaboration in the middle levels. *Proc. Natl Acad. Sci. USA 107*, 15 (2010), 6841–6846.

[28] BLANCA, M. J., ARNAU, J., LÓPEZ-MONTIEL, D., BONO, R., AND BENDAYAN, R. Skewness and kurtosis in real data samples. *Meth. Eur. J. Res. Meth. Behav. Soc. Sci. 9*, 2 (2013), 78–84.

[29] BLUM, K. I., AND ABBOTT, L. A model of spatial map formation in the hippocampus of the rat. *Neural computation 8*, 1 (1996), 85–93.

[30] BOGUNA, M., KRIOUKOV, D., ALMAGRO, P., AND SERRANO, M. Small worlds and clustering in spatial networks. *arXiv preprint arXiv:1909.00226* (2019).

[31] BORGATTI, S. P., AND HALGIN, D. S. Analyzing affiliation networks. *The Sage handbook of social network analysis 1* (2011), 417–433.

[32] BORGS, C., CHAYES, J., LOVÁSZ, L., SÓS, V. T., SZEGEDY, B., AND VESZTERGOMBI, K. Graph limits and parameter testing. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing* (2006), ACM, pp. 261–270.

[33] BOTVINICK, M., AND TOUSSAINT, M. Planning as inference. *Trends in cognitive sciences 16*, 10 (2012), 485–488.

[34] BOTVINICK, M., WEINSTEIN, A., SOLWAY, A., AND BARTO, A. Reinforcement learning, efficient coding, and the statistics of natural tasks. *Current Opinion in Behavioral Sciences 5* (Oct 2015), 71–77.

[35] BRADY, T. F., KONKLE, T., AND ALVAREZ, G. A. Compression in visual working memory: using statistical regularities to form more efficient memory representations. *Journal of experimental psychology 138 4* (2009), 487–502.

[36] BRAUN, D. A., MEHRING, C., AND WOLPERT, D. M. Structure learning in action. *Behavioural brain research 206*, 2 (2010), 157–165.

[37] BRAVO-HERMSDORFF, G., FELSO, V., RAY, E., GUNDERSON, L. M., HELANDAR, M. E., MARIA, J., AND NIV, Y. Gender and collaboration patterns in a temporal scientific authorship network. *To appear in Applied Network Science* (2019).

[38] BRAVO-HERMSDORFF*, G., AND GUNDERSON*, L. M. A unifying framework for spectrum-preserving graph sparsification and coarsening. In *Neural Information Processing Systems 33 (NeurIPS)* (2019).

[39] BRILLINGER, D. R. *Time series: data analysis and theory*, vol. 36. Siam, 1981.

[40] BRONSTEIN, M. M., BRUNA, J., LECUN, Y., SZLAM, A., AND VANDERGHEYNST, P. Geometric deep learning: Going beyond Euclidean data. *IEEE Signal Processing Magazine 34*, 4 (2017), 18–42.

[41] BRUNA, J., ZAREMBA, W., SZLAM, A., AND LECUN, Y. Spectral networks and locally connected networks on graphs. *International Conference on Learning Representations* (2014).

[42] BRUNSON, J. C. Triadic analysis of affiliation networks. *Network Science 3*, 4 (02 2015), 480–508.

[43] BRUNTON, B. W., BOTVINICK, M. M., AND BRODY, C. D. Rats and humans can optimally accumulate evidence for decision-making. *Science 340*, 6128 (2013), 95–98.

[44] BUTTS, C. T. A perfect sampling method for exponential family random graph models. *J. Math. Sociol. 42*, 1 (2018), 17–36.

[45] CAI, D., CAMPBELL, T., AND BRODERICK, T. Edge-exchangeable graphs and sparsity. In *Advances in Neural Information Processing Systems* (2016), pp. 4249–4257.

[46] CALLAWAY, F., LIEDER, F., DAS, P., GUL, S., KRUEGER, P. M., AND GRIFFITHS, T. A resource-rational analysis of human planning. In *CogSci* (2018).

[47] CAMPBELL, C., YANG, S., ALBERT, R., AND SHEA, K. A network model for plant–pollinator community assembly. *Proceedings of the National Academy of Sciences 108*, 1 (2011), 197–202.

[48] CANINI, K. R., GRIFFITHS, T. L., VANPAEMEL, W., AND KALISH, M. L. Revealing human inductive biases for category learning by simulating cultural transmission. *Psychonomic Bulletin & Review 21*, 3 (Jan 2014), 785–793.

[49] CARON, F., AND FOX, E. B. Sparse graphs using exchangeable random measures. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 79*, 5 (2017), 1295–1366.

[50] CHAKRABORTY, A., KRICHENE, H., INOUE, H., AND FUJIWARA, Y. Exponential random graph models for the japanese bipartite network of banks and firms. *Journal of Computational Social Science* (2019), 1–11.

[51] CHANDRA, A. K., RAGHAVAN, P., RUZZO, W. L., SMOLENSKY, R., AND TIWARI, P. The electrical resistance of a graph captures its commute and cover times. *Computational Complexity 6*, 4 (Dec 1996), 312–340.

[52] CHATTERJEE, S., AND DIACONIS, P. Estimating and understanding exponential random graph models. *The Annals of Statistics 41*, 5 (Oct 2013), 2428–2461.

[53] CHAZELLE, B. *The Discrepancy Method: Randomness and Complexity*. Cambridge University Press, 2000.

[54] CHAZELLE, B., AND WANG, C. Self-sustaining iterated learning. *arXiv preprint arXiv:1609.03960* (2016).

[55] CHAZELLE, B., AND WANG, C. Iterated learning in dynamic social networks. *The Journal of Machine Learning Research 20*, 1 (2019), 979–1006.

[56] CHEN, H., PEROZZI, B., HU, Y., AND SKIENA, S. HARP: Hierarchical representation learning for networks. *32nd AAAI Conference on Artificial Intelligence* (2018).

[57] CHIBA, N., AND NISHIZEKI, T. Arboricity and subgraph listing algorithms. *SIAM Journal on computing 14*, 1 (1985), 210–223.

[58] CHRISTIANO, P., KELNER, J. A., MADRY, A., SPIELMAN, D. A., AND TENG, S.-H. Electrical flows, Laplacian systems, and faster approximation of maximum flow in undirected graphs. *Proceedings of the 43rd Annual ACM Symposium on Theory of Computing* (2011), 273–282.

[59] CIMINI, G., SQUARTINI, T., SARACCO, F., GARLASCHELLI, D., GABRIELLI, A., AND CALDARELLI, G. The statistical physics of real-world networks. *Nature Reviews Physics 1*, 1 (2019), 58.

[60] COHEN, J. D., DUNBAR, K., AND MCCLELLAND, J. L. On the control of automatic processes: a parallel distributed processing account of the stroop effect. *Psychological review 97*, 3 (1990), 332.

[61] COHEN, M. B., KELNER, J., PEEBLES, J., PENG, R., RAO, A., SIDFORD, A., AND VLADU, A. Almost-linear-time algorithms for Markov chains and new spectral primitives for directed graphs. *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing* (2017), 410–419.

[62] COHEN, M. B., KYNG, R., MILLER, G. L., PACHOCKI, J. W., PENG, R., RAO, A. B., AND XU, S. C. Solving SDD linear systems in nearly $m \log^{1/2} n$ time. *Proceedings of the 46th Annual ACM Symposium on Theory of Computing* (2014).

[63] COLMAN, E. R., AND RODGERS, G. J. Complex scale-free networks with tunable power-law exponent and clustering. *Physica A: Statistical Mechanics and its Applications 392*, 21 (2013), 5501–5510.

[64] COUTINHO, B., HONG, S., ALBRECHT, K., DEY, A., BARABÁSI, A.-L., TORREY, P., VOGELSBERGER, M., AND HERNQUIST, L. The network behind the cosmic web. *arXiv preprint arXiv:1604.03236* (2016).

[65] COWAN, N. The magical mystery four: How is working memory capacity limited, and why? *Current directions in psychological science 19*, 1 (2010), 51–57.

[66] CRAIG, C. C. On a property of the semi-invariants of thiele. *The Annals of Mathematical Statistics 2*, 2 (1931), 154–164.

[67] CSARDI, G., NEPUSZ, T., ET AL. The igraph software package for complex network research. *InterJournal, Complex Systems 1695*, 5 (2006), 1–9.

[68] CURTICAPEAN, R., DELL, H., AND MARX, D. Homomorphisms are a good basis for counting small subgraphs. In *STOC* (2017), ACM, pp. 210–223.

[69] CZÉGEL, D., ZACHAR, I., AND SZATHMÁRY, E. Major evolutionary transitions as bayesian structure learning. *bioRxiv* (2018), 359596.

[70] DANISCH, M., BALALAU, O., AND SOZIO, M. Listing k-cliques in sparse real-world graphs. In *Proceedings of the 2018 World Wide Web Conference* (2018), International World Wide Web Conferences Steering Committee, pp. 589–598.

[71] DEL GENIO, C. I., GROSS, T., AND BASSLER, K. E. All scale-free networks are sparse. *Physical review letters 107*, 17 (2011), 178701.

[72] DEMAINE, E. D., REIDL, F., ROSSMANITH, P., VILLAAMIL, F. S., SIKDAR, S., AND SULLIVAN, B. D. Structural sparsity of complex networks: Bounded expansion in random models and real-world graphs. *Journal of Computer and System Sciences* (2019).

[73] DEVAINE, M., HOLLARD, G., AND DAUNIZEAU, J. The social bayesian brain: does mentalizing make a difference when we learn? *PLoS computational biology 10*, 12 (2014), e1003992.

[74] DI NARDO, E., AND SENATO, D. An umbral setting for cumulants and factorial moments. *European Journal of Combinatorics 27*, 3 (2006), 394–413.

[75] DOYA, K., ISHII, S., POUGET, A., AND RAO, R. P. *Bayesian brain: Probabilistic approaches to neural coding*. MIT press, 2007.

[76] EICHENBAUM, H. The role of the hippocampus in navigation is memory. *Journal of neurophysiology 117*, 4 (2017), 1785–1796.

[77] EMMONS, S., KOBOUROV, S., GALLANT, M., AND BÖRNER, K. Analysis of network clustering algorithms and cluster quality metrics at scale. *PloS one 11*, 7 (2016), e0159161.

[78] EULER, L. Solutio problematis ad geometriam situs pertinentis. *Commentarii Academiae Scientiarum Imperialis Petropolitanae 8* (1736), 128–140.

[79] FEYNMAN, R. P. Space-time approach to quantum electrodynamics. *Physical Review 76*, 6 (1949), 769.

[80] FIEDLER, M. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal 23* (1973), 298–305.

[81] FIELD, D. J. What the statistics of natural images tell us about visual coding. *Human Vision, Visual Processing, and Digital Display* (Aug 1989).

[82] FIENBERG, S. E. Introduction to papers on the modeling and analysis of network data. *Ann. Appl. Stat. 4* (2010), 1–4.

[83] FISHER, N. Unbiased estimation for some non-parametric families of distributions. *The Annals of Statistics* (1982), 603–615.

[84] FISHER, R. A. Moments and Product Moments of Sampling Distributions. *Proceedings of the London Mathematical Society 2*, 1 (1930), 199–238.

[85] FOMIN, F. V., LOKSHTANOV, D., RAMAN, V., SAURABH, S., AND RAO, B. R. Faster algorithms for finding and counting subgraphs. *Journal of Computer and System Sciences 78*, 3 (2012), 698–706.

[86] FUNG, W.-S., HARIHARAN, R., HARVEY, N. J., AND PANIGRAHI, D. A general framework for graph sparsification. *SIAM Journal on Computing 48*, 4 (2019), 1196–1223.

[87] GAO, X., XIAO, B., TAO, D., AND LI, X. A survey of graph edit distance. *Pattern Analysis and applications 13*, 1 (2010), 113–129.

[88] GENTILE, C., HERBSTER, M., AND PASTERIS, S. Online similarity prediction of networked data from known and unknown graphs. *Conference on Learning Theory* (2013), 662–695.

[89] GERSHMAN, S. J., HORVITZ, E. J., AND TENENBAUM, J. B. Computational rationality: A converging paradigm for intelligence in brains, minds, and machines. *Science 349*, 6245 (2015), 273–278.

[90] GLEISER, P. M., AND DANON, L. Community structure in jazz. *Advances in Complex Systems 6*, 04 (2003), 565–573.

[91] GNEDENKO, B., AND KOLMOGOROV, A. Limit distributions for sums of independent random variables. In *Am. Math. Soc* (1949), vol. 62, pp. 50–52.

[92] GOSWAMI, S., MURTHY, C., AND DAS, A. K. Sparsity measure of a network graph: Gini index. *Information Sciences 462* (2018), 16–39.

[93] GRIFFITHS, T. L., AND KALISH, M. L. Language evolution by iterated learning with bayesian agents. *Cognitive Science 31*, 3 (2007), 441–480.

[94] GRIFFITHS, T. L., AND REALI, F. Modelling minds as well as populations. *Proceedings of the Royal Society B: Biological Sciences 278*, 1713 (2011), 1773–1776.

[95] GUIDOLIN, M., AND TIMMERMANN, A. International asset allocation under regime switching, skew, and kurtosis preferences. *Rev. Financ. Stud. 21*, 2 (2008), 889–935.

[96] GUNDERSON, B., AND ALIAGA, M. Interactive statistics, 2005.

[97] HALD, A. The early history of the cumulants and the gram-charlier series. *International Statistical Review 68*, 2 (2000), 137–153.

[98] HALL, W., AND TIROPANIS, T. Web evolution and web science. *Computer Networks 56*, 18 (2012), 3859–3865.

[99] HAMILTON, W., YING, Z., AND LESKOVEC, J. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems* (2017), pp. 1024–1034.

[100] HAMILTON, W. L., YING, R., AND LESKOVEC, J. Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584* (2017).

[101] HAREL, D., AND KOREN, Y. A fast multi-scale method for drawing large graphs. *Graph Drawing* (2001), 183–196.

[102] HENAFF, M., BRUNA, J., AND LECUN, Y. Deep convolutional networks on graph-structured data. *arXiv:1506.05163* (2015).

[103] HENDRICKSON, B., AND LELAND, R. W. A multilevel algorithm for partitioning graphs. *Proceedings of the 1995 ACM/IEEE Conference on Supercomputing 95*, 28 (1995), 1–14.

[104] HERBSTER, M., PONTIL, M., AND WAINER, L. Online learning over graphs. *Proceedings of the 22nd International Conference on Machine Learning* (2005), 305–312.

[105] HIRANI, A., KALYANARAMAN, K., AND WATTS, S. Graph Laplacians and least squares on graphs. *IEEE International Parallel and Distributed Processing Symposium Workshop* (2015), 812–821.

[106] HOLLAND, P. W., AND LEINHARDT, S. An exponential family of probability distributions for directed graphs. *Journal of the American Statistical Association 76*, 373 (Mar 1981), 33–50.

[107] HOLME, P., AND KIM, B. J. Growing scale-free networks with tunable clustering. *Physical review E 65*, 2 (2002), 026107.

[108] HORNER, V., WHITEN, A., FLYNN, E., AND DE WAAL, F. B. Faithful replication of foraging techniques along cultural transmission chains by chimpanzees and children. *Proceedings of the National Academy of Sciences 103*, 37 (2006), 13878–13883.

[109] HORVÁT, S., CZABARKA, É., AND TOROCZKAI, Z. Reducing degeneracy in maximum entropy models of networks. *Phys. Rev. Lett. 114* (2015), 158701.

[110] HOWE, C., AND PURVES, D. The muller-lyer illusion explained by the statistics of image–source relationships. *Proceedings of the National Academy of Sciences 102*, 4 (2005), 1234–1239.

[111] HOWE, C., YANG, Z., AND PURVES, D. The poggendorff illusion explained by natural scene geometry. *Proceedings of the National Academy of Sciences 102*, 21 (2005), 7707–7712.

[112] HTTPS://EN.WIKIPEDIA.ORG/WIKI/CHINESE_WHISPERS. *Wikipedia page for the telephone game*.

[113] HTTPS://KINGJAMESPROGRAMMING.TUMBLR.COM/. *King James Programming website*, created on Dec 2013 by https://github.com/Barrucadu/markov, last checked on Sep 2019.

[114] HTTPS://OEIS.ORG/A000088. *The on-line encyclopedia of integer sequences (OEIS), entry A000088*, founded by Neil Sloane in 1964.

[115] HSU, D. J., KONTOROVICH, A., AND SZEPESVÁRI, C. Mixing time estimation in reversible markov chains from a single sample path. In *Advances in neural information processing systems* (2015), pp. 1459–1467.

[116] ISELLA, L., STEHLÉ, J., BARRAT, A., CATTUTO, C., PINTON, J.-F., AND DEN BROECK, W. V. What's in a crowd? Analysis of face-to-face behavioral networks. *Journal of Theoretical Biology 271*, 1 (2011), 166–180.

[117] JAMES, G. On moments and cumulants of systems of statistics. *Sankhyā: The Indian Journal of Statistics* (1958), 1–30.

[118] JARRELL, T. A., WANG, Y., BLONIARZ, A. E., BRITTIN, C. A., XU, M., THOMSON, J. N., ALBERTSON, D. G., HALL, D. H., AND EMMONS, S. W. The connectome of a decision-making neural network. *Science 337*, 6093 (2012), 437–444.

[119] JAYNES, E. T. Information theory and statistical mechanics. *Phy. Rev. 106*, 4 (1957), 620–630.

[120] JAYNES, E. T. Information theory and statistical mechanics. ii. *Phys. Rev. 108*, 2 (1957), 171–190.

[121] JIN, Y., AND JAJA, J. F. Network summarization with preserved spectral properties. *arXiv:1802.04447* (2018).

[122] JOYCE, T., AND HERRMANN, J. M. A review of no free lunch theorems, and their implications for metaheuristic optimisation. In *Nature-Inspired Algorithms and Applied Optimization*. Springer, 2018, pp. 27–51.

[123] JUYONG PARK, M. E. J. N. Solution of the two-star model of a network. *Phy. Rev. E 70*, 6 (2004), 066146.

[124] JUYONG PARK, M. E. J. N. Statistical mechanics of networks. *Phy. Rev. E 70*, 6 (2004), 066117.

[125] JUYONG PARK, M. E. J. N. Solution for the properties of a clustered network. *Phy. Rev. E 72*, 2 (2005), 026136.

[126] KACZMARZYK, M., FRANCIKOWSKI, J., ŁOZOWSKI, B., ROZPĘDEK, M., SAWCZYN, T., AND SUŁOWICZ, S. The bit value of working memory. *Psychology & Neuroscience 6*, 3 (2013), 345–349.

[127] KAPRALOV, M., LEE, Y. T., MUSCO, C., MUSCO, C. P., AND SIDFORD, A. Single pass spectral sparsification in dynamic streams. *SIAM Journal on Computing 46*, 1 (2017), 456–477.

[128] KARDAR, M. *Statistical Physics of Particles*. Cambridge Univ. Press, Cambridge, 2007.

[129] KARTUN-GILES, A., KRIOUKOV, D., GLEESON, J., MORENO, Y., AND BIANCONI, G. Sparse power-law network model for reliable statistical predictions based on sampled data. *Entropy 20*, 4 (2018), 257.

[130] KARYPIS, G., AND KUMAR, V. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing 20*, 1 (1998), 359–392.

[131] KENDALL, M. The derivation of multivariate sampling formulae from univariate formulae by symbolic operation. *Annals of Eugenics 10*, 1 (1940), 392–402.

[132] KENDALL, M. Proof of fisher's rules for ascertaining the sampling semi-invariants of k-statistics. *Annals of Eugenics 10*, 1 (1940), 215–222.

[133] KENDALL, M. Some properties of k-statistics. *Annals of Eugenics 10*, 1 (1940), 106–111.

[134] KIPF, T. N., AND WELLING, M. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[135] KIRBY, S., CORNISH, H., AND SMITH, K. Cumulative cultural evolution in the laboratory: An experimental approach to the origins of structure in human language. *Proceedings of the National Academy of Sciences 105*, 31 (2008), 10681–10686.

[136] KIRBY, S., GRIFFITHS, T., AND SMITH, K. Iterated learning and the evolution of language. *Current opinion in neurobiology 28* (2014), 108–114.

[137] KLEMM, K., AND EGUILUZ, V. M. Highly clustered scale-free networks. *Physical Review E 65*, 3 (2002), 036123.

[138] KNILL, D. C., AND RICHARDS, W. *Perception as Bayesian inference*. Cambridge University Press, 1996.

[139] KONDOR, R., SON, H. T., PAN, H., ANDERSON, B., AND TRIVEDI, S. Covariant compositional networks for learning graphs. *arXiv preprint arXiv:1801.02144* (2018).

[140] KOREN, Y., CARMEL, L., AND HAREL, D. ACE: A fast multiscale eigenvectors computation for drawing huge graphs. *IEEE Symposium on Information Visualization* (2002), 137–144.

[141] KOSKINEN, J. Exponential random graph models. *Wiley StatsRef: Statistics Reference Online* (2011), 1–12.

[142] KOSKINEN, J., WANG, P., ROBINS, G., AND PATTISON, P. Outliers and influential observations in exponential random graph models. *Psychometrika 83*, 4 (2018), 809–830.

[143] KOUTIS, I., AND XU, S. C. Simple parallel and distributed algorithms for spectral graph sparsification. *ACM Transactions on Parallel Computing (TOPC) 3*, 2 (2016), 14.

[144] KUBO, R. Generalized cumulant expansion method. *Journal of the Physical Society of Japan 17*, 7 (1962), 1100–1120.

[145] KYNG, R., PACHOCKI, J., PENG, R., AND SACHDEVA, S. A framework for analyzing resparsification algorithms. *Proceedings of the 38th Annual ACM-SIAM Symposium on Discrete Algorithms* (2017), 2032–2043.

[146] KYNG, R., RAO, A., SACHDEVA, S., AND SPIELMAN, D. A. Algorithms for Lipschitz learning on graphs. *Conference on Learning Theory* (2015).

[147] LAFON, S., AND LEE, A. Diffusion maps and coarse-graining: A unified framework for dimensionality reduction, graph partitioning, and data set parameterization. *IEEE Transactions on Pattern Analysis and Machine Intelligence 28*, 9 (Sep 2006), 1393–1403.

[148] LANDI, P., MINOARIVELO, H. O., BRÄNNSTRÖM, Å., HUI, C., AND DIECKMANN, U. Complexity and stability of ecological networks: a review of the theory. *Population ecology 60*, 4 (2018), 319–345.

[149] LANGLOIS, T., JACOBY, N., SUCHOW, J. W., AND GRIFFITHS, T. L. Uncovering visual priors in spatial memory using serial reproduction. In *CogSci* (2017).

[150] LE GALL, F. Powers of tensors and fast matrix multiplication. *Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation* (2014).

[151] LEE, J., KIM, H., LEE, J., AND YOON, S. Transfer learning for deep learning on graph-structured data. In *Thirty-First AAAI Conference on Artificial Intelligence* (2017).

[152] LEE, Y. T., AND SUN, H. An SDP-based algorithm for linear-sized spectral sparsification. *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing* (2017), 678–687.

[153] LEWANDOWSKY, S., GRIFFITHS, T. L., AND KALISH, M. L. The wisdom of individuals: Exploring people's knowledge about everyday events using iterated learning. *Cognitive science 33*, 6 (2009), 969–998.

[154] LEWICKI, M. S. Efficient coding of natural sounds. *Nature neuroscience 5*, 4 (2002), 356.

[155] LIBEN-NOWELL, D., AND KLEINBERG, J. The link-prediction problem for social networks. *Journal of the American society for information science and technology 58*, 7 (2007), 1019–1031.

[156] LINDLEY, D. V. *Bayesian statistics, a review*, vol. 2. SIAM, 1972.

[157] LONG, F., YANG, Z., AND PURVES, D. Spectral statistics in natural scenes predict hue, saturation, and brightness. *Proceedings of the National Academy of Sciences 103*, 15 (2006), 6013–6018.

[158] LOUKAS, A. Graph reduction with spectral and cut guarantees. *arXiv:1808.10650v2* (2018).

[159] LOUKAS, A., AND VANDERGHEYNST, P. Spectrally approximating large graphs with smaller graphs. *International Conference on Machine Learning 80* (2018), 3237–3246.

[160] LOVÁSZ, L. *Large networks and graph limits*, vol. 60. American Mathematical Soc., 2012.

[161] LOVÁSZ, L., AND SZEGEDY, B. Limits of dense graph sequences. *Journal of Combinatorial Theory, Series B 96*, 6 (2006), 933–957.

[162] LYNN, C. W., AND BASSETT, D. S. The physics of brain network structure, function and control. *Nature Reviews Physics* (2019), 1.

[163] MAGUIRE, E. A., SPIERS, H. J., GOOD, C. D., HARTLEY, T., FRACKOWIAK, R. S., AND BURGESS, N. Navigation expertise and the human hippocampus: a structural brain imaging analysis. *Hippocampus 13*, 2 (2003), 250–259.

[164] MATHY, F., AND FELDMAN, J. What's magic about magic numbers? chunking and data compression in short-term memory. *Cognition 122*, 3 (2012), 346–362.

[165] MCCULLAGH, P. *Tensor Methods in Statistics: Monographs on Statistics and Applied Probability*. Chapman and Hall/CRC, 2018.

[166] MCDERMOTT, J. H., SCHEMITSCH, M., AND SIMONCELLI, E. P. Summary statistics in auditory perception. *Nature neuroscience 16*, 4 (2013), 493.

[167] MESOUDI, A. Cultural evolution: integrating psychology, evolution and culture. *Current Opinion in Psychology 7* (2016), 17 – 22. Evolutionary psychology.

[168] MEYER, JR, C. D. Generalized inversion of modified matrices. *SIAM Journal on Applied Mathematics 24*, 3 (1973), 315–323.

[169] MONNIG, N. D., AND MEYER, F. G. The resistance perturbation distance: A metric for the analysis of dynamic networks. *Discrete Applied Mathematics 236* (2018), 347–386.

[170] NACHER, J. C., AND AKUTSU, T. Structural controllability of unidirectional bipartite networks. *Scientific reports 3* (2013), 1647.

[171] NEGAHBAN, S., OH, S., AND SHAH, D. Iterative ranking from pairwise comparisons. *Advances in Neural Information Processing Systems* (2012), 2474–2482.

[172] NEWMAN, M. *Networks*. Oxford university press, 2018.

[173] NEWMAN, M. E. Modularity and community structure in networks. *Proceedings of the national academy of sciences 103*, 23 (2006), 8577–8582.

[174] NEWMAN, M. E., BARABÁSI, A.-L. E., AND WATTS, D. J. *The structure and dynamics of networks*. Princeton university press, 2006.

[175] NOURI, M., TALATAHARI, S., AND SHAMLOO, A. S. Graph products and its applications in mathematical formulation of structures. *Journal of Applied Mathematics 2012* (2012).

[176] O'KEEFE, J., AND BURGESS, N. Theta activity, virtual navigation and the human hippocampus. *Trends in cognitive sciences 3*, 11 (1999), 403–406.

[177] OPSAHL, T., AND PANZARASA, P. Clustering in weighted networks. *Social Networks 31*, 2 (May 2009), 155–163.

[178] ORBÁN, G., FISER, J., ASLIN, R. N., AND LENGYEL, M. Bayesian learning of visual chunks by human observers. *Proceedings of the National Academy of Sciences 105*, 7 (2008), 2745–2750.

[179] ORSINI, C., DANKULOV, M. M., COLOMER-DE SIMÓN, P., JAMAKOVIC, A., MAHADEVAN, P., VAHDAT, A., BASSLER, K. E., TOROCZKAI, Z., BOGUÑÁ, M., CALDARELLI, G., AND ET AL. Quantifying randomness in real networks. *Nature Communications 6*, 1 (Oct 2015).

[180] OXLEY, J. G. *Matroid Theory*, vol. 3. Oxford University Press, USA, 2006.

[181] PEDRO G. LIND, M. C. G., AND HERRMANN, H. J. Cycles and clustering in bipartite networks. *Phys. Rev. E 72*, 5 (2005), 056127.

[182] PILCO, D. S., AND RIVERA, A. R. Graph learning network: A structure learning algorithm. *arXiv preprint arXiv:1905.12665* (2019).

[183] PINAR, A., SESHADHRI, C., AND VISHAL, V. Escape: Efficiently counting all 5-vertex subgraphs. In *Proceedings of the 26th International Conference on World Wide Web* (2017), pp. 1431–1440.

[184] PIROTTE, A., RENDERS, J.-M., SAERENS, M., AND FOUSS, F. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *IEEE Transactions on Knowledge & Data Engineering*, 3 (2007), 355–369.

[185] PUROHIT, M., PRAKASH, B. A., KANG, C., ZHANG, Y., AND SUBRAHMANIAN, V. Fast influence-based coarsening for large networks. *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2014), 1296–1305.

[186] RAFTERY, A. E., AND LEWIS, S. M. Implementing mcmc. *Markov chain Monte Carlo in practice* (1996), 115–130.

[187] RANJAN, G., AND ZHANG, Z.-L. Geometry of complex networks and topological centrality. *Physica A: Statistical Mechanics and its Applications 392*, 17 (2013), 3833–3845.

[188] RANJAN, G., ZHANG, Z.-L., AND BOLEY, D. Incremental computation of pseudoinverse of Laplacian. *Lecture Notes in Computer Science* (2014), 729–749.

[189] RICHARDSON, H. Development of brain networks for social functions: Confirmatory analyses in a large open source dataset. *Developmental cognitive neuroscience 37* (2019), 100598.

[190] RICHARDSON, H., AND SAXE, R. Development of predictive responses in theory of mind brain regions. *Developmental science* (2018), e12863.

[191] RIEDEL, K. S. A Sherman–Morrison–Woodbury identity for rank augmenting matrices with application to centering. *SIAM Journal on Matrix Analysis and Applications 13*, 2 (1992), 659–662.

[192] RIEKE, F., WARLAND, D., VAN STEVENINCK, R. D. R., AND BIALEK, W. *Spikes: Exploring the Neural Code*. MIT press, 1999.

[193] ROBINS, G., AND ALEXANDER, M. Small worlds among interlocking directors: Network structure and distance in bipartite graphs. *Computational & Mathematical Organization Theory 10*, 1 (2004), 69–94.

[194] ROBINS, G., PATTISON, P., KALISH, Y., AND LUSHER, D. An introduction to exponential random graph (p*) models for social networks. *Social Networks 29*, 2 (2007), 173–191.

[195] RON, D., SAFRO, I., AND BRANDT, A. Relaxation-based coarsening and multiscale graph organization. *SIAM Journal on Multiscale Modeling & Simulation 9*, 1 (2011), 407–423.

[196] ROTA, G.-C., AND SHEN, J. On the combinatorics of cumulants.

[197] SAERENS, M., FOUSS, F., YEN, L., AND DUPONT, P. The principal components analysis of a graph, and its relationships to spectral clustering. *European Conference on Machine Learning* (2004), 371–383.

[198] SAFRO, I., SANDERS, P., AND SCHULZ, C. Advanced coarsening schemes for graph partitioning. *Journal of Experimental Algorithmics 19* (Jan 2015), 1.1–1.24.

[199] SAMPAIO FILHO, C. I., MOREIRA, A. A., ANDRADE, R. F., HERRMANN, H. J., AND ANDRADE JR, J. S. Mandala networks: ultra-small-world and highly sparse graphs. *Scientific reports 5* (2015), 9082.

[200] SANBORN, S., BOURGIN, D., CHANG, M., AND GRIFFITHS, T. L. Representational efficiency outweighs action efficiency in human program induction. *CoRR abs/1807.07134* (2018).

[201] SARAMAKI, J., KIVELA, M., ONNELA, J.-P., KASKI, K., AND KERTESZ, J. Generalizations of the clustering coefficient to weighted complex networks. *Phys. Rev. E 75* (2007), 027105.

[202] SATULURI, V., PARTHASARATHY, S., AND RUAN, Y. Local graph sparsification for scalable clustering. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data* (2011), ACM, pp. 721–732.

[203] SAUL, Z. M., AND FILKOV, V. Exploring biological network structure using exponential random graph models. *Bioinformatics 23*, 19 (2007), 2604–2611.

[204] SCHWEINBERGER, M. Instability, sensitivity, and degeneracy of discrete exponential families. *J. Am. Stat. Assoc. 106*, 496 (2011), 1361–1370.

[205] SIMONCELLI, E. P. Vision and the statistics of the visual environment. *Current Opinion in Neurobiology 13*, 2 (2003), 144 – 149.

[206] SIMONCELLI, E. P., AND OLSHAUSEN, B. A. Natural image statistics and neural representation. *Annual review of neuroscience 24* (2001), 1193–216.

[207] SIMONOVSKY, M., AND KOMODAKIS, N. Dynamic edge-conditioned filters in convolutional neural networks on graphs. *IEEE Conference on Computer Vision and Pattern Recognition* (2017), 3693–3702.

[208] SNIJDERS, T. A., PATTISON, P. E., ROBINS, G. L., AND HANDCOCK, M. S. New specifications for exponential random graph models. *Sociological methodology 36*, 1 (2006), 99–153.

[209] SOLIS, R., BORKAR, V. S., AND KUMAR, P. A new distributed time synchronization protocol for multihop wireless networks. *Proceedings of the 45th IEEE Conference on Decision and Control* (2006), 2734–2739.

[210] SPEED, T. Cumulants and partition lattices 1. *Australian Journal of Statistics 25*, 2 (1983), 378–388.

[211] SPEED, T. Cumulants and partition lattices ii: Generalised k-statistics. *Journal of the Australian Mathematical Society 40*, 1 (1986), 34–53.

[212] SPIELMAN, D. A., AND SRIVASTAVA, N. Graph sparsification by effective resistances. *SIAM Journal on Computing 40*, 9 (2011), 1913–1926.

[213] SPIELMAN, D. A., AND TENG, S.-H. Spectral sparsification of graphs. *SIAM Journal on Computing 40*, 4 (2011), 981–1025.

[214] STACHENFELD, K. L., BOTVINICK, M. M., AND GERSHMAN, S. J. The hippocampus as a predictive map. *Nature neuroscience 20*, 11 (2017), 1643.

[215] STEHL'E, J., VOIRIN, N., BARRAT, A., CATTUTO, C., ISELLA, L., PINTON, J.-F., QUAGGIOTTO, M., VAN DEN BROECK, W., RÉGIS, C., LINA, B., ET AL. High-resolution measurements of face-to-face contact patterns in a primary school. *PloS One 6*, 8 (2011), e23176.

[216] STRAUSS, D. On a general class of models for interaction. *SIAM Review 28*, 4 (1986), 513–527.

[217] STROOP, J. R. Studies of interference in serial verbal reactions. *Journal of experimental psychology 18*, 6 (1935), 643.

[218] STROUSE, D., KLEIMAN-WEINER, M., TENENBAUM, J., BOTVINICK, M., AND SCHWAB, D. J. Learning to share and hide intentions using information regularization. In *Advances in Neural Information Processing Systems* (2018), pp. 10249–10259.

[219] SUCHOW, J. W., BOURGIN, D. D., AND GRIFFITHS, T. L. Evolution in mind: Evolutionary dynamics, cognitive processes, and bayesian inference. *Trends in cognitive sciences 21*, 7 (2017), 522–530.

[220] TENENBAUM, J. B., AND GRIFFITHS, T. L. Generalization, similarity, and bayesian inference. *Behavioral and brain sciences 24*, 4 (2001), 629–640.

[221] TENG, S.-H. The Laplacian paradigm: Emerging algorithms for massive graphs. *Theory and Applications of Models of Computation* (2010), 2–14.

[222] THIELE, T. N. *Theory of observations*. Charles & Edwin Layton, 1903.

[223] TUKEY, J. W. Some sampling simplified. *Journal of the American Statistical Association 45*, 252 (1950), 501–519.

[224] UDDENBERG, S., AND SCHOLL, B. J. Teleface: Serial reproduction of faces reveals a whiteward bias in race memory. *Journal of Experimental Psychology: General 147*, 10 (2018), 1466.

[225] ŠUBELJ, L., AND BAJEC, M. Robust network community detection using balanced propagation. *The European Physical Journal B 81*, 3 (May 2011), 353–362.

[226] VÄHÄMAA, S. Skewness and kurtosis adjusted black-scholes model: A note on hedging performance. *Finance Letters 1*, 5 (2003), 6–12.

[227] VAN MIEGHEM, P., DEVRIENDT, K., AND CETINAY, H. Pseudoinverse of the Laplacian and best spreader node in a network. *Physical Review E 96*, 3 (2017), 032311.

[228] VELIČKOVIĆ, P., CUCURULL, G., CASANOVA, A., ROMERO, A., LIO, P., AND BENGIO, Y. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).

[229] WASSERMAN, S., AND FAUST, K. *Social Network Analysis: Methods and applications*. Cambridge Univ. Press, Cambridge, 1994.

[230] WATTS, D. J., AND STROGATZ, S. H. Collective dynamics of "small-world" networks. *Nature 393*, 6684 (Jun 1998), 440–442.

[231] WOLFRAM RESEARCH, I. *Mathematica*. Wolfram Research, Inc., Champaign, Illinois, 2019.

[232] WOODBURY, M. A. *Inverting Modified Matrices*. Memorandum Rept 42, Statistical Research Group. Princeton University, Princeton, NJ, 1950.

[233] XU, J., DOWMAN, M., AND GRIFFITHS, T. L. Cultural transmission results in convergence towards colour term universals. *Proceedings of the Royal Society B: Biological Sciences 280*, 1758 (2013), 20123073.

[234] XU, J., AND GRIFFITHS, T. L. How memory biases affect information transmission: A rational analysis of serial reproduction. In *Advances in Neural Information Processing Systems* (2009), pp. 1809–1816.

[235] YEUNG, S., AND GRIFFITHS, T. L. Identifying expectations about the strength of causal relationships. *Cognitive psychology 76* (2015), 1–29.

[236] ZACHARY, W. W. An information flow model for conflict and fission in small groups. *Journal of anthropological research 33*, 4 (1977), 452–473.

[237] ZHANG, P., WANG, J., LI, X., LI, M., DI, Z., AND FAN, Y. Clustering coefficient and community structure of bipartite networks. *Phys. A: Statistical Mechanics and its Applications 387*, 27 (2008), 6869–6875.

[238] ZHAO, Z., WANG, Y., AND FENG, Z. Nearly-linear time spectral graph reduction for scalable graph partitioning and data visualization. *arXiv:1812.08942* (2018).

[239] ZUEV, K., BOGUNÁ, M., BIANCONI, G., AND KRIOUKOV, D. Emergence of soft communities from geometric preferential attachment. *Scientific reports 5* (2015), 9421.