# Design and Analysis of Experiments on Networks

by

Trang Bui

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Statistics

Waterloo, Ontario, Canada, 2024

**Examining Committee Membership**

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

| | |
|---|---|
| External Examiner: | Steven Gilmour<br>Professor, Department of Mathematics<br>King's College London |
| Supervisor(s): | Stefan Steiner<br>Professor, Dept. of Statistics and Actuarial Science,<br>University of Waterloo |
| | Nathaniel Stevens<br>Assistant Professor, Dept. of Statistics and Actuarial Science,<br>University of Waterloo |
| Internal Member: | Pengfei Li<br>Professor, Dept. of Statistics and Actuarial Science,<br>University of Waterloo |
| | Lucy Gao<br>Adjunct Professor, Dept. of Statistics and Actuarial Science,<br>University of Waterloo<br>Assistant Professor, Department of Statistics<br>University of British Columbia |
| Internal-External Member: | Thomas Parker<br>Associate Professor, Department of Economics<br>University of Waterloo |

## Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

**Abstract**

In the design and analysis of experiments, it is often assumed that experimental units are independent, in the sense that the treatment assigned to one unit will not affect the potential outcome of another unit. However, this assumption may not hold if the experiment is conducted on a network of experimental units. The treatment assignment of one unit can spread to its neighbors via their network connections. The growing popularity of online experiments conducted on social networks calls for more research on this topic. We investigate the problem of experiments on networks and propose new approaches to both the design and analysis of such experiments.

The first chapter gives a brief introduction to the problem of network experimentation. We begin by presenting some basic concepts of network data analysis and experimental design. We then provide a review of existing literature on experiments on networks, which can be categorized into model-based and design-based approaches. The model-based approaches start by positing a model for the experimental outcomes and optimal designs are found based on design criteria formulated from the model. However, the model-based approaches can be sensitive to model misspecification. In this thesis, we seek to enhance the robustness of the model-based approaches by extending the flexibility of the models and developing design methods that take into account one's uncertainty in model parameters.

In the second chapter, we propose the General Additive Network Effect (GANE) model, which encompasses many existing models from the literature while providing greater flexibility in modeling the experimental outcomes on networks. In addition, we establish an analysis framework that can be applied not only to the proposed GANE model but also to many other experimental network outcome models. In particular, we define causal effect quantities, hypothesis tests, and design criteria that are of interest in experiments on networks. We derive the quasi-likelihood-based estimation procedure and inferential properties of a specific family of specifications under the GANE framework. The performance of certain specifications of the GANE model is studied via simulations. We find that our proposed POW-DEG specification performs well under model misspecifications.

The third chapter considers network experiments with binary outcomes. In this case, models for continuous outcomes like those in Chapter 2 are no longer appropriate. We thus extend the GANE framework to binary data using link functions in a fashion similar to generalized linear models. The model inherits the flexibility of network effect modeling and inference from the GANE framework. Estimation and inference are carried out via the maximum likelihood framework. We investigate the

performance of different specifications of the model via simulations. We find that for trustworthy estimation and inference in experiments with binary outcomes, larger sample sizes are required than for their continuous-response counterpart. Binary models are also difficult to estimate if the outcome data lacks heterogeneity (i.e., a drastic imbalance of 1s and 0s). This requires careful consideration in the experimental design. Finally, we illustrate the applicability of our method in an agricultural insurance experiment.

As illustrated in Chapters 2 and 3, to capture the interference patterns on networks, models for network experiments can be complex. As a result, design criteria defined based on these models may not have an analytical form and/or they may involve unknown parameters. This limits the use of traditional design construction methods such as integer programming or gradient-based optimization algorithms. In Chapter 4, we focus on enhancing the robustness of model-based approaches by formulating a Bayesian design framework that takes into account prior distributions for the unknown parameters. To find optimal designs, we investigate and adapt a variety of general optimization algorithms. We investigate and compare the effectiveness of these algorithms over a variety of model specifications and data sets that have been proposed and used in the literature. Based on the resulting designs obtained from various algorithms and settings, we deduce desirable design characteristics for each model and provide general design guidelines for practitioners.

Finally, in Chapter 5, we summarize the contribution of the thesis and discuss topics for future research.

**Acknowledgements**

Words cannot describe my gratitude towards my supervisors, Dr. Stefan H. Steiner and Dr. Nathaniel T. Stevens. Without their understanding, patience, trust, and unwavering support, I would not be able to persist until today. Their generosity, which provides me with unique experiences and opportunities, has been instrumental in shaping my academic path. Their work ethic, dedication, and critical thinking will continue to inspire me for years to come.

I would like to thank my committee members, Drs Pengfei Li, Lucy Gao, Thomas Parker, and Steven Gilmour for spending their precious time and energy reviewing this thesis. I am thankful for their constructive comments, questions, and suggestions.

There are many other teachers and professors who have had great influences on my personal and academic journey. First, I would like to thank Dr. Geon-Ho Choe, for advising me to continue my study abroad to sharpen my skills before going back to Vietnam. His honest advice has broadened my perspectives and led me to where I am today. Next, I would like to thank Drs. Changbao Wu, Ali Ghodsi, Pengfei Li and Alexander Schied. Their clear and enjoyable teaching ignited my interest in statistics and inspired me to delve deeper into this discipline. I deeply appreciate the help from Drs. Alex Chin, Patrick Breheny, Mauricio Sarrias, Faisal Al-Faisal, Qingyuan Zhao, and Jae-Kwang Kim. Without even knowing who I was, they have generously dedicated their time to answering my emails, explaining the technical details in their papers, and giving me so much insight into their fields. In addition, I thank Drs. Matthias Schonlau and Mehdi Molkaraie for their encouragement and friendship, Drs. Sung-Ho Kim, Gyo-Taek Jin, Kil-Hyun Kwon, Yeonseung Chung, Seung-Hun Han, and Jihee Kim for their teaching and help at KAIST, and Dr. Alexander Schied for his teaching and lessons as my Master's supervisors.

During the Ph.D., I had been fortunate to spend three years with the Statistical Consulting and Survey Research Unit. This experience tremendously propelled my growth as a consultant and statistician. Special thanks to Dr. Joslin Goh for her enthusiasm, help, advice, opportunities, and great mentorship. In addition, I want to thank Drs. Martin Lysy, Glen McGee, Meixi Chen, and Kelly Ramsay for their guidance and advice. Finally, I would like to take this opportunity to thank Dr. Jonathan Farrar and his team for their collaboration and support.

I would like to extend my thanks to all the faculty, staff, and friends at University of Waterloo for providing me with an accommodating and nourishing academic

environment. In particular, I would like to thank Drs. Changbao Wu, Tony Wirjanto, Greg Rice, Reza Ramezan, Lan Wen, Mary Thompson, Chengguo Weng, Liqun Diao, Richard Cook, Kun Liang, Shoja'eddin Chenouri, Paul Marriott, and the late Professor Ken Seng Tan, for their guidance and support whenever I tried to reach out. I would like to thank Dina Dawoud, Divya Lala, Steve Van Doormaal, Gracia Dong, Diana Skrzydlo, Anton Mosunov, and Blake Madill for their advice and help with my teaching. I thank Mary Lou Dufton and Greg Preston for their excellent and reliable assistance. My heartfelt thanks go to Wenling Zhang and Chi-Kuang Yeh for accompanying me throughout this journey, for laughing and crying with me, and for sharing with me many meaningful memories. I would also like to specially mention room 4104+: Jie Jian, Shiyu He, Wenling Zhang, and Yuying Huang. I am grateful for their company, collaboration, and sisterhood. Finally, I want to thank my other classmates: Felix Go, Monica Rudd, Mahsa Panahi, Alex, Daeyoung Gam, Zhaohan Sun, Fangya Mao, Marzieh Mussavi Rizi, Yechao Meng, Xiyue Han, Nam-Hwui Kim, Ce Yang, Shuoshuo Liu, Chapman Lau, Wah-Tung Lau, Ruyi Pan, Dennis Deng, Jason Chia, Jingyi Wang, Rui Jie, Tishawa Pearson, Tamrah Brown, Shimeng Huang, Siqi Chen, Banafsheh Lashkari, Reza Mehrizi, Sophie He, Shahab Pirnia, Menglu Che, Qihuang Zhang, Takaaki Koike, Cong Jiang, Lingyu Cai, Danqiao Guo, Jingyue Huang, Yiran Wang and Lijia Wang.

I would not be who I am today without my lifelong friends, who have stayed with me throughout the years. I would like to thank Linh Pham, Loan Tran, Dao Phan for our 14-year and counting friendship. I thank my KAIST seniors Huy Hoang and Linh Nguyen for suggesting me University of Waterloo and for offering me great help when I first arrived in Canada. I thank my senior Nham Le and junior Duong Nguyen for their welcome, hospitality, and Vietnamese treats. It would now be remiss of me not to acknowledge my other seniors and juniors at KAIST: Viet Phuong Nguyen, Chanh Nguyen, Minh Nguyen, Thuong Nguyen, Thuy Nguyen, Thao Vu, Manh Do, Duc Pham, and many more. I thank them for keeping in touch and sharing their Ph.D. experiences with me, and encouraging me throughout this journey. I appreciate Youssef Medhat Aboutaleb and Sillas Teixeira Gonzaga for their sincere brotherhood and care. I thank Hien Do, My Bui, Nguyen Trang, Nguyen Quynh Nga, Huyen Tran, Thao Ly Nguyen, Cong Hoang, Tuan Lai, Ngoc Tran, Vinh Vu, Louis Long Nguyen, Bivan Alzacky Harmanto, and Sanghun Byeon for their friendship, nice words and encouragement. Finally, meeting my fellow Vietnamese: Duy Nguyen, Thi Xuan Vu, Minh Chau Nguyen, and Truong Le at University of Waterloo made me feel I am not alone here.

My Ph.D. journey would not be the same without my partner Ningsheng Zhao. I

am grateful for our discussions that taught me so much about love, life, and statistics. I would like to thank his company, patience, love, joy, and unwavering support.

Last but not least, I would like to thank my family: my father Bui Ngoc Hung, my mother Pham Thi Hoa, and my dear sister Bui Phuong Thao. I am deeply grateful for their patience and tolerance during my darkest time of depression. I am forever indebted to their fostering, life lessons, company, unconditional love, trust, and support. I would also like to extend my gratitude to my other family members. I thank my grandmothers for being healthy, my late grandfather for his kindness, and my relatives for caring for me and accompanying my parents while I was not there.

## Dedication

To my beloved grandmother, Nguyen Thi Nuoi, for being an inspiring role model of wisdom, sincerity, hard work, and persistence.

# Table of Contents

**List of Figures** xiv

**List of Tables** xvii

**List of Abbreviations** xix

**1 Introduction** 1

1.1 Basic Concepts and Notation . . . 2

1.1.1 Networks . . . 2

1.1.2 Designed Experiments . . . 3

1.2 Design and Analysis of Experiments on Networks . . . 5

1.2.1 Problem Setting . . . 6

1.2.2 Model-Based Approaches . . . 8

1.2.3 Design-Based Approaches . . . 11

1.3 Outline of the Thesis . . . 17

1.4 Contributions . . . 18

**2 General Additive Network Effects Model** 19

2.1 Motivation . . . 20

2.2 General Additive Network Effects (GANE) Model . . . 21

2.2.1 The GANE Model . . . 21

2.2.2 Model Specifications . . . 23

2.2.3 Causal Interpretation . . . 28
2.2.4 Hypothesis Testing . . . 31
2.2.5 Design Criteria . . . 32
2.3 Quasi-Maximum Likelihood Inference . . . 33
2.3.1 Estimation . . . 33
2.3.2 Asymptotic Results . . . 35
2.3.3 Inference for Causal Quantities . . . 36
2.4 Simulations . . . 36
2.4.1 The Distribution of the Estimates . . . 37
2.4.2 Hypothesis Testing . . . 39
2.4.3 Model Misspecification . . . 42
2.5 Conclusions . . . 45

**3 Analysis of Network Experiments with Binary Outcomes 47**
3.1 Binary GANE Extension . . . 47
3.1.1 Estimation . . . 48
3.1.2 Inference . . . 49
3.1.3 Global Treatment Effect . . . 49
3.2 Simulations . . . 50
3.2.1 The Distribution of the Estimates . . . 50
3.2.2 Hypothesis Testing . . . 51
3.2.3 GTE Estimation Under Model Misspecification . . . 54
3.3 Application to the Agricultural Insurance Data . . . 60
3.3.1 Data Description . . . 60
3.3.2 Our Analysis . . . 62
3.4 Conclusion . . . 65

**4 Optimal Bayesian Designs for Network A/B Testing** **67**
4.1 Design Criterion . . . 68
4.1.1 The Model-Based Design Problem . . . 68
4.1.2 Bayesian Design Criterion . . . 72
4.2 Design Construction Algorithms . . . 72
4.2.1 Meta-heuristic Search . . . 73
4.2.2 Bayesian Optimization . . . 76
4.2.3 Graph-cluster Randomization . . . 81
4.3 Simulations . . . 84
4.3.1 Response-Generating Models . . . 84
4.3.2 Other Simulation Details . . . 87
4.3.3 Results . . . 87
4.4 Conclusions . . . 92

**5 Conclusion and Future Research** **94**

**References** **96**

**APPENDICES** **107**

**A Appendices for Chapter 2** **108**
A.1 Mathematical Details for Section 2.3 . . . 108
A.1.1 Proof of Lemma 2.1 . . . 108
A.1.2 Assumptions Needed for Asymptotic Results . . . 109
A.1.3 Proof of Theorem 2.2 . . . 110
A.2 Additional Simulation Results . . . 117
A.2.1 Summary of the Fixed Designs . . . 117
A.2.2 Additional Simulation Results for Section 2.4.1 . . . 118
A.2.3 Additional Simulation Results for Section 2.4.2 . . . 123

**B Appendices for Chapter 3** **124**

B.1 Variance Derivation for the MLE . . . . . 124

B.1.1 Asymptotic Variances . . . . . 124

B.1.2 Robust Clustered Variances . . . . . 125

B.2 Additional Simulation Results . . . . . 126

B.2.1 Summary of the Fixed Designs . . . . . 126

B.2.2 Simulation Results on Distributions of Estimates for the Probit Model . . . . . 127

B.2.3 Simulation Results on Hypothesis Testing for the Probit Model 130

**C Appendices for Chapter 4** **132**

C.1 Comparison among Balanced Graph Clustering Algorithms . . . . . 132

C.2 Convergence of Monte-Carlo Approximations . . . . . 133

C.3 Running Times of the Algorithms Considered . . . . . 133

C.4 Other Design Characteristics . . . . . 135

# List of Figures


1.1 Boxplots of the effective sample sizes of $1,000$ random graph-cluster designs as $\theta$ changes. The designs are generated on three real-life networks. . . . . 15

1.2 Percentages of analyzed units having propensity scores $\pi_i(\varepsilon_k)$ less than 0.05 with respect to varying $\theta$. The boxplots are based on the results of $1,000$ random graph-cluster designs generated on three real-life networks. . . . . 16

2.1 Histograms of the degree distribution of the Caltech Facebook network. 26

2.2 The distribution of parameter estimates of the POW-DEG specification on the Caltech Facebook network with $\boldsymbol{\beta} = (0, 1, 0.5, 0.1)^\top$ and $\lambda \in \{0.5, 0.75, 1.00, 1.25\}$ over 1,000 simulation runs. . . . . 38

2.3 The variances of the estimates (left axes, lines) and coverage rates (right axes, bars) of POW-DEG specification on the Caltech Facebook network with $\boldsymbol{\beta} = (0, 1, 0.5, 0.1)^\top$ and $\lambda \in \{0.5, 0.75, 1.00, 1.25\}$ over 1,000 simulation runs. . . . . 38

2.4 (upper) The distribution of parameter estimates of the HOM specification with $\mu = 0$, $\tau = 1$, $\gamma_T = 0.5$, $\rho_T = \rho_C = 0.1$ over 1,000 simulation runs. (lower) The corresponding variances of the estimates (left axes, lines) and coverage rates (right axes, bars). . . . . 40

2.5 Rejection rates of hypothesis tests for POW-DEG specification on the Caltech Facebook network with varying parameters. . . . . 41

2.6 Model misspecification simulation results. The horizontal axis corresponds to the data-generating model while the vertical axis corresponds to the estimating model. . . . . 44

3.1 The distribution of parameter estimates for the POW-DEG specification of Model (3.1) with logit link over 1,000 runs, where $\boldsymbol{\beta} = (-2, 1, 0.5, 0.1)^\top$ and $\lambda \in \{0.25, 0.5, 0.75\}$. . . . . 52

3.2 The variances of the estimates (left axes, lines) and coverage rates (right axes, bars) for the POW-DEG specification of Model (3.1) with logit link over 1,000 simulation runs $\boldsymbol{\beta} = (-2, 1, 0.5, 0.1)^\top$ and $\lambda \in \{0.25, 0.5, 0.75\}$. . . . . 53

3.3 Rejection rates of hypothesis tests for POW-DEG specification of Model (3.1) with logit link and varying parameters. . . . . 55

3.4 Performances of different GTE estimators under Model (3.6) for the Caltech network. . . . . 58

3.5 Performances of different GTE estimators under Model (3.6) for the UMichigan network. . . . . 59

3.6 The within-village experimental design of the agricultural insurance experiment, adapted from Figure 1.1 of Cai et al. (2015). The numbers of households in each group in the raw data are given in brackets. . . 61

3.7 The distribution of the number of households in each village in the processed data set. . . . . 62

4.1 Efficiency of designs found by different algorithms under different models and networks with respect to the balanced randomization algorithm. . . . . 89

4.2 Characteristics of designs found by different algorithms for each model and network. . . . . 91

A.1 Simulation results of the POW-DEG specification on the Caltech Facebook network with $\mu = 0, \tau = 0.5, \gamma_T = 0.1, \gamma_C = 0.0$ and $\lambda \in \{0.5, 0.75, 1.00, 1.25\}$ over 1,000 simulation runs. . . . . 119

A.2 (upper) The distribution of parameter estimates of the POW-DEG specification on the UMichigan Facebook network with $\mu = 0, \tau = 1.0, \gamma_T = 0.5, \gamma_C = 0.1$ and $\lambda \in \{0.5, 0.75, 1.00, 1.25\}$ over 1,000 simulation runs. (lower) The corresponding variances of the estimates (left axes, lines) and coverage rates (right axes, bars). . . . . 120

A.3 (upper) The distribution of parameter estimates of the POW-DEG specification on the UMichigan Facebook network with $\mu = 0, \tau = 0.5, \gamma_T = 0.1, \gamma_C = 0.0$ and $\lambda \in \{0.5, 0.75, 1.00, 1.25\}$ over 1,000 simulation runs. (lower) The corresponding variances of the estimates (left axes, lines) and coverage rates (right axes, bars). . . . . 121

A.4 (upper) The distribution of parameter estimates of the HOM specification with $\mu = 0, \tau = 0.5, \gamma_T = 0.1, \rho_T = \rho_C = 0.0$ over 1,000 simulation runs. (lower) The corresponding variances of the estimates (left axes, lines) and coverage rates (right axes, bars). . . . . 122

A.5 Rejection rates of hypothesis tests for HOM specification on the Caltech and UMichigan Facebook networks with varying parameters. . . 123

B.1 Simulation results for the POW-DEG specification of Model (3.1) with probit link on the Caltech network over 1,000 runs. . . . . 128

B.2 (Upper) Distribution of estimates (upper), (Lower) variances and coverage rates for the POW-DEG specification of Model (3.1) with probit link on the UMichigan network over 1,000 runs. . . . . 129

B.3 Rejection rates of hypothesis tests for POW-DEG specification of Model (3.1) with probit link and varying parameters. . . . . 131

C.1 Results of simulations investigating the number of draws $L$ in Monte Carlo approximations. . . . . 134

C.2 Additional design characteristics results. . . . . 136

# List of Tables


1.1 Summary statistics of networks used in this thesis. . . . . 16

2.1 Parameters for the simulation in Section 2.4.3. . . . . 43

3.1 Model fitting results. The coefficients are rounded to the nearest 4 digits. Significance codes ***, **, *, and . correspond to significance levels 0.001, 0.01, 0.05, and 0.1, respectively. Robust clustered standard errors are given in brackets. Fixed effect estimates of individual villages are not directly of interest and thus are not displayed in this table. . . . . 66

4.1 Our implementation of tabu search. . . . . 75

4.2 Our implementation of simulated annealing. . . . . 76

4.3 Our implementation of the genetic algorithm. . . . . 77

4.4 Our implementation of deep surrogate Bayesian optimization. In our simulations, we use $m_{\text{neighbor}} = 100$. . . . . 79

4.5 Our implementation of deep reinforcement learning. . . . . 80

4.6 Our implementation of the Tree-Parzen estimator. We choose $m_{\text{initial}} = 1000$ and $m_{\text{candidate}} = 100$ for our simulations. . . . . 82

4.7 Parameter values and priors for models used in our simulations. . . . . 86

A.1 Summary statistics of the fixed designs used in the simulations of Section 2.4. . . . . 117

B.1 Summary statistics of the fixed designs used in the simulations of Section 3.2. . . . . 127

C.1 Performance of different clustering algorithms. Each entry is the average value over 30 runs. . . . . 133
C.2 The running times of each algorithm averaged over 30 runs. . . . . 135

# List of Abbreviations

**SUTVA** The Stable Unit Value Treatment Assumption.

**iid** independent and identically distributed.

**Enron** This refers to the network data `enron` in package `igraphdata` in R, which is transformed to become undirected and simple. Summary statistics of the network is given in Table 1.1.

**Caltech** This refers to the Caltech Facebook friendship network retrieved from the Network Repository Rossi and Ahmed (2015). The network contains snapshots of Facebook friendships in Caltech some time in 2005. The network is made undirected and simple. Summary statistics of the network is given in Table 1.1.

**UMichigan** This refers to the UMichigan Facebook friendship network retrieved from the Network Repository Rossi and Ahmed (2015). The network contains snapshots of Facebook friendships in UMichigan some time in 2005. The network is made undirected and simple. Summary statistics of the network is given in Table 1.1.

**HOM** The Homophily Model, i.e. Model (2.4).

**LNE** The Linear Network Effects Model proposed by Parker et al. (2017), i.e., Model (2.1).

**FNE** The Fraction Neighborhood Exposure Model used by Gui et al. (2015) which uses the percentage of treated neighbors to model the network effect, i.e., Model (2.5).

**LAV** The Local Average Model that uses the average of neighbors' outcomes to model the network effects, i.e. Model 2.6.

**LAG** The Local Aggregate Model that uses the sum of neighbors' outcomes to model the network effects, i.e. Model 2.7.

**POW-DEG** The power-transformed LNE model, i.e. Model 2.9.

**SUTVA** The model $Y_i = \mu + \tau + \epsilon_i$ where $\epsilon_i$'s are iid $\mathcal{N}(0, \sigma^2)$.

**GTE** The global treatment effect.

**DTE** The direct treatment effect.

**ITE** The indirect treatment effect.

**OLS** Ordinary Least Squares.

**ML** Maximum Likelihood.

**GANE** General Additive Network Effects.

**BFGS** Broyden-Fletcher-Goldfarb-Shanno.

**AIC** Akaike Information Criterion.

**POW-DEG-2** The POW-DEG specification but with different power coefficients $\lambda$ for each network effect term.

**RMSE** Root Mean Squared Error.

**MSE** Mean Squared Error.

**DiM** Difference-in-Means.

**CNAR** Conditional Network Autoregressive Model (4.6).

**BNTAR** Binary Network-Temporal Autoregressive (BNTAR) Model (4.5).

**NS** Normal Sum Model (4.15).

# Chapter 1

# Introduction

> *"We are all connected to everyone and everything in the universe. Therefore, everything one does as an individual affects the whole. All thoughts, words, images, prayers, blessings, and deeds are listened to by all that is."*
>
> - Serge King -

The world is an interconnected body where all entities are inseparable from one another. There exist connections and hence networks in every corner of life. Scientists have conducted studies on networks in logistics systems like transportation (Xu and Harriss, 2008) and waterways (Sattar et al., 2019); in biological bodies such as the brain (Xia et al., 2013) and proteins (Rual et al., 2005); in the citations of scientific papers (Börner et al., 2012); and in social structures like businesses, (Krebs, 1996; Owen-Smith and Powell, 2004), organizations (Tichy et al., 1979; Lewis and Sexton, 2004), and online social networks (Bakshy et al., 2012a,b; Bapna and Umyarov, 2015). It is thus insufficient, in many cases, to study individual entities separately without considering the connections among them.

This has been the motivation for the study of networks in the past century (Kolaczyk and Csárdi, 2014). However, the adoption of network studies in statistics has not been very popular until the recent computer age of the twenty-first century. One of the reasons was the lack of data, especially in biology and social sciences. To analyze a network, one must have a network at hand. This often requires network-constructing studies, for example, surveys where each individual nominates their connections (Latkin et al., 1995; Ennett et al., 2006), or research that reveals biological correlations among genes (Franke et al., 2006; Sardiello et al., 2009), or proteins

(Rual et al., 2005; Stelzl et al., 2005).

In recent decades, advances in science and technology have enabled us to collect, store, and manage data in much bigger quantities and with better quality. As a result, network data has also become more readily available. One of the most prominent examples comes from online social networks. From the SixDegrees.com website in 1997 to the global networks of Facebook and Twitter in 2006 (Boyd and Ellison, 2007), nowadays, social networks have become a part of our daily lives. They are platforms for connections, communications, networking, media, and propaganda. Consequently, there is an increasing number of problems that these social network companies face, from privacy issues (Hoadley et al., 2010), and misinformation (Christofides et al., 2009) to the selection of operational algorithms to improve their products (Xu et al., 2015). These problems often require the companies to conduct many experiments on their networks (Goel, 2014; Xu et al., 2015) in order to make informed and strategic decisions. This has motivated recent research on experimentation on networks (Bond et al., 2012; Goel, 2014; Eckles et al., 2016; Gupta et al., 2019; Larsen et al., 2023). In this chapter, we will introduce the basic concepts and current literature in this area. We will also present an outline of the thesis and the contributions it makes.

## 1.1 Basic Concepts and Notation

Experiments on networks consist of two components: experiments and networks. In the following subsections, we will introduce basic concepts from the fields of network analysis and experimental design. We will also define the notation that will be used throughout the thesis.

### 1.1.1 Networks

A *network* is defined by its *nodes* and the *links*, or connections, among the nodes. In graph theory, networks are referred to as *graphs*, nodes as *vertices*, and links as *edges*. In this thesis, these terms will be used interchangeably.

Formally, a network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of a node set $\mathcal{V}$ and an edge set $\mathcal{E}$. Nodes are labeled by positive integers, i.e., $\mathcal{V} = \{1, 2, ..., n\}$, where $0 < n \in \mathbb{N}$ is the number of nodes in the network. The edge set $\mathcal{E}$ contains pairs of nodes that are connected in $\mathcal{G}$

$$\mathcal{E} = \{\{i, j\} : \text{node } i \text{ is connected to node } j \text{ in } \mathcal{G}, 1 \leq i, j \leq n\}.$$

An edge can also encode direction. If a node $i$ is connected to node $j$, we say that there is a *directed edge* from node $i$ to node $j$. Likewise, if node $j$ is connected to node $i$, there is a directed edge from node $j$ to node $i$. In this case, it is possible that node $i$ is connected to node $j$ but node $j$ is not connected to node $i$. (Think about one-way roads!) However, if the edge has no direction, in the sense that if node $i$ is connected to node $j$ implies node $j$ is also connected to node $i$, we say that $\{i, j\}$ is an *undirected edge*. A network consisting of directed edges is called a *directed network*, while a network consisting of only undirected edges is called an *undirected network*.

If there is an undirected edge between node $i$ and node $j$, we say that node $i$ is a *neighbor* of node $j$ and vice versa. For each node $i$, the number of neighbors it has is called the *degree* of node $i$. If a node $i$ is connected to itself, we say that node $i$ has a *self-loop*. If there is more than one edge from node $i$ to node $j$, we call the edges a *multi-edge*. A *simple network* is a network without any self-loops or multi-edges.

A simple network $\mathcal{G}$ can be represented by its *adjacency matrix* $\mathbf{A}$ which contains information about the edges of $\mathcal{G}$.

$$\mathbf{A} = [A_{ij}]_{1 \le i,j \le n} \qquad \text{such that} \qquad A_{ij} = \begin{cases} 1 & \text{if } \{i, j\} \in \mathcal{E} \\ 0 & \text{otherwise.} \end{cases}$$

Since there are no self-loops in a simple network, the diagonal elements of $\mathbf{A}$ are all zeros, i.e., $A_{ii} = 0$, $1 \le i \le n$. For an undirected network $\mathcal{G}$, $\mathbf{A}$ is symmetric.

In a network, edges may contain more information than simply which nodes they are connecting. For example, edges may encode the strength of relationships, flow capacities, the number of edges for each multi-edge, etc. In this case, instead of using the binary adjacency matrix $\mathbf{A}$, the network can be represented using a *weight matrix* $\mathbf{W} = [W_{ij}]_{1 \le i,j \le n}$, where $W_{ij} \in \mathbb{R}$ is the weight of the edge $\{i, j\}$ indicating the strength of association between nodes $i$ and $j$. A zero-weighted edge implies the edge's nonexistence. A network with weighted edges is called a *weighted network*.

The above provided a brief overview of some important concepts in network analysis and graph theory. For a more detailed discussion, see Kolaczyk and Csárdi (2014).

### 1.1.2 Designed Experiments

In science and engineering, we often want to study a system and understand how it changes. For example, a farmer may want to know how to adjust the water level,

the amount of fertilizer, and the temperature for a bountiful harvest; an automotive engineer wants to know which engines can generate the highest acceleration; or a pharmacist might want to know whether a new drug is effective in treating a certain disease. In these examples, experiments are often required to help answer such causal questions of interest.

Formally, an *experiment* is an empirical investigation in which specific inputs of a system are *intentionally* manipulated so that the experimenters can observe whether, and quantify by how much, certain outputs of the system change (Montgomery, 2019). Such outputs of the system are called the *outcomes* (also called *outputs* or *responses*) of the experiment. The physical entities whose outcomes are measured in the experiment are called *experimental units*. For example, experimental units can be farming plots in an agricultural setting, cars in a manufacturing setting, and subjects in a pharmaceutical setting.

The input variables to be manipulated by the experimenters are called the *factors* of the experiment. Different values of a factor that are chosen to be used in the experiment are called *levels* of that factor. In the crop example, the factors may be water, fertilizer, and temperature, and the levels are certain amounts of these factors. A unique combination of the levels of the experimental factor(s), which forms a complete condition in which the system is observed, is called a *treatment*. When a treatment is applied to an experimental unit, we say that a *run* (or a *replication*) is conducted. The *experimental design* defines the treatments, determines the allocation of the treatments among the experimental units, and the order of the runs so that the experiment can help answer the question of interest *accurately* and *precisely*. In designing an experiment, one often needs to take into account certain constraints of the experiment, typically time and cost. Hence, experimental design is usually viewed as a constrained optimization problem.

Three basic principles of experimental design are *randomization*, *replication*, and *blocking*. First, depending on the experimental constraints, replication should be applied so that each treatment is assigned to more than one experimental unit. This helps to provide a more precise estimate of the treatment effects as well as an understanding of the variability of the outcomes, i.e., the experimental error. Second, since the outcomes of an experiment can be affected by other (uncontrolled) inputs besides the experimental factors, to neutralize the effects of these inputs, experiments are usually randomized, i.e., the experimental units are randomly assigned to the treatments. Third, whereas randomization handles uncontrolled inputs, to control for other nuisance inputs (i.e., the inputs that may affect the experiment outcomes but are not of analysis interest), it is helpful to group the experimental runs into

homogeneous blocks based on those inputs, and then allocate different treatments *within* each block. This helps remove the effects of the nuisance inputs and enhances the accuracy of treatment comparisons. Altogether, the three principles of experimental design enable us to conduct experiments with better internal and external validity. For a more detailed discussion about experimental design, see Montgomery (2019) and Wu and Hamada (2021). In this thesis, we focus on randomization and replication. There has been considerably less work that considers blocking in experiments on networks (Koutra et al., 2021), thus it is a promising avenue for future research.

## 1.2 Design and Analysis of Experiments on Networks

An *experiment on a network* is one in which the experimental units reside within a network. For example, users on social network websites connect by "friending" or "following" one another; streets in a city are physically connected; students in a university connect by taking the same courses; and researchers connect via collaborations. The following examples showcase possible applications of network experimentation.

**Example 1.1.** (LINKEDIN) LinkedIn is a social network for professionals. On LinkedIn, users can publish and update their CVs, connect with their past classmates and colleagues, and seek job opportunities or career advice. An important component of LinkedIn is the "People You May Know" feature, where LinkedIn users are presented with other users to whom they possibly want to connect. The system uses the LinkedIn network data and builds models that estimate the propensity of connections between two users. The effectiveness of these models can be measured by the number of connections a user initiates using the "People You May Know" feature. LinkedIn regularly updates the feature, and they conduct experiments on their users to evaluate the updates (Yin, 2021). △

**Example 1.2.** (SURVEILLANCE CAMERA) La Vigne et al. (2011) discuss a situation when the experimenter wants to evaluate the effectiveness of surveillance cameras in reducing crime rates. The experimenters selected certain streets for their experiment. Surveillance cameras are randomly installed on some streets, and not on others. The crime rate of each street will then be measured and analyzed to determine the effectiveness of the cameras. This is a network experiment because the streets are connected with one another in the traffic system. △

**Example 1.3.** (COSMETICS ADVERTISEMENT) As social network services have grown in popularity, it is common for businesses to utilize these platforms for advertising (Leskovec et al., 2007). Consider a cosmetic company that wants to know whether a new ad is more effective compared to an older one. In this case, suppose the company conducts an experiment on a social network platform's users. And, suppose the response of interest is the revenue from cosmetic sales. There is one experimental factor, which is the ad version. The factor has two levels: the old version and the new version. We will use this setting to provide illustrative examples throughout the thesis. △

Often, the goal of experimentation is to understand the effects of the treatments on the outcomes of individual experimental units, i.e., the *treatment effect*. However, in the context of an experiment on a network, the experimenter may also want to understand the influence of the network on the outcomes, i.e., the *network effect*, and/or how the treatment mediates such influence. In the following subsection, we will introduce the problem of network experimentation and the notation we use in this thesis. Then, in the subsections that follow, we review existing research on the subject.

### 1.2.1 Problem Setting

Consider an experiment to be conducted on $n$ experimental units, labeled $1, 2, ..., n$. Assume that the associations among the experimental units are described by a network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. In particular, the nodes of $\mathcal{G}$ represent the experimental units, and the edges of $\mathcal{G}$ represent their connections. Current studies (e.g., Gui et al., 2015; Eckles et al., 2016; Aronow et al., 2017; Basse and Airoldi, 2018) generally assume that the network is undirected and simple, and is observed and unchanged throughout the experiment. Let $\mathbf{A}$ denote the adjacency matrix of $\mathcal{G}$ and $\mathbf{K}$ denote the diagonal matrix whose diagonal element $K_{ii}$ is the degree (i.e., the number of neighbors) of unit $i$, for $i = 1, 2, ..., n$.

Let $Y_i$ denote the experimental outcome of unit $i$, for $i = 1, 2, ..., n$, and let $\mathbf{Y} = (Y_1, Y_2, ..., Y_n)^\top$ denote the vector of outcome values. In this thesis, we consider both cases where $\mathbf{Y}$ is continuous (Chapter 2) or binary (Chapter 3). The literature about experiments on networks mainly focuses on A/B testing problems, i.e., experiments with two treatments: the new condition whose effect is the interest of the experimenter (called the *treatment*), and the existing condition that the experimenter wants to compare the new condition to (called the *control*). So far, we have

seen that the word "treatment" has two meanings, one is the general notion of experimental conditions, and one is the specific "new" condition in the A/B test setting. For our purposes, we refer to the former meaning when we talk about the "treatment assignment" or in the plural form "treatments". And we refer to the latter meaning when we talk about the "treatment" on its own. We denote the treatment assignment vector by $\mathbf{Z}$, where $Z_i = 1$ if unit $i$ is assigned to treatment and $Z_i = 0$ if unit $i$ is assigned to control, $i = 1, 2, ..., n$. In this thesis, we will focus on investigating A/B tests. Cases where the experiment has one factor at multiple levels (i.e., A/B/n tests) or multiple factors (multivariate tests) are discussed as possible extensions in Chapter 5.

In the A/B test setting, once the treatment assignment is determined, treatments are operated concurrently. As such, we do not need to specify the order of the runs because the treatments are run in parallel. In this case, the *design* of the experiment refers to the decision of which experimental units (nodes) will be assigned to treatment, and which will be assigned to control, which is equivalent to the selection of the binary treatment assignment vector $\mathbf{Z}$. In this thesis, we will refer to $\mathbf{Z}$ as either the treatment assignment vector or the design, interchangeably. In contrast to the design of the experiment, the *analysis* of the experiment refers to the process of using observed experimental data to understand and infer the experimental results according to the goals set by the experimenters. Thus, the (statistical) problem of network experimentation concerns both the design and the analysis of experiments on networks.

To design and analyze an experiment on a network, one needs to make certain assumptions. We will start with the common assumptions made in general experimental design settings. The randomization and replication principles require randomly assigning each treatment to multiple experimental units (nodes) in the network. Hence, it is assumed that we can assign each unit to whichever treatment we want. We further assume that all experimental units respond and we can measure their outcomes. The problem of nonresponse is beyond the scope of the thesis. Another assumption that is usually made in the design and analysis of experiments is the Stable Unit Treatment Value Assumption (SUTVA) (Rubin, 1980), which states that the treatment assignment of an experimental unit will *not* affect the outcome of another. Under this assumption, many classical design and analysis methodologies have been developed, such as randomized block designs, factorial designs, and fractional factorial designs, etc. (Montgomery, 2019). However, in our setting where individual units are connected with one another on a network, this assumption often will not hold. The following example demonstrates how SUTVA may be violated in

the network experiment setting.

**Example 1.4.** Let us continue with Example 1.3. Suppose that the new ad is shown to Tracy, leading Tracy to buying more products from the cosmetics company. Now if Connie is a friend of Tracy on the social network, Tracy may share information about the products on their page, or introduce the products to Connie. In this case, even if Connie is only presented with the old ad, they may still buy more products from the company. In this sense, the treatment assignment of Tracy affected Connie's outcome. This violates SUTVA. △

As Example 1.4 shows, when the experimental units reside on a network $\mathcal{G}$, SUTVA may be violated. In particular, for an experiment on a network, the outcome $Y_i$ of a unit $i$ is not only affected by its own treatment assignment $Z_i$ as in traditional settings, but it may also be influenced by $\mathbf{Z}$, the treatment assignments of everyone in the network. As a consequence, common techniques in classical experimentation may be inadequate in the network setting, and more sophisticated approaches are required.

There have been several approaches considered in the growing literature on the design and analysis of experiments on networks. These approaches differ in many ways, from the goals and assumptions of the experiment to the design and analysis techniques proposed. In general, these approaches to the design and analysis of such experiments can be classified into two categories: model-based and design-based approaches. In the following subsections, we will review prominent research and discuss the advantages and disadvantages of each of these two experimental frameworks.

### 1.2.2 Model-Based Approaches

Model-based approaches begin by assuming a model for the experimental outcomes (i.e., the outcome vector $\mathbf{Y}$). With the assumed model, the analysis procedure can be formed and corresponding properties of the analysis can be studied. To accommodate such an analysis, design criteria are defined a priori according to the experimenters' interests, and algorithms to find satisfactory designs are developed.

#### 1.2.2.1 Models

Some studies suggest a *linear network effects model* (Gui et al., 2015; Parker et al., 2017; Koutra et al., 2021) in which the outcome of a unit is modeled by a linear

function of the *treatment effect* and the *network effect*. In particular, Parker et al. (2017) introduce the model

$$Y_i = \mu + \tau Z_i + \gamma_1 \sum_{j=1}^{n} A_{ij} Z_j + \gamma_2 \sum_{j=1}^{n} A_{ij}(1 - Z_j) + \epsilon_i, \tag{1.1}$$

where $\epsilon_i \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2)$, $\mu$ is the baseline outcome (i.e., the outcome when no treatment is applied and there is no network effect), and $\tau$ is the direct effect of the treatment. The network effect is accounted for using $\gamma_1$ and $\gamma_2$, which quantify the influence from the neighbors who are assigned to treatment or control, respectively. While (1.1) uses the *number* of neighbors to define the network effect, Gui et al. (2015) use the *percentage* of neighbors assigned to treatment

$$Y_i = \mu + \tau Z_i + \frac{\gamma}{K_{ii}} \sum_{j=1}^{n} A_{ij} Z_j + \epsilon_i. \tag{1.2}$$

Thus, (1.2) assumes that there is no influence from the neighbors assigned with control, while in (1.1), such an effect is parametrized by $\gamma_2$. We can see that both (1.1) and (1.2) are ordinary linear regression (OLS) models. If the model is correctly specified, these estimators are unbiased for their respective parameters, given a treatment assignment vector $\mathbf{Z}$. The variances of these estimators can be estimated following the usual OLS framework.

Another popular type of model is the *network-correlated outcome model* (Basse and Airoldi, 2018; Pokhilko et al., 2019). Similar to the linear network effects model, the treatment effect in the network-correlated outcome model is modeled by an additive term of individual treatment assignment. However, instead of modeling the network effects as functions of the treatment assignment vector, the network-correlated outcome model posits a correlation structure based on the network structure. This is motivated by a conjecture that connected units often share similar characteristics, which influence and thus create correlations in the experimental outcomes. This is referred to in network science as *homophily*. One version of a network-correlated outcome model is proposed by Basse and Airoldi (2018)

$$Y_i = U_i + \tau Z_i + \sum_{j=1}^{n} A_{ij} U_j + \epsilon_i, \tag{1.3}$$

where $U_i \overset{\text{iid}}{\sim} \mathcal{N}(\mu, \eta^2)$ denotes a latent variable that can be thought of as the "intrinsic baseline outcome" of unit $i$ in the absence of the network. In this model, the network

effect is modeled using the sum of the $U$ values of the neighbors and is unrelated to the treatment assignment vector. Another outcome model, suggested by Pokhilko et al. (2019), is adapted from the conditional autoregressive (CAR) models used in the spatial statistics literature

$$Y_i = \mu + \tau Z_i + \epsilon_i \quad \text{where} \quad \epsilon_i | \boldsymbol{\epsilon}_{-i} \sim \mathcal{N}\left(\rho \sum_{j \neq i} \frac{A_{ij}\epsilon_j}{K_{ii}}, \frac{\sigma^2}{K_{ii}}\right). \tag{1.4}$$

Here, $\boldsymbol{\epsilon}$ denotes the vector of $\epsilon_i$'s and $\boldsymbol{\epsilon}_{-i}$ denotes the vector without the $i$-th element. In this model, the correlation is modeled via the noise vector $\boldsymbol{\epsilon}$, where $\epsilon_i$ follows a normal distribution with the mean being the average of the $\epsilon$ values of its neighbors. It can be shown that under this model,

$$\boldsymbol{\epsilon} \sim \mathcal{N}\left(0, \sigma^2(\mathbf{K} - \rho\mathbf{A})^{-1}\right). \tag{1.5}$$

Thus, parameters of (1.3) and (1.4) can be estimated using Maximum Likelihood (ML), with inference being conducted in accordance with the usual ML procedures. Note that the above is not an exhaustive list of models, but instead a short introduction to build intuition and familiarity. Many other models exist, and we review and use some later in the thesis.

#### 1.2.2.2 Design

As experimenters want to design the experiment so that the analysis can be done accurately and efficiently, the design problem can be formulated into an optimization problem where we find the treatment assignment vector $\mathbf{Z}$ (called the *optimal design*) that optimizes a *design criterion*. In the model-based framework, design criteria can be defined based on the inferential properties of the parameters' estimators. For example, D-optimality is a popular design criterion in the experimental design literature (Pukelsheim, 2006; Fedorov and Leonov, 2013). A D-optimal design aims to *minimize* the determinant of the variance-covariance matrix of the parameter estimators, which is positively related to the volume of the confidence region of the parameters. In network experimentation, D-optimal designs have also been constructed with respect to certain models (Pokhilko et al., 2019; Koutra et al., 2021). While D-optimal designs aim to minimize the confidence region for *all* parameters, experimenters may have *specific* parameters of interest. In this case, a design criterion can be defined based on the variances of these parameters' estimators. For example, $\text{Var}[\hat{\tau}]$ has been considered as the design criterion for (1.1), (1.3), and (1.4). Parker et al. (2017) also consider $\text{Var}[\hat{\gamma}_1 - \hat{\gamma}_2]$ as a design criterion for (1.1).

Having selected the design criterion, we can proceed to find the optimal design. Certain design criteria for certain models can be written in a closed form that allows optimal designs to be found using integer programming (Pokhilko et al., 2019; Koutra et al., 2021; Zhang and Kang, 2022). However, more generally, design criteria do not have a closed-form formula, and heuristic optimization algorithms need to be used. A conceptually straightforward solution is *exhaustive search* (Parker et al., 2017), in which we calculate the design criterion for each of all possible designs, and choose the design with the best design criterion value. However, in an A/B test where each unit can be assigned to either treatment or control, there is a total of $2^n$ possible designs. This number will increase exponentially as the size of the network increases. Since most real-world networks have very large $n$, the aforementioned solution, although conceptually simple, is computationally prohibitive in practice.

To address this problem, Parker et al. (2017) consider two strategies. The first strategy is an exchange algorithm that iteratively changes the treatment assignment in the direction of optimizing the design criteria. Unfortunately, there is no guarantee about the convergence of this approach and early stopping criteria need to be used to control the running time. The second strategy is a random search, which involves randomly generating a large number of designs and choosing a design with the best design criterion value. Note that both procedures do not guarantee an optimal design, and so sub-optimal designs will be returned. Via simulations, Parker et al. (2017) observe that, given the same number of designs evaluated, the random search approach, despite its simplicity, yields similar or even better designs compared to the exchange algorithm.

To summarize, in the model-based approaches, treatment and network effects are defined via model parameters. In terms of analysis, parameter estimation and inference can be conducted using usual techniques such as the OLS or ML procedures. In terms of design, optimality criteria can be constructed based on the model and a good design can be found using suitable algorithms. A major drawback of model-based approaches, however, is the reliance on the model assumptions, where model misspecification is the main concern.

### 1.2.3 Design-Based Approaches

#### 1.2.3.1 Literature Review

In the model-based approaches, the model plays a central role, in which its distributional assumption and inferential properties drive the construction of the design

and analysis. On the other hand, in the design-based approaches, the focus is on the *design strategy*, i.e., the randomization scheme that generates the design $\mathbf{Z}$, and inferential uncertainty comes from the randomization scheme instead of any distributional assumption.

In the literature, design-based approaches have been motivated by the main goal of the network experiment. In many cases, the experimenters are interested in answering the question of whether or not to deploy the treatment to the whole population (network). In Example 1.1, it is the question of whether to apply the new "People You May Know" algorithm to the whole population of LinkedIn users; and in Example 1.3, it is whether to show all users the new ad. Therefore, an important goal of network experiments is to estimate the *global treatment effect* (GTE), i.e., the difference in the average outcome of the units when the whole network is assigned with treatment compared to when the whole network is assigned with control. Mathematically, the GTE is defined as

$$\frac{1}{n}\sum_{i=1}^{n}\Big\{\mathbb{E}[Y_i|\mathbf{Z}=\mathbf{1}_n]-\mathbb{E}[Y_i|\mathbf{Z}=\mathbf{0}_n]\Big\}, \tag{1.6}$$

where $\mathbf{1}_n$ and $\mathbf{0}_n$ are column vectors in $\mathbb{R}^n$ of ones and zeros, respectively. A naive experimental strategy would be to assign all experimental units in the network to treatment and then compare the average outcome to the average outcome before the experiment was conducted. However, temporal effects may confound the results. Unfortunately, both situations where the whole graph is assigned with treatment or with control, i.e., the factual and counterfactual outcomes, cannot be observed simultaneously. Therefore, in the experiment, only some of the units can be assigned to treatment while the others should be assigned to control, i.e., $\mathbf{Z}\neq\mathbf{1}_n$ and $\mathbf{Z}\neq\mathbf{0}_n$.

To estimate the GTE, a popular design strategy called *graph-cluster randomization* (Ugander et al., 2013; Gui et al., 2015; Eckles et al., 2016) has been proposed. The graph-cluster randomization strategy partitions the network into a reasonably large number of *clusters* of densely connected units and then randomly assigns all units in each cluster to either treatment or control. The dense connections among the clusters' units enable each unit to be surrounded by neighbors that share the same treatment assignment as theirs, hence mitigating network interference. In this case, the design strategy aims to simulate "two universes" (Ugander et al., 2013), one where the whole network is assigned to treatment and and one where all the units are assigned to control. Different graph clustering strategies, such as the $\epsilon$-net clustering algorithm (Ugander et al., 2013), the balanced label propagation algorithm (Gui et al., 2015), and the random walk-based algorithm (Backstrom and Kleinberg,

2011) have been studied and used in network experimentation. For a general review on graph clustering, see Fortunato (2010); Fortunato and Hric (2016); Abbe (2017); or Doreian et al. (2020).

In the design-based approaches, the analyses are conducted under the exposure framework (Gui et al., 2015; Eckles et al., 2016; Aronow et al., 2017). In this framework, units are classified into $R$ exposures $\varepsilon_1, \varepsilon_2, ..., \varepsilon_R$ depending on the treatment assignment vector $\mathbf{Z}$. A possible set of exposures is given in Example 1.5.

**Example 1.5.** Gui et al. (2015) propose the "neighborhood exposure" which comprises the following two exposures:

- the *treatment exposure:* units $i$ such that $Z_i = 1$ and the proportion of neighbors of $i$ assigned to treatment, $\sum_{j=1}^{n} A_{ij}Z_j/K_{ii}$, is greater than or equal to $\theta$; and
- the *control exposure:* units $i$ such that $Z_i = 0$ and the proportion of neighbors of $i$ assigned to treatment is less than or equal to $1 - \theta$,

where $0 \leq \theta \leq 1$ is a threshold chosen a priori. In this case, $R = 2$, and $\varepsilon_1$ denotes the treatment exposure and $\varepsilon_2$ denotes the control exposure. $\triangle$

Let us denote the *potential outcome* of unit $i$ if unit $i$ belongs to exposure $\varepsilon_k$ by $Y_i(\varepsilon_k)$. In this case,

$$\mu(\varepsilon_k) = \frac{1}{n}\sum_{i=1}^{n} Y_i(\varepsilon_k)$$

is the average potential outcome under exposure $\varepsilon_k$. Now, the experimenters may be interested in contrasts of these average potential outcomes, for example, between exposures $k$ and $l$

$$\mu(\varepsilon_k) - \mu(\varepsilon_l) = \frac{1}{n}\sum_{i=1}^{n} Y_i(\varepsilon_k) - \frac{1}{n}\sum_{i=1}^{n} Y_i(\varepsilon_l).$$

Note that we cannot obtain $\mu(\epsilon_k)$ directly from the experimental data because not every unit $i$ belongs to exposure $\varepsilon_k$. If $i$ does not belong to $\varepsilon_k$, $Y_i(\varepsilon_k)$ is unobserved. Hence, $\mu(\varepsilon_k)$ needs to be estimated by for example, inverse-probability-weighted difference-in-mean estimators, such as the Horvitz-Thompson estimator (Horvitz and

Thompson, 1952; Eckles et al., 2016; Aronow et al., 2017), or the Hajek estimator (Hájek, 1971; Eckles et al., 2016)). The Horvitz-Thompson estimator of $\mu(\epsilon_k)$ is

$$\hat{\mu}_{\text{HT}}(\varepsilon_k) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}(i \in \varepsilon_k)\frac{Y_i}{\hat{\pi}_i(\varepsilon_k)}, \tag{1.7}$$

where the *propensity score* $\pi_i(\varepsilon_k)$ is the probability that unit $i$ belongs to exposure $k$. The estimator $\hat{\pi}_i(\varepsilon_k)$ is obtained by randomly generating different designs from the design randomization scheme (such as the graph-cluster randomization), and calculating the proportion of times that unit $i$ belongs to exposure $k$. Unbiased or conservative estimators of the variances of these estimators have been proposed and studied (Aronow et al., 2017; Sussman and Airoldi, 2017; Li et al., 2019b).

#### 1.2.3.2 Limitations

The design-based approaches are nonparametric, model-free, and thus robust to model misspecification. This is a major advantage compared to the model-based approaches. However, there still remain challenges in the design-based approaches. First, the definitions of the exposures need to be specified by the experimenters. For example, if the experimenters want to use the two exposures defined in Example 1.5, they need to choose the threshold $\theta$ a priori. In particular, threshold $\theta$ needs to be set so that the potential outcomes under the exposures $Y_i(\varepsilon_k)$, $k = 1, 2$ are close to the unobserved (counterfactual) potential outcomes in the two "universes" where all units are assigned to the same treatment assignment (treatment or control). However, it is often unknown which value of $\theta$ will make the two sets of potential outcomes close enough. The effect of exposure misspecification still requires future research (Sävje, 2023).

The choice of $\theta$ also affects the effective sample sizes. Depending on the value of $\theta$, there may be some units that cannot be classified into any of the exposures. For example, any treated unit $i$ having less than $\theta \times 100\%$ of neighbors assigned to treatment will not be classified into either of the exposures defined in Example 1.5. In this case, the outcomes of these units will be discarded in the analysis. As a result, despite having many units involved in the experiment, only a fraction of the outcomes will be analyzed and the rest will be wasted. This happens regularly in real-life networks, even when graph-cluster randomization is used with the intention to simulate the treatment and control "universes". Due to the typical structure of a social network, it is unlikely that the network can be divided into similar-sized disconnected clusters (Chin, 2019). As a result, units from one cluster connect

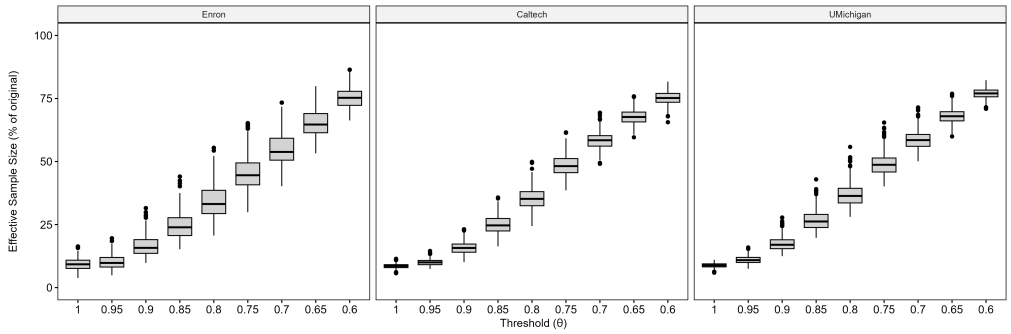



Figure 1.1: Boxplots of the effective sample sizes of $1,000$ random graph-cluster designs as $\theta$ changes. The designs are generated on three real-life networks.

with units from other clusters. These units will not be surrounded by neighbors with the same treatment assignment if the two clusters are assigned to different treatments. Thus, they may not be classified into either of the exposures in Example 1.5, especially if the threshold $\theta$ is large. However, as discussed above, it is preferred that $\theta$ is large so that the two exposures mimic the two "universes" more closely. Figure 1.1 illustrates the relationship between effective sample sizes (in percentage) and the threshold $\theta$. Each panel corresponds to one real-life network. The Enron network is retrieved from the `igraphdata` package in `R`, containing a network of emails exchanged among upper managers at Enron between 1998 and 2001. The Caltech and UMichigan networks are retrieved from the Network Repository (Rossi and Ahmed, 2015) and contain snapshots of Facebook friendship networks at Caltech and University of Michigan in 2005 (Traud et al., 2012). All networks are made simple and undirected. Summaries of these networks are given in Table 1.1. In each panel, each boxplot shows the distribution of the effective sample sizes (with respect to a certain value of $\theta$) of $1,000$ designs randomly generated using graph-cluster randomization. In particular, we use the balanced label propagation algorithm (Gui et al., 2015) to perform the network clustering. The algorithm returns 9 clusters for the Enron network, 12 clusters for the Caltech network, and 13 clusters for the UMichigan network. The designs are generated by randomly selecting half of the clusters and assigning all units in those clusters to treatment and the rest to control. We can see that even when we set $\theta = 0.75$ as suggested by Gui et al. (2015), about half of the network units will be discarded because they do not qualify for either of the two exposures in Example 1.5. The reduced effective sample size does not just reduce precision; if the discarded units are systematically different from the analyzed units, the analysis will be biased.

| Networks | # of nodes ($n$) | # of edges ($m$) |
|---|---|---|
| Enron network | 184 | 2097 |
| Caltech Facebook network | 770 | 16,656 |
| UMichigan Facebook network | 3,749 | 81,903 |

Table 1.1: Summary statistics of networks used in this thesis.

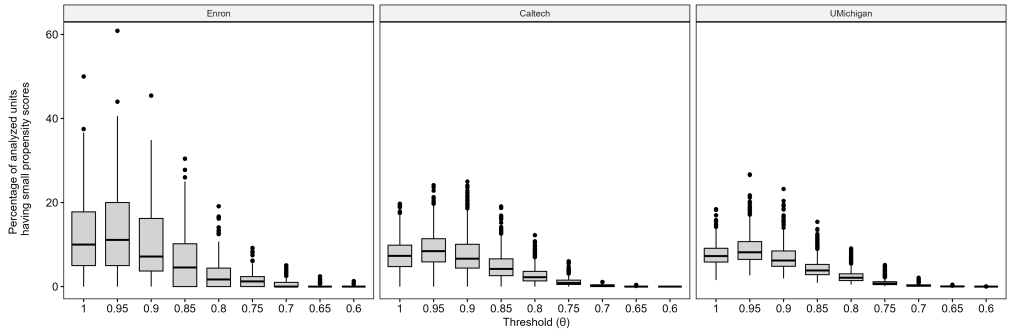



Figure 1.2: Percentages of analyzed units having propensity scores $\pi_i(\varepsilon_k)$ less than 0.05 with respect to varying $\theta$. The boxplots are based on the results of $1,000$ random graph-cluster designs generated on three real-life networks.

As discussed in Section 1.2.3.1, to compute inverse-probability-weighted estimators for the GTE, the propensity scores $\pi_i(\varepsilon_k)$, $i = 1, ..., n$, $k = 1, 2$ need to be estimated using Monte Carlo simulations (Aronow et al., 2017). This can be computationally challenging for large networks. Moreover, it is well-known that inverse-probability weighted estimators can be unstable if the propensity scores are small for certain units (Schafer and Kang, 2008; Crump et al., 2009; Khan and Tamer, 2010). To illustrate the small propensity score problem in the design-based analysis, for each network in Table 1.1, we generate 1,000 designs using graph-cluster randomization in the same fashion as used in Figure 1.1. For each design, we calculate the percentage of analyzed units (i.e., units that are categorizable into either the treatment or control exposure in Example 1.5) having propensity scores less than 0.05. The distributions of such quantities in the 1,000 designs are plotted as boxplots in Figure 1.2. Each boxplot corresponds to a certain value of $\theta$. We can see that the problem of small-propensity scores happens quite often, especially with high values of $\theta$. In some cases, the proportions of analyzed units with small propensity scores can range up to 50%. Together with the problem of effective sample size, this adds challenges to the analysis in design-based approaches.

## 1.3 Outline of the Thesis

The design-based approaches have been embraced in industry due to their model-free property. However, as discussed in Section 1.2.3.2, they also have unique challenges. On the other hand, the model-based approaches, with their inference advantages, remain promising. This thesis will focus on addressing the main weakness of the model-based approaches, i.e., the sensitivity to model misspecification. While Chapters 2 and 3 extend the modeling flexibility in the analysis, Chapter 4 enhances the robustness of experimental design to parameter specifications.

Existing model-based methodologies differ in their model specifications. In Chapter 2, we attempt to unify the existing models by proposing a general class of models called the general additive network effects (GANE) model. This general model provides greater flexibility in modeling while keeping the analysis simple. We then outline a model-based analysis framework for network experimentation which includes causal quantities, hypothesis tests, and design criteria. The estimation procedure and inferential properties of a subfamily of the GANE model are derived and studied via simulations. Simulation studies also provide evidence that the POW-DEG specification of the GANE model we propose in Section 2.2.2.3 performs well under model misspecification.

The GANE model in Chapter 2 is built for experiments with continuous outcomes. However, experiments with binary outcomes are also common in practice. For instance, the outcome might be the decision to purchase a product or adopt a policy. In this case, the use of continuous-outcome models is inappropriate. In Chapter 3, we consider a binary extension of the GANE model via a generalized linear specification with the Bernoulli distribution and binary link functions. Estimation and inferences of such binary models can be conducted using maximum likelihood theory. The performance of different specifications of the model is investigated via simulations. Our method is then applied to analyze the agricultural insurance experiment from Cai et al. (2015).

We shift our focus to the design of network experiments in Chapter 4. Under the model-based framework, the design is constructed by optimizing a design criterion formulated based on the postulated model. However, due to the complex specifications of models for network experiments, design criteria based on these models often involve unknown parameters. Thus, we propose the use of a Bayesian design criterion to incorporate prior information on these unknown parameters. However, these design criteria do not have a closed-form formula, which limits the use of classical optimal design algorithms. We thus propose the use of meta-heuristic algorithms

and Bayesian optimization techniques to construct the Bayesian optimal designs. We adapt these algorithms to our specific design problem and evaluate their performances on various models and data sets. We summarize the characteristics of good designs with respect to each model and provide general design guidelines for practitioners.

Chapter 5 concludes the thesis with a summary of our contribution and directions for future research.

## 1.4 Contributions

The chapters of this thesis correspond to work that has already been published or submitted for publication, or in preparation as follows.

- Chapter 2: Bui, T., Steiner, S.H., Stevens, N.T. (2023). General Additive Network Effect Models. *New England Journal of Statistics and Data Science*, 1-19, doi 10.51387/23-NEJSDS29.
- Chapter 3: Bui, T., Steiner, S.H., Stevens, N.T. (2024+). Analysis of Network Experiments with Binary Outcomes. *In preparation.*
- Chapter 4: Bui, T., Steiner, S.H., Stevens, N.T. (2024+). Optimal Bayesian Designs for Experiments on Networks. *Submitted to Technometrics.*

# Chapter 2

# General Additive Network Effects Model

Many statistical methods have been developed based on the assumption that the data of interest arise from, or can be aptly described by, a model. If a model's assumptions are found satisfactory, it can be very useful for learning about the system of interest (Flassig and Schenkendorf, 2018). In particular, additive models have been a fundamental tool in applied statistics (Rencher and Schaalje, 2008), especially in the design and analysis of experiments, thanks to their simple interpretation and generalizability. Therefore, it is intuitive to turn to linear models when it comes to the design and analysis of experiments on networks.

In fact, different linear models have been introduced and applied to observational (Manski, 1993; Christakis and Fowler, 2007; Bramoullé et al., 2009) and experimental (Gui et al., 2015; Parker et al., 2017; Advani and Malde, 2018; Basse and Airoldi, 2018) data on networks. In this chapter, we attempt to unify and extend these linear models by introducing the *general additive network effects* (GANE) model. Although no longer linear in the parameters, the model is additive in terms of causal quantities of interest. We then outline a framework to design, analyze, and interpret network experiments based on the GANE model. This includes the definition and estimation of causal quantities, hypotheses concerning the significance of these effects, and design criteria for optimal design. Last, we review, propose, and investigate several model specifications under the GANE framework and conduct simulations to study their characteristics.

## 2.1 Motivation

Consider an A/B test in which the experimental units are nodes of a fixed and observed network $\mathcal{G}$. In particular, we consider the problem setting of Section 1.2.1. To design and analyze such an experiment, Parker et al. (2017) introduce the linear network effects model

$$Y_i = \mu + \tau Z_i + \gamma_1 \sum_{j=1}^{n} A_{ij} Z_j + \gamma_2 \sum_{j=1}^{n} A_{ij}(1 - Z_j) + \epsilon_i, \tag{2.1}$$

where $\epsilon_i \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2)$. The parameters of Model (2.1) are easy to interpret: $\mu$ is the baseline outcome, $\tau$ is the treatment effect, $\gamma_1$ is the effect of having an additional treated neighbor and $\gamma_2$ is the effect of having an additional controlled neighbor. Because Model (2.1) is additive and linear, its parameters can be estimated using ordinary least squares in a straightforward manner. The inference procedure thus follows accordingly. Therefore, with Model (2.1), practitioners will find it easy to apply this familiar linear regression framework to analyze their network experiments.

It is noticeable that Model (2.1) uses the numbers of neighbors, i.e., the node degrees, to model the network effect. This can be clearly seen if we rewrite Model (2.1) into

$$\begin{aligned} Y_i &= \mu + \tau Z_i + (\gamma_1 - \gamma_2) \sum_{j=1}^{n} A_{ij} Z_j + \gamma_2 \sum_{j=1}^{n} A_{ij} + \epsilon_i \\ &= \mu + \tau Z_i + (\gamma_1 - \gamma_2) \sum_{j=1}^{n} A_{ij} Z_j + \gamma_2 K_{ii} + \epsilon_i. \end{aligned} \tag{2.2}$$

In this rearrangement of parameters, $\gamma_1 - \gamma_2$ parametrizes the effect of the number of neighbors that are assigned to treatment. This term involves both the treatment assignment vector $\mathbf{Z}$ and the network structure $\mathbf{A}$. Thus, we can think of this term as modeling the *interaction* between the network and the treatment. The other network effect term uses the node's degree $K_{ii}$ with parameter $\gamma_2$. When $\gamma_2 > 0$, the more neighbors unit $i$ has, the higher the expected outcome $Y_i$, and vice versa. This is reasonable in cases such as: a social network user with more friends will be expected to engage with the platform more, or an author with more collaborators will be expected to publish more papers. Such an effect can be referred to as the *popularity effect*.

Besides popularity, applied researchers also suggest other ways to model network effects for observational data. For example, the average outcomes of neighbors are commonly used to model a unit's outcome (Manski, 1993; Bramoullé et al., 2009; Gui et al., 2015; Advani and Malde, 2018). This is often referred to as the *homophily effect* (Shalizi and Thomas, 2011), which reflects the conjecture that "you are the average of the people around you" or that "people have the desire to conform to the average outcomes of their neighbors" (Advani and Malde, 2018). Yet, some may argue that the average outcome is not sufficient to represent the network effect because it does not reflect the popularity effect. In that case, the aggregate outcome, i.e., the sum of neighbors' outcomes, can be used as an alternative. Example 2.1 lists examples where the sum of neighbors' outcomes can be used to model the network effects.

**Example 2.1.** According to Advani and Malde (2018), modeling the network effect with the sum of neighbors' outcomes is suitable in cases such as

- a consumer decides to buy a certain product if more of their friends also decide to buy the product;
- a person's perceived cost of committing a crime is lower if their neighbors also engage in crimes (Bramoullé et al., 2014);
- a student will tend to put more effort into studying if their friends put more effort (Calvó-Armengol et al., 2009). △

The number of neighbors, the average outcomes of neighbors, and the sum of outcomes of neighbors are only some examples of the many proposed methods to model network effects (Christakis and Fowler, 2007; Advani and Malde, 2018). This underlines the need for a general model framework for experimental data that provides flexibility to model the network effect according to the experimenters' interest or domain knowledge, and at the same time inherits the clear inference procedure and interpretability of Model (2.1).

## 2.2 General Additive Network Effects (GANE) Model

### 2.2.1 The GANE Model

To generalize the idea of the linear network effects model proposed by Parker et al. (2017) and incorporate different functional forms of the network effects, we propose

the *general additive network effects* (GANE) model. Let $\mathbf{D} = \{\mathbf{A}, \mathbf{Z}, \mathbf{Y}, \mathbf{X}\}$, where $\mathbf{X}$ is the $n \times p$ matrix of the units' possible covariates. The GANE model is written as

$$Y_i = \mu + \tau Z_i + f_{T,i}(\mathbf{D}, \boldsymbol{\eta}) + f_{C,i}(\mathbf{D}, \boldsymbol{\eta}) + \epsilon_i, \tag{2.3}$$

where $f_{T,i}$ and $f_{C,i}$ model the network effects experienced by unit $i$ coming from treated (with $Z = 1$) and controlled units (with $Z = 0$), respectively. The separation of network effects according to different sources (treatment or control) makes it convenient to interpret and compare the sizes of network effects induced by different treatments. Example 2.2 demonstrates how a linear model in the network experiment literature can be re-formulated under the proposed GANE model and how the separation of network effects can be useful in practice.

**Example 2.2. The Homophily (HOM) Model:** Gui et al. (2015) incorporates the number of neighbors assigned to treatment (which they call the *spill-over effect*), and the homophily effect into a linear additive model:

$$Y_i = \mu + \tau Z_i + \gamma \sum_{j=1}^{n} A_{ij} Z_j + \frac{\rho}{K_{ii}} \sum_{j=1}^{n} A_{ij} Y_j + \epsilon_i; \tag{2.4}$$

Model (2.4) can be reparametrized under the GANE framework with

$$f_{T,i}(\mathbf{D}, \boldsymbol{\eta}) = \gamma_T \sum_{j=1}^{n} A_{ij} Z_j + \frac{\rho_T}{K_{ii}} \sum_{j=1}^{n} A_{ij} Y_j Z_j,$$
$$f_{C,i}(\mathbf{D}, \boldsymbol{\eta}) = \frac{\rho_C}{K_{ii}} \sum_{j=1}^{n} A_{ij} Y_j (1 - Z_j),$$

where $\boldsymbol{\eta} = (\rho_T, \rho_C, \gamma_T)^\top$. Note that to make the GANE parametrization equivalent to Model (2.4), we need to impose the constraints $\rho_T = \rho_C = \rho$ and let $\gamma_T = \gamma$. In this sense, Model (2.4) assumes that unit $i$'s neighbors, no matter treated or controlled, contribute with equal weights to the homophily effect experienced by unit $i$. On the other hand, without the constraint, the GANE framework allows the weights to be different among treated and controlled neighbors. This can be useful in cases where the treatment alters the strength of the network effect.

Consider Tracy and Connie in Example 1.4. Suppose Tracy is assigned to treatment while Connie is assigned to control (i.e., the new ad is shown to Tracy the old ad is shown to Connie). Now, due to the new ad, Tracy may spend more money on the company's cosmetic products. Even though she does not watch the new

ad, Connie, as a friend of Tracy, may observe Tracy's increased spending, and her spending may in turn become more similar to Tracy's spending than other friends' spending. In this case, the treatment increases the strength of the homophily effect (the $\rho$ term in Model (2.4)), and the proposed GANE model allows the modeling of such a phenomenon. $\triangle$

Note that the network effect functions $f_T$ and $f_C$ are functions of the experimental data $\mathbf{D}$ and can admit a parameter vector $\boldsymbol{\eta}$. That is, the network effects may depend on the network structure represented by the adjacency matrix $\mathbf{A}$, the treatment assignment vector $\mathbf{Z}$, the outcomes vector $\mathbf{Y}$ such as in Example 2.2, and/or covariates $\mathbf{X}$ as illustrated in Example 2.3 below.

**Example 2.3.** Continuing with the cosmetic ad setting of Example 1.3, the cosmetics purchase of a user $i$ may depend on the gender, age, or knowledge about cosmetics of the user's neighbor, which can be contained in a covariate matrix $\mathbf{X}$. $\triangle$

Using the GANE model, researchers can select the forms of the network effect functions $f_T$ and $f_C$ according to their domain knowledge or other pragmatic considerations. They will also be able to infer and interpret the parameters in a similar fashion to the linear network effects model (2.1). Thus, the proposed model inherits the interpretability of Model (2.1) while allowing the network effects to be modeled more flexibly.

By proposing the GANE model, we take a model-based approach to the analysis of experiments on networks. Compared to design-based approaches, although the model relies on potentially restrictive assumptions about the network effects (via functions $f_T$ and $f_C$), it is able to (i) utilize all units in the experiments; (ii) allow researchers to model different network interference patterns; and (iii) make predictions on the outcomes. Therefore, if correctly specified, the proposed model will be a useful tool to understand and quantify the treatment and network effects in the experiments.

### 2.2.2 Model Specifications

As previously discussed, with the proposed GANE model, the functional forms of network effects can be chosen flexibly. These choices depend on the domain knowledge or preferences of the experimenters, and/or how well these models fit the data. In this section, we will discuss some possibilities and considerations when specifying a GANE model.

#### 2.2.2.1 A Unification of Existing Linear Models

Most existing models for observational and experimental responses on networks are linear in their parameters, e.g., the linear-in-means models (Manski, 1993; Bramoullé et al., 2009), the linear network effects model (Parker et al., 2017), and so on. By adopting an additive structure, the GANE model unifies these models under a common framework. In Section 2.2.1, Example 2.2 showed how the HOM model can be written as a GANE model. In the examples below, we list some other existing models and illustrate how they can be written under the GANE framework.

**Example 2.4. The Linear Network Effect (LNE) Model:** Model (2.1) proposed by Parker et al. (2017) is a special case of the proposed GANE model with

$$f_{T,i}(\mathbf{D}, \boldsymbol{\eta}) = \gamma_T \sum_{j=1}^{n} A_{ij} Z_j \quad \text{and} \quad f_{C,i}(\mathbf{D}, \boldsymbol{\eta}) = \gamma_C \sum_{j=1}^{n} A_{ij}(1 - Z_j)$$

being the number of neighbors of unit $i$ that are assigned to treatment and control, respectively. In this case, $\boldsymbol{\eta} = (\gamma_T, \gamma_C)^\top$. $\triangle$

**Example 2.5. The Fraction Neighborhood Exposure (FNE) Model:** Gui et al. (2015) consider using the percentage of treated neighbors to model the network effect. This is equivalent to a GANE specification with

$$f_{T,i}(\mathbf{D}, \boldsymbol{\eta}) = \frac{\gamma_T}{k_i} \sum_{j=1}^{n} A_{ij} Z_j, \quad \text{and} \quad f_{C,i}(\mathbf{D}, \boldsymbol{\eta}) = 0. \tag{2.5}$$

Note that we cannot let $f_C$ be the percentage of controlled neighbors because then the model terms will be linearly dependent and hence inestimable. $\triangle$

**Example 2.6. The Local Average (LAV) Model:** As discussed in Section 2.1, in the economics literature, neighbors' average outcome has been used to model the network effect under the conjecture that units "have the desire to conform to the average outcomes of their neighbors" (Advani and Malde, 2018). Under the GANE framework, we can also use the local average (i.e., the average of neighbors' outcomes) to model the network effects, that is

$$f_{T,i}(\mathbf{D}, \boldsymbol{\eta}) = \rho_T \frac{\sum_{j=1} A_{ij} Z_j Y_j}{\sum_{j=1} A_{ij} Z_j}, \quad \text{and} \quad f_{C,i}(\mathbf{D}, \boldsymbol{\eta}) = \rho_C \frac{\sum_{j=1} A_{ij}(1 - Z_j) Y_j}{\sum_{j=1} A_{ij}(1 - Z_j)}. \tag{2.6}$$

Recall that we assumed the graph $\mathcal{G}$ to be simple with no self-loop, i.e., $A_{ii} = 0$ for all $i$. Thus in the LAV model, $Y_i$ does not appear on both sides of the equation. In this LAV model, $\rho_T$ and $\rho_C$ are coefficients of the average outcomes of treated and controlled neighbors, respectively. $\triangle$

**Example 2.7. The Local Aggregate (LAG) Model:** In Example 2.1, we discussed some situations where the sum, instead of the average, of the neighbors' outcomes might be preferred to model the network effect. Under the GANE framework, we can specify the network effect functions accordingly:

$$f_{T,i}(\mathbf{D}, \boldsymbol{\eta}) = \rho_T \sum_{j=1}^{n} A_{ij} Z_j Y_j, \quad \text{and} \quad f_{C,i}(\mathbf{D}, \boldsymbol{\eta}) = \rho_C \sum_{j=1}^{n} A_{ij}(1 - Z_j) Y_j. \quad \triangle \quad (2.7)$$

#### 2.2.2.2 The Use of Covariates

There are at least two ways that covariates can be incorporated into the GANE model. First, covariates can be used to calculate a weight matrix $\mathbf{W}$, whose elements represent the strength of connections among nodes in the graph. Using a weight matrix $\mathbf{W}$ instead of the dichotomous adjacency matrix $\mathbf{A}$ is practical if we posit that the network effects are heterogeneous among the connections (edges). Indeed, the concept of homophily tells us that people are more likely to connect and interact with others who are similar to them (Shalizi and Thomas, 2011). Thus, it may be reasonable to expect that the closer $\mathbf{x}_i$ and $\mathbf{x}_j$ are, the more important $i$ and $j$ are to each other. In this case, a weight matrix can be constructed using some measure of distance between the covariates. For example,

$$W_{ij} = \frac{A_{ij}}{||\mathbf{x}_i - \mathbf{x}_j||},$$

the operator $||\cdot||$ denotes a norm (e.g., Euclidean norm) of vectors. With this weighting scheme, as the distance between the covariates of units $i$ and $j$ increases, the influence they have on each other decreases. We can then replace the adjacency matrix $\mathbf{A}$ by this weight matrix $\mathbf{W}$ in the models discussed in Section 2.2.2.1.

Another possible use of covariates is to add them into the model equations, i.e.,

$$Y_i = \mu + \tau Z_i + \gamma_1 f_{T,i}(\mathbf{D}, \boldsymbol{\theta}) + \gamma_2 f_{C,i}(\mathbf{D}, \boldsymbol{\theta}) + \mathbf{x}_i^T \boldsymbol{\delta} + \epsilon_i, \quad (2.8)$$

where $\boldsymbol{\delta}$ is the vector of linear coefficients for covariate $\mathbf{x}_i$. This is similar to *regression adjustment* in causal inference literature for observational data. We do not consider regression adjustment in this section, however, the extension with regression adjustment is straightforward.

#### 2.2.2.3 Nonlinear Growth

Existing response models for connected units usually consider cases where the node degrees are small. For example, spatial statistics literature works on areal data where a district can only have a handful of neighboring districts. Parker et al. (2017) consider agricultural plots arranged on a lattice. Other applied work on peer effects (Christakis and Fowler, 2007; Cai et al., 2015; Li et al., 2019a,b) construct the network edges based on proximal relationships such as households or friendship or nomination (e.g., name up to three friends of yours). In all these cases, the number of connections for each node is likely small.

However, this does not apply to some social networks. Preferential attachment (also known as cumulative advantages or "the rich get richer") is a common phenomenon in networks (Vázquez, 2003). In particular, in a growing network, new nodes are more likely to connect with nodes with many existing connections, compared to those with only a few. This eventually makes several nodes become "popular" with many connections while the rest of the nodes have far fewer. Figure 2.1 shows the histogram of degrees for the Caltech network described in Table 1.1. We can see that there is a large number of nodes having less than 20 neighbors while a few nodes have more than 200 connections. In fact, the range of degrees is from 0 to 248. This range is usually large for large social networks.

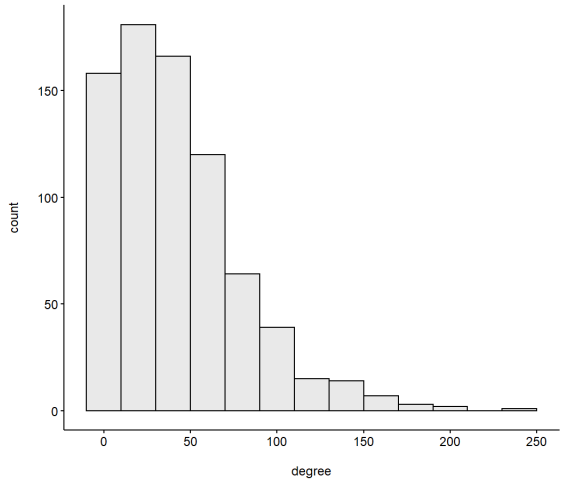



Figure 2.1: Histograms of the degree distribution of the Caltech Facebook network.

As noted, many existing models used in the literature assume linearity of the network effect. For example, with the LNE model, Parker et al. (2017) assume that

each additional treated neighbor will have an effect of size $\gamma_T$ on the node's response; likewise, each controlled neighbor will have an effect of size $\gamma_C$. Similarly, the LAG model assumes that the network effect is proportional to the sum of the neighbors' outcomes, which creates an approximately linear relationship between the network effect and the degree. However, this can be unrealistic, especially for popular nodes with hundreds of neighbors on a social network, as illustrated in Example 2.8.

**Example 2.8.** Let us continue to consider the two friends Tracy and Connie in Example 1.4. Since Tracy is assigned to treatment, Connie may spend more on buying cosmetic products simply because she is Tracy's friend. However, the increase in money spent by Connie due to her first treated friend may not necessarily be equal to the increase prompted by her 100th treated friend. More likely, the marginal impact due to each additional treated friend may decrease. This conjecture is similar to the law of diminishing marginal utility in economics (Gossen, 1983). △

On the other hand, it is also not reasonable to normalize the network effects by degree as in the FNE or the LAV models since it removes the *popularity* effect, as illustrated in Example 2.9.

**Example 2.9.** Continuing with Example 1.4, suppose Tracy and Connie both have 50% of their connections assigned to the new ad. According to the FNE model, Tracy and Connie will experience the same network effects. However, if Tracy has many more friends than Connie, we would expect Tracy to be more likely to be influenced by her friends. △

Therefore, both a linear growth of network effects with respect to degrees in the LNE and LAG models and a standardizing approach in the FNE and LAV models can be inadequate to model the network effects. This motivates something in between, where the network effects grow *sub-linearly* with respect to node degrees. The GANE framework can easily accommodate this. Example 2.8 further demonstrates the rationale of this idea.

We thus propose modeling the network effects sub-linearly with respect to node degree. We can achieve this by performing a power transformation on the number of neighbors of each treatment assignment

$$Y_i = \mu + \tau Z_i + \gamma_T \left( \sum_{j=1}^{n} A_{ij} Z_j \right)^{\lambda} + \gamma_C \left( \sum_{j=1}^{n} A_{ij} (1 - Z_j) \right)^{\lambda} + \epsilon_i. \tag{2.9}$$

We call this the POW-DEG specification because here the network effects are modeled as powers of the treatment and control degrees. The parameter $\lambda$ serves to

temper the effect of the treatment and control degrees. As discussed above, since a sub-linear growth might be reasonable, we expect $0 < \lambda < 1$. But in the interest of ample flexibility, e.g., super-linear growths, we do not make this assumption. Another possible option is to perform a log transformation

$$Y_i = \mu + \tau Z_i + \gamma_T \log\left(\sum_{j=1}^{n} A_{ij} Z_j\right) + \gamma_C \log\left(\sum_{j=1}^{n} A_{ij}(1 - Z_j)\right) + \epsilon_i. \tag{2.10}$$

This does not require additional parameters, which can be both an advantage and a disadvantage as the rate of growth is considered known. Another possibility is to put a threshold on the network effects

$$Y_i = \mu + \tau Z_i + \gamma_T \min\left(\sum_{j=1}^{n} A_{ij} Z_j, \lambda\right) + \gamma_C \min\left(\sum_{j=1}^{n} A_{ij}(1 - Z_j), \lambda\right) + \epsilon_i, \tag{2.11}$$

for a chosen threshold $\lambda > 0$. With this specification, the network effects are constrained to be at most $\lambda$ and cannot grow beyond this. Yet, in this case, functions $f_T$ and $f_C$ are not smooth. If smooth thresholded growths are desired, we can consider 4-parameter functions, such as log-logistic, log-normal, or Weibull functions from the dose-response literature (Holland-Letz and Kopp-Schneider, 2015); or the variogram functions, such as the exponential, spherical, or Gaussian variogram functions from spatial statistics literature (Cressie, 2015). Nevertheless, it is not our intention to explore and investigate all these possibilities; instead, these are just some suggested specifications of the GANE model that help us achieve a sublinear growth of the network effects.

### 2.2.3 Causal Interpretation

Section 2.2.2 presented a wide range of possibilities when it comes to specifying the GANE model. However, different specifications may lead to different parameter interpretations, depending on the form of the network effect functions $f_T$ and $f_C$. To attain a universal interpretation for the GANE model, we consider using definitions of causal effects from the causal inference literature. In particular, Hudgens and Halloran (2008) give definitions of *direct treatment effect* (DTE), *indirect treatment effect* (ITE), *overall treatment effect* (OTE) based on the comparison of two treatment assignment vectors, say $\mathbf{Z}_1$ and $\mathbf{Z}_2$.

$$\text{OTE}(\mathbf{Z}_1, \mathbf{Z}_2) = \frac{1}{n} \sum_{i=1}^{n} \Big\{ \mathbb{E}[Y_i | \mathbf{Z} = \mathbf{Z}_1] - \mathbb{E}[Y_i | \mathbf{Z} = \mathbf{Z}_2] \Big\}$$

$$\text{DTE}(\mathbf{Z}_1) = \frac{1}{n}\sum_{i=1}^{n}\Big\{\mathbb{E}[Y_i|Z_i=1,\mathbf{Z}_{-i}=\mathbf{Z}_{1,-i}]-\mathbb{E}[Y_i|Z_i=0,\mathbf{Z}_{-i}=\mathbf{Z}_{1,-i}]\Big\},$$
$$\text{ITE}(\mathbf{Z}_1,\mathbf{Z}_2) = \frac{1}{n}\sum_{i=1}^{n}\Big\{\mathbb{E}[Y_i|Z_i=0,\mathbf{Z}_{-i}=\mathbf{Z}_{1,-i}]-\mathbb{E}[Y_i|Z_i=0,\mathbf{Z}_{-i}=\mathbf{Z}_{2,-i}]\Big\}, \tag{2.12}$$

where $\mathbf{Z}_{-i}$ denotes the treatment assignment vector $\mathbf{Z}$ without the $i$th element. Saint-Jacques et al. (2019) give similar definitions, with $\mathbf{Z}_1$ replaced by $\mathbf{1}_n$ and $\mathbf{Z}_2$ replaced by $\mathbf{0}_n$. In that case, the overall treatment effect becomes the global treatment effect (GTE).

There are two difficulties with the definitions in (2.12). First, although it is easy to understand the overall treatment effect, the interpretations of the direct effect and indirect effect are less obvious. Taking the indirect effect as an example, it is not clear what the average of individual node's difference in expected outcome when all other nodes on the network are assigned to treatment versus when they are assigned to control, given the node itself is assigned to control, implies. Second, it is computationally expensive to calculate the direct and indirect effects defined in (2.12) for autoregressive models such as the LAG and LAV models. This is because the summands in (2.12) need to be calculated separately for each $i$, which involves generating the response vector $\mathbf{Y}$ corresponding to $\mathbf{Z} = \mathbf{e}_i$ (the $n$ vector of 1 at the $i$th position and 0 otherwise) and $\boldsymbol{\epsilon} = \mathbf{0}_n$ for each $i = 1, ..., n$. Considering these difficulties, we propose a new set of definitions for causal quantities of interest as presented below.

**Definition 2.1. Global treatment effect (GTE):** As discussed in Section 1.2.3.1, the GTE (1.6) has been treated as the estimand of interest in the design-based framework due to its important interpretation in business decision-making. Indeed, without loss of generality, suppose that higher responses are desired, then a significantly positive treatment effect will serve as evidence supporting a business decision to deploy the treatment on the whole network. Mathematically, the GTE is the overall treatment effect in (2.12) with $\mathbf{Z}_1$ replaced by $\mathbf{1}_n$ and $\mathbf{Z}_2$ replaced by $\mathbf{0}_n$. The terminology *global* used by Chin (2019) is informative because the GTE measures the treatment effect at the *global* level, instead of the individual level, taking into account the structure of the network and possible network effects. Furthermore, it also implies a "global" deployment of the treatment or control. Under the GANE framework, the GTE can be expressed as

$$\text{GTE} = \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}Y_i \,\middle|\, \mathbf{Z}=\mathbf{1}_n\right] - \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}Y_i \,\middle|\, \mathbf{Z}=\mathbf{0}_n\right]$$

$$= \tau + \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\big[f_{T,i}(\mathbf{D}_{\mathbf{Z}=\mathbf{1}_n}, \boldsymbol{\eta})\big] - \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\big[f_{C,i}(\mathbf{D}_{\mathbf{Z}=\mathbf{0}_n}, \boldsymbol{\eta})\big]$$

where the subscripts of $\mathbf{D}$ indicate that the data (including $\mathbf{Y}$) are evaluated when all the experimental units are assigned to treatment ($\mathbf{Z} = \mathbf{1}_n$) or control ($\mathbf{Z} = \mathbf{0}_n$), respectively. The expectations in the second equivalence are necessary when $f_T$ and/or $f_C$ are functions of the outcome $\mathbf{Y}$. As shown, the GTE can be expressed as a function of the model parameters. Moreover, under the GANE framework, the GTE can be decomposed into two components, the direct and indirect treatment effects (to be discussed below), which aid interpretation.

**Definition 2.2. Direct treatment effect (DTE):** We define the direct treatment effect as the expected difference in outcomes when a node is assigned to the treatment versus when a node is assigned to control, keeping the network effects fixed. Note that this definition is based on fixing the network effects, while definitions in (2.12) are based on a fixed treatment assignment vector (Hudgens and Halloran, 2008; Saint-Jacques et al., 2019). With this definition, under the GANE model framework, the direct treatment effect is simply

$$\text{DTE} = \tau. \tag{2.13}$$

The new definition is clear, easy to interpret, easy to calculate, and does not depend on any specific treatment assignment vector. Hence, it can be used across all specifications of the GANE model.

**Definition 2.3. Indirect treatment effect (ITE):** Interest can also lie in quantifying the amount of the global treatment effect induced by the network. We therefore define ITE as the difference between the global treatment effect and the direct treatment effect, which, under the GANE framework, is

$$\text{ITE} = \text{GTE} - \tau = \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\big[f_{T,i}(\mathbf{D}_{\mathbf{Z}=\mathbf{1}_n}, \boldsymbol{\eta})\big] - \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\big[f_{C,i}(\mathbf{D}_{\mathbf{Z}=\mathbf{0}_n}, \boldsymbol{\eta})\big]. \tag{2.14}$$

Hence, the indirect treatment effect can also be interpreted as the difference between the network effect induced by the treatment versus that induced by the control. Although the global treatment effect is the primary interest of most applied studies (Eckles et al., 2016; Chin, 2019), decomposing it into the direct and indirect treatment effects helps us better understand the global treatment effect. Especially when the GTE is zero, we may learn whether this is because the treatment does not have any effect at all or because the direct and indirect effects cancel out.

Overall, compared to the definitions in (2.12), the new set of causal quantities is easier to interpret and understand. Thus, these quantities will be particularly useful in practical applications. In addition, these quantities can be expressed as functions of the GANE model parameters, which is convenient for inference. In particular, when the expectations of $f_T$ and $f_C$ are known, estimates and hypothesis tests for these quantities can be developed based on parametric inference associated with the model. Overall, these definitions provide a systematic framework to interpret network experiments under a wide variety of outcome models, aligning the analysis results obtained from the model-based approaches with the ones obtained from the design-based approaches.

### 2.2.4 Hypothesis Testing

Corresponding to the causal quantities defined in Section 2.2.3, we identify a set of hypotheses that the experimenters may be interested in.

**Hypothesis test 1:** (DIRECT TREATMENT EFFECT) $H_{01}$ : DTE $= \tau = 0$ is the null hypothesis that the direct treatment effect is 0. That is, keeping the network effect fixed, a node's outcome is the same no matter if it is assigned to treatment or control.

**Hypothesis 2:** (SUTVA) $H_{02} : f_T = f_C = 0$ is the null hypothesis that there is no network effect and SUTVA is satisfied.

**Hypothesis 3:** (INDIRECT TREATMENT EFFECT) $H_{03}$ : ITE $= 0$ is the null hypothesis that the indirect treatment effect is 0, i.e., the network effects from treated and controlled neighbors are the same.

**Hypothesis 4:** (GLOBAL TREATMENT EFFECT) $H_{04}$ : GTE $= 0$ is the null hypothesis that the global treatment effect is 0, i.e., on average, treatment does not have an effect on the nodes' outcomes.

These hypotheses can be better understood using an illustrative example.

**Example 2.10.** In the cosmetic ad setting of Example 1.3, there are several hypotheses that the company may want to test. First, the company may be interested

in learning whether the new ad induces more sales for individual customers who do not engage in social media. This is equivalent to testing $H_{01}$. Second, on a social network, even without any ad, people sometimes share or recommend the cosmetic products that they use, which in turn may increase product sales. This is a result of a network effect and so testing $H_{02}$ helps the company understand if any network effect exists. Third, to see if the new ad affects the existing network effect, the company should compare the network effect induced by the new ad and the existing network effect induced by the old ad. This is done by testing $H_{03}$. Finally, the company wants to decide whether to deploy the new ad for all of their customers on the social network platform. Hence they want to test $H_{04}$. $\triangle$

Since DTE, ITE, and GTE are functions of the model parameters, Hypotheses 1, 3, and 4 can be tested using Wald-type tests. If the model is estimated using M- or Z-estimation (Van der Vaart, 2000), Hypothesis 2 can be tested using a likelihood ratio test or a score test, respectively.

### 2.2.5 Design Criteria

As we can see from the above discussions, parameter estimation is an important step for inference procedures such as interpretation and hypothesis testing. An important element that affects the model estimation that we can manipulate ourselves is the design, i.e., the decision about which nodes are assigned to treatment, and which nodes are assigned to control. This is equivalent to the choice of the treatment assignment vector $\mathbf{Z}$. As discussed in Section 1.2.2.2, in a model-based framework such as the GANE model, design criteria can be defined based on the efficiencies of estimators of the model parameters. These criteria can then be used to evaluate, compare, and select designs. An *optimal design* is a design that optimizes a given design criterion.

In the network experiment context, as discussed in Section 2.2.3, GTE (or possibly DTE or ITE) is the primary quantity of interest. In this thesis, we will focus on setting the mean squared error (MSE) of the GTE estimator as the design criterion. We will defer the discussion of the design problem to Chapter 4. Nevertheless, both the design and analysis of network experiments require point and variance estimation of GANE's model parameters. Thus, in the next section, we will discuss (quasi)-maximum likelihood estimation of the GANE model.

## 2.3 Quasi-Maximum Likelihood Inference

In this section, we develop quasi-likelihood theory for estimation and asymptotic inference in the context of the GANE model. Such theory is not readily available for this type of model with this type of data.

### 2.3.1 Estimation

Different specifications of the GANE model may require different estimation techniques. In fact, many of the GANE specifications can be estimated using (quasi)-maximum likelihood. Note that the term quasi-maximum likelihood is used instead of maximum likelihood when there is no distribution assumption on $\boldsymbol{\epsilon}$. To obtain the (quasi)-likelihood of the outcome vector $\mathbf{Y}$, we consider the family of GANE specifications where the outcome $Y_i$ either (i) does not depend on neighboring outcomes, or (ii) depends linearly on neighboring outcomes. That is,

$$\begin{aligned} f_{T,i}(\mathbf{D}, \boldsymbol{\eta}) &= \rho_T \sum_{j=1}^{n} W_{T,ij} Y_j + \gamma_T g_{T,i}(\boldsymbol{\varphi}), \\ f_{C,i}(\mathbf{D}, \boldsymbol{\eta}) &= \rho_C \sum_{j=1}^{n} W_{C,ij} Y_j + \gamma_C g_{C,i}(\boldsymbol{\varphi}), \end{aligned} \tag{2.15}$$

where $\boldsymbol{\eta} = (\rho_T, \rho_C, \gamma_T, \gamma_C, \boldsymbol{\varphi}^\top)^\top$ and $W_{T,ij}$ (or $W_{C,ij}$) is the $(i,j)^{th}$ element of a pre-specified weight matrix $\mathbf{W}_T$ (or $\mathbf{W}_C$). For example, these weight matrices can be set to the adjacency matrix $\mathbf{A}$. The diagonals of these weight matrices are zero, i.e., $W_{l,ii} = 0$ for $l \in \{T, C\}$ and $i = 1, ..., n$. In addition, $g_{T,i}(\boldsymbol{\varphi})$ and $g_{C,i}(\boldsymbol{\varphi})$ are real-valued functions, possibly depending on the parameter $\boldsymbol{\varphi}$, the experimental data $\mathbf{D}$, but not the outcome vector $\mathbf{Y}$. We can see that Model (2.15) generalizes all model specifications discussed in Section 2.2.2, in which the outcome of an experiment may depend linearly on other unit's outcomes and/or possibly nonlinearly on other covariates. Model (2.15), however, excludes cases where the outcome of unit $i$ is dependent on a *nonlinear* function of the outcome vector $\mathbf{Y}$, which complicates or precludes the (quasi)-maximum likelihood theory.

To perform estimation, we consider the matrix form of Model (2.15) as follows

$$\mathbf{Y} = \mu \mathbf{1}_n + \tau \mathbf{Z} + \big(\rho_T \mathbf{W}_T \mathbf{Y} + \gamma_T \mathbf{G}_T(\boldsymbol{\varphi})\big) + \big(\rho_C \mathbf{W}_C \mathbf{Y} + \gamma_C \mathbf{G}_C(\boldsymbol{\varphi})\big) + \boldsymbol{\epsilon}, \tag{2.16}$$

where $\mathbf{G}_T(\boldsymbol{\varphi})$ and $\mathbf{G}_C(\boldsymbol{\varphi})$ denote the $n \times 1$ vectors of $g_{T,i}(\boldsymbol{\varphi})$ or $g_{C,i}(\boldsymbol{\varphi})$ values respectively. Let $\mathbf{M}(\boldsymbol{\varphi}) = [\mathbf{1}_n \ \mathbf{Z} \ \mathbf{G}_T(\boldsymbol{\varphi}) \ \mathbf{G}_C(\boldsymbol{\varphi})]$ be the model matrix. Further, let $\boldsymbol{\beta} = (\mu, \tau, \gamma_T, \gamma_C)^\top$ and $\boldsymbol{\rho} = (\rho_T, \rho_C)^\top$. Then, the model may be rewritten, isolating for $\mathbf{Y}$ on the left-hand side, as follows

$$\begin{aligned} \mathbf{Y} &= \big(\rho_T \mathbf{W}_T + \rho_C \mathbf{W}_C\big)\mathbf{Y} + \mathbf{M}(\boldsymbol{\varphi})\boldsymbol{\beta} + \boldsymbol{\epsilon}, \\ &= \mathbf{S}(\boldsymbol{\rho})^{-1}\Big(\mathbf{M}(\boldsymbol{\varphi})\boldsymbol{\beta} + \boldsymbol{\epsilon}\Big), \end{aligned} \tag{2.17}$$

where $\mathbf{S}(\boldsymbol{\rho}) = \mathbf{I}_n - \rho_T \mathbf{W}_T - \rho_C \mathbf{W}_C$. The expression in (2.17) is well-defined if and only if $\mathbf{S}(\boldsymbol{\rho})$ is invertible. Lemma 2.1 gives sufficient conditions on $\boldsymbol{\rho}$ so that $\mathbf{S}(\boldsymbol{\rho})$ is nonsingular. Although the condition is based on any matrix norm, in practice, we can use the popular spectral norm (Horn and Johnson, 2012, sec. 5.6) to derive the constraints. The proof of Lemma 2.1 is given in Appendix A.1.1.

**Lemma 2.1.** *If*

$$\max\left(|\rho_T|, |\rho_C|\right) < \frac{1}{||\mathbf{W}_T|| + ||\mathbf{W}_C||},$$

*where* $||\cdot||$ *denotes a matrix norm (Horn and Johnson, 2012, sec. 5.6), then* $\mathbf{S}(\boldsymbol{\rho})$ *is invertible.*

With $\mathbf{Y}$ expressed as in Equation (2.17) and assuming $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}_n, \sigma^2 \mathbf{I}_n)$, the log-likelihood function for $\mathbf{Y}$ is

$$\begin{aligned} \log L(\boldsymbol{\theta}) = \ & -\frac{n}{2}\log(2\pi) - \frac{n}{2}\log(\sigma^2) + \log|\mathbf{S}(\boldsymbol{\rho})| \\ & - \frac{1}{2\sigma^2}\Big(\mathbf{S}(\boldsymbol{\rho})\mathbf{Y} - \mathbf{M}(\boldsymbol{\varphi})\boldsymbol{\beta}\Big)^\top \Big(\mathbf{S}(\boldsymbol{\rho})\mathbf{Y} - \mathbf{M}(\boldsymbol{\varphi})\boldsymbol{\beta}\Big), \end{aligned} \tag{2.18}$$

where $\boldsymbol{\theta} = (\boldsymbol{\rho}^\top, \boldsymbol{\beta}^\top, \boldsymbol{\varphi}^\top, \sigma^2)^\top$ is the vector of all model parameters. If the normality assumption is not made, then (2.18) becomes the *quasi-log-likelihood* (Wedderburn, 1974; Lee, 2004) and the estimators $\hat{\boldsymbol{\theta}}$ that maximize (2.18) are called the quasi maximum likelihood estimators.

To find the maximum likelihood estimates, we take the first-order derivatives with respect to $\boldsymbol{\beta}$ and $\sigma^2$ and equate them to zero to obtain

$$\hat{\boldsymbol{\beta}}(\boldsymbol{\rho}, \boldsymbol{\varphi}) = \Big(\mathbf{M}(\boldsymbol{\varphi})^\top \mathbf{M}(\boldsymbol{\varphi})\Big)^{-1} \mathbf{M}(\boldsymbol{\varphi})^\top \mathbf{S}(\boldsymbol{\rho})\mathbf{Y}; \tag{2.19}$$

$$\hat{\sigma}^2(\boldsymbol{\rho}, \boldsymbol{\varphi}) = \frac{1}{n}\Big(\mathbf{S}(\boldsymbol{\rho})\mathbf{Y} - \mathbf{M}(\boldsymbol{\varphi})\hat{\boldsymbol{\beta}}(\boldsymbol{\rho}, \boldsymbol{\varphi})\Big)^\top \Big(\mathbf{S}(\boldsymbol{\rho})\mathbf{Y} - \mathbf{M}(\boldsymbol{\varphi})\hat{\boldsymbol{\beta}}(\boldsymbol{\rho}, \boldsymbol{\varphi})\Big) > 0. \tag{2.20}$$

Note that these are the solutions to an ordinary least squares model that regresses the transformed outcome variable $\mathbf{S}(\boldsymbol{\rho})\mathbf{Y}$ on the covariate matrix $\mathbf{M}(\boldsymbol{\varphi})$ when $\boldsymbol{\rho}$ and $\boldsymbol{\varphi}$ are known. Plugging this into the log-likelihood (2.18), we can obtain the profile log-likelihood

$$\ell_p(\boldsymbol{\rho}, \boldsymbol{\varphi}) = -\frac{n}{2}\left[\log(2\pi) + 1\right] + \log|\mathbf{S}(\boldsymbol{\rho})| - \frac{n}{2}\log\hat{\sigma}^2(\boldsymbol{\rho}, \boldsymbol{\varphi}). \tag{2.21}$$

Then, we can find $\hat{\boldsymbol{\varphi}}$ and $\hat{\boldsymbol{\rho}}$ by maximizing $\ell p$ using numerical estimation techniques. The maximum likelihood estimates $\hat{\boldsymbol{\beta}}$ and $\hat{\sigma}^2$ can be obtained by plugging $\hat{\boldsymbol{\varphi}}$ and $\hat{\boldsymbol{\rho}}$ into Equations (2.19) and (2.20).

Note that in (2.19), it is implicitly required that $\mathbf{M}(\boldsymbol{\varphi})^\top\mathbf{M}(\boldsymbol{\varphi})$ is invertible, i.e., the columns of the model matrix $\mathbf{M}(\boldsymbol{\varphi})$ need to be linearly independent. Although unlikely, multicollinearity may exist. For instance, in the POW-DEG specification (2.9), when the graph is fully connected (i.e. when every node is connected with one another), or when the treatment and/or control degrees are the same for all nodes, the model matrix will have linearly dependent columns. It is thus important in the design stage to choose a design that ensures the model matrix has full rank.

### 2.3.2 Asymptotic Results

Here, we study the behavior of the (quasi)-maximum likelihood estimators as the network size increases to infinity. We use the subscript $n$ to denote the data for a given network size $n$. Model (2.17) then becomes

$$\mathbf{Y}_n = \mathbf{S}_n(\boldsymbol{\rho})^{-1}\Big(\mathbf{M}_n(\boldsymbol{\varphi})\boldsymbol{\beta} + \boldsymbol{\epsilon}_n\Big),$$

where $\mathbf{S}_n(\boldsymbol{\rho}) = \mathbf{I}_n - \rho_T\mathbf{W}_{Tn} - \rho_C\mathbf{W}_{Cn}$. Let $\boldsymbol{\theta}_0 = (\boldsymbol{\rho}_0^\top, \boldsymbol{\beta}_0^\top, \boldsymbol{\varphi}_0^\top, \sigma_0^2)^\top$ be the true parameter values. The consistency and asymptotic normality properties of the (quasi)-maximum likelihood estimators $\hat{\boldsymbol{\theta}}_n$ are given in Theorem 2.2 below.

**Theorem 2.2.** *Under Assumptions 1-6 (given in Appendix A.1.2), the (quasi)-maximum likelihood estimator $\hat{\boldsymbol{\theta}}_n$ obtained by maximizing the log-likelihood in (2.18) is consistent to $\boldsymbol{\theta}_0$. Further assuming that $\mathbf{J}_n(\boldsymbol{\theta}_0) = -\mathbb{E}\left[\frac{\partial \log L_n(\boldsymbol{\theta}_0)}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^\top}\right]$ is invertible and $\mathbf{V}_n(\boldsymbol{\theta}_0) = \mathbb{E}\left[\left(\frac{\partial \log L_n(\boldsymbol{\theta}_0)}{\partial\boldsymbol{\theta}}\right)\left(\frac{\partial \log L_n\boldsymbol{\theta}_0)}{\partial\boldsymbol{\theta}}\right)^\top\right]$ is positive definite, then*

$$[\mathbf{V}_n(\boldsymbol{\theta}_0)]^{-1/2}[\mathbf{J}_n(\boldsymbol{\theta}_0)](\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_0) \xrightarrow{d} \mathcal{N}(\mathbf{0}_{\dim(\boldsymbol{\theta})}, \mathbf{I}_{\dim(\boldsymbol{\theta})}),$$

*where* $\dim(\cdot)$ *denotes the length of a vector.*

The proof of Theorem 2.2 is given in Appendix A.1.3, following the ideas of Lee (2004), treating $\mathbf{S}_n(\boldsymbol{\rho})$ and $\mathbf{M}_n(\boldsymbol{\varphi})$ as non-stochastic for any given $\boldsymbol{\rho}$ and $\boldsymbol{\varphi}$. The random errors $\epsilon_{i,n}$ are assumed to be independent and identically distributed with mean zero and variance $\sigma_0^2$. When $\epsilon_{i,n}$ follow a normal distribution, $\hat{\boldsymbol{\theta}}_n$ is the maximum likelihood estimator (instead of a quasi maximum likelihood estimator), and we have $\mathbf{V}_n(\boldsymbol{\theta}) = \mathbf{J}_n(\boldsymbol{\theta})$ and

$$[\mathbf{J}_n(\boldsymbol{\theta}_0)]^{1/2}(\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_0) \xrightarrow{d} \mathcal{N}(\mathbf{0}_{\dim(\boldsymbol{\theta})}, \mathbf{I}_{\dim(\boldsymbol{\theta})}).$$

### 2.3.3 Inference for Causal Quantities

With the asymptotic normality result, inference for the parameters can be performed accordingly. The inference for the causal quantities given in Section 2.2.3 can then be carried out via the Delta method (Doob, 1935). In particular, the global treatment effect for Model (2.15) is calculated as

$$\begin{aligned}\mathrm{GTE}(\boldsymbol{\theta}) = \frac{1}{n}\mathbf{1}_n^\top \Bigg[ &\left(\mu + \tau + \frac{1}{n}\sum_{i=1}^n g_{T,i}(\mathbf{D}_{\mathbf{Z}=\mathbf{1}_n}, \boldsymbol{\varphi})\right)(\mathbf{I}_n - \rho_T \mathbf{W}_{T,\mathbf{Z}=\mathbf{1}_n})^{-1} \\ &- \left(\mu + \frac{1}{n}\sum_{i=1}^n g_{C,i}(\mathbf{D}_{\mathbf{Z}=\mathbf{0}_n}, \boldsymbol{\varphi})\right)(\mathbf{I}_n - \rho_C \mathbf{W}_{C,\mathbf{Z}=\mathbf{0}_n})^{-1}\Bigg]\mathbf{1}_n. \end{aligned} \tag{2.22}$$

Using the Delta method, the variance of the GTE can be written as

$$\mathrm{Var}\big[\mathrm{GTE}(\boldsymbol{\theta})\big] = \mathbf{t}^\top \mathrm{Var}(\boldsymbol{\theta})\mathbf{t}, \tag{2.23}$$

where $\mathbf{t} = \frac{\partial \mathrm{GTE}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^\top}$. As $\mathrm{DTE}(\boldsymbol{\theta}) = \tau$ and $\mathrm{ITE}(\boldsymbol{\theta}) = \mathrm{GTE}(\boldsymbol{\theta}) - \tau$, the variance for DTE and ITE can be derived in a similar manner. Variance estimators can be obtained by plugging in the estimate $\hat{\boldsymbol{\theta}}$ obtained from the (quasi)-maximum likelihood. Confidence intervals and hypothesis testing can thus be conducted accordingly.

## 2.4 Simulations

In this section, we use simulations to study the properties of different specifications of the GANE model. Specifically, we study our proposed POW-DEG specification (2.9) and the HOM specification (2.4) as an illustration of a spatially autoregressive specification. In order to study these model specifications on real-life networks, we

use the Caltech Facebook network and the UMichigan Facebook network described in Section 1.2.3.2. The summary statistics of these networks are given in Table 1.1. In our simulations, these networks provide realistic structures for the graph $\mathcal{G}$, however, the experiment and outcomes are hypothetical and simulated. Moreover, as our theoretical results concern the case where the treatment assignment vector $\mathbf{Z}$ is known, in this simulation, we choose a particular design where half of the nodes are randomly assigned to treatment and the other half are assigned to control. The summary statistics of the selected design for each network are given in Appendix A.2.1.

### 2.4.1 The Distribution of the Estimates

In this subsection, we investigate the asymptotic properties of the maximum likelihood estimates derived in Section 2.3. First, we investigate the results for the POW-DEG specification (2.9) by generating outcomes on the given network (either the Caltech or UMichigan Facebook network) with the following parameter settings: $\boldsymbol{\beta} = (0, 1, 0.5, 0.1)^\top$ and $\sigma = 1$. We further vary the power $\lambda$ within the set $\{0.5, 0.75, 1, 1.25\}$, where $\lambda = 1$ corresponds to the LNE specification (2.1). With each combination of parameters, 1,000 runs are conducted where the outcomes are generated and the maximum likelihood estimates are calculated accordingly.

The distribution of the parameter and GTE estimates for the POW-DEG specification (2.9) are plotted in Figure 2.2. We can see that the distributions of all estimates are reasonably bell-shaped and symmetric and centered around the true values (dashed vertical lines) as is expected given the asymptotic theory. While the distribution of $\hat{\tau}$ remains the same under different values of $\lambda$, the variances of the other estimators decrease when $\lambda$ increases. This is because the ranges of values within $\mathbf{G}_T$ and $\mathbf{G}_C$ in the model matrix $\mathbf{M}$ increase as $\lambda$ increases, which in turn decreases the variance of the parameter estimates.

The coverage of 95% asymptotic confidence intervals and variances of the parameter estimates are given in Figure 2.3, where left axes correspond to variances and right axes correspond to coverage. The blue lines depict the asymptotic variances derived from $J(\boldsymbol{\theta}_0)$ and the red lines depict the sample variance of the 1,000 parameter estimates. Generally, these red and blue lines agree closely, except for the small gaps in the variances of $\hat{\tau}$ and thus $\widehat{\text{GTE}}$. However, these gaps are close in the results for the UMichigan network in Figure A.2. This suggests that a larger sample size is required for the variances of the $\tau$ and GTE estimators to be accurately estimated by the asymptotic theory. Concerning coverage, the coverage rates

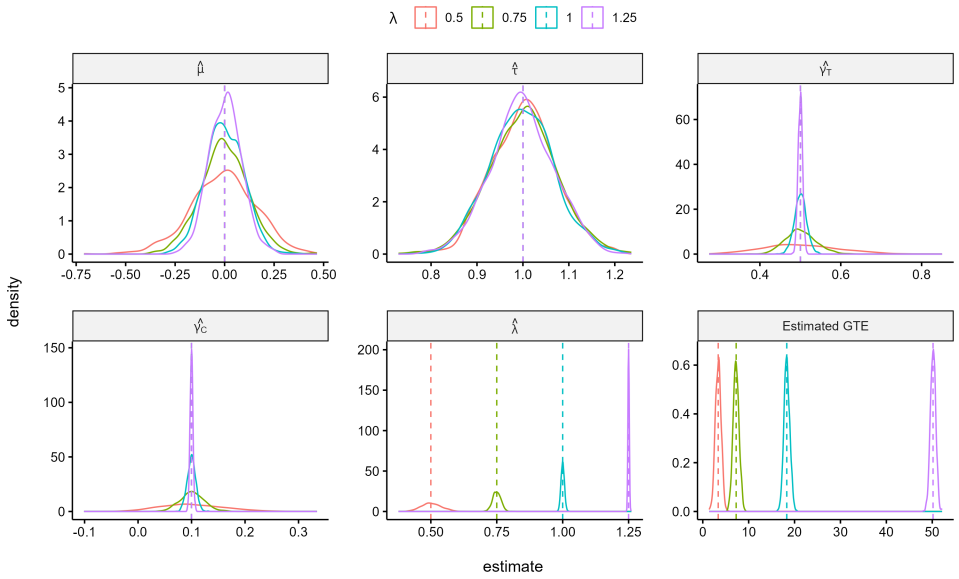



Figure 2.2: The distribution of parameter estimates of the POW-DEG specification on the Caltech Facebook network with $\boldsymbol{\beta} = (0, 1, 0.5, 0.1)^\top$ and $\lambda \in \{0.5, 0.75, 1.00, 1.25\}$ over 1,000 simulation runs.

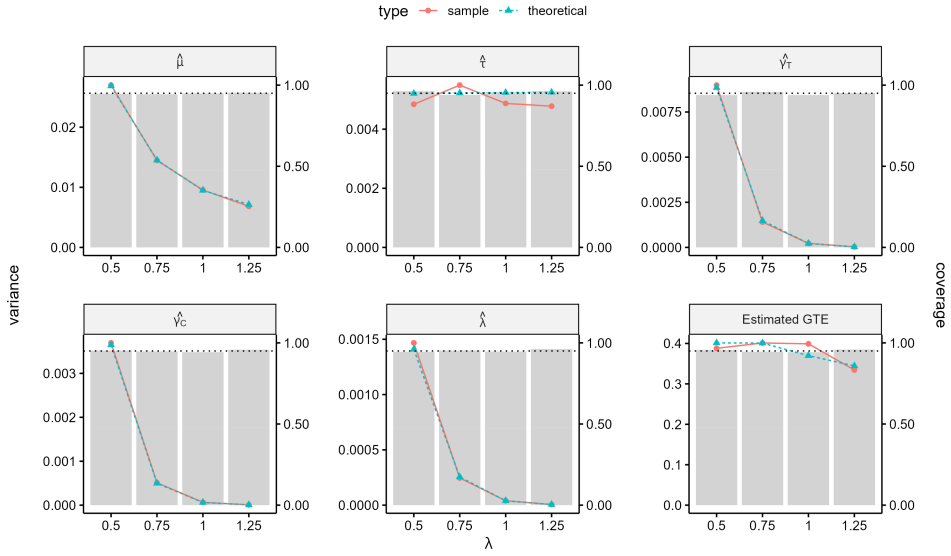



Figure 2.3: The variances of the estimates (left axes, lines) and coverage rates (right axes, bars) of POW-DEG specification on the Caltech Facebook network with $\boldsymbol{\beta} = (0, 1, 0.5, 0.1)^\top$ and $\lambda \in \{0.5, 0.75, 1.00, 1.25\}$ over 1,000 simulation runs.

for the 95% confidence intervals are plotted as grey bars on the right axes and the dotted lines serve as a reference at 0.95. We can see that the obtained confidence intervals have the correct coverage. To summarize, the simulation corroborates the asymptotic theory and indicates that maximum likelihood procedures work as expected for the POW-DEG specification (2.9). Simulation results for a different set of parameter values are included in Appendix A.2.2. These results suggest that when the network effect is small and the network size is moderate, consistent estimation of $\gamma_T$, $\gamma_C$, and $\lambda$ is more difficult. However, the battery of simulations was also run on the UMichigan Facebook network, whose size ($n = 3,749$) is almost 5 times that of the Caltech network, and we find that estimation of all parameters, whether the network and treatment effects are large or small, agrees with the asymptotic theory. These results are also available in Appendix A.2.2.

We conducted a similar simulation study on the HOM specification (2.4) with $\mu = 0$, $\tau = 1$, $\gamma_T = 0.5$, $\rho_T = \rho_C = 0.1$ and $\sigma^2 = 1$. The results for both the Caltech Facebook network and the UMichigan Facebook network are shown in Figure 2.4. As with the POW-DEG (2.9) estimates, and in agreement with the likelihood theory, the distributions of these parameter estimates are bell-shaped and centered at the true values. Moreover, since the UMichigan Facebook network is larger, the variation in the estimates decreases, as expected. Notice that the true values of GTE are different for the two networks, even though all parameters used are the same. This illustrates how the true value of GTE depends not only on the parameters but also on the structure of the graph. Variances and confidence interval coverage are also plotted in Figure 2.4. As we would expect, the asymptotic variances are suitable for inference and the asymptotic confidence intervals have acceptable coverage. To demonstrate the generality of these findings we present additional simulation results for another set of parameter values in Appendix A.2.2. Overall, the theory developed in Section 2.3 and the simulations presented here (for multiple GANE specifications, parameter values, and networks) demonstrate the general utility of maximum likelihood inference with GANE models.

### 2.4.2 Hypothesis Testing

As discussed in Section 2.2.3, under the GANE framework, we can test hypotheses about the DTE, SUTVA, ITE, and GTE. In particular, testing DTE $= 0$ is equivalent to testing $H_{01} : \tau = 0$; testing whether SUTVA is satisfied is equivalent to testing $H_{02} : f_T = f_C = 0$; the null hypothesis for testing the indirect treatment effect is $H_{03} :$ ITE $= 0$; and the null hypothesis for testing the global treatment effect is

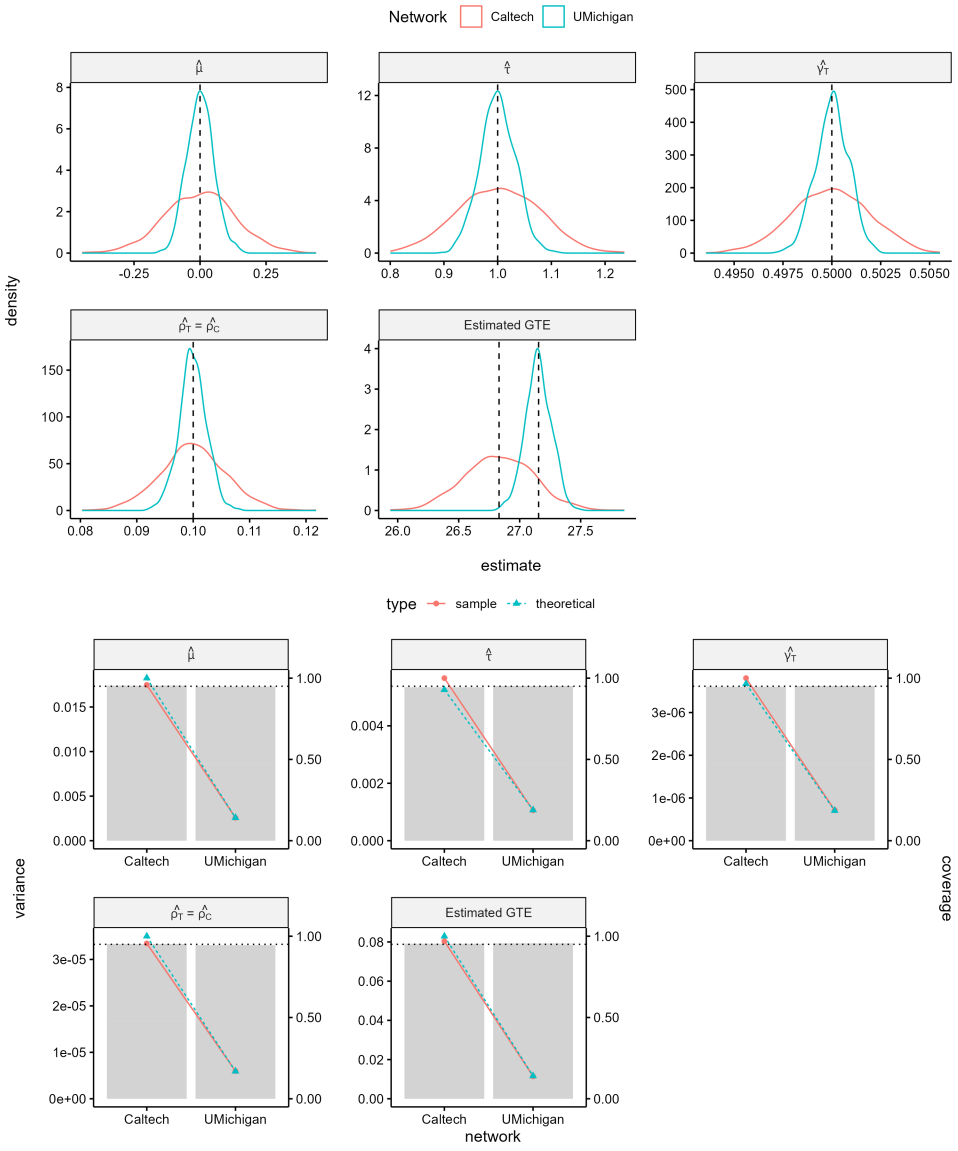



Figure 2.4: (upper) The distribution of parameter estimates of the HOM specification with $\mu = 0$, $\tau = 1$, $\gamma_T = 0.5$, $\rho_T = \rho_C = 0.1$ over 1,000 simulation runs. (lower) The corresponding variances of the estimates (left axes, lines) and coverage rates (right axes, bars).

$H_{04}$ : GTE $= 0$. In the maximum likelihood framework, Hypotheses 1, 3, and 4 can be tested using Wald-type tests, and Hypothesis 2 can be tested with a likelihood ratio test.

We study the characteristics of these tests via simulation. The parameters of the POW-DEG specification (2.9) are set at $\boldsymbol{\beta} = (0, 1, 0.5, 0.1)^\top$, $\sigma = 1$ and $\lambda \in \{0.5, 0.75, 1.00, 1.25\}$. Separate simulations are conducted to investigate each of the four hypothesis tests. For each simulation, values of certain parameters in $\boldsymbol{\beta}$ vary while the others stay as stated. In particular, in the simulation for Hypothesis 1, $\tau$ varies in the range $[0, 1]$; in the simulation for Hypothesis 2, $\gamma_T = \gamma_C$ and their values vary in the range $[0, 0.05]$; for Hypothesis 3, $\gamma_C$ is fixed at 0.1 and $\gamma_T - \gamma_C$ varies in the range $[0, 0.5]$. Hypothesis 4 with $H_{04}$ : GTE $= 0$ is also tested within each of the three simulations (with different $\lambda$) and the results are aggregated over different true values of GTE corresponding to different parameter combinations. All tests are done at a 5% significance level and 1,000 runs are conducted for each parameter combination. The results are presented in Figure 2.5. The dotted horizontal line serves as a reference at the 5% level.

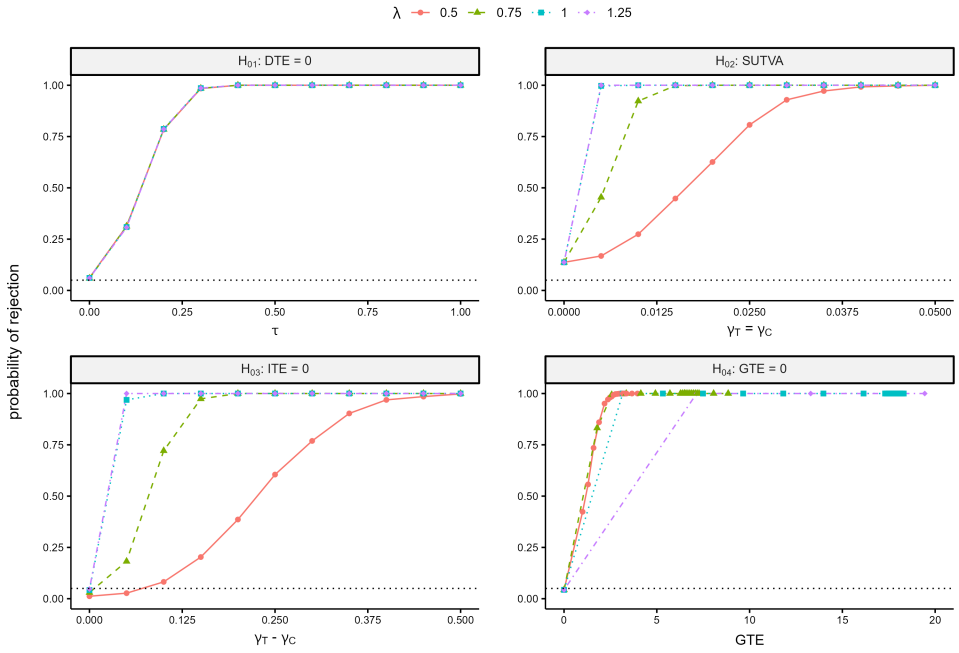



Figure 2.5: Rejection rates of hypothesis tests for POW-DEG specification on the Caltech Facebook network with varying parameters.

As expected, the rejection rates for each test increase as the respective parameter values depart from their null values. Moreover, tests for $H_{01} : \tau = 0$ seem to behave similarly over different values of $\lambda$. This is consistent with the model estimation results in Figure 2.2 where the variances for $\hat{\tau}$ and $\widehat{\text{GTE}}$ look similar over different values of $\lambda$ while the variances for $\hat{\gamma}$'s decrease as $\lambda$ increases. We also remark that the results at null values deviate slightly from the nominal 5% level. This can be attributed to the use of asymptotic (inexact) variances in these tests. Although we do not include the results for the UMichigan Facebook network, given its size and given the results from Section 2.4.1, we expect similar results to those presented here for the Caltech Facebook network.

We conducted similar simulations for the HOM specification (2.4) with $\mu = 0$, $\tau = 1$, $\gamma_T = 0.5$, $\rho_T = \rho_C = 0.1$, and varying $\tau$, $\gamma_T = \rho_C$ and $\gamma_T - \rho_C$ in different simulations for different hypothesis tests. The results are included in Appendix A.2.3. It can be noted that the results are similar in both networks, except for different values of $\tau$, signifying that this is an important parameter for the HOM specification (2.4). We also note that the rejection rates for Hypothesis 3 always stay at 100% even at $\gamma_T = \rho_C$ (i.e., when the scaling coefficients for $f_T$ and $f_C$ are equal). This shows that the indirect effect is not only affected by the sizes of the network effects but also by the functional forms of $f_T$ and $f_C$, which are different in the HOM specification (2.4) compared to the POW-DEG specification (2.9).

### 2.4.3 Model Misspecification

The simulations in Sections 2.4.1 and 2.4.2 explore properties of maximum likelihood inference for different GANE specifications when they are *correctly specified*. In this section, we further investigate the properties of these specifications under model misspecification. The specifications considered here are (i) the SUTVA specification, in which network effects do not exist and $f_T = f_C = 0$; (ii) the linear network effect (LNE) specification in (2.1); (iii) the POW-DEG specification in (2.9); (iv) the local aggregate (LAG) specification in (2.7); and (v) the homophily (HOM) specification in (2.4).

In this simulation, on the Caltech Facebook network, outcomes are generated 1,000 times for each of the listed model specifications. The data are then fitted using each of the five model specifications. Because the global treatment effect GTE is generally of primary interest, we use the GTE estimation and its inference results to compare performance among specifications. To make the comparison fair, parameters for each model specification are chosen such that the true global treatment effect

(GTE) is fixed at 2.0 and the average outcome variance is 1.0 in all data-generating scenarios. The exact parameter values for each specification are provided in Table 2.1.

| Specification | $\mu$ | $\tau$ | $\rho_T$ | $\rho_C$ | $\gamma_T$ | $\gamma_C$ | $\lambda$ | $\sigma$ |
|---|---|---|---|---|---|---|---|---|
| SUTVA | 0 | 2 | 0 | 0 | 0 | 0 | | 1 |
| LNE | 0 | 1 | 0 | 0 | 0.1231 | 0.1 | | 1 |
| POW-DEG | 0 | 1 | 0 | 0 | 0.2691 | 0.1 | 0.5 | 1 |
| LAG | 0 | 1 | 0.008492 | 0.001 | 0 | 0 | | 0.9977 |
| HOM | 0 | 1 | 0.1 | 0.1 | 0.01728 | 0 | | 0.9999 |

Table 2.1: Parameters for the simulation in Section 2.4.3.

Results of the simulation are plotted as heatmaps in Figure 2.6. The columns correspond to outcome-generating models and the rows correspond to estimating models. The top left panel shows the log ratio of the average estimated GTE to the true GTE. The desired value is 0, which is colored white. Red represents overestimation and blue represents underestimation. We see that all specifications can estimate the SUTVA specification well because it is nested within all GTE specifications. The POW-DEG specification seems to provide estimates with the lowest bias, even under model misspecification.

The top right panel shows the standard deviations of the GTE estimates where white represents low standard deviations and dark green represents high standard deviations. We can see that the SUTVA and HOM (2.4) specifications provide the lowest standard deviations while the highest standard deviations come from the LAG specification (2.7). Both the POW-DEG (2.9) and the LNE (2.1) specifications provide reasonably low standard deviations.

The bottom left panel shows the coverage rate of 95% confidence intervals for the GTE constructed by each estimating model. The results show that LNE (2.1), POW-DEG (2.9) and LAG (2.7) specifications have high coverage rates while the HOM specification (2.4) has lower coverage rates and the SUTVA specification has the worst. This is because the SUTVA specification does not capture the network effects introduced by other specifications.

Finally, on the bottom right, the model selection results by AIC (Akaike, 1998) are presented. Green represents high selection rates while white represents low selection rates. AIC works well as it selects the correct model specification most of the time, which is shown by the green diagonal. This supports the use of likelihood-based model selection criteria such as AIC for the GANE framework. Furthermore, it can be seen that POW-DEG specification (2.9) is selected fairly often no matter

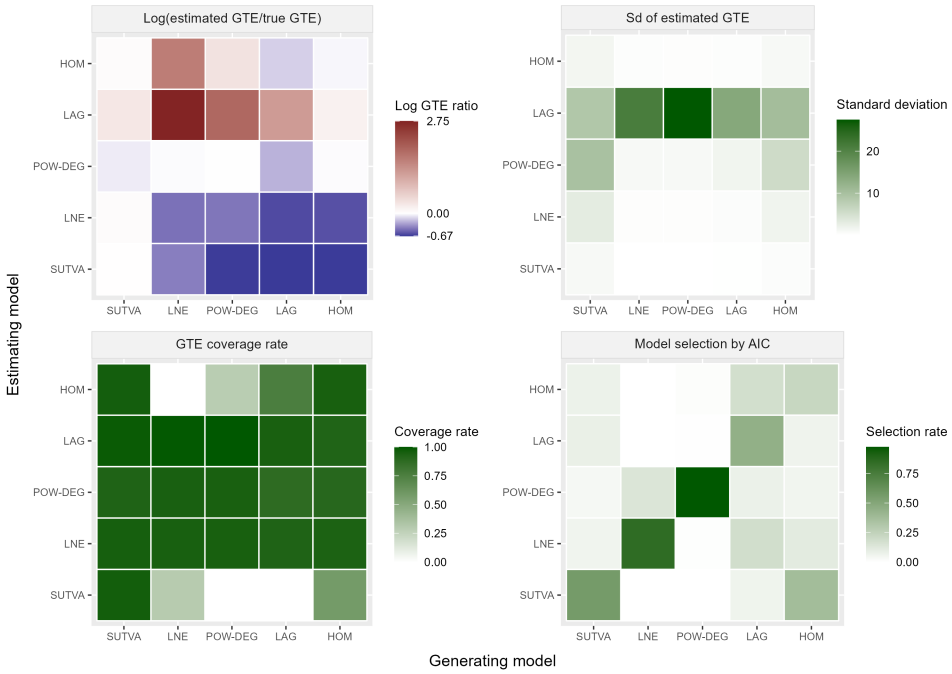



Figure 2.6: Model misspecification simulation results. The horizontal axis corresponds to the data-generating model while the vertical axis corresponds to the estimating model.

the data-generating model, which suggests that it fits the data reasonably well even under model misspecification.

As illustrated in Figure 2.6, the POW-DEG specification (2.9) is the only one that performs well in each dimension. This illustrates the flexibility of the POW-DEG model to capture a variety of network effects. Hence, we advocate its use generally, especially when there is no prior information or preference for another specification.

## 2.5 Conclusions

We introduce the general additive network effect model for network A/B tests, which unifies many existing models in the literature and enhances the modeling flexibility. We further bridge the model-based and the design-based frameworks by defining causal quantities of interest: the global treatment effect, the direct treatment effect, and the indirect treatment effect as functions of the model parameters. Thus, inference for all three quantities may be carried out via the inference of the model parameters.

Although the model is studied under the A/B testing setting where there are just two experimental conditions (treatment and control) and the outcomes are continuous, the GANE model framework can be extended for use in other settings. First, by expanding the model equation, the GANE model can be used to analyze experiments with more than two experimental conditions. Second, by introducing link functions and other distributional and functional assumptions, the framework can be extended to deal with non-normal distributions and discrete outcomes in manners similar to generalized linear models. In particular, the extension of the GANE model to experiments with binary outcomes is treated in Chapter 3.

Despite the GANE framework's flexibility, we recommend the POW-DEG specification (2.9), which models the network effect as powers of the treatment and control degrees. Via simulation, we found that the specification is robust against model misspecification in terms of inference for the global treatment effect. Thus we suggest the use of this specification, especially in the design stage, when there is no prior information or modeling preference. Optimal designs for the POW-DEG specification with respect to the MSE of the GTE are discussed in Section 4.

Finally, although AIC appears to work in the model misspecification simulation in Section 2.4.3, the use of AIC is only possible in the analysis stage once the data are observed, or when preliminary data are available. Model selection for the design

and analysis of experiments on networks thus remains an open problem for future research.

# Chapter 3

# Analysis of Network Experiments with Binary Outcomes

The GANE model proposed in Chapter 2 allows experiments with continuous outcomes to be modeled with generalized network effects. However, many experiments in practice concern binary outcomes, for example, experimenters may be interested in studying individuals' decisions to buy a certain product (Cai et al., 2015) or to adopt a policy (Park et al., 1976). Applying models designed for continuous outcomes to binary outcome data may be inadequate due to distributional misspecification. In this chapter, we extend the GANE framework in Chapter 2 for binary experimental outcomes. In particular, we assume that the experimental outcomes follow a Bernoulli distribution whose success probability is modeled using a non-autoregressive specification of the GANE model. We investigate the performance of such a model via extensive simulations. Our method is then applied to the agricultural insurance data set from Cai et al. (2015).

## 3.1 Binary GANE Extension

We consider the same problem setting as in Chapter 2, where we have access to $n$ experimental units. The experimental data $\mathbf{D}$ contains $\mathbf{A}, \mathbf{Z}, \mathbf{X}$, which respectively are the adjacency matrix, the vector of treatment assignments, and a possible matrix of covariates. We assume $\mathbf{D}$ to be fixed. The experiment being considered is again an A/B test with two treatments (treatment vs. control) on an undirected, simple, and fixed network. That is, for each unit $i$, $Z_i = 1$ indicates that the unit is assigned

to treatment, and $Z_i = 0$ indicates that the unit is assigned to control. For every pair of units $i$ and $j$, $A_{ij} = 1$ indicates that the two units are connected on the network, otherwise $A_{ij} = A_{ji} = 0$. Note that since the network is simple, $A_{ii} = 0$ for all $i = 1, ..., n$.

In this chapter, we consider the case where $Y_i$, $i = 1, ..., n$ are binary indicators taking the values 0 or 1. For each $i = 1, ..., n$, we consider the case where the experimental outcome $Y_i$ follows a Bernoulli distribution with success probability

$$\mathbb{P}(Y_i = 1) = h\Big(\mu + \tau Z_i + \gamma_T g_{T,i}(\boldsymbol{\varphi}) + \gamma_C g_{C,i}(\boldsymbol{\varphi})\Big), \tag{3.1}$$

where $h : \mathbb{R} \to [0, 1]$ is an inverse link function, which is often used in the GLM literature (McCullagh and Nelder, 1989, sec. 2.2.3), and $g_{T,i}$ and $g_{C,i}$ are real-valued functions that depend on parameters $\boldsymbol{\varphi}$ and the experimental data $\mathbf{D}$ but not the experimental outcome $\mathbf{Y}$. With Model (3.1), the outcome probability of a unit $i$ depends not only on its own treatment assignment $Z_i$ but also on other treated and controlled units via functions $g_{T,i}(\boldsymbol{\varphi})$ and $g_{C,i}(\boldsymbol{\varphi})$. Note that the regression within $h(\cdot)$ has a similar form as the GANE model in (2.15). The only difference is that Model (3.1) excludes the autoregressive component that involves the experimental outcome vector $\mathbf{Y}$. If we choose to include the outcome vector $\mathbf{Y}$ in the regression, we will not be able to specify the full likelihood for $\mathbf{Y}$. Indeed, including the outcome vector $\mathbf{Y}$ in the regression is considered infeasible in the spatial econometrics literature (Anselin, 2002; Klier and McMillen, 2008).

### 3.1.1 Estimation

Since Model (3.1) makes a Bernoulli distributional assumption for the outcome, we use maximum likelihood for the estimation and inference of Model (3.1). With this distributional assumption, asymptotic likelihood theory is well established and so theoretical developments like those in Section 2.3 are not necessary here. For brevity, we rewrite Model (3.1) as

$$p_i(\boldsymbol{\beta}, \boldsymbol{\varphi}) := \mathbb{P}(Y_i = 1) = h(\mathbf{m}_i(\boldsymbol{\varphi})^\top \boldsymbol{\beta}),$$

where $\mathbf{m}_i(\boldsymbol{\varphi}) = [\ 1 \quad Z_i \quad g_{T,i}(\boldsymbol{\varphi}) \quad g_{C,i}(\boldsymbol{\varphi})\ ]^\top$ and $\boldsymbol{\beta} = [\ \mu \quad \tau \quad \gamma_T \quad \gamma_C\ ]^\top$. Now, the log-likelihood can be written as

$$\ell(\boldsymbol{\beta}, \boldsymbol{\varphi}) = \sum_{i=1}^{n} \Big\{ Y_i \log p_i(\boldsymbol{\beta}, \boldsymbol{\varphi}) + (1 - Y_i) \log \big[1 - p_i(\boldsymbol{\beta}, \boldsymbol{\varphi})\big] \Big\}. \tag{3.2}$$

Note that Model (3.1) is a conventional generalized linear model for fixed values of $\boldsymbol{\varphi}$ since $g_{T,i}$ and $g_{C,i}$ are known up to $\boldsymbol{\varphi}$. Thus, we can estimate the parameters of Model (3.1) using the profile likelihood method. In particular, for a fixed value $\boldsymbol{\varphi}$, let $\hat{\boldsymbol{\beta}}(\boldsymbol{\varphi})$ be the ML estimate of $\boldsymbol{\beta}$ in the binary regression (3.1), which can be found using commands for generalized linear regression in standard statistical software. Then, the ML estimate for $\boldsymbol{\varphi}$ can be found by maximizing the profile log-likelihood $\ell_p(\boldsymbol{\varphi}) = \ell(\hat{\boldsymbol{\beta}}(\boldsymbol{\varphi}), \boldsymbol{\varphi})$ using numerical optimization techniques.

### 3.1.2 Inference

Let $\boldsymbol{\theta}_0 = [\ \boldsymbol{\beta}_0 \quad \boldsymbol{\varphi}_0\ ]^\top$ be the true parameter values of Model (3.1). Following the maximum likelihood theory (Serfling, 1980, sec. 4.2), the ML estimator is asymptotically unbiased and follows a normal distribution. The asymptotic variance of the ML estimates is given by the inverse of the Fisher information matrix, that is,

$$J^{-1}(\boldsymbol{\theta}_0) = \left( \mathbb{E}\left[ -\frac{\partial^2 \ell}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^\top} \Big|_{\boldsymbol{\theta}_0} \right] \right)^{-1}. \tag{3.3}$$

Thus, an estimator of the variance is the plug-in estimator $[J(\hat{\boldsymbol{\theta}})]^{-1}$. The subscript indicating the dependence of $J(\boldsymbol{\theta}_0)$ on $n$ is suppressed in this chapter for brevity. The detailed derivation of the variance estimator for Model (3.1) is given in Appendix B.1.1.

### 3.1.3 Global Treatment Effect

Recall that the global treatment effect (GTE) is the difference in the expected experimental outcomes, when all units in the network are assigned to treatment, versus when they are assigned to control. This quantity is still of interest in the context of binary responses. Hence, for Model (3.1), the GTE is given by

$$\mathrm{GTE}(\boldsymbol{\theta}) = \frac{1}{n} \sum_{i=1}^{n} \left\{ h\left[ \mathbf{m}_i(\mathbf{D}_{\mathbf{Z}=\mathbf{1}_n}, \boldsymbol{\varphi})^\top \boldsymbol{\beta} \right] - h\left[ \mathbf{m}_i(\mathbf{D}_{\mathbf{Z}=\mathbf{0}_n}, \boldsymbol{\varphi})^\top \boldsymbol{\beta} \right] \right\}, \tag{3.4}$$

where $\mathbf{m}_i(\mathbf{D}_{\mathbf{Z}=\mathbf{1}_n}, \boldsymbol{\varphi})$ is the value of $\mathbf{m}_i$ when $\mathbf{Z} = \mathbf{1}_n$ and $\mathbf{m}_i(\mathbf{D}_{\mathbf{Z}=\mathbf{0}_n}, \boldsymbol{\varphi})$ is defined similarly. Since the GTE in (3.4) is a function of the parameters, we can estimate the GTE from data using the plug-in estimator, namely $\mathrm{GTE}(\hat{\boldsymbol{\theta}})$. The variance of this estimator can be estimated using the Delta method by substituting $\hat{\boldsymbol{\theta}}$ for $\boldsymbol{\theta}$ in Equation (2.23). Asymptotic confidence intervals and hypothesis tests can be constructed accordingly.

## 3.2 Simulations

Next, we study the properties of the ML estimator for Model (3.1) via simulations. Note that in Model (3.1), if the network effect functions do not contain unknown parameters, i.e., $\boldsymbol{\varphi} = \emptyset$, Model (3.1) becomes a regular generalized linear regression. Since the behavior of ML estimators for generalized linear regressions are well-studied in the literature (McCullagh and Nelder, 1989), in this section, we focus on a nonlinear specification of Model (3.1), that is, the POW-DEG specification from Chapter 2 with

$$g_{T,i}(\boldsymbol{\varphi}) = \left(\sum_{j=1}^{n} A_{ij} Z_j\right)^{\lambda} \quad \text{and} \quad g_{C,i}(\boldsymbol{\varphi}) = \left(\sum_{j=1}^{n} A_{ij}(1 - Z_j)\right)^{\lambda}, \tag{3.5}$$

where $\boldsymbol{\varphi} = \{\lambda\}$. Similar to Chapter 2, to understand the estimator's properties in real-life network settings, we consider the Caltech and the UMichigan networks, whose summary statistics are given in Table 1.1. Since the ML theory for Model 3.1 applies for fixed values of $\mathbf{Z}$, in the simulation, we generate a fixed design $\mathbf{Z}$ for each network setting (either the Caltech or the UMichigan network), where half of the units are randomly selected and assigned to treatment and the rest of the units are assigned to control. Summary statistics of the designs we generated are provided in Appendix B.2.1. We use these designs throughout the simulation study.

### 3.2.1 The Distribution of the Estimates

First, we investigate the distribution of the ML estimator of Model (3.1) with the logit link

$$h(x) = \frac{1}{1 + \exp(-x)}.$$

Results for the probit regression version with $h(x) = \Phi(x)$, where $\Phi(\cdot)$ is the cumulative distribution function of the standard normal distribution, are given in Appendix B.2.2. For the parameters, we consider $\boldsymbol{\beta} = (-2, 1, 0.5, 0.1)^{\top}$ and $\lambda \in \{0.25, 0.5, 0.75\}$. For each combination of parameters and network, we conduct 1,000 simulation runs, in which we generate the experimental outcomes from Model (3.5) and estimate the parameters using the generated data. The parameter estimates and their estimated variances are recorded for investigation.

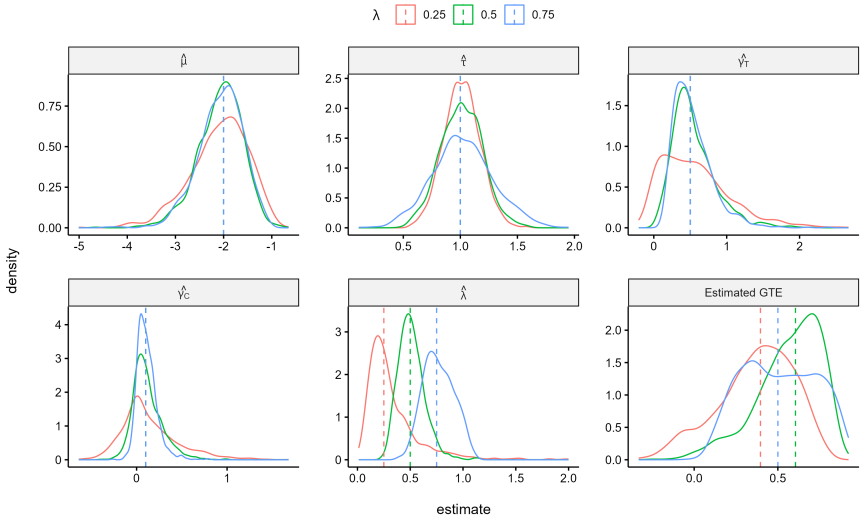
Figure 3.1a shows the distribution of the parameter and GTE estimates for the Caltech network. The dotted vertical lines represent true parameter values. We can

see that the distributions of the estimators are reasonably bell-shaped around the true parameter values as expected by asymptotic theory. Of all parameters, $\tau$ seems to be estimated the best as the sampling distribution of $\hat{\tau}$ is concentrated symmetrically at the true value. Skewed distributions are seen for the other parameters and the GTE. This happens because binary outcomes have less variation than continuous outcomes and are difficult to fit. Extreme estimates can happen when the binary responses are highly imbalanced. Hence, experiments with binary outcomes may require larger sample sizes for the asymptotic results to work. This conclusion is supported by the results for the larger UMichigan network in Figure 3.1b, where the sampling distributions for every parameter are more symmetric and concentrated around the true values.

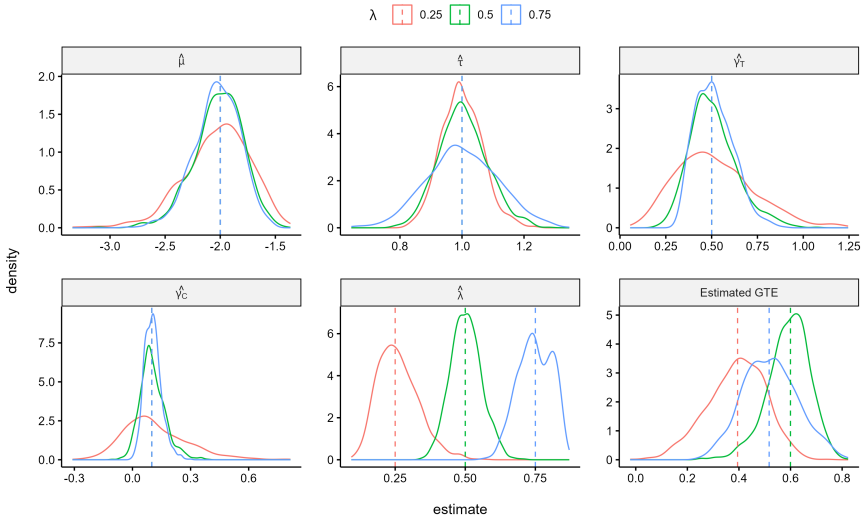
We further investigate the variance estimates of the ML estimators in Figure 3.2. The sample variance and theoretical variance (calculated using Equation (3.3)) are plotted with red and blue lines, respectively. The coverage rates of the associated 95% confidence intervals are plotted with grey bars, with dotted lines indicating the 0.95 level. We refer to the left axes for variances and to the right axes for coverage. We can see that the estimation results are clearly better for the larger UMichigan network, where sample and theoretical variances follow more closely and the coverage rates are more accurate. There are still some discrepancies in the sample and theoretical variances for $\lambda$, indicating that the parameter requires a larger sample size for the large sample inference to be accurate. Nevertheless, this again emphasizes that large sample sizes are required for ML estimation in experiments with binary outcomes. Similar conclusions are found in the results for the probit regression, which are given in Appendix B.2.2.

### 3.2.2 Hypothesis Testing

Due to the link function, the definitions of the direct and indirect effects in Section 2.2.3 are not relevant for Model (3.1). That is, $\tau$ is no longer the difference in expected outcomes when a node is assigned to treatment versus control while keeping the network effects fixed. Nevertheless, in the structure of Model (3.1), it is clear that $\tau$ governs the effect from individual treatment assignment, $\gamma_T$ and $g_{T,i}$ governs the effect from treated units, and $\gamma_C$ and $g_{C,i}$ are responsible for the effect from controlled units. Thus, the experimenters may still be interested in testing similar hypotheses considered in Section 2.4.2, i.e., $H_{01} : \tau = 0$, $H_{02} : \gamma_T = \gamma_C = 0$, $H_{03} : \gamma_T - \gamma_C = 0$, and $H_{04} : \text{GTE} = 0$. Hypotheses $H_{01}, H_{03}$ and $H_{04}$ are tested using a Wald-type test and Hypothesis $H_{02}$ is tested using a likelihood ratio test.

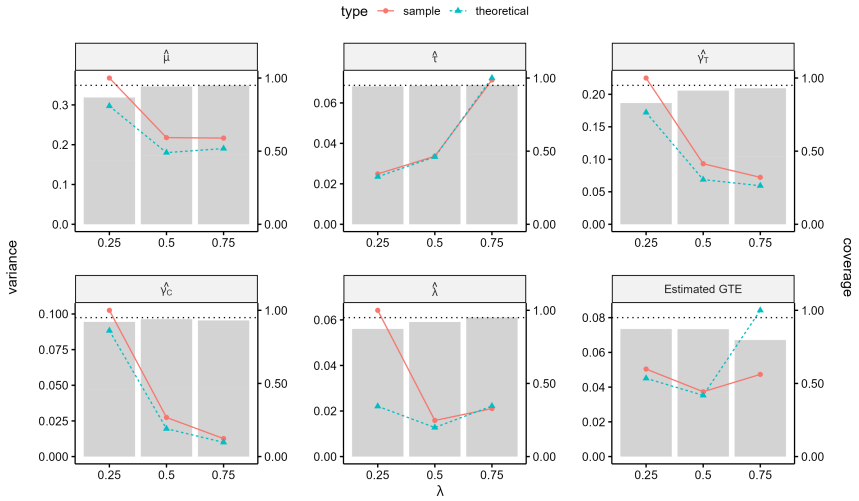



(a) Results for the Caltech network.

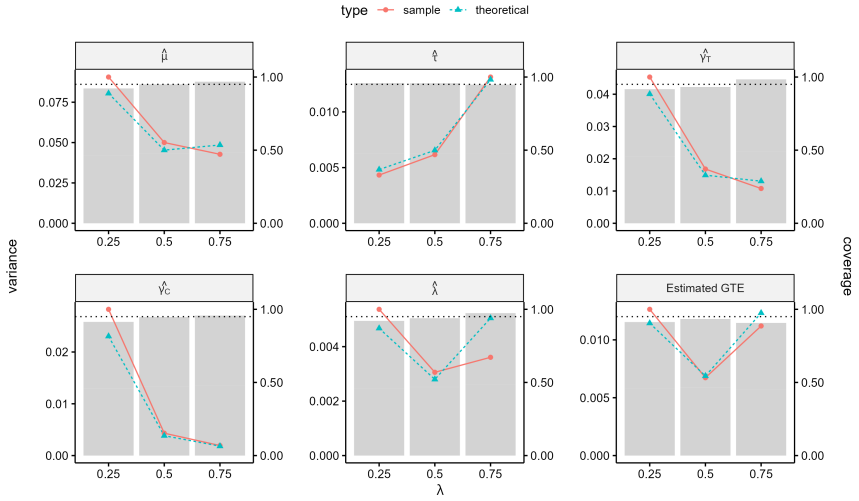



(b) Results for the UMichigan network.

Figure 3.1: The distribution of parameter estimates for the POW-DEG specification of Model (3.1) with logit link over 1,000 runs, where $\boldsymbol{\beta} = (-2, 1, 0.5, 0.1)^\top$ and $\lambda \in \{0.25, 0.5, 0.75\}$.




(a) Results for the Caltech network.




(b) Results for the UMichigan network.

Figure 3.2: The variances of the estimates (left axes, lines) and coverage rates (right axes, bars) for the POW-DEG specification of Model (3.1) with logit link over 1,000 simulation runs $\boldsymbol{\beta} = (-2, 1, 0.5, 0.1)^\top$ and $\lambda \in \{0.25, 0.5, 0.75\}$.

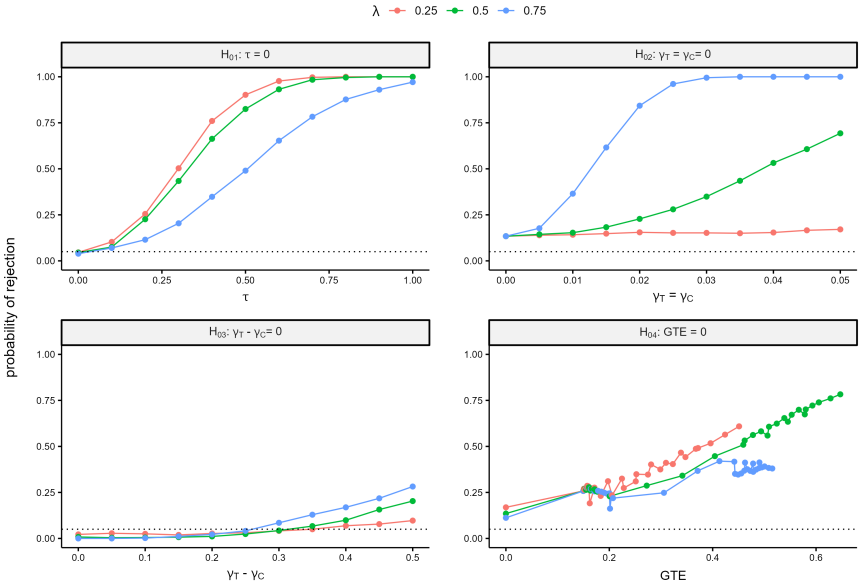
We conduct simulations to study the properties of these tests. We use the same set of parameters: $\boldsymbol{\beta} = (-2, 1, 0.5, 0.1)^\top$ and $\lambda \in \{0.25, 0.5, 0.75\}$, and run separate simulations to study each of the hypothesis tests. In each simulation, we vary some parameters while keeping the others fixed. In particular, for $H_{01}$, $\tau$ varies in $[0, 1]$; for $H_{02}$, $\gamma_T = \gamma_C$ and their values vary in the range $[0, 0.05]$; for $H_{03}$, $\gamma_C$ is fixed at 0.1 while $\gamma_T - \gamma_C$ vary in $[0, 0.5]$. Results for testing $H_{04}$ are accumulated through the simulations of the other three hypotheses, in which different parameter combinations result in different values of the GTE. Each scenario corresponding to a specific combination of parameters and network is investigated via 1,000 runs. Results for the logit link are shown in Figure 3.3 while results for the probit link are shown in Appendix B.2.3. Note that because the results for $H_{04}$ are accumulated throughout all parameter combinations in the simulations for the other three hypotheses, the GTE values in the $H_{04}$ plot are not regularly spaced.

Overall, the rejection rates increase as the parameters move away from the null values. Moreover, the tests become more accurate with higher power as the network size increases, namely results are better for the UMichigan network compared to the smaller Caltech network. However, we detect an anomaly in the test for $H_{03}$ when $\gamma_T - \gamma_C > 0.4$ in the UMichigan network. Specifically, the rejection rates drop rapidly when $\gamma_T$ increases beyond 0.5. This happens because larger parameter values have led to simulated data with many more 1s than 0s, making the estimation difficult and negatively impacting the performance of the hypothesis test. We can also see some irregularities at GTE $\approx 0.17$ in the test for $H_{04}$. This is caused by certain combinations of parameters in which $\tau$ is large, causing the generated data to be more imbalanced. Nevertheless, the general trend for $H_{04}$ rejection is still increasing with GTE as expected.

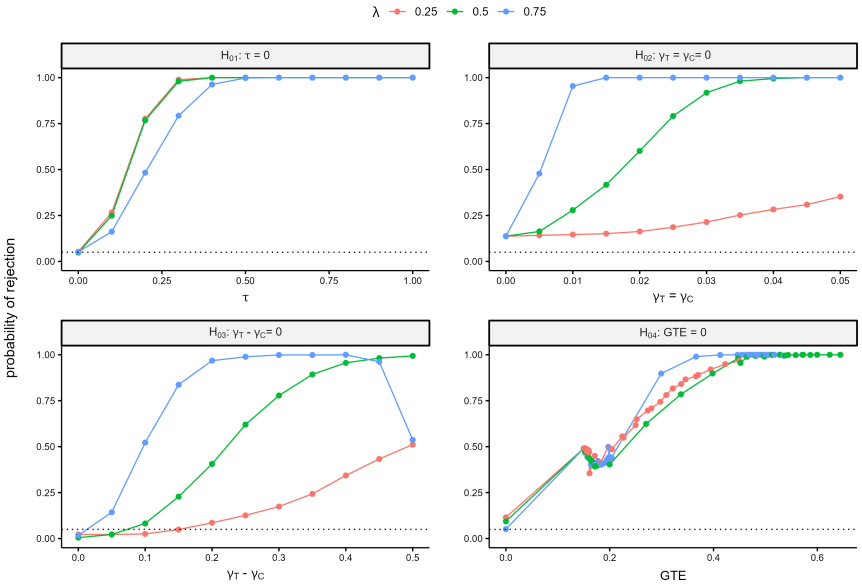
Appendix B.2.3 shows results for the probit link, which behave more closely to what is expected. The difference can be attributed to the difference in model structure and thus the difference in the scale of feasible parameter values for each model. Nevertheless, the simulation results suggest that experimenters should carefully design their binary-outcome experiments so that the numbers of 0s and 1s in the outcomes are ideally both reasonably large.

### 3.2.3 GTE Estimation Under Model Misspecification

In the previous two sections, we studied the properties of the ML estimators of the POW-DEG specification of Model (3.1) when the model is correctly specified. Moreover, in Chapter 2, we found that the POW-DEG specification for the continuous




(a) Results for the Caltech network.




(b) Results for the UMichigan network.

Figure 3.3: Rejection rates of hypothesis tests for POW-DEG specification of Model (3.1) with logit link and varying parameters.

GANE model had good performance under model misspecification. Does the POW-DEG specification still work well under model misspecification for binary data? In this section, we address this question by considering cases where the underlying data-generating model is not the same as the estimating model. In particular, we consider the following data-generating model

$$
\begin{aligned}
Y^*_{i,t} &= \mu + \tau Z_i + \gamma \frac{1}{K_{ii}} \sum_{j=1}^{n} A_{ij} Y_{j,t-1} + \epsilon_{i,t} \\
Y_{i,t} &= \mathbb{I}(Y^*_{i,t} > 0),
\end{aligned} \tag{3.6}
$$

where $\epsilon_{i,t} \sim \mathcal{N}(0, \sigma^2)$ and $0 \leq t \leq T$. In this model, the binary experimental outcome $Y_{i,t}$ of unit $i$ at time $t$ is determined by the value of a latent variable $Y^*_{i,t}$. The latent variable is influenced by unit $i$'s own treatment assignment $Z_i$, and by the outcomes of its neighbors in the previous time step. We can see that with this model, the effect from an individual's own treatment assignment is governed by the parameter $\tau$, while the effect from the network (here the neighbors) is governed by the parameter $\gamma$. Thus, the GTE will also be affected by these two parameters. In the literature, Model (3.6) has been used to evaluate the performance of different design-based GTE estimators (Gui et al., 2015; Eckles et al., 2016; Chin, 2019). We will use this model to evaluate the performance of different specifications of Model (3.1), while comparing them with the simple difference-in-means estimator. Specifically, we consider both linear and nonlinear specifications of the continuous GANE (2.3) and the binary GANE (3.6) models. The list of models being compared is given below.

(1) SUTVA model: This estimator corresponds to the SUTVA model in Section 2.4.3, namely a simple linear regression with the intercept and the treatment assignment vector as covariates. The GTE estimator corresponding to this model is equivalent to the difference-in-means estimator

$$
\hat{\tau} = \frac{\sum_{i=1}^{n} Z_i Y_i}{\sum_{i=1}^{n} Z_i} - \frac{\sum_{i=1}^{n} (1 - Z_i) Y_i}{\sum_{i=1}^{n} (1 - Z_i)}. \tag{3.7}
$$

(2) Gaussian-GLM: the LNE model in (2.1).

(3) Probit-GLM: the LNE specification of Model (3.1) with probit link.

(4) Logit-GLM: the LNE specification of Model (3.1) with logit link.

(5) Gaussian-POWDEG: the POW-DEG model in (2.9).

(6) Probit-POWDEG: the POW-DEG specification of Model (3.1) with probit link.

(7) Logit-POWDEG: the POW-DEG specification of Model (3.1) with logit link.

Following Eckles et al. (2016), in Model (3.6), we set the initial outcomes $Y_{i,0}$ to 0 for $i = 1, ..., n$. We also set $\mu = -1.5$, $\sigma = 1$, $T = 3$, while varying $\tau \in \{0, 0.25, 0.5, 0.75, 1\}$ and $\gamma \in \{0, 0.25, 0.5, 0.75, 1\}$. Each combination of network and parameters is used in 1,000 simulation runs, in which the outcomes are generated using Model (3.6) and the GTE is estimated using estimators based on Models (1)-(7). Note that in this simulation setting, the model assumptions of Models (1)-(7) are all wrong. This enables us to evaluate their performances under model misspecification. In particular, we evaluate the performance of these estimators using bias, standard deviation, root mean squared error (RMSE), coverage rate, and the proportion of times the estimator is selected among all estimators by the AIC. Results for the Caltech network are given in Figure 3.4, and the results for the UMichigan network are given in Figure 3.5.

In the figures, different colors represent different estimating models. Full lines represent linear specifications (when $\boldsymbol{\varphi} = \emptyset$) while dotted lines represent the nonlinear POW-DEG specifications. The columns of the panel grid correspond to different values of $\gamma$ in Model (3.6) while the rows correspond to different evaluation criteria. First, we notice that the SUTVA estimator (in red) performs very well with low bias, standard deviation, and RMSE when $\tau$ and $\gamma$ are small. In this case, it is best to use this simple estimator to estimate the GTE. However, as $\tau$ and $\gamma$ increase, the bias of the SUTVA estimator grows increasingly worse than the GTE estimators based on binary models (in blue and green). We also notice that the standard deviation of the SUTVA estimator is consistently low. This is because the SUTVA estimator's formula in (3.7) only takes into account the number of times $Y = 1$ in each treatment group, rather than which unit has which outcome. On the other hand, the other estimators take into account the network structure and hence are more sensitive to the change in the response vector. At higher values of $\tau$ and $\gamma$, the SUTVA estimator has increasingly large bias and poor coverage, which prompts the use of binary models as in (3.1) in these scenarios. This conclusion is further supported by the results for the UMichigan network in Figure 3.5. With a larger network, the standard deviations of the estimators from the binary models decrease, making the overall RMSE of these estimators lower than that of the SUTVA estimator.

Second, from Figure 3.4, although having correct coverage for the GTE, the RMSE of estimators from models for continuous outcomes (in yellow) increase as $\tau$ and $\gamma$ increase. On the other hand, the RMSE of estimators coming from binary

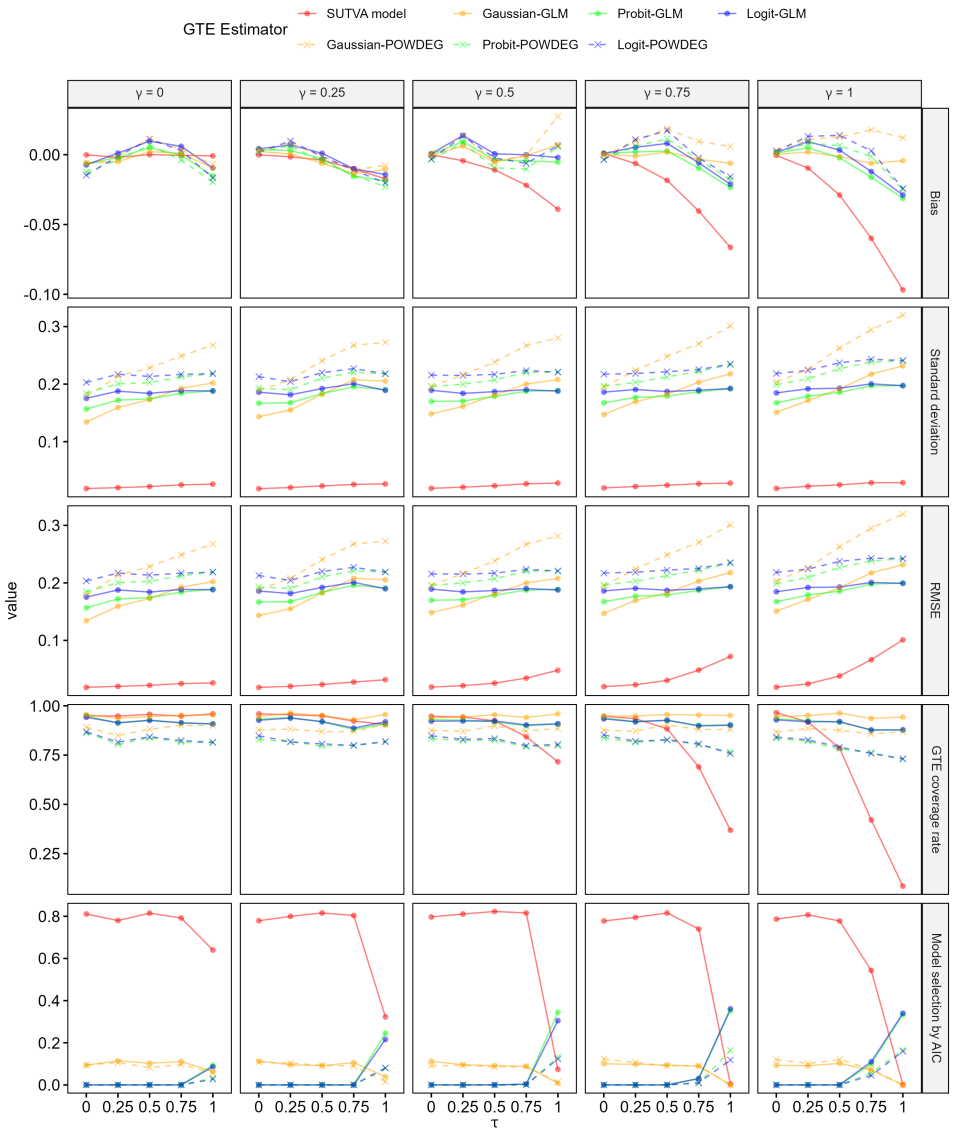



Figure 3.4: Performances of different GTE estimators under Model (3.6) for the Caltech network.

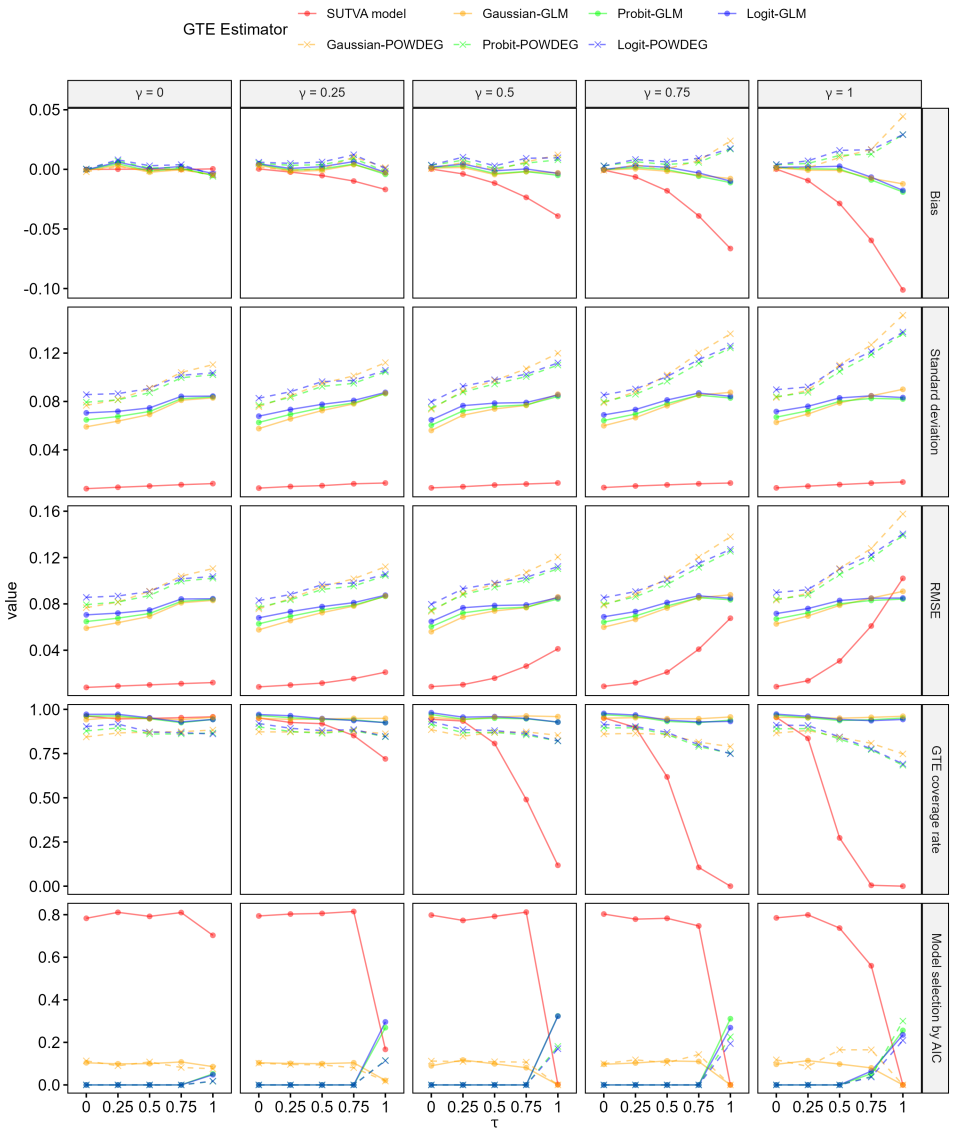



Figure 3.5: Performances of different GTE estimators under Model (3.6) for the UMichigan network.

models remain stable. This is because the predicted outcomes of binary models are contained between 0 and 1 while the predicted outcomes of continuous models can vary greatly as the size of treatment effect increases. Therefore, as is expected, experimenters should generally use binary models to analyze experiments with binary outcomes.

Finally, we can see that the linear specifications of the binary model in (3.1) perform relatively well for all evaluation criteria, including bias, variance, RMSE, and coverage. The nonlinear specifications perform worse due to uncertainty coming from additional parameters. However, this problem is alleviated with more data as shown in the results for the UMichigan network (Figure 3.5). Moreover, we find the choice of link functions does not seem to have any noticeable influence on the results. Since the AIC reasonably selects models that perform well across all evaluation criteria (the bottom row of Figures 3.4 and 3.5), we recommend that experimenters try both linear and nonlinear specifications of Model (3.1), and select the appropriate model using the AIC.

## 3.3 Application to the Agricultural Insurance Data

### 3.3.1 Data Description

In this section, we consider the agricultural insurance data from Cai et al. (2015). The data was collected from a large-scale experiment concerning farmers' insurance, which involved about 5,000 households in 250 villages in rural China. The experimenters were interested in verifying the hypothesis that a better understanding of the policy will enhance the insurance uptake rate. They were also interested in understanding the social effects of information spread that may influence the farmers' insurance purchase decision.

To evaluate this, the experimenters created two types of information sessions: a 20-minute simple information session and a 40-minute intensive information session. In this experiment, the simple information session acts as the control while the intensive information session acts as the treatment. The sessions were held in two rounds, each round comprised one simple and one intensive information session. The first round and the second round were held three days apart. This allowed some time for the information from the first round sessions to spread among friends and family, but not likely to the whole network. Friendship links were established by asking the farmers' household heads to list five close friends with whom they often discuss

rice production or financial issues. These friends could be in the same or different villages.

In each village, the farmers' households are randomized into one of the four information sessions. After that, farmers assigned to the second round are further randomized into one of the following three conditions: (a) farmers are shown the overall uptake rate of their village in the first round, (b) farmers are shown the detailed list of their village's first-round attendees and their purchase decisions, and (c) farmers are not shown any further information. The experimental design is summarized in Figure 3.6. The outcome of interest, i.e., the farmers' insurance purchase decisions, was recorded at the end of each information session.

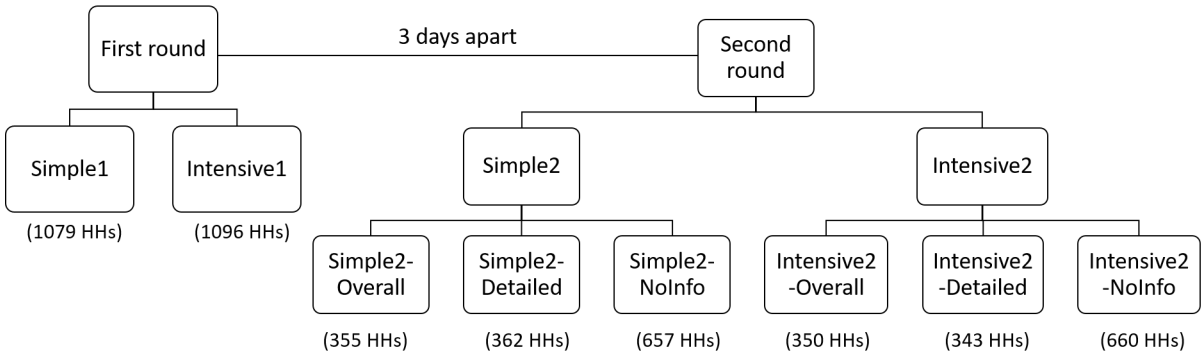



Figure 3.6: The within-village experimental design of the agricultural insurance experiment, adapted from Figure 1.1 of Cai et al. (2015). The numbers of households in each group in the raw data are given in brackets.

There are other experimental conditions on pricing and administrative style applied to the village level, but in this analysis we will only focus on the household-level experimental design. Following Cai et al. (2015), we model variations among villages using fixed effects in the regression and robust clustered variances (see Appendix B.1.2). Overall, the data set contains information (covariates) on individual households such as gender, age, literacy, etc. of the household head, their treatment assignment, their purchase decisions (outcome), and the friendship links among them.

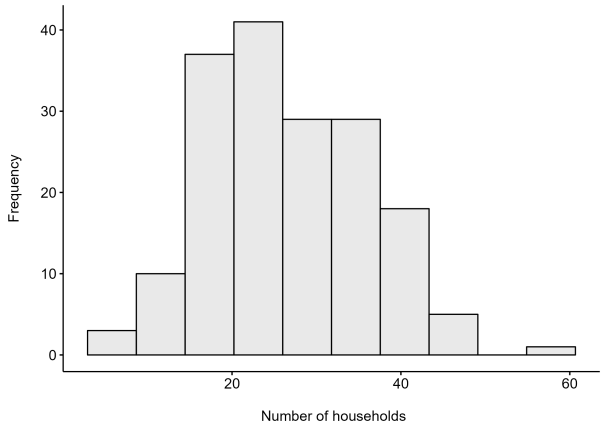



Figure 3.7: The distribution of the number of households in each village in the processed data set.

### 3.3.2 Our Analysis

We consider all households in both the first and second rounds of the experiment. Different from Chin (2019) who only consider the treatment and outcome, we add into our analysis covariates about the farmers' households. Following Cai et al. (2015), we consider (i) whether the household's head is male, (ii) the age of the household's head, (iii) household size, (iv) area of rice production, (v) whether the household's head is literate, and (iv) dummy variables for villages. We also add dummy variables indicating whether the second-round participants received overall or detailed uptake information from their fellow villagers who participated in the first round. Households with missing data are removed. This leaves us with a data set of 4,514 households (units) from 173 villages. The distribution of the number of households in each village is shown in Figure 3.7.

Since second-round households may receive information from their friends who participated in the first-round sessions, we model network effects as the (power of) number of friends who attended round 1 information sessions. We set the network effects to 0 for first-round participants. Thus, the network we consider consists of nodes representing households and edges coming from first-round participants to second-round participants. This is a directed network. Note that Model 3.1 handles network effects by functions $g_T$ and $g_C$, which we extend here for directed networks. In particular, we defined $g_T$ as the (power of) number of friends who attended the intensive session in round 1 and $g_C$ as the (power of) number of friends who attended the simple session in round 1.

Note that covariates can be incorporated into the regression formula by adding terms to the function $h(\cdot)$ in (3.1). In general, our regression formulas can be written as

$$\mu + \tau Z_i + \gamma_T q_T(U_i) + \gamma_C q_C(V_i) + \boldsymbol{\beta}^T \mathbf{x}_i$$

for each household $i$, where $\mathbf{x}_i$ is the vector of covariates for household $i$. When $i$ is a second-round participant, $U_i$ and $V_i$ are the numbers of friends of $i$ that participated in the first-round simple or intensive session, respectively. When $i$ is a first-round participant, $U_i$ and $V_i$ are set to 0. Functions $q_T$ and $q_C$ are set as one of

- Linear: $q_T$ and $q_C$ are identity functions, i.e., $q_T(x) = x$ and $q_C(x) = x$; or
- POW-DEG: $q_T(x) = x^\lambda$ and $q_C(x) = x^\lambda$; or
- POW-DEG-2: $q_T(x) = x^{\lambda_T}$ and $q_C(x) = x^{\lambda_C}$, where $\lambda_T$ can be different from $\lambda_C$.

For function $h(\cdot)$, we consider both the probit and logit links, and compare them with fitting a continuous model with identity link. We also compare these binary models to the simple SUTVA model (3.7) in Section 3.2.3. Following Cai et al. (2015), we calculate the robust clustered standard errors (see Appendix B.1.2) for all parameter estimates with the grouping variable being the village membership. The GTE estimates are calculated as the difference in average outcomes of second-round participants when all farmers go to intensive sessions versus when they all go to simple sessions.

Our analysis strategy differs from that of Cai et al. (2015) in various aspects. First, we consider the first-round and second-round participants together, instead of separately. This allows us to obtain a single estimate for the "direct" treatment effect from a larger data set. Second, while Cai et al. (2015) consider dummy variables for one, two, or more than two friends going to the first-round intensive session to consider nonlinear network effects, we take a more parsimonious approach using the POW-DEG and POW-DEG-2 specifications. This yields single estimates of network effects coming from treatment or control, which is more convenient for effect quantification and testing. Finally, Cai et al. (2015) use a simple linear regression for binary data, which can be problematic as we illustrated in the simulation in Section 3.2.3. We instead consider binary response models and compare them to continuous response models.

In addition, our analysis also differs from previous analyses from the network experimentation literature, such as Chin (2019), in that we are able to capture more

aspects of the data and the experimental design. For example, we incorporate covariates, consider the effect of the treatment delay (second-round vs first-round sessions), and calculate the robust clustered variances with respect to village membership. Moreover, Chin (2019) combine both first-round and second-round participants into a single network, which assumes that second-round participants can influence the decisions of first-round participants. This is not true because the first-round participants already made their decisions at the end of their sessions. Thus, our approach of considering only the influence of first-round participants to second-round participants aligns better with the experimental design.

Our results are given in Table 3.1. We can see that the results are relatively consistent across different models. The experimenters' hypothesis that a better understanding of the insurance policy will lead to an increased uptake rate is well-supported as the effect from intensive information sessions is found to be positive and highly significant. All models also agree that the GTE is highly significant. However, by ignoring the network effects, Model (1) estimates the GTE from 3 to 10 percentage points less than those of the other models. Model (1) is also not able to conduct tests for $H_{02}$ or $H_{03}$ like the other models.

In terms of network effects, all models agree that there are some network effects, since in all cases the test corresponding to $H_{02}$ is significant. The network effect from treatment is found to be significant in linear (Models (2)-(4)) and POW-DEG-2 models (Models (8-10)). The POW-DEG specification did not return a similar result. This implies that the test for specific network effects (coming from treatment or control) can depend on the network effect functions considered. Nevertheless, the POW-DEG-2 specifications have the smallest AIC values compared to their corresponding linear and POW-DEG counterparts. Moreover, note that the difference between the first- and second-round sessions can be attributed mostly to the network effects. While in the other network effect specifications, the effect of second round is still significant, in the POW-DEG-2 specifications, such an effect is no longer significant. This may imply that the POW-DEG-2 specifications manage to capture all the network effects received by second-round participants. Therefore, the POW-DEG-2 specifications seem to be the most appropriate network effect specifications for this data. According to the POW-DEG-2 specifications, both the network effects from the treatment and the control are significant, with the network effect coming from the treatment being positive and the network effect coming from the control being negative. This indicates that a household will be more likely to purchase the insurance if they have more friends who know the product well, which agrees with the findings by Cai et al. (2015) who used a different analysis approach. Moreover,

with the POW-DEG-2 specification, we also find that having friends who do not understand the policy well can decrease the purchase inclination. This is a new finding compared to the original analysis by Cai et al. (2015), where they also report a negative, but not significant network effect coming from friends in the simple session.

In terms of covariates, we find age and literacy of household heads highly significant for insurance purchase decisions across all models. On the other hand, the gender of household heads is not significant. Finally, the model selection result by AIC reiterates that the continuous models are not appropriate for binary data. Moreover, for this data set, logit models are preferred compared to probit models. Overall, the AIC suggests that we choose and hence draw conclusions using Model (10), i.e., the logit model with POW-DEG-2 network effect specifications.

## 3.4 Conclusion

In this chapter, we extended the GANE model framework to analyze network experiments with binary outcomes. We outlined the inference procedure and conducted simulation studies to investigate the performance of the model. We find that using models for continuous data may be unfavorable for binary data, especially for large treatment effects. We also find binary-outcome models are more difficult to estimate due to less variation in outcome values. Therefore, experiments with binary outcomes will typically require larger sample sizes than with continuous outcomes to achieve the same precision. The experimenters also need to be careful in designing the experiments so that the numbers of 1s and 0s in the outcomes are adequate for model estimation and inference. Finally, the applicability of our method is illustrated by a re-analysis of the agricultural insurance data from Cai et al. (2015). In particular, our framework is close to the regression framework that practitioners are familiar with while providing more flexibility with respect to network effect inference.

The popularity of experiments for binary outcomes on networks calls for more research in the future. As discussed, these experiments may require larger sample sizes. Thus, design and sample size calculation will be an important topic for future research. Furthermore, in this chapter, we did not consider the autoregressive effect of other units' outcomes on a unit. However, it is possible that observing/discussing the outcomes with other units may influence a unit's final decision. In such cases, other model structures, for example, binary models that incorporate latent variables (Klier and McMillen, 2008; Piras and Sarrias, 2023), may be considered. This is an exciting topic for future research.

| Models | (1) | (2) | (3) | (4) | (5) | (6) | (7) | (8) | (9) | (10) |
|---|---|---|---|---|---|---|---|---|---|---|
| ***Model specification*** | | | | | | | | | | |
| Link | Identity | Identity | Probit | Logit | Identity | Probit | Logit | Identity | Probit | Logit |
| Network functions | None | Linear | Linear | Linear | POW-DEG | POW-DEG | POW-DEG | POW-DEG-2 | POW-DEG-2 | POW-DEG-2 |
| ***Estimation*** | | | | | | | | | | |
| Intensive Session | 0.0788*** | 0.0761*** | 0.2058*** | 0.3397*** | 0.0765*** | 0.2066*** | 0.3418*** | 0.0757*** | 0.2049*** | 0.3392*** |
| | (0.0152) | (0.0152) | (0.0411) | (0.0675) | (0.0153) | (0.0413) | (0.0679) | (0.0152) | (0.0413) | (0.068) |
| Net.Eff from Treatment | | 0.0558*** | 0.1516*** | 0.2486*** | 0.0195 | 0.0493 | 0.0816 | 0.0453* | 0.1233*** | 0.1996*** |
| | | (0.0132) | (0.0364) | (0.059) | (0.0129) | (0.034) | (0.0567) | (0.0192) | (0.0354) | (0.0557) |
| Net.Eff from Control | | 0.0098 | 0.0282 | 0.0438 | 0.0089 | 0.024 | 0.0377 | -0.0599** | -0.1605** | -0.272** |
| | | (0.0115) | (0.0313) | (0.051) | (0.008) | (0.0228) | (0.036) | (0.0218) | (0.0575) | (0.0943) |
| Info - Overall | | -0.0431. | -0.1174. | -0.1923. | -0.0431. | -0.1183. | -0.1928. | -0.0434. | -0.1188. | -0.1949. |
| | | (0.0253) | (0.0707) | (0.1162) | (0.0254) | (0.0709) | (0.1165) | (0.0251) | (0.0703) | (0.1157) |
| Info - Detailed | | 0.0388 | 0.1074 | 0.1739 | 0.034 | 0.0957 | 0.1533 | 0.0508. | 0.1391. | 0.2279. |
| | | (0.0283) | (0.0763) | (0.1246) | (0.0282) | (0.0763) | (0.1246) | (0.0281) | (0.0761) | (0.1245) |
| Second Round | | -0.0739** | -0.2007** | -0.3283** | -0.0615* | -0.1641* | -0.2696* | -0.04 | -0.1088 | -0.1764 |
| | | (0.0268) | (0.0728) | (0.1181) | (0.026) | (0.0705) | (0.1145) | (0.0262) | (0.0757) | (0.1222) |
| Male | | 0.0359 | 0.098 | 0.1572 | 0.0348 | 0.0945 | 0.153 | 0.0378 | 0.1039 | 0.1649 |
| | | (0.0316) | (0.0874) | (0.142) | (0.0316) | (0.0872) | (0.1418) | (0.0316) | (0.0874) | (0.1421) |
| Age | | 0.0041*** | 0.0112*** | 0.019*** | 0.004*** | 0.0111*** | 0.0187*** | 0.004*** | 0.0111*** | 0.0188*** |
| | | (0.0008) | (0.0024) | (0.0036) | (0.0008) | (0.0024) | (0.0036) | (0.0008) | (0.0023) | (0.0035) |
| Household Size | | -0.0076* | -0.0216* | -0.0355* | -0.0073. | -0.0206* | -0.034* | -0.0079* | -0.0226* | -0.037* |
| | | (0.0038) | (0.0103) | (0.017) | (0.0038) | (0.0104) | (0.017) | (0.0038) | (0.0103) | (0.0169) |
| Rice Prod. Area | | 0.0018** | 0.0053 | 0.0117** | 0.0018** | 0.0053 | 0.0117** | 0.0018** | 0.0053 | 0.0118** |
| | | (0.006) | (0.0037) | (0.0038) | (0.006) | (0.0036) | (0.0038) | (0.006) | (0.0037) | (0.0038) |
| Literacy | | 0.0972*** | 0.2664*** | 0.4316*** | 0.0958*** | 0.2621*** | 0.4259*** | 0.0956*** | 0.2622*** | 0.4259*** |
| | | (0.0185) | (0.0511) | (0.0842) | (0.0185) | (0.0512) | (0.0844) | (0.0184) | (0.05) | (0.0825) |
| ***Model fitness*** | | | | | | | | | | |
| AIC | 6482.513 | 5910.036 | 5895.781 | 5890.616 | 6187.664 | 5893.323 | 5888.02 | 6186.015 | 5892.178 | 5886.358 |
| ***Other inferences*** | | | | | | | | | | |
| GTE | 0.0788*** | 0.1436*** | 0.1408*** | 0.1426*** | 0.1138*** | 0.1079*** | 0.1109*** | 0.1843*** | 0.1757*** | 0.1784*** |
| | (0.0152) | (0.0311) | (0.0305) | (0.0304) | (0.026) | (0.0255) | (0.0252) | (0.0362) | (0.0309) | (0.0306) |
| Test for $H_{02}$ | | *** | *** | *** | *** | *** | *** | *** | *** | *** |
| Test for $H_{03}$ | | ** | ** | ** | | | | *** | *** | *** |

Table 3.1: Model fitting results. The coefficients are rounded to the nearest 4 digits. Significance codes ***, **, *, and . correspond to significance levels 0.001, 0.01, 0.05, and 0.1, respectively. Robust clustered standard errors are given in brackets. Fixed effect estimates of individual villages are not directly of interest and thus are not displayed in this table.

# Chapter 4

# Optimal Bayesian Designs for Network A/B Testing

Given a model for the experimental outcomes, a good design for the experiment can be found by optimizing a design criterion. The design criterion should be deliberately chosen to be related to the efficiency of the analysis that will be conducted based on the model. As demonstrated in Chapter 2, to capture the complicated network interference patterns, outcome models for network experiments can be complex. Therefore, design criteria for network experiments usually involve unknown parameters and may not have a closed-form formula. This limits the use of classical optimal design methods.

In this chapter, we formulate a *Bayesian design criterion* based on the mean squared error of the GTE estimator. In optimizing such a criterion, we select a design that is *on average* near-optimal with respect to the prior distribution of model parameters. Since Bayesian design criteria often do not have a closed-form expression, we adapt and investigate a variety of general-use optimization algorithms to find the optimal design on networks. We investigate and compare the effectiveness of these algorithms over multiple model specifications and data sets. Our results help characterize generally good designs for network A/B testing.

## 4.1 Design Criterion

### 4.1.1 The Model-Based Design Problem

We consider the same setting as in Section 1.2.1, where an A/B test is conducted on a simple and undirected network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The network $\mathcal{G}$ has $|\mathcal{V}| = n$ nodes and is represented by the adjacency matrix $\mathbf{A}$. In an A/B test, there are only two experimental conditions, treatment and control. We denote the treatment assignment vector, or the design of the experiment, by $\mathbf{Z}$, where $Z_i = 1$ indicates that unit $i$ is assigned to treatment and $Z_i = 0$ indicates that unit $i$ is assigned to control. Let $\mathbf{X}$ be an $n \times p$ matrix of possible covariates and $\mathbf{Y}$ be the $n \times 1$ vector of experimental responses. In this chapter, we consider both cases when $\mathbf{Y}$ is continuous and binary.

As discussed in Section 1.2.2, model-based approaches to the design and analysis of experiments on networks impose a probabilistic model on the experimental response

$$\mathbf{Y} \sim p_{\mathbf{Y}}(\mathbf{D}, \boldsymbol{\theta}), \tag{4.1}$$

where $\mathbf{D} = \{\mathbf{A}, \mathbf{Z}, \mathbf{Y}, \mathbf{X})$ is the experimental data and $\boldsymbol{\theta} \in \boldsymbol{\Theta} \subset \mathbb{R}^d$ represents possible parameters of the distribution $p_{\mathbf{Y}}$. We assume that the covariates $\mathbf{X}$ and the network $\mathbf{A}$ are given, while parameters $\boldsymbol{\theta}$ are unknown and the experimental design $\mathbf{Z}$ is to be determined. An optimal design $\mathbf{Z}^*$ is defined as

$$\mathbf{Z}^* = \underset{\mathbf{Z} \in \{0,1\}^n}{\operatorname{argmin}}\, \phi(\mathbf{Z}),$$

where $\phi(\cdot)$ is a design criterion specified by the experimenter and derived based on the response model $p_{\mathbf{Y}}$.

As discussed in Section 2.2.5, in network A/B test setting, the global treatment effect (GTE) is often the quantity of primary interest. It is defined as the difference in average response when the *whole* network is assigned to treatment versus control

$$\text{GTE} = \frac{1}{n} \sum_{i=1}^{n} \left( \mathbb{E}\Big[ Y_i \Big| \mathbf{Z} = \mathbf{1}_n \Big] - \mathbb{E}\Big[ Y_i \Big| \mathbf{Z} = \mathbf{0}_n \Big] \right). \tag{4.2}$$

An estimator of the GTE can be determined prior to the experiment and the design could be chosen in order to minimize some efficiency measure of this GTE estimator. In this chapter, we focus on minimizing the MSE of the GTE estimator. Nevertheless, the methodologies presented in this chapter can be applied to other possible design

criteria such as the D-optimality criterion (Pukelsheim, 2006; Pokhilko et al., 2019) or the variances of specific model parameters (Parker et al., 2017; Koutra et al., 2021).

In classical experimental design, the design criterion can often be expressed in a closed-form formula and the formula is often free of unknown model parameters. Hence, exact optimization techniques such as gradient-based algorithms (Silvey et al., 1978; Foster et al., 2020) or integer programming (Pokhilko et al., 2019) can be used. However, models for network experiments are often complex in order to capture the interference patterns. This typically makes the design criterion depend on unknown parameters. Moreover, in some cases, the design criterion does not have a closed-form formula. We illustrate this via the examples below.

**Example 4.1. The Power-Degree (POW-DEG) Model:** In the POW-DEG model (2.9) of Chapter 2, the GTE can be expressed explicitly as

$$\text{GTE} = \tau + \frac{\gamma_T - \gamma_C}{n} \sum_{i=1}^{n} K_{ii}^{\lambda}, \tag{4.3}$$

where $K_{ii}$ is the degree (or the number of connections) of unit $i$ in the network. As discussed in Section 2.3.3, we can obtain an unbiased estimate of this GTE by plugging in the maximum likelihood estimates of $\tau, \gamma_T, \gamma_C$ and $\lambda$ into (4.3). The variance of this GTE estimator can then be derived using the Delta method (Doob, 1935; Van der Vaart, 2000)

$$\text{Var}(\widehat{\text{GTE}}) = \mathbf{d}^\top \boldsymbol{\Sigma}^{-1} \mathbf{d}, \tag{4.4}$$

where

$$\mathbf{d}^\top = [\ 0 \quad 1 \quad \tfrac{1}{n}\textstyle\sum_{i=1}^{n} K_{ii}^{\lambda} \quad -\tfrac{1}{n}\textstyle\sum_{i=1}^{n} K_{ii}^{\lambda} \quad \tfrac{\gamma_T - \gamma_C}{n}\textstyle\sum_{i=1}^{n} K_{ii}^{\lambda} \log K_{ii}\ ],$$

$$\boldsymbol{\Sigma} = \frac{1}{\sigma^2} \begin{bmatrix} \mathbf{M}^\top \mathbf{M} & \mathbf{M}^\top \dot{\mathbf{M}} \boldsymbol{\beta} & 0 \\ \mathbf{M}^\top \dot{\mathbf{M}} \boldsymbol{\beta} & \boldsymbol{\beta}^\top \dot{\mathbf{M}}^\top \dot{\mathbf{M}} \boldsymbol{\beta} & 0 \\ 0 & 0 & \frac{1}{2\sigma^2} \end{bmatrix},$$

in which

$$\mathbf{M} = [\ \mathbf{1}_n \quad \mathbf{Z} \quad \mathbf{G}_T \quad \mathbf{G}_C\ ], \qquad \dot{\mathbf{M}} = [\ \mathbf{0}_n \quad \mathbf{0}_n \quad \dot{\mathbf{G}}_T \quad \dot{\mathbf{G}}_C\ ],$$

$$\{\mathbf{G}_T\}_i = \left(\sum_{j=1}^{n} A_{ij} Z_j\right)^{\lambda}, \qquad \{\dot{\mathbf{G}}_T\}_i = \left(\sum_{j=1}^{n} A_{ij} Z_j\right)^{\lambda} \log\left(\sum_{j=1}^{n} A_{ij} Z_j\right),$$

$$\{\mathbf{G}_C\}_i = \left(\sum_{j=1}^{n} A_{ij}(1-Z_j)\right)^{\lambda}, \qquad \{\dot{\mathbf{G}}_C\}_i = \left(\sum_{j=1}^{n} A_{ij}(1-Z_j)\right)^{\lambda} \log\left(\sum_{j=1}^{n} A_{ij}(1-Z_j)\right),$$

and

$$\boldsymbol{\beta} = [\ \mu \quad \tau \quad \gamma_T \quad \gamma_C\ ]^\top.$$

As the maximum likelihood estimator is asymptotically unbiased, we use $\text{Var}(\widehat{\text{GTE}})$ as opposed to $\text{MSE}(\widehat{\text{GTE}})$ as our design criterion. It is clear from the formula above that this design criterion involves unknown parameters $\lambda$ and $\boldsymbol{\beta}$. $\triangle$

**Example 4.2. The Binary Network-Temporal Autoregressive (BNTAR) Model:** Consider the response generating model (3.6) from the simulation in Section 3.2.3.

$$\begin{aligned} Y^*_{i,t} &= \mu + \tau Z_i + \gamma \frac{1}{K_{ii}} \sum_{j=1}^{n} A_{ij} Y_{j,t-1} + \epsilon_{i,t}, \\ Y_{i,t} &= \mathbb{I}(Y^*_{i,t} > 0), \end{aligned} \tag{4.5}$$

where $Y_{i,t}$ is the response of unit $i$ at time step $t$ with $0 \le t \le T$, and the corresponding error $\epsilon_{i,t}$ follows the standard normal distribution $\mathcal{N}(0,1)$. Hence, Model (4.5) assumes a latent variable $Y^*_{i,t}$ at time $t$ that is dependent on the average response of neighbors at time $t-1$. The response $Y_{i,t}$ will be equal to 1 if this latent variable is greater than a threshold, here 0. The model is inspired by graphical games (Eckles et al., 2016), and is also adopted by Gui et al. (2015) and Chin (2019) as the response-generating model for their simulation studies. The model's popularity is due to its reasonable dynamics and nonlinear structure, which can be used to examine the performance of experimental analysis methods whose constructions are based on a linearity assumption. However, the complex structure of the model makes it hard to derive the formula for the GTE analytically. Moreover, the GTE also depends on specific values of the model parameters. Indeed, in Eckles et al. (2016), the true GTE is estimated using Monte Carlo approximation for each combination of parameter values. The MSE of any GTE estimator, which is the design criterion for this model, also needs to be estimated using Monte Carlo approximation. $\triangle$

**Example 4.3. The Conditional Network Autoregressive (CNAR) Model:** Pokhilko et al. (2019) and Zhang and Kang (2022) adopt the conditional autoregressive model from the spatial statistics literature (Besag, 1974) to model the response of a network experiment. A simple form of the model can be written as

$$Y_i = \mu + \tau Z_i + \epsilon_i,$$

$$\epsilon_i | \boldsymbol{\epsilon}_{-i} \sim \mathcal{N}\left(\rho \sum_{j=1}^{n} \frac{A_{ij}\epsilon_j}{K_{ii}}, \frac{\sigma^2}{K_{ii}}\right), \tag{4.6}$$

where $\boldsymbol{\epsilon}_{-i}$ is the error vector without the $i$th element. The CNAR model has four parameters: $\mu$ is the baseline response, $\tau$ is the effect of the treatment, $\sigma^2$ is the variance, and $0 \leq \rho < 1$ characterizes the strength of the network dependence between neighbors. Under the CNAR model, the GTE is $\tau$ and the joint distribution of the errors can be derived as (Pokhilko et al., 2019; Zhang and Kang, 2022)

$$\boldsymbol{\epsilon} \sim \mathcal{N}\left(\mathbf{0}, \sigma^2(\mathbf{K} - \rho\mathbf{A})^{-1}\right),$$

where $\mathbf{K}$ is the diagonal matrix of the degrees $K_{ii}$ (number of unit $i$'s connections). When the parameter $\rho$ is known, we can estimate the parameters $\boldsymbol{\beta} = (\mu, \tau)^\top$ using weighted least squares regression, which yields

$$\hat{\boldsymbol{\beta}} = (\mathbf{M}^\top(\mathbf{K} - \rho\mathbf{A})\mathbf{M})^{-1}\mathbf{M}^\top(\mathbf{K} - \rho\mathbf{A})\mathbf{Y}, \tag{4.7}$$

where $\mathbf{M} = [\ \mathbf{1}_n \quad \mathbf{Z}\ ]$ is the model matrix. This weighted least squares estimator is unbiased. Thus, we can set the design criterion as $\text{MSE}(\widehat{\text{GTE}}) = \text{Var}(\widehat{\text{GTE}}) = \text{Var}(\hat{\tau})$. We derive the variance-covariance matrix to be

$$\text{Var}(\hat{\boldsymbol{\beta}}) = \sigma^2(\mathbf{M}^\top(\mathbf{K} - \rho\mathbf{A})\mathbf{M})^{-1}, \tag{4.8}$$

where $\text{Var}(\hat{\tau})$ is the (2,2)th element. Both Pokhilko et al. (2019) and Zhang and Kang (2022) consider the case of a pre-specified $\rho$ and find the locally optimal design using integer programming. However, it may be difficult to choose a specific value for $\rho$ in practice. $\triangle$

In all examples, the design criterion is complicated, which involves unknown parameters or requires Monte Carlo approximation. In the case of unknown parameters, we can assume specific values for these parameters and work on the closed-form formula of the design criterion to find locally optimal designs using customized integer programming (Pokhilko et al., 2019; Zhang and Kang, 2022). However, it is unclear how to choose these values in practice. Finally, if the design criterion does not have an analytic formula, it is clear that we cannot use any traditional approach to construct optimal designs. Thus, we develop more generally useful design methods in this chapter.

### 4.1.2 Bayesian Design Criterion

To alleviate the problem of unknown parameters within design criteria for network experiments, we propose to set a prior $p_{\boldsymbol{\eta}}$ to describe their distribution (not depending on the yet-to-be-observed data), where $\boldsymbol{\eta} \subseteq \boldsymbol{\theta}$ denotes the unknown parameters. Such a prior can be specified to be either informative or noninformative according to the experimenters' domain knowledge. The design criterion can then be formulated as

$$\phi(\mathbf{Z}) = \int \phi_0(\mathbf{Z}, \boldsymbol{\eta}) p_{\boldsymbol{\eta}} d\boldsymbol{\eta}, \tag{4.9}$$

where $\phi_0$ is the target design criterion and the design criterion $\phi$ is the expected value of $\phi_0$ over the prior distribution of the model parameters. In our setting, $\phi_0 = \text{MSE}(\widehat{\text{GTE}})$. The design criterion $\phi$ in Equation (4.9) is called a Bayesian design criterion. Some authors may prefer to call the this the pseudo-Bayeisan optimal design criterion since the analysis is still carried out in the frequentist framework (Ryan et al., 2016). Other types of Bayesian design criteria can be found in Chaloner and Verdinelli (1995).

Since $p_{\boldsymbol{\eta}}$ can be freely chosen, the design criterion $\phi$ in Equation (4.9) generally does not have a closed-form expression. In this case, we can estimate $\phi(\mathbf{Z})$ using a Monte Carlo approximation

$$\phi(\mathbf{Z}) \approx \hat{\phi}(\mathbf{Z}) = \frac{1}{L} \sum_{l=1}^{L} \phi_0(\mathbf{Z}, \boldsymbol{\eta}_l), \quad \text{where } \boldsymbol{\eta}_l \sim p_{\boldsymbol{\eta}}, \tag{4.10}$$

for large $L$. Now, our problem amounts to finding the optimal design $\mathbf{Z} \in \{0, 1\}^n$ that optimizes $\hat{\phi}(\mathbf{Z})$. This is a discrete optimization problem where the objective function $\phi(\mathbf{Z})$ does not have a closed-form formula. In the next section, we will discuss algorithms to construct optimal designs for the defined Bayesian design criterion.

## 4.2 Design Construction Algorithms

As discussed in the previous section, the design construction problem is equivalent to a discrete optimization problem where the objective function, i.e. our design criterion $\phi(\mathbf{Z})$, does not have a closed-form formula. This impedes the use of optimal design strategies that are tailored to a specific model and/or require an analytical formula of the design criterion (Pokhilko et al., 2019; Koutra et al., 2021; Zhang

and Kang, 2022). Furthermore, as the network size $n$ increases, the search space ($|\{0,1\}^n| = 2^n$) increases exponentially, making it difficult to find the exact solution to the optimal design problem. In summary, our challenges include (i) a discrete and exponentially large search space of size $2^n$ and (ii) a non-differentiable and difficult-to-evaluate objective function. So, we seek to use general optimization algorithms that find near-optimal solutions. In this section, we describe three classes of design construction algorithms: (i) meta-heuristic search, (ii) Bayesian optimization, and (iii) graph cluster randomization. We discuss several specific algorithms under each of these strategies and we present how they can be applied in the network A/B test setting.

### 4.2.1 Meta-heuristic Search

Meta-heuristic searches (Gendreau et al., 2010) perform a partial search in the solution space. They usually involve a local improvement (*greedy*) step and a random step to help the search escape from local optima. Meta-heuristic algorithms are often used in problems with a large search space but constrained computational resources. They have been used in the experimental design literature to find near-optimal designs (García-Ródenas et al., 2020) especially when the design criterion does not have a closed-form formula. However, these applications are on continuous designs, instead of discrete designs as in our problem. There has also not been any application of these algorithms to the network A/B testing problem. Below, we introduce four popular meta-heuristic algorithms (Martin and Quinn, 1996), and explain how we adapt these algorithms to our network A/B testing problem.

#### 4.2.1.1 Random Search

One of the simplest meta-heuristic algorithms is *random search* (Spall, 2005), where a large number of designs is randomly generated from some distribution. The generated designs are then evaluated based on the prespecified design criterion, and the design with the smallest value of $\phi$ will be chosen as the "optimal" solution. Parker et al. (2017) compared random search with an exchange algorithm where $Z_i$, for $i = 1, ..., n$, is changed iteratively so as to decrease the value of $\phi$. They found that with respect to their LNE model, random search is able to achieve a comparable result while being less computationally complex.

Although the exact random search algorithms can be varied according to different design-generating distributions, in our case, we use the uniform distribution to

generate candidate designs, i.e.,

$$\mathbb{P}(\mathbf{Z} = \mathbf{z}) = \frac{1}{2^n}, \qquad \text{for } \mathbf{z} \in \{0,1\}^n. \tag{4.11}$$

This distribution is achieve by sampling $Z_i \sim \text{Bernoulli}(1/2)$ independently for $i = 1, 2, ..., n$.

#### 4.2.1.2 Tabu Search

While random search performs a "global search" where the designs are randomly generated from a distribution, *tabu search* (Kernighan and Lin, 1970; Reeves, 1993) performs a "local search". In each step, the search will move to the best neighboring design (defined below) relative to the current design. To prevent the search from getting trapped in suboptimal neighborhoods, the tabu search algorithm keeps track of recently visited designs in a tabu list and forbids the search to go back to that region, with exceptions following an aspiration criterion.

Implementations of tabu search vary in terms of the definition of a design's neighborhood, how the tabu list is updated, and the aspiration rules. In our case, we define neighbors of a design $\mathbf{Z} \in \{0,1\}^n$ as designs $\mathbf{Z}'$ which have at most $100\alpha_{\text{neighbor}}\%$ of its elements different from $\mathbf{Z}$. Since there can be a large number of such neighbors, we only randomly generate and evaluate a fixed number of neighbors $m_{\text{neighbor}}$ in each iteration. We also set the tabu list to be of fixed size $m_{tabu}$. Thus, if a new design is added, the oldest design is removed from the list. Finally, the aspiration criterion is determined by the event $U < \alpha_{\text{aspire}}$, where $U$ follows a uniform distribution in $[0,1]$ and $\alpha_{\text{aspire}}$ is called the aspiration rate. We set $\alpha_{\text{neighbor}} = \alpha_{\text{aspire}} = 0.1$ and $m_{\text{neighbor}} = m_{\text{tabu}} = 100$. The detailed algorithm is summarized in Table 4.1. We defer the discussion about the stopping criterion in Step 6 to Section 4.3.2.

#### 4.2.1.3 Simulated Annealing

Similar to tabu search, *simulated annealing* (Kirkpatrick et al., 1983) is a local search algorithm. However, instead of searching for the best neighboring design, the simulated annealing algorithm randomly chooses one neighboring design of the current design. The search will move to this new design with the acceptance probability

$$\mathbb{P}(\text{accept } \mathbf{Z}^{(\text{new})}) = \min\left\{1, \exp\left[-\frac{\phi(\mathbf{Z}^{(\text{current})}) - \phi(\mathbf{Z}^{(\text{new})})}{T_{\text{temp}}}\right]\right\}, \tag{4.12}$$

***Tabu Search***

| | |
|---|---|
| Step 1. | Randomly generate an initial design $\mathbf{Z}^{(0)}$. Set $t = 0$ and $\mathbf{Z}^* = \mathbf{Z}^{(0)}$. |
| Step 2. | Generate $m_{\text{neighbor}}$ neighboring designs, and evaluate them according to the given design criteria. Order the neighboring designs from the best to worst. |
| Step 3. | For each neighboring design in the ordered candidate list, starting with the best one: if the design does not lie in the tabu list or if it does but the aspiration criterion is satisfied, then set it to be $\mathbf{Z}^{(t+1)}$. Otherwise, examine the next best candidate. If none of the candidates is accepted, set $\mathbf{Z}^{(t+1)} = \mathbf{Z}^{(t)}$. |
| Step 4 | Add $\mathbf{Z}^{(t+1)}$ into the tabu list. Remove the oldest design in the tabu list if the tabu list now contains $m_{\text{tabu}} + 1$ designs. |
| Step 5 | If $\phi(\mathbf{Z}^{(t+1)}) < \phi(\mathbf{Z}^*)$, then set $\mathbf{Z}^* = \mathbf{Z}^{(t+1)}$. |
| Step 6 | Stop if the stopping criterion is met and return $\mathbf{Z}^*$, otherwise set $t = t+1$ and go to Step 2. |

Table 4.1: Our implementation of tabu search.

where $T_{\text{temp}} > 0$ is called the temperature. From Equation (4.12), we can see that the search will certainly move to the new design if the new design is better in terms of the design criterion. If the new design is worse than the current design, it will still be accepted based on a probability that depends on the difference in design criteria between the two designs and the temperature $T_{\text{temp}}$. When $T_{\text{temp}}$ is high, the acceptance probability is high and the search can explore more of the search space. However, when $T_{\text{temp}}$ is low, the search will tend to settle in a local optimum area. Inspired by the annealing process in physical chemistry where metal is heated and then cooled down, the simulated annealing algorithm starts with a high temperature for a large exploration and gradually decreases this temperature for a local greedy search. In our implementation of the simulated annealing algorithm, we define neighbors as in Section 4.2.1.2, set the initial temperature $T_{\text{temp}}$ at 1 and decrease it by $\alpha_{\text{cooling}} = 10\%$ every $m_{\text{cooling}} = 50$ iterations. The algorithm is summarized in Table 4.2.

#### 4.2.1.4 Genetic Algorithm

The genetic algorithm (Holland, 1992) was originally created to simulate species adaptations in nature, but eventually was shown to be a useful tool for optimization problems (Martin and Quinn, 1996). In the context of the genetic algorithm, each possible design is transformed into or regarded as a chromosome and the algorithm starts with a population of chromosomes. As the algorithm progresses, the chromo-

***Simulated Annealing***

| | |
|---|---|
| Step 1. | Randomly generate an initial design $\mathbf{Z}^{(0)}$. Set $t = 0$ and $\mathbf{Z}^* = \mathbf{Z}^{(0)}$. |
| Step 2. | Generate a neighbor $\mathbf{Z}^{(\text{new})}$, and evaluate it according to the given design criteria. Set $\mathbf{Z}^{(t+1)} = \mathbf{Z}^{(\text{new})}$ with probability given in Equation (4.12). Otherwise, $\mathbf{Z}^{(t+1)} = \mathbf{Z}^{(t)}$. |
| Step 3. | If $\phi(\mathbf{Z}^{(t+1)}) < \phi(\mathbf{Z}^*)$, then set $\mathbf{Z}^* = \mathbf{Z}^{(t+1)}$. |
| Step 4. | Decrease the temperature by $100\alpha_{\text{cooling}}\%$ every $m_{\text{cooling}}$ values of $t$. |
| Step 5. | Stop if the stopping criterion is met and return $\mathbf{Z}^*$, otherwise set $t = t+1$ and go to Step 2. |

Table 4.2: Our implementation of simulated annealing.

somes change by means of selection, crossover, and mutation so as to best "adapt" to the objective function. That is, the chromosomes with good performance in terms of the objective function will be born and thrive in the next generation.

Implementations of the genetic algorithm differ by genetic representation of the search space and definitions of selection, crossover, and mutation operations. In our case, as our search space is discrete, we can directly regard each design $\mathbf{Z} \in \{0,1\}^n$ as a chromosome, whose elements $Z_i$'s are regarded as genes. As the population evolves, our execution of selection, crossover, and mutation operations is as follows. During selection, the top $100\alpha_{\text{elite}}\%$ designs (in terms of the design criterion) in the population will be kept unchanged and moved to the next generation (iteration). Meanwhile, the top $100\alpha_{\text{parents}}\%$ of the designs will be eligible as parents and can mate to produce offspring in the new population. In particular, besides the elite designs, the rest of the new population is created by combining the designs of two randomly selected eligible parent designs so that half of the elements in the child design come from the father and the other half come from the mother. The exact indexes of the elements $i$ for which the values $Z_i$'s come from the mother (or father) are randomly chosen. Furthermore, the children will have mutations in $\alpha_{\text{mutation}}$ of their genes, that is, for each newly created child $\mathbf{Z}$, $100\alpha_{\text{mutation}}\%$ of $Z_i$'s will be randomly generated. We choose a fixed-sized population of $m_{\text{pop}} = 100$ designs for each iteration, and we set $\alpha_{\text{elite}} = \alpha_{\text{mutation}} = 0.1$ and $\alpha_{\text{parents}} = 0.5$. Our implementation is summarized in Table 4.3.

### 4.2.2 Bayesian Optimization

Recall that our optimization problem involves the design criterion (4.9) in the form of an integral. This integral is often intractable so it is typically evaluated using

***Genetic Algorithm***

| | |
|---|---|
| Step 1. | Randomly generate an initial population of $\mathcal{Z}^{(0)} = \{\mathbf{Z}^{(0,1)}, ..., \mathbf{Z}^{(0,m_{\text{pop}})}\}$. Set $t = 0$. |
| Step 2. | Evaluate the population and order them according to the design criterion $\phi(\cdot)$. |
| Step 3. | Move the top $100\alpha_{\text{elite}}\%$ of the current population $\mathcal{Z}^{(t)}$ to the next population $\mathcal{Z}^{(t+1)}$. |
| Step 4. | For the remaining $100(1 - \alpha_{\text{elite}})\%$ of $\mathcal{Z}^{(t+1)}$, perform crossover using two randomly selected parents from top $100\alpha_{\text{parents}}\%$ of the current population $\mathcal{Z}^{(t)}$. |
| Step 5. | For each newly created child in $\mathcal{Z}^{(t+1)}$: randomly modify $100\alpha_{\text{mutation}}\%$ of each unit's treatment assignment. |
| Step 6. | Stop if the stopping criterion is met and return the best solution in the current population as $\mathbf{Z}^*$. Otherwise set $t = t + 1$ and go to Step 2. |

Table 4.3: Our implementation of the genetic algorithm.

the Monte Carlo approximation given by (4.10). Depending on the target design criterion function $\phi_0$, a large number of draws $L$ may be required for a precise approximation of $\phi(\mathbf{Z})$ in (4.10). Thus, the objective function of our optimization problem is computationally taxing to evaluate. On top of that, to explore the large search space ($2^n$), we will need to evaluate $\phi(\mathbf{Z})$ many times. To avoid this, Bayesian optimization (Mockus, 1989; Garnett, 2023) techniques have been developed. One of the main applications of Bayesian optimization is in hyperparameter tuning for machine learning models. In this chapter, we consider applying Bayesian optimization for experimental design.

In our setting, following the principles of Bayesian optimization, we represent the mapping $\mathbf{Z} \mapsto \phi(\mathbf{Z})$ by a conditional "prior" distribution $p(\phi(\mathbf{Z})|\mathbf{Z})$ that reflects our belief and uncertainty about the design criterion function $\phi(\cdot)$. As the algorithm progresses and more data $\{\mathbf{Z}, \phi(\mathbf{Z})\}$ are collected, the posterior distribution is updated accordingly. We can decide which design to explore next based on an *acquisition function* derived from this posterior distribution. The most common Bayesian optimization technique is to employ Gaussian process prediction (Williams, 1998). However, since our search space is discrete, we will need other Bayesian optimization methods. In the following subsections, we discuss (i) local search using a surrogate model for the design criterion, (ii) reinforcement learning, and (iii) the Tree-Parzen estimator.

#### 4.2.2.1 Local Search with a Surrogate Model

Since $\mathbf{Z} \in \{0,1\}^n$ is high dimensional, it can be difficult to determine a reasonable choice for the conditional prior distribution $p(\phi(\mathbf{Z}) = u|\mathbf{Z})$ for $u \in \mathbb{R}$. We can utilize the flexibility and power of neural networks to approximate the relationship between $\mathbf{Z}$ and $\phi(\mathbf{Z})$. In this case, the neural network is called a *surrogate model* for the function $\phi(\cdot)$.

To train the neural networks, we generate initial data $\{\mathbf{Z}, \phi(\mathbf{Z})\}$ with the design criteria values $\phi(\mathbf{Z})$ evaluated using Monte Carlo approximation. Since Monte Carlo estimation is computationally intensive which limits the initial sample size, the initial surrogate model tends to be inaccurate and its predictions can have high variance. Following Lakshminarayanan et al. (2017), we estimate this variance using an ensemble of neural networks that have the same architecture. In particular, the point prediction $\tilde{\phi}(\mathbf{Z})$ is set as the average $\mu(\mathbf{Z})$ of predictions made by each neural network in the ensemble. Similarly, the uncertainty of $\tilde{\phi}(\mathbf{Z})$ is estimated by the standard deviation $\sigma(\mathbf{Z})$ among these predictions.

Via an acquisition function $h_{\text{acquisition}}(\cdot)$, we can decide to greedily move the search to solutions $\mathbf{Z}$ that have low values of $\tilde{\phi}(\mathbf{Z})$, or to explore regions where the uncertainty of prediction is high. Popular acquisition functions in Bayesian optimization are upper confidence bound (UCB) (Srinivas et al., 2010), Thompson sampling (Lu and Van Roy, 2017), or posterior mean. In this article, we use the UCB acquisition function, which favors low values of $h_{\text{acquisition}}(\mathbf{Z}) = \mu(\mathbf{Z}) - \sigma(\mathbf{Z})$, since Swersky et al. (2020) find that it produces the best results for discrete optimization.

Our implementation is summarized in Table 4.4. We find that for our problem, simple networks fit the design criteria better. Thus, for the simulations in Section 4.3, we use an ensemble of 10 neural networks, each with 1 hidden layer containing 4 nodes. The initial training data $\{\mathbf{Z}, \phi(\mathbf{Z})\}$ contains $m_{\text{initial}} = 1000$ observations, where $\mathbf{Z}$ is randomly generated according to (4.11) and $\phi(\mathbf{Z})$ is approximated using Monte Carlo approximation in Equation (4.10). The ensemble is trained for 100 epochs, with a batch size of 25, and a learning rate of 0.01 for each layer using mean squared error loss and sigmoid activation functions. The ensemble is also re-trained after every $m_{\text{update}} = 500$ iterations as more data is collected during the search. Overall, the use of the surrogate model allows us to examine more possible solutions $\mathbf{Z}$ while restricting the number of intensive Monte Carlo evaluations of $\phi(\mathbf{Z})$.

***Bayesian Optimization via A Local Search with A Surrogate Model***

| | |
|---|---|
| Step 1. | Randomly generate a collection of $\mathbf{Z}^{(1)}, ..., \mathbf{Z}^{(m_{\text{initial}})}$, evaluate them according to the design criterion $\phi(\cdot)$. Train the surrogate model using these data. |
| Step 2. | Set $t = m_{\text{initial}}$ and choose $\mathbf{Z}^{(m_{\text{initial}})}$ to be the design with the lowest value of $\phi(\cdot)$ in the initial data. Let $\mathbf{Z}^* = \mathbf{Z}^{(t)}$. |
| Step 3. | Randomly generate $m_{\text{neighbor}}$ neighbors of $\mathbf{Z}^{(t)}$ and evaluate them using the surrogate model. Select $\mathbf{Z}^{(t+1)}$ as the neighbor that minimizes the acquisition function $h_{\text{acquisition}}(\cdot)$. |
| Step 4. | If $\phi(\mathbf{Z}^{(t+1)}) < \phi(\mathbf{Z}^*)$, then set $\mathbf{Z}^* = \mathbf{Z}^{(t+1)}$. Update the surrogate model every $m_{\text{update}}$ iterations. |
| Step 5. | Stop if the stopping criterion is met and return $\mathbf{Z}^*$. Otherwise set $t = t + 1$ and go to Step 3. |

Table 4.4: Our implementation of deep surrogate Bayesian optimization. In our simulations, we use $m_{\text{neighbor}} = 100$.

#### 4.2.2.2 Reinforcement Learning

Instead of searching within the neighbors of the current solution as in the above local search approach, we can train another neural network to suggest the next designs to explore. This is called a *policy network*, which is used to improve the policy (i.e., distribution) of actions in reinforcement learning (Li, 2017). To use such techniques, we formalize our optimization problem into a reinforcement learning problem. Suppose at iteration $t$, we are considering the current design $\mathbf{Z}^{(t)}$. An action $\boldsymbol{\alpha}_t \in \{0, 1\}^n$ drawn from a policy distribution $\pi(\boldsymbol{\alpha}_t|\mathbf{Z}^{(t)})$ modifies $\mathbf{Z}^{(t)}$ into $\mathbf{Z}^{(t+1)}$ as follows:

$$Z_i^{(t+1)} = \begin{cases} Z_i^{(t)} & \text{if } \alpha_{i,t} = 0 \\ 1 - Z_i^{(t)} & \text{if } \alpha_{i,t} = 1 \end{cases}.$$

By moving from $\mathbf{Z}^{(t)}$ to $\mathbf{Z}^{(t+1)}$, the action $\boldsymbol{\alpha}_t$ is paid a reward $R_t$, which is defined as the decrease in value of the acquisition function

$$R_t = h_{\text{acquisition}}(\mathbf{Z}^{(t)}) - h_{\text{acquisition}}(\mathbf{Z}^{(t+1)}),$$

where $h_{\text{acquisition}}$ is obtained from a surrogate model as in Section 4.2.2.1. Note that larger rewards are desired. The motivation behind the use of a surrogate model is to avoid evaluating $\phi(\cdot)$ many times while training the policy network.

The policy network takes the current design $\mathbf{Z}^{(t)}$ as an input and outputs the probabilities $\pi_{i,t} = \mathbb{P}(\alpha_{i,t} = 1|\mathbf{Z}^{(t)})$. We train the policy network using the REINFORCE algorithm (Williams, 1992) where weights $\boldsymbol{\omega}$ of the policy network are

updated as

$$\boldsymbol{\omega}^{(t+1)} = \boldsymbol{\omega}^{(t)} - \text{learning rate} \times R_t \nabla_{\boldsymbol{\omega}} \log \pi(\boldsymbol{\alpha}_t | \mathbf{Z}^{(t)}; \boldsymbol{\omega}^{(t)}). \tag{4.13}$$

As each $\alpha_{i,t}$ can take the value 0 or 1, we model the policy distribution to be an independent multivariate Bernoulli distribution with

$$\pi(\boldsymbol{\alpha}_t | \mathbf{Z}^{(t)}) = \prod_{i=1}^{n} \pi_{i,t} = \prod_{i=1}^{n} \mathbb{P}(\alpha_{i,t} = 1 | \mathbf{Z}^{(t)}).$$

In our implementation, we choose a neural network with two hidden layers (32 and 8 nodes) to train our policy network. Again, we use sigmoid activation functions and learning rates of 0.01 for each layer. We follow Swersky et al. (2020) to take a population approach, that is, in each iteration, the policy network is trained using a population of $m_{\text{pop}} = 100$ designs for $m_{\text{epoch}} = 100$ epochs. The best $m_{\text{pop}}$ designs created during the training are chosen to move on to the next iteration. The surrogate model is trained similarly as in Section 4.2.2.1. However, we only use 30 epochs for surrogate training to make the running time reasonable. Details of the algorithm are given in Table 4.5.

***Bayesian Optimization with Reinforcement Learning***

| | |
|---|---|
| Step 1. | Randomly generate a population of $\tilde{\mathbf{Z}}^{(0,1)}, ..., \tilde{\mathbf{Z}}^{(0,m_{\text{initial}})}$ and evaluate them according to the design criterion $\phi(\cdot)$. Train the surrogate model using these data. Set $t = 0$ and $\mathbf{Z}^*$ as the best solution among the initial population. |
| Step 2. | Rearrange the population in Step 1 and choose the best $m_{\text{pop}}$ designs: $\mathcal{Z}^{(0)} = \{\mathbf{Z}^{(0,1)}, ..., \mathbf{Z}^{(0,m_{\text{pop}})}\}$. Initialize the policy network. |
| Step 3. | For each $m$ of $m_{\text{epoch}}$ epochs: Feed each of $\mathbf{Z}^{(t,j)}$ $(j = 1, ..., m_{\text{pop}})$ into the policy network and obtain $\mathbf{Z}^{(t,j,m)}$ by sampling the action $\boldsymbol{\alpha}_{t,j,m} \sim \pi(\cdot \vert \mathbf{Z}^{(t,j)})$. Update the policy network using the REINFORCE algorithm in (4.13). |
| Step 4. | Among $\mathbf{Z}^{(t,j,m)}$ $(j = 1, ..., m_{\text{pop}}$ and $m = 1, ..., m_{\text{epoch}})$, choose the best $m_{\text{pop}}$ designs (according to the acquisition function) into the next population $\mathcal{Z}^{(t+1)} = \{\mathbf{Z}^{(t+1,1)}, ..., \mathbf{Z}^{(t+1,m_{\text{pop}})}\}$. Evaluate the design criteria $\phi(\cdot)$ for each of these designs. |
| Step 5. | If $\phi(\mathbf{Z}^{(t+1)}) < \phi(\mathbf{Z}^*)$, then set $\mathbf{Z}^* = \mathbf{Z}^{(t+1)}$. Update the surrogate model every $m_{\text{update}}$ iterations. |
| Step 6. | Stop if the stopping criterion is met and return $\mathbf{Z}^*$. Otherwise set $t = t + 1$ and go to Step 3. |

Table 4.5: Our implementation of deep reinforcement learning.

#### 4.2.2.3 Tree-Parzen Estimator

The Tree-Parzen estimator (Bergstra et al., 2011) models the conditional distribution $\mathbb{P}(\mathbf{Z}|\phi(\mathbf{Z}) < u^*)$, where $u^*$ is the $\gamma$-quantile of $\phi(\mathbf{Z})$, i.e.,

$$\mathbb{P}(\phi(\mathbf{Z}) < u^*) = \gamma,$$

and $\gamma$ is pre-specified. The algorithm then uses this conditional distribution to identify designs that have high probabilities of having small values of the design criteria. In particular, candidate designs $\mathbf{Z}$ are generated based on the conditional distribution of $\mathbf{Z}|\phi(\mathbf{Z}) < u^*$. The design that has the highest values of expected improvement is chosen to be evaluated by calculating $\phi(\cdot)$. Recall that we do not want to evaluate $\phi(\cdot)$ many times but still want to explore more designs.

For a specific design $\mathbf{Z}$, the expected improvement in terms of design criterion is

$$\int_{-\infty}^{u^*} (u^* - u)\mathbb{P}(\phi(\mathbf{Z}) = u|\mathbf{Z})du \quad \propto \quad \left(\gamma + \frac{\mathbb{P}(\mathbf{Z}|\phi(\mathbf{Z}) \geq u^*)}{\mathbb{P}(\mathbf{Z}|\phi(\mathbf{Z}) < u^*)}(1-\gamma)\right)^{-1}.$$

Thus, the design that maximizes the expected improvement is the design that maximizes $\mathbb{P}(\mathbf{Z}|\phi(\mathbf{Z}) < u^*)/\mathbb{P}(\mathbf{Z}|\phi(\mathbf{Z}) \geq u^*)$. For our problem, as $\mathbf{Z} \in \{0,1\}^n$, we use independent Bernoulli distributions to model and estimate the conditional distributions $\mathbf{Z}|\phi(\mathbf{Z}) < u^*$ and $\mathbf{Z}|\phi(\mathbf{Z}) < u^*$, that is,

$$\mathbb{P}(\mathbf{Z}|\phi(\mathbf{Z}) < u^*) = \prod_{i=1}^{n} \mathbb{P}(Z_i|\phi(\mathbf{Z}) < u^*). \tag{4.14}$$

These conditional distributions are estimated and updated as the search progresses and more design evaluations $\{\mathbf{Z}, \phi(\mathbf{Z})\}$ are collected. The $\gamma$-quantile $u^*$ is also updated accordingly. Our implementation of the algorithm is summarized in Table 4.6. We set $\gamma = 0.15$ following Bergstra et al. (2011).

### 4.2.3 Graph-cluster Randomization

In the literature, there have been many attempts to characterize, in terms of graphical structure, a generally good design for experiments on networks without specifying a model. Parker et al. (2018) utilize graph symmetry to find units that have similar connection structures for a matching design. Jagadeesan et al. (2020) take a similar approach, where graph coloring is leveraged to conduct a matching design. Basse and

***Tree-Parzen Estimator***

| | |
|---|---|
| Step 1. | Randomly generate a collection of $\mathbf{Z}^{(1)}, ..., \mathbf{Z}^{(m_{\text{initial}})}$, evaluate them according to the design criterion $\phi(\cdot)$ to obtain the initial list of designs and corresponding design criteria $\{\mathbf{Z}^{(i)}, \phi(\mathbf{Z}^{(i)})\}_{i=1}^{m_{\text{initial}}}$. Set $t = m_{\text{initial}}$. |
| Step 2. | Let $u^*$ be the $\gamma$-quantile of the observed $\phi(\mathbf{Z})$ values and let $\mathbf{Z}^*$ be the best design in this initial collection. |
| Step 3. | Calculate the probabilities $\mathbb{P}(Z_i = 1 \vert \phi(\mathbf{Z}) < u^*)$ and $\mathbb{P}(Z_i = 1 \vert \phi(\mathbf{Z}) \geq u^*)$ from the current design-criteria list. |
| Step 4. | Generate $m_{\text{candidate}}$ candidates $\mathbf{Z} \sim \mathbb{P}(\mathbf{Z} \vert \phi(\mathbf{Z}) < u^*)$ using (4.14). Let $\mathbf{Z}^{(t+1)}$ to be the one that maximizes $\mathbb{P}(\mathbf{Z} \vert \phi(\mathbf{Z}) < u^*)/\mathbb{P}(\mathbf{Z} \vert \phi(\mathbf{Z}) \geq u^*)$. Add $(\mathbf{Z}^{(t+1)}, \phi(\mathbf{Z}^{(t+1)}))$ into the current design-criteria list. |
| Step 5. | Update $u^*$ and $\mathbf{Z}^*$. |
| Step 6. | Stop if the stopping criterion is met and return $\mathbf{Z}^*$. Otherwise set $t = t + 1$ and go to Step 3. |

Table 4.6: Our implementation of the Tree-Parzen estimator. We choose $m_{\text{initial}} = 1000$ and $m_{\text{candidate}} = 100$ for our simulations.

Airoldi (2018) observe that the best experimental designs for their model are the ones that "assign units with shared neighbors to different treatment groups, and avoid the assignment of entire clusters of units that are densely connected to either treatment or control". On the other hand, *graph cluster randomization* assigns the same treatment to closely connected clusters of units (Ugander et al., 2013; Eckles et al., 2016; Shalita et al., 2016). Eckles et al. (2016) show that graph cluster randomization is able to reduce bias in estimating the global treatment effect. Therefore, graph cluster randomization has become popular in industry when experimenting on networks (Gui et al., 2015; Saveski et al., 2017; Karrer et al., 2021). In Section 4.3, we will examine the conflicting views of Basse and Airoldi (2018) and Eckles et al. (2016) by investigating graph cluster randomization and other design approaches over a collection of response models for network experiments.

As discussed in Section 1.2.3, graph cluster randomization first partitions the network into clusters that have many connections within and much fewer connections between clusters. Given such a clustering, each cluster and thus all of its units are randomly assigned to either treatment or control. Graph cluster randomization limits potential interference by ensuring each unit has the same treatment assignment as (most of) its neighbors. This simulates universes (Ugander et al., 2013) where the whole graph is assigned to either treatment or control. Thus, graph cluster randomization is designed to reduce bias for estimating the GTE (Eckles et al., 2016).

Implementations of graph cluster randomization differ in terms of the clustering algorithm which assigns cluster labels $c_i$ to each node $i$ of the network, $1 \leq i \leq n$. The goodness of a clustering $\boldsymbol{c} = (c_1, c_2, ..., c_n)^\top$ is usually evaluated using *modularity* (Newman, 2006)

$$Q(\boldsymbol{c}) = \frac{1}{2m} \sum_{i,j} \left( A_{ij} - \frac{K_{ii} K_{jj}}{2m} \right) \mathbb{I}\{c_i = c_j\},$$

where $K_{ii} = \sum_{j=1}^{n} A_{ij}$ is the number of connections (*degree*) of node $i$ and $m$ is the number of edges in the network. Modularity measures the difference between the observed number of edges within clusters assigned by $\boldsymbol{c}$ and the expected number of such edges where the expectation is with respect to the degree distribution of the graph (Newman, 2006). The higher the value of $Q$, the better the clustering $\boldsymbol{c}$ in terms of grouping densely connected nodes together. Many clustering algorithms exist. We consider balanced label propagation and the Louvain method, which are described below.

Gui et al. (2015) argue that in addition to high modularity, a good clustering algorithm needs to produce clusters of similar sizes (i.e., *balanced*) so that the difference-in-means estimator (3.7) for the GTE has a small variance. They suggest a balanced label propagation algorithm, in which the first step is to randomly assign the cluster labels to nodes so that the cluster sizes are balanced. After that, the following two steps are alternated until convergence. The first step is label propagation, in which we iteratively check every pair of nodes and switch their labels if that increases the modularity. Label propagation continues until the modularity can no longer increase. The other step is random shuffling, where we randomly choose $100\alpha_{\text{shuffle}}\%$ of pairs and switch their labels. The random shuffling step is designed to help the algorithm break away from locally optimal solutions. Following Gui et al. (2015), we choose $\alpha_{\text{shuffle}} = 0.05$ in our implementation. There are other balanced graph clustering algorithms proposed in the literature, such as the social hash algorithm (Shalita et al., 2016; Karrer et al., 2021) and the restreaming linear deterministic greedy algorithm (Saveski et al., 2017), but we use the balanced label propagation algorithm because it produces clusterings with higher modularities (see Appendix C.1).

On the other hand, Karrer et al. (2021) observe that imbalanced clusters are better when regression adjustment is applied in the analysis. They suggest using the Louvain algorithm (Blondel et al., 2008), which is a fast and scalable algorithm that is able to produce a possibly imbalanced clustering but with high modularity. The algorithm starts by assigning each node to its own cluster. In each iteration, a node is moved into a cluster of its neighbors so that the move will contribute the highest gain

to modularity. Each cluster is then considered to be a new node and the algorithm continues until the modularity cannot be increased anymore. Our implementation of the Louvain algorithm is from the `igraph` package in `R`.

Note that balanced label propagation requires the number of clusters to be specified while the Louvain algorithm does not. In our implementation, we use the number of clusters determined by the Louvain algorithm as input to balanced label propagation. After a clustering is obtained, we randomly assign half of the clusters to treatment and the other half to control, yielding a graph cluster randomized design.

Besides the clustering-based design construction algorithms discussed above, in Section 4.3, we also consider balanced randomization, where we randomly assign half of the units to treatment and the other half to control. This is the default design construction algorithm for A/B tests with independent units and is thus a relevant comparator. In our simulations, we use the designs produced by balanced randomization as a baseline to investigate the relative efficiencies of the (optimal) designs produced by other algorithms.

## 4.3 Simulations

### 4.3.1 Response-Generating Models

To investigate the performance of the design construction algorithms discussed in Section 4.2, we consider a variety of models with different mean functions, error structures, and response types. Below we list the models that we consider in this chapter and discuss our choice of design criterion and prior specification for each model.

**The Conditional Network Autoregressive (CNAR) Model:** In Example 4.3, we discuss the CNAR model (4.6), in which parameter $\rho$ governs the correlation structure among the experimental units on the network. Since the weighted least squares estimator (4.7) is unbiased, we can consider $\text{MSE}(\widehat{\text{GTE}}) = \text{Var}(\widehat{\text{GTE}}) = \text{Var}(\hat{\tau})$, the (2,2) element of $\text{Var}(\hat{\boldsymbol{\beta}})$ in (4.8), as our target design criterion (i.e. $\phi_0$ in (4.9)). While Pokhilko et al. (2019) and Zhang and Kang (2022) construct optimal designs for known values of $\rho$, we aim to construct optimal designs with respect to a prior distribution of $\rho$ with the Bayesian design criterion (4.9). Since $0 \leq \rho < 1$, we assume a noninformative, uniform prior for $\rho$ in the range $[0, 1)$.

**The Normal Sum (NS) Model:** As discussed in Section 4.2.3, Basse and Airoldi (2018) find that designs with a clustering structure such as those generated by graph-cluster randomization are not desirable for their model. To verify this, we consider that model in our simulations. In particular, Basse and Airoldi (2018) posit a hierarchical model where the network correlation is induced by "intrinsic latent variables" $\boldsymbol{X} = (X_1, ..., X_n)^\top$, in which

$$\begin{aligned} X_j &\sim \mathcal{N}(\mu, \sigma^2) \\ Y_i(0) \mid \boldsymbol{X} &\sim \mathcal{N}\big(X_i + \sum_{j=1}^{n} A_{ij} X_j, \gamma^2\big) \\ Y_i &= Y_i(0) + \tau Z_i, \end{aligned} \tag{4.15}$$

where $Y_i(0)$ is the potential outcome when unit $i$ is assigned to control. Basse and Airoldi (2018) argue that the model arises naturally with an example where the response of interest $Y_i$ is the time user $i$ spends on a social media platform. Then $X_i$ can be thought of as the "intrinsic propensity of user $i$ to spend time on the website". From the structure of Model (4.15), the GTE in this model is $\tau$ and Basse and Airoldi (2018) estimate it using the difference-in-means estimator

$$\hat{\tau} = \frac{1}{n_1} \sum_{i=1}^{n} Z_i Y_i - \frac{1}{n_0} \sum_{i=1}^{n} (1 - Z_i) Y_i, \tag{4.16}$$

where $n_1$ and $n_0$ are the number of treated and controlled units, respectively. Under the assumed model, the estimator is biased, and the corresponding mean squared error can be derived as (Basse and Airoldi, 2018)

$$\text{MSE}(\hat{\tau}) = \mu^2 \left( \frac{1}{n_1} \sum_{i=1}^{n} Z_i K_{ii} - \frac{1}{n_0} \sum_{i=0}^{n} (1 - Z_i) K_{ii} \right)^2 + \gamma^2 \boldsymbol{\omega}^\top \boldsymbol{\omega} + \sigma^2 \boldsymbol{\omega}^\top \mathbf{A}^\top \mathbf{A} \boldsymbol{\omega}. \tag{4.17}$$

The vector $\boldsymbol{\omega} = (\omega_1, ..., \omega_n)^\top$ contains elements $\omega_i$ such that $\omega_i = \frac{1}{n_1}$ if unit $i$ is treated and $\omega_i = -\frac{1}{n_0}$ if unit $i$ is assigned to control. We can see that the MSE formula in (4.17) depends on unknown parameters $\mu$, $\gamma$ and $\sigma$. In the simulations, we posit priors on these unknown parameters. We choose a wide normal distribution centered at zero as the prior for $\mu$ and inverse gamma priors for $\gamma$ and $\sigma$. Specific details are given in Table 4.7.

**The Power-Degree (POW-DEG) Model:** In Example 4.1, we discuss that for the POW-DEG model (2.9), since the maximum likelihood estimator is asymptotically unbiased, we can consider using its asymptotic variance as our target design

criterion $\phi_0$. However, it is clear from (4.4) that the variance depends on model parameters. Thus, the Bayesian optimal criterion with specified priors can be used in the design stage when the parameters are unknown. For our simulations, we choose a set of priors for the regression coefficients similar to that of the NS model. As discussed in Section 2.2.2.3, since a sublinear growth is expected which renders $0 < \lambda \leq 1$, we set $\lambda \sim \text{Uniform}(0, 1]$. Specific details about our prior specification are summarized in Table 4.7.

**The Binary Network-Temporal Autoregressive (BNTAR) Model:** Finally, we consider the BNTAR model in Example 4.2. Following Eckles et al. (2016), we consider the difference-in-means estimator (4.16) for the GTE. Hence, our target design criterion $\phi_0$ will be the MSE of the difference-in-means estimator. However, due to the complex structure of Model (4.5), the design criterion cannot be derived analytically. Thus, we need to estimate it using Monte Carlo approximation. In particular, for a given design $\mathbf{Z}$, we generate the responses $Y_{i,T}$ in (4.5) 5,000 times, each run with different parameter values drawn from the prior distribution and different set of errors drawn from the standard normal distribution. A difference-in-means estimate (4.16) is computed for each run. The MSE of the estimator under the given design $\mathbf{Z}$ is then calculated with respect to the true GTE. The true GTE is estimated as the difference in average responses (over the 5,000 runs) when all units are assigned to treatment ($\mathbf{Z} = \mathbf{1}_n$) versus when all units are assigned to control ($\mathbf{Z} = \mathbf{0}_n$). Following Eckles et al. (2016), we set $\mu = -1.5$, $T = 3$, and $Y_{i,0} = 0$ for all $i \leq 1 \leq n$. For $\tau$ and $\gamma$, we impose uniform priors on the values considered in Eckles et al. (2016). Specific details are provided in Table 4.7.

| Models | Priors |
|---|---|
| CNAR Model (4.6) | $\rho \sim \text{Uniform}[0, 1)$ |
| NS Model (4.15) | $\mu \sim \mathcal{N}(0, 10^2)$<br>$\sigma^{-2}, \gamma^{-2} \sim \text{Gamma}(1, 1)$ |
| POW-DEG Model (2.9) | $\mu, \tau, \gamma_T, \gamma_C \sim \mathcal{N}(0, 10^2)$<br>$\lambda \sim \text{Uniform}(0, 1]$<br>$\sigma^{-2} \sim \text{Gamma}(1, 1)$ |
| BNTAR Model (4.5) | $\tau, \gamma \sim \text{Uniform}[0, 1]$ |

Table 4.7: Parameter values and priors for models used in our simulations.

### 4.3.2 Other Simulation Details

Both meta-heuristic and Bayesian optimization algorithms do not guarantee convergence, thus they require a stopping criterion in order to control the running time. There have been many stopping criteria proposed in the literature (Ribeiro et al., 2011; Ghoreishi et al., 2017; Dai et al., 2019; Makarova et al., 2022). However, which is best, both generally and specifically for each algorithm, has remained an open problem. In our case, in order to make a fair comparison of the algorithms, we will stop after a fixed number of design evaluations $\phi(\cdot)$. Specifically, the algorithms will stop after 5,000 design evaluations, where each evaluation uses a Monte Carlo approximation (4.10) based on $L = 5,000$ parameter draws. This value of $L$ is determined empirically as the design criteria considered in this article generally converge by $5,000$ parameter draws (see Appendix C.2).

To study the behaviors of the different response-generating models and design algorithms on realistic network structures, in our simulations, we use the Enron, Caltech, and UMichigan networks whose summary statistics are given in Table 1.1. Due to the stochastic nature of the design construction algorithms considered, for each scenario (network and model), we run each of the algorithms 30 times (to generate 30 approximately optimal designs). We evaluate the designs found by each algorithm in terms of efficiency compared to the naive balanced randomization (where half of the network is randomly selected and assigned to treatment and the other half is assigned to control). This enables us to make a fair comparison across designs and models. In particular, for each model and method, the efficiency of a design is calculated as

$$\Delta(\mathbf{Z}) = \frac{\bar{\phi}_{\text{balanced randomization}} - \hat{\phi}(\mathbf{Z})}{\bar{\phi}_{\text{balanced randomization}}} \times 100\%. \tag{4.18}$$

### 4.3.3 Results

#### 4.3.3.1 Performance of Design Construction Algorithms

The performance of the algorithms described in Section 4.2 for each of the models and networks are presented in Figure 4.1. Different rows of the grid show results for different models, and different columns show results for different networks. In each panel of the grid, a boxplot shows the distribution of efficiencies, given by (4.18), of the 30 "best" designs found by a particular algorithm. The vertical dashed line at 0% serves as a reference. Designs lying on the left-hand side of this reference line

perform worse than balanced randomization, and designs lying on the right-hand side perform better. Given the size of the UMichigan network, we only show its results for graph-cluster randomization, balanced randomization, genetic algorithm, and the Tree-Parzen estimator due to infeasibly long running times for the other algorithms.

The first row of the grid shows the results for the CNAR model (4.6). We can see that graph cluster randomization strategies, balanced or imbalanced, perform much worse than any other algorithms. This indicates that this popular design algorithm does not work for all response-generating models. Other algorithms perform slightly better than balanced randomization, with meta-heuristic algorithms performing better than Bayesian optimization algorithms.

Based on the MSE formula in (4.17), Basse and Airoldi (2018) point out that good designs for the NS model (4.15) should not assign the same treatment assignment to the entire cluster. Our results in the second row of Figure 4.1 corroborate this statement; the graph cluster randomization performs much worse than other design-finding methods. In fact, the efficiency loss can be up to more than 1000%. This is mainly attributed to the bias which scales with $\mu$ in (4.17), and the wide normal prior given to (unknown) $\mu$. In addition to graph cluster randomization, simple balanced randomization also occasionally produces very bad designs, causing the average design criteria $\bar{\phi}_{\text{balanced randomization}}$ (4.18) to be very large. As a result, designs produced by meta-heuristic and Bayesian optimization algorithms attain almost 100% efficiency gain.

In the third row of Figure 4.1, we present the results for the POW-DEG model (2.9). Unlike the previous two models, the POW-DEG model prefers graph cluster randomization. Such designs are the best designs for the Caltech and UMichigan networks, though the meta-heuristic algorithms and the genetic algorithm in particular tend not to be far behind.

Lastly, we take a look at the results for the BNTAR model (4.5) in the bottom row of Figure 4.1. Eckles et al. (2016) use the BNTAR model with fixed (known) parameters to show the effectiveness of graph cluster randomization. However, when the parameters are unknown and prior distributions are imposed, graph cluster randomization performs worse than even simple balanced randomization. Moreover, the efficiency gains of meta-heuristic and Bayesian optimization algorithms are so minimal that we may avoid their computational complexity and opt to use balanced randomization for this model.

We summarize these results with several conclusions. First, the performance of graph cluster randomization depends highly on the assumed response-generating

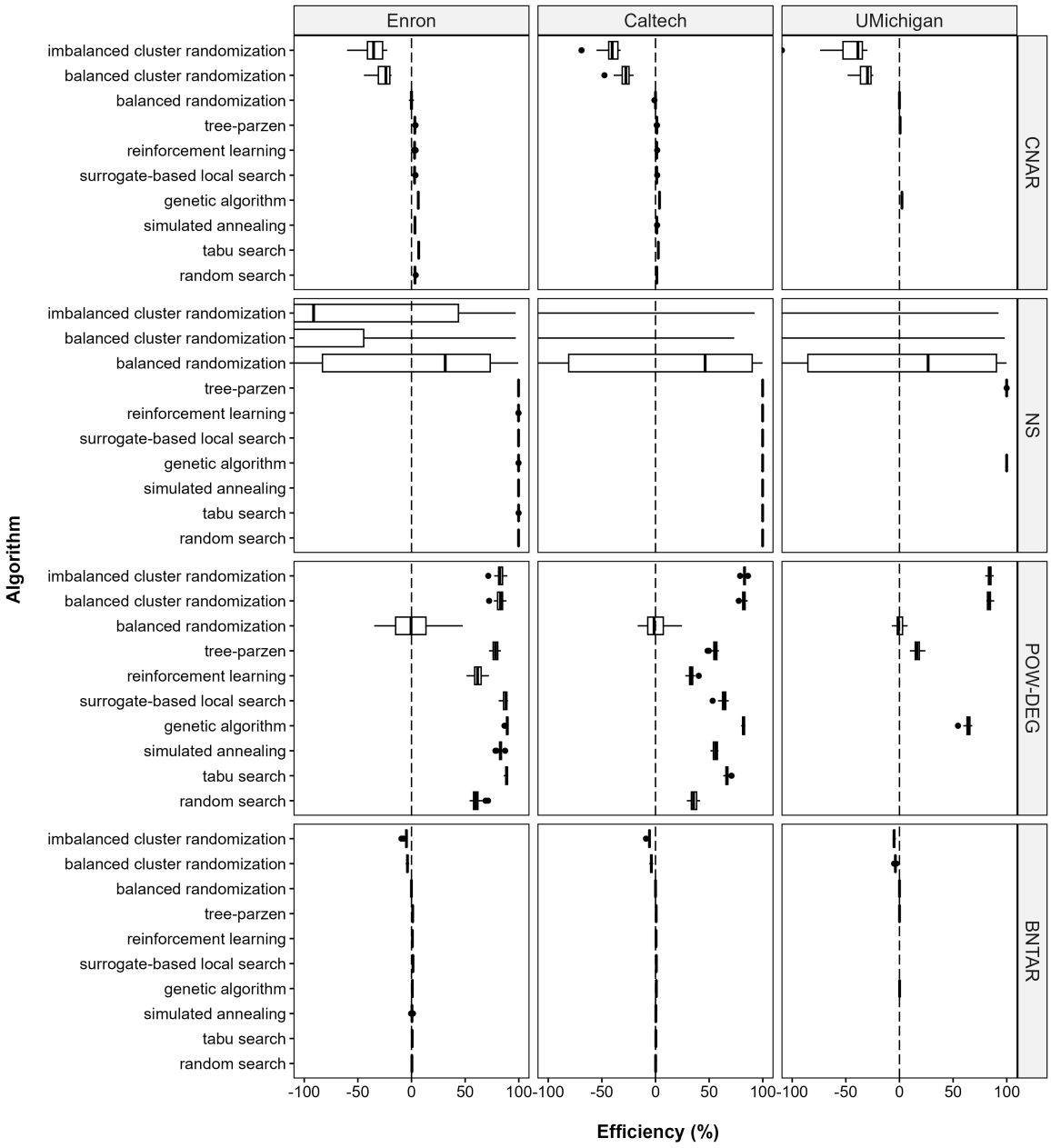



Figure 4.1: Efficiency of designs found by different algorithms under different models and networks with respect to the balanced randomization algorithm.

model. The algorithm may be the best for some models but the worst for others. In terms of balance for the graph clustering, balanced and imbalanced cluster randomization gives similar results, with balanced cluster randomization performing slightly better. Second, the performance of Bayesian optimization algorithms also varies with respect to the models. This may be attributed to how well the surrogate model can accurately represent the design criterion function $\phi(\cdot)$. Finally, the genetic algorithm seems to perform very well across all model and network settings considered. It also has the lowest running times among all meta-heuristic algorithms, which, in general, tend to be faster than the Bayesian optimization methods. See Appendix C.3 for an analysis of running times. We thus recommend the general use of the genetic algorithm for the problem of optimal Bayesian design on networks.

#### 4.3.3.2 Characteristics of Optimal Designs

To understand what makes a good design for a given model, we study the characteristics of the optimal designs generated across different algorithms for different networks and response-generating models. The distributions of these characteristics are plotted in Figure 4.2. Different rows represent different models and networks, while different columns represent different design characteristics. In each panel, each dot represents one design. By plotting the design characteristics against the efficiency, we can see how characteristics are distributed among good and bad designs.

In the first column of Figure 4.2, we compare the average degrees of treated and controlled nodes in different designs. The reference lines at 0 indicate perfect balance. We can see that balance in terms of degree is particularly important for the NS model (4.15). This is unsurprising given the first term of the MSE formula in (4.17): large difference in average degrees between treated and controlled nodes will increase the bias of the difference-in-means estimator. Degree balance also seems to be preferred for the CNAR (4.6) and the POW-DEG (2.9) models. The BNTAR model (4.5), on the other hand, seems to prefer assigning treatment to units with lower numbers of connections.

We further investigate balance in terms of *betweenness* in the second column of Figure 4.2. The betweenness of a unit measures the number of shortest paths (between pairs of other units) passing through that unit. If a unit lies on many shortest paths connecting two other units, it is considered to have high (betweenness) centrality. This is similar to a transportation hub that many routes need to pass through. We can see that betweenness balance is important across all models, especially in the NS model (4.15).

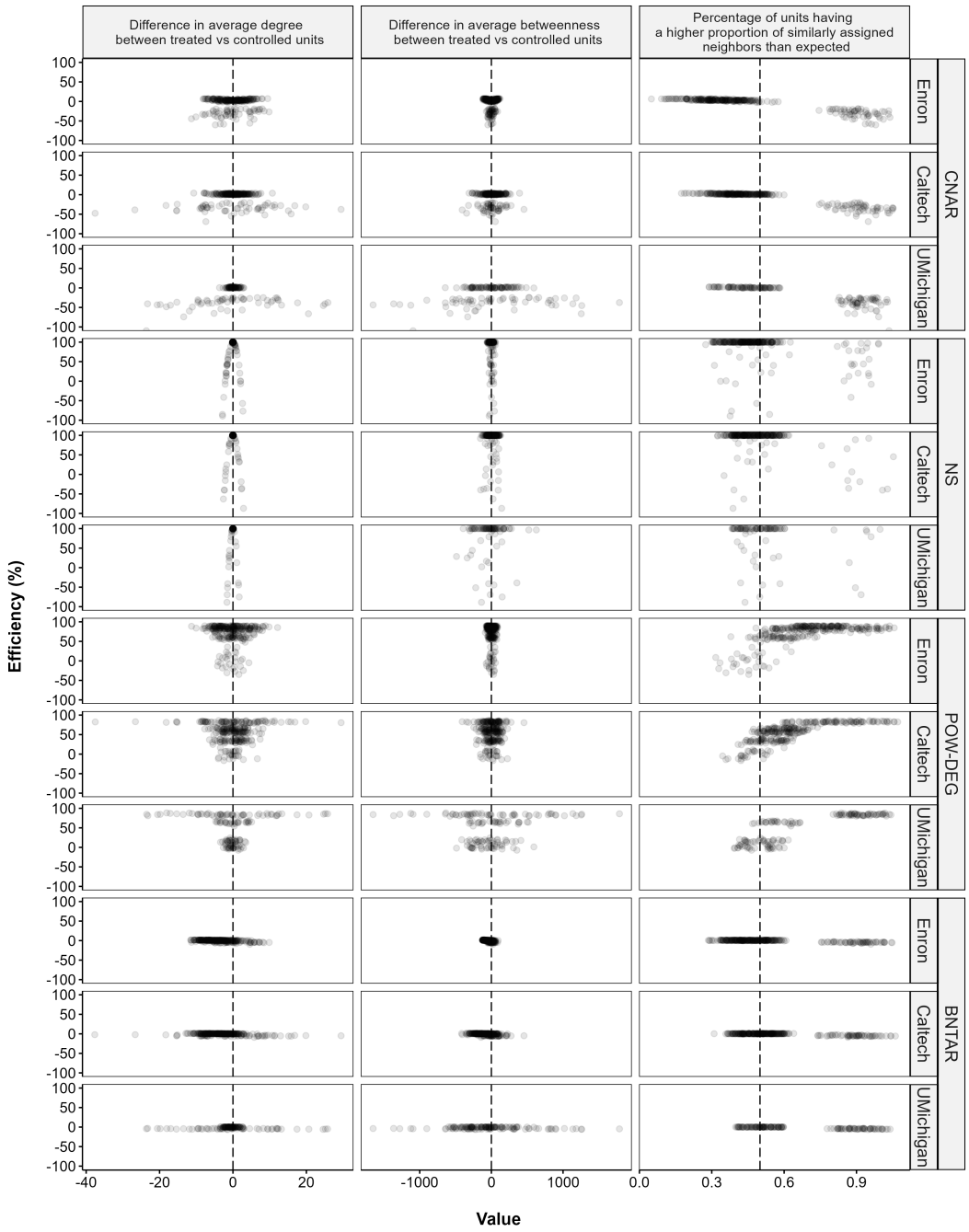



Figure 4.2: Characteristics of designs found by different algorithms for each model and network.

Finally, in the third column of Figure 4.2, we consider the clustering of the design by measuring the percentage of nodes having a higher proportion of similarly assigned neighbors than expected. If a design assigns $b\%$ of units to treatment, each unit is expected to have $b\%$ of neighbors assigned to treatment and $(100-b)\%$ of neighbors assigned to control. If a node is treated and has more than $b\%$ of neighbors assigned to treatment, or if the node is controlled and has more than $(100-b)\%$ of neighbors assigned to control, then the node tends to be surrounded by neighbors with the same treatment assignment as themselves. This is exactly what happens in graph-cluster randomization, where closely connected nodes are grouped into clusters and are given the same treatment assignment. If a design has a high percentage of nodes having a higher than expected number of similarly treated neighbors, then the design is more "clustered". We can see that the results in Figure 4.2 resonate with the results in Figure 4.1, where graph-cluster randomization is good for the POW-DEG model (2.9) while being poor for the CNAR (4.6) and the NS (4.15) models. For the BNTAR model (4.5), clustered and unclustered designs seem to perform similarly.

In addition to these three characteristics, we also considered the percentage of treated nodes (i.e., whether the design is balanced or not), and balance with respect to *closeness* centrality. The results are shown in Appendix C.4, but we do not see any notable patterns. We find that most designs, both good and bad, are balanced in these two measures, implying that balance is important, but these particular characteristics do not distinguish good designs from bad.

## 4.4 Conclusions

In this chapter, we formulate the problem of designing experiments on networks as an optimization problem, where the design criterion is the mean squared error of a GTE estimator averaged over the prior distributions of unknown parameters of the postulated model. By considering such a Bayesian design criterion, we mitigate the problem of unknown parameters in the design criteria formula, thus enhancing the robustness of model-based optimal design approaches. Since most Bayesian design criteria do not have closed-form formulas and need to be approximated using Monte Carlo simulations, classical optimal design algorithms become infeasible, and other general optimization algorithms need to be used. We adapted and investigated meta-heuristic, Bayesian optimization, and graph-based design construction algorithms. The effectiveness of these algorithms was evaluated using simulations across different response-generating models and realistic network structures. Although we consider

only the variance or MSE of the GTE estimator as the target design criterion in our simulations, our methods can be easily generalized to any other target design criteria.

In terms of the design construction algorithms, we find that the genetic algorithm works reasonably well across all models and networks considered in this chapter while being among the most computationally efficient. We thus recommend using the genetic algorithm for optimal design problems on networks. In terms of design characteristics, we find that balance between treatment and control groups, with respect to various graphical characteristics, is preferred across almost all settings. Thus, balanced designs in node characteristics may be generally favorable for network experimentation. Finally, we find that graph cluster randomization, although highly advocated in the literature, only performs well for some of the models considered. In others, it can cause substantial efficiency loss. This emphasizes that the goodness of a design depends on the intended analysis method. Although our results may be impacted by the choice of hyperparameters in the algorithm implementations, we choose hyperparameters according to the literature and obtain consistent results across different scenarios. This strengthens the generalizability of our conclusions.

There are some aspects of optimal Bayesian design on networks that can be further improved and investigated. First, the algorithms' performances may be further improved by employing more efficient coding, choosing different training distributions, and fine-tuning the hyper-parameters. Moreover, although we only consider individual algorithms in this article, it may be valuable to investigate combinations of these algorithms to see whether they can improve performance. Finally, our results show that the scalability of the optimization algorithms highly depends on the scalability of the evaluation of design criteria. This can be a topic for future research.

# Chapter 5

# Conclusion and Future Research

This thesis considers the problem of design and analysis of experiments when the experimental units are connected on a network in which case interference via network connections is common. This is an important problem because in many cases, such as certain agricultural, clinical, or social experiments, the classical independence assumption is not satisfied. Moreover, rather than an obstacle, interference may also be the subject of interest for applied researchers.

In particular, we focus on the model-based approach in which a model is postulated for the outcomes, and the design is constructed to optimize the efficiency of the analysis based on the model. Both design and analysis of experiments on networks are considered with a focus on enhancing flexibility, interpretability, and robustness. In particular, Chapter 2 expands the modeling possibilities by introducing a general class of parametric network effect specifications. Based on the additive structure of the model, we propose a unified framework for causal interpretation and derive the estimation and inference procedure for a family of specifications. Chapter 3 extends the framework in Chapter 2 to experiments with binary outcomes. We apply such a binary model to a real-world experiment and illustrate how our framework can be useful for the analysis and interpretation of network experiments. Finally, Chapter 4 improves the robustness of model-based design by incorporating prior information of unknown parameters into the design criterion. We formulate the optimal design construction problem into a discrete optimization where the objective function is computationally intensive. We approach this problem by adapting meta-heuristic and Bayesian optimization techniques. Overall, the thesis presents a systematic investigation of the problem of network experimentation in general and the model-based approach in particular. We provide practitioners with useful overviews and

insights, and we develop helpful tools that can be readily applied to the design and analysis of network A/B tests in practice.

Nevertheless, network experimentation is still a burgeoning research area with many possible extensions and open problems. First, since most current research focuses on network A/B testing, future attention can be paid to experiments with one or multiple factors at multiple levels. This is an important yet difficult problem since it is hard to quantify the network effect coming from each treatment on a given network with a fixed structure. Second, in practice, many experiments are conducted over an extended period of time, which may add uncertainty and temporal confounding to the experimental outcomes. Thus, another promising area is to incorporate, control, and leverage the time element of the experiment. Third, experiments are often conducted on a sampled network. Sampling the experimental network and generalizing the experimental results to the population network is also an important topic for future research. Finally, research on network experimentation does not only apply to network-correlated data. By properly defining the network relationship, they may be applied to cluster-correlated or spatially correlated experimental data. Causal inference on network-correlated observational data is also a related area.

# References


Abbe, E. (2017). Community detection and stochastic block models: Recent developments. *The Journal of Machine Learning Research*, 18(1):6446–6531.

Advani, A. and Malde, B. (2018). Methods to identify linear network models: A review. *Swiss Journal of Economics and Statistics*, 154(1):12.

Akaike, H. (1998). Information theory and an extension of the maximum likelihood principle. In *Selected papers of Hirotugu Akaike*, pages 199–213. Springer.

Anselin, L. (2002). Under the hood. Issues in the specification and interpretation of spatial regression models. *Agricultural Economics*, 27(3):247–267.

Aronow, P. M., Samii, C., et al. (2017). Estimating average causal effects under general interference, with application to a social network experiment. *The Annals of Applied Statistics*, 11(4):1912–1947.

Backstrom, L. and Kleinberg, J. (2011). Network bucket testing. In *Proceedings of the 20th International Conference on World Wide Web*, pages 615–624.

Bakshy, E., Eckles, D., Yan, R., and Rosenn, I. (2012a). Social influence in social advertising: Evidence from field experiments. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, pages 146–161.

Bakshy, E., Rosenn, I., Marlow, C., and Adamic, L. (2012b). The role of social networks in information diffusion. In *Proceedings of the 21st International Conference on World Wide Web*, pages 519–528.

Bapna, R. and Umyarov, A. (2015). Do your online friends make you pay? A randomized field experiment on peer influence in online social networks. *Management Science*, 61(8):1902–1920.

Basse, G. W. and Airoldi, E. M. (2018). Model-assisted design of experiments in the presence of network-correlated outcomes. *Biometrika*, 105(4):849–858.

Bergstra, J., Bardenet, R., Bengio, Y., and Kégl, B. (2011). Algorithms for hyper-parameter optimization. *Advances in Neural Information Processing Systems*, 24:2546–2554.

Besag, J. (1974). Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society: Series B (Methodological)*, 36(2):192–225.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008.

Bond, R. M., Fariss, C. J., Jones, J. J., Kramer, A. D., Marlow, C., Settle, J. E., and Fowler, J. H. (2012). A 61-million-person experiment in social influence and political mobilization. *Nature*, 489(7415):295–298.

Börner, K., Klavans, R., Patek, M., Zoss, A. M., Biberstine, J. R., Light, R. P., Larivière, V., and Boyack, K. W. (2012). Design and update of a classification system: The UCSD map of science. *PloS One*, 7(7):e39464.

Boyd, D. M. and Ellison, N. B. (2007). Social network sites: Definition, history, and scholarship. *Journal of Computer-Mediated Communication*, 13(1):210–230.

Bramoullé, Y., Djebbari, H., and Fortin, B. (2009). Identification of peer effects through social networks. *Journal of Econometrics*, 150(1):41–55.

Bramoullé, Y., Kranton, R., and D'amours, M. (2014). Strategic interaction and networks. *American Economic Review*, 104(3):898–930.

Cai, J., Janvry, A. D., and Sadoulet, E. (2015). Social networks and the decision to insure. *American Economic Journal: Applied Economics*, 7(2):81–108.

Calvó-Armengol, A., Patacchini, E., and Zenou, Y. (2009). Peer effects and social networks in education. *The Review of Economic Studies*, 76(4):1239–1267.

Cameron, A. C. and Miller, D. L. (2015). A practitioner's guide to cluster-robust inference. *Journal of Human Resources*, 50(2):317–372.

Chaloner, K. and Verdinelli, I. (1995). Bayesian experimental design: A review. *Statistical Science*, pages 273–304.

Chin, A. (2019). Regression adjustments for estimating the global treatment effect in experiments with interference. *Journal of Causal Inference*, 7(2).

Christakis, N. A. and Fowler, J. H. (2007). The spread of obesity in a large social network over 32 years. *New England Journal of Medicine*, 357(4):370–379.

Christofides, E., Muise, A., and Desmarais, S. (2009). Information disclosure and control on Facebook: Are they two sides of the same coin or two different processes? *Cyberpsychology & Behavior*, 12(3):341–345.

Cressie, N. (2015). *Statistics for spatial data*. John Wiley & Sons.

Crump, R. K., Hotz, V. J., Imbens, G. W., and Mitnik, O. A. (2009). Dealing with limited overlap in estimation of average treatment effects. *Biometrika*, 96(1):187–199.

Dai, Z., Yu, H., Low, B. K. H., and Jaillet, P. (2019). Bayesian optimization meets Bayesian optimal stopping. In *International Conference on Machine Learning*, pages 1496–1506. Proceedings of Machine Learning Research.

Doob, J. L. (1935). The limiting distributions of certain statistics. *The Annals of Mathematical Statistics*, 6(3):160–169.

Doreian, P., Batagelj, V., and Ferligoj, A. (2020). *Advances in network clustering and blockmodeling*. John Wiley & Sons.

Eckles, D., Karrer, B., and Ugander, J. (2016). Design and analysis of experiments in networks: Reducing bias from interference. *Journal of Causal Inference*, 5(1).

Ennett, S. T., Bauman, K. E., Hussong, A., Faris, R., Foshee, V. A., Cai, L., and DuRant, R. H. (2006). The peer context of adolescent substance use: Findings from social network analysis. *Journal of Research on Adolescence*, 16(2):159–186.

Fedorov, V. V. and Leonov, S. L. (2013). *Optimal design for nonlinear response models*. CRC Press.

Flassig, R. J. and Schenkendorf, R. (2018). Model-based design of experiments: Where to go. In *9th Vienna Internatioal Conference on Mathematical Modelling*, pages 875–876.

Fortunato, S. (2010). Community detection in graphs. *Physics Reports*, 486(3-5):75–174.

Fortunato, S. and Hric, D. (2016). Community detection in networks: A user guide. *Physics Reports*, 659:1–44.

Foster, A., Jankowiak, M., O'Meara, M., Teh, Y. W., and Rainforth, T. (2020). A unified stochastic gradient approach to designing Bayesian-optimal experiments. In *International Conference on Artificial Intelligence and Statistics*, pages 2959–2969. PMLR.

Franke, L., Van Bakel, H., Fokkens, L., De Jong, E. D., Egmont-Petersen, M., and Wijmenga, C. (2006). Reconstruction of a functional human gene network, with an application for prioritizing positional candidate genes. *The American Journal of Human Genetics*, 78(6):1011–1025.

Freedman, D. A. (2006). On the so-called "Huber sandwich estimator" and "robust standard errors". *The American Statistician*, pages 299–302.

García-Ródenas, R., García-García, J. C., López-Fidalgo, J., Martín-Baos, J. Á., and Wong, W. K. (2020). A comparison of general-purpose optimization algorithms for finding optimal approximate experimental designs. *Computational Statistics & Data Analysis*, 144:106844.

Garnett, R. (2023). *Bayesian optimization*. Cambridge University Press.

Gendreau, M., Potvin, J.-Y., et al. (2010). *Handbook of metaheuristics*, volume 2. Springer.

Ghoreishi, S. N., Clausen, A., and Jørgensen, B. N. (2017). Termination criteria in evolutionary algorithms: A survey. In *International Joint Conference on Computational Intelligence*, pages 373–384.

Goel, V. (2014). Facebook tinkers with users' emotions in news feed experiment, stirring outcry. *The New York Times*, 29.

Gossen, H. H. (1983). *The laws of human relations and the rules of human action derived therefrom*. MIT Press.

Gui, H., Xu, Y., Bhasin, A., and Han, J. (2015). Network A/B testing: From sampling to estimation. In *Proceedings of the 24th International Conference on World Wide Web*, pages 399–409.

Gupta, S., Kohavi, R., Tang, D., Xu, Y., Andersen, R., Bakshy, E., Cardin, N., Chandran, S., Chen, N., Coey, D., et al. (2019). Top challenges from the first practical online controlled experiments summit. *ACM SIGKDD Explorations Newsletter*, 21(1):20–35.

Hájek, J. (1971). Comment on "An essay on the logical foundations of survey sampling, Part One". *The Foundations of Survey Sampling*, 236.

Harville, D. A. (2008). *Matrix algebra from a statistician's perspective*. Springer Science & Business Media.

Hoadley, C. M., Xu, H., Lee, J. J., and Rosson, M. B. (2010). Privacy as information access and illusory control: The case of the Facebook news feed privacy outcry. *Electronic Commerce Research and Applications*, 9(1):50–60.

Holland, J. H. (1992). *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. MIT Press.

Holland-Letz, T. and Kopp-Schneider, A. (2015). Optimal experimental designs for dose-response studies with continuous endpoints. *Archives of Toxicology*, 89(11):2059–2068.

Horn, R. A. and Johnson, C. R. (2012). *Matrix analysis*. Cambridge University Press.

Horvitz, D. G. and Thompson, D. J. (1952). A generalization of sampling without replacement from a finite universe. *Journal of the American Statistical Association*, 47(260):663–685.

Hudgens, M. G. and Halloran, M. E. (2008). Toward causal inference with interference. *Journal of the American Statistical Association*, 103(482):832–842.

Jagadeesan, R., Pillai, N. S., and Volfovsky, A. (2020). Designs for estimating the treatment effect in networks with interference. *The Annals of Statistics*, 48(2):679 – 712.

Jennrich, R. I. (1969). Asymptotic properties of non-linear least squares estimators. *The Annals of Mathematical Statistics*, 40(2):633–643.

Karrer, B., Shi, L., Bhole, M., Goldman, M., Palmer, T., Gelman, C., Konutgan, M., and Sun, F. (2021). Network experimentation at scale. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data mining*, pages 3106–3116.

Kelejian, H. H. and Prucha, I. R. (2010). Specification and estimation of spatial autoregressive models with autoregressive and heteroskedastic disturbances. *Journal of Econometrics*, 157(1):53–67.

Kernighan, B. W. and Lin, S. (1970). An efficient heuristic procedure for partitioning graphs. *The Bell System Technical Journal*, 49(2):291–307.

Khan, S. and Tamer, E. (2010). Irregular identification, support conditions, and inverse weight estimation. *Econometrica*, 78(6):2021–2042.

Kirkpatrick, S., Gelatt Jr, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220(4598):671–680.

Klier, T. and McMillen, D. P. (2008). Clustering of auto supplier plants in the United States: Generalized method of moments spatial logit for large samples. *Journal of Business & Economic Statistics*, pages 460–471.

Kolaczyk, E. D. and Csárdi, G. (2014). *Statistical analysis of network data with R*, volume 65. Springer.

Koutra, V., Gilmour, S. G., and Parker, B. M. (2021). Optimal block designs for experiments on networks. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, n/a(n/a).

Krebs, V. (1996). Visualizing human networks. *Release*, 1:1–25.

La Vigne, N. G., Lowry, S. S., Markman, J. A., and Dwyer, A. M. (2011). Evaluating the use of public surveillance cameras for crime control and prevention. *Washington, DC: US Department of Justice, Office of Community Oriented Policing Services. Urban Institute, Justice Policy Center*.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6405–6416, Red Hook, NY, USA. Curran Associates Inc.

Larsen, N., Stallrich, J., Sengupta, S., Deng, A., Kohavi, R., and Stevens, N. T. (2023). Statistical challenges in online controlled experiments: A review of A/B testing methodology. *The American Statistician*, pages 1–15.

Latkin, C., Mandell, W., Oziemkowska, M., Celentano, D., Vlahov, D., Ensminger, M., and Knowlton, A. (1995). Using social network analysis to study patterns of drug use among urban drug users at high risk for HIV/AIDS. *Drug and Alcohol Dependence*, 38(1):1–9.

Lee, L.-F. (2004). Asymptotic distributions of quasi-maximum likelihood estimators for spatial autoregressive models. *Econometrica*, 72(6):1899–1925.

Leskovec, J., Adamic, L. A., and Huberman, B. A. (2007). The dynamics of viral marketing. *ACM Transactions on the Web (TWEB)*, 1(1):5–es.

Lewis, H. F. and Sexton, T. R. (2004). Network DEA: Efficiency analysis of organizations with complex internal structure. *Computers & Operations Research*, 31(9):1365–1410.

Li, T., Levina, E., and Zhu, J. (2019a). Prediction models for network-linked data. *The Annals of Applied Statistics*, 13(1):132–164.

Li, X., Ding, P., Lin, Q., Yang, D., and Liu, J. S. (2019b). Randomization inference for peer effects. *Journal of the American Statistical Association*.

Li, Y. (2017). Deep reinforcement learning: An overview. *CoRR*, abs/1701.07274.

Lu, X. and Van Roy, B. (2017). Ensemble sampling. *Advances in Neural Information Processing Systems*, 30:3260–3268.

Makarova, A., Shen, H., Perrone, V., Klein, A., Faddoul, J. B., Krause, A., Seeger, M., and Archambeau, C. (2022). Automatic termination for hyperparameter optimization. In *International Conference on Automated Machine Learning*, pages 7–1. Proceedings of Machine Learning Research.

Manski, C. F. (1993). Identification of endogenous social effects: The reflection problem. *The Review of Economic Studies*, 60(3):531–542.

Martin, A. D. and Quinn, K. M. (1996). A review of discrete optimization algorithms. *The Political Methodologist*, 7(2):6–10.

McCullagh, P. and Nelder, J. A. (1989). *Generalized Linear Models*, volume 37. CRC Press.

Mockus, J. (1989). *Bayesian approach to global optimization: Theory and applications*. Springer.

Montgomery, D. C. (2019). *Design and analysis of experiments*. John Wiley & Sons.

Newman, M. E. (2006). Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, 74(3):036104.

Owen-Smith, J. and Powell, W. W. (2004). Knowledge networks as channels and conduits: The effects of spillovers in the Boston biotechnology community. *Organization Science*, 15(1):5–21.

Park, H. J., Kincaid, D. L., Chung, K. K., Han, D. S., and Lee, S. B. (1976). The Korean mothers' club program. *Studies in Family Planning*, 7(10):275–283.

Parker, B. M., Gilmour, S. G., and Koutra, V. (2018). A graph-theoretic framework for algorithmic design of experiments. *arXiv*.

Parker, B. M., Gilmour, S. G., and Schormans, J. (2017). Optimal design of experiments on connected units with application to social networks. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 66(3):455–480.

Piras, G. and Sarrias, M. (2023). GMM estimators for binary spatial models in R. *Journal of Statistical Software*, 107:1–33.

Pokhilko, V., Zhang, Q., Kang, L., and D'arcy, P. M. (2019). D-Optimal design for network A/B testing. *Journal of Statistical Theory and Practice*, 13(4):61.

Pukelsheim, F. (2006). *Optimal design of experiments*. SIAM.

Reeves, C. R. (1993). *Modern heuristic techniques for combinatorial problems*. John Wiley & Sons, Inc.

Rencher, A. C. and Schaalje, G. B. (2008). *Linear models in statistics*. John Wiley & Sons.

Ribeiro, C. C., Rosseti, I., and Souza, R. C. (2011). Effective probabilistic stopping rules for randomized metaheuristics: GRASP implementations. In *International Conference on Learning and Intelligent Optimization*, pages 146–160. Springer.

Rossi, R. A. and Ahmed, N. K. (2015). The network data repository with interactive graph analytics and visualization. In *AAAI*.

Rual, J.-F., Venkatesan, K., Hao, T., Hirozane-Kishikawa, T., Dricot, A., Li, N., Berriz, G. F., Gibbons, F. D., Dreze, M., Ayivi-Guedehoussou, N., et al. (2005). Towards a proteome-scale map of the human protein-protein interaction network. *Nature*, 437(7062):1173–1178.

Rubin, D. B. (1980). Randomization analysis of experimental data: The Fisher randomization test comment. *Journal of the American Statistical Association*, 75(371):591–593.

Ryan, E. G., Drovandi, C. C., McGree, J. M., and Pettitt, A. N. (2016). A review of modern computational algorithms for Bayesian optimal design. *International Statistical Review*, 84(1):128–154.

Saint-Jacques, G., Varshney, M., Simpson, J., and Xu, Y. (2019). Using ego-clusters to measure network effects at LinkedIn. *arXiv*.

Sardiello, M., Palmieri, M., di Ronza, A., Medina, D. L., Valenza, M., Gennarino, V. A., Di Malta, C., Donaudy, F., Embrione, V., Polishchuk, R. S., et al. (2009). A gene network regulating lysosomal biogenesis and function. *Science*, 325(5939):473–477.

Sattar, A. M., Ertuğrul, Ö. F., Gharabaghi, B., McBean, E. A., and Cao, J. (2019). Extreme learning machine model for water network management. *Neural Computing and Applications*, 31(1):157–169.

Saveski, M., Pouget-Abadie, J., Saint-Jacques, G., Duan, W., Ghosh, S., Xu, Y., and Airoldi, E. M. (2017). Detecting network effects: Randomizing over randomized experiments. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1027–1035.

Sävje, F. (2023). Causal inference with misspecified exposure mappings: Separating definitions and assumptions. *Biometrika*, page asad019.

Schafer, J. L. and Kang, J. (2008). Average causal effects from nonrandomized studies: A practical guide and simulated example. *Psychological Methods*, 13(4):279.

Serfling, R. J. (1980). *Approximation theorems of mathematical statistics*. Wiley Series in Probability and Statistics. John Wiley & Sons, Inc.

Shalita, A., Karrer, B., Kabiljo, I., Sharma, A., Presta, A., Adcock, A., Kllapi, H., and Stumm, M. (2016). Social hash: An assignment framework for optimizing distributed systems operations on social networks. In *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*, pages 455–468.

Shalizi, C. R. and Thomas, A. C. (2011). Homophily and contagion are generically confounded in observational social network studies. *Sociological Methods & Research*, 40(2):211–239.

Silvey, S., Titterington, D., and Torsney, B. (1978). An algorithm for optimal designs on a design space. *Communications in Statistics: Theory and Methods*, 7(14):1379–1389.

Spall, J. C. (2005). *Introduction to stochastic search and optimization: Estimation, simulation, and control*. John Wiley & Sons.

Srinivas, N., Krause, A., Kakade, S. M., and Seeger, M. (2010). Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, pages 1015–1022, Madison, WI, USA. Omnipress.

Stelzl, U., Worm, U., Lalowski, M., Haenig, C., Brembeck, F. H., Goehler, H., Stroedicke, M., Zenkner, M., Schoenherr, A., Koeppen, S., et al. (2005). A human protein-protein interaction network: A resource for annotating the proteome. *Cell*, 122(6):957–968.

Sussman, D. L. and Airoldi, E. M. (2017). Elements of estimation theory for causal effects in the presence of network interference. *arXiv*.

Swersky, K., Rubanova, Y., Dohan, D., and Murphy, K. (2020). Amortized Bayesian optimization over discrete spaces. In *Conference on Uncertainty in Artificial Intelligence*, pages 769–778. Proceedings of Machine Learning Research.

Tichy, N. M., Tushman, M. L., and Fombrun, C. (1979). Social network analysis for organizations. *Academy of Management Review*, 4(4):507–519.

Traud, A. L., Mucha, P. J., and Porter, M. A. (2012). Social structure of Facebook networks. *Journal of Physics A*, 391(16):4165–4180.

Ugander, J., Karrer, B., Backstrom, L., and Kleinberg, J. (2013). Graph cluster randomization: Network exposure to multiple universes. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 329–337.

Van der Vaart, A. W. (2000). *Asymptotic statistics*, volume 3. Cambridge University Press.

Vázquez, A. (2003). Growing network with local rules: Preferential attachment, clustering hierarchy, and degree correlations. *Physical Review E*, 67(5):056104.

Wedderburn, R. W. (1974). Quasi-likelihood functions, generalized linear models, and the Gauss-Newton method. *Biometrika*, 61(3):439–447.

White, H. (1996). *Estimation, inference and specification analysis*. Cambridge University Press.

Williams, C. K. (1998). Prediction with Gaussian processes: From linear regression to linear prediction and beyond. In *Learning in Graphical Models*, pages 599–621. Springer.

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256.

Wu, C. J. and Hamada, M. S. (2021). *Experiments: Planning, analysis, and optimization*. John Wiley & Sons.

Xia, M., Wang, J., and He, Y. (2013). BrainNet Viewer: A network visualization tool for human brain connectomics. *PloS One*, 8(7):e68910.

Xu, Y., Chen, N., Fernandez, A., Sinno, O., and Bhasin, A. (2015). From infrastructure to culture: A/B testing challenges in large scale social networks. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 2227–2236.

Xu, Z. and Harriss, R. (2008). Exploring the structure of the US intercity passenger air transportation network: A weighted complex network approach. *GeoJournal*, 73(2):87.

Yin, Q. (2021). Optimizing people you may know (PYMK) for equity in network creation. *LinkedIn Engineering*.

Zeileis, A., Köll, S., and Graham, N. (2020). Various versatile variances: An object-oriented implementation of clustered covariances in R. *Journal of Statistical Software*, 95:1–36.

Zhang, Q. and Kang, L. (2022). Locally optimal design for A/B tests in the presence of covariates and network dependence. *Technometrics*, 64(3):358–369.

# APPENDICES

# Appendix A

# Appendices for Chapter 2

## A.1 Mathematical Details for Section 2.3

### A.1.1 Proof of Lemma 2.1

We use the following two lemmas.

**Lemma A.1.** *(Theorem 18.2.16 of Harville (2008)) Let* $\mathbf{A}$ *represent an* $n \times n$ *matrix. Then, the infinite series* $\mathbf{I}+\mathbf{A}+\mathbf{A}^2+\mathbf{A}^3+\cdots$ *converges if and only if* $\lim_{k\to\infty} \mathbf{A}^k = \mathbf{0}$*, in which case* $\mathbf{I}-\mathbf{A}$ *is nonsingular and*

$$(\mathbf{I}-\mathbf{A})^{-1} = \sum_{k=0}^{\infty} \mathbf{A}^k = \mathbf{I}+\mathbf{A}+\mathbf{A}^2+\mathbf{A}^3+\cdots,$$

*where* $\mathbf{A}^0 = \mathbf{I}$.

**Lemma A.2.** *(Lemma 5.6.11 of Horn and Johnson (2012)) Let* $\mathbf{A}$ *be an* $n \times n$ *given matrix. If there is a matrix norm* $||\cdot||$ *such that* $||A|| < 1$*, then* $\lim_{k\to\infty} \mathbf{A}^k = 0$*, that is, each entry of* $\mathbf{A}^k$ *tends to zero as* $k \to \infty$.

From the two lemmas, if we have $||\rho_T \mathbf{W}_T + \rho_C \mathbf{W}_C|| < 1$ for any matrix norm $||\cdot||$, then $\mathbf{S}(\boldsymbol{\rho})$ will be invertible. Now, if the condition of Lemma 2.1 is satisfied, using triangle inequality, we can derive

$$||\rho_T \mathbf{W}_T + \rho_C \mathbf{W}_C|| \leq |\rho_T|||\mathbf{W}_T|| + |\rho_C|||\mathbf{W}_C||,$$

$$
\begin{aligned}
&\leq \max(|\rho_T|, |\rho_C|) \left[||\mathbf{W}_T|| + ||\mathbf{W}_C||\right], \\
&< 1,
\end{aligned}
$$

i.e., $\mathbf{S}(\boldsymbol{\rho})$ is invertible.

### A.1.2 Assumptions Needed for Asymptotic Results

In order to achieve the asymptotic results in Theorem 2.2 in Appendix A.1.3, we make the following assumptions.

**Assumption A.1.** $\boldsymbol{\epsilon}_n = (\epsilon_{1n}, \epsilon_{2n}, \ldots, \epsilon_{nn})^\top$ are independently and identically distributed with mean 0 and variance $\sigma_0^2 > 0$. In addition, the moment $\mathbb{E}(|\epsilon_{i,n}|^{4+\eta})$ exists for some $\eta > 0$.

**Assumption A.2.** The true parameters $\boldsymbol{\rho}_0$ and $\boldsymbol{\varphi}_0$ lie in the interior of a compact parameter space $\mathrm{P} \times \boldsymbol{\Phi}$. The parameters are uniquely identifiable, in the sense that $\mathbb{P}\left(L_n(\boldsymbol{\theta}_1) = L_n(\boldsymbol{\theta}_2)\right) = 0$ for $\boldsymbol{\theta}_1 \neq \boldsymbol{\theta}_2$.

**Assumption A.3.** The elements of $\mathbf{W}_{Tn}$ and $\mathbf{W}_{Cn}$ are at most of order $h_n$ uniformly, i.e., $W_{ln,ij} = O(1/h_n)\ \forall i, j$ and $l \in \{T, C\}$. The sequence $h_n$ can be bounded or divergent. Furthermore, $\lim_{n\to\infty} h_n/n = 0$.

**Assumption A.4.** The matrix $\mathbf{S}_n(\boldsymbol{\rho}_0)$ is nonsingular.

**Assumption A.5.** The weight matrices $\mathbf{W}_{Tn}$, $\mathbf{W}_{Cn}$ and the matrix $\mathbf{S}_n(\boldsymbol{\rho}_0)^{-1}$ are uniformly bounded in both row and column sums. Moreover, $\mathbf{S}_n(\boldsymbol{\rho})^{-1}$ is uniformly bounded in either row or column sums.

**Assumption A.6.** For each $i$, the functions $g_{T,i}$ and $g_{C,i}$ are twice continuously differentiable with respect to $\boldsymbol{\varphi}$. The values of these functions and their derivatives are uniformly bounded $\forall \boldsymbol{\varphi} \in \boldsymbol{\Phi}$. Furthermore, $\forall \boldsymbol{\varphi} \in \boldsymbol{\Phi}$, $\lim_{n\to\infty} \mathbf{M}_n(\boldsymbol{\varphi})^\top \mathbf{M}_n(\boldsymbol{\varphi})/n$ exists and nonsingular.

Assumptions A.1, A.2 and the differentiability requirement for $g_{T,i}$ and $g_{C,i}$ in Assumption A.6 are usual regularity conditions for the consistency and asymptotic normality of nonlinear least squares regression (Jennrich, 1969). The identifiability requirement in Assumption A.2 contains the requirement that the columns of the model matrix $\mathbf{M}_n(\boldsymbol{\varphi})$ are linearly independent as discussed in Section 2.3.1.

Note that we cannot use the usual central limit theorems to derive the asymptotic behavior of Model (2.17) because as the sample size $n$ changes, the weight matrices

may also change, leading to changes in the outcomes $\mathbf{Y}_n$. For example, when a new unit is added to the network, it can be connected to other existing units, which in turn changes the degree $k_i$ for each existing unit $i$, $i = 1, 2, \ldots, n$. This results in a different set of weight matrices for autoregressive specifications such as the LAG (2.7) or the HOM (2.4) specifications. Therefore, we need to use the Central Limit Theorem for linear-quadratic forms of triangular arrays (Kelejian and Prucha, 2010). Assumptions A.3, A.5, and A.6 are introduced to satisfy the assumptions of this theorem. Essentially, the bounds in these assumptions control the spatial correlations to a manageable degree so that they do not diverge as $n$ goes to infinity (Lee, 2004). For example, suppose $\mathbf{W}_{Tn} = \mathbf{A}_n$, Assumption A.3 is satisfied as all elements of $\mathbf{A}$ are either 1 or 0, i.e., bounded. However, to satisfy Assumption A.5 in this case, we need to further require that the degree of each unit $i$, i.e., the number of connections, is bounded as $n$ goes to infinity. This is reasonable in social network settings as one will not have infinitely many friends.

Finally, Assumption A.4 makes sure that $\mathbf{Y}_n$ can be expressed in the reduced form as in (2.17).

### A.1.3 Proof of Theorem 2.2

To prove consistency, we use the following lemma.

**Lemma A.3.** *(Theorem 3.4 of White (1996)) Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a complete probability space, let $\boldsymbol{\Theta}$ be a compact subset of $\mathbb{R}^p$, $p \in \mathbb{N}$ and let $\{\boldsymbol{\Theta}_n\}$ be a sequence of compact subsets of $\boldsymbol{\Theta}$. Let $\{Q_n\}$ be a sequence of random functions continuous on $\boldsymbol{\Theta}$ a.s. and let $\hat{\boldsymbol{\theta}}_n = \arg\max_{\boldsymbol{\Theta}_n} Q_n(\cdot, \boldsymbol{\theta})$ a.s. If $Q_n(\cdot, \boldsymbol{\theta}) - \bar{Q}_n(\boldsymbol{\theta}) \to 0$ as $n \to \infty$ a.s. uniformly on $\boldsymbol{\Theta}$ and if $\{\bar{Q}_n : \boldsymbol{\Theta} \to \mathbb{R}\}$ has identifiably unique maximizers $\{\boldsymbol{\theta}_n^*\}$ on $\{\boldsymbol{\Theta}_n\}$ then $\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_n^* \to 0$ as $n \to \infty$ a.s.*

From the reduced form (2.17), consider

$$
\begin{aligned}
Q_n(\boldsymbol{\rho}, \boldsymbol{\varphi}) &= \max_{\boldsymbol{\beta}, \sigma^2} \mathbb{E}(\log L_n(\boldsymbol{\theta})), \\
&= \max_{\boldsymbol{\beta}, \sigma^2} \Bigg[ -\frac{n}{2}\log 2\pi - \frac{n}{2}\log(\sigma^2) + \log|\mathbf{S}_n(\boldsymbol{\rho})| \\
&\qquad - \frac{1}{2\sigma^2}\boldsymbol{\beta}^\top \mathbf{M}_n(\boldsymbol{\varphi})^\top \mathbf{M}_n(\boldsymbol{\varphi})\boldsymbol{\beta} - \frac{\sigma_0^2}{2\sigma^2}\operatorname{tr}\big(\mathbf{B}_n(\boldsymbol{\rho})\big) \\
&\qquad + \frac{1}{\sigma^2}\boldsymbol{\beta}^\top \mathbf{M}_n(\boldsymbol{\varphi})^\top \mathbf{S}_n(\boldsymbol{\rho})\mathbf{S}_n(\boldsymbol{\rho}_0)^{-1}\mathbf{M}_n(\boldsymbol{\varphi}_0)\boldsymbol{\beta}_0
\end{aligned}
$$

$$- \frac{1}{2\sigma^2}\boldsymbol{\beta}_0^\top \mathbf{M}_n(\boldsymbol{\varphi}_0)^\top \mathbf{S}_n(\boldsymbol{\rho}_0)^{-\top}\mathbf{S}_n(\boldsymbol{\rho})^\top \mathbf{S}_n(\boldsymbol{\rho})\mathbf{S}_n(\boldsymbol{\rho}_0)^{-1}\mathbf{M}_n(\boldsymbol{\varphi}_0)\boldsymbol{\beta}_0\Big],$$

where $\mathbf{B}_n(\boldsymbol{\rho}) = \mathbf{S}_n(\rho_0)^{-\top}\mathbf{S}_n(\boldsymbol{\rho})^\top\mathbf{S}_n(\boldsymbol{\rho})\mathbf{S}_n(\rho_0)^{-1}$. Taking the first derivative with respect to $\boldsymbol{\beta}$ and $\sigma^2$, we obtain the maximizers of $Q_n(\boldsymbol{\rho},\boldsymbol{\varphi})$ as follows:

$$\begin{aligned}
\boldsymbol{\beta}_n^*(\boldsymbol{\rho},\boldsymbol{\varphi}) =& \big[\mathbf{M}_n(\boldsymbol{\varphi})^\top\mathbf{M}_n(\boldsymbol{\varphi})\big]^{-1}\mathbf{M}_n(\boldsymbol{\varphi})^\top\mathbf{S}_n(\boldsymbol{\rho})\mathbf{S}_n(\boldsymbol{\rho}_0)^{-1}\mathbf{M}_n(\boldsymbol{\varphi}_0)\boldsymbol{\beta}_0;\\
\sigma_n^{*2}(\boldsymbol{\rho},\boldsymbol{\varphi}) =& \frac{1}{n}\Big\{\boldsymbol{\beta}_0^\top\mathbf{M}_n(\boldsymbol{\varphi}_0)^\top\mathbf{S}_n^{-\top}(\boldsymbol{\rho}_0)\mathbf{S}_n(\boldsymbol{\rho})^\top\big[\mathbf{I}_n - \mathbf{H}_n(\boldsymbol{\varphi})\big]\\
&\times\mathbf{S}_n(\boldsymbol{\rho})\mathbf{S}_n(\boldsymbol{\rho}_0)^{-1}\mathbf{M}_n(\boldsymbol{\varphi}_0)\boldsymbol{\beta}_0 + \sigma_0^2\mathrm{tr}\big(\mathbf{B}_n(\boldsymbol{\rho})\big)\Big\},
\end{aligned}\tag{A.1}$$

where $\mathbf{H}_n(\boldsymbol{\varphi}) = \mathbf{M}_n(\boldsymbol{\varphi})\big[\mathbf{M}_n(\boldsymbol{\varphi})^\top\mathbf{M}_n(\boldsymbol{\varphi})\big]^{-1}\mathbf{M}_n(\boldsymbol{\varphi})^\top$. Now,

$$\begin{aligned}
\ell_{p,n}(\boldsymbol{\rho},\boldsymbol{\varphi}) =& -\frac{n}{2}\log 2\pi - \frac{n}{2}\log\hat{\sigma}_n^2(\boldsymbol{\rho},\boldsymbol{\varphi}) + \log|\mathbf{S}_n(\boldsymbol{\rho})| - \frac{n}{2},\\
Q_n(\boldsymbol{\rho},\boldsymbol{\varphi}) =& -\frac{n}{2}\log 2\pi - \frac{n}{2}\log\sigma_n^{*2}(\boldsymbol{\rho},\boldsymbol{\varphi}) + \log|\mathbf{S}_n(\boldsymbol{\rho})| - \frac{n}{2},
\end{aligned}$$

where $\hat{\sigma}_n^2(\boldsymbol{\rho},\boldsymbol{\varphi})$ was given in (2.20). To use Lemma A.3, we first need to show

$$\frac{1}{n}\big\{\ell_{p,n}(\boldsymbol{\rho},\boldsymbol{\varphi}) - Q_n(\boldsymbol{\rho},\boldsymbol{\varphi})\big\} = -\frac{1}{2}\big\{\log\hat{\sigma}_n^2(\boldsymbol{\rho},\boldsymbol{\varphi}) - \log\sigma_n^{*2}(\boldsymbol{\rho},\boldsymbol{\varphi})\big\} = o_p(1).$$

Note that

$$\hat{\sigma}_n^2(\boldsymbol{\rho},\boldsymbol{\varphi}) - \sigma_n^{*2}(\boldsymbol{\rho},\boldsymbol{\varphi}) = 2R_{1n}(\boldsymbol{\rho},\boldsymbol{\varphi}) + R_{2n}(\boldsymbol{\rho},\boldsymbol{\varphi}) - \frac{1}{n}\sigma_0^2\mathrm{tr}\big(\mathbf{B}_n(\boldsymbol{\rho})\big),$$

where

$$R_{1n}(\boldsymbol{\rho},\boldsymbol{\varphi}) = \frac{1}{n}\boldsymbol{\beta}_0^\top\mathbf{M}_n(\boldsymbol{\rho}_0)^\top\mathbf{S}_n^{-\top}(\boldsymbol{\rho}_0)\mathbf{S}_n(\boldsymbol{\rho})^\top\big[\mathbf{I}_n - \mathbf{H}_n(\boldsymbol{\varphi})\big]\mathbf{S}_n(\boldsymbol{\rho})\mathbf{S}_n(\boldsymbol{\rho}_0)^{-1}\boldsymbol{\epsilon}_n,$$

and

$$\begin{aligned}
R_{2n}(\boldsymbol{\rho},\boldsymbol{\varphi}) =& \frac{1}{n}\boldsymbol{\epsilon}_n^\top\mathbf{S}_n^{-\top}(\boldsymbol{\rho}_0)\mathbf{S}_n(\boldsymbol{\rho})^\top\big[\mathbf{I}_n - \mathbf{H}_n(\boldsymbol{\varphi})\big]\mathbf{S}_n(\boldsymbol{\rho})\mathbf{S}_n(\boldsymbol{\rho}_0)^{-1}\boldsymbol{\epsilon}_n\\
=& \frac{1}{n}\boldsymbol{\epsilon}_n^\top\mathbf{S}_n^{-\top}(\boldsymbol{\rho}_0)\mathbf{S}_n(\boldsymbol{\rho})^\top\mathbf{S}_n(\boldsymbol{\rho})\mathbf{S}_n(\boldsymbol{\rho}_0)^{-1}\boldsymbol{\epsilon}_n\\
&- \frac{1}{n}\Big[\frac{1}{\sqrt{n}}\mathbf{M}_n(\boldsymbol{\varphi})^\top\mathbf{S}_n(\boldsymbol{\rho})\mathbf{S}_n(\boldsymbol{\rho}_0)^{-1}\boldsymbol{\epsilon}_n\Big]^\top\Big[\frac{1}{n}[\mathbf{M}_n(\boldsymbol{\varphi})^\top\mathbf{M}_n(\boldsymbol{\varphi})]^{-1}\Big]
\end{aligned}$$

$$\times \left[\frac{1}{\sqrt{n}}\mathbf{M}_n(\boldsymbol{\varphi})^\top \mathbf{S}_n(\boldsymbol{\rho})\mathbf{S}_n(\boldsymbol{\rho}_0)^{-1}\boldsymbol{\epsilon}_n\right].$$

It can be shown that $R_{1n}(\boldsymbol{\rho},\boldsymbol{\varphi}) = o_P(1)$ and $R_{2n}(\boldsymbol{\rho},\boldsymbol{\varphi}) - \frac{1}{n}\sigma_0^2 \mathrm{tr}\big(\mathbf{B}_n(\boldsymbol{\rho})\big) = o_P(1)$ using the following three lemmas given in Lee (2004) and assumptions on the bounds of matrices.

**Lemma A.4.** *Suppose the elements $A_{n,ij}$ of $n \times n$ matrices $\mathbf{A}_n$ are $O(1/h_n)$ uniformly for all $i, j$. If $n \times n$ matrices $\{\mathbf{B}_n\}$ are uniformly bounded in column sums (respectively, row sums), then the elements of $\mathbf{A}_n\mathbf{B}_n$ (respectively, $\mathbf{B}_n\mathbf{A}_n$) have the uniform order $O(1/h_n)$. For these cases, $\mathrm{tr}(\mathbf{A}_n\mathbf{B}_n) = \mathrm{tr}(\mathbf{B}_n\mathbf{A}_n) = O(n/h_n)$.*

**Lemma A.5.** *Suppose $\{\mathbf{A}_n\}$ are uniformly bounded either in row or column sums and their elements $A_{n,ij}$ have order $O(1/h_n)$ uniformly in $i$ and $j$. Then $\mathbb{E}(\boldsymbol{\varepsilon}_n^\top \mathbf{A}_n \boldsymbol{\varepsilon}_n) = \sigma_0^2 \mathrm{tr}(\mathbf{A}_n) = O(n/h_n)$ and $\mathrm{Var}(\boldsymbol{\varepsilon}_n^\top \mathbf{A}_n \boldsymbol{\varepsilon}_n) = O(n/h_n)$. If $\lim_{n\to\infty} h_n/n = 0$, then $(h_n/n)[\boldsymbol{\varepsilon}_n^\top \mathbf{A}_n \boldsymbol{\varepsilon}_n - \mathbb{E}(\boldsymbol{\varepsilon}_n^\top \mathbf{A}_n \boldsymbol{\varepsilon}_n)] = o_P(1)$, where $\epsilon_n$ satisfies Assumption A.1 (possibly without normality but with $\mathbb{E}(|\epsilon_n|^{4+\gamma}) < \infty$ for some $\gamma > 0$).*

**Lemma A.6.** *Suppose that $\mathbf{A}_n$ is a square matrix with its column sums being uniformly bounded and elements of the $n \times k$ matrix $\mathbf{Z}_n$ are uniformly bounded. Then $(1/\sqrt{n})\mathbf{Z}_n^\top \mathbf{A}_n \boldsymbol{\epsilon}_n = O_p(1)$. Furthermore, if the limit of $\mathbf{Z}_n^\top \mathbf{A}_n \mathbf{A}_n^\top \mathbf{Z}_n/n$ exists and is positive definite, then $(1/\sqrt{n})\mathbf{Z}_n' \mathbf{A}_n \boldsymbol{\epsilon}_n \xrightarrow{d} \mathcal{N}(0, \sigma_0^2 \lim_{n\to\infty} \mathbf{Z}_n^\top \mathbf{A}_n \mathbf{A}_n^\top \mathbf{Z}_n/n)$.*

Therefore, $\hat{\sigma}_n^2(\boldsymbol{\rho},\boldsymbol{\varphi}) - \sigma_n^{*2}(\boldsymbol{\rho},\boldsymbol{\varphi}) = o_P(1)$ uniformly on $\mathrm{P}\times\boldsymbol{\Phi}$. Hence, $\sup_{(\boldsymbol{\rho},\boldsymbol{\varphi})\in \mathrm{P}\times\boldsymbol{\Phi}} \frac{1}{n}\{\ell_{p,n}(\boldsymbol{\rho},\boldsymbol{\varphi}) - Q_n(\boldsymbol{\rho},\boldsymbol{\varphi})\} = o_P(1)$. Second, we need to prove the identification uniqueness condition that, for any $\epsilon_1, \epsilon_2 > 0$,

$$\limsup_{n\to\infty} \max_{\boldsymbol{\rho}\in \bar{N}_{\epsilon_1}(\boldsymbol{\rho}_0), \boldsymbol{\varphi}\in \bar{N}_{\epsilon_2}(\boldsymbol{\varphi}_0)} \frac{1}{n}\big[Q_n(\boldsymbol{\rho},\boldsymbol{\varphi}) - Q_n(\boldsymbol{\rho}_0,\boldsymbol{\varphi}_0)\big] < 0,$$

where $\bar{N}_{\epsilon_1}(\boldsymbol{\rho}_0)$ denotes the complement of an open neighborhood of $\boldsymbol{\rho}_0$ of diameter $\epsilon_1$ and likewise for $\boldsymbol{\varphi}$. To see this, we can write

$$\begin{aligned}\frac{1}{n}\big[Q_n(\boldsymbol{\rho},\boldsymbol{\varphi}) - Q_n(\boldsymbol{\rho}_0,\boldsymbol{\varphi}_0)\big] = \frac{1}{n}\Big\{\mathbb{E}[\log L_n(\boldsymbol{\rho},\boldsymbol{\beta}_0,\boldsymbol{\varphi}_0)] - \mathbb{E}[\log L_n(\boldsymbol{\rho}_0,\boldsymbol{\beta}_0,\boldsymbol{\varphi}_0)]\Big\} \\ - \frac{1}{2}\Big\{\log \sigma_n^{*2}(\boldsymbol{\rho},\boldsymbol{\varphi}) - \log\left(\frac{\sigma_0^2}{n}\mathrm{tr}(\mathbf{B}_n(\boldsymbol{\rho}))\right)\Big\}.\end{aligned}$$

The first term is less than 0 by Jensen's inequality and the identifiability condition of Assumption A.2. Furthermore, $\sigma_n^{*2}(\boldsymbol{\rho},\boldsymbol{\varphi}) \geq \frac{\sigma_0^2}{n}\mathrm{tr}(\mathbf{B}_n(\boldsymbol{\rho}))$ from (A.1) by the positive

semi-definiteness of the annihilator matrix $\mathbf{I} - \mathbf{H}_n(\boldsymbol{\varphi})$. Putting all of these together, we proved $\hat{\boldsymbol{\theta}}_n = \boldsymbol{\theta}_0 + o_P(1)$.

Now, to prove the asymptotic normality, we apply the mean-value theorem on the first order derivative of $\log L_n(\boldsymbol{\theta})$ at $\hat{\boldsymbol{\theta}}_n$ yielding

$$\frac{\partial \log L_n(\hat{\boldsymbol{\theta}}_n)}{\partial \boldsymbol{\theta}} = \mathbf{0} = \frac{\partial \log L_n(\boldsymbol{\theta}_0)}{\partial \boldsymbol{\theta}} + (\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_0)\frac{\partial^2 \log L_n(\tilde{\boldsymbol{\theta}}_n)}{\partial \boldsymbol{\theta}\partial \boldsymbol{\theta}^\top},$$

where $\tilde{\boldsymbol{\theta}}_n$ lies between $\hat{\boldsymbol{\theta}}_n$ and $\boldsymbol{\theta}_0$. Therefore

$$\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_0 = -\left[\frac{\partial^2 \log L_n(\tilde{\boldsymbol{\theta}}_n)}{\partial \boldsymbol{\theta}\partial \boldsymbol{\theta}^\top}\right]^{-1}\left(\frac{\partial \log L_n(\boldsymbol{\theta}_0)}{\partial \boldsymbol{\theta}}\right).$$

We can write down the first derivatives of $\log L_n(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$ as follows:

$$\begin{aligned}
\frac{\partial \log L_n(\boldsymbol{\theta})}{\partial \boldsymbol{\beta}} &= \frac{1}{\sigma^2}\boldsymbol{\epsilon}_n^\top \mathbf{M}_n(\boldsymbol{\varphi}),\\
\frac{\partial \log L_n(\boldsymbol{\theta})}{\partial \sigma^2} &= -\frac{n}{2\sigma^2} + \frac{1}{2\sigma^4}\boldsymbol{\epsilon}_n^\top \boldsymbol{\epsilon}_n,\\
\frac{\partial \log L_n(\boldsymbol{\theta})}{\partial \boldsymbol{\rho}_j} &= \frac{1}{\sigma^2}\boldsymbol{\epsilon}_n^\top \mathbf{W}_{jn}\mathbf{S}_n(\boldsymbol{\rho})^{-1}\mathbf{M}_n(\boldsymbol{\varphi})\boldsymbol{\beta}\\
&\quad + \left(\frac{1}{\sigma^2}\boldsymbol{\epsilon}^\top \mathbf{W}_{jn}\mathbf{S}_n(\boldsymbol{\rho})^{-1}\boldsymbol{\epsilon}_n - \operatorname{tr}(\mathbf{W}_{jn}\mathbf{S}_n^{-1}(\boldsymbol{\rho}))\right),\\
\frac{\partial \log L_n(\boldsymbol{\theta})}{\partial \boldsymbol{\varphi}} &= \frac{1}{\sigma^2}\boldsymbol{\epsilon}_n^\top \frac{\partial \mathbf{M}_n(\boldsymbol{\varphi})}{\partial \boldsymbol{\varphi}}\boldsymbol{\beta},
\end{aligned}$$

for $j \in \{T, C\}$. Note that these are linear and quadratic functions of $\boldsymbol{\epsilon}_n$. Therefore we can apply the Central Limit Theorem for linear-quadratic functions (Kelejian and Prucha, 2010) given as Lemma A.7 below.

**Lemma A.7.** *(Theorem A.1 of Kelejian and Prucha (2010)) Consider the linear quadratic forms* $(r = 1, \ldots, m)$

$$\mathbf{Q}_{r,n} = \boldsymbol{\epsilon}_n^\top \mathbf{A}_{r,n}\boldsymbol{\epsilon}_n + \mathbf{B}_{r,n}^\top \boldsymbol{\epsilon}_n,$$

*where* $\boldsymbol{\epsilon}_n = (\epsilon_{1,n}, \ldots, \epsilon_{n,n})^\top$ *is an* $n \times 1$ *random vector, and* $\mathbf{A}_{r,n} = (a_{ij,r,n})_{i,j=1,\ldots,n}$ *is an* $n \times n$ *non-stochastic real matrix, and* $\mathbf{B}_{r,n} = (b_{1,r,n}, \ldots, b_{n,r,n})^\top$ *is an* $n \times 1$ *non-stochastic real vector. We make the following assumptions:*

1. *The real-valued random variables of the array* $\{\epsilon_{i,n} : 1 \leq i \leq n, n \geq 1\}$ *satisfy* $\mathbb{E}[\epsilon_{i,n}] = 0$. *Furthermore, for each* $n \geq 1$, *the random variables* $\epsilon_{1,n}, \ldots, \epsilon_{n,n}$ *are totally independent.*

2. *For* $r = 1, \ldots, m$, *the elements of the array of real numbers* $\{a_{ij,r,n} : 1 \leq i, j \leq n, n \geq 1\}$ *satisfy* $a_{ij,r,n} = a_{ji,r,n}$ *and* $\sup_{1 \leq j \leq n, n \geq 1} \sum_{i=1}^{n} |a_{ij,r,n}| < \infty$. *The elements of the array of real numbers* $\{b_{i,r,n} : 1 \leq i \leq n, n \geq 1\}$ *satisfy* $\sup_n n^{-1} \sum_{i=1}^{n} |b_{i,r,n}|^{2+\eta_1} < \infty$ *for some* $\eta_1 > 0$.

3. *For* $r = 1, \ldots, m$, *one of the following two conditions holds*

   (a) $\sup_{1 \leq i \leq n, n \geq 1} \mathbb{E}|\epsilon_{i,n}|^{2+\eta_2} < \infty$ *for some* $\eta_2 > 0$ *and* $a_{ii,r,n} = 0$.

   (b) $\sup_{1 \leq i \leq n, n \geq 1} \mathbb{E}|\epsilon_{i,n}|^{4+\eta_2} < \infty$ *for some* $\eta_2 > 0$ *(but possibly* $a_{ii,r,n} \neq 0$*)* .

*Let*

$$\mathbf{U}_n = [\mathbf{Q}_{1,n}, \ldots, \mathbf{Q}_{m,n}]^\top,$$

*and* $\boldsymbol{\mu}_{\mathbf{U}_n} = \mathbb{E}[\mathbf{U}_n]$ *and* $\boldsymbol{\Sigma}_{\mathbf{U}_n}$ *denote the mean and variance-covariance matrix of* $\mathbf{U}_n$, *respectively. Suppose the assumptions hold and* $n^{-1}\lambda_{\min}(\boldsymbol{\Sigma}_{U_n}) \geq c$ *for some* $c > 0$. *Let* $\boldsymbol{\Sigma}_{\mathbf{U}_n} = \left(\boldsymbol{\Sigma}_{\mathbf{U}_n}^{1/2}\right)\left(\boldsymbol{\Sigma}_{\mathbf{U}_n}^{1/2}\right)^\top$, *then*

$$\boldsymbol{\Sigma}_{\mathbf{U}_n}^{-1/2}(\mathbf{U}_n - \boldsymbol{\mu}_{\mathbf{U}_n}) \xrightarrow{d} \mathcal{N}(\mathbf{0}, \mathbf{I}_m).$$

Therefore, we can apply Lemma A.7 to $\frac{\partial \log L_n(\boldsymbol{\theta}_0)}{\partial \boldsymbol{\beta}}$ with $m = \dim(\boldsymbol{\theta})$ since all the multipliers to $\boldsymbol{\epsilon}_n$ are bounded. Note that the assumption on the minimum eigenvalue of the variance-covariance matrix is to ensure that matrices $\boldsymbol{\Sigma}_{\mathbf{V}_n}$ stay invertible as $n \to \infty$, to which we have an equivalent condition in Theorem (2.2). The assumption of symmetry is W.L.O.G since $\boldsymbol{\epsilon}_n \mathbf{A}_n \boldsymbol{\epsilon}_n = \boldsymbol{\epsilon}_n^\top [(\mathbf{A}_n + \mathbf{A}_n^\top)/2] \boldsymbol{\epsilon}_n$ (Kelejian and Prucha, 2010). Hence we have

$$[\mathbf{V}_n(\boldsymbol{\theta}_0)]^{-1/2} \frac{\partial \log L_n(\boldsymbol{\theta}_0)}{\partial \boldsymbol{\theta}} \xrightarrow{d} \mathcal{N}(\mathbf{0}_{\dim(\boldsymbol{\theta})}, \mathbf{I}_{\dim.(\boldsymbol{\theta})}) \tag{A.2}$$

Now, what is left to be proved is

$$\frac{1}{n} \frac{\partial \log L_n(\tilde{\boldsymbol{\theta}}_n)}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^\top} = \frac{1}{n} \frac{\partial \log L_n(\boldsymbol{\theta}_0)}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^\top} + o_P(1) = \frac{1}{n} \mathbb{E}\left[\frac{\partial \log L_n(\boldsymbol{\theta}_0)}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^\top}\right] + o_P(1). \tag{A.3}$$

The second derivatives of $\log L_n(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$ are

$$\frac{\partial^2 \log L_n(\boldsymbol{\theta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^\top} = -\frac{1}{\sigma^2} \mathbf{M}_n(\boldsymbol{\varphi})^\top \mathbf{M}_n(\boldsymbol{\varphi}),$$

$$
\begin{aligned}
\frac{\partial^2 \log L_n(\boldsymbol{\theta})}{(\partial\sigma)^2} &= \frac{n}{2\sigma^4} - \frac{1}{\sigma^6}\boldsymbol{\epsilon}_n^\top\boldsymbol{\epsilon}_n, \\
\frac{\partial^2 \log L_n(\boldsymbol{\theta})}{\partial\boldsymbol{\rho}_j\partial\boldsymbol{\rho}_l} &= -\frac{1}{\sigma^2}\mathbf{Y}_n^\top\mathbf{W}_{ln}^\top\mathbf{W}_{jn}\mathbf{Y}_n \\
&\quad - \mathrm{tr}(\mathbf{W}_{ln}\mathbf{S}_n(\boldsymbol{\rho})^{-1}\mathbf{W}_{jn}\mathbf{S}_n(\boldsymbol{\rho})^{-1}), \\
\frac{\partial^2 \log L_n(\boldsymbol{\theta})}{\partial\boldsymbol{\varphi}\partial\boldsymbol{\varphi}^\top} &= \frac{1}{\sigma^2}\Bigg[-\boldsymbol{\beta}^\top\left(\frac{\partial\mathbf{M}_n(\boldsymbol{\varphi})}{\partial\boldsymbol{\varphi}}\right)^\top\left(\frac{\partial\mathbf{M}_n(\boldsymbol{\varphi})}{\partial\boldsymbol{\varphi}}\right)\boldsymbol{\beta} \\
&\quad + \boldsymbol{\beta}^\top\left(\frac{\partial^2\mathbf{M}_n(\boldsymbol{\varphi})}{\partial\boldsymbol{\varphi}\partial\boldsymbol{\varphi}^\top}\right)^\top\boldsymbol{\epsilon}_n\Bigg], \\
\frac{\partial^2 \log L_n(\boldsymbol{\theta})}{\partial\sigma^2\partial\boldsymbol{\beta}^\top} &= -\frac{1}{\sigma^4}\mathbf{M}_n(\boldsymbol{\varphi})^\top\boldsymbol{\epsilon}_n, \\
\frac{\partial^2 \log L_n(\boldsymbol{\theta})}{\partial\sigma^2\partial\boldsymbol{\rho}_j} &= -\frac{1}{\sigma^4}\mathbf{Y}_n^\top\mathbf{W}_{jn}^\top\boldsymbol{\epsilon}_n, \\
\frac{\partial^2 \log L_n(\boldsymbol{\theta})}{\partial\sigma^2\partial\boldsymbol{\varphi}^\top} &= -\frac{1}{\sigma^4}\boldsymbol{\beta}^\top\left(\frac{\partial\mathbf{M}_n(\boldsymbol{\varphi})}{\partial\boldsymbol{\varphi}}\right)^\top\boldsymbol{\epsilon}_n, \\
\frac{\partial^2 \log L_n(\boldsymbol{\theta})}{\partial\boldsymbol{\beta}\partial\boldsymbol{\rho}_j} &= -\frac{1}{\sigma^2}\mathbf{Y}_n^\top\mathbf{W}_{jn}^\top\mathbf{M}_n(\boldsymbol{\varphi}), \\
\frac{\partial^2 \log L_n(\boldsymbol{\theta})}{\partial\boldsymbol{\varphi}\partial\boldsymbol{\rho}_j} &= -\frac{1}{\sigma^2}\mathbf{Y}_n^\top\mathbf{W}_{jn}^\top\left(\frac{\partial\mathbf{M}_n(\boldsymbol{\varphi})}{\partial\boldsymbol{\varphi}}\right)\boldsymbol{\beta}, \\
\frac{\partial^2 \log L_n(\boldsymbol{\theta})}{\partial\boldsymbol{\beta}\partial\boldsymbol{\varphi}^\top} &= \frac{1}{\sigma^2}\Bigg[-\boldsymbol{\beta}^\top\left(\frac{\partial\mathbf{M}_n(\boldsymbol{\varphi})}{\partial\boldsymbol{\varphi}}\right)^\top\mathbf{M}_n(\boldsymbol{\varphi}) \\
&\quad + \boldsymbol{\epsilon}_n^\top\left(\frac{\partial\mathbf{M}_n(\boldsymbol{\varphi})}{\partial\boldsymbol{\varphi}}\right)\Bigg],
\end{aligned}
$$

for $l, j \in \{T, C\}$. Let $\mathbf{C}_{kn}(\boldsymbol{\rho}) = \mathbf{W}_{kn}\mathbf{S}_n(\boldsymbol{\rho})^{-1}$ for $k \in \mathsf{T}, C\}$. Using the mean-value theorem, we have

$$
\begin{aligned}
\frac{1}{n}\mathrm{tr}(\mathbf{W}_{ln}\mathbf{S}_n(\tilde{\boldsymbol{\rho}}_n)^{-1}\mathbf{W}_{jn}\mathbf{S}_n(\tilde{\boldsymbol{\rho}}_n)^{-1}) &= \frac{1}{n}\mathrm{tr}(\mathbf{C}_{ln}(\tilde{\boldsymbol{\rho}}_n)\mathbf{C}_{jn}(\tilde{\boldsymbol{\rho}}_n)) \\
&= \frac{1}{n}\mathrm{tr}(\mathbf{C}_{ln}(\boldsymbol{\rho}_0)\mathbf{C}_{jn}(\boldsymbol{\rho}_0)) + \frac{1}{n}(\tilde{\boldsymbol{\rho}}_n - \boldsymbol{\rho}_0)^\top \\
&\quad \times \begin{bmatrix} \cdots \\ \mathrm{tr}\Big\{\mathbf{C}_{kn}(\bar{\boldsymbol{\rho}}_n)\Big(\mathbf{C}_{ln}(\bar{\boldsymbol{\rho}}_n)\mathbf{C}_{jn}(\bar{\boldsymbol{\rho}}_n) + \mathbf{C}_{jn}(\bar{\boldsymbol{\rho}}_n)\mathbf{C}_{ln}(\bar{\boldsymbol{\rho}}_n)\Big)\Big\} \\ \cdots \end{bmatrix}
\end{aligned}
$$

$$= \frac{1}{n}\text{tr}(\mathbf{C}_{ln}(\boldsymbol{\rho}_0)\mathbf{C}_{jn}(\boldsymbol{\rho}_0)) + o_P(1),$$

where $\bar{\boldsymbol{\rho}}_n$ lies between $\tilde{\boldsymbol{\rho}}_n$ and $\boldsymbol{\rho}_0$. By consistency, $\hat{\boldsymbol{\theta}}_n = \boldsymbol{\theta}_0 + o_P(1)$, thus $\tilde{\boldsymbol{\theta}}_n = \boldsymbol{\theta}_0 + o_P(1)$ and $\bar{\boldsymbol{\rho}}_n = \boldsymbol{\theta}_0 + o_P(1)$. The elements of $\mathbf{C}_{kn}(\bar{\boldsymbol{\rho}}_n)\mathbf{C}_{ln}(\bar{\boldsymbol{\rho}}_n)\mathbf{C}_{jn}(\bar{\boldsymbol{\rho}}_n)$ and $\mathbf{C}_{kn}(\bar{\boldsymbol{\rho}}_n)\mathbf{C}_{jn}(\bar{\boldsymbol{\rho}}_n)\mathbf{C}_{ln}(\bar{\boldsymbol{\rho}}_n)$ are $O(1)$ and the elements of $\mathbf{Y}_n^\top\mathbf{W}_{ln}^\top\mathbf{W}_{jn}\mathbf{Y}_n$ are $O_P(n/h_n)$ by Lemma AA.6. Therefore

$$\frac{1}{n}\frac{\partial^2 \log L_n(\tilde{\boldsymbol{\theta}}_n)}{\partial\boldsymbol{\rho}_j\partial\boldsymbol{\rho}_l} = \frac{1}{n}\frac{\partial^2 \log L_n(\boldsymbol{\theta}_0)}{\partial\boldsymbol{\rho}_j\partial\boldsymbol{\rho}_l} + o_P(1).$$

For other partial derivative terms, we can use the fact that

$$\begin{aligned}
\mathbf{M}_n(\tilde{\boldsymbol{\varphi}}_n) &= \mathbf{M}_n(\boldsymbol{\varphi}_0) + o(1),\\
\frac{\partial\mathbf{M}_n(\tilde{\boldsymbol{\varphi}}_n)}{\partial\boldsymbol{\varphi}} &= \frac{\partial\mathbf{M}_n(\boldsymbol{\varphi}_0)}{\partial\boldsymbol{\varphi}} + o(1),\\
\frac{\partial^2\mathbf{M}_n(\tilde{\boldsymbol{\varphi}}_n)}{\partial\boldsymbol{\varphi}\partial\boldsymbol{\varphi}^\top} &= \frac{\partial^2\mathbf{M}_n(\boldsymbol{\varphi}_0)}{\partial\boldsymbol{\varphi}\partial\boldsymbol{\varphi}^\top} + o(1),
\end{aligned}$$

by Assumption A.6 and the continuous mapping theorem (Van der Vaart, 2000); the fact that elements of $\mathbf{M}_n(\boldsymbol{\varphi})$, $\partial\mathbf{M}_n(\boldsymbol{\varphi})/\partial\boldsymbol{\varphi}$ and $\partial^2\mathbf{M}_n(\boldsymbol{\varphi})/\partial\boldsymbol{\varphi}\partial\boldsymbol{\varphi}^\top$ are all $O(1)$; and Lemma A.4, A.5, and A.6. Similarly, we also can prove

$$\frac{1}{n}\frac{\partial \log L_n(\boldsymbol{\theta}_0)}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^\top} + o_P(1) = \frac{1}{n}\mathbb{E}\left[\frac{\partial \log L_n(\boldsymbol{\theta}_0)}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^\top}\right] + o_P(1),$$

with

$$\begin{aligned}
\mathbb{E}\left[\frac{\partial^2 \log L_n(\boldsymbol{\theta})}{\partial\boldsymbol{\beta}\partial\boldsymbol{\beta}^\top}\right] &= -\frac{1}{\sigma^2}\mathbf{M}_n(\boldsymbol{\varphi})^\top\mathbf{M}_n(\boldsymbol{\varphi}),\\
\mathbb{E}\left[\frac{\partial^2 \log L_n(\boldsymbol{\theta})}{(\partial\sigma)^2}\right] &= -\frac{n}{2\sigma^4},\\
\mathbb{E}\left[\frac{\partial^2 \log L_n(\boldsymbol{\theta})}{\partial\boldsymbol{\rho}_j\partial\boldsymbol{\rho}_l}\right] &= -\frac{1}{\sigma^2}\boldsymbol{\beta}^\top\mathbf{M}_n(\boldsymbol{\varphi})^\top\mathbf{C}_{ln}(\boldsymbol{\rho})^\top\mathbf{C}_{jn}(\boldsymbol{\rho})\mathbf{M}_n(\boldsymbol{\varphi})\boldsymbol{\beta}\\
&\quad - \text{tr}(\mathbf{C}_{ln}(\boldsymbol{\rho})^\top\mathbf{C}_{jn}(\boldsymbol{\rho})) - \text{tr}(\mathbf{C}_{ln}(\boldsymbol{\rho})\mathbf{C}_{jn}(\boldsymbol{\rho})),\\
\mathbb{E}\left[\frac{\partial^2 \log L_n(\boldsymbol{\theta})}{\partial\boldsymbol{\varphi}\partial\boldsymbol{\varphi}^\top}\right] &= -\frac{1}{\sigma^2}\boldsymbol{\beta}^\top\left(\frac{\partial\mathbf{M}_n(\boldsymbol{\varphi})}{\partial\boldsymbol{\varphi}}\right)^\top\left(\frac{\partial\mathbf{M}_n(\boldsymbol{\varphi})}{\partial\boldsymbol{\varphi}}\right)\boldsymbol{\beta},\\
\mathbb{E}\left[\frac{\partial^2 \log L_n(\boldsymbol{\theta})}{\partial\sigma^2\partial\boldsymbol{\beta}^\top}\right] &= \mathbf{0}_4,
\end{aligned}$$

$$
\begin{aligned}
\mathbb{E}\left[\frac{\partial^2 \log L_n(\boldsymbol{\theta})}{\partial \sigma^2 \partial \boldsymbol{\rho}_j}\right] &= -\frac{1}{\sigma^2}\mathrm{tr}(\mathbf{C}_{jn}), \\
\mathbb{E}\left[\frac{\partial^2 \log L_n(\boldsymbol{\theta})}{\partial \sigma^2 \partial \boldsymbol{\varphi}^\top}\right] &= \mathbf{0}_p, \\
\mathbb{E}\left[\frac{\partial^2 \log L_n(\boldsymbol{\theta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\rho}_j}\right] &= -\frac{1}{\sigma^2}\boldsymbol{\beta}^\top \mathbf{M}_n(\boldsymbol{\varphi})^\top \mathbf{C}_{jn}(\boldsymbol{\rho})^\top \mathbf{M}_n(\boldsymbol{\varphi}), \\
\mathbb{E}\left[\frac{\partial^2 \log L_n(\boldsymbol{\theta})}{\partial \boldsymbol{\varphi} \partial \boldsymbol{\rho}_j}\right] &= -\frac{1}{\sigma^2}\boldsymbol{\beta}^\top \mathbf{M}_n(\boldsymbol{\varphi})^\top C_{jn}(\boldsymbol{\rho})^\top \left(\frac{\partial \mathbf{M}_n(\boldsymbol{\varphi})}{\partial \boldsymbol{\varphi}}\right)\boldsymbol{\beta}, \\
\mathbb{E}\left[\frac{\partial^2 \log L_n(\boldsymbol{\theta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\varphi}^\top}\right] &= -\frac{1}{\sigma^2}\boldsymbol{\beta}^\top \left(\frac{\partial \mathbf{M}_n(\boldsymbol{\varphi})}{\partial \boldsymbol{\varphi}}\right)^\top \mathbf{M}_n(\boldsymbol{\varphi}),
\end{aligned}
$$

where $p$ is the number of parameters in $\boldsymbol{\varphi}$. This completes the proof for (A.3). Finally, from (A.2), (A.3), and Slutsky's theorem, Theorem 2.2 is proved.

## A.2 Additional Simulation Results

### A.2.1 Summary of the Fixed Designs

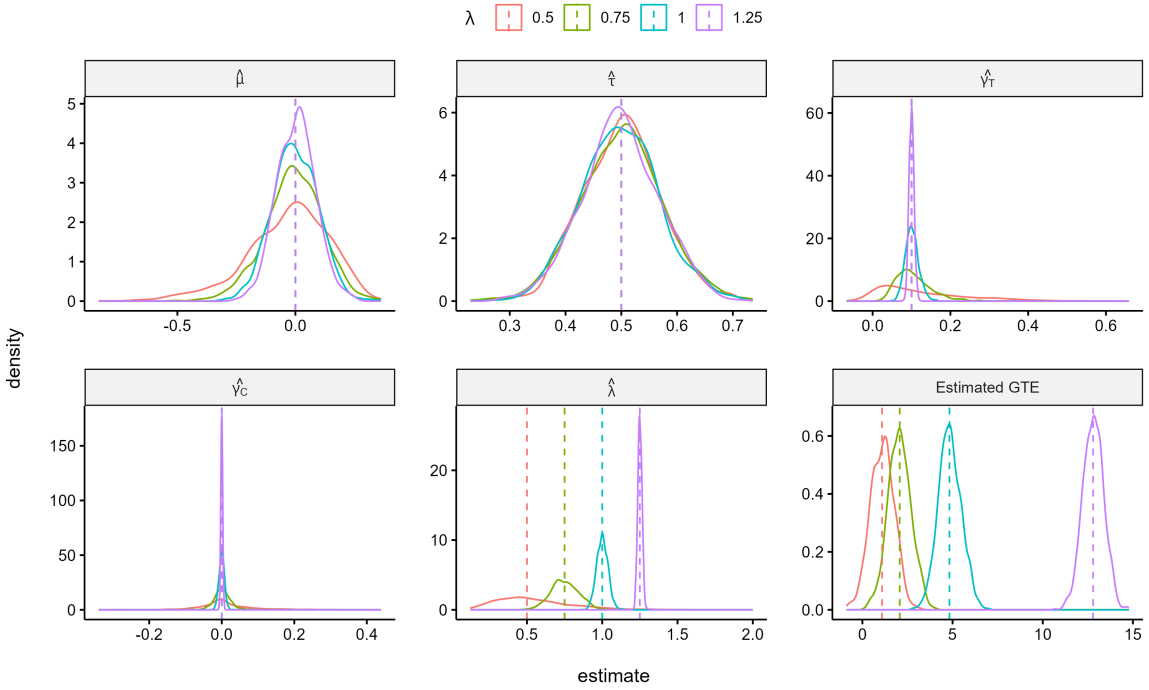
| Summary | Caltech | UMichigan |
|---|---|---|
| Proportion of treated units | 0.50 | 0.499 |
| Difference of average degree between treated and controlled units | 0.42 | -0.92 |
| Proportion of units having a higher proportion of similarly assigned neighbors than expected | 0.49 | 0.48 |

Table A.1: Summary statistics of the fixed designs used in the simulations of Section 2.4.

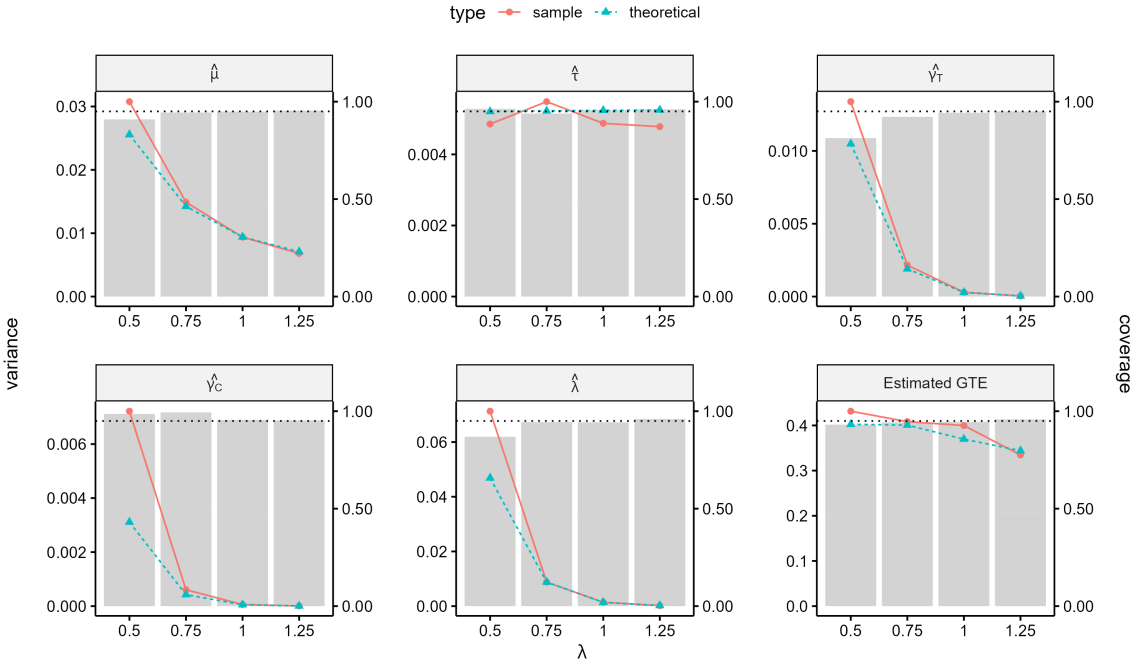
In the simulations of Section 2.4, we fix a design by randomly selecting half of the units in the network and assigning them to treatment. The rest of the units are assigned to control. We provide the summary statistics of the designs that we generated in Table A.1 below. Because we choose balanced randomization for each network, the proportion of treated units in each design is approximately 0.5. Both designs are balanced in terms of degree, since for each design, the difference in average

degree between treated and controlled units is less than 1. Finally, the proportion of units having a higher proportion of similarly assigned neighbors than expected illustrates the clustering of the design. If this number is large, then the majority of units are surrounded by neighbors who are assigned to the same treatment as themselves. We can see that this number is around 0.5 for each design, which implies that there is no clustering. Thus, the designs we generated are not uncommon, and our results are generalizable.

### A.2.2 Additional Simulation Results for Section 2.4.1




(a) The distribution of parameter estimates.




(b) The variances of the estimates (left axes, lines) and coverage rates (right axes, bars).

Figure A.1: Simulation results of the POW-DEG specification on the Caltech Facebook network with $\mu = 0, \tau = 0.5, \gamma_T = 0.1, \gamma_C = 0.0$ and $\lambda \in \{0.5, 0.75, 1.00, 1.25\}$ over 1,000 simulation runs.

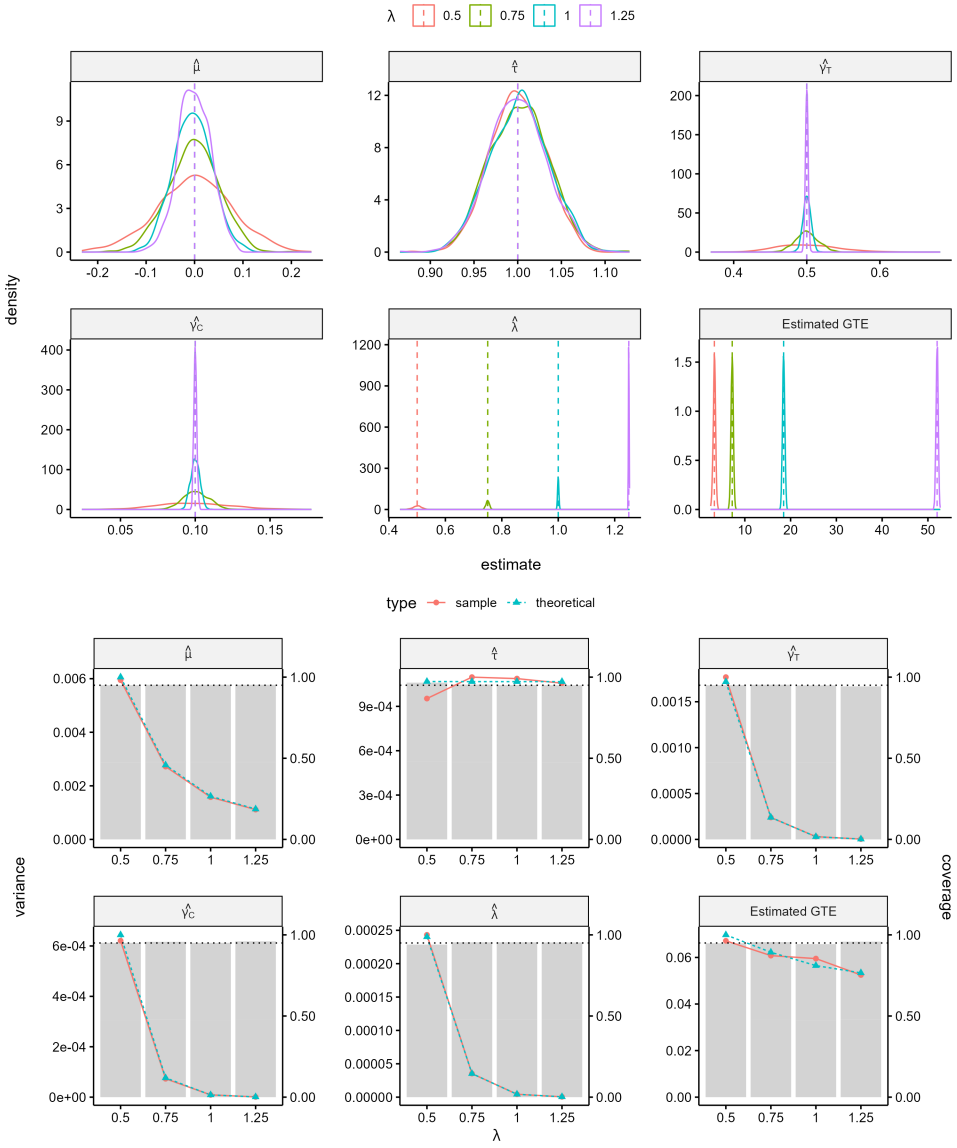



Figure A.2: (upper) The distribution of parameter estimates of the POW-DEG specification on the UMichigan Facebook network with $\mu = 0, \tau = 1.0, \gamma_T = 0.5, \gamma_C = 0.1$ and $\lambda \in \{0.5, 0.75, 1.00, 1.25\}$ over 1,000 simulation runs. (lower) The corresponding variances of the estimates (left axes, lines) and coverage rates (right axes, bars).

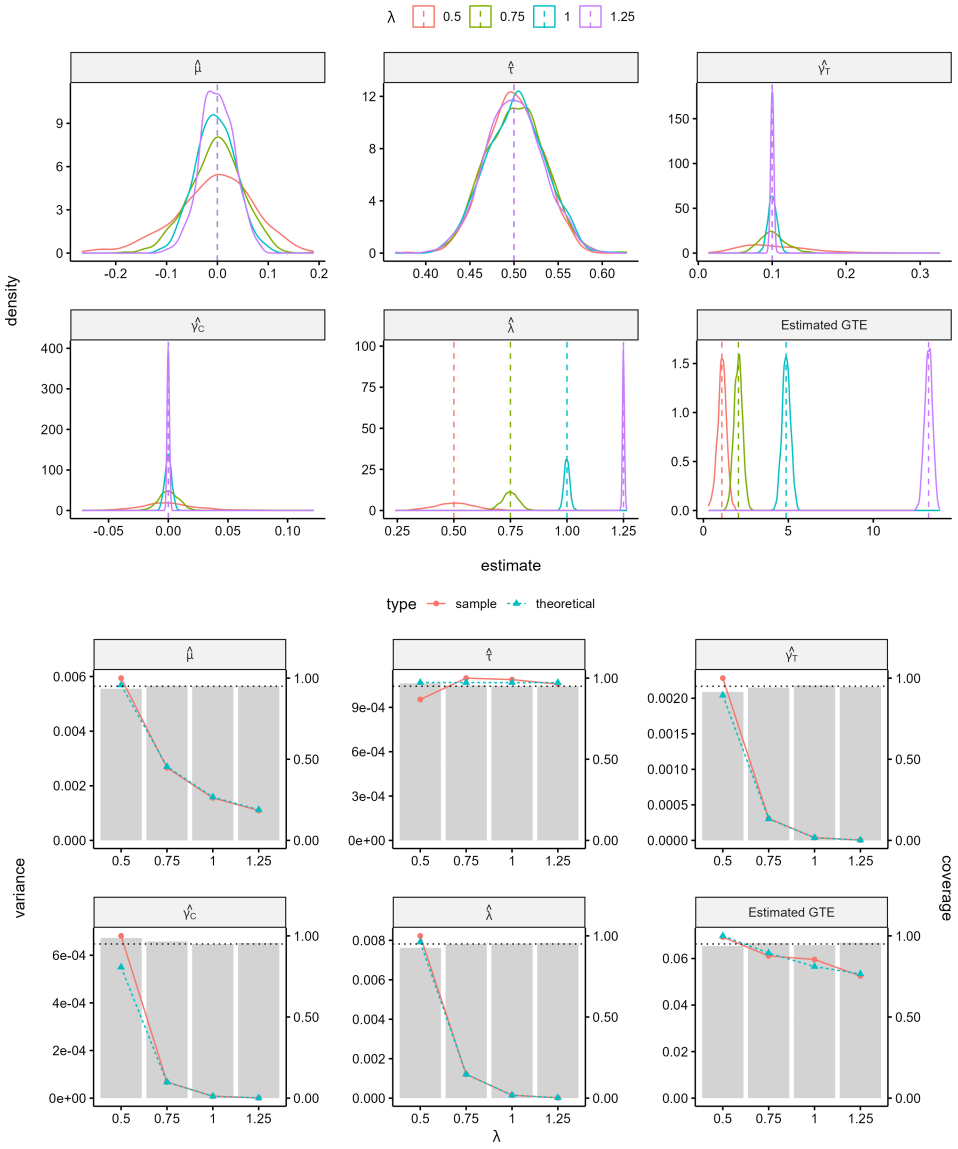



Figure A.3: (upper) The distribution of parameter estimates of the POW-DEG specification on the UMichigan Facebook network with $\mu = 0, \tau = 0.5, \gamma_T = 0.1, \gamma_C = 0.0$ and $\lambda \in \{0.5, 0.75, 1.00, 1.25\}$ over 1,000 simulation runs. (lower) The corresponding variances of the estimates (left axes, lines) and coverage rates (right axes, bars).

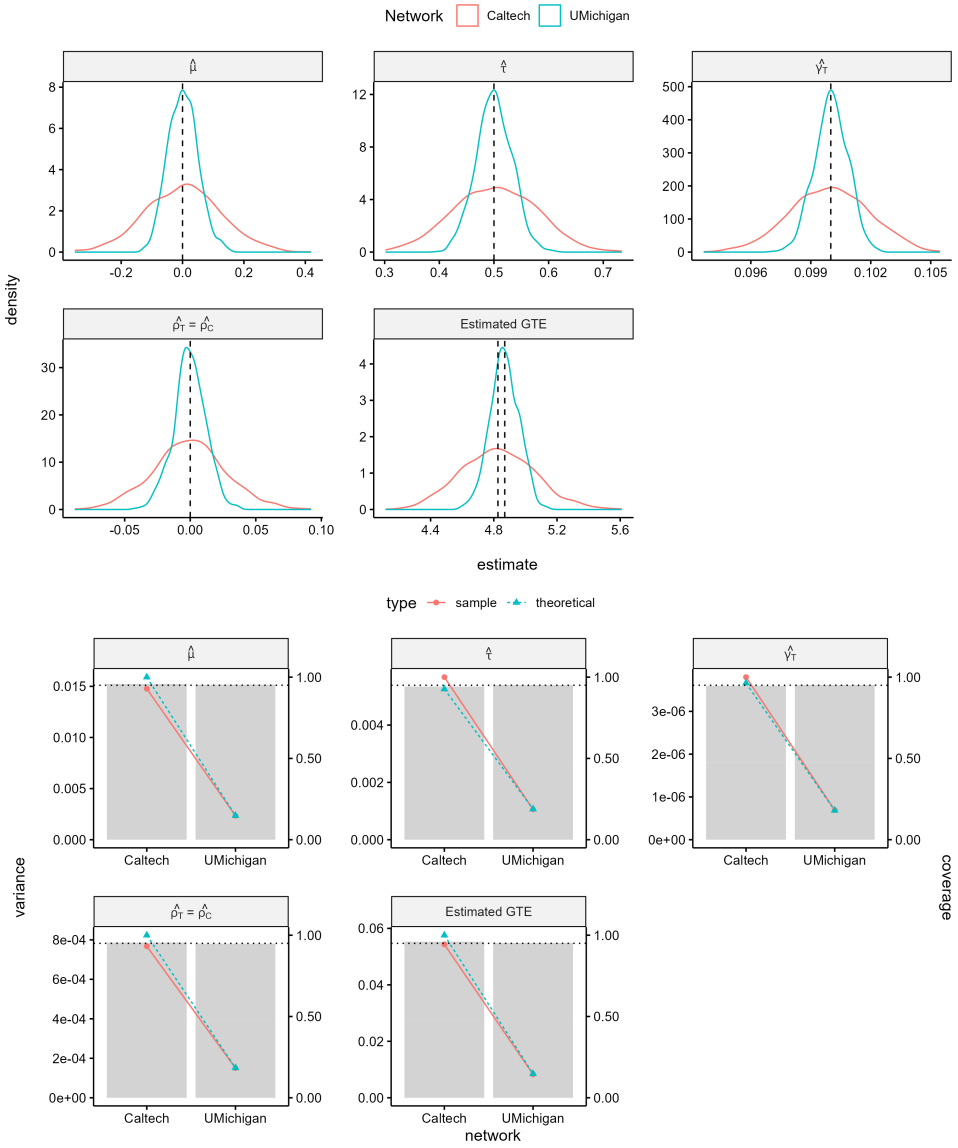



Figure A.4: (upper) The distribution of parameter estimates of the HOM specification with $\mu = 0, \tau = 0.5, \gamma_T = 0.1, \rho_T = \rho_C = 0.0$ over 1,000 simulation runs. (lower) The corresponding variances of the estimates (left axes, lines) and coverage rates (right axes, bars).

### A.2.3 Additional Simulation Results for Section 2.4.2

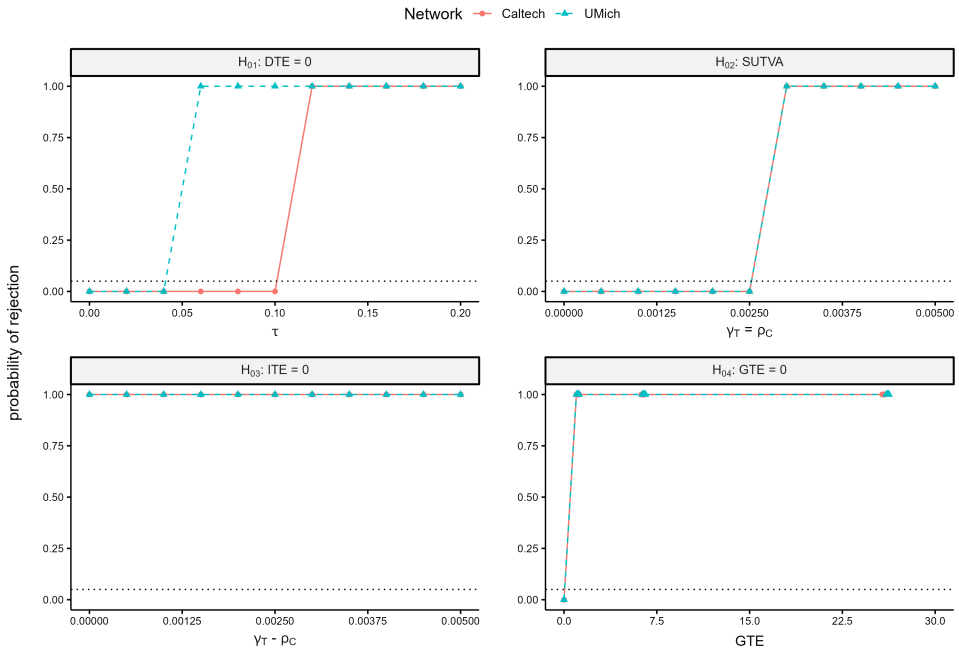



Figure A.5: Rejection rates of hypothesis tests for HOM specification on the Caltech and UMichigan Facebook networks with varying parameters.

# Appendix B

# Appendices for Chapter 3

## B.1 Variance Derivation for the MLE

### B.1.1 Asymptotic Variances

The information matrix is

$$\mathcal{I}(\boldsymbol{\theta}) = \begin{bmatrix} -\frac{\partial^2 \ell}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^\top} & -\frac{\partial^2 \ell}{\partial \boldsymbol{\beta} \partial \boldsymbol{\varphi}^\top} \\ -\frac{\partial^2 \ell}{\partial \boldsymbol{\varphi} \partial \boldsymbol{\beta}^\top} & -\frac{\partial^2 \ell}{\partial \boldsymbol{\varphi} \partial \boldsymbol{\varphi}^\top} \end{bmatrix}.$$

From Model (3.1) and the log-likelihood in (3.2), elements of matrix $\mathcal{I}(\boldsymbol{\theta})$ can be derived with

$$\begin{aligned} \frac{\partial^2 \ell}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^\top} &= \sum_{i=1}^{n} \left\{ \left( \frac{\partial^2 \ell}{\partial p_i^2} \right) \left( \frac{\partial p_i}{\partial \boldsymbol{\beta}} \right) \left( \frac{\partial p_i}{\partial \boldsymbol{\beta}} \right)^\top + \left( \frac{\partial \ell}{\partial p_i} \right) \left( \frac{\partial^2 p_i}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^\top} \right) \right\}, \\ \frac{\partial^2 \ell}{\partial \boldsymbol{\varphi} \partial \boldsymbol{\varphi}^\top} &= \sum_{i=1}^{n} \left\{ \left( \frac{\partial^2 \ell}{\partial p_i^2} \right) \left( \frac{\partial p_i}{\partial \boldsymbol{\varphi}} \right) \left( \frac{\partial p_i}{\partial \boldsymbol{\varphi}} \right)^\top + \left( \frac{\partial \ell}{\partial p_i} \right) \left( \frac{\partial^2 p_i}{\partial \boldsymbol{\varphi} \partial \boldsymbol{\varphi}^\top} \right) \right\}, \\ \left( \frac{\partial^2 \ell}{\partial \boldsymbol{\varphi} \partial \boldsymbol{\beta}^\top} \right)^\top = \frac{\partial^2 \ell}{\partial \boldsymbol{\beta} \partial \boldsymbol{\varphi}^\top} &= \sum_{i=1}^{n} \left\{ \left( \frac{\partial^2 \ell}{\partial p_i^2} \right) \left( \frac{\partial p_i}{\partial \boldsymbol{\beta}} \right) \left( \frac{\partial p_i}{\partial \boldsymbol{\varphi}} \right)^\top + \left( \frac{\partial \ell}{\partial p_i} \right) \left( \frac{\partial^2 p_i}{\partial \boldsymbol{\beta} \partial \boldsymbol{\varphi}^\top} \right) \right\}, \end{aligned} \tag{B.1}$$

where

$$\frac{\partial \ell}{\partial p_i} = \frac{Y_i}{p_i} - \frac{1 - Y_i}{1 - p_i},$$

$$
\begin{aligned}
\frac{\partial^2 \ell}{\partial p_i^2} &= -\frac{Y_i}{p_i^2} - \frac{(1-Y_i)}{(1-p_i)^2}, \\
\frac{\partial p_i}{\partial \boldsymbol{\beta}} &= h'\Big(\mathbf{m}_i(\boldsymbol{\varphi})^\top \boldsymbol{\beta}\Big)\mathbf{m}_i(\boldsymbol{\varphi}), \\
\frac{\partial p_i}{\partial \boldsymbol{\varphi}} &= h'\Big(\mathbf{m}_i(\boldsymbol{\varphi})^\top \boldsymbol{\beta}\Big)\frac{\partial \mathbf{m}_i(\boldsymbol{\varphi})}{\partial \boldsymbol{\varphi}}\boldsymbol{\beta}, \\
\frac{\partial^2 p_i}{\partial \boldsymbol{\beta}\partial \boldsymbol{\beta}^\top} &= h''\Big(\mathbf{m}_i(\boldsymbol{\varphi})^\top \boldsymbol{\beta}\Big)\mathbf{m}_i(\boldsymbol{\varphi})\mathbf{m}_i(\boldsymbol{\varphi})^\top, \\
\frac{\partial^2 p_i}{\partial \boldsymbol{\varphi}\partial \boldsymbol{\varphi}^\top} &= h''\Big(\mathbf{m}_i(\boldsymbol{\varphi})^\top \boldsymbol{\beta}\Big)\frac{\partial \mathbf{m}_i(\boldsymbol{\varphi})}{\partial \boldsymbol{\varphi}}\boldsymbol{\beta}\boldsymbol{\beta}^\top\left(\frac{\partial \mathbf{m}_i(\boldsymbol{\varphi})}{\partial \boldsymbol{\varphi}}\right)^\top + h'\Big(\mathbf{m}_i(\boldsymbol{\varphi})^\top \boldsymbol{\beta}\Big)\frac{\partial^2 \mathbf{m}_i(\boldsymbol{\varphi})}{\partial \boldsymbol{\varphi}\partial \boldsymbol{\varphi}^\top}\boldsymbol{\beta}, \\
\frac{\partial^2 p_i}{\partial \boldsymbol{\beta}\partial \boldsymbol{\varphi}^\top} &= h''\Big(\mathbf{m}_i(\boldsymbol{\varphi})^\top \boldsymbol{\beta}\Big)\mathbf{m}_i(\boldsymbol{\varphi})\boldsymbol{\beta}^\top\left(\frac{\partial \mathbf{m}_i(\boldsymbol{\varphi})}{\partial \boldsymbol{\varphi}}\right)^\top + h'\Big(\mathbf{m}_i(\boldsymbol{\varphi})^\top \boldsymbol{\beta}\Big)\left(\frac{\partial \mathbf{m}_i(\boldsymbol{\varphi})}{\partial \boldsymbol{\varphi}}\right)^\top.
\end{aligned}
$$

In the above derivation, we use the denominator layout. Since the regressors in Model (3.1) are considered fixed, to obtain the Fisher information matrix $J(\boldsymbol{\theta}) = \mathbb{E}[\mathcal{I}(\boldsymbol{\theta})]$, we only need to plug in the expectations

$$
\begin{aligned}
\mathbb{E}\left[\frac{\partial \ell}{\partial p_i}\right] &= 0, \\
\mathbb{E}\left[\frac{\partial^2 \ell}{\partial p_i^2}\right] &= -\frac{1}{p_i} - \frac{1}{1-p_i},
\end{aligned}
$$

in place of $\partial \ell / \partial p_i$ and $\partial^2 \ell / \partial p_i^2$ respectively in the derivative formulas (B.1), since $\mathbb{E}[Y_i] = p_i$.

### B.1.2 Robust Clustered Variances

Under regularity conditions, maximizing the log-likelihood in (3.2) is equivalent to solving the score equation

$$
\frac{\partial \ell}{\partial \boldsymbol{\theta}} = \begin{pmatrix} \frac{\partial \ell}{\partial \boldsymbol{\beta}} \\ \frac{\partial \ell}{\partial \boldsymbol{\varphi}} \end{pmatrix} = \sum_{i=1}^{n} \begin{pmatrix} \frac{\partial \ell}{\partial p_i}\frac{\partial p_i}{\partial \boldsymbol{\beta}} \\ \frac{\partial \ell}{\partial p_i}\frac{\partial p_i}{\partial \boldsymbol{\varphi}} \end{pmatrix} = \mathbf{0}. \tag{B.2}
$$

Thus, the asymptotic variance of the solution $\hat{\boldsymbol{\theta}}$ to the score equation (B.2) is given by (White, 1996)

$$
\left(\mathbb{E}\left[-\frac{\partial^2 \ell}{\partial \boldsymbol{\theta}\partial \boldsymbol{\theta}^\top}\right]\right)^{-1} \operatorname{Var}\left[\frac{\partial \ell}{\partial \boldsymbol{\theta}}\right] \left(\mathbb{E}\left[-\frac{\partial^2 \ell}{\partial \boldsymbol{\theta}\partial \boldsymbol{\theta}^\top}\right]\right)^{-1}. \tag{B.3}
$$

This reduces to (3.3) if the model is correctly specified and $\ell$ is the true log-likelihood of the data. Under model misspecification, we can use (B.3) to estimate the variance of $\hat{\boldsymbol{\theta}}$. In particular, we can estimate

$$\mathbb{E}\left[-\frac{\partial^2 \ell}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^\top}\right] \quad \text{by} \quad J(\hat{\boldsymbol{\theta}}),$$

with $J(\boldsymbol{\theta})$ given in Appendix B.1.1. The variance of $\partial \ell / \partial \boldsymbol{\theta}$ can be estimated by

$$\sum_{i=1}^{n} \begin{pmatrix} \frac{\partial \ell}{\partial p_i} \frac{\partial p_i}{\partial \boldsymbol{\beta}} \\ \frac{\partial \ell}{\partial p_i} \frac{\partial p_i}{\partial \boldsymbol{\varphi}} \end{pmatrix} \begin{pmatrix} \frac{\partial \ell}{\partial p_i} \frac{\partial p_i}{\partial \boldsymbol{\beta}} \\ \frac{\partial \ell}{\partial p_i} \frac{\partial p_i}{\partial \boldsymbol{\varphi}} \end{pmatrix}^\top.$$

If the data has a grouping structure, i.e., if units $1, ..., n$ can be partitioned into $G$ group, then the variance of $\partial \ell / \partial \boldsymbol{\theta}$ can be estimated by

$$\sum_{g=1}^{G} \left( \sum_{i \in g} \begin{pmatrix} \frac{\partial \ell}{\partial p_i} \frac{\partial p_i}{\partial \boldsymbol{\beta}} \\ \frac{\partial \ell}{\partial p_i} \frac{\partial p_i}{\partial \boldsymbol{\varphi}} \end{pmatrix} \right) \left( \sum_{i \in g} \begin{pmatrix} \frac{\partial \ell}{\partial p_i} \frac{\partial p_i}{\partial \boldsymbol{\beta}} \\ \frac{\partial \ell}{\partial p_i} \frac{\partial p_i}{\partial \boldsymbol{\varphi}} \end{pmatrix} \right)^\top. \tag{B.4}$$

Then the variance estimator is called the robust clustered variance estimator (Freedman, 2006; Cameron and Miller, 2015). Other types of robust clustered variance can be found in Zeileis et al. (2020).

## B.2 Additional Simulation Results

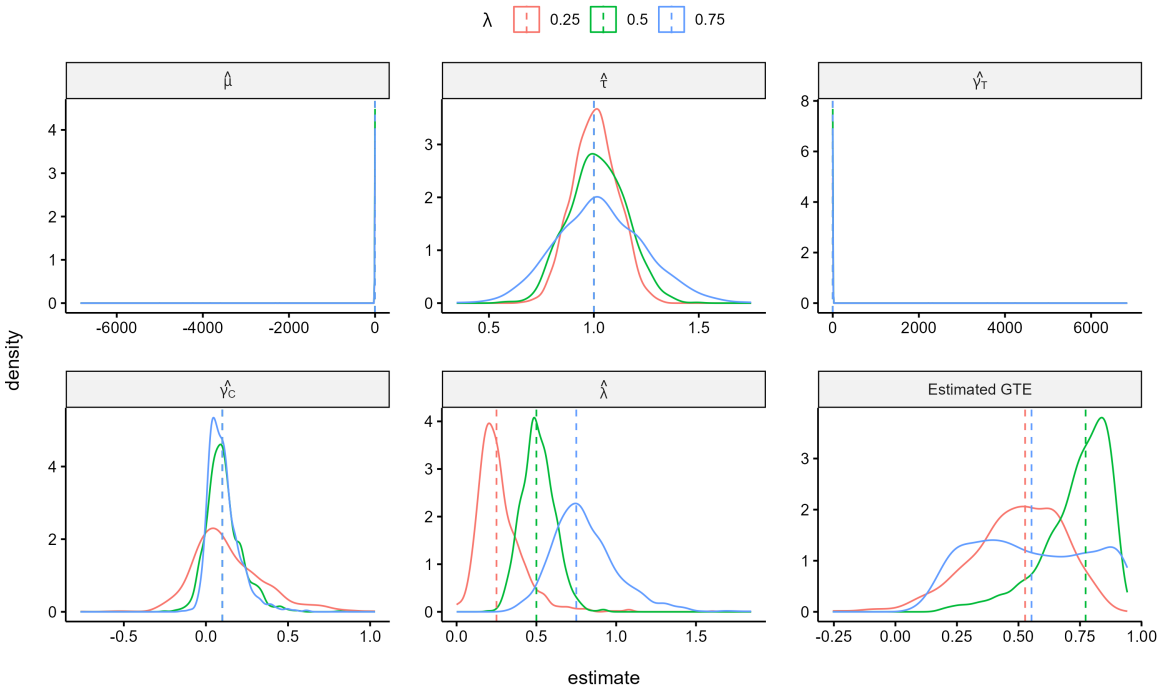
### B.2.1 Summary of the Fixed Designs

In the simulations of Section 3.2, we fix a design by randomly selecting and assigning half of the units to treatment and the other half to control. Table B.1 below shows the summary statistics of the designs. We use the same summary statistics as in Table A.1, which is described in Appendix A.2.1. We can see that the designs are balanced in terms of allocation and degree, and they do not show any notable clustering behavior. This means that the designs we generated are not typically rare and disproportionate, and so the results should be generalizable.

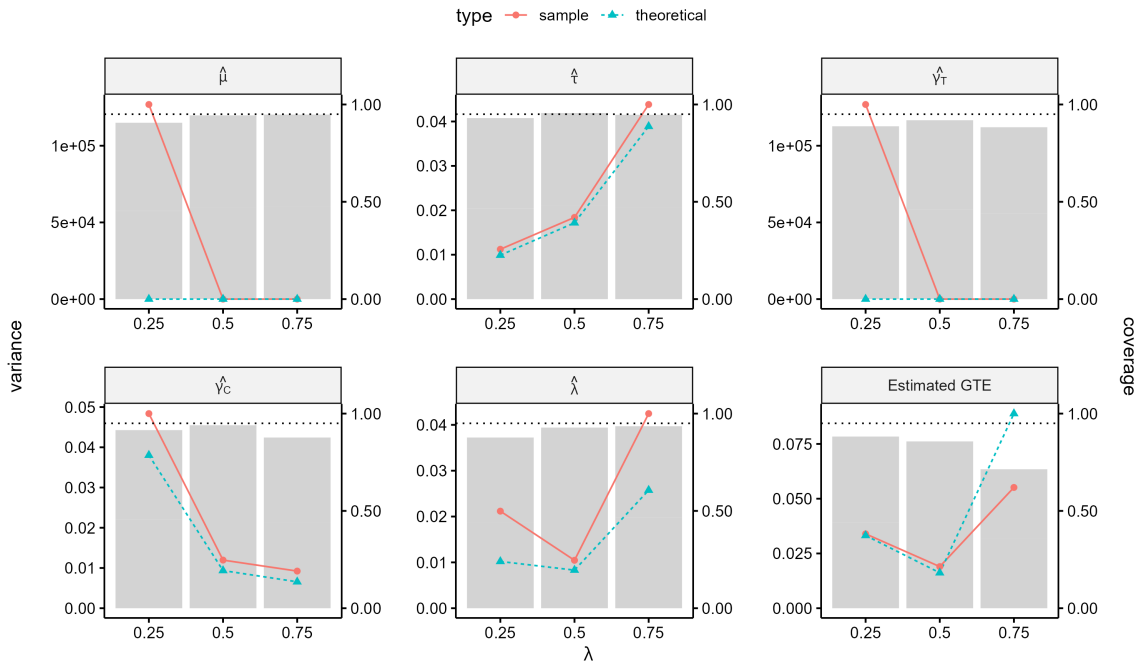
| Summary | Caltech | UMichigan |
|---|---|---|
| Proportion of treated units | 0.50 | 0.499 |
| Difference of average degree between treated and controlled units | 2.78 | 1.46 |
| Proportion of units having a higher proportion of similarly assigned neighbors than expected | 0.44 | 0.51 |

Table B.1: Summary statistics of the fixed designs used in the simulations of Section 3.2.

### B.2.2 Simulation Results on Distributions of Estimates for the Probit Model




(a) The distribution of parameter estimates.




(b) The variances of the estimates (left axes, lines) and coverage rates (right axes, bars).

Figure B.1: Simulation results for the POW-DEG specification of Model (3.1) with probit link on the Caltech network over 1,000 runs.

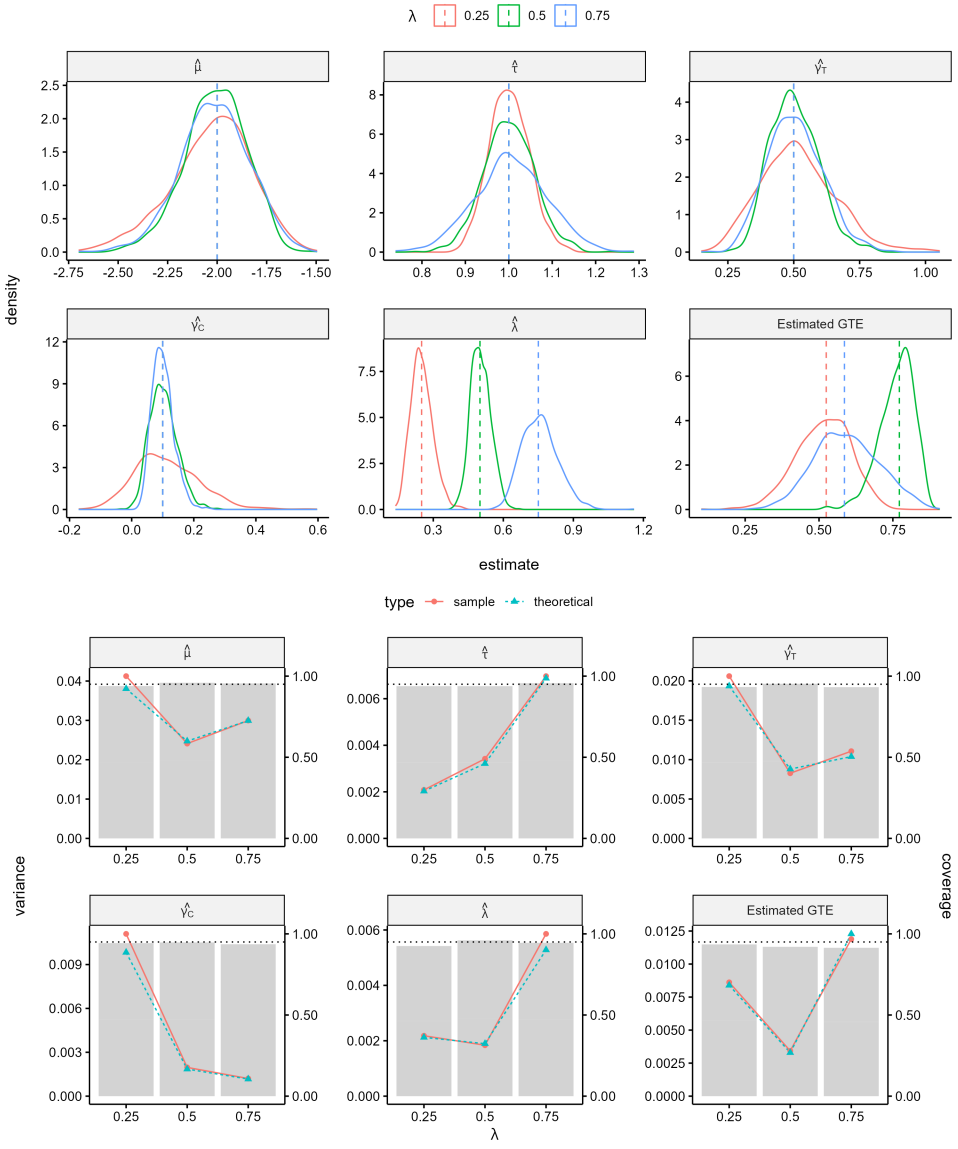
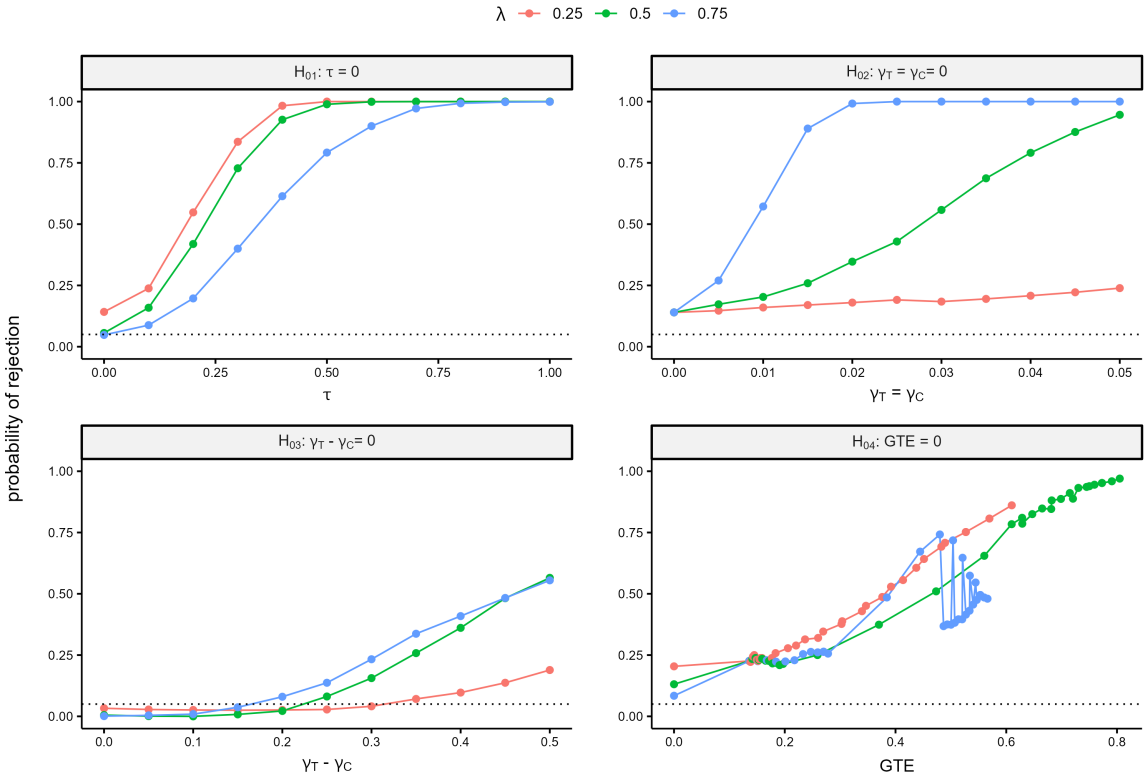
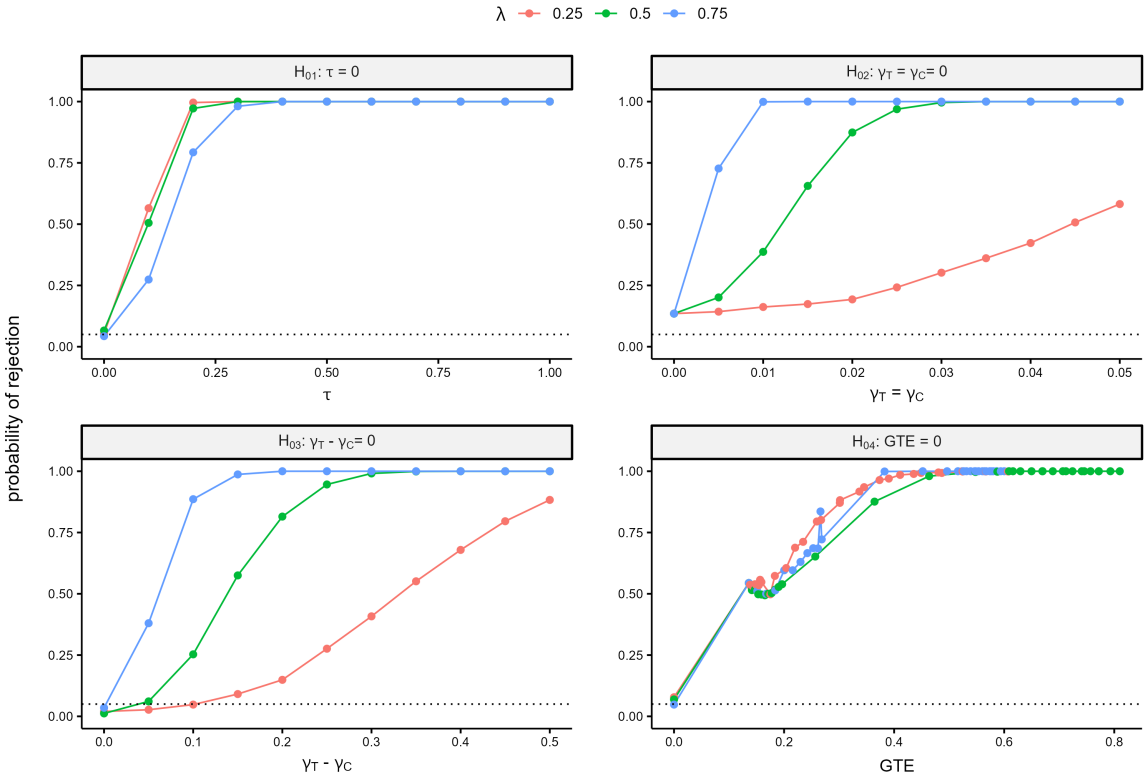



Figure B.2: (Upper) Distribution of estimates (upper), (Lower) variances and coverage rates for the POW-DEG specification of Model (3.1) with probit link on the UMichigan network over 1,000 runs.

### B.2.3 Simulation Results on Hypothesis Testing for the Probit Model




(a) Results for the Caltech network.




(b) Results for the UMichigan network.

Figure B.3: Rejection rates of hypothesis tests for POW-DEG specification of Model (3.1) with probit link and varying parameters.

# Appendix C

# Appendices for Chapter 4

## C.1 Comparison among Balanced Graph Clustering Algorithms

The clustering algorithm used in our simulations is the balanced label propagation (BLP), proposed early by Gui et al. (2015). Although it is able to produce balanced clusterings with high modularities, the algorithm is criticized for being too slow due to its exhaustive greedy step, which keeps exchanging cluster labels until modularity can no longer increase (Saveski et al., 2017). There have been other algorithms proposed in the literature, for example, the social hash algorithm for bipartite graphs (Shalita et al., 2016), and the restreaming linear deterministic greedy (reLDG) algorithm (Saveski et al., 2017). These algorithms sacrifice the high modularity and perfect balance from the BLP for a faster running time. In order to justify the use of BLP in Section 4.3, we run the two algorithms (each for 30 iterations), along with the BLP (Gui et al., 2015) and the Louvain imbalanced clustering (Blondel et al., 2008) algorithms on the Enron, Caltech, and UMichigan networks. We evaluate these algorithms based on (i) modularity (the higher the better the clustering), (ii) range of cluster sizes (the smaller, the more balanced the clustering), and (iii) running time. Table C.1 shows the mean performance of the four clustering algorithms over 30 runs.

According to the results, we choose the BLP algorithm as it returns clusterings with high modularities and perfect balance on our networks. This helps us investigate how graph cluster randomization behaves in an ideal situation. However, it is also

| Algorithm | Modularity | | | Cluster Size Range | | | Running Time | | |
|---|---|---|---|---|---|---|---|---|---|
| | Enron | Caltech | UMich | Enron | Caltech | UMich | Enron | Caltech | UMich |
| Louvain | 0.3492 | 0.3981 | 0.3964 | 33 | 158 | 644 | 00:00:00 | 00:00:00 | 00:00:00 |
| BLP | 0.3027 | 0.3445 | 0.3707 | 1 | 1 | 1 | 00:05:34 | 01:12:23 | 23:24:51 |
| reLDG | 0.0837 | 0.1273 | 0.1016 | 1.83 | 2.23 | 1.97 | 00:00:24 | 00:00:02 | 00:06:01 |
| Social hash | 0.1983 | 0.2100 | 0.2985 | 10.27 | 19.0 | 50.1 | 00:00:02 | 00:00:50 | 00:14:36 |

Table C.1: Performance of different clustering algorithms. Each entry is the average value over 30 runs.

notable that the algorithm does not scale well. The social hash algorithm seems to be a good alternative when the network is large.

## C.2 Convergence of Monte-Carlo Approximations

Our optimal design problems require Monte Carlo approximation in Equation (4.10) where $L$ is to be specified. Note that the Monte Carlo approximation will be called to evaluate any new design that our search points to. Therefore, the number of draws $L$ in Equation (4.10) needs to be small enough for computational feasibility. On the other hand, it also needs to be large enough for the approximation to be stable and precise. In order to find the number of draws $L$ for our simulations, we run a simulation study in which the design criterion $\phi(\mathbf{Z})$ is evaluated for a randomly generated design $\mathbf{Z}$. For each model in Section 4.3.1 and each of the two networks Enron and Caltech, we run the Monte Carlo approximation 10 times, each with 50,000 draws of parameters from corresponding prior distributions specified in Table 7. The results are shown in Figure C.1 with each line representing a run. We find that $L = 5,000$ (vertical dashed line) seems to give convergent results over all models.

## C.3 Running Times of the Algorithms Considered

In our simulations, we use parallel computing where the Monte Carlo approximations are distributed over 8 cores. Our results are obtained by running the algorithms on Linux servers with the following configurations: (i) Model: Dell PowerEdge R840; (ii) CPUs: four Intel Xeon Gold 6230 20-core 2.1 GHz (Cascade Lake); and Memory: 768 GB. We do not have control over the traffic of users on the servers, hence, recorded

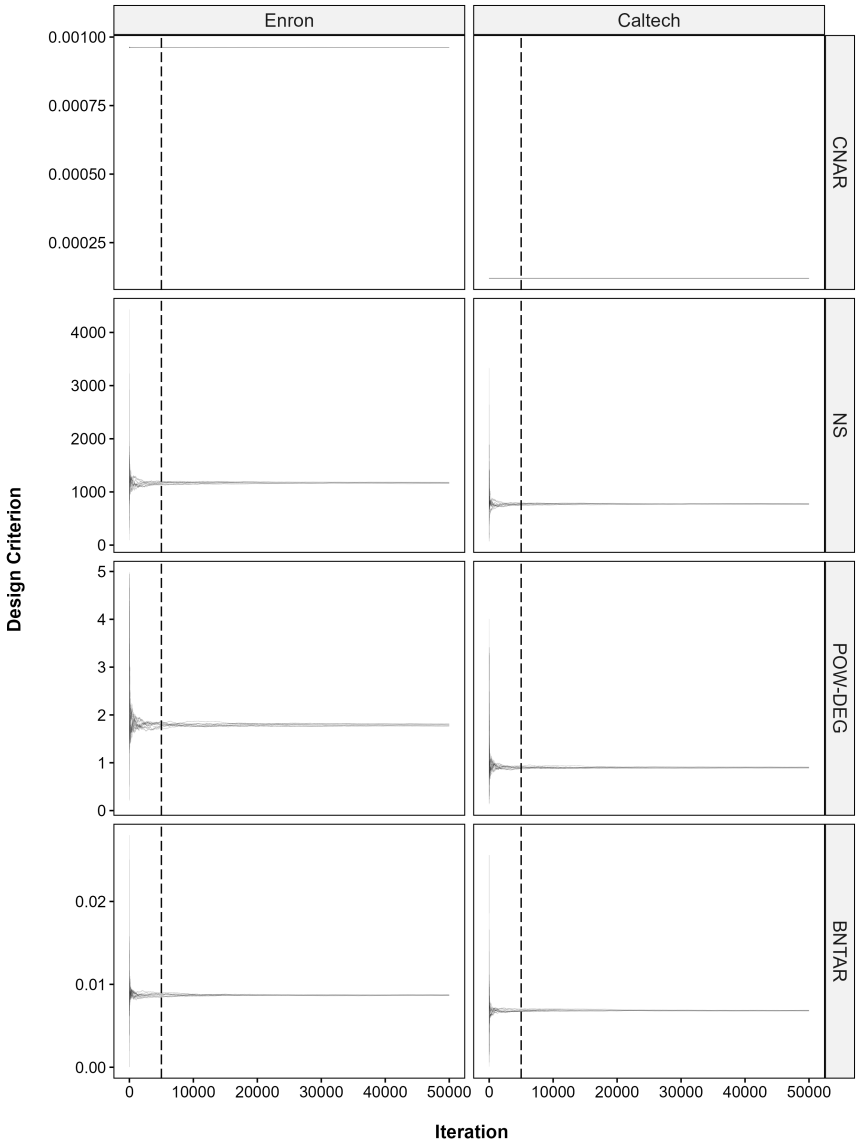



Figure C.1: Results of simulations investigating the number of draws $L$ in Monte Carlo approximations.

running times may differ from the case where algorithms are run on independent machines. We give the average running times for each algorithm and model in Table C.2. Although there can be uncertainties associated with these numbers, Table C.2 still provides us some understanding of how different algorithms compare and scale.

| Algorithms | CNAR model | | NS model | | POW-DEG model | | BNTAR model | |
|---|---|---|---|---|---|---|---|---|
| | Enron | Caltech | Enron | Caltech | Enron | Caltech | Enron | Caltech |
| Imbalanced cluster randomization | 00:00:00 | 00:00:00 | 00:00:00 | 00:00:00 | 00:00:00 | 00:00:00 | 00:00:00 | 00:00:00 |
| Balanced cluster randomization | 00:05:34 | 01:12:23 | 00:05:34 | 01:12:23 | 00:05:34 | 01:12:23 | 00:05:34 | 01:12:23 |
| Balanced randomization | 00:00:00 | 00:00:00 | 00:00:00 | 00:00:00 | 00:00:00 | 00:00:00 | 00:00:00 | 00:00:00 |
| Tree-Parzen | 00:23:06 | 01:24:51 | 00:36:55 | 02:01:29 | 01:21:27 | 03:00:25 | 01:29:36 | 03:21:25 |
| Surrogate-based local search | 05:13:14 | 06:41:23 | 05:03:21 | 06:19:30 | 05:59:08 | 07:37:19 | 06:56:08 | 08:18:19 |
| Reinforcement learning | 07:48:59 | 12:17:39 | 09:03:24 | 11:48:37 | 08:17:50 | 12:00:02 | 09:28:58 | 13:55:49 |
| Random search | 00:12:47 | 00:46:21 | 00:18:29 | 01:10:07 | 00:49:29 | 01:26:15 | 01:03:25 | 01:48:20 |
| Tabu search | 00:13:03 | 00:46:59 | 00:18:26 | 01:10:53 | 00:50:38 | 01:27:05 | 01:02:50 | 01:47:52 |
| Simulated annealing | 00:12:52 | 00:46:32 | 00:18:37 | 01:10:36 | 00:51:04 | 01:26:22 | 00:59:46 | 01:48:35 |
| Genetic algorithm | 00:11:48 | 00:43:02 | 00:16:42 | 01:01:58 | 00:43:40 | 01:23:04 | 00:57:12 | 01:38:17 |

Table C.2: The running times of each algorithm averaged over 30 runs.

## C.4 Other Design Characteristics

In addition to the design characteristics discussed in Section 4.3.3.2, we investigate two more characteristics: (i) the percentage of treated units and (ii) the difference in average closeness between treated and controlled units. First, the percentage of treated units demonstrates whether a design is balanced. A balanced design will have equal allocation among treatment and control, resulting in a value of 0.5 for panels in the first column of Figure C.2. Second, we also consider balance in terms of closeness. The closeness of a unit is measured by the inverse of the average distance between the unit to every other unit in the network. Similar to betweenness, closeness is a measure of centrality within a network. The results in Figure C.2 do not show any clear pattern where most designs are balanced in terms of both allocation and closeness.

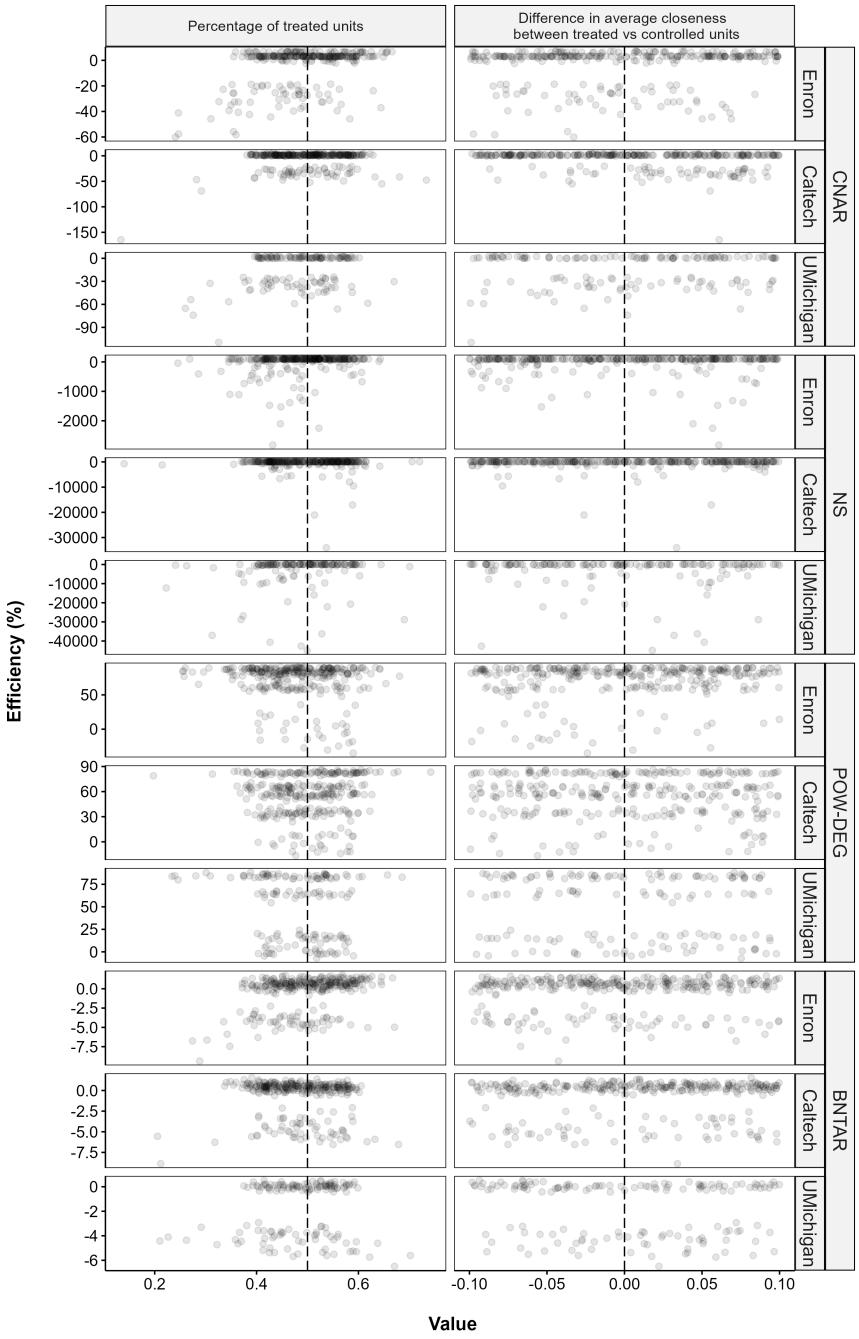



Figure C.2: Additional design characteristics results.