EFFICIENT ALGORITHMS FOR LEARNING COMBINATORIAL STRUCTURES FROM LIMITED DATA

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Asish Ghoshal

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

May 2019

Purdue University

West Lafayette, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF DISSERTATION APPROVAL

Dr. Jean Honorio, Chair

Department of Computer Science, Purdue University

Dr. Dan Goldwasser

Department of Computer Science, Purdue University

Dr. Elena Grigorescu

Department of Computer Science, Purdue University

Dr. Tommi Jaakkola

Department of Computer Science, Massachusetts Institute of Technology

Dr. Jennifer Neville

Department of Computer Science, Purdue University

Approved by:

Dr. Voicu Popescu

Graduate Committee Chair, Department of Computer Science, Purdue University

ACKNOWLEDGMENTS

First and foremost, I acknowledge the starring role played by my advisor Prof. Jean Honorio in the materialization of the thesis. I have not looked back since starting research with Jean in the middle of my third year at Purdue. His technical expertise, knowledge of the literature, and unique perspective on problems have ensured my productivity throughout my time with him. Beside research, he has been an excellent mentor in giving his inputs on matters relating to career, health, and life in general. I owe a great deal to him for my professional and academic development.

I am also thankful to my Ph.D. committee members, especially Prof. Jennifer Neville, for their sharp insights and valuable feedback on the dissertation.

The computer science department at Purdue has been a second home of sorts during my time here. I am thankful to the department leadership for providing a stimulating academic and learning environment, their financial support through various fellowships, and providing access to various computing and other resources that have been indispensable in my research. I am also grateful to the CS support staff who have been very diligent on various matters relating to travel and funding.

Surviving six Midwest winters would have been impossible had it not been for a strong community of friends that I have been fortunate enough to have at Purdue. Gaurav, Romila, Saurav, Pinaki, Miku, Prateek, and Aparajita have been like a loving family over the years. The numerous potlucks and Chai get-togethers that we have organized have been some of the most fun times at Purdue — something that I will fondly reminisce over the coming years. Akash, Abhiram, Rohit, Vikram, Mayank, GV, and many other friends in the CS department have ensured that I never had to endure a dull moment both inside and outside of the department. They have been instantly reachable for mindless and passionate discussions on everything under the sun including but not limited to: AI, politics, cricket, and religion. I would also like to

thank Adarsh, Kevin, and Chuyang and other members of our research group under Prof. Honorio for their time and feedback on mock presentations before conferences. Lastly, I would like to thank Akshay and Shivaram for giving me the opportunity to organize weekly Yoga sessions as part of the Purdue Hindu YUVA group, and in the process opening my mind to experiences and views that have significantly changed my life.

I will forever be indebted to my parents and sister for their love, encouragement, and support that has kept me going in the toughest of situations. I am especially thankful to my parents who have worked doubly hard in ensuring that I (along with my sister) received the best of education. Growing up at a time and place in India where we had no access to affordable quality education, my parents made our education the single most important purpose of their life. To say that I would be nowhere without their support is a huge understatement.

Lastly, I would like to thank my wife Shraddha who has been my rock in both the most turbulent and serene phases of my life at Purdue. Being a graduate student meant I spent many hours over weekends and holidays working on research — I am grateful to her for putting up with me during such times. I am thankful to her for being there through thick and thin and when I have been at my most vulnerable self. Her near constant chatter at home has made life tremendously fun and also increased my tolerance for annoyance.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Ghoshal, Asish PhD, Purdue University, May 2019. Efficient Algorithms for Learning Combinatorial Structures from Limited Data. Major Professor: Jean Honorio.

Recovering combinatorial structures from noisy observations is a recurrent problem in many application domains, including, but not limited to, natural language processing, computer vision, genetics, health care, and automation. For instance, dependency parsing in natural language processing entails recovering parse trees from sentences which are inherently ambiguous. From a computational standpoint, such problems are typically intractable and call for designing efficient approximation or randomized algorithms with provable guarantees. From a statistical standpoint, algorithms that recover the desired structure using an optimal number of samples are of paramount importance.

We tackle several such problems in this thesis and obtain computationally and statistically efficient procedures. We demonstrate optimality of our methods by proving fundamental lower bounds on the number of samples needed by any method for recovering the desired structures. Specifically, the thesis makes the following contributions:

(i) We develop polynomial-time algorithms for learning linear structural equation models — which are a widely used class of models for performing causal inference — that recover the correct directed acyclic graph structure under identifiability conditions that are weaker than existing conditions. We also show that the sample complexity of our method is information-theoretically optimal.

(ii) We develop polynomial-time algorithms for learning the underlying graphical game from observations of the behavior of self-interested agents. The key combinatorial problem here is to recover the Nash equilibria set of the true game

from behavioral data. We obtain fundamental lower bounds on the number of samples required for learning games and show that our method is statistically optimal.

(iii) Lastly, departing from the generative model framework, we consider the problem of structured prediction where the goal is to learn predictors from data that predict complex structured objects directly from a given input. We develop efficient learning algorithms that learn structured predictors by approximating the partition function and obtain generalization guarantees for our method. We demonstrate that randomization can not only improve efficiency but also generalization to unseen data.

1 INTRODUCTION

1.1 Background and Motivation

Machine learning is beginning to play a central role in modern society, where machine learning techniques are not only driving progress in a multitude of domains, for instance, health care, genetics, automation, and transportation, to name a few, but are also playing the role of automated arbiters in systems affecting law and order, national security, and social justice. From a computational standpoint, many of the problems that are being tackled using machine learning techniques, are *intractable* in the sense that they do not admit polynomial time algorithms that can solve all instances of the problem. Furthermore, learning algorithms that power much of these systems often have to simultaneously contend with both limited amount of data: for instance in domains like health care, where data collection can be both expensive and unethical or impractical, and *high-dimensional* regimes that involve jointly reasoning about millions or billions of variables. Due to many of these systems being mission critical, learning algorithms with strong computational and statistical guarantees are desired. The focus of this thesis is on intractable learning problems that arise in two important areas: causal inference and game theory. The problems that we consider, while being intractable, have a common theme in that they are concerned with learning *combinatorial structures* from data. We consider both the computational and statistical aspects of the problems and propose algorithms that are both polynomial time and (nearly) statistically optimal. In what follows, we describe the problems considered in this thesis at a high level and then summarize the main technical contributions of the thesis.

1.1.1 Directed Probabilistic Graphical Models

Directed graphical models, or Bayesian networks, provide a compact representation of joint distributions over many variables by representing the conditional independence relationships encoded in the distribution as a directed acyclic graph (DAG). Learning the DAG structure of Bayesian networks from observational data is of tremendous practical importance, since under suitable assumptions on the data generating process, they help recover cause-effect relationships from purely observational data. Estimating direct causal effects from data is the fundamental goal of causal inference. For instance, in health sciences, practitioners are often interested in estimating an outcome Y (severity of a disease) given a treatment X (drug). Similarly, in genetics, a pertinent problem is to determine how much a transcription factor X regulates the expression of a gene Y. These problems can be cast in the structural equation model (SEM) framework of Pearl [Pea09], which are a special case of Bayesian networks wherein each variable is written as a function of the variables that directly determine it — for instance, in the above examples we would write $Y = f(X)$ for some measurable function f. In the SEM framework, along with the set of algebraic equations, we have a directed acyclic graph (DAG) $G = (V, E)$, where the vertex set represents the variables under consideration ($V = \{X, Y\}$ for the preceding examples), and for each function describing the relationship between the quantities of interest, we have a directed arc from each variable on the right hand side of the equation to the variable on the left ($E = \{Y \leftarrow X\}$ for the preceding examples). The learning problem then corresponds to recovering the DAG G and the functions f given observations of the variables. The questions that the thesis tackles in this area are:

(i) **Identifiability:** Are the causal effects, i.e., the DAG and the functions quantifying the causal relationships, uniquely identifiable from observational or experimental data?

(ii) **Efficient algorithms:** Given that structure learning of SEMs is NP-complete [Chi96,Das99], are there broad sub-classes of SEMs that are poly-time learnable?

(iii) **Sample complexity:** What are the fundamental limits on the number of samples required by any procedure for estimating causal effects from data? Are there statistically efficient procedures for estimating causal effects? where by statistically efficient procedures we mean algorithms that attain the fundamental limits on the number of samples required for causal inference.

1.1.2 Game Theory

While in the previous section we considered scenarios that are best viewed through the lens of causal inference, many complex real-world data can be thought of as resulting from the behavior of a large number of self-interested agents trying to myopically or locally maximize some utility. Over the past several decades, non-cooperative game theory has emerged as a powerful mathematical framework for reasoning about such strategic interactions between self-interested agents. Traditionally, research in game theory has focused on computing the *Nash equilibria* (NE) (c.f. [BSK06] and [JLB11]) — which characterizes the stable outcome of the overall behavior of self-interested agents — *correlated equilibria* (c.f. [KKLO03]), and other solution concepts given a description of the game. Computing the *price of anarchy* (PoA) for graphical games, which in a sense quantifies the *inefficiency of equilibria*, is also of tremendous interest (c.f. [BZR11]). The aforementioned problems of computing the NE, correlated equilibria and PoA can be thought of as *inference problems* in graphical games, and require a description of the game, i.e., the payoffs of the players. In many real-world settings, however, only the behavior of the agents are observed, in which case inferring the latent payoffs of the players from observations of behavioral data becomes imperative. The learning problem then corresponds to recovering the structure and parameters of the player payoffs from observations of behavioral data such that, the

Nash equilibria of the game in some sense approximates the Nash equilibria of the true game. The key questions that the thesis pursues are:

(i) **Identifiability:** Given that the graph structure of a graphical game is not identifiable from the Nash equilibria set of the game, i.e., multiple graphs can result in the same Nash equilibria set under different payoff functions [HO15], is it possible to recover a game whose Nash equilibria are consistent with the game from which data was generated?

(ii) **Efficient algorithms:** Given that computing the Nash equilibria is intractable [CD06, DGP09], is it possible to learn games from behavioral data, without computing the Nash equilibria?

(iii) **Sample complexity:** What are the fundamental limits on the number of samples required by any procedure for learning games? Are there algorithms that achieve these fundamental lower bounds?

1.1.3 Structured Prediction

The aforementioned cases are instances of an approach to learning called generative modeling wherein we assume that data is generated from a specific family of models, viz. linear equation models or polymatrix games, and the goal is to develop estimators that recover a model from data that is as close as possible to the "true" model. *Discriminative learning* on the other hand provides an alternative approach for learning combinatorial structures from data, where the problem is to directly learn a *discriminant function* $f : \mathfrak{X} \to \mathfrak{Y}$ that maps input $x \in \mathfrak{X}$ to "structured outputs" $y \in \mathfrak{Y}$. Such problems are naturally dealt within the framework of *structured prediction* where for a given input x a prediction is made by first computing scores $\text{score}(x, y') \in \mathbb{R}$ for all $y' \in \mathfrak{Y}$ and then returning the output y that maximizes the score:

$$f(x) = \operatorname*{argmax}_{y' \in \mathfrak{Y}} \text{score}(x, y'), \tag{1.1}$$

with ties broken arbitrarily. The above *inference problem* (1.1) is often intractable and presents a number of computational and statistical challenges for learning optimal decoders f from a finite sample of observations of input-output pairs $\mathsf{S} = \{(x_i, y_i)\}_{i=1}^m$. The main questions that we seek to answer in this thesis are:

(i) **Efficient learning:** Is it possible to *efficiently* learn decoders $f : \mathfrak{X} \to \mathfrak{Y}$ from a finite sample $\mathsf{S} = \{(x_i, y_i)\}_{i=1}^m$ by solving the inference 1.1 problem approximately?

(ii) **Generalization guarantees:** Can decoders learned in such a way generalize to unseen examples?

1.2 Contributions

The overarching contribution of this thesis is to show that for many intractable problems in machine learning, there exists polynomial time algorithms that exactly solve broad sub-classes of problem instances in polynomial time, or that circumvent intractability by considering alternative objectives that are easier to solve while providing approximation guarantees on the original objectives. Specifically:

(i) We show that, SEMs are uniquely identifiable under conditions on the noise variances and data generating process that are more general than those known in the literature. We present truly polynomial time algorithms for solving such instances, obtain information-theoretic lower bounds on the number of samples required for recovering the structure of SEMs (or more generally Bayesian networks), and show that our algorithms are nearly statistically optimal.

(ii) We also develop polynomial time algorithms that learn games from behavioral data without computing the NE of the game, while still guaranteeing that the NE of the learned game is an epsilon-NE of the true game. We also obtain information-theoretic lower bounds for the problem of learning games from data

and show that the sample complexity of our algorithm is close to the information-theoretic limits.

(iii) We show that it is possible to learn structured predictors in polynomial time by solving the inference problem to a constant factor approximation. Furthermore, we obtain Rademacher-based generalization bounds for our structured predictors that guarantee generalization to unseen examples.

1.3 Outline and Previously Published Work

The rest of the report is organized as follows. Chapter 2 is concerned with learning SEMs, and describes in detail our main results, proofs, and juxtaposes our work with previous work. Chapter 3 delves into learning games, and describes our main results for the same, along with detailed comparison with prior work. Finally, Chapter 4 details our theoretical and empirical results for structured prediction.

The bulk of the report is based on the following papers [GH18a, GH18c, GH18b], which are joint work of mine with Jean Honorio. The first paper [GH18a] contains our results for learning SEMs, which in turn build on our previous papers [GH17a] and [GH17c]. The main results for learning games are described in the paper [GH18c], which in turn builds on our previous work related to learning games from [GH16, GH17b], while [GH18b] describes our main results for structure prediction.

1.4 General Notation

We will let $[p]$ denote the set $\{1, \ldots, p\}$. Since the first two chapters extensively relies on linear algebra, we denote vectors and matrices by lowercase and uppercase bold faced letters respectively to make the presentation clear. However, in the third chapter we do not make use of any special notation for vectors and matrices. For any two non-empty index sets s_r, s_c, the matrix $\mathbf{A}_{s_r,s_c}$ denotes the submatrix of $\mathbf{A}$ obtained by selecting the s_r rows and s_c columns of $\mathbf{A}$. With a slight abuse of

notation, we will allow the index sets s_r and s_c to be a single index, e.g., i, and we will denote the index set of all rows (or columns) by $*$ and the set $-\mathsf{i} \stackrel{\text{def}}{=} [p] \setminus \{i\}$. For any $p \times p$ matrix $\mathbf{A}$ (equivalently $p \times 1$ vector), we will denote its support set by: $\mathcal{S}(\mathbf{A}) = \{(i,j) \in [p] \times [p] \,|\, A_{i,j} \neq 0\}$. Vector ℓ_p norms are denoted by $\|\cdot\|_p$. For matrices, $\|\cdot\|_p$ denotes the induced (or operator) ℓ_p-norm and $|\cdot|_p$ denotes the elementwise ℓ_p norm, i.e., $|\mathbf{A}|_p \stackrel{\text{def}}{=} (\sum_{i,j} |A_{i,j}|^p)^{1/p}$. For two matrices $\mathbf{A}$ and $\mathbf{B}$, $\mathbf{A} \circ \mathbf{B}$ denotes the Hadamard product of $\mathbf{A}$ and $\mathbf{B}$, while $\mathbf{diag}(\mathbf{A})$ denotes the vector formed by taking the diagonal of $\mathbf{A}$. For a vector $\mathbf{v}$, $\mathbf{Diag}(\mathbf{v})$ denotes the diagonal matrix with $\mathbf{v}$ in the diagonal.

2 LEARNING STRUCTURAL EQUATION MODELS

Structural equation models (SEMs) is a commonly employed mathematical machinery for performing causal inference. Conditions under which SEMs can be uniquely identified from observational data have been recently characterized. Unfortunately, for linear SEMs, identifiability conditions have been rather limited, and existing structure learning algorithms are inefficient. In this chapter, we consider the problem of learning linear SEMs over p variables and bounded-degree d, from purely observational data, with arbitrary noise distributions having bounded second moment — including but not limited to the Gaussian distribution. We generalize existing identifiability conditions for learning linear SEMs, and present computationally and statistically efficient algorithms for learning the structure of linear SEMs when identifiable. We make the following technical contributions:

We present a new identifiability condition for learning linear SEMs from observational data that generalizes the homoscedastic Gaussian noise (equal noise variance) case considered by [PB14]. Our algorithm also works for the case when the noise variances are known up to a constant factor — a sufficient condition under which linear SEMs are identifiable as shown by [LB14]. This disproves an earlier conjecture by [LB14] that "variance scaling or non-Gaussianity is necessary in order to guarantee identifiability" of linear SEMs. Moreover, we show that our identifiability condition is necessary for ensuring identifiability of linear SEMs, in the sense that, if the identifiability condition is violated then there exist an exponential number of DAGs which induce the same covariance and precision matrix, and specify distributions that have the same conditional independence structures.

To the best of our knowledge, ours is the first method for learning SEMs with element-wise ℓ_∞ guarantees for recovering the autoregression matrix — the matrix of (directed) edge weights of the SEM. In contrast, score based approaches [VDGB13,

LB14] have guarantees on the score of the learned DAG structure. An unfortunate consequence of this is that, in order for these methods to recover the true DAG structure by finding the highest scoring DAG structure on the sample data set, the "score gap" between the true structure and the next best structure must scale as $\Omega(p)$ (see Equation 27 in [LB14]), which is unreasonable since the best DAG structure and the next best DAG structure might only differ on a constant number of edges, in which case the scores might differ by $o(p)$.

Our method is fully non-parametric, works for both Gaussian and non-Gaussian noise, and, to the best of our knowledge, the most efficient algorithm available for learning linear SEMs with provable guarantees. Given the inverse covariance (or precision) matrix, our method, which resembles a Cholesky factorization, can recover the structure and parameters of the SEM exactly in $\mathcal{O}(p(d + \log p))$ floating-point operations. In contrast [LB14]'s algorithm takes $\mathcal{O}\left(p2^{2(w+1)(w+d)}\right)$ time in the population setting, where w is the tree-width and d is the maximum degree of the graph. In the finite sample setting, our method involves estimating the precision matrix, which can be done by solving p linear programs (LPs) and then performing p iterations to learn the structure and parameters of the SEM by identifying and removing terminal (sink) vertices. If the estimated precision matrix is sparse, then each iteration involves solving at most d linear programs in at most d dimensions, leading to an overall smoothed complexity of $\widetilde{\mathcal{O}}(p^3 + pd^4)$. When the estimated precision matrix is dense our method has a smoothed complexity of $\widetilde{\mathcal{O}}(p^5)$. This is significantly better than [PB14]'s algorithm for learning linear Gaussian SEMs as well as [LB14]'s algorithm for learning SEMs with known noise variance. While the former is is exponential in p, the latter is exponential in d and the tree-width of the SEM when the estimated precision matrix is sparse and exponential in p for the dense case.

Our algorithm also works in the high-dimensional regime, when $n \ll p$ and $d = o(p)$, and has a sample complexity of $\mathcal{O}(\frac{d^4}{\varepsilon^2}\log(\frac{p}{\sqrt{\delta}}))$ and $\mathcal{O}(\frac{d^4}{\varepsilon^2}(\frac{p^2}{\delta})^{1/m})$ for sub-Gaussian noise and noise with bounded $4m$-th moment respectively, for recovering the autoregression matrix of the SEM up to ϵ additive error with probability at least $1-\delta$.

The sample complexity of our algorithm for sub-Gaussian noise is better than [LB14]'s algorithm, which has a sample complexity of $\mathcal{O}(p^2 \log p)$, and is therefore unsuitable for the high-dimensional regime. Moreover, unlike [LB14]'s algorithm, and other methods that use conditional independence tests, for instance, the PC algorithm for learning Gaussian SEMs [KP07], our algorithm does not require any faithfulness conditions, and only requires a weaker *causal minimality* condition. The PC algorithm and [LB14]'s algorithm can fail to recover the correct DAG for distributions that are not faithful to the DAG structure. Our results have the following significant yet hitherto unknown implication for learning Gaussian Bayesian networks. Given data generated from a Gaussian Bayesian network that is causal minimal to the true DAG structure, one can recover the DAG structure in polynomial time and sample complexity from a finite number of samples, under more general identifiability conditions than homoscedastic noise.

Lastly, we obtain several useful results about the theory of linear SEMs en route to developing our main algorithm for learning linear SEMs.

2.1 Preliminaries

We begin this section by introducing our notations and definitions before formalizing the problem of learning linear SEMs from observational data.

Let $\mathsf{G} = ([p], \mathsf{E})$ be a directed acyclic graph (DAG) where $[p]$ is the vertex set and $\mathsf{E} \subset [p] \times [p]$ is the set of directed edges. An edge $(i, j) \in \mathsf{E}$ implies the edge $i \leftarrow j$. We denote by $\pi_{\mathsf{G}}(i)$ and $\phi_{\mathsf{G}}(i)$ the parent set and the set of children of the i-th node respectively, in the graph G; and drop the subscript G when the clear from context. The set of neighbors of the i-th node is denoted by $\mathsf{N}_{\mathsf{G}}(i) = \pi_{\mathsf{G}}(i) \cup \phi_{\mathsf{G}}(i)$. A node j is a *descendant* of i in G if there exists a (directed) path from i to j in G. We will denote the set of descendants of i by $\mathsf{D}_{\mathsf{G}}(i)$. Similarly, we will denote the set of ancestors of i — nodes j such that there is a path from j to i in G — by the set $\mathsf{A}_{\mathsf{G}}(i)$. The Markov blanket of a node is defined as: $\mathsf{MB}_{\mathsf{G}}(i) = \mathsf{N}_{\mathsf{G}}(i) \cup \{k \in \pi_{\mathsf{G}}(j) \mid j \in \phi_{\mathsf{G}}(i)\}$.

A vertex $i \in [p]$ is a *terminal vertex* in G if $\phi_{\mathsf{G}}(i) = \varnothing$. For each $i \in [p]$ we have a random variable $X_i \in \mathbb{R}$, $X = (X_1, \ldots, X_p) \in \mathbb{R}^p$ is the p-dimensional vector of random variables, and $\mathbf{x} = (x_1, \ldots, x_p)$ is a joint assignment to X. Every DAG $\mathsf{G} = ([p], \mathsf{E})$ defines a set of topological orderings $\mathcal{T}_{\mathsf{G}}$ over $[p]$ that are compatible with the DAG G, i.e., $\mathcal{T}_{\mathsf{G}} = \{\tau \in \mathrm{S}_p \mid \tau(j) < \tau(i) \text{ if } (i,j) \in \mathsf{E}\}$, where S_p is the set of all possible permutations of $[p]$.

The random vector X follows a linear structural equation model (SEM), if each variable can be written as a linear combination of the variables in its parent set as follows:

$$X_i = \sum_{j \in \pi_{\mathsf{G}}(i)} B_{i,j} X_j + N_i \qquad (\forall i \in [p]), \tag{2.1}$$

where $\mathsf{G} = ([p], \mathsf{E})$ is a DAG, $N = (N_1, \ldots, N_p)$ are the noise or exogenous variables. We assume that the exogenous variables $N_i \perp\!\!\!\perp X_j$ for all $j \notin \mathsf{D}_{\mathsf{G}}(i)$ and $i \in [p]$. Without loss of generality, we assume that $\mathbb{E}[X_i] = \mathbb{E}[N_i] = 0, \forall i \in [p]$. As is typically the case in the literature of SEMs, we further assume that the noise variables N_i have bounded second moments and are independent. Thus $\mathrm{Cov}[N] = \mathbb{E}\left[NN^T\right] = \mathbf{Diag}(\sigma_1^2, \ldots, \sigma_p^2)$. We can then write (2.1) in vector form as follows:

$$X = \mathbf{B}X + N, \tag{2.2}$$

where $\mathbf{B} = (B_{i,j})$ is referred to as the *autoregression matrix* and $\mathcal{S}(\mathbf{B}) = \mathsf{E}$. Therefore, we will denote an SEM by the triple $(\mathsf{G}, \mathbf{B}, \{\sigma_i^2\})$ [1].

Given an SEM $(\mathsf{G}, \mathbf{B}, \{\sigma_i^2\})$, the joint distribution $\mathcal{P}(X)$ is completely determined and factorizes according to the DAG structure G:

$$\mathcal{P}(X; \mathsf{G}) = \prod_{i=1}^{p} \mathcal{P}_i(X_i | X_{\pi_{\mathsf{G}}(i)}; \mathsf{G}), \tag{2.3}$$

where $\mathcal{P}_i$ is the conditional distribution of the X_i. We then say that the distribution $\mathcal{P}$ is *Markov with respect to the DAG* G, i.e., X_i satisfies the Markov condition:

[1] An SEM is fully characterized by G, $\mathbf{B}$ and the distribution of the exogenous variables. However, since we are concerned with learning SEMs using second moments only, our notation captures all the required information.

$X_i \perp\!\!\!\perp X_j \mid X_{\pi(i)}$, $\forall i \in [p], \forall j \in [p] \setminus (\mathsf{D}(i) \cup \pi(i) \cup \{i\})$. Thus an SEM is equivalent to a *Bayesian network*. Specifically, if the noise variables are Gaussian, then $\mathcal{P}$ is a *Gaussian Bayesian network* (GBN), where the joint distribution $\mathcal{P}$ and the conditional distributions $\mathcal{P}_i$ are Gaussian. We obtain our theoretical results for the class of DAGs with Markov blanket at most d: $\mathcal{G}_{p,d} \stackrel{\text{def}}{=} \{\mathsf{G} \mid \mathsf{G} = ([p], \mathsf{E})$ is a DAG and $|\mathsf{MB}_{\mathsf{G}}(i)| \leq d, \forall i \in [p]\}$.

Next, we define the notion of *causal minimality*, introduced by [ZS08], which is important for ensuring identifiability of linear SEMs considered in this report.

Definition 2.1.1 (Causal Minimality) *Given a DAG G, a distribution $\mathcal{P}(X)$, that is Markov with respect to G, is causal minimal if $\mathcal{P}$ is not Markov with respect to a proper subgraph of G.*

Our assumption of $\mathcal{S}(\mathbf{B}) = \mathsf{E}$, ensures that Lemma 4 of [PMJS14] holds for all SEMs $(\mathsf{G}, \mathbf{B}, \{\sigma_i^2\})$. This in turn implies that the joint distribution $\mathcal{P}(X)$ determined by the SEM $(\mathsf{G}, \mathbf{B}, \{\sigma_i^2\})$ is causal minimal with respect to G (see Proposition 2 in [PMJS14]). Therefore, the SEMs considered in the report are causal minimal. Causal minimality is much weaker than faithfulness which requires that the distribution $\mathcal{P}(X)$ contain only those conditional independence assertions that are implied by the *d-separation* criteria of the DAG [SGS00]. However, faithfulness cannot be tested from data in full generality [ZS08] and algorithms that infer the DAG structure from a finite number of samples must require *strong faithfulness* [ZS02], which is a restrictive assumption.

Problem: The problem of learning the structure of an SEM is as follows. Given an $n \times p$ data matrix $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_p)$, with $\mathbf{x}_i \in \mathbb{R}^n$, drawn from an SEM $(\mathsf{G}^*, \mathbf{B}^*, \{\sigma_i^2\})$ with $\mathsf{G}^* \in \mathcal{G}_{p,d}$, we want to learn an SEM $(\widehat{\mathsf{G}}, \widehat{\mathbf{B}}, \{\widehat{\sigma}_i^2\})$ from $\mathbf{X}$ such that $\mathsf{G}^* = \widehat{\mathsf{G}}$.

2.2 Related Work

We start our discussion of existing literature by first presenting known identifiability conditions for learning SEMs and Bayesian networks. [PMJS14] proved identifiability of distributions drawn from a restricted SEM with additive noise, where

in the restricted SEM the functions are assumed to be non-linear and thrice continuously differentiable. Linear SEMs are identifiable if (a) the noise variables are non-Gaussian [SHHK06], (b) the noise variances are known up to a constant factor [LB14], and (c) noise variables are Gaussian and have the same variance [PB14] (homoscedastic noise). [PR17] introduced Quadratic Variance Function (QVF) DAG models — a class of Bayesian networks in which the conditional variance of a variable is a quadratic function of its conditional mean — and proved identifiability of the models from observational data. However, QVF DAG models cannot be expressed as SEMs in general, and the quadratic variance property holds for a handful of conditional distributions which includes Binomial, Poisson, Exponential, Gamma, and a few others.

The computational and statistical complexity landscape of learning linear SEMs is peppered by inefficient algorithms. This is in part justified by various hardness results known in the literature for learning DAGs from observational data [Chi96,Das99]. Algorithms for learning DAGs can be divided into two categories: independence test based methods and score based methods. Score based methods use a score function, typically penalized log-likelihood, to find the best scoring DAG among the space of all DAGs. Since the number of DAGs and degree-bounded DAGs is exponential in p [Rob77, GH17a] brute force methods, and existing score-based methods are exponential time. A popular score function for learning Gaussian SEMs is the ℓ_0-penalized Gaussian log-likelihood score proposed by [VDGB13]. [PB14] proposed using ℓ_0-penalized Gaussian log-likelihood score for learning homoscedastic noise linear Gaussian SEMs along with a heuristic greedy search algorithm which is not guaranteed to find the correct (highest-scoring) solution. [LB14] showed that under a faithfulness assumption, the sparsity pattern of the precision matrix corresponds to the edge structure of the *moral graph* of the underlying DAG. They exploit this property to devise an algorithm that searches for the highest-scoring DAG, using dynamic programming, that has the same moral graph as that given by the sparsity pattern of the precision matrix. Independence test based methods on the other hand require

restrictive faithfulness conditions to guarantee structure recovery. [KP07] proposed using the PC algorithm, which was originally proposed by [SGS00] and has a computational complexity of $\mathcal{O}\left(p^d\right)$, for learning Gaussian SEMs and proved asymptotic uniform consistency of the algorithm for recovering the Markov equivalence class, i.e., a CPDAG. However the PC algorithm is only efficient for learning very sparse Gaussian SEMs. Among computationally efficient algorithms, the *Direct-LiNGAM* algorithm [SIS$^+$11], which strictly requires non-Gaussianity of the noise variables, needs an infinite number of samples to guarantee structure recovery. This is because of the use of independence testing between a variable and its residuals to detect exogenous variables (variables with no parents). For the same reason, the correctness of *RESIT* [PMJS14], which is a computationally efficient algorithm for learning *non-linear SEMs*, is only guaranteed in the population setting. [GH17c] proposed a polynomial time algorithm, similar to the one proposed in this paper, for learning Gaussian SEMs (or Gaussian Bayesian networks) with a sample complexity of $\mathcal{O}\left(d^4 \log p\right)$. However, their method, theoretical guarantees and proofs crucially rely on the Gaussianity of the data distribution.

Other authors have proposed various approximation algorithms and heuristic methods for learning Bayesian networks, which can be used to learn Gaussian SEMs by using appropriate score functions. Popular heuristic methods are max-min hill climbing (MMHC) algorithm by [TBA06], and the Greedy Equivalence Search (GES) algorithm proposed by [Chi03]. [JSG$^+$10] proposed an LP-relaxation based method for learning Bayesian networks which is an approximation algorithm.

2.3 Learning SEMs with Unknown Error Variances

We start with presenting our main results for learning SEMs when the error variances are unknown. Our algorithm for learning SEMs works by constructing the SEM in a bottom-up fashion. The algorithm has p iterations. In each iteration it identifies and removes a terminal vertex, learning its parent set and edge weights along the

way. We show that, under a certain identifiability condition which generalizes other identifiability conditions known in the literature, e.g., homoscedastic errors, and without assuming faithfulness of the distribution to the DAG, each of these steps can be performed efficiently using only the precision matrix or an estimator of it.

2.3.1 Identifiability

The following assumption gives a sufficient condition under which the structure and parameters of an SEM can be uniquely recovered from observational data using Algorithm 1. The assumption is defined in terms of subgraphs of G obtained by removing terminal vertices sequentially. For any $\tau \in \mathcal{T}_{\mathsf{G}}$, we will consider sequence of graphs $\mathsf{G}[m, \tau] = (\mathsf{V}[m, \tau], \mathsf{E}[m, \tau])$, indexed by (m, τ), where $\mathsf{G}[m, \tau]$ is the induced subgraph of G over the first m vertices in the topological ordering τ, i.e., $\mathsf{V}[m, \tau] \stackrel{\text{def}}{=} \{i \in [p] \mid \tau(i) \leq m\}$ and $\mathsf{E}[m, \tau] \stackrel{\text{def}}{=} \{(i, j) \in \mathsf{E} \mid i \in \mathsf{V}[m, \tau] \wedge j \in \mathsf{V}[m, \tau]\}$.

Assumption 2.3.1 (Identifiability condition) *Given an SEM* $(\mathsf{G}, \mathbf{B}, \{\sigma_i^2\})$ *with* $\mathsf{G} \in \mathcal{G}_{p,d}$*, then* $\forall (i, j) \in \mathsf{V}[m, \tau] \times \mathsf{V}[m, \tau], m \in [p]$*, and* $\forall \tau \in \mathcal{T}_{\mathsf{G}}$*, such that* $\phi_{\mathsf{G}[m,\tau]}(i) = \varnothing \wedge \phi_{\mathsf{G}[m,\tau]}(j) \neq \varnothing$*:*

$$(\sigma_i^2)^{-1} < (\sigma_j^2)^{-1} + \sum_{l \in \phi_{\mathsf{G}[m,\tau]}(j)} (\sigma_l^2)^{-1} B_{l,j}^2, \tag{2.4}$$

As we will show later, Assumption 2.3.1 essentially lays down a condition under which terminal vertices, and subsequently the causal order, can be identified from the precision matrix. From Assumption 2.3.1, we immediately get the following special cases for identifiability of linear SEMs, where the first one is the homoscedastic case known in the literature, while the second case is new.

Proposition 2.3.1 (Sufficient conditions for identifiability)
Let $(\mathsf{G}, \mathbf{B}, \{\sigma_i^2\})$ *be an SEM satisfying Assumption 2.3.1, with precision matrix* $\mathbf{\Omega}$*. Then, either of the following two conditions are sufficient for uniquely identifying the autoregression matrix* $\mathbf{B}$ *and the DAG* G *from* $\mathbf{\Omega}$*:*

(i) $\forall i \in [p]$, $\sigma_i = \sigma$, *for some* $\sigma > 0$,

(ii) $1 < \sigma_i \leq B_{\min}$ $(\forall i \in [p])$, *where* $B_{\min} \stackrel{\text{def}}{=} \min\{|B_{i,j}| \mid (i,j) \in \mathsf{E}\}$

For detailed proofs, see Appendix A. At this point one might ask if the above assumption is *necessary* for identifiability of linear SEMs. We answer this question in affirmative in the following lemma, which states that if Assumption 2.3.1 is violated, then there exists an exponential number of DAGs structures that induce the same covariance and precision matrix, and determine joint distributions $\mathcal{P}(X)$ that are causal minimal and Markov to the DAG structures. In the following lemma we will equivalently denote an SEM by $(\mathsf{G}, \mathbf{B}, \mathbf{D})$ where $\mathbf{D}$ is a diagonal matrix with $D_{i,i} = \sigma_i^2$.

Lemma 2.3.2 *There exists* $\widetilde{\mathcal{G}}_{p,d} \subset \mathcal{G}_{p,d}$ *with* $\left|\widetilde{\mathcal{G}}_{p,d}\right| = 2^{\Theta(p)}$, *autoregression matrices* $\mathbf{B}(\beta)$ *parameterized by* β, *and diagonal matrices* $\mathbf{D}(v_1, v_2)$ *parameterized by* v_1, v_2, *such that for each* $\beta \in (-\infty, \infty)$ *and* $v_1 \in (0, \infty)$ *and* $v_2 > v_1$, *the family of SEMs* $\{(\mathsf{G}, \mathbf{B}(\beta), \mathbf{D}(v_1, v_2)) \mid G \in \widetilde{\mathcal{G}}_{p,d}\}$, *do not satisfy Assumption 2.3.1, induce the same covariance and precision matrix, and distribution* $\mathcal{P}(X)$ *that has the same conditional independence structure.*

Given that the true SEM can come from the aforementioned family, no algorithm, that uses only conditional independence tests and second moments, can consistently recover the true DAG structure if Assumption 2.3.1 is not satisfied. Next, we present a series of results building towards our main result for learning SEMs from precision matrix. In the following proposition we characterize the precision matrix of linear SEMs.

Proposition 2.3.2 *Let* $(\mathsf{G}, \mathbf{B}, \{\sigma_i^2\})$ *be an SEM over* X*, then the precision matrix is given as:* $\mathbf{\Omega} = (\mathbf{I} - \mathbf{B})^T \mathbf{D}^{-1} (\mathbf{I} - \mathbf{B})$*, where* $\mathbf{D} = \mathbf{Diag}(\sigma_1^2, \ldots, \sigma_p^2)$*. The entries of the precision matrix is given as:*

$$\Omega_{i,i} = (\sigma_i^2)^{-1} + \sum_{l \in \phi(i)} (\sigma_l^2)^{-1} B_{l,i}^2, \tag{2.5}$$

$$\Omega_{i,j} = -(\sigma_i^2)^{-1} B_{i,j} - (\sigma_j^2)^{-1} B_{j,i} + \sum_{l \in \phi(i) \cap \phi(j)} (\sigma_l^2)^{-1} B_{l,i} B_{l,j}.$$

The above characterization of the precision matrix motivates our indentifiability condition given by Assumption 2.3.1, and also provides a recipe for identifying terminal vertices from the precision matrix as is formalized by the following proposition.

Proposition 2.3.3 *Let* $(\mathsf{G}, \mathbf{B}, \{\sigma_i^2\})$ *be a SEM over* X *with precision matrix* $\mathbf{\Omega}$*, that satisfies the identifiability condition given by Assumption 2.3.1. Then,* i *is a terminal vertex in* G *if and only if* $i \in \operatorname{argmin}(\mathbf{diag}(\mathbf{\Omega}))$*. Further, if* i *is a terminal vertex then* $\sigma_i^2 = 1/\Omega_{i,i}$*.*

The next proposition, which follows directly from Proposition 2.3.3 and (2.5), states that for a terminal vertex the parent set and edge weights can be conveniently "read off" from the precision matrix. This is the key result which helps us avoid the faithfulness condition.

Proposition 2.3.4 *Let* $(\mathsf{G}, \mathbf{B}, \{\sigma_i^2\})$ *be an SEM over* X *with precision matrix* $\mathbf{\Omega}$*. If* i *is a terminal vertex in* G*, then* $\mathbf{B}_{i,*} = -\mathbf{\Omega}_{i,*}/\Omega_{i,i}$ *and* $\pi_{\mathsf{G}}(i) = \mathcal{S}(\mathbf{\Omega}_{i,*}) \setminus \{i\}$*.*

The following lemma is a useful result about linear SEMs with arbitrary noise distribution, that generalizes a result so far known only for the Gaussian distribution — for a terminal vertex i, the precision matrix over $X_{-\mathsf{i}}$ can be obtain by performing a Schur complement update of the precision matrix over X. While, the result for the Gaussian distribution holds for all variables, the analogous result for general SEMs holds only for terminal vertices.

Lemma 2.3.3 *Let* $(\mathsf{G}, \mathbf{B}, \{\sigma_i^2\})$ *be an SEM over* X *with precision matrix* $\mathbf{\Omega}$*. Let* i *be a terminal vertex in the* G*, then the precision matrix over* $X_{-\mathsf{i}}$*,* $\mathbf{\Omega}_{(-\mathsf{i})}$*, is given as:* $\mathbf{\Omega}_{(-\mathsf{i})} = \mathbf{\Omega}_{-\mathsf{i},-\mathsf{i}} - \Omega_{i,i}^{-1} \mathbf{\Omega}_{-\mathsf{i},i} \mathbf{\Omega}_{i,-\mathsf{i}}$*.*

Finally, the following lemma characterizes the entries of the precision matrix over $X_{-\mathsf{i}}$ and will be very useful in developing our finite-sample algorithm for learning SEMs.

Lemma 2.3.4 *Let $(\mathsf{G}, \mathbf{B}, \{\sigma_i^2\})$ be a SEM over X with precision matrix $\mathbf{\Omega}$. Let i be a terminal vertex in the G and let $\mathbf{\Omega}_{(-\mathsf{i})}$ denote the precision matrix over $X_{-\mathsf{i}}$. Then,*

$$(\mathbf{\Omega}_{(-\mathsf{i})})_{j,k} = \mathbf{\Omega}_{j,k},\ (\forall (j,k) \in -\mathsf{i} \times -\mathsf{i} \mid \{j,k\} \nsubseteq \pi_{\mathsf{G}}(i)),$$
$$\mathcal{S}((\mathbf{\Omega}_{(-\mathsf{i})})_{j,*}) \subseteq (\mathcal{S}(\mathbf{\Omega}_{j,*}) \setminus \{i\}) \cup \pi_{\mathsf{G}}(i)\ (\forall j \in \pi_{\mathsf{G}}(i)).$$

With the required results in place, we are now ready to present our main algorithm, detailed in Algorithm 1, for learning SEMs from the precision matrix. The role of the diagonal matrix $\mathbf{D}$ will become clear in the next section where we focus on the problem of learning SEMs with known error variances. For now we simply set $\mathbf{D}$ to the identity matrix $\mathbf{I}$. The following theorem proves the correctness of our algorithm in the population setting.

Algorithm 1 SEM structure learning algorithm.

Input: Precision matrix $\mathbf{\Omega}$, diagonal matrix $\mathbf{D}$.

Output: $\widehat{\mathsf{G}}, \widehat{\mathbf{B}}$.

1: $\widehat{\mathbf{B}} \leftarrow \mathbf{0}$.
2: **for** $t \in [p]$ **do**
3: $\quad i \leftarrow \text{argmin}(\mathbf{diag}(\mathbf{\Omega} \circ \mathbf{D}))$.
4: $\quad \mathbf{B}_{i,*} \leftarrow -\mathbf{\Omega}_{i,*}/\Omega_{i,i},\ B_{i,i} \leftarrow 0$.
5: $\quad \mathbf{\Omega} \leftarrow \mathbf{\Omega} - \frac{1}{\Omega_{i,i}} \mathbf{\Omega}_{*,i} \mathbf{\Omega}_{i,*}$.
6: $\quad \Omega_{i,i} \leftarrow \infty$.
7: **end for**
8: $\widehat{\mathsf{G}} \leftarrow ([p], \mathcal{S}(\widehat{\mathbf{B}}))$.

Algorithm 2 Updating a precision matrix, after removing a terminal vertex, using CLIME.

1: **function** UPDATE($\widehat{\mathbf{\Omega}}, i, \lambda_n$)
2: $\widehat{\pi}(i) \leftarrow \mathcal{S}(\widehat{\mathbf{\Omega}}_{i,*}) \setminus \{i\}$.
3: **for** $j \in \widehat{\pi}(i)$ **do**
4: $\widehat{\mathsf{S}}_j \leftarrow \left(\mathcal{S}(\widehat{\mathbf{\Omega}}_{j,*}) \setminus \{i\}\right) \cup \widehat{\pi}(i)$.
5: Compute $\bar{\boldsymbol{\omega}}_j$ by solving (2.7) for $\mathbf{\Sigma}^n_{\widehat{\mathsf{S}}_j, \widehat{\mathsf{S}}_j}$.
6: $\widehat{\mathbf{\Omega}}_{j,\widehat{\mathsf{S}}_j} = \widehat{\mathbf{\Omega}}_{\widehat{\mathsf{S}}_j,j} \leftarrow \bar{\boldsymbol{\omega}}_j$
7: **end for**
8: $\widehat{\mathbf{\Omega}}_{i,*} \leftarrow \mathbf{0}$ and $\widehat{\mathbf{\Omega}}_{*,i} \leftarrow \mathbf{0}$.
9: **return** $\widehat{\mathbf{\Omega}}$.
10: **end function**

Theorem 2.3.5 *Let* $(\mathsf{G}, \mathbf{B}, \{\sigma_i^2\})$ *be an SEM over* X*, with precision matrix* $\mathbf{\Omega}$*, satisfying Assumption 2.3.1. Then, given* $(\mathbf{\Omega}, \mathbf{I})$ *as input, Algorithm 1 returns a unique* $(\widehat{\mathsf{G}}, \widehat{\mathbf{B}})$ *such that* $\widehat{\mathsf{G}} = \mathsf{G}$ *and* $\widehat{\mathbf{B}} = \mathbf{B}$.

As a consequence of the above theorem we have the following corollary about identifiability of linear SEMs.

Corollary 2.3.6 *An SEM* $(\mathsf{G}, \mathbf{B}, \{\sigma_i^2\})$ *satisfying Assumption 2.3.1 is identifiable, and can be uniquely identified from the precision matrix* $\mathbf{\Omega}$.

2.3.2 Statistical Guarantees for Estimation

Algorithm 1 can be used to learn a SEM given an estimate of the precision matrix, computed from a finite number of samples, with a slight modification. In line 5 instead of using the Schur complement update, we use Algorithm 2 to update the precision matrix after a terminal vertex has been identified (and removed). The rationale behind this is that even if the estimated precision matrix is close to the true precision matrix, the Schur updates could still result in errors accumulating in the

precision matrix. In order to ensure that our algorithm is statistically efficient, we need more control over those errors, which in turns calls for some sort of penalization for estimating from a finite number of samples.

Inverse covariance matrix estimation. In the finite sample setting, our algorithm involves estimating the inverse covariance matrix, and subsequently updating the inverse covariance matrix after removing a terminal vertex. Due in part to its role in undirected graphical model selection, the problem of inverse covariance matrix estimation has received significant attention and many algorithms have been developed for the problem. In this paper we use the CLIME algorithm, proposed by [CLL11], to estimate the inverse covariance matrix and propose a modification of the CLIME algorithm for efficiently computing the inverse covariance matrix over the variables remaining after eliminating a terminal vertex in Algorithm 2.

The CLIME estimator, $\widehat{\mathbf{\Omega}}$, of the inverse covariance matrix $\mathbf{\Omega}$ is obtained as follows. First, we compute a potentially non-symmetric estimate $\bar{\mathbf{\Omega}} = (\bar{\omega}_{i,j})$ by solving the following:

$$\bar{\mathbf{\Omega}} = \underset{\mathbf{\Omega} \in \mathbb{R}^{p \times p}}{\operatorname{argmin}} |\mathbf{\Omega}|_1 \text{ s.t. } |\mathbf{\Sigma}^n \mathbf{\Omega} - \mathbf{I}|_\infty \leq \lambda_n, \tag{2.6}$$

where $\lambda_n > 0$ is the regularization parameter, $\mathbf{\Sigma}^n \stackrel{\text{def}}{=} (1/n)\mathbf{X}^T\mathbf{X}$ is the empirical covariance matrix, and $|\cdot|_1$ (respectively $|\cdot|_\infty$) denotes elementwise ℓ_1 (respectively ℓ_∞) norm. Finally, the symmetric estimator is obtained by selecting the smaller entry among $\bar{\omega}_{i,j}$ and $\bar{\omega}_{j,i}$, i.e., $\widehat{\mathbf{\Omega}} = (\widehat{\omega}_{i,j})$, where $\widehat{\omega}_{i,j} = \bar{\omega}_{i,j}\mathbf{1}\left[|\bar{\omega}_{i,j}| < |\bar{\omega}_{j,i}|\right] + \bar{\omega}_{j,i}\mathbf{1}\left[|\bar{\omega}_{j,i}| \leq |\bar{\omega}_{i,j}|\right]$. It is easy to see that (2.6) can be decomposed into p linear programs as follows. Let $\bar{\mathbf{\Omega}} = (\bar{\boldsymbol{\omega}}_1, \ldots, \bar{\boldsymbol{\omega}}_p)$, then

$$\bar{\boldsymbol{\omega}}_i = \underset{\boldsymbol{\omega} \in \mathbb{R}^p}{\operatorname{argmin}} \|\boldsymbol{\omega}\|_1 \text{ s.t. } |\mathbf{\Sigma}^n \boldsymbol{\omega} - \mathbf{e}_i|_\infty \leq \lambda_n, \tag{2.7}$$

where $\mathbf{e}_i = (e_{i,j})$ such that $e_{i,j} = 1$ for $j = i$ and $e_{i,j} = 0$ otherwise. The main result about the CLIME estimator that we use from [CLL11] is given by the following lemma, which is a minor reformulation of Theorem 6 in [CLL11]:

Lemma 2.3.7 ([CLL11]) *Let $(\mathsf{G}, \mathbf{B}, \{\sigma_i^2\})$ be an SEM over X, with covariance and precision matrix $\mathbf{\Sigma}$ and $\mathbf{\Omega}$ respectively. Let $\widehat{\mathbf{\Omega}}$ be the estimator of $\mathbf{\Omega}$ obtained by solving the optimization problem given by 2.7. Then if $\lambda_n \geq \|\mathbf{\Omega}\|_1 |\mathbf{\Sigma} - \mathbf{\Sigma}^n|_\infty$, then $\left|\mathbf{\Omega} - \widehat{\mathbf{\Omega}}\right|_\infty \leq 4 \|\mathbf{\Omega}\|_1 \lambda_n$. Further, if*

$$\min\{|\Omega_{i,j}| \mid (i,j) \in [p] \times [p] \wedge |\Omega_{i,j}| \neq 0\} > 4 \|\mathbf{\Omega}\|_1 \lambda_n,$$

then $\mathcal{S}(\mathbf{\Omega}) \subseteq \mathcal{S}(\widehat{\mathbf{\Omega}})$.

Next we state out finite sample identifiability condition. This differs from the population version in that we require a "gap" between the diagonal entries of the precision matrix for terminal and non-terminal vertices. This gap, as we show later, must scale as $\Omega\left(d\sqrt{\log p/n}\right)$ and $\Omega\left(d(p)^{1/m}/\sqrt{n}\right)$ for sub-Gaussian noise and bounded moment noise respectively. Condition (ii) of the below assumption also restricts how fast the "minimum" non-diagonal entry of the precision matrix must decay. Note that our conditions are weaker than those of [LB14] due to which we are able to achieve better sample complexity than their algorithm.

Assumption 2.3.8 (Finite Sample Identifiability Condition) *Let $(\mathsf{G}, \mathbf{B}, \{\sigma_i^2\})$ be an SEM with inverse covariance matrix $\mathbf{\Omega}$. Let $\mathbf{\Omega}_{(m,\tau)}$ denote the inverse covariance matrix over $X_{\mathsf{V}[m,\tau]}$, and*

$$M \stackrel{\text{def}}{=} \max\{\left\|\mathbf{\Omega}_{(m,\tau)}\right\|_1 \mid m \in [p], \tau \in \mathcal{T}_{\mathsf{G}}\}. \tag{2.8}$$

Then, we have that

(i) $\forall (i,j) \in \mathsf{V}[m,\tau] \times \mathsf{V}[m,\tau], m \in [p]$, *and* $\tau \in \mathcal{T}_{\mathsf{G}}$, *such that* $\phi_{\mathsf{G}[m,\tau]}(i) = \varnothing \wedge \phi_{\mathsf{G}[m,\tau]}(j) \neq \varnothing$*:*

$$\frac{1}{\sigma_i^2} < \frac{1}{\sigma_j^2} + \sum_{l \in \phi_{\mathsf{G}[m,\tau]}(j)} \frac{B_{l,j}^2}{\sigma_l^2} - 8M\lambda_n,$$

(ii) $\min\{\left|(\Omega_{(m,\tau)})_{i,j}\right| \mid (\Omega_{(m,\tau)})_{i,j} \neq 0, (i,j) \in \mathsf{V}[m,\tau] \times \mathsf{V}[m,\tau], m \in [p], \tau \in \mathcal{T}_{\mathsf{G}}\} > 4M\lambda_n$,

(iii) for all $i \in [p]$, $\sigma_i^2 \in o(1/4M\lambda_n)$.

The following lemma proves the correctness of Algorithm 2 which updates the precision matrix, after removing a terminal vertex.

Lemma 2.3.9 *Let* $(\mathsf{G}, \mathbf{B}, \{\sigma_i^2\})$ *be an SEM over* X *with precision matrix* $\mathbf{\Omega}$. *Let* $\widehat{\mathbf{\Omega}}$ *be an estimator of* $\mathbf{\Omega}$ *such that* $\left|\mathbf{\Omega} - \widehat{\mathbf{\Omega}}\right|_\infty \leq 4M\lambda_n$, *and* $\mathcal{S}(\mathbf{\Omega}) \subseteq \mathcal{S}(\widehat{\mathbf{\Omega}})$, *where* M *is defined in* (2.8). *Let* i *be a terminal vertex in the* G, $\mathbf{\Omega}_{(-\mathsf{i})}$ *be the true precision matrix over* $X_{-\mathsf{i}}$, *and let* $\widehat{\mathbf{\Omega}}'$ *be the matrix returned by the function* UPDATE. *Then,* $\left|\mathbf{\Omega}_{(-\mathsf{i})} - \widehat{\mathbf{\Omega}}'_{-\mathsf{i},-\mathsf{i}}\right|_\infty \leq 4M\lambda_n$ *and* $\mathcal{S}(\mathbf{\Omega}_{(-\mathsf{i})}) \subseteq \mathcal{S}(\widehat{\mathbf{\Omega}}')$.

Theorem 2.3.10 *Let* $(\mathsf{G}^*, \mathbf{B}^*, \{\sigma_i^2\})$ *be the true SEM, with covariance and precision matrix* $\mathbf{\Sigma}^*$ *and* $\mathbf{\Omega}^*$, *respectively, from which a data set* $\mathbf{X}$ *of* n *samples is drawn. If the regularization parameter satisfies* $\lambda_n \geq M\left|\mathbf{\Sigma}^n - \mathbf{\Sigma}^*\right|$, *then under Assumption 2.3.8, the Algorithm 1, with* $\mathbf{D}$ *set to* $\mathbf{I}$, *returns an estimator* $\widehat{\mathbf{B}}$ *such that* $\left|\mathbf{B}^* - \widehat{\mathbf{B}}\right| \leq c4M(1 + B_{\max})\sigma_{\max}^2\lambda_n$, $\mathcal{S}(\mathbf{B}^*) \subseteq \mathcal{S}(\widehat{\mathbf{B}})$, *and* $\mathcal{T}_{\widehat{\mathsf{G}}} \subseteq \mathcal{T}_{\mathsf{G}^*}$, *where* $c \leq \sigma_{\min}^2/(1-4M\lambda_n\sigma_{\min}^2)$ *is a constant.*

Next, we use known concentration results for the empirical covariance matrix to obtain finite sample results for noise distributions satisfying the following conditions.

Assumption 2.3.11 (Noise conditions) *For all* $i \in [p]$, *we have*
(i) ***Sub-Gaussian noise:*** N_i/σ_i *is sub-Gaussian with parameter* ν.
(ii) ***Bounded-moment noise:*** $(\mathbb{E}\left[N_i/\sigma_i\right])^{4m} \leq K_m$, *for a positive integer* m *and positive constant* K_m.

Theorem 2.3.12 (Sample complexity) *If* $\lambda_n \geq \eta_1(n, p, \delta)$ *and* $n \geq \eta_2(p, \varepsilon, \delta)$, *then* $\left|\widehat{\mathbf{B}} - \mathbf{B}^*\right| \leq \varepsilon$, *with probability* $1 - \delta$, *where*

(i) for sub-Gaussian noise (Assumption 2.3.11(i)):

$$\eta_1(n, p, \delta) = MC_1\sqrt{(2/n)\log\left(2p/\sqrt{\delta}\right)}$$
$$\eta_2(p, \varepsilon, \delta) = 2(C_1C/\varepsilon)^2\log\left(2p/\sqrt{\delta}\right),$$

(ii) for bounded moment noise (Assumption 2.3.11(ii)):

$$\eta_1(n,p,\delta) = MC_2\left({}^{p^2}\!/\!_{(n^m\delta)}\right)^{1/2m}$$
$$\eta_2(p,\varepsilon,\delta) = \left({}^{C_2C}\!/\!_{\varepsilon}\right)^2\left({}^{p^2}\!/\!_{\delta}\right)^{1/m}$$

with

$$C = c4M^2(1+B_{\max})\sigma^2_{\max},$$
$$C_1 = \sqrt{128}(1+4\nu^2)(\max_i \mathbf{\Sigma}^*_{i,i}),$$
$$C_2 = 2(\max_i \mathbf{\Sigma}^*_{i,i})(C_m(C_m(K_m+1)+1))^{1/2m},$$

and c is defined in Theorem 2.3.10. Further, thresholding $\widehat{\mathbf{B}}$ at the level ε we get that $\mathcal{S}(\widehat{\mathbf{B}}) = \mathcal{S}(\mathbf{B}^*)$ and $\widehat{\mathsf{G}} = \mathsf{G}^*$.

2.4 Learning SEMs with Known Error Variances

Next, we focus our attention on the problem of learning SEMs when the error variances are known upto a constant factor. We will consider SEMs $(\mathsf{G}, \mathbf{B}, \{\alpha\sigma_i^2\})$ where $\{\sigma_i^2\}_{i=1}^p$ are known (to the learner) and $\alpha > 0$ is some unknown constant. Identifiability of this class of SEMs was proved by [LB14] under a *faithfulness* assumption. However, we will merely assume that $(\mathsf{G}, \mathbf{B}, \{\alpha\sigma_i^2\})$ is causal minimal, i.e., $\mathcal{S}(\mathbf{B}) = \mathsf{E}$ — this ensures that the distribution $\mathcal{P}(X)$ defined by the SEM is causal minimal to the DAG $\mathsf{G} = ([p], \mathsf{E})$. An immediate consequence of Proposition 2.3.2 is the following observation about terminal vertices:

Proposition 2.4.1 *Let $(\mathsf{G}, \mathbf{B}, \{\alpha\sigma_i^2\})$ be an SEM over X with precision matrix $\mathbf{\Omega}$, $\{\sigma_i^2\}_{i=1}^p$ known and $\alpha > 0$ is some unknown constant. Then, i is a terminal vertex in G if and only if $i \in \operatorname{argmin} \mathbf{diag}(\mathbf{\Omega} \circ \mathbf{D})$, where $\mathbf{D} = \mathbf{Diag}(\sigma_1^2, \ldots, \sigma_p^2)$.*

Thus, when the error variances are known upto a constant factor, Algorithm 1 can be used to learn SEMs, under the assumption of causal minimality, by setting $\mathbf{D} = \mathbf{Diag}(\sigma_1^2, \ldots, \sigma_p^2)$. Consequently, we have the following result about learning SEMs with known error variances:

Theorem 2.4.1 *Let* $(\mathsf{G}, \mathbf{B}, \{\alpha\sigma_i^2\})$ *be an SEM over* X*, with precision matrix* $\mathbf{\Omega}$ *and* $\{\sigma_i^2\}_{i=1}^p$ *known. Then, if* $(\mathsf{G}, \mathbf{B}, \{\alpha\sigma_i^2\})$ *is causal minimal and given* $\mathbf{\Omega}$, $\mathbf{D} = \mathbf{Diag}(\sigma_1^2, \ldots, \sigma_p^2)$ *as input, Algorithm 1 returns a unique* $(\widehat{\mathsf{G}}, \widehat{\mathbf{B}})$ *such that* $\widehat{\mathsf{G}} = \mathsf{G}$ *and* $\widehat{\mathbf{B}} = \mathbf{B}$.

Misspecified Error Variances. Our algorithm can also be used to learn SEMs with misspecified error variances as considered by [LB14]. For instance, if the true SEM is $(\mathsf{G}, \mathbf{B}, \{\sigma_i^2\})$ while the input to Algorithm 1 is $\mathbf{D} = \mathbf{Diag}((\sigma_1')^2, \ldots, (\sigma_p')^2)$, then it is straightforward to verify that the following condition is sufficient to ensure that Algorithm 1 still recovers the structure and parameters of the SEM correctly:

$$\sum_{l \in \phi_{\mathsf{G}[m,\tau]}(j)} B_{l,j}^2 > \frac{\alpha_{\max}}{\alpha_{\min}} - 1,$$
$$(\forall j \in \mathsf{V}[m,\tau] \wedge \phi_{\mathsf{G}[m,\tau]}(j) \neq \varnothing, m \in [p], \tau \in \mathcal{T}),$$

where $\alpha_{\max} \stackrel{\text{def}}{=} \max\{{}^{(\sigma_i')^2}/_{\sigma_i^2} \mid i \in [p]\}$ (similarly $\alpha_{\min}$). Next, we obtain statistical guarantees for our algorithm for learning SEMs with known error variances.

2.4.1 Statistical Guarantees for Estimation

In order to learn SEMs with known error variances from a finite number of samples, we make the following assumptions:

Assumption 2.4.2 *Given an SEM* $(\mathsf{G}, \mathbf{B}, \{\alpha\sigma_i^2\})$ *with precision matrix* $\mathbf{\Omega}$ *and noise variances* $\{\sigma_i^2\}_{i=1}^p$ *known to the learner; let* $\mathbf{\Omega}_{(m,\tau)}$ *denote the inverse covariance matrix over* $X_{\mathsf{V}[m,\tau]}$*. Then,*

(i) $\forall i \in \mathsf{V}[m,\tau], m \in [p]$*, and* $\tau \in \mathcal{T}_{\mathsf{G}}$*, such that* $\phi_{\mathsf{G}[m,\tau]}(i) \neq \varnothing$*:*

$$\sum_{l \in \phi_{\mathsf{G}[m,\tau]}(i)} \left(\frac{\sigma_i^2}{\sigma_l^2}\right) B_{l,i}^2 > 8\alpha M \lambda_n,$$

(ii) $\min\{\left|(\Omega_{(m,\tau)})_{i,j}\right| \mid (\Omega_{(m,\tau)})_{i,j} \neq 0, (i,j) \in \mathsf{V}[m,\tau] \times \mathsf{V}[m,\tau], m \in [p], \tau \in \mathcal{T}_{\mathsf{G}}\} > 4M\lambda_n$,

(iii) for all $i \in [p]$, $\sigma_i^2 \in o(1/4\alpha M\lambda_n)$.

Using CLIME to estimate and update the precision matrix, it is easy to verify that Theorem 2.3.12 holds for SEMs with known error variances satisfying Assumption 2.4.2, with $\sigma^2_{\max}$ and $\sigma^2_{\min}$ replaced by $\alpha\sigma^2_{\max}$ and $\alpha\sigma^2_{\min}$, respectively. Thus, given a data set of n samples drawn from an SEM satisfying Assumption 2.4.2, with autoregression matrix $\mathbf{B}^*$ and DAG structure $\mathsf{G}^* = ([p], \mathsf{E}^*)$, we have the following results for sub-Gaussian and bounded-moment noise:

Remark 2.4.3 *If* $\lambda_n \geq \eta_1(n, p, \delta)$, *and* $n \geq \eta_2(p, \varepsilon, \delta)$, *then, under Assumption 2.4.2, Algorithm 1 with* $\mathbf{D} = \mathbf{Diag}(\{\sigma_i^2\})$ *returns an estimator* $\widehat{\mathbf{B}}$ *such that* $\left|\widehat{\mathbf{B}} - \mathbf{B}^*\right|_\infty \leq \varepsilon$, *with probability at least* $1 - \delta$, *where for*

(i) sub-Gaussian noise we have $\eta_1(n, p, \delta) = \mathcal{O}\left((d/\sqrt{n})\sqrt{\log(p/\sqrt{\delta})}\right)$ *and* $\eta_2(p, \varepsilon, \delta) = \mathcal{O}\left((d^4/\varepsilon^2)\log(p/\sqrt{\delta})\right)$, *and*

(ii) noise with bounded moments $\eta_1(n, p, \delta) = \mathcal{O}\left((d/\sqrt{n})(p/\sqrt{\delta})^{1/m}\right)$ *and* $\eta_2(p, \varepsilon, \delta) = \mathcal{O}\left((d^4/\varepsilon^2)(p^2/\delta)^{1/m}\right)$.

Further, thresholding $\widehat{\mathbf{B}}$ *at the level* ε, *we have* $\mathcal{S}(\widehat{\mathbf{B}}) = \mathsf{E}^*$.

The above remark follows from the fact that $M = \mathcal{O}(d)$ which follows from Proposition A.0.2 in Appendix A.

In the population setting, i.e., given the true precision matrix, our algorithm can be implemented by storing the diagonal of the precision matrix separately and sorting it once which takes $\mathcal{O}(p \log p)$ time. In each iteration, updating the precision matrix in line 5 takes $\mathcal{O}(d)$ time since $\mathbf{\Omega}_{*,i}$ and $\mathbf{\Omega}_{i,*}$ are d-sparse. Updating the diagonal takes $\mathcal{O}(d \log p)$ time, while searching for the minimum diagonal element takes $\mathcal{O}(\log p)$ time. Therefore, Algorithm 1 computes the $\widehat{\mathbf{B}}$ matrix in $\mathcal{O}(p(d + d \log p))$ time. In the population setting, the computational complexity of [LB14]'s algorithm is $\mathcal{O}\left(p2^{2(w+1)(w+d)}\right)$, where w is the tree-width of the DAG structure of the true SEM and $d = \max\{|\mathsf{N}(i)|\}$. Note that the population version of our algorithm can still

be used in the finite sample setting if the precision matrix is estimated accurately enough.

In the finite sample setting, the computational complexity of our algorithm is dominated by the steps for estimating and updating the precision matrix — the latter depends on how well the sparsity pattern of the precision matrix is estimated. First, we analyze the computational complexity of our algorithm assuming exact support recovery, then we analyze the worst-case performance of our algorithm without assuming sparsity of the estimated precision matrix. Estimating the precision matrix can be done by solving p linear programs in $2p$-dimension and with $4p$ constraints. The smoothed complexity of this step is $\mathcal{O}\left(p^3 \log(p/\sigma)\right)$ when using interior point LP solvers [DST11], where σ^2 is variance of the Gaussian perturbations [2]. Next observe that $\left|\mathbf{\Omega}^* - \widehat{\mathbf{\Omega}}\right|_\infty \leq \left|\mathbf{B}^* - \widehat{\mathbf{B}}\right|_\infty \leq \varepsilon$. By thresholding $\widehat{\mathbf{\Omega}}$ at the level ε, each time the precision matrix is updated, we can ensure exact support recovery in each iteration. Thus, in the UPDATE function $\widehat{\pi}(i) = \pi_{\mathsf{G}^*}(i)$ and $\left|\widehat{\mathsf{S}}_j\right| \leq d \leq p$. Therefore, the UPDATE function takes $\mathcal{O}\left(d^4 \log(d/\sigma)\right)$ operations, leading to an overall complexity of $\widetilde{\mathcal{O}}\left(p^3 + pd^4\right)$. In the worst case, i.e., without any thresholding, $\widehat{\mathbf{\Omega}}$ can be dense. Therefore, the UPDATE function might re-estimate the full precision matrix over $p-t$ variables in iteration t, which takes $\mathcal{O}\left((p-t)^4 \log((p-t)/\sigma)\right)$ operations, leading to an overall complexity of $\widetilde{\mathcal{O}}\left(p^5\right)$. Thus, in the finite sample setting the complexity of our algorithm is between $\widetilde{\mathcal{O}}\left(p^3 + pd^4\right)$ and $\widetilde{\mathcal{O}}\left(p^5\right)$. Note that [LB14]'s analysis of the computational complexity of their algorithm assumes perfect support recovery of the precision matrix. In this regime, the computational complexity of their method is $\mathcal{O}\left(p2^{2(w+1)(w+d)} + p^3\right)$, including the step to estimate the precision matrix using graphical Lasso [FHT08], where w is the tree-width of the true DAG. However, without thresholding the output of graphical Lasso can be dense leading to a worst-case computational complexity that is exponential in p.

[2]The worst-case complexity of interior point methods for solving LPs is $\mathcal{O}\left(p^3 L\right)$ where L " is a parameter measuring the precision needed to perform the arithmetic operations exactly" and grows as $\Omega\left(p\right)$ [ST03]. However, interior-point methods work much more efficiently in practice and have an average complexity of $\mathcal{O}\left(p^3 \log p\right)$ (see [ST03] and the references therein).

2.5 Information-theoretic Lower Bounds

In this section we obtain information-theoretic lower bounds on the number of samples required to learn the DAG structure SEMs exactly. Let $\mathbf{\Theta}(\mathsf{G}) = (\mathbf{B}, \{\sigma_i^2\})$ represent the set of parameters of an SEM satisfying Assumption 2.3.1, for a given DAG G. To obtain information-theoretic lower bounds we view the inference process as a communication channel where nature picks a true DAG G, and samples the parameters $\mathbf{\Theta}(\mathsf{G}) = (\mathbf{B}, \{\sigma_i^2\})$ for the SEM from some distribution $\mathcal{P}$ over parameters, and then generates the data matrix $\mathbf{X}$ from the SEM $(\mathsf{G}, \mathbf{B}, \{\sigma_i^2\})$. A decoder ζ then maps the data matrix $\mathbf{X}$ to a DAG $\zeta(\mathbf{X})$. We define the following minimax estimation error:

$$p_{\text{err}} = \inf_{\zeta} \sup_{\mathsf{G} \in \mathcal{G}_{p,d}} \sup_{\mathcal{P}} \int_{\mathbf{\Theta}(\mathsf{G})} \Pr\left\{\zeta(\mathbf{X}) \neq \mathsf{G} \mid \mathbf{\Theta}\right\} \mathcal{P}(\mathbf{\Theta}(\mathsf{G})), \tag{2.9}$$

where the second supremum in the equation above is over all distributions $\mathcal{P}$ over parameters $\mathbf{\Theta}$. Note that this is a stronger notion of minimax error than is typically considered in the literature for learning undirected graphical models, c.f. [SW12].

Theorem 2.5.1 *If the number of samples n is less than $c(d \log p + {}^{d^2}/_{p})$, where c is an absolute constant, then any decoder ζ fails to recover the correct structure with probability of error $p_{\text{err}} \geq {}^{1}/_{2}$.*

The proof of the above theorem follows from Theorem 2 in our paper [GH17c], and the fact that in our case the DAG structure is uniquely identifiable from the parameters $\mathbf{\Theta}$. In [GH17c] we obtained information-theoretic lower bounds on learning the structure of Bayesian networks from observational data. We showed that $\Omega\left(k \log p\right)$ samples are required by any method for recovering the DAG structure of a Bayesian network upto Markov equivalence, with high probability, where k is the maximum number of parents of a node. Our techniques involved computing tight lower bounds on the number of DAGs over p variables and at most k parents using combinatorial arguments, developing a new Fano's inequality for incorporating latent variables (unobserved parameters of the network), and finally deriving new inequalities for upper bounding the mutual information between two arbitrary Bayesian networks.

Thus, from Theorem 2.5.1 we conclude that our method for learning linear SEMs is optimal in the number of variables p.

2.6 Experiments

We validate our method using both synthetic experiments and on real-world data. The following paragraph describes our results on synthetic data.

2.6.1 Simulation Experiments

First we validate our theoretical results through simulation experiments. We generate random SEMs by first sampling Erdős-Rényi random DAGs and then set all the noise variances to $\sigma^2 = 0.8$. Note this is a sufficient condition for ensuring identifiability (Proposition 2.3.1). We sample edge weights from the uniform distribution over $[-1, -0.5] \cup [0.5, 1]$. To generate sub-Gaussian noise, we set the noise variables $N_i = \sigma_i R_i$, where R_i's are independent Rademacher random variables. We set the regularization parameter according to Theorem 2.3.12 and varied the number of samples as $Cd^2 \log p$, with C being the control parameter. Figure 2.1 shows the probability of correct structure recovery and the maximum absolute difference between the true edge weights and the learned edge weights, across 30 randomly sampled SEMs. From Figure 2.1 we observe that Theorem 2.3.12 indeed bears out in practice, and the results show a phase transition behavior for structure recovery.

Comparison with State-of-the-art Methods on Synthetic Data. We also compared the performance of our algorithm against three other state-of-the-art methods for learning SEMs, viz. MMHC [TBA06], GES [Chi03], and the PC algorithm [SGS00] on randomly generated SEMs. We used the implementation of the MMHC algorithm provided by the *bnlearn* R package, while the *pcalg* R package provided the implementations of the GES and PC algorithm. We implemented our method, along with the CLIME algorithm for inverse covariance estimation, in Python. We

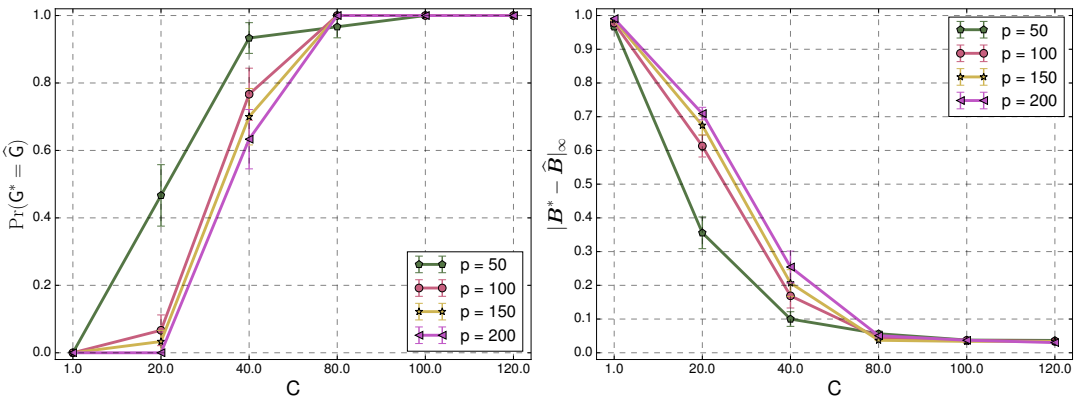


Figure 2.1.: (Left) Probability of correct structure recovery vs. number of samples, where the latter is set to $Cd^2 \log p$ with C being the control parameter and d being the maximum Markov blanket size. (Right) The maximum absolute difference between the true parameters and the learned parameters vs. number of samples.

performed two sets of experiments: in the first set of experiments we enforced the identifiability condition (Assumption 2.3.1) on the sampled SEMs, while in the second set of experiments we did not enforce the identifiability condition and generated the noise variances for each node using the uniform distribution over $[0.5, 1]$. To enforce identifiability of generated SEMs we simply set all the noise variances to $\sigma_i^2 = 0.8$. Note the equal noise variance is a sufficient condition for indentifiability (Proposition 2.3.1). Furthermore, The PC and MMHC algorithm require a parameter α, which is the target nominal type-I error rate of the conditional independence tests. We set to α to 0.05 for both the methods. Table 2.1 shows the mean accuracy, recall, execution time in seconds, for each method in the identifiable regime, computed across 30 randomly sampled SEMs, while Table 2.2 shows the results from the non-identifiable case. In the identifiable regime, our algorithm recovers the structure perfectly as is evident from the accuracy and recall scores while being comparable in speed to the other methods. Among the other methods, the PC algorithm has a recall score that is close to one indicating that it recovers the skeleton correctly most of the time. However, its poor accuracy, hovering at a mere 50%, indicates that it fails to orient most of the edges even though the true SEM is identifiable. MMHC and GES, which are heuristic algorithms, perform very poorly.

In the non-identifiable regime, as expected, our method is no-longer able to recover the graph perfectly. However, our method still achieves *close-to-perfect* structure recovery as is evidenced by its accuracy and recall scores, which are close to one. Also note that, while the PC algorithm has slightly better recall than our method, its accuracy is very poor. Therefore, our method is to be preferred over the PC algorithm. Other methods, on the other hand, achieve performance similar to that of the indentifiable case.

Next, we describe the performance of our method on real-world gene expression data sets.

Table 2.1.: Performance of our method vis-à-vis other state-of-the-art methods on Erdős-Rényi random DAGs that *satisfy the identifiability condition* given in Assumption 2.3.1.

Method	Accuracy	Recall	Seconds
p = 50			
Ours	**1.00 ± 0.00**	**1.00 ± 0.00**	**0.12 ± 0.01**
MMHC	0.53 ± 0.03	0.55 ± 0.03	0.25 ± 0.01
GES	0.24 ± 0.02	0.32 ± 0.02	0.32 ± 0.01
PC	0.56 ± 0.01	1.00 ± 0.00	0.18 ± 0.00
p = 100			
Ours	**1.00 ± 0.00**	**1.00 ± 0.00**	0.92 ± 0.01
MMHC	0.53 ± 0.02	0.57 ± 0.02	0.95 ± 0.02
GES	0.20 ± 0.01	0.34 ± 0.02	0.53 ± 0.01
PC	0.52 ± 0.01	0.99 ± 0.01	**0.41 ± 0.01**
p = 150			
Ours	**1.00 ± 0.00**	**1.00 ± 0.00**	3.16 ± 0.02
MMHC	0.46 ± 0.02	0.53 ± 0.02	2.17 ± 0.03
GES	0.18 ± 0.01	0.35 ± 0.02	**0.75 ± 0.01**
PC	0.51 ± 0.01	0.98 ± 0.00	0.88 ± 0.02
p = 200			
Ours	**1.00 ± 0.00**	**1.00 ± 0.00**	9.22 ± 0.03
MMHC	0.49 ± 0.01	0.59 ± 0.01	3.83 ± 0.04
GES	0.16 ± 0.01	0.34 ± 0.01	**1.07 ± 0.02**
PC	0.49 ± 0.01	0.98 ± 0.00	1.36 ± 0.01

Table 2.2.: Performance of our method vis-à-vis other state-of-the-art methods on Erdős-Rényi random DAGs, when the true SEM does not satisfy the identifiability condition given by Assumption 2.3.1.

Method	Accuracy	Recall	Seconds
p = 50			
Ours	**0.97 ± 0.01**	0.97 ± 0.01	**0.12 ± 0.01**
MMHC	0.53 ± 0.03	0.56 ± 0.03	0.25 ± 0.01
GES	0.27 ± 0.02	0.36 ± 0.03	0.31 ± 0.01
PC	0.55 ± 0.01	**1.00 ± 0.00**	0.19 ± 0.00
p = 100			
Ours	**0.95 ± 0.01**	0.96 ± 0.01	0.93 ± 0.01
MMHC	0.54 ± 0.02	0.59 ± 0.02	0.96 ± 0.02
GES	0.20 ± 0.01	0.34 ± 0.02	0.54 ± 0.01
PC	0.54 ± 0.01	**0.99 ± 0.01**	**0.41 ± 0.01**
p = 150			
Ours	**0.96 ± 0.01**	0.96 ± 0.01	3.24 ± 0.02
MMHC	0.49 ± 0.01	0.56 ± 0.02	2.12 ± 0.02
GES	0.18 ± 0.01	0.33 ± 0.01	**0.74 ± 0.01**
PC	0.53 ± 0.01	**0.99 ± 0.00**	0.81 ± 0.01
p = 200			
Ours	**0.96 ± 0.01**	0.96 ± 0.00	9.44 ± 0.04
MMHC	0.46 ± 0.01	0.56 ± 0.01	3.74 ± 0.03
GES	0.14 ± 0.01	0.31 ± 0.01	**1.04 ± 0.02**
PC	0.50 ± 0.01	**0.98 ± 0.00**	1.38 ± 0.01

2.6.2 Real-world Experiments

Table 2.3.: Number of samples and variables in the various gene expression data sets, where the number of sampled variables denotes the $0.8n$ highest variance variables.

Dataset	Disease	# Samples (n)	# Variables (p)	# Sampled Variables ($0.8n$)
GSE13294	Colon cancer	155	54,675	124.0
GSE1476	Colon cancer	150	59,381	120.0
GSE17951	Prostate cancer	154	54,675	123.0
GSE18105	Colon cancer	111	54,675	88.0
GSE18638	Colon cancer	98	235,826	78.0
GSE1898	Liver cancer	182	21,794	145.0
GSE22219	Breast cancer	216	24,332	172.0

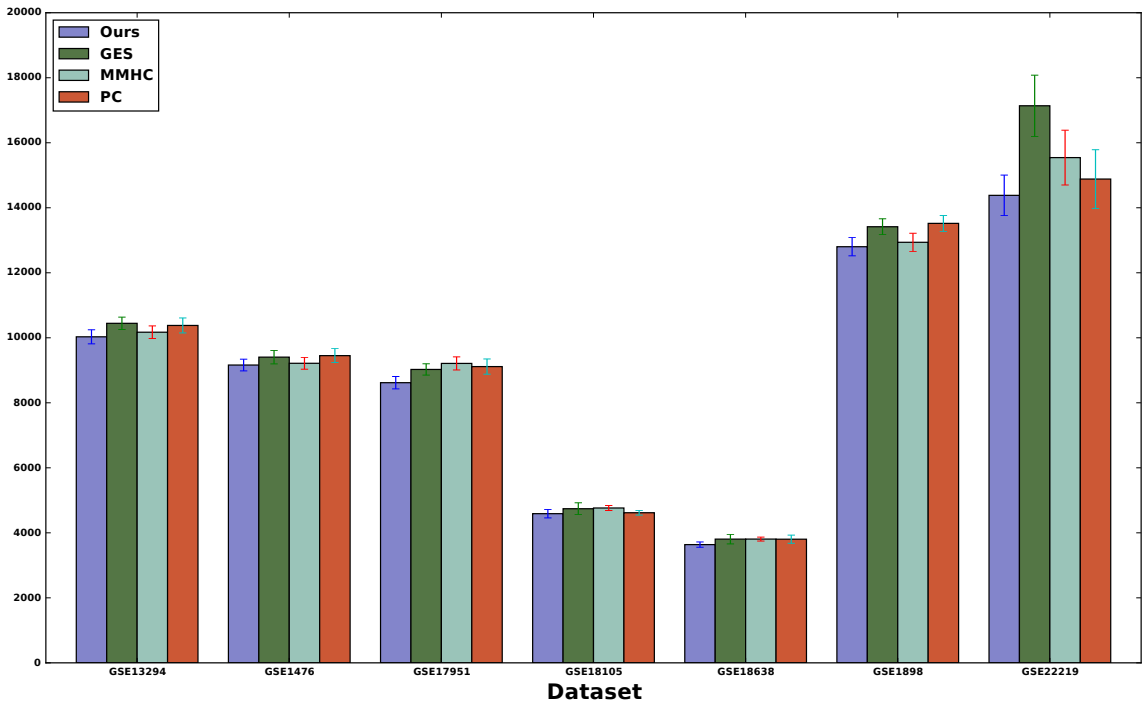


Figure 2.2.: The mean negative log likelihood of each method, on the test set, computed across 10 bootstrap runs.

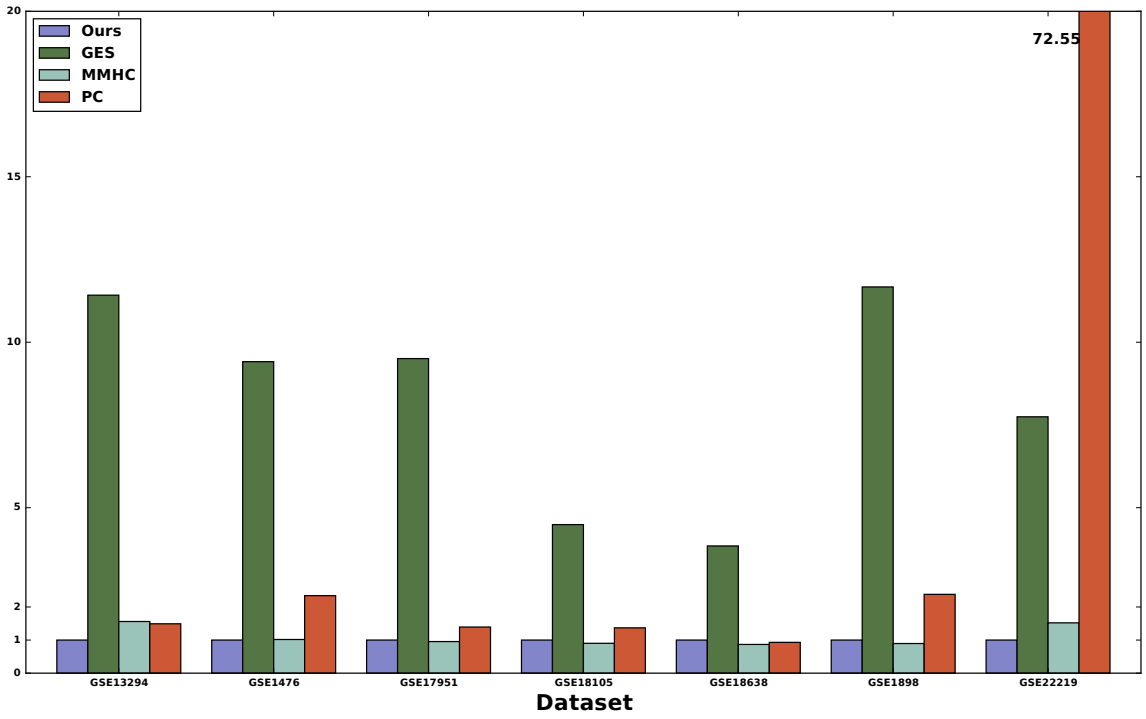


Figure 2.3.: The mean speed-up of our method vs. other state-of-the-art methods.

Finally, we compared the performance of our algorithm with the three state-of-the-art methods on 7 real-world gene expression data sets. The various attributes of the data sets, which are publicly available at the Gene Expression Omnibus (`http://www.ncbi.nlm.nih.gov/geo/`), are shown in Table 2.3. In these data sets, the ground truth DAG structure is not available. Therefore, the only way to evaluate the performance of the various algorithms is by comparing the test log-likelihood score of each method. While in the infinite sample limit the highest scoring DAG will coincide with the true DAG, when the number of samples is small, the DAG structure of the highest scoring DAG can be very different from the true DAG. In order to avoid a high-dimensional regime we selected the $\lfloor 0.8n \rfloor$ highest variance genes for analysis, as was done in [PB14]. We computed the average test negative-log-likelihood of each method on the 7 data sets across 10 bootstrap runs. In each bootstrap run, we created a training set by sampling n samples, with replacement, from the original data set and held out the remaining samples (those that were not picked in the sampling) as the test set. For our method, the regularization parameter was set to $0.01\sqrt{\log p/n}$, while for PC and MMHC the parameter α was set to 0.05. GES takes no parameters.

Figure 2.2 shows the mean test negative-log-likelihood, along with standard errors, of each method on the 7 gene expression data sets. Our method achieves the lowest test negative-log-likelihood on all seven data sets. This is noteworthy since MMHC and GES explicitly try to find the highest scoring structure while our method does not try to maximize any score. Further, unlike PC, MMHC, and GES, which return a PDAG, our method always returns a DAG.

Figure 2.3 shows the speed-up of our method with respect to the other three methods. On the largest and third largest data set (GSE22219) our method is close to 2 times faster than MMHC, 72 times faster than PC and around 10 times faster than GES. On five out of seven data sets our method achieves speed up of around 10 as compared to GES.

2.7 Summary

In this chapter, we considered the problem of learning linear structural equation models (SEMs) from observational data. We presented a polynomial time algorithm for learning linear SEMs under an identifiability condition that encompassed the homoscedastic noise case. The latter case had been considered by [LB14], who had presented a super-exponential time ℓ_0-penalized score-based method. Our polynomial time algorithm stemmed from the key observation that under our identifiability condition terminal vertices or sink nodes can be efficiently identified from the precision (inverse covariance) matrix. While we crucially leveraged this *structure* in our problem, our algorithm represents a new approach for learning DAGs from data: that is using the precision matrix to learn DAGs. The sparsity pattern of the precision matrix already gives the *moral graph* (undirected graph) of the corresponding DAG, which contains a super-set of the edges of the DAG, under faithfulness assumptions. By coming up with strategies to recover the DAG from the moral graph, by essentially pruning extra edges and orienting the remaining edges, we are able to leverage the plethora of computationally and statistically efficient algorithms available for learning undirected graphs from high-dimensional data. Another advantage of our approach is that we directly learn the edge weight matrix from data, whereas if one were to use the PC algorithm then recovering the edge weight matrix is a two step process involving learning the structure first and then performing regressions to learn the parameters (edge weights). An interesting avenue for future work is extending our approach to learning general (non-linear) SEMs.

3 LEARNING GRAPHICAL GAMES

In this chapter, we focus on the problem of learning polymatrix games — which are a class of multi-player graphical games — from data. Specifically, we assume that the behavior of a group of self-interested agents or players are described by a polymatrix game. We assume that we observe joint actions of these players, which may not be in equilibria, across multiple repetitions of the game. The goal is then to recover the structure and parameters of the game from the observations of the joint actions alone. In the next section, we formalize this problem introduce our notations, but first, we state our main contributions.

We propose an $\ell_{1,2}$ group-regularized logistic regression method to learn polymatrix games, which has been considered by [GJ16] and is a generalization of linear influence games considered by [GH17b]. We make no assumptions on the latent payoff functions and show that our polynomial time algorithm recovers an ε-Nash equilibrium of the true game [1], with high probability, if the number of samples (observations of joint actions) is $\mathcal{O}\left(m^4 d^4 \log(pd)\right)$, where p is the number of players, d is the maximum degree of the game graph and m is the maximum number of pure-strategies of a player. Under slightly more stringent separability conditions on the payoff functions of the underlying game, we show that our method recovers the Nash equilibria set exactly. We further generalize the observation model from [GH17b] in the sense that we allow strategy profiles (or joint actions) in the non-Nash equilibria set to have zero measure. This should be compared with the results of [GJ16] who show that learning tree-structured polymatrix games is NP-hard under a max-margin setting. We also obtain necessary conditions on learning polymatrix games and show that

[1] By the phrase "recovering the Nash equilibria" we mean that we learn a game with the same Nash equilibria as the true game. We use this phrase elsewhere in the paper for brevity.

$\Omega\left(d\log(pm)\right)$ samples are required by any method for recovering the PSNE set of a polymatrix

3.1 Preliminaries

First we introduce our notation and formally define the problem of learning polymatrix games from behavioral data.

Polymatrix games. A p-player *polymatrix game* is a graphical game where the set of nodes of the graph denote players and the edges correspond to two-player games. We will denote the graph by $G = ([p], E)$, where $[p] \stackrel{\text{def}}{=} \{1, \ldots, p\}$ is the vertex set and $E \subseteq [p] \times [p]$ is set of directed edges. An edge $(i, j) \in E$ denotes the directed edge $i \leftarrow j$. Each player i has a set of pure-strategies or actions $\mathcal{A}_i$, and the set of pure-strategy profiles or joint actions of all the p players is denoted by $\mathcal{A} = \times_{i\in[p]}\mathcal{A}_i$. We will denote $\mathcal{A}_{-\mathsf{i}} \stackrel{\text{def}}{=} \times_{j\in-\mathsf{i}}\mathcal{A}_j$. With each edge $(i, j) \in E$ is associated a payoff matrix $u^{i,j} : \mathcal{A}_i \times \mathcal{A}_j \to \mathbb{R}$, such that $u^{i,j}(x_i, x_j)$ gives the finite payoff of the i-th player (with respect to the j-th player), when player i plays $x_i \in \mathcal{A}_i$ and player j plays $x_j \in \mathcal{A}_j$. We assume that $(i, j) \in E$, if and only if $u^{i,j}(\cdot, \cdot) \neq 0$. Given a strategy profile $\mathbf{x} \in \mathcal{A}$, the total payoff, or simply the payoff, of the i-th player is given by the following potential function:

$$u^i(x_i, \mathbf{x}_{-\mathsf{i}}; G) = u^{i,i}(x_i) + \sum_{j\in\mathcal{N}_i} u^{i,j}(x_i, x_j), \tag{3.1}$$

where $\mathcal{N}_i(G) \stackrel{\text{def}}{=} \{j \in [p] | (i, j) \in E\}$ is the set of neighbors of i in the graph G, and $u^{i,i} : \mathcal{A}_i \to \mathbb{R}$ gives the (finite) *individual payoff* of i for playing x_i. We will denote the number of neighbors of player i by $d_i \stackrel{\text{def}}{=} |\mathcal{N}_i(G)|$, and the maximum degree of the graph G by $d = \max\{d_1, \ldots, d_p\}$. A polymatrix game $\mathcal{G} = (G, \mathcal{U})$ is then completely defined by a graph $G = ([p], E)$ and a collection of potential functions $\mathcal{U}(G) = \{u^i : \mathcal{A}_{-\mathsf{i}} \to \mathbb{R}\}_{i\in[p]}$, where each of the payoff functions $u^i(\cdot; G)$ decomposes according to (3.1). Finally, we will also assume that the number of strategies of

each player, $m_i \stackrel{\text{def}}{=} |\mathcal{A}_i|$, is non-zero and $\mathcal{O}(1)$ with respect to p and d, and that $m \stackrel{\text{def}}{=} \max\{m_i\}$.

Nash equilibria of polymatrix games. The set of pure-strategy Nash equilibria (PSNE) of the game $\mathcal{G} = (G, \mathcal{U})$ is given by the set of strategy profiles where no player has any incentive to unilaterally deviate from its strategy given the strategy profiles of its neighbors, and is defined as follows:

$$\mathcal{NE}(\mathcal{G}) = \left\{ \mathbf{x} \in \mathcal{A} \;\middle|\; x_i \in \operatorname*{argmax}_{a \in \mathcal{A}_i} u^i(a, \mathbf{x}_{-\mathrm{i}}) \right\}. \tag{3.2}$$

The set of ε-Nash equilibria of the game $\mathcal{G}$ are those strategy profiles where each player can gain at most ε payoff by deviating from its strategy, and is defined as follows:

$$\varepsilon\text{-}\mathcal{NE}(\mathcal{G}) = \Big\{ \mathbf{x} \in \mathcal{A} \mid u^i(x_i, \mathbf{x}_{-\mathrm{i}}) \geq u^i(a, \mathbf{x}_{-\mathrm{i}}) - \varepsilon, \forall a \in \mathcal{A}_i \text{ and } \forall i \in [p] \Big\}. \tag{3.3}$$

Observation model. Without getting caught up in the dynamics of gameplay — something that is difficult to observe or reason about in real-world scenarios — we abstract the learning problem as follows. Assume that we are given "noisy" observations of strategy profiles, or joint actions, $\mathcal{D} = \{\mathbf{x}^{(l)} \in \mathcal{A}\}_{l \in [n]}$ drawn from a game $\mathcal{G} = (G, \mathcal{U})$. The noise process models our uncertainty over the individual actions of the players due to observation noise, for instance, when we observe the actions through a noisy channel, or due to the unobserved dynamics of gameplay during which equilibrium is reached. By "observations drawn from a game" we simply mean that there exists a distribution $\mathcal{P}$, from which the strategy profiles are drawn, satisfying the following condition:

$$\forall \mathbf{x}, \mathbf{x}' \text{ such that } \mathbf{x} \in \mathcal{NE}(\mathcal{G}) \text{ and } \mathbf{x}' \in \mathcal{A} \setminus \mathcal{NE}(\mathcal{G}) : o\mathcal{P}(\mathbf{x}) > \mathcal{P}(\mathbf{x}').$$

The above condition ensures that the signal level is more than the noise level. This should be compared with the observation model of [GH17b], who assume that $\forall \mathbf{x}' \in \mathcal{A} \setminus \mathcal{NE}(\mathcal{G}), \mathcal{P}(\mathbf{x}') > 0$. Our observation model thus encompasses specific observation

models considered in prior literature [HO15, GH16]: the global and local noise model. The global noise model is parameterized by a constant $q \in ({}^{\mathcal{NE}(\mathcal{G})}/_{|\mathcal{A}|}, 1)$ such that the probability of observing a strategy profile $\mathbf{x} \in \mathcal{A}$ is given by a mixture of two uniform distributions:

$$\mathcal{P}_g(\mathbf{x}; \mathcal{G}) = q\frac{\mathbf{1}\left[\mathbf{x} \in \mathcal{NE}(\mathcal{G})\right]}{|\mathcal{NE}(\mathcal{G})|} + (1-q)\frac{\mathbf{1}\left[\mathbf{x} \notin \mathcal{NE}(\mathcal{G})\right]}{|\mathcal{A}| - |\mathcal{NE}(\mathcal{G})|}. \tag{3.4}$$

In the local noise model, we observe strategy profiles $\mathbf{x}$ from the PSNE set with each entry (strategy) corrupted independently. Therefore, in the local noise model we have the following distribution over strategy profiles:

$$\mathcal{P}_l(\mathbf{x}; \mathcal{G}) = \frac{1}{|\mathcal{NE}(\mathcal{G})|} \sum_{\mathbf{y} \in \mathcal{NE}(\mathcal{G})} \prod_{i=1}^{p} (q_i)^{\mathbf{1}[x_i = y_i]} \left(\frac{1-q_i}{m_i - 1}\right)^{\mathbf{1}[x_i \neq y_i]}, \tag{3.5}$$

with $q_i > 0.5$ for all $i \in [p]$.

In essence, we assume that we observe multiple "stable outcomes" of the game, which may or may-not be in equilibria. Treating the outcomes of the game as "samples" observed across multiple "plays" of the same game is a recurring theme in the literature for learning games (c.f. [HO15], [GH16], [GH17b], [GJ16]).

Problem: The learning problem then corresponds to recovering a game $\widehat{\mathcal{G}} = (\widehat{\mathsf{G}}, \widehat{\mathcal{U}})$ from $\mathcal{D}$ such that $\mathcal{NE}(\widehat{\mathcal{G}}) = \mathcal{NE}(\mathcal{G})$ with high probability.

Given that computing a single Nash equilibria is PPAD-complete [DGP09], any efficient learning algorithm must learn the game without explicitly computing or enumerating the Nash equilibria of the game. It has also been shown that even computing an ε-Nash equlibria is hard under the exponential time hypothesis for PPAD [Rub16]. We also emphasize that we do not observe any information about the latent player payoffs, and neither do we impose any restrictions on the payoffs for obtaining our ε-Nash equilibria guarantees. Also, note that in our definition of the learning problem, we do not impose any restriction on the "closeness" of the recovered graph $\widehat{\mathsf{G}}$ to the true graph G. This is because multiple graphs G can give rise to the same PSNE set under different payoff functions and thus be *unidentifiable* from observations of joint actions alone (see section 4.4.1 of [HO15] for a counter example.)

3.2 Related Work

Recovering the underlying game from behavioral data is an important tool in exploratory research in political science and behavioral economics, and recent times have seen a surge of interest in such problems (c.f. [IO14,HO15,GH16,GJ16,GH17b]). For instance, in political science, [IO14] identified the *most influential* senators in the U.S congress — a small coalition of senators whose collective behavior forced every other senator to a unique choice of action — by learning a *linear influence game* from congressional voting records. [GJ16] showed that a *tree-structured polymatrix game* [2] learned from U.S. Supreme Court data was able to recover the known ideologies of the justices. However, many open problems remain in this area of active research. One such problem is whether there exists efficient (polynomial time) methods for learning polymatrix games [Jan68] from noisy observations of strategic interactions. This is the focus of the current paper.

Various methods have been proposed for learning games from data. [HO15] proposed a maximum-likelihood approach to learn "linear influence games" — a class of parametric graphical games with linear payoffs. However, in addition to being exponential time, the maximum-likelihood approach of [HO15] also assumed a specific observation model for the strategy profiles. [GH16] proposed a polynomial time algorithm, based on ℓ_1-regularized logistic regression, for learning linear influence games. They again assumed the specific observation model proposed by [HO15] in which the strategy profiles (or joint actions) were drawn from a mixture of uniform distributions: one over the pure-strategy Nash equilibria (PSNE) set, and the other over the complement of the PSNE set. [GH17b] obtained necessary and sufficient conditions for learning linear influence games under arbitrary observation model. Finally, [GJ16] use a discriminative, max-margin based approach, to learn tree structured polymatrix games. However, their method is exponential time and they show that learning polymatrix games is NP-hard under this max-margin setting, even when the class of

[2] [GJ16] call their game a *potential game* even though the formulation of their game is similar to ours.

graphs is restricted to trees. Furthermore, all the aforementioned works, with the exception of [GJ16], consider binary strategies only. In this paper, we propose a polynomial time algorithm for learning polymatrix games, which are non-parametric graphical games where the pairwise payoffs between players are characterized by matrices (or pairwise potential functions). In this setting, each player has a finite number of pure-strategies.

Finally, we conclude this section by referring the reader to the work of [JRVS11] who analyze $\ell_{1,2}$-regularized logistic regression for learning undirected graphical models. However, our setting differs from that of learning discrete graphical models in many ways. First, unlike discrete graphical models, where the underlying distribution over the variables is described by a potential function that factorizes over the cliques of the graph, we make no assumptions whatsoever on the generative distribution of data. Further, we are interested in recovering the PSNE set of a game, since the graph structure in generally unidentifiable from observational data, whereas [JRVS11] obtain guarantees on the graph structure of the discrete graphical model. As a result, our theoretical analysis and proofs differ significantly from those of [JRVS11].

3.3 Method and Theoretical Guarantees

In this section, we describe our method for learning polymatrix games from observational data. The individual and pairwise payoffs can be equivalently written, in linear form, as follows:

$$u^{i,i}(x_i) = (\boldsymbol{\theta}^{i,0})^T \mathbf{f}^{i,0}(x_i),$$
$$u^{i,j}(x_i, x_j) = (\boldsymbol{\theta}^{i,j})^T \mathbf{f}^{i,j}(x_i, x_j),$$

where for $j \in \mathcal{N}_i$, $\mathbf{f}^{i,j}(x_i, x_j) = (\mathbf{1}\left[x_i = a, x_j = b\right])_{a \in \mathcal{A}_i, b \in \mathcal{A}_j}$ and

$$\boldsymbol{\theta}^{i,j} = (\theta^{i,j}_{a,b})_{a \in A_i, b \in A_j},$$
$$\boldsymbol{\theta}^{i,0} = (\theta^{i,0}_{a})_{a \in A_i}, \text{ and}$$
$$\mathbf{f}^{i,0}(x_i) = (\mathbf{1}\left[x_i = a\right])_{a \in A_i}.$$

Note that $\mathbf{f}^{i,j} \in \{0,1\}^{(m_i m_j)}$, $\boldsymbol{\theta}^{i,j} \in \mathbb{R}^{(m_i m_j)} \neq \mathbf{0}$, $\mathbf{f}^{i,0}(x_i) \in \{0,1\}^{m_i}$, and $\boldsymbol{\theta}^{i,0} \in \mathbb{R}^{m_i}$. Let

$$\begin{aligned}\boldsymbol{\theta}^i &\stackrel{\text{def}}{=} (\boldsymbol{\theta}^{i,0}, \boldsymbol{\theta}^{i,1}, \ldots, \boldsymbol{\theta}^{i,i-1}, \boldsymbol{\theta}^{i,i+1}, \ldots, \boldsymbol{\theta}^{i,p}),\\ \mathbf{f}^i(x_i, \mathbf{x}_{-\mathsf{i}}) &\stackrel{\text{def}}{=} (\mathbf{f}^{i,0}(x_i), \mathbf{f}^{i,1}(x_i, x_1), \ldots, \mathbf{f}^{i,i-1}(x_i, x_{i-1}),\\ &\qquad \mathbf{f}^{i,i+1}(x_i, x_{i+1}), \ldots, \mathbf{f}^{i,p}(x_i, x_p)),\end{aligned} \tag{3.6}$$

with

$$\begin{aligned}\boldsymbol{\theta}^{i,j} &= \mathbf{0} \text{ for } j > 0 \wedge j \notin \mathcal{N}_i,\\ \boldsymbol{\theta}^i &\in \mathbb{R}^{(m_i + \sum_{j \in -\mathsf{i}} m_i m_j)}, \text{ and}\\ \mathbf{f}^i(x_i, \mathbf{x}_{-\mathsf{i}}) &\in \{0,1\}^{(m_i + \sum_{j \in -\mathsf{i}} m_i m_j)}.\end{aligned}$$

Thus the payoff for the i-th player can be written, in linear form, as:

$$u^i(x_i, \mathbf{x}_{-\mathsf{i}}) = (\boldsymbol{\theta}^i)^T \mathbf{f}^i(x_i, \mathbf{x}_{-\mathsf{i}}). \tag{3.7}$$

The learning problem then corresponds to learning the parameters $\boldsymbol{\theta}^i$ for each player i. The sparsity pattern of $\boldsymbol{\theta}^i$ identifies the neighbors of i. The way this differs from the binary strategies considered by [GH17b] is that the parameters $\boldsymbol{\theta}^i$ have a *group-sparsity* structure, i.e., for all $j > 0 \wedge j \notin \mathcal{N}_i$ the entire group of parameters $\boldsymbol{\theta}^{i,j}$ is zero. In order to ensure that the payoffs are finite, we will assume that the parameters for the i-th player belong to the set $\Theta^i \stackrel{\text{def}}{=} \{\mathbf{y} \in \mathbb{R}^{(m_i + \sum_{j \in -\mathsf{i}} m_i m_j)} \mid \|\mathbf{y}\|_\infty < \infty\}$.

Our approach for estimating the parameters $\boldsymbol{\theta}^i$ is to perform one-versus-rest multinomial logistic regression with $\ell_{1,2}$ group-sparse regularization. In more detail, we obtain estimators $\widehat{\boldsymbol{\theta}^i}$ by solving the following optimization problem for each $i \in [p]$:

$$\widehat{\boldsymbol{\theta}^i} = \operatorname*{argmin}_{\boldsymbol{\theta} \in \Theta^i} L^i(\mathcal{D}; \boldsymbol{\theta}) + \lambda \|\boldsymbol{\theta}\|_{1,2}, \tag{3.8}$$

$$L^i(\mathcal{D}; \boldsymbol{\theta}) = \frac{1}{n} \sum_{l=1}^{n} \ell^i(\mathbf{x}^{(l)}; \boldsymbol{\theta}), \tag{3.9}$$

$$\ell^i(\mathbf{x}; \boldsymbol{\theta}) = -\log\left(\frac{\exp(\boldsymbol{\theta}^T \mathbf{f}^i(x_i, \mathbf{x}_{-\mathsf{i}}))}{\sum_{a \in \mathcal{A}_i} \exp(\boldsymbol{\theta}^T \mathbf{f}^i(a, \mathbf{x}_{-\mathsf{i}}))}\right), \tag{3.10}$$

where $\|\boldsymbol{\theta}\|_{1,2} = \sum_{j\in[p]} \|\boldsymbol{\theta}_j\|_2$, with $\boldsymbol{\theta}_j$ being the j-th group of $\boldsymbol{\theta}$. When referring to a block of a matrix or vector we will use bold letters, e.g, $\boldsymbol{\theta}_j$ denotes the j-th group or block of $\boldsymbol{\theta}$, while θ_j denotes the j-th element of $\boldsymbol{\theta}$. In general, we define the $\ell_{a,b}$ group structured norm as follows: $\|\boldsymbol{\theta}\|_{a,b} = \left\|\left(\|\boldsymbol{\theta}_1\|_b, \ldots, \|\boldsymbol{\theta}_p\|_b\right)\right\|_a$. Also, when using group structured norms, we will use the group structure as shown in (3.6), i.e., we will assume that there are p groups and, in the context of the i-th player, the sizes of the groups are: $\{m_i, m_i m_1, \ldots, m_i m_{i-1}, m_i m_{i+1}, \ldots, m_i m_p\}$. Finally, we will define the support set of $\boldsymbol{\theta}^i$ as the set of all indices corresponding to the active groups, i.e., $S_i = \{(j,k) | j \in \{0\} \cup \mathcal{N}_i \text{ and } k \in [m_i] \text{ for } j = 0, k \in [m_i m_j] \text{ for } j > 0\}$, where j can be thought of as indexing the groups, while k can be thought of as the indexing the elements within the j-th group. Thus, $|S_i| = m_i + \sum_{j\in\mathcal{N}_i} m_i m_j$.

After estimating the parameters $\widehat{\boldsymbol{\theta}}^i$ for each $i \in [p]$, the payoff functions are simply estimated to be $\widehat{u}^i(x_i, \mathbf{x}_{-\mathsf{i}}) = (\widehat{\boldsymbol{\theta}}^i)^T \mathbf{f}^i(x_i, \mathbf{x}_{-\mathsf{i}})$. Finally, the graph $\widehat{\mathsf{G}} = ([p], \widehat{E})$ is given by the group-sparsity structure of $\widehat{u}^i$s, i.e., $\widehat{u}^{i,j}(\cdot,\cdot) \neq 0 \implies (i,j) \in \widehat{E}$.

First, we obtain sufficient conditions on the number of samples n to ensure successful PSNE recovery. Since our theoretical results depend on certain properties of the Hessian of the loss function defined above, we introduce the Hessian matrix in this paragraph. Let $\mathbf{H}^i(\mathbf{x}; \boldsymbol{\theta})$ denote the Hessian of $\ell^i(\mathbf{x}; \boldsymbol{\theta})$. A little calculation shows that the (j,k)-th block of the Hessian matrix for the i-th player is given as:

$$\begin{aligned}\mathbf{H}^i_{j,k}(\mathbf{x}; \boldsymbol{\theta}) = \sum_{a\in\mathcal{A}_i} \sigma^i(a, \mathbf{x}_{-\mathsf{i}}; \boldsymbol{\theta}) \mathbf{f}^{i,j}(a, x_j)(\mathbf{f}^{i,k}(a, x_k))^T - \\ \Big\{ \Big(\sum_{a\in\mathcal{A}_i} \sigma^i(a, \mathbf{x}_{-\mathsf{i}}; \boldsymbol{\theta}) \mathbf{f}^{i,j}(a, x_j) \Big) \times \\ \Big(\sum_{a\in\mathcal{A}_i} \sigma^i(a, \mathbf{x}_{-\mathsf{i}}; \boldsymbol{\theta}) \mathbf{f}^{i,k}(a, x_k) \Big)^T \Big\}, \end{aligned} \tag{3.11}$$

$$\sigma^i(x, \mathbf{x}_{-\mathsf{i}}; \boldsymbol{\theta}) = \frac{\exp(\boldsymbol{\theta}^T \mathbf{f}^i(x, \mathbf{x}_{-\mathsf{i}}))}{\sum_{a\in\mathcal{A}_i} \exp(\boldsymbol{\theta}^T \mathbf{f}^i(a, \mathbf{x}_{-\mathsf{i}}))}, \tag{3.12}$$

where we have overloaded the notation $\mathbf{f}^{i,j}(x_i, x_j)$ to also include $\mathbf{f}^{i,0}(x_i)$, i.e., we let $\mathbf{f}^{i,0}(x_i, x_0) \stackrel{\text{def}}{=} \mathbf{f}^{i,0}(x_i)$. We will denote the i-th expected Hessian matrix at any parameter $\boldsymbol{\theta} \in \Theta^i$ as $\mathbf{H}^i(\boldsymbol{\theta}) = \mathbb{E}[\mathbf{x}]\, \mathbf{H}^i(\mathbf{x}; \boldsymbol{\theta})$, and the i-th Hessian matrix at the true

parameter $\boldsymbol{\theta}^i$ as $\mathbf{H}^i(\boldsymbol{\theta}^i)$. We will also drop the superscript i from the i-th Hessian matrix, whenever clear from context. We will denote the finite sample version of $\mathbf{H}^i(\boldsymbol{\theta}^i)$ by $\mathbf{H}^i(\mathcal{D}, \boldsymbol{\theta}^i)$, i.e., $\mathbf{H}^i(\mathcal{D}, \boldsymbol{\theta}^i) = \frac{1}{n}\sum_{l=1}^{n} \mathbf{H}^i(\mathbf{x}^{(l)}, \boldsymbol{\theta}^i)$. Finally, we will denote the Hessian matrix restricted to the true support set S_i by: $\mathbf{H}^i(\cdot; \boldsymbol{\theta}^i_{S_i}) \in \mathbb{R}^{|S_i|\times|S_i|}$. In order to prove our main result, we will present a series of technical lemmas slowly building towards our main result. Detailed proofs of the lemmas are given in Appendix B.

The following lemma states that the i-th population Hessian is positive definite. Specifically, the i-th population Hessian evaluated at the true parameter $\boldsymbol{\theta}^i$, are positive definite with the minimum eigenvalue being $C_{\min}$. We prove the following lemma by showing that the loss function given by (3.10), when restricted to an arbitrary line, is strongly convex as long as the payoffs are finite.

Lemma 3.3.1 (Minimum eigenvalue of population Hessian) *For $\boldsymbol{\theta}^i \in \Theta^i$ we have that $C_{\min} \stackrel{\text{def}}{=} \lambda_{\min}(\mathbf{H}^i(\boldsymbol{\theta}^i)) > 0$.*

Given that population Hessian matrices are positive-definite, we then show that the finite sample Hessian matrices, evaluated at any parameter $\boldsymbol{\theta}_{S_i}$, are positive definite with high probability. We use tools from random matrix theory developed by [Tro12] to prove the following lemma.

Lemma 3.3.2 (Minimum eigenvalue of finite sample Hessian) *Let $\boldsymbol{\theta} \in \Theta^i$ be any arbitrary vector and let $\lambda_{\min}(\mathbf{H}^i(\boldsymbol{\theta}_{S_i})) \stackrel{\text{def}}{=} \lambda_{\min} > 0$. Then, if the number of samples satisfies the following condition:*

$$n \geq \frac{8(d_i+1)}{\lambda_{\min}} \log\left(\frac{m_i(1+d_i m)}{\delta}\right),$$

then $\lambda_{\min}(\mathbf{H}^i(\mathcal{D}; \boldsymbol{\theta}_{S_i})) \geq \frac{\lambda_{\min}}{2}$ with probability at least $1-\delta$ for some $\delta \in (0,1)$.

Now that we have shown that the loss function given by (3.10) is strongly convex (Lemmas 3.3.1 and 3.3.2), we exploit strong convexity to control the difference between the true parameter and the estimator $\left\|\boldsymbol{\theta}^i - \widehat{\boldsymbol{\theta}}^i\right\|_{1,2}$. However, before proceeding further, we need to bound the $\ell_{\infty,2}$ norm of the gradient, as done in the following lemma. We prove the lemma by using McDiarmid's inequality to show that in each

group the finite sample gradient concentrates around the expected gradient, and then use a union bound over all the groups to control the $\ell_{\infty,2}$ norm.

Lemma 3.3.3 (Gradient bound) *Let* $\|\mathbb{E}_{\mathbf{x}}[\nabla \ell^i(\mathbf{x}; \boldsymbol{\theta}^i)]\|_{\infty,2} = \nu$, *then we have that*

$$\left\|\nabla L^i(\mathcal{D}; \boldsymbol{\theta}^i)\right\|_{\infty,2} \leq \nu + \sqrt{\frac{2}{n}\log\Big(\frac{2(d_i+1)}{\delta}\Big)},$$

with probability at least $1-\delta$.

Note that the expected gradient at the parameter $\boldsymbol{\theta}^i$ does not vanish, i.e., we have that $\|\mathbb{E}_{\mathbf{x}}[\nabla \ell^i(\mathbf{x}; \boldsymbol{\theta}^i)]\|_{\infty,2} = \nu$. This is because of the mismatch between the generating distribution $\mathcal{P}$ and the softmax distribution used for learning the parameters, as in (3.10). Indeed, if the data were drawn from a Markov random field, which induces a softmax distribution on the conditional distribution of node given the rest of the nodes, the parameter $\nu = 0$. However this is not the case for us. An unfortunate consequence of this is that, even with an infinite number of samples, our method will not be able to recover the parameters $\boldsymbol{\theta}^i$ exactly. Thus, without additional assumptions on the payoffs, our method only recovers the ε-Nash equilibrium of the game.

With the required technical results in place, we are now ready to bound the quantity $\left\|\boldsymbol{\theta}^i - \widehat{\boldsymbol{\theta}}^i\right\|_{1,2}$. Our analysis has two steps. First, we bound the norm in the true support set, i.e., $\left\|\boldsymbol{\theta}^i_{S_i} - \widehat{\boldsymbol{\theta}}^i_{S_i}\right\|_{1,2}$. Then, we show that the norm of the difference between the true parameter and the estimator, outside the support set, is a constant factor (specifically 3) of the difference in the support set. For the first step with use a proof technique originally developed by [RBLZ08] in a different context, while the second step follows from matrix algebra and optimality of the estimator $\widehat{\boldsymbol{\theta}}^i$ for the problem (3.8).

The following technical lemma, which will be used later on in our proof to bound $\left\|\widehat{\boldsymbol{\theta}}^i_S - \boldsymbol{\theta}^i_S\right\|_{1,2}$, lower bounds the minimum eigenvalue of the i-th population Hessian at an arbitrary parameter $\boldsymbol{\theta} \in \Theta^i$, in terms of the minimum eigenvalue of the i-th population Hessian at the true parameter $\boldsymbol{\theta}^i$.

Lemma 3.3.4 (Minimum population eigenvalue at arbitrary parameter) *Let $\boldsymbol{\theta} \in \Theta^i$ be any vector. Then the minimum eigenvalue of i-th population Hessian matrix evaluated at $\boldsymbol{\theta}_{S_i}$ is lower bounded as follows:*

$$\lambda_{\min}(\mathbf{H}^i(\boldsymbol{\theta}_{S_i})) \geq \lambda_{\min}(\mathbf{H}^i(\boldsymbol{\theta}^i_{S_i})) - \frac{1}{4}(d_i+1)m^2 \left\|\boldsymbol{\theta}_{S_i} - \boldsymbol{\theta}^i_{S_i}\right\|_{1,2}.$$

Now, we are ready to bound the difference between the true parameter $\boldsymbol{\theta}^i$ and its estimator $\widehat{\boldsymbol{\theta}}^i$, in the true support set S_i.

Lemma 3.3.5 (Error of the i-th estimator on the support set) *If the regularization parameter and number of samples satisfy the following condition:*

$$\lambda \geq 2\left(\nu + \sqrt{\frac{2}{n}\log\left(\frac{2(d_i+1)}{\delta}\right)}\right),$$
$$n > \frac{2}{N(m,d_i)}\log\left(\frac{2(d_i+1)}{\delta}\right),$$

where $N(m,d_i) = \left\{{}^{C_{\min}}\!/_{(36m^2(d_i+1)^2)} - \nu\right\}^2$, and $C_{\min} \stackrel{\text{def}}{=} \lambda_{\min}(\mathbf{H}^i(\boldsymbol{\theta}^i_{S_i}))$; then with probability at least $1-\delta$, for some $\delta \in (0,1)$, we have:

$$\left\|\widehat{\boldsymbol{\theta}}^i_{S_i} - \boldsymbol{\theta}^i_{S_i}\right\|_{1,2} \leq \frac{6(d_i+1)}{C_{\min}}\lambda. \tag{3.13}$$

Next, we bound the difference between the true parameter $\boldsymbol{\theta}^i$ and its estimator $\widehat{\boldsymbol{\theta}}^i$.

Lemma 3.3.6 (Error of the i-th parameter estimator) *Under the same conditions on the regularization parameter and number of samples as in Lemma 3.3.5 we have, with probability at least $1-\delta$ for some $\delta \in (0,1)$,*

$$\left\|\widehat{\boldsymbol{\theta}}^i - \boldsymbol{\theta}^i\right\|_{1,2} \leq \frac{24(d_i+1)}{C_{\min}}\lambda.$$

Now that we have control over $\left\|\boldsymbol{\theta}^i - \widehat{\boldsymbol{\theta}}^i\right\|_{1,2}$ for all $i \in [p]$, we are ready to prove our main result concerning the sufficient number of samples needed by our method to guarantee PSNE recovery with high probability.

Theorem 3.3.7 *Let $\mathcal{G} = (G, \mathcal{U})$, with $\mathcal{U} = \{u^i : \mathcal{A}_{-\mathrm{i}} \to \mathbb{R}\}_{i\in[p]}$, be the true potential graphical game over p players and maximum degree d, from which the data set $\mathcal{D}$ is*

drawn. Let $\widehat{\mathcal{G}} = (\widehat{G}, \widehat{\mathcal{U}})$*, with* $\widehat{U} = \{\widehat{u}^i : \mathcal{A}_{-\mathsf{i}} \to \mathbb{R}\}_{i\in[p]}$*, be the game learned from the data set* $\mathcal{D}$ *by solving the optimization problem* (3.8) *for each* $i \in [p]$*. Then if the regularization parameter and the number of samples satisfy the condition:*

$$\lambda \geq 2\left(\nu + \sqrt{\frac{2}{n}\log\left(\frac{2p(d+1)}{\delta}\right)}\right),$$
$$n > \max\left\{\frac{2}{N(m,d)}\log\left(\frac{2p(d+1)}{\delta}\right), \frac{8(d+1)}{C_{\min}}\log\left(\frac{m(1+dm)}{\delta}\right)\right\},$$

where $N(m,d) = \left\{{C_{\min}}/{(36m^2(d+1)^2)} - \nu\right\}^2$*, then we have that the following hold with probability at least* $1-\delta$*, for some* $\delta \in (0,1)$*:*

(i) $\mathcal{NE}(\widehat{\mathcal{G}}) = \varepsilon\text{-}\mathcal{NE}(\mathcal{G})$*, with* $\varepsilon = \frac{48(d_i+1)}{C_{\min}}\lambda$*.*

(ii) Additionally, if the true game $\mathcal{G}$ *satisfies,* $\forall i \in [p], \forall (x_i, \mathbf{x}_{-\mathsf{i}}), (x_i', \mathbf{x}_{-\mathsf{i}}) \in \mathcal{A}$*:*

$$(x_i, \mathbf{x}_{-\mathsf{i}}) \in \mathcal{NE}(\mathcal{G}) \text{ and } (x_i', \mathbf{x}_{-\mathsf{i}}) \notin \mathcal{NE}(\mathcal{G}) \implies u^i(x_i, \mathbf{x}_{-\mathsf{i}}) > u^i(x_i', \mathbf{x}_{-\mathsf{i}}) + \varepsilon.$$

Then, $\mathcal{NE}(\widehat{\mathcal{G}}) = \mathcal{NE}(\mathcal{G})$*.*

Remark 3.3.8 *The sufficient number of samples needed by our method to guarantee PSNE recovery, with probability at least* $1-\delta$*, scales as* $\mathcal{O}\left(m^4 d^4 \log({pd}/{\delta})\right)$*. This should be compared with the results of [JRVS11] for learning undirected graphical models. They show that* $\mathcal{O}\left(m^2 d^2 \log(m^2 p)\right)$ *are sufficient for learning* m*-ary discrete graphical models. However, their sample complexity hides a constant* K *that is related to the maximum eigenvalue of the scatter matrix, which we have upper bounded by* m^2d^2 *in our case, leading to a slightly higher sample complexity.*

Remark 3.3.9 *Note that as* $n \to \infty$*, the regularization parameter* $\lambda \to 2\nu$*, where* ν *is the maximum norm of the expected gradient at the true parameter* $\boldsymbol{\theta}^i$ *across all* $i \in [p]$*. Thus, even with an infinite number of samples, our method recovers the* ε*-Nash equlibria set of the true game with* $\varepsilon \to \frac{96(d_i+1)\nu}{C_{\min}}$ *as* $n \to \infty$*.*

3.4 Information-theoretic Lower Bounds

In this section, we obtain an information-theoretic lower bound on the number of samples needed to learn sparse polymatrix games. Let $\mathfrak{G}_{p,d,m}$ be set of polymatrix games over p players, with degree at most d, and maximum number of strategies per player being m. Our approach for doing so is to treat the inference procedure as a communication channel, where nature picks a game $\mathcal{G}^*$ from the set $\mathfrak{G}_{p,d,m}$ and then generates a data set $\mathcal{D}$ of n strategy profiles. A decoder $\psi : \mathcal{A}^n \to \mathfrak{G}_{p,d,m}$ then maps $\mathcal{D}$ to a game $\widehat{\mathcal{G}} \in \mathfrak{G}_{p,d,m}$. We wish to obtain lower bounds on the number of samples required by any decoder ψ to recover the true game consistently. In this setting, we define the *minimax* estimation error as follows:

$$p_{\text{err}} = \min_{\psi} \sup_{\mathcal{G}^* \in \mathfrak{G}_{p,d,m}} \Pr\left\{\mathcal{NE}(\psi(\mathcal{D})) \neq \mathcal{NE}(\mathcal{G}^*)\right\},$$

where the probability is computed over the data distribution. For obtaining necessary conditions on the sample complexity, we assume that the data distribution follows the global noise model described in (3.4). The following theorem prescribes the number of samples needed for learning sparse polymatrix games. Our proof of the theorem constitutes constructing restricted ensembles of "hard-to-learn" polymatrix games, from which nature picks a game uniformly at random and generates data. We then use the Fano's technique to lower bound the minimax error. The use of restricted ensembles is customary for obtaining information-theoretic lower bounds, c.f. [SW12, WWR10].

Theorem 3.4.1 *If the number of samples* $n \leq \frac{\log(m^d - m)\binom{p}{d}}{2\log 2} - 1$*, then estimation fails with* $p_{\text{err}} \geq 1/2$.

Remark 3.4.2 *From the above theorem we have that, the number of samples needed by any conceivable method, to recover the PSNE set consistently, is* $\Omega\left(d \log(pm)\right)$*, assuming that* $d = o(p)$*. Therefore, the method based on* $\ell_{1,2}$*-regularized logistic regression is information-theoretically optimal in the number of players, for learning sparse polymatrix games.*

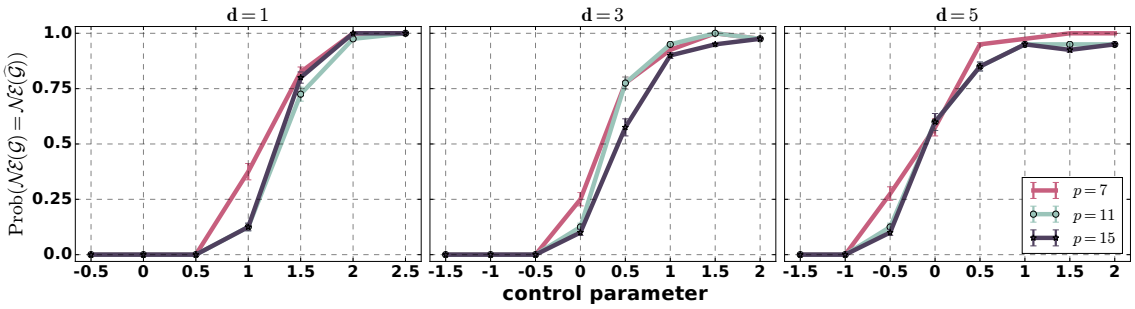


Figure 3.1.: Estimated probability of exact recovery of the PSNE set computed across 40 randomly sampled polymatrix games with the number of samples set to $n = 10^c(d+1)^2 \log(2p(d+1)/\delta)$, where c is the control parameter shown in the x-axis, and $\delta = 0.01$.

3.5 Experiments

3.5.1 Synthetic Experiments

We performed a series of experiments on synthetic data to validate our theoretical results. For these experiments, we first generated a random polymatrix game $\mathcal{G}$ by first generating random graphs over p players with degree exactly d, and number of pure strategies $m = 3$ per player. For each edge (i, j) in the graph, we set the payoffs as follows:

$$\begin{aligned} u^{i,i}(a) &= 0 && (\forall a \in [3]) \\ u^{i,j}(a,b) &\sim \mathcal{N}(0,2) && (\forall a \in [2] \wedge b \in [3]) \\ u^{i,j}(3,b) &= 0 && (\forall b \in [3]) \end{aligned}$$

We then generated a data set $\mathcal{D}$ from the game using the local noise model (3.5), with the noise parameter $q_i = 0.6$ for all $i \in [p]$. We then used our method to learn a game $\widehat{\mathcal{G}}$ from the data set $\mathcal{D}$, and computed $\mathbf{1}\left[\mathcal{NE}(\widehat{\mathcal{G}}) = \mathcal{NE}(\mathcal{G})\right]$. We then estimated the probability of successful PSNE recovery, $\Pr\left\{\mathcal{NE}(\widehat{\mathcal{G}}) = \mathcal{NE}(\mathcal{G})\right\}$, across 40 randomly sampled polymatrix games. Figure 3.1 plots the probability of successful

PSNE recovery as the number of samples is varied as $n = 10^c(d+1)^2 \log(2p(d+1)/\delta)$ and for various values of $d \in \{1, 3, 5\}$, with c being the control parameter and $\delta = 0.01$. We observe that the scaling of the sample complexity prescribed by Theorem 3.3.7 indeed holds in practice. The results show a phase transition behavior, where if the number of samples is less than $c(d+1)^2 \log(p(d+1)/\delta)$, for some constant c, then PSNE recovery fails with high probability, while if the number of samples is at least $C(d+1)^2 \log(p(d+1)/\delta)$, for some constant C, then PSNE recovery succeeds with high probability.

3.5.2 Real-world Experiments

We also evaluated our method on three publicly available real-world data sets containing (a) U.S. supreme court justices rulings, (b) voting records of senators from the 114th U.S. congress, and (c) roll-call votes in the U.N. General Assembly. We present evaluations of our method for each of the data set below.

Supreme court voting records. We analyzed two data sets of supreme court rulings: the first data set contains rulings of 9 justices across 512 cases spanning years 2010 to 2014, while the second data set contains rulings of 8 justices across 75 cases from year 2015 onwards [3]. We pre-processed the data, according the available code book, to map the vote of each justice, which was originally an integer between 1 to 8, to an integer between 1 to 3. Votes $\{1, 3, 4, 5\}$ were mapped to 1 and was interpreted as "voting with majority", votes $\{6, 7, 8\}$ were mapped to 2 and was interpreted as "not participating in the decision" , while vote 3 was mapped to 2 and was interpreted as "dissent". Thus, after pre-processing, each justice's vote was an integer between 1 to 3, with 1 corresponding to majority, 2 corresponding to abstention, and 3 corresponding to dissent.

After pre-processing the data, we learned a polymatrix game over supreme court justices using our algorithm. The regularization parameter λ was set according to

[3] All the data sets are publicly available at `http://scdb.wustl.edu`.

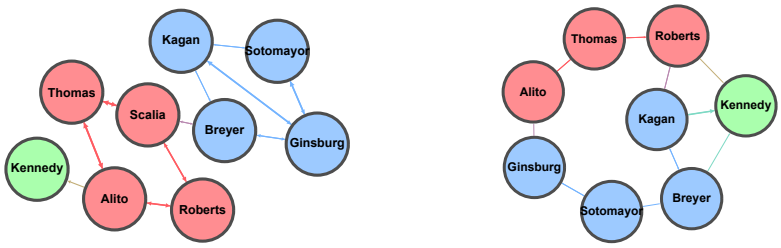


Figure 3.2.: The graphical game recovered from supreme court rulings data set 1 (years 2010-2014) on the left, and data set 2 (year 2015 onwards) on the right. Justice Thomas, Scalia, Roberts and Alito are widely known to be conservative and are denoted by the color ●, while Justice Breyer, Kagan, Sotomayor and Ginsburg, who are known to have a more liberal jurisprudence, are denoted by color ●. Justice Kennedy, who has a reputation of being moderate, is denoted by the color ●. The game graph was generated by adding all edges (i, j) if the corresponding payoff matrix $u^{i,j}$ was not all zeros. The average "influence" from j to i was calculated as the mean absolute payoff, i.e., $\frac{1}{6}\sum_{a=2}^{3}\sum_{b=1}^{3}|u^{i,j}(a,b)|$. The thickness of the edge denotes this influence of player j on i. Only the top 50% of the edges, in terms of influence, are shown.

Theorem 3.3.7 with reasonable values for different unknown population parameters. A more principled way to chose the regularization parameter λ is to assume a specific observation model, for instance, the global or local noise model, and then using crossvalidation to maximize the log-likelihood. The game graphs are shown in Figure 3.2 and the PSNE sets are shown in Table 3.1 for the two supreme court rulings data sets (years 2010-2014 and year 2015 onwards).

From 3.2 it is clear that our method recovers the well-established ideologies of the supreme court justices. This is especially evident for the graph learned from the first data set — there are two strongly connected components corresponding to the conservative and liberal bloc within the supreme court. The PSNE set recovered by our algorithm is also quite revealing. In both the data sets, a unanimous vote of 1 is a Nash equilibrium. Justice Kennedy, who has a moderate jurisprudence, always votes

Table 3.1.: The PSNE set learned from supreme court rulings data sets 1704 (top) and 1705 (bottom) respectively. Conservatives, liberal, and neutral justices are represented using ▭, ▭, and ▭ respectively. The actions 1, 2, and 3 correspond to "voted with majority", "abstention", and "dissent" respectively. The price of anarchy for the two data sets was computed to be 1.9 and 1.6 respectively.

Thomas	Scalia	Alito	Roberts	Kennedy	Breyer	Kagan	Ginsburg	Sotomayor
1	1	1	1	1	1	1	1	1
1	1	1	1	1	3	3	3	3
2	2	2	2	1	2	2	2	2
3	3	3	3	1	1	1	1	1
3	3	3	3	1	3	3	3	3

Thomas	Alito	Roberts	Kennedy	Breyer	Kagan	Ginsburg	Sotomayor
1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2
2	2	2	2	2	2	3	3
3	3	3	2	2	2	3	3

with the majority in the PSNE set. Further, strategy profiles where the conservative blocs and liberal blocs vote unanimously but dissent against each other are also in the PSNE set. In the second data set, there is a strongly connected component between the justice Kagan, Kennedy, and Breyer — this also bears out in the corresponding PSNE set where the strategies of the three justices are identical.

To compute the price of anarchy (PoA), we shifted all the payoff matrices by a constant to make the payoffs non-negative. Note that this does not change the PSNE set of the game. The price of anarchy was computed to be the ratio between the maximum welfare across all strategy profiles and the minimum welfare across all strategy profiles in the PSNE set. The PoA for the two data sets were, respectively, 1.9104 and 1.6115.

Senate voting records. We analyzed U.S. congressional voting records for the second session of the 114th congress (January 4, 2016 to January 3, 2017) [4]. The

[4]The data set is publicly available at `http://www.senate.gov/legislative/votes.htm`

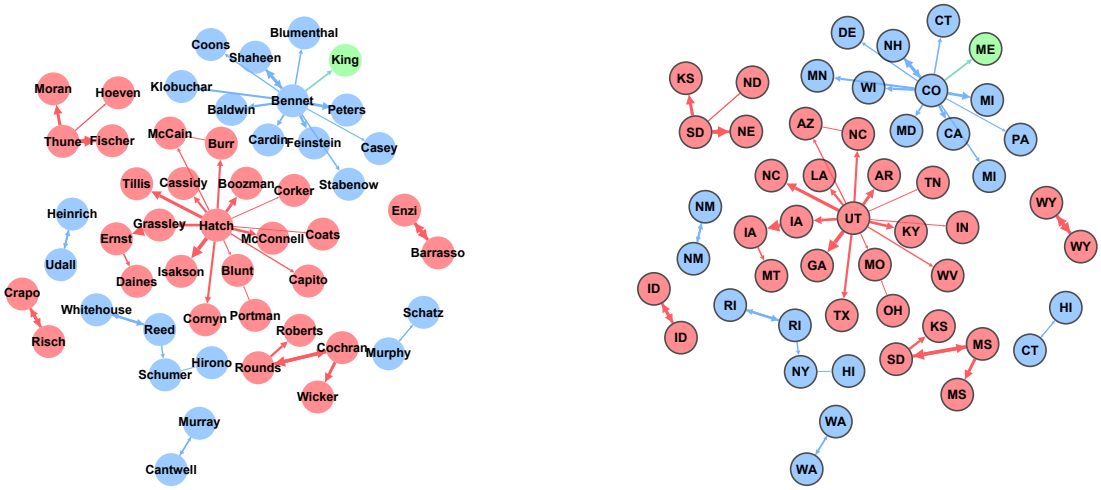


Figure 3.3.: The game graph learned from 114th U.S. congressional voting records. Only nodes with degree greater than one are shown. Democrats, Republicans, and Independent are denoted by ●, ●, and ● respectively. The graph on the right shows the states that the senators belong to. The thickness of the edges denote the amount of influence, computed as the mean absolute payoff, between the senators. Only nodes with degree at least 1 are shown.

data set comprised of the votes of 100 senators on 63 bills. The votes were pre-processed to take one of the three values: 1 ("yes"), 2 ("abstention"), and 3 ("no"). After pre-processing the data set we ran our algorithm to recover a polymatrix game from congressional voting records. Figure 3.3 shows the recovered game graph. Once again our method recovers the connected components corresponding the republicans and democrats. Interestingly, the connected components also have a nice geographic interpretation, for instance, the graph groups senators from Idaho, New Mexico, New York and midwestern states in their respective connected components. Strategy profiles where the overwhelming majority of senators in a connected component vote "yes" are in equilibria.

United Nations voting data. In our final real-world experiment we analyzed roll-call votes in the U.N. General Assembly. The data set contained votes of 193 countries

Table 3.2.: The PSNE set for the major connected components in the game graph learned from congressional voting records. The actions 1, 2, and 3 correspond to "yea", "abstain", and "nay" respectively. The combined number of Nash equilibria computed across senators with degree at least 1 was 144 and the price of anarchy was computed to be 2.6297.

Baldwin	Bennet	Blumenthal	Cardin	Casey	Coons	Feinstein	King	Klobuchar	Peters	Shaheen	Stabenow
1	1	1	1	1	1	1	1	1	1	1	1

Cochran	Roberts	Rounds	Wicker
1	1	1	1

Fischer	Hoeven	Moran	Thune
1	1	1	1
3	3	3	3

Hirono	Reed	Schumer	Whitehouse
1	1	1	1
3	3	3	3

Blunt	Boozman	Burr	Capito	Cassidy	Coats	Corker	Cornyn	Daines	Ernst	Grassley	Hatch	Isakson	McCain	McConnell	Portman	Tillis
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	3	1
1	1	1	1	1	1	1	1	1	1	1	2	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	2	1	1	1	2	1
1	3	3	3	1	3	3	3	1	1	3	3	1	3	3	1	3
3	1	1	1	1	1	1	1	1	1	1	2	1	1	1	3	1
3	3	3	3	1	3	3	3	1	1	3	3	1	3	3	2	3
3	3	3	3	1	3	3	3	1	1	3	3	1	3	3	3	3

for 847 U.N. resolutions [5]. Each vote could take one of the three values in $\{1, 2, 3\}$, with 1 denoting "yes", 2 denoting "abstention", and 3 denoting "no". The game graph learned from the data set is shown in Figure 3.4 while the PSNE set is shown in Table 3.3. As evident from Figure 3.4 our method recovered two major connected components: the first consisting of members of the Arab League, and the second consisting of majorly Southeast Asian countries and a few other Caribbean islands. The PSNE set once again comprised of strategy profiles where the overwhelming members of a connected component voted "yes". Within the component corresponding to the Arab league, Saudi Arabia, U.A.E., and Bahrain made up a small coalition of countries that voted identically in the PSNE set.

[5] The data set can be downloaded from `https://dataverse.harvard.edu/dataset.xhtml?persistentId=hdl:1902.1/12379`.

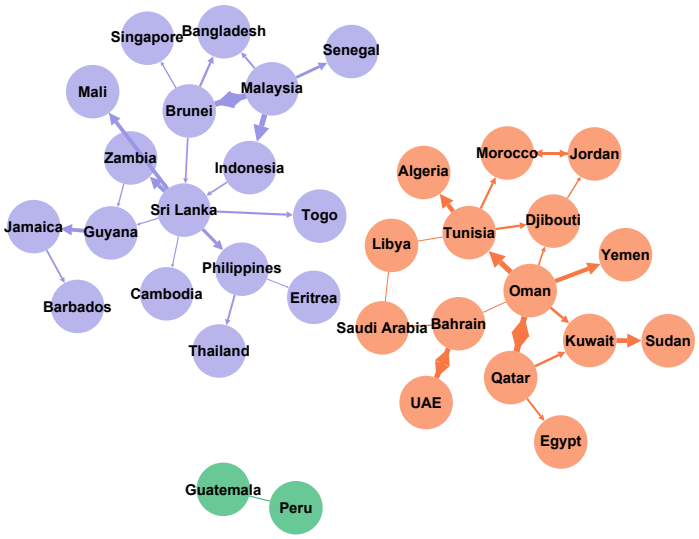


Figure 3.4.: The game graph learned from United Nations voting data set. Nodes belonging to the same connected component have the same color. Only countries with degree at least 1 are shown.

3.6 Summary

In this chapter we considered the problem of recovering the Nash equlibria set of a polymatrix game from observations of joint actions from repeated plays of the game. We proposed an $\ell_{1,2}$ group-regularized logistic regression method to learn

Table 3.3.: The PSNE set for the two major connected components in the game graph learned from United Nations voting data set. The total number of PSNE was 24 and the price of anarchy was computed to be 3.07.

Algeria	Bahrain	Djibouti	Egypt	Jordan	Kuwait	Libya	Morocco	Oman	Qatar	Saudi Arabia	Sudan	Tunisia	UAE	Yemen
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	2	1	1	1	1	1	1	1	1	2	1	1	2	1

Barbados	Bangladesh	Brunei	Cambodia	Eritrea	Guyana	Indonesia	Jamaica	Malaysia	Mali	Philippines	Senegal	Singapore	Sri Lanka	Thailand	Togo	Zambia
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	2	2	1	1	1	2	1	2	1	1	1	2	1	1	1	1
1	2	2	2	2	1	2	1	2	2	2	1	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2	1	2	2	2	2	2

polymatrix games with discrete pure strategies. We characterized the number of observations required by our method to recover an ε-Nash equilibria set of the game. We also obtained fundamental lower bounds on the number of samples required by any method to learn polymatrix games and showed that our method is close to optimal in the number of observations required. We evaluated our method on synthetic and real world data containing voting records of the U.S. supreme court, the United Nations, and U.S. Congress. By learning polymatrix games from these data sets we were able to quantify how supreme court justices, countries, and senators influence each other and make inferences about the steady-state (equilibria) behavior of these entities in a game-theoretic framework.

4 STRUCTURED PREDICTION

Structured prediction can be thought of as a generalization of binary classification to structured outputs, where the goal is to jointly predict several dependent variables. Predicting complex, structured data is of great significance in various application domains including computer vision (e.g., image segmentation, multiple object tracking), natural language processing (e.g., part-of-speech tagging, named entity recognition) and computational biology (e.g. protein structure prediction). However, unlike binary classification, structured prediction presents a set of unique computational and statistical challenges. The chief being that the number of structured outputs is exponential in the input size. For instance, in translation tasks, the number of parse trees of a sentence is exponential in the length of the sentence. Second, it is very common in such domains to have very few training examples as compared to the size of the output space thereby making generalization to unseen inputs difficult.

The key computational challenge in structured prediction stems from the *inference* problem, where a *decoder*, parameterized by a vector w of weights, predicts (or *decodes*) the latent structured output y given an observed input x. With the exception of a few special cases, the general inference problem in structured prediction is intractable. For instance in many cases the inference problem reduces to the maximum acyclic subgraph problem which is NP-hard and hard to approximate to within a factor of $1/2$ of the optimal solution [GMR08], or cardinality-constrained submodular maximization, which is also NP-hard and hard to compute a solution better than the $(1 - 1/\varepsilon)$-approximate solution returned by a greedy algorithm [NWF78]. The *learning* problem, where the goal is to learn the parameter w of the decoder from a set of labeled training instances, and which involves solving the inference problem as a subroutine, is therefore intractable for all but a few special cases. Hardness of max-margin learning (SVM) was shown by [SMGJ10].

Hardness results notwithstanding, various methods — which are exponential-time in the worst case— have been developed over the last decade for predicting structured data including conditional random fields [LMP01], and max-margin approaches [TGK03], to name a few. In these approaches, learning the parameter w of the decoder involves minimizing a loss function $L(w, \mathsf{S})$ over a data set S of m training pairs $\{(x_i, y_i)\}_{i=1}^m$. One could also take a Bayesian approach and learn a posterior distribution $\mathcal{Q}$ over decoder parameters w by minimizing the Gibbs loss $\mathbb{E}_{w\sim\mathcal{Q}}[L(w, \mathsf{S})]$. McAllester [McA07] showed, using the PAC-Bayesian framework, that the commonly used max-margin loss [TGK03] upper bounds the expected Gibbs loss over the data distribution, upto statistical error. Therefore, minimizing the max-margin loss provides a principled way for learning the parameters of a structured decoder. More recently, [HJ16] showed that minimizing a *surrogate* randomized loss, where the max-margin loss is computed over a small number of randomly sampled structured outputs, also bounds the Gibbs loss from above upto statistical error.

The above can be thought of as weight based perturbation models. The perturb-and-MAP framework introduced by [PY11], and henceforth referred to as MAP perturbation, provides an efficient way to generate samples from the Gibbs distribution by injecting random noise (that do not depend on the weights of the decoder w) in the potential or score function of the decoder and then computing the most likely assignment or energy configuration (MAP). MAP perturbation models are an attractive alternative to expensive Markov Chain Monte Carlo simulations for drawing samples from the Gibbs distribution, in that the former facilitates one-shot sampling. Moreover, learning MAP predictors for structured prediction problems is particularly attractive because the predictions are robust to random noise. However, learning the parameters of such MAP predictors involves solving the MAP problem, which in general is intractable. In this paper we obtain a provably polynomial time algorithm for learning the parameters of perturbed MAP predictors with structure based perturbations. In the following paragraph we summarize the main technical contributions of our paper.

Our contributions. To the best of our knowledge, we are the first to obtain generalization bounds for MAP-perturbation models with structure-based (Gumbel) perturbations — for detailed comparison with existing literature see Section 4.5. While it is well known that Gumbel perturbations induce a conditional random field (CRF) distribution over the structured outputs, we show that the generalization error is upper bounded by a CRF loss up to statistical error. We obtain Rademacher based uniform convergence guarantees for the latter. However, the main contribution of our paper is to obtain a provably polynomial time algorithm for learning MAP-perturbation models for general structured prediction problems. We propose a novel randomized *surrogate loss* that lower bounds the CRF loss and still upper bounds the expected loss over data distribution, upto approximation and statistical error terms that decay as $\widetilde{\mathcal{O}}\left(1/\sqrt{m}\right)$ with m being the number of samples. While it is NP-Hard to compute and approximate the CRF loss in general [Bar82, CSH08], our surrogate loss can be computed in polynomial time. Our results also imply that one can learn parameters of CRF models for structured prediction in polynomial time under certain conditions. Our work is inspired by the work of [HJ16] who also propose a polynomial time algorithm for learning the parameters of a structured decoder in the max-margin framework. In contrast to prior work which consider weight based perturbations, our work is concerned with structure based perturbations. Previous algorithms for learning MAP perturbation models, for instance, the hard-EM algorithm by [GHJ14] and the moment-matching algorithm by [PY11], are in general intractable and have no generalization guarantees. *Lastly, the main conceptual contribution of our work is to demonstrate that it is possible to efficiently learn the parameters of a structured decoder with generalization guarantees without solving the inference problem exactly.*

4.1 Preliminaries

We begin this section by introducing our notations and formalizing the problem of learning MAP-perturbation models. In structured prediction, we have an input

$x \in \mathfrak{X}$ and a set of feasible decodings of the input $\mathfrak{Y}(x)$. Without loss of generality, we assume that $|\mathfrak{Y}(x)| \leq r$ for all $x \in \mathfrak{X}$. Input-output pairs (x, y) are represented by a joint feature vector $\phi(x, y) \in \mathbb{R}^d$. For instance, when x is a sentence and y is a parse tree, the joint feature map $\phi(x, y)$ can be a vector of 0/1-indicator variables representing if a particular word is present in x and a particular edge is present in y. We will assume that $\min\{\phi_j(x, y) \neq 0 \mid j \in [d]\} \geq 1$ which commonly holds for structured prediction problems, for instance, when using binary features, or features that "count" number of components, edges, parts, etc.

A decoder $f_w : \mathfrak{X} \to \mathfrak{Y}$, parameterized by a vector $w \in \mathbb{R}^d$, returns an output $y \in \mathfrak{Y}(x)$ given an input x. We consider linear decoders of the form:

$$f_w(x) = \operatorname*{argmax}_{y \in \mathfrak{Y}(x)} \langle \phi(x, y), w \rangle, \tag{4.1}$$

which return the highest scoring structured output for a particular input x, where the score is linear in the weights w. As is traditionally the case in high-dimensional statistics, we will assume that the weight vectors are s-sparse, i.e., have at most s non-zero coordinates. We will denote the set of s-sparse d-dimensional vectors by $\mathbb{R}^{d,s}$.

In the perturb and MAP framework, a *stochastic decoder* first perturbs the linear score by injecting some independent noise for each structured output y, and then returns the structured output that maximizes the perturbed score. Gumbel perturbations are commonly used owing to the max-stability property of the Gumbel distribution. Denoting $\mathcal{G}(\beta)$ as the Gumbel distribution with location and scale parameters 0 and β respectively, we have the following stochastic decoder, where $\gamma \sim \mathcal{G}^r$ denotes a collection of r i.i.d. Gumbel-distributed random variables and γ_y denotes the Gumbel random variable associated with structured output y:

$$f_{w,\gamma}(x) = \operatorname*{argmax}_{y \in \mathfrak{Y}(x)} \langle \phi(x, y), w \rangle + \gamma_y. \tag{4.2}$$

For any weight vector w, and data set $\mathsf{S} = \{(x_i, y_i)\} \overset{\text{i.i.d.}}{\sim} \mathcal{D}^m$, we consider the following expected and empirical zero-one loss:

$$L(w, \mathcal{D}) = \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\mathbb{E}_{\gamma\sim\mathcal{G}^r} \left[\mathbf{1} \left[y \neq f_{w,\gamma}(x) \right] \right] \right], \tag{4.3}$$

$$L(w, \mathsf{S}) = \frac{1}{m} \sum_{i=1}^{m} \mathbb{E}_{\gamma\sim\mathcal{G}^r} \left[\mathbf{1} \left[y_i \neq f_{w,\gamma}(x_i) \right] \right], \tag{4.4}$$

where $\mathbf{1}[\cdot]$ denotes the indicator function and $\mathcal{D}$ is the unknown data distribution. We will let the scale parameter depend on the number of samples m and the weight vector w, and write $\beta(m, w) > 0$. The reason for this will become clear later, but intuitively one would expect that as the number of samples increases, the magnitude of perturbations should decrease in order to control the generalization error. Under Gumbel perturbations, $f_{w,\gamma}(x_i)$ is distributed according to following conditional random field (CRF) distribution $\mathcal{Q}(x_i, w)$ with pmf $q(\cdot; x_i, w)$ [Gum54, PY11]:

$$\begin{aligned} q(y_i; x_i, w) &= \Pr_{\gamma\sim\mathcal{G}^r(\beta)} \left\{ f_{w,\gamma}(x_i) = y_i \right\} \\ &= \frac{\exp\left(\langle \phi(x_i, y_i), w \rangle / \beta \right)}{Z(w, x_i)}, \end{aligned} \tag{4.5}$$

where $Z(w, x_i) = \sum_{y\in\mathcal{Y}(x)} \exp\left(\langle \phi(x_i, y), w \rangle / \beta \right)$ is the partition function. The empirical loss in (4.4) can then be computed as:

$$(\textbf{CRF loss})\ L(w, \mathsf{S}) = \frac{1}{m} \sum_{i=1}^{m} \Pr \left\{ f_{w,\gamma}(x_i) \neq y_i \right\}. \tag{4.6}$$

The ultimate objective of a learning algorithm is to learn a weight vector w that generalizes to unseen data. Therefore, minimizing the expected loss given by (4.3) is the best strategy towards that end. However, since the data distribution is unknown, one instead minimizes the empirical loss (4.4) on a finite number of labeled examples S.

4.2 Generalization Bound

As a first step we will show that the empirical loss (4.6) indeed bounds the expected perturbed loss (4.3) from above, upto statistical error that decays as $\widetilde{\mathcal{O}}\left(1/\sqrt{m}\right)$. We

have the following generalization bound whose proof, as well as proofs of other results in the rest of the chapter, can be found in Appendix C.

Theorem 4.2.1 (Rademacher based generalization bound)
With probability at least $1-\delta$ *over the choice of* m *samples* S*:*

$$(\forall w \in \mathbb{R}^{d,s})\ L(w, \mathcal{D}) \leq L(w, \mathsf{S}) + \varepsilon(d, s, m, r, \delta),$$

where

$$\varepsilon(d, s, m, r, \delta) = 2\sqrt{\frac{s(\ln d + 2\ln(mr))}{m}} + 3\sqrt{\frac{\ln 2/\delta}{2m}}.$$

As a direct consequence of the uniform convergence bound given by Theorem 4.2.1, we have that minimizing the CRF loss (4.6) is a consistent procedure for learning MAP-perturbation models.

4.3 Towards an Efficient Learning Algorithm

While Theorem 4.2.1 provides theoretical justification for learning perturbation models by minimizing the CRF loss (4.6), with the exception of a few special cases, computing the loss function is in general intractable. This is due to the need for computing the partition function $Z(w, x)$ which is an NP-hard problem [Bar82]. Further, even approximating $Z(w, x)$ with high probability and arbitrary precision is also known to be NP-hard [CSH08].

To counter this computational bottleneck, we propose an efficient stochastic decoder that decodes over a randomly sampled set of structured outputs. To elaborate further, given some $x \in \mathfrak{X}$, let $\mathcal{R}(x, w)$ be some *proposal distribution*, parameterized by x and w, over the structured outputs $\mathfrak{Y}(x)$. We generate a set T' of n structured outputs sampled independently from the distribution $\mathcal{R}$ and define the following *efficient* stochastic decoder:

$$f_{w,\gamma,\mathsf{T}'}(x) = \operatorname*{argmax}_{y \in \mathsf{T}'} \langle \phi(x, y), w \rangle + \gamma_y. \tag{4.7}$$

Therefore $f_{w,\gamma,\mathsf{T}'}(x)$ is distributed according to the CRF distribution $\mathcal{Q}(x, w, \mathsf{T}')$ with pmf $q(\cdot; x, w, \mathsf{T}')$ and support on T' as follows:

$$\begin{aligned} q(y; x, w, \mathsf{T}') &= \Pr_{\gamma \sim \mathcal{G}^n} \{f_{w,\gamma,\mathsf{T}'}(x) = y\} \\ &= \frac{\mathbf{1}[y \in \mathsf{T}']}{Z_{w,x,\mathsf{T}'}} \exp(\langle \phi(x,y), w \rangle / \beta), \end{aligned}$$

where $Z_{w,x,\mathsf{T}'} = \sum_{y' \in \mathsf{T}'} \exp(\langle \phi(x,y'), w \rangle / \beta)$. Note that the partition function $Z_{w,x,\mathsf{T}'}$ can be computed in time linear in n, since $|\mathsf{T}'| = n$. Now, let $\mathsf{T} = \{\mathsf{T}_i \mid x_i \in \mathsf{S}\}$ be the collection of n structured outputs sampled for each x_i in the data set, from the product distribution $\mathcal{R}(\mathsf{S}, w) \stackrel{\text{def}}{=} \times_{i=1}^{m} (\mathcal{R}(x_i)^n)$. *Note that the distribution $\mathcal{R}(\mathsf{S}, w)$ does not depend on the $\{y_i\}$'s in S. We denote the distribution over the collection of sets $\{\mathsf{T}_i\}$ by $\mathcal{R}(\mathsf{S}, w)$ to keep the notation light.* Additionally, we consider proposal distributions $\mathcal{R}(x, w)$ that are equivalent upto linearly inducible orderings of the structured output.

Definition 4.3.1 (Equivalence of proposal distributions [HJ16]) *For any $x \in \mathfrak{X}$, two proposal distributions $\mathcal{R}(x, w)$ and $\mathcal{R}(x, w')$, with probability mass functions $p(\cdot; x, w)$ and $p(\cdot; x, w')$, are equivalent if:*

$$\begin{gathered} \forall y, y' \in \mathfrak{Y}(x) \ : \langle \phi(x, y), w \rangle \leq \langle \phi(x, y'), w \rangle \\ \text{and } \langle \phi(x, y), w' \rangle \leq \langle \phi(x, y'), w' \rangle \\ \iff \forall y \in \mathfrak{Y}(x)\ p(y; x, w) = p(y; x, w'). \end{gathered}$$

We then write $\mathcal{R}(x, w) \equiv \mathcal{R}(x, w') \equiv \mathcal{R}(x, \pi(x))$, where $\pi(x)$ is the linear ordering over $\mathfrak{Y}(x)$ induced by w (and w').

Intuitively speaking, the above definition requires proposal distributions to depend only on the orderings of the values $\langle \phi(x, y_1), w \rangle, \ldots, \langle \phi(x, y_r), w \rangle$ and not on the actual value of $\langle \phi(x, y_j), w \rangle$.

To obtain an efficient learning algorithm with generalization guarantees, we will use *augmented* sets $\bar{\mathsf{T}} = \{\bar{\mathsf{T}}_i\}_{i=1}^{m}$, where $\bar{\mathsf{T}}_i = \mathsf{T}_i \cup \{y_i\}$. Then, given a random col-

lection of structured outputs T, we consider the following *augmented randomized* empirical loss for learning the parameters of the MAP-perturbation model:

$$L(w, \mathsf{S}, \bar{\mathsf{T}}) = \frac{1}{m} \sum_{i=1}^{m} \Pr_{\gamma \sim \mathcal{G}^n} \left\{ f_{w,\gamma,\bar{\mathsf{T}}_i}(x_i) \neq y_i \right\}. \tag{4.8}$$

As opposed to the loss function given by (4.6), the loss in (4.8) can be computed efficiently for small n. Our next result shows that the randomized augmented loss lower bounds the full CRF loss $L(w, \mathsf{S})$ as long as $\bar{\mathsf{T}}_i$ is a *set*, i.e., contains only unique elements.

Lemma 4.3.1 *For all data sets* S, $\mathsf{T}_i \subseteq \mathfrak{Y}(x_i)$, *and weight vectors* w*:*

$$\begin{aligned} L(w, \mathsf{S}, \bar{\mathsf{T}}) - L(w, \mathsf{S}) = \\ - \frac{1}{m} \sum_{i=1}^{m} \Pr_{\gamma} \left\{ f_{w,\gamma,\bar{\mathsf{T}}_i}(x_i) = y_i \right\} \times \\ \Pr_{\gamma} \left\{ f_{w,\gamma}(x_i) \in (\mathfrak{Y}(x_i) \setminus \bar{\mathsf{T}}_i) \right\} \leq 0 \end{aligned} \tag{4.9}$$

Remark 4.3.2 *If* $\bar{\mathsf{T}}_i = \mathfrak{Y}(x_i)$ *then* $L(w, \mathsf{S}) = L(w, \mathsf{S}, \bar{\mathsf{T}}_i)$.

Next, we will show that an algorithm that learns the parameter w of the MAP-perturbation model, by sampling a small number of structured outputs for each x_i and minimizing the empirical loss given by (4.8), generalizes under various choices of the proposal distribution $\mathcal{R}$. Our first step in that direction would be to obtain uniform convergence guarantees for the stochastic loss (4.8).

4.3.1 Generalization Bound

To obtain our generalization bound, we decompose the difference the exact and randomized loss $L(w, \mathsf{S}) - L(w, \mathsf{S}, \bar{\mathsf{T}})$ as follows:

$$L(w, \mathsf{S}) - L(w, \mathsf{S}, \bar{\mathsf{T}}) = A(w, \mathsf{S}) + B(w, \mathsf{S}, \bar{\mathsf{T}}), \tag{4.10}$$

$$A(w, \mathsf{S}) = L(w, \mathsf{S}) - \mathbb{E}_{\mathsf{T} \sim \mathcal{R}(\mathsf{S})} \left[L(w, \mathsf{S}, \bar{\mathsf{T}}) \right], \tag{4.11}$$

$$B(w, \mathsf{S}, \bar{\mathsf{T}}) = \mathbb{E}_{\mathsf{T} \sim \mathcal{R}(\mathsf{S})} \left[L(w, \mathsf{S}, \bar{\mathsf{T}}) \right] - L(w, \mathsf{S}, \bar{\mathsf{T}}), \tag{4.12}$$

where $A(w, \mathsf{S})$ can be thought of as the approximation error due to using a small number of structured outputs T_i's instead of the full sets $\mathfrak{Y}(x_i)$, while $B(w, \mathsf{S}, \bar{\mathsf{T}})$ be is the statistical error. In what follows, we will bound each of these errors from above.

From Lemma 4.3.1 it is clear that the proposal distribution plays a crucial role in determining how far the surrogate loss $L(w, \mathsf{S}, \bar{\mathsf{T}})$ is from the CRF loss $L(w, \mathsf{S})$. To bound the approximation error, we make the following assumption about the proposal distributions $\mathcal{R}(x, w)$.

Assumption 4.3.3 *For all $(x_i, y_i) \in \mathsf{S}$ and weight vectors $w \in \mathbb{R}^{d,s}$, the proposal distribution satisfies the following condition with probability at least $1 - \|w\|_1/\sqrt{m}$, for a constant $c \in [0, 1]$:*

(i) $\mathsf{T}_i = \{y_i\}$ *if* $\forall\, y \neq y_i \langle \phi(x_i, y_i), w\rangle > \langle \phi(x_i, y), w\rangle$,

(ii) $\frac{1}{n}\sum_{y \in \mathsf{T}_i} \langle \phi(x_i, y), w\rangle \geq \langle \phi(x_i, y_i), w\rangle + c\,\|w\|_1$ *otherwise,*

where the probability is taken over the set T_i.

Intuitively, Assumption 4.3.3 states that, if y_i is not the highest scoring structure under w, then the proposal distribution should return structures $\mathsf{T} = \{y\}$ whose average score is an additive constant factor away from the score of the observed structure y_i with high probability. Otherwise, the proposal distribution should return the singleton set $\mathsf{T} = \{y_i\}$ with high probability. Note that Assumption 4.3.3 is in comparison much weaker than the low-norm assumption of [HJ16], which requires that, in expectation, the norm of the difference between $\phi(x, y)$ and $\phi(x, y_i)$ (where y is sampled from the proposal distribution) should decay as $1/\sqrt{m}$. The following lemma bounds the approximation error from above.

Lemma 4.3.4 (Approximation Error) *If the scale parameter of the Gumbel perturbations satisfies: $\beta \leq \min(\|w\|_1/\log m, w_{\min}/\log((r-1)(\sqrt{m}-1)))$ for all $w \neq 0$, and $n \geq m^{0.5-c}$, then under Assumption 4.3.3 $A(w, \mathsf{S}) \leq \varepsilon_1(m, n, w)$, where*

$$\varepsilon_1(m, n, w) \stackrel{\text{def}}{=} \frac{\|w\|_1}{\sqrt{m}} + \frac{1}{1+\sqrt{m}},$$

and $w_{\min} = \min\{|w_j| \mid |w_j| \neq 0, j \in [d]\}$.

Note that for $c \geq 0.5$ the number of structured outputs needed is $n = 1$, while in the worst case ($c = 0$) $n = \sqrt{m}$. Furthermore, n needs to grow polynomially with respect to m in order to achieve $\mathcal{O}\left(1/\sqrt{m}\right)$ generalization error.

Lemma 4.3.5 (Statistical Error) *For any fixed data set* S*, the statistical error* $B(w, \mathsf{S}, \bar{\mathsf{T}})$ *is bounded, simultaneously for all proposal distributions* $\mathcal{R}(x_i, w)$ *over* $\{\mathsf{T}_i\}$*, as follows:*

$$\begin{aligned} &\Pr_{\mathsf{T}}\left\{(\forall w \in \mathbb{R}^{d,s})\ B(w, \mathsf{S}, \bar{\mathsf{T}}) \leq \varepsilon_2(d, s, n, r, m, \delta) \mid \mathsf{S}\right\} \\ &\quad \geq 1 - \delta, \end{aligned} \tag{4.13}$$

where

$$\varepsilon_2(d, s, n, r, m, \delta) \stackrel{\text{def}}{=} 2\sqrt{\frac{s(\ln d + 2\ln(nr))}{m}} + \sqrt{\frac{\ln 1/\delta}{2m}} + \sqrt{\frac{s(\ln d + 2\ln(mr)) + \ln 1/\delta}{2m}}.$$

Now, we are ready to present our main result proving uniform convergence of the randomized loss $L(w, \mathsf{S}, \bar{\mathsf{T}})$. More specifically, we provide $\widetilde{\mathcal{O}}\left(1/\sqrt{m}\right)$ generalization error.

Theorem 4.3.6 *With probability at least* $1 - 2\delta$ *over the choice of the data set* S *and the set of random structured outputs* T*, and simultaneously for all* $w \in \mathbb{R}^{d,s}$ *and proposal distributions* $\mathcal{R}(x, w)$*:*

$$L(w, \mathcal{D}) \leq L(w, \mathsf{S}, \bar{\mathsf{T}}) + \varepsilon_1 + \varepsilon_2, \tag{4.14}$$

where ε_1 *and* ε_2 *are defined in Lemma 4.3.4 and 4.3.5 respectively.*

Proof The claim follows directly from Lemma 4.3.4 and Lemma 4.3.5 by taking an expectation with respect to S. ■

4.3.2 Examples of Proposal Distributions

Having proved uniform convergence of our randomized procedure for learning the parameters of a MAP decoder, we turn our attention to the proposal distribution.

We want to construct proposal distributions of the form given by Definition 4.3.1 that satisfy Assumption 4.3.3 with a large enough constant c. Additionally, for our randomized procedure to run in polynomial time we want the proposal distribution to sample a structured output in constant time. The following algorithm is directly motivated by Assumption 4.3.3 where the set $\text{neighbors}_k(y)$ for an input x is defined as: $\text{neighbors}_k(y) \stackrel{\text{def}}{=} \{y' \in \mathfrak{Y}(x) \backslash \{y\} \mid H(y, y') \leq k\}$, with $H(\cdot, \cdot)$ being the Hamming distance.

Algorithm 3 An example algorithm implementing a proposal distribution that depends on $y_i \in \mathsf{S}$.

1: **Input:** Weight vector $w \in \mathbb{R}^{d,s}$, $(x_i, y_i) \in \mathsf{S}$, parameter $\alpha \in [0, 1]$ and $k \geq 1$.
2: **Output:** A structured output $y \in \mathfrak{Y}(x)$.
3: With probability α pick y' uniformly at random from $\mathfrak{Y}(x_i)$, and with probability $1 - \alpha$ set y' to y_i.
4: $y \leftarrow y'$.
5: **for** $y' \in \mathsf{neighbors}_k(y)$ **do**
6: **if** $\langle \phi(x, y'), w \rangle \geq \langle \phi(x, y), w \rangle$ **then**
7: $y \leftarrow y'$.
8: **end if**
9: **end for**
10: **Return** y.

Remark 4.3.7 *Setting $\alpha = \|w\|_1/\sqrt{m}$, Algorithm 3 satisfies the condition given in Definition 4.3.1 as well as Assumption 4.3.3. Since, for any $w, w' \in \mathbb{R}^{d,s}$ that induce the same linear ordering over $\mathfrak{Y}(x)$, conditioned on the y' sampled in Step 3, the algorithm returns the same y for both w and w' with probability* 1.

Also note that using a larger k ensures that the above algorithm satisfies Assumption 4.3.3 with a larger constant c, thereby reducing the number of structured outputs that need to be sampled (n), at the cost of increased computation for sampling a single structured output.

The parameter α in Algorithm 3 controls exploration vs exploitation. As α becomes smaller Algorithm 3 returns a proposal from within the neighborhood of y_i while for larger α it explores high scoring structures in the entire set of candidate structures.

Lastly, note that our results do not violate the hardness results of [SMGJ10], who essentially show that it is NP-hard to decide if the training data is linearly separable. Depending on whether or not the data is linearly separable, the loss $L(w, S)$ (4.6) can be large or small (for all or some weight vector). While computing $L(w, S)$ is intractable in general, we merely provide an efficiently computable lower bound $L(w, S, \mathsf{T})$ ((4.8)) that still upper bounds the expected loss $L(w, \mathcal{D})$.

4.3.3 Minimizing the CRF Loss

In this section we discuss strategies for minimizing the (randomized) CRF loss $L(w, \mathsf{S}, \bar{\mathsf{T}})$. Minimizing the randomized CRF loss $L(w, \mathsf{S}, \bar{\mathsf{T}})$ is equivalent to maximizing the *randomized CRF gain* $U(w, \mathsf{S}, \bar{\mathsf{T}}) = \frac{1}{m} \sum_{i=1}^{m} \Pr_\gamma \left\{ f_{w,\gamma,\bar{\mathsf{T}}_i}(x_i) = y_i \right\}$, which in turn is equivalent to maximizing $\log U(w, \mathsf{S}, \bar{\mathsf{T}})$. The latter can be accomplished by gradient based methods with the gradient of $\log U(w, \mathsf{S}, \bar{\mathsf{T}})$ given by:

$$\nabla_w \log U(w, \mathsf{S}, \bar{\mathsf{T}}) = \frac{\sum_{i=1}^{m} q_i (\phi(x_i, y_i) - \mathbb{E}\left[\phi(x_i, y)\right])}{\sum_{i=1}^{m} q_i}, \tag{4.15}$$

where $q_i \stackrel{\text{def}}{=} \Pr_\gamma \left\{ f_{w,\gamma,\bar{\mathsf{T}}_i}(x_i) = y_i \right\}$, and the expectation is taken with respect to $y \sim \mathcal{Q}(x_i, w, \bar{\mathsf{T}}_i)$. The exact CRF loss ($L(w, \mathsf{S})$) can similarly be minimized by using $\bar{\mathsf{T}}_i = \mathfrak{Y}(x_i)$, for all $x_i \in \mathsf{S}$, in the above. Note that by Jensen's inequality $\log U(w, \mathsf{S}, \bar{\mathsf{T}}) \geq \frac{1}{m} \sum_{i=1}^{m} \log \Pr_\gamma \left\{ f_{w,\gamma,\bar{\mathsf{T}}_i}(x_i) = y_i \right\}$, where the latter can be identified as the log likelihood of the data set S under the CRF distributions $\{\mathcal{Q}(x_i, w, \bar{\mathsf{T}}_i)\}$. Therefore, $L(w, \mathsf{S}, \bar{\mathsf{T}})$ can be equivalently minimized by minimizing the negative log-likelihood of the data, which in turn gives rise to the well known *moment-matching* rule known in the literature [PY11]. Thus, Algorithm 3 can be used with standard moment matching where the expectation is approximated by averaging over y's drawn from the distribution $\mathcal{Q}(x_i, w, \bar{\mathsf{T}}_i)$. While standard moment matching is in general

intractable, moment matching in conjunction with Algorithm 3 is always efficient. Indeed, (4.15) can be thought of as a "weighted" moment matching rule with weights q_i.

4.4 Experiments

4.4.1 Synthetic Experiments

In this section, we evaluate our proposed method (`CRF_RAND`) on synthetic data against three other methods: `CRF_ALL`, `SVM_RAND`, and `SVM`. The `CRF_RAND` method minimizes the randomized loss $L(w, \mathsf{S}, \bar{\mathsf{T}})$ (4.8) subject to ℓ_1 penalty (as prescribed by Lemma 4.3.4) by sampling structured outputs from the proposal distribution given by Algorithm 3. The `CRF_ALL` method minimizes the exact (exponential-time) loss $L(w, \mathsf{S})$ (4.6). Lastly, `SVM` is the widely used max-margin method of [TGK03], while `SVM_RAND` is the randomized SVM method proposed by [HJ16].

We generate a ground truth parameter $w^* \in \mathbb{R}^d$ with random entries sampled independently from a zero mean Gaussian distribution with variance 100. We then randomly set all but $s = \sqrt{d}$ entries to be zero. We then generate a training set of S of 100 samples. We used the following joint feature map $\phi(x, y)$ for an input output pair. For every pair of possible edges or elements i and j, we set $\phi(x, y)_{i,j} = \mathbf{1}[x_{i,j} = 1 \wedge i \in y \wedge j \in y]$. For instance, for directed spanning trees of v nodes, we have $x \in \{0, 1\}^{\binom{v}{2}}$ and $\phi(x, y) \in \mathbb{R}^{\binom{v}{2}}$. We considered directed spanning trees of 6 nodes, directed acyclic graphs of 5 nodes and 2 parents per node, and sets of 4 elements chosen from 15 possible elements. In order to generate each training sample $(x, y) \in S$, we generated a random vector x with independent Bernoulli entries with parameter $1/2$. After generating x, we set $y = f_{w^*}(x)$, i.e., we solved (4.1) in order to produce the latent structured output y from the observed input x and the parameter w^*.

We set the ℓ_1 regularization parameter to be 0.01 for all methods. We used 20 iterations of gradient descent with step size of $1/\sqrt{t}$ for all algorithms, where

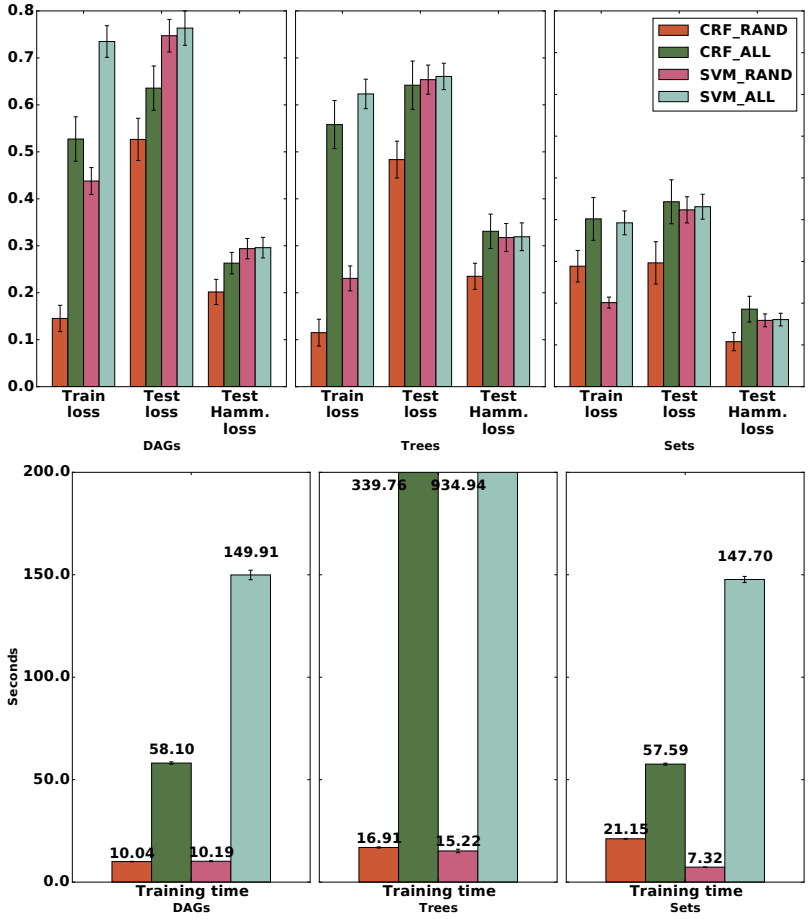


Figure 4.1.: (**Top**) Training and test set loss (4.6), and test set hamming loss of the exact method (`CRF_ALL`) and our randomized algorithm (`CRF_RAND`), the randomized SVM method by [HJ16] (`SVM_RAND`), and the exact SVM (`SVM_ALL`), a.k.a max-margin, method of [TGK03]. For the randomized algorithms, i.e., `CRF_RAND` and `SVM_RAND`, the training loss is the randomized training loss, i.e., $L(w, \mathsf{S}, \bar{\mathsf{T}})$ and $L(w, \mathsf{S}, \mathsf{T})$ respectively. (**Bottom**) Training time in seconds for the various methods.

t is the iteration, to learn the parameter w for both the exact method and our randomized algorithm. In order to simplify gradient calculations, we simply set $\beta = 1/\log((r-1))(\sqrt{m}-1))$ during training. For `CRF_RAND`, we used Algorithm 3 with $\alpha = {\|w\|_1}/{\sqrt{m}}$ and invoke the algorithm $\sqrt{m}$ number of times to generate the set T_i for each $i \in [m]$ and w. This results in $n = |\mathsf{T}_i| \leq \sqrt{m}$. To evaluate the generalization performance of our algorithm we generated a test set $S' = \{x'_i, y'_i\}_{i=1}^m$ of 100 samples and calculated two losses. The first was the full CRF loss (4.6) on the test set S', and the second was the test set hamming loss $\frac{1}{m}\sum_{i=1}^m \hat{H}(f_{\hat{w}}(x'_i), y'_i)$, where $\hat{H}(\cdot,\cdot)$ is the normalized Hamming distance, and $\hat{w}$ is the learned parameter. Hamming distance is a popular distortion function used in structured prediction, and provides a more realistic assessment of the performance of a decoder, since in most cases it suffices to recover most of the structure rather than predicting the structure exactly. For DAGs and trees the Hamming distance counts the number of different edges between the structured outputs, while for sets it counts the number of different elements. We normalize the Hamming distance to be between 0 and 1. We computed the mean and 95% confidence intervals of each of these metrics by repeating the above procedure 30 times.

Figure 4.1 shows the training and test set errors and the training time of the four different algorithms. `CRF_RAND` significantly outperformed other algorithms in both the test set loss and test set hamming loss, while being ≈ 6 times faster than the exact method (`CRF_ALL`) for DAGs, ≈ 20 times faster for trees, and ≈ 3 times faster for sets. The exact CRF method (`CRF_ALL`) was also significantly faster than the exact SVM (`SVM`) method while achieving similar test set loss and test set hamming loss.

4.4.2 Real-world Experiments

In this section, we evaluate the performance of our method on a image matching task. We used the Buffy Stickmen data set (available at `http://www.robots.ox.ac.`

uk/~vgg/data/stickmen/), containing stills (images) from the television (TV) show *Buffy the Vampire Slayer*. The data set contains 187 sequences from different episodes and scenes from the TV series. Each sequence has 4 video frames and we match the first and last frames in a sequence, resulting in 187 image pairs. Each image is annotated with 18 keypoints corresponding to 6 body parts (head, torso, etc.). Given two image pairs from a video sequence, the goal is to match the keypoints in the image. Specifically, the input $x = (x^0, x^1)$ corresponds to an RGB image pair, the corresponding output y is a permutation of $\{1, \ldots, 18\}$, where the i-th keypoint in x^0 is matched with the $y(i)$-th keypoint in x^1. Note that this is a weighted bipartite matching problem and can be solved in polynomial time, e.g., using the Hungarian algorithm. More specifically, given a weight vector w, $f_w(x)$ can be computed efficiently. We used the following feature for a input-output pair $\phi(x, y) = \frac{1}{18}\sum_{i=1}^{18}(\psi(x^0, i) - \psi(x^1, y(i)))^2$, where $\psi(\cdot, i) \in \mathbb{R}^{128}$ are the (normalized) SIFT descriptors with scale 5 computed at the i-th keypoint of the image. We compared our algorithm (`CRF_RAND`) against two other methods: the moment-matching method (`MM`), where we solve $f_w(x)$ exactly, and the `SVM_RAND` method of [HJ16]. For moment matching (`MM`), we learned using exact inference, and approximated the expectation (for computing the expected sufficient statistics) by averaging over 50 samples drawn using Gumbel perturbations. For `SVM_RAND`, we used $n = 5$, i.e., we draw 5 samples from the proposal distribution. For each of the methods we computed the mean test set error across 10 bootstrap runs, where in each run we sample 187 image pairs (with replacement) to generate the training set and use the complement of the training set as the test set. We tried four different ℓ_1 regularization parameters for each method: $0, 0.001$, and 0.01, and report the highest mean test set error for each method. The error is computed as the number of keypoints (out of a maximum of 18) that were matched incorrectly. Table 4.1 details the error of the three methods. Note that our method achieves similar performance to that of moment matching even though the latter uses exact inference during learning. Figure 4.2 shows the three best and three worst matches returned

Table 4.1.: Test set hamming error (number of mismatched key points) of the three methods on the image matching task.

Method	Test set hamming error	Train time (sec.)
`CRF_RAND`	6.94 ± 0.27	369.33 ± 2.36
`MM`	6.95 ± 0.3	487.58 ± 2.29
`SVM_RAND`	7.29 ± 0.5	665.42 ± 10.32

by our method for regularization parameter 0.01, and a train-test split where the odd numbered (resp. even numbered) sequences were used for training (resp. test).

4.5 Related Work

Significant body of work exists in computing a single MAP estimate by exploiting problem specific structure, for instance, super-modularity, linear programming relaxations to name a few. However, in this paper we are concerned with the problem of

Figure 4.2.: The three best (left) and three worst (right) matchings returned by our algorithm (`CRF_RAND`) on the test set of the image matching task.

learning the parameters of MAP perturbation models. Among generalization bounds for MAP perturbation models, [HMKJ13] prove PAC-Bayesian generalization bounds for weight based perturbations. [HMKJ13] additionally propose learning weight based MAP-perturbation models by minimizing the PAC-Bayesian upper bound on the generalization error. However, their method for learning the parameters involves constructing restricted families of posterior distributions over the weights w that lead to smooth, but not necessarily convex, generalization bounds that can be optimized using gradient based methods. For learning MAP-perturbation models with structure based (Gumbel) perturbations, [GHJ14] propose a hard-EM algorithm which is both worst-case exponential time and has no theoretical guarantees. [PY11] on the other hand, propose learning Gumbel MAP-perturbation models by using the moment matching method. However, such an approach is tractable only for energy functions for which the global minimum can be computed efficiently. Lastly, [HMJ13, OHSJ14] consider the problem of efficiently sampling from MAP perturbation models using low dimensional perturbations. [HJ12, HMJ13] additionally propose ways to approximate and bound the partition function. While such bounds on the partition function can be used, in principle, to approximately minimize the CRF loss (4.6), it is unclear if one can obtain uniform convergence guarantees for the same, given that computing or even approximating the partition function is NP-hard [Bar82, CSH08].

4.6 Summary

In this chapter, we proposed a provably polynomial time randomized procedure for learning the parameters of perturbed MAP predictors with generalization guarantees. Surprisingly, our results demonstrate that learning is possible in the MAP perturbation framework without solving the inference problem exactly, i.e., only upto a constant factor approximation. The main idea behind our approach was to use a proposal distribution to sample a small number of structured outputs in order to approximate the partition function. We obtained conditions on the proposal dis-

tributions under which the learned structured predictor can generalize to unseen examples. Our generalization bounds demonstrated that the proposed approach not only achieves computationally efficiency, but also has better generalization than exact methods. We evaluated the efficacy of our approach through synthetic and real-world experiments and compared our method against state-of-the-art exact and approximate learning algorithms where we demonstrated superior accuracy and computational efficiency. In the backdrop of results showing that it is hard to compute and accurately approximate the partition function [Bar82, CSH08], our results represent a significant advance in obtaining efficient learning algorithms by only roughly approximating the partition function.

5 CONCLUSION

This dissertation took an in-depth look at the general problem of recovering combinatorial structures from a limited amount of data. Such problems are pervasive in many application domains like computer vision, natural language processing, and genetics to name a few, and efficient algorithms with strong theoretical guarantees for the same are of paramount importance.

Specifically, in Chapter 2 we considered the problem of recovering DAGs corresponding to a linear structural equation model from purely observational data. Our work relaxed the homoscedastic noise condition to prove identifiability of linear SEMs under a slightly more general condition. Furthermore, our method was the first fully polynomial time algorithm for the problem and returns the correct DAG using an (almost) information-theoretically optimal number of samples. The key conceptual leap made by our work was to exploit the identifiability condition to propose a Cholesky factorization like algorithm for learning the edge weight matrix. Identifiability has been previously exploited to obtain computationally efficient algorithms in the LINGAM case. However, finite sample guarantees are yet to be obtained for the same.

In Chapter 3, we considered the problem of learning graphical games from behavioral data. Here the combinatorial structure of interest corresponded to the set of pure-strategy Nash equilibria of the "true game". We presented a polynomial-time algorithm based on one-vs-rest multinomial logistic regression that recovered the ϵ-Nash equilibria set of the true game. Our method can be considered as a convex relaxation of an objective that directly maximizes the number of empirical Nash equilibria [HO15]. While the latter objective is hard to optimize, our convex relaxation approach can be solved in polynomial time.

Lastly, in Chapter 4 we looked at the problem of learning (randomized) structured predictors. The key computational bottleneck here was computing the partition function during learning. We circumvented the problem by proposing an efficient sampling scheme for approximating the partition function that only computes a constant factor approximation. We obtained Rademacher-based generalization guarantees for our randomized learning procedure.

The three approaches summarized above represent three general techniques for coming up with efficient algorithms for combinatorial problems in machine learning namely, exploiting structure, convex relaxations, and randomization. While all of these techniques have been extensively used in the machine learning literature for a wide variety of problems, this dissertation explored such techniques for learning DAGs, games, and structured predictors. The proposed algorithms for the aforementioned problems are in many cases the first computational and statistically efficient procedures known in the literature.

REFERENCES

[Bar82] Francisco Barahona. On the computational complexity of ising spin glass models. *Journal of Physics A: Mathematical and General*, 15(10):3241, 1982.

[Ben56] Joseph F Bennett. Determination of the number of independent parameters of a score matrix from the examination of rank orders. *Psychometrika*, 21(4):383–393, 1956.

[BH60] Joseph F Bennett and William L Hays. Multidimensional unfolding: Determining the dimensionality of ranked preference data. *Psychometrika*, 25(1):27–43, 1960.

[BSK06] Ben Blum, Christian R Shelton, and Daphne Koller. A continuation method for nash equilibria in structured games. *Journal of Artificial Intelligence Research*, 25:457–502, 2006.

[BV04] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[BZR11] Oren Ben-Zwi and Amir Ronen. Local and global price of anarchy of graphical games. *Theoretical Computer Science*, 412(12-14):1196–1207, 2011.

[CD06] Xi Chen and Xiaotie Deng. Settling the complexity of two-player nash equilibrium. In *Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on*, pages 261–272. IEEE, 2006.

[Chi96] David Maxwell Chickering. Learning bayesian networks is np-complete. In *Learning from data*, pages 121–130. Springer, 1996.

[Chi03] David Maxwell Chickering. Optimal Structure Identification with Greedy Search. *J. Mach. Learn. Res.*, 3:507–554, March 2003.

[CLL11] Tony Cai, Weidong Liu, and Xi Luo. A Constrained L1 Minimization Approach to Sparse Precision Matrix Estimation. *Journal of the American Statistical Association*, 106(494):594–607, 2011.

[Cov67] Thomas M Cover. The number of linearly inducible orderings of points in d-space. *SIAM Journal on Applied Mathematics*, 15(2):434–439, 1967.

[CSH08] Venkat Chandrasekaran, Nathan Srebro, and Prahladh Harsha. Complexity of inference in graphical models. In *Proceedings of the Twenty-Fourth Conference on Uncertainty in Artificial Intelligence*, pages 70–78. AUAI Press, 2008.

[Das99] Sanjoy Dasgupta. Learning polytrees. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 134–141. Morgan Kaufmann Publishers Inc., 1999.

[DGP09] Constantinos Daskalakis, Paul W. Goldberg, and Christos H. Papadimitriou. The complexity of computing a Nash equilibrium. *SIAM Journal on Computing*, 39(1):195–259, 2009.

[DST11] John Dunagan, Daniel A. Spielman, and Shang-Hua Teng. Smoothed analysis of condition numbers and complexity implications for linear programming. *Mathematical Programming*, 126(2):315–350, February 2011.

[FHT08] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

[GH16] Asish Ghoshal and Jean Honorio. From behavior to sparse graphical games: Efficient recovery of equilibria. In *Communication, Control, and Computing (Allerton), 2016 54th Annual Allerton Conference on*, pages 1220–1227. IEEE, 2016.

[GH17a] Asish Ghoshal and Jean Honorio. Information-theoretic limits of Bayesian network structure learning. In *Artificial Intelligence and Statistics*, pages 767–775, 2017.

[GH17b] Asish Ghoshal and Jean Honorio. Learning graphical games from behavioral data: Sufficient and necessary conditions. In *Artificial Intelligence and Statistics*, pages 1532–1540, 2017.

[GH17c] Asish Ghoshal and Jean Honorio. Learning Identifiable Gaussian Bayesian Networks in Polynomial Time and Sample Complexity. In *Advances in Neural Information Processing Systems*, pages 6460–6469, 2017.

[GH18a] Asish Ghoshal and Jean Honorio. Learning Linear Structural Equation Models in Polynomial Time and Sample Complexity [Forthcoming]. In *Artificial Intelligence and Statistics*, 2018.

[GH18b] Asish Ghoshal and Jean Honorio. Learning maximum-a-posteriori perturbation models for structured prediction in polynomial time. In *International Conference on Machine Learning*, pages 1749–1757, 2018.

[GH18c] Asish Ghoshal and Jean Honorio. Learning sparse polymatrix games in polynomial time and sample complexity [Forthcoming]. In *Artificial Intelligence and Statistics*, 2018.

[GHJ14] Andreea Gane, Tamir Hazan, and Tommi Jaakkola. Learning with maximum a-posteriori perturbation models. In *Artificial Intelligence and Statistics*, pages 247–256, 2014.

[GJ16] Vikas Garg and Tommi Jaakkola. Learning Tree Structured Potential Games. In *Advances in Neural Information Processing Systems 29*, pages 1552–1560, 2016.

[GMR08] Venkatesan Guruswami, Rajsekar Manokaran, and Prasad Raghavendra. Beating the random ordering is hard: Inapproximability of maximum acyclic subgraph. In *Foundations of Computer Science, 2008. FOCS'08. IEEE 49th Annual IEEE Symposium on*, pages 573–582. IEEE, 2008.

[Gum54] Emil Julius Gumbel. Statistical theory of extreme valuse and some practical applications. *Nat. Bur. Standards Appl. Math. Ser. 33*, 1954.

[HJ12] Tamir Hazan and Tommi Jaakkola. On the partition function and random maximum a-posteriori perturbations. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pages 1667–1674. Omnipress, 2012.

[HJ16] Jean Honorio and Tommi Jaakkola. Structured prediction: from gaussian perturbations to linear-time principled algorithms. In *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*, pages 271–278. AUAI Press, 2016.

[HMJ13] Tamir Hazan, Subhransu Maji, and Tommi Jaakkola. On sampling from the gibbs distribution with random maximum a-posteriori perturbations. In *Advances in Neural Information Processing Systems*, pages 1268–1276, 2013.

[HMKJ13] Tamir Hazan, Subhransu Maji, Joseph Keshet, and Tommi Jaakkola. Learning efficient random maximum a-posteriori predictors with non-decomposable loss functions. In *Advances in Neural Information Processing Systems*, pages 1887–1895, 2013.

[HO15] Jean Honorio and Luis Ortiz. Learning the structure and parameters of large-population graphical games from behavioral data. *Journal of Machine Learning Research*, 16:1157–1210, 2015.

[IO14] Mohammad T Irfan and Luis E Ortiz. On influence, stable behavior, and the most influential individuals in networks: A game-theoretic approach. *Artificial Intelligence*, 215:79–119, 2014.

[Jan68] E. Janovskaja. Equilibrium situations in multi-matrix games. *Litovskiĭ Matematicheskiĭ Sbornik*, 8:381–384, 1968.

[JLB11] Albert Xin Jiang and Kevin Leyton-Brown. Polynomial-time computation of exact correlated equilibrium in compact games. In *Proceedings of the 12th ACM conference on Electronic commerce*, pages 119–126. ACM, 2011.

[JN91] Charles R. Johnson and Peter Nylen. Monotonicity properties of norms. *Linear Algebra and its Applications*, 148:43–58, April 1991.

[JRVS11] Ali Jalali, Pradeep Ravikumar, Vishvas Vasuki, and Sujay Sanghavi. On Learning Discrete Graphical Models using Group-Sparse Regularization. In *AISTATS*, pages 378–387, 2011.

[JSG+10] Tommi S. Jaakkola, David Sontag, Amir Globerson, Marina Meila, and others. Learning Bayesian Network Structure using LP Relaxations. In *AISTATS*, pages 358–365, 2010.

[KKLO03] Sham Kakade, Michael Kearns, John Langford, and Luis Ortiz. Correlated equilibria in graphical games. In *Proceedings of the 4th ACM Conference on Electronic Commerce*, pages 42–47. ACM, 2003.

[KP07] Markus Kalisch and Bühlmann Peter. Estimating High-Dimensional Directed Acyclic Graphs with the PC-Algorithm. *Journal of Machine Learning Research*, 8:613–636, 2007.

[LB14] Po-Ling Loh and Peter Bühlmann. High-dimensional learning of linear causal networks via inverse covariance estimation. *Journal of Machine Learning Research*, 15(1):3065–3105, 2014.

[LMP01] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[McA07] David McAllester. Generalization bounds and consistency. *Predicting structured data*, pages 247–261, 2007.

[NWF78] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions—I. *Mathematical Programming*, 14(1):265–294, 1978.

[OHSJ14] Francesco Orabona, Tamir Hazan, Anand Sarwate, and Tommi Jaakkola. On measure concentration of random maximum a-posteriori perturbations. In *International Conference on Machine Learning*, pages 432–440, 2014.

[PB14] J. Peters and P. Bühlmann. Identifiability of Gaussian structural equation models with equal error variances. *Biometrika*, 101(1):219–228, 2014.

[Pea09] Judea Pearl. *Causality*. Cambridge university press, 2009.

[PMJS14] Jonas Peters, Joris M Mooij, Dominik Janzing, and Bernhard Schölkopf. Causal Discovery with Continuous Additive Noise Models. *Journal of Machine Learning Research*, 15(June):2009–2053, 2014.

[PR17] Gunwoong Park and Garvesh Raskutti. Learning Quadratic Variance Function (QVF) DAG models via OverDispersion Scoring (ODS). *arXiv:1704.08783 [cs, stat]*, April 2017. arXiv: 1704.08783.

[PY11] George Papandreou and Alan L Yuille. Perturb-and-map random fields: Using discrete optimization to learn and sample from energy models. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 193–200. IEEE, 2011.

[RBLZ08] Adam J. Rothman, Peter J. Bickel, Elizaveta Levina, and Ji Zhu. Sparse permutation invariant covariance estimation. *Electronic Journal of Statistics*, 2:494–515, 2008.

[Rob77] R W Robinson. Counting unlabeled acyclic digraphs. *Combinatorial Mathematics V*, 622:28–43, 1977.

[Rub16] Aviad Rubinstein. Settling the complexity of computing approximate two-player Nash equilibria. *arXiv:1606.04550 [cs]*, June 2016. arXiv: 1606.04550.

[RWRY11] Pradeep Ravikumar, Martin J. Wainwright, Garvesh Raskutti, and Bin Yu. High-dimensional covariance estimation by minimizing ℓ_1-penalized log-determinant divergence. *Electronic Journal of Statistics*, 5(0):935–980, 2011.

[SGS00] Peter Spirtes, Clark N Glymour, and Richard Scheines. *Causation, prediction, and search*. MIT press, 2000.

[SHHK06] Shohei Shimizu, Patrik O Hoyer, Aapo Hyvärinen, and Antti Kerminen. A Linear Non-Gaussian Acyclic Model for Causal Discovery. *Journal of Machine Learning Research*, 7:2003–2030, 2006.

[SIS+11] Shohei Shimizu, Takanori Inazumi, Yasuhiro Sogawa, Aapo Hyvärinen, Yoshinobu Kawahara, Takashi Washio, Patrik O. Hoyer, and Kenneth Bollen. DirectLiNGAM: A Direct Method for Learning a Linear Non-Gaussian Structural Equation Model. *Journal of Machine Learning Research*, 12(Apr):1225–1248, 2011.

[SMGJ10] David Sontag, Ofer Meshi, Amir Globerson, and Tommi S Jaakkola. More data means less inference: A pseudo-max approach to structured learning. In *Advances in Neural Information Processing Systems*, pages 2181–2189, 2010.

[ST03] Daniel A. Spielman and Shang-Hua Teng. Smoothed analysis of termination of linear programming algorithms. *Mathematical Programming*, 97(1):375–404, 2003.

[SW12] Narayana P Santhanam and Martin J Wainwright. Information-theoretic limits of selecting binary graphical models in high dimensions. *Information Theory, IEEE Transactions on*, 58(7):4117–4134, 2012.

[TBA06] Ioannis Tsamardinos, Laura E Brown, and Constantin F Aliferis. The max-min hill-climbing bayesian network structure learning algorithm. *Machine learning*, 65(1):31–78, 2006.

[TGK03] Ben Taskar, Carlos Guestrin, and Daphne Koller. Max-margin Markov Networks. In *Proceedings of the 16th International Conference on Neural Information Processing Systems*, NIPS'03, pages 25–32, Cambridge, MA, USA, 2003. MIT Press.

[Tro12] Joel A Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of computational mathematics*, 12(May):389–434, 2012.

[VDGB13] Sara Van De Geer and Peter Bühlmann. L0-Penalized maximum likelihood for sparse directed acyclic graphs. *Annals of Statistics*, 41(2):536–567, 2013.

[WWR10] Wei Wang, Martin J Wainwright, and Kannan Ramchandran. Information-theoretic bounds on model selection for Gaussian Markov random fields. In *ISIT*, pages 1373–1377. Citeseer, 2010.

[Yu97] Bin Yu. *Festschrift for Lucien Le Cam: Research Papers in Probability and Statistics*, chapter Assouad, Fano, and Le Cam, pages 423–435. Springer New York, New York, NY, 1997.

[ZS02] Jiji Zhang and Peter Spirtes. Strong faithfulness and uniform consistency in causal inference. In *Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence*, pages 632–639. Morgan Kaufmann Publishers Inc., 2002.

[ZS08] Jiji Zhang and Peter Spirtes. Detection of unfaithfulness and robust causal inference. *Minds and Machines*, 18(2):239–271, 2008.

A DETAILED PROOFS FOR SEMS

Proof [Proof of Proposition 2.3.1] When $\sigma_i^2 = \sigma^2$ for all $i \in [p]$, then (2.4) reduces to:

$$\sum_{l \in \phi_{\mathsf{G}[m,\tau]}(j)} \frac{B_{l,j}^2}{\sigma_l^2} > 0,$$

which holds trivially by causal minimality since $B_{l,j}^2 > 0$ for $(l,j) \in \mathsf{E}$. This proves part (i).

Now under (ii), $1/\sigma_i^2 - 1/\sigma_j^2 < 1\,, \forall i,j \in [p]$. Also, $B_{l,j}^2/\sigma_l^2 \geq 1$ for all $(l,j) \in \mathsf{E}$. Thus (2.4) is satisfied. ■

Proof [Proof of Lemma 2.3.2] Consider the following two SEMs over three nodes, where the noise variances are shown within braces below each node, and the edge weights are shown on the edges.

Both the SEMs make the following conditional independence assertion: $X_1 \perp\!\!\!\perp X_3 \mid X_2$, and are therefore Markov and causal minimal to $\mathcal{P}(X)$. Set $b_2 = \sqrt{1 - \frac{v_1}{v_2}}$. Then using the formulas derived in Proposition 2.3.2 it can be verified that the full precision matrix and the precision matrix obtained after removing vertex 1 ($\mathbf{\Omega}_{(-1)}$), for both the SEMs is:

$$\mathbf{\Omega} = \frac{1}{v_1} \times \begin{bmatrix} 1 & -\beta & 0 \\ -\beta & 1+\beta^2 & -b_2 \\ 0 & -b_2 & 1 \end{bmatrix} \qquad \mathbf{\Omega}_{(-1)} = \frac{1}{v_1} \times \begin{bmatrix} 1 & -b_2 \\ -b_2 & 1 \end{bmatrix} \tag{A.1}$$

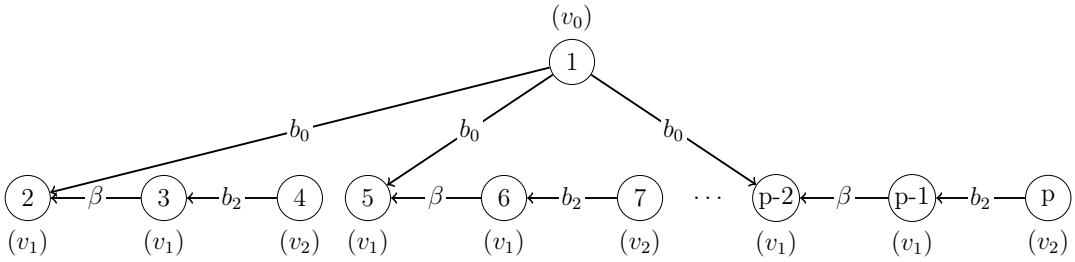
The SEM on the left does not satisfy Assumption 2.3.1 because vertex 3 is a non-terminal vertex but $3 \in \text{argmin}(\mathbf{diag}(\mathbf{\Omega}))$. The SEM on the right does not satisfy

Assumption 2.3.1 because after the vertex 1 is removed we have that vertex 2 is a non-terminal vertex but satisfies $2 \in \text{argmin}(\mathbf{diag}(\mathbf{\Omega}_{(-1)}))$.

Now we construct the subset $\widetilde{\mathcal{G}}_{p,d}$ with $p = 3k$ for $k = 1, 2, \ldots$, as follows. We randomly set the DAG structure over nodes $(3i-1), (3i)$ and $(3i+1)$ to one of the two configurations shown in the above figure. Therefore we have, $\left|\widetilde{\mathcal{G}}_{p,d}\right| = 2^{(p-1)/3}$. We generate matrices $\mathbf{B}(\beta)$ and $\mathbf{D}(v_1, v_2)$ as prescribed. The precision matrix block over the nodes $(3i-1), (3i)$, and $(3i+1)$, for $i \in [(p-1)/3]$, is given by (A.1), and all the other entries of the precision matrix are zeros. This proves our claim.

While the above constructions constructs a family of disconnected DAGs, with $d = 1$, it is easy to come up with subsets of DAGs that are connected and still satisfy the statement of the lemma. One such construction is shown below where $d = (p-1)/3$. The entries of the first row (and also the first column) of the precision matrix, for $i \in [(p-1)/3]$, are as follows:

$$\Omega_{1,1} = \frac{1}{v_0} + \frac{(p-1)b_0^2}{3v_1},\ \Omega_{1,3i-1} = -\frac{b_0}{v_1},\ \Omega_{1,3i} = \frac{b_0\beta}{v_1}.$$



As shown before, each triplet of nodes $(3i-1) \leftarrow (3i) \leftarrow (3i+1)$, for $i \in [(p-1)/3]$, can be oriented as $(3i-1) \leftarrow (3i) \rightarrow (3i+1)$ without changing the block of the precision matrix over the nodes $(3i-1), (3i)$ and $(3i+1)$, and the entries $\mathbf{\Omega}_{1,*}$ or $\mathbf{\Omega}_{*,1}$. ■

Proof [Proof of Proposition 2.3.2] From (2.2) we have that $(\mathbf{I} - \mathbf{B})X = N$, and since $(\mathbf{I} - \mathbf{B})$ is invertible, $X = (\mathbf{I} - \mathbf{B})^{-1}N$. Therefore:

$$\mathbf{\Sigma} = \mathbb{E}\left[XX^T\right] = \mathbb{E}\left[(\mathbf{I} - \mathbf{B})^{-1}NN^T(\mathbf{I} - \mathbf{B})^{-T}\right] = (\mathbf{I} - \mathbf{B})^{-1}\mathbf{D}(\mathbf{I} - \mathbf{B})^{-T}.$$

From which it follows that $\mathbf{\Omega} = (\mathbf{I}-\mathbf{B})^T\mathbf{D}^{-1}(\mathbf{I}-\mathbf{B})$, where $\mathbf{D}^{-1} = \mathbf{Diag}(1/\sigma_1^2, \ldots, 1/\sigma_p^2)$. From this the result for the entries of the precision matrix follows by sparsity pattern of $\mathbf{B}$. ■

Proof [Proof of Proposition 2.3.3] From (2.5) we have that for a terminal vertex i, $\Omega_{i,i} = 1/\sigma_i^2$, while for a non-terminal vertex j, $\Omega_{j,j} = 1/\sigma_j^2 + \sum_{l\in\phi(j)} B_{l,j}^2/\sigma_l^2$. Therefore, by Assumption 2.3.1 we have that for all non-terminal vertices j and terminal vertices i, $\Omega_{j,j} > \Omega_{i,i}$.

Now since every DAG has at least one terminal vertex, if $i \in \text{argmin}(\mathbf{diag}(\mathbf{\Omega}))$, then once again by Assumption 2.3.1, we have that i must be a terminal vertex. ■

Proof [Proof of Lemma 2.3.3] First note that since i is a terminal vertex, the autoregression matrix over $X_{-\mathsf{i}}$ is simply $\mathbf{B}_{-\mathsf{i},-\mathsf{i}}$. Denoting $\mathbf{D}' \stackrel{\text{def}}{=} \mathbf{Diag}(\sigma_1^2, \ldots, \sigma_{i-1}^2, \sigma_{i+1}^2, \sigma_p^2)$ and by Proposition 2.3.2 we have:

$$\begin{aligned}
\mathbf{\Omega}_{(-\mathsf{i})} &= (\mathbf{I} - \mathbf{B}_{-\mathsf{i},-\mathsf{i}})^T(\mathbf{D}')^{-1}(\mathbf{I} - \mathbf{B}_{-\mathsf{i},-\mathsf{i}}) = \sum_{j\in-\mathsf{i}} \frac{1}{\sigma_j^2}((\mathbf{e}_j)_{-\mathsf{i}} - \mathbf{B}_{j,-\mathsf{i}}^T)((\mathbf{e}_j)_{-\mathsf{i}}^T - \mathbf{B}_{-\mathsf{i},j}) \\
&= \sum_{j\in[p]} \frac{1}{\sigma_j^2}\left((\mathbf{e}_j - \mathbf{B}_{j,*}^T)(\mathbf{e}_j^T, -\mathbf{B}_{j,*})\right)_{-\mathsf{i},-\mathsf{i}} - \frac{1}{\sigma_i^2}\left((\mathbf{e}_i - \mathbf{B}_{i,*}^T)(\mathbf{e}_i^T - \mathbf{B}_{i,*})\right)_{-\mathsf{i},-\mathsf{i}} \\
&= \mathbf{\Omega}_{-\mathsf{i},-\mathsf{i}} - \frac{1}{\sigma_i^2}\left(\mathbf{B}_{i,-\mathsf{i}}^T\mathbf{B}_{i,-\mathsf{i}}\right) = \mathbf{\Omega}_{-\mathsf{i},-\mathsf{i}} - \Omega_{i,i}\frac{\mathbf{\Omega}_{i,-\mathsf{i}}^T}{\Omega_{i,i}}\frac{\mathbf{\Omega}_{i,-\mathsf{i}}}{\Omega_{i,i}} = \mathbf{\Omega}_{-\mathsf{i},-\mathsf{i}} - \frac{1}{\Omega_{i,i}}\mathbf{\Omega}_{-\mathsf{i},i}\mathbf{\Omega}_{i,-\mathsf{i}},
\end{aligned}$$

where in the last line we used the fact that for a terminal vertex $\Omega_{i,i} = 1/\sigma_i^2$ (Proposition 2.3.3), and $\mathbf{B}_{i,-\mathsf{i}} = -\mathbf{\Omega}_{i,-\mathsf{i}}/\Omega_{i,i}$ (Proposition 2.3.4). ■

Proof [Proof of Lemma 2.3.4] First consider the case when $j \notin \pi_{\mathsf{G}}(i)$. Then, for any $k \in [p] \setminus \{i, j\}$, $i \notin (\phi_{\mathsf{G}}(j) \cap \phi_{\mathsf{G}}(k))$. Therefore, by Proposition 2.3.2, $(\Omega_{(-\mathsf{i})})_{j,k} = \Omega_{j,k}$, and by symmetry of the precision matrix $(\Omega_{(-\mathsf{i})})_{k,j} = \Omega_{k,j}$. Thus, we have that for any (j, k) if at least one of $\{j, k\}$ is not in $\pi_{\mathsf{G}}(i)$, then $(\Omega_{(-\mathsf{i})})_{j,k} = \Omega_{j,k}$, which proves our first claim. Thus, the only remaining case to consider is when both $j, k \in \pi_{\mathsf{G}}(i)$. The are two ways is which the set $\mathcal{S}((\mathbf{\Omega}_{(-\mathsf{i})})_{j,*})$ can be larger than the set $\mathcal{S}(\mathbf{\Omega}_{j,*})$, i.e., the support set of the j-th node can increase after deleting the terminal node i.

The first being when $j, k \in \pi_{\mathsf{G}}(i)$ and either $(j,k) \in \mathsf{E}$ or $(k,j) \in \mathsf{E}$ but $\Omega_{j,k} = 0$, in which case we have:

$$\sum_{l \in \phi(j) \cap \phi(k)} {}^{(B_{l,j}B_{l,k})}/_{\sigma_l^2} = {}^{B_{j,k}}/_{\sigma_j^2} + {}^{B_{k,j}}/_{\sigma_k^2}.$$

Then, after removing the terminal node i, we have

$$(\Omega_{(-\mathsf{i})})_{j,k} = -{}^{B_{j,k}}/_{\sigma_j^2} - {}^{B_{k,j}}/_{\sigma_k^2} + \sum_{l \in (\phi(j) \cap \phi(k) \setminus \{i\})} {}^{(B_{l,j}B_{l,k})}/_{\sigma_l^2} \neq 0.$$

The other case is when $j, k \in \pi_{\mathsf{G}}(i)$, $(j,k) \notin \mathsf{E}$, $(k,j) \notin \mathsf{E}$ but $\Omega_{j,k} = 0$, in which case we have:

$$\sum_{l \in \phi(j) \cap \phi(k)} {}^{(B_{l,j}B_{l,k})}/_{\sigma_l^2} = 0.$$

Therefore, after removing the terminal node we have:

$$(\Omega_{(-\mathsf{i})})_{j,k} = \sum_{l \in (\phi(j) \cap \phi(k) \setminus \{i\})} {}^{(B_{l,j}B_{l,k})}/_{\sigma_l^2} \neq 0.$$

Thus, $\mathcal{S}((\mathbf{\Omega}_{(-\mathsf{i})})_{j,*}) \subseteq (\mathcal{S}(\mathbf{\Omega}_{j,*}) \setminus \{i\}) \cup \pi_{\mathsf{G}}(i)$. ■

Proof [Proof of Theorem 2.3.5] Let i_t be the terminal vertex identified in iteration t, $\mathcal{I}_t \stackrel{\text{def}}{=} \{i_1, \ldots, i_t\}$ and $\mathcal{R}_t \stackrel{\text{def}}{=} [p] \setminus \mathcal{I}_t$. Let $\mathbf{\Omega}_{(i)}$ be the precision matrix after iteration i. The correctness of the algorithm follows from the following loop invariants:

(i) By Lemma 2.3.3 we have that, $(\mathbf{\Omega}_{(t)})_{\mathcal{R}_t,\mathcal{R}_t}$ is the correct precision matrix over $X_{\mathcal{R}_t}$.

(ii) The algorithm identifies a correct terminal vertex in iteration t, since the matrix $(\mathbf{\Omega}_{(t-1)})_{\mathcal{R}_{t-1},\mathcal{R}_{t-1}}$ is the correct precision matrix over $X_{\mathcal{R}_{t-1}}$, the SEM over $X_{\mathcal{R}_{t-1}}$ satisfies Assumption 2.3.1 by definition, and $\forall i \in \mathcal{I}_{t-1}$, $\Omega_{i,i} = \infty$.

(iii) By proposition 2.3.3 we have that at the end of round t, the sub-matrix $\mathbf{B}_{\mathcal{I}_t,*}$ has been correctly set and that $\forall i \in \mathcal{I}_t$, $\pi_{\mathsf{G}}(i) = \mathcal{S}(\mathbf{B}_{i,*})$.

To see that the algorithm returns a unique autoregression matrix $\widehat{\mathbf{B}}$, consider the following. If at iteration t there is a unique minimizer of $\mathbf{diag}(\mathbf{\Omega}_{(t-1)})$, which implies

a single terminal vertex, then the algorithm selects it and the incoming edge weights of the node is uniquely determined. While, in iteration t if there are multiple terminal vertices, leading to multiple minimizers of $\mathbf{diag}(\mathbf{\Omega}_{(t-1)})$, then the order in which they are eliminated does not matter. Or in other words, once a vertex becomes a terminal vertex, for instance after deletion of its children, its edge weights do not change. To see this, assume that there are two terminal vertices, i and j after iteration $t-1$. Then i and j are not in each other's parent sets. Therefore, if node i is eliminated in iteration t, then by Lemma 2.3.4 we have that $(\Omega_{(t)})_{j,k} = (\Omega_{(t-1)})_{j,k}$, $\forall k \in \pi_{\mathsf{G}}(j)$. Hence, we have that $\mathbf{B}$ is the unique autoregression matrix returned by the algorithm. ■

Proof [Proof of Lemma 2.3.9] Let $\mathbf{\Omega}_{(-\mathsf{i})} = (\boldsymbol{\omega}_j)_{j\in-\mathsf{i}}$ be the true precision matrix over $X_{-\mathsf{i}}$ and let $\widehat{\mathbf{\Omega}}' = (\boldsymbol{\omega}'_j)_{j\in[p]}$ be the matrix returned by the function UPDATE. The estimator $\widehat{\mathbf{\Omega}}_{(-\mathsf{i})} = (\widehat{\boldsymbol{\omega}}_j)_{j\in-\mathsf{i}}$ of $\mathbf{\Omega}_{(-\mathsf{i})}$ can be obtained by solving (2.7) using $\mathbf{\Sigma}^n_{-\mathsf{i},-\mathsf{i}}$. By Lemma 2.3.7, and the facts that $\left|\mathbf{\Sigma}^n_{-\mathsf{i},-\mathsf{i}} - \mathbf{\Sigma}_{-\mathsf{i},-\mathsf{i}}\right|_\infty \leq \left|\mathbf{\Sigma}^n - \mathbf{\Sigma}\right|_\infty$ and $\left\|\mathbf{\Omega}_{(i)}\right\|_1 \leq M$, we have that $\left|\mathbf{\Omega}_{(-\mathsf{i})} - \widehat{\mathbf{\Omega}}_{(-\mathsf{i})}\right| \leq 4M\lambda_n$. Since i is a terminal vertex, by Proposition 2.3.4 we have $\pi_{\mathsf{G}}(i) = \mathcal{S}(\mathbf{\Omega}_{i,*}) \setminus \{i\}$. Further, since $\mathcal{S}(\mathbf{\Omega}_{j,*}) \subseteq \mathcal{S}(\widehat{\mathbf{\Omega}}_{j,*})$, $\forall j \in [p]$, we have by Assumption 2.3.8 (ii) that, $\pi_{\mathsf{G}}(i) \subseteq \widehat{\pi}(i) = \mathcal{S}(\widehat{\mathbf{\Omega}}_{i,*}) \setminus \{i\} \subseteq \widehat{\mathsf{S}}$. By Lemma 2.3.4 and Assumption 2.3.8 (ii) we have that $\forall j \in \widehat{\mathsf{S}}_j$, $\mathcal{S}(\boldsymbol{\omega}_j) \subseteq \mathcal{S}\left(\mathbf{\Omega}_{j,*} \setminus \{i\}\right) \cup \pi_{\mathsf{G}}(i) \subseteq \mathcal{S}\left(\widehat{\mathbf{\Omega}}_{j,*} \setminus \{i\}\right) \cup \widehat{\pi}(i) \stackrel{\text{def}}{=} \widehat{\mathsf{S}}_j$. Or in other words we have $\left(\mathbf{\Omega}_{(i)}\right)_{j,\widehat{\mathsf{S}}^c_j} = \left(\mathbf{\Omega}_{(i)}\right)_{\widehat{\mathsf{S}}^c_j,j} = \mathbf{0}$. Now for $j \in -\mathsf{i}$ we set $(\boldsymbol{\omega}'_j)_{\widehat{\mathsf{S}}_j} = \bar{\boldsymbol{\omega}}_j$ and $(\boldsymbol{\omega}'_j)_{\widehat{\mathsf{S}}^c_j} = \mathbf{0}$, where $\bar{\boldsymbol{\omega}}_j$ is obtained by solving:

$$\begin{aligned} \underset{\boldsymbol{\omega}\in\mathbb{R}^{|\widehat{\mathsf{S}}_j|}}{\text{argmin}} \quad & \|\boldsymbol{\omega}\|_1, \\ \text{sub. to} \quad & \left|\mathbf{\Sigma}^n_{k,\widehat{\mathsf{S}}_j}\boldsymbol{\omega}\right| \leq \lambda_n,\ \forall k \notin \{i,j\}, \\ & \left|\mathbf{\Sigma}^n_{j,\widehat{\mathsf{S}}_j}\boldsymbol{\omega} - 1\right| \leq \lambda_n. \end{aligned}$$

Since $\bar{\boldsymbol{\omega}}_j$ is a solution to the above linear program, we have that $\left|\mathbf{\Sigma}^n_{-\mathsf{i},-\mathsf{i}}\boldsymbol{\omega}'_j - \mathbf{e}_j\right| \leq \lambda_n$ and $\left\|\boldsymbol{\omega}'_j\right\|_1 \leq \|\widehat{\boldsymbol{\omega}}_j\|_1$. Therefore, $\left|\mathbf{\Omega}_{(-\mathsf{i})} - \widehat{\mathbf{\Omega}}'_{-\mathsf{i},-\mathsf{i}}\right| \leq 4M\lambda_n$. Moreover, by Assumption 2.3.8 (ii), and the fact that $\widehat{\mathbf{\Omega}}'_{i,*} = \widehat{\mathbf{\Omega}}'_{*,i} = \mathbf{0}$, we get: $\mathcal{S}(\mathbf{\Omega}_{(-\mathsf{i})}) \subseteq \mathcal{S}(\widehat{\mathbf{\Omega}}')$. ■

Proof [Proof of Theorem 2.3.10] Let i_t denote the terminal vertex identified in iteration t and let $\mathcal{I}_t \stackrel{\text{def}}{=} \{i_1, \ldots, i_t\}$. Let $\mathcal{R}_t \stackrel{\text{def}}{=} [p] \setminus \mathcal{I}_t$ denote the vertices remain-

ing after iteration t. Let $\widehat{\mathbf{\Omega}}_{(t)}$ denote the precision matrix at the end of iteration t, $\widehat{\mathbf{\Omega}}_{(\mathcal{R}_t)} \stackrel{\text{def}}{=} (\widehat{\mathbf{\Omega}}_{(t)})_{\mathcal{R}_t,\mathcal{R}_t}$, and $\mathbf{\Omega}^*_{(\mathcal{R}_t)}$ be the true precision matrix over $X_{\mathcal{R}_t}$. Since $\|\mathbf{\Omega}^*\|_1 \leq M$, where M is defined in (2.8), we have that $\lambda_n \geq M\,|\mathbf{\Sigma}^n - \mathbf{\Sigma}^*|_\infty \geq \|\mathbf{\Omega}^*\|_1\,|\mathbf{\Sigma}^n - \mathbf{\Sigma}^*|_\infty$. Therefore, by Lemma 2.3.7 and Assumption 2.3.8 (ii), we have that $\left|\widehat{\mathbf{\Omega}}_{(\mathcal{R}_0)} - \mathbf{\Omega}^*_{(\mathcal{R}_0)}\right|_\infty = \left|\widehat{\mathbf{\Omega}} - \mathbf{\Omega}^*\right|_\infty \leq 4M\lambda_n$, and $\mathcal{S}(\mathbf{\Omega}^*_{(\mathcal{R}_0)}) \subseteq \mathcal{S}(\widehat{\mathbf{\Omega}}_0)$. Therefore, by Assumption 2.3.8 we have that the Algorithm 1 identifies the correct terminal vertex in iteration 1. Therefore, by Lemma 2.3.9 we have that $\left|\mathbf{\Omega}^*_{(\mathcal{R}_{t_1})} - \widehat{\mathbf{\Omega}}_{(\mathcal{R}_{t_1})}\right| \leq 4M\lambda_n$ and $\mathcal{S}(\mathbf{\Omega}^*_{(\mathcal{R}_{t_1})}) \subseteq \widehat{\mathbf{\Omega}}_{(t_1)}$.

Let $\mathbf{E} = (\varepsilon_{i,j})$, where $\varepsilon_{i,j} = \Omega^*_{i,j} - \widehat{\Omega}_{i,j}$. To simplify notation in this paragraph, we will denote the i_1 vertex by simply i. Then, for any $j \neq i$, we have that

$$\begin{aligned}
\left|\widehat{B}_{i,j} - B^*_{i,j}\right| &= \left|\frac{\widehat{\Omega}_{i,j}}{\widehat{\Omega}_{i,i}} - \frac{\Omega^*_{i,j}}{\Omega^*_{i,i}}\right| = \left|\frac{\Omega^*_{ii}(\Omega^*_{i,j} - \varepsilon_{i,j}) - (\Omega^*_{i,i} - \varepsilon_{i,i})\Omega^*_{i,j}}{(\Omega^*_{i,i} - \varepsilon_{i,i})\Omega^*_{i,i}}\right| \\
&= \left|\frac{\Omega^*_{i,i}\varepsilon_{i,j} - \Omega^*_{i,j}\varepsilon_{i,i}}{(\Omega^*_{i,i} - \varepsilon_{i,i})\Omega^*_{i,i}}\right| = \left|\frac{\varepsilon_{i,i} - \sigma_i^2\Omega^*_{i,j}\varepsilon_{i,i}}{1/\sigma_i^2 - \varepsilon_{i,i}}\right| \\
&= \left|\frac{\varepsilon_{i,i} - B^*_{i,j}\varepsilon_{i,i}}{1/\sigma_i^2 - \varepsilon_{i,i}}\right| \\
&\leq \frac{4M\lambda_n(1 + \left|B^*_{i,j}\right|)}{|1/\sigma_i^2 - \varepsilon_{i,i}|} \leq 4cM(1 + \left|B^*_{i,j}\right|)\sigma_i^2\lambda_n,
\end{aligned}$$

where the second and third lines follow from the fact that i is a terminal vertex and therefore, $\Omega^*_{i,i} = 1/\sigma_i^2$ and $\Omega_{i,j} = -B_{i,j}/\sigma_i^2$. Therefore, we have that $\left|\mathbf{B}^*_{i_1,*} - \widehat{\mathbf{B}}_{i_1,*}\right|_\infty = 4cM(1 + B_{\max})\sigma^2_{\max}\lambda_n$.

Next, assume that the algorithm correctly identifies terminal vertices upto round t. Then $\left|\widehat{\mathbf{\Omega}}_{(\mathcal{R}_t)} - \mathbf{\Omega}^*_{(\mathcal{R}_t)}\right|_\infty \leq 4M\lambda_n$, $\mathcal{S}(\mathbf{\Omega}^*_{(\mathcal{R}_t)}) \subseteq \mathcal{S}(\widehat{\mathbf{\Omega}}_{(t)})$, and $\left|\mathbf{B}^*_{\mathcal{I}_t,\mathcal{I}_t} - \widehat{\mathbf{B}}_{\mathcal{I}_t,\mathcal{I}_t}\right| \leq 4cM(1+B_{\max})\sigma^2_{\max}\lambda_n$. Therefore, once again by Assumption 2.3.8, it follows that the algorithm identifies the correct terminal vertex in round $t+1$, $\left|\widehat{\mathbf{\Omega}}_{(\mathcal{R}_{t+1})} - \mathbf{\Omega}^*_{(\mathcal{R}_{t+1})}\right|_\infty \leq 4M\lambda_n$, $\mathcal{S}(\mathbf{\Omega}^*_{(\mathcal{R}_{t+1})}) \subseteq \mathcal{S}(\widehat{\mathbf{\Omega}}_{(t+1)})$, and $\left|\mathbf{B}^*_{\mathcal{I}_{t+1},\mathcal{I}_{t+1}} - \widehat{\mathbf{B}}_{\mathcal{I}_{t+1},\mathcal{I}_{t+1}}\right| \leq 4cM(1+B_{\max})\sigma^2_{\max}\lambda_n$. Hence, the final claim follows by induction. The claim that $\mathcal{S}(\mathbf{B}^*) \subseteq \mathcal{S}(\widehat{\mathbf{B}})$ follows from the fact that $\mathcal{S}(\mathbf{\Omega}^*) \subseteq \mathcal{S}(\widehat{\mathbf{\Omega}})$. Finally, since $\mathcal{S}(\mathbf{B}^*) \subseteq \mathcal{S}(\widehat{\mathbf{B}})$ implies that $\mathcal{T}_{\widehat{\mathsf{G}}} \subseteq \mathcal{T}_{\mathsf{G}^*}$.

■

Proof [Proof of Theorem 2.3.12] Given that the data was generated by the SEM $(\mathsf{G}^*, \mathbf{B}^*, \{\sigma_i^2\})$, each X_i can be written as follows:

$$X_i = \sum_{j \in \mathsf{A}_{\mathsf{G}^*}(i)} w_{i,j} N_j,$$

for some $w_{i,j} \neq 0$.

Sub-Gaussian case. N_i is sub-Gaussian with parameter $\sigma_i \nu$, X_i is sub-Gaussian with parameter $\nu \sqrt{\sum_{j \in \mathsf{A}_{\mathsf{G}^*}(i)} w_{i,j}^2 \sigma_i^2}$ and $\Sigma^*_{i,i} = \sum_{j \in \mathsf{A}_{\mathsf{G}^*}(i)} w_{i,j}^2 \sigma_i^2$. Therefore, it follows that $X_i / \sqrt{\Sigma^*_{i,i}}$ is sub-Gaussian with parameter ν. From Lemma 1 of [RWRY11] and Theorem 2.3.10 we have that the regularization parameter λ_n need to satisfy the following bound in order to guarantee that $\left| \widehat{\mathbf{B}} - \mathbf{B}^* \right|_\infty \leq \varepsilon$:

$$MC_1 \sqrt{\frac{2}{n} \log\left(\frac{2p}{\sqrt{\delta}}\right)} \leq \lambda_n \leq \frac{\varepsilon}{c4M(1 + B_{\max})\sigma^2_{\max}}. \tag{A.2}$$

The above holds in the regime where the number of samples scales as given in the statement of the Theorem.

Bounded moment case. In this case we have:

$$\left(\sqrt{\mathbf{\Sigma}^*_{i,i}}\right)^{4m} = \left(\sum_{j \in \mathsf{A}_{\mathsf{G}^*}(i)} w_{i,j}^2 \sigma_i^2\right)^{2m} \geq \sum_{j \in \mathsf{A}_{\mathsf{G}^*}(i)} (w_{i,j}\sigma_i)^{4m} \tag{A.3}$$

Now, by Rosenthal's inequality we have:

$$\begin{aligned}
\mathbb{E}\left[(X_i)^{4m}\right] &\leq C_m \left\{ \sum_{j \in \mathsf{A}_{\mathsf{G}^*}(i)} w_{i,j}^{4m} \mathbb{E}\left[N_j^{4m}\right] + \sum_{j \in \mathsf{A}_{\mathsf{G}^*}(i)} w_{i,j}^{4m} \mathrm{Var}\left[N_i\right]^{2m} \right\} \\
&\leq C_m \left\{ \sum_{j \in \mathsf{A}_{\mathsf{G}^*}(i)} w_{i,j}^{4m} \sigma_i^{4m} K_m + \sum_{j \in \mathsf{A}_{\mathsf{G}^*}(i)} w_{i,j}^{4m} \sigma_i^{4m} \right\} \\
&= C_m(K_m + 1) \sum_{j \in \mathsf{A}_{\mathsf{G}^*}(i)} (w_{i,j}\sigma_i)^{4m}
\end{aligned} \tag{A.4}$$

Combining (A.3) and (A.4) we have

$$\mathbb{E}\left[\left(\frac{X_i}{\sqrt{\Sigma^*_{i,i}}}\right)^{4m}\right] \leq C_m(K_m + 1). \tag{A.5}$$

From the above and invoking Lemma 2 of [RWRY11] we get:

$$|\mathbf{\Sigma}^n - \mathbf{\Sigma}^*|_\infty < C_2 \left(\frac{p^2}{n^m \delta}\right)^{1/2m}, \tag{A.6}$$

with probability at least $1-\delta$. From Theorem 2.3.10 and (A.6) we have that the regularization parameter λ should satisfy the following for $\left|\widehat{\mathbf{B}} - \mathbf{B}^*\right|_\infty \leq \varepsilon$ to hold:

$$MC_2 \left(\frac{p^2}{n^m \delta}\right)^{1/2m} \leq \lambda_n \leq \frac{\varepsilon}{c4M(1+B_{\max})\sigma^2_{\max}}. \tag{A.7}$$

The above holds in the regime where the number of samples scales as given in the statement of the Theorem. ■

Proposition A.0.1 *Let* $(\mathsf{G}, \mathbf{B}, \{\sigma_i^2\})$ *be an SEM over* X *with* $\mathsf{G} \in \mathcal{G}_{p,d}$ *and precision matrix* $\mathbf{\Omega}$*. Let* ρ *be the maximum degree of a node in* G*. Then,* $|\mathcal{S}(\mathbf{\Omega}_{i,*}) \setminus \{i\}| \leq \rho^2 \leq d$, $\forall i \in [p]$.

Proof [Proof of Proposition A.0.1] For any node i, we will define the following set: $\mathsf{S}_\mathsf{G}(i) = \{j \in -\mathsf{i} \mid (i,j) \notin \mathsf{E} \wedge (j,i) \notin \mathsf{E} \wedge |\Omega_{i,j}| \neq 0\}$. Then, from Proposition 2.3.2, we have: if $j \in \mathsf{S}_\mathsf{G}(i)$ then $\Omega_{i,j} = \sum_{l \in \phi(i) \cap \phi(j)} {}^{(B_{l,i}B_{l,j})}/_{\sigma_l^2} \neq 0$. In other words, if $j \in \mathsf{S}_\mathsf{G}(i)$ then i and j have at least one common child, i.e., $\phi_\mathsf{G}(i) \cap \phi_\mathsf{G}(j) \neq \varnothing$. Node i can have at most ρ children, and each child $k \in \phi_\mathsf{G}(i)$ can have at most $\rho - 1$ parents other than i making them all members of $\mathsf{S}(i)$. Thus, $\mathsf{S}(i) \leq \rho(\rho-1)$. Therefore, we have that $\mathcal{S}(\mathbf{\Omega}_{i,*}) \setminus \{i\} \subseteq \mathsf{N}_\mathsf{G}(i) \cup \mathsf{S}_\mathsf{G}(i)$. Then, using the inclusion-exclusion principle we have that:

$$\begin{aligned} |\mathcal{S}(\mathbf{\Omega}_{i,*}) \setminus \{i\}| &\leq |\mathsf{N}_\mathsf{G}(i)| + |\mathsf{S}_\mathsf{G}(i)| - |\mathsf{N}_\mathsf{G}(i) \cap \mathsf{S}_\mathsf{G}(i)| \\ &= |\mathsf{N}_\mathsf{G}(i)| + |\mathsf{S}_\mathsf{G}(i)| \leq \rho + \rho(\rho-1) = \rho^2. \end{aligned}$$

The SEM which achieves the above upper bound is precisely the one constructed in the proof, i.e., there exists a node i with exactly ρ children, each child in turn has $\rho - 1$ "other parents" which are all members of $\mathsf{S}_\mathsf{G}(i)$. ■

Proposition A.0.2 *Given an SEM* $(\mathsf{G}, \mathbf{B}, \{\sigma_i^2\})$ *with precision matrix* $\mathbf{\Omega}$*, if* $\sigma_i^2 = \mathcal{O}(1)$ *for all* $i \in [p]$*, and* $B_{i,j} = \mathcal{O}(1)$ *for all* $(i,j) \in \mathsf{E}$*, then the quantity* M *as defined in* (2.8) *is* $\mathcal{O}(d)$.

Proof [Proof of Proposition A.0.2] Let $\sigma^2_{\min} = \min\{\sigma_i^2\}$. Let $\phi_{ij} \stackrel{\text{def}}{=} \phi(i) \cap \phi(j)$ and let $C_i \stackrel{\text{def}}{=} \{j \neq i \mid \phi_{ij} \neq \varnothing\}$. Define:

$$f_i(\mathbf{B}) = \frac{1}{\sigma^2_{\min}} \sum_{j \in \mathsf{N}(i)} |B_{i,j} + B_{j,i}| + \frac{1}{\sigma^2_{\min}} \sum_{j \in C_i} \Big| \sum_{l \in \phi_{ij}} B_{l,i} B_{l,j} \Big| + \frac{1}{\sigma^2_{\min}} \sum_{l \in \phi(i)} B^2_{l,i} + \frac{1}{\sigma^2_{\min}} \tag{A.8}$$

Then by (2.5) and by definition of M in (2.8), $M \leq \max_{i=1}^{p} f_i(\mathbf{B})$. Now, $f_i(\mathbf{B})$ is maximized when $\mathsf{MB}(i) = d$. There are two cases to consider: Case (i), $\phi(i) = \sqrt{d}$, $\pi(i) = \varnothing$, $C_i = d - \sqrt{d}$ and $|\phi_{ij}| = 1$ for all $j \in C_i$. In this case, the first and third term of (A.8) are $\mathcal{O}\left(\sqrt{d}\right)$ while the second term is $\mathcal{O}\left(d - \sqrt{d}\right)$, and therefore $M = \mathcal{O}(d)$. Case (ii), $\pi(i) = d$ or $\phi(i) = d$. In this case $C_i = \varnothing$ and therefore, the first and third term in (A.8) dominate and $M = \mathcal{O}(d)$. Therefore, in the worst case $M = \mathcal{O}(d)$. ■

B DETAILED PROOFS FOR GAMES

Proof [Proof of Lemma 3.3.1 (Minimum eigenvalue of population Hessian)]
Fix any $\boldsymbol{\theta}^0, \boldsymbol{\theta}^1 \in \Theta^i$, with $\boldsymbol{\theta}^1 \neq \mathbf{0}$. For any $t \in (-\infty, \infty)$, let $F(t; x_i) \stackrel{\text{def}}{=} (\boldsymbol{\theta}^0 + t\boldsymbol{\theta}^1)^T \mathbf{f}(x_i, \mathbf{x}_{-\mathrm{i}})$. Then for $\mathbf{x} \in \mathcal{A}$,

$$\ell(\mathbf{x}; \boldsymbol{\theta}^0 + t\boldsymbol{\theta}^1) = -F(t; x_i) + \log(\sum_{a \in \mathcal{A}_i} \exp(F(t; a))). \tag{B.1}$$

A little calculation shows that the double derivative of $\ell(\mathbf{x}; \boldsymbol{\theta}^0 + t\boldsymbol{\theta}^1)$ with respect to t is as follows:

$$\frac{\partial^2 \ell(\mathbf{x}; \boldsymbol{\theta}^0 + t\boldsymbol{\theta}^1)}{\partial t^2} = \sum_{a \in \mathcal{A}_i} \sigma(t; a) F'(a)^2 - \Big(\sum_{a \in \mathcal{A}_i} \sigma(t; a) F'(a)\Big)^2, \tag{B.2}$$

$$\sigma(t; b) = \frac{\exp(F(t; b))}{\sum_{a \in \mathcal{A}_i} \exp(F(t; a))}, \ (b \in \mathcal{A}_i)$$

where $F'(a)$ is the derivative of $F(t; a)$ with respect to t. Since $F(t; a)$ is a linear function of t, $F'(a)$ is not a function of t. Also note that $\sum_{a \in \mathcal{A}_i} \sigma(t; a) = 1$. Since $\boldsymbol{\theta}^0, \boldsymbol{\theta}^1$ have bounded norm and $t \in (-\infty, \infty)$, we have that $\sigma(t; a) > 0, \forall a \in \mathcal{A}_i$. Therefore, from (B.2), the strict convexity of $(\cdot)^2$ and Jensen's inequality, we have:

$$\frac{\partial^2 \ell(\mathbf{x}; \boldsymbol{\theta}^0 + t\boldsymbol{\theta}^1)}{\partial t^2} > 0 \qquad\qquad (\forall t \in (-\infty, \infty)).$$

Thus we have that $\ell(\mathbf{x}, \boldsymbol{\theta})$ is strongly convex, i.e., $\lambda_{\min}(\mathbf{H}^i(\mathbf{x}; \boldsymbol{\theta})) > 0$, $\forall \boldsymbol{\theta} \in \Theta^i$. Finally, by concavity of $\lambda_{\min}(\cdot)$ [BV04] and the Jensen's inequality we have:

$$\lambda_{\min}(\mathbf{H}^i(\boldsymbol{\theta}^i)) = \lambda_{\min}(\mathbb{E}_{[\mathbf{x}]} \mathbf{H}^i(\mathbf{x}; \boldsymbol{\theta}^i)) \geq \mathbb{E}_{[\mathbf{x}]} \lambda_{\min}(\mathbf{H}^i(\mathbf{x}; \boldsymbol{\theta}^i)) > 0.$$

■

Proof [Proof of Lemma 3.3.2 (Minimum eigenvalue of finite sample Hessian)]
To simply notation in the proof we will denote S_i by S. The (j,k) block of $\mathbf{H}(\mathcal{D};\boldsymbol{\theta}_S)$, where $j,k \in \{0\} \cup \mathcal{N}_i$, can be written as:

$$\mathbf{H}_{j,k}(\mathcal{D};\boldsymbol{\theta}_S) = \underbrace{\sum_{l=1}^{n}\sum_{a\in\mathcal{A}_i} \sigma^i(a,\mathbf{x}_{-\mathsf{i}}^{(l)};\boldsymbol{\theta}_S)\mathbf{f}^{i,j}(a,x_j^{(l)})(\mathbf{f}^{i,k}(a,x_k^{(l)}))^T}_{\mathbf{B}_{j,k}(\mathcal{D};\boldsymbol{\theta}_S)} - \underbrace{\sum_{l=1}^{n}\sum_{a,b\in\mathcal{A}_i} \sigma^i(a,\mathbf{x}_{-\mathsf{i}}^{(l)};\boldsymbol{\theta}_S)\mathbf{f}^{i,j}(a,x_j^{(l)})\mathbf{f}^{i,k}(b,x_k^{(l)})^T}_{\mathbf{R}_{j,k}(\mathcal{D};\boldsymbol{\theta}_S)},$$

where the matrices $\mathbf{B}$ and $\mathbf{R}$ have been defined above (blockwise). Since the matrix $\mathbf{R}$ is positive semi-definite $\lambda_{\max}(\mathbf{H}(\mathcal{D};\boldsymbol{\theta}_S)) \leq \lambda_{\max}(\mathbf{B}(\mathcal{D};\boldsymbol{\theta}_S))$. Further, since $\mathbf{B}$ is positive semi-definite, we have, from Lemma B.0.1:

$$\begin{aligned}
&\lambda_{\max}(\mathbf{B}(\mathcal{D};\boldsymbol{\theta}_S)) \\
&\leq \sum_{j\in\{0\}\cup\mathcal{N}_i} \lambda_{\max}(\mathbf{B}_{j,j}(\mathcal{D};\boldsymbol{\theta}_S)) \\
&\leq (d_i+1)\max_{j\in\{0\}\cup\mathcal{N}_i} \lambda_{\max}\Big(\frac{1}{n}\sum_{l=1}^{n}\sum_{a\in\mathcal{A}_i}\sigma^i(a,\mathbf{x}_{-\mathsf{i}}^{(l)};\boldsymbol{\theta}_S)\mathbf{f}^{i,j}(a,x_j^{(l)})(\mathbf{f}^{i,j}(a,x_j^{(l)}))^T\Big) \\
&\leq \frac{(d_i+1)}{n}\max_{j\in\{0\}\cup\mathcal{N}_i} \sum_{l=1}^{n}\sum_{a\in\mathcal{A}_i}\sigma^i(a,\mathbf{x}_{-\mathsf{i}}^{(l)};\boldsymbol{\theta}_S)\lambda_{\max}\left(\mathbf{f}^{i,j}(a,x_j^{(l)})(\mathbf{f}^{i,j}(a,x_j^{(l)}))^T\right) \\
&= d_i+1.
\end{aligned}$$

Thus we have that $\lambda_{\max}(\mathbf{H}(\mathcal{D};\boldsymbol{\theta}_S)) \leq \lambda_{\max}(\mathbf{B}(\mathcal{D};\boldsymbol{\theta}_S)) \leq d_i+1 \stackrel{\text{def}}{=} R$. Also note that $\mathbf{H}(\mathcal{D};\boldsymbol{\theta}_S) \in \mathbb{R}^{|S|\times|S|}$, with $|S| \leq m_i(1+d_i m)$. Then using the matrix Chernoff bounds by [Tro12], we have:

$$\Pr\left\{\lambda_{\min}(\mathbf{H}(\mathcal{D};\boldsymbol{\theta}_S)) \leq (1-\delta)\lambda_{\min}\right\} \leq |S|\left(\frac{\exp(-\delta)}{(1-\delta)^{(1-\delta)}}\right)^{(n\lambda_{\min}/R)}$$

Setting $\delta = 1/2$ we get:

$$\Pr\left\{\lambda_{\min}(\mathbf{H}(\mathcal{D};\boldsymbol{\theta}_S)) \geq \frac{\lambda_{\min}}{2}\right\} \geq 1 - m_i(1+d_i m)\exp\left(-\frac{n\lambda_{\min}}{8(d_i+1)}\right)$$

Controlling the probability of error to be at most δ we obtain the lower bound on the number of samples. ■

Proof [Proof of Lemma 3.3.3 (Gradient bound)]

A simple calculation shows that

$$\frac{\partial \ell^i(\mathbf{x};\boldsymbol{\theta}^i)}{\partial \boldsymbol{\theta}^{i,j}} = -\mathbf{f}^{i,j}(x_i, x_j) + \sum_{a\in\mathcal{A}_i} \sigma^i(a, \mathbf{x}_{-\mathsf{i}};\boldsymbol{\theta}^i)\mathbf{f}^{i,j}(a, x_j), \tag{B.3}$$

where $\sigma^i(\cdot)$ has been defined in (3.12). Let

$$\mathbf{g}(\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(n)}) = (g_j(\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(n)}))_{j\in\{0\}\cup\mathcal{N}_i},$$

where $g_j(\cdot) = \left\| \frac{1}{n}\sum_{l=1}^n \frac{\partial \ell^i(\mathbf{x}^{(l)};\boldsymbol{\theta}^i)}{\partial \boldsymbol{\theta}^{i,j}} \right\|_2$. Then we have that $\|\mathbf{g}(\cdot)\|_\infty = \|\nabla L^i(\mathcal{D};\boldsymbol{\theta}^i)\|_{\infty,2}$ and $\|\mathbb{E}[\mathbf{x}]\,\mathbf{g}(\cdot)\|_\infty = \|\mathbb{E}[\mathbf{x}]\,\nabla\ell^i(\mathbf{x};\boldsymbol{\theta}^i)\|_{\infty,2} = \nu$. Then, for any $\mathbf{x}^{(l)} \neq \mathbf{x}^{(l)'}$ we have that:

$$\begin{aligned}
&\Big| g_j(\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(l)},\ldots,\mathbf{x}^{(n)}) - g_j(\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(l)'},\ldots,g_j(\mathbf{x}^{(n)}) \Big| \\
&= \frac{1}{n}\Big\| \mathbf{f}^{i,j}(x_i^{(l)'}, x_j^{(l)'}) - \mathbf{f}^{i,j}(x_i^{(l)}, x_j^{(l)}) \\
&\qquad + \sum_{a\in\mathcal{A}_i} \sigma^i(a, \mathbf{x}_{-\mathsf{i}}^{(l)};\boldsymbol{\theta}^i)\mathbf{f}^{i,j}(a, x_j^{(l)}) - \sigma^i(a, \mathbf{x}_{-\mathsf{i}}^{(l)'};\boldsymbol{\theta}^i)\mathbf{f}^{i,j}(a, x_j^{(l)'}) \Big\| \\
&\leq \frac{1}{n}\Big(2 + \sum_{a\in\mathcal{A}_i} (\sigma^i(a, \mathbf{x}_{-\mathsf{i}}^{(l)};\boldsymbol{\theta}^i))^2 + (\sigma^i(a, \mathbf{x}_{-\mathsf{i}}^{(l)'};\boldsymbol{\theta}^i))^2\Big)^{1/2} \leq \frac{1}{n}(2+2)^{1/2} = 2/n,
\end{aligned}$$

where in the last line we used the fact that $\sum_a \sigma^i(a,\cdot) = 1$ along with the Cauchy-Schwartz inequality. Then using the McDiarmid's inequality we have:

$$\Pr\{|g_j(\cdot) - \mathbb{E}[\mathbf{x}]\, g_j(\cdot)| \leq t\} \geq 1 - 2\exp\Big(\frac{-nt^2}{2}\Big).$$

Then using a union bound over all j we have:

$$\begin{aligned}
&\Pr\Big\{\max_j |g_j(\cdot) - \mathbb{E}[\mathbf{x}]\, g_j(\cdot)| \leq t\Big\} \geq 1 - 2(d_i+1)\exp\Big(\frac{-nt^2}{2}\Big) \\
\implies &\Pr\{\|\mathbf{g}(\cdot) - \mathbb{E}[\mathbf{x}]\,\mathbf{g}(\cdot)\|_\infty \leq t\} \geq 1 - 2(d_i+1)\exp\Big(\frac{-nt^2}{2}\Big) \\
\implies &\Pr\{\|\mathbf{g}(\cdot)\|_\infty - \|\mathbb{E}[\mathbf{x}]\,\mathbf{g}(\cdot)\|_\infty \leq t\} \geq 1 - 2(d_i+1)\exp\Big(\frac{-nt^2}{2}\Big) \\
\implies &\Pr\{\|\mathbf{g}(\cdot)\|_\infty \leq \nu + t\} \geq 1 - 2(d_i+1)\exp\Big(\frac{-nt^2}{2}\Big),
\end{aligned}$$

where in the third line we used the reverse triangle inequality. Setting the probability of error to be δ and solving for t, we prove our claim. ■

Proof [Proof of Lemma 3.3.4 (Minimum population eigenvalue at arbitrary parameter)]

To simply notation in the proof we will denote S_i by S. The population Hessian matrix at $\mathbf{H}(\boldsymbol{\theta}_S)$ can also be written as $\mathbf{H}(\boldsymbol{\theta}^i_S + \boldsymbol{\Delta}_S)$, where $\boldsymbol{\Delta}_S = \boldsymbol{\theta}_S - \boldsymbol{\theta}^i_S$. Using the variational characterization of the minimum eigenvalue of $\mathbf{H}(\boldsymbol{\theta}^i_S + \boldsymbol{\Delta}_S)$ and the Taylor's theorem, we have:

$$\begin{aligned}
\lambda_{\min}(\mathbf{H}(\boldsymbol{\theta}^i_S + \boldsymbol{\Delta}_S)) &= \min_{\{\mathbf{y}\in\mathbb{R}^{|S|} | \|\mathbf{y}\|_2=1\}} \sum_{i,j\in S} y_i\{H_{i,j}(\boldsymbol{\theta}^i_S) + (\nabla H_{i,j}(\bar{\boldsymbol{\theta}}_S))^T\boldsymbol{\Delta}_S\}y_j \\
&\geq \lambda_{\min}(\mathbf{H}(\boldsymbol{\theta}^i_S)) - \max_{\{\mathbf{y}\in\mathbb{R}^{|S|} | \|\mathbf{y}\|_2=1\}} \sum_{i,j\in S} y_i\{(\nabla H_{i,j}(\bar{\boldsymbol{\theta}}_S))^T\boldsymbol{\Delta}_S\}y_j \\
&\geq \lambda_{\min}(\mathbf{H}(\boldsymbol{\theta}^i_S)) - \max_{\{\mathbf{y}\in\mathbb{R}^{|S|} | \|\mathbf{y}\|_2=1\}} \sum_{i,j\in S} y_i\{\left|(\nabla H_{i,j}(\bar{\boldsymbol{\theta}}_S))^T\boldsymbol{\Delta}_S\right|\}y_j,
\end{aligned} \tag{B.4}$$

where $\bar{\boldsymbol{\theta}} = t\boldsymbol{\theta}^i_S + (1-t)\boldsymbol{\theta}_S$ for some $t \in [0,1]$, and the third line follows from the monotonicity property of the spectral norm $|||\cdot|||_2$ [JN91]. For any vector $\boldsymbol{\theta} \in \Theta^i$, let $\mathbf{A}(\boldsymbol{\theta}_S) = (A_{i,j}(\boldsymbol{\theta}_S))$, where $A_{i,j}(\boldsymbol{\theta}_S) = \left|(\nabla H_{i,j}(\boldsymbol{\theta}_S))^T\boldsymbol{\Delta}_S\right|$. Then,

$$|||\mathbf{A}(\boldsymbol{\theta}_S)|||_2 = |||\mathbb{E}\left[\mathbf{x}\right]\mathbf{A}(\mathbf{x};\boldsymbol{\theta}_S)|||_2 \leq \max_{\mathbf{x}\in\mathcal{A}} |||\mathbf{A}(\mathbf{x};\boldsymbol{\theta}_S)|||_2. \tag{B.5}$$

Now consider the (j,k) block of $\mathbf{A}(\mathbf{x};\boldsymbol{\theta}_S)$ for any $\mathbf{x} \in \mathcal{A}$, where $j,k \in \{0\}\cup\mathcal{N}_i$. Then, from (3.11) we have that:

$$\begin{aligned}
\mathbf{A}_{j,k}(\mathbf{x};\boldsymbol{\theta}) = &\underbrace{\sum_{a\in\mathcal{A}_i} \left|(\nabla\sigma^i(a,\mathbf{x}_{-\mathrm{i}};\boldsymbol{\theta}))^T\boldsymbol{\Delta}_S\right| \mathbf{f}^{i,j}(a,x_j)(\mathbf{f}^{i,k}(a,x_k))^T}_{\mathbf{B}_{j,k}(\mathbf{x};\boldsymbol{\theta})} \\
&\underbrace{- \sum_{a,b\in\mathcal{A}_i} \left|\left\{\sigma^i(b,\mathbf{x}_{-\mathrm{i}})\nabla\sigma^i(a,\mathbf{x}_{-\mathrm{i}}) + \sigma^i(a,\mathbf{x}_{-\mathrm{i}})\nabla\sigma^i(b,\mathbf{x}_{-\mathrm{i}})\right\}^T\boldsymbol{\Delta}_S\right| \mathbf{f}^{i,j}(a,x_j)\mathbf{f}^{i,k}(a,x_k)^T}_{\mathbf{R}_{j,k}(\mathbf{x};\boldsymbol{\theta})},
\end{aligned}$$

where in the above we have dropped the $\boldsymbol{\theta}$'s from the $\sigma(\cdot,\cdot;\boldsymbol{\theta})$ for notational convenience. Thus, $\mathbf{A}(\mathbf{x};\boldsymbol{\theta}) = \mathbf{B}(\mathbf{x};\boldsymbol{\theta}) - \mathbf{R}(\mathbf{x};\boldsymbol{\theta})$, where the matrices $\mathbf{B}$ and $\mathbf{R}$ have been defined above (block-wise). Observe that the matrix $\mathbf{R}$ is positive semi-definite. Therefore, $|||\mathbf{A}(\mathbf{x};\boldsymbol{\theta})|||_2 \leq |||\mathbf{B}(\mathbf{x};\boldsymbol{\theta})|||_2$. Finally, since $\mathbf{B}$ is positive semi-definite, the

spectral norm of $\mathbf{B}$ is at most the sum of the spectral norms of the diagonal blocks (c.f. Lemma B.0.1). Therefore, we have

$$|||\mathbf{B}(\mathbf{x};\boldsymbol{\theta})|||_2 \leq \sum_{j\in\{0\}\cup\mathcal{N}_i} |||\mathbf{B}_{j,j}(\mathbf{x};\boldsymbol{\theta})|||_2 \leq (d_i+1)\Big(\max_{j\in\{0\}\cup\mathcal{N}_i} |||\mathbf{B}_{j,j}(\mathbf{x};\boldsymbol{\theta})|||_2\Big). \tag{B.6}$$

A little calculation shows that

$$\frac{\partial\,\sigma^i(a,\mathbf{x}_{-\mathrm{i}};\boldsymbol{\theta})}{\partial\boldsymbol{\theta}_j} = \sigma^i(a,\mathbf{x}_{-\mathrm{i}};\boldsymbol{\theta})\Big\{\mathbf{f}^{i,j}(a,x_j) - \sum_{a'\in\mathcal{A}_i}\sigma^i(a',\mathbf{x}_{-\mathrm{i}};\boldsymbol{\theta})\mathbf{f}^{i,j}(a',x_j)\Big\},$$

and $\|{}^{\partial\sigma^i(a,\mathbf{x}_{-\mathrm{i}};\boldsymbol{\theta})}/_{\partial\boldsymbol{\theta}_j}\|_\infty \leq {}^1/_4$. Further, since for any given $a\in\mathcal{A}_i$, at most m_j+1 elements of the partial derivative vector above is non-zero, we have $\|{}^{\partial\sigma^i(a,\mathbf{x}_{-\mathrm{i}};\boldsymbol{\theta})}/_{\partial\boldsymbol{\theta}_j}\|_2 \leq {}^{(m_j+1)}/_4$ and $\|\nabla\sigma^i(a,\mathbf{x}_{-\mathrm{i}};\boldsymbol{\theta})\|_{\infty,2} \leq {}^{(m_j+1)}/_4 \leq {}^{(m+1)}/_4$. Then from Cauchy-Schwartz inequality and the monotonicity property of spectral norm [JN91] we have:

$$\begin{aligned}|||\mathbf{B}_{j,j}(\mathbf{x};\boldsymbol{\theta})|||_2 &\leq \Big|\Big|\Big|\sum_{a\in\mathcal{A}_i}\big\|\nabla\sigma^i(a,\mathbf{x}_{-\mathrm{i}};\boldsymbol{\theta})\big)\big\|_{\infty,2}\,\|\boldsymbol{\Delta}_S\|_{1,2}\,\mathbf{f}^{i,j}(a,x_j)(\mathbf{f}^{i,j}(a,x_j))^T\Big|\Big|\Big|_2 \\ &\leq \frac{1}{4}m_i m\,\|\boldsymbol{\Delta}_S\|_{1,2}\end{aligned} \tag{B.7}$$

Putting together (B.4), (B.5), (B.6) and (B.7) we get

$$\begin{aligned}\lambda_{\min}(\mathbf{H}(\boldsymbol{\theta}^i_S+\boldsymbol{\Delta}_S)) &\geq \lambda_{\min}(\mathbf{H}(\boldsymbol{\theta}^i_S)) - |||\mathbf{A}(\boldsymbol{\theta}_S)|||_2 \\ &\geq \lambda_{\min}(\mathbf{H}(\boldsymbol{\theta}^i_S)) - (d_i+1)\Big(\max_{\mathbf{x}\in\mathcal{A}}\max_{j\in\{0\}\cup\mathcal{N}_i}|||\mathbf{B}_{j,j}(\mathbf{x};\boldsymbol{\theta})|||_2\Big) \\ &\geq \lambda_{\min}(\mathbf{H}(\boldsymbol{\theta}^i_S)) - \frac{1}{4}(d_i+1)m_i m\,\|\boldsymbol{\Delta}_S\|_{1,2}\,.\end{aligned}$$

■

Proof [Proof of Lemma 3.3.5 (Error of the i-th estimator on the support set)]
To simplify notation in the proof, we will write S instead of S_i. Recall that $L^i(\mathcal{D};\boldsymbol{\theta})$ is the empirical loss for the i-th player for parameter $\boldsymbol{\theta}$. For the purpose of the proof we will often write $L(\boldsymbol{\theta})$ instead of $L^i(\mathcal{D};\boldsymbol{\theta})$. Let $F(\boldsymbol{\theta}) = L(\boldsymbol{\theta}) + \lambda\,\|\boldsymbol{\theta}\|_{1,2}$. For any $\boldsymbol{\theta}\in\Theta^i$, let $\boldsymbol{\Delta}_S = \boldsymbol{\theta}_S - \boldsymbol{\theta}^i_S$ denote the difference between $\boldsymbol{\theta}$ and the true parameter $\boldsymbol{\theta}^i$ on the true support set S. We introduce the following shifted and reparameterized regularized loss function:

$$\widetilde{F}(\boldsymbol{\Delta}_S) = \underbrace{L(\boldsymbol{\theta}^i_S+\boldsymbol{\Delta}_S) - L(\boldsymbol{\theta}^i_S)}_{\text{term 1}} + \underbrace{\lambda(\big\|\boldsymbol{\theta}^i_S+\boldsymbol{\Delta}_S\big\|_{1,2} - \big\|\boldsymbol{\theta}^i_S\big\|_{1,2})}_{\text{term 2}}, \tag{B.8}$$

which takes the value 0 at the true parameter $\boldsymbol{\theta}^i$, i.e., $\widetilde{F}(\mathbf{0}) = 0$. Let $\widehat{\boldsymbol{\Delta}}_S = \widehat{\boldsymbol{\theta}}^i_S - \boldsymbol{\theta}^i_S$, where $\widehat{\boldsymbol{\theta}}^i$ minimizes $F(\boldsymbol{\theta})$. Since $\widehat{\boldsymbol{\theta}}^i$ minimizes $F(\boldsymbol{\theta})$, we must have that $\widetilde{F}(\widehat{\boldsymbol{\Delta}}_S) \leq 0$. Thus, in order to upper bound $\left\|\widehat{\boldsymbol{\Delta}}_S\right\|_{1,2} = \left\|\widehat{\boldsymbol{\theta}}_S - \boldsymbol{\theta}^i_S\right\|_{1,2} \leq b$, we show that there exists an $\ell_{1,2}$ ball of radius b such that function $\widetilde{F}(\boldsymbol{\Delta}_S)$ is strictly positive on the surface of the ball. To see this, assume the contrary, i.e, $\forall \boldsymbol{\Delta} \in \Theta^i \wedge \|\boldsymbol{\Delta}_S\|_{1,2} = b$, $\widetilde{F}(\boldsymbol{\Delta}_S) > 0$, but $\widehat{\boldsymbol{\Delta}}_S$ lies outside the ball, i.e., $\left\|\widehat{\boldsymbol{\Delta}}_S\right\|_{1,2} > b$. Then, there exists a $t \in (0,1)$ such that $(1-t)\mathbf{0} + t\widehat{\boldsymbol{\Delta}}_S$ lies on the surface of the ball, i.e., $\left\|(1-t)\mathbf{0} + t\widehat{\boldsymbol{\Delta}}_S\right\|_{1,2} = b$. However, by convexity of $\widetilde{F}$ we have that

$$0 < \widetilde{F}((1-t)\mathbf{0} + t\widehat{\boldsymbol{\Delta}}_S) \leq (1-t)\widetilde{F}(\mathbf{0}) + t\widetilde{F}(\widehat{\boldsymbol{\Delta}}_S) = t\widetilde{F}(\widehat{\boldsymbol{\Delta}}_S),$$

which implies that $\widetilde{F}(\widehat{\boldsymbol{\Delta}}_S) > 0$ and therefore is a contradiction to the fact that $\widetilde{F}(\widehat{\boldsymbol{\Delta}}_S) \leq 0$. Going forward, our strategy would be to lower bound $\widetilde{F}(\boldsymbol{\Delta}_S)$ in terms of $\|\boldsymbol{\Delta}_S\|_{1,2} = b$. We then set the lower bound to 0 and solve for b, to obtain the radius of the $\ell_{1,2}$ ball on which the function is non-negative. Towards that end we first lower bound the first term of (B.8).

Using the Taylor's theorem and the Cauchy-Schwartz inequality, for some $t \in [0,1]$, we have:

$$\begin{aligned}
&L(\boldsymbol{\theta}^i_S + \boldsymbol{\Delta}_S) - L(\boldsymbol{\theta}^i_S) \\
&\quad = \nabla L(\boldsymbol{\theta}^i_S)^T \boldsymbol{\Delta}_S + \boldsymbol{\Delta}_S^T \nabla^2 L(\boldsymbol{\theta}^i_S + t\boldsymbol{\Delta}_S)\boldsymbol{\Delta}_S, \\
&\quad \geq -\left\|\nabla L(\boldsymbol{\theta}^i_S)\right\|_{\infty,2} \|\boldsymbol{\Delta}_S\|_{1,2} + \|\boldsymbol{\Delta}_S\|_2^2 \, \lambda_{\min}(\mathbf{H}(\mathcal{D}; \boldsymbol{\theta}^i_S + t\boldsymbol{\Delta}_S)) \\
&\quad \geq -\frac{b\lambda}{2} + \frac{\|\boldsymbol{\Delta}_S\|_{1,2}^2}{d_i + 1} \lambda_{\min}(\mathbf{H}(\mathcal{D}; \boldsymbol{\theta}^i_S + t\boldsymbol{\Delta}_S)) \\
&\quad \geq -\frac{b\lambda}{2} + \frac{b^2}{2(d_i+1)} \left(C_{\min} - \frac{m^2 b (d_i+1)}{4} \right) \\
&\quad \geq -\frac{b\lambda}{2} + \frac{b^2 C_{\min}}{4(d_i+1)},
\end{aligned} \tag{B.9}$$

where the third follows from our assumption that $\|\nabla L(\boldsymbol{\theta}^i)\|_{\infty,2} \leq \lambda/2$ and the fact for any vector $\mathbf{x}$, $\|\mathbf{x}\|_2 \geq (1/\sqrt{g}) \|\mathbf{x}\|_{1,2}$ where the $\ell_{1,2}$ norm is evaluated over g groups. The fourth line follows from Lemma 3.3.4 with $t = 1$ and Lemma 3.3.2. Finally, in the last line we assumed that $b \leq 2C_{\min}/(m^2(d_i+1))$ — an assumption that we will verify

momentarily. The second term of (B.8) is easily lower bounded using the reverse triangle inequality as follows:

$$\lambda(\left\|\boldsymbol{\theta}^i_S + \boldsymbol{\Delta}_S\right\|_{1,2} - \left\|\boldsymbol{\theta}^i_S\right\|_{1,2}) \geq -\lambda \left\|\boldsymbol{\Delta}_S\right\|_{1,2} = -b\lambda \tag{B.10}$$

Putting together (B.8), (B.9) and (B.10) we get:

$$\widetilde{F}(\boldsymbol{\Delta}_S) \geq -\frac{b\lambda}{2} + \frac{b^2 C_{\min}}{4(d_i+1)} - b\lambda.$$

Setting the above to zero and solving for b we get:

$$b = \frac{6\lambda(d_i+1)}{C_{\min}}.$$

Finally, coming back to our assumption that $b \leq {}^{2C_{\min}}/_{(m^2(d_i+1))}$, it is easy to show that the assumption holds if the regularization parameter λ satisfies:

$$\lambda \leq \frac{C^2_{\min}}{3m^2(d_i+1)^2},$$

The lower bound on the number of samples is obtained by ensuring that the lower bound on λ is less than the upper bound. The final claim follows from using the high probability bound on $\left\|\nabla L(\boldsymbol{\theta}^i)\right\|_{\infty,2}$ from Lemma 3.3.3. ■

Proof [Proof of Lemma 3.3.6 (Error of the i-th parameter estimator)]
$\boldsymbol{\Delta} \stackrel{\text{def}}{=} \widehat{\boldsymbol{\theta}}_i - \boldsymbol{\theta}^i$. We will denote the true support of $\boldsymbol{\theta}^i$ by S, and the complement of S by S^c. We will also simply write $L(\boldsymbol{\theta})$ instead of $L^i(\mathcal{D};\boldsymbol{\theta})$. For any vector $\mathbf{y}$, let $\mathbf{y}_{\bar{S}}$ denote the vector $\mathbf{y}$ with elements not in the support set S zeroed out, i.e.,

$$(\mathbf{y}_{\bar{S}})_j = \begin{cases} y_j & j \in S, \\ 0 & \text{otherwise} \end{cases}$$

Then by definition of S, we have that $\left\|\boldsymbol{\theta}^i_{\bar{S}}\right\|_{1,2} = \left\|\boldsymbol{\theta}^i\right\|_{1,2}$.

$$\begin{aligned} \left\|\widehat{\boldsymbol{\theta}}^i\right\|_{1,2} &= \left\|\boldsymbol{\theta}^i + \boldsymbol{\Delta}\right\|_{1,2} = \left\|\boldsymbol{\theta}^i_{\bar{S}} + \boldsymbol{\Delta}_{\bar{S}} + \boldsymbol{\Delta}_{\bar{S}^c}\right\|_{1,2} \\ &= \left\|\boldsymbol{\theta}^i_{\bar{S}} + \boldsymbol{\Delta}_{\bar{S}}\right\|_{1,2} + \left\|\boldsymbol{\Delta}_{\bar{S}^c}\right\|_{1,2} \\ &\geq \left\|\boldsymbol{\theta}^i_{\bar{S}}\right\|_{1,2} - \left\|\boldsymbol{\Delta}_{\bar{S}}\right\|_{1,2} + \left\|\boldsymbol{\Delta}_{\bar{S}^c}\right\|_{1,2}, \end{aligned}$$

where in the second line follows from the fact that the index sets S and S^c have non-overlapping groups, and in the last line we used the reverse triangle inequality. Rearranging the terms of the previous equation, and from the fact that $\left\|\boldsymbol{\theta}^i_{\bar{S}}\right\|_{1,2} = \|\boldsymbol{\theta}^i\|_{1,2}$, we get:

$$\left\|\boldsymbol{\theta}^i\right\|_{1,2} - \left\|\widehat{\boldsymbol{\theta}}^i\right\|_{1,2} \leq \|\boldsymbol{\Delta}_{\bar{S}}\|_{1,2} - \|\boldsymbol{\Delta}_{\bar{S}^c}\|_{1,2} \tag{B.11}$$

Next, by optimality of $\widehat{\boldsymbol{\theta}}^i$ we have that $L(\boldsymbol{\theta}^i) + \lambda \|\boldsymbol{\theta}^i\|_{1,2} \geq L(\widehat{\boldsymbol{\theta}}^i) + \lambda \left\|\widehat{\boldsymbol{\theta}}^i\right\|_{1,2}$. Rearranging the terms and continuing, we get

$$\begin{aligned}
\lambda(\left\|\boldsymbol{\theta}^i\right\|_{1,2} - \left\|\widehat{\boldsymbol{\theta}}^i\right\|_{1,2}) &\geq L(\widehat{\boldsymbol{\theta}}^i) - L(\boldsymbol{\theta}^i) \\
&\geq (\nabla L(\widehat{\boldsymbol{\theta}}^i))^T(\widehat{\boldsymbol{\theta}}^i - \boldsymbol{\theta}^i) \\
&\geq -\left\|\nabla L(\widehat{\boldsymbol{\theta}}^i))^T\right\|_{\infty,2} \|\boldsymbol{\Delta}\|_{1,2} \\
&\geq -\frac{\lambda}{2} \|\boldsymbol{\Delta}\|_{1,2}\,,
\end{aligned} \tag{B.12}$$

where the third line follows from the convexity of $L(\cdot)$, the fourth line follows from the Cauchy-Schwartz inequality and the last line follows from our assumption that $\lambda \geq 2\|\nabla L(\boldsymbol{\theta}^i)\|_{\infty,2}$. Thus, from (B.11) and (B.12) we have that

$$\begin{aligned}
\frac{1}{2}\|\boldsymbol{\Delta}\|_{1,2} &\geq \|\boldsymbol{\Delta}_{\bar{S}^c}\|_{1,2} - \|\boldsymbol{\Delta}_{\bar{S}}\|_{1,2} \\
\implies \frac{1}{2}\|\boldsymbol{\Delta}_{\bar{S}}\|_{1,2} + \frac{1}{2}\|\boldsymbol{\Delta}_{\bar{S}^c}\|_{1,2} &\geq \|\boldsymbol{\Delta}_{\bar{S}^c}\|_{1,2} - \|\boldsymbol{\Delta}_{\bar{S}}\|_{1,2} \\
\implies 3\|\boldsymbol{\Delta}_{\bar{S}}\|_{1,2} &\geq \|\boldsymbol{\Delta}_{\bar{S}^c}\|_{1,2}\,.
\end{aligned}$$

Finally, from the above inequality, we have $\|\boldsymbol{\Delta}\|_{1,2} = \|\boldsymbol{\Delta}_{\bar{S}}\|_{1,2} + \|\boldsymbol{\Delta}_{\bar{S}^c}\|_{1,2} \leq 4\|\boldsymbol{\Delta}_S\|_{1,2}$. The final result follows from the upper bound on $\|\boldsymbol{\Delta}_S\|_{1,2}$ derived in Lemma 3.3.5. ■

Lemma B.0.1 (Max eigenvalue of block positive semi-definite matrix) *Let $\mathbf{X} \in \mathbb{R}^{n\times n} \succeq \mathbf{0}$ be any positive semi-definite matrix, with $\mathbf{X}_{i,i}$ being the i-th diagonal block of $\mathbf{X}$. Then*

$$\lambda_{\max}(\mathbf{X}) \leq \sum_i \lambda_{\max}(\mathbf{X}_{i,i})$$

Proof We will prove the result by decomposing $\mathbf{X}$ into two blocks as follows:

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_{1,1} & \mathbf{X}_{1,2} \\ \mathbf{X}_{2,1} & \mathbf{X}_{2,2} \end{bmatrix},$$

where $\mathbf{X}_{1,1} \in \mathbb{R}^{n_1 \times n_1}$, $\mathbf{X}_{2,2} \in \mathbb{R}^{n_2 \times n_2}$ and $n_1 + n_2 = 1$. The general result for multiple diagonal blocks is obtained by recursively decomposing the blocks $\mathbf{X}_{1,1}$ and $\mathbf{X}_{2,2}$. Any unit vector $\mathbf{x}$ can be written as $\mathbf{x} = c_1(\mathbf{x})\mathbf{x}_1(\mathbf{x}) + c_2(\mathbf{x})\mathbf{x}_2(\mathbf{x})$, with $\mathbf{x}_1(\mathbf{x}) = ({}^{x_1}/_{\|\mathbf{x}_1(\mathbf{x})\|_2}, \ldots, {}^{x_{n_1}}/_{\|\mathbf{x}_1(\mathbf{x})\|_2}, \mathbf{0})$, $\mathbf{x}_2(\mathbf{x}) = (\mathbf{0}, {}^{x_{n_2}}/_{\|\mathbf{x}_2(\mathbf{x})\|_2}, \ldots, {}^{x_n}/_{\|\mathbf{x}_2(\mathbf{x})\|_2})$, and $c_1(\mathbf{x}) = \|\mathbf{x}_1(\mathbf{x})\|_2$ (similarly $c_2(\mathbf{x})$). For notational simplicity we will drop the $(\mathbf{x})s$. Note that $c_1^2 + c_2^2 = 1$, thus $\mathbf{c} = (c_1, c_2)$ is also a unit vector. Further, for any unit vector $\mathbf{x}$, we have $\mathbf{x}^T\mathbf{X}\mathbf{x} = \mathbf{c}^T\mathbf{Y}\mathbf{c}$, where

$$\mathbf{Y} = \begin{bmatrix} \mathbf{x}_1^T\mathbf{X}\mathbf{x}_1 & \mathbf{x}_1^T\mathbf{X}\mathbf{x}_2 \\ \mathbf{x}_2^T\mathbf{X}\mathbf{x}_1 & \mathbf{x}_2^T\mathbf{X}\mathbf{x}_2 \end{bmatrix} \in \mathbb{R}^{2\times 2}.$$

Note that $\mathbf{x}_1^T\mathbf{X}\mathbf{x}_1 \leq \lambda_{\max}(\mathbf{X}_{1,1})$ and $\mathbf{x}_2^T\mathbf{X}\mathbf{x}_2 \leq \lambda_{\max}(\mathbf{X}_{2,2})$ for all $\mathbf{x}$. Thus, using the variational characterization of the maximum eigenvalue of $\mathbf{X}$ we get:

$$\begin{aligned}
\lambda_{\max}(\mathbf{X}) &= \max_{\|\mathbf{x}\|_2=1} \mathbf{x}^T\mathbf{X}\mathbf{x} \\
&= \max_{\{\mathbf{c}=(\|\mathbf{x}_1(\mathbf{x})\|_2, \|\mathbf{x}_2(\mathbf{x})\|_2) \,:\, \|\mathbf{x}\|_2=1\}} \mathbf{c}^T\mathbf{Y}\mathbf{c} \\
&\leq \max_{\|\mathbf{c}\|_2=1} \mathbf{c}^T\mathbf{Y}\mathbf{c} = \lambda_{\max}(\mathbf{Y}) \leq \mathrm{Tr}\,(\mathbf{Y}) \quad \text{(since } \mathbf{Y} \text{ is positive semi-definite)} \\
&\leq \lambda_{\max}(\mathbf{X}_{1,1}) + \lambda_{\max}(\mathbf{X}_{2,2}),
\end{aligned}$$

where the third line follows from the fact that the maximization is over a superset of the set $\{\mathbf{c} = (\|\mathbf{x}_1(\mathbf{x})\|_2, \|\mathbf{x}_2(\mathbf{x})\|_2) \,:\, \|\mathbf{x}\|_2 = 1\}$. ■

Proof [Proof of Theorem 3.3.7] Note that:

$$\left\|\mathbf{f}^i(x_i, \mathbf{x}_{-\mathrm{i}})\right\|_{\infty,2} = \max\{\left\|\mathbf{f}^{i,0}(x_i)\right\|_2, \left\|\mathbf{f}^{i,1}(x_i, x_1)\right\|_2, \ldots, \left\|\mathbf{f}^{i,p}(x_i, x_p)\right\|_2\} = 1, \quad \text{(B.13)}$$

for any $\mathbf{x} \in \mathcal{A}$, since each binary vector $\mathbf{f}^{i,j}(x_i, x_j)$ has a single "1" at exactly one location. Then, from the Cauchy-Schwartz inequality, Lemma 3.3.6, and a union bound over all players, we have that:

$$\begin{aligned}
&(\forall \mathbf{x} \in \mathcal{A},\, \forall i \in [p]) \;\; \left|\widehat{u}^i(x_i, \mathbf{x}_{-\mathrm{i}}) - u^i(x_i, \mathbf{x}_{-\mathrm{i}})\right| \\
&= \left|(\widehat{\boldsymbol{\theta}}^i - \boldsymbol{\theta}^i)^T \mathbf{f}^i(x_i, \mathbf{x}_{-\mathrm{i}})\right| \\
&\leq \left\|\widehat{\boldsymbol{\theta}}^i - \boldsymbol{\theta}^i\right\|_{1,2} \left\|\mathbf{f}^i(x_i, \mathbf{x}_{-\mathrm{i}})\right\|_{\infty,2} \\
&= \left\|\widehat{\boldsymbol{\theta}}^i - \boldsymbol{\theta}^i\right\|_{1,2} \leq \frac{24(d_i+1)}{C_{\min}}\lambda = \frac{\varepsilon}{2}, \qquad \text{(B.14)}
\end{aligned}$$

with probability at least $1 - p\delta$. Now consider any $\mathbf{x} \in \mathcal{NE}(\widehat{\mathcal{G}})$ and any $i \in [p]$. Since $\mathbf{x} \in \mathcal{NE}(\widehat{\mathcal{G}})$, we have from (B.14), $(\forall x_i' \in \mathcal{A}_i)$:

$$\begin{aligned}
u^i(x_i, \mathbf{x}_{-\mathrm{i}}) + \varepsilon/2 \geq \widehat{u}^i(x_i, \mathbf{x}_{-\mathrm{i}}) &\geq \widehat{u}^i(x_i', \mathbf{x}_{-\mathrm{i}}) \\
\implies u^i(x_i, \mathbf{x}_{-\mathrm{i}}) &\geq \widehat{u}^i(x_i', \mathbf{x}_{-\mathrm{i}}) - \varepsilon/2 \\
\implies u^i(x_i, \mathbf{x}_{-\mathrm{i}}) &\geq u^i(x_i', \mathbf{x}_{-\mathrm{i}}) - \varepsilon,
\end{aligned}$$

where the last line again follows from (B.14). This proves that $\mathcal{NE}(\widehat{\mathcal{G}}) \subseteq \varepsilon\text{-}\mathcal{NE}(\mathcal{G})$. Using exactly the same arguments as above, we can also show that for any $\mathbf{x} \in \mathcal{NE}(\mathcal{G})$:

$$\widehat{u}^i(x_i, \mathbf{x}_{-\mathrm{i}}) \geq \widehat{u}^i(x_i, \mathbf{x}_{-\mathrm{i}}) - \varepsilon \qquad\qquad (\forall x_i' \in \mathcal{A}_i),$$

which proves that $\mathcal{NE}(\mathcal{G}) \subseteq \varepsilon\text{-}\mathcal{NE}(\widehat{\mathcal{G}})$. Thus we have that $\mathcal{NE}(\widehat{\mathcal{G}}) = \varepsilon\text{-}\mathcal{NE}(\mathcal{G})$, i.e., the set of joint strategy profiles $\mathbf{x} \in \mathcal{NE}(\widehat{\mathcal{G}})$ form an ε-Nash equilibrium set of the true game $\mathcal{G}$. This proves our first claim. For our second claim, consider any $(x_i, \mathbf{x}_{-\mathrm{i}}) \in \mathcal{NE}(\mathcal{G})$ and $(x_i', \mathbf{x}_{-\mathrm{i}}) \notin \mathcal{NE}(\mathcal{G})$. Then:

$$\begin{aligned}
u^i(x_i, \mathbf{x}_{-\mathrm{i}}) &> u^i(x_i', \mathbf{x}_{-\mathrm{i}}) + \varepsilon \\
\implies \widehat{u}^i(x_i, \mathbf{x}_{-\mathrm{i}}) + \varepsilon/2 &> \widehat{u}^i(x_i', \mathbf{x}_{-\mathrm{i}}) - \varepsilon/2 + \varepsilon \\
\implies \widehat{u}^i(x_i, \mathbf{x}_{-\mathrm{i}}) &> \widehat{u}^i(x_i', \mathbf{x}_{-\mathrm{i}}) \\
\implies (x_i, \mathbf{x}_{-\mathrm{i}}) \in \mathcal{NE}(\widehat{\mathcal{G}}) &\wedge (x_i', \mathbf{x}_{-\mathrm{i}}) \notin \mathcal{NE}(\widehat{\mathcal{G}}),
\end{aligned}$$

where the first line holds by assumption, and the second line again follows from (B.14). Thus we have that $\mathcal{NE}(\mathcal{G}) = \mathcal{NE}(\widehat{\mathcal{G}})$. By setting the probability of error $p\delta = \delta'$ for

some $\delta' \in (0,1)$ we prove our claim. The second part of the lower bound on the number of samples is due to Lemma 3.3.2. ■

Proof [Proof of Theorem 3.4.1] Consider the following restricted ensemble $\widetilde{\mathfrak{G}} \subset \mathfrak{G}_{p,d,m}$ of p-player polymatrix games with degree d, and the set of pure-strategies of each player being $\mathcal{A}_i = [m]$. Each $\mathcal{G} = (G, \mathcal{U}) \in \widetilde{\mathfrak{G}}_{p,d,m}$ is characterized by a set $\mathcal{I}$ of *influential players*, and a set $\mathcal{I}^c \stackrel{\text{def}}{=} [p] \setminus \mathcal{I}$ of *non-influential* players, with $|\mathcal{I}| = d$. The graph G is a complete (directed) bipartite graph from the set $\mathcal{I}$ to $\mathcal{I}^c$. After picking the graph structure G, nature fixes the strategies of the influential players to some $\mathbf{a} \in \{\mathbf{b} \in [m]^{|\mathcal{I}|} \mid \exists i, j \in \mathcal{I} \text{ such that } b_i \neq b_j\}$. Finally, the payoff matrices are chosen as follows:

$$\begin{aligned} u^{i,i}(x_i) &= \mathbf{1}\left[x_i = a_i\right] && (\forall i \in \mathcal{I}) \\ u^{j,j}(x_j) &= \frac{1}{(2x_j)} && (\forall j \in \mathcal{I}^c) \\ u^{j,i}(x_j, x_i) &= \mathbf{1}\left[x_j = x_i\right] && (\forall i \in \mathcal{I} \wedge j \in \mathcal{I}^c). \end{aligned}$$

Therefore, each $\mathcal{G} \in \widetilde{\mathfrak{G}}$ game has a exactly one unique Nash equilibrium where the influential players play $\mathbf{a}$ (decided by nature) and the non-influential players play $\mathfrak{maj}(\mathbf{a})$ — where $\mathfrak{maj}(\mathbf{a})$ returns the majority strategy among $\mathbf{a}$, and in case of a tie between two or more strategies it returns the numerically lowest strategy (recall that the pure-strategy set for each player is $[m]$). Thus we have that $\left|\widetilde{\mathfrak{G}}\right| = (m^d - m)\binom{p}{d}$. Nature picks a game $\mathcal{G}$ uniformly at random from $\widetilde{\mathfrak{G}}$ by randomly selecting a set of d players as "influential", and then selecting a strategy profile $\mathbf{a}$ uniformly at random for the influential players and setting the payoff matrices as described earlier. Nature then generates a dataset $\mathcal{D}$ using the global noise model with parameter $q \in (1/m^p, 2/(m^p+1)]$. Then from the Fano's inequality we have that:

$$p_{\text{err}} \geq 1 - \frac{I(\mathcal{D}; \mathcal{G}) + \log 2}{H(\mathcal{G})}, \tag{B.15}$$

where $I(\cdot;\cdot)$ and $H(\cdot)$ denote mutual information and entropy respectively. The mutual information $I(\mathcal{D};\mathcal{G})$ can be bounded, using a result by [Yu97], as follows:

$$I(\mathcal{D};\mathcal{G}) \leq \frac{1}{\left|\widetilde{\mathfrak{G}}\right|^2} \sum_{\mathcal{G}_1 \in \widetilde{\mathfrak{G}}} \sum_{\mathcal{G}_2 \in \widetilde{\mathfrak{G}}} \mathbb{KL}\left(\mathcal{P}_{\mathcal{D}|\mathcal{G}=\mathcal{G}_1} \middle\| \mathcal{P}_{\mathcal{D}|\mathcal{G}=\mathcal{G}_2}\right), \tag{B.16}$$

where $\mathcal{P}_{\mathcal{D}|\mathcal{G}=\mathcal{G}_1}$ (respectively $\mathcal{P}_{\mathcal{D}|\mathcal{G}=\mathcal{G}_2}$) denotes the data distribution under $\mathcal{G}_1$ (respectively $\mathcal{G}_2$). The KL divergence term from B.16 can be bounded as follows:

$$\begin{aligned}
&\mathbb{KL}\left(\mathcal{P}_{\mathcal{D}|\mathcal{G}=\mathcal{G}_1} \middle\| \mathcal{P}_{\mathcal{D}|\mathcal{G}=\mathcal{G}_2}\right) \\
&= n \sum_{\mathbf{x} \in \mathcal{A}} \mathcal{P}_{\mathcal{D}|\mathcal{G}=\mathcal{G}_1} \log \frac{\mathcal{P}_{\mathcal{D}|\mathcal{G}=\mathcal{G}_1}}{\mathcal{P}_{\mathcal{D}|\mathcal{G}=\mathcal{G}_2}} \\
&= n \Bigg\{ \sum_{\mathbf{x} \in \mathcal{NE}(\mathcal{G}_1)} q \log \frac{q(m^p-1)}{1-q} + \\
&\qquad \sum_{\mathbf{x} \in \mathcal{NE}(\mathcal{G}_2)} \frac{(1-q)}{m^p-1} \log \frac{1-q}{q(m^p-1)} \Bigg\} \\
&= \frac{n(qm^p-1)}{m^p-1} \log\left(\frac{q(m^p-1)}{1-q}\right) \\
&\leq n \log\left(\frac{q(m^p-1)}{1-q}\right) \leq n \log 2,
\end{aligned} \tag{B.17}$$

where the first line follows from the fact that the samples are i.i.d. , the second line follows from the fact the the distributions $\mathcal{P}_{\mathcal{D}|\mathcal{G}=\mathcal{G}_1}$ and $\mathcal{P}_{\mathcal{D}|\mathcal{G}=\mathcal{G}_2}$ assign the same probability to $\mathbf{x} \in \mathcal{A} \setminus (\mathcal{NE}(\mathcal{G}_1) \cup \mathcal{NE}(\mathcal{G}_2))$, and the last line follows from the fact that $q \in (1/m^p, 2/(m^p+1)]$. Putting together (B.15), (B.16) and (B.17), we have that if

$$n \leq \frac{\log(m^d - m)\binom{p}{d}}{2 \log 2} - 1,$$

then $p_{\text{err}} \geq 1/2$. Since, learning the ensemble $\mathfrak{G}$ is at least as hard as learning a subset of $\mathfrak{G}$, our claim follows. ■

C DETAILED PROOFS FOR STRUCTURED PREDICTION

Proof [Proof of Theorem 4.2.1] Let

$$g_w(x,y) \stackrel{\text{def}}{=} \Pr_{\gamma\sim\mathcal{G}^r(\beta)}\left\{y \neq f_{w,\gamma}(x)\right\},$$
$$\mathfrak{G} \stackrel{\text{def}}{=} \{g_w \mid w \in \mathbb{R}^{d,s}\}.$$

Then by Rademacher based uniform convergence, with probability at least $1-\delta$ over the choice of m samples, we have that:

$$(\forall w \in \mathbb{R}^{d,s})\ L(w,\mathcal{D}) \leq L(w,\mathsf{S}) + 2\widehat{\mathfrak{R}}_{\mathsf{S}}(\mathfrak{G}) + 3\sqrt{\frac{\log 2/\delta}{2m}}, \tag{C.1}$$

where $\widehat{\mathfrak{R}}_{\mathsf{S}}(\mathfrak{G})$ denotes the empirical Rademacher complexity of $\mathfrak{G}$. Let $\sigma = (\sigma_i)_{i=1}^m$ be independent Rademacher variables. Also define $\mathcal{W} \stackrel{\text{def}}{=} \{w/\beta(w,m) \mid w \in \mathbb{R}^{d,s}\}$. Then,

$$\begin{aligned}
&\widehat{\mathfrak{R}}_{\mathsf{S}}(\mathfrak{G}) \\
&= \mathbb{E}_\sigma\left[\sup_{w\in\mathbb{R}^{d,s}} \frac{1}{m}\sum_{i=1}^m \sigma_i g_w(x_i,y_i)\right] \\
&= \frac{1}{m}\mathbb{E}_\sigma\left[\sup_{w\in\mathbb{R}^{d,s}} \sum_{i=1}^m \sigma_i \Pr_{\gamma\sim\mathcal{G}^r(\beta)}\left\{y_i \neq f_{w,\gamma}(x_i)\right\}\right] \\
&\stackrel{(a)}{=} \frac{1}{m}\mathbb{E}_\sigma\left[\sup_{w\in\mathcal{W}} \sum_{i=1}^m \sigma_i \Pr_{\gamma\sim\mathcal{G}^r(1)}\left\{y_i \neq f_{w,\gamma}(x_i)\right\}\right] \\
&\leq \frac{1}{m}\mathbb{E}_{\gamma\sim\mathcal{G}^r(1)}\left[\mathbb{E}_\sigma\left[\sup_{w\in\mathcal{W}} \sum_{i=1}^m \sigma_i \mathbf{1}\left[y_i \neq f_{w,\gamma}(x_i)\right]\right]\right] \\
&\stackrel{(b)}{\leq} \frac{1}{m}\mathbb{E}_{\gamma\sim\mathcal{G}^r(1)}\left[\mathbb{E}_\sigma\left[\sup_{w\in\mathbb{R}^{d,s}} \sum_{i=1}^m \sigma_i \mathbf{1}\left[y_i \neq f_{w,\gamma}(x_i)\right]\right]\right],
\end{aligned}$$

where step (a) follows from $\Pr_{\gamma\sim\mathcal{G}^r(\beta)}\left\{y_i \neq f_{w,\gamma}(x_i)\right\} = \Pr_{\gamma\sim\mathcal{G}^r(1)}\left\{y_i \neq f_{w/\beta,\gamma}(x_i)\right\}$, and step (b) follows from $\mathcal{W} \subseteq \mathbb{R}^{d,s}$. We will enumerate the structured outputs $\mathfrak{Y}(x_i)$ as $y_{i,1},\ldots,y_{i,r}$. For any fixed γ, the weight vector w induces a linear ordering $\pi_i(\cdot;\gamma)$ over the structured outputs $\mathfrak{Y}(x_i)$, i.e., $\langle\phi(x_i,y_{i,\pi_i(1;\gamma)}),w\rangle + \gamma_1 > \langle\phi(x_i,y_{i,\pi_i(2;\gamma)}),w\rangle + \gamma_2 >$

$\ldots > \langle \phi(x_i, y_{i,\pi_i(r;\gamma)}), w \rangle + \gamma_r$. Let $\pi(\gamma) = \{\pi_i\}$ be the orderings over all m data points induced by a fixed weight vector w and fixed γ, and let $\Pi(\gamma)$ be the collection of all orderings $\pi(\gamma)$ over all $w \in \mathbb{R}^{d,s}$ for a fixed γ. Since w is s-sparse we have, from results by [Ben56, BH60, Cov67], that the number of possible linear orderings is $|\Pi(\gamma)| \leq \binom{d}{s}(mr)^{2s} \leq d^s(mr)^{2s}$. Therefore we have:

$$\begin{aligned}
&\widehat{\mathfrak{R}}_{\mathsf{S}}(\mathfrak{G}) \\
&\leq \frac{1}{m}\mathbb{E}_{\gamma\sim\mathcal{G}^r(\beta)}\left[\mathbb{E}_{\sigma}\left[\sup_{\pi(\gamma)\in\Pi(\gamma)}\sum_{i=1}^{m}\sigma_i \mathbf{1}\left[y_i \neq y_{i,\pi_i(1;\gamma)}\right]\right]\right] \\
&\overset{(a)}{\leq} \frac{1}{m}\sqrt{s(\log d + 2\log(mr))}\sqrt{m} \\
&= \sqrt{\frac{s(\log d + 2\log(mr))}{m}},
\end{aligned}$$

where the inequality (a) follows from the Massart's finite class lemma. ■

Proof [Proof of Lemma 4.3.1] For any $x \in \mathfrak{X}$, $\mathsf{T} \subseteq \mathfrak{Y}(x)$, $y \in \mathsf{T}$ and weight vector w:

$$\begin{aligned}
&\Pr_{\gamma}\{f_{w,\gamma}(x) = y\} - \Pr_{\gamma}\{f_{w,\gamma,\mathsf{T}}(x) = y\} \\
&= e^{\langle \phi(x,y), w\rangle}\left\{\frac{Z(w,x,\mathsf{T}) - Z(w,x)}{Z(w,x)Z(w,x,\mathsf{T})}\right\} \\
&= \frac{e^{\langle \phi(x,y), w\rangle}}{Z(w,x,\mathsf{T})}\frac{1}{Z(w,x)}\left\{-\sum_{y'\in\mathfrak{Y}(x)\setminus\mathsf{T}} e^{\langle \phi(x,y'), w\rangle}\right\} \\
&= -\Pr_{\gamma}\{f_{w,\gamma,\mathsf{T}}(x) = y\}\Pr_{\gamma}\{f_{w,\gamma}(x) \in \mathfrak{Y}(x) \setminus \mathsf{T}\}.
\end{aligned}$$

Since by construction $y_i \in \bar{\mathsf{T}}_i$, the final claim follows. ■

Proof [Proof of Lemma 4.3.4] Let

$$A_i(w,\mathsf{S}) \overset{\text{def}}{=} \Pr_{\gamma\sim\mathcal{G}(\beta)}\{f_{w,\gamma}(x_i) \neq y_i\} - \mathbb{E}_{\mathsf{T}_i}\left[\Pr_{\gamma\sim\mathcal{G}(\beta)}\left\{f_{w,\gamma,\bar{\mathsf{T}}_i}(x_i) \neq y_i\right\}\right]$$

be the i-th term of $A(w,\mathsf{S})$. We will consider two cases.

Case I: y_i is strictly the highest scoring structure for x_i under w, i.e., $\forall y \neq y_i \, \langle \phi(x_i, y_i), w \rangle > \langle \phi(x_i, y), w \rangle$. First note that:

$$A_i(w, \mathsf{S}) \leq \Pr_{\gamma \sim \mathcal{G}(\beta)} \left\{ f_{w,\gamma}(x_i) \neq y_i \right\}. \tag{C.2}$$

We will prove that $\Pr_{\gamma \sim \mathcal{G}(\beta)} \left\{ f_{w,\gamma}(x_i) \neq y_i \right\} \leq 1/\sqrt{m}$. Assume instead that the following holds: $\Pr_{\gamma \sim \mathcal{G}(\beta)} \left\{ f_{w,\gamma}(x_i) \neq y_i \right\} > 1/\sqrt{m}$. Then

$$\sum_{y \neq y_i} (\sqrt{m} - 1) e^{\langle \phi(x_i, y), w \rangle / \beta} > e^{\langle \phi(x_i, y_i), w \rangle / \beta}$$

Let $y' \in \mathfrak{Y}(x_i) \setminus \{y_i\}$ be such that $\langle \phi(x_i, y'), w \rangle$ is maximized. Then, $(r-1)(\sqrt{m} - 1) e^{\langle \phi(x_i, y'), w \rangle / \beta}$ upper bounds the left-hand side of the above equation. Taking log on both sides we get:

$$\beta > \frac{\langle \phi(x_i, y_i) - \phi(x_i, y'), w \rangle}{\log((r-1)(\sqrt{m} - 1))}$$

Since y_i is the unique maximizer of the score $\langle \phi(x_i, y_i), w \rangle$, $\phi(x_i, y')$ and $\phi(x_i, y_i)$ must differ on at least one element in the support set of w. This implies, from above and the assumption that the minimum non-zero element of $\phi(x, y)$ is at least 1:

$$\beta > \frac{w_{\min}}{\log((r-1)(\sqrt{m} - 1))},$$

which violates Assumption 4.3.3. Therefore from (C.2) we have that $A_i(w, \mathsf{S}) \leq 1/\sqrt{m}$.

Case II: $\exists y \neq y_i \, : \langle \phi(x_i, y), w \rangle \geq \langle \phi(x_i, y_i), w \rangle$. Let $\Delta_i(y) \stackrel{\text{def}}{=} \phi(x_i, y) - \phi(x_i, y_i)$. In this case,

$$\begin{aligned}
A_i(w, \mathsf{S}) &\stackrel{(a)}{\leq} \mathbb{E}_{\mathsf{T}_i} \left[\Pr_{\gamma} \left\{ f_{w,\gamma,\bar{\mathsf{T}}_i}(x_i) = y_i \right\} \right] \\
&= \mathbb{E}_{\mathsf{T}_i} \left[\frac{\exp(\langle \phi(x_i, y_i), w \rangle / \beta)}{Z(w, x_i, \bar{\mathsf{T}}_i)} \right] \\
&\stackrel{(b)}{=} \mathbb{E}_{\mathsf{T}_i} \left[\frac{1}{1 + \sum_{y \in \mathsf{T}_i} e^{\langle \Delta_i(y), w \rangle / \beta}} \right] \\
&\stackrel{(c)}{\leq} \mathbb{E}_{S_i} \left[\frac{1}{1 + n e^{S_i / \beta}} \right],
\end{aligned} \tag{C.3}$$

where we have defined $S_i \stackrel{\text{def}}{=} \frac{1}{n}\sum_{y\in\mathsf{T}_i}\langle\Delta_i(y), w\rangle$. In the above, in step (a) we dropped the term $\Pr_\gamma\{f_{w,\gamma}(x_i) = y_i\}$ to get an upper bound. Step (b) follows from dividing the numerator and denominator by $\exp(\langle\phi(x_i, y_i), w\rangle)$ and that $y_i \in \bar{\mathsf{T}}_i$. Step (c) follows from Jensen's inequality. Now,

$$\begin{aligned}
&\mathbb{E}_{S_i}\left[\frac{1}{1+ne^{S_i/\beta}}\right] \\
&= \mathbb{E}_{S_i}\left[\frac{1}{1+ne^{S_i/\beta}} \mid S_i \geq c\left\|w\right\|_1\right]\Pr\left\{S_i \geq c\left\|w\right\|_1\right\} \\
&\quad + \mathbb{E}_{S_i}\left[\frac{1}{1+ne^{S_i/\beta}} \mid S_i < c\left\|w\right\|_1\right]\Pr\left\{S_i < c\left\|w\right\|_1\right\} \\
&\stackrel{(a)}{\leq} \mathbb{E}_{S_i}\left[\frac{1}{1+ne^{S_i/\beta}} \mid S_i \geq c\left\|w\right\|_1\right] + \frac{\left\|w\right\|_1}{\sqrt{m}} \\
&\stackrel{(b)}{\leq} \mathbb{E}_{S_i}\left[\frac{1}{1+ne^{S_i\log m/\left\|w\right\|_1}} \mid S_i \geq c\left\|w\right\|_1\right] + \frac{\left\|w\right\|_1}{\sqrt{m}} \\
&= \mathbb{E}_{S_i}\left[\frac{1}{1+nm^{S_i/\left\|w\right\|_1}} \mid S_i \geq c\left\|w\right\|_1\right] + \frac{\left\|w\right\|_1}{\sqrt{m}} \\
&\leq \frac{1}{1+nm^c} + \frac{\left\|w\right\|_1}{\sqrt{m}}, \qquad \text{(C.4)}
\end{aligned}$$

where inequality (a) follows from Assumption 4.3.3 and (b) follows from the fact that $\beta \leq {\left\|w\right\|_1}/{\log m}$. Thus from (C.3) and (C.4) we have that $A_i(w, \mathsf{S}) \leq {1}/{(1+nm^c)} + {\left\|w\right\|_1}/{\sqrt{m}}$.

The final claim follows from Case I and II. ■

Proof [Proof of Lemma 4.3.5] We adapt the proof of Rademacher based uniform convergence for our purpose. Fix the distribution over T to $\mathcal{R}(\mathsf{S}, w')$ for some w'. Recall that $\bar{\mathsf{T}} = \{\bar{\mathsf{T}}_i\}$ with $\bar{\mathsf{T}}_i = \{y_i\} \cup \mathsf{T}_i$ and the elements of T_i are drawn i.i.d. from $\mathcal{R}(x_i, w')$. Since the only random part in $\bar{\mathsf{T}}_i$ is T_i and $y_i \in \mathsf{S}$, it suffices to show concentration of $\mathbb{E}_{\mathsf{T}}[L(w, \mathsf{S}, \mathsf{T})] - L(w, \mathsf{S}, \mathsf{T})$ for all w and S. For a fixed S, we will consider $L(w, \mathsf{S}, \mathsf{T})$ to be a function of T and w and denote it by $L(\mathsf{T}, w; \mathsf{S})$. In what follows, we will consider T to be an mn-dimensional vector whose elements (structured outputs) are conditionally independent (but not identically distributed) given a data set S. Define,

$$\varphi(\mathsf{T}; \mathsf{S}) \stackrel{\text{def}}{=} \sup_{w\in\mathbb{R}^{d,s}} \mathbb{E}_{\mathsf{T}\sim\mathcal{R}(\mathsf{S},w')}[L(\mathsf{T}, w; \mathsf{S})] - L(\mathsf{T}, w; \mathsf{S}). \qquad \text{(C.5)}$$

$\varphi(\mathsf{T};\mathsf{S})$ is $(1/m)$-Lipschitz and the elements of T are independent. Therefore, by McDiarmid's inequality, we have that:

$$\Pr_{\mathsf{T}}\left\{\mathbb{E}_{\mathsf{T}}\left[\varphi(\mathsf{T};\mathsf{S})\right]-\varphi(\mathsf{T};\mathsf{S})\leq\sqrt{\frac{\ln(1/\delta)}{2m}}\mid\mathsf{S}\right\}\geq 1-\delta. \tag{C.6}$$

Therefore, with probability at least $1-\delta$ over the choice of T:

$$\begin{aligned}&(\forall w\in\mathbb{R}^{d,s})\ \mathbb{E}_{\mathsf{T}}\left[L(\mathsf{T},w;\mathsf{S})\right]-L(\mathsf{T},w;\mathsf{S})\\&\leq\sup_{w\in\mathbb{R}^{d,s}}\mathbb{E}_{\mathsf{T}}\left[L(\mathsf{T},w;\mathsf{S})\right]-L(\mathsf{T},w;\mathsf{S})=\varphi(\mathsf{T};\mathsf{S})\\&\leq\mathbb{E}_{\mathsf{T}}\left[\varphi(\mathsf{T};\mathsf{S})\right]+\sqrt{\frac{\ln 1/\delta}{2m}}.\end{aligned} \tag{C.7}$$

Next, we will use a symmetrization argument to bound $\mathbb{E}_{\mathsf{T}}\left[\varphi(\mathsf{T};\mathsf{S})\right]$. Let $\mathsf{T}'\sim\mathcal{R}(\mathsf{S})$ be an independent copy of T. Observe that:

$$\begin{aligned}\mathbb{E}_{\mathsf{T}'}\left[L(\mathsf{T},w;\mathsf{S})\mid\mathsf{T}\right]&=L(\mathsf{T},w;\mathsf{S})\\\mathbb{E}_{\mathsf{T}'}\left[L(\mathsf{T}',w;\mathsf{S})\mid\mathsf{T}\right]&=\mathbb{E}_{\mathsf{T}}\left[L(\mathsf{T},w;\mathsf{S})\right].\end{aligned}$$

Now,

$$\begin{aligned}&\mathbb{E}_{\mathsf{T}}\left[\varphi(\mathsf{T})\right]\\&=\mathbb{E}_{\mathsf{T}}\left[\sup_{w\in\mathbb{R}^{d,s}}\mathbb{E}_{\mathsf{T}}\left[L(\mathsf{T},w;\mathsf{S})\right]-L(\mathsf{T},w;\mathsf{S})\right]\\&=\mathbb{E}_{\mathsf{T}}\left[\sup_{w\in\mathbb{R}^{d,s}}\mathbb{E}_{\mathsf{T}'}\left[L(\mathsf{T}',w;\mathsf{S})\mid\mathsf{T}\right]-\mathbb{E}_{\mathsf{T}'}\left[L(\mathsf{T},w;\mathsf{S})\mid\mathsf{T}\right]\right]\\&\leq\mathbb{E}_{\mathsf{T},\mathsf{T}'}\left[\sup_{w\in\mathbb{R}^{d,s}}\frac{1}{m}\sum_{i=1}^{m}z_i'-z_i\right],\end{aligned}$$

where $z_i'=\Pr_{\gamma}\left\{f_{w,\gamma,\mathsf{T}'}(x_i)\neq y_i\right\}$ and $z_i=\Pr_{\gamma}\left\{f_{w,\gamma,\mathsf{T}}(x_i)\neq y_i\right\}$. Since $z_i'-z_i$ has a distribution that is symmetric around zero, $z_i'-z_i$ and $\sigma_i(z_i'-z_i)$ have the same

distribution, where σ_i's are independent Rademacher variables. Continuing the above derivation,

$$\begin{aligned}
&\mathbb{E}_{\mathsf{T}}\left[\varphi(\mathsf{T})\right] \\
&\leq \mathbb{E}_{\mathsf{T},\mathsf{T}',\sigma}\left[\sup_{w\in\mathbb{R}^{d,s}} \frac{1}{m}\sum_{i=1}^{m}\sigma_i(z_i' - z_i)\right] \\
&= \frac{2}{m}\mathbb{E}_{\mathsf{T},\sigma}\left[\sup_{w\in\mathbb{R}^{d,s}} \sum_{i=1}^{m}\sigma_i \mathrm{Pr}_{\gamma}\left\{f_{w,\gamma,\mathsf{T}}(x_i)\neq y_i\right\}\right] \\
&= 2\mathbb{E}_{\mathsf{T}}\left[\widehat{\mathfrak{R}}_{\mathsf{T}}(\mathcal{G})\right],
\end{aligned}$$

where $\widehat{\mathfrak{R}}_{\mathsf{T}}(\mathcal{G})$ is the empirical Rademacher complexity of the function class $\mathcal{G} = \{g_w \mid w \in \mathbb{R}^{d,s}\}$ with respect to T, with $g_w(x,y) = \mathrm{Pr}_{\gamma}\left\{f_{w,\gamma,\mathsf{T}}(x)\neq y\right\}$. Next, using the same argument as in the proof of Theorem 4.2.1, we can bound $\widehat{\mathfrak{R}}_{\mathsf{T}}(\mathcal{G})$ for any set T, and get the following bound:

$$\mathbb{E}_{\mathsf{T}}\left[\varphi(\mathsf{T})\right] \leq 2\sqrt{\frac{s(\log d + 2\log(nr))}{m}} \tag{C.8}$$

Note that the above differs from the bound in Theorem 4.2.1 in the log factor since we need to consider linear orderings of nr structured outputs. Therefore from (C.7) and (C.8) we have that:

$$\begin{aligned}
\mathrm{Pr}_{\mathsf{T}}\{(\forall w\in\mathbb{R}^{d,s})\ \mathbb{E}_{\mathsf{T}}\left[L(\mathsf{T},w;\mathsf{S})\right] - L(\mathsf{T},w;\mathsf{S}) \\
\leq \varepsilon_2(d,s,n,r,m,\delta) \mid \mathsf{S}\} \geq 1-\delta.
\end{aligned} \tag{C.9}$$

By Definition 4.3.1 and from the results by [Ben56, BH60, Cov67], there are at most $\binom{d}{s}(mr)^{2s}$ effective (equivalence classes) proposal distributions $\mathcal{R}(.)$ Taking a union bound over all such proposal distributions we prove our claim. ■