Methods on Model Selection: Bayes Factor Approximation and False Discovery Rate Control

Methods on Model Selection: Bayes Factor Approximation and False Discovery Rate Control

A DISSERTATION PRESENTED
BY
CHENGUANG DAI
TO
DEPARTMENT OF STATISTICS

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
IN THE SUBJECT OF
STATISTICS

HARVARD UNIVERSITY
CAMBRIDGE, MASSACHUSETTS
APRIL 2020

Thesis Advisor: Professor Jun S. Liu Chenguang Dai

Methods on Model Selection: Bayes Factor Approximation and False Discovery Rate Control

Abstract

Model selection can be an art. In many scientific fields including genetics, climate sciences and social sciences, a proper model can simplify the computation in the analysis as well as increase the interpretability of the result. In this dissertation, we articulate two methods, under two different guiding principles, to facilitate model selection.

The first method concerns the computational challenge in Bayesian model comparison. We show that the Bayes factor can be approximated using the Wang-Landau algorithm, based on a mixture formulation between the posterior distribution and a user-defined surrogate distribution. The proposed Wang-Landau mixture method is applicable as long as an effective Markov kernel invariant to the posterior is available. Further refinements are carefully discussed, including accelerating the convergence using the momentum method, and facilitating global jumps between the posterior and the surrogate using the Multiple-try Metropolis.

The second method concerns a desired Frequentist property in feature selection. Specifically, we form a proper statistic via data splitting to rank the importance of each feature. The statistic enjoys a useful property, that is, it is symmetric about 0 for null features, and relatively large for relevant features. We show that by carefully choosing a data-dependent cutoff, we can achieve asymptotic false discovery rate control under proper conditions. The proposed method is free of calculating p-values, and is applicable to a wide class of statistical models including the linear model, the generalized linear model, and the Gaussian graphical model.

Contents

This thesis is dedicated to my loving parents.

Acknowledgments

My deepest gratitude goes first to my advisor Prof. Jun S. Liu, without whom my research achievements in the past couple of years would never have been possible. You set a high bar for being a motivated researcher and a thoughtful advisor. Words can barely express how grateful I am, for your insightful guidance on improving my work, your huge effort on polishing my writings, and your constant attention on helping me thrive. I cannot thank you enough for always being approachable.

I am extremely honored to have Prof. Xiao-Li Meng and Prof. Natesh S. Pillai in my thesis committee. I would like to thank you for your time and your valuable feedback. My sincere gratitude also goes to Prof. Pierre Jacob. Your passion and integrity as a researcher are truly inspiring.

I am grateful to all the faculty and administrative staff members of the Department of Statistics at Harvard University, for your endless effort in creating an incredibly inclusive culture.

I would like to thank my collaborators Chan Duo, Xin Xing and Buyu Lin. It has been a pleasant experience learning from all of you. My warmest thanks go to my wonderful peers and friends, for sticking by my side and supporting me in my hour of need.

Finally, I dedicate my greatest thanks to my parents, Hailin Dai and Huafang Qu, for your unconditional love and care. I have no words to acknowledge the support I received and the sacrifices you made ever since I was born.

0

Introduction

Model selection has been a well-focused problem in Statistics. Many methodological developments, from both the Frequentist and the Bayesian perspectives, have been documented to help practitioners select a suitable model among a pool of candidate models. We articulate two guiding principles for model selection in the following, along with a brief overview of our contributions.

0.1 Maximizing the Marginal Likelihood of Data

Given data $\boldsymbol{y}$, the marginal likelihood of data under model $\mathcal{M}_k$, sometimes called the normalizing constant, is defined as

$$p(\boldsymbol{y} \mid \mathcal{M}_k) = \int \gamma(\boldsymbol{\theta}_k \mid \boldsymbol{y}, \mathcal{M}_k) d\boldsymbol{\theta}_k = \int p(\boldsymbol{\theta}_k \mid \mathcal{M}_k) p(\boldsymbol{y} \mid \boldsymbol{\theta}_k, \mathcal{M}_k) d\boldsymbol{\theta}_k, \tag{1}$$

in which $\boldsymbol{\theta}_k$ denotes the associated parameters of model $\mathcal{M}_k$, $p(\boldsymbol{\theta}_k \mid \mathcal{M}_k)$ denotes the prior, $p(\boldsymbol{y} \mid \boldsymbol{\theta}_k, \mathcal{M}_k)$ denotes the model likelihood, and $\gamma(\boldsymbol{\theta}_k \mid \boldsymbol{y}, \mathcal{M}_k)$ denotes the unnormalized posterior distribution.

For a finite set of competing models $\{\mathcal{M}_k\}$, Bayesian methods typically compare two models $\mathcal{M}_i$ and $\mathcal{M}_j$ by computing the *Bayes factor* B_{ij} defined as below,

$$B_{ij} = \frac{p(\boldsymbol{y} \mid \mathcal{M}_i)}{p(\boldsymbol{y} \mid \mathcal{M}_j)}. \tag{2}$$

With the uniform prior on model, that is, $p(\mathcal{M}_k) \propto 1$, $B_{i,j} > 1$ indicates that model $\mathcal{M}_i$ is more favorable than model $\mathcal{M}_j$, in terms of the posterior odds, given the current data $\boldsymbol{y}$.

While model comparison via the Bayes factor is conceptually straightforward, it is also likely to be computationally challenging. There are two general strategies to exercise this principle. First, we can estimate all the normalizing constants in parallel. Second, we can include the model index $\mathcal{M}_k$ as a parameter in the joint posterior distribution specified as below,

$$p(\boldsymbol{\theta}_k, \mathcal{M}_k \mid \boldsymbol{y}) \propto p\left(\boldsymbol{y} \mid \boldsymbol{\theta}_k, \mathcal{M}_k\right) p\left(\boldsymbol{\theta}_k \mid \mathcal{M}_k\right) p\left(\mathcal{M}_k\right), \tag{3}$$

and use Markov chain Monte Carlo (MCMC) algorithms to traverse the joint model and parameter space. The ratio between the time that the Markov chain spends in different models, adjusted by the

model prior, consistently estimates the Bayes factor.

Neither analytical calculation nor numerical integration of the normalizing constant is feasible except for trivial models. Instead, various approximation schemes have been proposed in the literature. (i) The Bayesian information criterion (BIC). Konishi and Kitagawa (2008) derived BIC as an Laplace approximation of the marginal likelihood of data . (ii) The variational approximation. The normalizing constant is equivalent to the free energy in statistical physics up to a sign. Thus, variational free energies, including the Gibbs free energy and the Bethe free energy, can be exploited from statistical physics (Mezard and Montanari, 2009) to approximate the normalizing constant. (iii) Monte Carlo approximation. Based upon effective sampling from the posterior distribution using Markov chain Monte Carlo (MCMC) algorithms, many efforts have been made to construct accurate Monte Carlo estimators, including importance-sampling-based methods (Liu, 2008), Chib's method (Chib, 1995; Chib and Jeliazkov, 2001), the inverse logistic regression (Geyer, 1991). Monte Carlo methods will be the focus of Chapter 2, and a review of relevant methods is given in Section 2.5.

When it appears over-demanding to estimate the marginal likelihood of data for each model, sampling from the joint posterior distribution (3) serves as an attractive alternative. Different models potentially sit in different dimensional spaces, thus it requires to construct efficient trans-dimensional jumping mechanisms Brooks et al. (2003) to enable effective reversible jump MCMC (Green, 1995). In order to maintain a reasonably high acceptance probability when jumping across different models, we propose to employ the Multiple-try Metropolis (Liu et al., 2000) to sample along informative jumping directions, which can be identified by pre-locating the modes of within-model posteriors. Two jumping mechanisms, including the fixed-directional jump and the adaptive-directional jump, are detailed in Section 2.4.

Model selection via the Bayes factor can be misleading if vague (e.g. uniform priors on arbitrarily wide intervals) or improper priors are involved. Indeed, statisticians are debating on whether priors should be taken into account in model comparison (Kass and Raftery, 1995; Robert, 2007). Possible

solutions have been proposed in the literature, by properly modifying the Bayes factor to enhance its robustness. We refer the readers to O'Hagan (1995); Berger and Pericchi (1996); Berger et al. (1998, 2001).

0.2 Controlling the False Discovery Rate

In view of the explosive feature collection capabilities, model selection in the high-dimensional setting has been of recent interest. In the following, we focus on regression models, although the general idea is applicable in a broad class of multiple testing problems. Let n be the sample size. For each sample, suppose there is a set of p associated features $(X_1, \cdots, X_p)$ targeting some response variable y, with p potentially being much larger than n. We assume that only a small fraction of features directly interact with the response variable, which appears plausible in many real-world applications. Let $S^\star$ and S_0 be the index set corresponding to relevant features and null features, respectively.

One of the guiding principles to achieve controlled model selection is to maximize the selection power while controlling the false discovery rate (FDR) under a pre-specified threshold $q \in (0, 1)$. The selection power is defined as

$$\text{Power} = \frac{\#\{j : j \in S^\star, j \in \widehat{S}\}}{\#\{j : j \in S^\star\}}, \tag{4}$$

in which $\widehat{S}$ denotes the set of selected features. FDR is defined as

$$\text{FDR} = \mathbb{E}[\text{FDP}], \quad \text{FDP} = \frac{\#\{j : j \in S_0,\, j \in \widehat{S}\}}{\#\{j \in \widehat{S}\} \vee 1}, \tag{5}$$

where FDP refers to the false discovery proportion. The expectation is taken with respect to the randomness both in the data and in the selection procedure if it is not deterministic. FDR control is a desired property in practice, which ensures that the proportion of the false positives is, on average, be-

low q. More stringent control criteria including the family-wise error rate control and the directional FDR control have also been considered in the literature.

Perhaps the most prominent FDR control procedure is the Benjamin-Hochberg (BHq) procedure (Benjamini and Hochberg, 1995), which is applicable when individual p-value p_j is accessible for each feature X_j. More precisely, for a given FDR control level $q \in (0, 1)$, BHq first ranks all the p-values in an increasing order, and then find the largest $k \in [p]$ such that $p_{(k)} \leq kq/p$. The selected index set of features is $\{j : p_j \leq p_{(k)}\}$. (Benjamini and Hochberg, 1995) showed that BHq achieves an exact FDR control if all the p-pvalues are independent. An extension of BHq, the Benjamin-Yekutieli (BYq) procedure, is considered in Benjamini and Yekutieli (2001) to handle potential positive dependency on the test statistics associated with null features.

Regularization methods (Bühlmann and Van De Geer, 2011; Wainwright, 2019) are generally required to fit high-dimensional regression models. Thus, calculating individual p-value for each feature X_j can be highly non-trivial. Three possible solutions have been considered in the literature. (i) data splitting (Wasserman and Roeder, 2009). That is, we split the data into two halves, and screen out redundant features using the first half of data. The validity of the statistical inference carried out on a smaller model using the second half of data is guaranteed as long as all the relevant features are selected in the screening stage. (2) post-selection inference. Recent work (Taylor et al., 2014; Lee et al., 2016) showed that for the high-dimensional linear model, exact inference conditioning on the model selection result is possible for several popular selection procedures including Lasso (Tibshirani, 1996) and the stepwise regression. (3) debiased methods. For the linear model (Javanmard and Montanari, 2014; Zhang and Zhang, 2014) and the generalized linear models (Van de Geer et al., 2014), it is possible to obtain asymptotically valid p-values after we properly debiase the regularized estimators.

The knockoff filter is another class of FDR control procedures that have been developed and popularized recently (Barber and Candès, 2015; Candes et al., 2018). The idea is to pair a knockoff for each feature, and examine how significance the knockoff is compared to the original feature. The knockoff

filter is free of calculating p-values, and is applicable in arbitrary models with provable finite-sample FDR control. On the other hand, in the large-p-small-n settings, it requires the knowledge of the joint distribution of all features so as to construct knockoffs. When correlations among the features appear non-negligible, the power of the knockoff filter can be inferior. Intuitively, this is because there is not enough degree of freedom to construct a knockoff significantly different to the original feature, but maintaining the correlation structure with the rest of features.

Xin et al. (2019) recently proposed the Gaussian mirror method for FDR control on the linear regression model. The idea is to substitute each feature X_j in the linear model by a pair of perturbed features

$$X_j^+ = X_j + c_j Z_j \quad \text{and} \quad X_j^- = X_j - c_j Z_j, \tag{6}$$

in which Z_j follows $N(0,1)$, and c_j is a proper-chosen scalar so that the corresponding regression coefficient estimates $\widehat{\beta}_j^+$ and $\widehat{\beta}_j^-$ are independent. The Gaussian mirror method then introduces a mirror statistic M_j for each feature X_j in the form of

$$M_j = \left|\widehat{\beta}_j^+ + \widehat{\beta}_j^-\right| - \left|\widehat{\beta}_j^+ - \widehat{\beta}_j^-\right|. \tag{7}$$

The mirror statistic enjoys the symmetric property, that is, for $j \in S_0$, M_j is symmetric about 0. On the other hand, for $j \in S^\star$, M_j is relatively large so that relevant features can be differentiated with null features using the mirror statistics. It remains to select a data-dependent cutoff τ_q so that the selected index set $\{j, M_j > \tau_q\}$ has a desirable FDR control under level q. The key is to upper bound the number of false positives, in which we exploit the symmetric property of M_j for $j \in S_0$, that is, for any $t > 0$,

$$\#\{j \in S_0 : M_j > t\} \approx \#\{j \in S_0 : M_j < -t\} \leq \#\{j : M_j < -t\}.$$

The Gaussian mirror method achieves the asymptotic FDR control under the assumption that the mirror statistics for null features are not too correlated.

0.3 Outline of Dissertation

In Chapter 1, we investigate the Wang-Landau algorithm (Wang and Landau, 2001b) from the optimization perspective, which will be the building block of the proposed Bayes factor approximation method we present in Chapter 2. We show that the Wang-Landau algorithm can be formulated as a (stochastic) gradient descent algorithm minimizing a convex and smooth objective function, of which the gradient is estimated using MCMC simulations. Various acceleration schemes can be exploited to speed up the convergence of the Wang-Landau algorithm. In particular, we find that the Adaptive Moment Estimation (Adam) method (Kingma and Ba, 2014) is effective, and illustrate it on canonical physical model including the two-dimensional Ising model and the two-dimensional Potts model.

In Chapter 2, we present a new Monte Carlo method to estimate the marginal likelihood of data. The idea is to match the posterior distribution by a surrogate distribution with a known normalizing constant, and formulate a mixture distribution to combine the posterior and the surrogate. The mixing parameter is an easy function of the two normalizing constants, and we estimate it using the Wang-Landau algorithm. The proposed method is applicable as long as an effective Markov kernel invariant to the posterior distribution is available. Detailed discussion on further refinement of the method can be found in Section 2.3 and 2.4.

In Chapter 3, motivated by the Gaussian mirror method, we present a new FDR control procedure, which requires neither calculating p-values nor constructing knockoffs. Instead, for each feature X_j, we estimate two regression coefficients via data splitting, upon which we construct the mirror statistic. The mirror statistic associated with null features enjoy the symmetric property as long as the regression coefficient estimates for null features are symmetric about 0. To stabilize the selection result and

overcome the potential power loss resulting from data splitting, we propose a multiple data-splitting framework to aggregate the selection results obtained from repeated data splits. Detailed discussions on specializations of the proposed method for different statistical models can be found in Section 3.4.1 to 3.4.4.

1

The Wang-Landau Algorithm as Stochastic Optimization and Its Acceleration

Contribution This chapter is based on a published article from *Physical Review E* (Dai and Liu, 2020), jointly with Prof. Jun S. Liu.

1.1 Abstract

We show that the Wang-Landau algorithm can be formulated as a stochastic gradient descent algorithm minimizing a smooth and convex objective function, of which the gradient is estimated using Markov chain Monte Carlo iterations. The optimization formulation provides us another way to establish the convergence rate of the Wang-Landau algorithm, by exploiting the fact that almost surely, the density estimates (on the logarithmic scale) remain in a compact set, upon which the objective function is strongly convex. The optimization viewpoint motivates us to improve the efficiency of the Wang-Landau algorithm using popular tools including the momentum method and the adaptive learning rate method. We demonstrate the accelerated Wang-Landau algorithm on a two-dimensional Ising model and a two-dimensional ten-state Potts model.

1.2 Introduction

The Wang-Landau (WL) algorithm (Wang and Landau, 2001a,b; Landau et al., 2004) has been proven useful in solving a wide range of computational problems in statistical physics, including spin-glass models (Brown and Schulthess, 2005; Torbrügegge and Schnack, 2007; Alder et al., 2004; Snider and Clare, 2005; Okabe et al., 2002; Zhou et al., 2006; Wu and Machta, 2005; Malakis and Fytas, 2006; Hernández and Ceva, 2008; Fytas and Malakis, 2008; Tsai et al., 2007; Yamaguchi and Okabe, 2001), fluid phase equilibria (Mastny and de Pablo, 2005; Shell et al., 2002), polymers (Taylor et al., 2009; Strathmann et al., 2008), lattice gauge theory (Langfeld et al., 2012), protein folding (Rathore and de Pablo, 2002; Rathore et al., 2003, 2004), free energy profile (Calvo, 2002), and numerical integration (Tröster and Dellago, 2005; Li et al., 2007). Its successful applications in statistics have also been documented (Liang, 2005; Atchadé and Liu, 2010; Bornn et al., 2013). The WL algorithm directly targets the density of states (the number of all possible configurations for an energy level of a system), thus allowing us to calculate thermodynamic quantities over an arbitrary range of temperature within

a single run of the algorithm.

Much effort has been made to understand the dynamics of the WL algorithm, along with numerous proposed improvements, of which we highlight three here. (i) Optimizing the modification factor (flatness criterion) (Belardinelli and Pereyra, 2007; Zhou and Bhatt, 2005; Zhou and Su, 2008; Dayal et al., 2004). Belardinelli and Pereyra (Belardinelli and Pereyra, 2007) proposed that instead of reducing the modification factor exponentially, the log modification factor should be scaled down at the rate of $1/t$ in order to avoid the saturation in the error. (ii) Employing a parallelization scheme. Wang and Landau (Wang and Landau, 2001b) suggested that multiple random walkers working simultaneously on the same density of states can accelerate the convergence of the WL algorithm. The efficiency of the parallelization scheme can be further enhanced using the replica-exchange framework (Vogel et al., 2013). (iii) Incorporating efficient Monte Carlo trial moves (Wüst and Landau, 2009; Yamaguchi and Kawashima, 2002; Wu et al., 2005).

In this chapter, we consider the WL algorithm from an optimization perspective and formulate it as a first-order method. We derive the corresponding smooth and convex objective function, of which the gradient involves the unknown density of states. Wang and Landau (Wang and Landau, 2001b) used a random-walk based Metropolis algorithm Metropolis et al. (1953) to estimate the gradient. In general, any suitable Markov chain Monte Carlo (MCMC) strategies Liu (2008) can be employed for this purpose. Therefore, the WL algorithm is essentially a stochastic gradient descent algorithm.

The optimization viewpoint enables us to establish the convergence rate of the WL algorithm. Following (Fort et al., 2015) and using the standard stochastic approximation theory (Fort et al., 2011), we first show that the density estimates (on the logarithmic scale) almost surely stay in a compact set. Based on this, we exploit the strong convexity of the objective function, restricted on this compact set, to prove the convergence rate. We note that the gradient estimator output from the MCMC iterations is generally biased, thus a critical step is to show that the bias vanishes properly as $t \to \infty$.

The optimization framework also provides us with a new direction for improving the WL algo-

rithm. We explore one possible improvement, by combining the momentum method (Polyak, 1964) and the adaptive learning rate method (Duchi et al., 2011; Zeile, 2012). The general goal is to accelerate the transient phase (Darken and Moody, 1992) of the WL algorithm before it enters the fine local convergence regime. The effectiveness of the acceleration method is demonstrated on a two-dimensional Ising model and a two-dimensional ten-state Potts model, in which the learning in the transient phase is considerably demanding.

The rest of the paper is organized as follows. Section 1.3 discusses the optimization formulation of the WL algorithm, and establishes the convergence rate from an optimization perspective. Section 1.4 introduces possible strategies to accelerate the WL algorithm using optimization tools. Section 2.6 demonstrates the accelerated WL algorithm on two benchmark examples. Finally, Section 3.6 concludes with a few remarks.

1.3 An Optimization Formulation

Denote the space of all microscopic configurations as X. Suppose there are in total N energy levels, $E_1 < \cdots < E_N$, for the underlying physical model. For a microscopic configuration $x \in \mathsf{X}$, we use $E(x)$ to denote its energy. Let $\{g(E_n)\}_{n=1}^N$ be the normalized density of states, i.e.,

$$g(E_n) \propto \#\{x \in \mathsf{X}, E(x) = E_n\}, \qquad \sum_{n=1}^{N} g(E_n) = 1. \tag{1.1}$$

After initializing $g_0(E_n)$ as $1/N$, the WL algorithm iterates between the following two steps. (i) Propose a transition configuration and accept it with probability $\min\{1, g_t(E_i)/g_t(E_j)\}$, where E_i and E_j refer to the energy levels before and after this transition, respectively. This is essentially a step of

the Metropolis algorithm (Metropolis et al., 1953) with the corresponding stationary distribution:

$$\pi_t(x) \propto \sum_{n=1}^{N} \frac{1}{g_t(E_n)} 1\left(E(x) = E_n\right). \tag{1.2}$$

(ii) Update the density of states. If $E(x_{t+1}) = E_n$, multiply $g_t(E_n)$ by a modification factor $f_{t+1} > 1$. That is, $g_{t+1}(E_n) \leftarrow g_t(E_n) \times f_{t+1}$.

The modification factor f_t should be properly scaled down in order to guarantee the convergence of the algorithm. There is a rich literature on how to adapt f_t online, including the flat/minimum histogram criterion, and the $1/t$ rule (Belardinelli and Pereyra, 2007) with its various extensions (Jayasri et al., 2005; Poulain et al., 2006). Under a proper scaling rule, the magnitude of the modification factor f_t is informative of the estimation error (Zhou and Bhatt, 2005). Thus, a commonly used stopping criteria for the WL algorithm is that f_t is small enough (say, below $\exp(10^{-8})$).

In the following, we will work on the logarithmic scale of the density of states. Denote $u_n^{(t)} = \log(g_t(E_n))$ for $n \in [N]$, and let $\boldsymbol{u} = (u_1, \cdots, u_N)$. The density update in the WL algorithm can be rewritten as

$$u_n^{(t+1)} \leftarrow u_n^{(t)} + \eta_{t+1} 1(E(x_{t+1}) = E_n), \tag{1.3}$$

where $\eta_{t+1} = \log f_{t+1}$, which will be referred to as the learning rate henceforth. The intermediate target distribution $\pi_t(x)$ defined in Equation (1.2) can also be formulated in terms of $\boldsymbol{u}^{(t)}$. We define

$$\pi_{\boldsymbol{u}}(x) \propto \sum_{n=1}^{N} \exp(-u_n) 1\left(E(x) = E_n\right), \tag{1.4}$$

and denote $P_{\boldsymbol{u}}$ as a general transition kernel invariant to $\pi_{\boldsymbol{u}}(x)$. For notational convenience, we use $\pi_t(x)$ to refer to $\pi_{\boldsymbol{u}^{(t)}}(x)$, and use P_t to refer to the transition kernel invariant to $\pi_t(x)$. After each density update, we normalize $\boldsymbol{u}^{(t)}$ to sum to 0, i.e., $u_n^{(t)} \leftarrow u_n^{(t)} - \sum_{i=1}^{N} u_i^{(t)}/N$, so that $\boldsymbol{u}^{(t)}$ stays in a compact set (see Proposition 1.3.1). The WL algorithm can be slightly rephrased as in Algorithm 1.

Algorithm 1: The Wang-Landau algorithm.

1. Initialization. $u_n^{(0)} = 0$ for $n \in [N]$.
2. For $t \geq 1$, iterate between the following steps.
   (a) Sample x_{t+1} from $P_t(x_t, \cdot)$.
   (b) Update $\boldsymbol{u}^{(t+1)}$ following Equation (1.3).
   (c) Normalize $\boldsymbol{u}^{(t+1)}$ to sum to 0.
   (d) Scale down the learning rate η_t properly.
3. Stop when the learning rate η_t is smaller than a prescribed threshold.

Let us consider the following optimization problem:

$$\begin{aligned} &\min_{\boldsymbol{u}\in\mathbb{R}^N} h(\boldsymbol{u}) = \log\left(\sum_{n=1}^{N} \exp(u_n^\star - u_n)\right), \\ &\text{subject to } \sum_{n=1}^{N} u_n = 0, \end{aligned} \tag{1.5}$$

in which $u_n^\star = \log(g(E_n)) - \frac{1}{N}\sum_{i=1}^{N}\log(g(E_i))$. We write $\boldsymbol{u}^\star = (u_1^\star, \cdots, u_N^\star)$. It is not difficult to see that this is a convex optimization problem because the objective function $h(\boldsymbol{u})$ is a log-sum-exp function and the constraint is linear. It has a unique solution at $u_n = u_n^\star$ for $n \in [N]$, in which $\exp(u_n^\star)$ equals to the density of states $g(E_n)$ up to an multiplicative constant.

The projected gradient descent algorithm is a standard approach to solve the constrained optimization problem (2.9). The gradient of the objective function $h(\boldsymbol{u})$ is

$$\frac{\partial h(\boldsymbol{u})}{\partial u_n} = -\frac{\exp\left(u_n^\star - u_n\right)}{\sum_{i=1}^{N}\exp\left(u_i^\star - u_i\right)}, \quad n \in [N], \tag{1.6}$$

which is not directly available because it involves the unknown density of states. However, one can think of approximating the gradient function defined in Equation (1.6) by one-step or multiple-step Monte Carlo simulations, leading to a stochastic version of the projected gradient descent algorithm.

More precisely, a gradient descent step for minimizing $h(\boldsymbol{u})$ takes the following form:

$$u_n^{(t+1)} \leftarrow u_n^{(t)} + \frac{\eta_{t+1} \exp(u_n^\star - u_n^{(t)})}{\sum_{i=1}^N \exp(u_i^\star - u_i^{(t)})}. \tag{1.7}$$

Denote the probability of the set $\{x \in \mathsf{X} : E(x) = E_n\}$ with respect to $\pi_t(x)$ as $\pi_t(E_n)$. Since the probability $\pi_t(E_n)$ is proportional to $\exp(u_n^\star - u_n^{(t)})$, the density update in Equation (1.7) is essentially

$$u_n^{(t+1)} \leftarrow u_n^{(t)} + \eta_{t+1}\pi_t(E_n). \tag{1.8}$$

A crude approximation to $\pi_t(E_n)$ is the indicator function $1\left(E(x_{t+1}) = E_i\right)$, given that after several steps of Monte Carlo simulations according to the transition kernel P_t invariant to $\pi_t(x)$, x_{t+1} is approximately a sample from $\pi_t(x)$. This corresponds to the density update in Equation (1.3).

We note that the projection step to the set $\Pi = \{\boldsymbol{u} \in \mathbb{R}^N, \sum_{n=1}^N u_n = 0\}$ is equivalent to the normalization step (see Algorithm 1 step 2(c)). Thus, we have shown that the stochastic projected gradient descent algorithm solving the constrained optimization problem (2.9), which estimates the probability $\pi_t(E_n)$ by $1\left(E(x_{t+1}) = E_n\right)$ using the output from Monte Carlo simulations, is equivalent to the WL algorithm.

The above optimization formulation has the following immediate implications. First, the parallel WL algorithm estimates the negative gradient $\pi_t(E_n)$ by $1/m \sum_{k=1}^m [1(E(x_t^{(k)}) = E_n)]$, in which m denotes the total number of random walkers, and $x_t^{(k)}$ denotes the kth random walker. Therefore, it reduces the variance of the gradient estimate by a factor m. Second, instead of implementing a single transition step, the separation strategy mentioned in (Zhou and Bhatt, 2005) implements multiple transition steps within each iteration, so that the law of the random walker gets closer to the interme-

diate target distribution $\pi_t(x)$ defined in Equation (1.4). Therefore, it reduces the bias of the gradient estimate.

The optimization formulation also points out another approach to establish the convergence rate of the WL algorithm. We first state a required assumption, which assumes that the transition kernels are (uniformly) geometrically ergodic over the space Π.

Assumption 1.3.1. There exists a constant $\rho \in (0, 1)$ such that for all $\boldsymbol{u} \in \Pi$, $x \in \mathsf{X}$, $k \in \mathbb{N}$, we have

$$\sup_{\boldsymbol{u}\in\Pi} \sup_{x\in\mathsf{X}} ||P_{\boldsymbol{u}}^k(x, \cdot) - \pi_{\boldsymbol{u}}||_{\mathrm{TV}} \leq 2(1-\rho)^k, \tag{1.9}$$

in which for a signed measure μ, the total variation norm is defined as

$$||\mu||_{\mathrm{TV}} = \sup_{|q|\leq 1} \left| \int_{\mathsf{X}} q(x)\mu(dx) \right|. \tag{1.10}$$

We note that sufficient conditions for Assumption 1.3.1 exist in the literature (e.g., condition A2 in (Fort et al., 2015)), and relaxation of Assumption 1.3.1 is also possible (Fort et al., 2011). We have the following result.

Proposition 1.3.1. Under Assumption 1.3.1, if we scale down the learning rate η_t in the order of $O(1/t)$, the following two statements hold.

1. Almost-sure convergence.

   (a) There exists a compact set $\mathcal{K} \subseteq \Pi$ such that for any $t \geq 0$, $\boldsymbol{u}^{(t)} \in \mathcal{K}$ almost surely.

   (b) $\mathbb{P}(\lim_{t\to\infty} \boldsymbol{u}^{(t)} = \boldsymbol{u}^\star) = 1$.

2. Convergence rate. There exists a constant $C > 0$ such that

$$\mathbb{E}||\boldsymbol{u}^{(t)} - \boldsymbol{u}^\star||^2 \leq C/t. \tag{1.11}$$

The proof of Proposition 1.3.1 is given in the Appendix A.

The first part of Proposition 1.3.1 follows similarly as (Fort et al., 2015). The main idea is to rewrite the WL update, including the density update and the normalization step, as

$$\boldsymbol{u}^{(t+1)} \leftarrow \boldsymbol{u}^{(t)} + \eta_{t+1}\boldsymbol{r}(\boldsymbol{u}^{(t)}) + \eta_{t+1}(\boldsymbol{R}(x_{t+1}) - \boldsymbol{r}(\boldsymbol{u}^{(t)})),$$

in which $R_n(x) = 1(E(x) = E_n) - 1/N$, and $r(\boldsymbol{u})$ is the mean-field function defined as

$$\boldsymbol{r}(\boldsymbol{u}) = \int_{\mathsf{X}} \boldsymbol{R}(x)\pi_{\boldsymbol{u}}(x)dx = \frac{\exp(\boldsymbol{u}^\star - \boldsymbol{u})}{\sum_{n=1}^N \exp(u_n^\star - u_n)} - \frac{1}{N}.$$

The proof of the almost-sure convergence concludes by applying the standard stochastic approximation theory (Theorems 2.2 and 2.3 in (Andrieu et al., 2005)) after we establish the following two facts. (i) The remainder term $\eta_{t+1}(\boldsymbol{R}(x_{t+1}) - \boldsymbol{r}(\boldsymbol{u}^{(t)}))$ vanishes properly as $t \to \infty$. (ii) There exists a Lyapunov function $V(\boldsymbol{u})$ specified below,

$$V(\boldsymbol{u}) = \frac{1}{N}\sum_{n=1}^N \exp(u_n^\star - u_n) - 1, \tag{1.12}$$

with respect to the mean-field function $r(\boldsymbol{u})$, such that

$$\langle \nabla V(\boldsymbol{u}), \boldsymbol{r}(\boldsymbol{u})\rangle < 0, \ \forall\, \boldsymbol{u} \neq \boldsymbol{u}^\star \ \text{ and } \ \langle \nabla V(\boldsymbol{u}^\star), \boldsymbol{r}(\boldsymbol{u}^\star)\rangle = 0.$$

The second part of Proposition 1.3.1 is our main theoretical contribution. There are two essential ingredients in establishing the convergence rate. (i) Strong convexity. The objective function $h(\boldsymbol{u})$ is only convex but not strongly convex on $\mathbb{R}^N$. However, because $\boldsymbol{u}^{(t)}$ stays in a compact set $\mathcal{K} \subseteq \Pi$ almost surely (see Proposition 1.3.1, part 1(a)), we are able to establish the strong convexity of $h(\boldsymbol{u})$ restricted on this compact set $\mathcal{K}$.

Lemma 1.3.1. Under Assumption 1.3.1, there exists a constant $\ell > 0$ such that for any $t \geq 0$, almost surely, it holds

$$\langle \nabla h(\boldsymbol{u}^{(t)}), \boldsymbol{u}^{(t)} - \boldsymbol{u}^{\star} \rangle \geq \ell ||\boldsymbol{u}^{(t)} - \boldsymbol{u}^{\star}||^2. \tag{1.13}$$

(ii) Vanishing bias. Because x_{t+1} is only an approximate sample from the intermediate target distribution $\pi_t(x)$, the indicator $1\left(E(x_{t+1}) = E_n\right)$ is not an unbiased estimator to the negative gradient $\pi_t(E_n)$. The following Lemma C.1.5 shows that the bias of the gradient estimator vanishes properly, as fast as the learning rate, when $t \to \infty$.

Lemma 1.3.2. Under Assumption 1.3.1, there exists a constant $C > 0$ such that

$$\mathbb{E}||\pi_t - P_t(x_t, \cdot)||_{\mathrm{TV}} \leq C\eta_{t+1}. \tag{1.14}$$

The convergence rate of the WL algorithm has been established in different forms in the literature. Zhou and Bhatt (Zhou and Bhatt, 2005) show that the discrete probability distribution $\{\pi_t(E_n)\}_{n=1}^N$ will be attracted, in terms of the KL-divergence, to the vicinity of the uniform distribution ($\pi_\infty(E_n) = 1/N$) as $t \to \infty$. In addition, they show that the standard deviation of $\exp(u_n^\star - u_n^{(t)})$ roughly scales like $\sqrt{\log f_t}$ when the modification factor f_t is close to 1. Although we are looking at the L^2 error of $\boldsymbol{u}^{(t)}$, which is slightly different from the aforementioned standard deviation, their convergence rate is consistent with our result because $\sqrt{\log f_t} = \sqrt{\eta_t}$ is in the order of $O(1/\sqrt{t})$ if we scale down the learning rate η_t in the order of $O(1/t)$. It is also worthwhile to mention that a corresponding central limit theorem in the original density space is provided in (Fort et al., 2015).

1.4 Accelerating Wang-Landau Algorithm

The optimization formulation motivates us to further improve the WL algorithm using optimization tools (Ruder, 2016). Our goal is to accelerate the convergence in the transient phase. The transient

phase (Darken and Moody, 1992) generally refers to the initial stage of running a stochastic gradient descent algorithm. For instance, if we scale down the learning rate according to the flat/minimum histogram criterion, we can refer to the transient phase as the running period from the beginning up to the time when the flat/minimum histogram criterion is first satisfied.

When the transient phase appears noticeable, the acceleration tools can be very effective in practice, and have been widely used in large-scale systems such as deep neural networks (Sutskever et al., 2013). In this chapter, we restrict ourselves on the first-order acceleration methods, and leave other possibilities for future explorations. In particular, we find that both the momentum method and the adaptive learning rate method are effective in accelerating the WL algorithm. Before we go into details, we note that improvement in the asymptotic convergence rate of the stochastic gradient descent algorithm is hard to achieve (or even impossible) (Nemirovski et al., 2009; Jain et al., 2017), except for some well-structured objective functions such as finite sums.

The momentum method exponentially accumulates a momentum vector, denoted as $\boldsymbol{m}_t$ in the following, to amplify the persistent gradient across iterations. The basic momentum update operates as follows:

$$\begin{aligned} \boldsymbol{m}^{(t)} &\leftarrow \beta \boldsymbol{m}^{(t-1)} + \eta_{t+1} \nabla h(\boldsymbol{u}^{(t)}), \\ \boldsymbol{u}^{(t+1)} &\leftarrow \boldsymbol{u}^{(t)} - \boldsymbol{m}^{(t)}, \end{aligned} \tag{1.15}$$

where we initialize the momentum vector to be $\boldsymbol{m}^{(0)} = 0$. We note that the momentum update essentially adds a fraction β of the previously accumulated gradients $\boldsymbol{m}^{(t-1)}$ into the current update vector $\boldsymbol{m}^{(t)}$. The weighting factor β is a tuning parameter and is commonly set to be 0.9 or higher.

In the setting of the WL algorithm, the momentum update in Equation (1.15) becomes

$$\begin{aligned} m_n^{(t)} &\leftarrow \beta m_n^{(t-1)} - \eta_{t+1} 1(E(x_{t+1}) = E_n), \\ u_n^{(t+1)} &\leftarrow u_n^{(t)} - m_n^{(t)}, \quad \forall n \in [N]. \end{aligned} \tag{1.16}$$

The intuition behind the momentum acceleration for the WL algorithm can be heuristically described as follows. The event $E(x_{t+1}) = E_n$ suggests that $\pi_t(E_n)$ is likely larger than $1/N$, and thus the Markov kernel P_t has a better chance to transit the microscopic configuration x_t into the energy level E_n. Therefore, in order to push $\pi_t(E_n)$ toward $1/N$, that is, downweight the probability mass in the energy level E_n, we increase $u_n^{(t)}$ by η_{t+1}, which corresponds to the density update in Equation (1.3). In contrast to the WL algorithm, which only increases $u_n^{(t)}$ by η_{t+1} at the current iteration t, we keep increasing $u_n^{(t)}$ for a few more iterations by an exponentially decay momentum $m_n^{(t)}$ to achieve a faster convergence.

The adaptive learning rate method helps standardize the gradient across different coordinates of the parameter $\boldsymbol{u}$, so that they scale in a similar magnitude. Otherwise, it can be challenging to find a suitable global learning rate η_t over different coordinates. Popular algorithms along this research direction include AdaGrad (Duchi et al., 2011), AdaDelta (Zeile, 2012), and RMSprop (an unpublished method proposed by Geoffrey Hinton). The RMSprop update operates as follows:

$$\begin{aligned} \boldsymbol{G}^{(t)} &\leftarrow \gamma \boldsymbol{G}^{(t-1)} + (1-\gamma)\nabla h(\boldsymbol{u}^{(t)})^2, \\ \boldsymbol{u}^{(t+1)} &\leftarrow \boldsymbol{u}^{(t)} - \eta_{t+1}[\boldsymbol{G}^{(t)}]^{-1/2}\nabla h(\boldsymbol{u}^{(t)}), \end{aligned} \tag{1.17}$$

in which both the square and the square root are taken elementwise. $\boldsymbol{G}^{(t)}$ represents the moving average of the squared gradients, so that the current gradient $\nabla h(\boldsymbol{u}^{(t)})$, standardized by $[\boldsymbol{G}^{(t)}]^{1/2}$, is in a similar magnitude across different coordinates. The weighting factor γ is a tuning parameter, which is commonly set to be 0.9 in order to prevent the updates from diminishing too fast. In the setting of the WL algorithm, the RMSprop update in Equation (1.17) becomes

$$\begin{aligned} G_n^{(t)} &\leftarrow \gamma G_n^{(t-1)} + (1-\gamma)1(E(x_{t+1}) = E_n), \\ u_n^{(t+1)} &\leftarrow u_n^{(t)} - \eta_{t+1}[G_n^{(t)}]^{-1/2}1(E(x_{t+1}) = E_n). \end{aligned} \tag{1.18}$$

The combination of the momentum method and the adaptive learning rate method leads to the Adaptive Moment Estimation (Adam) method (Kingma and Ba, 2014). The Adam update operates as follows:

$$\begin{aligned} \boldsymbol{m}^{(t)} &\leftarrow \beta \boldsymbol{m}^{(t-1)} + (1-\beta)\nabla h(\boldsymbol{u}^{(t)}), \\ \boldsymbol{G}^{(t)} &\leftarrow \gamma \boldsymbol{G}^{(t-1)} + (1-\gamma)\nabla h(\boldsymbol{u}^{(t)})^2, \\ \boldsymbol{u}^{(t+1)} &\leftarrow \boldsymbol{u}^{(t)} - \eta_{t+1}[\boldsymbol{G}^{(t)}]^{-1/2}\boldsymbol{m}^{(t)}. \end{aligned} \tag{1.19}$$

In the setting of the WL algorithm, we note that, although β and γ can be potentially two tuning parameters, if we set $\beta = \gamma$ and initialize $\boldsymbol{m}^{(0)}$ and $\boldsymbol{G}^{(0)}$ to be 0, we have $\boldsymbol{G}^{(t)} = -\boldsymbol{m}^{(t)}$, since $-\nabla h(\boldsymbol{u}^{(t)})$ is approximated by a one-hot vector, which contains only a single "1" with the remaining elements being 0. This simplification leads to Algorithm 2, which we refer to as the AWL algorithm henceforth.

Algorithm 2: Accelerated Wang-Landau algorithm.

---

1. Initialization. $u_n^{(0)} = 0$, $m_n^{(0)} = 0$ for $n \in [N]$.
2. For $t \geq 1$, iterate between the following steps.
   (a) Sample x_{t+1} from $P_t(x_t, \cdot)$.
   (b) Update $\boldsymbol{m}^{(t)}$ and $\boldsymbol{u}^{(t+1)}$ as follows.

$$\begin{aligned} m_n^{(t)} &\leftarrow \beta m_n^{(t-1)} + (1-\beta)1(E(x_{t+1}) = E_n), \\ u_n^{(t+1)} &\leftarrow u_n^{(t)} + \eta_{t+1}[m_n^{(t)}]^{1/2}. \end{aligned} \tag{1.20}$$

   (c) Normalize $\boldsymbol{u}^{(t+1)}$ to sum to 0.
   (d) Scale down the learning rate η_t properly.
3. Stop when the learning rate η_t is smaller than a prescribed threshold.

We remark that for large-scale systems, a naive implementation of Equation (1.20) can be very inefficient, as we have to loop over every coordinate of $\boldsymbol{m}^{(t)}$ and $\boldsymbol{u}^{(t)}$ in each iteration. A simple solution

is to introduce a vector $\boldsymbol{s} = (s_1, \cdots, s_N)$, in which s_n records the last time when m_n and u_n are updated. With the help of s_n, instead of updating m_n and u_n in each iteration, we shall update them only when the energy level E_n is involved in the Monte Carlo simulations.

1.5 Illustrations

We compare the AWL algorithm with the original WL algorithm on two benchmark examples: (i) a nearest-neighbour Ising model; (ii) a nearest-neighbour ten-state Potts model. Both models are defined on a two-dimensional $L \times L$ square lattice equipped with the periodic boundary condition.

For the Ising model, the energy $E(x)$ is given by the Hamiltonian:

$$E(x) = -\sum_{<i,j>} J_{ij} x_i x_j - \psi \sum_j b_j x_j, \tag{1.21}$$

where $x_i \in \{\pm 1\}$. The subscripts i, j denote the lattice sites, and the notation $< i, j >$ implies that the site i and the site j are nearest neighbors. For the ten-state Potts model, the energy $E(x)$ is given by:

$$E(x) = -\sum_{<i,j>} J_{ij} 1(x_i = x_j) - \psi \sum_j b_j x_j, \tag{1.22}$$

where $x_i \in \{1, \cdots, 10\}$. For both models, we assume that $J_{ij} \equiv 1$ and $b_j \equiv 0$ (no external magnetic field). If $b_j \equiv 0$, the two-dimensional Ising model exhibits a second-order phase transition. Otherwise, in the presence of an external magnetic field, the two-dimensional Ising model exhibits a first-order phase transition. When $b_j \equiv 0$, the two-dimensional Potts model exhibits a first-order phase transition when the number of states is larger than 4.

Let $\{H_t(E_n)\}_{n=1}^N$ be the histogram of all energy levels at iteration t. We initialize $H_0(E_n) = 0$ for $n \in [N]$. At each iteration t, the AWL algorithm and the WL algorithm update $\boldsymbol{u}^{(t)}$ according to Algorithms 2 and 1, respectively. In addition, we update the energy histogram as $H_t(E_n) =$

$H_{t-1}(E_n) + 1(E(x_{t+1}) = E_n)$.

The adaptation of the learning rate η_t follows (Belardinelli and Pereyra, 2007), which is detailed in the following.

1. After every 1,000 MC sweeps, we check $\{H_t(E_n)\}$. If $\min_n H_t(E_n) > 0$, we set $\eta_{t+1} = \eta_t/2$, and reset $H_t(E_n) = 0$ for each energy level E_n. Otherwise if $\min_n H_t(E_n) = 0$, we keep $\eta_{t+1} = \eta_t$.

2. If $\eta_{t+1} \leq N/t$, then $\eta_t = N/t$ for all the subsequent iterations. $H_t(E_n)$ is discarded and the above step is not executed any more.

We note that each MC sweep contains L^2 iterations, in which each iteration refers to a single round of parameter update. That is, steps 2(a)–2(c) in Algorithms 1 and 2. The energy histogram $\{H_t(E_n)\}$ essentially represents the number of visits to each energy level up to iteration t, since the last update of the learning rate.

We implement one step of the Metropolis algorithm to estimate the gradient, i.e., step 2(a) in Algorithms 1 and 2. The proposal schemes for the Ising model and the Potts model are described as follows. Given the current configuration x_t, we randomly pick up a site and change its value. For the Ising model, we flip its sign. For the ten-state Potts model, we set it to be a number uniformly sampled from $\{1, \cdots, 10\}$.

To illustrate the efficiency of the AWL algorithm, we investigate the following four perspectives. (i) The scaling of the first equilibration time, in terms of the number of MC sweeps, with respect to the dimension L. The first equilibration time, which corresponds to the transient phase as we discussed in Section 1.4, is defined to be $\min\{t : \min_n H_t(E_n)\} > 0$. That is, the first time when the energy histogram becomes nonzero everywhere. According to the adaptation rule of the learning rate η_t, the equilibration time is also the first time we decrease the learning rate. (ii) The scaling of the first equilibration time, in terms of the CPU time, with respect to the dimension L. Because the

AWL algorithm requires additional computations in updating the momentum vector, the comparison between the two algorithms on the actual CPU time is necessary to see whether the implementation of the acceleration method is indeed worthwhile. (iii) The dynamics of the estimation error $\epsilon(t)$ defined as below following (Belardinelli and Pereyra, 2007) for $L = 80$,

$$\epsilon(t) = \frac{1}{N-1} \sum_{n=1}^{N} \left| 1 - \frac{\log(g_t(E_n))}{\log(g(E_n))} \right|. \tag{1.23}$$

For the Ising model, the exact density of states $g(E_n)$ is available, and can be calculated using a publicly available Mathematica program (Beale, 1996). For the Potts model, no exact solution of $g(E_n)$ is available, and thus we pre-run a $1/t$ WL simulation for 5×10^7 MC sweeps, in which the final learning rate is 2×10^{-8}. We then treat the density estimates as an approximation to the exact density of states. (iv) The accuracy in the task of estimating the specific heat for the Ising model with $L = 80$.

We compare the AWL algorithm and the WL algorithm with different initializations of the learning rate, $\eta_0 = 0.05$, 0.10 and 1.00. We test out the two algorithms for different sizes of the two-dimensional square lattice, $L = 50, 60, 70, 80, 90, 100$. The computations were run on the FASRC Cannon cluster supported by the FAS Division of Science Research Computing Group at Harvard University.

Figure 1.1 summarizes the computational overheads of the two algorithms on the Ising model. The reported results are based on 50 independent runs of both algorithms, in which the dot represents the empirical mean and the error bar represents the empirical standard deviation. We see that the AWL algorithm takes significantly fewer MC sweeps as well as less CPU time to reach the first equilibration among all settings with different lattice sizes and different initializations of the learning rate. Figure 1.2 summarizes the computational overheads of the two algorithms on the Potts model. Similar to the case of the Ising model, the AWL algorithm is more efficient than the WL algorithm in terms of the first equilibration time measured by the number of MC sweeps and the CPU time.

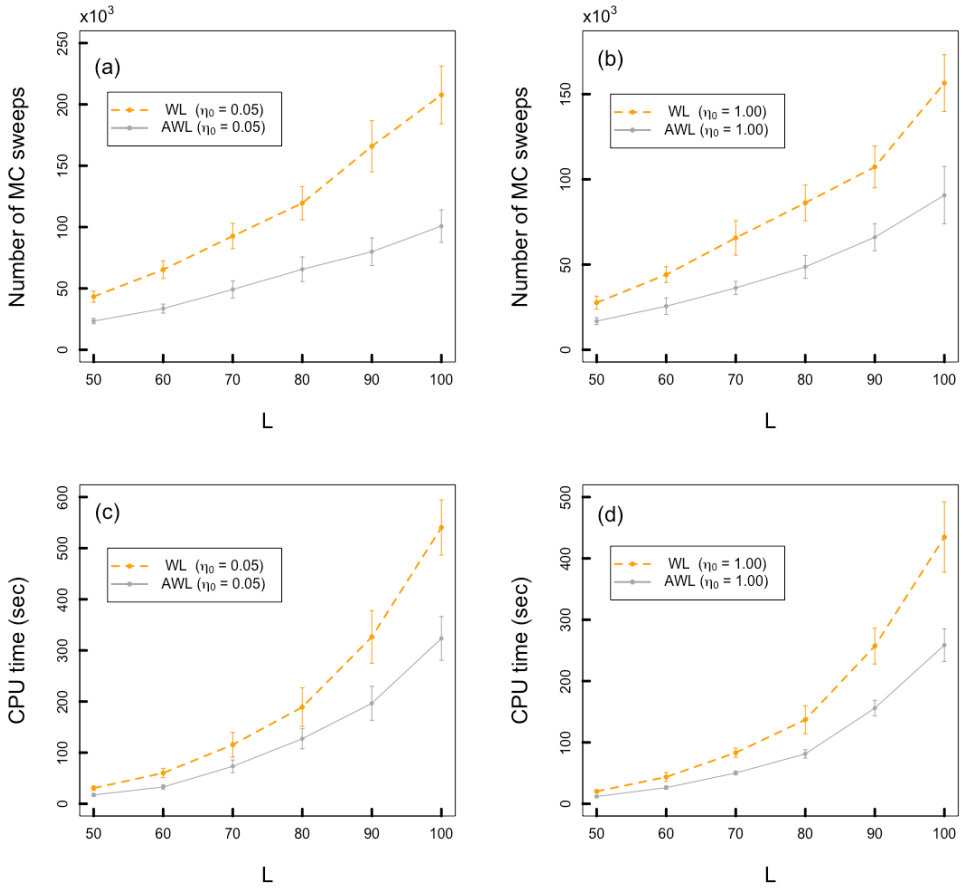


Figure 1.1: The computational overheads, in terms of the number of MC sweeps and the CPU time, that the AWL algorithm and the WL algorithm take to reach the first equilibration on the Ising model. Two initializations of the learning rate are tested out, including $\eta_0 = 0.05$ and $\eta_0 = 1.00$. The reported results are based on 50 independent runs of both algorithms. The dot represents the empirical mean and the error bar represents the empirical standard deviation.

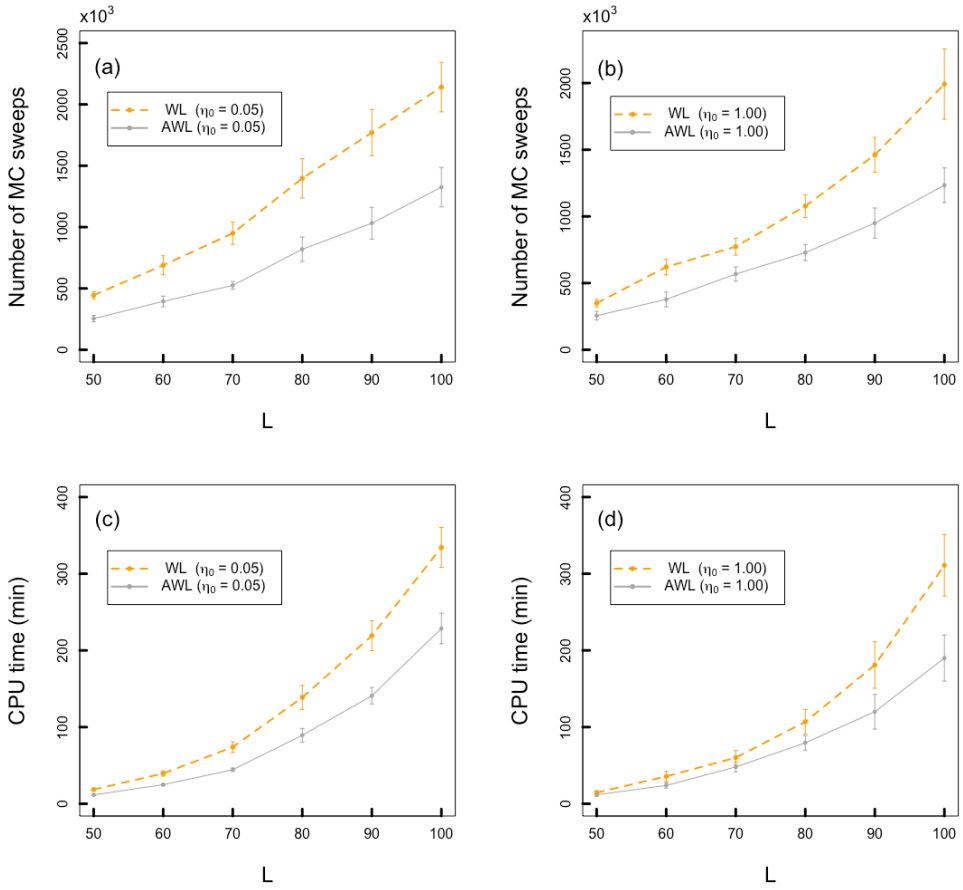


Figure 1.2: The computational overheads, in terms of the number of MC sweeps and the CPU time, that the AWL algorithm and the WL algorithm take to reach the first equilibration on the Potts model. Two initializations of the learning rate are tested out, including $\eta_0 = 0.05$ and $\eta_0 = 1.00$. The reported results are based on 50 independent runs of both algorithms. The dot represents the empirical mean and the error bar represents the empirical standard deviation.

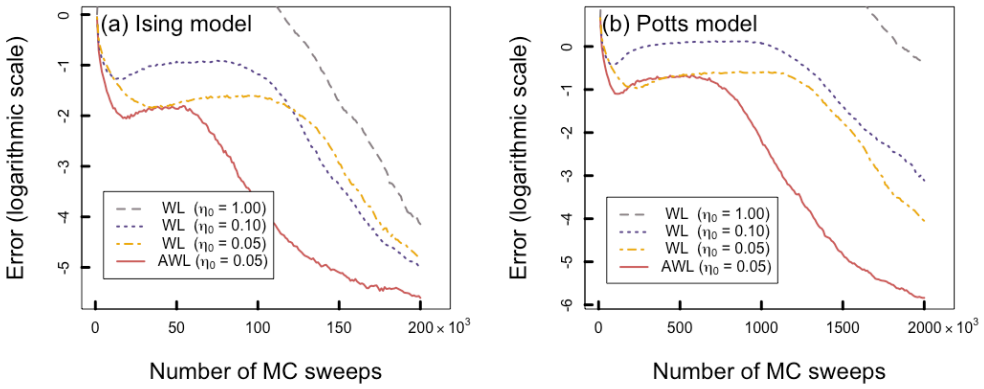


Figure 1.3: The dynamics of the estimation error $\epsilon(t)$ (on the logarithmic scale), averaging over 50 independent runs, of the AWL algorithm and the WL algorithm. Panel (a) shows the result for the Ising model, and panel (b) shows the result for the Potts model. η_0 denotes the initialization of the learning rate.

Figure 1.3 shows the empirical dynamics of $\epsilon(t)$, averaged over 50 independent runs of both algorithms. The first 100×10^3 MC sweeps for the Ising model and the first 1500×10^3 MC sweeps for the Potts model are representative for the transient phase. We see that in the transient phase, the convergence speed of the AWL algorithm, in terms of the number of MC sweeps, is significantly faster than the convergence speed of the WL algorithm with different initializations of the learning rate.

For the Ising model with $L = 80$, Table 1.1 compares the accuracy of the two algorithms in the calculation of the specific heat defined as

$$C(T) = \frac{\langle E^2 \rangle_T - \langle E \rangle_T^2}{T^2}, \tag{1.24}$$

in which T denotes the temperature. We test out temperatures ranging from 0.4 to 8 incremented by 0.1. The internal energy $\langle E \rangle_T$ is defined as

$$\langle E \rangle_T = \frac{\sum_n E_n g(E_n) \exp(-E_n/T)}{\sum_n g(E_n) \exp(-E_n/T)}. \tag{1.25}$$

	100×10^3 MC sweeps			150×10^3 MC sweeps			200×10^3 MC sweeps		
Quantiles	25%	50%	75%	25%	50%	75%	25%	50%	75%
AWL ($\eta_0 = 0.05$)	2.9%	6.3%	17.7%	0.9%	2.0%	4.6%	0.5%	1.2%	2.9%
WL ($\eta_0 = 0.05$)	10.5%	18.9%	41.4%	4.6%	9.1%	17.7%	1.1%	2.0%	4.4%
WL ($\eta_0 = 0.10$)	12.2%	24.0%	44.0%	2.4%	4.6%	10.9%	0.7%	2.4%	5.1%
WL ($\eta_0 = 1.00$)	47.1%	57.6%	74.4%	8.0%	16.0%	27.5%	2.8%	4.6%	8.4%

Table 1.1: The relative errors of the AWL algorithm and the WL algorithm in the calculation of the specific heat for the Ising model with $L = 80$. The relative errors are calculated based on the mean of 50 independent estimates produced by each algorithm. The quantiles of the relative errors are over the temperature interval $T \in [0.4, 8]$. η_0 denotes the initialization of the learning rate.

The fluctuation expression $\langle E^2 \rangle_T$ is defined similarly. We note that the theoretical value of the specific heat at a given temperature T can be evaluated exactly when the exact density of states is available, which is the case for the two-dimensional Ising model. We independently run each algorithm 50 times to obtain 50 independent estimates of the specific heat at each temperature. The relative error at each temperature is calculated based on the mean of the 50 independent estimates. Table 1.1 summarizes the quantiles of the relative errors for $T \in [0.4, 8]$, by running each algorithm for 100×10^3, 150×10^3, and 200×10^3 MC sweeps, respectively. Compared to the WL algorithm, the AWL algorithm yields significantly more accurate estimates of the specific heat especially in the transient phase.

More details of this numerical study can be found in the Appendix A. First, within the first 2×10^5 MC sweeps and 2×10^6 MC sweeps for the Ising model and the Potts model, respectively, we report the number of equilibrations that the AWL algorithm and the WL algorithm have reached (equivalently, the number of changes of the learning rate η_t), for different lattice sizes L and different initializations of the learning rate η_0. We also report the corresponding first eight equilibration time in terms of the number of MC sweeps (see Table A.1 and A.2). Second, for the Ising model with $L = 80$, we provide a graphical comparison of the estimated specific heat obtained by the AWL algorithm and the WL algorithm, over the temperature region $T \in [0.4, 8]$ (see Figure A.1).

1.6 Concluding Remarks

To summarize, in this chapter we present a new interpretation of the WL algorithm from the optimization perspective. We show that the WL algorithm is essentially a stochastic (projected) gradient descent algorithm minimizing a smooth and convex function, in which MCMC steps are used to estimate the unknown gradient. The optimization formulation intuitively explains that because of using more accurate gradient estimates, some notable modifications of the algorithm, such as utilizing multiple random walkers, can improve the WL algorithm. In addition, using the (strong) convexity of the objective function, we provide a new approach to establish the convergence rate of the WL algorithm, which is more explicit compared to the existing results (Fort et al., 2015; Zhou and Bhatt, 2005). We expect that our contributions are useful for further theoretical investigations of the WL algorithm.

The optimization interpretation also opens a new way to improve the efficiency of the WL algorithm. There are rich tools in the optimization literature to accelerate the stochastic gradient descent algorithm, including but not restricted to the methods we mentioned in Section 1.4. Different methods can be favorable for different applications. In the presence of noisy gradients, it usually requires some careful tuning to successfully apply the acceleration tools. We demonstrate one possible acceleration approach, using the momentum method and the adaptive learning rate strategy, on a two-dimensional Ising model and a two-dimensional ten-state Potts model.

2

Monte Carlo Approximation of Bayes Factors via Mixing with Surrogate Distributions

Contribution This chapter is based on a paper (Dai and Liu, 2019) jointly with Prof. Jun S. Liu.

2.1 Abstract

By mixing the posterior distribution with a surrogate distribution, of which the normalizing constant is tractable, we propose a new method for normalizing constant estimation using the Wang-Landau algorithm. We then show that faster convergence of the proposed method is achievable based on momentum acceleration. Two implementation suggestions are detailed, including (i) facilitating global jumps between the posterior and the surrogate via the Multiple-try Metropolis; (ii) constructing the surrogate via the variational approximation. When a surrogate is difficult to come by, we describe a new jumping mechanism for general reversible jump Markov chain Monte Carlo algorithms that combines the Multiple-try Metropolis and the directional sampling algorithm. We illustrate the proposed methods on several statistical models, including the Log-Gaussian Cox process, the Bayesian Lasso, the logistic regression, and the g-prior Bayesian variable selection.

2.2 Introduction

Given data $\boldsymbol{y}$, we consider a finite sequence of competing models $\{\mathcal{M}_k\}$ associated with parameters $\{\boldsymbol{\theta}_k\}$. The marginal likelihood of data under model $\mathcal{M}_k$, also referred to as the normalizing constant, is defined as

$$p(\boldsymbol{y} \mid \mathcal{M}_k) = \int \gamma(\boldsymbol{\theta}_k \mid \boldsymbol{y}, \mathcal{M}_k) d\boldsymbol{\theta}_k = \int p(\boldsymbol{\theta}_k \mid \mathcal{M}_k) p(\boldsymbol{y} \mid \boldsymbol{\theta}_k, \mathcal{M}_k) d\boldsymbol{\theta}_k,$$

in which $p(\boldsymbol{\theta}_k \mid \mathcal{M}_k)$ is the prior, $p(\boldsymbol{y} \mid \boldsymbol{\theta}_k, \mathcal{M}_k)$ is the likelihood, and $\gamma(\boldsymbol{\theta}_k \mid \boldsymbol{y}, \mathcal{M}_k)$ is the unnormalized posterior distribution. To compare different models, Bayesian methods typically compute the Bayes factor, which is defined as the ratio of the normalizing constants under different models, that is, $B_{i,j} = p(\boldsymbol{y} \mid \mathcal{M}_i)/p(\boldsymbol{y} \mid \mathcal{M}_j)$. With the uniform prior on model $\mathcal{M}_i$ and $\mathcal{M}_j$, $B_{i,j} > 1$ indicates that model $\mathcal{M}_i$ is more favorable than model $\mathcal{M}_j$ given the current data $\boldsymbol{y}$.

We can approximate the Bayes factor by estimating the normalizing constant of each model. For simplicity, we will drop the dependency on $\boldsymbol{y}$ and the model index k in $\gamma(\boldsymbol{\theta}_k \mid \boldsymbol{y}, \mathcal{M}_k)$ when the context is clear, and use Z_γ to denote the normalizing constant of $\gamma(\boldsymbol{\theta})$. Let $\gamma^\star(\boldsymbol{\theta}) = \gamma(\boldsymbol{\theta})/Z_\gamma$ be the corresponding normalized distribution. Computing Z_γ is essentially a task of calculating an integral. However, in many interesting cases, the complex form of the unnormalized density $\gamma(\boldsymbol{\theta})$, sometimes with high dimensionality, prohibits us from obtaining neither analytic solutions nor easy numerical approximations. Various Monte Carlo strategies have been developed to tackle this problem, such as Chib's method (Chib, 1995), inverse logistic regression (Geyer, 1994), importance sampling (Gelfand and Smith, 1990), bridge sampling (Meng and Schilling, 1996; Meng and Wong, 1996), path sampling (Ogata, 1989), sequential importance sampling (Hammersley and Morton, 1954; Rosenbluth and Rosenbluth, 1955; Kong et al., 1994), and sequential Monte Carlo (SMC) (Liu and Chen, 1998; Doucet et al., 2000; Del Moral et al., 2006). A nice overview of nineteen methods for normalizing constant estimation in the context of Bayesian phylogenetics is given in Fourment et al. (2020).

In this chapter, we present a mixture approach for normalizing constant estimation using the Wang-Landau (WL) algorithm (Wang and Landau, 2001b). Our idea is to construct a matching surrogate distribution $q(\boldsymbol{\theta})$ with its normalizing constant Z_q known, and combine $\gamma(\boldsymbol{\theta})$ and $q(\boldsymbol{\theta})$ to form a mixture distribution with an adjustable mixing parameter tuned through the WL algorithm. The ratio $r = Z_\gamma/Z_q$ is then an easy function of the mixing parameter. Let $q^\star(\boldsymbol{\theta})$ be the normalized surrogate distribution. Many of the aforementioned methods also use a surrogate, and the idea of using the WL algorithm to estimate the ratio $r = Z_\gamma/Z_q$ also appears in Liang (2005) and Atchadé and Liu (2010) in more restricted settings.

The proposed WL mixture method is different from existing methods in the following perspectives. First, when we apply the WL algorithm in our setting, there is a natural partition of the parameter space indicated by the two (or more if needed) mixture components. Second, unlike the method in Liang (2005), we do not require $\gamma^\star(\boldsymbol{\theta})$ and $q^\star(\boldsymbol{\theta})$ being well-separated. In fact, jumps between the

posterior and the surrogate becomes more flexible if $\gamma^\star(\boldsymbol{\theta})$ and $q^\star(\boldsymbol{\theta})$ are mixed together. Third, the WL mixture method does not require $\gamma^\star(\boldsymbol{\theta})$ and $q^\star(\boldsymbol{\theta})$ to have any overlap. With the help of mode jumping algorithms such as the Multiple-try Metropolis (MTM) (Liu et al., 2000), it appears much more robust than importance sampling based methods such as bridge sampling, which crucially rely on the amount of overlaps between $\gamma^\star(\boldsymbol{\theta})$ and $q^\star(\boldsymbol{\theta})$.

Following Dai and Liu (2020), we can achieve faster convergence of the proposed method using momentum acceleration. The idea is to formulate the WL algorithm as a (stochastic) gradient descent algorithm minimizing a convex and smooth function, of which the gradient is estimated using Markov chain Monte Carlo (MCMC) iterations. Under this optimization framework, we are able to exploit acceleration tools to speed up the convergence of the WL algorithm. Empirically, we find that the simple momentum method improves the efficiency of our algorithm. We illustrate the accelerated WL mixture method on two statistical models, the Log-Gaussian Cox process and the Bayesian Lasso.

Many aforementioned Monte Carlo techniques, including MTM and the WL weight adjustment, are also potentially useful in other Bayesian model selection approaches. In particular, we describe an MTM-based reversible jump MCMC framework, which can be useful when it is challenging to propose an appropriate surrogate. Specifically, we can include the model index $\mathcal{M}_k$ as a parameter in the full posterior distribution specified as

$$p(\boldsymbol{\theta}_k, \mathcal{M}_k \mid \boldsymbol{y}) \propto p\left(\boldsymbol{y} \mid \boldsymbol{\theta}_k, \mathcal{M}_k\right) p\left(\boldsymbol{\theta}_k \mid \mathcal{M}_k\right) p\left(\mathcal{M}_k\right), \tag{2.1}$$

and use MCMC to traverse the joint model and parameter space. The ratio between the proportions of time that the Markov chain spends in model $\mathcal{M}_i$ and model $\mathcal{M}_j$, adjusted by the prior on models, consistently estimates the Bayes factor $B_{i,j}$. A reversible jump MCMC (RJMCMC) (Green, 1995) algorithm is often required to sample across different dimensional spaces. However, it is well-known that constructing an efficient trans-dimensional proposal is challenging (Brooks et al., 2003).

To enable efficient RJMCMC, we propose to combine MTM and the directional sampling (Liu et al., 2000) algorithm, which will be most effective if $p\left(\boldsymbol{\theta}_k \mid \boldsymbol{y}, \mathcal{M}_k\right)$ is uni-modal for each model $\mathcal{M}_k$, and the mode $\widehat{\boldsymbol{\theta}}_k$ can be located reasonably well beforehand. We note that the proposed method is different from the MTM version of RJMCMC algorithms proposed in Pandolfi et al. (2014). Their method mainly focuses on using a computationally favourable weight function in MTM to avoid evaluating the target density, which can be expensive in complex statistical models. Our method is perhaps most similar to the mode jumping algorithm proposed in Tjelmeland and Hegstad (2001). While they design a mixture of Metropolis-Hastings proposals guided by deterministic local optimization to enable large step-size jumps, we utilize the more flexible MTM.

The rest of the article is organized as follows. Section 2.3.1 reviews the general WL algorithm. Section 2.3.2 proposes our mixture formulation, and explains how we adapt the WL algorithm in the mixture setting to estimate the normalizing constant. Section 2.3.3 introduces an accelerated version of the WL mixture method. Section 2.3.4 describes a principled way of using the variational approximation to construct the surrogate distribution. Section 2.4.1 explains how to use MTM to jump between the two mixture components if $q^{\star}(\boldsymbol{\theta})$ and $\gamma^{\star}(\boldsymbol{\theta})$ are not well aligned and relatively separated. Section 2.4.2 describes an efficient MTM-RJMCMC algorithm to sample the model space. Section 2.5 reviews existing methods in the literature, and makes connections and comparisons to our proposed methods. Section 2.6 illustrates the utility of the proposed methods on several numerical examples including a Bayesian evaluation of the Log-Gaussian Cox process fitting, a hyper-parameter selection problem for Bayesian Lasso regression, marginal likelihood estimation for a logistic regression model, and Bayesian variable selection for linear models under the spike-and-slab g-prior. Section 3.6 concludes with some final remarks.

2.3 A Surrogate Mixture Approach

2.3.1 The Wang-Landau algorithm

In order to improve the convenience and efficiency of the multicanonical sampling (Berg and Neuhaus, 1992), Wang and Landau (2001a,b) proposed a simple stochastic adaptive updating algorithm, which quickly becomes a popular Monte Carlo method for sampling complex physical systems. Given a target distribution $p(\boldsymbol{\theta})$ and a user-specified partition of the target space $\Theta = \cup_{i=1}^{s} \Theta_i$, where s is the total number of subregions, we can use the WL algorithm to estimate the probability mass of $p(\boldsymbol{\theta})$ within each subregion, i.e., $\psi(i) = \mathbb{P}(\boldsymbol{\theta} \in \Theta_i)$. The main steps of the WL algorithm are outlined in Algorithm 3.

Algorithm 3: The Wang-Landau algorithm.

---

1. Sample $\boldsymbol{\theta}_t$ from $K_{t-1}(\boldsymbol{\theta}_{t-1}, \cdot)$.
2. Update $\psi_t(i) \leftarrow \psi_{t-1}(i)\left[1 + \eta_t 1(\boldsymbol{\theta}_t \in \Theta_i)\right]$ for $i \in [s]$.
3. Normalize $\{\psi_t(i)\}_{i=1}^{s}$ to sum 1.

To initialize the algorithm, we can simply set $\psi_0(i) = 1/s$, and sample $\boldsymbol{\theta}_0$ from some initial distribution. K_t is a Markov kernel invariant to the adaptive target distribution $p_t^{\dagger}(\boldsymbol{\theta})$ defined as below,

$$p_t^{\dagger}(\boldsymbol{\theta}) \propto \sum_{i=1}^{s} \frac{p(\boldsymbol{\theta})}{\psi_t(i)} 1(\boldsymbol{\theta} \in \Theta_i). \tag{2.2}$$

The parameter η_t is the learning rate, and typically we shall scale it down following the flat/minimum histogram criterion (Zhou and Bhatt, 2005; Belardinelli and Pereyra, 2007) so as to guarantee the convergence of the algorithm. The convergence of $\psi_t(i)$ to $\psi(i)$ for $i \in [s]$ has been established in Atchadé and Liu (2010) and Fort et al. (2015) under proper conditions. Thus, $\boldsymbol{\theta}_t$ will spend equal amount of time in each subregion Θ_i as $t \to \infty$. We note that the magnitude of the learning rate η_t

is informative of the estimation error (Zhou and Bhatt, 2005). Therefore, a commonly used stopping criteria for the WL algorithm is that η_t is small enough.

2.3.2 The Wang-Landau mixture method

Suppose we have a (unnormalized) surrogate distribution $q(\boldsymbol{\theta})$ with its normalizing constant Z_q known. In addition, we assume that we have two effective Markov kernels K_γ and K_q in hand so that we can sample from $\gamma^\star(\boldsymbol{\theta})$ and $q^\star(\boldsymbol{\theta})$ sufficiently well. Since we control the construction of the surrogate $q^\star(\boldsymbol{\theta})$, we can typically make it easy to sample from without using MCMC. More details on constructing the surrogate $q^\star(\boldsymbol{\theta})$ are deferred to Section 2.3.4.

The proposed method relies on the following mixture formulation:

$$\pi(\boldsymbol{\theta}) = \gamma(\boldsymbol{\theta}) + q(\boldsymbol{\theta}). \tag{2.3}$$

The key is to recognize that the normalizing constants Z_γ and Z_q are proportional to the relative probability masses of the two components, $\gamma^\star(\boldsymbol{\theta})$ and $q^\star(\boldsymbol{\theta})$, in the mixture distribution $\pi(\boldsymbol{\theta})$. Therefore, we can directly apply the WL algorithm to estimate the ratio Z_γ/Z_q [*], where the two mixture components $\gamma^\star(\boldsymbol{\theta})$ and $q^\star(\boldsymbol{\theta})$ naturally defines a "partition" of the parameter space Θ. In view of step 1 in Algorithm 3, it requires an efficient kernel K_t to sample from the adaptive mixture distribution $\pi_t^\dagger(\boldsymbol{\theta})$ defined as below,

$$\pi_t^\dagger(\boldsymbol{\theta}) \propto \frac{\gamma(\boldsymbol{\theta})}{\psi_t(\gamma)} + \frac{q(\boldsymbol{\theta})}{\psi_t(q)}, \tag{2.4}$$

in which the γ and q in the brackets serve as indexes (the same as i in Algorithm 3), and should not be misinterpreted as function arguments. In the case where $q^\star(\boldsymbol{\theta})$ and $\gamma^\star(\boldsymbol{\theta})$ are well mixed, we exploit the data augmentation strategy and perform a Gibbs sampling step (Diebolt and Robert, 1994) using K_γ and K_q. To be specific, we define a binary indicator I_t to denote the mixture component

[*] For numerical stability, we recommend to work on the logarithmic scale.

from which $\boldsymbol{\theta}_t$ comes. Given $\{\psi_{t-1}(\gamma), \psi_{t-1}(q)\}$ and $\{\boldsymbol{\theta}_{t-1}, I_{t-1}\}$, if $I_{t-1} = 1$, we sample $\boldsymbol{\theta}_t$ from $K_\gamma(\boldsymbol{\theta}_{t-1}, \cdot)$, otherwise we sample $\boldsymbol{\theta}_t$ from $K_q(\boldsymbol{\theta}_{t-1}, \cdot)$. Given $\boldsymbol{\theta}_t$, we then sample I_t from a Bernoulli distribution with probability

$$\mathbb{P}(I_t = 1 \mid |\boldsymbol{\theta}_t) \propto \gamma(\boldsymbol{\theta}_t)/\psi_{t-1}(\gamma), \quad \mathbb{P}(I_t = 0 \mid |\boldsymbol{\theta}_t) \propto q(\boldsymbol{\theta}_t)/\psi_{t-1}(q). \tag{2.5}$$

In the case where $q^\star(\boldsymbol{\theta})$ and $\gamma^\star(\boldsymbol{\theta})$ are not well aligned, we complement Gibbs sampling with the Multiple-try Metropolis (MTM) if informative jumping directions are identified beforehand. We defer a detailed discussion on MTM to Section 2.4.1.

A detailed algorithm implementing the WL mixture method is summarized in Algorithm 4. The generic Markov kernel K_t invariant to $\pi_t^\dagger(\boldsymbol{\theta})$ can be substituted by the Gibbs sampling kernel discussed above or the MTM kernel. In addition, instead of terminating the algorithm after a prespecified number of iterations S, we can also use the stopping criteria that the learning rate η_t is small enough (say, below 10^{-3}).

Tunable parameter sequences $\{\xi_t(\gamma),\ \xi_t(q)\}$ are introduced to help check the flat histogram criterion, that is, whether the Markov chain has spent equal amount of time in each of the two mixture components $\gamma^\star(\boldsymbol{\theta})$ and $q^\star(\boldsymbol{\theta})$. If this is approximately satisfied to the extent controlled by a threshold $c \in (0, 1)$ (see Equation (2.7)), we decrease the learning rate and refresh $\xi_t(\gamma) = \xi_t(q) = 0$ so that it can start monitoring the next stage of the algorithm. Empirically we find that the performance of the WL mixture method is robust to the choice of c (see Figure B.1 in the Appendix B). For the numerical examples in the paper, we set $c = 0.2$.

The WL mixture method naturally adapts to the missing data framework. The marginal likelihood of the observed-data can be formulated as follows:

$$L(\boldsymbol{y}_{\text{obs}}) = \int p(\boldsymbol{y}_{\text{obs}} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta}) d\boldsymbol{\theta} = \int \int p(\boldsymbol{y}_{\text{obs}}, \boldsymbol{y}_{\text{mis}} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta}) d\boldsymbol{y}_{\text{mis}} d\boldsymbol{\theta},$$

Algorithm 4: The Wang-Landau mixture method.

1. Algorithmic setup. Choose a decreasing positive sequence $\{\eta_t\}$ as the sequence of learning rate. Set $a_0 = 1$, $c \in (0,1)$, $\xi_0(\gamma) = \xi_0(q) = 0$, and $\psi_0(\gamma) = \psi_0(q) = 1/2$. Set the total number of iterations to be S. We exclude the first b iterations in estimation.

2. At $t = 0$: initialize $\boldsymbol{\theta}_0$ from some initial distribution, and sample a binary indicator I_0 with probability $\mathbb{P}(I_0 = 1) \propto \gamma(\boldsymbol{\theta}_0)$ and $\mathbb{P}(I_0 = 0) \propto q(\boldsymbol{\theta}_0)$.

3. For $t \in [S]$: given $(\boldsymbol{\theta}_{t-1}, I_{t-1})$, iterate between the following steps.

   (a) Sample $\boldsymbol{\theta}_t$ from $K_{t-1}(\boldsymbol{\theta}_{t-1}, \cdot)$, which is invariant to the the adaptive mixture distribution $\pi^{\dagger}_{t-1}(\boldsymbol{\theta})$ defined by $\{\psi_{t-1}(\gamma), \psi_{t-1}(q)\}$ as in Equation (2.4).

   (b) Sample a binary indicator I_t with probability specified in Equation (2.5).

   (c) Update $\{\xi_t(\gamma), \xi_t(q)\}$ and $\{\psi_t(\gamma), \psi_t(q)\}$ as follows:

$$\begin{aligned} &\xi_t(\gamma) \leftarrow \xi_{t-1}(\gamma) + 1\left(I_t = 1\right), \quad \psi_t(\gamma) \leftarrow \psi_{t-1}(\gamma)\left[1 + \eta_{a_t} 1\left(I_t = 1\right)\right], \\ &\xi_t(q) \leftarrow \xi_{t-1}(q) + 1\left(I_t = 0\right), \quad \psi_t(q) \leftarrow \psi_{t-1}(q)\left[1 + \eta_{a_t} 1\left(I_t = 0\right)\right]. \end{aligned} \tag{2.6}$$

   (d) Normalize $\{\psi_t(\gamma), \psi_t(q)\}$ to sum 1.

   (e) If the following condition is satisfied:

$$\frac{\max\{\xi_t(\gamma), \xi_t(q)\}}{\xi_t(\gamma) + \xi_t(q)} - \frac{1}{2} \leq \frac{c}{2}, \tag{2.7}$$

   update $a_{t+1} = a_t + 1$ and reset $\xi_t(\gamma) = \xi_t(q) = 0$. Otherwise set $a_{t+1} = a_t$.

4. Output the estimators $\log \widehat{Z}_\gamma = \log \widehat{r} + \log Z_q$, where

$$\log \widehat{r} = \frac{1}{S - b} \sum_{t=b+1}^{S} [\log \xi_t(\gamma) - \log \xi_t(q)].$$

in which $p(\boldsymbol{y}_{\text{obs}}, \boldsymbol{y}_{\text{mis}} \mid \boldsymbol{\theta})$ is the complete-data distribution, and $p(\boldsymbol{\theta})$ is the prior. When the integral $\int p(\boldsymbol{y}_{\text{obs}}, \boldsymbol{y}_{\text{mis}} \mid \boldsymbol{\theta}) d\boldsymbol{y}_{\text{mis}}$ can be analytically calculated, such as the finite mixture model in which $\boldsymbol{y}_{\text{mis}}$ are discrete, we can directly apply the WL mixture method to estimate the normalizing constant $L(\boldsymbol{y}_{\text{obs}})$. More generally, we can treat the missing data $\boldsymbol{y}_{\text{mis}}$ as parameters, and apply the WL mixture method to estimate the normalizing constant of the (unnormalized) complete-data posterior distribution $\gamma(\boldsymbol{\theta}, \boldsymbol{y}_{\text{mis}} \mid \boldsymbol{y}_{\text{obs}}) = p(\boldsymbol{y}_{\text{obs}}, \boldsymbol{y}_{\text{mis}} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})$.

Another extension of the WL mixture method is to introduce a sequence of auxiliary distributions $\{\eta_0^\star(\boldsymbol{\theta}), \cdots, \eta_p^\star(\boldsymbol{\theta})\}$ between the posterior and the surrogate, i.e., $\eta_0^\star(\boldsymbol{\theta}) = q^\star(\boldsymbol{\theta})$ and $\eta_p^\star(\boldsymbol{\theta}) = \gamma^\star(\boldsymbol{\theta})$, and estimate the ratios of the normalizing constants of any two adjacent auxiliary distributions in parallel. The idea of constructing auxiliary distributions has been widely used in Monte Carlo simulations, including bridge sampling, annealed importance sampling, sequential importance sampling, and SMC algorithms. However, these earlier methods typically cannot be easily parallelized.

This generic idea is beneficial if the sequence of auxiliary distributions is properly chosen. Discussions and references on constructing auxiliary distributions are given in Section 2.5.1. We refer to this multiple-step WL mixture method as the parallel WL (PWL) method henceforth, and illustrate it in Section 2.6.1 on the Log-Gaussian Cox process, in which we set the prior as the surrogate, and employ a geometric sequence of auxiliary distributions. We believe that the parallel WL method can complement the WL mixture method when a good surrogate $q^\star(\boldsymbol{\theta})$ is not easy to construct.

2.3.3 Acceleration of the Wang-Landau mixture method

The efficiency of the WL mixture method can be further improved using the acceleration idea discussed in Dai and Liu (2020). The update of the reweighting factor $\psi_t(\gamma)$ (step 3(c) in Algorithm 4)

can be approximated as follows:

$$\begin{aligned}\log \psi_t(\gamma) &= \log \psi_{t-1}(\gamma) + \log\left[1 + \eta_{a_t} 1\left(I_t = 1\right)\right] \\ &\approx \log \psi_{t-1}(\gamma) + \eta_{a_t} 1\left(I_t = 1\right).\end{aligned} \tag{2.8}$$

We note that the above approximation is fairly accurate since the learning rate η_t is small.

Let $u_t(\gamma) = \log \psi_t(\gamma)$ and $u_t(q) = \log \psi_t(q)$. We consider the following optimization problem:

$$\begin{aligned}&\min_{u_1, u_2 \in \mathbb{R}} f(u_1, u_2) = \log\left(\exp(u^\star(\gamma) - u_1) + \exp(u^\star(q) - u_2)\right), \\ &\text{subject to } \; u_1 + u_2 = 0,\end{aligned} \tag{2.9}$$

in which $u^\star(\gamma) = \log Z_\gamma - \log Z_q$ and $u^\star(q) = \log Z_q - \log Z_\gamma$. It is not difficult to see that this is a convex optimization problem because the objective function $f(u_1, u_2)$ is a log-sum-exp function and the constraint is linear. It has a unique solution at $(u^\star(\gamma), u^\star(q))$.

To solve the constrained optimization problem (2.9), we use the projected gradient descent algorithm. More precisely, one-step update of $u_t(\gamma)$ is

$$\begin{aligned}u_t(\gamma) &= u_{t-1}(\gamma) - \eta_{a_t} \frac{\partial f}{\partial u_1}\left(u_{t-1}(\gamma), u_{t-1}(q)\right) \\ &= u_{t-1}(\gamma) + \eta_{a_t} \mathbb{P}\left(I_t = 1\right),\end{aligned} \tag{2.10}$$

in which $\mathbb{P}(I_t = 1)$ equals the weight of $\gamma^\star(\boldsymbol{\theta})$ in the mixture distribution $\pi^\dagger_{t-1}(\boldsymbol{\theta})$ (see Equation (2.4)). $u_t(q)$ is updated in the same way. We note that anayltical evaluation of $\mathbb{P}\left(I_t = 1\right)$ involves the unknown normalizing constant Z_γ, thus is not feasible in practice. However, we can implement one-step or multiple-step Monte Carlo simulations, and approximate $\mathbb{P}\left(I_t = 1\right)$ by $1\left(I_t = 1\right)$. This recovers the WL update in Equation (2.8). In addition, the projection step to the set $\{u_1, u_2 \in \mathbb{R}, u_1 + u_2 = 0\}$ is equivalent to the normalization step (see step 3(d) in Algorithm 4). There-

fore, the WL algorithm is equivalent to the stochastic projected gradient descent algorithm solving the constrained optimization problem (2.9).

Once we have the optimization perspective, various acceleration tools can be employed to improve the efficiency of the WL mixture method. One simple tool we find useful is the momentum method, which exponentially accumulates a momentum vector to amplify the persistent gradient across iterations, thus reducing the oscillation caused by the noise in the gradient estimate. More precisely, we modify step 3(c) in Algorithm 4 as below.

3 (c′) (Momentum accelerated WL updates)

(i) Update the momentum vector:

$$m_t(\gamma) \leftarrow \beta m_{t-1}(\gamma) - \eta_{a_t} 1\left(I_t = 1\right), \quad m_t(q) \leftarrow \beta m_{t-1}(q) - \eta_{a_t} 1\left(I_t = 0\right).$$

(ii) Update the reweighting vector:

$$\log \psi_t(\gamma) \leftarrow \log \psi_{t-1}(\gamma) - m_t(\gamma), \qquad \log \psi_t(q) \leftarrow \log \psi_{t-1}(q) - m_t(q).$$

(iii) Update $\{\xi_t(\gamma), \xi_t(q)\}$ as in step 3(c) in Algorithm 4.

We initialize the momentum vector as $m_0(\gamma) = m_0(q) = 0$. β is commonly set to be 0.9 or higher, which calibrates the fraction of the accumulated past gradients that we want to incorporate into the current update. Numerical illustrations of the accelerated WL mixture method is given in Figure 2.2 on two statistical models, the Log-Gaussian Cox process and the Bayesian Lasso.

2.3.4 Constructing the surrogate distribution

In principle, any posterior approximation with a known normalizing constant, such as the Laplace approximation and the variational approximation, can be used to construct the surrogate distribution.

We can also use MCMC methods to obtain posterior samples, and fit some parametric distribution to them. In this section, we describe how to construct a surrogate $q(\boldsymbol{\theta})$ using the variational approximation (Jordan et al., 1999; Blei et al., 2017). The variational approach enjoys two main advantages. First, it is computationally efficient and does not require MCMC sampling to explore $\gamma^\star(\boldsymbol{\theta})$. Second, it provides a reasonable approximation to $\gamma^\star(\boldsymbol{\theta})$ in a wide class of statistical models (Wainwright and Jordan, 2008).

The variational approximation aims at finding the closest distribution $q^\star(\boldsymbol{\theta})$ to $\gamma^\star(\boldsymbol{\theta})$ in the KL divergence within a particular class of distributions Q, that is,

$$q^\star(\boldsymbol{\theta}) = \underset{p(\boldsymbol{\theta})\in Q}{\arg\min}\, \mathrm{KL}\left(p(\boldsymbol{\theta})||\gamma^\star(\boldsymbol{\theta})\right). \tag{2.11}$$

$\mathrm{KL}\left(p(\boldsymbol{\theta})||\gamma^\star(\boldsymbol{\theta})\right)$ is not computable as it involves the unknown normalizing constant Z_γ. However, we can equivalently reformulate the optimization problem (2.11) as follows:

$$q^\star(\boldsymbol{\theta}) = \underset{p(\boldsymbol{\theta})\in Q}{\arg\max}\, \mathrm{ELBO}(p) = \underset{p(\boldsymbol{\theta})\in Q}{\arg\max} \left\{ \mathbb{E}_p\left[\log \gamma(\boldsymbol{\theta})\right] - \mathbb{E}_p\left[\log p(\boldsymbol{\theta})\right] \right\}, \tag{2.12}$$

in which Z_γ is no longer involved. ELBO refers to the *evidence lower bound* of $\log Z_\gamma$ since

$$\log Z_\gamma = \mathrm{KL}\left(q^\star(\boldsymbol{\theta})||\gamma^\star(\boldsymbol{\theta})\right) + \mathrm{ELBO}(q^\star) \geq \mathrm{ELBO}(q^\star). \tag{2.13}$$

We note that the EM algorithm (Dempster et al., 1977) can also be formulated as a two-step iterative algorithm maximizing the ELBO with respect to $p(\boldsymbol{\theta})$ and the relevant model parameters (Tzikas et al., 2008).

Before solving the optimization problem (2.12), we need to specify the variational family Q. A commonly considered class of distributions Q is the mean-field variational family, which assumes that $q^\star(\boldsymbol{\theta})$ is a product of univariate distributions, that is, $q^\star(\boldsymbol{\theta}) = \prod_{j=1}^d q_j^\star(\theta_j)$. We assume that

$q_j^\star(\theta_j)$ belongs to some parametric family Q_j whose probability density function can be evaluated exactly. To solve the optimization problem (2.12), we can use the *coordinate ascent variational inference* (CAVI) algorithm (Bishop, 2006). CAVI, detailed in Algorithm 5, iteratively maximizes the ELBO in a coordinate-wise fashion. We note that the optimization problem in (2.15) can be further simplified in conjugate cases. For each $j \in [d]$, conditioning on all the other components $q_i^\star(\theta_i),\ i \neq j$, $\text{ELBO}(q_j)$ can be rewritten as

$$\text{ELBO}(q_j) = -\text{KL}\left(q_j(\theta_j) || q_j^{\text{opt}}(\theta_j)\right) + \text{constant}, \tag{2.14}$$

in which $q_j^{\text{opt}}(\theta_j) \propto \exp\left[\mathbb{E}_{-j}\left(\log \gamma(\theta_j, \boldsymbol{\theta}_{-j})\right)\right]$. If $q_j^{\text{opt}}(\theta_j) \in Q_j$ (conjugacy), the optimal $q_j(\theta_j)$ is $q_j^{\text{opt}}(\theta_j)$ since the KL divergence is non-negative.

Algorithm 5: The coordinate ascent variational inference (CAVI) (Blei et al., 2017).

---

1. Initialize each $q_j^\star(\theta_j) \in Q_j$ for $j \in [d]$.
2. For each $j \in [d]$, fix all the other components $q_i^\star(\theta_i),\ i \neq j$, update $q_j^\star(\theta_j)$ with

$$q_j^\star(\theta_j) = \underset{q_j \in Q_j}{\arg\max} \left\{ \mathbb{E}_j\left[\mathbb{E}_{-j}\left(\log \gamma(\theta_j, \boldsymbol{\theta}_{-j})\right)\right] - \mathbb{E}_j\left[\log q_j(\theta_j)\right] \right\}. \tag{2.15}$$

3. Calculate $\text{ELBO}(q^\star)$ where $q^\star(\boldsymbol{\theta}) = \prod_{j=1}^d q_j^\star(\theta_j)$. If ELBO hasn't converged, go back to step 2. Otherwise output $q^\star(\boldsymbol{\theta})$.

2.4 Global Jump via Multiple-try Metropolis

2.4.1 The Multiple-try Metropolis

In many problems, the two mixture components $q^\star(\boldsymbol{\theta})$ and $\gamma^\star(\boldsymbol{\theta})$ may not be well aligned and sometimes can be completely separated. Thus, naive Metropolis-Hastings proposals may be easily trapped

in one of the components and cannot efficiently traverse the whole space. We here describe an approach based on the Multiple-try Metropolis (MTM) method (Liu et al., 2000) for constructing a proper Markov kernel that enables easy jumps between different modes.

Given a target distribution $\pi(\boldsymbol{x})$ defined on $\mathbb{R}^d$ and a proposal transition function $T(\boldsymbol{x}, \boldsymbol{y})$, a version of MTM is described in in Algorithm 6. Heuristically, MTM aims at *biasing* the multiple proposals with a proper weight function $w(\boldsymbol{x}, \boldsymbol{y})$:

$$w(\boldsymbol{x}, \boldsymbol{y}) = \pi(\boldsymbol{x}) T(\boldsymbol{x}, \boldsymbol{y}) \lambda(\boldsymbol{x}, \boldsymbol{y}), \tag{2.16}$$

in which $\lambda(\boldsymbol{x}, \boldsymbol{y})$ is a user-chosen nonnegative symmetric function. Briefly, MTM draws m proposals $\{\boldsymbol{y}^{(1)}, \cdots, \boldsymbol{y}^{(m)}\}$ from the transition function $T(\boldsymbol{x}_t, \cdot)$, and then selects $\boldsymbol{y}$ from $\{\boldsymbol{y}^{(1)}, \cdots, \boldsymbol{y}^{(m)}\}$ with probability proportional to $w(\boldsymbol{y}^{(j)}, \boldsymbol{x})$. A proper acceptance-rejection rule (steps 3 and 4 in Algorithm 6) is employed to ensure the reversibility of the Markov chain.

Algorithm 6: The Multiple-try Metropolis (Liu et al., 2000).

1. Sample $\boldsymbol{y}^{(1)}, \cdots, \boldsymbol{y}^{(m)}$ i.i.d from $T(\boldsymbol{x}_t, \cdot)$. Compute the weight function $w(\boldsymbol{y}^{(j)}, \boldsymbol{x})$.
2. Sample $\boldsymbol{y}$ from $\boldsymbol{y}^{(1)}, \cdots, \boldsymbol{y}^{(m)}$ with probability proportional to $w(\boldsymbol{y}^{(j)}, \boldsymbol{x})$.
3. Given $\boldsymbol{y}$, sample $\boldsymbol{x}^{(1)}, \cdots, \boldsymbol{x}^{(m-1)}$ i.i.d from $T(\boldsymbol{y}, \cdot)$. Set $\boldsymbol{x}^{(m)} = \boldsymbol{x}_t$.
4. Accept $\boldsymbol{y}$ with probability:

$$\alpha = \min\left\{1, \quad \frac{w(\boldsymbol{y}^{(1)}, \boldsymbol{x}) + \cdots + w(\boldsymbol{y}^{(m)}, \boldsymbol{x})}{w(\boldsymbol{x}^{(1)}, \boldsymbol{y}) + \cdots + w(\boldsymbol{x}^{(m)}, \boldsymbol{y})}\right\}. \tag{2.17}$$

A special choice of λ is $\lambda(\boldsymbol{x}, \boldsymbol{y}) = [T(\boldsymbol{x}, \boldsymbol{y}) + T(\boldsymbol{y}, \boldsymbol{x})]^{-1}$. If $T(\boldsymbol{x}, \boldsymbol{y})$ is also a symmetric proposal, the corresponding acceptance probability simplifies to:

$$\alpha = \min\left\{1, \quad \frac{\pi(\boldsymbol{y}^{(1)}) + \cdots + \pi(\boldsymbol{y}^{(m)})}{\pi(\boldsymbol{x}^{(1)}) + \cdots + \pi(\boldsymbol{x}^{(m)})}\right\}. \tag{2.18}$$

This special case is referred to as MTM (II) in Liu et al. (2000). MTM is particularly useful when it is combined with the directional sampling algorithm. For instance, if we know a desirable jumping direction, we can use MTM to explore a wide range along it. Let $\boldsymbol{e}$ denote the jumping direction. For the simple case where $\boldsymbol{e}$ is fixed and independent of the current state $\boldsymbol{x}_t$, we outline the main steps in Algorithm 7. More generally, we can choose the jumping direction $\boldsymbol{e}$ based on the current state $\boldsymbol{x}_t$. Some detailed discussion on a special form of this adaptive strategy can be found in Section 2.4.2.

Algorithm 7: The Multiple-try Metropolis combined with directional sampling.

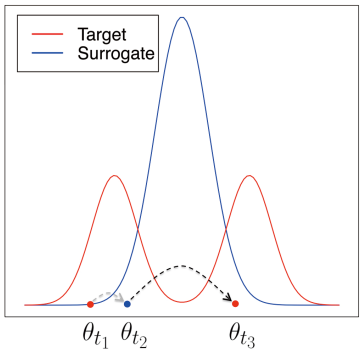
---

1. Sample $r^{(1)}, \cdots, r^{(m)}$ from a user-chosen distribution $p(r)$. Let $\boldsymbol{y}^{(j)} = \boldsymbol{x}_t + r^{(j)} \cdot \boldsymbol{e}$. Compute the target density $\pi(\boldsymbol{y}^{(j)})$.
2. Sample $\boldsymbol{y}$ from $\boldsymbol{y}^{(1)}, \cdots, \boldsymbol{y}^{(m)}$ with probability proportional to $\pi(\boldsymbol{y}^{(j)})$. Set $\boldsymbol{x}^{(j)} = \boldsymbol{y} - r^{(j)} \cdot \boldsymbol{e}$.
3. Accept $\boldsymbol{x}_{t+1} = \boldsymbol{y}$ with probability:

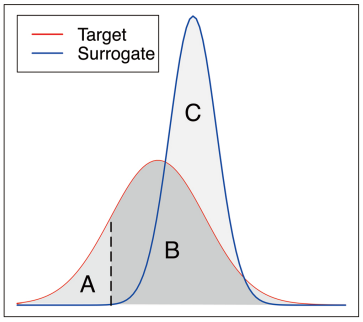
$$\alpha = \min \left\{ 1, \quad \frac{\pi(\boldsymbol{y}^{(1)}) + \cdots + \pi(\boldsymbol{y}^{(m)})}{\pi(\boldsymbol{x}^{(1)}) + \cdots + \pi(\boldsymbol{x}^{(m)})} \right\}. \tag{2.19}$$

When we incorporate MTM in the WL mixture method, assuming that we are equipped with an efficient kernel K_γ, some pre-MCMC runs should help us pin down informative jumping directions, such as the directions connecting the modes of the posterior and the surrogate. Thus, we can substitute step 3(a) in Algorithm 4 by randomly alternating between MTM and Gibbs sampling.

When the posterior $\gamma^\star(\boldsymbol{\theta})$ is multimodal, the proposed surrogate mixture framework can potentially help identify the multimodality of $\gamma^\star(\boldsymbol{\theta})$, upon which a better K_γ can be designed using MTM. The idea is similar to parallel tempering (Geyer, 1991), which relies on a sequence of auxiliary distributions so that global jumps are possible by "transporting" samples back and forth from the posterior to auxiliary distributions. In our setting, the surrogate $q^\star(\boldsymbol{\theta})$ plays a similar role as the auxiliary distributions used in parallel tempering, and the WL weight adjustment ensures that transitions between



(a) Demonstration of mode jumping. The color indicates the mixture component that the sample comes from. Initially, $\boldsymbol{\theta}_{t_1}$ is considered to be more likely a sample from $\gamma^\star(\boldsymbol{\theta})$ (step 3(b) in Algorithm 4). The WL mixture method keeps downweighting the mixture component $\gamma^\star(\boldsymbol{\theta})$ by increasing its reweighting factor $\psi_{t_1}(\gamma)$ (step 3(c) in Algorithm 4). In the meantime, we perform Gibbs sampling steps, which locally moves the chain using the Markov kernel K_γ, thus the chain still stays around the same local mode. After $\gamma^\star(\boldsymbol{\theta})$ has been downweighted enough, at some point t_2, $\boldsymbol{\theta}_{t_2}$ is considered to be more likely a sample from $q^\star(\boldsymbol{\theta})$. In the next step we directly sample from $q^\star(\boldsymbol{\theta})$, leading to a global jump from $\boldsymbol{\theta}_{t_2}$ to $\boldsymbol{\theta}_{t_3}$, which identifies the other mode of $\gamma^\star(\boldsymbol{\theta})$.



(b) Comparison between the WL mixture method and importance sampling. Importance sampling only estimates the normalizing constant of the target restricted on the region B, thus yields an underestimated normalizing constant. In contrast, the WL mixture method has approximately equal chance to explore the whole high-density regions of the target and the surrogate distributions, respectively, thus produces more accurate, rather than underestimated, normalizing constant estimates.

Figure 2.1

the posterior and the auxiliary distribution are sufficiently frequent. As shown in Figure 2.1a, we consider a setting where $\gamma^\star(\boldsymbol{\theta})$ is bimodal and $q^\star(\boldsymbol{\theta})$ covers both modes, but we are unaware of the multimodality of $\gamma^\star(\boldsymbol{\theta})$ in the first place and K_γ only enables local moves. As is explained in the caption of Figure 2.1a, by leveraging the surrogate $q^\star(\boldsymbol{\theta})$ under the help of the WL reweighting, the algorithm helps the chain cross the two modes of $\gamma^\star(\boldsymbol{\theta})$.

2.4.2 An extension: MTM for reversible jump MCMC

We discuss possible utilizations of some aforementioned ideas in the classical reversible-jump MCMC (RJMCMC) (Green, 1995) framework for Bayesian model comparison. That is, instead of estimating the normalizing constant of each model, we incorporate the model index into the joint posterior distribution defined in Equation (2.1), and sample the model indicator and model parameters simultaneously. It generally requires RJMCMC since the posterior distribution $p\left(\boldsymbol{\theta}_k, \mathcal{M}_k \mid \boldsymbol{y}\right)$ is potentially trans-dimensional.

Effective jumping mechanisms are necessary for successful traverse across different model spaces. We believe both MTM and the WL weight adjustment are potentially useful in constructing efficient trans-dimensional proposals. (i) The MTM directional sampling may guide the Markov chain to directly jump towards the mode of the within-model posterior, so that the acceptance probability can be much higher than other generic jumping mechanisms. (ii) The WL weight adjustment can be used to balance the probability masses of different models in the joint posterior distribution $p\left(\boldsymbol{\theta}_k, \mathcal{M}_k \mid \boldsymbol{y}\right)$, so as to facilitate trans-dimensional jumps. In this chapter, we detail the discussion on the MTM directional sampling in RJMCMC, and leave the idea of using the WL weight adjustment in RJMCMC for future explorations.

We note that the MTM-based reversible jump algorithm (MTM-RJMCMC) is most useful if (i) $p(\boldsymbol{\theta}_k \mid \boldsymbol{y}, \mathcal{M}_k)$ is approximately unimodal for each model $\mathcal{M}_k$; (ii) we can estimate the mode $\widehat{\boldsymbol{\theta}}_k$ reasonably well before running the algorithm. For $i \neq j$, suppose we want to move from model $\mathcal{M}_i$ to model $\mathcal{M}_j$. Since $\boldsymbol{\theta}_i \in \mathbb{R}^{d_i}$ and $\boldsymbol{\theta}_j \in \mathbb{R}^{d_j}$ are potentially in different dimensions, we first match the dimension of $\boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_j$ by introducing auxiliary parameters $\boldsymbol{u} \in \mathbb{R}^{d_j}$ and $\boldsymbol{v} \in \mathbb{R}^{d_i}$ so that the dimension and domain of $(\boldsymbol{\theta}_i, \boldsymbol{u})$ matches those of $(\boldsymbol{v}, \boldsymbol{\theta}_j)$. We note that this is just one principled way to match the parameter spaces. For specific problems, more efficient designs may exist and should be considered.

We define the augmented posterior distributions as

$$
\begin{aligned}
p_i\left(\boldsymbol{\theta}_i, \boldsymbol{u}, \mathcal{M}_i \mid \boldsymbol{y}\right) &= p(\boldsymbol{\theta}_i \mid \boldsymbol{y}, \mathcal{M}_i) q_i(\boldsymbol{u}) p(\mathcal{M}_i), \\
p_j\left(\boldsymbol{v}, \boldsymbol{\theta}_j, \mathcal{M}_j \mid \boldsymbol{y}\right) &= p(\boldsymbol{\theta}_j \mid \boldsymbol{y}, \mathcal{M}_j) q_j(\boldsymbol{v}) p(\mathcal{M}_j),
\end{aligned}
\tag{2.20}
$$

in which $q_i(\boldsymbol{u})$ and $q_j(\boldsymbol{v})$ are user-chosen unimodal distributions with modes denoted as $\widehat{\boldsymbol{u}}$ and $\widehat{\boldsymbol{v}}$. The above construction implies that $\boldsymbol{\theta}_i \perp \boldsymbol{u}$ and $\boldsymbol{\theta}_j \perp \boldsymbol{v}$. In general, we can consider introducing dependence structures between $\boldsymbol{\theta}_i, \boldsymbol{u}$ and $\boldsymbol{\theta}_j, \boldsymbol{v}$.

The multiple-try trans-dimensional move from model $\mathcal{M}_i$ to model $\mathcal{M}_j$ is summarized in Algorithm 8 and briefly explained here. Given the current state $\boldsymbol{\theta}_i$, we first sample $\boldsymbol{u}$ from $q_i(\boldsymbol{u})$, then construct m proposals $(\boldsymbol{v}^{(k)}, \boldsymbol{\theta}_j^{(k)}) = (\boldsymbol{\theta}_i, \boldsymbol{u}) + r^{(k)} \cdot \boldsymbol{e}$ for $k \in [m]$. Two types of directional jumping mechanisms can be exploited: (i) fixed-directional jump; (ii) adaptive-directional jump. For the fixed-directional jump, the jumping direction is defined by the two pre-located modes of the augmented posteriors $p_i\left(\boldsymbol{\theta}_i, \boldsymbol{u}, \mathcal{M}_i \mid \boldsymbol{y}\right)$ and $p_j\left(\boldsymbol{v}, \boldsymbol{\theta}_j, \mathcal{M}_j \mid \boldsymbol{y}\right)$, and is fixed throughout the algorithm. For the adaptive-directional jump, the jumping direction is defined by the current state of the chain $(\boldsymbol{\theta}_i, \boldsymbol{u})$ and the mode of the augmented posterior $p_j\left(\boldsymbol{v}, \boldsymbol{\theta}_j, \mathcal{M}_j \mid \boldsymbol{y}\right)$.

There are subtle differences in the implementation of the two jumping mechanisms. If we use the adaptive-directional jump to *jump towards* a mode, that is, $\boldsymbol{e} = (\widehat{\boldsymbol{v}} - \boldsymbol{\theta}_i, \widehat{\boldsymbol{\theta}}_j - \boldsymbol{u})/||(\widehat{\boldsymbol{v}} - \boldsymbol{\theta}_i, \widehat{\boldsymbol{\theta}}_j - \boldsymbol{u})||$, the sampling distribution $p(r)$ of the jumping distance r is required to be a centered symmetric distribution in order that the acceptance probability can be simplified as in Equation (2.23). A more general $p(r)$ is allowed if we use a generalized form of MTM in Liu et al. (2000). In contrast, for the fixed-directional jump, we can simply set the jumping direction as $\boldsymbol{e} = (\widehat{\boldsymbol{v}} - \widehat{\boldsymbol{\theta}}_i, \widehat{\boldsymbol{\theta}}_j - \widehat{\boldsymbol{u}})$ without standardization, and sample the jumping distance r from an arbitrary distribution, not necessarily being symmetric and centered at 0. In fact, to push the chain directly jump into the mode $(\widehat{\boldsymbol{v}}, \widehat{\boldsymbol{\theta}}_j)$, we recommend to center $p(r)$ at 1. We note that the acceptance probability of the adaptive-directional

jump involves an additional Jacobian defined as below,

$$J((\boldsymbol{\theta}_i, \boldsymbol{u}), (\boldsymbol{v}, \boldsymbol{\theta}_j)) = \left| 1 - \frac{||(\boldsymbol{v} - \boldsymbol{\theta}_i, \boldsymbol{\theta}_j - \boldsymbol{u})||}{||(\widehat{\boldsymbol{v}} - \boldsymbol{\theta}_i, \widehat{\boldsymbol{\theta}}_j - \boldsymbol{u})||} \right|^{d_i + d_j - 1}. \tag{2.21}$$

For the fixed-directional jump, the Jacobian is not required.

Each of the two jumping mechanisms has its own advantages depending on the scenarios. For instance, if the local variations around two modes $(\widehat{\boldsymbol{\theta}}_i, \widehat{\boldsymbol{u}})$ and $(\widehat{\boldsymbol{v}}, \widehat{\boldsymbol{\theta}}_j)$ differ significantly, the adaptive-directional jump is more favorable, as the fixed jumping direction can be misleading when the chain jumps from the relatively wider mode to the narrower one. On the other hand, since $p(r)$ for the fixed-directional jump is more flexible, e.g., $p(r)$ can be centered at 1, when the jumping direction $\boldsymbol{e} = (\widehat{\boldsymbol{v}} - \widehat{\boldsymbol{\theta}}_i, \widehat{\boldsymbol{\theta}}_j - \widehat{\boldsymbol{u}})$ is indeed informative, it appears more efficient than the adaptive-directional jump in which $p(r)$ centers at 0.

We then sample $(\boldsymbol{v}, \boldsymbol{\theta}_j)$ from the multiple proposals $\{(\boldsymbol{v}^{(k)}, \boldsymbol{\theta}_j^{(k)})\}_{k=1}^m$, with probability proportional to the augmented posterior density $p_j(\boldsymbol{v}^{(k)}, \boldsymbol{\theta}_j^{(k)}, \mathcal{M}_j \mid \boldsymbol{y})$. After obtaining $(\boldsymbol{v}, \boldsymbol{\theta}_j)$, we set $(\boldsymbol{\theta}_i^{(k)}, \boldsymbol{u}^{(k)}) = (\boldsymbol{v}, \boldsymbol{\theta}_j) - r^{(k)} \cdot \boldsymbol{e}$ for $k \in [m]$. We accept the trans-dimensional proposal $(\boldsymbol{v}, \boldsymbol{\theta}_j)$ with probability α given in Equations (2.22) and (2.23), depending on which jumping mechanism we use. We note that the proposed trans-dimensional move should be combined with local MCMC moves within each model $\mathcal{M}_k$. Since the current setting is slightly different from that of a typical MTM, we provide here a theoretical justification in the following proposition. The proof of Proposition 1 can be found in the Appendix B.

Proposition 1: The proposed trans-dimensional move, equipped with either the fixed-directional jumping mechanism or the adaptive-directional jumping mechanism, leaves the posterior distribution $p(\boldsymbol{\theta}_k, \mathcal{M}_k \mid \boldsymbol{y})$ invariant.

Algorithm 8: The Multiple-try Metropolis reversible jump MCMC algorithm.

For $i \neq j$, suppose the current posterior draw $\boldsymbol{\theta}_i$ is from model $\mathcal{M}_i$, the trans-dimensional move to model $\mathcal{M}_j$ is accomplished as follows.

1. Sample the auxiliary variable $\boldsymbol{u}$ from $q_i(\boldsymbol{u})$, of which the dimension and domain match those of $\boldsymbol{\theta}_j$ in model $\mathcal{M}_j$.

2. Set the jumping direction and sample the jumping distances.

   (a) (Fixed-directional jump) Set the jumping direction as $\boldsymbol{e} = (\widehat{\boldsymbol{v}} - \widehat{\boldsymbol{\theta}}_i, \widehat{\boldsymbol{\theta}}_j - \widehat{\boldsymbol{u}})$. Sample the jumping distances $r^{(1)}, \cdots, r^{(m)}$ from an arbitrary distribution $p(r)$ (recommend to center $p(r)$ at 1).

   (b) (Adaptive-directional jump) Set the jumping direction as $\boldsymbol{e} = (\widehat{\boldsymbol{v}} - \boldsymbol{\theta}_i, \widehat{\boldsymbol{\theta}}_j - \boldsymbol{u})/||(\widehat{\boldsymbol{v}} - \boldsymbol{\theta}_i, \widehat{\boldsymbol{\theta}}_j - \boldsymbol{u})||$. Sample the jumping distances $r^{(1)}, \cdots, r^{(m)}$ from a *symmetric* distribution $p(r)$ centered at 0.

3. Propose multiple tries: set $(\boldsymbol{v}^{(k)}, \boldsymbol{\theta}_j^{(k)}) = (\boldsymbol{\theta}_i, \boldsymbol{u}) + r^{(k)} \cdot \boldsymbol{e}$ for $k \in [m]$.

4. Sample $(\boldsymbol{v}, \boldsymbol{\theta}_j)$ from $\{(\boldsymbol{v}^{(k)}, \boldsymbol{\theta}_j^{(k)})\}_{k=1}^m$ with probability proportional to $p_j(\boldsymbol{v}^{(k)}, \boldsymbol{\theta}_j^{(k)}, \mathcal{M}_j \mid \boldsymbol{y})$.

5. Given $(\boldsymbol{v}, \boldsymbol{\theta}_j)$, set $(\boldsymbol{\theta}_i^{(k)}, \boldsymbol{u}^{(k)}) = (\boldsymbol{v}, \boldsymbol{\theta}_j) - r^{(k)} \cdot \boldsymbol{e}$ for $k \in [m]$.

6. Accept $(\boldsymbol{v}, \boldsymbol{\theta}_j)$ with probability α specified as below.

   (a) (Fixed-directional jump)

$$\alpha = \min \left\{ 1, \quad \frac{\sum_{k=1}^m p_j(\boldsymbol{v}^{(k)}, \boldsymbol{\theta}_j^{(k)}, \mathcal{M}_j \mid \boldsymbol{y})}{\sum_{k=1}^m p_i(\boldsymbol{\theta}_i^{(k)}, \boldsymbol{u}^{(k)}, \mathcal{M}_i \mid \boldsymbol{y})} \right\}. \tag{2.22}$$

   (b) (Adaptive-directional jump)

$$\alpha = \min \left\{ 1, \quad \frac{\sum_{k=1}^m p_j(\boldsymbol{v}^{(k)}, \boldsymbol{\theta}_j^{(k)}, \mathcal{M}_j \mid \boldsymbol{y})}{\sum_{k=1}^m p_i(\boldsymbol{\theta}_i^{(k)}, \boldsymbol{u}^{(k)}, \mathcal{M}_i \mid \boldsymbol{y})} \times J((\boldsymbol{\theta}_i, \boldsymbol{u}), (\boldsymbol{v}, \boldsymbol{\theta}_j)) \right\}. \tag{2.23}$$

2.5 Review of Existing Methods with Comparisons

2.5.1 Importance sampling and sequential Monte Carlo

It is known that the performance of importance sampling is determined by how closely the proposal distribution tracks the target distribution. In a good importance sampler, high probability regions of the proposal and target distributions overlap substantially, and the proposal typically has a heavier tail than the target. Otherwise, the variance of the importance sampling estimator can be unacceptably large so that the resulting estimation is misleading. If the dimension of the problem is high, it is generally hard to construct an appropriate proposal distribution.

Figure 2.1b provides a cartoon illustration of the setting that the surrogate distribution has a smaller domain (and thinner tail) compared to the target distribution. In this case, importance sampling is likely to underestimate the normalizing constant (see the caption in Figure 2.1b). This phenomenon is illustrated on two realistic examples in Section 2.6, the Log-Gaussian Cox process and the Bayesian Lasso.

Sequential Monte Carlo (SMC), developed upon importance sampling, employs a sequential scheme to handle the high-dimensionality of the target distribution (Liu et al., 2001; Liu, 2008). More precisely, we first decompose $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_p)$, and construct a sequence of (unnormalized) auxiliary distributions $\eta_1(\theta_1), \eta_2(\theta_1, \theta_2), \ldots, \eta_p(\boldsymbol{\theta})$ sequentially approaching the target distribution, i.e., $\eta_p(\boldsymbol{\theta}) = \gamma(\boldsymbol{\theta})$. We then initialize by drawing n samples, $\theta_1^{(1)}, \cdots, \theta_1^{(n)}$, from a (normalized) proposal distribution $q_1(\theta_1)$ and attach to each with weight $w_1^{(i)} = \eta_1(\theta_1^{(i)})/q_1(\theta_1^{(i)})$. The average weight $\overline{w}_1$ serves as an estimate of the normalizing constant of $\eta_1(\theta_1)$. In the next step, we can either resample the obtained "particles" $\{(\theta_1^{(i)}, w_1^{(i)})\}_{i=1}^n$ with probability proportional to, say $(w_1^{(i)})^\alpha$ with $\alpha \in [0, 1]$, and modify the new weights to $(w_1^{(i)})^{1-\alpha}$, or proceed directly to sample $\theta_2^{(i)}$ from a (normalized)

user-chosen sampling distribution $q_2(\theta_2 \mid \theta_1^{(i)})$. We then update the weights

$$w_2^{(i)} = w_1^{(i)} \times \frac{\eta_2(\theta_1^{(i)}, \theta_2^{(i)})}{\eta_1(\theta_1^{(i)}) q_2(\theta_2^{(i)} \mid \theta_1^{(i)})}.$$

Similarly, $\overline{w}_2$ is an estimate of the normalizing constant of $\eta_2(\theta_1, \theta_2)$. This sequential update is carried out up to step p, and the final average weight $\overline{w}_p$ is an estimate of the normalizing constant Z_γ. It is easy to show that $\overline{w}_p$ is unbiased, if no resampling is involved, and is always consistent (Del Moral, 2004).

We can generalize the above SMC framework to cases where no dimensional changes are involved, that is, $\eta_1, \ldots, \eta_p$ are all defined on the full space of $\boldsymbol{\theta}$. In this case SMC looks very similar to bridge sampling, path sampling and the parallel WL mixture methodL. Recent work has demonstrated its potential in normalizing constant estimation for applications in Bayesian phylogenetics (Wang et al., 2020), nonlinear ordinary differential equation (ODE) models, and positron emission tomography (PET) compartmental models (Zhou et al., 2016). It is generally nontrivial to design an appropriate sequence of auxiliary and sampling distributions. Various proposals have been documented in the literature. (i) The geometric path, $\eta_p(\boldsymbol{\theta}) = \gamma(\boldsymbol{\theta})^{\lambda_p} q(\boldsymbol{\theta})^{1-\lambda_p}$ with $0 = \lambda_0 < \lambda_1 < \cdots < \lambda_p = 1$. (ii) The posterior distribution with partial data (Chopin, 2002). In this case, a common choice for the sampling distributions is some form of prior/posterior predictive distributions. (iii) The path of level sets defined by the likelihood function (Salomone et al., 2018) or specific functions associated with tasks of rare event estimation (Cérou et al., 2012). We note that the selection of the auxiliary distributions in the aforementioned forms can be potentially made automatic using the conditional effective sample size criterion proposed in Zhou et al. (2016).

Another related method is the stepping stone (SS) method (Xie et al., 2011), which has been successfully applied in Bayesian phylogenetics. The stepping stone method takes the prior as the proposal distribution, and employs a geometric sequence of auxiliary distributions. By expanding the ratio

Z_γ/Z_q into a series of telescopic product, i.e., $Z_r/Z_q = \prod_{j=1}^{p} Z_j/Z_{j-1}$, in which Z_j is the normalizing constant of the auxiliary distribution $\eta_j(\boldsymbol{\theta})$, it uses importance sampling to estimate each ratio $r_j = Z_j/Z_{j-1}$ by

$$\widehat{r}_j = \frac{1}{n}\sum_{i=1}^{n} p(\boldsymbol{y} \mid \boldsymbol{\theta}_{ji})^{\lambda_j - \lambda_{j-1}}, \tag{2.24}$$

in which $p(\boldsymbol{y} \mid \boldsymbol{\theta})$ is the likelihood function, and $\boldsymbol{\theta}_{ji}$ is the i^{th} MCMC samples from $\eta_j(\boldsymbol{\theta})$. A generalized stepping stone method is proposed in Fan et al. (2011), which suggests to use a different proposal better tracking the posterior distribution if the prior is too diffuse. An illustration of SMC, the parallel WL mixture method and the stepping stone method on the Log-Gaussian Cox process is given in Section 2.6.1.

2.5.2 Bridge sampling and path sampling

Bridge sampling provides an efficient way of utilizing samples from both the proposal and target distributions. Given the (unnormalized) target $\gamma(\boldsymbol{\theta})$ and proposal $q(\boldsymbol{\theta})$, bridge sampling inserts a bridge distribution $\gamma_{1/2}(\boldsymbol{\theta})$ between $\gamma(\boldsymbol{\theta})$ and $q(\boldsymbol{\theta})$, and estimates the ratio Z_γ/Z_q based on the following identity:

$$r = \frac{Z_\gamma}{Z_q} = \frac{\mathbb{E}_q\left[\gamma_{1/2}(\boldsymbol{\theta})/q(\boldsymbol{\theta})\right]}{\mathbb{E}_\gamma\left[\gamma_{1/2}(\boldsymbol{\theta})/\gamma(\boldsymbol{\theta})\right]}. \tag{2.25}$$

The corresponding bridge sampling estimator is

$$\widehat{r} = \frac{(1/n_q)\sum_{i=0}^{n_q}\gamma_{1/2}(\boldsymbol{\theta}_{qi})/q(\boldsymbol{\theta}_{qi})}{(1/n_\gamma)\sum_{i=0}^{n_\gamma}\gamma_{1/2}(\boldsymbol{\theta}_{\gamma i})/\gamma(\boldsymbol{\theta}_{\gamma i})}, \tag{2.26}$$

in which $\boldsymbol{\theta}_{q1}, \cdots, \boldsymbol{\theta}_{qn_q}$ and $\boldsymbol{\theta}_{\gamma 1}, \cdots, \boldsymbol{\theta}_{\gamma n_\gamma}$ are n_q samples and n_γ samples from $q^\star(\boldsymbol{\theta})$ and $\gamma^\star(\boldsymbol{\theta})$, respectively.

The bridge distribution helps create more connections between the target and the proposal. In addition, since bridge sampling also utilizes samples from the target, it helps resolve the issue of underes-

timating the normalizing constant illustrated in Figure 2.1b and discussed in Section 2.5.1. However, the efficiency of bridge sampling is still sensitive to the "distance" between $q^\star(\boldsymbol{\theta})$ and $\gamma^\star(\boldsymbol{\theta})$. For simplicity, let us assume $n_q = n_\gamma = n$, and consider the optimal bridge $\gamma_{\text{opt}}(\boldsymbol{\theta}) = (q^\star(\boldsymbol{\theta})^{-1} + \gamma^\star(\boldsymbol{\theta})^{-1})^{-1}$ that minimizes the asymptotic variance of $\log \widehat{r}$ under the assumption that all the samples are independent draws. The corresponding optimal asymptotic variance is:

$$\begin{aligned} V_{\text{opt}} &= \frac{2}{n}\left[\left(\int \frac{2q^\star(\boldsymbol{\theta})\gamma^\star(\boldsymbol{\theta})}{q^\star(\boldsymbol{\theta}) + \gamma^\star(\boldsymbol{\theta})} d\boldsymbol{\theta}\right)^{-1} - 1\right] \\ &\geq \frac{2}{n}\left[\left(\int 2\min\{q^\star(\boldsymbol{\theta}), \gamma^\star(\boldsymbol{\theta})\} d\boldsymbol{\theta}\right)^{-1} - 1\right]. \end{aligned} \tag{2.27}$$

We see that the lower bound increases if we push the proposal $q^\star(\boldsymbol{\theta})$ and the target $\gamma^\star(\boldsymbol{\theta})$ further apart. In contrast, with the help of global jumping algorithms such as MTM, the WL mixture method is more robust to this separation issue.

We empirically compare the performance of bridge sampling and the WL mixture method (equiped with MTM) on a 20-dimensional multivariate normal distribution. The target is $N(0, I_{20})$, and the surrogate (proposal) is $N(\mu \times 1_{20}, I_{20})$ with $\mu = 1, 2, 3, 4, 5$. The target has been normalized so that the true log normalizing constant is 0. The fixed jumping direction is $\boldsymbol{e} = \pm(\mu \times 1_{20})$, and we use 8 tries in each multiple-try iteration. For simplicity, we substitute the local MCMC moves around the two mixture components by directly sampling from either the target or the surrogate. We set $n_\gamma = n_q = 5,000$, and run 5,000 iterations for the WL mixture method. The results are summarized in Table 2.1.

Method	$\mu = 1$	$\mu = 2$	$\mu = 3$	$\mu = 4$	$\mu = 5$
WL	$0.00(0.05)$	$0.01(0.04)$	$0.00(0.04)$	$-0.00(0.04)$	$0.01(0.05)$
BS	$-0.00(0.11)$	$0.21(3.15)$	$-0.35(5.43)$	$-1.27(6.93)$	$1.60(7.90)$

Table 2.1: Comparisons of the WL mixture method and bridge sampling for estimating the logarithm of the integral of the multivariate normal density, which is exactly 0 in all cases. The reported values are empirical means and standard deviations (in the bracket) based on 10 independent runs.

We see that the WL mixture method has robust performances for different μ's, whereas bridge sampling performs worse as the target and the proposal distributions become more and more separated. We note that the comparison is not entirely fair because we pre-locate the mode of the target for MTM. Our point is that the WL mixture method should be classified as an MCMC-based method, and behaves very differently from bridge sampling and other importance sampling based methods. The performance of the WL mixture method relies on an efficient strategy to sample from the adaptive mixture distribution $\pi_t^\dagger(\boldsymbol{\theta})$ (see Equation (2.4)), rather than the amount of overlaps between the target and the surrogate distributions.

It is conceivable that bridge sampling can overcome the separation between the target and the proposal by creating multiple bridge distributions. Gelman and Meng (1998) pointed out that when the number of bridge distributions goes to infinity, bridge sampling is equivalent to path sampling. Given a *continuous path* $\{\eta_t(\boldsymbol{\theta})\}$ from $q(\boldsymbol{\theta})$ to $\gamma(\boldsymbol{\theta})$ parameterized by $t \in [0, 1]$ (e.g., the geometric path), path sampling utilizes the identity of thermodynamic integration. That is,

$$\log \frac{Z_\gamma}{Z_q} = \int_0^1 \mathbb{E}_t \left[\frac{d}{dt} \log \eta_t(\boldsymbol{\theta}) \right] dt,$$

in which $\mathbb{E}_t$ is taken with respect to $\eta_t(\boldsymbol{\theta})$. The corresponding path sampling estimator can be constructed using numerical integration over t with samples from $\eta_t(\boldsymbol{\theta})$.

2.5.3 Chib's method

The method proposed by Chib (1995) is effective for estimating normalizing constants for a class of Bayesian models and has been widely adopted. For any $\boldsymbol{\theta}^\star$ such that $p(\boldsymbol{\theta}^* \mid \boldsymbol{y}) > 0$, we have

$$\log Z_\gamma = \log p(\boldsymbol{y} \mid \boldsymbol{\theta}^\star) + \log p(\boldsymbol{\theta}^\star) - \log \gamma^\star(\boldsymbol{\theta}^\star \mid \boldsymbol{y}), \tag{2.28}$$

in which $p(\boldsymbol{y} \mid \boldsymbol{\theta}^\star)$ and $p(\boldsymbol{\theta}^\star)$ are the likelihood function and the prior evaluated at $\boldsymbol{\theta}^\star$, respectively. Consequently, if we can estimate well the normalized posterior density at $\boldsymbol{\theta}^\star$, that is, $\gamma^\star(\boldsymbol{\theta}^\star \mid \boldsymbol{y})$, we have an estimate of the normalizing constant Z_γ.

Chib (1995) showed that this is feasible using Gibbs outputs. For example, suppose $\boldsymbol{\theta}$ can be decomposed into two blocks, $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$, and we have an efficient Gibbs sampler in hand, targeting the posterior distribution $\gamma(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \mid \boldsymbol{y})$, which iteratively samples from the two conditional distributions $\gamma^\star(\boldsymbol{\theta}_1 \mid \boldsymbol{\theta}_2, \boldsymbol{y})$ and $\gamma^\star(\boldsymbol{\theta}_2 \mid \boldsymbol{\theta}_1, \boldsymbol{y})$. We further assume that we can evaluate the two conditional distributions exactly. With the Gibbs outputs $\{(\boldsymbol{\theta}_1^{(i)}, \boldsymbol{\theta}_2^{(i)})\}_{i=1}^n$, we can estimate the normalized posterior density at $\boldsymbol{\theta}^\star$ as below,

$$\widehat{\gamma}^\star(\boldsymbol{\theta}^\star \mid \boldsymbol{y}) = \widehat{\gamma}^\star(\boldsymbol{\theta}_1^\star \mid \boldsymbol{y})\gamma^\star(\boldsymbol{\theta}_2^\star \mid \boldsymbol{\theta}_1^\star, \boldsymbol{y}) = \left[\frac{1}{n}\sum_{i=1}^n \gamma^\star(\boldsymbol{\theta}_1^\star \mid \boldsymbol{\theta}_2^{(i)}, \boldsymbol{y})\right] \gamma^\star(\boldsymbol{\theta}_2^\star \mid \boldsymbol{\theta}_1^\star, \boldsymbol{y}), \quad (2.29)$$

which utilizes the identity that $\gamma^\star(\boldsymbol{\theta}_1 \mid \boldsymbol{y}) = \int \gamma^\star(\boldsymbol{\theta}_1 \mid \boldsymbol{\theta}_2, \boldsymbol{y})\gamma^\star(\boldsymbol{\theta}_2 \mid \boldsymbol{y}) d\boldsymbol{\theta}_2$. For a better statistical efficiency, it is recommended to select $\boldsymbol{\theta}^\star$ close to the posterior mode. The above scheme can be generalized to cases in which $\boldsymbol{\theta}$ is decomposed into an arbitrary number of blocks, and also cases with missing data (Chib, 1995).

We see that Chib's method is particularly useful and easy to implement when we have an efficient Gibbs sampler with all the conditional distributions being tractable. In Section 2.6.2, we compare Chib's method and the WL mixture method on the Bayesian Lasso example, in which we indeed have a closed form Gibbs sampler. The performances of the two methods are comparable. Chib and Jeliazkov (2001) extended the method to the setting with intractable conditional densities, but its applicability can still be limited if we encounter other types of MCMC algorithms such as Hamiltonian Monte Carlo (HMC), Metropolis-adjusted Langevin algorithm (MALA), etc. In contrast, a major advantage of the WL mixture method is that it can be built on any type of MCMC samplers, and is reasonably easy to implement.

2.6 Illustrations

2.6.1 Log-Gaussian Cox process

We consider estimating the normalizing constant of a Log-Gaussian Cox process on the pine forest data set studied in Penttinen et al. (1992) and Stoyan and Stoyan (1994). The data contains the locations of 126 Scots pine saplings on a 10×10 m^2 square (see Figure 10(a) in Møller et al. (1998)). We first standardize the locations into unit square and then discretize the unit square into an $M \times M$ regular grid. Let $\boldsymbol{y} = (y_m)_{m \in [M]^2}$ denote the number of pine saplings in each grid cell, and let $= (\lambda_m)_{m \in [M]^2}$ denote the latent intensity process. We assume the following model:

$$[y_m \mid \lambda_m] \sim \text{Poisson}\left(a\lambda_m\right),$$

in which $a = M^{-2}$ is the area of each grid cell. The dimension of is M^2, and in this example, we test out $M = 10,\ 20,\ 30$, thus the dimension of the problem is $100,\ 400,\ 900$, respectively. We transform $\boldsymbol{\theta} = \log$ so that all the parameters are defined on $\mathbb{R}$. We specify a Gaussian process prior given as below with constant mean μ_0 and exponential covariance function on $\boldsymbol{\theta} = (\theta_m)_{m \in [M]^2}$,

$$\Sigma_0\left(m, n\right) = \sigma^2 \exp\left(-\frac{1}{M\beta}\left|m - n\right|\right), \quad m, n \in [M]^2,$$

where we follow the same parameters setting in Møller et al. (1998): $\sigma^2 = 1.91$, $\beta = 1/33$ and $\mu_0 = \log(126) - \sigma^2/2$. The Poisson likelihood is as follows:

$$L\left(\boldsymbol{\theta} \mid \boldsymbol{y}\right) = \prod_{m \in [M]^2} \exp\left(\theta_m y_m - a \exp\left(\theta_m\right)\right),$$

thus the unnormalized posterior distribution is $\gamma(\boldsymbol{\theta} \mid \boldsymbol{y}) = N(\boldsymbol{\theta}; \mu_0, \Sigma_0) L(\boldsymbol{\theta} \mid \boldsymbol{y})$. An approximate mode $\widehat{\boldsymbol{\theta}}$ of $\gamma(\boldsymbol{\theta} \mid \boldsymbol{y})$ is obtained using the Newton-Raphson method.

With this example, we compare the performances of two classes of methods: (i) single-step methods including the WL mixture method and importance sampling; (ii) multiple-step methods including an adaptive SMC algorithm, the parallel WL (PWL) mixture method, and the stepping stone (SS) method proposed in (Xie et al., 2011). The multiple-step methods insert a sequence of auxiliary distributions between the surrogate (proposal) and the posterior, whereas the single-step methods not.

For the single-step methods, we use $N(\boldsymbol{\mu}_q, \sigma_q^2 I)$ as the surrogate, in which $\boldsymbol{\mu}_q = \widehat{\boldsymbol{\theta}}$, and $\sigma_q = 1.0,\ 1.2,\ 1.3$ for $M = 10,\ 20,\ 30$, respectively, in order to approximately match the marginal posterior standard deviations. For the WL mixture method, we use HMC local moves around the mixture component γ. The gradient of the log likelihood is

$$\nabla \log L(\boldsymbol{\theta} \mid \boldsymbol{y}) = \boldsymbol{y} - a \exp(\boldsymbol{\theta}),$$

and the HMC kernel contains 10 leapfrog steps with step size 0.25. We run in total $S = 5 \times 10^4$ iterations for $M = 10$, and $S = 1 \times 10^5$ iterations for $M = 20,\ 30$, with $b = S/2$. In addition, we test out different thresholds c (from 0.10 to 0.30 incremented by 0.05) used in the flat histogram criterion. For importance sampling, we use 1×10^6 samples to match the computation time of the WL mixture method (see Table 2.2).

The algorithmic settings of the mutliple-step methods are described below. The SMC algorithm is detailed in Algorithm 14 in the Appendix B. The proposal distribution is the prior distribution $N(\boldsymbol{\theta}; \mu_0, \Sigma_0)$. We use 500 particles, and run 10 HMC rejuvenation steps (step 2(g) in Algorithm 14) for each auxiliary distribution to diversify the particles. The conditional effective sample size (CESS) adaptation criterion (Zhou et al., 2016) is set to be $\kappa = 0.9$, and a systematic resampling step is carried out if the normalized effective sample size (ESS) drops below 0.5. On average there are 35, 42 and 45

intermediate steps for $M = 10, 20, 30$, respectively. For both the parallel WL mixture method and the stepping stone method, we employ the same sequence of auxiliary distributions adaptively selected by the SMC algorithm, and run $S = 1.5 \times 10^3$ iterations using the aforementioned HMC kernel invariant to each auxiliary distribution. The computation time for the three methods are comparable (see Table 2.3).

The results are summarized in Tables 2.2 and 2.3. We find that all the three multiple-step methods, as well as the WL mixture method, produced similar estimates of the log normalizing constant under all settings, whereas importance sampling underestimated the log normalizing constant for both $M = 20, 30$, consistent with our discussion in Section 2.5.1. The SMC algorithm is the most accurate method on this example, and the (parallel) WL mixture method also performed reasonably well, having slightly larger standard deviations compared to the SMC algorithm. Figure B.1 in the Appendix B shows that the performance of the WL mixture method is robust to the choice of the threshold c used in the flat histogram criterion in the region $[0.1, 0.3]$. For this and the Bayesian Lasso example in the next section, we also compare the convergence speeds of the standard and the accelerated WL algorithms (see Section 2.3.3). Figure 2.2 shows that the accelerated algorithm converges much faster than the standard one.

Log normalizing constant estimates			
Dimension	100	400	900
WL	474.4(0.1)	490.7(0.3)	496.6(0.6)
IS	474.1(0.2)	487.1(1.0)	476.0(1.3)
Computation time (second)			
Dimension	100	400	900
WL	15.5(0.2)	157.8(13.4)	985.4(60.3)
IS	17.4(0.4)	150.5(12.6)	929.8(55.2)

Table 2.2: Log normalizing constant estimates of the Log-Gaussian-Cox process using single-step methods. WL and IS refer to the WL mixture method and importance sampling, respectively. The reported values are empirical means and standard deviations (in the bracket) based on 10 independent runs.

Log normalizing constant estimates			
Dimension	100	400	900
SMC	474.4(0.1)	490.7(0.1)	497.6(0.1)
PWL	474.6(0.2)	490.7(0.3)	497.3(0.4)
Stepping stone	474.6(0.1)	491.9(0.2)	502.5(0.8)
Computation time (second)			
Dimension	100	400	900
SMC	21.6(0.7)	212.2(17.9)	1054.2(66.2)
PWL	26.9(0.8)	189.8(15.4)	1123.3(89.1)
Stepping stone	20.5(0.5)	176.7(14.6)	1055.2(70.4)

Table 2.3: Log normalizing constant estimates of the Log-Gaussian-Cox process using multiple-step methods. SMC, PWL and Stepping stone refer to the sequential Monte Carlo method, the parallel WL mixture method, and the stepping stone method (Xie et al., 2011), respectively. The reported time for PWL and the stepping stone method is the computation time without parallelization. The reported values are empirical means and standard deviations (in the bracket) based on 10 independent runs.

2.6.2 Hyper-parameter determination for Bayesian Lasso

We consider the Bayesian Lasso method proposed in Park and Casella (2008), which assumes a hierarchical prior on the linear regression coefficients so that the posterior mode corresponds to the Lasso estimator (Tibshirani, 1996). Given a centered and standardized $n \times p$ design matrix X, the response vector $\boldsymbol{y}$ follows $N(X\boldsymbol{\beta}, \sigma^2 I_n)$. Following Park and Casella (2008), we specify a prior $N(0_p, \sigma^2 D_\tau)$ on $\boldsymbol{\beta}$, where D_τ is a diagonal matrix $\mathrm{diag}(\tau_1^2, \cdots, \tau_p^2)$. Besides, we specify an independent hyper-prior $\mathrm{Exp}(\lambda^2/2)$ on τ_j^2 for $j \in [p]$, and specify an improper prior $p(\sigma^2) \propto 1/\sigma^2$ on σ^2. This completes the full model specification and the unnormalized posterior distribution is given by

$$\gamma(\boldsymbol{\beta}, , \sigma^2 \mid X, \boldsymbol{y}) = \frac{1}{\sigma^2} N(\boldsymbol{y}; X\boldsymbol{\beta}, \sigma^2 I_n) \prod_{j=1}^{p} \mathrm{Exp}\left(\tau_j^2 \mid \lambda^2/2\right).$$

We transform the parameters $\eta_j = \log \tau_j^2$ for $j \in [p]$ and $\xi = \log \sigma^2$ so that all the parameters are defined on $\mathbb{R}$.

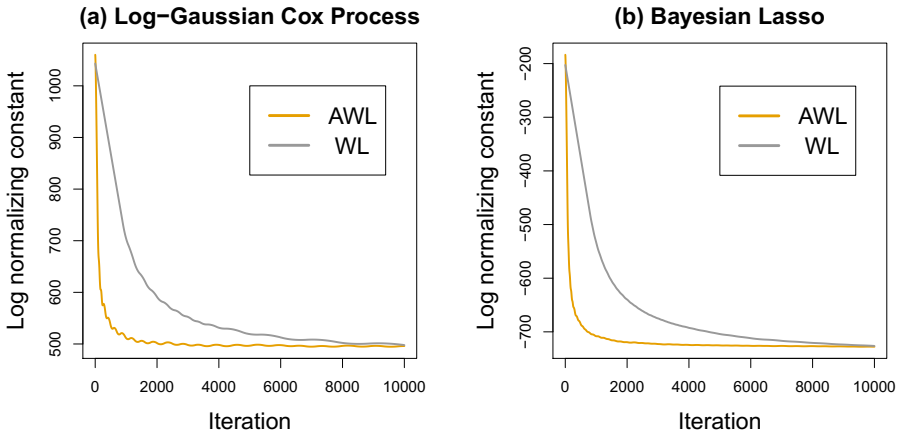


Figure 2.2: Demonstration of the accelerated Wang-Landau algorithm. AWL and WL refer to the accelerated Wang-Landau algorithm and the standard Wang-Landau algorithm, respectively. (a) The Log-Gaussian Cox process discussed in Section 2.6.1 with $M = 30$. (b) The Bayesian Lasso example discussed in Section 2.6.2 with SNR = 0.1 and $\lambda = 20$.

In this example, we simulate the data set as in Yang et al. (2016). We define

$$\boldsymbol{\beta}^\star = \text{SNR}\sqrt{\sigma_0^2 \frac{\log p}{n}}(2, -3, 2, 2, -3, 3, -2, 3, -2, 3, 0, \cdots, 0)^\intercal \in \mathbb{R}^p$$

with $p = 100$, $n = 500$, SNR $\in \{0.1, 1, 3\}$ (signal-to-noise ratio), and $\sigma_0^2 = 1$. The dimension of the posterior distribution is $2 \times p + 1 = 201$. The design matrix X is generated from a centered multivariate normal distribution with covariance matrix $\Sigma_{ij} = \exp(-|i-j|)$. The response variable $\boldsymbol{y}$ is generated from $N(X\boldsymbol{\beta}^\star, \sigma_0^2 I_n)$. The task is to estimate the marginal likelihood of data for a set of regularization parameters $\lambda \in \{5, 10, 15, 20\}$ under different SNRs.

We compare the WL mixture method, Chib's method and importance sampling in this example. The surrogate distribution used in the WL mixture method, which is also the proposal distribution used in importance sampling, is constructed using the variational approximation discussed in Section 2.3.4. We consider the Normal mean-field variational family where $q(\beta_j)$ is $N(m_j, s_j^2)$, $q(\eta_j)$ is $N(\phi_j, \zeta_j^2)$, and $q(\xi)$ is $N(u, v^2)$. The CAVI updates are summarized in Section B.2.2 in the Ap-

pendix B. For the WL mixture method, the Gibbs move proposed in Park and Casella (2008) is used to move around the mixture component γ. For completeness, we detail below the conditional posterior distributions required by the Gibbs sampler:

$$
\begin{aligned}
&[\boldsymbol{\beta} \mid \text{rest}] \sim N(C_\tau^{-1} X^\intercal \boldsymbol{y}, \sigma^2 C_\tau^{-1}), \quad C_\tau = X^\intercal X + D_\tau^{-1}, \\
&[\tau_j^{-2} \mid \text{rest}] \sim \text{Inverse-Gaussian}(\lambda\sigma/|\beta_j|, \lambda^2), \\
&[\sigma^2 \mid \text{rest}] \sim \text{Inv-}\chi^2 \left(n + p, \left(||\boldsymbol{y} - X\boldsymbol{\beta}||_2^2 + \boldsymbol{\beta}^\intercal D_\tau^{-1} \boldsymbol{\beta}\right)/(n+p)\right).
\end{aligned}
$$

We run a total of $S = 1 \times 10^4$ iterations, and set $b = 2,000$. For importance sampling, we use 5×10^5 samples. For Chib's method, the parameters are cut into three blocks, $\boldsymbol{\beta}$, ^{2} and σ^2, and we run the Gibbs sampler for 5×10^3 iterations and burn-in the first 10% samples. The computation time for the three methods are comparable (see the caption in Figure 2.3),

The results are summarized in Figure 2.3. We see that under all settings, the WL mixture method and Chib's method produced similar and stable estimates of the log normalizing constant, whereas importance sampling underestimated the log normalizing constant. The regularization parameter that maximizes the marginal likelihood of data are $\lambda = 10, 20, 20$ for SNR $= 3, 1, 0.1$, respectively, which corresponds to our intuition that it requires more regularization for estimating the regression coefficients when there exists larger noises in the data.

2.6.3 Logistic regression

We consider a Bayesian logistic regression model for the classic German credit data set (available from the UCI repository (Frank and Asuncion, 2011)). There are in total $n = 1,000$ personal records in the data set. For each records, there are 24 associated attributes including sex, age, and credit amount. The binary response variable $\boldsymbol{y}$ indicates good or bad credit risks. Let $X_{n \times p}$ be the design matrix after we standardize all the predictors. In particular, we include an intercept and all pairwise interactions.

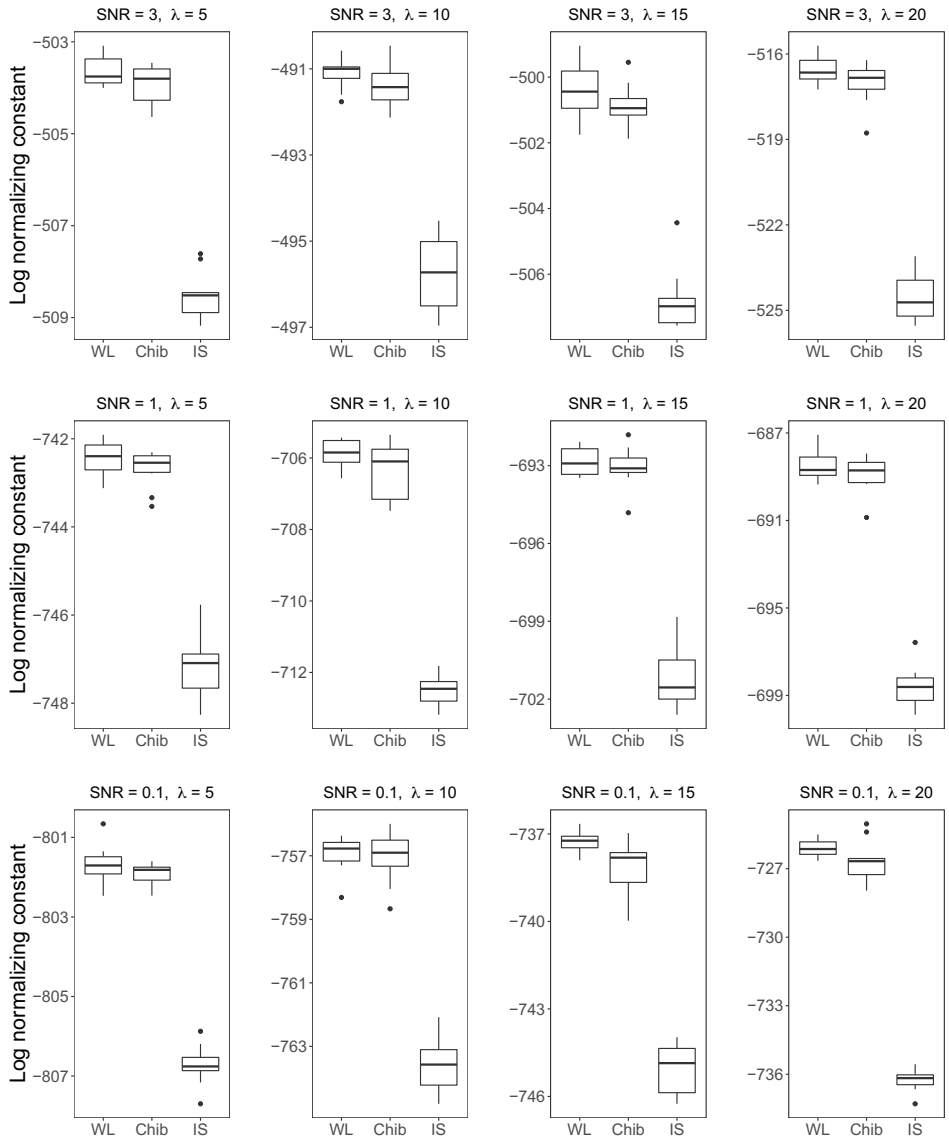


Figure 2.3: Results summary of the Bayesian Lasso example. WL, Chib and IS refer to the WL mixture method, Chib's method and importance sampling, respectively. The box plots are based on 10 independent runs of the algorithms. The computation time for the WL mixture method, Chib's method, and importance sampling are 32.1 (± 0.6) seconds, 43.5 (± 2.0) seconds and 47.1 (± 1.2) seconds, respectively.

The dimension of the problem is $p = 24 + 24 \times 23/2 + 1 = 301$. We consider the following logistic regression model:

$$\mathbb{P}(y_i = 1 \mid \boldsymbol{x}_i, \alpha, \boldsymbol{\beta}) = \frac{\exp(\alpha + \boldsymbol{\beta}^\intercal \boldsymbol{x}_i)}{1 + \exp(\alpha + \boldsymbol{\beta}^\intercal \boldsymbol{x}_i)}, \tag{2.30}$$

in which $y_i \in \{0, 1\}$, $\boldsymbol{x}_i \in \mathbb{R}^{300}$, $\alpha \in \mathbb{R}$, $\boldsymbol{\beta} \in \mathbb{R}^{300}$, $i \in [n]$. All the observations are assumed to be independent. We set up similar priors on the parameters as in Heng and Jacob (2019),

$$[\alpha \mid s^2] \sim N\left(0, s^2\right), \quad [\boldsymbol{\beta} \mid s^2] \sim N\left(0_{300}, s^2 I_{300}\right), \quad s^2 \sim \text{Exp}(\lambda),$$

with $\lambda \in \{0.01, 1.00\}$. This leads to the unnormalized posterior distribution:

$$\begin{aligned}\gamma(\alpha, \boldsymbol{\beta}, s^2 \mid \boldsymbol{y}, X) &= p(\alpha, \boldsymbol{\beta} \mid s^2) p(s^2) \prod_{i=1}^{n} p(y_i \mid \boldsymbol{x}_i) \\ &= \lambda e^{-\lambda s^2} N(\alpha; 0, s^2) \prod_{j=1}^{300} N(\boldsymbol{\beta}_j; 0, s^2) \prod_{i=1}^{n} p_i^{y_i} (1 - p_i)^{1 - y_i},\end{aligned}$$

in which $p_i = \mathbb{P}(y_i = 1 \mid \boldsymbol{x}_i, \alpha, \boldsymbol{\beta})$ as defined in Equation (2.30). We transform s^2 to the logarithmic scale $\log s^2$ so that all the parameters are defined on $\mathbb{R}$. The task is to estimate the log normalizing constant of γ.

We compare the WL mixture method and bridging sampling (BS). We use the same surrogate (proposal) distribution, constructed by the Laplace approximation method detailed below, for both methods. We first run an HMC algorithm to obtain posterior samples from $\gamma(\alpha, \boldsymbol{\beta}, \log s^2 \mid \boldsymbol{y}, X)$. Then, we fit a multivariate normal distribution on the posterior samples, and choose it as the surrogate distribution. Each HMC step contains 10 leapfrog steps with step size adjusted to be 0.03. For bridge sampling, we use the R package *bridgesampling* (Gronau et al., 2017), and obtain n samples from the posterior using RStan (Stan Development Team, 2019). Correspondingly, we run $2 \times n$ iterations for the WL mixture method so that approximately we also use n samples from the posterior.

For this example, we tested out $n = 1000,\ 1500,\ 2000,\ 2500$ for $\lambda \in \{0.01,\ 1.00\}$. The results are summarized in Figure 2.4. We see that the WL mixture method has a much better estimation efficiency compared to bridge sampling. Bridge sampling approaches to the vicinity of the correct estimate only after 2,500 iterations/samples for both cases $\lambda = 0.01$ and $\lambda = 1.00$.

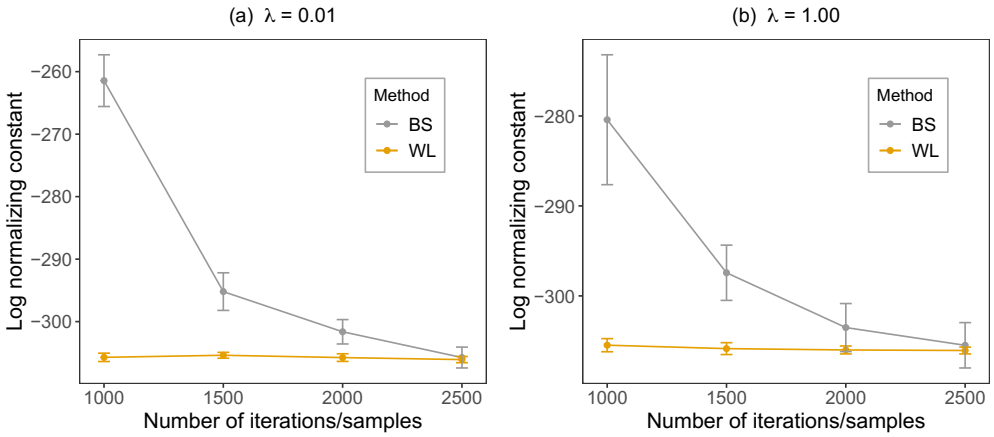


Figure 2.4: Comparison between the WL mixture method and bridge sampling (BS). For bridge sampling, the x-axis represents the number of samples we draw from the posterior and the proposal distributions. For the WL mixture method, the x-axis represents half of the total number of iterations we run (see the second to last paragraph in this section). The error bars represent the standard deviations of the log normalizing constant estimates based on 10 independent runs.

2.6.4 g-Prior variable selection

We compare the performance of MTM-RJMCMC proposed in Section 2.4.2 and that of a standard birth-and-death RJMCMC (BD-RJMCMC, detailed below) in the setting of Bayesian variable selection for the pollution data (McDonald and Schwing, 1973). The response variable $\boldsymbol{y}$ is the age-adjusted mortality rate obtained for the years 1959-1961 in 201 standard metropolitan statistical areas. There are in total $n = 60$ observations. The design matrix X contains $p = 15$ predictors including the average annual precipitation, the average temperature in January and July, and the population per household. We consider the standard linear model assuming that $[\boldsymbol{y} \mid X, \boldsymbol{\beta}, \sigma^2]$ follows

$N\left(X\boldsymbol{\beta}, \sigma^2 I\right)$. We center the response variable $\boldsymbol{y}$ so that there is no intercept in the model, and standardize each predictor in the design matrix X.

Let $\boldsymbol{\gamma} \in \{0,1\}^p$ be the binary indicator such that $\gamma_j = 1$ represents that the predictor X_j is selected into the model. We employ the g-prior on parameters $\boldsymbol{\beta}$:

$$\left[\boldsymbol{\beta}_{\boldsymbol{\gamma}} \mid \boldsymbol{\gamma}, \sigma^2\right] \sim N\left(0_{\boldsymbol{\gamma}}, g\sigma^2\left(X_{\boldsymbol{\gamma}}^{\intercal} X_{\boldsymbol{\gamma}}\right)^{-1}\right).$$

The g-prior enables us to integrate out $\boldsymbol{\beta}$ so that we can obtain the marginal distribution of $\boldsymbol{\gamma}$:

$$p(\boldsymbol{\gamma} \mid \boldsymbol{y}, X) \propto (g+1)^{-q_{\boldsymbol{\gamma}}/2}\left[\boldsymbol{y}^{\intercal}\boldsymbol{y} - \frac{g}{g+1}\boldsymbol{y}^{\intercal} X_{\boldsymbol{\gamma}}\left(X_{\boldsymbol{\gamma}}^{\intercal} X_{\boldsymbol{\gamma}}\right)^{-1} X_{\boldsymbol{\gamma}}\boldsymbol{y}\right]^{-n/2}, \tag{2.31}$$

in which $q_{\boldsymbol{\gamma}}$ denotes the number of selected predictors. We see that g controls the sparsity of the model, and a larger g induces a sparser model. For σ^2, we use a noninformative prior $p\left(\sigma^2\right) \propto 1/\sigma^2$. This completes the full model specification. The task is to estimate the marginal probability of each predictor being selected. The ground truth is obtained by enumerating all 32,768 possible $\boldsymbol{\gamma}$ and calculating the marginal probability using Equation (2.31). To compare MTM-RJMCMC and BD-RJMCMC, we pretend that we do not have the privilege to integrate out $\boldsymbol{\beta}$, thus we will sample from the trans-dimensional joint posterior distribution $p\left(\boldsymbol{\beta}_{\boldsymbol{\gamma}}, \boldsymbol{\gamma}, \sigma^2 \mid \boldsymbol{y}, X\right)$.

We use the Gibbs sampler to iterate between the following conditional distributions:

$$\left[\boldsymbol{\beta}_{\boldsymbol{\gamma}}, \boldsymbol{\gamma} \mid \sigma^2, \boldsymbol{y}, X\right] \sim \left(2\pi g\sigma^2\right)^{-\frac{q_{\boldsymbol{\gamma}}}{2}}\left|X_{\boldsymbol{\gamma}}^{\intercal} X_{\boldsymbol{\gamma}}\right|^{\frac{1}{2}} \exp\left(-\frac{1}{2\sigma^2}\left[\frac{g+1}{g}\left|\left|X_{\boldsymbol{\gamma}}\boldsymbol{\beta}_{\boldsymbol{\gamma}}\right|\right|^2 - 2\boldsymbol{\beta}_{\boldsymbol{\gamma}}^{\intercal} X_{\boldsymbol{\gamma}}^{\intercal}\boldsymbol{y}\right]\right),$$

$$\left[\sigma^2 \mid \boldsymbol{y}, X, \boldsymbol{\beta}_{\boldsymbol{\gamma}}, \boldsymbol{\gamma}\right] \sim \text{Inv-Gamma}\left(\frac{n+q_{\boldsymbol{\gamma}}}{2}, \frac{1}{2}\left[\frac{1}{g}\left|\left|X_{\boldsymbol{\gamma}}\boldsymbol{\beta}_{\boldsymbol{\gamma}}\right|\right|^2 + \left|\left|\boldsymbol{y} - X_{\boldsymbol{\gamma}}\boldsymbol{\beta}_{\boldsymbol{\gamma}}\right|\right|^2\right]\right).$$

Given $\boldsymbol{\gamma}_t$, the jumping rule for $\boldsymbol{\gamma}_{t+1}$ as described below is the same for both algorithms. We first flip a coin to decide whether we stay in the current model ($\boldsymbol{\gamma}_{t+1} = \boldsymbol{\gamma}_t$) or move to a different model ($\boldsymbol{\gamma}_{t+1} \neq \boldsymbol{\gamma}_t$). If we choose to leave the current model (a trans-dimensional move), we randomly move

into a higher dimension (add a predictor) or move into a lower dimension (exclude a predictor) with equal probability 0.5. When the chain is at the boundary (q_γ is 1 or 15), the proposal going out of the range is automatically rejected.

Given $\boldsymbol{\gamma}_{t+1}$, for the within-dimensional move ($\boldsymbol{\gamma}_{t+1} = \boldsymbol{\gamma}_t$), we implement an Metropolis-within-Gibbs step, with proposal distribution $N(0, 0.5^2)$, to sequentially update each coordinate of $\boldsymbol{\beta}_{\gamma_t}$. For the trans-dimensional move, MTM-RJMCMC and BD-RJMCMC use different proposals. For MTM-RJMCMC, we follow the fixed-directional jumping mechanism detailed in Algorithm 8. Since we only add or remove one predictor in each trans-dimensional move, the algorithm requires only one auxiliary variable. We choose the auxiliary distribution to be $N(0, 1)$. We sample the jumping distance r from $N(1, 1)$, and set the number of tries to be $m = 5$. For BD-RJMCMC, if we choose to add a predictor, we propose it from $N\left(0, 0.5^2\right)$. We run 5×10^4 iterations for MTM-RJMCMC and 1.5×10^5 iterations for BD-RJMCMC so that the computation time for the two algorithms are comparable (see the caption in Figure 2.5). For both algorithms, we burn-in the first 10% samples.

The estimation results for $g = \exp(10)$ and $g = \exp(15)$, respectively, are summarized in Figure 2.5. We see that MTM-RJMCMC produced more accurate estimation results than BD-RJMCMC. In particular, we see that BD-RJMCMC might have been stuck in a local mode thus mistakenly selected two wrong predictors X_{12} and X_{13}. Intuitively, we see that the directional jumping in MTM-RJMCMC is much more informative than the blind proposal used in BD-RJMCMC, thus preventing the algorithm from getting stuck in local modes.

2.7 Concluding Remarks

We have described a general strategy to construct a mixture of the unnormalized posterior distribution and a surrogate distribution with a known normalizing constant to estimate the model likelihood. Such a mixture formulation allows us to use the generalized WL algorithm and the MTM machinery

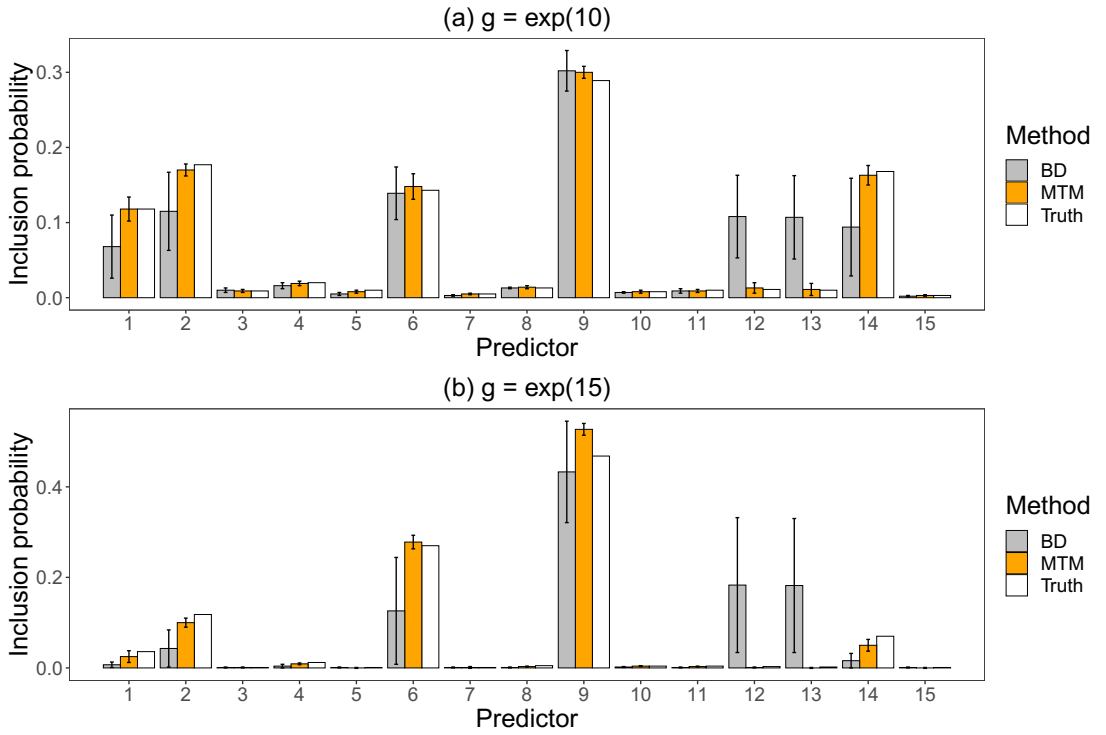


Figure 2.5: Estimates of the marginal probability of each predictor being selected. The bar plots are based on 10 independent runs of both algorithms. The computation time for MTM-RJMCMC and BD-RJMCMC are 19.0 (± 1.0) seconds and 29.8 (± 1.2) seconds, respectively.

for fast MCMC mixing and accurate estimation of the unknown normalizing constant. We have also designed acceleration schemes to further improve its performance.

By efficiently jumping back and forth between the posterior and the surrogate distributions, possibly with the help of mode jumping algorithms such as MTM, the performance of the WL mixture method is less sensitive to the potential separation between the posterior and the surrogate distributions compared to importance sampling based methods. The WL mixture method also has more general applicability compared to Chib's method, when the sampler of the posterior involves more sophisticated MCMC steps beyond the closed-form Gibbs sampler (i.e., all conditional distributions are easy to sample from) or standard Metropolis-Hasting algorithms. In addition, the WL mixture method re-

quires less effort in delicate tuning in its implementation compared to other advanced methods such as path sampling, reversible-jump MCMC, and sequential Monte Carlo methods.

There are several future directions that we would like to follow. First, although we have shown the power of the WL mixture method, a rigorous theoretical framework is required to better understand the nature of the method. Second, instead of mixing the posterior with a single surrogate distribution, a multiple-component mixture formulation can be considered. Third, although the intuitive idea of first using some deterministic algorithm to find modes and then conducting MCMC to do mode jumping has been around, an efficient way of achieving the intended goal has not been formulated precisely. Our proposed MTM-enhanced jumping strategy, together with the WL weight adjustment, can help achieve the goal. It is particularly useful to identify some specific classes of models where this general methodology is straightforward and effective to apply.

3

False Discovery Rate Control via Data Splitting

Contribution This chapter is based on a paper (Dai et al., 2020) jointly with Prof. Jun S. Liu, Buyu Lin and Xin Xing.

3.1 Abstract

Selecting relevant features associated with a given response variable is an important issue in many scientific fields. Quantifying quality and uncertainty of the selection via the false discovery rate (FDR) control has been of recent interest. This paper introduces a way of using data-splitting strategies to asymptotically control FDR for various feature selection techniques while maintaining a high power. For each feature, the method estimates two independent significance coefficients via data splitting and constructs a contrast statistic. The FDR control is achieved by taking advantage of the statistic's property that, for any null feature, its sampling distribution is (asymptotically) symmetric about 0. We further propose a strategy to aggregate multiple data splits (MDS) to stabilize the selection result and boost the power. Interestingly, this multiple data-splitting approach appears capable of overcoming the power loss caused by data splitting with FDR still under control. The proposed framework is applicable to canonical statistical models including the linear model, the generalized linear model, the Gaussian graphical model, and the deep neural network. Simulation results, as well as a real data application, show that the proposed approaches, especially the multiple data-splitting strategy, control FDR well and are often more powerful than existing methods including the Benjamini-Hochberg procedure and the knockoff filter.

3.2 Introduction

3.2.1 Motivation for the FDR control in regression models

Scientific researchers in the current big data era often have the privilege of collecting or accessing a large number of explanatory features targeting a specific response variable. For instance, population geneticists often need to profile thousands of single nucleotide polymorphisms (SNPs) in the genome-wide association study. A ubiquitous belief is that the response variable only depends on a small frac-

tion of the collected features. Therefore, researchers are of primary interest to identify those relevant features, so that the computability of the downstream analysis, the reproducibility of the reported results, and the interpretability of the scientific findings can be largely enhanced. Throughout the article, we denote the explanatory features as $(X_1, \cdots, X_p)$, with p being potentially large, and denote the response variable as y. We remark that this paper is presented in the context of feature selection (regression models), although all the methodological development can be possibly adapted to solve general multiple testing problems.

Many methodological contributions to the feature selection problem have been made by statisticians, including but not limited to stepwise regression, Lasso (Tibshirani, 1996), Dantzig selector (Candes and Tao, 2007), SCAD (Fan and Li, 2001), and some Bayesian methods (O'Hara and Sillanpää, 2009). A desired property of the selection procedure is the capability of controlling the number of false positives, which can be mathematically calibrated by the false discovery rate (FDR) (Benjamini and Hochberg, 1995) defined as below,

$$\text{FDR} = \mathbb{E}[\text{FDP}], \quad \text{FDP} = \frac{\#\{j : j \in S_0,\ j \in \widehat{S}\}}{\#\{j \in \widehat{S}\} \vee 1}, \tag{3.1}$$

where S_0 denotes the set of null features (irrelevant features), $\widehat{S}$ denotes the set of selected features, and FDP refers to the false discovery proportion. The expectation is taken with respect to the randomness both in the data and in the selection procedure if it is not deterministic.

The first class of approaches is the Benjamin-Hochberg (BHq) procedure (Benjamini and Hochberg, 1995) and its extension, the Benjamin-Yekutieli (BYq) procedure (Benjamini and Yekutieli, 2001). BHq and BYq are only applicable when valid p-values are available for each feature. BHq guarantees an exact FDR control when all the p-values are independent, while BYq extends its applicability to the settings with dependent p-values. A further generalization of BHq, which is commonly referred to as the q-value approach, is detailed in Storey et al. (2004).

The second class of approaches is the knockoff filter, including the fixed-design knockoff filter (Barber and Candès, 2015) and the model-X knockoff filter (Candes et al., 2018), which manages to control FDR by creating "knockoff" features in a similar spirit as the spike-ins in biological experiments. The knockoff filter does not require calculating individual p-values, and can be applied to fairly general settings without having to know the underlying true relationship between the response variable and the explanatory features. Further developments of the knockoff filter include the multilayer knockoff filter (Katsevich and Sabatti, 2019) and DeepPINK (Lu et al., 2018). More detailed discussions and comparisons of these methods are postponed to Section 3.3.4 after we introduce our method.

The third approach is the recently proposed Gaussian mirror method (Xin et al., 2019). Its essential idea is to perturb the features one by one and examine the corresponding impact. Specifically, for each feature X_j, the method creates a pair of perturbed "mirror variables", $X_j^+ = X_j + c_j Z_j$ and $X_j^- = X_j - c_j Z_j$, where c_j is an adjustable scalar and Z_j follows $N(0,1)$ independently, and constructs a statistic by contrasting the regression coefficients of the mirror variables obtained by the ordinary least squares (OLS) or Lasso. Their constructed statistic satisfies the property that its sampling distribution is symmetric about 0 if the underlying feature is a null feature, which is crucial to guarantee its asymptotic FDR control.

3.2.2 A review of data-splitting methods for the FDR control

We focus here on related data-splitting methods applicable to multiple testing problems (feature selection). Other applications of the data-splitting strategy include evaluating statistical predictions (cross validation) (Stone, 1974) and selecting efficient test statistics (Moran, 1973; Cox, 1975). In high-dimensional inference, a common practice of data splitting is to reduce the dimension of the problem. For linear models, two notable contributions are made by Wasserman and Roeder (2009) and Barber and Candès (2019). Wasserman and Roeder (2009) proposed to split the data into three parts to implement a three-stage regression method. In the first stage, the user fits a suite of candidate models

to the data, with different tuning parameters, using the first part of the data. In the second stage, the second part of the data is used for selecting one of those models based on cross validations. In the third stage, null features are eliminated using hypothesis testing based on the third part of the data. Barber and Candès (2019) use the data-splitting strategy to extend the fixed-design knockoff framework to the high-dimensional setting. Specifically, the data is split into two parts, while the first part of data is used to screen out enough null features so that the fixed-design knockoff framework can be applied to the selected features on the second part of the data. We note that both methods rely on the so-called screening property, i.e., all relevant features are selected in the first step before the hypothesis testing or the knockoff filtering step. Moving beyond linear models, an assumption-free inference framework is proposed in Rinaldo et al. (2016) by combining data splitting and bootstrapping (or the normal approximation).

Another use of data splitting is to boost the power of multiple testing procedures. Two notable methods are proposed in Rubin et al. (2006) and Ignatiadis et al. (2016). Rubin et al. (2006) derive the optimal test statistic cutoffs that maximize the expected number of true positives, which depend on the underlying data generating process. The authors thus proposed to use data splitting, so that one part of the data is used to estimate the optimal cutoffs, while the rest of the data is used for testing. Ignatiadis et al. (2016) employ a hypothesis-weighting approach to improve the power of multiple testing. In particular, the authors proposed to use data splitting, in which one part of the data is used to determine proper weights for each individual hypotheses, and the other part of the data is used for large-scale multiple testing.

The undesirable randomness in data splitting can be lessened by repeating the procedure multiple times. Methodological developments along this line include methods proposed in van de Wiel et al. (2009), Meinshausen et al. (2009), and Romano and DiCiccio (2019), all of which aim at combining p-values obtained over multiple data splits. van de Wiel et al. (2009) proposed to aggregate p-values using the median, in testing the prediction error difference between two predictors constructed using

each part of the data. A more sophisticated approach of combining p-values uses a properly scaled $\gamma-$quantile, as proposed in Meinshausen et al. (2009), which gives asymptotic control of FDR under the screening property and an additional rank assumption on the design matrix. Romano and DiCiccio (2019) introduced several alternative approaches, based on concentration inequalities, or the limiting distribution of the averaged p-value. Our proposed approach is very different to the aforementioned methods, as it is built upon the inclusion rates estimated from multiple data splits rather than p-values.

3.2.3 Main contributions of the paper

In contrast to the Gaussian mirror method, which perturbs the data by adding and subtracting a Gaussian noise to each feature, we propose to impose a bootstrap-type perturbation via random data splitting, which is both conceptually simpler and computationally cheaper (can be done for all features simultaneously). Specifically, we split the whole data set into two halves, and apply two potentially different statistical learning procedures to each part of the data. The idea of using data splitting to make valid statistical inferences has been around for some time, and a review of related methods is given in Section 3.2.2. For most existing methods, the main motivation for splitting the data is to obtain valid p-values for each feature. Our proposed approach is different in the sense that, instead of aiming at p-values, we focus on perturbing the data to obtain two independent measurements of the importance of each feature so that a proper contrast between the two measurements of the same feature can be used to control FDR.

Ways of estimating the number of false positives without requiring p-values have been described in Barber and Candès (2015) for knockoff filters and Xin et al. (2019) for the Gaussian mirror method. The main idea is to construct a contrasting statistic M_j, called the "mirror statistic" in Xin et al. (2019), for each feature X_j, which enjoys the following two key properties as illustrated by Figure 3.1:

1. A feature with a larger mirror statistic is more likely to be a relevant feature.

2. The sampling distribution of the mirror statistic of a null feature is symmetric about 0.

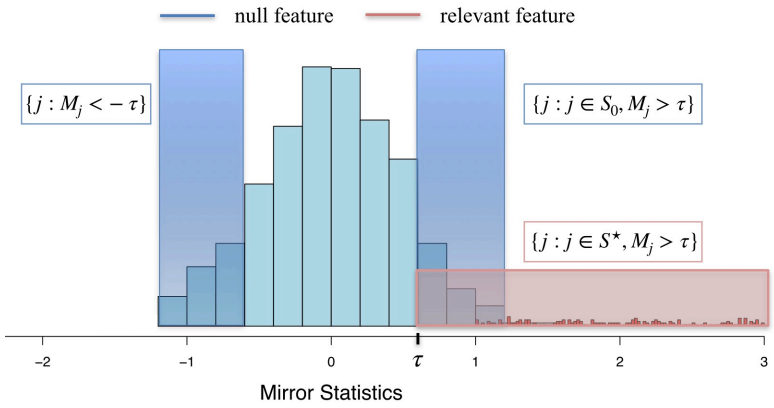


Figure 3.1: A cartoon illustration of the mirror statistic. M_j denotes the mirror statistic associated with feature X_j. S_0 denotes the set of null features, and $S^\star$ denotes the set of relevant features. Features associated with a mirror statistic larger than the cutoff τ are selected.

Property 1 suggests that we can rank the importance of each feature by its mirror statistic, and select those features with their mirror statistics greater than a cutoff value (τ in Figure 3.1). Property 2 implies that we can estimate (conservatively) the number of false positives $\#\{j : j \in S_0,\ M_j > \tau\}$ by $\#\{j : M_j < -\tau\}$, if the mirror statistics of the null features are not too correlated. Based upon this principle, given any FDR control level $q \in (0, 1)$, we can choose a data-driven cutoff τ_q so that our proposed approach can achieve an asymptotic FDR control. As we will see, Property 1 is naturally satisfied due to the construction of the mirror statistic, thus our main concern is Property 2.

Another main contribution of this work is to propose a multiple data-splitting (MDS) approach, which both reduces the variability of the selection result and boosts the power. For the ease of presentation, we refer to the single data-splitting approach and the multiple data-splitting approach as DS and MDS, respectively. Instead of ranking the features by the mirror statistics, we rank them by their inclusion rates estimated through multiple data splits. We show that FDR can be still under control if

the rank of the inclusion rate is reasonably consistent with the rank of the feature importance. Empirically, we observe that MDS has the capability to almost retrieve the full power without splitting the data. We back up the empirical result by studying a simple Normal mean model, and we prove that the inclusion rate is a monotone decreasing function of the p-value calculated using the full data set. In particular, MDS can be regarded as a Rao-Blackwell improvement of DS in terms of ranking the features.

We apply the data-splitting approaches to four canonical models including the linear model, the generalized linear model, the Gaussian graphical model, and the deep neural network. For the linear model, we focus on the high-dimensional setting, in which two strategies are considered. The simpler strategy is to first select preliminary features using one part of the data via a high-dimensional feature selection procedure such as Lasso, and then run OLS on the selected features using the other part of the data. Property 2 is satisfied if all relevant features are selected in the first step (the so-called screening property). The more involved strategy aims at directly symmetrizing the Lasso estimator using the debiasing method discussed in Van de Geer et al. (2014).

For the generalized linear model, we consider two asymptotic regimes. The moderate-dimensional setting concerns $p/n \to \kappa \in (0, 1/2)$, in which n is the sample size, and the factor $1/2$ accounts for data splitting. The mirror statistics are built upon the maximal likelihood estimator (MLE). As shown in Sur and Candès (2019), for the logistic regression model, the Fisher information does not correctly target the asymptotic variance of the MLE. Therefore, it is at the risk of FDR inflation or power loss to simply standardize the MLE by the Fisher information and use BHq. Sur and Candès (2019) showed that adjusted inference is possible for the logistic regression model and the probit regression model. However, moving beyond these special cases, to the best of our knowledge, it remains an open problem to quantify the asymptotic uncertainty of the MLE for generalized linear models including the Possion regression model and the multinomial logistic regression model. On the other hand, the selection result of the data-splitting method is invariant to any constant rescaling of the mirror statistics.

Therefore, we are free of estimating the asymptotic variance of the MLE. For the high-dimensional setting, we use the debiasing method similar as the case of the high-dimensional linear model. We show that the data-splitting method achieves an asymptotic FDR control under proper sparsity conditions and regularity conditions.

Methods designed for the linear model are applicable to the Gaussian graphical model because of the linear representation of the conditional dependence structure (Lauritzen, 1996). Given an FDR control level q, we apply DS or MDS to each nodewise linear regression with designated FDR control level $q/2$, and then combine the nodewise selection results using the OR rule (Meinshausen and Bühlmann, 2006). Our simulation study shows that DS and MDS performed significantly better than existing approaches including BHq based on the partial correlation test and GFC proposed in Liu (2013).

For the deep neural network, we train two identically structured networks using each part of the data. Two types of mirror statistics are considered in order to cope with the issue of non-identifiability of labels of the hidden units in each network. The first one is built on weight multiplications as in Lu et al. (2018), while the second one utilizes the influence function (Hechtlinger, 2016) to measure each feature's importance. The influence function approach appears to be also applicable to more sophisticated networks including convolutional and recurrent neural networks. For the fully-connected feed-forward neural networks, the performances of the two approaches are similar based on our empirical studies.

The rest of the paper is structured as follows. Section 3.3.1 introduces DS, a proposed FDR control approach based on a single data split. Section 3.3.2 and 3.3.3 detail MDS, accompanied by useful theoretical insights on a simple Normal means model. Section 3.3.4 compares our approaches with existing methods including BHq, the knockoff filter, and the Gaussian mirror method. Section 3.4 discusses the applications of the proposed data-splitting approaches on four popular models including the linear model, the generalized linear model, the Gaussian graphical model, and the deep neural

network. Section 3.5 demonstrates the competitive performances of DS and MDS through simulation studies on the aforementioned models. Section 3.5.5 applies DS and MDS to the task of identifying mutations associated with drug resistance in the HIV-1 data. Section 3.6 concludes with a few final remarks.

3.3 Data Splitting for the FDR Control

3.3.1 Single data split

Suppose there is a set of random features $\{X_1, \cdots, X_p\}$, which jointly follow some p-dimensional probability distribution. Denote $\boldsymbol{X}_{n\times p} = (\boldsymbol{X}_1, \cdots, \boldsymbol{X}_p)$ as the n independent observations of these features, also known as the *design matrix*. The bold case $\boldsymbol{X}_j = (\boldsymbol{X}_{1j}, \cdots, \boldsymbol{X}_{nj})^\intercal$ denotes the vector containing n independent realizations of feature X_j, while the bold case $\boldsymbol{x}_i = (\boldsymbol{X}_{i1}, \cdots, \boldsymbol{X}_{ip})^\intercal$ denotes the i-th observation. For each set of observed features $(X_{i1}, \cdots, X_{ip})$, there is an associated response variable y_i, for $i \in [n]$. Let $\boldsymbol{y} = (y_1, \cdots, y_n)^\intercal$ be the vector of n independent responses. For each subset S of $\{1, \cdots, p\}$, we denote $X_S = \{X_j : j \in S\}$ and $X_{-S} = \{X_j : j \notin S\}$. The task of variable or feature selection is to find the smallest set $S^\star$ such that

$$y \perp\!\!\!\perp X_{-S^\star} \mid X_{S^\star}.$$

In the following, we denote $S^\star$ as the set of indexes for relevant features (non-null features). Thus, $S_0 = \{1, \cdots, p\} \backslash S^\star$ is the complementary set of indexes, and $X_{S_0} = X_{-S^\star}$ is the collection of all null features. In practice, one should first de-mean and standardize the features *a priori* so as to make the features comparable to each other. Let $p_0 = |S_0|$ and $p_1 = |S^\star|$ be the number of null features and relevant features, respectively.

The selection of relevant features for a statistical learning model such as regression commonly relies

on a set of coefficients (or other measures of "impact") $\{\widehat{\beta}_1, \cdots, \widehat{\beta}_p\}$, each corresponding to one feature, which are estimated based on the observed data. The larger the $|\widehat{\beta}_j|$ is, the more likely we believe that feature X_j is useful in predicting y. For example, in the linear regression setting, $\widehat{\beta}_j$ can be the regression coefficient estimated via OLS or Lasso. Contrast to the commonly practiced approaches that select features based on a single coefficient estimate $\widehat{\beta}_j$ for each feature X_j, we propose to construct two independent estimates of the coefficient, $\widehat{\beta}_j^{(1)}$ and $\widehat{\beta}_j^{(2)}$, potentially with two different statistical procedures, in order to set up an FDR control framework.

The independence of the two coefficient estimates can be ensured by employing a data-splitting strategy. To be specific, we split the n observations into two groups, denoted as $(\boldsymbol{y}^{(1)}, \boldsymbol{X}^{(1)})$ and $(\boldsymbol{y}^{(2)}, \boldsymbol{X}^{(2)})$. Then, we obtain $\widehat{\beta}_j^{(1)}$ based on $(\boldsymbol{y}^{(1)}, \boldsymbol{X}^{(1)})$, and $\widehat{\beta}_j^{(2)}$ based on $(\boldsymbol{y}^{(2)}, \boldsymbol{X}^{(2)})$. We note that the data-splitting procedure can be arbitrary, not necessary to be completely at random. The sample sizes for the two groups can also be different. The only requirement is that it should be independent of the response variables $\boldsymbol{y}$, which can be easily satisfied if we split the data without looking at $\boldsymbol{y}$.

Motivated by Xin et al. (2019), we construct feature X_j's *mirror statistic* as

$$M_j = \left|\widehat{\beta}_j^{(1)} + \widehat{\beta}_j^{(2)}\right| - \left|\widehat{\beta}_j^{(1)} - \widehat{\beta}_j^{(2)}\right|. \tag{3.2}$$

Given a designated FDR control level $q \in (0, 1)$, the goal is to choose a data-dependent cutoff τ_q, so that FDR among the selected features $\{j : M_j > \tau_q\}$ is under q. To achieve this asymptotically, we need to impose a few essential requirements on the M_j's for both relevant and null features. For a relevant feature, its M_j tends to be large if (i) $\widehat{\beta}_j^{(1)}$ and $\widehat{\beta}_j^{(2)}$ have the same sign; (ii) $|\widehat{\beta}_j^{(1)}|$ and $|\widehat{\beta}_j^{(2)}|$ are both relatively large. For a null feature, we require the following assumption on the sampling distribution of either $\widehat{\beta}_j^{(1)}$ or $\widehat{\beta}_j^{(2)}$ to control FDR. Without loss of generality, we state the assumption in terms of $\widehat{\beta}_j^{(2)}$.

Assumption 3.3.1. (Symmetric Assumption) For each null feature index $j \in S_0$, conditioning on the design matrix $\boldsymbol{X}^{(2)}$, the sampling distribution of $\widehat{\beta}_j^{(2)}$ is symmetric about 0.

We note that (i) the symmetric assumption is only imposed on null features, and is not required for relevant features; (ii) for the purpose of FDR control, it is sufficient that either $\widehat{\beta}_j^{(1)}$ or $\widehat{\beta}_j^{(2)}$ satisfies the symmetric assumption. We will show that the symmetric assumption can be satisfied with high probability for some standard statistical models, such as the linear model and the Gaussian graphical model, under some proper conditions. In addition, we can relax the assumption from finite-sample exact symmetric to asymptotic symmetric for specific models (see our discussion on the high-dimensional (generalized) linear model using the debiasing method in Section 3.4). For more general statistical learning models, there is no guarantee for the symmetric assumption, although empirically we find that our proposed approach can be applied effectively to complex models such as fully connected deep neural networks. A detailed discussion of the symmetric assumption for different statistical models is deferred to Section 3.4. We have the following property for the mirror statistics:

Lemma 3.3.1. Under Assumption 3.3.1, regardless of the data-splitting procedure, the sampling distribution of M_j is symmetric about 0 for each $j \in S_0$.

The proof is elementary and thus omitted. The simultaneous symmetric property of the mirror statistics for null features leads us to approximately upper bound the number of false discoveries as follows:

$$\#\{j \in S_0 : M_j > t\} \approx \#\{j \in S_0 : M_j < -t\} \leq \#\{j : M_j < -t\}, \quad \forall t > 0.$$

Therefore, an over-estimated FDP of our selection $\widehat{S}_t = \{j : M_j > t\}$ is

$$\widehat{\mathrm{FDP}}(t) = \frac{\#\{j : M_j < -t\}}{\#\{j : M_j > t\} \vee 1}.$$

For any designated FDR control level q, we can choose the data-driven cutoff τ_q as follows:

$$\tau_q = \min\{t > 0 : \widehat{\mathrm{FDP}}(t) \leq q\}, \tag{3.3}$$

and our final selection will be $\widehat{S}_{\tau_q} = \{j : M_j > \tau_q\}$. A summary of our proposed method can be found in Algorithm 9. To prove that the procedure asymptotically control FDR under a pre-specified level q as $p \to \infty$, we require the following weak dependency assumption for all the mirror statistics associated with the null features.

Assumption 3.3.2. (Weak dependency) The mirror statistics $M_j's$ are continuous random variables, and there exist constants $C > 0$ and $\alpha \in (0, 2)$ such that

$$\mathrm{Var}\Big(\sum_{j \in S_0} 1(M_j > t)\Big) \leq C p_0^{\alpha}, \quad \forall\, t \in \mathbb{R}.$$

To understand this assumption, we note that if the mirror statistics M_j's, associated with the null features, are perfectly correlated, or can be clustered into a fixed number of groups so that their within-group correlation is 1, then α has to be 2 and the assumption does not hold. However, if the dependence between M_j and M_{j+k} decays at a reasonable rate of k, Assumption 3.3.2 holds. Under this assumption, the proposition below justifies our proposed approach for controlling FDR.

Proposition 3.3.1. For any given FDR control level q, under Assumptions 3.3.1 and 3.3.2, if $p_0 \to \infty$ as $p \to \infty$, we have

$$\limsup_{p \to \infty} \mathbb{E}\left[\frac{\#\{j : j \in S_0, j \in \widehat{S}_{\tau_q}\}}{\#\{j : j \in \widehat{S}_{\tau_q}\} \vee 1}\right] \leq q.$$

The proof of Proposition 3.3.1 can be found in the Appendix C.

Algorithm 9: False discovery rate control via single data split.

1. Split the data set into two groups $(\boldsymbol{y}^{(1)}, \boldsymbol{X}^{(1)})$ and $(\boldsymbol{y}^{(2)}, \boldsymbol{X}^{(2)})$, independent of the response variable $\boldsymbol{y}$.
2. Estimate the "impact" coefficients on each part of the data to obtain $\widehat{\beta}_j^{(1)}$ and $\widehat{\beta}_j^{(2)}$. The two estimation procedures can be potentially different.
3. Calculate the mirror statistics $M_j = \left|\widehat{\beta}_j^{(1)} + \widehat{\beta}_j^{(2)}\right| - \left|\widehat{\beta}_j^{(1)} - \widehat{\beta}_j^{(2)}\right|$.
4. Given a designated FDR level $q \in (0, 1)$, calculate the cutoff τ_q as below:
$$\tau_q = \min\left\{t > 0 : \widehat{\mathrm{FDP}}(t) = \frac{\#\{j : M_j < -t\}}{\#\{j : M_j > t\} \vee 1} \leq q\right\}.$$
5. Select the features $\{j : M_j > \tau_q\}$.

3.3.2 Multiple data splits

There are two major concerns about the single data-splitting procedure (DS). First, splitting the data into two halves inflates the variances of the estimated coefficients, thus the procedure can potentially suffer from a power loss. Second, the selection result is not stable and can vary substantially across different data splits. A natural way to remedy these issues is to repeat the procedure multiple times and appropriately aggregate the selection results. In the settings when p-values are available, Meinshausen et al. (2009) proposed a method to aggregate the dependent p-values obtained from multiple sample splits. We here propose a different approach, detailed in Algorithm 10, to aggregate multiple selection results without requiring p-values, which can almost achieve the power of using the full data.

Suppose we repeat the data-splitting procedure m times independently. Each time the set of selected features is denoted as $\widehat{S}^{(k)}$, $k \in [m]$. For each feature X_j, we define its population inclusion

rate I_j and the corresponding empirical inclusion rate $\widehat{I}_j$ as

$$I_j = \mathbb{E}\left[\frac{1(j \in \widehat{S})}{|\widehat{S}| \vee 1}\right], \quad \widehat{I}_j = \frac{1}{m}\sum_{k=1}^{m} \frac{1(j \in \widehat{S}^{(k)})}{|\widehat{S}^{(k)}| \vee 1}, \tag{3.4}$$

where the expectation is taken with respect to both the randomness in $\boldsymbol{y}$ and the randomness in data splitting. The proposed aggregation approach is most useful if the following informal statement is approximately true: if a feature is selected less frequently in the m independent data splits, it is less likely to be a relevant feature. In other words, the rank of each feature in terms of the inclusion rate should roughly reflect the feature's importance. If this holds, we only need to choose a inclusion rate cutoff so that the features with their inclusion rates greater than the cutoff are selected.

Our intuition is based on the fact that conditioning on the design matrix $\boldsymbol{X}$, if we can regenerate m independent sets of the response variable $\boldsymbol{y}$ and apply the proposed method, on average, the false discovery proportions should be (asymptotically) no larger than q based on Proposition 3.3.1. Although there is no way for us to regenerate data, we propose a backtracking procedure based on this fact, by treating the m dependent selection results obtained from repeated data splits as an approximation to m independent selection results obtained with data regeneration.

Specifically, we first sort the features with respect to their empirical inclusion rates in an increasing order. Denote the sorted inclusion rates as $0 \leq \widehat{I}_{(1)} \leq \widehat{I}_{(2)} \leq \cdots \leq \widehat{I}_{(p)}$. The cutoff is then chosen to be the largest $\ell \in [p]$ such that $\widehat{I}_{(1)} + \cdots + \widehat{I}_{(\ell)} \leq q$. A heuristic justification for this selection is as follows. Imagine that the set of features $\{j : \widehat{I}_j > \widehat{I}_{(\ell)}\}$ are all relevant features, and the set of features $\{j : \widehat{I}_j \leq \widehat{I}_{(\ell)}\}$ are all null features. Then, if we review the m selection results, we can find that the average false discovery proportions of the m data-splitting selection procedures is the largest under q.

In terms of the FDR control, we provide a theoretical justification of the multiple data-splitting (MDS) approach at the population level. That is, we proceed as if we have access to the population

Algorithm 10: Aggregating selection results from multiple data splits

1. Sort the features with respect to their empirical inclusion rates in an increasing order. Denote the sorted empirical inclusion rates as $0 \leq \widehat{I}_{(1)} \leq \widehat{I}_{(2)} \leq \cdots \leq \widehat{I}_{(p)}$.
2. Find the largest $\ell \in [p]$ such that $\widehat{I}_{(1)} + \cdots + \widehat{I}_{(\ell)} \leq q$.
3. Select the features $\widehat{S} = \{j : \widehat{I}_j > \widehat{I}_{(\ell)}\}$.

inclusion rate I_j for $j \in [p]$, where in practice, the empirical inclusion rate $\widehat{I}_j$ serves as an unbiased estimator to I_j. We require the following assumptions.

Assumption 3.3.3.

(a) (Null exchangeability) The distribution of $\{1(j \in \widehat{S}),\ j \in S_0\}$ is exchangeable.

(b) (Rank faithfulness) For any $\alpha \in (0, 1)$, we have

$$\limsup_{p\to\infty} \frac{1}{p_1} \sum_{j\in S^\star} 1\left(I_j \leq \frac{\alpha}{p_0}\right) \leq \alpha,$$

where $p_0 = |S_0|, p_1 = |S^\star|$.

(c) (Procedure faithfulness) For each single data split, the procedure has an asymptotic FDR control, that is,

$$\limsup_{p\to\infty} \mathbb{E}\left[\frac{\#\{j : j \in S_0, j \in \widehat{S}\}}{\#\{j : j \in \widehat{S}\} \vee 1}\right] = \limsup_{p\to\infty} \sum_{j\in S_0} I_j \leq q.$$

Assumption 3.3.3(a) also appears in Meinshausen and Bühlmann (2010) (Theorem 1), and it directly implies that for any $i, j \in S_0$, $I_i = I_j$. Assumption 3.3.3(b) guarantees that the rank of a feature, in terms of the inclusion rate, is more informative of the feature's importance than random guessing. Under Assumption 3.3.3, we have the following proposition.

Proposition 3.3.2. For any FDR control level $q \in (0, 1)$, let ℓ be the largest value in $[p]$ such that

$I_{(1)} + \cdots + I_{(\ell)} \leq q$, in which $0 \leq I_{(1)} \leq I_{(2)} \leq \cdots \leq I_{(p)}$ are the order statistics of the population inclusion rates. Under Assumption 3.3.3, we have

$$\limsup_{p\to\infty} \frac{\sum_{j\in S_0} 1(I_j > I_{(\ell)})}{\sum_{j=1}^{p} 1(I_j > I_{(\ell)}) \vee 1} \leq q, \tag{3.5}$$

in the asymptotic regime $p_0 \to \infty$ and $\liminf_{p\to\infty} p_1/p_0 > 0$.

The proof of Proposition 3.3.2 can be found in the Appendix C.

3.3.3 A theoretical study of MDS for the Normal means model

We next consider a simple Normal means model, upon which we show that MDS can achieve almost the full power without splitting the data. For $i \in [n]$, we assume X_{ij} follow $N(\mu_j, \sigma^2)$, where $j \in [p]$ and σ^2 is known. We assume all X_{ij} are independent. To test whether μ_j is 0, the standard p-value is given by $p_j = \Phi(-|\sqrt{n}\bar{\boldsymbol{X}}_j/\sigma|)$, where $\bar{\boldsymbol{X}}_j = \sum_{i=1}^{n} X_{ij}/n$, and Φ is the CDF of the standard Normal distribution. Under this setup, we have the following proposition.

Proposition 3.3.3. For the Normal means model described above, the population inclusion rate defined in Equation (3.4) is monotone with respect to the p-value calculated using the full data. Mathematically, this means

$$\mathbb{E}\left[\frac{1(j \in \widehat{S})}{|\widehat{S}| \vee 1} \,\middle|\, p_j\right]$$

is a monotone decreasing function of p_j for all $j \in [p]$.

The proof of Proposition 3.3.3 can be found in the Appendix C. Proposition 3.3.3 implies that for this simple model, the rank of the features, in terms of the inclusion rates, can be as informative as the rank in terms of p-values. Therefore, if the empirical inclusion rate $\widehat{I}_j$ is a reasonable estimate to the population inclusion rate I_j, MDS is possibly as powerful as those methods based on the rank of p-values, which are calculated using the full data set.

Proposition 3.3.3 also suggests why MDS is superior to DS in terms of ranking the features. For DS, we can substitute the ranks by the mirror statistics with the ranks by $1(j \in \widehat{S})/(|\widehat{S}| \vee 1)$, which does not decrease the number of correctly ranked pairs. That is, for any pair of feature indexes (i, j), if the rank by the mirror statistics aligns with the rank by the p-values, say $p_i \leq p_j$ and $M_i \geq M_j$, then the rank by $1(j \in \widehat{S})/(|\widehat{S}| \vee 1)$ also aligns with the rank by the p-values, i.e., $1(i \in \widehat{S})/(|\widehat{S}| \vee 1) \geq 1(j \in \widehat{S})/(|\widehat{S}| \vee 1)$. For MDS, all the features are ranked by $\mathbb{E}[1(j \in \widehat{S})/(|\widehat{S}| \vee 1)\big|\boldsymbol{X}]$*, where the expectation is taken with respect to the randomness in data splitting. In this simple model, MDS yields a rank better aligned with the rank of p-values, because conditioning on the p-values,

$$\text{Var}\left(\frac{1(j \in \widehat{S})}{|\widehat{S}| \vee 1}\right) \geq \text{Var}\left(\mathbb{E}\left[\frac{1(j \in \widehat{S})}{|\widehat{S}| \vee 1}\middle|\boldsymbol{X}\right]\right),$$

where the variance on the left hand side is taken with respect to both the randomness in the data and the randomness in data splitting, while the variance on the right hand side is taken with respect to the randomness only in the data. Therefore, MDS is essentially a Rao-Blackwell improvement of DS.

We empirically check Assumption 3.3.3(b) and Proposition 3.3.3 on the Normal means model. We set $\sigma = 1$, $p = 800$, $p_1 = 160$, $n = 1000$, and $m = 400$. For $j \in [p_1]$, μ_j is sampled from $N(0, \delta^2)$, whereas the rest μ_j's are set to be 0. The left panel of Figure 3.2 shows that Assumption 3.3.3(b) is satisfied for various signal strengths $\delta = 0.1, 0.2, 1$, in which the red line represents the upper bound αp_1, and the other three lines represent the number of less-frequently selected relevant features, $\sum_{j \in S^\star} 1(I_j \leq \alpha/p_0)$. In the right panel of Figure 3.2, we set $\delta = 0.5$, and plot the empirical inclusion rates (red "$*$") and the mirror statistics based on a single split (blue "+") against the p-values calculated using the full data set. We see that the empirical inclusion rate is approximately a monotone decreasing function of the p-value. Besides, the rank of the empirical inclusion rates aligns better with the rank of the p-values, and is much more informative than the rank of the mirror statistics.

*In more general cases where we conditioning on the design matrix $\boldsymbol{X}$ and the randomness comes from the response variable $\boldsymbol{y}$, MDS ranks all the features by $\mathbb{E}[1(j \in \widehat{S})/|(\widehat{S}| \vee 1)\big|\boldsymbol{y}]$.

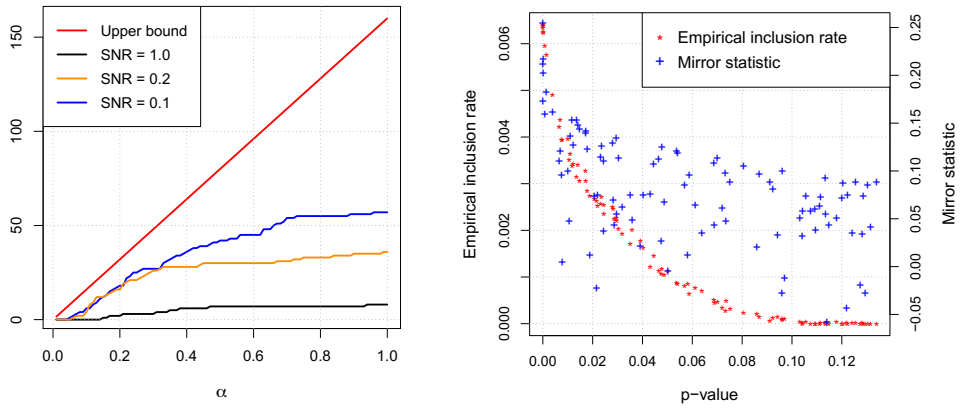


Figure 3.2: (Left) Check of Assumption 3.3.3(b). The red line represents the upper bound αp_1, while the other three lines represent the number of less-frequently selected relevant features, $\sum_{j \in S^\star} 1(I_j \leq \alpha/p_0)$, under different signal-to-noise ratios (SNR) $\{0.1, 0.2, 1\}$, respectively. (Right) Empirical inclusion rates and mirror statistics against p-values calculated using the full data set.

3.3.4 Comparison with existing methods

The proposed data-splitting methods are perhaps most useful in the settings when practicing exact or asymptotic hypothesis testing is not straightforward. That is, a valid p-value from the significance test for each feature is not available, upon which some popular FDR control procedures rely, including BHq, BYq, and the q-value approach.

The high-dimensional linear model serves as a canonical example where constructing valid p-values is difficult. Much effort has been made in the literature for obtaining valid p-values for the selected features using the theory of post-selection inference. Popular selection procedures including Lasso (Lee et al., 2016), forward stepwise regression, and least angle regression (Tibshirani et al., 2016) have been considered. However, this type of theory is mostly developed case by case, and can not be easily generalized to other selection procedures such as SCAD (Fan and Li, 2001) and elastic net (Zou and Hastie, 2005). DS and MDS are more flexible in the sense that as long as the screening property holds, i.e., all the relevant features are estimated to be nonzero in $\widehat{\boldsymbol{\beta}}^{(1)}$ (see Section 3.4.1), the selection result

enjoys an asymptotic FDR control.

The knockoff filter is another class of methods that provides an exact FDR control without requiring p-values. The fixed-design knockoff filter (Barber and Candès, 2015) is proposed to exactly control FDR for the linear model in low-dimensional settings ($n \geq 2p$). The model-X knockoff filter (Candes et al., 2018) largely generalizes its applicability to arbitrary models between y and X in high-dimensional settings. The knockoff filter is theoretically superior to DS and MDS in the sense that it guarantees a finite sample FDR control instead of an asymptotic FDR control. However, DS and MDS can be more favorable from the following perspectives. First, in high-dimensional settings, the model-X knockoff filter is only applicable if the joint distribution of all the features is known, otherwise estimating the joint distribution itself can be very challenging. Because DS and MDS require no knowledge of the feature generating process, they are expected to be more robust in real applications. Second, DS and MDS are computationally more efficient compared to the model-X knockoff filter because generating knockoff features in high-dimensional settings can be quite expensive. Third, empirically we find that when the features are highly correlated, the model-X knockoff filter can be too conservative to detect relevant features (see Section 3.5.1).

The proposed data-splitting approaches are also computationally more favorable compared to the Gaussian mirror method (Xin et al., 2019). For the linear model, the Gaussian mirror method requires running p times linear fittings, whereas DS and MDS only requires running 2 and $2m$ linear fittings, respectively, which can be much smaller than p. For the Gaussian graphical model, DS requires running $2p$ times nodewise linear fittings (see Section 3.4.3). The Gaussian mirror method is also potentially applicable to the Gaussian graphical model. However, it requires p^2 times nodewise linear fittings, which is unacceptable in high-dimensional settings unless using parallel computing resources.

3.4 Specializations for Different Statistical Models

In this section, we discuss how we construct the "impact" coefficients for several popular statistical models. Our main concern is that the "impact" coefficient $\widehat{\boldsymbol{\beta}}^{(2)}$, calculated based on $(\boldsymbol{y}^{(2)}, \boldsymbol{X}^{(2)})$, shall satisfy the symmetric assumption (Assumption 3.3.1). We assume that the data-splitting procedure is done so that the discussion throughout this section is conditioning on the data-splitting result.

3.4.1 Linear model

The linear model assumes that the true data generating process is $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta}^\star + \boldsymbol{\epsilon}$, where $\boldsymbol{\beta}^\star$ denotes the true regression coefficient, and the noise $\boldsymbol{\epsilon}$ follows $N(0, \sigma^2 I_n)$. In the context of feature selection, $\boldsymbol{\beta}^\star$ is often assumed to be nonzero restricted on a subset $S^\star \subseteq \{1, \cdots, p\}$. In the low-dimensional setting where $\boldsymbol{X}$ is full rank ($n \geq p$), we can simply run the ordinary least regression (OLS), and take the estimated regression coefficient as the "impact" coefficient $\widehat{\boldsymbol{\beta}}^{(2)}$. Because the sampling distribution of $\widehat{\boldsymbol{\beta}}^{(2)}$, with respect to the randomness in $\boldsymbol{y}^{(2)}$, is $N(\boldsymbol{\beta}^\star, \sigma^2(\boldsymbol{X}^{(2)\mathsf{T}}\boldsymbol{X}^{(2)})^{-1})$, the symmetric assumption is satisfied. $\widehat{\boldsymbol{\beta}}^{(1)}$ can be estimated using either OLS or any other regularization methods such as Lasso.

In the high-dimensional setting where $p > n$, we consider two approaches. The first approach relies on the so-called screening property, which means that after applying some feature selection method to $(\boldsymbol{X}^{(1)}, \boldsymbol{y}^{(1)})$, the selected feature set contains all the relevant features, that is, $\widehat{S}^{(1)} \supseteq S^\star$. For Lasso, under the restricted eigenvalue condition (Bickel et al., 2009), which rules out the scenario that the design matrix has unacceptably high pairwise correlation, and the "beta-min" condition (Dezeure et al., 2015), which ensures that the signal strength is large enough, the screening property holds with high probability. We remark that the "beta-min" condition calibrates the minimum requirement for the signal strength as a function of $p_1 = |S^\star|$ and p, thus it does not contradict Assumption 3.3.2.

The first approach still uses OLS to construct the "impact" coefficient $\widehat{\boldsymbol{\beta}}^{(2)}$, but on the subset of features $\widehat{S}^{(1)}$, selected using the first half of data $(\boldsymbol{y}^{(1)}, \boldsymbol{X}^{(1)})$. The symmetric assumption holds for $\widehat{\boldsymbol{\beta}}^{(2)}$ as long as the screening property holds for $\widehat{\boldsymbol{\beta}}^{(1)}$. In the following, we refer to the procedure, which applies Lasso to $(\boldsymbol{X}^{(1)}, \boldsymbol{y}^{(1)})$ and OLS to $(\boldsymbol{X}^{(2)}, \boldsymbol{y}^{(2)})$, as the Lasso + OLS procedure.

The second approach symmetrizes the Lasso estimator via the debiasing method. The debiased Lasso estimator $\widehat{\boldsymbol{\beta}}^{d}$ takes the form of

$$\widehat{\boldsymbol{\beta}}^{d} = \widehat{\boldsymbol{\beta}} + \frac{1}{n} D \boldsymbol{X}^{\intercal}(\boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}), \tag{3.6}$$

where D is a "decorrelating" matrix. Plugging in $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta}^{\star} + \boldsymbol{\epsilon}$, we obtain the following decomposition:

$$\sqrt{n}(\widehat{\boldsymbol{\beta}}^{d} - \boldsymbol{\beta}^{\star}) = \boldsymbol{Z} + \boldsymbol{\Delta} \tag{3.7}$$

with

$$\boldsymbol{Z}|\boldsymbol{X} \sim N(0, \sigma^2 D\widehat{\Sigma}D^{\intercal}) \quad \text{and} \quad \boldsymbol{\Delta} = \sqrt{n}(D\widehat{\Sigma} - I)(\boldsymbol{\beta}^{\star} - \widehat{\boldsymbol{\beta}}), \tag{3.8}$$

where $\widehat{\Sigma} = (\boldsymbol{X}^{\intercal}\boldsymbol{X})/n$ is the sample covariance matrix. Let Σ be the corresponding population covariance matrix of $(X_1, \cdots, X_p)$.

Various proposals of D have been documented in the literature to minimize both the bias term $\boldsymbol{\Delta}$ and the variance term $D\widehat{\Sigma}D$. In this dissertation, we follow the approach outlined in Javanmard and Montanari (2013), Zhang and Zhang (2014) and Van de Geer et al. (2014), and set D to be an estimator of the precision matrix $\Omega = \Sigma^{-1}$. The construction of $\widehat{\Omega}$ is detailed in Algorithm 11, which enjoys a convergence rate $||\widehat{\Omega} - \Omega||_{\infty} = o(1/\sqrt{\log p})$, under the sparsity condition specified in Assumption 3.4.1. For alternative constructions of D, we refer the readers to an optimization approach outlined in Javanmard and Montanari (2014).

Let $\Lambda = D\widehat{\Sigma}D^{\top} = \widehat{\Omega}\widehat{\Sigma}\widehat{\Omega}^{\top}$. When the bias term $\boldsymbol{\Delta}$ vanishes properly as $n, p \to \infty$, the asymp-

Algorithm 11: Construction of the "decorrelating" matrix $\widehat{\Omega}$.

1. Nodewise Lasso regression.
   (a) Linear model. For $j \in [p]$, let
   $$\widehat{\boldsymbol{\gamma}}_j = \underset{\boldsymbol{\gamma}_j \in \mathbb{R}^{p-1}}{\arg\min} \left\{ \frac{1}{2n} ||\boldsymbol{X}_j - \boldsymbol{X}_{-j}\boldsymbol{\gamma}||_2^2 + \lambda_j ||\boldsymbol{\gamma}||_1 \right\}.$$
   (b) GLM. For $j \in [p]$, let
   $$\widehat{\boldsymbol{\gamma}}_j = \underset{\boldsymbol{\gamma}_j \in \mathbb{R}^{p-1}}{\arg\min} \left\{ \frac{1}{2n} ||\boldsymbol{X}_{\widehat{\boldsymbol{\beta}},j} - \boldsymbol{X}_{\widehat{\boldsymbol{\beta}},-j}\boldsymbol{\gamma}||_2^2 + \lambda_j ||\boldsymbol{\gamma}||_1 \right\}.$$
2. Define a matrix $\widehat{C}$, in which $\widehat{C}_{j,k} = -\widehat{\gamma}_{j,k}$, $\widehat{C}_{i,i} = 1$. $\widehat{\gamma}_{j,k}$ is the k^{th} entry of $\widehat{\boldsymbol{\gamma}}_j$.
3. Let $\widehat{\Omega} = \widehat{G}^{-2}\widehat{C}$, in which $\widehat{G} = \text{diag}(\widehat{\tau}_1^2, \cdots, \widehat{\tau}_p^2)$ with
   (a) $\widehat{\tau}_j^2 = (\boldsymbol{X}_j - \boldsymbol{X}_{-j}\widehat{\boldsymbol{\gamma}}_j)^\intercal \boldsymbol{X}_j / n$ for the linear model;
   (b) $\widehat{\tau}_j^2 = (\boldsymbol{X}_{\widehat{\boldsymbol{\beta}},j} - \boldsymbol{X}_{\widehat{\boldsymbol{\beta}},-j}\widehat{\boldsymbol{\gamma}}_j)^\intercal \boldsymbol{X}_j / n$ for the GLM.

totic variance of the debiased Lasso estimator is $\sigma^2\Lambda$, which is inhomogeneous across different coordinate $j \in [p]$. Therefore, it requires rescaling $\widehat{\beta}_j^{(d)}$ to $T_j = \sqrt{n}\widehat{\beta}_j^d / \Lambda_{jj}^{1/2}$ so that all the mirror statistics are in the same magnitude. The method proposed in Javanmard and Javadi (2019) calculates the asymptotic p-value for each feature X_j based on the statistic $T_j/\widehat{\sigma}$, in which $\widehat{\sigma}$ is an estimator to the noise level σ obtained using the scaled Lasso method (Sun and Zhang, 2012). Although the scaled Lasso estimator $\widehat{\sigma}$ is consistent, its finite-sample efficiency can be unsatisfying especially in the cases when the features are correlated. We remark that the selection result of the data-splitting methods is invariant to any constant scaling of the mirror statistics, thus we do not need to estimate the noise level σ. Empirically, we find that this scaling-free property allows the data-splitting methods being more robust and powerful compared to the BHq method in Javanmard and Javadi (2019).

We show in Proposition 3.4.1 that under Assumption 3.4.1, the data-splitting method detailed in

Algorithm 12 enjoys an asymptotic FDR control. The proof of Proposition 3.4.1 can be found in the Appendix C.

Assumption 3.4.1.

1. The sparsity conditions.

   (a) Ω is sparse. $s = \max_{i\in[p]} |\{j \in [p],\ \Omega_{ij} \neq 0\}| = o(\sqrt{n}/\log p)$.

   (b) Signals are sparse. $s^\star = |\{j \in [p], \beta_j^\star \neq 0\}| = o(\sqrt{n}/\log p)$.

2. The conditions on the design matrix $\boldsymbol{X}$.

   (a) The rows of $\boldsymbol{X}\Omega^{1/2}$ are sub-Gaussian. That is, there exists a constant $C > 0$ such that

   $$||\boldsymbol{x}_i^\intercal \Omega^{1/2}||_{\psi_2} = \sup_{p\geq 1} p^{-1/2}\left[\mathbb{E}|\boldsymbol{x}_i^\intercal \Omega^{1/2}|^p\right]^{1/p} < C$$

   (b) $1/C' \leq \sigma_{\min}(\Sigma) \leq \sigma_{\max}(\Sigma) \leq C'$ for some constant $C' > 0$.

3. The required sample size. $\sqrt{n}/\log p \to \infty$.

Proposition 3.4.1. For any given FDR control level $q \in (0, 1)$, using the debiased Lasso estimator for the high-dimensional linear model, under Assumption 3.4.1, we have

$$\limsup_{n,p\to\infty} \mathbb{E}\left[\frac{\#\{j : j \in S_0, j \in \widehat{S}_{\tau_q}\}}{\#\{j : j \in \widehat{S}_{\tau_q}\} \vee 1}\right] \leq q.$$

3.4.2 Generalized linear models

We consider a generalized linear model (GLM) with a canonical link specified as below,

$$p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\beta}^\star) = \prod_{i=1}^{n} c(y_i) \exp\left(y_i \boldsymbol{x}_i^\intercal \boldsymbol{\beta}^\star - \varphi(\boldsymbol{x}_i^\intercal \boldsymbol{\beta}^\star)\right), \tag{3.9}$$

Algorithm 12: FDR control using debiased Lasso for high-dimensional GLM.

1. Split the data set into two groups $(\boldsymbol{y}^{(1)}, \boldsymbol{X}^{(1)})$ and $(\boldsymbol{y}^{(2)}, \boldsymbol{X}^{(2)})$, independent of the response variable $\boldsymbol{y}$. Let n_1 and n_2 be the sample size of each part of the data.
2. Construct $\widehat{\Omega}^{(1)}$ and $\widehat{\Omega}^{(2)}$ using each part of the data following Algorithm 11. Calculate the variance estimator $\widehat{\sigma}_j^2$ for $j \in [p]$ as below.

$$\widehat{\sigma}_j^{2(1)} = \left(\widehat{\Omega}^{(1)}\widehat{\Sigma}^{(1)}\widehat{\Omega}^{(1)\top}\right)_{j,j}, \quad \widehat{\sigma}_j^{2(2)} = \left(\widehat{\Omega}^{(2)}\widehat{\Sigma}^{(2)}\widehat{\Omega}^{(2)\top}\right)_{j,j},$$

   in which $\widehat{\Sigma}^{(1)}$ and $\widehat{\Sigma}^{(2)}$ are the sample covariance matrix of $\boldsymbol{X}^{(1)}$ and $\boldsymbol{X}^{(2)}$, respectively.
3. On each part of the data, calculate the debiased Lasso estimator $\widehat{\boldsymbol{\beta}}^{(1,d)}$ and $\widehat{\boldsymbol{\beta}}^{(2,d)}$. For $j \in [p]$, calculate

$$T_j^{(1)} = \sqrt{n_1}\widehat{\beta}_j^{(1,d)}/\widehat{\sigma}_j^{(1)}, \quad T_j^{(2)} = \sqrt{n_2}\widehat{\beta}_j^{(2,d)}/\widehat{\sigma}_j^{(2)}.$$

4. Calculate the mirror statistics $M_j = \left|T_j^{(1)} + T_j^{(2)}\right| - \left|T_j^{(1)} - T_j^{(2)}\right|$.
5. Select the features using Algorithm 9.

in which $\boldsymbol{X}_{n\times p}$ is the design matrix, $\boldsymbol{y}_{n\times 1}$ is the response vector, and $\boldsymbol{\beta}^\star_{p\times 1}$ is the regression coefficient. The application of the data-splitting methods is discussed in two asymptotic regimes, including the moderate-dimensional setting and the high-dimensional setting, in which the mirror statistics are built upon the MLE and the debiased Lasso estimator, respectively.

For the ease of presentation, we introduce the following notations. Let $\ell(u, v)$ be the loss function associated with the GLM, i.e., the negative log-likelihood up to a constant,

$$\ell(u, v) = -uv + \varphi(v), \tag{3.10}$$

in which u corresponds to y_i, and v corresponds to $\boldsymbol{x}_i^\intercal\boldsymbol{\beta}^\star$. The (scalar) first and second derivatives of $\ell(u, v)$ with respect to v are denoted as $\dot{\ell}$ and $\ddot{\ell}$, respectively. With a bit abuse of notations, the

gradient and the hessian of $\ell(y, \boldsymbol{x}^\intercal\boldsymbol{\beta})$ with respect to $\boldsymbol{\beta}$ are denoted as $\dot{\ell}_{\boldsymbol{\beta}}$ and $\ddot{\ell}_{\boldsymbol{\beta}}$, respectively. For a general mapping $f(y, \boldsymbol{x})$, denote $P_n f = \sum_{i=1}^n f(y_i, \boldsymbol{x}_i)/n$ and $Pf = \mathbb{E}\left[P_n f\right]$. Let W_β be the diagonal matrix with $W_{i.i} = [\ddot{\ell}(y_i, \boldsymbol{x}_i^\intercal\boldsymbol{\beta})]^{1/2}$. Then $P_n\ddot{\ell}_{\boldsymbol{\beta}} = \boldsymbol{X}_{\boldsymbol{\beta}}^\intercal\boldsymbol{X}_{\boldsymbol{\beta}}/n$, in which $\boldsymbol{X}_{\boldsymbol{\beta}} = W_\beta\boldsymbol{X}$ is the weighted version of the design matrix $\boldsymbol{X}$.

The Moderate-dimensional Setting

In the moderate-dimensional setting, we assume that $p/n \to \kappa \in (0, 1/2)$, in which the factor $1/2$ accounts for data splitting. We note that $\kappa \to 0$ corresponds to the classical setting with fixed p. We consider the random design setting, in which we assume $\boldsymbol{x}_i$ are i.i.d observations following $N(0, \Sigma)$. Besides, we assume the signal strength $\mathrm{Var}(\boldsymbol{x}_i^\intercal\boldsymbol{\beta}^\star) \to \omega^2$ as $n, p \to \infty$.

Denote $\varphi(\boldsymbol{X}\boldsymbol{\beta}) = (\varphi(\boldsymbol{x}_1^\intercal\boldsymbol{\beta}), \cdots, \varphi(\boldsymbol{x}_n^\intercal\boldsymbol{\beta}))^\intercal$. The MLE of the regression coefficient $\boldsymbol{\beta}^\star$ is calculated as below,

$$\widehat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\arg\min}\left\{\frac{1}{n}1^\intercal\varphi(\boldsymbol{X}\boldsymbol{\beta}) - \frac{1}{n}\boldsymbol{y}^\intercal\boldsymbol{X}\boldsymbol{\beta}\right\}. \tag{3.11}$$

First of all, the MLE may not exist. Second, in the cases when the MLE exists, $\widehat{\boldsymbol{\beta}}$ can be asymptotically biased, and the Fisher information may not give the correct asymptotic variance of the MLE.

For the logistic regression model, the phase transition curve for the existence of MLE, parametrized in terms of κ and ω, has been given in Candès and Sur (2020). In addition, as shown in Zhao et al. (2020), we have for $j \in [p]$,

$$\frac{\sqrt{n}(\widehat{\beta}_j - \alpha_\star\beta_j^\star)}{\sigma_\star/\tau_j} \xrightarrow{p} N(0, 1). \tag{3.12}$$

$\tau_j^2 = 1/\Omega_{jj}$, corresponding to the conditional variance $\mathrm{Var}(X_j|\boldsymbol{X}_{-j})$, in which $\Omega = \Sigma^{-1}$ is the precision matrix. We note that τ_j^2 can be reasonably well estimated using nodewise regression, i.e., regressing X_j onto $\boldsymbol{X}_{-j}$. An unbiased estimator of τ_j^2 is $\widehat{\tau}_j^2 = \mathrm{RSS}_j/(n - p + 1)$, in which RSS_j denotes the residual sum of squares. $(\alpha_\star, \sigma_\star)$ is the unique solution of some deterministic system (see Equation (18) in Zhao et al. (2020)) depending on the underlying GLM as well as κ and ω. Sur and

Candès (2019) proposed the *ProbFrontier* method to estimate the signal strength ω based on the phase transition curve. With ω, we can numerically calculate $(\alpha_\star, \sigma_\star)$ so as to obtain asymptotic p-values for each feature.

We remark that the asymptotic normality presented in Equation (3.12) holds more generally for GLMs. However, it requires case-by-case study to derive the phase transition curve, as well as the system determining $(\alpha_\star, \sigma_\star)$. To the best of our knowledge, there is no unified approach to achieve this goal, and results are not known even for commonly-used GLMs including the multinomial logistic regression model and the Poisson regression model. Consequently, BHq can not be easily exercised in practice for the purpose of FDR control.

On the other hand, as discussed in Section 3.4.1, the data-splitting methods do not require estimating $(\alpha_\star, \sigma_\star)$, thus enjoys more general applicability for GLMs as long as the MLE exists for each part of the data. We detail the method in Algorithm 13, and show that it achieves an asymptotic FDR control in Proposition 3.4.2, under mild conditions specified in Assumption 3.4.2.

Assumption 3.4.2.

1. $1/c_1 \le \sigma_{\min}(\Sigma) \le \sigma_{\max}(\Sigma) \le c_1$ for some constant $c_1 > 0$.

2. The required number of null features, $\limsup p/p_0 < +\infty$.

Proposition 3.4.2. For any given FDR control level $q \in (0, 1)$, using Algorithm 13, we have

$$\limsup_{n,p\to\infty} \mathbb{E}\left[\frac{\#\{j : j \in S_0, j \in \widehat{S}_{\tau_q}\}}{\#\{j : j \in \widehat{S}_{\tau_q}\} \vee 1}\right] \le q$$

under Assumption 3.4.2.

Algorithm 13: FDR control for GLM in the moderate-dimensional setting.

1. Split the data set into two equal-sized groups $(\boldsymbol{y}^{(1)}, \boldsymbol{X}^{(1)})$ and $(\boldsymbol{y}^{(2)}, \boldsymbol{X}^{(2)})$.
2. For $j \in [p]$, regress $X_j^{(1)}$ onto $\boldsymbol{X}_{-j}^{(1)}$, and $X_j^{(2)}$ onto $\boldsymbol{X}_{-j}^{(2)}$. Let

$$\widehat{\tau}_j^{2(1)} = \frac{\text{RSS}_j^{(1)}}{n/2 - p + 1}, \quad \widehat{\tau}_j^{2(2)} = \frac{\text{RSS}_j^{(2)}}{n/2 - p + 1},$$

   in which RSS_j is the residual sum of squares.
3. Find the MLE $\widehat{\boldsymbol{\beta}}^{(1)}$ and $\widehat{\boldsymbol{\beta}}^{(2)}$ on each part of data. For $j \in [p]$, let

$$T_j{}^{(1)} = \widehat{\tau}_j^{(1)} \widehat{\beta}_j^{(1)}, \quad T_j{}^{(2)} = \widehat{\tau}_j^{(2)} \widehat{\beta}_j^{(2)}.$$

4. Calculate the mirror statistics $M_j = \left|T_j^{(1)} + T_j^{(2)}\right| - \left|T_j^{(1)} - T_j^{(2)}\right|$.
5. Select the features using Algorithm 9.

The high-dimensional setting

In the high-dimensional setting, we take a similar approach as in the case of the high-dimensional linear model, and use the debiasing method to symmetrize the Lasso estimator. The Lasso estimator for GLMs is defined as below,

$$\widehat{\boldsymbol{\beta}}(y, \boldsymbol{X}; \lambda) = \underset{\boldsymbol{\beta} \in \mathbb{R}^p}{\arg\min} \left\{ \frac{1}{2n} \sum_{i=1}^{n} \ell(y_i, \boldsymbol{x}_i^\intercal \boldsymbol{\beta}) + \lambda ||\boldsymbol{\beta}||_1 \right\}, \tag{3.13}$$

based upon which, the debiased Lasso estimator $\widehat{\boldsymbol{\beta}}^d$ is defined similarly as,

$$\widehat{\boldsymbol{\beta}}^d = \widehat{\boldsymbol{\beta}} - \widehat{\Omega} P_n \dot{\ell}_{\widehat{\boldsymbol{\beta}}}. \tag{3.14}$$

$\widehat{\Omega}$ analogizes to the "decorrelating" matrix D. Let $\Sigma = \mathbb{E}[\boldsymbol{X}_{\boldsymbol{\beta}^\star}^\intercal \boldsymbol{X}_{\boldsymbol{\beta}^\star}]/n$ be the population Hessian matrix. We follow the debiasing method in Van de Geer et al. (2014), and set $\widehat{\Omega}$ to be an estimator of $\Omega = \Sigma^{-1}$, constructed based on Algorithm 11. We note that the population version $\boldsymbol{\gamma}_j$ of $\widehat{\boldsymbol{\gamma}}_j$ is defined as below,

$$\boldsymbol{\gamma}_j = \underset{\boldsymbol{\gamma} \in \mathbb{R}^{p-1}}{\arg\min}\, \mathbb{E}\left[||\boldsymbol{X}_{\boldsymbol{\beta}^\star,j} - \boldsymbol{X}_{\boldsymbol{\beta}^\star,-j}\boldsymbol{\gamma}||_2^2\right]. \tag{3.15}$$

Let $\boldsymbol{\eta}_j = X_{\boldsymbol{\beta}^\star,j} - \boldsymbol{X}_{\boldsymbol{\beta}^\star,-j}\boldsymbol{\gamma}_j$ for $j \in [p]$. $\tau_j^2 = \mathbb{E}[||\boldsymbol{\eta}_j||_2^2/n]$ is the corresponding population version of $\widehat{\tau}_j^2$, calibrating the conditional variance $\mathrm{Var}(X_j|\boldsymbol{X}_{-j})$.

It requires to properly normalize the debiased Lasso estimator $\widehat{\boldsymbol{\beta}}^d$ so that the resulting mirror statistics scale in the same magnitude. Under Assumption 3.4.3, the asymptotic covariance of $\sqrt{n}\widehat{\boldsymbol{\beta}}^d$ is Ω following Van de Geer et al. (2014). Since

$$\sigma_j^2 := (\Omega\mathbb{E}[P_n\dot{\ell}_{\boldsymbol{\beta}^\star}\dot{\ell}_{\boldsymbol{\beta}^\star}^\top]\Omega)_{j,j} = (\Omega\Sigma\Omega)_{j,j} = \Omega_{j,j},$$

we normalize $\widehat{\beta}_j^{(d)}$ to $T_j = \sqrt{n}\widehat{\beta}_j^d/\widehat{\sigma}_j$, where $\widehat{\sigma}_j^2 = (\widehat{\Omega}P_n\dot{\ell}_{\widehat{\boldsymbol{\beta}}}\dot{\ell}_{\widehat{\boldsymbol{\beta}}}^\top\widehat{\Omega}^\top)_{j,j}$ serves as an estimator to σ_j^2.

Similar to the decomposition in Equation (3.7), we have $T_j = Z_j + \Delta_j$ in which

$$Z_j = -\frac{\sqrt{n}\Omega_{j,\cdot}P_n\dot{\ell}_{\boldsymbol{\beta}^\star}}{\sigma_j} = -\frac{\sqrt{n}}{\sigma_j}\sum_{i=1}^n \Omega_{j,\cdot}\boldsymbol{x}_i\dot{\ell}(y_i, \boldsymbol{x}_i^\intercal\boldsymbol{\beta}^\star), \tag{3.16}$$

where $\Omega_{j,\cdot}$ denotes the j-th row of Ω. For $j \in S_0$, Z_j no longer follows a normal distribution (conditioning on the design matrix) as in the case of the linear model. However, we can easily quantify the discrepancy between the law of Z_j and the standard normal distribution using the Berry-Essen Theorem. For the bias term Δ_j, we show that it is asymptotically negligible under Assumption 3.4.3.

Assumption 3.4.3.

1. The sparsity conditions in Assumption 3.4.1.

2. The regularity conditions on the loss function $\ell(u, v)$.

   (a) There exists $\delta \in (0, 1)$ such that

$$\sup_{(\boldsymbol{x},y)} \sup_{|v-\boldsymbol{x}^\intercal\boldsymbol{\beta}^\star|\leq\delta} \frac{|\ddot{\ell}(y, v) - \ddot{\ell}(y, \boldsymbol{x}^\intercal\boldsymbol{\beta}^\star)|}{|v - \boldsymbol{x}^\intercal\boldsymbol{\beta}^\star|} \leq 1. \tag{3.17}$$

   (b) There exists a constant $C_1 > 0$ such that

$$1/C_1 \leq \inf_{(\boldsymbol{x},y)} |\ddot{\ell}(y, \boldsymbol{x}^\intercal\boldsymbol{\beta}^\star)| \leq \sup_{(\boldsymbol{x},y)} |\dot{\ell}(y, \boldsymbol{x}^\intercal\boldsymbol{\beta}^\star)| \leq C_1, \tag{3.18}$$

   and

$$\sup_{(\boldsymbol{x},y)} \sup_{|v-\boldsymbol{x}^\intercal\boldsymbol{\beta}^\star|\leq\delta} |\ddot{\ell}(y, v)| \leq C_1. \tag{3.19}$$

3. The conditions on the design matrix $\boldsymbol{X}$ and the weighted design matrix $\boldsymbol{X}_{\boldsymbol{\beta}^\star}$.

   (a) There exists a constant $C_2 > 0$ such that

$$||\boldsymbol{X}||_\infty \leq C_2, \quad ||\boldsymbol{X}_{\boldsymbol{\beta}^\star}||_\infty \leq C_2, \quad ||\boldsymbol{X}_{\boldsymbol{\beta}^\star,-j}\boldsymbol{\gamma}_j||_\infty \leq C_2. \tag{3.20}$$

   (b) $1/C_3 \leq \sigma_{\min}(\Sigma) \leq \sigma_{\max}(\Sigma) \leq C_3$, for some constant $C_3 > 0$.

4. The required sample size. $\sqrt{n}/\log p \to \infty$.

Proposition 3.4.3. Under Assumption 3.4.1, we have $||\Delta||_\infty = o_p(1)$.

The proof of Proposition 3.4.3 can be found in the Appendix C. Therefore, for $j \in S_0$, T_j is asymptotically normal, thus asymptotically symmetric. Proposition 3.4.4 shows that the data-splitting method outlined in Algorithm 12 achieves an asymptotic FDR control.

Proposition 3.4.4. For any given FDR control level $q \in (0,1)$, using the debiasing method in Algorithm 12 for high-dimensional GLMs, we have

$$\limsup_{n,p\to\infty} \mathbb{E}\left[\frac{\#\{j : j \in S_0, j \in \widehat{S}_{\tau_q}\}}{\#\{j : j \in \widehat{S}_{\tau_q}\} \vee 1}\right] \leq q.$$

under Assumption 3.4.1.

The proof of Proposition 3.4.4 can be found in the Appendix C.

3.4.3 Gaussian graphical model

Suppose $\boldsymbol{X} = (X_1, \cdots, X_p)$ follows a p-dimensional multivariate normal distribution $N(\boldsymbol{\mu}, \Sigma)$ with $\Sigma = (\sigma_{ij})$. Without loss of generality, we assume $\boldsymbol{\mu} = 0$. One can define a corresponding Gaussian graphical model (V, E), in which the set of vertexes is $V = (X_1, \cdots, X_p)$, and there is an edge between two different vertexes X_i and X_j if X_i and X_j are conditionally independent, that is, $X_i \perp\!\!\!\perp X_j | \{X_k, k \neq i, j\}$. The estimation of the graphical structure is equivalent to the selection of the precision matrix $\Lambda = \Sigma^{-1} = (\lambda_{ij})$. For each vertex X_j, we can write $X_j = \boldsymbol{X}_{-j}^{\mathsf{T}} \boldsymbol{\beta}^j + \epsilon_j$, in which ϵ_j (independent of $\boldsymbol{X}_{-j}$) follows a centered normal distribution, and $\boldsymbol{\beta}^j = -\lambda_{jj}^{-1} \Lambda_{j,-j}$. Therefore, $\lambda_{i,j} = 0$ implies that X_i and X_j are conditionally independent. Let the neighborhood of each vertex X_j be $ne_j = \{k \in [p] : k \neq j,\ \beta_k^j \neq 0\}$, and its complement be $ne_j^c = \{k \in [p] : k \neq j,\ \beta_k^j = 0\}$.

Given i.i.d samples $\boldsymbol{X}_1, \cdots, \boldsymbol{X}_n$ from $N(\boldsymbol{\mu}, \Sigma)$, it is natural to consider first recovering the support of each $\boldsymbol{\beta}^j$ using feature selection methods such as Lasso (Meinshausen and Bühlmann, 2006), then combining all the nodewise selection results properly to estimate the global graph structure. In view of this, we propose an FDR control approach, with a designated FDR control level q, for the estimation of Gaussian graphical models. The proposed method mainly contains the following two steps:

1. Apply the proposed data-splitting approaches for linear models (see Section 3.4.1) to each nodewise edge selection, with designated FDR control level $q/2$. Denote the nodewise selection results as $\widehat{ne}_j = \{k \in [p] : k \neq j,\ \widehat{\beta}^j_k \neq 0\}$ for $j \in [p]$.

2. Combine the nodewise selection results using the OR rule to estimate the graph structure:

$$\widehat{E}_{\text{OR}} = \{(i,j) \in [p] \times [p],\ i \in \widehat{ne}_j \text{ or } j \in \widehat{ne}_i\}. \tag{3.21}$$

A heuristic justification of the proposed approach is given as below:

$$\begin{aligned}
\text{FDP} &= \frac{\#\{(i,j) \in \widehat{E}_{\text{OR}}, (i,j) \notin E\}}{|\widehat{E}_{\text{OR}}| \vee 1} \leq \frac{\sum_{j=1}^p \#\{i \in ne^c_j, i \in \widehat{ne}_j\}}{\frac{1}{2}\sum_{j=1}^p \#\{i \in \widehat{ne}_j\} \vee 1} \\
&= \frac{\sum_{j=1}^p \#\{i \in ne^c_j, M_{ji} > \tau^j_{q/2}\}}{\frac{1}{2}\sum_{j=1}^p \#\{M_{ji} > \tau^j_{q/2}\} \vee 1} \approx \frac{\sum_{j=1}^p \#\{i \in ne^c_j, M_{ji} < -\tau^j_{q/2}\}}{\frac{1}{2}\sum_{j=1}^p \#\{M_{ji} > \tau^j_{q/2}\} \vee 1} \\
&\leq 2 \max_{j \in [p]} \frac{\#\{i \in ne^c_j, M_{ji} < -\tau^j_{q/2}\}}{\#\{M_{ji} > \tau^j_{q/2}\} \vee 1} \leq q.
\end{aligned} \tag{3.22}$$

For $j \in [p]$, $\tau^j_{q/2}$ is the cutoff of the mirror statistics in the nodewise edge selection of the vertex X_j, and M_{ji} is the mirror statistic associated with X_i for $i \in [p]\backslash\{j\}$. The first inequality is based on the fact that each edge can be selected at most twice. The approximation in the middle utilizes the symmetric property of the mirror statistics. The second to last inequality comes from the elementary inequality $\sum_n a_n / \sum_n b_n \leq \max_n a_n/b_n$ for $a_n \geq 0,\ b_n > 0$.

In the following, we assume that the Lasso + OLS procedure is used in each nodewise selection. We first show that the symmetric property of the mirror statistics, or equivalently the screening property of the Lasso selection, is simultaneously satisfied in all p nodewise selections, with probability approaching 1 as $p \to \infty$, under the following assumptions.

Assumption 3.4.4.

(a) Regularity condition. There exist positive constants c_0, c_1, c_2 such that

$$\lambda_{\min}(\Sigma) \geq c_0, \quad \max_{1\leq j\leq p} \sigma_{jj} \leq c_1, \quad \max_{1\leq j\leq p} \lambda_{jj} \leq c_2.$$

(b) Sparsity condition. There exists some $\xi \in [0, 1)$ such that

$$\max_{1\leq j\leq p} |ne_j| = O(n_1^{\xi/2}).$$

(c) The minimum-signal strength condition:

$$\min\{|\lambda_{ij}| : \lambda_{ij} \neq 0, 1 \leq i < j \leq p\} \gtrsim \sqrt{\frac{\log p}{n_1^{1-\xi}}}.$$

Here n_1 is the sample size of the first half of data. Similar assumptions as Assumption 3.4.4(a) and 3.4.4(b) also appear in Liu (2013) and Meinshausen and Bühlmann (2006), respectively. Assumption 3.4.4(c) calibrates the minimum strengthen of the signal required for detection. It is commonly referred to as the "beta-min" condition in high-dimensional linear regressions (Dezeure et al., 2015). Under Assumption 3.4.4, we have the following proposition.

Proposition 3.4.5. Under Assumption 3.4.4, if we apply the Lasso + OLS procedure to each nodewise regression with the regularizaton parameter properly chosen in the order of $O(\sqrt{\log p/n_1})$, then in the regime $n_1,\ p \to \infty$ and $\log p/n_1^{1-\xi/2} \to 0$, the symmetric assumption (Assumption 3.3.1) is simultaneously satisfied in all nodewise edge selection with probability approaching 1.

The proof of Proposition 3.4.5 relies on Theorem 1 in Raskutti et al. (2010) and Theorem 7.2 in Bickel et al. (2009), and is postponed to the Appendix C.

Under the symmetric assumption, we require some technical assumptions to asymptotically control the FDR of the estimated graph structure. The essential assumption is that for any $p \in \mathbb{N}_+$, $j \in [p], t \in \mathbb{R}$, the set of Bernoulli random variables $\{1(M_{ji} > t), i \in ne_j^c\}$ are only weakly depen-

dent. For any subset $A \subseteq ne_j^c$, $k \in ne_j^c \backslash A$, and $t \in \mathbb{R}$, we define the following random variable to measure the conditional dependency,

$$\Delta_{p,j}^{k,A} = \left| \sum_{i \in ne_j^c} \mathbb{P}\left(M_{ji} > t \middle| M_{jk} > t, \mathscr{F}_A\right) - \sum_{i \in ne_j^c} \mathbb{P}\left(M_{ji} > t \middle| M_{jk} \leq t, \mathscr{F}_A\right) \right|, \tag{3.23}$$

in which $\mathscr{F}_A$ denotes the sigma algebra generated by $\{1(M_{ji} > t),\ i \in A\}$.

Assumption 3.4.5.

(a) There exist constants $C > 0$ and $\alpha \in (0, 1/2)$, such that for any $p \in \mathbb{N}_+$, $j \in [p]$, subset $A \subseteq ne_j^c$, $k \in ne_j^c \backslash A$, and $t \in \mathbb{R}$, we have $\Delta_{p,j}^{k,A} \leq C|ne_j^c|^\alpha$ almost surely.

(b) $\lim_{p\to\infty} \min_{j\in[p]} |ne_j^c|^{1-2\alpha} / \log p = \infty$.

We remark that when all the indicators $\{1(M_{ji} > t), i \in ne_j^c\}$ are independent, $\Delta_{p,j}^{k,A} \equiv 0$ for any subset $A \subseteq ne_j^c$ and $k \in ne_j^c \backslash A$. On the other hand, when all the indicators are perfectly correlated, i.e., $M_{j1} = \cdots = M_{jp}$, we have $\Delta_{p,j}^{k,\emptyset} = |ne_j^c|$ for any $k \in ne_j^c$. We assume $\Delta_{p,j}^{k,A} \leq C|ne_j^c|^\alpha$ for some $\alpha \in (0, 1/2)$ almost surely to allow some weak dependency among $\{1(M_{ji} > t), i \in ne_j^c\}$. Under Assumption 3.4.5, we have the following proposition. The proof is postponed to the Appendix C.

Proposition 3.4.6. Assume the symmetric assumption is satisfied in each nodewise edge selection. If we set the designated FDR control level for each nodewise edge selection to be $q/2$ and apply the Lasso + OLS procedure, then under Assumption 3.4.5, we can control the FDR of the estimated edge set $\widehat{E}_{\mathrm{OR}}$ at the level q asymptotically as $p \to \infty$.

We note that the nodewise selection procedure and the GFC approach proposed in Liu (2013) are effective in quite different scenarios. GFC tends to work well only if the underlying true graph is ultra-sparse, in the order of $o(\sqrt{n}/(\log p)^{3/2})$. The nodewise selection procedure is capable of handling the case when the graph is not too sparse, but can suffer from the ultra-sparsity. This can be seen by

considering an extreme scenario where $|ne_j| = 1$ for some vertex X_j. Suppose with probability θ_j, $\max\{M_{ji} : i \in ne_j^c, M_{ji} > 0\}$ is strictly larger than $-\min\{M_{ji} : i \in ne_j^c, M_{ji} < 0\}$. Then the FDR of the nodewise selection of the vertex X_j is at least $0.5\theta_j$, since the FDP is at least 0.5. Therefore, if $\theta_j > q$, it is impossible to control FDR below $q/2$. We note that in the case where all $\{M_{ji} : i \in ne_j^c\}$ are independent and symmetric about 0, $\theta_j = 0.5$. A similar issue also exists in general knockoff-based approaches, which is pointed out in Remark 3.2 in Li and Maathuis (2019).

3.4.4 Deep neural network

In this section, we integrate the proposed data-splitting approaches into neural networks to achieve reproducible feature selection. We restrict ourselves here to fully-connected forward neural networks, although the procedure is easily applicable to more complex networks including convolutional neural networks and recurrent neural networks.

The recipe of our method is to first split the data into two halves, $(\boldsymbol{X}^{(1)}, \boldsymbol{y}^{(1)})$ and $(\boldsymbol{X}^{(2)}, \boldsymbol{y}^{(2)})$, and then train two neural networks with the same structure independently for each part of the data. Denote the number of units in each layer as m_h for $h \in [0 : k]$. The input layer is the 0-th layer with length $m_0 = p$, and the response layer is the k-th layer with length $m_k = 1$. For $h \in [k]$, let $\boldsymbol{H}^{(h)}$ be the vector of hidden units in layer h. In particular, we also write $\boldsymbol{H}^{(0)} = (X_1, \cdots, X_p)$. Denote $\boldsymbol{W}^{(h)}$ as the $m_{h-1} \times m_h$ weight matrix between layer $h-1$ and layer h. Let ϕ_h be the activation function used between layer $h-1$ and layer h. Thus, we have $\boldsymbol{H}_\ell^{(h)} = \phi_h(\boldsymbol{H}^{(h-1)\mathsf{T}}\boldsymbol{W}_{\cdot\ell}^{(h)})$ for $\ell \in [m_h]$. Let the estimated weights be $\{\boldsymbol{W}^{(1,h)}\}_{h=1}^k$ and $\{\boldsymbol{W}^{(2,h)}\}_{h=1}^k$, where $\boldsymbol{W}^{(1,h)}$ and $\boldsymbol{W}^{(2,h)}$ are the $m_{h-1} \times m_h$ weight matrices between layer $h-1$ and layer h in the two trained neural networks, respectively. A cartoon illustration is given in Figure 3.3 .

The main challenge of applying our FDR control approach to neural networks is to define a proper mirror statistic. A naive way is to calculate the mirror statistic only using the weights between the input layer and the first hidden layer. However, this proposal is problematic because the h-th layer's hidden

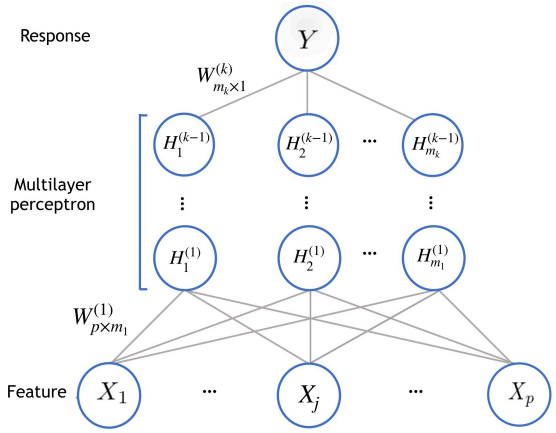


Figure 3.3: A demonstration of the fully-connected neural network.

units in the two neural networks, which are trained independently with different data and potentially different initialization, do not necessarily correspond to each other. Therefore, it is not sensible to directly contrast $\boldsymbol{W}^{(1,1)}_{j\ell}$ to $\boldsymbol{W}^{(2,1)}_{j\ell}$.

We discuss two alternatives to bypass the aforementioned non-identifiability problem of the hidden units. The first one is to use weight multiplication as in Lu et al. (2018). For each feature X_j, we define its "impact" coefficient $\widehat{\beta}^{(1)}_j$ to be the j-th coordinate of the matrix product among all weights, that is, $\boldsymbol{W}^{(1,1)}\boldsymbol{W}^{(1,2)}\cdots\boldsymbol{W}^{(1,k)}$. Similarly, we can define $\widehat{\beta}^{(2)}_j$. This construction of the "impact" coefficient may only apply to fully-connected networks since in more complex networks such as convolutional neural networks, the weight tensor can be 3 or 4 dimensions, thus the multiplication between weight tensors are not well defined.

The second construction of the "impact" coefficient, which is able to handle more complex network structures, uses the influence function (Hechtlinger, 2016). It essentially calibrates the influence of feature X_j to the response variable y using gradient. Mathematically, the influence function, $\boldsymbol{\mathcal{I}} = (\mathcal{I}_1, \cdots, \mathcal{I}_p)$, is a vector of length p, with each coordinate defined to be $\mathcal{I}_j = \mathbb{E}[\partial f(X)/\partial X_j]$, where the expectation is taken with respect to the joint distribution of $(y, X_1, \cdots, X_p)$. f represents the neural network model. Specifically, for fully-connected neural networks, the influence function

can be explicitly written as:

$$\boldsymbol{\mathcal{I}} = \mathbb{E}\left[\boldsymbol{W}^{(k)}\Phi^{(k)}\boldsymbol{W}^{(k-1)}\Phi^{(k-1)}\cdots\boldsymbol{W}^{(1)}\Phi^{(1)}\right]. \tag{3.24}$$

$\Phi^{(h)}$ is a $m_h \times m_h$ diagonal matrix with ℓ-th element being the gradient of ϕ_h evaluated at the point $\boldsymbol{H}^{(h-1)\intercal}\boldsymbol{W}^{(h)}_{\cdot\ell}$ for $\ell \in [m_h]$. We proceed to approximate the influence function $\boldsymbol{\mathcal{I}}$ using the sample mean across training data, and its j-th coordinate will serve as the "impact" coefficient for feature X_j. We note that in most popular deep learning platforms, the gradient can be calculated using built-in functions, thus the computation is fairly user-friendly.

After obtaining the "impact" coefficient, the mirror statistic can be calculated following Equation (3.2). We note that the weight multiplication approach, which simply ignores all the activation functions between hidden layers, can be considered as a special case of the influence function approach. Although the two approaches seem to perform similarly in fully-connected forward networks based on our limited experiences, we believe that the influence function approach has more general applicability in real data examples.

3.5 Illustrations

3.5.1 Linear model

We consider the high-dimensional linear regression using Lasso. Throughout we fix the sample size $n = 500$ and the number of signals (sparsity) $s^\star = 50$. We simulate the response variable $\boldsymbol{y}$ from the linear model $\boldsymbol{y}_{n\times 1} = \boldsymbol{X}_{n\times p}\boldsymbol{\beta}_p + \boldsymbol{\epsilon}_{n\times 1}$, in which the noise $\boldsymbol{\epsilon}$ are sampled from $N(0, I_n)$. We set $\beta_j = 0$ if $j > s^\star$, and sample β_j from $N(0, \delta\sqrt{\log p/n})$ for $j \in [s^\star]$. We consider the following two scenarios:

1. Vary the pairwise correlation among features. We independently sample each row of the design

matrix $\boldsymbol{X}_{n\times p}$ from $N(0_p, \Sigma)$, where $\Sigma_{jj} = 1$ and $\Sigma_{ij} = \rho$ for all $i \neq j$. We test out $\rho \in \{0.0, 0.2, 0.4, 0.6, 0.8\}$. Throughout this scenario, we fix the signal strength and set $\delta = 10$.

2. Vary the signal strength. We test out $\delta \in \{4, 8, 12, 16, 20\}$. Throughout this scenario, we sample independently each row of $\boldsymbol{X}_{n\times p}$ from $N(0_p, \Sigma)$, where $\Sigma_{jj} = 1$ and $\Sigma_{ij} = 0.5$ for all $i \neq j$..

For each scenario, we also vary the total number of features $p \in \{500, 1000, 1500, 2000\}$.

We compare eight methods within three classes: (1) BHq and BYq; (2) the knockoff filter including the model-X knockoff filter (M-knockoff) (Candes et al., 2018), and the fixed-design knockoff filter (F-knockoff) based on data splitting, with the data recycling strategy proposed in Barber and Candès (2019); (3) DS and MDS. For BHq and BYq, we first randomly split the data into two halves. We then use one half of the data for signal screening using Lasso, and calculate the p-values for the selected features by running a OLS regression on the other half of the data. The corresponding multiple data-splitting versions (MBHq and MBYq) following Meinshausen et al. (2009), which combines the p-values obtained across multiple data splits, are also tested out using the R package *hdi*. We implement the model-X knockoff filter using the R package *knockoff*. For the multiple data-splitting approaches MBHq, MBYq and MDS, we independently split the data 50 times. The designated FDR control level is set to be $q = 0.1$ in all simulation settings.

Figure 3.4 summarizes the results for the varying correlation scenario, in which $p \in \{500, 2000\}$ and $\rho \in \{0.0, 0.8\}$. More detailed results are given in Table C.3 in the Appendix C. The FDRs of all eight methods are reasonably under control across different simulation settings. The empirical performances show that MDS is promising. First, it significantly improves DS, in the sense that it simultaneously reduces the FDR and boosts the power. Second, from Table C.3, we see that it has the highest power among all 8 methods when features are correlated ($\rho \geq 0.2$). When features are independent ($\rho = 0.0$), the power of MDS is second to the best, slightly lower than the power of the

model-X knockoff filter. We note that $\rho = 0.0$ is considered to be the easiest setting for generating knockoff features as all the features are independent.

The following observations are also worthwhile to mention. First, when the correlation ρ is large, the model-X knockoff filter tends to be very conservative. From Table C.3, we see that when $\rho \geq 0.4$, except for the relatively low-dimensional case $p = 500$, the model-X knockoff filter becomes powerless. On the other hand, the fixed-design knockoff filter is more robust to the dependency between features, and performs competitively across different simulation settings. Second, BYq is more conservative compared to BHq, which consistently yields a smaller FDR but also a lower power compared to BHq. We find that the p-value aggregation strategy proposed in Meinshausen et al. (2009) is effective. In most cases, MBHq and MBYq enjoy zero FDRs and competitive powers. The results for the varying signal strength scenario are summarized in Figure 3.5 and Table C.4 in the Appendix C, which provides similar evidence to demonstrate the effectiveness of the proposed data-splitting approaches.

3.5.2 Generalized linear model

The moderate-dimensional setting

Logistic regression model. We consider two moderate-dimensional scenarios for the logistic regression model. The first scenario is the classical small-n-and-p setting, in which the sample size $n = 500$, the dimension $p = 60$, and the corresponding dimension-to-sample ratio is $\kappa = p/n = 0.12$. The second scenario concerns with the large-n-and-p setting, in which the sample size $n = 3000$, the dimension $p = 500$, and the corresponding dimension-to-sample ratio is $\kappa = p/n = 1/6$.

In both scenarios, we consider five competing methods based on the MLE, including the single data-splitting method (DS) and its multiple version (MDS), the Benjamini-Hochberg procedure along with its adjusted version, and the Gaussian mirror (GM) method. The standard Benjamini-Hochberg

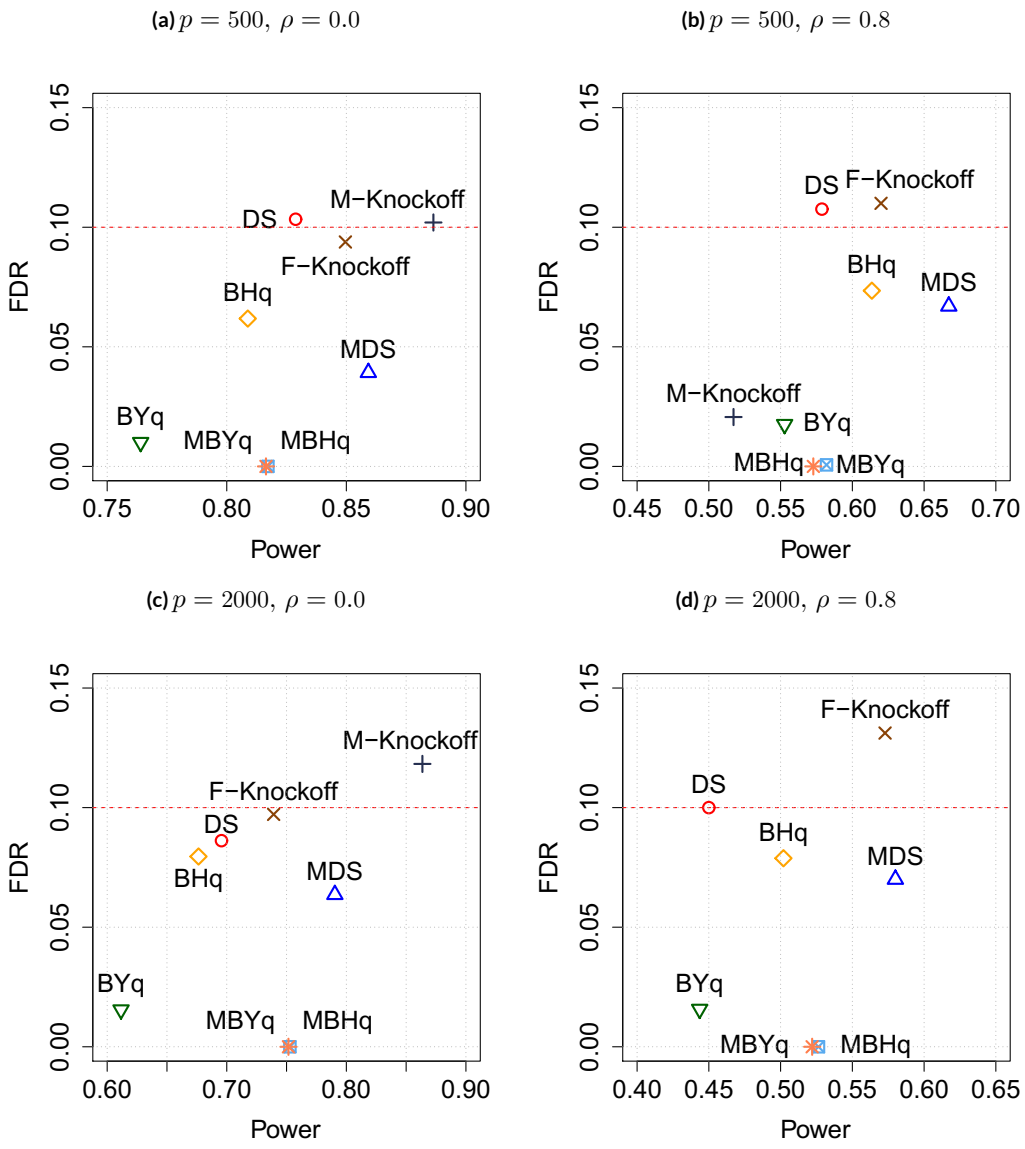


Figure 3.4: Empirical FDRs and powers on the linear model, where the design matrix has pairwise constant correlation ρ. The number of true features is 50 across all settings. The signal-to-noise ratio is $10 \times \sqrt{\log p/n}$. The designated FDR control level is $q = 0.1$ (the red dashed line). The eight methods are, the single data-splitting method (DS), the multiple data-splitting method (MDS), the model-X knockoff filter (M-Knockoff), the fixed-design knockoff filter with data recycling (F-Knockoff), the Benjamini-Hochberg method (BHq) and its multiple data-splitting version (MBHq), the Benjamini-Yekutieli method (BYq) and its multiple data-splitting version (MBYq). The reported results are the empirical means of 50 independent runs. In panel (d), the coordinates (Power, FDR) of the model-X knockoff filter (M-Knockoff) are (0.05, 0.00), which is out of range of the figure.

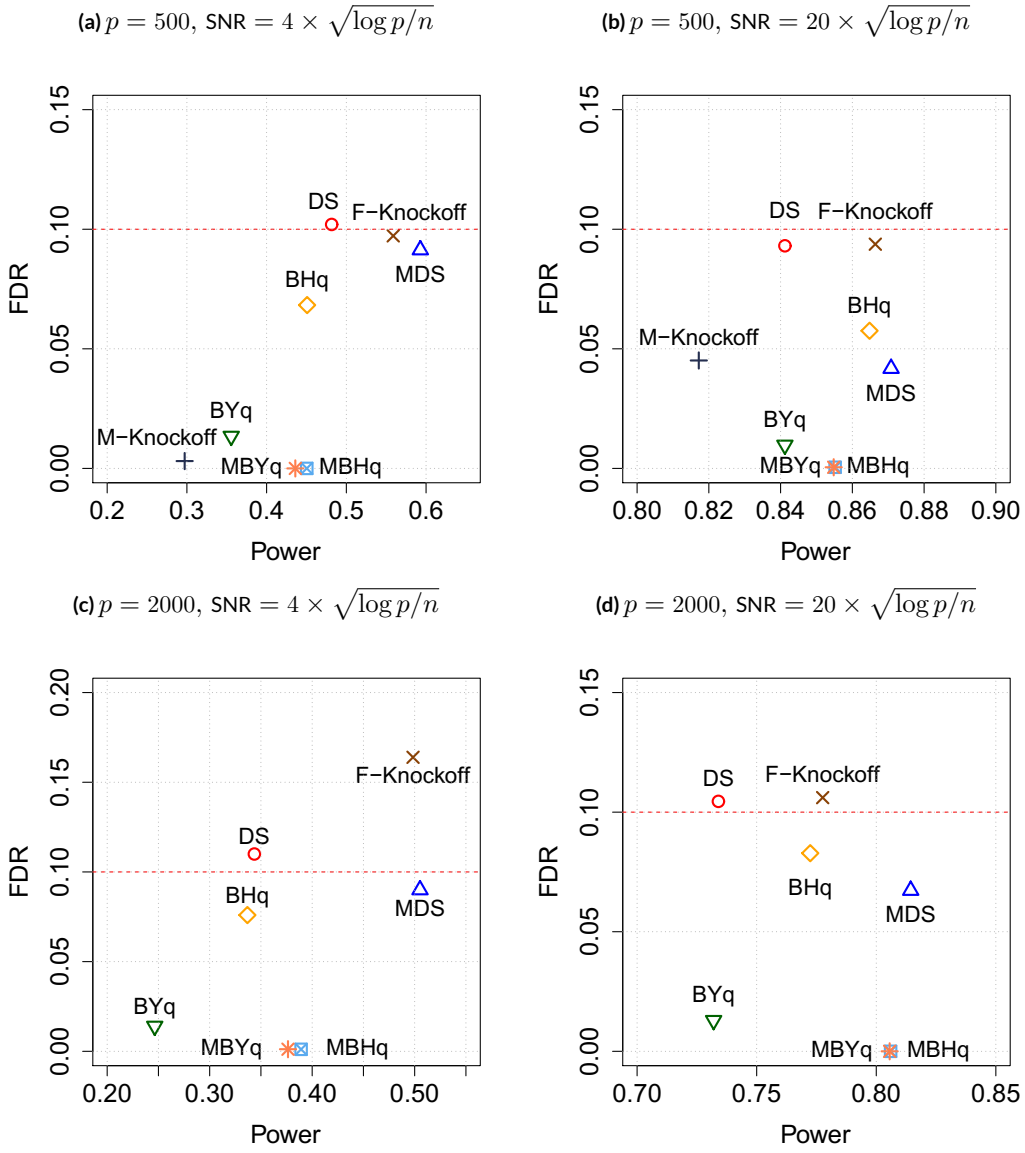


Figure 3.5: Empirical FDRs and powers on the linear model, where the design matrix follows a multivariate normal distribution with constant pairwise correlation 0.5. SNR represents the signal-to-noise ratio. The number of true features is 50 across all settings. The designated FDR control level is $q = 0.1$. The eight methods are, the single data-splitting method (DS), the multiple data-splitting method (MDS), the model-X knockoff filter (M-Knockoff), the fixed-design knockoff filter with data recycling (F-Knockoff), the Benjamini-Hochberg method (BHq) and its multiple-splitting version (MBHq), the Benjamini-Yekutieli method (BYq) and its multiple-splitting version (MBYq). The reported results are the empirical means of 50 independent runs. In panel (c) and panel (d), the coordinates (Power, FDR) of the model-X knockoff filter (M-Knockoff) are (0.04, 0.00) and (0.19, 0.00), respectively, which are out of range of the corresponding figures.

procedure (BHq) utilizes the classical asymptotic p-values calculated via the Fisher information, whereas the adjusted Benjamini-Hochberg procedure (ABHq) is based on the adjusted asymptotic p-values (Sur and Candès, 2019).

Each row of the design matrix independently follows the multivariate normal distribution with constant pairwise correlation ρ. In particular, we test out the cases in which ρ increases from 0.0 to 0.5 incremented by 0.1. The mean and the variance of each feature are 0 and $1/n$, respectively. There are in total 50 relevant features, and the corresponding true regression coefficients are independently sampled from ± 6.5 with equal probability. The designated FDR control level is 0.1 across all settings.

The empirical FDRs and powers of different methods in the small-n-and-p setting are summarized in Figure 3.6. The FDRs of the five competing methods are under control across all settings. In terms of power, BHq and MDS are the two leading methods and perform similarly. GM is as powerful as BHq and MDS when the correlation among the features is small (e.g., $\rho \leq 0.2$), but appears to be inferior when the correlation becomes larger (e.g., $\rho \geq 0.3$). We see that ABHq is less powerful than BHq. One possible reason is that the asymptotics for the p-value adjustment is not ready to kick in when the sample size n and the dimension p are relatively small.

The empirical FDRs and powers of different methods in the large-n-and-p setting are summarized in Figure 3.7. We see that BHq lost FDR control, which implies that the classical asymptotic p-value of the null feature, calculated based on the Fisher information, is non-uniform and skew to the left. On the other hand, ABHq still enjoys FDR control and also has the best power, which verifies the adjusted asymptotic distribution of the MLE derived in Sur and Candès (2019). MDS is the second-best method, and is more robust compared to GM when the pairwise correlation among the features is high. In both the small-n-and-p setting and the large-n-and-p setting, we observe that MDS performs significantly better than DS, which simultaneously reduces the FDR and boots the power.

Negative binomial regression model. We consider a negative binomial regression model, in which the dispersion parameter is 2, i.e., the target number of successful trials is 2. We set the sample size

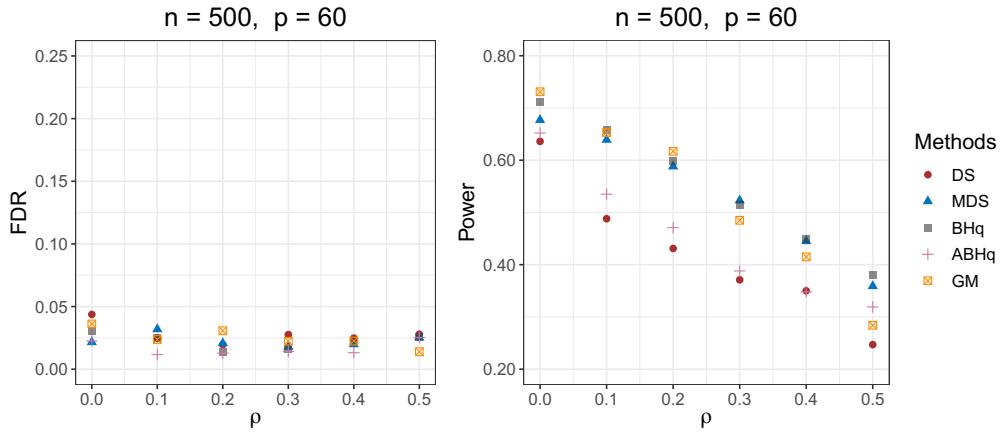


Figure 3.6: Empirical FDRs and powers on the logistic regression model in the small-n-and-p setting. Each row of the design matrix independently follows the multivariate normal distribution with constant pairwise correlation ρ. The number of relevant features is 50, and the true regression coefficients are independently sampled from ± 6.5 with equal probability. The designated FDR control level is $q = 0.1$. The reported results are the empirical means of 50 independent runs.

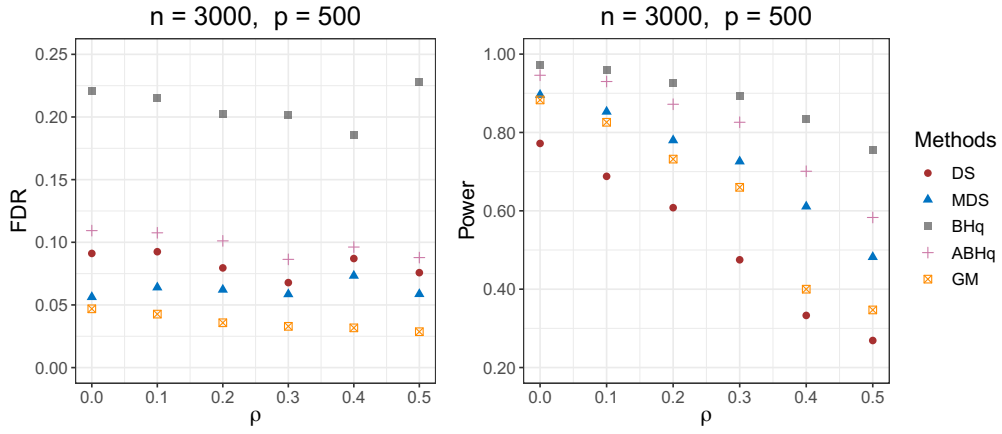


Figure 3.7: Empirical FDRs and powers on the logistic regression model in the large-n-and-p setting. Detailed simulation settings can be found in the caption of Figure 3.6.

$n = 3000$, and the dimension $p = 500$, thus the corresponding dimension-to-sample ratio is $\kappa = p/n = 1/6$. We simulate the design matrix the same as the case of the logistic regression model. There are in total 50 relevant features, and the corresponding true regression coefficients are independently sampled from ± 6 with equal probability. The designated FDR control level is 0.1 across all settings.

We consider four competing methods based on the MLE, including the single data-splitting method (DS) and its multiple version (MDS), the Benjamini-Hochberg procedure (BHq), and the Gaussian mirror (GM) method. We note that BHq is based upon the classical asymptotic p-values calculated based on the Fisher information. Although we expect such p-values to be non-uniform for the null features, the exact asymptotic distribution of the MLE under this moderate-dimensional setting has not been derived in the literature, thus no proper adjustment of the p-values exists to the best of our knowledge.

The empirical FDRs and powers of different methods are summarized in Figure 3.8. We see that DS, MDS and GM enjoys FDR control, whereas BHq not, because of the non-uniformity (skew to the left) of the p-values for the null features. Among the methods with FDR control, GM has the leading performance, with power slightly higher than MDS across all settings.

The high-dimensional setting

Logistic regression model. We consider a high-dimensional logistic regression model with the sample size $n = 800$, and the dimension $p \in \{800, 2000\}$. Each row of the design matrix independently follows the multivariate normal distribution $N(0, \Sigma)$. We consider a similar setup as in Ma et al. (2020), in which $\Sigma = 0.1 \times \Sigma_B$, where Σ_B is a $p \times p$ blockwise diagonal matrix of 10 identical unit diagonal Toeplitz matrices whose off-diagonal entries descend from 0.08 to 0. We test out different levels of signal sparsity, that is, the number of relevant features takes values from 40 to 80 incremented by 10, and the corresponding true regression coefficients are independently sampled from ± 4 with equal probability.

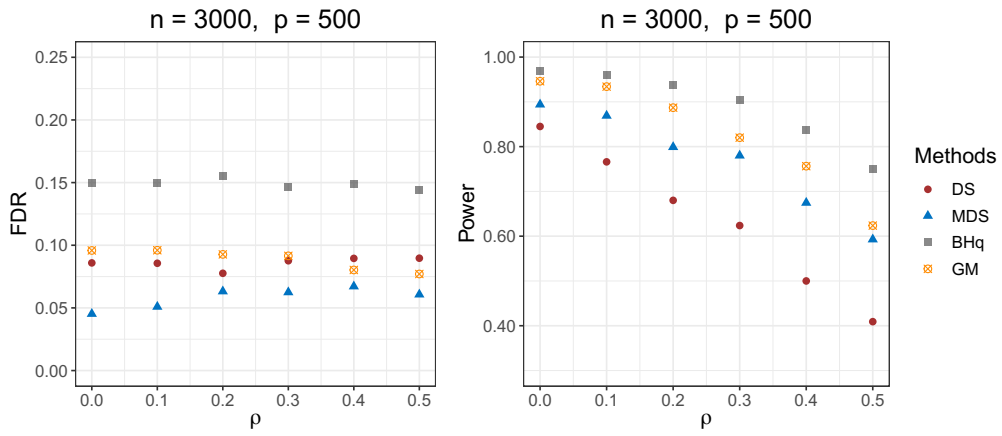


Figure 3.8: Empirical FDRs and powers on the negative binomial regression model. Each row of the design matrix independently follows the multivariate normal distribution with constant pairwise correlation ρ. The number of relevant features is 50, and the true regression coefficients are independently sampled from ± 6 with equal probability. The designated FDR control level is $q = 0.1$. The reported results are the empirical means of 50 independent runs.

We consider five competing methods, including the single data-splitting method (DS) and its multiple version (MDS), two Benjamini-Hochberg procedures (BHq-I and BHq-II), and the model-X knockoff filter. The data-splitting methods are based on the debiasing Lasso estimator constructed using Algorithm 12. BHq-II also uses the same debiasing approach, whereas BHq-I refers to the method proposed in Ma et al. (2020), which utilizes a different debiasing approach. For the model-X knockoff filter, we use the second-order method to create multivariate normal knockoffs. Throughout, the designated FDR control level is 0.1.

The empirical FDRs and powers of different methods are summarized in Figure 3.9. We see that except for BHq-II, all the other methods enjoy FDR control. In terms of power, when $p = 800$, MDS and the model-X knockoff filter perform similarly, and appear to be more powerful than DS and BHq-I. We find that MDS is more robust to the potential high dimensionality of the selection problem than the model-X knockoff filter. In particular, when $p = 2000$, MDS has the highest power among the methods with FDR control, whereas the model-X knockoff filter becomes the least

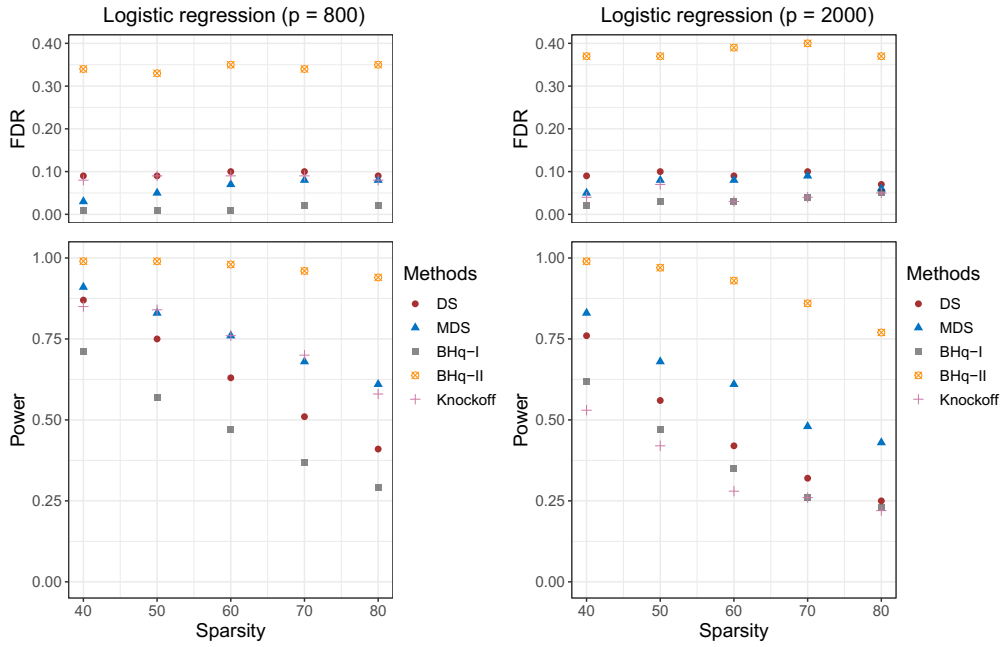


Figure 3.9: Empirical FDRs and powers on the logistic regression model with the sample size $n = 800$. Each row of the design matrix independently follows the multivariate normal distribution $N(0_p, 0.1 \times \Sigma)$ with Σ being a block diagonal matrix. The x-axis refers to the signal sparsity, i.e., the number of relevant features. The true regression coefficients for the relevant features are set to be $\beta_j^\star = \pm 4$ with equal probability. The designated FDR control level is $q = 0.1$. The reported results are the empirical means of 50 independent runs.

powerful method. We note that even DS is more powerful than BHq-I, which implies that the p-values constructed following Ma et al. (2020) can be highly non-informative (skew to the right) in the finite-sample case.

3.5.3 Gaussian graphical model

We consider two types of graphs:

1. Banded graph. The precision matrix Λ is constructed such that $\lambda_{jj} = 1$, $\lambda_{ij} = \text{sign}(a) \cdot |a|^{|i-j|/c}$ if $0 < |i-j| \leq \rho$, and $\lambda_{ij} = 0$ if $|i-j| > \rho$. Following Li and Maathuis (2019), we set $c = 1.5$ throughout this simulation study. Other parameters including the sample size

n, the dimension p, the signal strength (partial correlation) a, and the nodewise sparsity ρ will be specified case by case.

2. Blockwise diagonal graph. The precision matrix Λ is blockwise diagonal with equally sized squared blocks generated in the same fashion. We fix the block size to be 25×25 throughout this simulation study. In each block, all the diagonal elements are set to be 1, and the off-diagonal elements are independently sampled from some distribution u specified case by case.

The designated FDR control level is set to be $q = 0.2$. We note that for both graphs, the resulting precision matrix Λ is not necessarily positive definite. If $\lambda_{\min}(\Lambda) < 0$, we re-set $\Lambda \leftarrow \Lambda + (\lambda_{\min}(\Lambda) + 0.005)I_p$ following Liu (2013).

Three classes of competing methods are tested out: (1) DS and MDS; (2) BHq and BYq; (3) GFC-L and GFC-SL (two FDR control methods for Gaussian graphical model using Lasso or scaled Lasso proposed in Liu (2013)). The p-values used in BHq and BYq are calculated based on the pairwise partial correlation test using the R package *ppcor* (Kim, 2015). We use the R package *SILGGM* (Zhang et al., 2018) to implement GFC-L and GFC-SL.

For the banded graph, we test out the following four settings:

(a) Fix $p = 100$, $\rho = 8$, $a = -0.6$. and vary the sample size $n \in \{500, 1000, 1500, 2000, 2500\}$.

(b) Fix $n = 1000$, $\rho = 8$, $a = -0.6$, and vary the dimension $p \in \{50, 100, 150, 200, 250\}$.

(c) Fix $n = 1000$, $p = 100$, $a = -0.6$, and vary the nodewise sparsity $\rho \in \{4, 6, 8, 10, 12\}$.

(d) Fix $n = 1000$, $p = 100$, $\rho = 8$, and vary the signal strength $a \in \{-0.5, -0.6, -0.7, -0.8, -0.9\}$.

For the blockwise diagonal graph, we fix $n = 500$, and test out three different scenarios, in which the sampling distribution u of the off-diagonal elements are set to be Unif$(-0.8, -0.4)$ (negative partial correlation), Unif$(0.4, 0.8)$ (positive partial correlation), and Unif$((-0.8, -0.4) \cup (0.4, 0.8))$ (balanced partial correlation), respectively. For MDS, we independently split the data 50 times and aggregate the selection results using Algorithm 10.

The results for the banded graphs are summarized in Figure 3.10. DS and MDS outperform the four competing methods across all simulation settings. In particular, MDS consistently yields a lower FDR and a higher power compared to DS. BYq has the lowest FDR, but appears to be too conservative as it also has the lowest power. FDRs of GFC-L, GFC-SL, and BHq are similar to each other, and are higher than FDRs of DS and MDS. In terms of power, GFC-L and GFC-SL perform similar, having a slightly higher power than BHq. Panel (d) in Figure 3.10 is interesting, since the powers of BHq and BYq have an opposite trend compared to the other methods. One possible reason is that when we decrease the signal strength, or equivalently increase a from -0.9 to -0.5, the pairwise correlations decrease from 0.54 to 0.20 so that the independent p-value assumption becomes better justified. Therefore, we see a power increase for BHq and BYq.

The results for the blockwise diagonal graphs are summarized in Figure 3.11. In the settings with balanced signals and negative signals, MDS outperforms all the other competing methods. In particular, in the setting with negative signals, DS and MDS are the only two effective methods with reasonable powers. In the setting with positive partial correlation, BHq and BYq are the two leading methods. In this scenario, DS and MDS perform reasonably well only in the low-dimensional setting.

3.5.4 Deep neural network

We consider the single-index model $y = f(\boldsymbol{x}^\intercal \boldsymbol{\beta}) + \epsilon$, in which $f(t)$ is some unknown link function, and ϵ is the noise. In this simulation study, we test out three different cases detailed below.

1. Power function. We set $f_1(t) = t^3/2$.
2. Exponential function. We set $f_2(t) = \exp(t/10)$.
3. Sigmoid function. We set $f_3(t) = 1/(1 + \exp(-t))$.

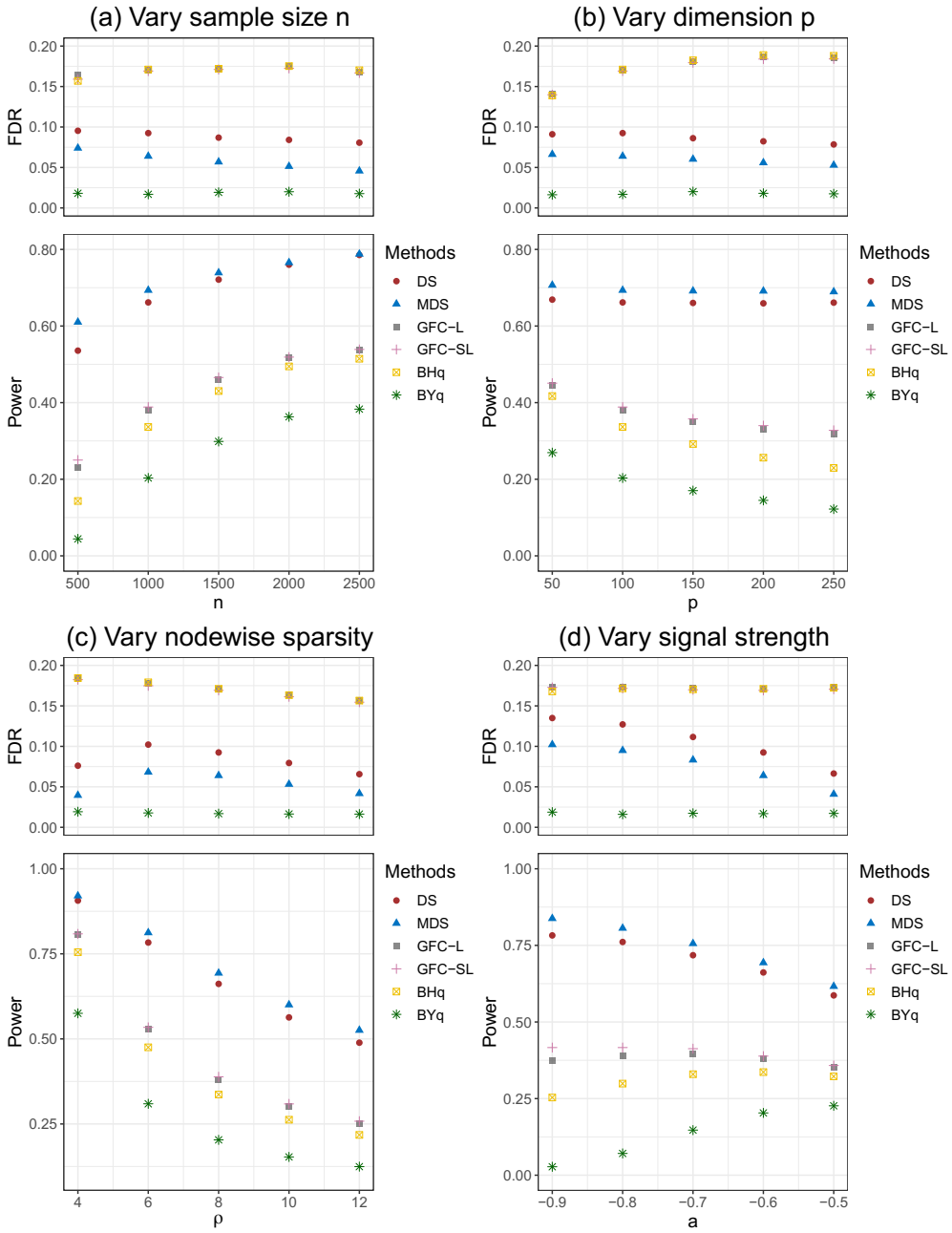


Figure 3.10: Empirical FDRs and powers on the banded graphs of the six methods, the single data-splitting method (DS), the multiple data-splitting method (MDS), the GFC methods with Lasso (GFC-L) and scaled Lasso (GFC-SL) proposed in Liu (2013), the Benjamini-Hochberg method (BHq), and the Benjamini-Yekutieli method (BYq). The designated FDR control level is $q = 0.2$ in all settings. The reported results are the empirical means of 50 independent runs.

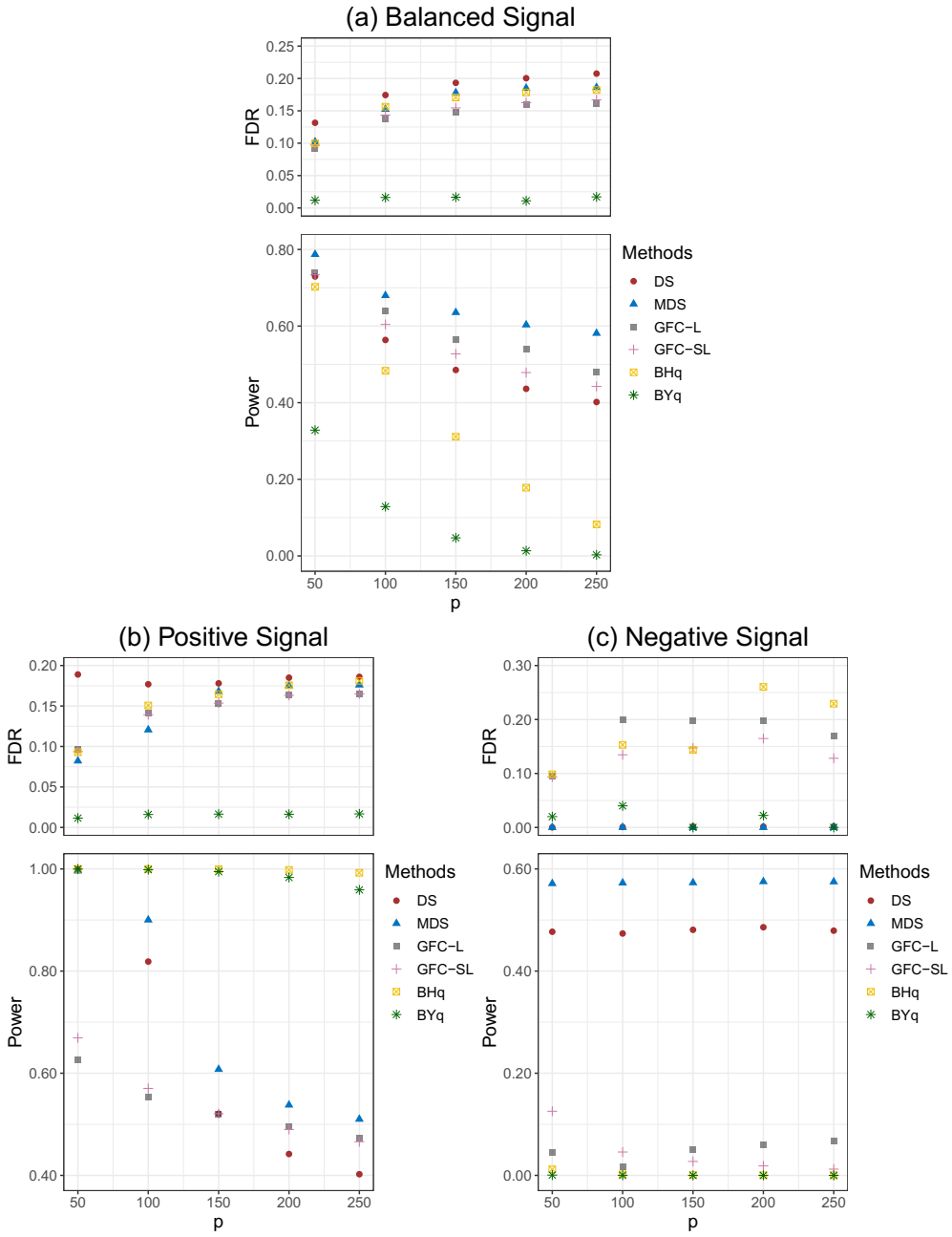


Figure 3.11: Empirical FDRs and powers on the blockwise diagonal graphs with different partial correlation ranges. The six methods are, the single data-splitting method (DS), the multiple data-splitting method (MDS), the GFC methods with Lasso (GFC-L) and scaled Lasso (GFC-SL) proposed in Liu (2013), the Benjamini-Hochberg method (BHq), and the Benjamini-Yekutieli method (BYq). The designated FDR control level is $q = 0.2$ in all settings. The reported results are the empirical means of 50 independent runs.

For the power function and the exponential function, we set the number of signals (sparsity) $s^\star = 30$, and sample the randomly located nonzero β_j from $N(0, 20 \times \sqrt{\log p/n})$. For the sigmoid function, we set the number of signals (sparsity) $s^\star = 50$, and sample the randomly located nonzero β_j from Unif$(0.5, 1)$. For the power function, we assume ϵ follows $N(0, 1)$. For the exponential function and the sigmoid function, we decrease the noise level by assuming ϵ follows $N(0, 0.1^2)$. Throughout we fix the sample size $n = 1000$, and vary $p \in \{500, 1000, 1500, 2000, 3000\}$ for all three cases. The designated FDR control level is fixed to be $q = 0.1$. For the design matrix $\boldsymbol{X}$, in which each row is independently simulated from $N(0, \Sigma)$, we test out two cases: (1) we assume the precision matrix $\Sigma^{-1}_{ij} = \rho^{|i-j|}$; (2) we assume the covariance matrix $\Sigma_{ij} = \rho^{|i-j|}$. ρ is fixed to be 0.5 across all settings.

We compare the proposed data-splitting methods using the influence function or the weight multiplication, and the DeepPINK method proposed in Lu et al. (2018). For DS and MDS, we build a four-layer fully-connected neural network. The input layer is of size p, the output layer is of size 1, and the two hidden layers in the middle are of size $20 \times \log(p)$ and $10 \times \log(p)$, respectively. We choose sigmoid as the activation function between the first three layers, and add a ℓ_1 regularization term of the order $O(\sqrt{\log p/n})$ for the two hidden layers. For DeepPINK, we test out two architectures of the neural network. DeepPINK-I has the same architecture as described in Lu et al. (2018) (see the caption of Figure 1), whereas DeepPINK-II has the same architecture as DS and MDS. We set the batch size to be 128 and set the initializing learning rate to be 0.001. We use the Adam algorithm (Kingma and Ba, 2014) to train the neural network with respect to the mean squared error loss for a total of 500 epochs. For MDS, we randomly split the data 50 times, and aggregate the selection results based on Algorithm 10. For the DeepPINK method, we thank the authors for kindly providing their code through personal communication.

The results for $p = 2000$ are summarized in Figure 3.12, and more detailed results for different p are given in Table C.1 and Table C.2 in the Appendix C. Compared to the DeepPINK method, DS and MDS consistently yield smaller FDR and significantly higher power. For DS and MDS, the

influence function approach and the weight multiplication approach yield similar results. Compared with DS, MDS always has a lower FDR value, but also slightly lower power in cases $f_2(t)$ and $f_3(t)$.

3.5.5 Real data application: HIV drug resistance

We apply the proposed approaches to detect mutations in the Human Immunodeficiency Virus Type 1 (HIV-1) that are associated with drug resistance. The data set, which has also been analyzed in Rhee et al. (2006), Barber and Candès (2015), and Lu et al. (2018), contains resistance measurements of 7 drugs for protease inhibitors (PIs), 6 drugs for nucleoside reverse-transcriptase inhibitors (NRTIs), and 3 drugs for nonnucleoside reverse transcriptase inhibitors (NNRTIs). We focus on the first two classes of inhibitors, PI and NRTI, as in Lu et al. (2018).

The response variable $\boldsymbol{y}$ calibrates the log-fold-increase of lab-tested drug resistance. The design matrix $\boldsymbol{X}$ is binary, in which the jth column indicates the presence or absence of the jth mutation. The task is to select relevant mutations for each inhibitor against different drugs. The data is preprocessed as follows. First, we remove the patients with missing drug resistance information. Second, we only focus on those mutations that appear at least three times across all the patients. The sample size n and the number of mutations p vary from drug to drug, but are all in hundreds with n/p ranging from 1.5 to 4 (see Figures 3.13 and 3.14). We assume an additive linear model between the response variable and the features with no interactions.

Five methods are compared, including DeepPINK with model-X knockoffs (Lu et al., 2018), the fixed-design knockoff filter based on a Gaussian linear model (Barber and Candès, 2015), BHq, DS, and MDS. For DeepPINK, knockoff, and BHq, we report the selection results obtained in Lu et al. (2018). The designated FDR control level is $q = 0.2$ throughout. The results are evaluated based on the selected mutations with existing treatment-selected mutation (TSM) panels (Rhee et al., 2005), as discussed in Barber and Candès (2015).

Numbers of discovered mutations for PI within each drug class, including both true and false pos-

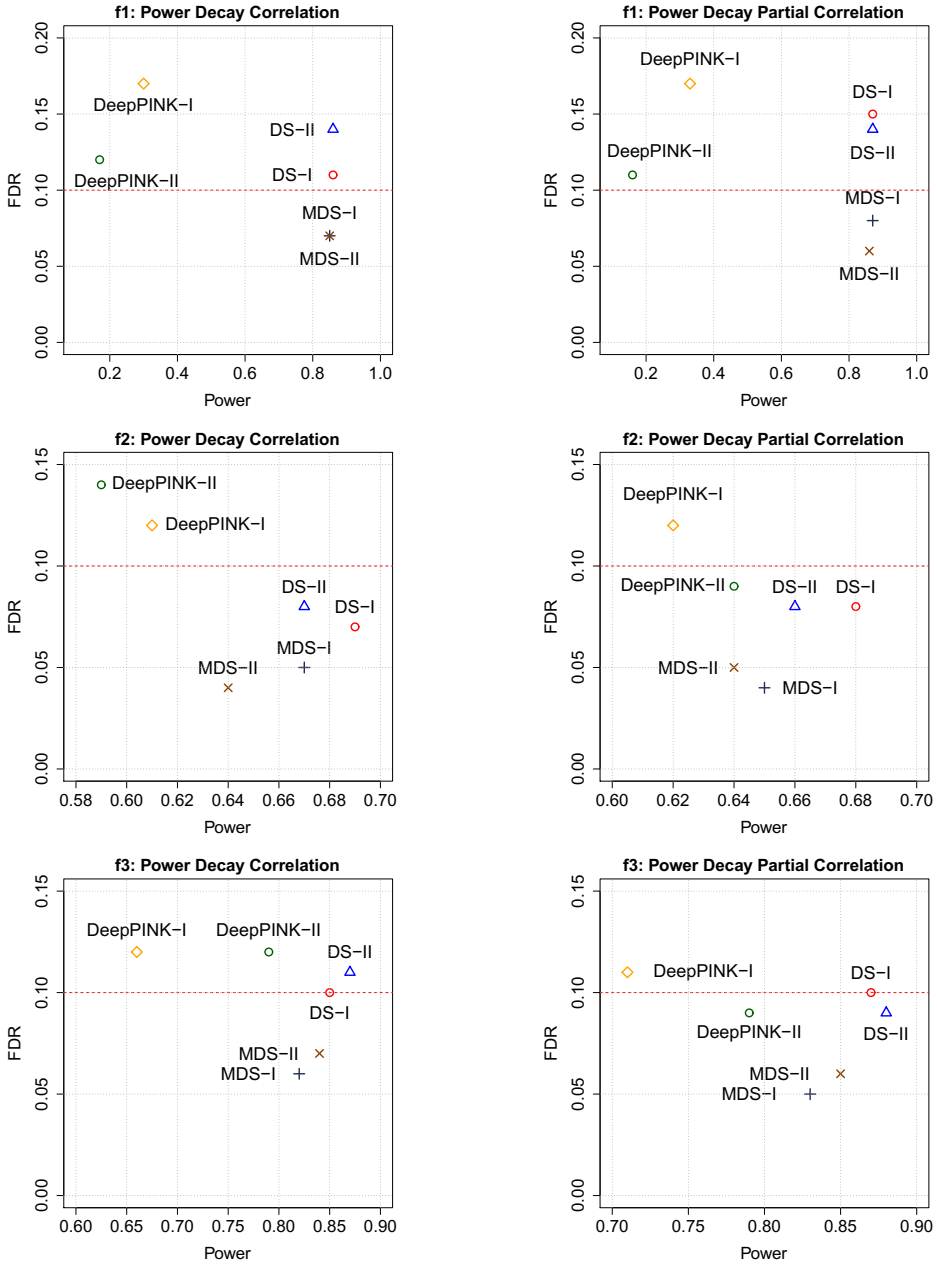


Figure 3.12: Empirical FDRs and powers on the single-index models for $p = 2000$. The design matrix follows multivariate normal distribution with pairwise power decay correlation (the left column) or pairwise power decay partial correlation (the right column). The six methods are, the single data-splitting method using the influence function (DS-I) and its multiple data-splitting version (MDS-I), the single data-splitting method using the weight multiplication (DS-II) and its multiple data-splitting version (MDS-II), and the DeepPINK method with two different network architectures (see the text for details). The designated FDR control level is $q = 0.1$ (the red dashed line) in all settings. The reported results are the empirical means of 20 independent runs.

itives, are summarized in Figure 3.13. We see that MDS performs the best for 4 out of 7 PI drugs, including ATV, LPV, NFV, and SQV. For the remaining drugs, APV, IDV and RTV, MDS is comparable to DeepPINK, and both are superior to the fixed-design knockoff filter and BHq. Similarly, Figure 3.14 shows the numbers of the identified mutations for the NRTI drugs. Among the 6 NRTI drugs, MDS performs the best in 5, including ABC, AZT, D4T, DDI, and X3TC. For TDF, MDS is comparable to DeepPINK, and both are much better than BHq and the fixed-design knockoff filter. In particular, the fixed-design knockoff filter has no power and does not select any mutation for DDI, TDF, and X3TC.

3.6 Concluding Remarks

We have described a general framework for the FDR control in the task of high-dimensional feature selection. The proposed data-splitting approaches (DS and MDS) allow us to asymptotically control FDR in canonical statistical models including the linear model and the Gaussian graphical model. We have also empirically demonstrated its applications to more complex models such as deep neural networks. The multiple data-splitting approach (MDS) proposed here is of particular interest, which helps stabilize the selection result and remedy the potential power loss. Both DS and MDS are conceptually simple and easy to implement based upon existing softwares for high-dimensional feature selection methods.

We conclude by pointing out several directions for future work. First, for the linear model, an interesting extension of the Lasso + OLS procedure is to consider features with a group structure. A natural strategy is to substitute Lasso with group Lasso. However, unlike Lasso, group Lasso can potentially select more than n features (n is the sample size), thus the companion OLS step, which guarantees the symmetric assumption, may not be easily applied. Second, we would like to apply the proposed framework, equipped with the influence function, to convolutional neural networks and re-

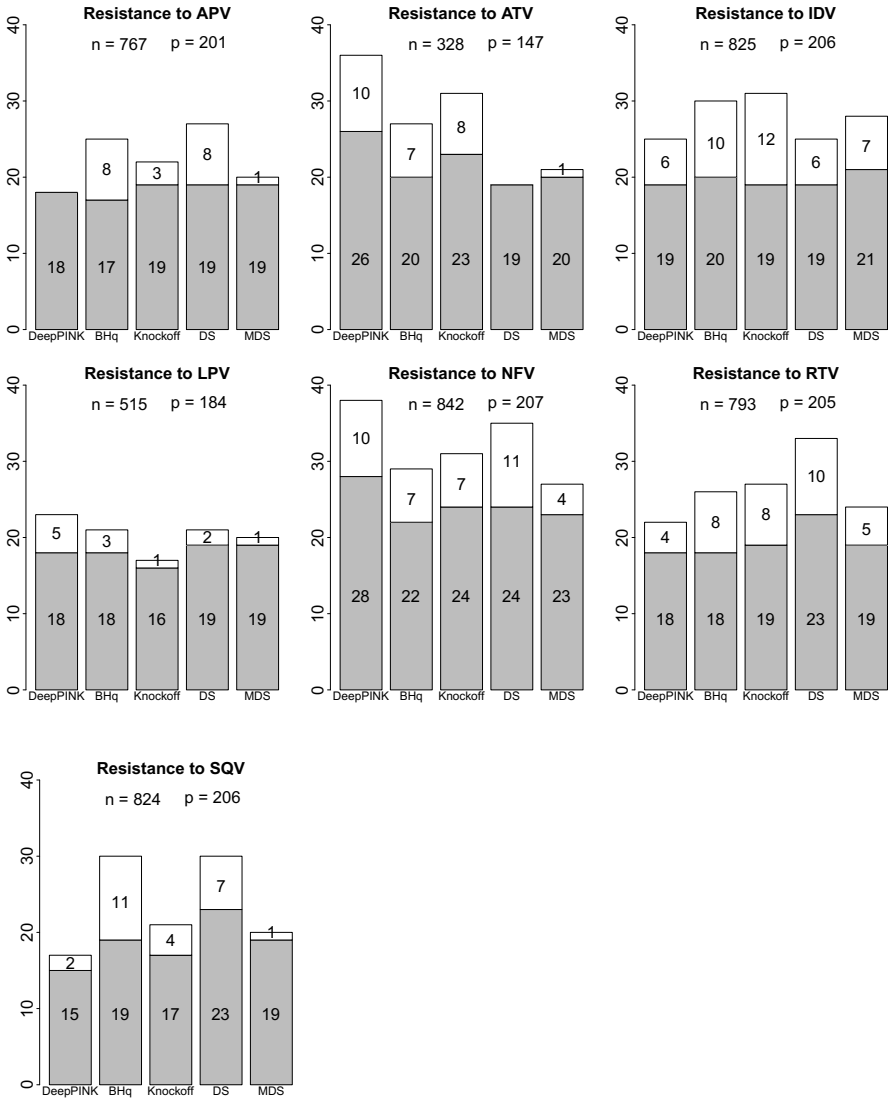


Figure 3.13: Numbers of discovered mutations corresponding to the 7 PI drugs. The grey bar represents the number of true positives, while the white bar represents the number of false positives. The designated FDR control level is $q = 0.2$.

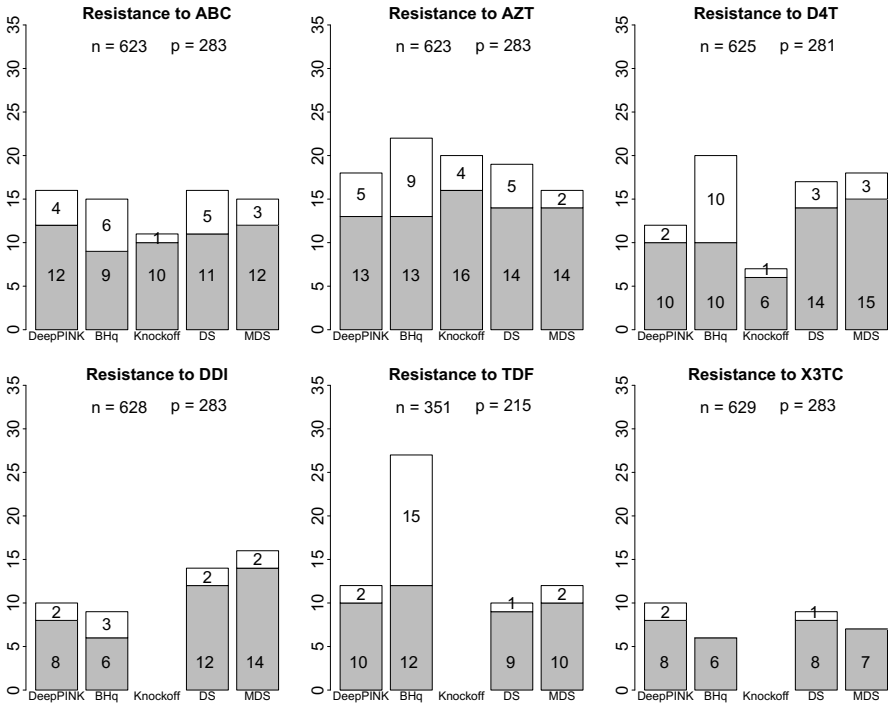


Figure 3.14: Numbers of discovered mutations corresponding to the 6 NRTI drugs. The grey bar represents the number of true positives, while the white bar represents the number of false positives. The designated FDR control level is $q = 0.2$.

current neural networks, in order to handle more complex data such as images and natural languages. Third, we are interested in investigating the multiple testing problem of sparse high-dimensional covariance matrices, which are commonly estimated using some thresholding estimator. Our proposed framework is applicable as long as the estimator to the zero covariance element is symmetric about 0. Last but not the least, extensions of the DS and MDS frameworks to data containing dependent observations or having hierarchical structures can be of immediate interest.

A

Supplemental Materials of Chapter 1

A.1 Proofs

A.1.1 Proof of the almost-sure convergence

The proof of the almost-sure convergence follows from Theorem 2.2 and Theorem 2.3 in Andrieu et al. (2005). Lemma A.1.1 and Lemma A.1.5 in the following verify the necessary conditions.

Lemma A.1.1. We define a Lyapunov function $V(\boldsymbol{u})$ as below,

$$V(\boldsymbol{u}) = \frac{1}{N}\sum_{n=1}^{N}\exp(u_n^\star - u_n) - 1, \tag{A.1}$$

which satisfies the following properties:

1. $V(\boldsymbol{u})$ is non-negative and continuously differentiable on $\Pi = \{\boldsymbol{u} \in \mathbb{R}^N, \sum_{n=1}^N u_n = 0\}$.

2. For any $M > 0$, $\{\boldsymbol{u} : V(\boldsymbol{u}) \leq M\}$ is a compact set.

3. For any $u \in \Pi$, we have $\langle \nabla V(\boldsymbol{u}), \boldsymbol{r}(\boldsymbol{u})\rangle \leq 0$. $\boldsymbol{r}(\boldsymbol{u})$ is the mean-field function defined as below,

$$\boldsymbol{r}(\boldsymbol{u}) = \int_{\mathsf{X}} \boldsymbol{R}(x)\pi_{\boldsymbol{u}}(x)dx = \frac{\exp(\boldsymbol{u}^\star - \boldsymbol{u})}{\sum_{n=1}^N \exp(u_n^\star - u_n)} - \frac{1}{N}, \tag{A.2}$$

in which $R_n(x) = 1(E(x) = E_n) - 1/N$ for $n \in [N]$. In addition,

$$\{\boldsymbol{u} \in \mathbb{R}^N, \langle \nabla V(\boldsymbol{u}), \boldsymbol{r}(\boldsymbol{u})\rangle = 0\} = \{\boldsymbol{u}^\star\}.$$

Proof of Lemma A.1.1:

1. $V(\boldsymbol{u})$ is continuously differentiable on Π. By Jensen's inequality, we have

$$V(\boldsymbol{u}) \geq \exp\left(\frac{1}{N}\sum_{n=1}^{N}(u_n^\star - u_n)\right) - 1 = 0.$$

2. For any $M > 0$ and $V(\boldsymbol{u}) \leq M$, we have

$$u_n \geq u_n^\star - \log((1+M)N) \geq \inf_{k\in[N]} u_k^\star - \log((1+M)N).$$

On the other hand, since $\sum_{n=1}^{N} u_n = 0$, we have

$$u_n \leq (N-1)\log((1+M)N) - (N-1)\inf_{k\in[N]} u_k^\star.$$

Because $V(\boldsymbol{u})$ is a continuous function, the set $\{\boldsymbol{u} : V(\boldsymbol{u}) \leq M\}$ is a compact set.

3. The n-th coordinate of the gradient is $\nabla_n V(\boldsymbol{u}) = -\exp(u_n^\star - u_n)/N$. By Cauchy-Schwartz inequality, we have

$$\langle \nabla V(\boldsymbol{u}), \boldsymbol{r}(\boldsymbol{u})\rangle = -\frac{1}{N}\frac{\sum_{n=1}^{N}\exp\left(2(u_n^\star - u_n)\right)}{\sum_{n=1}^{N}\exp(u_n^\star - u_n)} + \frac{1}{N^2}\sum_{n=1}^{N}\exp(u_n^\star - u_n) \leq 0.$$

The equality is satisfied if and only if $u_n = u_n^\star$ for $n \in [N]$.

Lemma A.1.2. For any $t \geq 0$,

$$||\pi_{t+1} - \pi_t||_{\text{TV}} \leq 2N(N-1)\eta_{t+1}. \tag{A.3}$$

Proof of Lemma A.1.2: using Lemma 4.6 in Fort et al. (2015), we have

$$\begin{aligned}
||\pi_{t+1} - \pi_t||_{\text{TV}} &\leq 2(N-1)\sum_{n=1}^{N}\left|1 - \exp(u_n^{(t)} - u_n^{(t+1)})\right| \\
&= 2(N-1)\sum_{n=1}^{N}|1 - \exp\left(-\eta_{t+1}R_n(x_{t+1})\right)| \\
&= 2(N-1)\left[(N-1)(e^{\frac{1}{N}\eta_{t+1}} - 1) + 1 - e^{-\frac{N-1}{N}\eta_{t+1}}\right] \\
&\leq 2N(N-1)\eta_{t+1}.
\end{aligned}$$

The last line follows from the following elementary inequality: $\forall\, 0 < x \leq 1,\ N \geq 1$,

$$(N-1)e^{\frac{1}{N}x} + 1 - e^{-\frac{N-1}{N}x} - Nx \leq 0.$$

Lemma A.1.3. For any $t \geq 0$,

$$\sup_{x \in \mathsf{X}} ||P_t(x, \cdot) - P_{t+1}(x, \cdot)||_{\text{TV}} \leq 4(e+1)\eta_{t+1}. \tag{A.4}$$

Proof of Lemma A.1.3: using Lemma 4.7 in Fort et al. (2015), we have

$$\begin{aligned}
\sup_{x \in \mathsf{X}} ||P_t(x, \cdot) - P_{t+1}(x, \cdot)||_{\text{TV}} &\leq 4 \sup_{n \in [N]} \left|1 - \exp\left(u_n^{(t+1)} - u_n^{(t)}\right)\right| \\
&+ 4 \sup_{n \in [N]} \left|1 - \exp\left(u_n^{(t)} - u_n^{(t+1)}\right)\right| \\
&\leq 4\left(e^{\frac{N-1}{N}\eta_{t+1}} - 1\right) \vee \left(1 - e^{-\frac{1}{N}\eta_{t+1}}\right) \\
&+ 4\left(1 - e^{-\frac{N-1}{N}\eta_{t+1}}\right) \vee \left(e^{\frac{1}{N}\eta_{t+1}} - 1\right) \\
&\leq 4(e+1)\eta_{t+1}.
\end{aligned}$$

The last line is based on the following elementary inequalities: $\forall\, 0 < x \leq 1,\ N \geq 1$,

$$ex \geq e^{\frac{N-1}{N}x} - 1 \geq 1 - e^{-\frac{1}{N}x}, \quad \max\left\{1 - e^{-\frac{N-1}{N}x}, e^{\frac{1}{N}x} - 1\right\} \leq x.$$

Lemma A.1.4. Under Assumption 1, there exists a function $\widehat{\boldsymbol{R}}_{\boldsymbol{u}}(x)$ solving the Poisson equation

$$\widehat{\boldsymbol{R}}_{\boldsymbol{u}}(x) - P_{\boldsymbol{u}}\widehat{\boldsymbol{R}}_{\boldsymbol{u}}(x) = \boldsymbol{R}(x) - \pi_{\boldsymbol{u}}(\boldsymbol{R}(x)). \tag{A.5}$$

In particular,

$$\sup_{\boldsymbol{u} \in \Pi} \sup_{x \in \mathsf{X}} \left|\left|\widehat{\boldsymbol{R}}_{\boldsymbol{u}}(x)\right|\right|_{\infty} < +\infty. \tag{A.6}$$

Besides, there exists a constant $C > 0$, such that for any $\boldsymbol{u}, \boldsymbol{u}' \in \Pi$, we have

$$\begin{aligned}\sup_{x\in\mathsf{X}}\left|P_{\boldsymbol{u}}\widehat{\boldsymbol{R}}_{\boldsymbol{u}}(x) - P_{\boldsymbol{u}'}\widehat{\boldsymbol{R}}_{\boldsymbol{u}'}(x)\right| \leq C\sup_{x\in\mathsf{X}}||P_{\boldsymbol{u}}(x,\cdot) - P_{\boldsymbol{u}'}(x,\cdot)||_{\mathrm{TV}} \\ + C\sup_{x\in\mathsf{X}}||\pi_{\boldsymbol{u}}(x,\cdot) - \pi_{\boldsymbol{u}'}(x,\cdot)||_{\mathrm{TV}}.\end{aligned} \tag{A.7}$$

Proof of Lemma A.1.4: $\widehat{\boldsymbol{R}}_{\boldsymbol{u}}(x)$ is specified as follows:

$$\widehat{\boldsymbol{R}}_{\boldsymbol{u}}(x) = \sum_{k\geq 0}\left[P_{\boldsymbol{u}}^k\boldsymbol{R}_{\boldsymbol{u}}(x) - \pi_{\boldsymbol{u}}(\boldsymbol{R}(x))\right].$$

Since $\sup_n \sup_x |R_n(x)| \leq 1$, by Assumption 1, we have

$$\begin{aligned}\sup_{\boldsymbol{u}\in\Pi}\sup_{x\in\mathsf{X}}\left|\left|\widehat{\boldsymbol{R}}_{\boldsymbol{u}}(x)\right|\right|_\infty &\leq \sum_{k\geq 0}\sup_{\boldsymbol{u}\in\Pi}\sup_{x\in\mathsf{X}}\left|\left|P_{\boldsymbol{u}}^k\boldsymbol{R}_{\boldsymbol{u}}(x) - \pi_{\boldsymbol{u}}(\boldsymbol{R}(x))\right|\right|_\infty \\ &\leq \sum_{k\geq 0}||P_{\boldsymbol{u}}^k(x,\cdot) - \pi_{\boldsymbol{u}}||_{\mathrm{TV}} \\ &\leq \sum_{k\geq 0}2(1-\rho)^k \leq +\infty.\end{aligned}$$

The proof of the second part of Lemma A.1.4 follows from Lemma 4.2 in Fort et al. (2011).

Lemma A.1.5. Under Assumption 1, if we scale down η_t in the order of $O(1/t)$, almost surely, we have

$$\limsup_{t\to\infty}\sup_{\ell\geq k}\left|\sum_{t=k}^{\ell}\eta_{t+1}\left(\boldsymbol{R}(x_{t+1}) - \boldsymbol{r}(\boldsymbol{u}^{(t)})\right)\right| = 0. \tag{A.8}$$

The proof of Lemma A.1.5 follows from the proof of Proposition 4.10 in Fort et al. (2015).

A.1.2 Proof of the convergence rate

Lemma A.1.6. Under Assumption 1, there exists a constant $\ell > 0$ such that for any $t \geq 0$, almost surely, it holds

$$\langle \nabla h(\boldsymbol{u}^{(t)}), \boldsymbol{u}^{(t)} - \boldsymbol{u}^\star \rangle \geq \ell ||\boldsymbol{u}^{(t)} - \boldsymbol{u}^\star||^2, \tag{A.9}$$

where $h(\boldsymbol{u}) = \log\left(\sum_{n=1}^N \exp\left(u_n^\star - u_n\right)\right)$.

Proof of Lemma A.1.6: since $\boldsymbol{u}^{(t)}$ almost surely stays in a compact set, there exist constants $C_1 > 0$ and $C_2 < 0$, such that for any $t \geq 0, n \in [N]$, we have $C_2 \leq u_n^{(t)} \leq C_1$. We proceed to show that the ℓ specified below satisfies the condition,

$$\ell = \left[N(\max_{n\in[N]} u_n^\star - C_2)^3 \exp(-(\min_{u\in[N]} u_n^\star - C_1)) \sum_{n=1}^N \exp(u_n^\star - C_2)\right]^{-1}.$$

Indeed,

$$\begin{aligned}
\langle \nabla h(\boldsymbol{u}^{(t)}), \boldsymbol{u}^{(t)} - \boldsymbol{u}^\star \rangle &= \frac{\sum_{n=1}^N (u_n^\star - u_n^{(t)}) \exp(u_n^\star - u_n^{(t)})}{\sum_{n=1}^N \exp(u_n^\star - u_n^{(t)})} \\
&\geq \ell N(\max_{n\in[N]} u_n^\star - C_2)^3 \exp(-(\min_{u\in[N]} u_n^\star - C_1)) \sum_{n=1}^N (u_n^\star - u_n^{(t)}) \exp(u_n^\star - u_n^{(t)}) \\
&\geq \ell \exp(-(\min_{u\in[N]} u_n^\star - C_1)) \sum_{n=1}^N (u_n^\star - u_n^{(t)})^3 \sum_{n=1}^N (u_n^\star - u_n^{(t)}) \exp(u_n^\star - u_n^{(t)}) \\
&\geq \ell \exp(-(\min_{u\in[N]} u_n^\star - C_1)) \left[\sum_{n=1}^N (u_n^\star - u_n^{(t)})^2 \exp\left(\frac{u_n^\star - u_n^{(t)}}{2}\right)\right]^2 \\
&\geq \ell ||\boldsymbol{u}^{(t)} - \boldsymbol{u}^\star||^2.
\end{aligned}$$

In the first and the second inequality, we use an elementary inequality

$$\sum_{n=1}^N (u_n^\star - u_n^{(t)}) \exp(u_n^\star - u_n^{(t)}) \geq \sum_{n=1}^N (u_n^\star - u_n^{(t)}) = 0.$$

Lemma A.1.7. Under Assumption 1, there exists a constant $C > 0$ such that

$$\mathbb{E}||\pi_t - P_t(x_t, \cdot)||_{\text{TV}} \leq C\eta_{t+1}. \tag{A.10}$$

Proof of Lemma C.1.5: we prove by induction. Suppose the statement is true for $t-1$, we consider the case for t.

$$\begin{aligned}
\mathbb{E}||\pi_t - P_t(x_t, \cdot)||_{\text{TV}} &= \mathbb{E} \sup_{|f|\leq 1} |\pi_t(f) - P_t f(x_t)| \\
&\leq \mathbb{E}||\pi_t - \pi_{t-1}||_{\text{TV}} + \mathbb{E} \sup_{x\in\mathsf{X}} ||P_t(x, \cdot) - P_{t-1}(x, \cdot)||_{\text{TV}} \\
&+ \mathbb{E} \sup_{|f|\leq 1} |\pi_{t-1}(f) - P_{t-1} f(x_t)|.
\end{aligned} \tag{A.11}$$

The first two terms can be upper bounded using Lemma A.1.2 and Lemma A.1.3. For the third term, since x_t is a sample from $P_{t-1}(x_{t-1}, \cdot)$, we have

$$\begin{aligned}
\mathbb{E} \sup_{|f|\leq 1} |\pi_{t-1}(f) - P_{t-1} f(x_t)| &= \mathbb{E} \sup_{|f|\leq 1} |\pi_{t-1}(f) - P_{t-1}^2 f(x_{t-1})| \\
&\leq \mathbb{E}\big[||\pi_{t-1} - P_{t-1}(x_{t-1}, \cdot)||_{\text{TV}} \sup_{x\in\mathsf{X}} \sup_{|f|\leq 1} |\pi_{t-1}(f) - P_{t-1} f(x)|\big] \\
&= \mathbb{E}\big[||\pi_{t-1} - P_{t-1}(x_{t-1}, \cdot)||_{\text{TV}} \sup_{x\in\mathsf{X}} ||\pi_{t-1} - P_{t-1}(x, \cdot)||_{\text{TV}}\big] \\
&\leq 2(1-\rho)\mathbb{E}||\pi_{t-1} - P_{t-1}(x_{t-1}, \cdot)||_{\text{TV}} \\
&\leq 2(1-\rho)C\eta_t.
\end{aligned} \tag{A.12}$$

The last to second line uses Assumption 1, and the last line uses the induction. Without loss of generality, we assume $2(1-\rho) < 1/2$, otherwise we can keep decreasing the iteration index until

$2(1-\rho)^k < 1/2$ for some $k \in \mathbb{N}_+$. Applying the above upper bound to Equation (A.11), we have

$$\mathbb{E}||\pi_t - P_t(x_t, \cdot)||_{\mathrm{TV}} \leq 2N(N-1)\eta_t + 4(e+1)\eta_t + 2(1-\rho)C\eta_t \leq C\eta_{t+1}$$

as long as

$$C \geq \frac{2N(N-1) + 4(e+1)}{1/2 - 2(1-\rho)}.$$

Proof of the convergence rate: denote $G_n(x) = R_n(x) + 1/N$ for $n \in [N]$. Conditioning on $\mathcal{F}_t$, we first notice that

$$\mathbb{E}\left[||\boldsymbol{R}(x_{t+1})||^2 \middle| \mathcal{F}_t\right] = \left(-\frac{1}{N}\right)^2 (N-1) + \left(\frac{N-1}{N}\right)^2 \leq 1.$$

It follows that

$$\begin{aligned}
\mathbb{E}\left[||\boldsymbol{u}^{(t+1)} - \boldsymbol{u}^\star||^2 \middle| \mathcal{F}_t\right] &= \mathbb{E}\left[||\boldsymbol{u}^{(t)} + \boldsymbol{R}(x_{t+1})\eta_{t+1} - \boldsymbol{u}^\star||^2 \middle| \mathcal{F}_t\right] \\
&= ||\boldsymbol{u}^{(t)} - \boldsymbol{u}^\star||^2 + \eta_{t+1}^2 \mathbb{E}\left[||\boldsymbol{R}(x_{t+1})||^2 \middle| \mathcal{F}_t\right] \\
&\quad + 2\eta_{t+1}\langle \mathbb{E}\left[\boldsymbol{R}(x_{t+1}) \middle| \mathcal{F}_t\right], \boldsymbol{u}^{(t)} - \boldsymbol{u}^\star \rangle \\
&\leq ||\boldsymbol{u}^{(t)} - \boldsymbol{u}^\star||^2 + \eta_{t+1}^2 + 2\eta_{t+1}\langle \pi_t(\boldsymbol{G}), \boldsymbol{u}^{(t)} - \boldsymbol{u}^\star \rangle \\
&\quad + 2\eta_{t+1}\langle P_t\boldsymbol{G}(x_t) - \pi_t(\boldsymbol{G}), \boldsymbol{u}^{(t)} - \boldsymbol{u}^\star \rangle \\
&\leq (1 - 2\eta_{t+1}\ell)||\boldsymbol{u}^{(t)} - \boldsymbol{u}^\star||^2 + \eta_{t+1}^2 \\
&\quad + 2\eta_{t+1}N||P_t(x_t, \cdot) - \pi_t||_{\mathrm{TV}} \sup_{n \in [N]} \left|u_n^{(t)} - u_n^\star\right|,
\end{aligned}$$

which follows from Lemma A.1.6 in which ℓ is defined. Since $\boldsymbol{u}^{(t)}$ almost surely stays in a compact set, by Lemma C.1.5, there exists a constant $C_1 > 0$ such that

$$\mathbb{E}\left[||\boldsymbol{u}^{(t+1)} - \boldsymbol{u}^\star||^2\right] \leq (1 - 2\eta_{t+1}\ell)\mathbb{E}\left[||\boldsymbol{u}^{(t)} - \boldsymbol{u}^\star||^2\right] + C_1\eta_{t+1}^2.$$

Let $\eta_{t+1} = 1/(\ell t)$ and $C = \max\{||\boldsymbol{u}^{(1)} - \boldsymbol{u}^\star||^2, C_1/\ell^2\}$. We prove the statement by induction. The case $t = 1$ trivially holds. Suppose the convergence rate holds for t. For the case $t + 1$, we have

$$\mathbb{E}\left[||\boldsymbol{u}^{(t+1)} - \boldsymbol{u}^\star||^2\right] \leq \left(1 - \frac{2}{t}\right)\frac{C}{t} + \frac{C_1}{\ell^2 t^2} \leq \left(1 - \frac{2}{t}\right)\frac{C}{t} + \frac{C}{t^2} = \left(\frac{1}{t} - \frac{1}{t^2}\right)C \leq \frac{C}{t+1}.$$

A.2 More Simulation Details

A.2.1 Equilibration time

In this section, we report the total number of equilibrations, as well as the corresponding equilibration time, that the AWL algorithm and the WL algorithm have reached on the Ising model and the Potts model. Three initializations of the learning rate $\eta_0 = 0.05$, $\eta_0 = 0.10$, $\eta_0 = 1.00$, and six lattice sizes $L = 50, 60, 70, 80, 90, 100$ are tested out. We run in total 2×10^5 and 2×10^6 MC sweeps for the Ising model and the Potts model, respectively. The results are summarized in Table A.1 and A.2.

A.2.2 Estimating Specific Heat

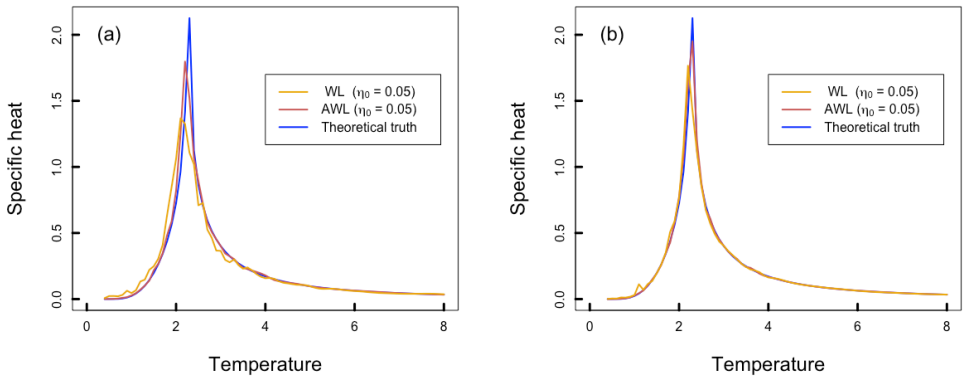


Figure A.1: Comparison of the estimated specific heat (Ising model with $L = 80$) obtained by the AWL algorithm and the WL algorithm, over the temperature region $T \in [0.4, 8]$. At each temperature ranging from 0.4 to 8 incremented by 0.1, the reported results are the mean of 50 independent estimates of the specific heat produced by each algorithm. Panel (a) and (b) show the results of running each algorithm for 150×10^3 and 200×10^3 MC sweeps, respectively. η_0 denotes the initialization of the learning rate.

Initialization of the learning rate $\eta_0 = 0.05$										
Method	L	# equilibration	First eight equilibration time ($\times 10^3$ MC sweeps)							
AWL	50	12.1 ± 0.9	23.4	27.7	32.1	36.6	42.2	49.8	61.4	76.1
WL	50	10.8 ± 0.9	43.4	47.9	52.4	58.1	65.1	72.5	84.8	101.3
AWL	60	10.9 ± 0.6	33.5	38.4	44.0	49.1	56.5	66.0	79.8	94.1
WL	60	9.2 ± 0.7	65.3	71.6	77.9	84.6	93.7	105.1	118.3	145.1
AWL	70	9.5 ± 0.9	49.2	56.7	63.2	68.7	78.4	90.7	105.6	127.6
WL	70	8.0 ± 0.9	92.7	101.9	110.4	119.3	131.3	145.0	166.2	193.7
AWL	80	8.7 ± 0.6	65.6	73.4	82.6	90.7	103.6	114.8	132.1	155.7
WL	80	6.1 ± 0.9	119.5	133.2	143.8	156.9	170.0	187.2	—	—
AWL	90	7.7 ± 0.8	79.9	92.1	102.0	113.9	127.1	144.6	165.8	195.3
WL	90	2.8 ± 1.1	165.9	180.8	196.2	—	—	—	—	—
AWL	100	6.4 ± 1.0	100.8	114.9	128.2	142.0	157.8	174.6	—	—
WL	100	0.7 ± 0.9	—	—	—	—	—	—	—	—
Initialization of the learning rate $\eta_0 = 0.10$										
Method	L	# equilibration	First eight equilibration time ($\times 10^3$ MC sweeps)							
AWL	50	13.0 ± 0.9	17.8	21.4	24.9	28.4	33.4	37.8	44.9	53.0
WL	50	12.1 ± 0.9	33.7	37.9	42.2	46.4	52.2	58.6	67.5	79.5
AWL	60	11.9 ± 0.5	28.9	35.6	41.8	47.6	53.2	58.4	68.7	81.8
WL	60	10.5 ± 0.9	54.0	60.8	67.3	72.6	78.9	88.6	100.5	115.5
AWL	70	10.5 ± 0.8	40.4	46.5	53.5	62.1	71.7	81.0	93.1	107.5
WL	70	9.1 ± 0.8	74.5	81.4	89.3	96.9	105.0	116.0	128.9	149.1
AWL	80	9.5 ± 0.7	57.4	64.6	74.7	84.1	91.6	101.6	116.7	139.5
WL	80	7.5 ± 0.7	102.8	114.2	123.9	134.9	145.5	158.3	174.6	196.3
AWL	90	8.4 ± 0.9	73.9	85.5	100.5	112.7	124.9	139.0	156.2	176.5
WL	90	5.3 ± 0.9	134.4	148.0	165.0	176.9	188.2	—	—	—
AWL	100	6.7 ± 0.9	95.9	110.8	124.6	143.8	157.4	174.1	189.8	—
WL	100	2.5 ± 1.1	165.6	181.9	198.0	—	—	—	—	—
Initialization of the learning rate $\eta_0 = 1.00$										
Method	L	# equilibration	First eight equilibration time ($\times 10^3$ MC sweeps)							
AWL	50	15.9 ± 0.8	16.8	20.8	25.1	28.8	32.4	36.2	39.3	43.9
WL	50	14.6 ± 0.6	27.7	32.4	37.7	43.0	47.4	51.9	55.7	61.7
AWL	60	14.8 ± 0.9	25.6	31.7	37.4	41.9	47.7	52.7	57.5	63.9
WL	60	13.5 ± 0.8	44.1	52.0	59.1	66.4	72.6	78.5	84.9	91.1
AWL	70	13.7 ± 0.9	36.3	43.4	51.1	59.7	69.0	76.8	83.9	91.2
WL	70	12.2 ± 1.0	65.7	75.3	83.4	94.0	101.8	109.8	117.4	125.5
AWL	80	11.9 ± 0.8	48.7	60.9	73.9	85.0	93.5	103.0	112.3	120.5
WL	80	9.9 ± 0.9	86.2	97.9	110.0	123.3	134.7	143.8	153.7	162.3
AWL	90	10.5 ± 0.7	66.1	78.9	89.6	101.3	113.9	124.1	135.4	145.4
WL	90	6.8 ± 1.0	107.3	119.2	140.3	156.5	169.0	185.9	197.2	—
AWL	100	7.4 ± 1.0	90.7	106.2	122.7	133.1	149.5	162.3	174.5	185.8
WL	100	2.7 ± 1.1	156.5	180.0	197.4	—	—	—	—	—

Table A.1: The total number of equilibrations, as well as the corresponding first eight equlibration time, that the AWL algorithm and the WL algorithm have reached within 2×10^5 MC sweeps on the Ising model. The reported results are the empirical means over 50 independent runs.

Initialization of the learning rate $\eta_0 = 0.05$										
Method	L	# equilibration	First eight equilibration time ($\times 10^4$ MC sweeps)							
AWL	50	14.6 ± 1.1	26.6	29.3	31.8	34.2	36.6	40.0	44.2	50.2
WL	50	13.0 ± 0.9	43.6	46.6	49.8	53.0	56.4	60.7	66.3	74.0
AWL	60	12.8 ± 0.7	41.9	45.6	49.3	53.0	56.4	60.4	66.4	73.9
WL	60	11.8 ± 0.6	63.7	68.0	72.4	76.8	80.8	86.5	92.3	102.6
AWL	70	11.7 ± 0.7	56.5	61.5	65.9	70.8	81.7	89.5	98.9	112.9
WL	70	9.9 ± 0.9	99.9	106.0	112.4	118.9	125.0	131.9	143.0	151.4
AWL	80	10.5 ± 0.9	79.6	85.9	91.7	98.3	104.5	112.8	122.3	134.3
WL	80	7.8 ± 0.8	127.8	135.2	143.6	151.9	158.8	170.8	181.3	195.6
AWL	90	9.2 ± 0.9	100.8	108.8	117.6	124.8	131.7	140.5	153.0	164.4
WL	90	3.2 ± 0.9	176.7	186.1	196.6	—	—	—	—	—
AWL	100	6.6 ± 1.1	133.2	143.0	152.1	163.7	174.3	186.3	197.3	—
WL	100	0.3 ± 0.7	—	—	—	—	—	—	—	—
Initialization of the learning rate $\eta_0 = 0.10$										
Method	L	# equilibration	First eight equilibration time ($\times 10^4$ MC sweeps)							
AWL	50	15.2 ± 0.8	23.8	26.7	29.2	31.7	34.4	37.4	41.5	46.4
WL	50	14.0 ± 0.9	42.5	45.9	49.3	52.8	56.5	60.0	64.3	70.8
AWL	60	14.4 ± 0.9	34.0	37.7	41.7	45.2	48.2	52.9	57.9	63.7
WL	60	12.9 ± 0.7	58.6	63.5	68.2	72.6	77.2	82.8	88.5	95.0
AWL	70	12.7 ± 0.7	48.6	54.0	59.4	64.0	69.4	75.5	81.5	88.5
WL	70	10.9 ± 0.8	93.8	100.5	107.1	113.2	120.0	126.1	132.4	141.2
AWL	80	11.4 ± 0.7	71.5	78.5	85.3	91.4	99.1	105.7	112.4	121.9
WL	80	9.1 ± 0.6	117.5	126.2	135.4	143.6	151.4	159.2	166.0	179.2
AWL	90	10.4 ± 0.9	92.2	100.8	109.4	118.0	125.8	132.6	141.3	152.2
WL	90	4.9 ± 0.9	153.4	164.5	176.6	186.6	196.5	—	—	—
AWL	100	6.5 ± 0.7	131.1	142.2	151.5	160.8	169.2	180.7	191.5	—
WL	100	1.0 ± 0.9	191.4	—	—	—	—	—	—	—
Initialization of the learning rate $\eta_0 = 1.00$										
Method	L	# equilibration	First 8 equilibration time ($\times 10^4$ MC sweeps)							
AWL	50	18.9 ± 0.9	25.0	28.8	32.4	35.9	39.0	41.6	44.4	47.1
WL	50	17.9 ± 0.9	33.8	38.0	42.1	46.1	49.8	53.2	57.0	59.8
AWL	60	17.1 ± 0.8	37.7	42.9	48.9	53.7	57.8	62.5	66.0	70.0
WL	60	15.9 ± 0.6	54.3	60.3	66.2	71.8	77.0	82.0	86.7	91.3
AWL	70	15.1 ± 0.8	57.6	66.8	74.5	83.2	89.3	94.4	99.5	104.2
WL	70	13.9 ± 1.0	78.4	86.4	94.5	102.3	109.4	115.8	121.8	127.8
AWL	80	13.5 ± 0.7	72.8	81.7	94.3	102.8	110.8	117.6	124.1	130.0
WL	80	9.8 ± 0.9	115.5	126.8	137.3	148.0	157.1	165.3	173.1	181.0
AWL	90	10.4 ± 1.0	100.9	112.4	126.1	137.8	148.1	156.2	161.8	167.5
WL	90	5.0 ± 0.7	141.2	150.9	163.1	174.1	186.9	—	—	—
AWL	100	6.1 ± 0.9	126.4	140.8	152.1	164.8	176.9	187.1	—	—
WL	100	0.9 ± 0.9	194.2	—	—	—	—	—	—	—

Table A.2: The total number of equilibrations, as well as the corresponding first eight equlibration time, that the AWL algorithm and the WL algorithm have reached within 2×10^6 MC sweeps on the Potts model. The reported results are the empirical means over 50 independent runs.

B

Supplemental Materials of Chapter 2

B.1 Proofs

Proof of Proposition 1: We first consider the fixed-directional jump. In the following, we calculate the transition probability $\mathbb{P}\left((\boldsymbol{\theta}_j, \mathcal{M}_j) \mid (\boldsymbol{\theta}_i, \mathcal{M}_i)\right)$. Suppose the transition from $(\boldsymbol{\theta}_i, \mathcal{M}_i)$ to $(\boldsymbol{\theta}_j, \mathcal{M}_j)$

is achieved by some jumping distance r , then we have:

$$\boldsymbol{\theta}_j = \boldsymbol{u} + r(\widehat{\boldsymbol{\theta}}_j - \widehat{\boldsymbol{u}}) \text{ and } \boldsymbol{v} = \boldsymbol{\theta}_i + r(\widehat{\boldsymbol{v}} - \widehat{\boldsymbol{\theta}}_i).$$

Let

$$S_j = \sum_{k=1}^{m} p_j(\boldsymbol{v}^{(k)}, \boldsymbol{\theta}_j^{(k)}, \mathcal{M}_j \mid \boldsymbol{y}) \quad \text{and} \quad S_i = \sum_{k=1}^{m} p_i\left(\boldsymbol{\theta}_i^{(k)}, \boldsymbol{u}^{(k)}, \mathcal{M}_i \mid \boldsymbol{y}\right). \tag{B.1}$$

It follows that

$$\begin{aligned}
\mathbb{P}\left((\boldsymbol{\theta}_j, \mathcal{M}_j) \mid (\boldsymbol{\theta}_i, \mathcal{M}_i)\right) &= \int q_i(\boldsymbol{u}) \times p_j\left(\boldsymbol{v}, \boldsymbol{\theta}_j, \mathcal{M}_j \mid \boldsymbol{y}\right)/S_j \times \min\left\{1, \quad S_j/S_i\right\} \\
&\quad \times \prod_{k=1}^{m-1} p(r^{(k)}) p(r) dr^{(1)} \cdots dr^{(m-1)} \\
&= \int p\left(\boldsymbol{\theta}_j, \mathcal{M}_j \mid \boldsymbol{y}\right) q_i(\boldsymbol{u}) q_j(\boldsymbol{v}) p(r) \min\left\{1/S_j, 1/S_i\right\} \\
&\quad \times \prod_{k=1}^{m-1} p(r^{(k)}) dr^{(1)} \cdots dr^{(m-1)}
\end{aligned}$$

where

$$\begin{aligned}
\boldsymbol{\theta}_j^{(k)} &= \boldsymbol{u} + r^{(k)}(\widehat{\boldsymbol{\theta}}_j - \widehat{\boldsymbol{u}}), \quad \boldsymbol{v}^{(k)} = \boldsymbol{\theta}_i + r^{(k)}(\widehat{\boldsymbol{v}} - \widehat{\boldsymbol{\theta}}_i), \\
\boldsymbol{\theta}_i^{(k)} &= \boldsymbol{v} - r^{(k)}(\widehat{\boldsymbol{v}} - \widehat{\boldsymbol{\theta}}_i), \quad \boldsymbol{u}^{(k)} = \boldsymbol{\theta}_j - r^{(k)}(\widehat{\boldsymbol{\theta}}_j - \widehat{\boldsymbol{u}}).
\end{aligned}$$

We note that when we jump back from $(\boldsymbol{v}, \boldsymbol{\theta}_j)$ to $(\boldsymbol{\theta}_i, \boldsymbol{u})$, we flip the sign of the jumping direction, that is, changing the jumping direction to $(\widehat{\boldsymbol{u}} - \widehat{\boldsymbol{\theta}}_j, \widehat{\boldsymbol{\theta}}_i - \widehat{\boldsymbol{v}})$, and keep the same jumping distance r. This is the reason why we do not require the sampling distribution of the jumping distance $p(r)$ to be symmetric and centered at 0. Since the Jacobian between $(\boldsymbol{v}, \boldsymbol{\theta}_j)$ and $(\boldsymbol{\theta}_i, \boldsymbol{u})$ is simply 1, and the multiple integral is symmetric in the index i and j, the transition kernel satisfies the detailed balance condition, thus leaves $p\left(\boldsymbol{\theta}_k, \mathcal{M}_k \mid \boldsymbol{y}\right)$ invariant.

For the adaptive-directional jump, we first standardize the jumping direction, that is, setting $\boldsymbol{e} =$

$(\widehat{\boldsymbol{v}} - \boldsymbol{\theta}_i, \widehat{\boldsymbol{\theta}}_j - \boldsymbol{u})/||(\widehat{\boldsymbol{v}} - \boldsymbol{\theta}_i, \widehat{\boldsymbol{\theta}}_j - \boldsymbol{u})||$, so that the jumping distance is independent to the current state of the chain $(\boldsymbol{\theta}_i, \boldsymbol{u})$. Besides, because the jumping direction always points to the modes of the augmented posterior distributions defined in (2.20), we should flip the sign of the jumping distance when we jump back from $(\boldsymbol{v}, \boldsymbol{\theta}_j)$ to $(\boldsymbol{\theta}_i, \boldsymbol{u})$ as we won't flip the sign of the jumping direction. Consequently, the sampling distribution of the jumping distance $p(r)$ is required to be symmetric and centered at 0.

The proof of the reversibility of the transition kernel equipped with the adaptive-directional jump follows similarly (thus is omitted) as the case of the fixed-directional jump, but requires additional calculations of the Jacobian, which are detailed as below. Suppose the transition from $(\boldsymbol{\theta}_i, \mathcal{M}_i)$ to $(\boldsymbol{\theta}_j, \mathcal{M}_j)$ is achieved by some jumping distance r, then we have:

$$\boldsymbol{\theta}_j = \boldsymbol{u} + r\frac{\widehat{\boldsymbol{\theta}}_j - \boldsymbol{u}}{||(\widehat{\boldsymbol{v}} - \boldsymbol{\theta}_i, \widehat{\boldsymbol{\theta}}_j - \boldsymbol{u})||} \quad \text{and} \quad \boldsymbol{v} = \boldsymbol{\theta}_i + r\frac{\widehat{\boldsymbol{v}} - \boldsymbol{\theta}_i}{||(\widehat{\boldsymbol{v}} - \boldsymbol{\theta}_i, \widehat{\boldsymbol{\theta}}_j - \boldsymbol{u})||}.$$

For notational convenience, we define $\boldsymbol{x} \in \mathbb{R}^{d_i+d_j}$ as follows. For $1 \leq k \leq d_j$, let $x_k = (\widehat{\theta}_{jk} - u_k)/||(\widehat{\boldsymbol{v}} - \boldsymbol{\theta}_i, \widehat{\boldsymbol{\theta}}_j - \boldsymbol{u})||^{3/2}$. For $1 \leq l \leq d_i$, let $x_{d_j+l} = (\widehat{v}_l - \theta_{il})/||(\widehat{\boldsymbol{v}} - \boldsymbol{\theta}_i, \widehat{\boldsymbol{\theta}}_j - \boldsymbol{u})||^{3/2}$. Then we have:

$$\left|\frac{\partial(\boldsymbol{\theta}_j, \boldsymbol{v})}{\partial(\boldsymbol{u}, \boldsymbol{\theta}_i)}\right| = \det\left((1 - r||\boldsymbol{x}||^2)I_{d_j} + r\boldsymbol{x}\boldsymbol{x}^\intercal\right) = \left[1 - \frac{r}{||(\widehat{\boldsymbol{v}} - \boldsymbol{\theta}_i, \widehat{\boldsymbol{\theta}}_j - \boldsymbol{u})||}\right]^{d_i+d_j-1}.$$

B.2 More Simulation Details

B.2.1 The Log-Gaussian Cox Process

We provide more details of the Log-Gaussian Cox process discussed in Section 2.6.1. We first detail the adaptive SMC algorithm in Algorithm 14. We then examine the effect of the threshold c used in the flat histogram criterion (see Algorithm 4, step 3(e)). Figure B.1 shows that the WL mixture

Algorithm 14: An adaptive sequential Monte Carlo (SMC) sampler (Zhou et al., 2016).

Input: proposal distribution $q(\boldsymbol{\theta})$, Markov kernels $\{K_t\}$.

1. Initialization.
   (a) Sample $\boldsymbol{\theta}_0^{(i)}$ from $q(\boldsymbol{\theta})$ for $i \in [n]$ independently.
   (b) Set $w_0^{(i)} = 1/n$ for $i \in [n]$. Set $\lambda_0 = 0$ and $t = 0$.
2. While $\lambda_t < 1$, iterate between the following steps.
   (a) Set $t \leftarrow t + 1$.
   (b) For some pre-specified $\kappa \in (0, 1)$, using binary search to find $\lambda_t \in (\lambda_{t-1}, 1]$ such that

$$n^{-1}\text{CESS}_t(\lambda_t) = \frac{\left(\sum_{i=1}^n w_{t-1}^{(i)}(\gamma/q)(\boldsymbol{\theta}_{t-1}^{(i)})^{\lambda_t - \lambda_{t-1}}\right)^2}{\sum_{i=1}^n w_{t-1}^{(i)}(\gamma/q)(\boldsymbol{\theta}_{t-1}^{(i)})^{2(\lambda_t - \lambda_{t-1})}} = \kappa. \tag{B.2}$$

   If $n^{-1}\text{CESS}_t(1) > \kappa$, set $\lambda_t = 1$ and $T = t$.
   (c) Compute the unnormalized weights $w_t^{(i)} = (\gamma_t/\gamma_{t-1})(\boldsymbol{\theta}_{t-1}^{(i)})$ for $i \in [n]$. The geometric sequence of the auxiliary distributions $\gamma_t(\boldsymbol{\theta})$ is defined as

$$\gamma_0(\boldsymbol{\theta}) = q(\boldsymbol{\theta}),\ \gamma_T(\boldsymbol{\theta}) = \gamma(\boldsymbol{\theta}),\ \text{and}\ \gamma_t(\boldsymbol{\theta}) = q(\boldsymbol{\theta})^{\lambda_t}\gamma(\boldsymbol{\theta})^{1-\lambda_t}.$$

   (d) Compute $r_t = \sum_{i=1}^n w_{t-1}^{(i)} w_t^{(i)}$, which is an estimate of Z_t/Z_{t-1}. Z_t is the normalizing constant of γ_t.
   (e) Normalize the weights $\{w_t^{(i)}\}_{i\in[n]}$ to sum 1.
   (f) Compute the (normalized) effective sample size

$$\text{ESS}_t = \frac{1}{n\sum_{i=1}^n \left(w_t^{(i)}\right)^2}. \tag{B.3}$$

   If $\text{ESS}_t \leq 0.5$, resample particles using systematic resampling, and set $w_t^{(i)} = 1/n$ for $i \in [n]$.
   (g) Move particles $\boldsymbol{\theta}_t^{(i)}$ according to the Markov kernel K_t for $i \in [n]$.

Output: normalizing constant estimate $\widehat{Z}_\gamma = \prod_{t=1}^T r_t$.

method is robust to the choice of c in the region $[0.1, 0.3]$.

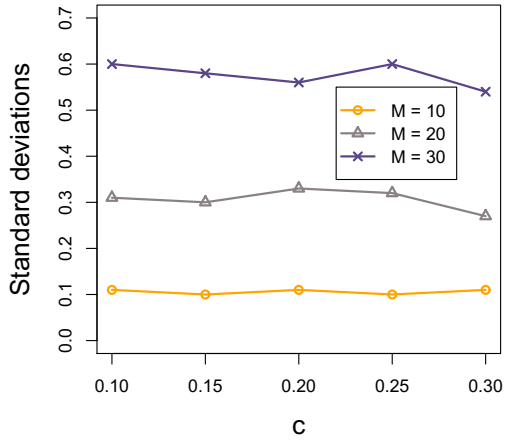


Figure B.1: Demonstration of the effect of the threshold c used in the flat histogram criterion (see Algorithm 4, step 3(e)) on the Log-Gaussian Cox process. The y-axis is the empirical standard deviations of the log normalizing constant estimates over 10 independent runs. The unit square is discretized into an $M \times M$ regular grid (see Section 2.6.1).

B.2.2 Variational Approximation for Bayesian Lasso

Algorithm 15 details the coordinate ascent variational inference (CAVI) algorithm used in constructing the surrogate distribution for the Bayesian Lasso example (see Section 2.6.2). We use the mean-field variational family, in which we assume

$$q(\beta_j) \sim N(m_j, s_j^2), \quad q(\eta_j) \sim N(\phi_j, \zeta_j^2), \quad \text{for } j \in [p], \quad \text{and} \quad q(\xi) \sim N(u, v^2). \tag{B.4}$$

The definitions of $\boldsymbol{\beta}$, $\boldsymbol{\eta}$ and ξ can be found in Section 2.6.2.

Algorithm 15: The CAVI updates for the Bayesian Lasso.

1. Given $q(\beta_i)$ for $i \neq j$, $q(\boldsymbol{\eta})$ and $q(\xi)$, update $q(\beta_j)$.

$$(m_j, s_j) = \underset{(m,\, s>0)}{\arg\max} \left\{ e^{-u+v^2/2} \big[A_{-j} m - \frac{1}{2} B_j \left(m^2 + s^2 \right) \big] + \frac{1}{2} \log s^2 \right\},$$

   in which

$$A_{-j} = \sum_{k=1}^{n} X_{kj} y_k - \sum_{k \neq j} (X^\intercal X)_{kj} m_k, \qquad B_j = (X^\intercal X)_{jj} + e^{-\phi_j + \zeta_j^2/2}.$$

2. Given $q(\eta_i)$ for $i \neq j$, $q(\boldsymbol{\beta})$ and $q(\xi)$, update $q(\eta_j)$.

$$(\phi_j, \zeta_j) = \underset{(\phi,\, \zeta>0)}{\arg\max} \left\{ \phi - \lambda^2 e^{\phi + \zeta^2/2} - (m_j^2 + s_j^2) e^{-u+v^2/2} e^{-\phi+\zeta^2/2} + \log \zeta^2 \right\}.$$

3. Given $q(\boldsymbol{\beta})$, $q(\boldsymbol{\eta})$, update $q(\xi)$.

$$(u, v) = \underset{(u,\, v>0)}{\arg\max} \left\{ -(n+p)u - C e^{-u+v^2/2} + \log v^2 \right\},$$

   in which

$$C = \boldsymbol{y}^\intercal \boldsymbol{y} - 2 \boldsymbol{y}^\intercal X m + \text{tr} \left[(m m^\intercal + \text{diag}(s_k^2)) X^\intercal X \right] + \sum_{k=1}^{p} (m_k^2 + s_k^2) e^{-\phi_k + \zeta_k^2/2}.$$

C

Supplemental Materials of Chapter 3

C.1 Proofs

C.1.1 Proof of Proposition 3.3.1

For the ease of presentation, we introduce the following notations. For $t \in \mathbb{R}$, denote

$$\begin{aligned} \widehat{G}_p^0(t) &= \frac{1}{p_0}\sum_{j\in S_0} 1(M_j > t), \quad \widehat{V}_p^0(t) = \frac{1}{p_0}\sum_{j\in S_0} 1(M_j < -t), \\ \widehat{G}_p^1(t) &= \frac{1}{p_1}\sum_{j\in S^\star} 1(M_j > t), \quad G_p^0(t) = \frac{1}{p_0}\sum_{j\in S_0} \mathbb{P}(M_j > t). \end{aligned} \tag{C.1}$$

Let $r_p = p_1/p_0$. In addition, denote

$$\mathrm{FDP}_p(t) = \frac{\widehat{G}_p^0(t)}{\widehat{G}_p^0(t) + r_p\widehat{G}_p^1(t)}, \quad \mathrm{FDP}_p^\dagger(t) = \frac{\widehat{V}_p^0(t)}{\widehat{G}_p^0(t) + r_p\widehat{G}_p^1(t)}, \quad \overline{\mathrm{FDP}}_p(t) = \frac{G_p^0(t)}{G_p^0(t) + r_p\widehat{G}_p^1(t)}. \tag{C.2}$$

Lemma C.1.1. Under Assumption 3.3.2, if $p_0 \to \infty$ as $p \to \infty$, we have in probability,

$$\sup_{t\in\mathbb{R}} \left|\widehat{G}_p^0(t) - G_p^0(t)\right| \longrightarrow 0, \quad \sup_{t\in\mathbb{R}} \left|\widehat{V}_p^0(t) - G_p^0(t)\right| \longrightarrow 0. \tag{C.3}$$

Proof of Lemma C.1.1. For any $\epsilon \in (0,1)$, denote $-\infty = \alpha_0^p < \alpha_1^p < \cdots < \alpha_{N_\epsilon}^p = \infty$ in which $N_\epsilon = \lceil 2/\epsilon \rceil$, such that $G_p^0(\alpha_{k-1}^p) - G_p^0(\alpha_k^p) \le \epsilon/2$ for $k \in [N_\epsilon]$. Such a sequence $\{\alpha_k^p\}$ exists because by Assumption 3.3.2, all the mirror statistics M_j's are continuous random variables so that

$G_p^0(t)$ is a continuous function with respect to $t \in \mathbb{R}$. We have

$$\begin{aligned}\mathbb{P}\big(\sup_{t\in\mathbb{R}} \widehat{G}_p^0(t) - G_p^0(t) > \epsilon\big) &\le \mathbb{P}\big(\bigcup_{k=1}^{N_\epsilon} \sup_{t\in[\alpha_{k-1}^p,\alpha_k^p)} \widehat{G}_p^0(t) - G_p^0(t) > \epsilon\big) \\ &\le \sum_{k=1}^{N_\epsilon} \mathbb{P}\big(\sup_{t\in[\alpha_{k-1}^p,\alpha_k^p)} \widehat{G}_p^0(t) - G_p^0(t) > \epsilon\big).\end{aligned} \tag{C.4}$$

We note that both $\widehat{G}_p^0(t)$ and $G_p^0(t)$ are monotonic decreasing function. Therefore, for any $k \in [N_\epsilon]$, we have

$$\sup_{t\in[\alpha_{k-1}^p,\alpha_k^p)} \widehat{G}_p^0(t) - G_p^0(t) \le \widehat{G}_p^0(\alpha_{k-1}^p) - G_p^0(\alpha_k^p) \le \widehat{G}_p^0(\alpha_{k-1}^p) - G_p^0(\alpha_{k-1}^p) + \epsilon/2. \tag{C.5}$$

Based on Equation (C.4), Assumption 3.3.2, and the Chebyshev's inequality, it follows that

$$\mathbb{P}\big(\sup_{t\in\mathbb{R}} \widehat{G}_p^0(t) - G_p^0(t) > \epsilon\big) \le \sum_{k=1}^{N_\epsilon} \mathbb{P}\big(\widehat{G}_p^0(\alpha_{k-1}^p) - G_p^0(\alpha_{k-1}^p) > \frac{\epsilon}{2}\big) \le \frac{4CN_\epsilon}{p_0^{2-\alpha}\epsilon^2} \to 0, \tag{C.6}$$

as $p \to \infty$. Similarly, we can show that

$$\mathbb{P}\big(\inf_{t\in\mathbb{R}} \widehat{G}_p^0(t) - G_p^0(t) < -\epsilon\big) \le \sum_{k=1}^{N_\epsilon} \mathbb{P}\big(\widehat{G}_p^0(\alpha_k^p) - G_p^0(\alpha_k^p) < -\frac{\epsilon}{2}\big) \le \frac{4CN_\epsilon}{p_0^{2-\alpha}\epsilon^2} \to 0, \tag{C.7}$$

as $p \to \infty$. This concludes the proof that $\sup_{t\in\mathbb{R}} \left|\widehat{G}_p^0(t) - G_p^0(t)\right| \to 0$ in probability. The convergence of $\sup_{t\in\mathbb{R}} \left|\widehat{V}_p^0(t) - G_p^0(t)\right|$ can be shown similarly using the symmetric assumption of the mirror statistics M_j for $j \in S_0$.

Proof of Proposition 3.3.1. Notice that

$$\begin{aligned}
\limsup_{p\to\infty} \mathrm{FDR} &= \limsup_{p\to\infty} \mathbb{E}\left[\mathrm{FDP}_p\left(\tau_q\right)\right] \\
&\le \limsup_{p\to\infty} \mathbb{E}\left|\mathrm{FDP}_p\left(\tau_q\right) - \overline{\mathrm{FDP}}_p\left(\tau_q\right)\right| + \limsup_{p\to\infty} \mathbb{E}\left[\mathrm{FDP}_p^{\dagger}\left(\tau_q\right)\right] \\
&\quad + \limsup_{p\to\infty} \mathbb{E}\left|\mathrm{FDP}_p^{\dagger}\left(\tau_q\right) - \overline{\mathrm{FDP}}_p\left(\tau_q\right)\right| \qquad \text{(C.8)} \\
&\le \limsup_{p\to\infty} \mathbb{E}\left[\sup_{t>0}\left|\mathrm{FDP}_p(t) - \overline{\mathrm{FDP}}_p(t)\right|\right] + \limsup_{p\to\infty} \mathbb{E}\left[\mathrm{FDP}_p^{\dagger}\left(\tau_q\right)\right] \\
&\quad + \limsup_{p\to\infty} \mathbb{E}\left[\sup_{t>0}\left|\mathrm{FDP}_p^{\dagger}(t) - \overline{\mathrm{FDP}}_p(t)\right|\right].
\end{aligned}$$

The first two terms are 0 based on Lemma C.1.1 and the dominated convergence theorem. For the last term, we have $\mathrm{FDP}_p^{\dagger}\left(\tau_q\right) \le q$ almost surely based on the definition of τ_q. This concludes the proof of Proposition 3.3.1.

C.1.2 Proof of Proposition 3.3.2

Lemma C.1.2. For any FDR level $q \in (0,1)$, let ℓ be the largest value in $[p]$ such that $I_{(1)}+\cdots+I_{(\ell)} \le q$, where $0 \le I_{(1)} \le I_{(2)} \le \cdots \le I_{(p)}$ are the order statistics of the population inclusion rates. Let $p_0 = |S_0|$, $p_1 = |S^{\star}|$, and $p = p_0 + p_1$. Under Assumption 3.3.3(a) and 3.3.3(c), when $p_0 \to \infty$, we have

$$\liminf_{p\to\infty} \frac{\ell}{p_0} \ge 1. \qquad \text{(C.9)}$$

Proof of Lemma C.1.2. Because ℓ is the largest index in $[p]$ such that $I_{(1)} + \cdots + I_{(\ell)} \le q$, we have $I_{(1)}+\cdots+I_{(\ell+1)} > q$. Therefore, $I_{(\ell+1)} > q/(\ell+1)$. By Assumption 3.3.3(c), for any $\epsilon > 0$, there exists $\tilde{p}$, such that when $p \ge \tilde{p}$, we have $\sum_{j\in S_0} I_j \le q + \epsilon$. By Assumption 3.3.3(a), we have $I_j \le (q+\epsilon)/p_0$ for any $j \in S_0$. We next consider the following two cases. (i) If $q/(\ell+1) \ge (q+\epsilon)/p_0$,

we have $I_j < I_{(\ell+1)}$ for any $j \in S_0$. Therefore $\ell \geq p_0$. (ii) If $q/(\ell+1) < (q+\epsilon)/p_0$, we have

$$\frac{\ell}{p_0} > \frac{q}{q+\epsilon} - \frac{1}{p_0}. \tag{C.10}$$

Because ϵ can be arbitrarily small and $p_0 \to \infty$, we conclude the proof.

Proof of Proposition 3.3.2. By Assumption 3.3.3(c), for any $\epsilon > 0$, there exists $\tilde{p}'$, such that when $p \geq \tilde{p}'$, we have $\sum_{j \in S_0} I_j \leq q + \epsilon$. By Assumption 3.3.3(a), we have $I_j \leq (q+\epsilon)/p_0$ for any $j \in S_0$. By Assumption 3.3.3(b), with the same ϵ as before and $\alpha = (q+\epsilon)/p_0$, there exists $\tilde{p}''$, such that when $p \geq \tilde{p}''$, we have

$$\frac{1}{p_1} \# \left\{ j : j \in S^\star, I_j \leq \frac{q+\epsilon}{p_0} \right\} \leq q + 2\epsilon. \tag{C.11}$$

By Lemma C.1.2, with the same ϵ as before, there exists $\tilde{p}'''$, such that when $p \geq \tilde{p}'''$, we have $\ell/p_0 \geq 1 - \epsilon$. In the following, we assume $p \geq \max\{\tilde{p}', \tilde{p}'', \tilde{p}'''\}$. We next consider the following two cases. (1) If $I_{(\ell)} > (q+\epsilon)/p_0$, then $I_{(\ell)} > I_j$ for any $j \in S_0$. This implies that none of the null features will be selected, thus the FDR is simply 0. (2) If $I_{(\ell)} \leq (q+\epsilon)/p_0$, we have

$$\begin{aligned}
\frac{\sum_{j \in S_0} 1(I_j > I_{(\ell)})}{\sum_{j=1}^p 1(I_j > I_{(\ell)}) \vee 1} &= 1 - \frac{\sum_{j \in S^\star} 1(I_j > I_{(\ell)})}{\sum_{j=1}^p 1(I_j > I_{(\ell)}) \vee 1} \leq 1 - \frac{\sum_{j \in S^\star} 1(I_j > (q+\epsilon)/p_0)}{p - \ell} \\
&\leq 1 - \frac{p_1 - (q+2\epsilon)p_1}{p - \ell} \leq 1 - \frac{p_1 - (q+2\epsilon)p_1}{p - p_0(1-\epsilon)} \\
&= q + \frac{(1 - q + p_1/p_0)\epsilon}{p_1/p_0 + \epsilon}.
\end{aligned} \tag{C.12}$$

Because ϵ can be arbitrarily small and $\liminf_{p \to \infty} p_1/p_0 > 0$, we conclude the proof.

C.1.3 Proof of Proposition 3.3.3

Suppose $p_j \leq p'_j$. Let $Z_j = \bar{\boldsymbol{X}}_j^{(1)} - \bar{\boldsymbol{X}}_j^{(2)}$ be the difference of the two sample means, which follows $N(0, 4\sigma^2/n)$ regardless of the data split. Z_j is also independent to $\bar{\boldsymbol{X}}_j = (\bar{\boldsymbol{X}}_j^{(1)} + \bar{\boldsymbol{X}}_j^{(2)})/2$ because of the normality. The mirror statistic is thus $M_j = |2\bar{\boldsymbol{X}}_j| - |Z_j|$. It is sufficient for us to prove that $\mathbb{E}[1(j \in \widehat{S})/(|\widehat{S}| \vee 1) \big| |\bar{\boldsymbol{X}}_j|]$ is a monotone increasing function of $|\bar{\boldsymbol{X}}_j|$. By conditioning on Z_j, we have

$$\begin{aligned} &\mathbb{E}\left[\frac{1(j \in \widehat{S})}{|\widehat{S}| \vee 1} \,\middle|\, \bar{\boldsymbol{X}}_j\right] - \mathbb{E}\left[\frac{1(j \in \widehat{S}')}{|\widehat{S}'| \vee 1} \,\middle|\, \bar{\boldsymbol{X}}'_j\right] \\ &\qquad = \mathbb{E}\left(\mathbb{E}\left[\frac{1(j \in \widehat{S})}{|\widehat{S}| \vee 1} \,\middle|\, \bar{\boldsymbol{X}}_j, Z_j\right] - \mathbb{E}\left[\frac{1(j \in \widehat{S}')}{|\widehat{S}'| \vee 1} \,\middle|\, \bar{\boldsymbol{X}}'_j, Z_j\right]\right). \end{aligned} \tag{C.13}$$

By conditioning on $\bar{\boldsymbol{X}}_j, Z_j$ and $\bar{\boldsymbol{X}}'_j, Z_j$, we have $M_j \geq M'_j$. For any realization of the rest of data X_{ik} for $i \in [n]$ and $k \neq j$, which is independent to $\bar{\boldsymbol{X}}_j, Z_j$ and $\bar{\boldsymbol{X}}'_j, Z_j$, we have $1(j \in \widehat{S})/(|\widehat{S}| \vee 1) \geq 1(j \in \widehat{S}')/(|\widehat{S}'| \vee 1)$. This can be argued by considering the three cases including $\tau_q \leq M'_j$, $M'_j \leq \tau_q < M_j$, and $\tau_q \geq M_j$, where τ_q is the cutoff of mirror statistics defined in Equation (3.3). In the first case, we have $1(j \in \widehat{S})/(|\widehat{S}| \vee 1) = 1(j \in \widehat{S}')/(|\widehat{S}'| \vee 1) = 1/(|\widehat{S}| \vee 1)$, while in the third case, we have $1(j \in \widehat{S})/|\widehat{S}| = 1(j \in \widehat{S}')/|\widehat{S}'| = 0$. In the second case, since $1(j \in \hat{S}') = 0$, we have $1(j \in \widehat{S})/(|\widehat{S}| \vee 1) \geq 1(j \in \widehat{S}')/(|\widehat{S}'| \vee 1)$.

For mathematical convenience, we consider the mirror statistics defined to be

$$M_j = \frac{1}{4}\left|T_j^{(1)} + T_j^{(2)}\right|^2 - \frac{1}{4}\left|T_j^{(1)} - T_j^{(2)}\right|^2 = T_j^{(1)} T_j^{(2)} \tag{C.14}$$

in the proofs of Propositions 3.4.1, 3.4.2, and 3.4.4. The proofs can be adapted to the cases where we use the mirror statistics

$$M_j = \left|T_j^{(1)} + T_j^{(2)}\right| - \left|T_j^{(1)} - T_j^{(2)}\right|. \tag{C.15}$$

Let $Q(t) = 1 - \Phi(t)$, where Φ is the CDF of the standard normal distribution. Let $\phi(t)$ be the probability density function of the standard normal distribution. Denote $H(t) = \mathbb{P}(Z_1 Z_2 > t)$ for $t \in \mathbb{R}$, where Z_1 and Z_2 are independent, following the standard normal distribution.

C.1.4 Proof of Proposition 3.4.1

We define the following normalized versions of Ω and Λ,

$$\Omega^0_{ij} = \frac{\Omega_{ij}}{\sqrt{\Omega_{ii}\Omega_{jj}}}, \quad \Lambda^0_{ij} = \frac{\Lambda_{ij}}{\sqrt{\Lambda_{ii}\Lambda_{jj}}}. \tag{C.16}$$

Technical lemmas

Lemma C.1.3. Let $Z = (Z_1, Z_2)^\top$ follow a centered and standardized bivariate normal distribution with correlation ρ. For any $t_1, t_2 \in \mathbb{R}$, we have

$$\mathbb{P}\left(Z_1 \geq t_1, Z_2 \geq t_2\right) \leq \frac{1+|\rho|}{\sqrt{1-\rho^2}} Q\left(\frac{t_1}{\sqrt{1+|\rho|}}\right) Q\left(\frac{t_2}{\sqrt{1+|\rho|}}\right). \tag{C.17}$$

Proof of Lemma C.1.3. Denote Σ_Z as the covariance matrix of Z. The eigenvalues of Σ_Z^{-1} are $1/(1-\rho)$ and $1/(1+\rho)$. Therefore,

$$\begin{aligned}
\mathbb{P}\left(Z_1 \geq t_1, Z_2 \geq t_2\right) &= \frac{1}{2\pi\sqrt{1-\rho^2}} \int_{t_1}^{\infty}\int_{t_2}^{\infty} \exp\left(-\frac{1}{2} x^\top \Sigma_Z^{-1} x\right) dx \\
&\leq \frac{1}{2\pi\sqrt{1-\rho^2}} \int_{t_1}^{\infty} \exp\left(-\frac{x_1^2}{1+|\rho|}\right) dx_1 \int_{t_2}^{\infty} \exp\left(-\frac{x_2^2}{1+|\rho|}\right) dx_2 \\
&= \frac{1+|\rho|}{\sqrt{1-\rho^2}} Q\left(\frac{t_1}{\sqrt{1+|\rho|}}\right) Q\left(\frac{t_2}{\sqrt{1+|\rho|}}\right).
\end{aligned} \tag{C.18}$$

Lemma C.1.4. Under Assumption 3.4.1, for any $\epsilon > 0$, there exist constants $c_1, c_2, c_3, c_4 > 0$ such

that for large enough n,

$$\mathbb{P}\left(\max_{i,j\in[p]}|\Lambda_{ij}-\Omega_{ij}|\leq\frac{1}{\sqrt{\log p}}\right)\geq 1-c_1p^2\exp\left(-c_2\frac{n}{\log p}\right)-c_3p^2\exp\left(-c_4n\right)-\epsilon. \quad \text{(C.19)}$$

Proof of Lemma C.1.4. The proof follows similarly as the proof of Lemma 7.2 in Javanmard and Montanari (2013). For any $\epsilon > 0$, by Proposition 5.1 in Javanmard and Montanari (2013) (also see Van de Geer et al. (2014)), there exists n_ϵ such that for $n \geq n_\epsilon$, we have

$$\mathbb{P}\left(||\widehat{\Omega}-\Omega||_\infty\leq\frac{1}{\sqrt{\log p}}\right)\geq 1-\epsilon. \quad \text{(C.20)}$$

In the following, we assume $n \geq n_\epsilon$, and condition on the event $||\widehat{\Omega}-\Omega||_\infty \leq 1/\sqrt{\log p}$. Let $\boldsymbol{v}=\Omega^\top\boldsymbol{e}_i, \boldsymbol{u}=\Omega^\top\boldsymbol{e}_j, \boldsymbol{\delta}=(\Omega-\widehat{\Omega})^\top\boldsymbol{e}_i, \boldsymbol{\eta}=(\Omega-\widehat{\Omega})^\top\boldsymbol{e}_j$. We have the following decomposition,

$$\begin{aligned}\Lambda_{ij}-\Omega_{i,j}&=(\boldsymbol{v}-\boldsymbol{\delta})^\top\widehat{\Sigma}(\boldsymbol{u}-\boldsymbol{\eta})-\Omega_{i,j}\\&=\left(\boldsymbol{v}^\top\widehat{\Sigma}\boldsymbol{u}-\Omega_{i,j}\right)-\boldsymbol{v}^\top\widehat{\Sigma}\boldsymbol{\eta}-\boldsymbol{\delta}^\top\widehat{\Sigma}\boldsymbol{u}+\boldsymbol{\delta}^\top\widehat{\Sigma}\boldsymbol{\eta}.\end{aligned} \quad \text{(C.21)}$$

We proceed to bound each term. Consider first $\boldsymbol{v}^\top\widehat{\Sigma}\boldsymbol{u}-\Omega_{i,j}$. We note that

$$\mathbb{E}[\boldsymbol{v}^\top\widehat{\Sigma}\boldsymbol{u}]=\boldsymbol{v}^\top\Sigma\boldsymbol{u}=\Omega_{ij}.$$

Thus, it follows that

$$\begin{aligned}(\boldsymbol{v}-\boldsymbol{\delta})^\top\widehat{\Sigma}(\boldsymbol{u}-\boldsymbol{\eta})-\Omega_{i,j}&=\boldsymbol{v}^\top\widehat{\Sigma}\boldsymbol{u}-E[\boldsymbol{v}^\top\widehat{\Sigma}\boldsymbol{u}]\\&=\frac{1}{n}\sum_{k=1}^n\boldsymbol{e}_i^\top\Omega\left(\boldsymbol{X}_k\boldsymbol{X}_k^\top-E[\boldsymbol{X}_kX_k^\top]\right)\Omega^\top\boldsymbol{e}_j.\end{aligned} \quad \text{(C.22)}$$

Denote $\xi_k = \boldsymbol{e}_i^\top\Omega\left(\boldsymbol{X}_k\boldsymbol{X}_k^\top-E[\boldsymbol{X}_k\boldsymbol{X}_k^\top]\right)\Omega^\top\boldsymbol{e}_j$ for $k \in [n]$. ξ_k's are independent, and have

sub-exponential tails because

$$||\xi_k||_{\psi_1} \leq 2||(\boldsymbol{e}_i^\top \Omega \boldsymbol{X}_k)^2||_{\psi_1} \leq 4||\boldsymbol{e}_i^\top \Omega \boldsymbol{X}_k||_{\psi_2}^2 \leq 4C^2/\sigma_{\min}^2(\Sigma) := K, \tag{C.23}$$

where the first inequality follows from Vershynin (2010) (Remark 5.18), the second inequality follows from Lemma C.1 in Javanmard and Montanari (2013), and the last inequality follows from Assumption 3.4.1. Without loss of generality, we assume $K \geq 1$. Using a Bernstein-type inequality (Proposition 5.26 in Vershynin (2010)), we have

$$\mathbb{P}\left(\left|\frac{1}{n}\sum_{k=1}^{n}\xi_k\right| \geq t\right) \leq 2\exp\left(-cn\min\left(\frac{t^2}{K^2}, \frac{t}{K}\right)\right) \tag{C.24}$$

Plugging in $t = 1/\sqrt{\log p}$ and employing the union bound, we obtain

$$\mathbb{P}\left(\max_{i,j\in[p]}\left|\boldsymbol{v}^\top \widehat{\Sigma}\boldsymbol{u} - \Omega_{i,j}\right| \geq \frac{1}{\sqrt{\log p}}\right) \leq 2p^2\exp\left(-c_2\frac{n}{\log p}\right) \tag{C.25}$$

for some constant c_1^2.

Next we consider $\boldsymbol{\delta}^\top \widehat{\Sigma}\boldsymbol{\eta}$. Since $\widehat{\Sigma} \succeq 0$, we have $|\boldsymbol{\delta}^\top \widehat{\Sigma}\boldsymbol{\eta}| \leq [\boldsymbol{\delta}^\top \widehat{\Sigma}\boldsymbol{\delta}]^{1/2}[\boldsymbol{\eta}^\top \widehat{\Sigma}\boldsymbol{\eta}]^{1/2}$ using Cauchy-Schwartz inequality. Therefore, it is sufficient for us to upper bound $\boldsymbol{\delta}^\top \widehat{\Sigma}\boldsymbol{\delta}$ and $\boldsymbol{\eta}^\top \widehat{\Sigma}\boldsymbol{\eta}$. We have

$$\boldsymbol{\delta}^\top \Sigma\boldsymbol{\delta} = \sum_{i,j\in[p]} \widehat{\Sigma}_{ij}\delta_i\delta_j \leq |\widehat{\Sigma}|_\infty||\boldsymbol{\delta}||_1^2 \leq |\widehat{\Sigma}|_\infty||\Omega - \widehat{\Omega}||_\infty^2 \leq \frac{|\widehat{\Sigma}|_\infty}{\log p}. \tag{C.26}$$

The tail probability $\mathbb{P}(|\widehat{\Sigma}|_\infty \geq 2)$ can be bounded as follows

$$\mathbb{P}(|\widehat{\Sigma}|_\infty \geq 2) \leq \sum_{i,j\in[p]} \mathbb{P}(|\widehat{\Sigma}_{ij}| \geq 2) \leq \sum_{i,j\in[p]} \mathbb{P}(|\widehat{\Sigma}_{ij} - \mathbb{E}\widehat{\Sigma}_{ij}| \geq 1). \tag{C.27}$$

Let $\xi_k = \boldsymbol{e}_i^\top \boldsymbol{X}_k \boldsymbol{X}_k^\top \boldsymbol{e}_j - \Sigma_{ij}$, we have

$$\begin{aligned} ||\xi_k||_{\psi_1} &\leq 2||(\boldsymbol{e}_i^\top \boldsymbol{X}_k)^2||_{\psi_1} \leq 4||\boldsymbol{e}_i^\top \boldsymbol{X}_k||_{\psi_2}^2 \\ &\leq 4||\boldsymbol{e}_i^\top \Omega^{-1/2}\Omega^{1/2}\boldsymbol{X}_k||_{\psi_2}^2 \leq 4\sigma_{\max}^2(\Sigma)C^2. \end{aligned} \tag{C.28}$$

Thus ξ_k's are independent sub-exponential random variables. By the Bernstein inequality, we have

$$\mathbb{P}(|\widehat{\Sigma}|_\infty \geq 2) \leq 2p^2 \exp(-c_4 n) \tag{C.29}$$

for some constant c_4. We remark that although $\boldsymbol{\delta} = (\Omega - \widehat{\Omega})^\top \boldsymbol{e}_i$ is implicitly associated with the index i, the upper bound $|\widehat{\Sigma}|_\infty ||\Omega - \widehat{\Omega}||_\infty^2$ in Equation (C.26) is irrelevant to the index i. Therefore, we have shown that

$$\mathbb{P}\left(\max_{i,j\in[p]} \boldsymbol{\delta}^\top \widehat{\Sigma}\boldsymbol{\eta} \geq \frac{2}{\log p}\right) \leq 4p^2 \exp(-c_4 n) + 2\epsilon, \tag{C.30}$$

where the maximum is taken with respect to the implicit index i, j associated with $\boldsymbol{\delta}$ and $\boldsymbol{\eta}$.

We now consider $\boldsymbol{v}^\top \widehat{\Sigma}\boldsymbol{\eta}$. We have

$$\begin{aligned} \max_{i,j\in[p]} |\boldsymbol{v}^\top \widehat{\Sigma}\boldsymbol{\eta}| &\leq \max_{i,j\in[p]} [\boldsymbol{v}^\top \widehat{\Sigma}\boldsymbol{v}]^{1/2} [\boldsymbol{\eta}^\top \widehat{\Sigma}\boldsymbol{\eta}]^{1/2} \\ &\leq \max_{i,j\in[p]} [|\boldsymbol{v}^\top \widehat{\Sigma}\boldsymbol{v} - \Omega_{ii}| + |\Omega_{ii}|]^{1/2} [\boldsymbol{\eta}^\top \widehat{\Sigma}\boldsymbol{\eta}]^{1/2} \\ &\leq [\max_{i\in[p]} |\boldsymbol{v}^\top \widehat{\Sigma}\boldsymbol{v} - \Omega_{ii}| + \max_{i\in[p]} |\Omega_{ii}|]^{1/2} \max_{j\in[p]} [\boldsymbol{\eta}^\top \widehat{\Sigma}\boldsymbol{\eta}]^{1/2}. \end{aligned} \tag{C.31}$$

Under Assumption 3.4.1, $\max_{i\in[p]} |\Omega_{ii}|$ is upper bounded by some constant. Combining the inequalities in Equation (C.25) and Equation (C.30), we have

$$\mathbb{P}\left(\max_{i,j\in[p]} |\boldsymbol{v}^\top \widehat{\Sigma}\boldsymbol{\eta}| \geq \frac{c}{\sqrt{\log p}}\right) \leq 2p^2 \exp\left(-c_2 \frac{n}{\log p}\right) + 2p^2 \exp(-c_4 n) + \epsilon. \tag{C.32}$$

The decomposition in Equation (C.21), along with the inequalities in Equation (C.25), Equation (C.30) and Equation (C.32) together imply the claim in Lemma C.1.4.

Corollary C.1.1. The high probability bound in Lemma C.1.4 also applies to the normalized versions Λ^0 and Ω^0. That is, under Assumption 3.4.1, for any $\epsilon > 0$, there exist constants $c_1, c_2, c_3, c_4 > 0$ such that for large enough n,

$$\mathbb{P}\left(\max_{i,j\in[p]} \left|\Lambda^0_{ij} - \Omega^0_{ij}\right| \leq \frac{1}{\sqrt{\log p}}\right) \geq 1 - c_1 p^2 \exp\left(-c_2 \frac{n}{\log p}\right) - c_3 p^2 \exp\left(-c_4 n\right) - \epsilon. \quad \text{(C.33)}$$

Proof of Corollary C.1.1. Under Assumption 3.4.1, there exists some constants K_1, K_2 such that

$$0 < K_1 \leq \min_{i\in[p]} \Omega_{ii} \leq \max_{i\in[p]} \Omega_{ii} \leq K_2. \quad \text{(C.34)}$$

We proceed to show that

$$\max_{i,j\in[p]} \left|\Lambda^0_{ij} - \Omega^0_{ij}\right| \leq c/\sqrt{\log p} \quad \text{(C.35)}$$

for some constant c, conditioning on the high probability event $\left\{\max_{i,j\in[p]} |\Lambda_{ij} - \Omega_{ij}| \leq 1/\sqrt{\log p}\right\}$. We have

$$\begin{aligned}
\max_{i,j\in[p]} \left|\Lambda^0_{ij} - \Omega^0_{ij}\right| &= \max_{i,j\in[p]} \left|\frac{\Lambda_{ij}}{\sqrt{\Lambda_{ii}\Lambda_{jj}}} - \frac{\Omega_{ij}}{\sqrt{\Omega_{ii}\Omega_{jj}}}\right| \\
&\leq \max_{i,j\in[p]} \left|\frac{\Lambda_{ij}}{\sqrt{\Lambda_{ii}\Lambda_{jj}}} - \frac{\Lambda_{ij}}{\sqrt{\Omega_{ii}\Omega_{jj}}}\right| + \max_{i,j\in[p]} \left|\frac{\Lambda_{ij}}{\sqrt{\Omega_{ii}\Omega_{jj}}} - \frac{\Omega_{ij}}{\sqrt{\Omega_{ii}\Omega_{jj}}}\right|.
\end{aligned} \quad \text{(C.36)}$$

Consider the first term. For large enough p, we have

$$\frac{|\Lambda_{ij}|}{\sqrt{\Lambda_{ii}\Lambda_{jj}}} \leq \frac{|\Lambda_{ij} - \Omega_{ij}| + \Omega_{ij}}{\sqrt{[\Omega_{ii} - |\Lambda_{ii} - \Omega_{ii}|][\Omega_{jj} - |\Lambda_{jj} - \Omega_{jj}|]}} \leq K_3 \quad \text{(C.37)}$$

for some constant K_3. It follows that

$$\begin{aligned}
\left|\frac{\Lambda_{ij}}{\sqrt{\Lambda_{ii}\Lambda_{jj}}}-\frac{\Lambda_{ij}}{\sqrt{\Omega_{ii}\Omega_{jj}}}\right| &= \frac{|\Lambda_{ij}|}{\sqrt{\Lambda_{ii}\Lambda_{jj}}}\left|\left(\frac{\Lambda_{ii}\Lambda_{jj}}{\Omega_{ii}\Omega_{jj}}\right)^{1/2}-1\right| \leq K_3\left|\left(\frac{\Lambda_{ii}\Lambda_{jj}}{\Omega_{ii}\Omega_{jj}}\right)^{1/2}-1\right| \\
&\leq K_3\left|\left(1+\frac{1}{\sqrt{\log p}\,\Omega_{ii}}\right)^{1/2}\left(1+\frac{1}{\sqrt{\log p}\,\Omega_{jj}}\right)^{1/2}-1\right| \\
&\leq \left|1+\frac{2}{K_1\sqrt{\log p}}+\frac{1}{K_1^2\log p}-1\right| \\
&\leq \frac{K_4}{\sqrt{\log p}}
\end{aligned} \tag{C.38}$$

for some constant K_4. In the last to second inequality above, we use an elementary inequality $\sqrt{1+x} \leq 1+x$ for $x > 0$.

For the second term, we have

$$\left|\frac{\Lambda_{ij}}{\sqrt{\Omega_{ii}\Omega_{jj}}}-\frac{\Omega_{ij}}{\sqrt{\Omega_{ii}\Omega_{jj}}}\right| \leq \frac{|\Lambda_{ij}-\Omega_{ij}|}{K_1} \leq \frac{1}{K_1\sqrt{\log p}}. \tag{C.39}$$

We note that both the upper bounds in Equation (C.38) and Equation (C.39) do not depend on the index i, j. Therefore, we have shown that

$$\max_{i,j\in[p]}\left|\Lambda^0_{ij}-\Omega^0_{ij}\right| \leq (K_4+1/K_1)/\sqrt{\log p}. \tag{C.40}$$

This implies the claim in Corollary C.1.1.

Lemma C.1.5. Under Assumption 3.4.1, as $n, p \to \infty$, we have

$$\begin{aligned}
\sup_{t\in\mathbb{R},\, j\in S_0}\left|\mathbb{P}(T_j > t) - Q(t)\right| &\longrightarrow 0, \\
\sup_{t\in\mathbb{R},\, j\in S_0}\left|\mathbb{P}(T_j < t) - Q(-t)\right| &\longrightarrow 0.
\end{aligned} \tag{C.41}$$

Proof of Lemma C.1.5. Recall that

$$T_j = \sqrt{n}\widehat{\beta}_j^d/\Lambda_{jj}^{1/2} \quad \text{and} \quad \widetilde{Z}_j = Z_j/\Lambda_{jj}^{1/2} \Big| X \sim N(0,1).$$

By Lemma C.1.4, for large enough n, $\min_{j\in[p]} \Lambda_{jj} \geq c > 0$ for some constant c with high probability. Throughout we condition on this high probability event. Recall the decomposition in Equation (3.7). For $j \in S_0$, i.e., $\beta_j^\star = 0$, $T_j = \widetilde{Z}_j + \Delta_j/\Lambda_{jj}^{1/2}$. For any $t \in \mathbb{R}$, we have

$$\begin{aligned}
\mathbb{P}\left(T_j > t\right) - Q(t) &= \mathbb{P}(\widetilde{Z}_j > t - \Delta_j/\Lambda_{jj}^{1/2}) - Q(t) \\
&\leq \mathbb{P}(\widetilde{Z}_j > t - \epsilon) + \mathbb{P}\left(||\Delta||_\infty \geq \epsilon\sqrt{c}\right) - Q(t) \\
&\leq Q(t-\epsilon) - Q(t) + \epsilon \\
&\leq c_1\epsilon
\end{aligned} \tag{C.42}$$

for some constant $c_1 > 0$, in which $\epsilon = s^\star \log p/\sqrt{n} \to 0$. We use Theorem 2.3 in Javanmard and Montanari (2013) to control the bias term Δ via $\mathbb{P}\left(||\Delta||_\infty \geq \epsilon\sqrt{c}\right) \leq \epsilon$. The last inequality follows from the simple fact that for any $t \in \mathbb{R}$, $\epsilon > 0$,

$$Q(t-\epsilon) - Q(t) = \int_{t-\epsilon}^{t} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) dx \leq \frac{\epsilon}{\sqrt{2\pi}}. \tag{C.43}$$

Similarly, we have

$$\begin{aligned}
\mathbb{P}\left(T_j > t\right) - Q(t) &\geq \mathbb{P}(\widetilde{Z}_j > t + \epsilon)\mathbb{P}(||\Delta||_\infty < \epsilon\sqrt{c}) - Q(t) \\
&\geq Q(t+\epsilon)(1-\epsilon) - Q(t) \\
&\geq -c_1^2\epsilon.
\end{aligned} \tag{C.44}$$

for some constant $c_1^2 > 0$. Since the bounds in Equation (C.42) and Equation (C.44) are irrelevant to

t and the index j, the first claim in Lemma C.1.5 follows. The second claim can be shown similarly.

Lemma C.1.6. Under Assumption 3.4.1, as $n, p \to \infty$, we have

$$\begin{aligned} \sup_{t\in\mathbb{R},\, j\in S_0} |\mathbb{P}(M_j > t) - H(t)| &\longrightarrow 0, \\ \sup_{t\in\mathbb{R},\, j\in S_0} |\mathbb{P}(M_j < -t) - H(t)| &\longrightarrow 0. \end{aligned} \tag{C.45}$$

Proof of Lemma C.1.6. We prove the first claim. The second claim follows similarly. Recall that $M_j = T_j^{(1)} T_j^{(2)}$, where $T_j^{(1)}$ and $T_j^{(2)}$ are independent. For $t \in \mathbb{R}$, $H(t) = \mathbb{P}(Z_1 Z_2 > t)$, where Z_1 and Z_2 are independent following the standard normal distribution. By Lemma C.1.5, for $\epsilon = s^\star \log p/\sqrt{n} \to 0$, and any $t \in \mathbb{R}$, $j \in S_0$, we have

$$\begin{aligned} Q(t) - \epsilon \le P(T_j > t) \le Q(t) + \epsilon, \\ Q(-t) - \epsilon \le P(T_j < t) \le Q(-t) + \epsilon \end{aligned} \tag{C.46}$$

for large enough n and p. Denote $p_{T_j}(x)$ as the probability density function of T_j. It follows that

$$\begin{aligned} \mathbb{P}(M_j > t) &= \int_0^{+\infty} \mathbb{P}\left(T_j^{(2)} > \frac{t}{x}\right) p_{T_j^{(1)}}(x)dx + \int_{-\infty}^0 \mathbb{P}\left(T_j^{(2)} < \frac{t}{x}\right) p_{T_j^{(1)}}(x)dx \\ &\le \int_0^{+\infty} Q\left(t/x\right) p_{T_j^{(1)}}(x)dx + \int_{-\infty}^0 Q\left(-t/x\right) p_{T_j^{(1)}}(x)dx + \epsilon \\ &= \int_0^{+\infty} \mathbb{P}\left(T_j^{(1)} > \frac{t}{x}\right) \phi(x)dx + \int_{-\infty}^0 \mathbb{P}\left(T_j^{(2)} > -\frac{t}{x}\right) \phi(x)dx + \epsilon \\ &\le \int_0^{+\infty} Q\left(t/x\right) \phi(x)dx + \int_{-\infty}^0 Q\left(-t/x\right) \phi(x)dx + 2\epsilon \\ &= H(t) + 2\epsilon. \end{aligned} \tag{C.47}$$

Similarly, we can show that $\mathbb{P}(M_j > t) \ge H(t) - 2\epsilon$. This completes the proof of Lemma C.1.6.

Lemma C.1.7. Under Assumption 3.4.1, as $n, p \to \infty$, we have

$$\begin{aligned} \sup_{t\in\mathbb{R}} \mathrm{Var}\left(\frac{1}{p_0}\sum_{j\in S_0} 1(M_j > t)\right) &\longrightarrow 0, \\ \sup_{t\in\mathbb{R}} \mathrm{Var}\left(\frac{1}{p_0}\sum_{j\in S_0} 1(M_j < -t)\right) &\longrightarrow 0. \end{aligned} \tag{C.48}$$

Proof of Lemma C.1.7. We prove the first claim. The second claim follows similarly. Denote the correlated set as $\Gamma = \{(i,j) : i,j \in S_0,\ \Omega_{ij} \neq 0\}$, and the uncorrelated set as $\Gamma^c = \{(i,j) : i,j \in S_0,\ \Omega_{ij} = 0\}$. We have the following decomposition.

$$\begin{aligned} 0 \leq & \sup_{t\in\mathbb{R}} \mathrm{Var}\left(\frac{1}{p_0}\sum_{j\in S_0} 1(M_j > t)\right) \\ & \leq \frac{1}{p_0^2}\sup_{t\in\mathbb{R}} \sum_{(i,j)\in\Gamma} \left[\mathbb{P}(M_i > t, M_j > t) - \mathbb{P}(M_i > t)\mathbb{P}(M_j > t)\right] \\ & + \frac{1}{p_0^2}\sup_{t\in\mathbb{R}} \sum_{(i,j)\in\Gamma^c} \left[\mathbb{P}(M_i > t, M_j > t) - \mathbb{P}(M_i > t)\mathbb{P}(M_j > t)\right] \\ & \leq \frac{|\Gamma|}{p_0^2} + \frac{1}{p_0^2}\sum_{(i,j)\in\Gamma^c} \sup_{t\in\mathbb{R}} \left[\mathbb{P}(M_i > t, M_j > t) - H^2(t)\right] \\ & + \frac{1}{p_0^2}\sum_{(i,j)\in\Gamma^c} \sup_{t\in\mathbb{R}} \left[\mathbb{P}(M_i > t)\mathbb{P}(M_j > t) - H^2(t)\right]. \end{aligned} \tag{C.49}$$

By Assumption 3.4.1 (1), we have $|\Gamma|/p_0^2 \leq p_0\sqrt{n}/(p_0^2 \log p) \to 0$. Further, by Lemma C.1.6, we have

$$\sup_{t\in\mathbb{R}, i,j\in S_0} \left|\mathbb{P}(M_i > t)\mathbb{P}(M_j > t) - H^2(t)\right| \to 0. \tag{C.50}$$

Therefore, it is sufficient for us to establish a vanishing upper bound (converge to 0 as $n, p \to \infty$) of $\sup_{t\in\mathbb{R},(i,j)\in\Gamma^c} [\mathbb{P}(M_i > t, M_j > t) - H^2(t)]$.

In the following, we condition on the below high probability event

$$\left\{\min_{j\in[p]}\Lambda_{jj}\geq c\right\}\cap\left\{\max_{i,j\in[p]}|\Lambda^0_{ij}-\Omega^0_{ij}|\leq\frac{1}{\sqrt{\log p}}\right\}\cap\left\{||\Delta||_\infty\leq\epsilon\sqrt{c}\right\} \tag{C.51}$$

with $\epsilon = s^\star\log p/\sqrt{n}\to 0$ and some constant $c>0$. Besides, by Lemma C.1.5, for any $t\in\mathbb{R}$, $j\in S_0$, we have

$$\begin{aligned} &Q(t)-\epsilon\leq P(T_j>t)\leq Q(t)+\epsilon,\\ &Q(-t)-\epsilon\leq P(T_j<t)\leq Q(-t)+\epsilon \end{aligned} \tag{C.52}$$

for large enough n and p. We have the following decomposition.

$$\begin{aligned} \mathbb{P}(M_i>t,M_j>t) =& \int_0^{+\infty}\int_0^{+\infty}\mathbb{P}\left(T_i^{(2)}>\frac{t}{x},T_j^{(2)}>\frac{t}{y}\right)p_{T_i^{(1)},T_j^{(1)}}(x,y)dxdy\\ &+\int_0^{+\infty}\int_{-\infty}^{0}\mathbb{P}\left(T_i^{(2)}>\frac{t}{x},T_j^{(2)}<\frac{t}{y}\right)p_{T_i^{(1)},T_j^{(1)}}(x,y)dxdy\\ &+\int_{-\infty}^{0}\int_0^{+\infty}\mathbb{P}\left(T_i^{(2)}<\frac{t}{x},T_j^{(2)}>\frac{t}{y}\right)p_{T_i^{(1)},T_j^{(1)}}(x,y)dxdy\\ &+\int_{-\infty}^{0}\int_{-\infty}^{0}\mathbb{P}\left(T_i^{(2)}<\frac{t}{x},T_j^{(2)}<\frac{t}{y}\right)p_{T_i^{(1)},T_j^{(1)}}(x,y)dxdy\\ :=& I_1+I_2+I_3+I_4. \end{aligned} \tag{C.53}$$

$p_{T_i,T_j}(x,y)$ is the joint probability density of (T_i,T_j). For the ease of presentation, we introduce the following notations.

$$\begin{aligned} H_1(t)&=\mathbb{P}\left(Z_1Z_2>t,\ Z_1>0,\ Z_2>0\right),\\ H_2(t)&=\mathbb{P}\left(Z_1Z_2>t,\ Z_1>0,\ Z_2<0\right),\\ H_3(t)&=\mathbb{P}\left(Z_1Z_2>t,\ Z_1<0,\ Z_2>0\right),\\ H_4(t)&=\mathbb{P}\left(Z_1Z_2>t,\ Z_1<0,\ Z_2<0\right). \end{aligned} \tag{C.54}$$

Thus, it is sufficient for us to establish a vanishing upper bound (converge to 0 as $n,p\to\infty$) for each of $\{\sup_{t\in\mathbb{R},(i,j)\in\Gamma^c}\left[I_k-H_k^2(t)\right],\ k\in[4]\}$, respectively.

We detail the proof of the first one, while the other three follow similarly. Let $\rho = 1/\sqrt{\log p} \to 0$. For $(i,j) \in \Gamma^c$ and any $t_1, t_2 \in \mathbb{R}$, we have

$$\begin{aligned}\mathbb{P}(T_i > t_1, T_j > t_2) &\leq \mathbb{P}(\widetilde{Z}_i > t_1 - \epsilon, \widetilde{Z}_j > t_2 - \epsilon) \\ &\leq \frac{1+\rho}{\sqrt{1-\rho^2}} Q\left(\frac{t_1-\epsilon}{\sqrt{1+\rho}}\right) Q\left(\frac{t_2-\epsilon}{\sqrt{1+\rho}}\right) \\ &\leq \frac{1+\rho}{\sqrt{1-\rho^2}} Q\left(\frac{t_1}{\sqrt{1+\rho}}\right) Q\left(\frac{t_2}{\sqrt{1+\rho}}\right) + c_1\epsilon \end{aligned} \tag{C.55}$$

for some constant $c_1 > 0$. The second inequality follows from Lemma C.1.3. We note that for $(i,j) \in \Gamma^c$, once we condition on the event $\{\max_{i,j\in[p]} |\Lambda^0_{ij} - \Omega^0_{ij}| \leq 1/\sqrt{\log p}\}$, we have $|\Lambda^0_{i,j}| \leq 1/\sqrt{\log p}$. The third inequality follows from Equation (C.43). Plugging the inequality in Equation (C.55) into I_1, we have

$$I_1 \leq \frac{(1+\rho)^2}{1-\rho^2} \mathbb{P}\left(Z_1 Z_2 > \frac{t}{\sqrt{1+\rho}},\ Z_1 > 0,\ Z_2 > 0\right)^2 + c_1^2\epsilon. \tag{C.56}$$

for some constant $c_1^2 > 0$. Consequently, it follows that

$$\sup_{t\in\mathbb{R},(i,j)\in\Gamma^c} \left[I_1 - H_1^2(t)\right] \leq c_3\left(1 - \frac{1}{1+\rho}\right) + \left[\frac{(1+\rho)^2}{1-\rho^2} - 1\right] + c_1^2\epsilon \to 0 \tag{C.57}$$

as $n, p \to \infty$. This completes the proof of Lemma C.1.7.

Proof of Proposition 3.4.1

In addition to the notations introduced in Section C.1.1, we denote

$$G_p^0(t) = \frac{1}{p_0}\sum_{j\in S_0} \mathbb{P}(M_j > t), \quad V_p^0(t) = \frac{1}{p_0}\sum_{j\in S_0} \mathbb{P}(M_j < -t). \tag{C.58}$$

The definition of $\overline{\mathrm{FDP}}_p(t)$ is slightly modified to be

$$\overline{\mathrm{FDP}}_p(t) = \frac{H(t)}{H(t) + r_p \widehat{G}_p^1(t)}. \tag{C.59}$$

The proof of Proposition 3.4.1 is essentially the same as the proof of Proposition 3.3.1, with the help of the following Lemma C.1.8.

Lemma C.1.8. Under Assumption 3.4.1, as $n, p \to \infty$, we have in probability,

$$\begin{aligned} \sup_{t\in\mathbb{R}} \left| \widehat{G}_p^0(t) - H(t) \right| &\longrightarrow 0, \\ \sup_{t\in\mathbb{R}} \left| \widehat{V}_p^0(t) - H(t) \right| &\longrightarrow 0. \end{aligned} \tag{C.60}$$

Proof of Lemma C.1.8. We prove the first claim. The second claim follows similarly. Notice that $H(t)$ is symmetric about 0. By Lemma C.1.6, it is sufficient for us to show that

$$\begin{aligned} \sup_{t\in\mathbb{R}} \left| \widehat{G}_p^0(t) - G_p^0(t) \right| &\longrightarrow 0, \\ \sup_{t\in\mathbb{R}} \left| \widehat{V}_p^0(t) - V_p^0(t) \right| &\longrightarrow 0. \end{aligned} \tag{C.61}$$

The proof follows similarly as the proof of Lemma C.1.1 using an ϵ-net argument, except that we use Lemma C.1.7 instead of the Chebyshev's inequality in Equations (C.6) and (C.7).

C.1.5 Proof of Proposition 3.4.2

For the ease of presentation, we introduce the following notations. Let $\langle \boldsymbol{u}, \boldsymbol{v} \rangle_\Sigma = \boldsymbol{u}^\intercal \Sigma \boldsymbol{v}$ for any $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^p$, and $||\boldsymbol{u}||_\Sigma^2 = \boldsymbol{u}^\intercal \Sigma \boldsymbol{u}$. Denote $\Sigma = LL^\intercal$ as the Cholesky decomposition of the covariance matrix Σ. Let

$$\boldsymbol{\theta}^\star = L^\intercal \boldsymbol{\beta}^\star, \quad \widehat{\boldsymbol{\theta}}^{(1)} = L^\intercal \widehat{\boldsymbol{\beta}}^{(1)}, \quad \widehat{\boldsymbol{\theta}}^{(2)} = L^\intercal \widehat{\boldsymbol{\beta}}^{(2)}.$$

By Proposition 2.1 in Zhao et al. (2020), $\widehat{\boldsymbol{\Omega}}$ has the same distribution as the MLE (using half data) of the underlying GLM with true regression coefficient $\boldsymbol{\theta}^\star$ and features drawn i.i.d from $N(0, I_p)$. Let

$$\sigma_n^{(1)} = ||P_{\boldsymbol{\theta}^\star}^{\perp} \widehat{\boldsymbol{\theta}}^{(1)}||_2, \quad \sigma_n^{(2)} = ||P_{\boldsymbol{\theta}^\star}^{\perp} \widehat{\boldsymbol{\theta}}^{(2)}||_2,$$

in which $P_{\boldsymbol{\theta}^\star}^{\perp}$ is the projection matrix onto the orthogonal space of $\boldsymbol{\theta}^\star$. Define $\boldsymbol{\xi}^{(1)}, \boldsymbol{\xi}^{(2)}$ as below,

$$\begin{aligned} \boldsymbol{\xi}^{(1)} &= \boldsymbol{W}^{(1)} - \frac{1}{\omega_n^2} \langle \boldsymbol{W}^{(1)}, \boldsymbol{\beta}^\star \rangle_\Sigma \boldsymbol{\beta}^\star, \\ \boldsymbol{\xi}^{(2)} &= \boldsymbol{W}^{(2)} - \frac{1}{\omega_n^2} \langle \boldsymbol{W}^{(2)}, \boldsymbol{\beta}^\star \rangle_\Sigma \boldsymbol{\beta}^\star, \end{aligned} \tag{C.62}$$

where $\boldsymbol{W}^{(1)}, \boldsymbol{W}^{(2)}$ are i.i.d $N(0, \Omega)$ random vectors, independent to everything else.

Define $Z_j^{(1)} = \tau_j W_j^{(1)}$ and $Z_j^{(2)} = \tau_j W_j^{(2)}$ for $j \in [p]$. Thus $\boldsymbol{Z}^{(1)}$ and $\boldsymbol{Z}^{(2)}$ independently follow $N(0, R)$ with $R_{ij} = \tau_i \tau_j \Omega_{ij}$. Denote $\omega_n^2 = \mathrm{Var}(\boldsymbol{x}_i^\intercal \boldsymbol{\beta}^\star)$. We assume $\omega_n \to \omega$ as $n, p \to \infty$. With a bit abuse of notation, denote $\kappa = 2p/n \in (0, 1)$ as the ratio of dimension over sample size after data splitting. Recall that $\sigma_\star$ is defined to be the unique optimizer of the optimization problem (??).

For $j \in [p]$, let

$$T_j^{(1)} = \sqrt{\frac{n}{2}} \frac{\widehat{\tau}_j^{(1)} \widehat{\beta}_j^{(1)}}{\sigma_\star}, \quad \widetilde{T}_j^{(1)} = \sqrt{\frac{n}{2}} \frac{\tau_j \widehat{\beta}_j^{(1)}}{\sigma_n^{(1)}/\sqrt{\kappa}}, \quad \widetilde{Z}_j^{(1)} = \frac{\sqrt{p} Z_j^{(1)}}{||\boldsymbol{\xi}^{(1)}||_\Sigma}. \tag{C.63}$$

$T_j^{(2)}$, $\widetilde{T}_j^{(2)}$ and $\widetilde{Z}_j^{(2)}$ are defined similarly. We note that since the selection result, using the data-splitting approach, is scaling invariant with respect to $T_j^{(1)}$ and $T_j^{(2)}$, thus in practice, we can simply drop the constants $\sqrt{n/2}$ and $\sigma_\star$ (see Algorithm 13). For $j \in [p]$, define

$$\widetilde{M}_j = \widetilde{T}_j^{(1)} \widetilde{T}_j^{(2)}, \tag{C.64}$$

which serves as an approximation to the mirror statistic $M_j = T_j^{(1)} T_j^{(2)}$.

Technical lemmas

Lemma C.1.9. Let $\boldsymbol{Z}^{(1)}, \boldsymbol{Z}^{(2)} \in \mathbb{R}^p$ independently follow $N(0, R)$, in which $R_{jj} = 1$ for $j \in [p]$. We have

$$\sup_{t\in\mathbb{R}} \mathbb{E}\left[(\widehat{F}(t) - H(t))^2\right] \leq \frac{1}{4p} + C||R||_1, \tag{C.65}$$

where $C > 0$ is a universal constant, and

$$\widehat{F}(t) = \frac{1}{p}\sum_{j=1}^{p} 1(\boldsymbol{Z}^{(1)}\boldsymbol{Z}^{(2)} > t), \quad ||R||_1 = \frac{1}{p^2}\sum_{i,j=1}^{p} |R_{ij}|. \tag{C.66}$$

Proof of Lemma C.1.9. Lemma C.1.9 is a generalization of Theorem 1 in Azriel and Schwartzman (2015), and the proof follows similarly. We have

$$\begin{aligned}\sup_{t\in\mathbb{R}} \mathbb{E}\left[(\widehat{F}(t) - H(t))^2\right] &\leq \frac{1}{p^2}\sum_{j=1}^{p}\sup_{t\in\mathbb{R}}[H(t)(1-H(t)] \\ &\quad + \frac{1}{p^2}\sum_{i\neq j}\sup_{t\in\mathbb{R}}\left[\mathbb{P}(Z_j^{(1)}Z_j^{(2)} > t, Z_i^{(1)}Z_i^{(2)} > t) - H^2(t)\right] \\ &:= I_1 + I_2.\end{aligned} \tag{C.67}$$

By the Cauchy-Schwartz inequality, $I_1 \leq 1/(4p)$.

For I_2, we decompose it into four terms conditioning on the signs of $Z_j^{(2)}$ and $Z_i^{(2)}$. We detail one term in the following, and the rest three terms follow similarly. Let

$$I_{21} = \mathbb{P}(Z_j^{(1)}Z_j^{(2)} > t, Z_i^{(1)}Z_i^{(2)} > t, Z_j^{(2)} < 0, Z_i^{(2)} < 0) - \mathbb{P}(Z_1Z_2 > t, Z_2 < 0)^2. \tag{C.68}$$

Recall the Mehler's identity (Kotz et al., 2000), that is, for any $t_1, t_2 \in \mathbb{R}$,

$$\Phi_r(t_1, t_2) = \Phi(t_1)\Phi(t_2) + \sum_{n=1}^{\infty}\frac{r^n}{n!}\phi^{(n-1)}(t_1)\phi^{(n-1)}(t_2), \tag{C.69}$$

in which Φ_r is the bivariate CDF of standard normals with correlation r, and $\phi^{(n)}$ is the nth derivative of ϕ. Using the Mehler's identity twice, we have

$$\begin{aligned} I_{21} &= 2\int_{-\infty}^{0}\int_{-\infty}^{0}\left[\sum_{n=1}^{\infty}\frac{R_{ij}^n}{n!}\phi^{(n-1)}(t/u_1)\phi^{(n-1)}(t/u_2)\right]\phi(u_1)\phi(u_2)du_1du_2 \\ &\leq 2|R_{ij}|\int_{-\infty}^{0}\int_{-\infty}^{0}\sum_{n=1}^{\infty}\frac{\left[\sup_{t\in\mathbb{R}}\phi^{(n-1)}(t)\right]^2}{n!}\phi(u_1)\phi(u_2)du_1du_2 \\ &\leq C|R_{ij}|, \end{aligned} \tag{C.70}$$

for some universal constant $C > 0$, in which we use Lemma 1 in Azriel and Schwartzman (2015), that is,

$$\sum_{n=1}^{\infty}\frac{\left[\sup_{t\in\mathbb{R}}\phi^{(n-1)}(t)\right]^2}{n!} < \infty. \tag{C.71}$$

This completes the proof of Lemma C.1.9.

Lemma C.1.10. For $j \in S_0$, we have

$$\widetilde{M}_j \overset{d}{=} \widetilde{Z}_j^{(1)}\widetilde{Z}_j^{(2)}. \tag{C.72}$$

Proof of Lemma C.1.10. By the stochastic representation (Lemma 2.1, Proposition A.1) in Zhao et al. (2020), we have

$$\widetilde{T}_j^{(1)} = \frac{\sqrt{p}\tau_j\widehat{\beta}_j^{(1)}}{\sigma_n^{(1)}} = \frac{\sqrt{p}\tau_j L^{-\intercal}\widehat{\Omega}_j^{(1)}}{\sigma_n^{(1)}} \overset{d}{=} \frac{\sqrt{p}\tau_j L^{-\intercal}P_{\mathbf{\Omega}^\star}^{\perp}\boldsymbol{Z}}{||P_{\mathbf{\Omega}^\star}^{\perp}\boldsymbol{Z}||_2} \overset{d}{=} \frac{\sqrt{p}\tau_j\boldsymbol{\xi}^{(1)}}{||\boldsymbol{\xi}^{(1)}||_\Sigma} \overset{d}{=} \widetilde{Z}_j^{(1)}, \tag{C.73}$$

in which $\boldsymbol{Z}$ following $N(0, I_p)$ is independent to everything else. The last to second equivalence follows from the fact that $L^{-\intercal}P_{\mathbf{\Omega}^\star}^{\perp}\boldsymbol{Z} \overset{d}{=} \boldsymbol{\xi}^{(1)}$, and the last equivalence is only true for $j \in S_0$. We have a similar representation for $\widetilde{T}_j^{(2)}$. The proof thus completes.

Lemma C.1.11. We have $\sigma_n \xrightarrow{p} \sqrt{\kappa}\sigma_\star$ as $n, p \to \infty$.

Proof of Lemma C.1.11.

Lemma C.1.12. There exists a constant $c > 0$ such that for any $\epsilon \in (0, 1)$, we have

$$\mathbb{P}\left(\max_{j\in[p]}\left|\widehat{\tau}_j^{2(1)}/\tau_j^2 - 1\right| > \epsilon\right) \leq p\exp(-c(n/2 - p + 1)\epsilon^2). \tag{C.74}$$

A similar statement also holds for $\max_{j\in[p]}\left|\widehat{\tau}_j^{2(2)}/\tau_j^2 - 1\right|$.

Proof of Lemma C.1.12. We first notice that

$$\begin{aligned}
&\min_{j\in[p]}\tau_j^2 = 1/\max_{j\in[p]}\Omega_{jj} \geq 1/\sigma_{\max}(\Omega) = \sigma_{\min}(\Sigma) \geq 1/c_1 > 0,\\
&\max_{j\in[p]}\tau_j^2 = 1/\min_{j\in[p]}\Omega_{jj} \leq 1/\sigma_{\min}(\Omega) = \sigma_{\max}(\Sigma) \leq c_1 < \infty,
\end{aligned} \tag{C.75}$$

by Assumption 3.4.2. Thus, it is sufficient for us to consider $\max_{j\in[p]}\left|\widehat{\tau}_j^{2(1)} - \tau_j^2\right|$. Since $\mathrm{RSS}_j^{(1)} \sim \tau_j^2\chi^2_{n/2-p+1}$, by the union bound and a Bernstein-type inequality, for any $\epsilon \in (0, 1)$, we have

$$\begin{aligned}
\mathbb{P}\left(\max_{j\in[p]}\left|\widehat{\tau}_j^{2(1)} - \tau_j^2\right| > \epsilon\right) &\leq \sum_{j=1}^{p}\mathbb{P}\left(\left|\widehat{\tau}_j^{2(1)} - \tau_j^2\right| > \epsilon\right)\\
&\leq p\exp(-c(n/2 - p + 1)\epsilon^2),
\end{aligned} \tag{C.76}$$

for some constant $c > 0$.

Lemma C.1.13. For any Lipschitz continuous function $f : \mathbb{R} \to \mathbb{R}$, we have

$$\frac{1}{p_0}\sum_{j\in S_0}[f(M_j) - f(\widetilde{M}_j)] \xrightarrow{p} 0. \tag{C.77}$$

Proof of Lemma C.1.13. Denote the Lipschitz coefficient as L. We have

$$\begin{aligned}\left|\frac{1}{p_0}\sum_{j\in S_0}[f(M_j)-f(\widetilde{M}_j)]\right| &\le \frac{L}{p_0}\sum_{j\in S_0}\left|M_j-\widetilde{M}_j\right| = \frac{L}{p_0}\sum_{j\in S_0}\left|T_j^{(1)}T_j^{(2)}-\widetilde{T}_j^{(1)}\widetilde{T}_j^{(2)}\right| \\ &\le \frac{L}{p_0}\sum_{j\in S_0}\left|T_j^{(2)}\right|\left|T_j^{(1)}-\widetilde{T}_j^{(1)}\right| + \frac{L}{p_0}\sum_{j\in S_0}\left|\widetilde{T}_j^{(1)}\right|\left|T_j^{(2)}-\widetilde{T}_j^{(2)}\right|.\end{aligned} \tag{C.78}$$

In the following, we detail a high probability upper bound for the first term, and the second term can be treated similarly.

For any $\epsilon \in (0,1)$, we condition on the following three events:

$$\begin{aligned}E_1 &= \{|\sqrt{\kappa}/\sigma_n - 1/\sigma_\star| < \epsilon\}, \\ E_2 &= \{\widehat{\boldsymbol{\beta}}^{\mathsf{T}(1)}\Sigma\widehat{\boldsymbol{\beta}}^{(1)} \le c\} \cap \{\widehat{\boldsymbol{\beta}}^{\mathsf{T}(2)}\Sigma\widehat{\boldsymbol{\beta}}^{(2)} \le c\}, \\ E_3 &= \Big\{\max_{j\in[p]}\left|\widehat{\tau}_j^{(1)}/\tau_j - 1\right| < \epsilon\Big\} \cap \Big\{\max_{j\in[p]}\left|\widehat{\tau}_j^{(2)}/\tau_j - 1\right| < \epsilon\Big\}.\end{aligned} \tag{C.79}$$

We note that for large enough n and p, E_2 holds for large enough constant c with high probability by Theorem 4 in Sur and Candès (2018), and E_1, E_3 hold with high probability by Lemma C.1.11 and Lemma C.1.12, respectively.

We have

$$\sum_{j\in S_0}\left(\tau_j\widehat{\beta}_j^{(1)}\right)^2 \le \max_{j\in[p]}\tau_j^2\sum_{j\in[p]}\widehat{\beta}_j^{2(1)} \le \frac{\sigma_{\max}(\Sigma)}{\sigma_{\min}(\Sigma)}\sigma_{\min}(\Sigma)||\widehat{\boldsymbol{\beta}}^{(1)}||_2^2 \le c_1^2 c, \tag{C.80}$$

by Assumption 3.4.2 and the event E_2. It follows that

$$\begin{aligned}\sum_{j\in S_0}\left[\widehat{\tau}_j^{(1)}\widehat{\beta}_j^{(1)}\right]^2 &\le 2\sum_{j\in S_0}\left[(\widehat{\tau}_j^{(1)}-\tau_j)\widehat{\beta}_j^{(1)}\right]^2+2\sum_{j\in S_0}\left[\tau_j\widehat{\beta}_j^{(1)}\right]^2\\ &\le 2\max_{j\in[p]}\left|\widehat{\tau}_j^{(1)}/\tau_j-1\right|^2\sum_{j\in S_0}\left[\tau_j\widehat{\beta}_j^{(1)}\right]^2+2\sum_{j\in S_0}\left[\tau_j\widehat{\beta}_j^{(1)}\right]^2\\ &\le 2(\epsilon^2+1)c_1^2c,\end{aligned} \tag{C.81}$$

by the event E_3. The same upper bounds also hold for $\sum_{j\in S_0}\left[\tau_j\widehat{\beta}_j^{(2)}\right]^2$ and $\sum_{j\in S_0}\left[\widehat{\tau}_j^{(2)}\widehat{\beta}_j^{(2)}\right]^2$, respectively.

We note that

$$\begin{aligned}\left|T_j^{(1)}-\widetilde{T}_j^{(1)}\right| &\le \sqrt{\frac{n}{2}}\left|\widehat{\tau}_j^{(1)}\widehat{\beta}_j^{(1)}\right|\left|\frac{\sqrt{\kappa}}{\sigma_n}-\frac{1}{\sigma_\star}\right|+\sqrt{\frac{n}{2}}\frac{\sqrt{\kappa}}{\sigma_n}\left|\tau_j\widehat{\beta}_j^{(1)}\right|\left|\frac{\widehat{\tau}_j^{(1)}}{\tau_j}-1\right|\\ &\le \sqrt{n/2}\left|\widehat{\tau}_j^{(1)}\widehat{\beta}_j^{(1)}\right|\epsilon+\sqrt{n/2}\left|\tau_j\widehat{\beta}_j^{(1)}\right|\epsilon(\epsilon+1/\sigma_\star).\end{aligned} \tag{C.82}$$

Therefore, by the Cauchy-Schwartz inequality, we have

$$\begin{aligned}\frac{L}{p_0}\sum_{j\in S_0}\left|T_j^{(2)}\right|\left|T_j^{(1)}-\widetilde{T}_j^{(1)}\right| &\le \frac{nL\epsilon(\epsilon+1/\sigma_\star)}{2p_0\sigma_\star}\sum_{j\in S_0}\left|\widehat{\tau}_j^{(2)}\widehat{\beta}_j^{(2)}\right|\left|\tau_j^{(1)}\widehat{\beta}_j^{(1)}\right|\\ &\quad+\frac{nL\epsilon}{2p_0\sigma_\star}\sum_{j\in S_0}\left|\widehat{\tau}_j^{(2)}\widehat{\beta}_j^{(2)}\right|\left|\widehat{\tau}_j^{(1)}\widehat{\beta}_j^{(1)}\right|\\ &\le \frac{nL\epsilon(\epsilon+1/\sigma_\star)}{2p_0\sigma_\star}\left[\sum_{j\in S_0}\left(\widehat{\tau}_j^{(2)}\widehat{\beta}_j^{(2)}\right)^2\right]^{1/2}\left[\sum_{j\in S_0}\left(\tau_j^{(1)}\widehat{\beta}_j^{(1)}\right)^2\right]^{1/2}\\ &\quad+\frac{nL\epsilon}{2p_0\sigma_\star}\left[\sum_{j\in S_0}\left(\widehat{\tau}_j^{(2)}\widehat{\beta}_j^{(2)}\right)^2\right]^{1/2}\left[\sum_{j\in S_0}\left(\widehat{\tau}_j^{(1)}\widehat{\beta}_j^{(1)}\right)^2\right]^{1/2}\\ &\le c'\epsilon,\end{aligned} \tag{C.83}$$

for some constant $c'>0$, since $p/n\to\kappa/2$ and $\limsup p/p_0<+\infty$ by Assumption 3.4.2. This completes the proof of Lemma C.1.13.

Lemma C.1.14. For any Lipschitz continuous function $f : \mathbb{R} \to \mathbb{R}$, we have

$$\frac{1}{p_0}\sum_{j\in S_0}[f(\widetilde{Z}_j^{(1)}\widetilde{Z}_j^{(2)}) - f(Z_j^{(1)}Z_j^{(2)})] \xrightarrow{p} 0. \tag{C.84}$$

Proof of Lemma C.1.14. Denote the Lipschitz coefficient as L. We have

$$\begin{aligned}\left|\frac{1}{p_0}\sum_{j\in S_0}[f(\widetilde{Z}_j^{(1)}\widetilde{Z}_j^{(2)}) - f(Z_j^{(1)}Z_j^{(2)})]\right| &\le \frac{L}{p_0}\sum_{j\in S_0}\left|\widetilde{Z}_j^{(1)}\widetilde{Z}_j^{(2)} - Z_j^{(1)}Z_j^{(2)}\right| \\ &\le \frac{L}{p_0}\left|\frac{\sqrt{p}}{||\boldsymbol{\xi}^{(1)}||_\Sigma}\frac{\sqrt{p}}{||\boldsymbol{\xi}^{(2)}||_\Sigma} - 1\right|\sum_{j\in S_0}\left|Z_j^{(1)}Z_j^{(2)}\right|.\end{aligned} \tag{C.85}$$

We proceed to show that

$$\frac{1}{p_0}\sum_{j\in S_0}\left|Z_j^{(1)}Z_j^{(2)}\right| = O_p(1) \quad \text{and} \quad \frac{||\boldsymbol{\xi}^{(1)}||_\Sigma}{p} \xrightarrow{p} 1. \tag{C.86}$$

The first claim follows by noticing that

$$\mathbb{E}\left[\frac{1}{p_0}\sum_{j\in S_0}\left|Z_j^{(1)}Z_j^{(2)}\right|\right] = \mathbb{E}\left|Z_j^{(1)}Z_j^{(2)}\right| \le \left[\mathbb{E}(Z_j^{(1)})^2\mathbb{E}(Z_j^{(2)})^2\right]^{1/2} = 1. \tag{C.87}$$

For the second claim, by the definition of $\boldsymbol{\xi}^{(1)}$ in Equation (C.62), we have

$$\frac{||\boldsymbol{\xi}^{(1)}||_\Sigma^2}{p} = \frac{||\boldsymbol{W}^{(1)}||_\Sigma^2}{p} - \frac{1}{p}\langle \boldsymbol{W}^{(1)}, \frac{\boldsymbol{\beta}^\star}{\omega_n}\rangle_\Sigma^2. \tag{C.88}$$

For the first term, we have

$$\frac{||\boldsymbol{W}^{(1)}||_\Sigma^2}{p} = \frac{1}{p}\boldsymbol{W}^{(1)\intercal}\Sigma\boldsymbol{W}^{(1)} = \frac{\boldsymbol{Z}^\intercal\boldsymbol{Z}}{p} \xrightarrow{p} 1, \tag{C.89}$$

in which $\boldsymbol{Z} \sim N(0, I_p)$. Besides, since $\langle \boldsymbol{W}^{(1)}, \boldsymbol{\beta}^\star/\omega_n\rangle_\Sigma$ follows a standard normal distribution, the

second term converges to 0 in probability. The proof of Lemma C.1.14 thus completes.

Lemma C.1.15. For any $\epsilon \in (0,1)$, we have as $n, p \to \infty$,

$$\sup_{t\in\mathbb{R}} \mathbb{P}\left(\left|\frac{1}{p_0}\sum_{j\in S_0} 1(M_j > t) - H(t)\right| > \epsilon\right) \longrightarrow 0. \tag{C.90}$$

Proof of Lemma C.1.15. We prove an one-sided result, whereas the other side follows similarly. Given $t \in \mathbb{R}$ and $\epsilon \in (0,1)$, we define the follow Lipschitz continuous function,

$$f_{t,\epsilon}(x) = \begin{cases} 1, & x \in [t, +\infty), \\ 1 + 4(x-t)/\epsilon, & x \in (t-\epsilon/4, t), \\ 0, & x \in (-\infty, t-\epsilon/4]. \end{cases} \tag{C.91}$$

We note that by integrating with respect to the density of the product of two independent normals, we have

$$\mathbb{E}[f_{t,\epsilon}(Z_1Z_2)] - H(t) = \mathbb{E}[f_{t,\epsilon}(Z_1Z_2) - 1(Z_1Z_2 > t)] \le \epsilon/4. \tag{C.92}$$

We thus have the following decomposition:

$$\begin{aligned} \mathbb{P}\left(\frac{1}{p_0}\sum_{j\in S_0} 1(M_j > t) - H(t) > \epsilon\right) &\le \mathbb{P}\left(\frac{1}{p_0}\sum_{j\in S_0} f_{t,\epsilon}(M_j) - \mathbb{E}[f_{t,\epsilon}(Z_1Z_2)] > \epsilon/2\right) \\ &\le I_1 + I_2 + I_3, \end{aligned} \tag{C.93}$$

in which

$$\begin{aligned} I_1 &= \mathbb{P}\left(\frac{1}{p_0}\sum_{j\in S_0}\left[f_{t,\epsilon}(M_j) - f_{t,\epsilon}(\widetilde{M}_j)\right] > \epsilon\right), \\ I_2 &= \mathbb{P}\left(\frac{1}{p_0}\sum_{j\in S_0}\left[f_{t,\epsilon}(\widetilde{Z}_j^{(1)}\widetilde{Z}_j^{(2)}) - f_{t,\epsilon}(Z_j^{(1)}Z_j^{(2)})\right] > \epsilon/8\right), \\ I_3 &= \mathbb{P}\left(\frac{1}{p_0}\sum_{j\in S_0} f_{t,\epsilon}(Z_j^{(1)}Z_j^{(2)}) - \mathbb{E}[f_{t,\epsilon}(Z_1Z_2)] > \epsilon/8\right). \end{aligned} \tag{C.94}$$

By Lemma C.1.13, we have $\sup_t I_1 \to 0$ because the Lipschitz coefficient of $f_{t,\epsilon}(x)$ is $4/\epsilon$ independent to t. Similarly, by Lemma C.1.14, we have $\sup_t I_2 \to 0$. For $\sup_t I_3$, we follow the proof of Corollary A.2 in Zhao et al. (2020) to show that it converges to 0. Since $f_{t,\epsilon}(x)$ is continuous and in the range of $[0,1]$, we partition the unit interval $-\infty = u_0 < u_1 < \cdots < u_m = +\infty$ so that for any $x \in [u_{i-1}, u_i)$, we have

$$|f_{t,\epsilon}(x) - f_{t,\epsilon}(u_{i-1})| \leq \epsilon/32. \tag{C.95}$$

We define

$$\widehat{f}_{t,\epsilon}(x) = \sum_{i=1}^{m} f_{t,\epsilon}(u_{i-1})\, 1(x \in [u_{i-1}, u_i)). \tag{C.96}$$

Then we have for any $x \in \mathbb{R}$,

$$\left| f_{t,\epsilon}(x) - \widehat{f}_{t,\epsilon}(x) \right| \leq \epsilon/32. \tag{C.97}$$

With the following decomposition,

$$\begin{aligned} &\frac{1}{p_0} \sum_{j \in S_0} f_{t,\epsilon}(Z_j^{(1)} Z_j^{(2)}) - \mathbb{E}[f_{t,\epsilon}(Z_1 Z_2)] \\ &\qquad \leq \frac{1}{p_0} \sum_{j \in S_0} \widehat{f}_{t,\epsilon}(Z_j^{(1)} Z_j^{(2)}) - \mathbb{E}\left[\widehat{f}_{t,\epsilon}(Z_1 Z_2)\right] + \epsilon/16, \end{aligned} \tag{C.98}$$

we have

$$\begin{aligned} \sup_{t \in \mathbb{R}} I_3 &\leq \sup_{t \in \mathbb{R}} \mathbb{P}\left(\frac{1}{p_0} \sum_{j \in S_0} \widehat{f}_{t,\epsilon}(Z_j^{(1)} Z_j^{(2)}) - \mathbb{E}[\widehat{f}_{t,\epsilon}(Z_1 Z_2)] > \epsilon/16 \right) \\ &\leq \sup_{t \in \mathbb{R}} \sum_{i=1}^{m-1} \mathbb{P}\left(\frac{1}{p_0} \sum_{j \in S_0} 1(Z_j^{(1)} Z_j^{(2)} \in [u_i, u_{i+1})) - \mathbb{P}(Z_1 Z_2 \in [u_i, u_{i+1})) > \epsilon/16 \right) \\ &\longrightarrow 0 \end{aligned} \tag{C.99}$$

by Lemma C.1.9. This completes the proof of Lemma C.1.15.

Proof of Proposition 3.4.2

The proof follows similarly as the proof of Proposition 3.4.1 thus is omitted.

C.1.6 Proof of Proposition 3.4.3 and 3.4.4

Technical lemmas

Lemma C.1.16. Under Assumption 3.4.3, for any $\epsilon > 0$, there exists a constant $C_\epsilon > 0$, such that $\mathbb{P}(\cap_{i=1}^4 E_i^\epsilon) \geq 1 - \epsilon$, in which the events $E_1^\epsilon, E_2^\epsilon, E_3^\epsilon, E_4^\epsilon$ are defined as follows:

$$\begin{aligned} E_1^\epsilon &= \left\{ \max_{j\in[p]} ||\widehat{\boldsymbol{\gamma}}_j - \boldsymbol{\gamma}_j||_1 \leq C_\epsilon s\sqrt{\log p/n} \right\}, \\ E_2^\epsilon &= \left\{ \max_{j\in[p]} ||\widehat{\boldsymbol{\gamma}}_j - \boldsymbol{\gamma}_j||_2 \leq C_\epsilon \sqrt{s\log p/n} \right\}, \\ E_3^\epsilon &= \left\{ \frac{1}{n}||\boldsymbol{X}(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star)||_2^2 \leq C_\epsilon s^\star \log p/n \right\}, \\ E_4^\epsilon &= \left\{ ||\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star||_1 \leq C_\epsilon s^\star \sqrt{\log p/n} \right\}. \end{aligned} \tag{C.100}$$

Proof of Lemma C.1.16. Lemma C.1.16 follows from standard arguments in Bickel et al. (2009), Raskutti et al. (2010), Bühlmann and Van De Geer (2011), Van de Geer et al. (2014).

Lemma C.1.17. Under Assumption 3.4.3, we have

$$\begin{aligned} &\max_{j\in[p]} \left|\widehat{\tau}_j^2 - \tau_j^2\right| = O_p(\sqrt{s\log p/n}), \\ &\max_{j\in[p]} \left|1/\widehat{\tau}_j^2 - 1/\tau_j^2\right| = O_p(\sqrt{s\log p/n}). \end{aligned} \tag{C.101}$$

Proof of Lemma C.1.17. The proof follows similarly as the proof of Theorem 3.2 in Van de Geer et al. (2014), combined with some enriched arguments that we detail below. For any $\epsilon > 0$, we condition on the high probability event $\{\cap_{i=1}^4 E_i^\epsilon\}$. By Assumption 3.4.3, we assume that p is large enough so

that

$$\max_{i\in[n]} |\boldsymbol{x}_i^\intercal(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star)| \leq ||\boldsymbol{x}_i||_\infty ||\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star||_1 \leq C_2 C_\epsilon s^\star \sqrt{\log p/n} \leq \delta.$$

Thus, $\boldsymbol{x}_i^\intercal \widehat{\boldsymbol{\beta}}$ lies in the δ-neighborhood of the $\boldsymbol{x}_i^\intercal \boldsymbol{\beta}^\star$. Since $\boldsymbol{X}_{\widehat{\boldsymbol{\beta}}} = W_{\widehat{\boldsymbol{\beta}}} W_{\boldsymbol{\beta}^\star}^{-1} X_{\boldsymbol{\beta}^\star}$, we have the following decomposition,

$$\widehat{\tau}_j^2 - \tau_j^2 = \frac{1}{n} \boldsymbol{X}_{\widehat{\boldsymbol{\beta}},j}^\top \left(\boldsymbol{X}_{\widehat{\boldsymbol{\beta}},j} - \boldsymbol{X}_{\widehat{\boldsymbol{\beta}},-j} \widehat{\boldsymbol{\gamma}}_j \right) - \tau_j^2 := I_1 + I_2, \tag{C.102}$$

in which

$$\begin{aligned} I_1 &= \frac{1}{n} \boldsymbol{X}_{\boldsymbol{\beta}^\star,j}^\top \left(\boldsymbol{X}_{\boldsymbol{\beta}^\star,j} - \boldsymbol{X}_{\boldsymbol{\beta}^\star,-j} \widehat{\boldsymbol{\gamma}}_j \right) - \tau_j^2, \\ I_2 &= \frac{1}{n} \boldsymbol{X}_{\boldsymbol{\beta}^\star,j}^\top \left(W_{\widehat{\boldsymbol{\beta}}}^2 W_{\boldsymbol{\beta}^\star}^{-2} - I \right) \left(\boldsymbol{X}_{\boldsymbol{\beta}^\star,j} - \boldsymbol{X}_{\boldsymbol{\beta}^\star,-j} \widehat{\boldsymbol{\gamma}}_j \right). \end{aligned} \tag{C.103}$$

We proceed to upper bound I_1 and I_2.

For I_1, we have the following further decomposition $I_1 = I_{11} + I_{12} + I_{13} + I_{14}$, using the fact that $\boldsymbol{X}_{\boldsymbol{\beta}^\star,j} = \boldsymbol{X}_{\boldsymbol{\beta}^\star,-j} \boldsymbol{\gamma}_j + \boldsymbol{\eta}_j$, in which

$$\begin{aligned} I_{11} &= \frac{1}{n} \boldsymbol{\eta}_j^\top \boldsymbol{\eta}_j - \tau_j^2, \quad I_{12} = \frac{1}{n} \boldsymbol{\eta}_j^\top \boldsymbol{X}_{\boldsymbol{\beta}^\star,-j} (\boldsymbol{\gamma}_j - \widehat{\boldsymbol{\gamma}}_j), \\ I_{13} &= \frac{1}{n} \boldsymbol{\gamma}_j^\top \boldsymbol{X}_{\widehat{\boldsymbol{\beta}},-j}^\top \left(\boldsymbol{X}_{\widehat{\boldsymbol{\beta}},j} - \boldsymbol{X}_{\widehat{\boldsymbol{\beta}},-j} \widehat{\boldsymbol{\gamma}}_j \right), \\ I_{14} &= \frac{1}{n} \boldsymbol{\gamma}_j^\top \boldsymbol{X}_{\boldsymbol{\beta}^\star,-j}^\top \left(I - W_{\widehat{\boldsymbol{\beta}}}^2 W_{\boldsymbol{\beta}^\star}^{-2} \right) \left(\boldsymbol{X}_{\boldsymbol{\beta}^\star,j} - \boldsymbol{X}_{\boldsymbol{\beta}^\star,-j} \widehat{\boldsymbol{\gamma}}_j \right). \end{aligned} \tag{C.104}$$

For I_{11}, by Assumption 3.4.3, we have

$$||\boldsymbol{\eta}_j||_\infty \leq ||\boldsymbol{X}_{\boldsymbol{\beta}^\star,j}||_\infty + ||\boldsymbol{X}_{\boldsymbol{\beta}^\star,-j} \boldsymbol{\gamma}_j||_\infty \leq 2C_2. \tag{C.105}$$

Since $\mathbb{E}[\boldsymbol{\eta}_j^\top \boldsymbol{\eta}_j / n] = \tau_j^2$, we have

$$\max_{j\in[p]} I_{11} = O_p(\sqrt{\log p/n}),$$

using the union bound and the Hoeffding's inequality.

For I_{12}, since $\mathbb{E}[\boldsymbol{\eta}_j^\top \boldsymbol{X}_{\boldsymbol{\beta}^\star,-j}] = 0$, we have

$$\max_{j\in[p]} ||\boldsymbol{\eta}_j^\top \boldsymbol{X}_{\boldsymbol{\beta}^\star,-j}/n||_\infty = O_p(\sqrt{\log p/n}),$$

by the union bound and the Hoeffding's inequality. It follows that

$$\max_{j\in[p]} I_{12} \leq \max_{j\in[p]} ||\boldsymbol{\eta}_j^\top \boldsymbol{X}_{\widehat{\boldsymbol{\beta}},-j}/n||_\infty \max_{j\in[p]} ||\widehat{\boldsymbol{\gamma}}_j - \boldsymbol{\gamma}_j||_1 = O_p(s\log p/n). \tag{C.106}$$

For I_{13}, since $\boldsymbol{\gamma}_j$ is $s-$sparse, by Assumption 3.4.3, we have

$$\max_{j\in[p]} ||\boldsymbol{\gamma}_j||_1 \leq \sqrt{s}\max_{j\in[p]} ||\boldsymbol{\gamma}_j||_2 = \sqrt{s}\max_{j\in[p]} [\Sigma_{j,-j}\Sigma_{-j,-j}^{-2}\Sigma_{-j,j}]^{1/2} \leq \sqrt{s}C_2^2/C_3. \tag{C.107}$$

In addition, by the KKT condition of the j^{th} nodewise Lasso regression, we have

$$\max_{j\in[p]} ||\boldsymbol{X}_{\widehat{\boldsymbol{\beta}},-j}^\top \left(\boldsymbol{X}_{\widehat{\boldsymbol{\beta}},j} - \boldsymbol{X}_{\widehat{\boldsymbol{\beta}},-j}\widehat{\boldsymbol{\gamma}}_j\right)/n||_\infty \leq \lambda_j. \tag{C.108}$$

It follows that

$$\begin{aligned}\max_{j\in[p]} I_{13} &\leq \max_{j\in[p]} ||\boldsymbol{\gamma}_j||_1 \max_{j\in[p]} ||\boldsymbol{X}_{\widehat{\boldsymbol{\beta}},-j}^\top \left(\boldsymbol{X}_{\widehat{\boldsymbol{\beta}},j} - \boldsymbol{X}_{\widehat{\boldsymbol{\beta}},-j}\widehat{\boldsymbol{\gamma}}_j\right)/n||_\infty \\ &= O_p(\sqrt{s\log p/n})\end{aligned} \tag{C.109}$$

For I_{14}, we first notice that

$$\begin{aligned}||\boldsymbol{X}_{\boldsymbol{\beta}^\star,j} - \boldsymbol{X}_{\boldsymbol{\beta}^\star,-j}\widehat{\boldsymbol{\gamma}}_j||_\infty &\leq ||\boldsymbol{\eta}_j||_\infty + ||\boldsymbol{X}_{\boldsymbol{\beta}^\star,-j}(\boldsymbol{\gamma}_j - \widehat{\boldsymbol{\gamma}}_j)||_\infty \\ &\leq 2C_2 + C_2||\boldsymbol{\gamma}_j - \widehat{\boldsymbol{\gamma}}_j||_1 \leq 3C_2\end{aligned} \tag{C.110}$$

by Assumption 3.4.3 and Equation (C.105). By Assumption 3.4.3, it follows that

$$\begin{aligned}\max_{j\in[p]} I_{14} &\leq \frac{3C_2^2}{n}\sum_{i=1}^{n}\left|\frac{\ddot{\ell}(y_i, \boldsymbol{x}_i^\intercal\widehat{\boldsymbol{\beta}}) - \ddot{\ell}(y_i, \boldsymbol{x}_i^\intercal\boldsymbol{\beta}^\star)}{\ddot{\ell}(y_i, \boldsymbol{x}_i^\intercal\boldsymbol{\beta}^\star)}\right| \\ &\leq 3C_1C_2^2||\boldsymbol{X}(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star)||_2/\sqrt{n}) \\ &= O_p(\sqrt{s^\star \log p/n}),\end{aligned} \tag{C.111}$$

in which the second inequality follows from Cauchy-Schwarz inequality. Combining the upper bound on $\max_{j\in[p]} I_{11}$, $\max_{j\in[p]} I_{12}$, $\max_{j\in[p]} I_{13}$ and $\max_{j\in[p]} I_{14}$, we have shown that $\max_{j\in[p]} I_1 = O_p(\sqrt{s\log p/n})$.

Finally, $\max_{j\in[p]} I_2$ can be upper bounded similarly as $\max_{j\in[p]} I_{14}$. This completes the proof of the claim in Lemma C.1.17. The second claim follows by noticing that $1/\tau_j^2$ is uniformly upper bounded because

$$1/\tau_j^2 = \Omega_{j,j} \leq \sigma_{\max}(\Omega) = 1/\sigma_{\min}(\Sigma) \leq C_3. \tag{C.112}$$

This completes the proof of Lemma C.1.17

Lemma C.1.18. Under Assumption 3.4.3, we have

$$\begin{aligned}&\max_{j\in[p]} ||\widehat{\Omega}_{j,\cdot} - \Omega_{j,\cdot}||_1 = O_p(s\sqrt{\log p/n}), \\ &\max_{j\in[p]} ||\widehat{\Omega}_{j,\cdot} - \Omega_{j,\cdot}||_2 = O_p(\sqrt{s\log p/n}), \\ &\max_{j\in[p]} |\widehat{\Omega}_{j,\cdot}\Sigma\widehat{\Omega}_{j,\cdot}^\top - \Omega_{j,j}| = O_p(\sqrt{s\log p/n}).\end{aligned} \tag{C.113}$$

Proof of Lemma C.1.18. By Lemma C.1.16 and C.1.17, for any $\epsilon > 0$, there exists a constant $C_\epsilon > 0$,

such that $\mathbb{P}(\cap_{i=1}^6 E_i^\epsilon) \geq 1-\epsilon$, in which E_i^ϵ for $i \in [4]$ are specified in Lemma C.1.16, and

$$\begin{aligned} E_5^\epsilon &= \left\{ \max_{j\in[p]} \left|\widehat{\tau}_j^2 - \tau_j^2\right| \leq C_\epsilon \sqrt{s\log p/n} \right\}, \\ E_6^\epsilon &= \left\{ \max_{j\in[p]} \left|1/\widehat{\tau}_j^2 - 1/\tau_j^2\right| \leq C_\epsilon \sqrt{s\log p/n} \right\}. \end{aligned} \tag{C.114}$$

In the following, we condition on this high probability event $\cap_{i=1}^6 E_i^\epsilon$. By Assumption 3.4.3, we assume n is large enough so that $C_\epsilon\sqrt{s\log p/n} \leq C_3$.

We first note that by Equation (C.112),

$$1/\widehat{\tau}_j^2 \leq |1/\widehat{\tau}_j^2 - 1/\tau_j^2| + 1/\tau_j^2 \leq 2C_3.$$

Thus, by Equation (C.107), we have

$$\begin{aligned} \max_{j\in[p]} ||\widehat{\Omega}_{j,\cdot} - \Omega_{j,\cdot}||_1 &= \max_{j\in[p]} ||\widehat{C}_{j,\cdot}/\widehat{\tau}_j^2 - C_{j,\cdot}/\tau_j^2||_1 \\ &\leq \max_{j\in[p]} ||\widehat{\boldsymbol{\gamma}}_j - \boldsymbol{\gamma}_j||_1 \max_{j\in[p]} 1/\widehat{\tau}_j^2 + \max_{j\in[p]} ||\boldsymbol{\gamma}_j||_1 \max_{j\in[p]} \left|1/\widehat{\tau}_j^2 - 1/\tau_j^2\right| \\ &\leq 2C_3C_\epsilon s\sqrt{\log p/n} + C_2^2/C_3C_\epsilon\sqrt{s}\sqrt{s\log p/n} \\ &= O_p(s\sqrt{\log p/n}). \end{aligned} \tag{C.115}$$

Similarly, we have

$$\begin{aligned} \max_{j\in[p]} ||\widehat{\Omega}_{j,\cdot} - \Omega_{j,\cdot}||_2 &= \max_{j\in[p]} ||\widehat{C}_{j,\cdot}/\widehat{\tau}_j^2 - C_{j,\cdot}/\tau_j^2||_2 \\ &\leq \max_{j\in[p]} ||\widehat{\boldsymbol{\gamma}}_j - \boldsymbol{\gamma}_j||_2 \max_{j\in[p]} 1/\widehat{\tau}_j^2 + \max_{j\in[p]} ||\boldsymbol{\gamma}_j||_2 \max_{j\in[p]} \left|1/\widehat{\tau}_j^2 - 1/\tau_j^2\right| \\ &\leq 2C_3C_\epsilon\sqrt{s\log p/n} + C_2^2/C_3C_\epsilon\sqrt{s\log p/n} \\ &= O_p(\sqrt{s\log p/n}). \end{aligned} \tag{C.116}$$

For the last claim in Lemma C.1.18, we employ the following decomposition.

$$\begin{aligned}\widehat{\Omega}_{j,\cdot}\Sigma\widehat{\Omega}_{j,\cdot}^{\top}-\Omega_{j,j} &= (\widehat{\Omega}_{j,\cdot}-\Omega_{j,\cdot})\Sigma(\widehat{\Omega}_{j,\cdot}-\Omega_{j,\cdot})^{\top}+2\Omega_{j,\cdot}\Sigma(\widehat{\Omega}_{j,\cdot}-\Omega_{j,\cdot})^{\top}\\ &:= I_1+I_2.\end{aligned} \tag{C.117}$$

For I_1, by Assumption 3.4.3, we have

$$\max_{j\in[p]} I_1 \le \sigma_{\max}(\Sigma)\max_{j\in[p]}||\widehat{\Omega}_{j,\cdot}-\Omega_{j,\cdot}||_2^2 = O_p(s\log p/n). \tag{C.118}$$

For I_2, we have

$$\max_{j\in[p]} I_2 = 2\max_{j\in[p]} e_j^{\top}(\widehat{\Omega}_{j,\cdot}-\Omega_{j,\cdot})^{\top} = 2\max_{j\in[p]}\left|1/\widehat{\tau}_j^2-1/\tau_j^2\right| = O_p(\sqrt{s\log p/n}). \tag{C.119}$$

This completes the proof of Lemma C.1.18.

Lemma C.1.19. Under Assumption 3.4.3, we have

$$\begin{aligned}&\max_{j\in[p]}|\widehat{\sigma}_j-\sigma_j| = O_p(s\sqrt{\log p/n}),\\ &\max_{j\in[p]}|1/\widehat{\sigma}_j-1/\sigma_j| = O_p(s\sqrt{\log p/n}).\end{aligned} \tag{C.120}$$

Proof of Lemma C.1.19. By Assumption 3.4.3,

$$\sigma^2=\Omega_{j,j}\ge\sigma_{\min}(\Omega)\ge 1/C_3>0.$$

Therefore, we only need to consider the first claim in the lemma. In addition, since

$$\max_{j\in[p]}|\widehat{\sigma}_j-\sigma_j|\le\max_{j\in[p]}|\widehat{\sigma}_j^2-\sigma_j^2|/\sigma_j\le\sqrt{C_3}\max_{j\in[p]}|\widehat{\sigma}_j^2-\sigma_j^2|, \tag{C.121}$$

it is sufficient to show that

$$\max_{j\in[p]} |\widehat{\sigma}_j^2 - \sigma_j^2| = O_p(s\sqrt{\log p/n}).$$

We note that

$$\max_{j\in[p]} |\widehat{\sigma}_j^2 - \sigma_j^2| \leq \max_{j\in[p]} |\widehat{\Omega}_{j,\cdot}\Sigma\widehat{\Omega}_{j,\cdot}^\top - \Omega_{j,j}| + \max_{j\in[p]} |\widehat{\Omega}_{j,\cdot}\widehat{\Sigma}\widehat{\Omega}_{j,\cdot}^\top - \widehat{\Omega}_{j,\cdot}\Sigma\widehat{\Omega}_{j,\cdot}^\top|. \tag{C.122}$$

By Lemma C.1.18, the first term is $O_p(\sqrt{s\log p/n})$. Using the same arguments as in the proof of Theorem 3.1 in Van de Geer et al. (2014) (page 1198), we can show that the second term also scales as $O_p(s\sqrt{\log p/n})$. This completes the proof of Lemma C.1.19.

Proof of Proposition 3.4.3

We first note that for $j \in [p]$, the bias term Δ_j can be decomposed into the following three terms,

$$\begin{aligned}
R_{1,j} &= \frac{\sqrt{n}}{\sigma_j}\Omega_{j,\cdot}P_n\dot{\ell}_{\boldsymbol{\beta}^\star} - \frac{\sqrt{n}}{\widehat{\sigma}_j}\widehat{\Omega}_{j,\cdot}P_n\dot{\ell}_{\boldsymbol{\beta}^\star}, \\
R_{2,j} &= -\frac{\sqrt{n}}{\widehat{\sigma}_j}\left(\widehat{\Omega}_{j,\cdot}P_n\ddot{\ell}_{\widehat{\boldsymbol{\beta}}} - \boldsymbol{e}_j^\top\right)(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star), \\
R_{3,j} &= -\frac{\sqrt{n}}{\widehat{\sigma}_j}\widehat{\Omega}_{j,\cdot}P_n(\dot{\ell}_{\widehat{\boldsymbol{\beta}}} - \dot{\ell}_{\boldsymbol{\beta}^\star}) + \frac{1}{\sqrt{n}\widehat{\sigma}_j}\widehat{\Omega}_{j,\cdot}\boldsymbol{X}^\top W_{\widehat{\boldsymbol{\beta}}}^2\boldsymbol{X}(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star).
\end{aligned} \tag{C.123}$$

We proceed to bound each term. For any $\epsilon > 0$, there exists a constant $C_\epsilon > 0$ such that $\mathbb{P}(\cap_{i=1}^{10} E_i^\epsilon) \geq 1-\epsilon$, in which E_i^ϵ for $i \in [4]$ are defined in Lemma C.1.16, E_5^ϵ and E_6^ϵ are defined in Lemma C.1.18,

and $E_7^\epsilon, E_8^\epsilon, E_9^\epsilon, E_{10}^\epsilon$ are defined as follows,

$$\begin{aligned} E_7^\epsilon &= \left\{ \max_{j\in[p]} ||\widehat{\Omega}_{j,\cdot} - \Omega_{j,\cdot}||_1 \leq C_\epsilon s\sqrt{\log p/n} \right\}, \\ E_8^\epsilon &= \left\{ \max_{j\in[p]} ||\widehat{\Omega}_{j,\cdot} - \Omega_{j,\cdot}||_2 \leq C_\epsilon \sqrt{s\log p/n} \right\}, \\ E_9^\epsilon &= \left\{ \max_{j\in[p]} |\widehat{\sigma}_j - \sigma_j| \leq C_\epsilon s\sqrt{\log p/n} \right\}, \\ E_{10}^\epsilon &= \left\{ \max_{j\in[p]} |1/\widehat{\sigma}_j - 1/\sigma_j| \leq C_\epsilon s\sqrt{\log p/n} \right\}. \end{aligned} \tag{C.124}$$

In the following, we condition on this high probability event $\cap_{i=1}^{10} E_i^\epsilon$. By Assumption 3.4.3, we further assume that p is large enough so that $C_\epsilon s\sqrt{\log p/n} \vee C_\epsilon\sqrt{s\log p/n} \leq 1$ and

$$\max_{i\in[n]} |\boldsymbol{x}_i^\intercal(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star)| \leq ||\boldsymbol{x}_i||_\infty ||\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star||_1 \leq C_2 C_\epsilon s^\star\sqrt{\log p/n} \leq \delta.$$

For $R_{1,j}$, we have

$$\begin{aligned} \max_{j\in[p]} R_{1,j} &\leq \max_{j\in[p]} \left| \frac{\sqrt{n}}{\sigma_j} \left(\Omega_{j,\cdot} - \widehat{\Omega}_{j,\cdot}\right) P_n\dot{\ell}_{\boldsymbol{\beta}^\star} \right| + \max_{j\in[p]} \left| \sqrt{n}\widehat{\Omega}_{j,\cdot} P_n\dot{\ell}_{\boldsymbol{\beta}^\star} \left(\frac{1}{\widehat{\sigma}_j} - \frac{1}{\sigma_j}\right) \right| \\ &:= I_1 + I_2. \end{aligned} \tag{C.125}$$

For I_1, since $\mathbb{E}[P_n\dot{\ell}_{\boldsymbol{\beta}^\star}] = 0$ and $||\dot{\ell}_{\boldsymbol{\beta}^\star}||_\infty \leq C_1C_2$ by Assumption 3.4.3, we have

$$||P_n\dot{\ell}_{\boldsymbol{\beta}^\star}||_\infty = O_p(\sqrt{\log p/n})$$

using the union bound and the Hoeffding's inequality. Since $\sigma^2 \geq 1/C_3$ (see the proof of Lemma

C.1.19), it follows that

$$\begin{aligned} I_1 &\leq \sqrt{C_3}\sqrt{n}||P_n\dot{\ell}_{\boldsymbol{\beta}^\star}||_\infty \max_{j\in[p]}||\widehat{\Omega}_{j,\cdot} - \Omega_{j,\cdot}||_1 \\ &= \sqrt{n}O_p(\sqrt{\log p/n})O_p(s\sqrt{\log p/n}) \\ &= o_p(1) \end{aligned} \tag{C.126}$$

by Assumption 3.4.3. For I_2, using the same argument, we have

$$\begin{aligned} I_2 &\leq \sqrt{n}\max_{j\in[p]}||\widehat{\Omega}_j||_\infty||P_n\dot{\ell}_{\boldsymbol{\beta}^\star}||_\infty \max_{j\in[p]}\left|\frac{1}{\widehat{\sigma}_j} - \frac{1}{\sigma_j}\right| \\ &\leq \sqrt{n}\left(||\widehat{\Omega} - \Omega||_\infty + ||\Omega||_\infty\right)O_p(\sqrt{\log p/n})O_p(s\sqrt{\log p/n}) \\ &= o_p(1). \end{aligned} \tag{C.127}$$

For $R_{2,j}$, using the KKT condition of the nodewise Lasso regression, we have

$$\left|\left|\widehat{\Omega}_{j,\cdot}P_n\ddot{\ell}_{\widehat{\boldsymbol{\beta}}} - \boldsymbol{e}_j^\top\right|\right|_\infty \leq \lambda_j/\widehat{\tau}_j^2. \tag{C.128}$$

Since

$$\begin{aligned} \max_{j\in[p]} 1/\widehat{\tau}_j^2 &\leq \max_{j\in[p]}|1/\widehat{\tau}_j^2 - 1/\tau_j^2| + \max_{j\in[p]} 1/\tau_j^2 \\ &\leq C_\epsilon\sqrt{s\log p/n} + \max_{j\in[p]} 1/\Omega_{j,j} \\ &\leq 1 + C_3 \end{aligned} \tag{C.129}$$

and

$$\begin{aligned} \max_{j\in[p]} 1/\widehat{\sigma}_j &\leq \max_{j\in[p]}|1/\widehat{\sigma}_j - 1/\sigma_j| + \max_{j\in[p]} 1/\sigma_j \\ &\leq C_\epsilon s\sqrt{\log p/n} + 1/\sigma_{\min}(\Sigma) \\ &\leq 1 + \sqrt{C_3} \end{aligned} \tag{C.130}$$

by Assumption 3.4.3, we have

$$R_{2,j} \leq (1+C_3)\sqrt{n}\left|\left|\widehat{\Omega}_{j,\cdot}P_n\ddot{\ell}_{\widehat{\boldsymbol{\beta}}} - \boldsymbol{e}_j^\top\right|\right|_\infty ||\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star||_1 = o_p(1) \tag{C.131}$$

by Assumption 3.4.3.

For $R_{3,j}$, the first term is $o_p(1)$ following the proof of Theorem 3.1 in Van de Geer et al. (2014). For the second term, we note that

$$\begin{aligned}
||\boldsymbol{X}\widehat{\Omega}_{j,\cdot}^\top||_\infty &\leq ||\boldsymbol{X}\Omega_{j,\cdot}^\top||_\infty + ||\boldsymbol{X}(\widehat{\Omega}_{j,\cdot} - \Omega_{j,\cdot})^\top||_\infty \\
&\leq ||\boldsymbol{X}||_\infty + ||\boldsymbol{X}_{-j}\boldsymbol{\gamma}_j||_\infty + ||\boldsymbol{X}||_\infty||(\widehat{\Omega}_{j,\cdot} - \Omega_{j,\cdot})^\top||_1 \\
&\leq ||\boldsymbol{X}||_\infty + ||\boldsymbol{X}_{\boldsymbol{\beta}^\star,-j}\boldsymbol{\gamma}_j||_\infty / \inf_{(\boldsymbol{x},y)} |\ddot{\ell}(y, \boldsymbol{x}^\intercal\boldsymbol{\beta}^\star)| + ||\boldsymbol{X}||_\infty||(\widehat{\Omega}_{j,\cdot} - \Omega_{j,\cdot})^\top||_1 \\
&\leq C_2 + C_2/C_1^2 + C_2
\end{aligned} \tag{C.132}$$

by Assumption 3.4.3. By the mean value Theorem and Assumption 3.4.3, it follows that

$$R_{3,j} \leq \frac{1}{\sqrt{n}\widehat{\sigma}_j}||\boldsymbol{X}\widehat{\Omega}_{j,\cdot}^\top||_\infty||\boldsymbol{X}(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star)||_2^2 = o_p(1) \tag{C.133}$$

by Assumption 3.4.3. This completes the proof of Proposition 3.4.3.

Proof of Proposition 3.4.4

The proof of Proposition 3.4.4 essentially follows the same as the proof of Proposition 3.4.1. The only change is that $Z_j = -\sqrt{n}\Omega_{j,\cdot}P_n\dot{\ell}_{\boldsymbol{\beta}^\star}/\sigma_j$ is not exactly normal, but asymptotically normal. The discrepancy between the joint law of $(Z_1, \cdots, Z_p)$ and the corresponding multivariate normal distribution (with the covariance matrix Σ) can be quantified using the Berry-Esseen Theorem, which is in the order of $o(1/\sqrt{n})$. Therefore, we can establish the GLM version of Lemma C.1.5 similarly.

C.1.7 Proof of Proposition 3.4.5

Denote $s = \max_{1\le j\le p} |ne_j|$. We first show that the restricted eigenvalue condition in Bickel et al. (2009), page 1710, holds with probability approaching 1 in the regime $\log p/n_1^{1-\xi/2} \to 0$. For the jth nodewise egde selection, for any $J_0 \subseteq [1:p]\backslash\{j\}$ with $|J_0| \le s$, and any $\boldsymbol{v} \ne 0$ satisfying $||\boldsymbol{v}_{J_0^c}||_1 \le ||\boldsymbol{v}_{J_0}||_1$, we have $||\boldsymbol{v}||_1 \le 2||\boldsymbol{v}_{J_0}||_1 \le 2\sqrt{s}||\boldsymbol{v}||_2$. By Theorem 1 in Raskutti et al. (2010) and Cauchy interlacing theorem, with high probability, we have:

$$\frac{||\boldsymbol{X}_{-j}^{(1)}\boldsymbol{v}||_2}{\sqrt{n_1}} \ge \left(\frac{1}{4}\lambda_{\min}(\Sigma) - 18\max_{1\le j\le p}\sigma_{jj}\sqrt{\frac{s\log p}{n_1}}\right)||\boldsymbol{v}||_2. \tag{C.134}$$

Under Assumption 3.4.4, we have

$$\max_{1\le j\le p}\sigma_{jj}\sqrt{\frac{s\log p}{n_1}} \to 0, \tag{C.135}$$

thus $\kappa(s,1) \gtrsim \sqrt{c_0}$ with probability approaching 1. Now we apply the high probability ℓ_1-bound (Equation (7.4) in Bickel et al. (2009)), which is asymptotically no larger than $(\log p/n_1^{1-\xi})^{1/2}$. Therefore by Assumption 3.4.4(c), we conclude that the screening property, thus the symmetric assumption, will hold with probability approaching 1 in the regime $\log p/n_1^{1-\xi/2} \to 0$.

C.1.8 Proof of Proposition 3.4.6

For the ease of presentation, we introduce the following notations. For $j \in [p]$ and $t \in \mathbb{R}$, denote

$$\begin{aligned}
\widehat{G}_{p,j}^0(t) &= \frac{1}{|ne_j^c|}\sum_{i\in ne_j^c} 1(M_{ji} > t), \quad \widehat{V}_{p,j}^0(t) = \frac{1}{|ne_j^c|}\sum_{i\in ne_j^c} 1(M_{ji} < -t), \\
\widehat{G}_{p,j}^1(t) &= \frac{1}{|ne_j|}\sum_{i\in ne_j} 1(M_{ji} > t), \quad G_{p,j}^0(t) = \frac{1}{|ne_j^c|}\sum_{i\in ne_j^c} \mathbb{P}(M_{ji} > t).
\end{aligned} \tag{C.136}$$

Let $\pi^0_{p,j} = |ne^c_j|/\sum_{j=1}^p |ne^c_j|$, $\pi^1_{p,j} = |ne_j|/\sum_{j=1}^p |ne_j|$, and $r_{p,j} = \sum_{j=1}^p |ne_j|/\sum_{j=1}^p |ne^c_j|$. In addition, denote

$$\begin{aligned}
\mathrm{FDP}_p(t_1,\cdots,t_p) &= \frac{\sum_{j=1}^p \pi^0_{p,j}\widehat{G}^0_{p,j}(t_j)}{\sum_{j=1}^p \pi^0_{p,j}\widehat{G}^0_{p,j}(t_j) + r_{p,j}\sum_{j=1}^p \pi^1_{p,j}\widehat{G}^1_{p,j}(t_j)},\\
\mathrm{FDP}^\dagger_p(t_1,\cdots,t_p) &= \frac{\sum_{j=1}^p \pi^0_{p,j}\widehat{V}^0_{p,j}(t_j)}{\sum_{j=1}^p \pi^0_{p,j}\widehat{G}^0_{p,j}(t_j) + r_{p,j}\sum_{j=1}^p \pi^1_{p,j}\widehat{G}^1_{p,j}(t_j)},\\
\overline{\mathrm{FDP}}_p(t_1,\cdots,t_p) &= \frac{\sum_{j=1}^p \pi^0_{p,j}G^0_{p,j}(t_j)}{\sum_{j=1}^p \pi^0_{p,j}G^0_{p,j}(t_j) + r_{p,j}\sum_{j=1}^p \pi^1_{p,j}\widehat{G}^1_{p,j}(t_j)}.
\end{aligned} \tag{C.137}$$

Lemma C.1.20. Under Assumption 3.4.5, as $p \to \infty$, we have

$$\begin{aligned}
\sup_{t_1,\cdots,t_p}\left|\sum_{j=1}^p \pi^0_{p,j}\left(\widehat{G}^0_{p,j}(t_j) - G^0_{p,j}(t_j)\right)\right| &\longrightarrow 0 \quad \text{in probability},\\
\sup_{t_1,\cdots,t_p}\left|\sum_{j=1}^p \pi^0_{p,j}\left(\widehat{V}^0_{p,j}(t_j) - G^0_{p,j}(t_j)\right)\right| &\longrightarrow 0 \quad \text{in probability}.
\end{aligned} \tag{C.138}$$

Proof of Lemma C.1.20. For any $j \in [p]$, we first show that in probability, $\sup_t |\widehat{G}^0_{p,j}(t) - G^0_{p,j}(t)|$ converges to 0 as $p \to \infty$ exponentially fast. For any $\epsilon \in (0,1)$, denote $-\infty = \alpha^{p,j}_0 < \alpha^{p,j}_1 < \cdots < \alpha^{p,j}_{N_\epsilon} = \infty$ in which $N_\epsilon = \lceil 2/\epsilon \rceil$, such that $G^0_{p,j}(\alpha^{p,j}_{k-1}) - G^0_{p,j}(\alpha^{p,j}_k) \le \epsilon/2$ for $k \in [N_\epsilon]$. We have

$$\begin{aligned}
\mathbb{P}\big(\sup_t \widehat{G}^0_{p,j}(t) - G^0_{p,j}(t) > \epsilon\big) &\le \mathbb{P}\Big(\bigcup_{k=1}^{N_\epsilon} \sup_{t\in[\alpha^{p,j}_{k-1},\alpha^{p,j}_k)} \widehat{G}^0_{p,j}(t) - G^0_{p,j}(t) > \epsilon\Big)\\
&\le \sum_{k=1}^{N_\epsilon} \mathbb{P}\Big(\sup_{t\in[\alpha^{p,j}_{k-1},\alpha^{p,j}_k)} \widehat{G}^0_{p,j}(t) - G^0_{p,j}(t) > \epsilon\Big).
\end{aligned} \tag{C.139}$$

We note that both $\widehat{G}^0_{p,j}(t)$ and $G^0_{p,j}(t)$ are monotonic decreasing function. Therefore, for any $k \in$

$[N_\epsilon]$, we have

$$\sup_{t\in\left[\alpha_{k-1}^{p,j},\alpha_k^{p,j}\right)} \widehat{G}_{p,j}^0(t) - G_{p,j}^0(t) \le \widehat{G}_{p,j}^0(\alpha_{k-1}^{p,j}) - G_{p,j}^0(\alpha_k^{p,j}) \le \widehat{G}_{p,j}^0(\alpha_{k-1}^{p,j}) - G_{p,j}^0(\alpha_{k-1}^{p,j}) + \epsilon/2. \tag{C.140}$$

Based on Equation (C.139), Assumption 3.4.5(a), and the Azuma-Hoeffding inequality, it follows that

$$\begin{aligned}\mathbb{P}\big(\sup_t \widehat{G}_{p,j}^0(t) - G_{p,j}^0(t) > \epsilon\big) &\le \sum_{k=1}^{N_\epsilon} \mathbb{P}\big(\widehat{G}_{p,j}^0(\alpha_{k-1}^{p,j}) - G_{p,j}^0(\alpha_{k-1}^{p,j}) > \frac{\epsilon}{2}\big) \\ &\le N_\epsilon \exp\left(-C|ne_j^c|^{1-2\alpha}\epsilon^2\right)\end{aligned} \tag{C.141}$$

for some constant $C > 0$. Similarly, we can show that

$$\begin{aligned}\mathbb{P}\big(\inf_t \widehat{G}_{p,j}^0(t) - G_{p,j}^0(t) < -\epsilon\big) &\le \sum_{k=1}^{N_\epsilon} \mathbb{P}\big(\widehat{G}_{p,j}^0(\alpha_k^{p,j}) - G_{p,j}^0(\alpha_k^{p,j}) < -\frac{\epsilon}{2}\big) \\ &\le N_\epsilon \exp\left(-C|ne_j^c|^{1-2\alpha}\epsilon^2\right).\end{aligned} \tag{C.142}$$

It follows that as $p \to \infty$,

$$\begin{aligned}\mathbb{P}\left(\sup_{t_1,\cdots,t_p}\left|\sum_{j=1}^p \pi_{p,j}^0\left(\widehat{G}_{p,j}^0(t_j) - G_{p,j}^0(t_j)\right)\right| > \epsilon\right) &\le \mathbb{P}\left(\bigcup_{j=1}^p \sup_{t_j\in\mathbb{R}}\left|\widehat{G}_{p,j}^0(t_j) - G_{p,j}^0(t_j)\right| > \epsilon\right) \\ &\le \sum_{j=1}^p \mathbb{P}\big(\sup_{t\in\mathbb{R}}\left|\widehat{G}_{p,j}^0(t) - G_{p,j}^0(t)\right| > \epsilon\big) \\ &\le 2N_\epsilon \sum_{j=1}^p \exp\left(-C|ne_j^c|^{1-2\alpha}\epsilon^2\right) \\ &\le 2N_\epsilon p \exp\big(-C\min_{j\in[p]}|ne_j^c|^{1-2\alpha}\epsilon^2\big) \\ &\to 0,\end{aligned} \tag{C.143}$$

under Assumption 3.4.5(b). The second convergence statement involves $\widehat{V}^0_{p,j}(t)$ can be shown similarly. This concludes the proof of Lemma C.1.20.

Proof of Proposition 3.4.6. Following Equation (3.22), we have

$$\begin{aligned}
\limsup_{p\to\infty} \mathrm{FDR} &\le \limsup_{p\to\infty} 2\mathbb{E}\left[\mathrm{FDP}_p\left(\tau^1_{q/2},\cdots,\tau^p_{q/2}\right)\right] \\
&\le \limsup_{p\to\infty} 2\mathbb{E}\left|\mathrm{FDP}_p\left(\tau^1_{q/2},\cdots,\tau^p_{q/2}\right) - \overline{\mathrm{FDP}}_p\left(\tau^1_{q/2},\cdots,\tau^p_{q/2}\right)\right| \\
&+ \limsup_{p\to\infty} 2\mathbb{E}\left|\mathrm{FDP}^\dagger_p\left(\tau^1_{q/2},\cdots,\tau^p_{q/2}\right) - \overline{\mathrm{FDP}}_p\left(\tau^1_{q/2},\cdots,\tau^p_{q/2}\right)\right| \\
&+ \limsup_{p\to\infty} 2\mathbb{E}\left[\mathrm{FDP}^\dagger_p\left(\tau^1_{q/2},\cdots,\tau^p_{q/2}\right)\right] \qquad \text{(C.144)} \\
&\le \limsup_{p\to\infty} 2\mathbb{E}\left[\sup_{t_1,\cdots,t_p>0}\left|\mathrm{FDP}_p(t_1,\cdots,t_p) - \overline{\mathrm{FDP}}_p(t_1,\cdots,t_p)\right|\right] \\
&+ \limsup_{p\to\infty} 2\mathbb{E}\left[\sup_{t_1,\cdots,t_p>0}\left|\mathrm{FDP}^\dagger_p(t_1,\cdots,t_p) - \overline{\mathrm{FDP}}_p(t_1,\cdots,t_p)\right|\right] \\
&+ \limsup_{p\to\infty} 2\mathbb{E}\left[\mathrm{FDP}^\dagger_p\left(\tau^1_{q/2},\cdots,\tau^p_{q/2}\right)\right].
\end{aligned}$$

The first two terms are 0 based on Lemma C.1.20 and the dominated convergence theorem. For the last term, we have

$$\limsup_{p\to\infty} 2\mathbb{E}\left[\mathrm{FDP}^\dagger_p\left(\tau^1_{q/2},\cdots,\tau^p_{q/2}\right)\right] \le \limsup_{p\to\infty} 2\mathbb{E}\left[\max_{j\in[p]} \frac{\#\{i\in ne^c_j, M_{ji} < -\tau^j_{q/2}\}}{\#\{M_{ji} > \tau^j_{q/2}\}\vee 1}\right] \le q \qquad \text{(C.145)}$$

based on the definition of $\tau^j_{q/2}$. This concludes the proof of Proposition 3.4.6.

C.2 More Simulation Details

Link function	$p = 500$		$p = 1000$		$p = 1500$		$p = 2000$		$p = 3000$	
$f_1(t)$	FDR	Power	FDR	Power	FDR	Power	FDR	Power	FDR	Power
DS-I	0.10	0.81	0.07	0.86	0.10	0.84	0.11	0.86	0.12	0.80
DS-II	0.07	0.80	0.10	0.83	0.11	0.82	0.14	0.86	0.11	0.80
MDS-I	0.05	0.79	0.03	0.83	0.06	0.82	0.07	0.85	0.08	0.79
MDS-II	0.02	0.78	0.07	0.86	0.09	0.84	0.07	0.85	0.09	0.82
DeepPINK-I	0.18	0.43	0.12	0.31	0.16	0.26	0.17	0.30	0.15	0.32
DeepPINK-II	0.14	0.41	0.15	0.24	0.13	0.17	0.12	0.17	0.14	0.13
$f_2(t)$	FDR	Power	FDR	Power	FDR	Power	FDR	Power	FDR	Power
DS-I	0.10	0.85	0.10	0.76	0.12	0.68	0.07	0.69	0.09	0.65
DS-II	0.10	0.87	0.09	0.72	0.11	0.66	0.08	0.67	0.08	0.62
MDS-I	0.05	0.83	0.06	0.74	0.08	0.66	0.05	0.67	0.06	0.63
MDS-II	0.05	0.84	0.04	0.71	0.07	0.65	0.04	0.64	0.05	0.61
DeepPINK-I	0.15	0.68	0.16	0.67	0.14	0.62	0.12	0.61	0.15	0.62
DeepPINK-II	0.11	0.59	0.10	0.61	0.18	0.56	0.14	0.59	0.07	0.54
$f_3(t)$	FDR	Power	FDR	Power	FDR	Power	FDR	Power	FDR	Power
DS-I	0.08	0.94	0.09	0.87	0.08	0.87	0.10	0.85	0.12	0.83
DS-II	0.10	0.96	0.10	0.89	0.10	0.88	0.11	0.87	0.13	0.83
MDS-I	0.04	0.90	0.06	0.84	0.05	0.84	0.06	0.82	0.06	0.80
MDS-II	0.04	0.92	0.05	0.86	0.07	0.85	0.07	0.84	0.09	0.81
DeepPINK-I	0.10	0.83	0.13	0.74	0.11	0.70	0.12	0.66	0.12	0.65
DeepPINK-II	0.09	0.88	0.09	0.84	0.14	0.83	0.12	0.79	0.10	0.68

Table C.1: Empirical FDRs and powers on the single-index models. The design matrix follows multivariate normal distribution with pairwise power decay correlation. The five methods are, the single data-splitting method using the influence function (DS-I) and its multiple data-splitting version (MDS-I), the single data-splitting method using the weight multiplication (DS-II) and its multiple data-splitting version (MDS-II), and the DeepPINK method with two different network architectures (see the text for details). The designated FDR control level is $q = 0.1$ in all settings. The reported results are the empirical means of 20 independent runs.

Link function	$p = 500$		$p = 1000$		$p = 1500$		$p = 2000$		$p = 3000$	
$f_1(t)$	FDR	Power	FDR	Power	FDR	Power	FDR	Power	FDR	Power
DS-I	0.10	0.82	0.13	0.87	0.11	0.85	0.15	0.87	0.13	0.85
DS-II	0.09	0.83	0.13	0.87	0.09	0.84	0.14	0.87	0.13	0.85
MDS-I	0.06	0.80	0.07	0.84	0.08	0.83	0.08	0.87	0.07	0.84
MDS-II	0.05	0.81	0.06	0.86	0.04	0.82	0.06	0.86	0.07	0.84
DeepPINK-I	0.15	0.37	0.12	0.21	0.16	0.27	0.17	0.33	0.19	0.36
DeepPINK-II	0.13	0.41	0.14	0.23	0.12	0.18	0.11	0.16	0.16	0.13
$f_2(t)$	FDR	Power	FDR	Power	FDR	Power	FDR	Power	FDR	Power
DS-I	0.14	0.87	0.11	0.75	0.08	0.66	0.08	0.68	0.09	0.66
DS-II	0.11	0.85	0.07	0.71	0.07	0.66	0.08	0.66	0.09	0.64
MDS-I	0.08	0.85	0.07	0.73	0.05	0.64	0.04	0.65	0.05	0.64
MDS-II	0.04	0.83	0.05	0.68	0.06	0.64	0.05	0.64	0.05	0.62
DeepPINK-I	0.09	0.63	0.12	0.63	0.14	0.66	0.12	0.62	0.14	0.62
DeepPINK-II	0.07	0.61	0.11	0.64	0.17	0.65	0.09	0.64	0.12	0.58
$f_3(t)$	FDR	Power	FDR	Power	FDR	Power	FDR	Power	FDR	Power
DS-I	0.11	0.96	0.10	0.90	0.09	0.89	0.10	0.87	0.10	0.84
DS-II	0.10	0.98	0.08	0.89	0.11	0.89	0.09	0.88	0.09	0.84
MDS-I	0.05	0.94	0.05	0.87	0.04	0.85	0.05	0.83	0.06	0.80
MDS-II	0.03	0.95	0.05	0.86	0.07	0.85	0.06	0.85	0.05	0.81
DeepPINK-I	0.11	0.87	0.15	0.78	0.12	0.70	0.11	0.71	0.14	0.74
DeepPINK-II	0.10	0.90	0.13	0.86	0.12	0.85	0.09	0.79	0.13	0.72

Table C.2: Empirical FDRs and powers on the single-index models based on 20 independent runs. The design matrix follows multivariate normal distribution with pairwise power decay partial correlation. The six methods are as per Figure C.1. The designated FDR control level is $q = 0.1$ in all settings.

Methods	$\rho = 0.0$		$\rho = 0.2$		$\rho = 0.4$		$\rho = 0.6$		$\rho = 0.8$	
($p = 500$)	FDR	Power	FDR	Power	FDR	Power	FDR	Power	FDR	Power
DS	0.10	0.83	0.10	0.81	0.07	0.80	0.10	0.73	0.11	0.58
MDS	0.04	0.86	0.04	0.86	0.05	0.83	0.05	0.77	0.06	0.67
M-knockoff	0.10	0.88	0.07	0.84	0.04	0.74	0.02	0.61	0.02	0.51
F -knockoff	0.09	0.84	0.10	0.85	0.09	0.82	0.10	0.76	0.11	0.62
BHq	0.06	0.81	0.06	0.80	0.06	0.77	0.09	0.72	0.08	0.61
BYq	0.01	0.76	0.01	0.72	0.01	0.73	0.01	0.67	0.02	0.55
MBHq	0.00	0.82	0.00	0.80	0.00	0.77	0.00	0.71	0.00	0.57
MBYq	0.00	0.82	0.00	0.79	0.00	0.77	0.00	0.71	0.00	0.58
($p = 1000$)	FDR	Power	FDR	Power	FDR	Power	FDR	Power	FDR	Power
DS	0.10	0.82	0.09	0.78	0.10	0.76	0.09	0.72	0.08	0.56
MDS	0.04	0.86	0.06	0.85	0.06	0.83	0.06	0.78	0.07	0.66
M-Knockoff	0.13	0.90	0.07	0.79	0.00	0.28	0.00	0.26	0.00	0.26
F -Knockoff	0.10	0.84	0.10	0.83	0.11	0.81	0.10	0.77	0.09	0.64
BHq	0.06	0.81	0.07	0.79	0.07	0.77	0.09	0.70	0.09	0.60
BYq	0.02	0.76	0.01	0.73	0.01	0.71	0.01	0.66	0.01	0.54
MBHq	0.00	0.82	0.00	0.80	0.00	0.78	0.00	0.73	0.00	0.58
MBYq	0.00	0.81	0.00	0.80	0.00	0.78	0.00	0.73	0.00	0.57
($p = 1500$)	FDR	Power	FDR	Power	FDR	Power	FDR	Power	FDR	Power
DS	0.10	0.74	0.10	0.72	0.10	0.67	0.10	0.62	0.11	0.51
MDS	0.07	0.81	0.06	0.81	0.08	0.77	0.09	0.73	0.09	0.60
M-Knockoff	0.11	0.86	0.03	0.68	0.00	0.16	0.00	0.09	0.00	0.08
F -Knockoff	0.11	0.79	0.11	0.78	0.12	0.74	0.13	0.70	0.12	0.58
BHq	0.07	0.74	0.07	0.72	0.11	0.67	0.05	0.64	0.08	0.52
BYq	0.01	0.68	0.01	0.66	0.03	0.61	0.01	0.58	0.01	0.45
MBHq	0.00	0.77	0.00	0.76	0.00	0.72	0.00	0.66	0.00	0.53
MBYq	0.00	0.77	0.00	0.76	0.00	0.72	0.00	0.66	0.00	0.53
($p = 2000$)	FDR	Power	FDR	Power	FDR	Power	FDR	Power	FDR	Power
DS	0.08	0.69	0.10	0.69	0.08	0.64	0.11	0.56	0.10	0.45
MDS	0.07	0.78	0.05	0.78	0.07	0.74	0.06	0.69	0.07	0.58
M-Knockoff	0.12	0.86	0.02	0.62	0.00	0.15	0.00	0.08	0.00	0.05
F -Knockoff	0.09	0.74	0.13	0.76	0.09	0.71	0.12	0.67	0.13	0.57
BHq	0.08	0.68	0.05	0.68	0.08	0.64	0.08	0.60	0.07	0.50
BYq	0.02	0.61	0.00	0.63	0.01	0.59	0.01	0.54	0.02	0.44
MBHq	0.00	0.75	0.00	0.74	0.00	0.70	0.00	0.64	0.00	0.53
MBYq	0.00	0.75	0.00	0.73	0.00	0.69	0.00	0.63	0.00	0.52

Table C.3: Empirical FDRs and powers on the linear model, where the design matrix has pairwise constant correlation ρ. The number of true features is 50 across all settings. The signal-to-noise ratio is $10 \times \sqrt{\log p/n}$. The designated FDR control level is $q = 0.1$. The eight methods are, the single data-splitting method (DS), the multiple data-splitting method (MDS), the model-X knockoff filter (M-Knockoff), the fixed-design knockoff filter with data recycling (F-Knockoff), the Benjamini-Hochberg method (BHq) and its multiple data-splitting version (MBHq), the Benjamini-Yekutieli method (BYq) and its multiple data-splitting version (MBYq). The reported results are the empirical means of 50 independent runs.

SNR	$4\sqrt{\log p/n}$		$8\sqrt{\log p/n}$		$12\sqrt{\log p/n}$		$16\sqrt{\log p/n}$		$20\sqrt{\log p/n}$	
$(p = 500)$	FDR	Power	FDR	Power	FDR	Power	FDR	Power	FDR	Power
DS	0.10	0.48	0.09	0.70	0.10	0.78	0.10	0.82	0.09	0.84
MDS	0.09	0.59	0.06	0.76	0.05	0.83	0.04	0.86	0.04	0.87
M-Knockoff	0.00	0.30	0.01	0.59	0.03	0.73	0.03	0.78	0.05	0.82
F -Knockoff	0.10	0.56	0.10	0.74	0.09	0.82	0.09	0.85	0.09	0.87
BHq	0.07	0.45	0.07	0.69	0.07	0.78	0.05	0.83	0.06	0.86
BYq	0.01	0.36	0.01	0.62	0.01	0.74	0.01	0.80	0.01	0.84
MBHq	0.00	0.45	0.00	0.68	0.00	0.77	0.00	0.82	0.00	0.85
MBYq	0.00	0.44	0.00	0.68	0.00	0.77	0.00	0.82	0.00	0.85
$(p = 1000)$	FDR	Power	FDR	Power	FDR	Power	FDR	Power	FDR	Power
DS	0.010	0.44	0.10	0.67	0.10	0.76	0.10	0.80	0.10	0.83
MDS	0.09	0.58	0.07	0.76	0.05	0.82	0.05	0.86	0.05	0.87
M-Knockoff	0.00	0.10	0.00	0.26	0.00	0.37	0.00	0.44	0.00	0.47
F -Knockoff	0.12	0.55	0.10	0.72	0.10	0.81	0.09	0.84	0.09	0.86
BHq	0.05	0.42	0.06	0.67	0.07	0.76	0.06	0.82	0.06	0.85
BYq	0.01	0.31	0.01	0.60	0.01	0.71	0.01	0.78	0.01	0.83
MBHq	0.00	0.44	0.00	0.68	0.00	0.77	0.00	0.83	0.00	0.85
MBYq	0.00	0.42	0.00	0.68	0.00	0.77	0.00	0.83	0.00	0.85
$(p = 1500)$	FDR	Power	FDR	Power	FDR	Power	FDR	Power	FDR	Power
DS	0.10	0.40	0.09	0.62	0.10	0.70	0.10	0.74	0.10	0.77
MDS	0.09	0.54	0.09	0.72	0.07	0.79	0.06	0.82	0.05	0.84
M-Knockoff	0.00	0.05	0.00	0.11	0.00	0.16	0.00	0.19	0.00	0.25
F -Knockoff	0.15	0.53	0.12	0.70	0.10	0.76	0.11	0.79	0.10	0.81
BHq	0.08	0.38	0.07	0.62	0.06	0.72	0.06	0.78	0.05	0.81
BYq	0.01	0.29	0.01	0.56	0.01	0.66	0.01	0.74	0.01	0.78
MBHq	0.00	0.41	0.00	0.65	0.00	0.74	0.00	0.79	0.00	0.83
MBYq	0.00	0.40	0.00	0.65	0.00	0.74	0.00	0.79	0.00	0.83
$(p = 2000)$	FDR	Power	FDR	Power	FDR	Power	FDR	Power	FDR	Power
DS	0.11	0.34	0.10	0.56	0.10	0.66	0.10	0.70	0.10	0.73
MDS	0.09	0.51	0.09	0.68	0.07	0.76	0.07	0.79	0.06	0.81
M-Knockoff	0.00	0.04	0.00	0.09	0.00	0.14	0.00	0.17	0.00	0.19
F -Knockoff	0.16	0.50	0.12	0.65	0.11	0.73	0.11	0.76	0.11	0.78
BHq	0.08	0.34	0.08	0.55	0.08	0.68	0.07	0.72	0.08	0.77
BYq	0.01	0.25	0.01	0.48	0.02	0.62	0.01	0.67	0.01	0.73
MBHq	0.00	0.39	0.00	0.61	0.00	0.71	0.00	0.77	0.00	0.80
MBYq	0.00	0.38	0.00	0.61	0.00	0.71	0.00	0.77	0.00	0.80

Table C.4: Empirical FDRs and powers on the linear model, where the design matrix follows a multivariate normal distribution with constant pairwise correlation 0.5. SNR represents the signal-to-noise ratio. The number of true features is 50 across all settings. The designated FDR control level is $q = 0.1$. The eight methods are, the single data-splitting method (DS), the multiple data-splitting method (MDS), the model-X knockoff filter (M-Knockoff), the fixed-design knockoff filter with data recycling (F-Knockoff), the Benjamini-Hochberg method (BHq) and its multiple-splitting version (MBHq), the Benjamini-Yekutieli method (BYq) and its multiple-splitting version (MBYq). The reported results are the empirical means of 50 independent runs.

References

Alder, S., S. Trebst, A. K. Hartmann, and M. Troyer (2004). Dynamics of the Wang-Landau algorithm and complexity of rare events for the three-dimensional bimodal Ising spin glass. *Journal of Statistical Mechanics: Theory and Experiment 2004*(07), P07008.

Andrieu, C., E. Moulines, and P. Priouret (2005). Stability of stochastic approximation under verifiable conditions. *SIAM Journal on Control and Optimization 44*(1), 283–312.

Atchadé, Y. F. and J. S. Liu (2010). The Wang-Landau algorithm in general state spaces: Applications and convergence analysis. *Statistica Sinica*, 209–233.

Azriel, D. and A. Schwartzman (2015). The empirical distribution of a large number of correlated normal variables. *Journal of the American Statistical Association 110*(511), 1217–1228.

Barber, R. F. and E. J. Candès (2015). Controlling the false discovery rate via knockoffs. *The Annals of Statistics 43*(5), 2055–2085.

Barber, R. F. and E. J. Candès (2019). A knockoff filter for high-dimensional selective inference. *The Annals of Statistics 47*(5), 2504–2537.

Beale, P. D. (1996). Exact distribution of energies in the two-dimensional Ising model. *Physical Review Letters 76*(1), 78.

Belardinelli, R. E. and V. D. Pereyra (2007). Fast algorithm to calculate density of states. *Physical Review E 75*(4), 046701.

Benjamini, Y. and Y. Hochberg (1995). Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 57*(1), 289–300.

Benjamini, Y. and D. Yekutieli (2001). The control of the false discovery rate in multiple testing under dependency. *The Annals of Statistics 29*(4), 1165–1188.

Berg, B. A. and T. Neuhaus (1992). Multicanonical ensemble: a new approach to simulate first-order phase transitions. *Physical Review Letters 68*(1), 9.

Berger, J. O. and L. R. Pericchi (1996). The intrinsic Bayes factor for model selection and prediction. *Journal of the American Statistical Association 91*(433), 109–122.

Berger, J. O., L. R. Pericchi, J. K. Ghosh, T. Samanta, and F. De Santis (2001). Objective Bayesian methods for model selection: Introduction and comparison. *Lecture Notes-Monograph Series*, 135–207.

Berger, J. O., L. R. Pericchi, and J. A. Varshavsky (1998). Bayes factors and marginal distributions in invariant situations. *Sankhyā: The Indian Journal of Statistics, Series A*, 307–321.

Bickel, P. J., Y. Ritov, and A. B. Tsybakov (2009). Simultaneous analysis of Lasso and Dantzig selector. *The Annals of Statistics 37*(4), 1705–1732.

Bishop, C. M. (2006). Pattern recognition and machine learning (information science and statistics) springer-verlag new york. *Inc. Secaucus, NJ, USA*.

Blei, D. M., A. Kucukelbir, and J. D. McAuliffe (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association 112*(518), 859–877.

Bornn, L., P. E. Jacob, P. D. Moral, and A. Doucet (2013). An adaptive interacting Wang-Landau algorithm for automatic density exploration. *Journal of Computational and Graphical Statistics 22*(3), 749–773.

Brooks, S. P., P. Giudici, and G. O. Roberts (2003). Efficient construction of reversible jump Markov chain Monte Carlo proposal distributions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 65*(1), 3–39.

Brown, G. and T. C. Schulthess (2005). Wang-Landau estimation of magnetic properties for the Heisenberg model. *Journal of Applied Physics 97*(10), 477.

Bühlmann, P. and S. Van De Geer (2011). *Statistics for high-dimensional data: Methods, theory and applications*. Springer Science & Business Media.

Calvo, F. (2002). Sampling along reaction coordinates with the Wang-Landau method. *Molecular Physics 100*(21), 3421–3427.

Candes, E., Y. Fan, L. Janson, and J. Lv (2018). Panning for gold: model-X knockoffs for high-dimensional controlled variable selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 80*(3), 551–577.

Candes, E. and T. Tao (2007). The Dantzig selector: Statistical estimation when p is much larger than n. *The Annals of Statistics 35*(6), 2313–2351.

Candès, E. J. and P. Sur (2020). The phase transition for the existence of the maximum likelihood estimate in high-dimensional logistic regression. *The Annals of Statistics 48*(1), 27–42.

Cérou, F., P. Del Moral, T. Furon, and A. Guyader (2012). Sequential Monte Carlo for rare event estimation. *Statistics and Computing 22*(3), 795–808.

Chib, S. (1995). Marginal likelihood from the Gibbs output. *Journal of the American Statistical Association 90*(432), 1313–1321.

Chib, S. and I. Jeliazkov (2001). Marginal likelihood from the Metropolis-Hastings output. *Journal of the American Statistical Association 96*(453), 270–281.

Chopin, N. (2002). A sequential particle filter method for static models. *Biometrika 89*(3), 539–552.

Cox, D. R. (1975). A note on data-splitting for the evaluation of significance levels. *Biometrika 62*(2), 441–444.

Dai, C., B. Lin, X. Xing, and J. S. Liu (2020). False discovery rate control via data splitting. *arXiv:2002.08542*.

Dai, C. and J. S. Liu (2019). Monte Carlo approximation of Bayes factors via mixing with surrogate distributions. *arXiv:1909.05922*.

Dai, C. and J. S. Liu (2020). Wang-Landau algorithm as stochastic optimization and its acceleration. *Physical Review E 101*(3), 033301.

Darken, C. and J. Moody (1992). Towards faster stochastic gradient search. In *Advances in Neural Information Processing Systems*, pp. 1009–1016.

Dayal, P., S. Trebst, S. Wessel, D. Wuertz, M. Troyer, S. Sabhapandit, and S. N. Coppersmith (2004). Performance limitations of flat-histogram methods. *Physical Review Letters 92*(9), 097201.

Del Moral, P. (2004). Feynman-kac formulae: Genealogical and interacting particle systems with applications.

Del Moral, P., A. Doucet, and A. Jasra (2006). Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 68*(3), 411–436.

Dempster, A. P., N. M. Laird, and D. B. Rubin (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological) 39*(1), 1–22.

Dezeure, R., P. Bühlmann, L. Meier, and N. Meinshausen (2015). High-dimensional inference: Confidence intervals, p-values and R-software hdi. *Statistical Science*, 533–558.

Diebolt, J. and C. P. Robert (1994). Estimation of finite mixture distributions through Bayesian sampling. *Journal of the Royal Statistical Society: Series B (Methodological) 56*(2), 363–375.

Doucet, A., S. Godsill, and C. Andrieu (2000). On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing 10*(3), 197–208.

Duchi, J., E. Hazan, and Y. Singer (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research 12*(Jul), 2121–2159.

Fan, J. and R. Li (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association 96*(456), 1348–1360.

Fan, Y., R. Wu, M. Chen, L. Kuo, and P. O. Lewis (2011). Choosing among partition models in Bayesian phylogenetics. *Molecular Biology and Evolution 28*(1), 523–532.

Fort, G., B. Jourdain, E. Kuhn, T. Lelièvre, and G. Stoltz (2015). Convergence of the Wang-Landau algorithm. *Mathematics of Computation 84*(295), 2297–2327.

Fort, G., E. Moulines, and P. Priouret (2011). Convergence of adaptive and interacting Markov chain Monte Carlo algorithms. *The Annals of Statistics 39*(6), 3262–3289.

Fourment, M., A. F. Magee, C. Whidden, A. Bilge, F. A. Matsen IV, and V. N. Minin (2020). 19 dubious ways to compute the marginal likelihood of a phylogenetic tree topology. *Systematic Biology 69*(2), 209–220.

Frank, A. and A. Asuncion (2011). Uci machine learning repository, 2010. *URL http://archive. ics. uci. edu/ml 15*, 22.

Fytas, N. G. and A. Malakis (2008). Phase diagram of the 3D bimodal random-field Ising model. *The European Physical Journal B 61*(1), 111–120.

Gelfand, A. E. and A. F. M. Smith (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American statistical association 85*(410), 398–409.

Gelman, A. and X. Meng (1998). Simulating normalizing constants: From importance sampling to bridge sampling to path sampling. *Statistical Science*, 163–185.

Geyer, C. (1991). Markov chain Monte Carlo maximum likelihood. In E. Keramigas (Ed.), *Computing Science and Statistics: he 23rd symposium on the interface*, Fairfax, pp. 156–163. Interface Foundation.

Geyer, C. J. (1994). Estimating normalizing constants and reweighting mixtures.

Green, P. J. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika 82*(4), 711–732.

Gronau, Q. F., H. Singmann, and E. Wagenmakers (2017). Bridgesampling: An R package for estimating normalizing constants. *arXiv:1710.08162*.

Hammersley, J. M. and K. W. Morton (1954). Poor man's Monte Carlo. *Journal of the Royal Statistical Society: Series B (Methodological) 16*(1), 23–38.

Hechtlinger, Y. (2016). Interpretation of prediction models using the input gradient. *arXiv:1611.07634*.

Heng, J. and P. E. Jacob (2019). Unbiased Hamiltonian Monte Carlo with couplings. *Biometrika 106*(2), 287–302.

Hernández, L. and H. Ceva (2008). Wang-Landau study of the critical behavior of the bimodal 3D random field Ising model. *Physica A: Statistical Mechanics and its Applications 387*(12), 2793–2801.

Ignatiadis, N., B. Klaus, J. B. Zaugg, and W. Huber (2016). Data-driven hypothesis weighting increases detection power in genome-scale multiple testing. *Nature methods 13*(7), 577.

Jain, P., S. M. Kakade, R. Kidambi, P. Netrapalli, and A. Sidford (2017). Accelerating stochastic gradient descent. *stat 1050*, 26.

Javanmard, A. and H. Javadi (2019). False discovery rate control via debiased Lasso. *Electronic Journal of Statistics 13*(1), 1212–1253.

Javanmard, A. and A. Montanari (2013). Nearly optimal sample size in hypothesis testing for high-dimensional regression. In *2013 51st Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 1427–1434. IEEE.

Javanmard, A. and A. Montanari (2014). Confidence intervals and hypothesis testing for high-dimensional regression. *The Journal of Machine Learning Research 15*(1), 2869–2909.

Jayasri, D., V. S. S. Sastry, and K. P. N. Murthy (2005). Wang-Landau Monte Carlo simulation of isotropic-nematic transition in liquid crystals. *Physical Review E 72*(3), 036702.

Jordan, M. I., Z. Ghahramani, T. S. Jaakkola, and L. K. Saul (1999). An introduction to variational methods for graphical models. *Machine Learning 37*(2), 183–233.

Kass, R. E. and A. E. Raftery (1995). Bayes factors. *Journal of the American Statistical Association 90*, 773–795.

Katsevich, E. and C. Sabatti (2019). Multilayer knockoff filter: Controlled variable selection at multiple resolutions. *The Annals of Applied Statistics 13*(1), 1–33.

Kim, S. (2015). ppcor: An R package for a fast calculation to semi-partial correlation coefficients. *Communications for Statistical Applications and Methods 22*(6), 665.

Kingma, D. P. and J. Ba (2014). Adam: A method for stochastic optimization. *arXiv:1412.6980*.

Kong, A., J. S. Liu, and W. H. Wong (1994). Sequential imputations and Bayesian missing data problems. *Journal of the American Statistical Association 89*(425), 278–288.

Konishi, S. and G. Kitagawa (2008). *Information criteria and statistical modeling*. Springer Science & Business Media.

Kotz, S., N. Balakrishnan, and N. L. Johnson (2000). Bivariate and trivariate normal distributions. *Continuous multivariate distributions 1*, 251–348.

Landau, D. P., S. Tsai, and M. Exler (2004). A new approach to Monte Carlo simulations in statistical physics: Wang-Landau sampling. *American Journal of Physics 72*(10), 1294–1302.

Langfeld, K., B. Lucini, and A. Rago (2012). Density of states in gauge theories. *Physical Review Letters 109*(11), 111601.

Lauritzen, S. L. (1996). Graphical models.

Lee, J. D., D. L. Sun, Y. Sun, and J. E. Taylor (2016). Exact post-selection inference, with application to the Lasso. *The Annals of Statistics 44*(3), 907–927.

Li, J. and M. H. Maathuis (2019). Nodewise knockoffs: False discovery rate control for Gaussian graphical models. *arXiv:1908.11611*.

Li, Y. W., T. Wüst, D. P. Landau, and H. Q. Lin (2007). Numerical integration using Wang-Landau sampling. *Computer Physics Communications 177*(6), 524–529.

Liang, F. (2005). A generalized Wang-Landau algorithm for Monte Carlo computation. *Journal of the American Statistical Association 100*(472), 1311–1327.

Liu, J. S. (2008). *Monte Carlo Strategies in Scientific Computing*. Springer Science & Business Media.

Liu, J. S. and R. Chen (1998). Sequential Monte Carlo methods for dynamic systems. *Journal of the American statistical association 93*(443), 1032–1044.

Liu, J. S., R. Chen, and T. Logvinenko (2001). A theoretical framework for sequential importance sampling with resampling. In *Sequential Monte Carlo methods in practice*, pp. 225–246. Springer.

Liu, J. S., F. Liang, and W. H. Wong (2000). The multiple-try method and local optimization in Metropolis sampling. *Journal of the American Statistical Association 95*(449), 121–134.

Liu, W. (2013). Gaussian graphical model estimation with false discovery rate control. *The Annals of Statistics 41*(6), 2948–2978.

Lu, Y., Y. Fan, J. Lv, and W. S. Noble (2018). DeepPINK: reproducible feature selection in deep neural networks. In *Advances in Neural Information Processing Systems*, pp. 8676–8686.

Ma, R., T. T. Cai, and H. Li (2020). Global and simultaneous hypothesis testing for high-dimensional logistic regression models. *Journal of the American Statistical Association*, 1–15.

Malakis, A. and N. G. Fytas (2006). Lack of self-averaging of the specific heat in the three-dimensional random-field Ising model. *Physical Review E 73*(1), 016109.

Mastny, E. A. and J. J. de Pablo (2005). Direct calculation of solid-liquid equilibria from density-of-states Monte Carlo simulations. *The Journal of Chemical Physics 122*(12), 124109.

McDonald, G. C. and R. C. Schwing (1973). Instabilities of regression estimates relating air pollution to mortality. *Technometrics 15*(3), 463–481.

Meinshausen, N. and P. Bühlmann (2006). High-dimensional graphs and variable selection with the Lasso. *The Annals of Statistics 34*(3), 1436–1462.

Meinshausen, N. and P. Bühlmann (2010). Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 72*(4), 417–473.

Meinshausen, N., L. Meier, and P. Bühlmann (2009). p-values for high-dimensional regression. *Journal of the American Statistical Association 104*(488), 1671–1681.

Meng, X. and S. Schilling (1996). Fitting full-information item factor models and an empirical investigation of bridge sampling. *Journal of the American Statistical Association 91*(435), 1254–1267.

Meng, X. and W. H. Wong (1996). Simulating ratios of normalizing constants via a simple identity: A theoretical exploration. *Statistica Sinica*, 831–860.

Metropolis, N., A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics 21*(6), 1087–1092.

Mezard, M. and A. Montanari (2009). *Information, physics, and computation*. Oxford University Press.

Møller, J., A. R. Syversveen, and R. P. Waagepetersen (1998). Log Gaussian Cox processes. *Scandinavian Journal of Statistics 25*(3), 451–482.

Moran, P. A. P. (1973). Dividing a sample into two parts a statistical dilemma. *Sankhyā: The Indian Journal of Statistics, Series A*, 329–333.

Nemirovski, A., A. Juditsky, G. Lan, and A. Shapiro (2009). Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization 19*(4), 1574–1609.

Ogata, Y. (1989). A Monte Carlo method for high dimensional integration. *Numerische Mathematik 55*(2), 137–157.

O'Hagan, A. (1995). Fractional Bayes factors for model comparison. *Journal of the Royal Statistical Society: Series B (Methodological) 57*(1), 99–118.

O'Hara, R. B. and M. J. Sillanpää (2009). A review of Bayesian variable selection methods: What, how and which. *Bayesian Analysis 4*(1), 85–117.

Okabe, Y., Y. Tomita, and C. Yamaguchi (2002). Application of new Monte Carlo algorithms to random spin systems. *Computer Physics Communications 146*(1), 63–68.

Pandolfi, S., F. Bartolucci, and N. Friel (2014). A generalized multiple-try version of the reversible jump algorithm. *Computational Statistics & Data Analysis 72*, 298–314.

Park, T. and G. Casella (2008). The Bayesian Lasso. *Journal of the American Statistical Association 103*(482), 681–686.

Penttinen, A., D. Stoyan, and H. M. Henttonen (1992). Marked point processes in forest statistics. *Forest Science 38*(4), 806–824.

Polyak, B. T. (1964). Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics 4*(5), 1–17.

Poulain, P., F. Calvo, R. Antoine, M. Broyer, and P. Dugourd (2006). Performances of Wang-Landau algorithms for continuous systems. *Physical Review E 73*(5), 056704.

Raskutti, G., M. J. Wainwright, and B. Yu (2010). Restricted eigenvalue properties for correlated Gaussian designs. *Journal of Machine Learning Research 11*(Aug), 2241–2259.

Rathore, N. and J. J. de Pablo (2002). Monte Carlo simulation of proteins through a random walk in energy space. *The Journal of Chemical Physics 116*(16), 7225–7230.

Rathore, N., T. A. K. IV, and J. J. de Pablo (2003). Density of states simulations of proteins. *The Journal of Chemical Physics 118*(9), 4285–4290.

Rathore, N., Q. Yan, and J. J. de Pablo (2004). Molecular simulation of the reversible mechanical unfolding of proteins. *The Journal of Chemical Physics 120*(12), 5781–5788.

Rhee, S. Y., W. J. Fessel, A. R. Zolopa, L. Hurley, T. Liu, J. Taylor, D. P. Nguyen, S. Slome, D. Klein, and M. Horberg (2005). HIV-1 protease and reverse-transcriptase mutations: correlations with antiretroviral therapy in subtype B isolates and implications for drug-resistance surveillance. *The Journal of Infectious Diseases 192*(3), 456–465.

Rhee, S. Y., J. Taylor, G. Wadhera, A. Ben-Hur, D. L. Brutlag, and R. W. Shafer (2006). Genotypic predictors of human immunodeficiency virus type 1 drug resistance. *Proceedings of the National Academy of Sciences 103*(46), 17355–17360.

Rinaldo, A., L. Wasserman, M. G'Sell, and J. Lei (2016). Bootstrapping and sample splitting for high-dimensional, assumption-free inference. *arXiv:1611.05401*.

Robert, C. (2007). *The Bayesian choice: from decision-theoretic foundations to computational implementation*. Springer Science & Business Media.

Romano, J. P. and C. DiCiccio (2019). Multiple data splitting for testing.

Rosenbluth, M. N. and A. W. Rosenbluth (1955). Monte Carlo calculation of the average extension of molecular chains. *The Journal of Chemical Physics 23*(2), 356–359.

Rubin, D., S. Dudoit, and M. V. der Laan (2006). A method to increase the power of multiple testing procedures through sample splitting. *Statistical Applications in Genetics and Molecular Biology 5*(1).

Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv:1609.04747*.

Salomone, R., L. F. South, C. C. Drovandi, and D. P. Kroese (2018). Unbiased and consistent nested sampling via sequential Monte Carlo. *arXiv:1805.03924*.

Shell, M. S., P. G. Debenedetti, and A. Z. Panagiotopoulos (2002). Generalization of the Wang-Landau method for off-lattice simulations. *Physical Review E 66*(5), 056703.

Snider, J. and C. Y. Clare (2005). Absence of dipole glass transition for randomly dilute classical Ising dipoles. *Physical Review B 72*(21), 214203.

Stan Development Team (2019). RStan: the R interface to Stan. R package version 2.19.2.

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 36*(2), 111–133.

Storey, J. D., J. E. Taylor, and D. Siegmund (2004). Strong control, conservative point estimation and simultaneous conservative consistency of false discovery rates: A unified approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 66*(1), 187–205.

Stoyan, D. and H. Stoyan (1994). *Fractals, random shapes, and point fields: methods of geometrical statistics*, Volume 302. John Wiley & Sons Inc.

Strathmann, J. L., F. Rampf, W. Paul, and K. Binder (2008). Transitions of tethered polymer chains. *The Journal of Chemical Physics 128*, 064903.

Sun, T. and C. H. Zhang (2012). Scaled sparse linear regression. *Biometrika 99*(4), 879–898.

Sur, P. and E. J. Candès (2018). Supporting information to: A modern maximum-likelihood theory for high-dimensional logistic regression.

Sur, P. and E. J. Candès (2019). A modern maximum-likelihood theory for high-dimensional logistic regression. *Proceedings of the National Academy of Sciences 116*(29), 14516–14525.

Sutskever, I., J. Martens, G. Dahl, and G. Hinton (2013). On the importance of initialization and momentum in deep learning. In *International Conference on Machine Learning*, pp. 1139–1147.

Taylor, J., R. Lockhart, R. J. Tibshirani, and R. Tibshirani (2014). Exact post-selection inference for forward stepwise and least angle regression. *arXiv:1401.3889 7*, 10–1.

Taylor, M. P., W. Paul, and K. Binder (2009). Phase transitions of a single polymer chain: A Wang-Landau simulation study. *The Journal of Chemical Physics 131*(11), 114907.

Tibshirani, R. (1996). Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 58*(1), 267–288.

Tibshirani, R. J., J. Taylor, R. Lockhart, and R. Tibshirani (2016). Exact post-selection inference for sequential regression procedures. *Journal of the American Statistical Association 111*(514), 600–620.

Tjelmeland, H. and B. K. Hegstad (2001). Mode jumping proposals in MCMC. *Scandinavian Journal of Statistics 28*(1), 205–223.

Torbrüegge, S. and J. Schnack (2007). Sampling the two-dimensional density of states g(E, M) of a giant magnetic molecule using the Wang-Landau method. *Physical Review B 75*(5), 054403.

Tröster, A. and C. Dellago (2005). Wang-Landau sampling with self-adaptive range. *Physical Review E 71*(6), 066705.

Tsai, S., F. Wang, and D. P. Landau (2007). Critical endpoint behavior in an asymmetric Ising model: Application of Wang-Landau sampling to calculate the density of states. *Physical Review E 75*(6), 061108.

Tzikas, D. G., A. C. Likas, and N. P. Galatsanos (2008). The variational approximation for Bayesian inference. *IEEE Signal Processing Magazine 25*(6), 131–146.

Van de Geer, S., P. Bühlmann, Y. Ritov, and R. Dezeure (2014). On asymptotically optimal confidence regions and tests for high-dimensional models. *The Annals of Statistics 42*(3), 1166–1202.

van de Wiel, M. A., J. Berkhof, and W. N. van Wieringen (2009). Testing the prediction error difference between 2 predictors. *Biostatistics 10*(3), 550–560.

Vershynin, R. (2010). Introduction to the non-asymptotic analysis of random matrices. *arXiv:1011.3027*.

Vogel, T., Y. W. Li, T. Wüst, and D. P. Landau (2013). Generic, hierarchical framework for massively parallel Wang-Landau sampling. *Physical Review Letters 110*(21), 210603.

Wainwright, M. J. (2019). *High-dimensional statistics: A non-asymptotic viewpoint*, Volume 48. Cambridge University Press.

Wainwright, M. J. and M. I. Jordan (2008). Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning 1*(1–2), 1–305.

Wang, F. and D. P. Landau (2001a). Determining the density of states for classical statistical models: A random walk algorithm to produce a flat histogram. *Physical Review E 64*(5), 056101.

Wang, F. and D. P. Landau (2001b). Efficient, multiple-range random walk algorithm to calculate the density of states. *Physical Review Letters 86*(10), 2050.

Wang, L., S. Wang, and A. Bouchard-Côté (2020). An annealed sequential Monte Carlo method for Bayesian phylogenetics. *Systematic Biology 69*(1), 155–183.

Wasserman, L. and K. Roeder (2009). High-dimensional variable selection. *The Annals of Statistics 37*(5A), 2178.

Wu, Y., M. Körner, L. Colonna-Romano, S. Trebst, H. Gould, J. Machta, and M. Troyer (2005). Overcoming the slowing down of flat-histogram Monte Carlo simulations: Cluster updates and optimized broad-histogram ensembles. *Physical Review E 72*(4), 046704.

Wu, Y. and J. Machta (2005). Ground states and thermal states of the random field Ising model. *Physical Review Letters 95*(13), 137208.

Wüst, T. and D. P. Landau (2009). Versatile approach to access the low temperature thermodynamics of lattice polymers and proteins. *Physical Review Letters 102*(17), 178101.

Xie, W., P. O. Lewis, Y. Fan, L. Kuo, and M. H. Chen (2011). Improving marginal likelihood estimation for Bayesian phylogenetic model selection. *Systematic biology 60*(2), 150–160.

Xin, X., Z. Zhao, and J. S. Liu (2019). Controlling false discovery rate using Gaussian mirrors. *arXiv:1911.09761*.

Yamaguchi, C. and N. Kawashima (2002). Combination of improved multibondic method and the Wang-Landau method. *Physical Review E 65*(5), 056710.

Yamaguchi, C. and Y. Okabe (2001). Three-dimensional antiferromagnetic q-state Potts models: Application of the Wang-Landau algorithm. *Journal of Physics A: Mathematical and General 34*(42), 8781.

Yang, Y., M. J. Wainwright, and M. I. Jordan (2016). On the computational complexity of high-dimensional Bayesian variable selection. *The Annals of Statistics 44*(6), 2497–2532.

Zeile, M. D. (2012). AdaDelta: An adaptive learning rate method. *arXiv:1212.5701*.

Zhang, C. H. and S. S. Zhang (2014). Confidence intervals for low dimensional parameters in high dimensional linear models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 76*(1), 217–242.

Zhang, R., Z. Ren, and W. Chen (2018). SILGGM: An extensive R package for efficient statistical inference in large-scale gene networks. *PLoS Computational Biology 14*(8), e1006369.

Zhao, Q., P. Sur, and E. J. Candès (2020). The asymptotic distribution of the mle in high-dimensional logistic models: Arbitrary covariance. *arXiv:2001.09351*.

Zhou, C. and R. N. Bhatt (2005). Understanding and improving the Wang-Landau algorithm. *Physical Review E 72*(2), 025701.

Zhou, C., T. C. Schulthess, S. Torbrügge, and D. P. Landau (2006). Wang-Landau algorithm for continuous models and joint density of states. *Physical Review Letters 96*(12), 120201.

Zhou, C. and J. Su (2008). Optimal modification factor and convergence of the Wang-Landau algorithm. *Physical Review E 78*(4), 046705.

Zhou, Y., A. M. Johansen, and J. A. D. Aston (2016). Toward automatic model comparison: An adaptive sequential Monte Carlo approach. *Journal of Computational and Graphical Statistics 25*(3), 701–726.

Zou, H. and T. Hastie (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 67*(2), 301–320.