Bayesian Multi-Task Variable Selection with an Application to Differential DAG Analysis

Guanxun Li and Quan Zhou
Department of Statistics, Texas A&M University, College Station, TX

1. Introduction
In machine learning, multi-task learning refers to the paradigm where we simultaneously learn multiple related tasks instead of learning each task independently (Zhang and Yang 2021). In the context of model selection, we can formulate the problem as follows: given K observed datasets where the kth dataset is generated from some statistical model $\Omega^{(k)}$, simultaneously estimate $\Omega^{(1)}, \ldots, \Omega^{(K)}$ so that the estimation of $\Omega^{(k)}$ (for any $k = 1, \ldots, K$) uses information from all K datasets. In real problems where the K models tend to share many common features, this joint estimation approach is expected to have better performance than separate estimation (i.e., estimating $\Omega^{(k)}$ using only the kth dataset). In this work, we consider multi-task model selection problems where each task may be variable selection or structure learning.

We first study the multi-task variable selection problem, where each dataset is generated from a sparse linear regression model. The majority of the existing research has been conducted under the strict assumption that the “activated” covariates (i.e., covariates with nonzero regression coefficients) are shared across all datasets (Lounici et al. 2009, 2011). Recent works have relaxed this assumption by taking a more adaptable strategy that splits each regression coefficient into a shared and an individual component (Jalali et al. 2010; Hernández-Lobato, Hernández-Lobato, and Ghahramani 2015). We propose a more flexible Bayesian method which generalizes the well-known spike-and-slab prior (George and McCulloch 1993; Ishwaran and Rao 2005) and allows a covariate to be activated in an arbitrary number of datasets with varying effect sizes. We prove the posterior consistency for our model in high-dimensional scenarios. While there is a large literature on frequentists’ approaches to multi-task learning, the corresponding Bayesian methodology has received less attention and in particular theoretical results are lacking (Bonilla, Chai, and Williams 2007; Guo, Zoeter, and Archambeau 2011; Hernández-Lobato, Hernández-Lobato, and Ghahramani 2015). To our knowledge, this is the first work that establishes the theoretical guarantee for the high-dimensional Bayesian multi-task variable selection problem.

The traditional method for obtaining the posterior distribution for a Bayesian model is to use Markov chain Monte Carlo (MCMC) sampling, which is often computationally intensive, especially for multi-task learning problems where the space of candidate models can be enormous. A more scalable alternative is variational Bayes (VB), which recasts posterior approximation as an optimization problem (Ray and Szabó 2021). To carry out efficient VB inference, we approximate our spike-and-slab prior model using a novel multi-task sum of single effects (muSuSiE) model, which extends the sum of single effects (SuSiE) model of Wang et al. (2020a) to multiple datasets. Then, we propose to fit muSuSiE using an iterative Bayesian stepwise selection (IBSS) method, which may be thought of as a coordinate ascent algorithm for maximizing the evidence lower bound over a particular variational family.

To illustrate the application of the proposed methodology to more complex multi-task learning problems, we consider differential network analysis based on directed acyclic graphs (DAGs), which is essentially a multi-task structure learning problem. Differential network analysis has emerged as a significant topic in biology and received increasing attention over recent years. Its application can be found in the analysis of various diseases and biological mechanisms such as lung cancer (Li et al. 2020), breast cancer (Liu, Sun, and Liu 2019), Parkinson’s disease (Lee and Cao 2022), brain connectivity network (Zhang et al. 2020).
and the study of phosphorylated proteins and phospholipid components (Castelletti et al. 2020). Because learning a DAG model can be equivalently viewed as a set of variable selection problems when the order of nodes is known (Agrawal, Uhler, and Broderick 2018), learning multiple DAG models with a known order is likewise equivalent to a set of multi-task variable selection problems. However, when the order is not known (which is usually the case in practice), learning the order of nodes from the data can be very challenging. To overcome this issue, we employ MCMC sampling over the permutation space to average over the uncertainty in learning the order of nodes and then compute the DAG model for each given order via the proposed Bayesian multi-task variable selection approach. Simulation studies and a real data example are used to demonstrate the effectiveness of the proposed method.

The rest of this article is organized as follows. In Section 2, we introduce our model for Bayesian multi-task variable selection, prove the high-dimensional posterior consistency and describe the VB algorithm for model-fitting. Section 3 presents simulation results for the multi-task variable selection problem. In Section 4, we generalize our method to joint estimation of multiple DAG models and propose an order MCMC sampler. Simulation studies and real data analysis for differential DAG analysis are presented in Sections 5 and 6, respectively. Section 7 concludes the article with a brief discussion. Proofs, additional simulation results and more details about the algorithm implementation are deferred to the appendices in supplementary materials.

2. Bayesian Multi-Task Variable Selection

2.1. Model, Prior and Posterior Distributions

We introduce some notation to be used throughout the article. Denote the cardinality of a set $S$ by $|S|$. For any $k \in \mathbb{N}$, let $[k] = \{1, 2, \ldots, k\}$, and let $2^{|k|} = \{S : S \subseteq [k]\}$ denote the power set on it; note that $|2^{|k|}| = 2^k$. For any vector $b$ and matrix $A$, let $b_S$ be the subvector of $b$ with index set $S$ and $A_S$ be the submatrix of $A$ containing columns indexed by $S$. Let $\mathbb{1}$ denote the indicator function.

For the multi-task variable selection problem, let $K$ denote the number of datasets we have, which is treated as fixed in this article. We assume the same $p$ covariates are observed in all $K$ datasets. For the $k$th dataset, let $n_k$ denote the sample size, $y^{(k)} \in \mathbb{R}^{n_k}$ the response vector, and $X^{(k)} \in \mathbb{R}^{n_k \times p}$ the design matrix containing $n_k$ observations of the $p$ covariates. Consider the linear regression model

$$y^{(k)} = X^{(k)}\beta^{(k)} + \epsilon^{(k)},$$

where $\epsilon^{(k)} \sim N_n(0, \sigma^2 I_n)$, $\forall k \in [K]$.

(1)

where $N_n$ denotes the $n$-dimensional normal distribution, $I_n$ denotes the $n$-dimensional identity matrix, and the vector of regression coefficients, $\beta^{(k)}$, is assumed to be sparse. For ease of presentation, we assume the error variance $\sigma^2$ is the same across all datasets, but this assumption can be relaxed straightforwardly in the theory and algorithms to be developed in this article. For now, we also assume that $\sigma^2$ is known, and we will explain in Appendix B, supplementary materials how to estimate it in practice.

The main parameter of interest is the set-valued vector $\gamma \in (2^{[K]})^p$, where $\gamma_j = 1$ means that the $j$th covariate has a nonzero regression coefficient (i.e., it is activated) in the $k$th dataset for each $k \in [K]$. For instance, $\gamma_1 = \{1, 2\}$ indicates that the first covariate is activated in both the first and second datasets, and $\gamma_2 = \emptyset$ indicates that the second covariate is deactivated across all datasets. Let $|\gamma| = \sum_{j=1}^p \mathbb{1}_{\{\gamma_j \neq \emptyset\}}$ denote the number of covariates that are activated in at least one dataset, and let $a_k(\gamma) = |\{j \in [p] : \gamma_j = k\}|$ be the number of covariates that are activated in $k$ distinct datasets. Note that $|\gamma| = a_1 + \cdots + a_K$. The main idea behind our construction of the prior on $(\gamma, (\beta^{(k)})_{k=1}^K)$, denoted by $\Pi(\gamma, (\beta^{(k)})_{k=1}^K)$, is similar to the spike-and-slab prior for single-task variable selection. First, given $\gamma$, we assume that $\beta_j^{(k)} = 0$ if $k \notin \gamma_j$ and put a normal prior on $\beta_j^{(k)}$ otherwise. Next, to achieve sparsity, we put a prior on $\gamma$ that favors sparser models. Explicitly, our prior is given by

$$\beta_j^{(k)} | \gamma \sim \mathbb{1}_{\{|\gamma_j| \leq 1\}} \delta_0 + \mathbb{1}_{\{\gamma_j \neq \emptyset\}} N_1(0, \tau_j^{(k)}),$$

$\forall j \in [p], k \in [K],$

$$\Pi(\gamma) \propto \mathbb{1}_{\{|\gamma_j| \leq L\}} f(|\gamma|, L) \prod_{k=1}^K P^{-a_k}(\gamma),$$

(2)

(3)

where $L \in \mathbb{N}$, $\tau_j^{(k)} > 0$ for $j \in [p], k \in [K]$ and $\omega_k > 0$ for $k \in [K]$ are hyperparameters, and $\delta_0$ denotes the Dirac measure at 0. The function $f(|\gamma|, L)$ is introduced for generality, and in our theoretical analysis it will be assumed to be “asymptotically negligible” compared to the product term in (3). Hence, the sparsity is mainly promoted by the hard threshold $L$, which is the maximum number of activated covariates (in at least one dataset) we allow, and the hyperparameters $(\omega_k)_{k=1}^K$. We can view $\omega_k$ as the “cost” we pay for activating one covariate simultaneously in $k$ datasets.

For most multi-task variable selection problems in reality, it is reasonable to assume that activated covariates tend to be shared across datasets, and to reflect this prior belief, we propose to choose $(\omega_k)_{k=1}^K$ such that

$$\frac{\omega_k}{K} < \frac{\omega_{k-1}}{K-1} < \frac{\omega_{k-2}}{K-2} < \cdots < \frac{\omega_2}{2} < \omega_1.$$  

(4)

To see the reasoning behind (4), consider the case $K = 2$ where the above condition is reduced to $\omega_2 < 2\omega_1$. Suppose that the first two covariates are identical in both datasets, and consider two models $\gamma, \gamma'$ such that $\gamma_1 = \{1\}, \gamma_2 = \{2\}, \gamma'_1 = \{1, 2\}$, $\gamma'_2 = \emptyset$ and $\gamma_j = \gamma'_j = \emptyset$ for any $j > 2$. Then, $\gamma, \gamma'$ have the same marginal likelihood, but $a_1(\gamma) = 2, a_2(\gamma) = 0$ and $a_1(\gamma') = 2, a_2(\gamma') = 1$. It can be seen that $\omega_2 < 2\omega_1$ ensures we favor $\gamma'$. An analogous argument for the general case with $K \geq 2$ leads to (4). Note that the choice of $\omega_1, \ldots, \omega_K$ only reflects the experimenter’s prior belief on $\gamma$, and one can even use $\omega_k \ll \omega_1$ for all $K \geq 2$ if prior information reveals that the majority of activated covariates must be shared in multiple datasets. However, in all of our numerical studies, we only use $(\omega_k)_{k=1}^K$ such that (4) is satisfied and $\omega_1 \leq \omega_2 \leq \cdots \leq \omega_K$, the latter of which appears to be a natural condition in situations where not much prior information is available. We will refer to the model specified by (1)–(3) as muSSVS (multi-task Spike-and-Slab Variable Selection).
2.2. Posterior Consistency for Multi-Task Spike-and-Slab Variable Selection

In this section, we prove the posterior consistency for the mSSVS model, which generalizes the existing results for single-task variable selection (Johnson and Rossell 2012; Narisetty and He 2014; Yang, Wainwright, and Jordan 2016; Jeong and Ghosal 2021). We only consider in our proof the special case \(n_k = n\) and \(t_{j}^{(k)} = \tau\) for \(k \in [K]\) and \(j \in [p]\). Analogous arguments can be used to prove the posterior consistency in the more general case where \((t_{j}^{(k)})_{k \in [K], j \in [p]}\) are bounded and \(n_1, \ldots, n_K\) are different with \(\min_{k \in [K]} n_k\) being sufficiently large.

Suppose the data is generated by (1) with \(\beta^{(k)}_{\text{true}}\) being the vector of true regression coefficients for the \(k\)th dataset. Our goal is to show that covariates with a relatively high signal strength (aggregated over multiple datasets) can be recovered with high probability. To this end, define the “true” model \(\gamma^*\) as follows. Let \(C_{B,1}, \ldots, C_{B,K}\) be constants that depend on \(n, p, \sigma^2, \) and \(\tau\). For each \(j \in [p]\), define

\[
m_j^* = \max \left\{ m \in [K]: \{k \in [K]: (\beta_{j}^{(k)*})^2 \geq C_{B, m}\} = m \right\},
\]

and set \(\gamma^* = \{k \in [K]: (\beta_{j}^{(k)*})^2 \geq C_{B, m^*}\}\). If \(k \in \gamma^*_j\), we say the \(j\)th covariate is “influential” in the \(k\)th dataset (a “non-influential” covariate may have a small but nonzero regression coefficient). In words, \(C_{B, k}\) can be seen as the detection threshold for covariates that have relatively large nonzero regression coefficients in \(k\) distinct datasets. For our posterior consistency result, we will assume that \(C_{B, 1} > \cdots > C_{B, K}\), which reflects the advantage of multi-task learning: if a covariate is activated in more datasets, the signal size in each data set required for detection can be smaller.

We assume the following five conditions hold for \(k = 1, \ldots, K\), which were also used in the consistency analysis for single-task variable selection conducted in Yang, Wainwright, and Jordan (2016). However, since we use an independent normal prior on the nonzero entries of \(\beta^{(k)}\) while Yang, Wainwright, and Jordan (2016) considered the g-prior (which significantly simplifies the calculation), some of our conditions are slightly more stringent. We use

\[
S_k(\gamma^*) = \{j \in [p]: k \in \gamma^*_j\}
\]

to denote the set of covariates that are activated in the \(k\)th dataset, and we simply denote the set of truly influential covariates by \(S_k^* = S_k(\gamma^*)\).

(1) The first condition is on the true regression coefficients \(\beta^{(k)*}\).

(1a) For some \(B_1 \geq 1\), \(\frac{1}{n} \|X^{(k)} \beta^{(k)*}\|_2^2 \leq B_1 \sigma^2 \log p\).

(1b) For some \(B_2 \geq 0\), \(\frac{1}{n} \|X^{(k)} \beta^{(k)*}\|_2^2 \leq B_2 \sigma^2 \log p\).

Condition (1a) requires that the order of the total signal size in each dataset, \(\|X^{(k)} \beta^{(k)*}\|_2^2\), is at most \(n \log p\), and Condition (1b) requires that non-influential covariates cannot contribute significantly to the variation in \(\gamma^{(k)}\). Both are reasonable assumptions for most high-dimensional problems. If one assumes all nonzero entries of \(\beta^{(k)*}\) are sufficiently large in absolute value, then \(\beta^{(k)*}_{|S_k^*|} = 0\) and Condition (1b) holds trivially. If one further assumes the influential covariates have bounded regression coefficients (i.e., coefficients do not grow with \(n\)), Condition (1a) allows each dataset to have \(O(\log p)\) independent influential covariates, which is not restrictive when \(p \gg n\). More discussion on Condition (1a) will be given after Condition (5).

(2) The second condition is on the design matrix. For any symmetric matrix \(A\), denote its smallest eigenvalue by \(\lambda_{\min}(A)\).

(2a) \(\|X^{(k)}\|_2^2 = n\) for all \(j = 1, \ldots, p\).

(2b) For some \(\nu \in (0, 1)\), \(\min_{|S| \leq L} \lambda_{\min}\left(\frac{1}{n} X^{(k)} X^{(k)\top}\right) \geq \nu^2\).

(2c) Let \(Z \sim \mathcal{N}_n(0,I)\). For some \(B_3 \geq 8/\nu^2\), we have

\[
\frac{1}{\sqrt{n}} E_Z \max_{|S| \leq L, j \in S} \left| X^{(k)} (I - \Psi^{(k)}_S) X^{(k)}_j \right| \\
\leq \frac{1}{2} B_3 \nu^2 \log p,
\]

where \(\Psi^{(k)}_S = X^{(k)} (X^{(k)}_S)^\top X^{(k)}_S\) is the projection matrix.

Condition (2a) assumes all columns of \(X^{(k)}\) are normalized and is used to simplify the calculation. Condition (2b) is known as the lower restricted eigenvalue condition and is a modest constraint necessary for theoretical analysis of Bayesian variable selection problems (Narisetty and He 2014). Condition (2c) is called the sparse projection condition (Yang, Wainwright, and Jordan 2016). Since Condition (2a) ensures that \(\|I - \Psi^{(k)}_S X^{(k)}_j\|_2 \leq \sqrt{n}\) for all \(k \in [K]\), \(|S| \leq L\) and \(j \in [p]\), one can use a standard inequality for maximum of Gaussian random variables to show that Condition (2c) always holds for some \(B_3 = O(L \nu^{-1})\). But when the design matrix consists of independent covariates, \(B_3\) can be much smaller; see Yang, Wainwright, and Jordan (2016) for more details.

(3) The third condition is on the choice of prior hyperparameters. Let \(\tilde{\tau} = \tau / \sigma^2\), and \(C\) denote some universal constant (i.e., a constant that does not depend on \(n\)).

(3a) \(1 + n \tilde{\tau} \leq C \rho^{\eta}\) for some \(\eta > 0\).

(3b) \(L \leq C \rho^\frac{\eta}{1}\) for some \(\tilde{\eta} \in (0, 1)\).

(3c) \(\omega_k^k \geq 1\) satisfies (4) and \(\omega_k^k \geq \frac{1}{2} \left( \frac{B_1}{\nu^2} + B_2 + B_3 \right) + \tilde{\eta} + 2\).

(3d) The function \(f\) in (3) satisfies \(1 \leq f(s + 1, L) \leq L\) for every \(s \in \mathbb{N}\).

Condition (3a) is only used to bound a determinant term in the posterior distribution of \(\gamma^*\). In high-dimensional settings with \(n \ll p\), both Conditions (3a) and (3b) are very natural and easy to satisfy. Condition (3c) requires the parameter \(\omega_k\) to be sufficiently large, which is needed to ensure that the posterior mass concentrates on sparse models. Condition (3d) implies that \(f(\gamma^*) \mathbb{I}_{|L|} \leq L \mathbb{I}_{|L|}\). By Condition (3c), we have \(\omega_k > 2K \geq 2\), and thus the product term in (3) is at most \(p^{-2K}\). Since \(L = o(p)\) by Condition (3b), we see that the magnitude of \(\Pi(\gamma^*)\) depends little on the function \(f(\gamma^*)\).

(4) The true sparsity level \(|S_k^*|\) satisfies \(\max \{1, |S_k^*|\} \leq \frac{n}{\nu^2 \log p}\).

(5) The constant \(C_{B, k}\) is given by

\[
C_{B, k} = \left\lfloor \left( \frac{\omega_k}{8} + 2 + \eta \right) \left( \frac{B_1}{\nu^2} \right) \sigma^2 \log p \right\rfloor.
\]
Condition (5) is known as the beta-min condition (Yang, Wainwright, and Jordan 2016). By inequality (4), it further implies that $C_{β,k} < C_{β,k-1} < \cdots < C_{β,1}$; that is, the more datasets in which the covariate is influential, the lower the signal strength level required to detect it. To gain further insights into this condition, consider the case where $η, B_1, ν, \tilde{τ}, σ^{-2}$ are all universal constants. Then, the order of $C_{β,k}$ is given by $\frac{ω_k log p}{k}$, which typically goes to zero in the high-dimensional asymptotic regimes considered in the literature, implying that we can identify activated covariates with diminishing signal sizes. Note that Conditions (1a) and (5) are compatible with each other. For example, assuming $ω_k/k$ is a constant, to satisfy Condition (5), all entries of $(β(k)^*)^2$ corresponding to influential covariates only need to have order $n^{-1} log p$; in this case, we have $\|X(β)^*\|_2^2 = O(|S^*| log p)$, which is much smaller than the order $n log p$ required by Condition (1a).

**Theorem 1.** Suppose for each $k$, $y(k)$ is generated by (1) with $β(k) = β(k)^*$. If Conditions (1) to (5) hold, we have

$$P \left( \prod (y^* | (y(k))_{k \in [K]}) \geq 1 - c_1 p^{-1} \right) \geq 1 - c_2 p^{-c_3},$$

where $\prod(· | (y(k))_{k \in [K]})$ denotes the posterior measure for the model specified by (1)–(3), $P$ denotes the probability measure for the true data-generating process, and $c_1$, $c_2$, and $c_3$ are positive universal constants.

**Proof.** We defer the proof to Appendix A, supplementary materials.  

**Remark 1.** The main difference between Theorem 1 and existing consistency results for single-task spike-and-slab variable selection (Narisetty and He 2014; Yang, Wainwright, and Jordan 2016) is that the detection threshold $C_{β,k}$ in our Condition (5) depends on $k$. When (4) holds, $C_{β,k}$ is smaller for larger $k$, which means that by combining information from multiple datasets and properly choosing $(ω_k)^K_{k=1}$ (see Condition (3c)), we can detect activated covariates with smaller signal sizes. This rigorously justifies the advantage of multi-task variable selection over separate analysis.

### 2.3. Multi-Task Sum of Single Effects Model

For Bayesian problems, posterior distributions are typically calculated through Markov chain Monte Carlo (MCMC) sampling. But in our case, the huge discrete model space can make the sampling converge very slowly. In this section, we approximate our muSSVS model by a multi-task sum of single effects (muSuSiE) model, generalizing the recently developed sum of single effects (SuSiE) model of Wang et al. (2020a) for single-task variable selection. The muSuSiE model assumes that for each $k \in [K]$, $y(k) ~ N_{\pi_k}(X(β)^{(k)}, σ^2 I_{\pi_k})$, where $β(k) = \sum_{l=1}^L β^{(k,l)}$, and each $β^{(k,l)} \in \mathbb{R}^p$ has at most one nonzero entry; that is, we decompose each $β(k)$ into a “sum of single effects.” We will call each $β^{(k,l)}$ a single-effect regression coefficient vector. Similarly, we introduce $L$ set-valued single-effect selection vectors $y^{(1)}, \ldots, y^{(L)}$ such that $y^{(l)} \in \{1\}$ means that $β^{(k,l)}$ is nonzero for each $k \in [L]$ (i.e., covariate $j$ is the $l$th single effect and is activated in the datasets indexed by $I_l$). Let $χ$ denote a probability distribution on $2^{[K]} \setminus \emptyset$ and Unif([p]) denote the uniform distribution on [p]. The prior distribution we put on $\{y^{(l)} : l \in [L]\}$ encodes the following procedure for selecting and activating covariates: for each $l \in [L]$, we draw $χ_l \sim \text{Bernoulli}(π_τ)$, $u_l \sim \text{Unif([p])}$, and $I_l \sim χ$; if $χ_l = 1$, we activate the $u_l$th covariate in the datasets indexed by $I_l$, and we do nothing if $χ_l = 0$. So $χ_l$ indicates whether the $l$th single effect is indeed activated. For each activated covariate in each dataset, we still use a normal prior distribution on its effect size as in (2). Note that we assume $u_1, \ldots, u_L$ are generated independently and thus a covariate can be activated multiple times, which is the key difference between muSuSiE and muSSVS.

Formally, the prior distribution of muSuSiE can be expressed as follows:

\[
\begin{align*}
\pi_1 \sim \text{Uniform([p])}, & \quad \forall l \in [L], \\
\zeta_l \sim \text{Bernoulli}(π_τ), & \quad \forall l \in [L], \\
y^{(l)} | (u_l, \zeta_l) \sim (1 - \zeta_l) \mathbb{1}_{(u_l=0)} δ_0 + \zeta_l \mathbb{1}_{(u_l=1)} X, & \quad \forall j \in [p], l \in [L], \\
β^{(l,k)} | (y^{(l)})_{l \in [L]} & \sim \mathbb{1}_{(k \notin y^{(l)}_{\{j\}})} δ_0 + \mathbb{1}_{(k \in y^{(l)}_{\{j\}})} N_1(0, r^{(k,l)}_{\{j\}}), & \forall j \in [p], k \in [K], l \in [L], \\
\end{align*}
\]

where $(r^{(k,l)}_{\{j\}})_{k,l}$ are hyperparameters and $δ_0$ denotes the Dirac measure that assigns unit probability mass to the empty set. Though in (6) we write $β(k)$ as the sum of $L$ terms, the actual sparsity is controlled by the hyperparameter $π_τ$. Each $y^{(l)}$ has zero (if $χ_l = 0$) or one (if $χ_l = 1$) covariate activated.

We now discuss how to choose the probability distribution $χ$. We introduce hyperparameters $π_1 > π_2 > \cdots > π_K > 0$ and set $χ(\{l : χ_l = 1\}) = p \pi_{|l|}$, $χ(\{l : χ_l = 0\}) = 1 - p \pi_{|l|}$, $χ(l = 0) = 1 - p \pi_{|l|}$. Assume $π_1, \ldots, π_K$ are normalized so that $χ(2^{[K]} \setminus \emptyset) = 1$. Let $s_{\pi} = \{l : χ_l = 1\}$ denote the number of activated single effects, $(y^{(l)}_{\{j\}})_{l \in s_{\pi}}$ denote the unordered set of activated single-effect selection vectors, and $l_{\pi}$ denote the value of $y^{(l)}_{\{j\}}$. Note that
\( \{ \mathbf{y}^{(l)} : \zeta_l = 1 \} \) is completely determined by \( \{(u_l, I_l)\}_{l \in [L]} \), since we always have \( \gamma_j^{(l)} = 0 \) for any \( j \neq u_l \). Let \( \hat{I} \) denote the probability measure under the muSuSiE model given by (7). If no covariate is activated more than once (i.e., for any \( l \neq l' \) such that \( \zeta_l = \zeta_{l'} = 1 \), we have \( u_l \neq u_l' \),

\[
\hat{I}(\{ \mathbf{y}^{(l)} : \zeta_l = 1 \}) = f(s_l, L) \prod_{l=1}^{L} \pi_{\zeta}(1 - \pi_{\zeta})^{1 - 0} \pi_{\zeta}^{0}, \tag{8}
\]

where \( f(s, L) = L \times (L - 1) \times \cdots \times (L - s + 1) \) satisfies Condition (3d). A straightforward calculation shows that (8) and (3) are equivalent if

\[
\frac{\pi_\zeta \pi_k}{1 - \pi_\zeta} = p^{-\text{eq}}, \tag{9}
\]

for each \( k \in [K] \). This shows why muSuSiE is an approximation to the muSSVS model. Again, the two models are not equivalent because we may have \( u_l = u_{l'} \) for some \( l \neq l' \) in (7), though this happens with very small probability when \( p \) is large. While the repeated activation of a covariate may seem artificial and slightly unnatural, this feature enables us to propose an efficient VB method (to be introduced in the next section) which can quickly yield an approximate Bayesian solution to the multi-task variable selection problem.

\textbf{Remark 2.} While muSuSiE is based on the SuSiE model proposed by Wang et al. (2020a) for single-task variable selection, our model (7) with \( K = 1 \) still differs from SuSiE in that we use Bernoulli random variables \( \zeta_1, \ldots, \zeta_L \) to control the actual sparsity of \( \{ \mathbf{b}^{(k)} \}_{k \in [K]} \). The prior distribution used in model (7) assumes that the number of activated covariates (including duplicates) follows Binomial(\( L, \pi_{\zeta} \)), and given a sufficiently large sample size, the model (7) is able to learn the actual number of activated covariates, which can range from 0 to \( L \). This also implies that an increase in the value of \( L \) is not likely to have a significant impact on the posterior distribution. In contrast, SuSiE assumes there are exactly \( L \) activated single effects and relies on an ad-hoc procedure to determine which covariates are truly activated from the output of a VB algorithm.

### 2.4. Iterative Bayesian Stepwise Selection for Fitting muSuSiE

We propose an iterative Bayesian stepwise selection (IBSS) method for fitting the model given in (7) by generalizing the IBSS algorithm of Wang et al. (2020a). The main idea is to iteratively find \( \hat{I} \) for \( l = 1, \ldots, L \) in the muSuSiE model by conditioning on the other \( L - 1 \) single effects. The starting point for our algorithm is the muSuSiE model with \( L = 1 \), which we will refer to as the "multi-task single-effect regression" (muSER) model and we recall below with superscript \( l \) dropped:

\[
\mathbf{y}^{(k)} \sim \mathcal{N}_{\mathbf{y}_k} (\mathbf{X}^{(k)} \mathbf{b}^{(k)}, \sigma^2 \mathbf{I}_{\mathbf{y}_k}), \quad \forall k \in [K],
\]

\[
u \overset{\text{ind}}{\sim} \text{Uniform}([p]),
\]

\[
\zeta \overset{\text{ind}}{\sim} \text{Bernoulli}(\pi_{\zeta}),
\]

\[
\gamma_j | (u, \zeta) \overset{\text{ind}}{\sim} (1 - \zeta) \mathcal{U}_{[u_{(j)}]} \delta_0 + \zeta \mathcal{U}_{[u_{(j)}]} \mathbf{X}, \quad \forall j \in [p],
\]

\[
\beta_j^{(k)} | \mathbf{y} \overset{\text{ind}}{\sim} \mathcal{U}_{[k \notin \gamma_j]} \delta_0 + 1_{(k \in \gamma_j)} \mathcal{N}_{1}(0, \tau), \quad \forall j \in [p], k \in [K],
\]

Since we only allow at most one covariate to be activated in (10), the joint posterior distribution of \( (\mathbf{y}, 1 - \zeta) \) given \( \sigma^2 \) and \( \tau \) can be quickly calculated, which is given by a multinomial distribution with

\[
\begin{align*}
\Pi_{\text{muSER}}(\zeta = 0 | (\mathbf{y}^{(k)})_{k \in [K]}) &= \alpha_0, \\
\Pi_{\text{muSER}}(\gamma_j | (\mathbf{y}^{(k)})_{k \in [K]}) &= \alpha_{j,k},
\end{align*}
\]

where expressions for \( \alpha_{j,k} \) and \( \alpha_0 \) are given in Appendix B, supplementary materials. By definition, \( \alpha_0 + \sum_{j \in [p]} \sum_{l \neq 0} \alpha_{j,l} = 1 \). Further, the posterior distribution of \( \beta_j^{(k)} \) given \( \zeta = 1, u = j, k \in \gamma_j \) (i.e., the \( j \)th covariate is activated in the \( k \)th dataset) is

\[
\beta_j^{(k)} | (y^{(k)})_{k \in [K]}, \sigma^2, \tau, \zeta = 1, u = j, k \in \gamma_j \sim \mathcal{N}(\mu_j^{(k)}, \phi_j^{(k)}),
\]

where we defer the explicit expressions for \( \mu_j^{(k)} \) and \( \phi_j^{(k)} \) to Appendix B, supplementary materials. (Note that whenever \( \zeta = 0, u \neq j \) or \( k \notin \gamma_j \), the posterior distribution of \( \beta_j^{(k)} \) is \( \delta_0 \).

For ease of notation, we introduce a function, \( f_{\text{muSER}} \), which returns the posterior distribution for \( \beta \) under the muSER model. Since this posterior distribution is determined by the values of \( \alpha_0, \alpha = (\alpha_{j,l})_{l \in [p], j \in [K]}, \mathbf{m}(k) = (\mu_1^{(k)}, \ldots, \mu_p^{(k)}) \) and \( \phi = (\phi_1^{(k)}, \ldots, \phi_p^{(k)}) \) for \( k = 1, \ldots, K \), we define \( f_{\text{muSER}} \) by

\[
f_{\text{muSER}}((\mathbf{y}^{(k)})_{k \in [K]}; \sigma^2, \tau) := (\alpha, \alpha_0, (\mathbf{m}(k))_{k \in [K]}, (\phi(k))_{k \in [K]}) \quad \text{for each } K.
\]

Observe that for the muSuSiE model, if \( \beta^{(k,l')} : l' \in [L] \) and \( l' \neq l \) is given, calculating the posterior distribution of \( \beta^{(k,l)} \) is very straightforward: one just needs to fit the muSER model by substituting the residual \( y^{(k)} - \mathbf{X}^{(k)} \mathbf{b}^{(k)} \) for the response \( y^{(k)} \) for each \( k \) in the muSER model (10). This suggests an iterative strategy for fitting muSuSiE, which we detail in Algorithm 1. The implementation of our algorithm is analogous to the IBSS algorithm for the original SuSiE model.

Let \( \hat{\beta}^{(k,l)} \) be as given in the output of Algorithm 1, which denotes the estimated \( l \)th single-effect regression coefficient vector for the \( k \)th dataset. We can express the posterior mean regression coefficient vector for the \( k \)th dataset by

\[
\hat{\beta}^{(k,l)} = \frac{L}{l = 1} \sum \hat{\beta}^{(k,l)}, \tag{12}
\]

Further, taking all \( L \) single-effect selection vectors into account, we can approximate the probability that the \( j \)th covariate is activated in the \( k \)th dataset by

\[
r_j^{(k,l)} = 1 - \prod_{l=1}^{L} \left( 1 - r_j^{(k,l)} \right), \quad \text{where } r_j^{(k,l)} := \sum_{l' : k \notin l'} \alpha_{j,l'}. \tag{13}
\]

is the probability that the \( j \)th coordinate is activated in the \( k \)th dataset in the \( L \)th single-effect model, conditioning on the other \( L - 1 \) single effects.

By an argument similar to that in Wang et al. (2020a), we can show that this IBSS algorithm coincides with the coordinate ascent variational inference (CAVI) algorithm (Blei, Kucukelbir, and McAuliffe 2017 for maximizing the evidence lower bound over a particular variational family for the muSuSiE model; see Appendix B, supplementary materials, where we also explain how to choose the stopping criterion and estimate \( \sigma^2 \) and \( \tau \) empirically in Algorithm 1.)
Algorithm 1: Iterative Bayesian stepwise selection (IBSS) for fitting muSuSiE

Require: data \( \{X^{(k)}\}_{k=1}^{K}, \{y^{(k)}\}_{k=1}^{K} \), number of single effects \( L \)

Require: a function \( \text{fmuSER} \) which is defined in (11)

initialize posterior means \( \hat{\beta}^{(k,l)} = 0 \) for \( l = 1, \ldots, L \) and \( k = 1, \ldots, K \)

initialize \( \hat{\sigma}^2 \) and \( (\tau^{(l)})_{l=1}^{L} \)

if the stopping criterion is not satisfied then

for \( l = 1, \ldots, L \) do

for \( k = 1, \ldots, K \) do

\( \hat{y}^{(k,l)} \leftarrow y^{(k)} - X^{(k)} \sum_{l' \neq l} \hat{\beta}^{(k,l')} \)

end for

estimate \( \tau^{(l)} \) by maximizing (B.2) in Appendix B.1

\( (a^{(l)}, \phi^{(l)}, (\mu^{(k,l)})_{k=1}^{K}, (\phi^{(k,l)})_{k=1}^{K}) \leftarrow \text{fmuSER}((\gamma^{(k,l)})_{k \in [K]}; \hat{\sigma}^2, \tau^{(l)}) \), for \( k = 1, \ldots, K \) do

for \( j = 1, \ldots, p \) do

\( \hat{\beta}^{(k,l)}_j \leftarrow \mu_j^{(k,l)} \sum_{l' \in [L]} \alpha_j^{(l',k)} \)

end for

end for

end for

update \( \hat{\sigma}^2 \) by (B.8) in Appendix B.2

end if

return \( \hat{\sigma}^2, (a^{(l)})_{l=1}^{L}, (\hat{\beta}^{(k,l)})_{l \in [L], k \in [K]} \)

Remark 3. We can also implement the VB algorithm for the model proposed in Section 2 by generalizing VB methods for single-task variable selection (Carbonetto and Stephens 2012; Huang, Wang, and Liang 2016; Ormerod, You, and Müller 2017; Ray and Szabó 2022). However, a key advantage of the IBSS algorithm for SuSiE/muSuSiE is that, in addition to being fast, it does not use a variational family that assumes independence among \( \gamma_1, \ldots, \gamma_p \) (in single-task variable selection, \( \gamma \) indicates whether the \( j \)th covariate is activated), which is particularly important for high-dimensional applications where high collinearity is expected. We refer readers to Wang et al. (2020a) for more discussion on why this “sum of single effects” representation can effectively overcome collinearity and the advantage of IBSS over deterministic search algorithms that return a single best model.

3. Simulation Studies for Bayesian Multi-Task Variable Selection

We conduct simulation studies to illustrate the benefits of performing variable selection for multiple datasets jointly rather than independently. We generate datasets according to (1) using the same \( \sigma^2 \) for all \( K \) datasets. For the true model, we consider two types of activated covariates. For the first type, each covariate is activated in all \( K \) datasets. We denote the set of these covariates by \( S^*_{\text{com}} \) and let \( s^*_1 = |S^*_{\text{com}}| \) (subscript “com” means “common”). For the second type, each covariate is activated in only one dataset. We choose some \( s^*_2 > 0 \) and draw \( s^*_2 \) covariates of the second type for each dataset; denote the set of covariates that are only activated in the \( k \)th dataset by \( S^*_{\text{pri},k} \) (subscript “pri” means “private”). The true model size is given by \( s^* = s^*_1 + K s^*_2 \), and \( S^*_k = S^*_{\text{com}} \cup S^*_{\text{pri},k} \) is the true set of activated covariates for the \( k \)th dataset. For each activated covariate, we sample its regression coefficient \( \beta_j^{(k)} \) independently from the normal distribution \( \mathcal{N}(0, 0.6^2) \). For the design matrix, we sample each entry of \( X^{(k)} \in \mathbb{R}^{n \times p} \) from the standard normal distribution. Finally, we generate the response data by drawing \( y^{(k)} \sim \mathcal{N}(X^{(k)} \beta^{(k)}, \sigma^2 I) \).

After generating the dataset \( \{(X^{(k)}, y^{(k)})\}_{k=1}^{K} \), we run the IBSS algorithm to fit the muSuSiE model, which does variable selection simultaneously for \( K \) datasets. For comparison, we also fit the SuSiE model using the algorithm of Wang et al. (2020a) for each dataset separately. We will refer to the former as the multi-task method and the latter as the separate single-task analysis. When running simulations, we set \( L = s^* + K \) for the multi-task method and \( L = s^*_1 + s^*_2 + 1 \) for the separate analysis method. We have also tried other values of \( L \) and observed that as long as \( L \) is larger than the true number of activated covariates, its choice has negligible effect on the estimates; the reason was explained in Remark 2. For the hyperparameter \( \pi \) in the muSuSiE model, we set it by (9), and thus it suffices to specify \( \omega_1, \ldots, \omega_K \). When \( K = 2 \), we use \( p^{-\omega_1} = p^{-1.1/2} \) and \( p^{-\omega_2} = p^{-1.25} \); when \( K = 5 \), we use \( \omega_k = 1.25 + 0.15k \) for each \( k \). Additionally, we tried joint Markov chain Monte Carlo (MCMC), separate MCMC, and LASSO methods, for which the results and implementation details are deferred to Appendix C, supplementary materials.

For the multi-task method, recall that the probability of the \( j \)th covariate being activated in the \( k \)th dataset, \( r_j^{(k)} \), is defined in (13). Setting the threshold to 0.5, we define the selected activated covariates from our multi-task method by \( S_{\text{mult}} = \{ j : r_j^{(k)} \geq 0.5 \} \) (subscript “mu” means “multi-task”). For the standard SuSiE method, we use the susie function from the susieR package (Wang et al. 2020a) to find the set of activated covariates, which we denote by \( S_{\text{ull}} \) (subscript “si” means “single-task”). To compare the performance of two approaches, we calculate the sensitivity (sens) and precision (prec) by sens(\( S_k \)) = \( \frac{|S_k \cap S^*_k|}{|S^*_k|} \), prec(\( S_k \)) = \( \frac{|S_k \cap S^*_k|}{|S_k|} \), where we let \( S_k = S_{\text{mult}} \) for the multi-task method and \( S_k = S_{\text{ull}} \) for the single-task approach.

Table 1 shows the simulation results for \( \sigma^2 = 1 \) and \( K = 2 \). We consider two scenarios: one with \( p = 600 \) and \( n = 100 \), and the other with \( p = 1000 \) and \( n = 500 \). From Table 1, we observe that when the sample size is small (\( n = 100 \)), the multi-task method identifies more activated covariates than the single-task approach, resulting in higher sensitivity and precision. When the sample size is increased to 500, the multi-task method still improves the sensitivity but has a slightly smaller precision, because the multi-task method tends to treat the covariates with a very strong signal strength in only one dataset as simultaneously activated in two datasets. Nevertheless, considering the significant improvement in sensitivity, the overall performance of the multi-task method seems much better. To further examine this phenomenon, we plot the sensitivity and specificity for \( |S^*_k| = 10 \) and \( |S^*_k| = 2 \) in Appendix C.1, supplementary materials; all other settings yield similar plots.

The simulation results for \( \sigma^2 = 1 \) and \( K = 5 \) are shown in Table C.1 in Appendix C.1, supplementary materials. It is worth noting that when the sample size is small, compared with the case \( K = 2 \), the advantage of the multi-task method with
Table 1. Simulation results for two data sets with $\sigma = 1$.

| $p$ | $n$ | $s_1^2$ | $s_2^2$ | sens_mu | sens_si | prec_mu | prec_si |
|-----|-----|---------|---------|---------|---------|---------|---------|
| 600 | 100 | 10     | 2       | 0.4526  | 0.2632  | 0.9884  | 0.9365  |
| 600 | 100 | 10     | 5       | 0.3456  | 0.2045  | 0.9747  | 0.9258  |
| 1000| 500 | 10     | 2       | 0.8121  | 0.7073  | 0.9862  | 1       |
| 1000| 500 | 10     | 5       | 0.7905  | 0.7011  | 0.9928  | 0.9996  |
| 1000| 500 | 25     | 2       | 0.8191  | 0.6963  | 0.9962  | 1       |
| 1000| 500 | 25     | 5       | 0.804   | 0.6949  | 0.9964  | 0.9999  |

NOTE: For each setting, the result is averaged over 500 replicates.

$K = 5$ becomes much more significant and it outperforms the single-task method significantly in terms of both sensitivity and precision. When the sample size is large, the multi-task method is still better than the single-task method, but the performance is similar to that for $K = 2$. The simulation results for $\sigma^2 = 4$ (which represents a higher noise level) are shown in Appendix C.1, supplementary materials, where we have made very similar observations for the behavior of the two methods.

In Appendix C.2, supplementary materials, we show the average computation time of the multi-task and separate single-task methods for each setting across 500 replicates. The two methods take a similar amount of time when $K = 2$. However, as $K$ increases to 5, the multi-task method takes more time than the separate analysis. For the latter, the time increases linearly with respect to $K$, while the computational time of mmSuSiE increases exponentially. Additionally, when the number of individually activated covariates is small ($|s_2^2| = 2$), the multi-task method is significantly faster than in the case with $|s_2^2| = 5$. The stability of our algorithm with respect to the choice of $\omega$ is discussed in Appendix C.3, supplementary materials.

### 4. Differential DAGs Analysis via Multi-Task Variable Selection

#### 4.1. From Multi-Task Variable Selection to Joint Estimation of Multiple DAG models

A highly useful application of the proposed Bayesian multi-task variable selection method is that it can be naturally extended to the multi-task structure learning problem, that is, joint estimation of multiple DAG models. The existing Bayesian literature on the statistical learning of multiple DAG models mostly focuses on directed graphical models; see, for example, Danaheer, Wang, and Witten (2014), Peterson, Stingo, and Vannucci (2015), Goncalves, Von Zuben, and Banerjee (2016), Niu, Sun, and Sun (2018), Peterson et al. (2020), Shaddox et al. (2020), and Peterson and Stingo (2021). For the learning of multiple DAG models, Oyen and Lane (2012) proposed a greedy search algorithm, Yajima et al. (2015) devised a MCMC sampler generalizing the method of Fronk and Giudici (2004), and Lee and Cao (2022) proposed a method based on the joint empirical sparse Cholesky (JESC) prior. Castelletti et al. (2020) developed the Bayesian methodology and MCMC algorithm for learning multiple essential graphs. For frequentists’ approaches, Liu, Sun, and Liu (2019) proposed the MPenPC method, a two-stage approach based on the PC-stable algorithm, Chen et al. (2021) proposed an iterative constrained optimization algorithm for calculating an $\ell^1/\ell^2$-regularized maximum likelihood estima-
tor, Wang, Segarra, and Uhler (2020b) extended the well-known greedy equivalence search (GES) algorithm of Chickering (2002) to the case of multiple DAGs, and Ghoshal, Bello, and Honorio (2019) offered an algorithm that learns the difference between DAGs efficiently but seems only applicable to the case $K = 2$. The method we will propose in this section is motivated by the observation that once the order of variables is given, the IBSS algorithm for multi-task variable selection can be applied to quickly learn multiple DAG models simultaneously. Hence, all we need is just to combine IBSS with an MCMC sampler that traverses the order space. Compared with frequentists’ methods, our algorithm can quantify the learning uncertainty since the estimators are averaged over the posterior distribution.

Consider learning the DAG model for a single data set first. Let $G = (V, E)$ be a DAG with vertices $V = \{1, \ldots, p\}$ and set of directed edges $E \subseteq V \times V$. Let $|G|$ denote the cardinality of the edge set $E$. Let $B \in \mathbb{R}^{p \times p}$ be the weighted adjacency matrix of the DAG $G$ such that $B_{ij} \neq 0$ if and only if $(i, j) \in E$. Suppose that the observed data matrix, denoted by $X \in \mathbb{R}^{n \times p}$, is generated by the following linear structural equation model (SEM),

$$X_j = \sum_{i=1}^{p} B_{ij} X_i + e_j, \quad \text{for } j = 1, \ldots, p,$$

where $X_j$ denotes the $j$th column of $X$, and for each $j$, the error vector $e_j$ independently follows $N_0(0, \sigma_j^2 I)$. That is, each row of $X$ is an iid copy of a random vector $X = (X_1, \ldots, X_p)$, whose distribution is given by $X = B^T X + e$ with $e \sim N_p(0, \text{diag}(\sigma_1^2, \ldots, \sigma_p^2))$.

Since $G$ is acyclic, there exists at least one permutation (i.e., order) $\prec \in S_p$ such that $B_{ij} = 0$ for any $j < i$ (i.e., $j$ precedes $i$ in the permutation $\prec$), where $S_p$ is the symmetric group of order $p$. Hence, if the rows and columns of $B$ are permuted according to $\prec$, the resulting matrix is strictly upper triangular. To determine which entries in $B$ are not zero, we can convert this problem to $p$ variable selection problems. If we know that the DAG is consistent with the order $\prec$, for each $j$, we only need to identify the parent nodes for $j$ from the set $\{i \in [p]: i < j\}$, which can be seen as a variable selection problem with response variable $X_j$ and candidate explanatory variables $\{X_i: i < j\}$. Combining the results for all $p$ variable selection problems, we get an estimate for the DAG model underlying the distribution of $X$. Unfortunately, the true order of nodes is usually unknown in practice and needs to be learned from the data. Since the order space $S_p$ has cardinality $p!$, searching over $S_p$ can be very time consuming, which is one major challenge in structure learning. To overcome this, various order MCMC methods have been proposed in the literature for efficiently generating samples from posterior distributions defined on $S_p$ (Koller and Friedman 2009; Kuipers and Moffa 2017; Agrawal, Uhler, and Broderick 2018; Kuipers, Suter, and Moffa 2022).

Next, consider the joint learning of multiple DAG models from $K$ datasets, one for each dataset. This problem, which henceforth is referred to as differential DAG analysis, is motivated by differential gene regulatory network (GRN) analysis in biology, where we may have gene data for samples from different tissues, developmental phases or case-control studies, and the goal is to see how the GRN changes across different samples (Li...
et al. 2020). Since the advent of the single-cell technology, differential GRN analysis has become increasingly important (Fiers et al. 2018; Van de Sande et al. 2020). As in the multi-task variable selection problem, we assume the same $p$ covariates are observed in $K$ datasets, and use $X^{(k)} \in \mathbb{R}^{n_k \times p}$ to denote the data matrix for the $k$th dataset with sample size $n_k$. Denote the $K$ DAGs we want to learn by $(\hat{G}^{(k)} = (V, E^{(k)}))_{k=1}^K$, which share the same node set $V = [p]$ and, a priori, are believed to share a large proportion of common edges. We further assume that $\hat{G}^{(1)}, \ldots, \hat{G}^{(K)}$ are “permutation compatible,” which means that for any $i \neq j$, if $(i, j) \in E^{(k)}$ for some $k \in [K]$, then $(j, i) \notin E^{(k)}$ for any $k' \in [K]$. In other words, we assume there exists an order shared by all the $K$ DAGs. This assumption has been widely used in the literature (Liu, Sun, and Liu 2019; Chen et al. 2021; Lee and Cao 2022), and is very reasonable for problems such as GRN analysis, where an edge may occur only in some data sets but generally does not change direction across datasets. Observe that if the order $< \in S_p$ is known, learning $K$ DAGs can be converted to $p$ multi-task variable selection problems. One just needs to repeatedly apply the IBSS algorithm we have proposed to select the parent nodes for each $j \in [p]$. Denote the resulting $K$ DAGs by $(G^{(k)})_{k=1}^K$. We are interested in the case where the ordering is unknown. To average over the order space, we follow the existing order MCMC works to devise a Metropolis-Hastings algorithm on $\mathcal{S}_p$, which we describe in detail in the next section.

4.2. An Order MCMC Sampler for Differential DAG Analysis

We propose to consider the following Gibbs posterior distribution (Jiang and Tanner 2008),

$$P(\prec | (X^{(k)})_{k=1}^K) \propto P((G^{(k)})_{k=1}^K | \prec) \prod_{k=1}^K \tilde{P}(X^{(k)} | G^{(k)}_{\prec}),$$

\forall \prec \in \mathcal{S}_p, \tag{15}$$

where $(G^{(k)})_{k=1}^K$ denotes the DAGs we obtain by applying the IBSS algorithm with ordering $\prec$. The product term in (15) denotes the “estimated” likelihood function, which gives the estimated probability of observing the data given that $G^{(k)}_{\prec}$ is the underlying DAG model for the $k$th dataset. Denote by $A^\prec = \{i \in [p]: i < j\}$ the index set of variables preceding $X_j$ in the order $\prec$. Let

$$\hat{a}_{j, \prec} = \left(\hat{\sigma}_{j, \prec}, (\hat{a}_{j, \prec}^{(k)})_{k=1}^K, ((X^{(k)}_{i}: i \in A^\prec), (X^{(k)}_{j})_{k=1}^K, L)\right) \leftarrow \text{IBSS}$$

$$\left(\tilde{\beta}_{j, \prec}^{(k)} = \frac{\sum_{l=1}^L \hat{\beta}_{j, \prec}^{(k)}}{L} \right)_{k=1}^K, \tag{16}$$

denote the output of Algorithm 1 for the multi-task variable selection problem with response vector $X_j$ and covariates $\{X_i: i \in A^\prec\}$. As in (12), let $\hat{\sigma}_{j, \prec} = \sum_{l=1}^L \hat{\beta}_{j, \prec}^{(k)}$ denote the posterior mean aggregated over $L$ single effects. Then, we can estimate the likelihood of the DAGs $(G^{(k)})_{k=1}^K$ by plugging in the estimates $(\hat{\sigma}_{j, \prec})_{k=1}^K$ and $(\hat{\beta}_{j, \prec})_{k=1}^K$, which yields

$$\prod_{k=1}^K \tilde{P}(X^{(k)} | G^{(k)}_{\prec}) = \prod_{k=1}^K \prod_{i=1}^p \prod_{j=1}^p \Phi \left(\frac{X_{ij} - X_{i,A^\prec}^{(k)} \hat{\beta}_{j, \prec}^{(k)}}{\hat{\sigma}_{j, \prec}}\right), \tag{17}$$

where $\Phi(x)$ is the density function for the standard normal distribution and $X^{(k)}_{i,A^\prec} \overset{\text{row}}{\sim} \mathcal{N}(0, \mathbb{I})$. The first term $P((G^{(k)})_{k=1}^K | \prec)$ in (15) is the prior probability of the DAGs $(G^{(k)})_{k=1}^K$ given order $\prec$, or more generally can be any positive function that penalizes DAGs with more edges.

Analogously to (13), given $(\alpha^{(k)})_{k=1}^K$, we define $\bar{a}_{j} = 1 - \prod_{i=1}^p (1 - \alpha^{(k)}_{i,j})$, and we let $\bar{a}_{i,j}^{(k)} < \prec$ denote the corresponding quantity when $(\alpha^{(k)})_{k=1}^K = (1 - \alpha^{(k)}_{i,j})_{k=1}^K$, where $(\alpha^{(k)}_{i,j})_{k=1}^K$ is defined in (16). Write $a_{i,j} = (a_{i,j}^{(k)})_{k=1}^K$, and define $a_i(\alpha_{j, \prec}) = \sum_{i=1}^p \sum_{l=1}^L \bar{a}_{i,j}^{(k)}$, which gives the estimated number of covariates that are activated in exactly $k$ distinct datasets. We define the prior term in (15) by

$$P((G^{(k)})_{k=1}^K | \prec) = \prod_{k=1}^P \prod_{j=1}^P \rho^{-\omega \bar{a}_i(\alpha_{j, \prec})}. \tag{18}$$

Recall that $\omega_1, \ldots, \omega_p$ are the hyperparameters introduced in (3) for muSSVS and can be seen as a reparameterization of $\pi$ by (9). The reasoning behind (18) is the same as that behind (3). Combining (17) and (18), we get a closed-form expression for the posterior defined in (15). For later use, let $R_{(k)}^{(i,j)} \in [0, 1]^{p \times p}$ be the matrix such that

$$R_{(k)}^{(i,j)}_{ij} = \mathbb{I}_{\{i \in A^\prec\}} \sum_{l=1}^L \bar{a}_{i,j}^{(k)}. \tag{19}$$

That is, $(R_{(k)}^{(i,j)})_{ij}$ is the estimated probability of the edge $(i, j)$ being in the $k$th dataset given the order $\prec$. Given the target posterior distribution defined in (15), we are now ready to introduce our Metropolis-Hastings algorithm for differential DAG analysis. Given the current state $\prec \in \mathcal{S}_p$, we propose another state $\prec'$ from some proposal distribution $q(\cdot | \prec')$ and accept it with probability

$$\min \left\{ 1, \frac{P(\prec' | (X^{(k)})_{k=1}^K)}{P(\prec | (X^{(k)})_{k=1}^K)} q(\prec | \prec') \right\}. \tag{20}$$

We choose $q(\cdot | \prec')$ to be the uniform distribution on the set of permutations that can be obtained from $\prec$ by an adjacent transposition. That is, we randomly pick $j \in [p-1]$ with equal probability and then propose to move from $\prec$ to $\prec' = (i_1, \ldots, i_{j+1}, i_j, i_{j+1}, \ldots, i_p)$. Clearly, $q(\prec | \prec') = q(\prec' | \prec)$, and thus the proposal ratio in (20) is always equal to 1. Note that to calculate $P(\prec' | (X^{(k)})_{k=1}^K)$, we need to run IBSS to find the DAGs $(G^{(k)})_{k=1}^K$. Running this Metropolis-Hastings sampler for $T$ iterations (excluding burn-in), we obtain a sequence of permutations denoted by $(\prec)_t$. For each $\prec_t$, let $R_{(k)}^{(l)} \in [0, 1]^{p \times p}$ be the matrix defined in (19), and then $(R_{(k)}^{(l)})_{t=1}^T$ can be used for making posterior inference. For example, to estimate the probability of the edge $i \rightarrow j$ being in the $k$th DAG model, we can simply calculate the time average

$$R_{(k)}^{(i,j)}_{ji} = \frac{1}{T} \sum_{t=1}^T (R_{(k)}^{(l)})_{ij}. \tag{21}$$
5. Simulation Studies for Bayesian Differential DAG Analysis

We use simulation studies to investigate the performance of the order MCMC sampler described in Section 4.2, which we denote by muSuSiE-DAG, in two scenarios: $K = 2$, $n_1 = n_2 = 300$, and $K = 5$, $n_1 = \cdots = n_5 = 240$. We fix the number of nodes $p$ to 100 for all experiments. For each experiment, we generate the data according to the linear SEM (14) with true order given by $\omega= (1, 2, \ldots, p)$. Hence, the true weighted adjacency matrices of the $K$ DAGs are strictly upper triangular. The true DAGs $(G^{(k)})_{k=1}^{K}$ are then generated as follows. First, we generate a random edge set $\mathcal{E}_{\text{com}}$ consistent with $\omega$ such that each edge in $\mathcal{E}_{\text{com}}$ is activated in all the $K$ datasets. Second, for each $k \in [K]$, we generate an edge set $\mathcal{E}_{\text{pri}}$ which consists of edges that are only activated in the $k$th dataset. Let $N_{\text{com}} = |\mathcal{E}_{\text{com}}|$ denote the number of edges shared by all the $K$ DAGs and $N_{\text{pri}} = |\mathcal{E}_{\text{pri}}|$ denote the number of private edges unique to each dataset.

We consider $N_{\text{com}} \in \{50, 100\}$, and $N_{\text{pri}} \in \{20, 50\}$ in the simulation studies. To generate the matrix $B^{(k)}$ corresponding to DAG $G^{(k)}$ and the error variances of the $p$ variables, we follow Wang, Segarra, and Uhler (2020b) to sample the nonzero entries of $B^{(k)}$ (determined by $G^{(k)}$) independently from the uniform distribution on $[-1, -0.1] \cup [0.1, 1]$ and sample the error variance of each variable independently from the uniform distribution on $[1, 2.25]$. Note that for each edge in $\mathcal{E}_{\text{com}}$, its weights in the $K$ datasets are drawn independently.

For each simulation setting, we generate 50 replicates; the true DAG models and the data $(X^{(k)})_{k=1}^{K}$ are re-sampled for each replicate. We compare the performance of six methods: PC algorithm or GES applied independently to each data set (Spirtes et al. 2000; Chickering 2002; Harris and Drton 2013), the joint GES algorithm proposed by Wang, Segarra, and Uhler (2020b) which is a state-of-the-art method for joint learning multiple DAG models with theoretical guarantees, MPenPC method of (Liu, Sun, and Liu 2019), JESC method (Lee and Cao 2022), and muSuSiE-DAG. We implement PC and GES algorithms using the R package pcalg (Kalisch et al. 2012), and MPenPC and JESC using publicly available code with default parameters. In the ensuing results, we select parameter values that yield the most robust empirical performance across our experiments. For the PC algorithm, we let the significance level used in the conditional independent tests be 0.005, and for GES and joint GES methods, we let $\lambda = 2$, where $\lambda$ is the $\ell_0$-penalization parameter (scaled by $\log p$). For the muSuSiE-DAG method, we need to set the penalty parameters $\omega_1, \ldots, \omega_K$. For $K = 2$, we use $p^{-\omega_1} = p^{-2}/2$ and $p^{-\omega_2} = p^{-2.25}$, and the choice for $K = 5$ is given in Appendix D, supplementary materials. The results for the four methods obtained by using other parameter values are also provided in Appendix D, supplementary materials.

Table 2 shows the results for $K = 2$, and the results for $K = 5$ are given in Appendix D, supplementary materials. For each method, we calculate the average number of incorrect edges, denoted by $N_{\text{wrong}}$, the average true positive rate (TP) and the average false positive (FP) rate by ignoring the edge directions. As expected, joint GES and muSuSiE-DAG have significantly larger true positive rates than PC and GES methods, since the former two methods are able to use information from all the $K$ datasets to infer common edges, which is particularly useful when an edge has a relatively small signal size in both datasets. Meanwhile, the two joint methods tend to have slightly larger false positive rates as well, since an edge with a very large signal size in one dataset is likely to be identified by the joint method as existing concurrently in both datasets. However, note that the false positive rate of muSuSiE-DAG is still comparable to that of PC and GES and is much smaller than that of joint GES. Both MPenPC and JESC have high TP and FP rates, and JESC seems to perform significantly better than MPenPC. Overall, muSuSiE-DAG has the best performance among all the six methods in all settings, and its advantage is more significant when the ratio $N_{\text{com}}/N_{\text{pri}}$ is larger. The convergence of our order MCMC is discussed in Appendix D.1, supplementary materials.

6. A Real Data Example for Differential DAG Analysis

To evaluate the performance of the proposed muSuSiE-DAG method in real data analysis, we consider a pre-processed gene expression microarray dataset used in Wang, Segarra, and Uhler (2020b), which consists of two groups of patients with ovarian cancer. The first group has 83 patients who have enhanced survival rate. The second group has 168 patients who have ovarian cancer of other subtypes. For both groups, we observe the expression levels of $p = 76$ genes, which, according to
the KEGG database (Kanahasaka et al. 2012), participate in the apoptotic pathway. For more details about the original dataset, see Tothill et al. (2008). Let $G_1$ denote the underlying DAG model for the first group and $G_2$ denote that for the second. The objective of this real data analysis is to detect the differences between the two DAGs $G_1, G_2$, which may be associated with the survival rate. As in Section 5, we compare the performance of four methods: PC, GES, joint GES and muSSVS-DAG. Table 3 lists the number of edges detected by each method. The results for all four methods obtained by using other parameter values are provided in Appendix E, supplementary materials, where one can also find results obtained by combining PC, GES and joint GES with stability selection (Meinshausen and Bühlmann 2010). The results clearly illustrate the differences between the four methods. First, the percentage of shared edges in the two estimated DAGs (i.e., the “ratio” column in Table 3) is much larger for the two joint methods, which is consistent with both our theory and simulation results. For PC and GES, this ratio is always less than 0.3 in all parameter settings we have tried; see Tables E.1 and E.2 in Appendix E, supplementary materials. This shows that when the sample size is not large, applying a structure learning method to two datasets separately is very likely to miss some gene-gene interactions existing in both gene regulatory networks. Second, joint GES has the largest shared ratio, and it is often much larger than that of muSSVS-DAG. This is probably because joint GES is a two-step procedure where the first step is to learn a large DAG $G^{\text{union}}$, and in the second step $G_1$ and $G_2$ are constructed separately under the constraint that they must be sub-DAGs of $G^{\text{union}}$. If an edge only exists in one DAG or it exists in both but has very different regression coefficients in the two SEMs, it is not very likely to be included in $G^{\text{union}}$ and thus cannot be detected in the second step of joint GES. Indeed, since $p = 76$ is relatively large and $n_1 = 83$ and $n_2 = 168$, we expect that more edges (especially those with small signal sizes) can be detected in $G_2$ than in $G_1$, which is observed for PC, GES, and muSSVS-DAG.

### Supplementary Materials

**Appendices:** Appendixes.pdf gives the proof, additional simulation results and more details about the algorithm implementation.

**code:** Folder code includes the R code for replicating the simulation study and real data analysis presented in the article.

### Disclosure Statement

No potential conflict of interest was reported by the author(s).

### Acknowledgments

We thank Yuhao Wang for sharing with us the code for the joint GES method and the pre-processed real dataset.

### Funding

QZ was supported in part by NSF grant DMS-2245591.

### ORCID

Guanxun Li http://orcid.org/0000-0002-2449-4469

### References

Agrawal, R., Uhler, C., and Broderick, T. (2018), “Minimal I-MAP MCMC for Scalable Structure Discovery in Causal DAG Models,” in International Conference on Machine Learning, pp. 89–98. [36,41]

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017), “Variational Inference: A Review for Statisticians,” Journal of the American Statistical Association, 112, 859–877. [39]
Bonilla, E. V., Chai, K., and Williams, C. (2007), “Multi-Task Gaussian Process Prediction,” in *Advances in Neural Information Processing Systems* (Vol. 20). [35]

Carbonetto, P., and Stephens, M. (2012), “Scalable Variational Inference for Bayesian Variable Selection in Regression, and its Accuracy in Genetic Association Studies,” *Bayesian Analysis*, 7, 73–108. [40]

Castelli, F., Locca, L., Peluso, S., Stingo, F. C., and Consonni, G. (2020), “Bayesian Learning of Multiple Directed Networks from Observational Data,” *Statistics in Medicine*, 39, 4745–4766. [36,41,43]

Chen, X., Sun, H., Ellington, C., Xing, E., and Song, L. (2021), “Multi-Task Learning of Order-Consistent Causal Graphs,” in *Advances in Neural Information Processing Systems* (Vol. 34). [41,42]

Chickering, D. M. (2002), “Optimal Structure Identification with Greedy Sampling,” *Journal of Machine Learning Research*, 3, 507–554. [41,43]

Danaher, P., Wang, P., and Witten, D. M. (2014), “The Joint Graphical Lasso for Inverse Covariance Estimation Across Multiple Classes,” *Journal of the Royal Statistical Society, Series B*, 76, 373–397. [41]

Ellis, B., and Wong, W. H. (2008), “Learning Causal Bayesian Network Structures from Experimental Data,” *Journal of the American Statistical Association*, 103, 778–789. [43]

Fiers, M. W. E. J., Minnoye, L., Aibar, S., Bravo González-Blas, C., Kalendar, Z., and Aerts, S. (2018), “Mapping Gene Regulatory Networks from Single-Cell Omics Data,” *Briefings in Functional Genomics*, 17, 246–254. [42]

Fronk, E.-M., and Giudici, P. (2004), “Markov Chain Monte Carlo Model Selection for DAG Models,” *Statistical Methods and Applications*, 13, 259–273. [41]

George, E. I., and McCulloch, R. E. (1993), “Variable Selection via Gibbs Sampling,” *Journal of the American Statistical Association*, 88, 881–889. [35]

Ghoshal, A., Bello, K., and Honorio, J. (2019), “Direct Learning with Guarantees of the Difference DAG between Structural Equation Models,” arXiv preprint arXiv:1906.12024. [41]

Goñi, G., Goñi, C., Calle, P., and Sanz, J. (2002), “Variable Selection for Gaussian Bayesian Networks,” in *Advances in Neural Information Processing Systems* (Vol. 14), 730–737. [35]

Guo, S., Zue, O., and Archambeau, C. (2011), “Sparse Bayesian Multi-Task Learning,” in *Advances in Neural Information Processing Systems* (Vol. 24). [35]

Harrington, N., and Zhu, Y. (2013), “A New Method for Multitask Learning,” *Journal of Machine Learning Research*, 14, 3365–3383. [43]

Hernández-Lobato, D., Hernández-Lobato, J. M., and Ghahramani, Z. (2015), “A Probabilistic Model for Dirty Multi-Task Feature Selection,” in *International Conference on Machine Learning*, pp. 1073–1082, PMLR. [35]

Huang, X., Wang, J., and Liang, F. (2016), “A Variational Algorithm for Sparse Structure Learning with Gaussian Copula Models,” *The Journal of Machine Learning Research*, 17, 1205–1234. [41]

Guo, S., Zue, O., and Archambeau, C. (2011), “Sparse Bayesian Multi-Task Learning,” in *Advances in Neural Information Processing Systems* (Vol. 24). [35]

Harris, N., and Drton, M. (2013), “PC Algorithm for Nonparanormal Graphical Models,” *Journal of Machine Learning Research*, 14, 3365–3383. [43]

Hernández-Lobato, D., Hernández-Lobato, J. M., and Ghahramani, Z. (2015), “A Probabilistic Model for Dirty Multi-Task Feature Selection,” in *International Conference on Machine Learning*, pp. 1073–1082, PMLR. [35]

Huang, X., Wang, J., and Liang, F. (2016), “A Variational Algorithm for Bayesian Variable Selection,” arXiv preprint arXiv:1602.07640. [40]

Ishwaran, H., and Rao, J. S. (2005), “Splice and Slab Variable Selection: Frequentist and Bayesian Strategies,” *The Annals of Statistics*, 33, 730–773. [35]

Jalali, A., Sanghavi, S., Ruan, C., and Ravikumar, P. (2010), “A Dirty Model for Multi-Task Learning,” in *Advances in Neural Information Processing Systems* (Vol. 23). [35]

Jeong, S., and Ghosal, S. (2021), “Unified Bayesian Theory of Sparse Linear Regression with Nuisance Parameters,” *Electronic Journal of Statistics*, 15, 3040–3111. [37]

Jiang, W., and Tanner, M. A. (2008), “Gibbs Posterior for Variable Selection in High-Dimensional Classification and Data Mining,” *The Annals of Statistics*, 36, 2207–2231. [42]

Johnson, V. E., and Rossell, D. (2012), “Bayesian Model Selection in High-Dimensional Settings,” *Journal of the American Statistical Association*, 107, 649–660. [37]

Kalisch, M., Bühlmann, P., Colombo, D., Maathuis, M. H., and Bühlmann, P. (2012), “Causal Inference Using Graphical Models with the R Package pcalg,” *Journal of Statistical Software*, 47, 1–26. [43]

Kanehisa, M., Goto, S., Yamanishi, M., and Mi, H. (2012), “KEGG for Integration and Interpretation of Large-Scale Molecular Data Sets,” *Nucleic Acids Research*, 40, D109–D114. [43]

Koller, D., and Friedman, N. (2009), *Probabilistic Graphical Models: Principles and Techniques*, Cambridge, MA: MIT Press. [41]
Wang, G., Sarkar, A., Carbonetto, P., and Stephens, M. (2020), "A Simple New Approach to Variable Selection in Regression, with Application to Genetic Fine Mapping," *Journal of the Royal Statistical Society, Series B*, 82, 1273–1300. [35,38,39,40,44]

Wang, Y., Segarra, S., and Uhler, C. (2020b), "High-Dimensional Joint Estimation of Multiple Directed Gaussian Graphical Models," *Electronic Journal of Statistics*, 14, 2439–2483. [41,43]

Yajima, M., Telesca, D., Ji, Y., and Müller, P. (2015), "Detecting Differential Patterns of Interaction in Molecular Pathways," *Biostatistics*, 16, 240–251. [41]

Yang, Y., Wainwright, M. J., and Jordan, M. I. (2016), "On the Computational Complexity of High-Dimensional Bayesian Variable Selection," *The Annals of Statistics*, 44, 2497–2532. [37,38]

Zhang, A., Zhang, G., Calhoun, V. D., and Wang, Y.-P. (2020), "Causal Brain Network in Schizophrenia by a Two-Step Bayesian Network Analysis," in *Medical Imaging 2020: Imaging Informatics for Healthcare, Research, and Applications* (Vol. 11318), pp. 316–321, SPIE. [35]

Zhang, Y., and Yang, Q. (2021), "A Survey on Multi-Task Learning," *IEEE Transactions on Knowledge and Data Engineering*, 34, 5586–5609. [35]