A Appendix of Method

A.1 Backward mapping for Exponential Families

The solution of vanilla graphical model MLE can be expressed as a backward mapping[29] for an exponential family distribution. It estimates the model parameters (canonical parameter θ) from certain (sample) moments. We provide detailed explanations about backward mapping of exponential families, backward mapping for Gaussian special case and backward mapping for differential network of GGM in this section.

Backward mapping: Essentially the vanilla graphical model MLE can be expressed as a backward mapping that computes the model parameters corresponding to some given moments in an exponential family distribution. For instance, in the case of learning GGM with vanilla MLE, the backward mapping is $\widehat{\Sigma}^{-1}$ that estimates Ω from the sample covariance (moment) $\widehat{\Sigma}$.

Suppose a random variable $X \in \mathbb{R}^p$ follows the exponential family distribution:

$$\mathbb{P}(X;\theta) = h(X)\exp\{<\theta, \phi(\theta) > -A(\theta)\} \tag{A.1}$$

Where $\theta \in \Theta \subset \mathbb{R}^d$ is the canonical parameter to be estimated and Θ denotes the parameter space. $\phi(X)$ denotes the sufficient statistics as a feature mapping function $\phi : \mathbb{R}^p \to \mathbb{R}^d$, and $A(\theta)$ is the log-partition function. We then define mean parameters v as the expectation of $\phi(X)$: $v(\theta) := \mathbb{E}[\phi(X)]$, which can be the first and second moments of the sufficient statistics $\phi(X)$ under the exponential family distribution. The set of all possible moments by the moment polytope:

$$\mathcal{M} = \{v | \exists p \text{ is a distribution s.t. } \mathbb{E}_p[\phi(X)] = v\} \tag{A.2}$$

Mostly, the graphical model inference involves the task of computing moments $v(\theta) \in \mathcal{M}$ given the canonical parameters $\theta \in H$. We denote this computing as **forward mapping** :

$$\mathcal{A} : H \to \mathcal{M} \tag{A.3}$$

The learning/estimation of graphical models involves the task of the reverse computing of the forward mapping, the so-called **backward mapping** [29]. We denote the interior of $\mathcal{M}$ as $\mathcal{M}^0$. **backward mapping** is defined as:

$$\mathcal{A}^* : \mathcal{M}^0 \to H \tag{A.4}$$

which does not need to be unique. For the exponential family distribution,

$$\mathcal{A}^* : v(\theta) \to \theta = \nabla A^*(v(\theta)). \tag{A.5}$$

Where $A^*(v(\theta)) = \sup_{\theta \in H} < \theta, v(\theta) > -A(\theta)$.

Backward Mapping: Gaussian Case If a random variable $X \in \mathbb{R}^p$ follows the Gaussian Distribution $N(\mu, \Sigma)$. then $\theta = (\Sigma^{-1}\mu, -\frac{1}{2}\Sigma^{-1})$. The sufficient statistics $\phi(X) = (X, XX^T)$, $h(x) = (2\pi)^{-\frac{k}{2}}$, and the log-partition function

$$A(\theta) = \frac{1}{2}\mu^T\Sigma^{-1}\mu + \frac{1}{2}\log(|\Sigma|) \tag{A.6}$$

When performing the inference of Gaussian Graphical Models, it is easy to estimate the mean vector $v(\theta)$, since it equals to $\mathbb{E}[X, XX^T]$.

When learning the GGM, we estimate its canonical parameter θ through vanilla MLE. Because Σ^{-1} is one entry of θ we can use the backward mapping to estimate Σ^{-1}.

$$\begin{aligned} \theta &= (\Sigma^{-1}\mu, -\frac{1}{2}\Sigma^{-1}) = \mathcal{A}^*(v) = \nabla A^*(v) \\ &= ((\mathbb{E}_\theta[XX^T] - \mathbb{E}_\theta[X]\mathbb{E}_\theta[X]^T)^{-1}\mathbb{E}_\theta[X], \\ &\quad -\frac{1}{2}(\mathbb{E}_\theta[XX^T] - \mathbb{E}_\theta[X]\mathbb{E}_\theta[X]^T)^{-1}). \end{aligned} \tag{A.7}$$

By plugging in Eq. (A.6) into Eq. (A.5), we get the backward mapping of Ω as $(\mathbb{E}_\theta[XX^T] - \mathbb{E}_\theta[X]\mathbb{E}_\theta[X]^T)^{-1}) = \widehat{\Sigma}^{-1}$, easily computable from the sample covariance matrix.

A.1.1 Backward Mapping for Differential Network of Two GGMs

When the random variables $X_c, X_d \in \mathbb{R}^p$ follows the Gaussian Distribution $N(\mu_c, \Sigma_c)$ and $N(\mu_d, \Sigma_d)$, their density ratio (defined by [17]) essentially is a distribution in exponential families:

$$\begin{aligned} r(x, \Delta) &= \frac{p_d(x)}{p_c(x)} \\ &= \frac{\sqrt{\det(\Sigma_c)}\exp\left(-\frac{1}{2}(x-\mu_d)^T\Sigma_d^{-1}(x-\mu_d)\right)}{\sqrt{\det(\Sigma_d)}\exp\left(-\frac{1}{2}(x-\mu_c)^T\Sigma_c^{-1}(x-\mu_c)\right)} \\ &= \exp(-\frac{1}{2}(x-\mu_d)^T\Sigma_d^{-1}(x-\mu_d) \\ &+ \frac{1}{2}(x-\mu_c)^T\Sigma_c^{-1}(x-\mu_c) \\ &- \frac{1}{2}(\log(\det(\Sigma_d)) - \log(\det(\Sigma_c)))) \\ &= \exp\left(-\frac{1}{2}\Delta x^2 + \mu_\Delta x - A(\mu_\Delta, \Delta)\right) \end{aligned} \tag{A.8}$$

Here $\Delta = \Sigma_d^{-1} - \Sigma_c^{-1}$ and $\mu_\Delta = \Sigma_d^{-1}\mu_d - \Sigma_c^{-1}\mu_c$.

The log-partition function

$$\begin{aligned} A(\mu_\Delta, \Delta) = \frac{1}{2}\mu_d^T\Sigma_d^{-1}\mu_d - \frac{1}{2}\mu_c^T\Sigma_c^{-1}\mu_c + \\ \frac{1}{2}\log(\det(\Sigma_d)) - \frac{1}{2}\log(\det(\Sigma_c)) \end{aligned} \tag{A.9}$$

The canonical parameter

$$\begin{aligned} \theta &= \left(\Sigma_d^{-1}\mu_d - \Sigma_c^{-1}\mu_c, -\frac{1}{2}(\Sigma_d^{-1} - \Sigma_c^{-1})\right) \\ &= \left(\Sigma_d^{-1}\mu_d - \Sigma_c^{-1}\mu_c, -\frac{1}{2}(\Delta)\right) \end{aligned} \tag{A.10}$$

The sufficient statistics $\phi([X_c, X_d])$ and the log-partition function $A(\theta)$:

$$
\begin{aligned}
&\phi([X_c, X_d]) = ([X_c, X_d], [X_c X_c^T, X_d X_d^T]) \\
&A(\theta) = \frac{1}{2}\mu_d^T \Sigma_d^{-1} \mu_d - \frac{1}{2}\mu_c^T \Sigma_c^{-1} \mu_c + \qquad \text{(A.11)} \\
&\frac{1}{2}\log(\det(\Sigma_d)) - \frac{1}{2}\log(\det(\Sigma_c))
\end{aligned}
$$

And $h(x) = 1$.

Now we can estimate this exponential distribution (θ) through vanilla MLE. By plugging Eq. (A.11) into Eq. (A.5), we get the following backward mapping via the conjugate of the log-partition function:

$$
\begin{aligned}
\theta = &\left(\Sigma_d^{-1}\mu_d - \Sigma_c^{-1}\mu_c, -\frac{1}{2}(\Sigma_d^{-1} - \Sigma_c^{-1})\right) \qquad \text{(A.12)} \\
=&\mathcal{A}^*(v) = \nabla A^*(v)
\end{aligned}
$$

The mean parameter vector $v(\theta)$ includes the moments of the sufficient statistics $\phi()$ under the exponential distribution. It can be easily estimated through $\mathbb{E}[([X_c, X_d], [X_c X_c^T, X_d X_d^T])]$.

Therefore the backward mapping of θ becomes,

$$
\begin{aligned}
\widehat{\theta} =&(((\mathbb{E}_\theta[X_d X_d^T] - \mathbb{E}_\theta[X_d]\mathbb{E}_\theta[X_d]^T)^{-1}\mathbb{E}_\theta[X_d] \\
&- (\mathbb{E}_\theta[X_c X_c^T] - \mathbb{E}_\theta[X_c]\mathbb{E}_\theta[X_c]^T)^{-1}\mathbb{E}_\theta[X_c]), \\
&- \frac{1}{2}((\mathbb{E}_\theta[X_d X_d^T] - \mathbb{E}_\theta[X_d]\mathbb{E}_\theta[X_d]^T)^{-1} - \\
&(\mathbb{E}_\theta[X_c X_c^T] - \mathbb{E}_\theta[X_c]\mathbb{E}_\theta[X_c]^T)^{-1})).
\end{aligned}
\qquad \text{(A.13)}
$$

Because the second entry of the canonical parameter θ is $(\Sigma_d^{-1} - \Sigma_c^{-1})$, we get the backward mapping of Δ as

$$
\begin{aligned}
&((\mathbb{E}_\theta[X_d X_d^T] - \mathbb{E}_\theta[X_d]\mathbb{E}_\theta[X_d]^T)^{-1} \\
&-(\mathbb{E}_\theta[X_c X_c^T] - \mathbb{E}_\theta[X_c]\mathbb{E}_\theta[X_c]^T)^{-1}) \qquad \text{(A.14)} \\
&=\widehat{\Sigma}_d^{-1} - \widehat{\Sigma}_c^{-1}
\end{aligned}
$$

This can be easily inferred from two sample covariance matrices $\widehat{\Sigma}_d$ and $\widehat{\Sigma}_c$ (Att: when under low-dimensional settings).

A.2 Appendix:Proof

A.2.1 Derivation of Theorem (2.1)

DIFFEE formulation Eq. (2.11) and EE-sGGM Eq. (2.3) are special cases of the following generic formulation:

$$
\begin{aligned}
&\underset{\theta}{\operatorname{argmin}}\, \mathcal{R}(\theta) \\
&\text{subject to:} \mathcal{R}^*(\theta - \widehat{\theta}_n) \le \lambda_n
\end{aligned}
\qquad \text{(A.15)}
$$

Where $\mathcal{R}^*(\cdot)$ is the dual norm of $\mathcal{R}(\cdot)$,

$$
\mathcal{R}^*(v) := \sup_{u \neq 0} \frac{< u, v >}{\mathcal{R}(u)} = \sup_{\mathcal{R}(u) \le 1} < u, v > . \qquad \text{(A.16)}
$$

Connecting Eq. (2.11) and Eq. (A.15), $\mathcal{R}()$ is the $\ell 1$ norm, $\mathcal{R}^*()$ is the ℓ_∞-norm, and ℓ_∞-norm is the dual norm of ℓ_1-norm. $\widehat{\theta}_n$ represents a backward mapping (or proxy backward mapping well-defined in high-dimensional settings) of θ , which is a close approximation of θ^*.

Following the unified framework [20], we first decompose the parameter space into a subspace pair$(\mathcal{M}, \bar{\mathcal{M}}^\perp)$, where $\bar{\mathcal{M}}$ is the closure of $\mathcal{M}$. Here $\bar{\mathcal{M}}^\perp := \{v \in \mathbb{R}^p | < u, v >= 0, \forall u \in \bar{\mathcal{M}}\}$. $\mathcal{M}$ is the **model subspace** that typically has a much lower dimension than the original high-dimensional space. $\bar{\mathcal{M}}^\perp$ is the **perturbation subspace** of parameters. For further proofs, we assume the regularization function in Eq. (A.15) is **decomposable** w.r.t the subspace pair $(\mathcal{M}, \bar{\mathcal{M}}^\perp)$.

(C1) $\mathcal{R}(u + v) = \mathcal{R}(u) + \mathcal{R}(v)$, $\forall u \in \mathcal{M}, \forall v \in \bar{\mathcal{M}}^\perp$.

[20] showed that most regularization norms are decomposable corresponding to a certain subspace pair.

Definition A.1. ***Subspace Compatibility Constant***
Subspace compatibility constant is defined as $\Psi(\mathcal{M}, |\cdot|) := \sup_{u \in \mathcal{M} \setminus \{0\}} \frac{\mathcal{R}(u)}{|u|}$ *which captures the relative value between the error norm* $|\cdot|$ *and the regularization function* $\mathcal{R}(\cdot)$.

For simplicity, we assume there exists a true parameter θ^* which has the exact structure w.r.t a certain subspace pair. Concretely:

(C2) $\exists$ a subspace pair $(\mathcal{M}, \bar{\mathcal{M}}^\perp)$ such that the true parameter satisfies $\text{proj}_{\mathcal{M}^\perp}(\theta^*) = 0$

Then we have the following theorem.

Theorem A.2. *Suppose the regularization function in Eq. (A.15) satisfies condition* ***(C1)****, the true parameter of Eq. (A.15) satisfies condition* ***(C2)****, and* λ_n *satisfies that* $\lambda_n \ge \mathcal{R}^*(\widehat{\theta}_n - \theta^*)$. *Then, the optimal solution* $\widehat{\theta}$ *of Eq. (A.15) satisfies:*

$$
\mathcal{R}^*(\widehat{\theta} - \theta^*) \le 2\lambda_n \qquad \text{(A.17)}
$$

$$
||\widehat{\theta} - \theta^*||_2 \le 4\lambda_n \Psi(\bar{\mathcal{M}}) \qquad \text{(A.18)}
$$

$$
\mathcal{R}(\widehat{\theta} - \theta^*) \le 8\lambda_n \Psi(\bar{\mathcal{M}})^2 \qquad \text{(A.19)}
$$

For the proposed DIFFEE model, $\mathcal{R} = ||\cdot||_1$. Based on the results in[20], $\Psi(\bar{\mathcal{M}}) = \sqrt{k}$, where k is the total number of nonzero entries in Δ. Using $\mathcal{R} = ||\cdot||_1$ in Theorem (A.2), we have the following theorem (the same as Theorem (2.1)),

Theorem A.3. *Suppose that* $\mathcal{R} = ||\cdot||_1$ *and the true parameter* Δ^* *satisfy the conditions* ***(C1)(C2)*** *and* $\lambda_n \ge \mathcal{R}^*(\widehat{\Delta} - \Delta^*)$, *then the optimal point* $\widehat{\Delta}$ *of Eq. (2.11) has the following error bounds:* $||\widehat{\Delta} - \Delta^*||_\infty \le 2\lambda_n$, $||\widehat{\Delta} - \Delta^*||_2 \le 4\sqrt{k}\lambda_n$, *and* $||\widehat{\Delta} - \Delta^*||_1 \le 8k\lambda_n$

A.2.2 Proof of Theorem (A.2)

Proof. Let $\delta := \widehat{\theta} - \theta^*$ be the error vector that we are interested in.

$$\begin{aligned}&\mathcal{R}^*(\widehat{\theta} - \theta^*) = \mathcal{R}^*(\widehat{\theta} - \widehat{\theta}_n + \widehat{\theta}_n - \theta^*)\\ &\leq \mathcal{R}^*(\widehat{\theta}_n - \widehat{\theta}) + \mathcal{R}^*(\widehat{\theta}_n - \theta^*) \leq 2\lambda_n\end{aligned} \tag{A.20}$$

By the fact that $\theta^*_{\mathcal{M}^\perp} = 0$, and the decomposability of $\mathcal{R}$ with respect to $(\mathcal{M}, \bar{\mathcal{M}}^\perp)$

$$\begin{aligned}&\mathcal{R}(\theta^*)\\ &= \mathcal{R}(\theta^*) + \mathcal{R}[\Pi_{\bar{\mathcal{M}}^\perp}(\delta)] - \mathcal{R}[\Pi_{\bar{\mathcal{M}}^\perp}(\delta)]\\ &= \mathcal{R}[\theta^* + \Pi_{\bar{\mathcal{M}}^\perp}(\delta)] - \mathcal{R}[\Pi_{\bar{\mathcal{M}}^\perp}(\delta)]\\ &\leq \mathcal{R}[\theta^* + \Pi_{\bar{\mathcal{M}}^\perp}(\delta) + \Pi_{\bar{\mathcal{M}}}(\delta)] + \mathcal{R}[\Pi_{\bar{\mathcal{M}}}(\delta)]\\ &- \mathcal{R}[\Pi_{\bar{\mathcal{M}}^\perp}(\delta)]\\ &= \mathcal{R}[\theta^* + \delta] + \mathcal{R}[\Pi_{\bar{\mathcal{M}}}(\delta)] - \mathcal{R}[\Pi_{\bar{\mathcal{M}}^\perp}(\delta)]\end{aligned} \tag{A.21}$$

Here, the inequality holds by the triangle inequality of norm. Since Eq. (A.15) minimizes $\mathcal{R}(\widehat{\theta})$, we have $\mathcal{R}(\theta^* + \Delta) = \mathcal{R}(\widehat{\theta}) \leq \mathcal{R}(\theta^*)$. Combining this inequality with Eq. (A.21), we have:

$$\mathcal{R}[\Pi_{\bar{\mathcal{M}}^\perp}(\delta)] \leq \mathcal{R}[\Pi_{\bar{\mathcal{M}}}(\delta)] \tag{A.22}$$

Moreover, by Hölder's inequality and the decomposability of $\mathcal{R}(\cdot)$, we have:

$$\begin{aligned}&||\Delta||_2^2 = \langle\delta, \delta\rangle \leq \mathcal{R}^*(\delta)\mathcal{R}(\delta) \leq 2\lambda_n\mathcal{R}(\delta)\\ &= 2\lambda_n[\mathcal{R}(\Pi_{\bar{\mathcal{M}}}(\delta)) + \mathcal{R}(\Pi_{\bar{\mathcal{M}}^\perp}(\delta))] \leq 4\lambda_n\mathcal{R}(\Pi_{\bar{\mathcal{M}}}(\delta))\\ &\leq 4\lambda_n\Psi(\bar{\mathcal{M}})||\Pi_{\bar{\mathcal{M}}}(\delta)||_2\end{aligned} \tag{A.23}$$

where $\Psi(\bar{\mathcal{M}})$ is a simple notation for $\Psi(\bar{\mathcal{M}}, ||\cdot||_2)$.

Since the projection operator is defined in terms of $||\cdot||_2$ norm, it is non-expansive: $||\Pi_{\bar{\mathcal{M}}}(\Delta)||_2 \leq ||\Delta||_2$. Therefore, by Eq. (A.23), we have:

$$||\Pi_{\bar{\mathcal{M}}}(\delta)||_2 \leq 4\lambda_n\Psi(\bar{\mathcal{M}}), \tag{A.24}$$

and plugging it back to Eq. (A.23) yields the error bound Eq. (A.18).

Finally, Eq. (A.19) is straightforward from Eq. (A.22) and Eq. (A.24).

$$\begin{aligned}&\mathcal{R}(\delta) \leq 2\mathcal{R}(\Pi_{\bar{\mathcal{M}}}(\delta))\\ &\leq 2\Psi(\bar{\mathcal{M}})||\Pi_{\bar{\mathcal{M}}}(\delta)||_2 \leq 8\lambda_n\Psi(\bar{\mathcal{M}})^2.\end{aligned} \tag{A.25}$$

□

A.2.3 Useful lemma(s)

Lemma A.4. *(Theorem 1 of [26]). Let* δ *be* $\max_{ij}|[\frac{X^TX}{n}]_{ij} - \Sigma_{ij}|$. *Suppose that* $v > 2\delta$. *Then, under the conditions (C-Sparse*Σ*), and as* $\rho_v(\cdot)$ *is a soft-threshold function, we can deterministically guarantee that the spectral norm of error is bounded as follows:*

$$|||T_v(\widehat{\Sigma}) - \Sigma|||_\infty \leq 5v^{1-q}c_0(p) + 3v^{-q}c_0(p)\delta \tag{A.26}$$

Lemma A.5. *(Lemma 1 of [23]). Let* $\mathcal{A}$ *be the event that*

$$||\frac{X^TX}{n} - \Sigma||_\infty \leq 8(\max_i \Sigma_{ii})\sqrt{\frac{10\tau\log p'}{n}} \tag{A.27}$$

where $p' := \max n, p$ *and* τ *is any constant greater than 2. Suppose that the design matrix* X *is i.i.d. sampled from* Σ*-Gaussian ensemble with* $n \geq 40\max_i \Sigma_{ii}$. *Then, the probability of event* $\mathcal{A}$ *occurring is at least* $1 - 4/p'^{\tau-2}$.

To prove the bound of $||\Delta^* - ([T_v(\widehat{\Sigma}_d)]^{-1} - [T_v(\widehat{\Sigma}_c)]^{-1})||_\infty$, we first prove the bound of $||\Omega_c^* - [T_v(\widehat{\Sigma}_c)]^{-1}||_\infty$. In the following proof, we first derive the inequality $||\Omega_c^* - [T_v(\widehat{\Sigma}_c)]^{-1}||_\infty \leq |||[T_v(\widehat{\Sigma}_c)]^{-1}|||_\infty|||\Omega_c^*|||_\infty||T_v(\widehat{\Sigma}_c) - \Sigma_c^*||_\infty$, which is bounded by multiplication of three parts. Then we use the above Lemmas and two conditions to prove the bound of each part. Finally, we combine the three results to have the whole bound of $||\Omega_c^* - [T_v(\widehat{\Sigma}_c)]^{-1}||_\infty$.

A.2.4 Proof of Corollary (2.2)

Proof. In the following proof, we first prove $||\Omega_c^* - [T_v(\widehat{\Sigma}_c)]^{-1}||_\infty \leq \lambda_{n_c}$. Here $\lambda_{n_c} = \frac{4\kappa_1 a}{\kappa_2}\sqrt{\frac{\log p'}{n_c}}$ and $p' = \max(p, n_c)$

The condition (C-SparseΣ) and condition (C-MinInfΣ) also hold for Ω_c^* and Σ_c^*. In order to utilize Theorem (A.3) for this specific case, we only need to show that $||\Omega_c^* - [T_v(\widehat{\Sigma}_c)]^{-1}||_\infty \leq \lambda_{n_c}$ for the setting of $\lambda_{n_c} = \frac{4\kappa_1 a}{\kappa_2}\sqrt{\frac{\log p'}{n_c}}$:

$$\begin{aligned}&||\Omega_c^* - [T_v(\widehat{\Sigma}_c)]^{-1}||_\infty = ||[T_v(\widehat{\Sigma}_c)]^{-1}(T_v(\widehat{\Sigma}_c)\Omega_c^* - I)||_\infty\\ &\leq |||[T_v(\widehat{\Sigma}_c)w]|||_\infty||T_v(\widehat{\Sigma}_c)\Omega_c^* - I||_\infty\\ &= |||[T_v(\widehat{\Sigma}_c)]^{-1}|||_\infty||\Omega_c^*(T_v(\widehat{\Sigma}_c) - \Sigma_c^*)||_\infty\\ &\leq |||[T_v(\widehat{\Sigma}_c)]^{-1}|||_\infty|||\Omega_c^*|||_\infty||T_v(\widehat{\Sigma}_c) - \Sigma_c^*||_\infty.\end{aligned} \tag{A.28}$$

We first compute the upper bound of $|||[T_v(\widehat{\Sigma}_c)]^{-1}|||_\infty$. By the selection v in the statement, Lemma (A.4) and Lemma (A.5) hold with probability at least $1 - 4/p'^{\tau-2}$. Armed with Eq. (A.26), we use the triangle inequality of norm and the condition (C-SparseΣ): for any w,

$$\begin{aligned} ||T_v(\widehat{\Sigma}_c)w||_\infty &= ||T_v(\widehat{\Sigma}_c)w - \Sigma w + \Sigma w||_\infty \\ &\geq ||\Sigma w||_\infty - ||(T_v(\widehat{\Sigma}_c) - \Sigma)w||_\infty \\ &\geq \kappa_2||w||_\infty - ||(T_v(\widehat{\Sigma}_c) - \Sigma)w||_\infty \\ &\geq (\kappa_2 - ||(T_v(\widehat{\Sigma}_c) - \Sigma)w||_\infty)||w||_\infty \end{aligned} \tag{A.29}$$

Where the second inequality uses the condition (C-SparseΣ). Now, by Lemma (A.4) with the selection of v, we have

$$|||T_v(\widehat{\Sigma}_c) - \Sigma|||_\infty \leq c_1(\frac{\log p'}{n_c})^{(1-q)/2} c_0(p) \tag{A.30}$$

where c_1 is a constant related only on τ and $\max_i \Sigma_{ii}$. Specifically, it is defined as $6.5 \times (16(\max_i \Sigma_{ii})\sqrt{10\tau})^{1-q}$. Hence, as long as $n_c > (\frac{2c_1 c_0(p)}{\kappa_2})^{\frac{2}{1-q}} \log p'$ as stated, so that $|||T_v(\widehat{\Sigma}_c) - \Sigma|||_\infty \leq \frac{\kappa_2}{2}$, we can conclude that $||T_v(\widehat{\Sigma}_c)w||_\infty \geq \frac{\kappa_2}{2}||w||_\infty$, which implies $||||[T_v(\widehat{\Sigma}_c)]^{-1}|||_\infty \leq \frac{2}{\kappa_2}$.

The remaining term in Eq. (A.28) is $||T_v(\widehat{\Sigma}_c) - \Sigma_c^*||_\infty$; $||T_v(\widehat{\Sigma}_c) - \Sigma_c^*||_\infty \leq ||T_v(\widehat{\Sigma}_c) - \widehat{\Sigma}_c||_\infty + ||\widehat{\Sigma}_c - \Sigma_c^*||_\infty$. By construction of $T_v(\cdot)$ in (C-Thresh) and by Lemma (A.5), we can confirm that $||T_v(\widehat{\Sigma}_c) - \widehat{\Sigma}_c||_\infty$ as well as $||\widehat{\Sigma}_c - \Sigma_c^*||_\infty$ can be upper-bounded by v.

Similarly, the $[T_v(\widehat{\Sigma}_d)]^{-1}$ has the same result.

Finally,

$$||\Delta^* - \left([T_v(\widehat{\Sigma}_d)]^{-1} - [T_v(\widehat{\Sigma}_c)]^{-1}\right)||_\infty \tag{A.31}$$

$$\leq ||\Omega_d - [T_v(\widehat{\Sigma}_d)]^{-1}||_\infty + ||\Omega_c - [T_v(\widehat{\Sigma}_c)]^{-1}||_\infty \tag{A.32}$$

$$\leq \frac{4\kappa_1 a}{\kappa_2}\sqrt{\frac{\log p'}{n_c}} + \frac{4\kappa_1 a}{\kappa_2}\sqrt{\frac{\log p'}{n_c}} \tag{A.33}$$

Suppose $p > \max(n_c, n_d)$, we have that

$$||\Delta^* - \left([T_v(\widehat{\Sigma}_d)]^{-1} - [T_v(\widehat{\Sigma}_c)]^{-1}\right)||_\infty \leq \frac{8\kappa_1 a}{\kappa_2}\sqrt{\frac{\log p}{\min(n_c, n_d)}} \tag{A.34}$$

Similarly, we also have that

$$||\Delta^* - \left([T_v(\widehat{\Sigma}_d)]^{-1} - [T_v(\widehat{\Sigma}_c)]^{-1}\right)||_F \leq \frac{32\kappa_1 a}{\kappa_2}\sqrt{\frac{k \log p}{\min(n_c, n_d)}} \tag{A.35}$$

, and

$$||\Delta^* - \left([T_v(\widehat{\Sigma}_d)]^{-1} - [T_v(\widehat{\Sigma}_c)]^{-1}\right)||_1 \leq \frac{64\kappa_1 a}{\kappa_2} k \sqrt{\frac{\log p}{\min(n_c, n_d)}} \tag{A.36}$$

By combining all together, we can confirm that the selection of λ_n satisfies the requirement of Theorem (A.3), which completes the proof. □

B Details of Experimental Setup

Evaluation Metrics: We evaluate DIFFEE and the baseline methods on both contexts of effectiveness and scalability.

- F1-score: We first use the edge-level F1-score to compare the predicted versus true differential graph. Here, F1 $= \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision}+\text{Recall}}$, where Precision $= \frac{\text{TP}}{\text{TP+FP}}$ and Recall $= \frac{\text{TP}}{\text{TP+FN}}$. TP (true positive) means the number of true edges correctly estimated by the predicted differential network. FP (false positive) and FN (false negative) are the number of incorrectly predicted nonzero entries and zero entries respectively. We repeat the experiment 10 times for each method and use the average metrics for comparison. The better method achieves a higher F1-score.

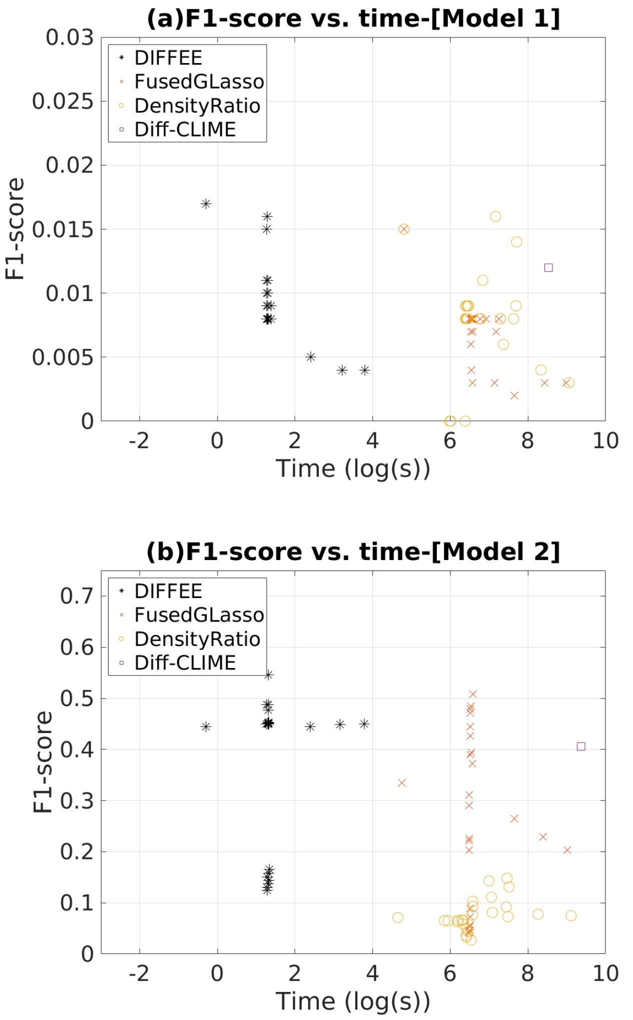
- Time Cost: We use the execution time (measured in seconds or log(seconds)) for a method as a measure of its scalability. To ensure a fair comparison, we try 30 different λ (or λ_2) and measure the total time of execution for each method. The better method uses less time[6].

- Low F1 values on Model 1 datasets: The F1-score of all cases in Figure 2(a) appear quite low. This is due to the fact that simulated differential networks from Model 1 are extremely sparse (e.g., only 0.1% non-zero edges among all possible edges). For example, if the estimated $\widehat{\Delta}$ only predicts 5% zero entries incorrectly (i.e., FP=5%) and correctly predicts all the rest entries (TP = 0.1%, TN = 94.9%). The precision equals to $\frac{\text{TP}}{\text{TP + FP}} = \frac{0.1\%}{0.1\%+5\%} \approx 0.02$, which is a small number. The recall equals to $\frac{\text{TP}}{\text{TP + FN}} = \frac{0.1\%}{0.1\%+0\%} = 1$. Then F1 $= \frac{\text{precision} \cdot \text{recall}}{2(\text{precision+recall})} \approx 0.01$, which is also a relatively small number. However, the estimator only wrongly inferred 5% zero entries, which is still a good result. Therefore, low F1-score doesn't mean that the estimator is bad when the differential network is extremely sparse.

  This extreme sparsity also influences other evaluation metrics. For instance, if the estimated $\widehat{\Delta}$ only includes 1% zero entries and 0.05% non-zero entries incorrectly (i.e., FP=5% and FN=0.05%) and correctly predicts all the rest entries (TP=0.05% and TN=94.9%). The TPR $= \frac{0.05\%}{0.05\%+0.05\%} = 0.5$ and FPR $= \frac{5\%}{5\%+94.9\%} \approx 0.2$. If you plot this point in the FPR vs. TPR curve, it is not good. However from the angle of accuracy, this method only predicts wrongly around 5% edges, which indicates that it performs well.

[6] The machine that we use for experiments is an Intel(R) Core(TM) i7-6850k CPU @ 3.60GHz with a 64GB memory.

Figure 4: F1-score versus Time Cost(log(seconds)) for different methods and synthetic data models (a) F1-score vs. Time for Model 1. (b)F1-score vs. Time for Model 2.



Simulated Data Generation: We first simulate precision matrices Ω_c and Ω_d by Model 1 or Model 2. To simulate data for the control block, we generate n_c data samples following multivariate gaussian distribution with mean 0 and covariance matrix $(\Omega_c)^{-1}$. We use the multivariate distribution method from stochastic simulation [24] to sample the simulated data blocks. In our implementation, we directly use the R function "**mvrnorm**" in **MASS** package. We repeat the same process for the case group with Ω_d. Then, we apply DIFFEE and baseline methods to obtain the estimated differential networks.

C Detailed Empirical Results

Figure 4 (a) and (b) summarize DIFFEE's better performance in both scalability and effectiveness for all experiment settings in Model 1 and Model 2, respectively. Each point in Figure4 represents both the F1-Score and Time Cost of a method. Most of the DIFFEE points lie in the top left area, indicating lesser Time Cost and higher F1-scores compared to the other baselines.

Table 3 and Table 4 present the detailed results on the simulated datasets, comparing the scalability to p of our proposed method DIFFEE with the baselines FusedGLasso, Density Ratio, and Diff-CLIME. The Table 3 and Table 4 are obtained by experimental settings under Model 1 and Model 2 respectively. We vary number of features p in the set of $\{100, 200, 300, 400, 500\}$. The computation time for each case is the summation of the computational time for the method over a range of $\lambda_n \in \{0.01 \times \sqrt{\frac{\log p}{\min(n_c, n_d)}} \times i | i \in \{1, 2, 3, \dots, 30\}\}$. The F1-score for each case is the best result over a range of $\lambda_n \in \{0.01 \times \sqrt{\frac{\log p}{\min(n_c, n_d)}} \times i | i \in \{1, 2, 3, \dots, 30\}\}$. The Diff-CLIME cannot finish any tasks in one day. So all the results in the column "Diff-CLIME" are indicated by "NA". In most of the synthetic datasets, DIFFEE achieves a higher F1-Score and less computation time than other baselines. This proves that DIFFEE outperforms the baselines in both effectiveness and scalability.

Table 5 and Table 6 present the detailed performance results of our proposed method DIFFEE and others by varying the sparsity level s. The Table 5 and Table 6 are obtained by Model 1 and Model 2 respectively. We vary the sparsity parameter s in the set of $\{0.1, 0.2, \dots, 0.7\}$. The computation time and F1-Score are measured similar to Table 3 and Table 4. In all of the synthetic datasets, DIFFEE performs better as indicated by its higher F1-score and lesser computation time than other baselines.

Table 7 and Table 8 present the detailed results of our proposed method–DIFFEE versus the corresponding baselines FusedGLasso, Density Ratio, and Diff-CLIME on the simulated datasets varying different n_c and n_d in a high-dimensional setting $(p > \max(n_c, n_d))$. The Table 7 and Table 8 are obtained by Model 1 and Model 2, respectively. We vary the number of samples n_c and n_d in the set of $\{p/2, p/4\}$. The computation time and F1-Score are measured similar to Table 3 and Table 4. In most of the synthetic datasets, DIFFEE achieves a higher F1-Score and less computation time than other baselines.

Table 9 and Table 10 present the performance of our proposed method–DIFFEE and other methods with varying n_c and n_d in a low-dimensional setting $(p > \max(n_c, n_d))$. The Table 9 and Table 10 correspond to Model 1 and Model 2, respectively. We vary the number of samples n_c and n_d in the set of $\{p, 2p, 3p\}$. The computation time and F1-Score are measured similar to Table 3 and Table 4. In most of the synthetic

Table 3: Model 1 varying p

	Model	DIFFEE	FusedGLasso	Slower	Density Ratio	Slower	Diff-CLIME	Slower
F1-score	p = 50	**0.029**	0		0.027		0.016	
	p = 100	**0.017**	0.015		0.015		0.012	
	p = 200	**0.009**	0.008		0.009		NA	
	p = 300	0.005	0.002		**0.006**		NA	
	p = 400	**0.004**	0.003		0.004		NA	
	p = 500	**0.004**	0.003		0.003		NA	
Time (s)	p = 50	**0.296**	45.61	154×	24.903	84×	56.37	190×
	p = 100	**0.748**	121.537	162×	122.596	163×	5094.796	6811×
	p = 200	**3.645**	715.672	196×	611.341	167×	NA	
	p = 300	**11.064**	2106.681	190×	1584.262	143×	NA	
	p = 400	**24.763**	4551.419	183×	4159.019	167×	NA	
	p = 500	**44.54**	8008.809	179×	8575.529	192×	NA	

Table 4: Model 2 varying p

	Model	DIFFEE	FusedGLasso	Slower	Density Ratio	Slower	Diff-CLIME	Slower
F1-score	p = 50	**0.581**	0.401		0.082		0.422	
	p = 100	**0.444**	0.335		0.071		0.406	
	p = 200	**0.45**	0.311		0.066		NA	
	p = 300	0.444	0.265		0.073		NA	
	p = 400	**0.449**	0.229		0.078		NA	
	p = 500	**0.45**	0.203		0.075		NA	
Time (s)	p = 50	**0.274**	43.57	159×	19.35	70×	116.712	425×
	p = 100	**0.751**	115.049	153×	104.53	139×	11640.82	15500×
	p = 200	**3.528**	657.147	186×	538.842	152×	NA	
	p = 300	**10.887**	2106.415	193×	1780.176	163×	NA	
	p = 400	**23.462**	4406.156	187×	3859.082	164×	NA	
	p = 500	**44.163**	8164.19	184×	9054.507	205×	NA	

Table 5: Model 1 varying sparsity

	Model	DIFFEE	FusedGLasso	Slower	Density Ratio	Slower
F1-score	s = 0.1	0.008	0.003		**0.009**	
	s = 0.2	**0.009**	0.008		0.009	
	s = 0.3	**0.008**	0.008		0.008	
	s = 0.4	**0.011**	0.008		0.008	
	s = 0.5	**0.008**	0.006		0.008	
	s = 0.6	**0.008**	0.008		0.008	
	s = 0.7	**0.008**	0.007		0.008	
Time (s)	s = 0.1	**3.606**	712.682	197×	631.582	175×
	s = 0.2	**3.993**	712.365	178×	598.191	149×
	s = 0.3	**3.97**	719.859	181×	595.246	149×
	s = 0.4	**3.65**	721.785	197×	598.009	163×
	s = 0.5	**3.632**	679.94	187×	631.062	173×
	s = 0.6	**3.693**	679.263	183×	608.358	164×
	s = 0.7	**3.679**	686.979	186×	624.632	169×

Table 6: Model 2 varying sparsity

	Model	DIFFEE	FusedGLasso	Slower	Density Ratio	Slower
F1-score	s = 0.1	**0.165**	0.089		0.066	
	s = 0.2	**0.158**	0.073		0.059	
	s = 0.3	**0.15**	0.057		0.05	
	s = 0.4	**0.144**	0.053		0.044	
	s = 0.5	**0.137**	0.042		0.036	
	s = 0.6	**0.13**	0.046		0.033	
	s = 0.7	**0.124**	0.043		0.027	
Time (s)	s = 0.1	**3.817**	671.255	175×	564.679	147×
	s = 0.2	**3.763**	671.499	178×	559.455	148×
	s = 0.3	**3.62**	674.941	186×	609.633	168×
	s = 0.4	**3.741**	664.363	177×	635.302	169×
	s = 0.5	**3.691**	662.802	179×	603.838	163×
	s = 0.6	**3.619**	659.336	182×	611.441	168×
	s = 0.7	**3.596**	648.885	180×	689.137	191×

Table 7: model1 varying n_c and n_d in high-dimensional setting

	Model	**DIFFEE**	FusedGLasso	Slower	Density Ratio	Slower
F1-score	$n_c = p/4, n_d = p/4$	**0.008**	0.008		0	
	$n_c = p/4, n_d = p/2$	**0.008**	0.008		0	
	$n_c = p/2, n_d = p/4$	**0.016**	0.008		0	
	$n_c = p/2, n_d = p/2$	**0.009**	0.008		0.009	
Time (s)	$n_c = p/4, n_d = p/4$	**3.647**	696.742	191×	398.226	109×
	$n_c = p/4, n_d = p/2$	**3.61**	704.943	195×	590.044	163×
	$n_c = p/2, n_d = p/4$	**3.609**	697.858	193×	408.149	113×
	$n_c = p/2, n_d = p/2$	**3.582**	654.147	182×	642.168	179×

Table 8: model2 varying n_c and n_d in high-dimensional setting

	Model	DIFFEE	FusedGLasso	Slower	Density Ratio	Slower
F1-score	$n_c = p/4, n_d = p/4$	**0.45**	0.221		0.065	
	$n_c = p/4, n_d = p/2$	**0.45**	0.226		0.063	
	$n_c = p/2, n_d = p/4$	**0.45**	0.29		0.065	
	$n_c = p/2, n_d = p/2$	**0.45**	0.203		0.066	
Time (s)	$n_c = p/4, n_d = p/4$	**3.74**	654.227	174×	381.686	102×
	$n_c = p/4, n_d = p/2$	**3.748**	654.822	174×	484.77	129×
	$n_c = p/2, n_d = p/4$	**3.717**	653.657	175×	346.148	93×
	$n_c = p/2, n_d = p/2$	**3.528**	657.147	186×	494.066	140×

datasets, DIFFEE achieves a higher F1-Score and less computation time than other baselines.

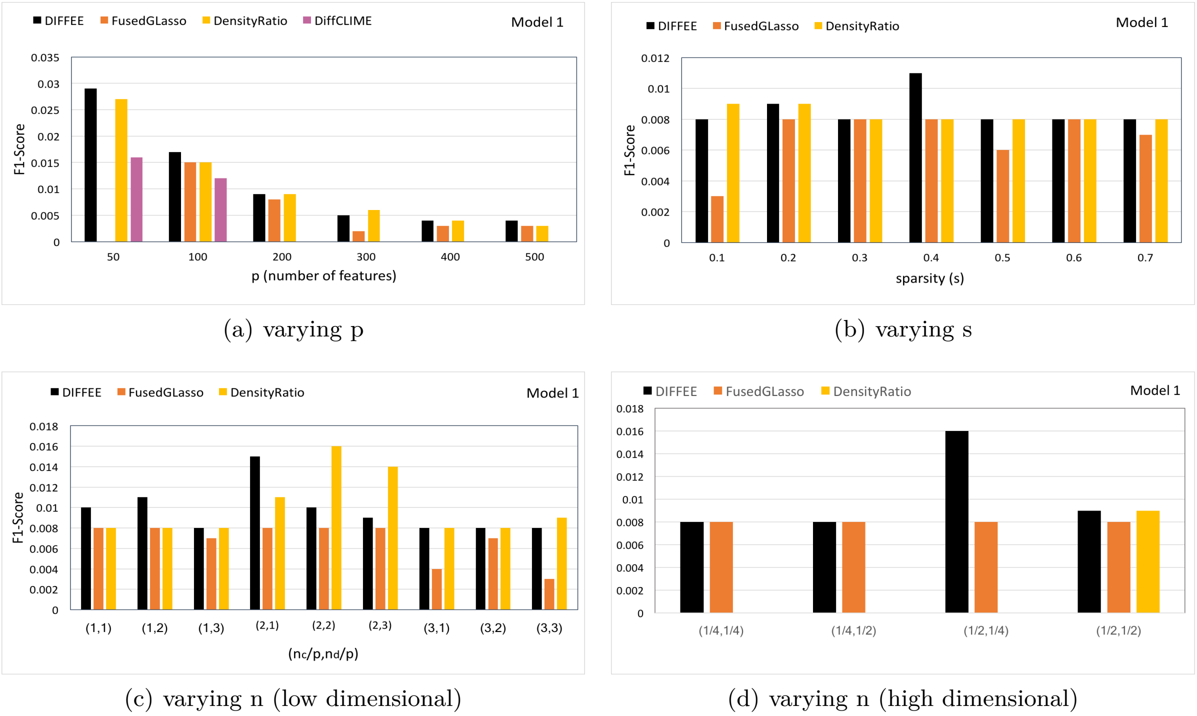
Figure 5 and Figure 6 summarize F1-Scores for DIFFEE and the baseline methods: FusedGLasso and DensityRatio for all simulations under varying p, s and (n_c, n_d) for Model 1 and Model 2, respectively.

Table 9: model1 varying n_c and n_d in low-dimensional setting

	Model	DIFFEE	FusedGLasso	Slower	Density Ratio	Slower
F1-score	$n_c = p, n_d = p$	**0.01**	0.008		0.008	
	$n_c = p, n_d = 2p$	**0.011**	0.008		0.008	
	$n_c = p, n_d = 3p$	**0.008**	0.007		0.008	
	$n_c = 2p, n_d = p$	**0.015**	0.008		0.011	
	$n_c = 2p, n_d = 2p$	0.01	0.008		**0.016**	
	$n_c = 2p, n_d = 3p$	0.009	0.008		**0.014**	
	$n_c = 3p, n_d = p$	**0.008**	0.004		0.008	
	$n_c = 3p, n_d = 2p$	**0.008**	0.007		0.008	
	$n_c = 3p, n_d = 3p$	0.008	0.003		**0.009**	
Time (s)	$n_c = p, n_d = p$	**3.643**	691.581	189×	838.863	230×
	$n_c = p, n_d = 2p$	**3.569**	1023.507	286×	1468.593	411×
	$n_c = p, n_d = 3p$	**3.62**	1319.354	364×	2054.228	567×
	$n_c = 2p, n_d = p$	**3.578**	700.539	195×	932.511	260×
	$n_c = 2p, n_d = 2p$	**3.568**	875.55	245×	1291.795	362×
	$n_c = 2p, n_d = 3p$	**3.553**	1406.44	395×	2224.744	626×
	$n_c = 3p, n_d = p$	**3.587**	696.087	194×	882.885	246×
	$n_c = 3p, n_d = 2p$	**3.578**	725.195	202×	1464.343	409×
	$n_c = 3p, n_d = 3p$	**3.592**	1264.346	351×	2191.003	609×

Table 10: model2 varying n_c and n_d in low-dimensional setting

	Model	DIFFEE	FusedGLasso	Slower	Density Ratio	Slower
F1-score	$n_c = p, n_d = p$	**0.45**	0.372		0.076	
	$n_c = p, n_d = 2p$	**0.453**	0.394		0.081	
	$n_c = p, n_d = 3p$	**0.452**	0.39		0.092	
	$n_c = 2p, n_d = p$	**0.451**	0.426		0.093	
	$n_c = 2p, n_d = 2p$	**0.477**	0.471		0.111	
	$n_c = 2p, n_d = 3p$	**0.488**	0.479		0.131	
	$n_c = 3p, n_d = p$	**0.452**	0.445		0.103	
	$n_c = 3p, n_d = 2p$	**0.488**	0.484		0.143	
	$n_c = 3p, n_d = 3p$	**0.546**	0.508		0.148	
Time (s)	$n_c = p, n_d = p$	**3.658**	707.735	193×	714.371	195×
	$n_c = p, n_d = 2p$	**3.746**	688.608	183×	1192.792	318×
	$n_c = p, n_d = 3p$	**3.673**	676.806	184×	1707.516	464×
	$n_c = 2p, n_d = p$	**3.69**	673.112	182×	723.656	196×
	$n_c = 2p, n_d = 2p$	**3.691**	676.597	183×	1164.175	315×
	$n_c = 2p, n_d = 3p$	**3.57**	677.65	189×	1830.678	512×
	$n_c = 3p, n_d = p$	**3.692**	673.364	182×	717.752	194×
	$n_c = 3p, n_d = 2p$	**3.692**	682.499	184×	1090.64	295×
	$n_c = 3p, n_d = 3p$	**3.732**	719.733	192×	1739.274	466×



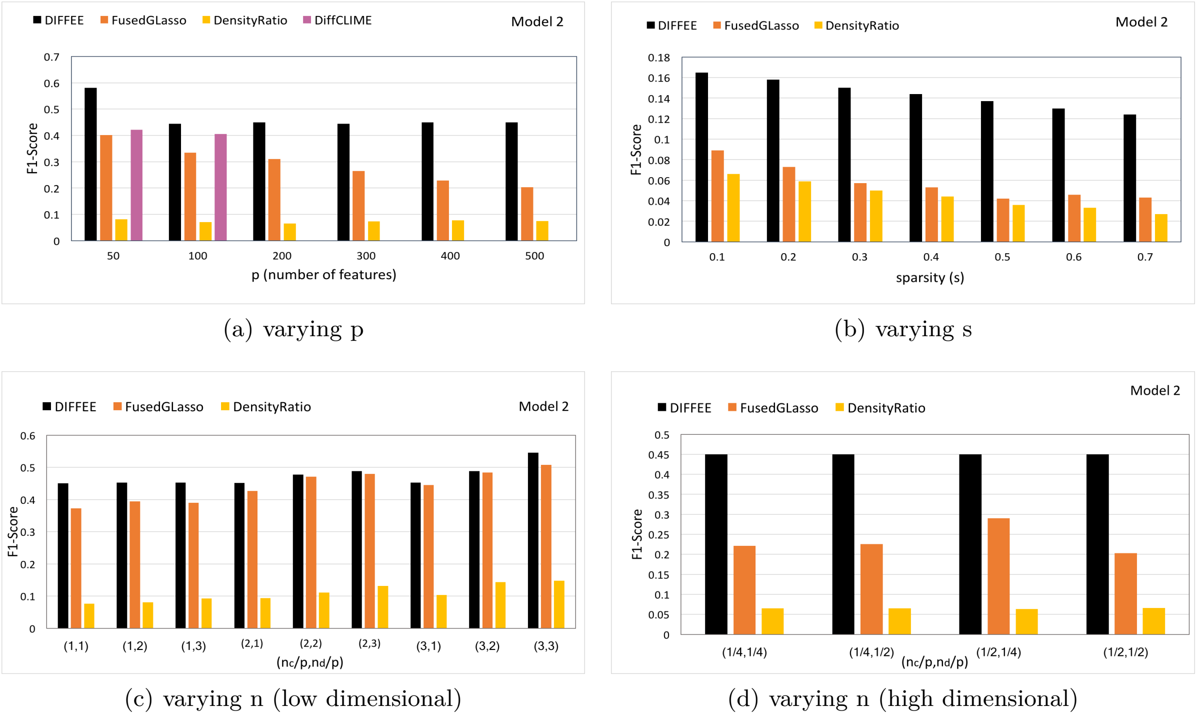
(a) varying p

(b) varying s

(c) varying n (low dimensional)

(d) varying n (high dimensional)

Figure 5: F1-Score of DIFFEE and baseline methods for Simulated Model 1



(a) varying p

(b) varying s

(c) varying n (low dimensional)

(d) varying n (high dimensional)

Figure 6: F1-Score of DIFFEE and baseline methods for Simulated Model 2