Differentially Private and Federated Structure Learning in Bayesian Networks

Ghita Fassy El Fehri
PreMeDICal, Inria, Idesp,
Inserm, University of Montpellier

Aurélien Bellet
PreMeDICal, Inria, Idesp,
Inserm, University of Montpellier

Philippe Bastien
L'Oréal Research and Innovation,
Aulnay-sous-Bois, France

Abstract

Learning the structure of a Bayesian network from decentralized data poses two major challenges: (i) ensuring rigorous privacy guarantees for participants, and (ii) avoiding communication costs that scale poorly with dimensionality. In this work, we introduce Fed-Sparse-BNSL, a novel federated method for learning linear Gaussian Bayesian network structures that addresses both challenges. By combining differential privacy with greedy updates that target only a few relevant edges per participant, Fed-Sparse-BNSL efficiently uses the privacy budget while keeping communication costs low. Our careful algorithmic design preserves model identifiability and enables accurate structure estimation. Experiments on synthetic and real datasets demonstrate that Fed-Sparse-BNSL achieves utility close to non-private baselines while offering substantially stronger privacy and communication efficiency.

1 INTRODUCTION

Bayesian networks (BNs) compactly represent joint distributions and causal dependencies among random variables. Their graph structure, in the form of a directed acyclic graph (DAG), encodes conditional independencies, yielding interpretable models widely used in biology, medicine, and social sciences. A key challenge is to infer the BN structure directly from data, a task known as Bayesian network structure learning (BNSL). Among existing approaches, continuous optimization methods have shown strong empirical performance on linear Gaussian BNs, where each variable is modeled as a linear function of its parents plus Gaussian noise (Zheng et al., 2018; Ng et al., 2019; Zheng et al., 2020; Ng et al., 2020; Vowels et al., 2021). However, these methods assume centralized access to data. In many real-world scenarios, data are distributed across multiple institutions or participants and cannot be aggregated due to privacy concerns. This motivates the study of federated BNSL, which aims to collaboratively reconstruct a global DAG from decentralized data (Ng and Zhang, 2022). Federated BNSL faces two key challenges: (i) ensuring strong privacy guarantees for participants' data, and (ii) preventing communication costs from growing excessively with dimensionality. These challenges are further compounded by the fact that the number of possible edges in a DAG grows quadratically with the number of variables.

In this work, we address both challenges with Fed-Sparse-BNSL, a federated method for learning linear Gaussian Bayesian network structures, and its privacy-preserving variant DP-Fed-Sparse-BNSL. Our approach combines differential privacy (DP) with a greedy, coordinate-wise update strategy that restricts communication to only a few relevant edges per participant at each iteration. This leads to two key advantages compared to prior work. First, communication costs are greatly reduced since sparse edge updates are transmitted instead of dense matrices. Second, privacy budget scales with the number of updated dependencies per participant, not with the full dimension. The core component of our method is a Proximal Greedy Coordinate Descent (PGCD) algorithm for the local updates, which produces high-quality sparse solutions without requiring data standardization—which is known to compromise the identifiability of the DAG (Ng et al., 2024; Loh and Bühlmann, 2014). For the private variant, DP-Fed-Sparse-BNSL, we improve upon the Differentially Private Proximal Greedy Coordinate Descent (DP-PGCD) algorithm (Mangold et al., 2023) by minimizing the noise added under DP, leveraging mechanisms that can be analyzed un-

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

der zero-concentrated differential privacy (zCDP) for tighter privacy accounting. This results in favorable privacy-utility trade-off, as demonstrated by both theoretical guarantees and empirical results. Besides the above benefits, we empirically show that Fed-Sparse-BNSL supports participant-level personalization in heterogeneous settings, where the structure is shared but edge weights differ across participants.

Contributions. Our main contributions can be summarized as follows: (i) a communication-efficient federated BNSL algorithm that exploits sparsity, reducing communication costs; (ii) a differentially private variant with formal (ε, δ)-DP guarantees, that achieves strong utility even in high-dimensional settings; (iii) an empirical evaluation on synthetic homogeneous and heterogeneous data as well as a real dataset, demonstrating the effectiveness of our method in terms of convergence, communication costs, privacy-utility trade-offs, robustness to dimensionality, and effective participant-level personalization.

Organization of the paper. The paper is organized as follows. Section 2 formalizes the problem setting; Section 3 reviews related work; Section 4 presents the federated algorithm Fed-Sparse-BNSL; Section 5 introduces the private variant DP-Fed-Sparse-BNSL and its guarantees; and finally, Section 6 reports experimental results, and Section 7 concludes.

2 PROBLEM SETTING

Structure learning in linear Gaussian BNs. We consider the task of learning the structure of a Bayesian Network (BN) from observational data. A BN (Koller and Friedman, 2009) is a directed acyclic graph (DAG) where each node X_i represents a random variable, and edges encode conditional dependencies. We represent a (weighted) DAG using a weighted adjacency matrix $W \in \mathbb{R}^{d\times d}$, where each entry $W_{i,j}$ encodes the strength of a directed edge from variable X_i to X_j. A nonzero $W_{i,j}$ indicates a direct causal effect of X_i on X_j, whereas $W_{i,j} = 0$ indicates no direct effect.

We focus on linear Gaussian BNs, where each variable is a linear function of its parents in the DAG plus additive Gaussian noise:

$$X_i = \sum_{j\in \mathrm{Pa}(X_i)} W_{ji} X_j + Z_i, \qquad Z_i \sim \mathcal{N}(0, \sigma_i^2),$$

where $\mathrm{Pa}(X_i) = \{X_j : W_{j,i} \neq 0\}$ denotes the set of parents of X_i in the DAG.

The structure learning problem is to recover W from a dataset $X \in \mathbb{R}^{n\times d}$, which is known to be identifiable when noise variances are equal or known (Peters and Bühlmann, 2013; Loh and Bühlmann, 2014). The combinatorial nature of the problem is addressed by reformulating it as a continuous optimization task using the NOTEARS framework (Zheng et al., 2018):

$$\min_{W\in\mathbb{R}^{d\times d}} L(W;X) + \lambda\|W\|_1 \qquad \text{s.t.} \quad h(W) = 0, \quad (1)$$

where $L(W;X) = \frac{1}{2n}\|X - XW\|_F^2$ is the least squared loss, $\lambda\|W\|_1$ promotes sparsity, and

$$h(W) = \mathrm{tr}(e^{W\circ W}) - d = 0$$

ensures acyclicity, where $\circ$ is the Hadamard product.

Federated learning. In this work, we consider a setting with P participants, each holding a private local dataset $X^{(p)} = \{X_1^{(p)}, \ldots, X_{n_p}^{(p)}\} \subset \mathbb{R}^d$, with n_p the number of samples held by participant p. The goal is to learn a shared DAG structure with the coordination of a central server, while keeping all data decentralized.

Problem (1) can then be equivalently reformulated as a consensus problem (Ng and Zhang, 2022):

$$\begin{aligned} \min_{B,W} \quad & \mathcal{L}(B;W) = \textstyle\sum_{p=1}^{P} L(B_p; X^{(p)}) + \lambda\|W\|_1 \\ \text{s.t.} \quad & B_p = W, \qquad \forall p \in \{1,\ldots,P\}, \\ & h(W) = 0, \end{aligned} \quad (2)$$

where $B = [B_1, \ldots, B_P]$ and $B_p \in \mathbb{R}^{d\times d}$ is the local matrix for participant p and W the global consensus adjacency matrix.

Differential privacy. To protect the participants' data in the federated learning process, we aim to enforce formal differential privacy (DP) guarantees (Dwork et al., 2006). Intuitively, a mechanism satisfies DP if replacing a single data point has only a limited impact on the algorithm's output distribution.

Definition 2.1 (Differential privacy)**.** *Let $\varepsilon, \delta > 0$. A randomized algorithm $\mathcal{A}$ satisfies (ε, δ)-DP if, for any two datasets D_1 and D_2 of fixed size that differ in exactly one record, and for any possible $O \subseteq \mathrm{range}(\mathcal{A})$,*

$$P(\mathcal{A}(D_1) \in O) \leq e^{\varepsilon} P(\mathcal{A}(D_2) \in O) + \delta.$$

In our federated setting, we aim to limit the influence of any single participant's data on the information shared during training, thereby providing formal privacy guarantees against both external observers and an honest-but-curious server.

3 RELATED WORK

Bayesian networks structure learning (BNSL) aims at recovering the graph structure of a BN from observational data (Koller and Friedman, 2009). Traditional structure learning approaches are based on

discrete optimization and can be categorized into two families: constraint-based methods, which rely on conditional independence tests, and score-based methods, which optimize a goodness-of-fit score. For a detailed survey, we refer the reader to Kitson et al. (2023).

Recently, continuous optimization methods have gained popularity, notably with the introduction of NOTEARS (Zheng et al., 2018), which proposed the first differentiable acyclicity constraint in the context of linear Gaussian BNs. This allows BNSL to be formulated as a continuous optimization problem, avoiding the combinatorial search of discrete methods. Several extensions of NOTEARS have since been proposed (Ng et al., 2019; Zheng et al., 2020; Ng et al., 2020), see also Vowels et al. (2021, Section 5). However, these methods require centralized access to data.

Federated BNSL extensions have been developed for both continuous and discrete optimization approaches. Discrete optimization methods (van Daalen et al., 2024; Torrijos et al., 2024) are primarily designed for discrete data and are not directly applicable to continuous data, which is the focus of our work. For continuous optimization, Fed-BNSL (Ng and Zhang, 2022) adapts the NOTEARS formulation to the federated setting via the alternating direction method of multipliers (ADMM) to solve (2), enabling participants to collaboratively learn a consensus DAG without centralizing their data. FedDAG (Gao et al., 2023), built on the GOLEM framework (Ng et al., 2020), relies on FedAvg (McMahan et al., 2017) to learn and aggregate local structures across participants. Both methods require participants to transmit $d \times d$ matrices at each round, which becomes impractical in high-dimensions, and neither provides formal privacy guarantees.

Differentially private BNSL methods have been proposed in the context of synthetic data generation. PrivBayes (Zhang et al., 2017) and subsequent work (Bao et al., 2021) propose approaches to learn BNs under DP, enabling the release of synthetic data with formal privacy guarantees. Related efforts have appeared in the causal discovery community (Wang et al., 2020). However, these methods are designed for discrete data, so they cannot be applied to our continuous setting, and assume centralized access to all data.

Federated and private BNSL has been explored in recent work. In (Mian et al., 2023), participants compute scores on local DAGs and send noisy scores to a central server, which then identifies a consensus DAG minimizing a notion of regret. However, the method lacks explicit sensitivity bounds for DP guarantees and does not scale, requiring as many communication rounds as graph edges. Wang et al. (2023) extend constraint-based methods to the federated setting by performing conditional independence tests locally and aggregating results via secure computation, but this approach is again limited to discrete data.

4 FEDERATED ALGORITHM

In existing federated BNSL approaches, each participant must transmit a dense $d \times d$ matrix to the central server (Ng and Zhang, 2022; Gao et al., 2023), resulting in a communication cost that scales quadratically with the number of variables, despite the underlying DAG typically being sparse. In this section, we propose a communication-efficient alternative that exploits this sparsity, which is also crucial for the design of an effective differentially private version (Section 5).

4.1 Algorithm Overview

We first slightly modify the formulation (2) of Ng and Zhang (2022), redefining the objective function $\mathcal{L}(B, W)$ by transferring the ℓ_1 sparsity penalty from the global consensus matrix W to the local matrices:

$$\begin{aligned} \min_{B,W} \quad & \mathcal{L}(B; W) = \textstyle\sum_{p=1}^{P} L(B_p; X^{(p)}) + \lambda \|B_p\|_1 \\ \text{s.t.} \quad & B_p = W, \qquad \forall p \in \{1, \dots, P\}, \\ & h(W) = 0. \end{aligned} \tag{3}$$

Note that this reformulation yields an equivalent problem because of the consensus constraints in (2). Following the approach of Fed-BNSL, we solve this constrained optimization problem in a federated way using the ADMM-based augmented Lagrangian method, leading to the following iterative update rules:

$$\begin{aligned} B_p^{t+1} &= \arg\min_{B_p} (L(B_p; X^{(p)}) + \lambda \|B_p\|_1 \\ &\quad + \operatorname{tr}\left(\beta_p^t (B_p - W^t)^\top\right) + \frac{\rho_2}{2} \|B_p - W^t\|_F^2), \end{aligned} \tag{4}$$

$$\begin{aligned} W^{t+1} &= \arg\min_W (\textstyle\sum_{p=1}^{P} \operatorname{tr}\left(\beta_p^t (B_p^{t+1} - W)^\top\right) \\ &\quad + \tfrac{\rho_2^t}{2} \textstyle\sum_{p=1}^{P} \|B_p^{t+1} - W\|_F^2 + \alpha^t h(W) + \tfrac{\rho_1}{2} h(W)^2), \end{aligned} \tag{5}$$

$$\alpha^{t+1} = \alpha^t + \rho_1 h(W^{t+1}), \tag{6}$$

$$\beta_p^{t+1} = \beta_p^t + \rho_2 (B_p^{t+1} - W^{t+1}), \tag{7}$$

where $\rho_1, \rho_2 > 0$ are the penalty coefficients and $\alpha \in \mathbb{R}$ and $\beta_1, \dots, \beta_P \in \mathbb{R}^{d \times d}$ are the dual variables (Lagrange multipliers). Algorithm 1 summarizes the proposed federated BNSL algorithm.

Imposing the ℓ_1 penalty directly on the local matrices leverages the inherent sparsity of Bayesian networks to reduce communication. Since solutions to the local subproblem (4) are expected to be sparse, participants only need to exchange the nonzero entries of their estimates, reducing the communication cost from $O(d^2)$ to

Algorithm 1 Fed-Sparse-BNSL

Initialize $B_p^0 = 0, W^0 = 0, \alpha^0 = 0, \beta_p^0 = 0$
for $t = 1$ **to** T **do**
  for each participant p in parallel **do**
    Update B_p^{t+1} by solving Eq. 4 with PGCD
    Send sparse update B_p^{t+1} to server
  end for
  Update W^{t+1} by solving Eq. 5 with L-BFGS
  Update dual variables $\alpha^{t+1}, \beta_p^{t+1}$ (Eq. 6-7)
  Server sends back sparse W^{t+1} to participants
end for

the number of identified dependencies. This sparsity propagates to the shared structure W: for all entries that are zero across the local matrices, any local minimizer of (5) will likewise assign a zero to that entry.

It remains to select the appropriate local solver for (4), which is a subtle but crucial design choice.

4.2 Choice of Local Solver

The local subproblem (4) is a LASSO-type optimization problem, for which many solvers exist (Hastie et al., 2015; Jaggi, 2013). However, these solvers typically assume that data has been standardized (features centered and rescaled). Standardization ensures that all variables are on a comparable scale, allowing the ℓ_1 penalty to be applied uniformly across coefficients (Hastie et al., 2015). However, in the context of linear Gaussian structural equation model with equal error variances (Peters and Bühlmann, 2013), standardization is known to invalidate the theoretical conditions under which the true DAG can be identified (Ng et al., 2024; Loh and Bühlmann, 2014) We therefore require a solver that can meaningfully handle LASSO problems without rescaling the data.

For this reason, we rely on the *Proximal Greedy Coordinate Descent* (PGCD) algorithm (Tseng and Yun, 2009; Nutini et al., 2015; Karimireddy et al., 2019). PGCD is an iterative method that updates one coordinate at a time, selecting the coordinate that yields the greatest potential improvement in the objective. Consider participant p with initialization $B^0 \in \mathbb{R}^{d\times d}$. At each iteration k, PGCD evaluates each coordinate (i, j) using the score

$$S_{i,j} = \sqrt{M_{i,j}}\Big| \operatorname{prox}_{\frac{\lambda}{M_{i,j}}|\cdot|}\Big(B_{i,j}^k - \frac{1}{M_{i,j}}\big(\nabla_{i,j}\mathcal{L}(B^k; X^{(p)})\big)\Big) - B_{i,j}^k\Big|, \tag{8}$$

where $\mathcal{L}(B^k; X^{(p)}) = L(B^k; X^{(p)}) + \operatorname{tr}(\beta(B^k - W)^\top) + \frac{\rho}{2}\|B^k - W\|_F^2$ is the smooth part of the local objective, $\nabla_{i,j}\mathcal{L}(B^k; X^{(p)})$ is the partial derivative of $\mathcal{L}$ with respect to $B_{i,j}^k$, $\operatorname{prox}_{\lambda|\cdot|}(x) = \operatorname{sign}(x)\cdot\max(|x| - \lambda, 0)$ is the soft-thresholding operator, and $M_{i,j}$ is the coordinate-wise smoothness constant of $\mathcal{L}$. The coordinate with the highest score, $(m, n) = \arg\max_{i,j} S_{i,j}$, is then greedily updated by a proximal gradient step:

$$B_{m,n}^{k+1} = \operatorname{prox}_{\frac{\lambda\gamma}{M_{m,n}}|\cdot|}\big(B_{m,n}^k - \frac{\gamma}{M_{m,n}}\nabla_{m,n}\mathcal{L}(B^k; X^{(p)})\big) \tag{9}$$

PGCD is particularly well-suited to our setting for two reasons. First, the coordinate-wise smoothness constants naturally normalize updates, eliminating the need for standardization and thereby preserving the identifiability conditions of the underlying DAG. We empirically validate the importance of this property in Appendix C.1, where we show that classical solvers fail on raw data and that standardization compromises structure recovery. Second, the greedy coordinate selection combined with proximal updates enforces sparsity, retaining only the most relevant dependencies, thereby significantly reducing communication costs.

Furthermore, because the ℓ_1-regularized local subproblem (4) satisfies both coordinate-wise smoothness and strong convexity, the PGCD algorithm benefits from a linear convergence guarantee to the exact local optimum (Karimireddy et al., 2019, Theorem 1).

In the next section, we further leverage the properties of PGCD to ensure the privacy budget is effectively spent on the most informative updates—a key factor for achieving good privacy-utility trade-offs in high-dimensional settings.

5 PRIVATE ALGORITHM

5.1 Motivation

In the original Fed-BNSL algorithm (Ng and Zhang, 2022), the local subproblem in B_p—corresponding to (4) without the ℓ_1 penalty—is solved via the closed-form expression:

$$B_p^{t+1} = (\Sigma_p + \rho_2 I)^{-1}(\rho_2 W^t - \beta_p^t + \Sigma_p), \tag{10}$$

where $\Sigma_p = \frac{1}{n_p} X^{(p)\top} X^{(p)} \in \mathbb{R}^{d\times d}$ denotes the empirical covariance matrix of participant p. Since the server receives B_p and has access to W^t, ρ_2, and β_p^t, it can directly reconstruct Σ_p, as detailed in Appendix A.1. This poses a significant privacy risk: covariance matrices can leak detailed information about individual data points, enabling attribute inference and data reconstruction attacks (Creţu et al., 2024; Huth et al., 2023). Although methods exist for constructing differentially private covariance matrices (Wang, 2018; Amin et al., 2019), they become impractical for high-dimensional datasets unless additional assumptions are made, since

Algorithm 2 DP-PGCD

Initialize $B^0 = 0$ (or use warm-start)
for $k = 1$ **to** K **do**
  Compute noisy scores $\tilde{S}_{i,j}$ $\forall (i,j)$ (Eq. 11)
  Select (m,n) with highest noisy score
  Update $B^{k+1}_{m,n}$ (Eq. 12)
  $B^{k+1}_{i,j} = B^k_{i,j}$ for $(i,j) \neq (m,n)$
end for
Output: Sparse matrix B^K

the privacy-induced error grows quadratically with the number of variables d (Amin et al., 2019).

Our approach, presented in Section 4, naturally produces sparse updates for B_p, which reduces the amount of information exposed to the server. Nevertheless, it is well known that sharing model updates in federated learning can still be vulnerable to a range of privacy attacks (Nasr et al., 2019; Geiping et al., 2020). In this section, we introduce a differentially private variant of our federated BNSL algorithm that leverages the greedy updates of PGCD, avoiding the quadratic cost in the dimensionality p.

5.2 Differentially Private Local Solver

In Fed-Sparse-BNSL, the only part that directly accesses the data is the participants' local update (4). Therefore, it is sufficient to privatize this step; the post-processing and composition properties of differential privacy then provide guarantees for the overall algorithm. We thus propose to rely on a differentially private version of our local solver, PGCD.

Mangold et al. (2023) introduced a private version of PGCD which relies on the addition of Laplace noise to privatize both the greedy coordinate selection and the gradient update, providing pure ϵ-DP guarantees, and resorted to the advanced composition theorem (Dwork and Roth, 2014) to track the privacy loss across iterations. However, this composition is overly pessimistic, requiring unnecessarily large amounts of noise at each step and resulting in a poor privacy–utility trade-off, as the excessive noise degrades the quality of the learned matrix.

In contrast, our improved DP-PGCD uses the exponential mechanism via the "Gumbel max trick" (Ding et al., 2021) for private coordinate selection and Gaussian noise for gradient updates. This design enables us to leverage zero-Concentrated Differential Privacy (zCDP) (Bun and Steinke, 2016), under which the exponential mechanism admits a tighter privacy analysis (Dong et al., 2020) and composition behaves more favorably. We thus achieve tighter privacy accounting and require less noise for the same privacy budget. A detailed theoretical and numerical comparison with the approach of Mangold et al. (2023) is provided in Appendices A.4 and C.4, demonstrating a provable improvement in privacy loss of at least a factor of $\sqrt{2}$ for the same noise variance.

Our DP-PGCD algorithm (Algorithm 2) operates similarly to its non-private counterpart, with two key modifications. First, it uses noisy scores computed by adding independent Gumbel noise to the non-private scores defined in (8):

$$\tilde{S}_{i,j} = S_{i,j} + \text{Gumbel}(0, \beta). \tag{11}$$

Second, the gradient used in the coordinate update (9) is perturbed with Gaussian noise:

$$B^{k+1}_{m,n} = \text{prox}_{\frac{\lambda\gamma}{M_{m,n}}|\cdot|}\big(B^k_{m,n} - \frac{\gamma}{M_{m,n}}(\nabla_{m,n}\mathcal{L}(B^k; X^{(p)}) + \mathcal{N}(0, \sigma^2))\big). \tag{12}$$

Note that this private version reduces to the non-private algorithm when $\beta = \sigma = 0$.

Each iteration of DP-PGCD computes the full gradient but updates only a single coordinate. While using the full gradient would require privatizing all d^2 coordinates, even though only a few are typically relevant in sparse DAGs, the exponential mechanism allows PGCD to select the most promising coordinate while incurring a privacy cost of only $O(\log d)$ (Dwork and Roth, 2014). In the following, we refer to DP-Fed-Sparse-BNSL as the Fed-Sparse-BNSL algorithm using DP-PGCD as the local solver.

5.3 Privacy Guarantees

We provide DP guarantees for DP-Fed-Sparse-BNSL.

Theorem 5.1 (Privacy of DP-Fed-Sparse-BNSL)**.** *Let $\varepsilon, \delta > 0$ and $\Delta = 2L_{i,j}/n_p$ where $L_{i,j}$ is the coordinate-wise Lipschitz constant of $\mathcal{L}$. Suppose DP-Fed-Sparse-BNSL (Algorithm 1) runs for T global rounds, with local updates performed with K iterations of DP-PGCD (Algorithm 2). If the Gumbel and Gaussian noise parameters are chosen as*

$$\beta = \frac{\sigma}{\sqrt{M_{i,j}}} = \frac{\Delta\sqrt{KT}(\sqrt{\log(\frac{1}{\delta}) + \varepsilon} + \sqrt{\log(\frac{1}{\delta})})}{\varepsilon},$$

then DP-Fed-Sparse-BNSL is (ε, δ)-differentially private with respect to each participant's dataset.

Sketch of proof. We first show that gradients have ℓ_2-sensitivity bounded by Δ, and that scores have sensitivity bounded by $\Delta/\sqrt{M_{i,j}}$, leveraging the non-expansiveness of the prox. Each DP-PGCD iteration applies two private mechanisms: (i) coordinate

selection via the exponential mechanism (with Gumbel noise), and (ii) coordinate update via Gaussian noise. With $\beta = \sigma/\sqrt{M_{i,j}}$, each one satisfies ρ-zCDP, where $\rho = \frac{\Delta^2}{2\beta^2}$. By additive composition, running T rounds of DP-Fed-Sparse-BNSL, each with K updates of DP-PGCD, yields total privacy loss $\rho_{\text{tot}} = 2K\rho = \frac{K\Delta^2}{\beta^2}$. Converting to (ε, δ)-DP gives $\varepsilon = \rho_{\text{tot}} + 2\sqrt{\rho_{\text{tot}} \log(1/\delta)}$, which determines the stated noise scales. Full derivations are in Appendix A.2. □

Gradient clipping. Since coordinate-wise Lipschitz constants are difficult to bound tightly, in practice we instead enforce bounded sensitivity through gradient clipping. Following Mangold et al. (2022), each gradient coordinate $\nabla_{i,j}\mathcal{L}(B^k; X^{(p)})$ is clipped at $C_{i,j} = \sqrt{M_{i,j}/\sum_{i,j} M_{ij}}C$ for some global threshold $C > 0$. This scheme adjusts for the relative scale of each coordinate while reducing tuning to a single parameter.

Knowledge of $M_{i,j}$. We assume that the coordinate-wise smoothness constants $M_{i,j}$ are available (or upper bounded). When this is not the case, they can be privately estimated using the Gaussian mechanism. Further details are provided in Appendices A.3 and C.3.

5.4 Extension to More General Bayesian Networks

While this work focuses on linear Gaussian BNs—a setting where DAG identifiability is well understood and strong baselines allow us to isolate the contribution of our FL/DP design—our framework is not fundamentally limited to this class.

Our approach can be naturally extended to broader classes of BNs, provided the following conditions hold: (i) the underlying DAG is identifiable under suitable assumptions, (ii) the global loss $L(W; X)$ decomposes as a sum of local losses $L(B_p; X^{(p)})$ across participants, (iii) the smooth part of each local score $L(B_p; X^{(p)})$ is differentiable and coordinate-wise smooth (so that the PGCD solver can be applied), and (iv) the corresponding gradients can be bounded to derive sensitivity bounds for the DP mechanisms.

When these conditions are satisfied, extending our framework only requires replacing the squared loss with the appropriate score function and re-deriving the associated smoothness constants and sensitivity bounds.

6 EXPERIMENTS

We empirically evaluate Fed-Sparse-BNSL and its differentially private variant DP-Fed-Sparse-BNSL on synthetic data and on real data. We assess the effectiveness of our method in terms of convergence, structural accuracy of the estimated DAG, communication efficiency and privacy-utility trade-offs. The code is available at `https://gitlab.com/ghitafassy/fed-sparse-bnsl/`.

6.1 Experimental Setup

Datasets We consider the following datasets.

Homogeneous synthetic data. Following Zheng et al. (2018) and Ng and Zhang (2022), we generate Erdös-Rényi random DAGs with $d = 20$ nodes and an expected number of d edges. Observations are drawn from a linear Gaussian BN with equal noise variance across variables. Data are partitioned across $P = 8$ participants, each with $n_p = 5000$ samples. We also consider higher-dimensional versions with d up to 200. Details are provided in Appendix B.1.1.

Heterogenous synthetic data. We use the same DAG generation, but allow participant-specific regression coefficients, while keeping the underlying DAG structure shared. We consider $P = 5$ participants, each with $n_p = 5000$ samples. This setting allows to evaluate both the structural accuracy of the estimated shared DAG, and the ability to estimate participant-specific coefficients. Details are in Appendix B.1.2.

Real data. As in previous work (Zheng et al., 2018; Ng and Zhang, 2022), we use the Sachs protein signaling dataset (Sachs et al., 2005), which has $d = 11$ variables, $n = 7466$ samples, and a ground-truth DAG with 18 edges. To simulate a federated setting, we split the data across $P = 3$ participants.

Metrics. We consider the following metrics.

Structural accuracy. We assess the quality of the estimated DAGs using standard structure learning metrics. The Structural Hamming Distance (SHD) counts the number of edge insertions, deletions or reversals needed to convert the estimated DAG into the true one. The True Positive Rate (TPR) represents the proportion of true edges that are correctly recovered, while the False discovery rate (FDR) represents the proportion of incorrect edges among predicted edges.

Communication cost. We measure the total cost as the number of bytes transmitted between the server and the participants throughout the entire training procedure, reported in megabytes (MB).

Personalization. In the heterogeneous setting, after learning the shared structure, we compute per-participant normalized mean squared error (MSE) between the true regression coefficients and those estimated after local refitting.

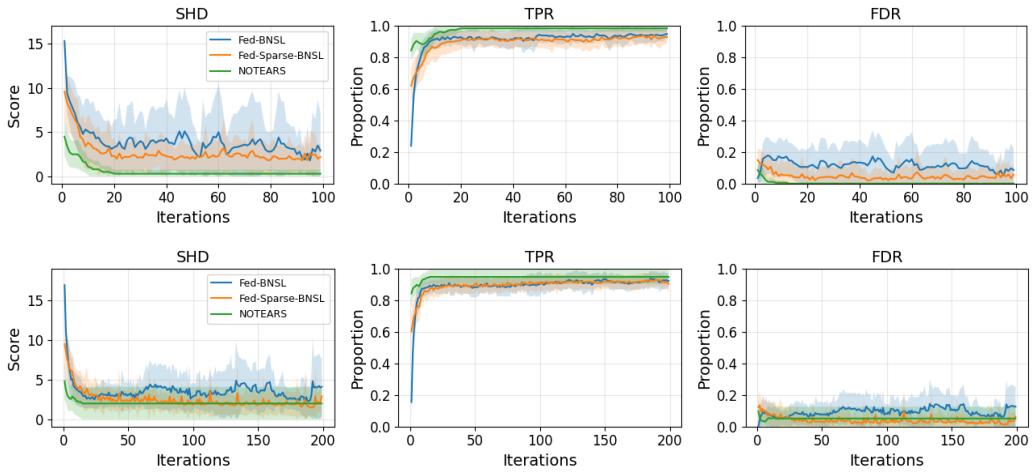


Figure 1: Convergence of Fed-Sparse-BNSL, Fed-BNSL and centralized NOTEARS. Top: homogeneous synthetic data; Bottom: heterogeneous synthetic data. We report SHD, TPR and FDR across iterations.

Evaluation protocol. We report the mean and standard deviation (shaded bands in figures) across the 10 runs. For each configuration of dataset and privacy budget, hyperparameters are tuned separately for each method. The procedure, search ranges and final values are detailed in Appendix B.3.

6.2 Non-Private Setting

We evaluate the performance of Fed-Sparse-BNSL on synthetic data and compare to Fed-BNSL (Ng and Zhang, 2022). We also include centralized NOTEARS as a reference point, which corresponds to running NOTEARS on the union of all participants' data and thus represents the best achievable performance.

Convergence on homogeneous data. Figure 1 (top) shows that Fed-Sparse-BNSL consistently achieves lower SHD than Fed-BNSL across iterations and stabilizes earlier, indicating faster and more stable convergence. Variability is also narrower for Fed-Sparse-BNSL. Both methods quickly reach high TPR and remain close thereafter, but Fed-Sparse-BNSL maintains a lower FDR throughout training. This gap persists over iterations and is accompagnied by tighter variability, suggesting that Fed-Sparse-BNSL avoids over-selecting spurious edges while converging to more accurate structures.

Convergence on heterogeneous data. As shown in Figure 1 (bottom), Fed-Sparse-BNSL maintains its advantages on heterogeneous data: it achieves lower SHD with earlier stabilization, matches Fed-BNSL in TPR, and consistently attains lower FDR, demonstrating accurate and robust recovery of the consensus structure despite cross-participant variability.

Communication efficiency. Communication costs are reported in Table 1. For $d = 20$, Fed-Sparse-BNSL reduces communication by approximately 64% compared to Fed-BNSL, while maintaining strong structural accuracy. As expected, for $d = 200$, the difference is even more pronounced: Fed-Sparse-BNSL achieves roughly 97.2% reduction in communication costs and clearly outperforms Fed-BNSL, with lower SHD and FDR while keeping TPR high. These results highlight the effectiveness of Fed-Sparse-BNSL 's design, demonstrating substantial communication savings without sacrificing performance.

Participant-level personalization. For this experiment, we consider a low-sample setting with 20 participants, each holding only 50 samples. We compare three approaches: (i) Fed-Sparse-BNSL without personalization, using the consensus weighted adjacency matrix learned collectively, (ii) Fed-Sparse-BNSL with personalization, where each participant locally refits the edge weights via linear regression on its own data after fixing the learned DAG structure and (iii) local NOTEARS, where each participant independently runs centralized NOTEARS on its own data. Figure 2 shows the per-participant normalized MSE between estimated and ground-truth participant-specific weights for each approach, showing that personalization consistently reduces MSE across participants, with lower medians in all boxplots. This demonstrates that, despite heterogeneity in edge weights, learning the network structure collectively provides a strong foundation, and subsequent local refitting produces participant-specific estimates that closely align with the ground truth.

Furthermore, while local NOTEARS already achieves

Method	Dimension d	Communication cost	SHD	TPR	FDR
Fed-Sparse-BNSL	20	1.99 ± 0.28	2.2 ± 1.81	0.93 ± 0.048	0.057 ± 0.078
Fed-BNSL	20	5.12 ± 0	2.9 ± 4.483	0.95 ± 0.033	0.086 ± 0.145
Fed-Sparse-BNSL	200	13.7 ± 0.8	27.4 ± 14.516	0.922 ± 0.03	0.084 ± 0.05
Fed-BNSL	200	512 ± 0	50.7 ± 12.859	0.934 ± 0.007	0.19 ± 0.045

Table 1: Communication costs (MB) and structural accuracy for $d = 20$ and $d = 200$.

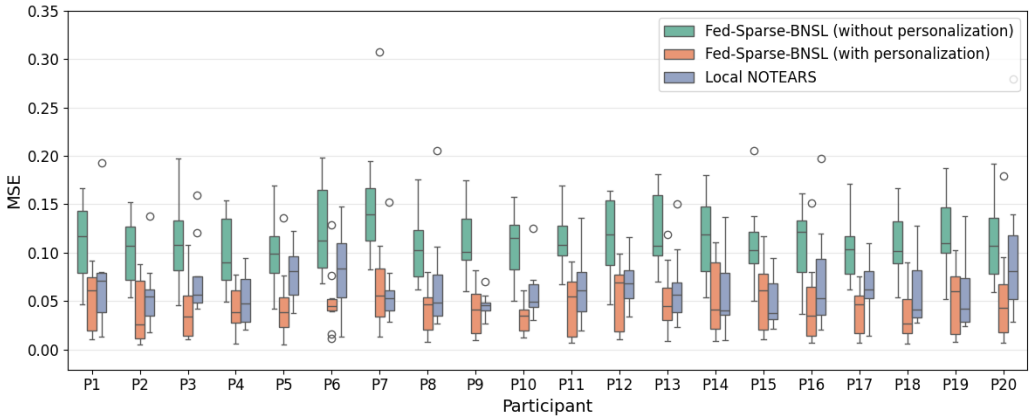


Figure 2: Participant-level personalization: per-participant normalized MSE between true and estimated parameters, for Fed-Sparse-BNSL with and without personalization, compared to local NOTEARS.

reasonable MSE across participants, indicating that client-level estimates are not degenerate, Fed-Sparse-BNSL with personalization consistently achieves the lowest median errors. This performance gap is deeply rooted in the quality of the underlying learned DAG: as detailed in Appendix C.2 (Figure 6), Fed-Sparse-BNSL attains significantly better structural metrics than local NOTEARS. This confirms that aggregating information across participants improves structural recovery beyond what is achievable locally.

6.3 Private Setting

Privacy-utility trade-off. We compare our non-private algorithm (Fed-Sparse-BNSL) to its differentially private variant (DP-Fed-Sparse-BNSL) on homogeneous synthetic data, varying the privacy budget $\varepsilon \in [0.5, 1, 2, 5, 10, 25, 50]$ while keeping $\delta = \frac{1}{n_p^2}$ fixed. The results are shown in Figure 3, where Fed-Sparse-BNSL is shown as a dashed reference to visualize the utility gap under privacy. In the high-privacy regime ($\varepsilon \leq 2$), DP-Fed-Sparse-BNSL exhibits higher SHD and lower TPR, while FDR is highest. As ε increases, SHD decreases and TPR rises monotonically, approaching the non-private baseline. For moderate to large budgets ($\varepsilon \geq 5$), SHD and TPR are close to Fed-Sparse-BNSL, variance shrinks, and FDR drops toward the non-private reference. Overall, increasing ε narrows the privacy-utility gap, with $\varepsilon = 5$ providing a strong compromise and $\varepsilon = 10$ nearly matching the non-private performance.

We evaluate the effect of dimensionality under differential privacy by comparing DP-Fed-Sparse-BNSL to a baseline, DP-Fed-BNSL, in which each participant privatizes its covariance matrix using the Gaussian mechanism (Wang, 2018) before running Fed-BNSL. Using homogeneous synthetic data, we vary the dimension $d \in [20, 50, 100, 200]$ while keeping the number of participants $P = 8$ and the per-participant sample size $n_p = 5000$ fixed. The privacy budget is set to $\varepsilon = 10$ for all dimensions.

As shown in Figure 4, SHD increases with d for both methods, reflecting the growth in the number of edges. At small dimensions ($d = 20, 50$), DP-Fed-BNSL performs similarly to DP-Fed-Sparse-BNSL. However, its structural accuracy deteriorates rapidly as d increases. At $d = 200$, DP-Fed-BNSL 's SHD is roughly twice that of DP-Fed-Sparse-BNSL (about 140 vs. 70), and its TPR drops to around 0.5 compared to over 0.8 for DP-Fed-Sparse-BNSL. This is expected, since privatizing full covariance matrices scales quadratically with dimension, whereas DP-Fed-Sparse-BNSL handles higher-dimensional settings more effectively due to its greedy approach.

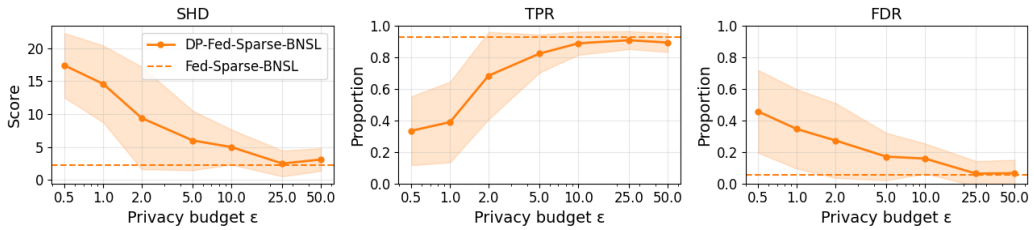


Figure 3: Privacy-utility trade-off: SHD, TPR and FDR of DP-Fed-Sparse-BNSL under varying privacy budgets ε, compared to non-private Fed-Sparse-BNSL.

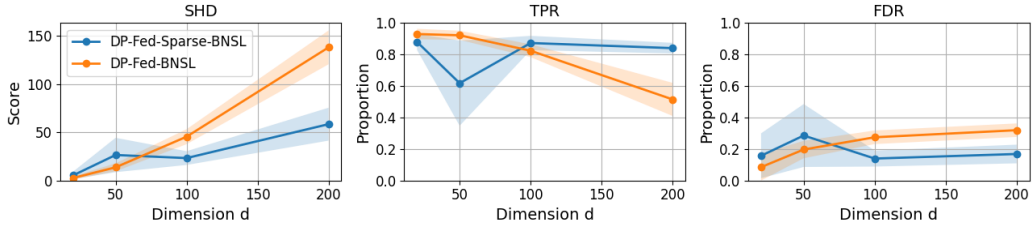


Figure 4: Dimensional robustness: performance of DP-Fed-Sparse-BNSL vs DP-Fed-BNSL as dimension d increases, under fixed privacy budget $\varepsilon = 10$.

6.4 Real data

We evaluate Fed-Sparse-BNSL, Fed-BNSL and their DP variants on the Sachs dataset (Sachs et al., 2005), a well-known benchmark for structure learning with a biologically validated ground-truth DAG. To simulate a federated setting, the data was partitioned across 3 participants. Experimental details and hyperparameter settings are provided in Appendix B.1.3.

Both Fed-Sparse-BNSL and Fed-BNSL recover a DAG with 18 edges, with SHD= 20 for Fed-Sparse-BNSL and SHD= 23 for Fed-BNSL, consistent with previously reported NOTEARS results (16 edges, SHD= 22). Direct comparison with Fed-BNSL from original paper is not possible since their experiments were conducted on a smaller dataset. As we have no guarantee that the ground-truth DAG is identifiable in real datasets, we also assess the Markov equivalence class, by looking at the skeleton (undirected edges) and v-structures. For a detailed introductions to these notions, we refer the reader to Koller and Friedman (2009). Neither method recovers ground-truth v-structures. However, Fed-Sparse-BNSL correctly identifies 12 out of 18 undirected edges, compared to 8 out of 18 undirected edges for Fed-BNSL.

For the DP variants, we fix the privacy budget to $\varepsilon = 5, \delta = \frac{1}{n_p^2}$. Both methods experienced a moderate performance drop: DP-Fed-Sparse-BNSL achieved SHD= 21 while still recovering 11 correct edges out of 18, whereas DP-Fed-BNSL obtained SHD= 25 with 8 correct edges out of 19 estimated.

7 CONCLUSION

We introduced Fed-Sparse-BNSL, a federated method for learning linear Gaussian Bayesian networks that addresses privacy and communication challenges. By combining sparse, greedy updates with differential privacy, Fed-Sparse-BNSL achieves accurate structure recovery with low communication overhead. Experiments on synthetic and real datasets demonstrate its effectiveness, scalability, and support for participant-level personalization.

Providing formal convergence guarantees for our proposed algorithms is an interesting direction for future work. While non-private PGCD enjoys linear convergence (Karimireddy et al., 2019), extending these guarantees to the differentially private setting introduces significant technical challenges. As noted by Mangold et al. (2023), deriving convergence guarantees for private greedy coordinate descent algorithms with proximal operators remains an open problem. Moreover, even in the centralized setting and assuming exact solutions to the subproblems, the convergence of the NOTEARS formulation itself is not fully established: the acyclicity constraint violates the regularity conditions typically required for global convergence guarantees in augmented Lagrangian methods (Ng et al., 2022).

In future work, we also plan to extend our approach to hybrid Bayesian networks to handle data with both continuous and discrete features.

Acknowledgements

We thank the anonymous reviewers for their valuable feedback and constructive comments.

This work was partially supported by L'Oréal, by grant ANR-20-CE23-0015 (Project PRIDE), and by the ANR 22-PECY-0002 IPOP (Interdisciplinary Project on Privacy) project of the Cybersecurity PEPR.

Some experiments presented in this paper were carried out using the Grid'5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations (see `https://www.grid5000.fr`).

References

Amin, K., Dick, T., Kulesza, A., Medina, A. M., and Vassilvitskii, S. (2019). Differentially private covariance estimation. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 14190–14199.

Bao, E., Xiao, X., Zhao, J., Zhang, D., and Ding, B. (2021). Synthetic data generation with differential privacy via bayesian networks. *Journal of Privacy and Confidentiality*, 11(3).

Bun, M. and Steinke, T. (2016). Concentrated differential privacy: Simplifications, extensions, and lower bounds. In Hirt, M. and Smith, A., editors, *Theory of Cryptography*, pages 635–658, Berlin, Heidelberg. Springer Berlin Heidelberg.

Creţu, A.-M., Guépin, F., and de Montjoye, Y.-A. (2024). Correlation inference attacks against machine learning models. *Science Advances*, 10(28):eadj9260.

Ding, Z., Kifer, D., E., S. M. S. N., Steinke, T., Wang, Y., Xiao, Y., and Zhang, D. (2021). The permute-and-flip mechanism is identical to report-noisy-max with exponential noise. *CoRR*, abs/2105.07260.

Dong, J., Durfee, D., and Rogers, R. (2020). Optimal differential privacy composition for exponential mechanisms. In III, H. D. and Singh, A., editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 2597–2606. PMLR.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. (2006). Calibrating noise to sensitivity in private data analysis. In *Proceedings of the Third Conference on Theory of Cryptography*, TCC'06, page 265–284, Berlin, Heidelberg. Springer-Verlag.

Dwork, C. and Roth, A. (2014). *The Algorithmic Foundations of Differential Privacy*. Now Publishers Inc.

Gao, E., Chen, J., Shen, L., Liu, T., Gong, M., and Bondell, H. D. (2023). Feddag: Federated DAG structure learning. *Trans. Mach. Learn. Res.*, 2023.

Geiping, J., Bauermeister, H., Dröge, H., and Moeller, M. (2020). Inverting gradients - how easy is it to break privacy in federated learning? In *NeurIPS*.

Hastie, T., Tibshirani, R., and Wainwright, M. (2015). *Statistical Learning with Sparsity: The Lasso and Generalizations*. Chapman & Hall/CRC.

Huth, M., Arruda, J., Gusinow, R., Contento, L., Tacconelli, E., and Hasenauer, J. (2023). Accessibility of covariance information creates vulnerability in federated learning frameworks. *Bioinformatics*, 39(9):btad531.

Jaggi, M. (2013). Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In Dasgupta, S. and McAllester, D., editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 427–435, Atlanta, Georgia, USA. PMLR.

Kalainathan, D. and Goudet, O. (2019). Causal discovery toolbox: Uncover causal relationships in python.

Karimireddy, S. P., Koloskova, A., Stich, S. U., and Jaggi, M. (2019). Efficient Greedy Coordinate Descent for Composite Problems. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2887–2896. PMLR.

Kitson, N. K., Constantinou, A. C., Guo, Z., Liu, Y., and Chobtham, K. (2023). A survey of bayesian network structure learning. *Artif. Intell. Rev.*, 56(8):8721–8814.

Koller, D. and Friedman, N. (2009). *Probabilistic Graphical Models: Principles and Techniques*. Adaptive computation and machine learning. MIT Press.

Loh, P.-L. and Bühlmann, P. (2014). High-dimensional learning of linear causal networks via inverse covariance estimation. *Journal of Machine Learning Research*, 15(88):3065–3105.

Mangold, P., Bellet, A., Salmon, J., and Tommasi, M. (2022). Differentially Private Coordinate Descent for Composite Empirical Risk Minimization. In *ICML*.

Mangold, P., Bellet, A., Salmon, J., and Tommasi, M. (2023). High-dimensional private empirical risk minimization by greedy coordinate descent. In Ruiz, F.,

Dy, J., and van de Meent, J.-W., editors, *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pages 4894–4916. PMLR.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and Arcas, B. A. y. (2017). Communication-Efficient Learning of Deep Networks from Decentralized Data. In Singh, A. and Zhu, J., editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 1273–1282. PMLR.

Mian, O., Kaltenpoth, D., Kamp, M., and Vreeken, J. (2023). Nothing but regrets — privacy-preserving federated causal discovery. In Ruiz, F., Dy, J., and van de Meent, J.-W., editors, *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pages 8263–8278. PMLR.

Nasr, M., Shokri, R., and Houmansadr, A. (2019). Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *IEEE Symposium on Security and Privacy*.

Ng, I., Fang, Z., Zhu, S., and Chen, Z. (2019). Masked gradient-based causal structure learning. *ArXiv*, abs/1910.08527.

Ng, I., Ghassami, A., and Zhang, K. (2020). On the role of sparsity and dag constraints for learning linear dags. *ArXiv*, abs/2006.10201.

Ng, I., Huang, B., and Zhang, K. (2024). Structure learning with continuous optimization: A sober look and beyond. In Locatello, F. and Didelez, V., editors, *Proceedings of the Third Conference on Causal Learning and Reasoning*, volume 236 of *Proceedings of Machine Learning Research*, pages 71–105. PMLR.

Ng, I., Lachapelle, S., Rosemary Ke, N., Lacoste-Julien, S., and Zhang, K. (2022). On the convergence of continuous constrained optimization for structure learning. In Camps-Valls, G., Ruiz, F. J. R., and Valera, I., editors, *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pages 8176–8198. PMLR.

Ng, I. and Zhang, K. (2022). Towards federated bayesian network structure learning with continuous optimization. In Camps-Valls, G., Ruiz, F. J. R., and Valera, I., editors, *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pages 8095–8111. PMLR.

Nutini, J., Schmidt, M., Laradji, I., Friedlander, M., and Koepke, H. (2015). Coordinate Descent Converges Faster with the Gauss-Southwell Rule Than Random Selection. In *International Conference on Machine Learning*, pages 1632–1641. PMLR.

Peters, J. and Bühlmann, P. (2013). Identifiability of gaussian structural equation models with equal error variances. *Biometrika*, 101(1):219–228.

Rogers, R. and Steinke, T. (2021). A better privacy analysis of the exponential mechanism. DifferentialPrivacy.org. `https://differentialprivacy.org/exponential-mechanism-bounded-range/`.

Sachs, K., Perez, O., Pe'er, D., Lauffenburger, D. A., and Nolan, G. P. (2005). Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721):523–529.

Torrijos, P., Gámez, J. A., and Puerta, J. M. (2024). Fedges: A federated learning approach for bayesian network structure learning. In Pedreschi, D., Monreale, A., Guidotti, R., Pellungrini, R., and Naretto, F., editors, *Discovery Science - 27th International Conference, DS 2024, Pisa, Italy, October 14-16, 2024, Proceedings, Part II*, volume 15244 of *Lecture Notes in Computer Science*, pages 83–98. Springer.

Tseng, P. and Yun, S. (2009). A coordinate gradient descent method for nonsmooth separable minimization. *Mathematical Programming*, 117(1):387–423.

van Daalen, F., Ippel, L., Dekker, A., and Bermejo, I. (2024). Vertibayes: learning bayesian network parameters from vertically partitioned data with missing values. *Complex & Intelligent Systems*, 10(4):5317–5329.

Vowels, M. J., Camgoz, N. C., and Bowden, R. (2021). D'ya like dags? a survey on structure learning and causal discovery. *ACM Computing Surveys*, 55:1 – 36.

Wang, L., Pang, Q., and Song, D. (2020). Towards practical differentially private causal graph discovery. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 5516–5526. Curran Associates, Inc.

Wang, Y. (2018). Revisiting differentially private linear regression: optimal and adaptive prediction & estimation in unbounded domain. In Globerson, A. and Silva, R., editors, *Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence, UAI 2018, Monterey, California, USA, August 6-10, 2018*, pages 93–103. AUAI Press.

Wang, Z., Ma, P., and Wang, S. (2023). Towards practical federated causal structure learning. In Koutra, D., Plant, C., Rodriguez, M. G., Baralis, E., and Bonchi, F., editors, *Machine Learning and Knowledge Discovery in Databases: Research Track - European Conference, ECML PKDD 2023, Turin, Italy, September 18-22, 2023, Proceedings, Part II*, volume 14170 of *Lecture Notes in Computer Science*, pages 351–367. Springer.

Zhang, J., Cormode, G., Procopiuc, C. M., Srivastava, D., and Xiao, X. (2017). Privbayes: Private data release via bayesian networks. *ACM Trans. Database Syst.*, 42(4).

Zheng, X., Aragam, B., Ravikumar, P., and Xing, E. P. (2018). Dags with no tears: continuous optimization for structure learning. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 9492–9503, Red Hook, NY, USA. Curran Associates Inc.

Zheng, X., Dan, C., Aragam, B., Ravikumar, P., and Xing, E. (2020). Learning sparse nonparametric dags. In Chiappa, S. and Calandra, R., editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 3414–3425. PMLR.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. **Yes**
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. **No**
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. **Yes**

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. **Yes**
   (b) Complete proofs of all theoretical results. **Yes**
   (c) Clear explanations of any assumptions. **Yes**

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). **Yes**
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). **Yes**
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). **Yes**
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). **Yes**.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. **Yes**, as detailed in Appendix B.4.
   (b) The license information of the assets, if applicable. **Yes**, as detailed in Appendix B.4.
   (c) New assets either in the supplemental material or as a URL, if applicable. **Not Applicable**
   (d) Information about consent from data providers/curators. **Not Applicable**
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. **Not Applicable**

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. **Not Applicable**
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. **Not Applicable**
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. **Not Applicable**

A Theoretical Derivations and Proofs

A.1 Reconstruction of Covariance Matrix

Starting from the closed-form solution of the local subproblem in participant p,

$$B = (\Sigma + \rho I_d)^{-1}(\rho W - \beta + \Sigma),$$

we can isolate Σ by multiplying both sides by $\Sigma + \rho I_d$ and rearranging terms:

$$(\Sigma + \rho I_d)B = \rho W - \beta + \Sigma \quad \Rightarrow \quad \Sigma(B - I_d) = \rho(W - B) - \beta.$$

Now, assume for the moment that $B - I_d$ is invertible. Then, the covariance matrix can be written as

$$\Sigma = (B - I_d)^{-1}\big(\rho(W - B) - \beta\big).$$

Regarding the invertibility of $B - I_d$, note that at the first iteration where $W = 0$ and $\beta = 0$, we have

$$B - I_d = (\Sigma + \rho I_d)^{-1}\Sigma - I_d = -\rho(\Sigma + \rho I_d)^{-1}.$$

This matrix is invertible because $\Sigma + \rho I_d$ is strictly positive definite: Σ is the empirical covariance matrix (positive semidefinite) and $\rho > 0$. Consequently, $B - I_d$ is invertible at the first iteration.[1]

The above derivation shows that the server can reconstruct the empirical covariance Σ from B, W, β and ρ at the first iteration of the algorithm.

A.2 Privacy Proofs

A.2.1 Preliminaries on Differential Privacy

We recall the main definitions and results used in our privacy analysis.

Definition A.1 (Bun and Steinke 2016)**.** *A randomized algorithm $\mathcal{A}$ satisfies ρ-zero-concentrated differential privacy (ρ-zCDP), if for any two datasets D_1 and D_2 of fixed size that differ in exactly one record and all $\alpha \in (1, \infty)$, we have :*

$$D_\alpha(\mathcal{A}(D_1) \| \mathcal{A}(D_2)) \leq \rho\alpha$$

where $D_\alpha(\mathcal{A}(D_1) \| \mathcal{A}(D_2))$ is the α-Rényi divergence between the distributions of $\mathcal{A}(D_1)$ and $\mathcal{A}(D_2)$.

Theorem A.1 (Bun and Steinke 2016)**.** *If $\mathcal{A}_1$ satisfies $\rho^{(1)}$-zCDP and $\mathcal{A}_2$ satisfies $\rho^{(2)}$-zCDP, then the composition of $\mathcal{A}_1$ and $\mathcal{A}_2$ satisfies $(\rho^{(1)} + \rho^{(2)})$-zCDP.*

Theorem A.2 (Bun and Steinke 2016)**.** *If $\mathcal{A}$ satisfies ρ-zCDP, then $\mathcal{A}$ satisfies $(\rho + 2\sqrt{\rho \log(1/\delta)}, \delta)$-DP for any $\delta > 0$.*

Theorem A.3 (Bun and Steinke 2016)**.** *Let $\mathcal{A}$ be the Gaussian mechanism applied to a query with ℓ_2-sensitivity Δ, using noise parameter σ. Then $\mathcal{A}$ satisfies ρ-zCDP with $\rho = \frac{\Delta^2}{2\sigma^2}$.*

Theorem A.4 (Rogers and Steinke 2021)**.** *Let $\mathcal{A}$ be the Gumbel max trick mechanism applied to a query with ℓ_1-sensitivity Δ, using noise parameter β. Then $\mathcal{A}$ satisfies ρ-zCDP with $\rho = \frac{\Delta^2}{2\beta^2}$.*

A.2.2 Proof of Theorem 5.1

Proof. The smooth part of the local objective is

$$\mathcal{L}(B; X) = L(B; X) + \operatorname{tr}\big[\beta^\top (B - W)\big] + \frac{\rho_2}{2}\|B - W\|_F^2,$$

where only $L(B; X)$ depends on the data:

$$L(B; X) = \frac{1}{2n}\|X - XB\|_F^2 = \frac{1}{n}\sum_{i=1}^{n} \ell(B; x_i), \qquad \ell(B; x_i) = \frac{1}{2}\|x_i - B^\top x_i\|_2^2.$$

The (i, j)-th partial derivative of ℓ is $\nabla_{i,j}\ell(B; x_i)$.

[1] In practice, we find that this property generally holds also in subsequent iterations.

Gradient sensitivity. For neighboring datasets X, X' differing in a single sample at index k, we have

$$\Delta = \sup_{X,X'} \|\nabla_{i,j}\mathcal{L}(B;X) - \nabla_{i,j}\mathcal{L}(B;X')\|_2 = \frac{1}{n}\left|\nabla_{i,j}\ell(B;x_k) - \nabla_{i,j}\ell(B;x'_k)\right| \le \frac{2L_{i,j}}{n},$$

where $L_{i,j}$ is the coordinate-wise Lipschitz constant of ℓ.

Sensitivity of the coordinate selection score. The non-private score is

$$S_{i,j} = \sqrt{M_{i,j}}\left|\operatorname{prox}_{\frac{\lambda}{M_{i,j}}|\cdot|}\left(B^k_{i,j} - \frac{1}{M_{i,j}}\nabla_{i,j}\mathcal{L}(B^k;X)\right) - B^k_{i,j}\right|.$$

By the non-expansiveness of the proximal operator, the sensitivity satisfies:

$$\begin{aligned}
\Delta(S_{i,j}) &= \left|\sqrt{M_{i,j}}\left|\operatorname{prox}_{\frac{\lambda}{M_{i,j}}|\cdot|}\left(B_{i,j} - \frac{1}{M_{i,j}}\nabla_{i,j}\mathcal{L}(B;X)\right) - B_{i,j}\right|\right. \\
&\quad \left. - \sqrt{M_{i,j}}\left|\operatorname{prox}_{\frac{\lambda}{M_{i,j}}|\cdot|}\left(B_{i,j} - \frac{1}{M_{i,j}}\nabla_{i,j}\mathcal{L}(B;X')\right) - B_{i,j}\right|\right| \\
&\le \sqrt{M_{i,j}}\left|\left(B_{i,j} - \frac{1}{M_{i,j}}\nabla_{i,j}\mathcal{L}(B;X)\right) - \left(B_{i,j} - \frac{1}{M_{i,j}}\nabla_{i,j}\mathcal{L}(B;X')\right)\right| \\
&= \frac{1}{\sqrt{M_{i,j}}}\left|\nabla_{i,j}\mathcal{L}(B;X) - \nabla_{i,j}\mathcal{L}(B;X')\right| \le \frac{2L_{i,j}}{n\sqrt{M_{i,j}}} \\
&= \frac{\Delta}{\sqrt{M_{i,j}}}.
\end{aligned} \tag{13}$$

Base mechanisms. We now analyze the privacy guarantees of the two base mechanisms we use:

1. **Coordinate selection via Gumbel-max (Exponential mechanism).** The non-private score $S_{i,j}$ from (13) has sensitivity $\Delta(S_{i,j})$. From Theorem A.4, adding Gumbel noise $\text{Gumbel}(0, \beta_{i,j})$ to this score ensures ρ_{EM}-zCDP with
$$\rho_{\text{EM}} = \frac{\Delta(S_{i,j})^2}{2\beta^2_{i,j}} = \frac{1}{M_{i,j}}\frac{\Delta^2}{2\beta^2_{i,j}}$$
2. **Gradient update via Gaussian mechanism.** Each gradient coordinate $\nabla_{i,j}\mathcal{L}(B^k;X)$ has ℓ_2-sensitivity Δ. Adding Gaussian noise $\mathcal{N}(0,\sigma^2_{i,j})$ to this coordinate ensures ρ_{Gauss}-zCDP (Theorem A.3) with
$$\rho_{\text{Gauss}} = \frac{\Delta^2}{2\sigma^2_{i,j}}.$$
3. **Matching privacy costs.** To balance the privacy budget between the two mechanisms, we set
$$\beta_{i,j} = \frac{\sigma_{i,j}}{\sqrt{M_{i,j}}}.$$
With this choice, both mechanisms contribute equally to the total zCDP cost:
$$\rho = \frac{\Delta^2}{2\sigma^2_{i,j}}.$$

Composition and DP conversion. Since there are $2KT$ queries in total, by the composition theorem (Theorem A.1),

$$\rho_{\text{total}} = 2KT\rho = \frac{KT\Delta^2}{\sigma^2_{i,j}}.$$

Applying the zCDP to (ε,δ)-DP conversion (Theorem A.2) yields

$$\varepsilon = \rho_{\text{total}} + 2\sqrt{\rho_{\text{total}}\log(1/\delta)}.$$

Therefore, to satisfy (ε,δ)-DP, we need to set the noise scales to

$$\beta_{i,j} = \frac{\sigma_{i,j}}{\sqrt{M_{i,j}}} = \frac{\Delta\sqrt{KT}(\sqrt{\log(\frac{1}{\delta}) + \varepsilon} + \sqrt{\log(\frac{1}{\delta})})}{\varepsilon}.$$

□

A.3 Smoothness Constants

In our method, we assume the coordinate-wise smoothness constants $M_{i,j}$ known or upper bounded. When it is not the case, these constants can be estimated privately.

A.3.1 Estimation of the Constants

Coordinate-wise smoothness. A differentiable function $f : \mathbb{R}^{d\times d} \to \mathbb{R}$ is called *coordinate-wise smooth* if there exists $M \in \mathbb{R}^{d\times d}$ such that for all $X \in \mathbb{R}^{d\times d}$ and for all $1 \le i, j \le d$,

$$|\nabla_{i,j} f(X + uE_{i,j}) - \nabla_{i,j} f(X)| \le M_{i,j}|u|, \quad \forall u \in \mathbb{R},$$

where $E_{i,j}$ is the matrix with 1 in entry (i, j) and 0 elsewhere.

Consider the local objective:

$$\mathcal{L}(B; X) = L(B; X) + \operatorname{tr}\big[\beta^\top (B - W)\big] + \frac{\rho_2}{2}\|B - W\|_F^2,$$

where $X \in \mathbb{R}^{n\times d}$ is the local data matrix. The Hessian with respect to B is

$$\nabla^2 \mathcal{L}(B; X) = \frac{1}{n} X^\top X \otimes I_d + \rho_2 I_{d^2}.$$

Hence, the coordinate-wise smoothness constant for coordinate (i, j) is:

$$\begin{aligned} M_{i,j}(X) = M_j(X) &= \frac{1}{n}\|X_{:,j}\|_2^2 + \rho_2 \qquad \forall i \in 1, \dots, d \\ &= \frac{1}{n}\sum_{k=1}^{n} X_{k,j}^2 + \rho_2 \end{aligned} \tag{14}$$

As can be seen from the formula above, $M_{i,j}(X)$ does not depend on i, hence the notation $M_j(X)$.

A.3.2 Differentially Private Estimation

We now explain how to privately estimate the coordinate-wise smoothness constants. Following Mangold et al. (2023), we assume we know an upper bound b_j on the squared value of the j-th feature, i.e., $X_j^2 \le b_j$.

Theorem A.5 (Privacy of smoothness constants)**.** *Let $\varepsilon_M, \delta_M > 0$ and $\Delta_{M,j} = b_j/n_p$ where b_j is such that $X_j^2 \le b_j$ and n_p is the sample size of participant p. If the Gaussian noise parameter is chosen as*

$$\sigma_j = \frac{\Delta_{M,j}\sqrt{d}(\sqrt{\log(1/\delta_M) + \varepsilon_M} + \sqrt{\log(1/\delta_M)})}{\sqrt{2}\varepsilon_M}, \quad \forall j \in \{1, \dots, d\},$$

then participant p can locally estimate all $M_{i,j}$'s with $(\varepsilon_M, \delta_M)$-DP in a preprocessing step using d calls to the Gaussian mechanism with noise scales $\sigma_1, \dots, \sigma_d$.

Proof. As shown in (14), we only need to estimate $M_1(X), \dots, M_d(X)$ to obtain all coordinate-wise smoothness constants. Let X and X' two neighboring datasets that differ in exactly one datapoint. The sensitivity of M_j is

$$\Delta_{M,j} = \sup_{X, X'} \|M_j(X) - M_j(X')\|_2 = \frac{1}{n_p}|X_{k,j}^2 - X_{k,j}'^2| \le \frac{b_j}{n_p}.$$

To privatize M_j, we clip and add Gaussian noise:

$$\tilde{M}_j(X) = \frac{1}{n_p}\sum_{k=1}^{n_p} \min(X_{k,j}^2, b_j) + \rho_2 + \mathcal{N}(0, \sigma_j^2).$$

By Theorem A.3, releasing $\tilde{M}_j(X)$ satisfies $\rho^{(j)}$-zCDP with

$$\rho^{(j)} = \frac{\Delta_{M,j}^2}{2\sigma_j^2}.$$

Since there are d constants to estimate privately, by the composition theorem (Theorem A.1),

$$\rho_{\text{total}} = d\frac{\Delta_{M,j}^2}{2\sigma_j^2}.$$

Applying the zCDP to $(\varepsilon_M, \delta_M)$-DP conversion (Theorem A.2) yields

$$\varepsilon_M = \rho_{\text{total}} + 2\sqrt{\rho_{\text{total}} \log(1/\delta)}.$$

Therefore, to satisfy $(\varepsilon_M, \delta_M)$-DP, we need to set the noise scale to

$$\sigma_j = \frac{\Delta_M \sqrt{d}(\sqrt{\log(1/\delta_M) + \varepsilon_M} + \sqrt{\log(1/\delta_M)})}{\sqrt{2}\varepsilon_M}.$$ □

Remark. The above theorem shows how each participant p can estimate the coordinate-wise smoothness constants based on its local data. These participant-level constants can then be shared and aggregated to obtain more accurate global estimates.

A.4 Theoretical Comparison of DP-PGCD Privacy Bounds

We provide a detailed mathematical comparison of the privacy guarantees between our improved DP-PGCD algorithm (Algorithm 2) and the original approach introduced by Mangold et al. (2023), which relied on Laplace noise. The following analysis highlights why our method requires less noise to achieve the same privacy budget.

The privacy loss for our approach is given by:

$$\varepsilon_{\text{new}} = \frac{4L_{ij}\sqrt{KT\log(1/\delta)}}{n\sigma_{ij}} + \frac{4L_{ij}^2 KT}{n^2\sigma_{ij}^2},$$

while the original approach of Mangold et al. (2023), it is:

$$\varepsilon_{\text{old}} = \frac{4L_{ij}\sqrt{KT\log(1/\delta)}}{n\lambda_{ij}} + \frac{4L_{ij}KT}{n\lambda_{ij}}\Big(e^{\frac{2L_{ij}}{n\lambda_{ij}}} - 1\Big),$$

where λ_{ij} is the scale parameter of a Laplace distribution.

To compare these expressions, we match the noise variance: for a Gaussian distribution $\mathcal{N}(0, \sigma^2)$, the variance is σ^2, while for a Laplace distribution $\text{Lap}(0, \lambda)$, it is $2\lambda^2$. Setting $\sigma^2 = 2\lambda^2$ (i.e., $\lambda = \frac{\sigma}{\sqrt{2}}$) yields:

$$\varepsilon_{\text{new}} = \frac{4L_{ij}\sqrt{KT\log(1/\delta)}}{n\sigma_{ij}} + \frac{4L_{ij}^2 KT}{n^2\sigma_{ij}^2},$$

$$\varepsilon_{\text{old}} = \frac{4\sqrt{2}L_{ij}\sqrt{KT\log(1/\delta)}}{n\sigma_{ij}} + \frac{4\sqrt{2}L_{ij}KT}{n\sigma_{ij}}\Big(e^{\frac{2\sqrt{2}L_{ij}}{n\sigma_{ij}}} - 1\Big).$$

We analyze the ratio $\frac{\varepsilon_{\text{old}}}{\varepsilon_{\text{new}}}$:

$$\frac{\varepsilon_{\text{old}}}{\varepsilon_{\text{new}}} = \sqrt{2}\frac{1 + \sqrt{\frac{KT}{\log(1/\delta)}}\Big(e^{2\sqrt{2}\frac{L_{ij}}{n\sigma_{ij}}} - 1\Big)}{1 + \sqrt{\frac{KT}{\log(1/\delta)}}\frac{L_{ij}}{n\sigma_{ij}}}.$$

Using the inequality $e^x \geq 1 + x$ for all $x \in \mathbb{R}$, we have:

$$e^{2\sqrt{2}\frac{L_{ij}}{n\sigma_{ij}}} - 1 \geq 2\sqrt{2}\frac{L_{ij}}{n\sigma_{ij}} > \frac{L_{ij}}{n\sigma_{ij}}.$$

By multiplying both sides by $\sqrt{\frac{KT}{\log(1/\delta)}}$ and adding 1, we have:

$$1+\sqrt{\frac{KT}{\log(1/\delta)}}(e^{2\sqrt{2}\frac{L_{ij}}{n\sigma_{ij}}}-1) > 1+\sqrt{\frac{KT}{\log(1/\delta)}}\frac{L_{ij}}{n\sigma_{ij}}.$$

Equivalently,

$$\frac{1+\sqrt{\frac{KT}{\log(1/\delta)}}(e^{2\sqrt{2}\frac{L_{ij}}{n\sigma_{ij}}}-1)}{1+\sqrt{\frac{KT}{\log(1/\delta)}}\frac{L_{ij}}{n\sigma_{ij}}} > 1.$$

Finally, by multiplying both sides by $\sqrt{2}$, we obtain:

$$\frac{\varepsilon_{\text{old}}}{\varepsilon_{\text{new}}} \geq \sqrt{2}.$$

Thus, our DP-PGCD (Algorithm 2) provides at least a $\sqrt{2}$ improvement in privacy loss compared to Mangold et al. (2023), for the same noise variance.

B Experimental Setup and Implementation

B.1 Datasets

B.1.1 Synthetic Data: Homogeneous Setting

To simulate homogeneous data, we reproduce and generalization the data generation procedure used in (Zheng et al., 2018; Ng and Zhang, 2022). We generate a dataset for P participants, each having n_p samples, across d variables. This setup assumes a shared underlying structure and identical edge weights across all participants.

- **DAG structure generation:** A $d \times d$ binary adjacency matrix is initially created. This is achieved by generating an Erdös-Rényi random graph with d nodes and an expected number of d edges. The generation process ensures the resulting graph is acyclic.
- **Edge weight assignment:** Once the DAG structure (represented by the binary adjacency matrix) is defined, non-zero weights are assigned to its edges. These weights are uniformly sampled from two disjoint intervals: $[-2, -0.5] \cup [0.5, 2]$. This results in a weighted adjacency matrix.
- **Data generation:** Observations are then simulated from the linear Gaussian Structural Equation Model (SEM) defined in Section 2. In this model, the value of each variable is determined by a linear combination of its parents' value (as defined by the weighted adjacency matrix), plus an independent Gaussian noise term. This additive noise term is sampled, for each variable, from a standard normal distribution $\mathcal{N}(0, 1)$. This implies a uniform noise variance of 1 across all variables, a condition that ensures the identifiability of the DAG structure (Peters and Bühlmann, 2013; Loh and Bühlmann, 2014).

B.1.2 Synthetic Data: Heterogeneous Setting

For heterogeneous data generation, we consider $P = 5$ participants, each having $n_p = 5000$ samples, and we assume that all P participants share the same underlying DAG structure but differ in their edge weights. Instead of being identical for all participants, we use a hierarchical design: each participant's weight for an edge present in the DAG is independently sampled from a Gaussian distribution centered at the corresponding global weight, with variance $\sigma^2 = 0.1$. Global weights are uniformly drawn from the disjoint intervals $[-2, -0.5] \cup [0.5, 2]$ as in the homogeneous setting. Observations for each participant are then generated from a linear Gaussian SEM using their individual edge weights.

B.1.3 Real Data

The Sachs dataset (Sachs et al., 2005), composed of $d = 11$ variables representing protein signaling molecules and $n = 7466$ samples, features a biologically validated ground-truth DAG with 18 edges. To simulate a federated setting, the dataset was partitioned across $P = 3$ participants, resulting in $n_p = 2488$ samples per participant.

B.2 Evaluation Metrics

B.2.1 Communication cost

The total communication cost is defined as the cumulative volume of data exchanged between the server and all participants over the entire training procedure, reported in megabytes (MB). Each floating-point value is encoded using 8 bytes.

When sparsity is used, the transmission of each non-zero coefficient requires sending its matrix index, encoded using $\lceil \log_2(d^2)/8 \rceil$ bytes. Hence, the total size of one transmitted coefficient-index pair is $b_{\text{entry}} = 8 + \lceil \log_2(d^2)/8 \rceil$ bytes.

In Fed-BNSL, a dense $d \times d$ matrix is transmitted from each of the P participants to the server and back at every iteration, resulting in a total communication cost accumulated over T iterations of

$$\text{Cost} = 2 \times T \times P \times d^2 \times 8\text{bytes}.$$

Since all entries are transmitted, no index encoding is required in this case.

In contrast, Fed-Sparse-BNSL communicates only non-zero coefficients and their indices are transmitted from each of the P participants to the server and back at every iteration, resulting in a total communication cost accumulated over T iterations of

$$\text{Cost} = \sum_{t=1}^{T} \Big(\underbrace{\sum_{p=1}^{P} e_{\text{local}}^{(p,t)} \times b_{\text{entry}}}_{\text{participants}} + \underbrace{P \times e_{\text{global}}^{(t)} \times b_{\text{entry}}}_{\text{server}} \Big),$$

where $e_{\text{local}}^{(p,t)}$ denotes the number of coefficients sent by participant p at iteration t, and $e_{\text{global}}^{(t)}$ denotes the number of global coefficients broadcast by the server.

B.2.2 Personalization

In the heterogeneous setting, each participant p has its own regression coefficients $B_{\text{true}}^{(p)}$ within a shared DAG structure. After learning the consensus structure, Fed-Sparse-BNSL returns a global weighted adjacency matrix $\widehat{B}_{\text{global}}$. Then, each participant locally re-estimates its coefficients $\widehat{B}^{(p)}$ by ordinary least squares regression on its own data, keeping the DAG fixed.

The quality of estimated parameters is measured by the normalized mean squared error (MSE):

$$\text{MSE}_p(\widehat{B}) = \frac{\|\widehat{B} - B_{\text{true}}^{(p)}\|_F^2}{\|B_{\text{true}}^{(p)}\|_F^2}.$$

In Figure 2, $\text{MSE}_p(\widehat{B}^{(p)})$ is Fed-Sparse-BNSL with personalization and $\text{MSE}_p(\widehat{B}_{\text{global}})$ is Fed-Sparse-BNSL without personalization.

B.3 Hyperparameters

The selection of appropriate hyperparameters is critical for the performance and fair comparison of algorithms. This appendix details the hyperparameter tuning methodology employed for Fed-Sparse-BNSL, DP-Fed-Sparse-BNSL, Fed-BNSL and DP-Fed-BNSL across all experimental settings: homogeneous and heterogeneous synthetic data, real data, across both non-private and private settings. Our general goal was to ensure that each method operated as its optimal performance for each specific scenario.

B.3.1 Synthetic Data: General Tuning Protocol

For all settings, the metric used to tune the hyperparameters is the Structural Hamming Distance (SHD).

Tuning on held-out problem instances. For experiments with dimension $d = 20$, hyperparameters for each algorithm were tuned on an independent dataset generated with seed= 1. For higher-dimensional experiments

Table 2: Grid search for hyperparameters tuning in the non-private homogeneous setting.

Hyperparameter	Grid search
ρ_1	$\{10, 50, 100, 1000, 10000\}$
ρ_2	$\{1, 10, 100, 1000\}$
γ	$\{0.1, 0.5, 1\}$
λ	$\{0.001, 0.01, 0.1, 0.5, 1\}$

Table 3: Grid search for hyperparameters tuning in the non-private heterogeneous setting.

Hyperparameter	Grid search
ρ_1	$\{100, 1000, 10000\}$
ρ_2	$\{1, 10, 100\}$
γ	$\{0.5, 1\}$
λ	$\{0.01, 0.1, 1\}$

($d \in \{50, 100, 200\}$), hyperparameters were tuned based on the average performance across two independent datasets, generated with seed= 100 and seed= 400. This approach helps to ensure robustness of the chosen hyperparameters against specific dataset realizations in high dimensions.

Final evaluation. Once the optimal hyperparameters were determined for a given scenario using the procedure above, the final results (mean and standard deviation) reported in Section 6 were obtained by running each algorithm on 10 new datasets generated with 10 independent seeds: $\{2, \ldots, 11\}$.

B.3.2 Synthetic Data: Non-Private Setting

In the non-private setting, we tuned the ADMM penalty parameters (ρ_1, ρ_2), the step size for gradient descent in Fed-Sparse-BNSL (γ), and the regularization parameter (λ). As done in prior work (Zheng et al., 2018; Ng and Zhang, 2022), we also use a threshold set at 0.3 for edge pruning in a post-processing step.

Homogeneous data. For homogeneous synthetic data, the grid search ranges for each hyperparameter are detailed in Table 2.

The best configurations of hyperparameters, selected based on the lowest SHD on the tuning dataset (seed= 1) are:

- For Fed-Sparse-BNSL: $\rho_1 = 1000$, $\rho_2 = 1$ $\lambda = 0.1$ and $\gamma = 0.5$
- For Fed-BNSL: $\rho_1 = 1000$, $\rho_2 = 1$ $\lambda = 0.01$

Heterogeneous data. For heterogeneous synthetic data, the grid search ranges were slightly adjusted. The grid search ranges are detailed in Table 3.

The best configurations of hyperparameters, selected based on the lowest SHD on the tuning dataset (seed= 1) are:

- For Fed-Sparse-BNSL: $\rho_1 = 1000$, $\rho_2 = 1$ $\lambda = 0.1$ and $\gamma = 0.5$
- For Fed-BNSL: $\rho_1 = 100$, $\rho_2 = 1$ $\lambda = 0.1$

B.3.3 Synthetic Data: Private Setting

In the private setting, we focused on tuning parameters specific to differentially private mechanisms.

Privacy-utility study (Figure 3). For DP-Fed-Sparse-BNSL, we fixed the non-private hyperparameters $(\rho_1, \rho_2, \gamma, \lambda)$ to their best-performing values identified in the non-private setting. This allowed us to isolate the

impact of privacy-specific parameters. We then tuned the following privacy-specific hyperparameters: C is the clipping threshold for gradients in DP-Fed-Sparse-BNSL, T the number of ADMM iterations and K the number of local PGCD iterations performed by each participant per ADMM iteration.

The grid search ranges are given in Table 4.

Table 4: Grid search ranges for privacy-specific hyperparameters tuning in the private homogeneous setting for the privacy-utility study (Figure 3).

Hyperparameter	**Grid search**
C	$\{3, 5, 7, 10, 20, 30\}$
T	$\{10, 20, 50, 100\}$
K	$\{10, 20, 30, 40, 50, 100\}$

The optimal combination of these hyperparameters for each privacy budget $\varepsilon \in \{0.5, 1, 2, 5, 10, 25, 50\}$ are presented Table 5.

Table 5: Optimal hyperparameters for DP-Fed-Sparse-BNSL in the privacy-utility study (Figure 3), for each privacy budget ε.

ε	C	T	K
0.5	10	10	10
1	10	10	10
2	5	100	10
5	5	100	20
25	7	100	30
50	5	100	50

Dimensionality robustness study (Figure 4). This study evaluated the performance of DP-Fed-Sparse-BNSL compared to the baseline DP-Fed-BNSL as the data dimension d increased, under a fixed privacy budget of $\varepsilon = 10$. For this analysis, a broader set of hyperparameters, including λ and γ, were re-tuned for each dimension. For DP-Fed-BNSL, b is the sensitivity bound used for the Gaussian mechanism applied to privatize the covariance matrices, as described by Wang (2018). The grid search ranges for this study are given in Table 6.

Table 6: Grid search ranges for hyperparameters tuning in the private dimensionality robustness study (Figure 4).

Hyperparameter	**Grid search**
C	$\{5, 10, 20, 30\}$
b	$\{5, 10, 20, 30\}$
T	$\{50, 100\}$
K	$\{20, 50, 100\}$
λ	$\{0.01, 0.1, 1\}$
γ	$\{0.5, 1\}$

The optimal hyperparameters for each method and dimension in the dimensionality robustness study are presented in Table 7.

B.3.4 Real Data: Non-Private Setting

In contrast to synthetic data, where we can select hyperparameters on held-out problem instances, it is not clear how to implement a similar validation procedure on the Sachs dataset. Consistent with prior work (Zheng et al.,

Table 7: Optimal hyperparameters for each method and dimension, with privacy budget $\varepsilon = 10$, in the private dimensionality robustness study (Figure 4).

Dimension	DP-Fed-Sparse-BNSL					DP-Fed-BNSL		
	C	T	K	λ	γ	b	T	λ
20	10	100	30	0.1	0.5	7	300	0.01
50	5	50	50	0.1	1	10	300	0.1
100	30	100	50	0.1	1	20	300	0.01
200	30	100	100	0.1	1	10	300	0.01

2018; Ng and Zhang, 2022), we select the best hyperparameters for each method based on the final structure learning metrics.

We recall that the hyperparameters in the non-private settings are the ADMM penalty parameters (ρ_1, ρ_2), the ℓ_1 regularization parameter (λ), the step size for gradient descent in Fed-Sparse-BNSL (γ). For this dataset, we also tuned the threshold used for edge pruning in a post-processing step.

For the non-private evaluation on the Sachs dataset, the goal of hyperparameter tuning was to identify configurations that yielded the best structural accuracy. Given the known ground-truth DAG has 18 edges, we specifically looked for configurations that estimated approximately 18 edges while maximizing the number of correctly identified undirected edges.

The tuning process involved a grid search over the hyperparameters for the two methods. The grid search for Fed-Sparse-BNSL and Fed-BNSL included:

Hyperparameter	**Grid searchs**
ρ_1	$\{100, 1000, 10000, 100000\}$
ρ_2	$\{1, 5, 10, 100\}$
γ	$\{0.5, 1\}$
λ	$\{0.01, 0.1, 1\}$
threshold	$\{0, 0.1, 0.3\}$

Note that, wile λ, ρ_1, ρ_2 and threshold are shared across both methods, γ (step size for local updates) is a hyperparameter unique to Fed-Sparse-BNSL.

Tuning metrics. During tuning, we looked at several metrics, including the number of estimated edges, the number of correctly estimated undirected edges, the number of estimated v-structures, the number of correctly estimated v-structures and SHD.

Consistent with prior work, we observed that recovering ground-truth v-structures was particularly challenging for both methods on this dataset (0 correct v-structures found). Therefore, we prioritized configurations that resulted in an estimated number of edges close to the true 18 edges of the Sachs DAG, while maximizing correctly identified undirected edges.

The best configurations of hyperparameters are:

- For Fed-Sparse-BNSL: $\rho_1 = 10000$, $\rho_2 = 5$ $\lambda = 1, \gamma = 0.1$ and threshold$= 0.1$.
- For Fed-BNSL: $\rho_1 = 100000$, $\rho_2 = 10$ $\lambda = 0.1$ and threshold$= 0.1$.

B.3.5 Real Data: Private Setting

We recall that the privacy-specific hyperparameters are C the clipping threshold for gradients in DP-Fed-Sparse-BNSL, b the sensitivity bound used for the Gaussian mechanism applied to privatize the covariance matrices in DP-Fed-BNSL, T the number of ADMM iterations and K the number of local PGCD iterations performed by each participant per ADMM iteration in DP-Fed-Sparse-BNSL.

For the differentially private evaluation on the Sachs dataset, a fixed privacy budget of $\varepsilon = 5$ (with $\delta = 1/n_p^2$) was used for both DP-Fed-Sparse-BNSL and DP-Fed-BNSL. To reduce the tuning complexity and build upon the non-private baselines, the non-private hyperparameters $(\rho_1, \rho_2, \lambda, \gamma)$ were fixed to their optimal values found in the non-private setting. The tuning then focused on privacy-specific parameters.

Tuned privacy-specific hyperparameters. The grids used for privacy-specific parameters for each algorithm are given in Tables 8 and 9.

Table 8: Grid search ranges for hyperparameters tuning in the private setting for DP-Fed-Sparse-BNSL

Hyperparameter	**Grid searchs**
C	$\{1, 10, 100, 10000, 15000, 20000\}$
T	$\{30, 50\}$
K	$\{10, 20, 30, 50\}$
threshold	$\{0, 0.1, 0.3\}$

Table 9: Grid search ranges for hyperparameters tuning in the private setting for DP-Fed-BNSL

Hyperparameter	**Grid searchs**
b	$\{1, 2, 3, 4, 5, 100, 10000\}$
T	$\{100, 300\}$
threshold	$\{0, 0.1, 0.3\}$

The goal was to find the combination of privacy-specific hyperparameters that maintained the best possible utility (low SHD, high correct edges) while operating under the fixed privacy budget $\varepsilon = 5$. We aimed for configurations whose performance was close to the non-private baseline.

The selected optimal private configurations are:

- For DP-Fed-Sparse-BNSL: $C = 15000$, $T = 30$ $K = 30$, $\lambda = 1$, $\gamma = 0.1$ and threshold$= 0$.
- For DP-Fed-BNSL: $b = 4, T = 100, \lambda = 0.1$ and threshold$= 0.1$.

B.4 Implementation and Computing Resources

All experiments were conducted on CPUs, either on a personal computer or using the Grid5000 cluster (`https://www.grid5000.fr/w/Grid5000:Home`).

The implementation relies on several open-source components. Parts of the code were adapted from the open-source implementation of Ng and Zhang (2022) (available at `https://github.com/ignaviernng/notears-admm/tree/master/notears_admm`), which itself is based on the original NOTEARS implementation (Zheng et al., 2018) (available at `https://github.com/xunzheng/notears/tree/master/notears`). Additionally, the code for the privatization of the covariance matrix for DP-Fed-BNSL was adapted from the repository at `https://github.com/BorjaBalle/analytic-gaussian-mechanism`.

The Sachs dataset (Sachs et al., 2005) was obtained through the Causal Discovery Toolbox (CDT) Python package (Kalainathan and Goudet, 2019).

Regarding licenses, the reused code for NOTEARS-ADMM (Ng and Zhang, 2022) and NOTEARS (Zheng et al., 2018) and the analytic-gaussian-mechanism are all released under the Apache License. The Causal Discovery Toolbox (CTD) is released under the MIT License.

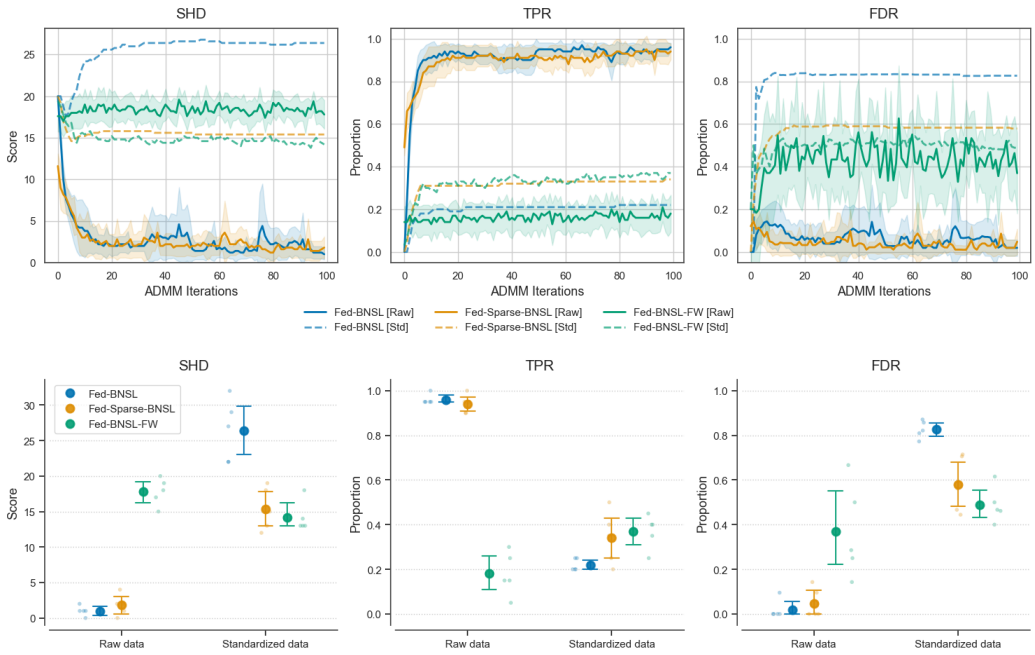


Figure 5: Convergence of Fed-BNSL, Fed-BNSL-FW and Fed-Sparse-BNSL on raw and standardized data. Top: SHD, TPR and FDR across iterations. Solid lines represent raw data, while dashed lines represent standardized data.; Bottom: Final SHD, TPR and FDR (iteration 100).

C Additional Empirical Evaluations

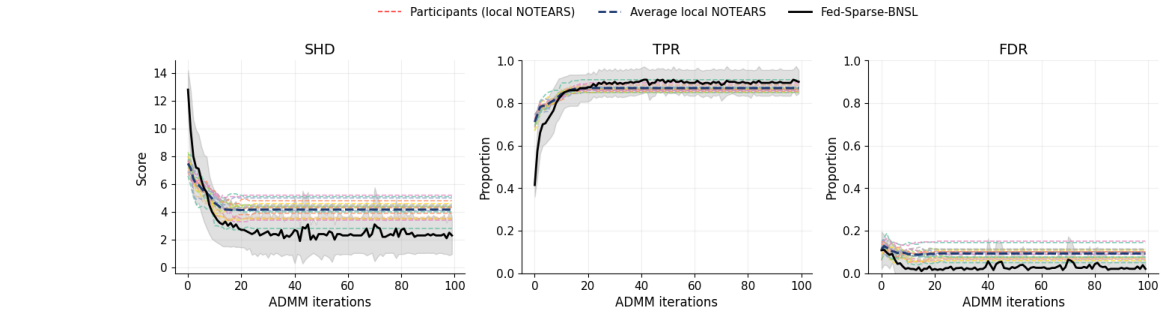
C.1 Impact of Data Standardization on Identifiability

In Section 4.2, we theoretically justified the choice of Proximal Greedy Coordinate Descent (PGCD) over classical LASSO solvers (such as Frank-Wolfe). Classical solvers typically require standardized data to apply the ℓ_1 penalty uniformly. However, in the setting of linear Gaussian BNs, standardizing the data violates the equal error variance assumption, thereby destroying the identifiability of the true DAG (Ng et al., 2024; Loh and Bühlmann, 2014). To empirically validate this crucial design choice, we evaluate the performance of:

- Fed-BNSL: the baseline (Ng and Zhang, 2022) that solves the unpenalized local subproblem using the exact closed-form solution (Eq. 10);
- Fed-Sparse-BNSL: solves ℓ_1-penalized subproblem using PGCD;
- Fed-BNSL-FW: a variant of our Fed-Sparse-BNSL, which solves the ℓ_1-penalized subproblem using Frank-Wolfe (Jaggi, 2013), a classical LASSO solver.

We run each algorithm on both raw data, and standardized data across 5 runs. The results are presented in Figure 5, showing the optimization trajectories (top) and the final structural utility (bottom). The results confirm that standardization destroys the identifiability of the DAG: when the data is standardized, all three algorithms, including the unpenalized baseline, fail to converge. On raw data, identifiability is preserved, and the closed-form baseline (Fed-BNSL) recovers the true graph. However, for Fed-BNSL-FW which solves the ℓ_1-penalized problem using Frank-Wolfe, the lack of standardization prevents the solver from applying the ℓ_1 penalty uniformly, which leads to a highly inaccurate DAG. In contrast, our proposed approach, Fed-Sparse-BNSL, which employs PGCD as the solver, effectively resolves this dilemma. The coordinate-wise smoothness constants M_j inherently normalize the gradient steps, enabling PGCD to handle the ℓ_1-penalized problem on unstandardized data. As a result, Fed-Sparse-BNSL achieves the high structural utility of the unpenalized baseline while simultaneously enforcing the sparsity necessary for communication efficiency.

Figure 6: Convergence of Fed-Sparse-BNSL compared to local NOTEARS on a heterogeneous dataset ($P = 20, n_p = 50$).



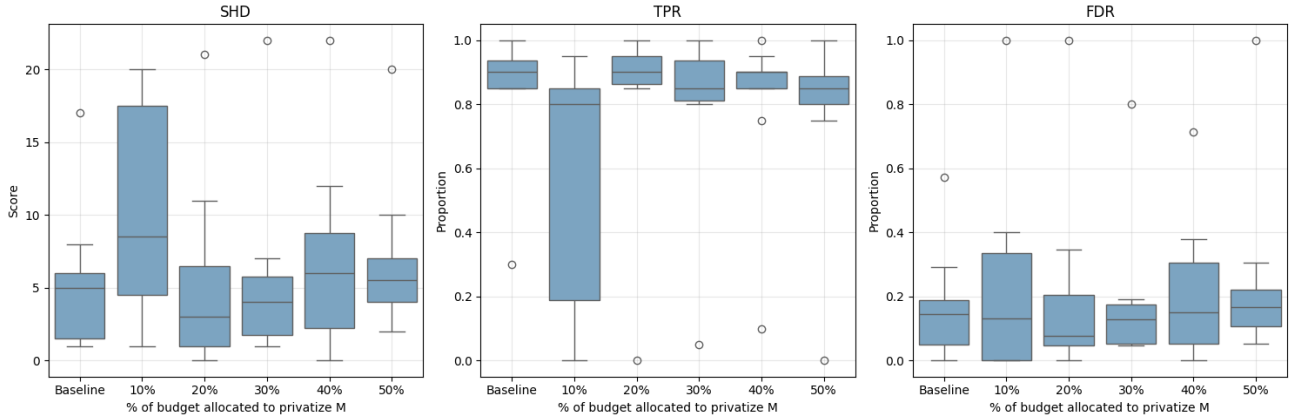


Figure 7: Utility (SHD, TPR, FDR) of DP-Fed-Sparse-BNSL with $\varepsilon_{\text{total}} = 5$ for varying percentages of the total privacy budget allocated to privatize the smoothness constants M_j. 'Baseline' represents DP-Fed-Sparse-BNSL with $\varepsilon = 5$ using exact, non-private smoothness constants.

C.2 Structural Convergence: Federated vs. Local Learning

In Section 6, we demonstrated that Fed-Sparse-BNSL with participant-level personalization yields lower estimation errors (MSE) than local NOTEARS training. Figure 6 provides further insights into this result as the structural level. Although individual local NOTEARS runs converge to reasonably accurate DAGS, Fed-Sparse-BNSL approach consistently attains lower SHD and FDR while maintaining comparable TPR. This indicates that aggregating information across participants improves structural recovery beyond what is achievable locally, even when local solutions are already moderately accurate.

C.3 Privacy-Utility Trade-off for Smoothness Estimation

We empirically evaluate the impact of allocating a fraction of the total privacy budget ε to the private estimation of M_j. Let $\varepsilon_{\text{total}} = \varepsilon_M + \varepsilon$ be the total privacy budget, where ε_M is the budget dedicated to estimating the smoothness constants, and ε is the budget reserved for the main structure learning algorithm. Figure 7 illustrates the end-to-end performance of DP-Fed-Sparse-BNSL with $\varepsilon_{\text{total}} = 5$, across different allocation ratios $\varepsilon_M/\varepsilon_{\text{total}} \in \{10\%, 20\%, 30\%, 40\%, 50\%\}$, compared to a baseline where the exact smoothness constants are assumed to be known (labeled as 'Baseline'). The results are reported over 10 runs.

The results reveal a clear privacy-utility trade-off governed by the budget split:

- **Under-allocation** (10%): allocating too little budget to ε_M results in highly noisy estimates of M_j. Since M_j acts both as a scaling factor in the score function (Eq. 8) and as the inverse step size in the proximal gradient update (Eq. 9), the noise severely impacts the optimization process. This is clearly demonstrated by the large variance and severe degradation in all metrics.

- **Over-allocation** $(40\%, 50\%)$: conversely, allocating too much budget to ε_M yields more accurate smoothness constants but starves the main optimization routine ε. Consequently, the gradients and greedy coordinate selections become too noisy, leading to a degradation in structural accuracy.
- **Balanced allocation** $(20\%, 30\%)$: when allocating approximately 20% to 30% of the budget to ε_M, the algorithm achieves an optimal balance. The performance distributions in this regime are statistically comparable to DP-Fed-Sparse-BNSL with the non-private M baseline.

Conclusion. This empirical evaluation confirms that the assumption of perfectly known smoothness constants can be safely relaxed in practice. By allocating a small portion of the privacy budget, participants can privately estimate their local constants without incurring any significant loss in the final utility.

C.4 Empirical Comparison of DP-PGCD Variants

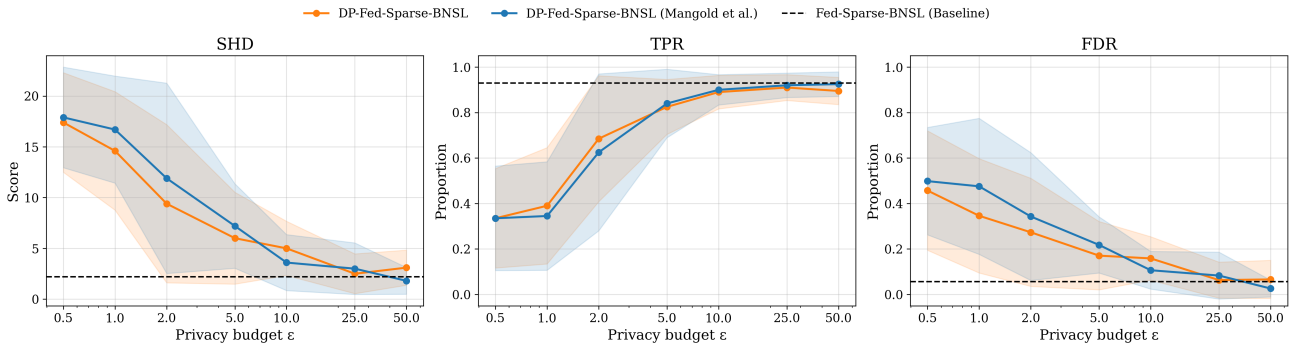


Figure 8: Empirical comparison of our DP-Fed-Sparse-BNSL vs. a variant using Mangold et al. (2023), and the non-private baseline.

To complement the theoretical analysis showing that our improved DP-PGCD algorithm achieves a stronger privacy guarantee for the same amount of noise (see Appendix C.4, Figure 8 shows empirical results comparing DP-Fed-Sparse-BNSL (using our DP-PGCD) with a variant of DP-Fed-Sparse-BNSL that uses the original DP-PGCD of Mangold et al. (2023), as well as with the non-private baseline. The comparison spans SHD, TPR, and FDR metrics as a function of the privacy budget ε. These results confirm that our method consistently achieves higher utility at the same privacy level.