From graphs to DAGs: a low-complexity model and a scalable algorithm

Shuyu Dong, Michèle Sebag

From graphs to DAGs: a low-complexity model and a scalable algorithm

Shuyu Dong[1]✉ and Michèle Sebag[2]

[1] TAU, LISN, INRIA, Université Paris-Saclay, 91190 Gif-sur-Yvette, France
[2] TAU, LISN, CNRS, INRIA, Université Paris-Saclay, 91190 Gif-sur-Yvette, France
shuyu.dong@inria.fr, michele.sebag@lri.fr

Abstract. Learning directed acyclic graphs (DAGs) is long known a critical challenge at the core of probabilistic and causal modeling. The NoTears approach of Zheng et al. [23], through a differentiable function involving the matrix exponential trace $\mathrm{tr}(\exp(\cdot))$, opens up a way to learning DAGs via continuous optimization, though with a $O(d^3)$ complexity in the number d of nodes. This paper presents a low-complexity model, called LoRAM for Low-Rank Additive Model, which combines low-rank matrix factorization with a sparsification mechanism for the continuous optimization of DAGs. The main contribution of the approach lies in an efficient gradient approximation method leveraging the low-rank property of the model, and its straightforward application to the computation of projections from graph matrices onto the DAG matrix space. The proposed method achieves a reduction from a cubic complexity to quadratic complexity while handling the same DAG characteristic function as NoTears and scales easily up to thousands of nodes for the projection problem. The experiments show that the LoRAM achieves efficiency gains of orders of magnitude compared to the state-of-the-art at the expense of a very moderate accuracy loss in the considered range of sparse matrices, and with a low sensitivity to the rank choice of the model's low-rank component.

Keywords: Bayesian networks · low-rank matrix factorization · matrix exponential

1 Introduction

The learning of directed acyclic graphs (DAGs) is an important problem for probabilistic and causal inference [15,17] with important applications in social sciences [11], genome research [21] and machine learning itself [16,2,18]. Through the development of probabilistic graphical models [15,4], DAGs are a most natural mathematical object to describe the causal relations among a number of variables. In today's many application domains, the estimation of DAGs faces intractability issues as an ever growing number d of variables is considered, due to the fact that estimating DAGs is NP-hard [6]. The difficulty lies in how to enforce the acyclicity of graphs. Shimizu et al. [19] combined independent component analysis with the combinatorial linear assignment method to optimize a

linear causal model (LiNGAM) and later proposed a direct and sequential algorithm [20] guaranteeing global optimum of the LiNGAM, for $O(d^4)$ complexities.

Recently, Zheng et al. [23] proposed an optimization approach to learning DAGs. The breakthrough in this work, called NoTears, comes with the characterization of the DAG matrices by the zero set of a real-valued differentiable function on $\mathbb{R}^{d\times d}$, which shows that an $d \times d$ matrix A is the adjacency matrix of a DAG if and only if the *exponential trace* satisfies

$$h(A) := \mathrm{tr}(\exp(A \odot A)) = d, \tag{1}$$

and thus the learning of DAG matrices can be cast as a continuous optimization problem subject to the constraint $h(A) = d$. The NoTears approach broadens the way of learning complex causal relations and provides promising perspectives to tackling large-scale inference problems [10,22,24,14]. However, NoTears is still not suitable for large-scale applications as the complexity of computing the exponential trace and its gradient is $O(d^3)$. More recently, Fang et al. [7] proposed to represent DAGs by low-rank matrices with both theoretical and empirical validation of the low-rank assumption for a range of graph models. However, the adaptation of the NoTears framework [23] to low-rank model still yields a complexity of $O(d^3)$ due to the DAG characteristic function in (1).

The contribution of the paper is to propose a new computational framework and a scalable algorithm to obtain DAGs via low-rank matrix models. We notice that the Hadamard product $\odot$ in characteristic functions as in (1) poses real obstacles to scaling up the optimization of NoTears [23] and NoTears-low-rank [7]. To address these difficulties, we present a low-complexity model, named *Low-Rank Additive Model* (LoRAM), which is a composition of low-rank matrix factorization with sparsification, and then propose a novel approximation method compatible with LoRAM to compute the gradients of the exponential trace in (1). Formally, the gradient approximation—consisting of matrix computation of the form $(A, C, B) \mapsto (\exp(A) \odot C)B$, where $A, C \in \mathbb{R}^{d\times d}$ and B is a thin low-rank matrix—is inspired from the numerical analysis of [1] for the matrix action of $\exp(A)$. We apply the new method to the computation of projections from graphs to DAGs through optimization with the differentiable DAG constraint.

Empirical evidence is presented to identify the cost and benefits of the approximation method combined with Nesterov's accelerated gradient descent [12], depending on the considered range of problem parameters (number of nodes, rank approximation, sparsity of the target graph).

The main contributions are summarized as follows:

- The LoRAM model, combining a low-rank structure with a flexible sparsification mechanism, is proposed to represent DAG matrices, together with a DAG characteristic function generalizing the exponential trace function of NoTears [23].
- An efficient gradient approximation method, exploiting the low-rank and sparse nature of the LoRAM model, is proposed. Under the low-rank assumption ($r \leq C \ll d$), the complexity of the proposed method is quadratic

($O(d^2)$) instead of $O(d^3)$ as shown in Table 1. Large efficiency gains, with insignificant loss of accuracy in some cases, are demonstrated experimentally in the considered range of application.

Table 1. Computational properties of LoRAM and algorithms in related work.

	Search space	Memory req.	Cost for ∇h
NoTears [23]	$\mathbb{R}^{d\times d}$	$O(d^2)$	$O(d^3)$
NoTears-low-rank [7]	$\mathbb{R}^{d\times r}\times\mathbb{R}^{d\times r}$	$O(dr)$	$O(d^3)$
LoRAM (ours)	$\mathbb{R}^{d\times r}\times\mathbb{R}^{d\times r}$	$O(dr)$	$O(d^2r)$

2 Notation and formal background

A graph on d nodes is defined and denoted as a pair $\mathcal{G}=(\mathcal{V},\mathcal{E})$, where $|\mathcal{V}|=d$ and $\mathcal{E}\subset\mathcal{V}\times\mathcal{V}$. By default, a directed graph is simply referred to as a graph. The adjacency matrix of a graph $\mathcal{G}$, denoted as $\mathbb{A}(\mathcal{G})$, is defined as the matrix such that $[\mathbb{A}(\mathcal{G})]_{ij}=1$ if $(i,j)\in\mathcal{E}$ and 0 otherwise. The i-th canonical basis vector in $\mathbb{R}^d$ is denoted by e_i. Let $A_{\mathcal{G}}:=A\in\mathbb{R}^{d\times d}$ be any *weighted* adjacency matrix of a graph $\mathcal{G}$ on d nodes, then by definition, the adjacency matrix $\mathbb{A}(\mathcal{G})$ indicates the nonzeros of A. The (0-1) adjacency matrix $\mathbb{A}(\mathcal{G})$ corresponds exactly to the *support* of A, denoted as $\mathrm{supp}(A)$, i.e., $(i,j)\in\mathrm{supp}(A)$ if and only if $A_{ij}\neq 0$. The number of nonzeros of A is denoted as $\|A\|_0$. We call a matrix A a *DAG matrix* if $\mathrm{supp}(A)$ is the adjacency matrix of a directed acyclic graph (DAG), and denote the set of DAG matrices as $\mathcal{D}_{d\times d}:=\{A\in\mathbb{R}^{d\times d}:\mathrm{supp}(A)\text{ defines a DAG}\}$.

We recall the following theorem that characterizes acyclic graphs using the matrix *exponential trace*—$\mathrm{tr}(\exp(\cdot))$—where $\exp(\cdot)$ denotes the matrix exponential function. The matrix exponential will be denoted as $e^{\cdot}$ and $\exp(\cdot)$ indifferently. The operator $\odot$ denotes the matrix Hadamard product that acts on two matrices of the same size by elementwise multiplications.

Theorem 1 ([23]). *A matrix $A\in\mathbb{R}^{d\times d}$ is a DAG matrix if and only if*

$$\mathrm{tr}(\exp(A\odot A))=d.$$

The following corollary is a straightforward extension of the theorem above:

Corollary 1. *Let $\sigma:\mathbb{R}^{d\times d}\to\mathbb{R}^{d\times d}$ be an operator such that: (i) $\sigma(A)\geq 0$ and (ii) $\mathrm{supp}(A)=\mathrm{supp}(\sigma(A))$, for any $A\in\mathbb{R}^{d\times d}$. Then, $A\in\mathbb{R}^{d\times d}$ is a DAG matrix if and only if $\mathrm{tr}(\exp(\sigma(A)))=d$.*

In view of the property above, we will refer to the composition of $\mathrm{tr}(\exp(\cdot))$ and the operator σ as a *DAG characteristic function*.

Next, we show some more properties of the exponential trace. All proofs in this paper are given in the supplementary material.

Proposition 1. *The exponential trace* $\tilde{h} : \mathbb{R}^{d\times d} \to \mathbb{R} : A \mapsto \mathrm{tr}(\exp(A))$ *satisfies: (i) For all* $\bar{A} \in \mathbb{R}_+^{d\times d}$, $\mathrm{tr}(\exp(\bar{A})) \geq d$ *and* $\mathrm{tr}(\exp(\bar{A})) = d$ *if and only if* $\bar{A}$ *is a DAG matrix. (ii)* $\tilde{h}$ *is nonconvex on* $\mathbb{R}^{d\times d}$*. (iii) The Fréchet derivative of* $\tilde{h}$ *at* $A \in \mathbb{R}^{d\times d}$ *along any direction* $\xi \in \mathbb{R}^{d\times d}$ *is*

$$\mathrm{D}\tilde{h}(A)[\xi] = \mathrm{tr}(\exp(A)\xi),$$

and the gradient of $\tilde{h}$ *at* A *is* $\nabla\tilde{h}(A) = (\exp(A))^{\mathrm{T}}$*.*

3 LoRAM: a low-complexity model

In this section, we describe a low-complexity matrix representation of the adjacency matrices of directed graphs, and then a generalized DAG characteristic function for the new matrix model.

In the spirit of searching for best low-rank singular value decomposition and taking inspiration from [7], the search of a full $d \times d$ (DAG) matrix A is replaced by the search of a pair of thin factor matrices (X, Y), in $\mathbb{R}^{d\times r} \times \mathbb{R}^{d\times r}$ for $1 \leq r < d$, and the $d \times d$ candidate graph matrix is represented by the product XY^{T}. This matrix product has a rank bounded by r, with number of parameters $2dr \leq d^2$. However, the low-rank representation $(X, Y) \mapsto XY^{\mathrm{T}}$ generally gives a dense $d \times d$ matrix. Since in many scenarios the sought graph (or Bayesian network) is usually sparse, we apply a *sparsification* operator on XY^{T} in order to trim abundant entries in XY^{T}. Accordingly, we combine the two operations and introduce the following model.

Definition 1 (LoRAM). *Let* $\Omega \subset [d] \times [d]$ *be a given index set. The* low-rank additive model *(LoRAM), noted* A_Ω*, is defined from the matrix product of* $(X, Y) \in \mathbb{R}^{d\times r} \times \mathbb{R}^{d\times r}$ *sparsified according to* Ω*:*

$$A_\Omega(X, Y) = \mathcal{P}_\Omega(XY^{\mathrm{T}}), \tag{2}$$

where $\mathcal{P}_\Omega : \mathbb{R}^{d\times d} \to \mathbb{R}^{d\times d}$ *is a mask operator such that* $[\mathcal{P}_\Omega(A)]_{ij} = A_{ij}$ *if* $(i, j) \in \Omega$ *and* 0 *otherwise. The set* Ω *is referred to as the* candidate set *of* LoRAM*.*

The candidate set Ω is to be fixed according to the specific problem. In the case of projection from a given graph to the set of DAGs, Ω can be fixed as the index set of the given graph's edges.

The DAG characteristic function on the LoRAM search space is defined as follows:

Definition 2. *Let* $\tilde{h} : \mathbb{R}^{d\times d} \to \mathbb{R} : A \mapsto \mathrm{tr}(\exp(A))$ *denote the exponential trace function. We define* $h : \mathbb{R}^{d\times r} \times \mathbb{R}^{d\times r} \to \mathbb{R}$ *by*

$$h(X, Y) = \mathrm{tr}(\exp(\sigma(A_\Omega(X, Y)))), \tag{3}$$

where $\sigma : \mathbb{R}^{d\times d} \to \mathbb{R}^{d\times d}$ *is one of the following elementwise operators:*

$$\sigma_2(Z) := Z \odot Z \quad \textit{and} \quad \sigma_{\mathrm{abs}}(Z) := \sum_{i,j=1}^{d} |Z_{ij}| e_i {e_j}^{\mathrm{T}}. \tag{4}$$

The operators σ_2 and σ_{abs} (4) are two natural choices for $\text{tr}(\exp(\sigma(\cdot)))$ to be a valid DAG characteristic function (see Corollary 1), since they both produce a nonnegative surrogate matrix of A_Ω while preserving the support of A_Ω.

3.1 Representativity

In the construction of a LoRAM matrix (2), the low-rank component of the model—XY^{T} with $(X, Y) \in \mathbb{R}^{d\times r} \times \mathbb{R}^{d\times r}$—has a rank smaller or equal to r (equality attained when X and Y have full column ranks), and the subsequent sparsification operator $\mathcal{P}_\Omega$ generally induces a change in the rank of the final matrix $A_\Omega = \mathcal{P}_\Omega(XY^{\mathrm{T}})$ (2). Indeed, the rank of $\mathcal{P}_\Omega(XY^{\mathrm{T}})$ depends on an interplay between $(X, Y) \in \mathbb{R}^{d\times r} \times \mathbb{R}^{d\times r}$ and the discrete set $\Omega \subset [d] \times [d]$. The following examples illustrate the two extreme cases of such interplay:

(i) The first extreme case: let Ω be the index set of the edges of a sparse graph $\mathcal{G}_\Omega$, and let $(X, Y) = (\mathbf{1}, \mathbf{1}) \in \mathbb{R}^{d\times 1} \times \mathbb{R}^{d\times 1}$ (matrices of ones), for $r = 1$, then the LoRAM matrix $\mathcal{P}_\Omega(XY^{\mathrm{T}}) = \mathbb{A}(\mathcal{G}_\Omega)$, i.e., the adjacency matrix of $\mathcal{G}_\Omega$. Hence $\text{rank}(\mathcal{P}_\Omega(XY^{\mathrm{T}})) = \text{rank}(\mathbb{A}_\Omega)$, which depends solely on Ω and is generally much larger than $r = 1$.

(ii) The second extreme case: let Ω be the full $[d] \times [d]$ index set, then $\mathcal{P}_\Omega$ reduces to identity such that $\mathcal{P}_\Omega(XY^{\mathrm{T}}) = XY^{\mathrm{T}}$ and $\text{rank}(\mathcal{P}_\Omega(XY^{\mathrm{T}})) = \text{rank}(XY^{\mathrm{T}}) \leq r$ for any $(X, Y) \in \mathbb{R}^{d\times r} \times \mathbb{R}^{d\times r}$.

In the first extreme case above, optimizing LoRAM (2) for DAG learning boils down to choosing the most relevant edge set Ω, which is a NP-hard combinatorial problem [6]. In the second extreme case, the optimization of LoRAM (2) reduces to learning the most pertinent low-rank matrices $(X, Y) \in \mathbb{R}^{d\times r} \times \mathbb{R}^{d\times r}$, which coincides with optimizing the NoTears-low-rank [7] model.

In this work, we are interested in settings between the two extreme cases above such that both $(X, Y) \in \mathbb{R}^{d\times r} \times \mathbb{R}^{d\times r}$ and the candidate set Ω have sufficient degrees of freedom. Consequently, the representativity of LoRAM depends on both the rank parameter r and Ω.

Next, we present a way of quantifying the representativity of LoRAM with respect to a subset $\mathcal{D}^\star_{d\times d}$ of DAG matrices. The restriction to a subset $\mathcal{D}^\star_{d\times d}$ is motivated by the revelation that certain types of DAG matrices—such as those with many hubs—can be represented by low-rank matrices [7].

Definition 3. *Let $\mathcal{D}^\star_{d\times d} \subset \mathcal{D}_{d\times d}$ be a given set of nonzero DAG matrices. For $Z_0 \in \mathcal{D}^\star_{d\times d}$, let $A^*_\Omega(Z_0)$ denote any* LoRAM *matrix* (2) *such that $\|A^*_\Omega(Z_0) - Z_0\| = \min_{(X,Y)\in\mathbb{R}^{d\times r}\times\mathbb{R}^{d\times r}} \|A_\Omega(X, Y) - Z_0\|$, then the relative error of* LoRAM *w.r.t. $\mathcal{D}^\star_{d\times d}$ is defined and denoted as $\epsilon^\star_{r,\Omega} = \max_{Z\in\mathcal{D}^\star_{d\times d}} \left\{ \frac{\|A^*_\Omega(Z) - Z\|_{\max}}{\|Z\|_{\max}} \right\}$, where $\|Z\|_{\max} := \max_{ij} |Z_{ij}|$ denotes the matrix max-norm. For $Z_0 \in \mathcal{D}^\star_{d\times d}$, $A^*_\Omega(Z_0)$ is referred to as an $\epsilon^\star_{r,\Omega}$-quasi DAG matrix.*

Note that the existence of $A^*_\Omega(Z_0)$ for any $Z_0 \in \mathcal{D}^\star_{d\times d}$ is guaranteed by the closeness of the image set of LoRAM (2).

Based on the relative error above, the next proposition provides a quasi DAG characterization of the LoRAM via function h (3).

Proposition 2. *Given a set $\mathcal{D}^{\star}_{d\times d} \subset \mathcal{D}_{d\times d}$ of nonzero DAG matrices. For any $Z_0 \in \mathcal{D}^{\star}_{d\times d}$ such that $\|Z_0\|_{\max} \leq 1$, without loss of generality, the minima of $\min_{(X,Y)\in\mathbb{R}^{d\times r}\times\mathbb{R}^{d\times r}} \|A_\Omega(X,Y) - Z_0\|$ belong to the set*

$$\{(X,Y) \in \mathbb{R}^{d\times r} \times \mathbb{R}^{d\times r} : h(X,Y) - d \leq C_0 \epsilon^{\star}_{r,\Omega}\} \tag{5}$$

where $\epsilon^{\star}_{r,\Omega}$ is given in Definition 3 and $C_0 = \big(C_1\|Z_0\|_0 + \sum_{ij}[e^{\sigma(Z_0)}]_{ij}\big)\|Z_0\|_{\max}$ for a constant $C_1 \geq 0$.

Remark 1. The constant C_0 in Proposition 2 can be seen as a measure of total capacity of passing from one node to other nodes, and therefore depends on d, $\|Z_0\|_{\max}$ (bounded by 1), the sparsity and the average degree of the graph of Z_0. For DAG matrices with sparsity $\rho_0 \sim 10^{-3}$ and $d \lesssim 10^3$, one can expect that $C_0 \leq Cd$ for a constant C. □

The result of Proposition 2 establishes that, under the said conditions, a given DAG matrix Z_0 admits low rank approximations $A_\Omega(X,Y)$ satisfying $h(X,Y) - d \leq \delta_\epsilon$ for a small enough parameter δ_ϵ. In other words, the low-rank projection with a relaxed DAG constraint admits solutions.

The general case of projecting a non-acyclic graph matrix Z_0 onto a low-rank matrix under a relaxed DAG constraint is considered in next section.

4 Scalable projection from graphs to DAGs

Given a (generally non-acyclic) graph matrix $Z_0 \in \mathbb{R}^{d\times d}$, let us consider the projection of Z_0 onto the feasible set (5):

$$\min_{(X,Y)\in\mathbb{R}^{d\times r}\times\mathbb{R}^{d\times r}} \frac{1}{2}\|A_\Omega(X,Y) - Z_0\|_{\mathrm{F}}^2 \quad \text{subject to } h(X,Y) - d \leq \delta_\epsilon, \tag{6}$$

where $A_\Omega(X,Y)$ is the LoRAM matrix (2), h is given by Definition 2, and $\delta_\epsilon > 0$ is a tolerance parameter. Based on Proposition 2 and given the objective function of (6), the solution to (6) provides a quasi DAG matrix closest to Z_0 and thus enables finding a projection of Z_0 onto the DAG matrix space $\mathcal{D}_{d\times d}$. More precisely, we tackle problem (6) using the penalty method and focus on the primal problem, for a given penalty parameter $\lambda > 0$, followed by elementwise hard thresholding:

$$(X^*, Y^*) = \operatorname*{arg\,min}_{(X,Y)\in\mathbb{R}^{d\times r}\times\mathbb{R}^{d\times r}} h(X,Y) + \frac{1}{2\lambda}\|A_\Omega(X,Y) - Z_0\|_{\mathrm{F}}^2, \tag{7}$$

$$A^* = \mathbb{T}_{\epsilon^{\star}_{r,\Omega}}(A_\Omega(X^*, Y^*)), \tag{8}$$

where $\mathbb{T}_{\epsilon^{\star}_{r,\Omega}}(z) = z\delta_{|z|\geq\epsilon^{\star}_{r,\Omega}}$ is the elementwise hard thresholding operator. The choice of λ and $\epsilon^{\star}_{r,\Omega}$ for problem (6) is discussed in the supplementary.

Remark 2. To obtain any DAG matrix closest to (the non-acyclic) Z_0, it is necessary to break the cycles in $\mathcal{G}(Z_0)$ by suppressing certain edges. Hence, we assume that the edge set of the sought DAG is a strict subset of $\mathrm{supp}(Z_0)$ and thus fix the candidate set Ω to be $\mathrm{supp}(Z_0)$. □

4.1 Gradient of the DAG characteristic function and an efficient approximation

Problems (6) and (7) are nonconvex due to: (i) the matrix exponential trace $A \mapsto \operatorname{tr}(\exp(A))$ in h (3) is nonconvex (as in [23]), and (ii) the matrix product $(X, Y) \mapsto XY^{\mathrm{T}}$ in $A_\Omega(X, Y)$ (2) is nonconvex.

In view of the DAG characteristic constraint of (6), the augmented Lagrangian algorithms of NoTears [23] and NoTears-low-rank [7] can be applied for the same objective as problem (6); it suffices for NoTears and NoTears-low-rank to replace the LoRAM matrix $A_\Omega(X, Y)$ in (6) by the $d \times d$ matrix variable and the dense matrix product XY^{T} respectively. However, the NoTears-based methods of [23,7] have an $O(d^3)$ complexity due to the composition of the elementwise operations (the Hadamard product $\odot$) with the matrix exponential in function h (1) and (3). We elaborate this argument in the rest of this subsection and then propose a new computational method for computations involving the gradient of function h (3).

Lemma 1. *For any $Z \in \mathbb{R}^{d\times d}$ and $\xi \in \mathbb{R}^{d\times d}$, the differentials of σ_2 and σ_{abs} (4) are*

$$\mathrm{D}\sigma_2(Z)[\xi] = 2Z \odot \xi \quad \textit{and} \quad \hat{\mathrm{D}}\sigma_{\mathrm{abs}}(Z)[\xi] = \operatorname{sign}(Z) \odot \xi, \tag{4b}$$

where $\operatorname{sign}(\cdot)$ is the element-wise sign function such that $[\operatorname{sign}(Z)]_{ij} = \frac{Z_{ij}}{|Z_{ij}|}$ if $Z_{ij} \neq 0$ and 0 otherwise.

Theorem 2. *The gradient of h (3) is*

$$\nabla h(X, Y) = (\mathcal{S}Y, \mathcal{S}^{\mathrm{T}}X) \in \mathbb{R}^{d\times r} \times \mathbb{R}^{d\times r}, \tag{9}$$

where $\mathcal{S} \in \mathbb{R}^{d\times d}$ has the following expressions, depending on the choice of σ in (4): with $A_\Omega := A_\Omega(X, Y)$ for brevity,

$$\mathcal{S}_2 = 2(\exp(\sigma_2(A_\Omega))^{\mathrm{T}}) \odot A_\Omega, \tag{10}$$

$$\mathcal{S}_{\mathrm{abs}} = (\exp(\sigma_{\mathrm{abs}}(A_\Omega))^{\mathrm{T}}) \odot \operatorname{sign}(A_\Omega). \tag{11}$$

Proof. From Proposition 1-(iii), the Fréchet derivative of the exponential trace $\tilde{h}$ at $A \in \mathbb{R}^{d\times d}$ is $\mathrm{D}\tilde{h}(A)[\xi] = \operatorname{tr}(\exp(A)\xi)$ for any $\xi \in \mathbb{R}^{d\times d}$. By the chain rule and Lemma 1, the Fréchet derivative of h (3) for $\sigma = \sigma_2$ is as follows: with $A_\Omega = \mathcal{P}_\Omega(XY^{\mathrm{T}})$,

$$\begin{aligned}
\mathrm{D}_X h(X, Y)[\xi] &= \operatorname{tr}\big(\exp(\sigma(A_\Omega))\mathrm{D}\sigma(A_\Omega)\mathrm{D}\mathcal{P}_\Omega(XY^{\mathrm{T}})[\xi Y^{\mathrm{T}}]\big) \\
&= \operatorname{tr}\big(\exp(\sigma(A_\Omega))\mathrm{D}\sigma(A_\Omega)[\mathcal{P}_\Omega(\xi Y^{\mathrm{T}})]\big) && (12) \\
&= 2\operatorname{tr}\big(\exp(\sigma(A_\Omega))(A_\Omega \odot \mathcal{P}_\Omega(\xi Y^{\mathrm{T}}))\big) \\
&= 2\operatorname{tr}\big(\exp(\sigma(A_\Omega))(A_\Omega \odot (\xi Y^{\mathrm{T}}))\big) && (13) \\
&= 2\operatorname{tr}\big((\exp(\sigma(A_\Omega)) \odot {A_\Omega}^{\mathrm{T}})(\xi Y^{\mathrm{T}})\big) && (14) \\
&= 2\operatorname{tr}\big(Y^{\mathrm{T}}(\exp(\sigma(A_\Omega)) \odot {A_\Omega}^{\mathrm{T}})\xi\big),
\end{aligned}$$

where (13) holds, i.e., $\mathcal{P}_\Omega(XY^{\mathrm{T}}) \odot \mathcal{P}_\Omega(\xi Y^{\mathrm{T}}) = \mathcal{P}_\Omega(XY^{\mathrm{T}}) \odot \xi Y^{\mathrm{T}}$ because $A \odot \mathcal{P}_\Omega(B) = \mathcal{P}_\Omega(A) \odot B$ (for any A and B) and $\mathcal{P}_\Omega^2 = \mathcal{P}_\Omega$, and (14) holds because $\mathrm{tr}(A(B \odot C)) = \mathrm{tr}((A \odot B^{\mathrm{T}})C)$ for any A, B, C (with compatible sizes). By identifying the above equation with $\mathrm{tr}(\nabla_X h(X,Y)^{\mathrm{T}} \xi)$, we have $\nabla_X h(X,Y) = \underbrace{2\big(\exp(\sigma(Z))^{\mathrm{T}} \odot A_\Omega\big)}_{\mathcal{S}} Y$, hence the expression (10) for $\mathcal{S}$. The expression of $\mathcal{S}_{\mathrm{abs}}$ for $\sigma = \sigma_{\mathrm{abs}}$ can be obtained using the same calculations and (4b). □

The computational bottleneck for the exact gradient (9) lies in the computation of the $d \times d$ matrix $\mathcal{S}$ (10)–(11) and is due to the difference between Hadamard product and matrix multiplication; see the supplementary material for details. Nevertheless, we note that the multiplication $(\mathcal{S}, X) \mapsto \mathcal{S}X$ is similar to the action of a matrix exponential $(A, X) \mapsto \exp(A)X$, which can be computed using only a number of repeated multiplications of a $d \times d$ matrix with the thin matrix $X \in \mathbb{R}^{d \times r}$ based on Al-Mohy and Higham's results [1].

A difficulty in adapting the method of [1] also lies in the presence of Hadamard product in $\mathcal{S}$ (10)–(11). Once the sparse $d \times d$ matrix $A := \sigma(A_\Omega)$ in (10)–(11) is obtained (see the supplementary) the exact computation of $(A, C, B) \mapsto (\exp(A) \odot C)B$, using the Taylor expansion of $\exp(\cdot)$ to a certain order m_*, is to compute $\frac{1}{k!}(A^k \odot C)B$ at each iteration, which inevitably requires the computation of the $d \times d$ matrix product A^k (in the form of $A(A^{k-1})$) before computing the Hadamard product, which requires an $O(d^3)$ cost.

To alleviate the obstacle above, we propose to use inexact[3] incremental multiplications; see Algorithm 1.

Algorithm 1 Approximation of $(A, C, B) \mapsto (\exp(A) \odot C)B$

Input: $d \times d$ matrices A and C, thin matrix $B \in \mathbb{R}^{d \times r}$, tolerance $\mathrm{tol} > 0$
Output: $F \approx (\exp(A) \odot C)B \in \mathbb{R}^{d \times r}$

1: Estimate the Taylor order parameter m_* from A # using [1, Algorithm 3.2]
2: Initialize: let $F = (I \odot C)B$
3: **for** $k = 1, \ldots, m_*$ **do**
4: $\quad B \leftarrow \frac{1}{k+1}(A \odot C)B$
5: $\quad F \leftarrow F + B$
6: $\quad B \leftarrow F$
7: **end for**
8: Return F.

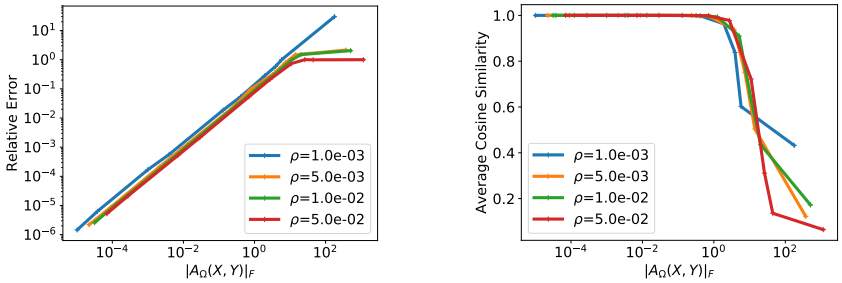
In line 1 of Algorithm 1, the value of m_* is obtained from numerical analysis results of [1, Algorithm 3.2]; often, the value of m_*, depending on the spectrum of A, is a bounded constant (independent of the matrix size d). Therefore, the dominant cost of Algorithm 1 is $2m_*|\Omega|r \lesssim d^2 r$, since each iteration (lines 4–6)

[3] It is inexact because $((A \odot C)^{k+1})B \neq (A^{k+1} \odot C)B$.

costs $(2|\Omega|r+dr) \approx 2|\Omega|r$ flops. Table 1 summarizes this computational property in comparison with existing methods.

Reliability of Algorithm 1. The accuracy of Algorithm 1 with respect to the exact computation of $(A, C, B) \mapsto (\exp(A) \odot C)B$ depends notably on the scale of A, since the differential $\mathrm{D}\exp(A)$ at A has a greater operator norm when the norm of A is greater; see [8, Theorem 2.b] and the supplementary.

To illustrate the remark above, we approximate $\nabla h(X, Y)$ (9) by Algorithm 1 on random points of $\mathbb{R}^{d\times r} \times \mathbb{R}^{d\times r}$ with different scales, with Ω defined from the edges of a random sparse graph; the results are shown in Figure 1.



(a) Relative error (b) Cosine similarity

Fig. 1. Average accuracy of Algorithm 1 in approximating $\nabla h(X, Y)$ (9) at (X, Y) with different scales and sparsity levels $\rho = \frac{|\Omega|}{d^2}$. Number of nodes $d = 200$, $r = 40$.

We observe from Figure 1 that Algorithm 1 is reliable, i.e., having gradient approximations with cosine similarity close to 1, when the norm of $A_\Omega(X, Y)$ is sufficiently bounded. More precisely, for $c_0 = 10^{-1}$, Algorithm 1 is reliable in the following set

$$\mathcal{D}(c_0) = \{(X, Y) \in \mathbb{R}^{d\times r} \times \mathbb{R}^{d\times r} : \|A_\Omega(X, Y)\|_{\mathrm{F}} \leq c_0\}. \tag{15}$$

The degrading accuracy of Algorithm 1 outside $\mathcal{D}(c_0)$ (15) can nonetheless be avoided for the projection problem (6), in particular, through *rescaling* of the input matrix Z_0. Note that the edge set of any graph is invariant to the scaling of its weighted adjacency matrix, and that any DAG A^* solution to (6) satisfies $\|A^*\|_{\mathrm{F}} \lesssim \|Z_0\|_{\mathrm{F}}$ since $\mathrm{supp}(A^*) \subset \mathrm{supp}(Z_0)$ (see Remark 2). Hence it suffices to rescale Z_0 with a small enough scalar, e.g., replace Z_0 with $Z_0' = \frac{c_0}{10\|Z_0\|_{\mathrm{F}}} Z_0$, without loss of generality, in (6)–(7). Indeed, this rescaling ensures that both the input matrix Z_0' and matrices like $A^{*\prime} = \frac{c_0}{10\|Z_0\|_{\mathrm{F}}} A^*$—a DAG matrix equivalent to A^*—stay confined in the image (through LoRAM) of $\mathcal{D}(c_0)$ (15), in which the gradient approximations by Algorithm 1 are reliable.

4.2 Accelerated gradient descent

Given the gradient computation method (Algorithm 1) for the h function, we adapt Nesterov's accelerated gradient descent [12,13] to solve (7). The accelerated gradient descent is used for its superior performance than vanilla gradient descent in many convex and nonconvex problems while it also only requires first-order information of the objective function. Details of this algorithm for our LoRAM optimization is given in Algorithm 2.

Algorithm 2 Accelerated Gradient Descent of LoRAM (LoRAM-AGD)

Input: Initial point $x_0 = (X_0, Y_0) \in \mathbb{R}^{d\times r} \times \mathbb{R}^{d\times r}$, objective function F, tolerance ϵ.

Output: $x_t \in \mathbb{R}^{d\times r} \times \mathbb{R}^{d\times r}$

1: Make a gradient descent: $x_1 = x_0 - s_0 \nabla F(x_0)$ for an initial stepsize $s_0 > 0$
2: Initialize: $y_0 = x_0$, $y_1 = x_1$, $t = 1$.
3: **while** $\|\nabla F(x_t)\| > \epsilon$ **do**
4: Compute $\nabla F(y_t) = \nabla f(y_t) + \alpha \nabla h(y_t)$ # using Algorithm 1 for $\nabla h(y_t)$
5: Compute the Barzilai–Borwein stepsize: $s_t = \frac{\|z_{t-1}\|^2}{\langle z_{t-1}, w_{t-1}\rangle}$, where $z_{t-1} = y_t - y_{t-1}$ and $w_{t-1} = \nabla F(y_t) - \nabla F(y_{t-1})$.
6: Updates with Nesterov's acceleration:

$$\begin{aligned} x_{t+1} &= y_t - s_t \nabla F(y_t), \\ y_{t+1} &= x_{t+1} + \frac{t}{t+3}(x_{t+1} - x_t). \end{aligned} \tag{16}$$

7: $t = t + 1$
8: **end while**

Specifically, in line 5 of Algorithm 2, the Barzilai–Borwein (BB) stepsize [3] is used for the descent step (16). The computation of the BB stepsize s_t requires evaluating the inner products $\langle z_{t-1}, w_{t-1}\rangle$ and the norm $\|z_{t-1}\|$, where $z_{t-1}, w_{t-1} \in \mathbb{R}^{d\times r} \times \mathbb{R}^{d\times r}$; we choose the Euclidean inner product as the metric on $\mathbb{R}^{d\times r} \times \mathbb{R}^{d\times r}$: for any pair of points $z = (z^{(1)}, z^{(2)})$ and $w = (w^{(1)}, w^{(2)})$ on $\mathbb{R}^{d\times r} \times \mathbb{R}^{d\times r}$, $\langle z, w\rangle = \mathrm{tr}({z^{(1)}}^{\mathrm{T}} w^{(1)}) + \mathrm{tr}({z^{(2)}}^{\mathrm{T}} w^{(2)})$. Note that one can always use backtracking line search based on the stepsize estimation (line 5). We choose to use the BB stepsize directly since it does not require any evaluation of the objective function, and thus avoids non-negligible costs for computing the matrix exponential trace in h (3). We refer to [5] for a comprehensive view on accelerated methods for nonconvex optimization.

Due to the nonconvexity of h (3) (see Proposition 1) and thus (7), we aim at finding stationary points of (7). In particular, empirical results in Section 5.2 show that the solutions by the proposed method, with close-to-zero or even zero SHDs to the true DAGs, are close to global optima in practice.

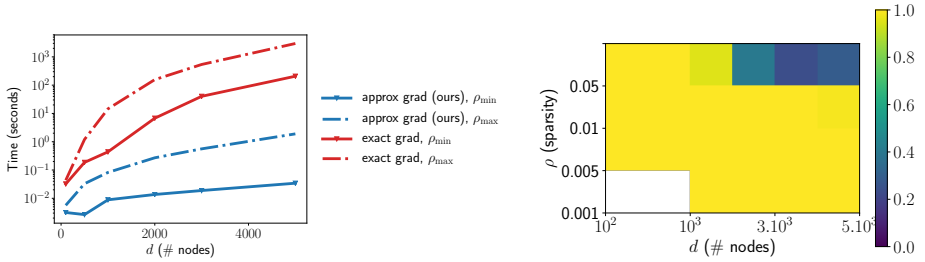
5 Experimental validation

This section investigates the performance (computational gains and accuracy loss) of the proposed gradient approximation method (Algorithm 1) and thereafter reports on the performance of the LoRAM projection (6), compared to NoTears [23]. Sensitivity to the rank parameter r of the proposed method is also investigated.

The implementation is available at https://github.com/shuyu-d/loram-exp.

5.1 Gradient computations

We compare the performance of Algorithm 1 in gradient approximations with the exact computation in the following settings: the number of nodes $d \in \{100, 500, 10^3, 2.10^3, 3.10^3, 5.10^3\}$, $r = 40$, and the sparsity ($\frac{|\Omega|}{d^2}$) of the index set Ω tested are $\rho \in \{10^{-3}, 5.10^{-3}, 10^{-2}, 5.10^{-2}\}$. The results shown in Figure 2 are based on the computation of $\nabla h(X, Y)$ (9) on randomly generated points $(X, Y) \in \mathbb{R}^{d\times r} \times \mathbb{R}^{d\times r}$, where X and Y are Gaussian matrices.



(a) Computation time (b) Accuracy

Fig. 2. (a): Runtime (in log-scale) for computing $\nabla h(X, Y)$. (b): Cosine similarities between the approximate and the exact gradients.

From Figure 2 (a), Algorithm 1 shows a significant reduction in the runtime for computing the gradient of h (3) at the expense of a very moderate loss of accuracy (Figure 2 (b)): the approximate gradients are mostly sufficiently aligned with exact gradients in the considered range of graph sizes and sparsity levels.

5.2 Projection from graphs to DAGs

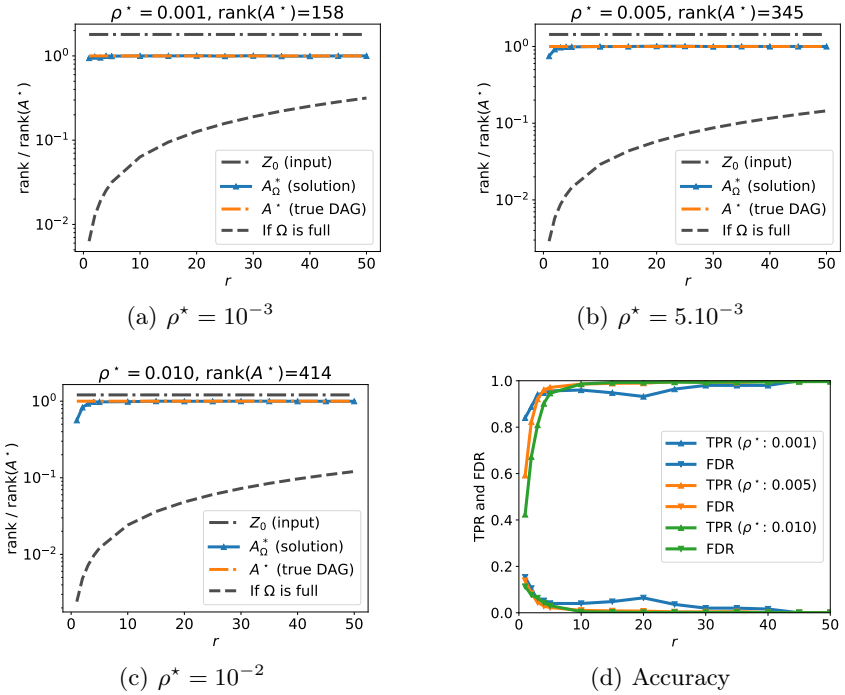
In the following experiments, we generate the input matrix Z_0 of problem (6) by

$$Z_0 = A^\star + E, \tag{17}$$

where $A^\star$ is a given $d \times d$ DAG matrix and E is a graph containing additive noisy edges that break the acyclicity of the observed graph Z_0.

The ground truth DAG matrix $A^\star$ is generated from the acyclic Erdős-Rényi (ER) subset, in the same way as in [23], with a sparsity rate $\rho \in \{10^{-3}, 5.10^{-3}, 10^{-2}\}$. The noise graph E of (17) is defined as $E = \sigma_E {A^\star}^{\mathrm{T}}$, which consists of edges that create a confusion between causes and effects since these edges are *reversed*, pointing from the (true) effects to their respective causes.

Sensitivity to the rank parameter r. We evaluate the performance of LoRAM-AGD in the proximal mapping computation (7) for $d = 500$ and different values of the rank parameter r. In all these evaluations, the candidate set Ω is fixed to be $\mathrm{supp}(Z_0)$; see Remark 2. We measure the accuracy of solutions by the false discovery rate (FDR, lower is better), false positive rate (FPR), true positive rate (TPR, higher is better), and the structural Hamming distance (SHD, lower is better) of the solution compared to the DAG $\mathcal{G}(A^\star)$.



(a) $\rho^\star = 10^{-3}$ (b) $\rho^\star = 5.10^{-3}$

(c) $\rho^\star = 10^{-2}$ (d) Accuracy

Fig. 3. (a)–(c): Rank profiles $\mathrm{rank}(\cdot)/\,\mathrm{rank}(A^\star)$, and (d) projection accuracy for different values of r (number of columns of X and Y in (2)). The number of nodes $d = 500$. Sparsity of $Z_0 = A^\star + \sigma_E {A^\star}^{\mathrm{T}}$ (17) are $\rho^\star \in \{10^{-3}, 5.10^{-3}, 10^{-2}\}$.

The results in Figure 3 suggest that:

(i) For each sparsity level, increasing the rank parameter r generally improves the projection accuracy of the LoRAM.
(ii) While the rank parameter r of (2) attains at most around 5, which is only $\frac{1}{100}$-th the rank of the input matrix Z_0 and the ground truth $A^\star$, the rank of the solution $A^*_\Omega = A_\Omega(X^*, Y^*)$ attains the same value as rank$(A^\star)$. This means that the rank representativity of the LoRAM goes beyond the value of r. This phenomenon is understandable in the present case where the candidate set $\Omega = \text{supp}(Z_0)$ is fairly close to the sparse edge set supp$(A^\star)$.
(iii) The projection accuracy in TPR and FDR (and also SHD, see the supplementary) of LoRAM-AGD is close to optimum on a wide interval $25 \leq r \leq 50$ of the tested ranks and are fairly stable on this interval.

Scalability. We examine two different types of noisy edges (in E) as follows. Case (a): Bernoulli-Gaussian $E = E(\sigma_E, p)$, where $E_{ij}(\sigma_E, p) \neq 0$ with probability p and all nonzeros of $E(\sigma_E, p)$ are i.i.d. samples of $\mathcal{N}(0, \sigma_E)$. Case (b): cause-effect confusions $E = \sigma_E A^{\star\mathrm{T}}$ as in the beginning of Section 5.2.

The initial factor matrices $(X, Y) \in \mathbb{R}^{d\times r} \times \mathbb{R}^{d\times r}$ are random Gaussian matrices. For the LoRAM, we set Ω to be the support of Z_0; see Remark 2. The penalty parameter λ of (7) is varied in $\{2.0, 5.0\}$ with no fine tuning.

In case (a), we test with various noise levels for $d = 500$ nodes. In case (b), we test on various graph dimensions, for $(d, r) \in \{100, 200, \ldots, 2000\} \times \{40, 80\}$. The results are given in Table 2–Table 3 respectively.

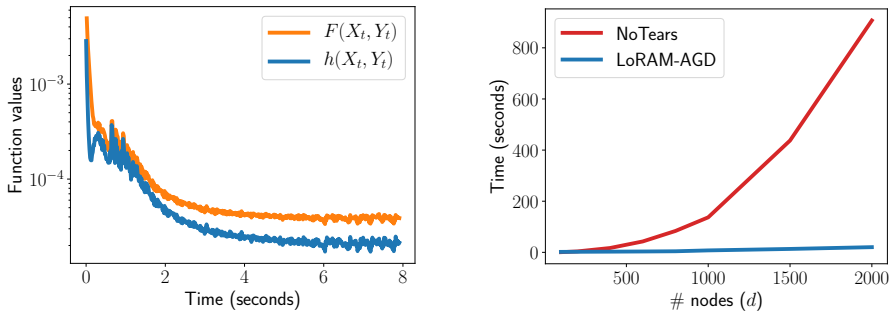
Table 2. Results in case (a): the noise graph is $E(\sigma_E, p)$ for $p = 5.10^{-4}$ and $d = 500$.

σ_E	LoRAM (ours) / NoTears			
	Runtime (sec)	TPR	FDR	SHD
0.1	1.34 / 5.78	1.0 / 1.0	9.9e-3 / 0.0e+0	25.0 / 0.0
0.2	2.65 / 11.58	1.0 / 1.0	9.5e-3 / 0.0e+0	24.0 / 0.0
0.3	1.35 / 28.93	1.0 / 1.0	9.5e-3 / 8.0e-4	24.0 / 2.0
0.4	1.35 / 18.03	1.0 / 1.0	9.9e-3 / 3.2e-3	25.0 / 9.0
0.5	1.35 / 12.52	1.0 / 1.0	9.9e-3 / 5.2e-3	25.0 / 13.0
0.6	2.57 / 16.07	1.0 / 1.0	9.5e-3 / 4.4e-3	24.0 / 11.0
0.7	1.35 / 18.72	1.0 / 1.0	9.9e-3 / 5.2e-3	25.0 / 13.0
0.8	1.35 / 32.03	1.0 / 1.0	9.9e-3 / 4.8e-3	25.0 / 15.0

The results in Table 2–Table 3 show that: (i) in case (a), the solutions of LoRAM-AGD are close to the ground truth despite slightly higher errors than NoTears in terms of FDR and SHD; (ii) in case (b), the solutions of LoRAM-AGD are almost identical to the ground truth $A^\star$ in (17) in all performance indicators (also see the supplementary); (iii) in terms of computation time (see Figure 4), the proposed LoRAM-AGD achieves significant speedups (around 50 times faster when $d = 2000$) compared to NoTears and also has a smaller growth rate with respect to the problem dimension d, showing good scalability.

Table 3. Results in case (b): the noise graph $E = \sigma_E A^{\star\mathrm{T}}$ contains cause-effect confusions for $\sigma_E = 0.4$.

(α, r)	d	LoRAM (ours) / NoTears			
		Runtime (sec)	TPR	FDR	SHD
(5.0, 40)	100	1.82 / 0.67	1.00 / 1.00	0.00e+0 / 0.0	0.0 / 0.0
(5.0, 40)	200	2.20 / 3.64	0.98 / 0.95	2.50e-2 / 0.0	1.0 / 2.0
(5.0, 40)	400	2.74 / 16.96	0.98 / 0.98	2.50e-2 / 0.0	4.0 / 4.0
(5.0, 40)	600	3.40 / 42.65	0.98 / 0.96	1.67e-2 / 0.0	6.0 / 16.0
(5.0, 40)	800	4.23 / 83.68	0.99 / 0.97	7.81e-3 / 0.0	5.0 / 22.0
(2.0, 80)	1000	7.63 / 136.94	1.00 / 0.96	0.00e+0 / 0.0	0.0 / 36.0
(2.0, 80)	1500	13.34 / 437.35	1.00 / 0.96	8.88e-4 / 0.0	2.0 / 94.0
(2.0, 80)	2000	20.32 / 906.94	1.00 / 0.96	7.49e-4 / 0.0	3.0 / 148.0



(a) Iterations of LoRAM-AGD (for $d = 1000$)

(b) Runtime vs number of nodes

Fig. 4. (a): An iteration history of LoRAM-AGD for (7) with $d = 1000$. (b): Runtime comparisons for different number d of nodes.

6 Discussion and Perspectives

This paper tackles the computation of projection from graphs to DAGs, motivated by problems for causal DAG discovery. The line of research built upon the LiNGAM algorithms [19,20] has recently been revisited through the formal characterization of DAGness in terms of a continuously differentiable constraint by [23]. The NoTears approach of [23] however suffers from an $O(d^3)$ complexity in the number d of variables, precluding its usage for large-scale problems.

Unfortunately, this difficulty is not related to the complexity of the model (number of parameters of the model): the low-rank approach investigated by NoTears-low-rank [7] also suffers from the same $O(d^3)$ complexity, incurred in the gradient-based optimization phase.

The present paper addresses this difficulty by combining a sparsification mechanism with a low-rank matrix model and using a new approximated gradient computation. This approximated gradient takes inspiration from the ap-

proach of [1] for computing the action of exponential matrices based on truncated Taylor expansion. This approximation eventually yields a complexity of $O(d^2 r)$, where the rank parameter is small ($r \leq C \ll d$). The experimental validation of the approach shows that the approximated gradient entails no significant error with respect to the exact gradient, for LoRAM matrices with a bounded norm, in the considered range of graph sizes (d) and sparsity levels. The proposed algorithm combining the approximated gradient with a Nesterov's acceleration method [12,13] yields gains of orders of magnitude in computation time compared to NoTears on the same artificial benchmark problems. The approximation performance indicators reveal almost no performance loss for the projection problem in the setting of case (b) (where the matrix to be projected is perturbed with anti-causal links), while it incurs minor losses in terms of false discovery rate (FDR) in the setting of case (a) (with random additive spurious links).

The proposed method has the general purpose of computing proximate DAGs from given graphs. Further developments aim to apply this method to the identification of causal DAGs from observational data. A longer term perspective is to extend LoRAM to non-linear causal models, building upon latent causal variables and taking inspiration from non-linear independent component analysis and generalized contrastive losses [9]. Another perspective relies on the use of auto-encoders to yield a compressed representation of high-dimensional data, while constraining the structure of the encoder and decoder modules to enforce the acyclic property.

Acknowledgement

The authors warmly thank Fujitsu Laboratories LTD who funded the first author, and in particular Hiroyuki Higuchi and Maruashi Koji for many discussions.

References

1. Al-Mohy, A.H., Higham, N.J.: Computing the action of the matrix exponential, with an application to exponential integrators. SIAM journal on scientific computing **33**(2), 488–511 (2011)
2. Arjovsky, M., Bottou, L., Gulrajani, I., Lopez-Paz, D.: Invariant risk minimization. arXiv preprint arXiv:1907.02893 (2019)
3. Barzilai, J., Borwein, J.M.: Two-point step size gradient methods. IMA journal of numerical analysis **8**(1), 141–148 (1988)
4. Bühlmann, P., Peters, J., Ernest, J.: Cam: Causal additive models, high-dimensional order search and penalized regression. The Annals of Statistics **42**(6), 2526–2556 (2014)
5. Carmon, Y., Duchi, J.C., Hinder, O., Sidford, A.: Accelerated methods for non-convex optimization. SIAM Journal on Optimization **28**(2), 1751–1772 (2018)
6. Chickering, D.M.: Learning bayesian networks is NP-complete. In: Learning from data, pp. 121–130. Springer (1996)

7. Fang, Z., Zhu, S., Zhang, J., Liu, Y., Chen, Z., He, Y.: Low rank directed acyclic graphs and causal structure learning. arXiv preprint arXiv:2006.05691 (2020)
8. Haber, H.E.: Notes on the matrix exponential and logarithm. Santa Cruz Institute for Particle Physics, University of California: Santa Cruz, CA, USA (2018)
9. Hyvarinen, A., Sasaki, H., Turner, R.: Nonlinear ICA using auxiliary variables and generalized contrastive learning. In: International Conference on Artificial Intelligence and Statistics (AISTATS) (2019)
10. Kalainathan, D., Goudet, O., Guyon, I., Lopez-Paz, D., Sebag, M.: Structural agnostic modeling: Adversarial learning of causal graphs. arXiv preprint arXiv:1803.04929 (2018)
11. Morgan, S.L., Winship, C.: Counterfactuals and causal inference. Cambridge University Press (2015)
12. Nesterov, Y.: A method for solving the convex programming problem with convergence rate $O(1/k^2)$. Soviet Mathematics Doklady **27:372–376** (1983)
13. Nesterov, Y.: Introductory Lectures on Convex Optimization, vol. 87. Springer Publishing Company, Incorporated, 1 edn. (2004). https://doi.org/10.1007/978-1-4419-8853-9, http://link.springer.com/10.1007/978-1-4419-8853-9
14. Ng, I., Ghassami, A., Zhang, K.: On the role of sparsity and DAG constraints for learning linear DAGs. Advances in Neural Information Processing Systems **33**, 17943–17954 (2020)
15. Pearl, J.: Causality. Cambridge university press (2009)
16. Peters, J., Bühlmann, P., Meinshausen, N.: Causal inference by using invariant prediction: identification and confidence intervals. Journal of the Royal Statistical Society. Series B (Statistical Methodology) pp. 947–1012 (2016)
17. Peters, J., Janzing, D., Schölkopf, B.: Elements of causal inference: foundations and learning algorithms. The MIT Press (2017)
18. Sauer, A., Geiger, A.: Counterfactual generative networks. In: International Conference on Learning Representations (ICLR) (2021)
19. Shimizu, S., Hoyer, P.O., Hyvärinen, A., Kerminen, A.: A linear non-Gaussian acyclic model for causal discovery. Journal of Machine Learning Research **7**(10) (2006)
20. Shimizu, S., Inazumi, T., Sogawa, Y., Hyvärinen, A., Kawahara, Y., Washio, T., Hoyer, P.O., Bollen, K.: DirectLiNGAM: A direct method for learning a linear non-Gaussian structural equation model. The Journal of Machine Learning Research **12**, 1225–1248 (2011)
21. Stephens, M., Balding, D.J.: Bayesian statistical methods for genetic association studies. Nature Reviews Genetics **10**(10), 681–690 (2009)
22. Yu, Y., Chen, J., Gao, T., Yu, M.: DAG-GNN: DAG structure learning with graph neural networks. In: International Conference on Machine Learning. pp. 7154–7163. PMLR (2019)
23. Zheng, X., Aragam, B., Ravikumar, P.K., Xing, E.P.: DAGs with NO TEARS: Continuous optimization for structure learning. In: Advances in Neural Information Processing Systems. vol. 31 (2018), https://proceedings.neurips.cc/paper/2018/file/e347c51419ffb23ca3fd5050202f9c3d-Paper.pdf
24. Zheng, X., Dan, C., Aragam, B., Ravikumar, P., Xing, E.: Learning sparse nonparametric DAGs. In: International Conference on Artificial Intelligence and Statistics. pp. 3414–3425. PMLR (2020)