β-th order Acyclicity Derivatives for DAG Learning

Madhumitha Shridharan
Columbia University

Garud Iyengar
Columbia University

Abstract

We consider a non-convex optimization formulation for learning the weighted adjacency matrix W of a directed acyclic graph (DAG) that uses acyclicity constraints that are functions of $|W_{ij}|^\beta$, for $\beta \in \mathbb{N}$. State-of-the-art algorithms for this problem use gradient-based Karush-Kuhn-Tucker (KKT) optimality conditions which only yield useful search directions for $\beta = 1$. Therefore, constraints with $\beta \geq 2$ have been ignored in the literature, and their empirical performance remains unknown. We introduce β-th Order Taylor Series Expansion Based Local Search (β-LS) which yields actionable descent directions for any $\beta \in \mathbb{N}$. Our empirical experiments show that 2-LS obtains solutions of higher quality than 1-LS, 3-LS and 4-LS. 2-LSopt, an optimized version of 2-LS, obtains high quality solutions significantly faster than the state of the art which uses $\beta = 1$. Moreover, 2-LSopt does not need any graph-size specific hyperparameter tuning. We prove that β-LSopt is guaranteed to converge to a Coordinate-Wise Local Stationary Point (Cst) for any $\beta \in \mathbb{N}$. If the objective function is convex, β-LSopt converges to a local minimum. To facilitate reproducibility, we provide our implementation at GitHub.

1 Introduction

Learning weighted directed acyclic graphs (DAGs) from data is a fundamental challenge in several scientific fields, including causal inference [14], biology [13], climate science [19] and economics [6]. In this paper, we propose an optimization algorithm for DAG learning that computes high quality solutions significantly faster than the state of the art, without graph-size specific hyperparameter tuning.

Let $W \in \mathbb{R}^{d\times d}$ denote a weighted real-valued adjacency matrix. For $\beta \in \mathbb{N}$, define the function $g_\beta : \mathbb{R}^{d\times d} \to \mathbb{R}^{d\times d}$ as follows:

$$[g_\beta(W)]_{ij} := |W_{ij}|^\beta, \qquad i,j \in \{1,\dots,d\}. \tag{1}$$

Define the class of acyclicity functions

$$\mathcal{H} := \Big\{ h : \mathbb{R}_+^{d\times d} \mapsto \mathbb{R}_+ \Big| \, h(A) = \sum_{p=1}^{d} c_p \, \mathbf{Tr}(A^p), \text{ where } c_p > 0, \forall p \Big\}. \tag{2}$$

Thus, for any $h \in \mathcal{H}$, $\frac{\partial h(A)}{\partial A_{ij}}$ is well defined for all (i,j) (where we interpret the partial derivative to be a one-sided derivative when $A_{ij} = 0$). We have the following result.

Theorem 1.1 (Theorem 1 in [15]). *A directed graph is acyclic if and only if its weighted adjacency matrix W satisfies $h(g_\beta(W)) = 0$ for some $h \in \mathcal{H}$, and $\beta \in \mathbb{N}$.*

A critical component of DAG Learning is the score function. While research on continuous optimization methods for DAG Learning began with the least squares score function [20], score functions in DAG Learning is an active area of research (see e.g. [3, 7, 12, 22, 11]). Hence, to accommodate a variety of score functions, we consider optimization problem [5]

$$\begin{aligned} \min_W \quad & f(W), \\ \text{s.t.} \quad & (h \circ g_\beta)(W) = 0, \end{aligned} \tag{3}$$

where score function $f : \mathbb{R}^{d\times d} \mapsto \mathbb{R}$ is differentiable, but is not required to be convex, $h \in \mathcal{H}$ and $\beta \in \mathbb{N}$. The function $(h \circ g_\beta)(W)$ denotes the composite function $h(g_\beta(W))$.

To our knowledge, the impact of the choice of the nonnegative function g_β on the solutions local search algorithms obtain is not well studied. Although the research on continuous optimization approaches for DAG learning began with $\beta = 2$ [20, 1], the use of any

Proceedings of the 28th International Conference on Artificial Intelligence and Statistics (AISTATS) 2025, Mai Khao, Thailand. PMLR: Volume 258.

function with $\beta \geq 2$ has fallen out of favor because state-of-the-art algorithms use KKT optimality conditions which neither yield actionable descent directions in local search algorithms nor characterize their output when $\beta \geq 2$. (see Section 2 for details) [15, 5]. Hence, these functions have been ignored in the literature, and their empirical performance remains unknown [15, 5]. Hence, there is a need to design new local search algorithms which yield actionable search directions for any g_β but also converge to a meaningful approximation of a critical point.

Our contributions are as follows.

(a) In Section 3, we propose *β-th Order Taylor Series Expansion Based Local Search (β-LS)* which in every iteration, finds the steepest feasible descent direction under the β-th order Taylor series approximation of the acyclicity constraint $h \circ g_\beta$ at a feasible point of (3). β-LS adopts the bi-level optimization framework introduced in [5] which iterates over permutations of nodes via node swaps until convergence, but selects good descent directions significantly more precisely. We empirically test the performance of β-LS for $\beta \in \{1, 2, 3, 4\}$. We show that, for any fixed set of hyperparameters, the quality of solutions obtained by 2-LS, is superior to using 1-LS, 3-LS and 4-LS.

(b) In Section 4, we propose 2-LSopt, an optimized version of 2-LS, that is able to compute high quality solutions significantly faster than the state of the art. Furthermore, the state of the art depends on multiple graph-size dependent hyperparameters whose recommended values are only available for a limited set of graph sizes [5]. Hence, these hyperparameters have to be retuned by the user when their graph size is not in this set. On the other hand, our algorithm contains a single graph-size independent input hyperparameter, that only depends on the score and acyclicity function. Thus, the hyperparameter needs to only be tuned *once* for given score and acyclicity functions. 2-LSopt can then be an off-the-shelf tool to obtain high quality results on graphs of any size.

(c) In Section 5, we prove that β-LSopt is guaranteed to converge to a *Coordinate-Wise Local Stationary Point (Cst)*, an approximation of a critical point. If the objective function is convex, β-LSopt converges to a local minimum.

2 The Problem with the State-of-the-Art

We discuss why KKT-informed state-of-the-art algorithms for problem (3) can only use $\beta = 1$. The KKT condition for (3) is given by

$$\nabla f(W) + \lambda \nabla (h \circ g_\beta)(W) = 0 \tag{4}$$

where Lagrange multiplier $\lambda \in \mathbb{R}$. [15] noted that KKT conditions cannot be used to identify descent directions in local search algorithms and characterize their output when $\beta = 2$. Theorem 2.1 extends this result for all $\beta \geq 2$. Its proof is in Appendix 9.1.

Theorem 2.1. *Suppose $\beta \geq 2$ and $\overline{W}$ is a feasible solution of (3). Then we have*

$$\nabla (h \circ g_\beta)(\overline{W}) = \mathbf{0}.$$

Theorem 2.1 highlights two problems with KKT conditions when $\beta \geq 2$. First, the only way KKT condition (4) can be satisfied is if $\nabla f(W) = 0$. In particular, if f is convex (e.g. least squares function), only the global minimum of f is eligible to be a KKT point; therefore, if the global minimum does not correspond to the weighted adjacency matrix of a DAG, no feasible points of (3) can be a KKT point. Hence, no algorithm can provably converge to a KKT point, rendering it a moot target. Secondly, Theorem 2.1 implies that at a feasible point W of (3), the first order approximation $\nabla(h \circ g)(W)$ does not provide any feasible descent directions, and hence cannot be used in iterative algorithms. Thus KKT-informed local search algorithms can only use $\beta = 1$.

3 β-th order Taylor series Expansion Based Local Search

Let $G = (V, E)$ denote a graph with node set $V = \{1, \ldots, d\}$. Note that G is a DAG if, and only if, it has a topological ordering, i.e. there exists an ordering $\pi : \{1, \ldots, d\} \to V$ such that if $t < s$, then $(\pi(s), \pi(t)) \notin E$. Since π is a permutation of V, a weighted graph G is a DAG if and only if there exists a permutation π such that $W_{\pi(s),\pi(t)} = 0, \forall t < s$. Let

$$\begin{aligned} f^\pi \coloneqq \min_W \quad & f(W) \\ \text{s.t.} \quad & W_{\pi(s),\pi(t)} = 0, \quad \forall\ t < s,\ s, t = 1, \ldots, d, \end{aligned} \tag{5}$$

denote the lowest value of the score function with the permutation fixed at π. Then we have the following result:

Lemma 3.1. *The problem*

$$\min_{\pi \in \Pi} \quad f^\pi, \tag{6}$$

where Π denotes the set of all permutations of the node set V, is equivalent to problem (3).

Proof: The problem (6) is equivalent to the problem:

$$\begin{aligned} \min_{\pi, W} \quad & f(W) \\ \text{s.t.} \quad & W_{\pi(s),\pi(t)} = 0, \forall t < s \end{aligned} \tag{7}$$

Consider any feasible solution (π, W) of problem (7) with objective value $f(W)$. Since $W_{\pi(s),\pi(t)} = 0, \forall t < s$, W is the weighted adjacency matrix of a DAG with topological ordering π. It follows that W is feasible for problem (3) with the same objective value $f(W)$. Similarly, consider any feasible solution W of problem (3) with objective value $f(W)$. By Theorem 1.1, W is the weighted adjacency matrix of a DAG. Hence, it must have a topological ordering π. It follows that (π, W) is feasible for problem (7) with the same objective value $f(W)$. Hence, the two problems are equivalent. ■

We use the bi-level optimization framework proposed in [5] to solve problem (6) as follows:

1. Initialize arbitrary permutation π
2. Update π to a new permutation π_{ij} such that $f^{\pi_{ij}} < f^{\pi}$ via edge addition. If edges cannot be added, update π via swaps.
3. Repeat until convergence.

We iteratively jump to better local minimizers of (3) until convergence.

3.1 Updating π via edge addition

Let W^π denote a local minimum for (5) given π. Let $G^\pi = (V, E^\pi)$ denote the DAG where $(i,j) \in E^\pi \iff W^\pi_{ij} \neq 0$. Note $\pi^{-1}(i)$ denotes the position of node i in permutation π. We want to update π by adding an edge between some node pair (i,j) with $\pi^{-1}(i) > \pi^{-1}(j)$.

Define $A^\pi := g_\beta(W^\pi)$, and let $\mathcal{Z}(\pi, W^\pi)$ denote the set

$$\left\{(i,j) : \pi^{-1}(i) > \pi^{-1}(j), \frac{\partial h}{\partial A^\pi_{ij}} = 0, [\nabla f(W^\pi)]_{ij} \neq 0\right\} \tag{8}$$

Lemma 3.2. *For every edge $(i,j) \in \mathcal{Z}(\pi, W^\pi)$, we can add (i,j) to E^π with weight $W_{ij} \propto -\nabla_{ij} f(W^\pi)$ to decrease f without adding cycles.*

Proof. Since $\pi^{-1}(i) > \pi^{-1}(j)$, we have $W^\pi_{ij} = 0$, i.e. $(i,j) \notin E^\pi$. Furthermore, since

$$0 = \frac{\partial h}{\partial A^\pi_{ij}} = \left[\sum_{p=1}^{d} p c_p (g_\beta(W^\pi))^{p-1}\right]_{ji}, \tag{9}$$

there are no walks from j to i in G^π. Since $[\nabla f(W^\pi)]_{ij} \neq 0$, we can add edge (i,j) to E^π with a weight $W_{ij} \propto -[\nabla f(W^\pi)]_{ij}$ to decrease the score function f, while ensuring the graph remains a DAG. □

Suppose $\mathcal{Z}(\pi, W^\pi) \neq \emptyset$, and let (i,j) denote the edge in the set which decreases f the most when added to E^π. Then we set $E^{\pi_{ij}} \leftarrow E^\pi \cup (i,j)$, and update π to the topological sort of the new DAG.

3.2 Updating π via swaps

Suppose $\mathcal{Z}(\pi, W^\pi) = \emptyset$. Then, for all (i,j) with $\pi^{-1}(i) > \pi^{-1}(j)$ and $\frac{\partial h}{\partial A^\pi_{ij}} = 0$, we have $\nabla_{ij} f(W^\pi) = 0$. Thus, to improve score, we restrict our search to edges (i,j) in set

$$\mathcal{S}(\pi, W^\pi) := \left\{(i,j) : \pi^{-1}(i) > \pi^{-1}(j), \frac{\partial h}{\partial A^\pi_{ij}} > 0\right\}$$

Each edge in this set will create a cycle if added to E^π. Hence, we add a score-improving edge (i,j) from this set by swapping nodes i and j in π. **Our key contribution is identifying a small subset of $\mathcal{S}(\pi, W^\pi)$ such that the best swap in the set quickly decreases f.**

First, we solve for the steepest descent direction $\delta^* \in \mathbb{R}^{d\times d}$ at W^π which is feasible (upto a tolerance) under the β-th order Taylor series approximation of $(h \circ g_\beta)$.

Definition 3.3 (β-th order Taylor Series Approximation)**.** *Let $\boldsymbol{\mu} \in \mathbb{Z}^{d\times d} \geq 0$ and $\|\boldsymbol{\mu}\|_1 = \sum_{ij} \mu_{ij}$. The β-th order Taylor series approximation $T_\beta(h \circ g_\beta)(W^\pi, \delta)$ of $(h \circ g)$ at W^π in the direction $\delta \in \mathbb{R}^{d\times d}$ is*

$$T_\beta(h \circ g_\beta)(W^\pi, \delta) := \sum_{\boldsymbol{\mu} \in M^{(\beta)}(\delta)} D^{(\boldsymbol{\mu})}(W^\pi) \Pi_{i,j} \frac{\delta_{ij}^{\mu_{ij}}}{\mu_{ij}!}$$

where

$$D^{(\boldsymbol{\mu})}(W^\pi) = \lim_{\delta \to 0^+} \frac{\partial^{\|\boldsymbol{\mu}\|_1}(h \circ g_\beta)}{\partial W_{11}^{\mu_{11}} \dots W_{dd}^{\mu_{dd}}}(W^\pi + \gamma\delta)$$

and

$$M^{(\beta)}(\delta) = \Big\{\boldsymbol{\mu} : \mathbb{Z}^{d\times d} \geq 0, \quad \|\boldsymbol{\mu}\|_1 \leq \beta, \quad \mu_{ij} = 0, \quad \forall i,j \text{ s.t. } \delta_{ij} = 0\Big\}. \tag{10}$$

Definition 3.3 is motivated by the Clarke Subdifferential for non-smooth functions [4]. See Appendix 7 for details.

Our main result in this section is Theorem 3.4, which shows that $T_\beta(h \circ g_\beta)(W^\pi, \delta)$ has a simple equivalent form. Its proof is in Appendix 9.2.

Theorem 3.4.

$$T_\beta(h \circ g_\beta)(W^\pi, \delta) = \frac{C}{\beta!} \sum_{\{i,j : W^\pi_{ij} = 0\}} \frac{\partial h}{\partial A^\pi_{ij}} |\delta_{ij}|^\beta$$

where constant $C = \Pi_{k=0}^{\beta-1}(\beta - k)$.

Then δ^* is the solution of

$$\begin{array}{ll} \min_{\delta} & \sum_{(i,j)\in\mathcal{S}(\pi,W^\pi)}[\nabla f(W^\pi)]_{ij}\delta_{ij}, \\ \text{s.t.} & \sum_{(i,j)\in\mathcal{S}(\pi,W^\pi)} \frac{\partial h}{\partial A^\pi_{ij}}|\delta_{ij}|^\beta \leq 1 \\ & \delta_{ij} = 0, \forall (i,j) \notin \mathcal{S}(\pi, W^\pi) \end{array} \tag{11}$$

For $\beta \geq 2$, we have $|\delta^*_{ij}| \propto \left(\frac{|[\nabla f(W^\pi)]_{ij}|}{\frac{\partial h}{\partial A^\pi_{ij}}}\right)^{\frac{1}{\beta-1}}, \forall (i,j) \in \mathcal{S}(\pi, W^\pi)$ [2]. Note that $x^{\frac{1}{\beta-1}}$ is increasing on $[0,\infty)$. Hence,we add edges sequentially in decreasing order of $\gamma_{ij} = \frac{|[\nabla f(W^\pi)]_{ij}|}{\frac{\partial h}{\partial A^\pi_{ij}}}$ to the candidate set. Let $\gamma_{[n]}$ denote the reversed order statistics for the set $\{\gamma_{ij} : (i,j) \in \mathcal{S}(\pi, W^\pi)\}$, i.e. $\gamma_{[1]} = \max\{\gamma_{ij} : (i,j) \in \mathcal{S}(\pi, W^\pi)\}$, and $\gamma_{[|\mathcal{S}(\pi,W^\pi)|]} = \min\{\gamma_{ij} : (i,j) \in \mathcal{S}(\pi, W^\pi)\}$, where $|\mathcal{S}(\pi, W^\pi)|$ denotes the cardinality of $\mathcal{S}(\pi, W^\pi)$. We construct candidate set

$$C_n = \{(i,j) : \gamma_{ij} \geq \gamma_{[n]}\} \tag{12}$$

where n is a hyperparameter. Note that we construct C_n in this manner for $\beta = 1$ as well, although δ^* puts all its weight on one edge [2]. In every iteration, we search over set C_n for a pair of nodes (i,j), which when swapped in permutation π, decreases the score f the most.

Finally, we repeat until there is no sufficient improvement in the score. We summarize β-th order Taylor series Expansion Based Local Search (β-LS) in Algorithm 1.

Remark 3.5. *If the set* $\mathcal{Z}(\pi, W^\pi) = \emptyset$ *in practice (e.g. if* f *is the least squares score function), then the procedure* `FreeEdges`(π, W^π) *in Algorithm 1 can be eliminated in implementation. See Appendix 8 for other strategies for efficient implementation.*

Empirical experiments in [5] demonstrate that the RANDOM-TOPO algorithm which uses $\beta = 1$ obtains higher quality solutions than any previously known DAG learning algorithm, including GOLEM [7], NOTEARS [21] and NOFEARS [16]. The candidate set of swaps in RANDOM-TOPO is parametrized by two parameters, τ and ζ. Hence, RANDOM-TOPO considers all edges, which when added to G^π, lead to a change in score function lower bounded by ζ, and a deviation from acyclicity upper bounded by τ. On the other hand, our candidate set in 1-LS is parameterized by a single parameter n. Our candidate set consists of the top n edges which lead to the highest change in score per unit deviation from acyclicity. Unlike RANDOM-TOPO, we do not remove from consideration edges which do not meet user-defined and

Algorithm 1 β-th Order Taylor Series Expansion Based Local Search (β-LS)

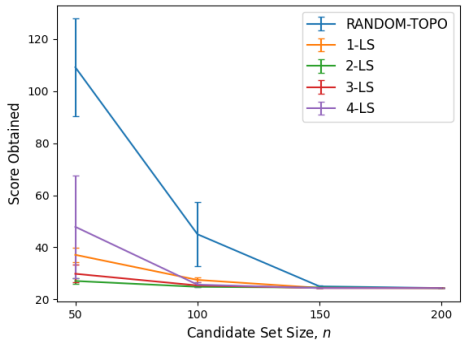
Input: (i) score function f, (ii) $\beta \in \mathbb{N}$, (iii) initial topological ordering π, (iv) candidate set size n
Output: final topological ordering π and weighted adjacency matrix estimator W^π
Function `FreeEdges`(π, W^π)**:**
  Compute set $\mathcal{Z}(\pi, W^\pi)$ from (8)
  while $\mathcal{Z}(\pi, W^\pi) \neq \emptyset$ **do**
    for $(i,j) \in \mathcal{Z}(\pi, W^\pi)$ **do**
      Add edge (i,j) to G^π to obtain DAG $G^{\pi_{ij}}$
      Compute topological ordering π_{ij} of $G^{\pi_{ij}}$
      Compute local optimal solution $W^{\pi_{ij}}$ of (5)
    Select $(\bar{i},\bar{j}) \in \arg\min_{(i,j)\in\mathcal{Z}(\pi,W^\pi)} f(W^{\pi_{ij}})$
    Update $\pi \leftarrow \pi_{\bar{i}\bar{j}}$
    Compute local optimal solution W^π of (5)
  return π, W^π

Compute local optimal solution W^π of (5)
Update $\pi, W^\pi \leftarrow$ `FreeEdges`(π, W^π)
$C \leftarrow C_n$ from (12)

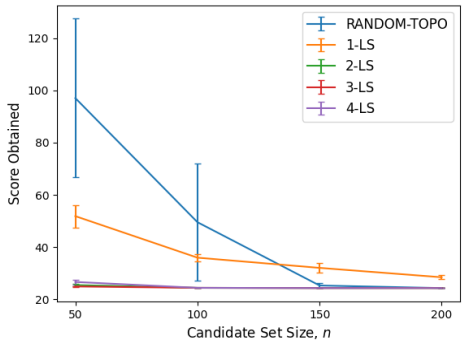
while $C \neq \emptyset$ **do**
  for $(i,j) \in C$ **do**
    Swap nodes i and j in π to obtain topological ordering π_{ij}.
    Compute local optimal solution $W^{\pi_{ij}}$ of (5)
  Select $(\bar{i},\bar{j}) \in \arg\min_{(i,j)\in C} f(W^{\pi_{ij}})$

  if $f(W^{\pi_{\bar{i}\bar{j}}}) < f(W^\pi)$ **then**
    Update $\pi \leftarrow \pi_{\bar{i}\bar{j}}$
    Compute local optimal solution W^π of (5)
    Update $\pi, W^\pi \leftarrow$ `FreeEdges`(π, W^π)
    $C \leftarrow C_n$ from (12)
  else
    return π, W^π
return π, W^π

arbitrary thresholds for score improvement and acyclicity deviation. This leads to the empirical performance gains seen in Figure 1, where we show that for any fixed candidate set size, the quality of solutions obtained by β-LS, $\beta \in \{1, 2, 3, 4\}$ is significantly superior to that obtained by RANDOM-TOPO. Furthermore, the quality of solutions obtained by 2-LS is superior to those obtained by 1-LS, 3-LS and 4-LS. We present results for the state of the art DAGMA acyclicity function [1], and the matrix polynomial acyclicity function [18]. See Appendix **??** for additional results on scale-free graphs and graphs of different degrees.



(a) DAGMA Acyclicity Function



(b) Matrix Polynomial Acyclicity Function

Figure 1: Scores obtained by β-LS, $\beta \in \{1, 2, 3, 4\}$ and RANDOM-TOPO for various candidate set sizes. Ground truth graphs are linear ER4 DAGs with equal-variance Gaussian noise. The score is least squares, and $d = 50$. Error bars denote finite sample estimates of the standard error of the mean.

4 Optimized β-th Order Taylor Series Expansion Based Local Search

Figure 1 suggests that β-LS can obtain high quality solutions even with small candidate sets of size d. Furthermore, 2-LS either matches or outperforms 1-LS, 3-LS and 4-LS for both acyclicity functions. Hence, we propose 2-LSopt, an optimized version of 2-LS (see Algorithm 2) with two key differences from 2-LS:

1. In every iteration, 2-LSopt searches over candidate sets C_d. 2-LSopt does not require the hyperparameter n.
2. While 2-LS terminates if a score decreasing swap cannot be found, 2-LSopt expands the size of the candidate set. In particular, we search over the set of candidate edges

$$C_n(\tau) = \{(i, j) : \gamma_{ij} \geq \tau \gamma_{[n]}\} \tag{13}$$

where $\tau \in (0, 1)$ is a hyperparameter selected to work well empirically across graph sizes.

Tables 1, 2, 3 and 4 report the performance of 2-LSopt and RANDOM-TOPO for two score functions: least squares, and a neural network function with two hidden layers and randomly initialized weights. For least squares, 2-LSopt attains solutions of similar quality to RANDOM-TOPO, but is significantly faster for large graphs. For the neural network, 2-LSopt is significantly faster than RANDOM-TOPO even for small graphs. Note that we limit experiments to small graphs ($d = 10$) for the neural network since it is significantly more challenging to optimize. We attribute 2-LSopt's performance to its ability to identify good descent directions precisely (see Section 3). We present results for the DAGMA acyclicity function in Tables 1 and 2 and the matrix polynomial acyclicity function in Tables 3 and 4. See Appendix 10.4 for additional results on scale-free graphs and graphs of different degrees.

5 Convergence analysis

Define the coordinate-wise function $g_{(ij)}(\delta) = f(\delta, W^{\pi}_{-(i,j)})$, i.e. set $W_{ij} = \delta$ and $W_{st} = W^{\pi}_{st}$ for all $(s, t) \neq (i, j)$. We call W^{π} a coordinate-wise local stationary point (Cst) if W^{π}_{ij} is a stationary point for $g_{(i,j)}(W)$ for all edges (i, j) such that $G = (V, E^{\pi} \cup \{(i, j)\})$ is a DAG. Since the score function $f(W)$ is differentiable, the following definition for Cst immediately follows.

Definition 5.1 (Coordinate-Wise Local Stationary Point (Cst)). *W^{π} is a Coordinate-Wise Local Stationary point (Cst) if $[\nabla f(W^{\pi})]_{ij} = 0$ for any (i, j) such that $G = (V, E^{\pi} \cup \{(i, j)\})$ is a DAG.*

Recall that from (9) we have that $\frac{\partial h}{\partial A^{\pi}_{ij}} = 0$ implies that there are no walks from j to i in G^{π}, and therefore, the edge (i, j) is a candidate for perturbation, i.e. $G = (V, E^{\pi} \cup \{(i, j)\})$ is a DAG. Lemma 5.2 is an algebraic formulation for Cst which follows by combining the definition of Cst with (9).

Algorithm 2 Optimized 2-nd Order Taylor Series Expansion Based Local Search (2-LSopt)

Input: (i) score function f, (ii) initial topological ordering π, (iii) hyperparameter $\tau \in (0, 1)$
Output: final topological ordering π and weighted adjacency matrix estimator W^π

Function `FreeEdges`(π, W^π)**:**
  Compute set $\mathcal{Z}(\pi, W^\pi)$ from (8)
  while $\mathcal{Z}(\pi, W^\pi) \neq \emptyset$ **do**
    for $(i, j) \in \mathcal{Z}(\pi, W^\pi)$ **do**
      Add edge (i, j) to G^π to obtain DAG $G^{\pi_{ij}}$
      Compute topological ordering π_{ij} of $G^{\pi_{ij}}$
      Compute local optimal solution $W^{\pi_{ij}}$ of (5)
    Select $(\bar{i}, \bar{j}) \in \arg\min_{(i,j)\in\mathcal{Z}(\pi,W^\pi)} f(W^{\pi_{ij}})$
    Update $\pi \leftarrow \pi_{\bar{i}\bar{j}}$
    Compute local optimal solution W^π of (5)
  return π, W^π

Compute local optimal solution W^π of (5)
Update $\pi, W^\pi \leftarrow$ `FreeEdges`(π, W^π)
$C \leftarrow C_d$ from (12)
`large` $\leftarrow$ FALSE
while $C \neq \emptyset$ **do**
  for $(i, j) \in C$ **do**
    Swap nodes i and j in π to obtain topological ordering π_{ij}.
    Compute local optimal solution $W^{\pi_{ij}}$ of (5)
  Select $(\bar{i}, \bar{j}) \in \arg\min_{(i,j)\in C} f(W^{\pi_{ij}})$

  if $f(W^{\pi_{\bar{i}\bar{j}}}) < f(W^\pi)$ **then**
    Update $\pi \leftarrow \pi_{\bar{i}\bar{j}}$
    Compute local optimal solution W^π of (5)
    Update $\pi, W^\pi \leftarrow$ `FreeEdges`(π, W^π)
    $C \leftarrow C_d$ from (12)
    if *large* $=$ TRUE **then**
      `large` $\leftarrow$ FALSE
  else if *large* $=$ TRUE **then**
    return π, W^π
  else
    $C \leftarrow C_d(\tau)$ from (13)
    `large` $\leftarrow$ TRUE
return π, W^π

Table 1: RANDOM-TOPO vs 2-LSopt with DAGMA acyclicity function and least squares score function. Ground truth graphs are linear ER4 DAGs with equal variance Gaussian noise. Error bars denote finite sample estimates of the standard error of the mean. False edges denotes false positive edges i.e. edges that appear in our recovered graph, but there is no edge (in either direction) in the original graph. The numbers are high because we compare methods **without** applying a post-processing threshold to their solutions.

Method	Metric	$d = 10$	$d = 50$	$d = 100$
RANDOM TOPO	Score	4.96 ±0.02	24.33 ±0.04	47.40 ±0.08
	# Edges recovered	39.2 ±0.3	193.0 ±1.0	388.2 ±0.8
	# False edges	5.5 ±0.2	1030.0 ±0.9	4558.6 ±0.9
	Runtime	**4.88** ±0.98	**72.54** **±5.10**	**990.51** ±37.26
2-LSopt	Score	4.96 ±0.02	24.33 ±0.04	47.40 ±0.08
	# Edges recovered	39.2 ±0.3	193.5 ±1.0	388.5 ±1.0
	# False edges	5.5 ±0.2	1029.6 ±0.8	4558.2 ±0.9
	Runtime	**4.51** **±0.96**	**28.36** **±0.86**	**374.74** **±12.55**

Table 2: RANDOM-TOPO vs 2-LSopt with DAGMA acyclicity function and neural network score function. Ground truth graphs are linear ER4 DAGs with equal variance Gaussian noise. Error bars denote finite sample estimates of the standard error of the mean.

Method	Metric	$d = 10$	$d = 50$	$d = 100$
RANDOM TOPO	Score	0.39 ±0.01	-	-
	Runtime	**46.40** **±5.93**	-	-
2-LSopt	Score	0.39 ±0.01	-	-
	Runtime	**28.50** ±3.87	-	-

Table 3: RANDOM-TOPO Algorithm vs 2-LSopt with the matrix polynomial acyclicity function [18] and least squares score function. Ground truth graphs are linear ER4 DAGs with equal variance Gaussian noise. Error bars denote finite sample estimates of the standard error of the mean. False edges denotes false positive edges i.e. edges that appear in our recovered graph, but there is no edge (in either direction) in the original graph. We compare methods **without** applying a post-processing threshold to their solutions.

Method	Metric	$d = 10$	$d = 50$	$d = 100$
RANDOM TOPO	Score	4.96 ±0.02	24.33 ±0.04	47.40 ±0.08
	# Edges recovered	39.2 ±0.3	193.2 ±0.9	388.0 ±1.0
	# False edges	5.5 ±0.2	1029.9 ±0.8	4558.4 ±0.9
	Runtime	**1.19 ±0.02**	**80.57 ±4.54**	**1031.73 ±43.85**
2-LSopt	Score	4.96 ±0.02	24.33 ±0.04	47.40 ±0.08
	# Edges recovered	39.2 ±0.3	193 ±1.0	387.6 ±1.1
	# False edges	5.5 ±0.2	1029.8 ±0.8	4559.3 ±1.1
	Runtime	**0.94 ±0.01**	**37.76 ±1.44**	**490.76 ±19.13**

Table 4: RANDOM-TOPO Algorithm vs 2-LSopt with the matrix polynomial acyclicity function [18] and neural network score function. Ground truth graphs are linear ER4 DAGs with equal variance Gaussian noise. We compare methods **without** applying a post-processing threshold to their solutions.

Method	Metric	$d = 10$	$d = 50$	$d = 100$
RANDOM TOPO	Score	0.39 ±0.01	-	-
	Runtime	**44.47 ±4.55**	-	-
2-LSopt	Score	0.39 ±0.01	-	-
	Runtime	**26.63 ±3.11**	-	-

Lemma 5.2. *W^π is a Cst, if and only if, for all pairs (i, j), $i \neq j$,*

$$\frac{\partial h}{\partial A^\pi_{ij}} = 0 \implies [\nabla f(W^\pi)]_{ij} = 0. \tag{14}$$

Proof:

Suppose W^π is a Cst for (3). Suppose there exist nodes s, t such that $\frac{\partial h}{\partial A^\pi_{st}} = 0$, and $|[\nabla f(W^\pi)]_{st}| > 0$. Then there are no walks from t to s, and so adding edge (s, t) to W^π retains a DAG. We arrive at a contradiction. On the other hand suppose W^π satisfies (14), but is not a Cst. Then there exists nodes s, t such that adding edge (s, t) to W^π retains a DAG (i.e. $\frac{\partial h}{\partial A^\pi_{st}} = 0$) but $[\nabla f(W^\pi)]_{st} \neq 0$. This contradicts (14). ■

We have the following convergence result for β-LSopt.

Theorem 5.3. *Suppose f is convex (resp. non-convex). Then β-LSopt converges to a local minimum (resp. Coordinate-Wise Local Stationary Point) of (3) for any $\beta \in \mathbb{N}$.*

Proof: See Appendix 9.3. ■

Note that even if f is convex, problem (3) is not a convex problem because the acyclicity constraint is not a convex constraint. Therefore, one can only hope of converging to local minima. Our algorithm strategically constructs candidate swaps to jump to better local minimizers in every iteration until convergence. Note that the current state of the art TOPO also converges to a local minimum even when f is convex; however, it is significantly less efficient compared to our method.

6 Concluding Remarks

We propose an optimization algorithm for DAG learning that computes high quality solutions significantly faster than the state of the art, without graph-size specific hyperparameter tuning. We now discuss limitations of our approach. Score functions in DAG Learning is an active area of research, with several recent attempts to design score functions in specific problem settings [10, 11, 17]. β-LSopt's hyperparameter is sensitive to the choice of score function, and needs to be retuned whenever new score functions are released. Hence, an important direction of future work is designing optimization methods with hyperparameters which work empirically well across score functions. Furthermore, we focus on the one-parameter-per-edge setting. A next step is to generalize our theory and algorithms to non-linear models with multiple parameters per edge to make them practically applicable in more scenarios. Finally, as mentioned in Chapter 1, the key challenge in the observational data setting is the

presence of unobserved confounders. Hence, extending the methods in this chapter to a setting which accounts for unobserved confounders is a critical open problem to our knowledge.

References

[1] Kevin Bello, Bryon Aragam, and Pradeep Ravikumar. Dagma: Learning dags via m-matrices and a log-determinant acyclicity characterization, 2023.

[2] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[3] Peter Bühlmann, Jonas Peters, and Jan Ernest. Cam: Causal additive models, high-dimensional order search and penalized regression. 2014.

[4] Frank H Clarke. *Optimization and nonsmooth analysis*. SIAM, 1990.

[5] Chang Deng, Kevin Bello, Bryon Aragam, and Pradeep Ravikumar. Optimizing notears objectives via topological swaps, 2023.

[6] Kevin D Hoover. Causality in economics and econometrics. *Available at SSRN 930739*, 2006.

[7] Ignavier Ng, AmirEmad Ghassami, and Kun Zhang. On the role of sparsity and dag constraints for learning linear dags. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 17943–17954. Curran Associates, Inc., 2020.

[8] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library, 2019.

[9] Kaare Brandt Petersen and Michael Syskind Pedersen. *Matrix Cookbook*. Version 20081115, 2008.

[10] Alexander Reisach, Christof Seiler, and Sebastian Weichwald. Beware of the simulated dag! causal discovery benchmarks may be easy to game. *Advances in Neural Information Processing Systems*, 34:27772–27784, 2021.

[11] Alexander Reisach, Myriam Tami, Christof Seiler, Antoine Chambaz, and Sebastian Weichwald. A scale-invariant sorting criterion to find a causal order in additive noise models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 785–807. Curran Associates, Inc., 2023.

[12] Seyed Saman Saboksayr, Gonzalo Mateos, and Mariano Tepper. Colide: Concomitant linear dag estimation, 2024.

[13] Xinpeng Shen, Sisi Ma, Prashanthi Vemuri, and Gyorgy Simon. Challenges and opportunities with causal discovery algorithms: application to alzheimer's pathophysiology. *Scientific reports*, 10(1):2975, 2020.

[14] Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, prediction, and search*. MIT press, 2001.

[15] Dennis Wei, Tian Gao, and Yue Yu. Dags with no fears: A closer look at continuous optimization for learning bayesian networks. *Advances in Neural Information Processing Systems*, 33:3895–3906, 2020.

[16] Dennis Wei, Tian Gao, and Yue Yu. Dags with no fears: A closer look at continuous optimization for learning bayesian networks. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 3895–3906. Curran Associates, Inc., 2020.

[17] Sascha Xu, Osman A Mian, Alexander Marx, and Jilles Vreeken. Inferring cause and effect in the presence of heteroscedastic noise. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 24615–24630. PMLR, 17–23 Jul 2022.

[18] Yue Yu, Jie Chen, Tian Gao, and Mo Yu. Dag-gnn: Dag structure learning with graph neural networks. In *International Conference on Machine Learning*, pages 7154–7163. PMLR, 2019.

[19] David D. Zhang, Harry F. Lee, Cong Wang, Baosheng Li, Qing Pei, Jane Zhang, and Yulun An. The causality analysis of climate change and large-scale human crisis. *Proceedings of the National Academy of Sciences*, 108(42):17296–17301, 2011.

[20] Xun Zheng, Bryon Aragam, Pradeep Ravikumar, and Eric P. Xing. Dags with no tears: Continuous optimization for structure learning, 2018.

[21] Xun Zheng, Bryon Aragam, Pradeep K Ravikumar, and Eric P Xing. Dags with no tears: Continuous optimization for structure learning. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[22] Xun Zheng, Chen Dan, Bryon Aragam, Pradeep Ravikumar, and Eric Xing. Learning sparse nonparametric dags. In Silvia Chiappa and Roberto Calandra, 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, 26–28 Aug 2020.

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. [Yes]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [No]

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]
   (b) The license information of the assets, if applicable. [Yes]
   (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]

7 Motivation for Definition 3.3

Suppose the derivative

$$\frac{\partial^{\|\boldsymbol{\mu}\|_1}(h \circ g_\beta)}{\partial W_{11}^{\mu_{11}} \dots \partial W_{dd}^{\mu_{dd}}}(W^\pi + \gamma\delta)$$

exists for every $\gamma > 0$. Then, motivated by the Clarke Sub-differential for non-smooth functions [4], we define

$$D^{(\boldsymbol{\mu})}(W^\pi) \coloneqq \lim_{\gamma \to 0^+} \frac{\partial^{\|\boldsymbol{\mu}\|_1}(h \circ g_\beta)}{\partial W_{11}^{\mu_{11}} \dots \partial W_{dd}^{\mu_{dd}}}(W^\pi + \gamma\delta)$$

The generic term in the Taylor's series expansion of $(h \circ g_\beta)(W)$ at the point W^π in the direction δ is given by $D^{(\boldsymbol{\mu})}(W^\pi)\prod_{i,j}\frac{\delta_{ij}^{\mu_{ij}}}{\mu_{ij}!}$. This term is zero when $\delta_{ij} = 0$ and $\mu_{ij} > 0$, and therefore only $\boldsymbol{\mu}$ in the set

$$M^{(\beta)}(\delta) \coloneqq \{\boldsymbol{\mu} \in \mathbb{Z}^{d\times d} \geq \mathbf{0} : \|\boldsymbol{\mu}\|_1 \leq \beta \text{ and } \mu_{ij} = 0, \forall i, j \text{ s.t. } \delta_{ij} = 0\}, \tag{15}$$

contribute to the β-th order Taylor series approximation. And, since $(h \circ g_\beta)(W^\pi) = 0$, the β^{th}-order Taylor series approximation $T_\beta(h \circ g_\beta)(W^\pi, \delta)$ of $(h \circ g_\beta)(W)$ at the point W^π in the direction δ is given by

$$T_\beta(h \circ g_\beta)(W^\pi, \delta) \coloneqq \sum_{\boldsymbol{\mu} \in M^{(\beta)}(\delta)} D^{(\boldsymbol{\mu})}(W^\pi) \prod_{i,j} \frac{\delta_{ij}^{\mu_{ij}}}{\mu_{ij}!}, \tag{16}$$

provided the limit $D^{(\boldsymbol{\mu})}$ exists for all $\boldsymbol{\mu} \in M^{(\beta)}(\delta)$. Lemma 9.5 shows that this is, indeed, the case.

8 Separability of Score Function

Lemma 8.1 shows that if the score function is separable (e.g. least squares), then several weights in W^π can be reused in $W^{\pi_{ij}}$ when nodes i and j are swapped. Hence, we do not need to compute $W^{\pi_{ij}}$ from scratch.

Lemma 8.1. *Suppose $f(W)$ is separable, i.e.,*

$$f(W) = \sum_{k=1}^{d} f_k(w_k)$$

where function $f_k : \mathbb{R}^d \to \mathbb{R}$ and $w_k \in \mathbb{R}^d$ denotes the k-th column of W. Suppose $\pi^{-1}(i) > \pi^{-1}(j)$. Then the weights in at most $\pi^{-1}(i) - \pi^{-1}(j) + 1$ columns need to be updated to obtain $W^{\pi_{ij}}$ from W^π after swapping nodes i and j.

Proof: Let $w_k(i) = W_{ik}$. Let

$$\begin{aligned} f_k^\pi = \min_{w_k} \quad & f_k(w_k) \\ \text{s.t.} \quad & w_k(\ell) = 0, \forall \pi^{-1}(\ell) > \pi^{-1}(k) \end{aligned} \tag{17}$$

For any k such that $k < \pi^{-1}(j)$ or $k > \pi^{-1}(i)$ we have:

$$\{\ell \in \{1, \dots, d\} : \pi^{-1}(\ell) > \pi^{-1}(k)\} = \left\{\ell \in \{1, \dots, d\} : \pi_{ij}^{-1}(\ell) > \pi_{ij}^{-1}(k)\right\}$$

It follows that

$$f_k^{\pi_{ij}} = f_k^\pi \tag{18}$$

Let w_k^π denote the k-th column of W^π. From (5), we have $f_k^\pi = f_k(w_k^\pi)$. Hence, from (18), we have

$$f_k(w_k^\pi) = f_k^{\pi_{ij}}$$

It follows that we can set $w_k^{\pi_{ij}} = w_k^\pi$ for all k such that $k < \pi(j)$ or $k > \pi(i)$. Hence, only the weights of W^π from columns $\pi(j)$ to $\pi(i)$ need to potentially be updated to obtain $W^{\pi_{ij}}$. ■

9 Proof of Results

9.1 Proof of Theorem 2.1

Lemma 9.1. *Suppose $\overline{W}$ is the weighted adjacency matrix of a DAG, $p \geq 1$ and $\alpha, \beta > 0$. Then we have*

$$\left(g_\alpha(\overline{W})^{p-1}\right)^\top \odot g_\beta(\overline{W}) = \mathbf{0}, \tag{19}$$

where $A \odot B$ denotes the Hadamard product of $A, B \in \mathbb{R}^{d\times d}$, and $A^0 = \boldsymbol{I}$ for any $A \in \mathbb{R}^{d\times d}$.

Proof: For $p = 1$, $\left(g_\alpha(\overline{W})^{p-1}\right)^\top \odot g_\beta(\overline{W}) = \boldsymbol{I} \odot g_\beta(\overline{W}) = \mathbf{0}$, where the last equality follows from the fact that $\overline{W}_{ii} = 0$ implies that $[g_\beta(\overline{W})]_{ii} = 0$. For the same reason, for all $p \geq 2$, $[g_\alpha(\overline{W})^{p-1}]_{ii}[g_\beta(\overline{W})]_{ii} = 0$ for all $i = 1, \ldots, d$.

Next, consider $i \neq j$. Let $G = (V, E)$ denote the DAG where $(i, j) \in E$ if, and only if, $\overline{W}_{ij} \neq 0$. We will interpret $\overline{W}$ as particular edge weights for G. Suppose

$$[g_\alpha(\overline{W})^{p-1}]_{ji}\ [g_\beta(\overline{W})]_{ij} \neq 0$$

Then both $[g_\alpha(\overline{W})^{p-1}]_{ji} \neq 0$ and $[g_\beta(\overline{W})]_{ij} \neq 0$. Since $g_\alpha(\overline{W}) \geq 0$ can be interpreted as non-negative on the arcs E, $[g_\alpha(\overline{W})^{p-1}]_{ji}$ denotes the sum of the weights of all paths from j to i with $p-1$ arcs. Thus, $[g_\alpha(\overline{W})^{p-1}]_{ji} > 0$ implies that there exists a path with $p-1$ arcs from j to i. Since $[g_\beta(\overline{W})]_{ij} \neq 0$ if, and only if, $\overline{W}_{ij} \neq 0$, we have that $(i, j) \in E$. Thus, $[g_\alpha(\overline{W})^{p-1}]_{ji}\ [g_\beta(\overline{W})]_{ij} \neq 0$ implies that there is a directed cycle in the graph G. A contradiction. ■

Lemma 9.1 immediately implies that

$$\sum_{p=1}^{m} c_p p\left(g_\alpha(\overline{W})^{p-1}\right)^\top \odot g_\beta(\overline{W}) = \mathbf{0}. \tag{20}$$

In particular, let $\alpha = \beta = 1$ and $\overline{A} = g_1(\overline{W})$. Standard matrix calculus (see, e.g.[9]) implies that

$$\frac{\partial h}{\partial \overline{A}_{ij}} = \Big[\sum_{p=1}^{d} p c_p (\overline{A})^{p-1}\Big]_{ji}$$

Then (20) implies

$$\frac{\partial h}{\partial \overline{A}} \odot \left|\overline{W}\right| = \mathbf{0} \tag{21}$$

Theorem 2.1. *Suppose $\beta \geq 2$ and $\overline{W}$ is a feasible solution of (3). Then we have*

$$\nabla(h \circ g_\beta)(\overline{W}) = \mathbf{0}.$$

Proof: When $W_{ij} \neq 0$, we have

$$\frac{\partial g_\beta}{\partial W_{ij}} = C\,|W_{ij}|^{\beta-1}$$

where $C = \beta sgn(W_{ij})$. Suppose $W_{ij} = 0$. We have

$$\begin{aligned}
\lim_{W_{ij}\to 0^+} \frac{[g_\beta(W)]_{ij} - |0|^\beta}{W_{ij} - 0} &= \lim_{W_{ij}\to 0^+} \frac{[g_\beta(W)]_{ij}}{W_{ij}} \\
&= \lim_{W_{ij}\to 0^+} \frac{W_{ij}^\beta}{W_{ij}} \\
&= \lim_{W_{ij}\to 0^+} W_{ij}^{\beta-1} \\
&= 0
\end{aligned}$$

and we have

$$\begin{aligned}\lim_{W_{ij}\to 0^-}\frac{[g_\beta(W)]_{ij}-|0|^\beta}{W_{ij}-0} &= \lim_{W_{ij}\to 0^-}\frac{[g_\beta(W)]_{ij}}{W_{ij}}\\ &= \lim_{W_{ij}\to 0^-}\frac{(-W_{ij})^\beta}{W_{ij}}\\ &= \lim_{W_{ij}\to 0^-}(-1)^\beta W_{ij}^{\beta-1}\\ &= 0\end{aligned}$$

We hence have

$$\lim_{W_{ij}\to 0}\frac{[g_\beta(W)]_{ij}-|0|^\beta}{W_{ij}-0}=0$$

It follows that the derivative

$$\frac{\partial g_\beta}{\partial W_{ij}}=C\,|W_{ij}|^{\beta-1}$$

everywhere. Let $\overline{A}=g_\beta(\overline{W})$. We have

$$\begin{aligned}\left[\nabla(h\circ g_\beta)\left(\overline{W}\right)\right]_{ij} &= \frac{\partial h}{\partial \overline{A}_{ij}}\frac{\partial g_\beta}{\partial \overline{W}_{ij}}\\ &= \overline{C}\frac{\partial h}{\partial \overline{A}_{ij}}\left|\overline{W}_{ij}\right|^{\beta-1}\end{aligned}\tag{22}$$

where $\overline{C}=\beta sgn(\overline{W}_{ij})$. From Theorem 1.1, $\overline{W}$ is the weighted adjacency matrix of a DAG. Hence from (21), we have

$$\frac{\partial h}{\partial \overline{A}_{ij}}\,|\overline{W}_{ij}|=0\tag{23}$$

From (22) and (23), it follows that

$$\nabla(h\circ g_\beta)(\overline{W})=\mathbf{0}$$

■

9.2 Proof of Theorem 3.4

Let $G^C=(V,E^C)$ denote the complete DAG over all d nodes i.e. for every pair of nodes $i,j\in V$, we have $(i,j)\in E^C$. A walk ω from node i to node j is a sequence of nodes $n_0,\ldots,n_\ell\in V$ such that $n_0=i$ and $n_\ell=j$. The sequence of edges traversed by the walk ω is given by $(n_0,n_1),\ldots,(n_{\ell-1},n_\ell)\in E^C$. Note that each edge (n_k,n_{k+1}) in walk ω can be traversed multiple times. Hence, let $t_{ij}^{(\omega)}$ denote the number of times edge (i,j) is traversed in walk ω. That is,

$$t_{ij}^{(\omega)}=\left|\left\{(n_k,n_{k+1}):n_k=i,n_{k+1}=j\right\}\right|$$

Let $\mathcal{W}_{ij}^{(\ell)}$ denote the set of all walks from i to j of length ℓ. That is,

$$\mathcal{W}_{ij}^{(\ell)}:=\left\{\omega:n_0=i,n_\ell=j,\sum_{i,j}t_{ij}^{(\omega)}=\ell\right\}$$

If $i = j$, then $\mathcal{W}_{ij}^{(\ell)}$ is the set of cycles of length ℓ that contain the node i (or j).

Fix $p \in \{1, \ldots, d\}$, and let $h^{(p)}(A) \coloneqq \mathbf{Tr}(A^p)$. We have:

$$h^{(p)}(A) = \sum_{k=1}^{d} \sum_{\omega \in \mathcal{W}_{kk}^{(p)}} L^{(\omega)}(A) \tag{24}$$

where

$$L^{(\omega)}(A) \coloneqq \prod_{i,j} (A_{ij})^{t_{ij}^{(\omega)}} \tag{25}$$

For any node k and cycle $\omega \in \mathcal{W}_{kk}^{(p)}$, Lemma 9.2 shows that

$$(L^{(\omega)} \circ g_\beta)(W^\pi) = 0$$

Lemma 9.2. *For any node k and cycle $\omega \in \mathcal{W}_{kk}^{(p)}$, we have*

$$(L^{(\omega)} \circ g_\beta)(W^\pi) = 0$$

Proof: For any node k and cycle $\omega \in \mathcal{W}_{kk}^{(p)}$, from (25), we have:

$$\begin{aligned} (L^{(\omega)} \circ g_\beta)(W^\pi) &= \prod_{i,j} \left(|W_{ij}^\pi|^\beta\right)^{t_{ij}^{(\omega)}} \\ &= \prod_{i,j} |W_{ij}^\pi|^{\beta t_{ij}^{(\omega)}} \end{aligned} \tag{26}$$

We claim

$$\prod_{i,j} |W_{ij}^\pi|^{\beta t_{ij}^{(\omega)}} = 0 \tag{27}$$

Suppose not. Then for every (i, j) such that $t_{ij}^{(\omega)} \geq 1$, we have

$$|W_{ij}^\pi| \neq 0 \tag{28}$$

Hence, for every edge (i, j) which is traversed at least once by the cycle ω, we have $|W_{ij}^\pi| \neq 0$. Since $\omega \in \mathcal{W}_{kk}^{(p)}$, (28) implies that node k is part of a directed cycle in G^π, a contradiction. Hence, (27) must hold. By (26), we have

$$(L^{(\omega)} \circ g_\beta)(W^\pi) = 0$$

■

Lemma 9.3. *Fix node k and cycle $\omega \in \mathcal{W}_{kk}^{(p)}$. Let $\delta \in \mathbb{R}^{d \times d}$ and $\boldsymbol{\mu} \in M^{(\beta)}(\delta)$ such that $\mu_{ij} < \beta, \forall i, j$. We have*

$$\lim_{\gamma \to 0^+} \frac{\partial^{\|\boldsymbol{\mu}\|_1} \left(L^{(\omega)} \circ g_\beta\right)}{\partial W_{11}^{\mu_{11}} \ldots \partial W_{dd}^{\mu_{dd}}} (W^\pi + \gamma\delta) = 0$$

Proof:

For any node k and cycle $\omega \in \mathcal{W}_{kk}^{(p)}$, we have:

$$(L^{(\omega)} \circ g_\beta)(W) = \prod_{i,j} \left(|W_{ij}|^\beta\right)^{t_{ij}^{(\omega)}} \tag{29}$$

Since $\boldsymbol{\mu} \in M^{(\beta)}(\delta)$, for every i, j we have

$$\mu_{ij} = 0, \forall i, j \text{ s.t. } \delta_{ij} = 0 \tag{30}$$

For any i, j such that $\mu_{ij} > 0$ and $W^{\pi}_{ij} = 0$, we have from (30)

$$W^{\pi}_{ij} + \gamma\delta_{ij} \neq 0$$

for all $\gamma > 0$. For any i, j such that $\mu_{ij} > 0$ and $W^{\pi}_{ij} \neq 0$, we have from (30)

$$W^{\pi}_{ij} + \gamma\delta_{ij} \neq 0$$

for all $\gamma \in \left(0, \left|\frac{W^{\pi}_{ij}}{\delta_{ij}}\right|\right)$. From (29) it follows that the derivative

$$\frac{\partial^{\|\boldsymbol{\mu}\|_1}(L^{(\omega)} \circ g_\beta)}{\partial W_{11}^{\mu_{11}} \dots \partial W_{dd}^{\mu_{dd}}}(W^{\pi} + \gamma\delta)$$

exists for all $\gamma > 0$ if $W^{\pi}_{ij} = 0$ for all i, j such that $\mu_{ij} > 0$. If there exists i, j such that $\mu_{ij} > 0$ and $W^{\pi}_{ij} \neq 0$, then the derivative exists for all $\gamma \in \left(0, \min_{i,j:\mu_{ij}>0, W^{\pi}_{ij}\neq 0} \left|\frac{W^{\pi}_{ij}}{\delta_{ij}}\right|\right)$. If the derivative exists, from (29), we have

$$\frac{\partial^{\|\boldsymbol{\mu}\|_1}(L^{(\omega)} \circ g_\beta)}{\partial W_{11}^{\mu_{11}} \dots \partial W_{dd}^{\mu_{dd}}}(W^{\pi} + \gamma\delta) = C \left(\prod_{i,j} (sgn(W^{\pi}_{ij} + \gamma\delta_{ij}))^{\mu_{ij}}\right) \left(\prod_{i,j} |W^{\pi}_{ij} + \gamma\delta_{ij}|^{\beta t^{(\omega)}_{ij} - \mu_{ij}}\right)$$

where constant $C = \prod_{i,j} \prod_{m=0}^{\mu_{ij}-1} (\beta t^{(\omega)}_{ij} - m)$. Hence, we have

$$\begin{aligned} \lim_{\gamma\to 0^+} \frac{\partial^{\|\boldsymbol{\mu}\|_1}(L^{(\omega)} \circ g_\beta)}{\partial W_{11}^{\mu_{11}} \dots \partial W_{dd}^{\mu_{dd}}}(W^{\pi} + \gamma\delta) &= C \left(\lim_{\gamma\to 0^+} \prod_{i,j} \left(sgn(W^{\pi}_{ij} + \gamma\delta_{ij})\right)^{\mu_{ij}}\right) \left(\lim_{\gamma\to 0^+} \prod_{i,j} |W^{\pi}_{ij} + \gamma\delta_{ij}|^{\beta t^{(\omega)}_{ij} - \mu_{ij}}\right) \\ &= C \left(\lim_{\gamma\to 0^+} \prod_{i,j} \left(sgn(W^{\pi}_{ij} + \gamma\delta_{ij})\right)^{\mu_{ij}}\right) \left(\prod_{i,j} |W^{\pi}_{ij}|^{\beta t^{(\omega)}_{ij} - \mu_{ij}}\right) \end{aligned} \tag{31}$$

We have

$$\lim_{\gamma\to 0^+} \left(sgn(W^{\pi}_{ij} + \gamma\delta_{ij})\right)^{\mu_{ij}} = \left\{ \begin{array}{ll} (sgn(\delta_{ij}))^{\mu_{ij}}, & \text{if } W^{\pi}_{ij} = 0 \\ \left(sgn(W^{\pi}_{ij})\right)^{\mu_{ij}}, & \text{otherwise} \end{array} \right\}$$

We now claim

$$\prod_{i,j} |W^{\pi}_{ij}|^{\beta t^{(\omega)}_{ij} - \mu_{ij}} = 0 \tag{32}$$

By Lemma 9.2, we have

$$(L^{(\omega)} \circ g_\beta)(W^{\pi}) = 0$$

From (29) this implies

$$\prod_{i,j} |W^{\pi}_{ij}|^{\beta t^{(\omega)}_{ij}} = 0$$

Hence, there exist nodes i', j' such that $|W^{\pi}_{i'j'}|^{\beta t^{(\omega)}_{i'j'}} = 0$ i.e. $\beta t^{(\omega)}_{i'j'} > 0$ and $W^{\pi}_{i'j'} = 0$. Since $\beta t^{\omega}_{i'j'} - \mu_{i'j'} \geq \beta - \mu_{i'j'} > 0$, it follows that

$$|W^{\pi}_{i'j'}|^{\beta t^{(\omega)}_{i'j'} - \mu_{i'j'}} = 0$$

and (32) is true. By (31), we have

$$\lim_{\gamma \to 0^+} \frac{\partial^{\|\boldsymbol{\mu}\|_1} (L^{(\omega)} \circ g_\beta)}{\partial W_{11}^{\mu_{11}} \dots \partial W_{dd}^{\mu_{dd}}} (W^{\pi} + \gamma \delta) = 0$$

■

Fix node k and cycle $\omega \in \mathcal{W}^{(p)}_{kk}$. Let $\delta \in \mathbb{R}^{d \times d}$. Suppose $\boldsymbol{\mu} \in M^{(\beta)}(\delta)$ and $\boldsymbol{\mu} = \beta J^{ij}$ for some i, j. From (15), we have $\delta_{ij} \neq 0$. We have

$$\lim_{\gamma \to 0^+} \frac{\partial^{\|\boldsymbol{\mu}\|_1} (L^{(\omega)} \circ g_\beta)}{\partial W_{11}^{\mu_{11}} \dots \partial W_{dd}^{\mu_{dd}}} (W^{\pi} + \gamma \delta) = \lim_{\gamma \to 0^+} \frac{\partial^{\beta} (L^{(\omega)} \circ g_\beta)}{\partial W_{ij}^{\beta}} (W^{\pi} + \gamma \delta) \tag{33}$$

From (25), we have:

$$(L^{(\omega)} \circ g_\beta)(W) = \prod_{i',j'} \left(|W_{i'j'}|^{\beta} \right)^{t^{(\omega)}_{i'j'}} \tag{34}$$

Suppose $t^{(\omega)}_{ij} = 0$. Then we have

$$\frac{\partial^{\beta} (L^{(\omega)} \circ g)}{\partial W_{ij}^{\beta}} = 0$$

and so we have

$$\lim_{\gamma \to 0^+} \frac{\partial^{\beta} (L^{(\omega)} \circ g_\beta)}{\partial W_{ij}^{\beta}} (W^{\pi} + \gamma \delta) = 0 \tag{35}$$

Suppose $t^{(\omega)}_{ij} \geq 1$. That is, the edge (i, j) is traversed at least once in cycle ω, and the cycle ω contains the nodes i and j. Without loss of generality, suppose the sequence of nodes in cycle ω is given by $n_0, \dots, n_{\ell-1}, n_\ell$ where $n_0 = n_\ell = j$ and $n_{\ell-1} = i$. Let $\omega \setminus (i, j)$ denote the walk from nodes j to i obtained by removing edge (i, j) from cycle ω. That is, the sequence of nodes in walk $\omega \setminus (i, j)$ is given by $n_0, \dots, n_{\ell-1}$. Lemma 9.4 shows that (33) is not necessarily zero.

Lemma 9.4. *Fix node k and cycle $\omega \in \mathcal{W}^{(p)}_{kk}$. Let $\delta \in \mathbb{R}^{d \times d}$ such that $\delta_{ij} \neq 0$. Suppose $t^{(\omega)}_{ij} \geq 1$. Then we have*

$$\lim_{\gamma \to 0^+} \frac{\partial^{\beta} (L^{(\omega)} \circ g_\beta)}{\partial W_{ij}^{\beta}} (W^{\pi} + \gamma \delta) = \left\{ \begin{array}{ll} C \left[\left(L^{(\omega \setminus (i,j))} \circ g_\beta \right) (W^{\pi}) \right] (sgn(\delta_{ij}))^{\beta}, & \text{if } W^{\pi}_{ij} = 0 \\ 0, & \text{otherwise} \end{array} \right\}$$

where $C = \prod_{k=0}^{\beta-1} (\beta - k)$ is a constant.

Proof: Suppose $W^{\pi}_{ij} = 0$. We have

$$W^{\pi}_{ij} + \gamma \delta_{ij} \neq 0$$

for all $\gamma > 0$. Suppose $W_{ij}^{\pi} \neq 0$. Then we have

$$W_{ij}^{\pi} + \gamma\delta_{ij} \neq 0$$

for all $0 < \gamma < \left|\frac{W_{ij}^{\pi}}{\delta_{ij}}\right|$. From (34), it follows that the derivative

$$\frac{\partial^{\beta}(L^{(\omega)} \circ g_{\beta})}{\partial W_{ij}^{\beta}}(W^{\pi} + \gamma\delta)$$

exists for all $\gamma > 0$ if $W_{ij}^{\pi} = 0$ and exists for all $0 < \gamma < \left|\frac{W_{ij}^{\pi}}{\delta_{ij}}\right|$ if $W_{ij}^{\pi} \neq 0$. If the derivative exists, from (34) we have

$$\begin{aligned}
\frac{\partial^{\beta}(L^{(\omega)} \circ g_{\beta})}{\partial W_{ij}^{\beta}} &= \left(\prod_{(i',j')\neq(i,j)} |W_{i'j'}|^{\beta t_{i'j'}^{(\omega)}}\right)\left(C\left(sgn(W_{ij})\right)^{\beta}|W_{ij}|^{\beta t_{ij}^{(\omega)}-\beta}\right) \\
&= C\left(sgn(W_{ij})\right)^{\beta}\left(\prod_{(i',j')\neq(i,j)} |W_{i'j'}|^{\beta t_{i'j'}^{(\omega)}}\right)|W_{ij}|^{\beta t_{ij}^{(\omega)}-\beta} \\
&= C\left(sgn(W_{ij})\right)^{\beta}\left(\prod_{(i',j')\neq(i,j)} |W_{ij}|^{\beta t_{ij}^{(\omega)}}\right)\left(|W_{ij}|^{\beta}\right)^{t_{ij}^{(\omega)}-1} \\
&= C\left(sgn(W_{ij})\right)^{\beta}\left(L^{(\omega\setminus(i,j))} \circ g_{\beta}\right)(W)
\end{aligned}$$

where constant $C = \prod_{k=0}^{\beta-1}(\beta - k)$. Hence, it follows that

$$\begin{aligned}
\lim_{\gamma\to 0^+}\frac{\partial^{\beta}\left(L^{(\omega)} \circ g\right)}{\partial W_{ij}^{\beta}}(W^{\pi} + \gamma\delta) &= C\left(\lim_{\gamma\to 0^+}\left(sgn(W_{ij}^{\pi} + \gamma\delta_{ij})\right)^{\beta}\right)\left(\lim_{\gamma\to 0^+}\left(L^{(\omega\setminus(i,j))} \circ g_{\beta}\right)(W^{\pi} + \delta\gamma)\right) \\
&= C\left(\lim_{\gamma\to 0^+}\left(sgn(W_{ij}^{\pi} + \gamma\delta_{ij})\right)^{\beta}\right)\left(L^{(\omega\setminus(i,j))} \circ g_{\beta}\right)(W^{\pi}) \qquad (36)
\end{aligned}$$

We have

$$\lim_{\gamma\to 0^+}\left(sgn(W_{ij}^{\pi} + \gamma\delta_{ij})\right)^{\beta} = \left\{\begin{array}{ll} (sgn(\delta_{ij}))^{\beta}, & \text{if } W_{ij}^{\pi} = 0 \\ \left(sgn(W_{ij}^{\pi})\right)^{\beta}, & \text{otherwise} \end{array}\right\}$$

Hence, from (36), we have

$$\lim_{\gamma\to 0^+}\frac{\partial^{\beta}(L^{(\omega)} \circ g)}{\partial W_{ij}^{\beta}}(W^{\pi} + \gamma\delta) = \left\{\begin{array}{ll} C(sgn(\delta_{ij}))^{\beta}\left[\left(L^{(\omega\setminus(i,j))} \circ g_{\beta}\right)(W^{\pi})\right], & \text{if } W_{ij}^{\pi} = 0 \\ C(sgn(W_{ij}^{\pi}))^{\beta}\left[\left(L^{(\omega\setminus(i,j))} \circ g_{\beta}\right)(W^{\pi})\right], & \text{otherwise} \end{array}\right\}$$

Suppose $W_{ij}^{\pi} \neq 0$. We claim

$$(sgn(W_{ij}^{\pi}))^{\beta}\left[\left(L^{(\omega\setminus(i,j))} \circ g_{\beta}\right)(W^{\pi})\right] = 0 \qquad (37)$$

Let $G = (V, E)$ denote the DAG where $(i, j) \in E$ if, and only if, $W_{ij}^{\pi} \neq 0$. We will interpret W^{π} as particular edge weights for G. Since $W_{ij}^{\pi} \neq 0$ and G is a DAG, we must have $i \neq j$. Suppose

$$(sgn(W_{ij}^{\pi}))^{\beta}\left[\left(L^{(\omega\setminus(i,j))}\circ g_{\beta}\right)(W^{\pi})\right]\neq 0$$

Then both $\left[\left(L^{(\omega\setminus(i,j))}\circ g_{\beta}\right)(W^{\pi})\right]\neq 0$ and $(sgn(W_{ij}^{\pi}))^{\beta}\neq 0$. $\left[\left(L^{(\omega\setminus(i,j))}\circ g_{\beta}\right)(W^{\pi})\right]\neq 0$ implies that there exists a walk with $p-1$ edges from j to i. Since $(sgn(W_{ij}^{\pi}))^{\beta}\neq 0$ if, and only if, $W_{ij}^{\pi}\neq 0$, we have that $(i,j)\in E$. Thus, $(sgn(W_{ij}^{\pi}))^{\beta}\left[\left(L^{(\omega\setminus(i,j))}\circ g_{\beta}\right)(W^{\pi})\right]\neq 0$ implies that there is a directed cycle in the graph G. A contradiction. Thus, (37) holds. It follows that

$$\lim_{\gamma\to 0^+}\frac{\partial^{\beta}(L^{(\omega)}\circ g)}{\partial W_{ij}^{\beta}}(W^{\pi}+\gamma\delta)=\left\{\begin{array}{ll} C(sgn(\delta_{ij}))^{\beta}\left[\left(L^{(\omega\setminus(i,j))}\circ g_{\beta}\right)(W^{\pi})\right], & \text{if } W_{ij}^{\pi}=0\\ 0, & \text{otherwise}\end{array}\right\}$$

■

From (35) and Lemma 9.4, we have

$$\lim_{\gamma\to 0^+}\frac{\partial^{\beta}(L^{(\omega)}\circ g_{\beta})}{\partial W_{ij}^{\beta}}(W^{\pi}+\gamma\delta)=\left\{\begin{array}{ll} C\left[\left(L^{(\omega\setminus(i,j))}\circ g_{\beta}\right)(W^{\pi})\right](sgn(\delta_{ij}))^{\beta}, & \text{if } t_{ij}^{(\omega)}\geq 1 \text{ and } W_{ij}^{\pi}=0\\ 0, & \text{otherwise}\end{array}\right\} \tag{38}$$

Lemma 9.5. *Let $A^{\pi}=g_{\beta}(W^{\pi})$. Suppose $\delta\in\mathbb{R}^{d\times d}$ and $\boldsymbol{\mu}\in M^{(\beta)}(\delta)$. Then we have that*

$$D^{(\boldsymbol{\mu})}(W^{\pi})=\begin{cases} C(sgn(\delta_{ij}))^{\beta}\left(\frac{\partial h}{\partial A_{ij}^{\pi}}\right), & \boldsymbol{\mu}=\beta\boldsymbol{e}_i\boldsymbol{e}_j^{\top} \text{ and } W_{ij}^{\pi}=0, i,j,\in\{1,\ldots,d\},\\ 0 & \text{otherwise}\end{cases}$$

where $\boldsymbol{e}_i$, $i=1,\ldots,d$, denotes the i-th basis vector in $\mathbb{R}^d$, the constant $C=\prod_{k=0}^{\beta-1}(\beta-k)$, and $sgn(x)=1$ for $x>0$, -1 for $x<0$, and 0 for $x=0$.

Proof:

Suppose $\mu_{ij}<\beta,\forall i,j$. By Lemma 9.3 and linearity (see (24)), we have

$$\lim_{\gamma\to 0^+}\frac{\partial^{\|\boldsymbol{\mu}\|_1}(h^{(p)}\circ g)}{\partial W_{11}^{\mu_{11}}\ldots\partial W_{dd}^{\mu_{dd}}}(W^{\pi}+\gamma\delta)=0$$

Applying linearity again (see (2)), we have

$$D^{(\boldsymbol{\mu})}(W^{\pi})=0$$

Suppose $\boldsymbol{\mu}=\beta J^{ij}$ for some i,j. By linearity (see (24)), we have

$$\lim_{\gamma\to 0^+}\frac{\partial^{\beta}(h^{(p)}\circ g_{\beta})}{\partial W_{ij}^{\beta}}(W^{\pi}+\gamma\delta)=\sum_{k=1}^{d}\sum_{\omega\in\mathcal{W}_{kk}^{(p)}}\lim_{\gamma\to 0^+}\frac{\partial^{\beta}(L^{(\omega)}\circ g_{\beta})}{\partial W_{ij}^{\beta}}(W^{\pi}+\gamma\delta) \tag{39}$$

From (38) have

$$\lim_{\gamma\to 0^+}\frac{\partial^{\beta}(L^{(\omega)}\circ g_{\beta})}{\partial W_{ij}^{\beta}}(W^{\pi}+\gamma\delta)=\left\{\begin{array}{ll} C\left[\left(L^{(\omega\setminus(i,j))}\circ g_{\beta}\right)(W^{\pi})\right](sgn(\delta_{ij}))^{\beta}, & \text{if } t_{ij}^{(\omega)}\geq 1 \text{ and } W_{ij}^{\pi}=0\\ 0, & \text{otherwise}\end{array}\right\}$$

Hence from (39), we have

$$\lim_{\gamma\to 0^+}\frac{\partial^{\beta}(h^{(p)}\circ g_{\beta})}{\partial W_{ij}^{\beta}}(W^{\pi}+\gamma\delta)=\left\{\begin{array}{ll} C(sgn(\delta_{ij}))^{\beta}\sum_{k=1}^{d}\sum_{\omega\in\mathcal{W}_{kk}^{(p)}:t_{ij}^{(\omega)}\geq 1}\left(L^{(\omega\setminus(i,j))}\circ g_{\beta}\right)(W^{\pi}), & \text{if } W_{ij}^{\pi}=0\\ 0, & \text{otherwise}\end{array}\right\}$$

Note we have

$$\sum_{k=1}^{d} \sum_{\omega \in \mathcal{W}_{kk}^{(p)} : t_{ij}^{(\omega)} \geq 1} \left(L^{(\omega \setminus (i,j))} \circ g_\beta \right) = \sum_{\omega \in \mathcal{W}_{ji}^{(p-1)}} \left(L^{(\omega)} \circ g_\beta \right) (W^\pi)$$

Hence,

$$\lim_{\gamma \to 0^+} \frac{\partial^\beta (h^{(p)} \circ g_\beta)}{\partial W_{ij}^\beta} (W^\pi + \gamma \delta) = \left\{ \begin{array}{ll} C(sgn(\delta_{ij}))^\beta \sum_{\omega \in \mathcal{W}_{ji}^{(p-1)}} \left(L^{(\omega)} \circ g_\beta \right) (W^\pi), & \text{if } W_{ij}^\pi = 0 \\ 0, & \text{otherwise} \end{array} \right\}$$

Since $[g_\beta(W^\pi)^{p-1}]_{ji}$ denotes the sum of the weights of all walks from j to i of length $p-1$, we have

$$[g_\beta(W^\pi)^{p-1}]_{ji} = \sum_{\omega \in \mathcal{W}_{ji}^{(p-1)}} \left(L^{(\omega)} \circ g_\beta \right) (W^\pi)$$

It follows that

$$\lim_{\gamma \to 0^+} \frac{\partial^\beta (h^{(p)} \circ g_\beta)}{\partial W_{ij}^\beta} (W^\pi + \gamma \delta) = \left\{ \begin{array}{ll} C(sgn(\delta_{ij}))^\beta [g_\beta(W^\pi)^{p-1}]_{ji}, & \text{if } W_{ij}^\pi = 0 \\ 0, & \text{otherwise} \end{array} \right\}$$

Applying linearity again (see (2)), we have

$$\begin{aligned} D^{(\boldsymbol{\mu})}(W^\pi) &= \left\{ \begin{array}{ll} C(sgn(\delta_{ij}))^\beta \left(\sum_{p=1}^{d} c_p p \left[g_\beta(W^\pi)^{p-1} \right]_{ji} \right), & \text{if } W_{ij}^\pi = 0 \\ 0, & \text{otherwise} \end{array} \right\} \\ &= \left\{ \begin{array}{ll} C(sgn(\delta_{ij}))^\beta \left(\frac{\partial h}{\partial A_{ij}^\pi} \right), & \text{if } W_{ij}^\pi = 0 \\ 0, & \text{otherwise} \end{array} \right\} \end{aligned} \tag{40}$$

where (40) follows from standard matrix calculus [9]. ■

We now have all prerequisite results to prove our main result of Section 3, Theorem 3.4.

Theorem 3.4.

$$T_\beta(h \circ g_\beta)(W^\pi, \delta) = \frac{C}{\beta!} \sum_{\{i,j : W_{ij}^\pi = 0\}} \frac{\partial h}{\partial A_{ij}^\pi} |\delta_{ij}|^\beta$$

where constant $C = \Pi_{k=0}^{\beta-1}(\beta - k)$.

Proof. Lemma 9.5 implies that

$$T_\beta(h \circ g_\beta)(W^\pi, \delta) = \frac{C}{\beta!} \sum_{\{i,j : \delta_{ij} \neq 0, W_{ij}^\pi = 0\}} \frac{\partial h}{\partial A_{ij}^\pi} sgn(\delta_{ij})^\beta \delta_{ij}^\beta = \frac{C}{\beta!} \sum_{\{i,j : W_{ij}^\pi = 0\}} \frac{\partial h}{\partial A_{ij}^\pi} |\delta_{ij}|^\beta$$

□

9.3 Proof of Theorem 5.3

Lemma 9.6. *Suppose* f *is convex. Let* $A^\pi = g_\beta(W^\pi)$. *Then if*

$$\frac{\partial h}{\partial A_{ij}^\pi} = 0 \implies [\nabla f(W^\pi)]_{ij} = 0, \forall i, j, i \neq j \tag{41}$$

then W^π *is a local minimum for* (3).

Proof: Let $\mathcal{P} := \{(i,j) : \frac{\partial h}{\partial A^{\pi}_{ij}} > 0\}$. Consider the problem

$$\begin{array}{ll} \min_W & f(W), \\ \text{s.t.} & W_{ij} = 0, (i,j) \in \mathcal{P}, \end{array} \tag{42}$$

Since f is convex, the sufficient conditions for optimality in problem (42) are:

$$[\nabla f(W)]_{ij} = 0, (i,j) \notin \mathcal{P} \tag{43}$$
$$W_{ij} = 0, (i,j) \in \mathcal{P} \tag{44}$$

We now show that W^{π} satisfies sufficient conditions (43) and (44). From Theorem 1.1, W^{π} is the weighted adjacency matrix of a DAG. Hence (21) implies that

$$\frac{\partial h}{\partial A^{\pi}_{ij}} \ |W^{\pi}_{ij}| = 0, \forall i,j \tag{45}$$

It follows that W^{π} satisfies (44). By (41), W^{π} satisfies (43). Hence, W^{π} is optimal for (42).

Since $\frac{\partial h}{\partial A_{ij}}(g_\beta(W))$ is a continuous function of W, there exists a sufficiently small $\epsilon > 0$ such that, for all $W \in \mathbb{R}^{d \times d}$ such that $\|W - W^{\pi}\|_F < \epsilon$, we have

$$\frac{\partial h}{\partial A^{\pi}_{ij}} > 0 \implies \frac{\partial h}{\partial A_{ij}}(g_\beta(W)) > 0, \forall i,j \tag{46}$$

Let $\widehat{W}$ denote a feasible solution for (3) which satisfies $\|W^{\pi} - \widehat{W}\|_F < \epsilon$. From Theorem 1.1, $\widehat{W}$ is the weighted adjacency matrix of a DAG. Hence (21) implies that

$$\frac{\partial h}{\partial A_{ij}}(g_\beta(\widehat{W})) \ |\widehat{W}_{ij}| = 0, \forall i,j \tag{47}$$

From (46), we have

$$\frac{\partial h}{\partial A_{ij}}(g_\beta(\widehat{W})) > 0, \forall (i,j) \in \mathcal{P}$$

Hence from (47), we have

$$\widehat{W}_{ij} = 0, \forall (i,j) \in \mathcal{P}$$

It follows that $\widehat{W}$ satisfies (44). Hence $\widehat{W}$ is a feasible solution for problem (42). But since W^{π} is the minimizer of problem (42), we must have $f(W^{\pi}) \leq f(\widehat{W})$. Hence, W^{π} is a local minimum for (3). ■

Theorem 5.3. *Suppose f is convex (resp. non-convex). Then β-LSopt converges to a local minimum (resp. Coordinate-Wise Local Stationary Point) of (3) for any $\beta \in \mathbb{N}$.*

Proof: Let the topological sort and weighted adjacency matrix estimator output by β-LSopt be π and W^{π} respectively. Since W^{π} is a local optimal solution of (5), it is feasible for (3) . Let $A^{\pi} = g_\beta(W^{\pi})$. We claim that the set $\mathcal{Z}(\pi, W^{\pi})$ defined in (8) is empty at termination. Suppose $\mathcal{Z}(\pi, W^{\pi}) \neq \emptyset$. That is, there exists edge (i,j) such that $\pi(i) > \pi(j)$, $\left|[\nabla f(W^{\pi})]_{ij}\right| > 0$ and $\frac{\partial h}{\partial A^{\pi}_{ij}} = 0$. Let $G^{\pi} = (V, E)$ denote the DAG where $(i,j) \in E$ if, and only if, $W^{\pi}_{ij} \neq 0$. Since $\pi(i) > \pi(j)$, from (5), we have $W^{\pi}_{ij} = 0$ i.e. there is no edge (i,j) in G^{π}. Since $\left|[\nabla f(W^{\pi})]_{ij}\right| > 0$, we can add edge (i,j) to G^{π} to decrease the score function f. Furthermore, from (9), we have

$$\frac{\partial h}{\partial A^{\pi}_{ij}} = \Big[\sum_{p=1}^{d} p c_p (g_\beta(W^{\pi}))^{p-1}\Big]_{ji}$$

Since $\frac{\partial h}{\partial A^{\pi}_{ij}} = 0$, we have

$$\Big[\sum_{p=1}^{d} p c_p (g_\beta(W^\pi))^{p-1}\Big]_{ji} = 0 \tag{48}$$

(48) implies that there are no walks from j to i in G^π. Hence we can add edge (i, j) to G^π and decrease the score function without creating a cycle in G^π. It follows that we must have $\mathcal{Z}(\pi, W^\pi) = \emptyset$ at termination.

Hence, for any pair of nodes i, j, we have two cases. If $\pi(i) > \pi(j)$, then since $\mathcal{Z}(\pi, W^\pi) = \emptyset$ at termination, we have

$$\frac{\partial h}{\partial A^\pi_{ij}} = 0 \implies [\nabla f(W^\pi)]_{ij} = 0$$

On the other hand, if $\pi(i) < \pi(j)$, there are no walks from j to i in G^π. Hence, we have $\frac{\partial h}{\partial A^\pi_{ij}} = 0$. Furthermore, since W^π is a local optimal solution of (5), we have $[\nabla f(W^\pi)]_{ij} = 0$. It follows that:

$$\frac{\partial h}{\partial A^\pi_{ij}} = 0 \implies [\nabla f(W^\pi)]_{ij} = 0, \forall i, j, i \neq j$$

From Lemma 5.2, the algorithm returns a Cst. From Lemma 9.6, the algorithm returns a local minimum if f is convex. ■

10 Experiments

10.1 Experimental Set-up

Our ground truth graphs are Erdos-Renyi DAGs sampled uniformly from all DAGs with d nodes and $4d$ edges (i.e. ER4 graphs). Given a random graph, we assign edge weights independently from Unif($[-2, -0.01] \cup [0.01, 2]$) to obtain a ground truth weighted adjacency matrix $W \in \mathbb{R}^{d\times d}$. While prior work sample edge weights from Unif($[-2, -0.5]\cup[0.5, 2]$) [5, 15], we opt to sample our weights from a wider range of values which also accommodates very small coefficients, since this is more representative of real world situations. Given W, we generate $n = 1000$ samples of $X = W^\top X + z \in \mathbb{R}^d$, where $z \in \mathcal{N}(0, I_{d\times d})$, to a create dataset $\boldsymbol{X} \in \mathbb{R}^{n\times d}$. The least squares score function is $\frac{1}{2n}||\boldsymbol{X} - \boldsymbol{X}W||_F^2$. The neural network is a fully connected neural network with two hidden layers of size d^2 and the nonlinear sigmoid function. The weights of the neural network are randomly intialized from the uniform distribution $\mathcal{U}\left(-\sqrt{\frac{1}{k}}, \sqrt{\frac{1}{k}}\right)$ where k denotes the number of input features to the layer.

RANDOM TOPO was implemented using code from the github repository for [5], with hyperparameters set to the recommended values in [5]. Code for β-LS and β-LSopt was also adopted from this repository (License: Non-exclusive license to distribute).

We use the Dual Xeon Gold 6226R CPU in a high performance computing cluster to run all experiments. None of our jobs required more than 5GB RAM. Note that this research required more compute than the experiments reported in the paper due to the need for hyperparameter tuning.

10.2 Minimizing (5)

For the least squares score function, we follow [5]'s code and optimize (5) using scipy's LinearRegression library, since it is significantly more efficient than computing closed form solutions locally. For the neural network, we initialize edge weights independently from the uniform distribution $\mathcal{U}[-2, 2]$, and implement gradient descent in Pytorch [8] until convergence.

10.3 Hyperparameters

Table 5 lists the hyperparameter γ used for each score and acyclicity function across graph sizes. The hyperparameters for the RANDOM-TOPO algorithm are set to values recommended in [5].

Table 5: Hyperparameter γ for β-LSopt

Score	Acyclicity Function	γ
Least Squares	DAGMA	0.1
	Matrix Polynomial	0.25
Neural Network	DAGMA	10^{-6}
	Matrix Polynomial	10^{-4}

10.4 Additional Ground Truth Simulation Setups

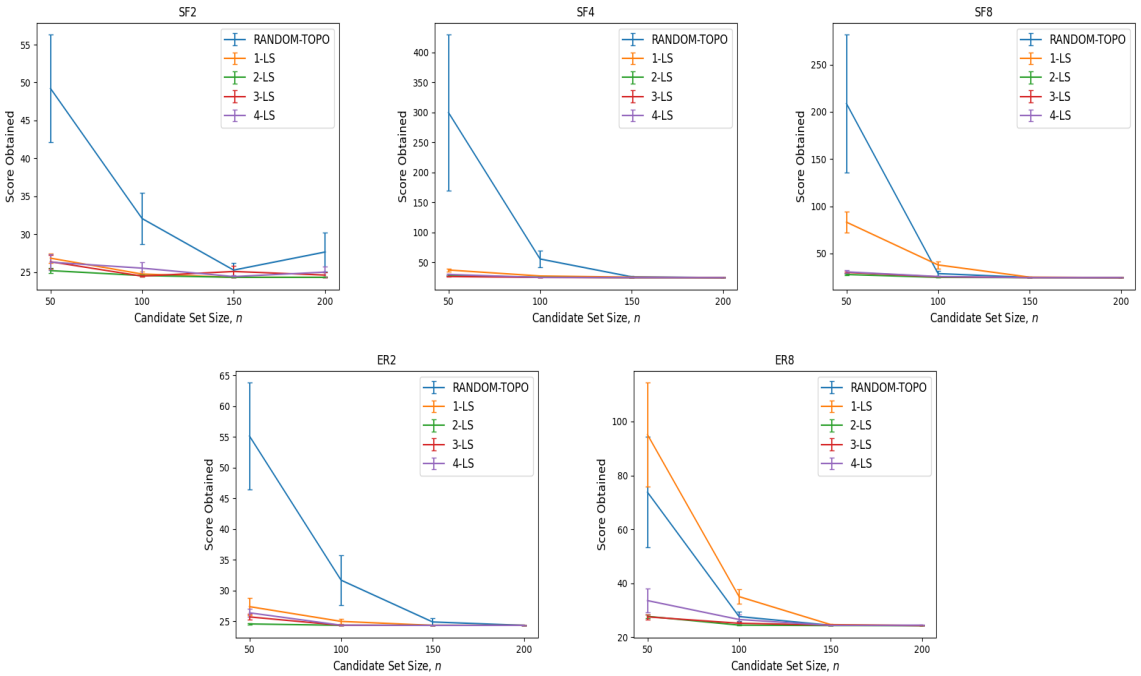


Figure 2: Scores obtained by β-LS, $\beta \in \{1, 2, 3, 4\}$ and RANDOM-TOPO with DAGMA acyclicity function for various candidate set sizes in a variety of ground truth simulation set-ups. The score is least squares, and $d = 50$. Error bars denote finite sample estimates of the standard error of the mean.

Table 6: RANDOM-TOPO vs 2-LSopt with DAGMA acyclicity function in a variety of ground truth simulation set-ups. The score is least squares, and $d = 50$. We compare methods **without** applying a post-processing threshold to their solutions. Error bars denote finite sample estimates of the standard error of the mean.

Method	Metric	$SF2$	$SF4$	$SF8$	$ER2$	$ER8$
RANDOM TOPO	Score	24.33 ±0.04	24.33 ±0.04	24.33 ±0.04	24.33 ±0.04	24.33 ±0.04
	# Edges Recovered	94.1 ±0.5	184.5 ±0.9	351.9 ±0.9	96.7 ±0.6	391.0 ±1.2
	SHD	1130.6 ±0.5	1038.5 ±0.7	865.5 ±0.5	1127.9 ±0.6	828.5 ±0.7
	Runtime	**123.33** ±8.73	**102.68** ±8.37	**60.22** ±8.59	**92.68** ±5.91	**55.56** **±4.80**
2-LSopt	Score	24.32 ±0.04	24.33 ±0.04	24.37 ±0.06	24.32 ±0.04	24.36 ±0.05
	# Edges Recovered	93.9 ±0.7	185.6 ±0.7	352.0 ±0.7	96.5 ±0.6	390.6 ±1.1
	SHD	1130.4 ±0.7	1037.5 ±0.5	865.5 ±0.7	1128.1 ±0.6	829.3 ±0.9
	Runtime	**80.33** ±7.64	**43.89** ±4.21	**22.33** ±1.64	**59.42** ±5.54	**27.27** ±0.83

10.5 Comparision with NOTEARS and NOFEARS

We include results which show the improvement of 2-LSopt over additional baselines NOTEARS and NOFEARS below.

Table 7: Comparison of different methods on linear ER4 DAGs with equal variance Gaussian noise. We compare methods after applying a post-processing threshold of 0.3 to their solutions. Error bars denote the standard error of the mean.

Method	Metric	10	50	100
2-LSopt	Score	4.96 ±0.02	24.33 ±0.05	47.40 ±0.08
	SHD	6.20 ±1.04	36.60 ±2.34	74.40 ±3.58
	Runtime	0.69 ±0.11	30.62 ±1.37	392.56 ±40.57
NOTEARS	Score	7.88 ±0.84	619.89 ±261.86	638.02 ±141.07
	SHD	16.00 ±1.21	216.80 ±3.87	120.40 ±4.90
	Runtime	3.32 ±0.50	299.41 ±22.48	1016.09 ±54.91
NOFEARS	Score	8.11 ±0.79	690.02 ±326.09	727.16 ±176.12
	SHD	16.50 ±1.38	214.30 ±3.36	121.70 ±5.13
	Runtime	3.08 ±0.37	285.17 ±22.39	898.12 ±62.11

Table 8: Comparison of different methods on linear SF4 DAGs with equal variance Gaussian noise. We compare methods after applying a post-processing threshold of 0.3 to their solutions. Error bars denote the standard error of the mean.

Method	Metric	10	50	100
2-LSopt	Score	4.96 ±0.02	24.34 ±0.04	47.40 ±0.08
	SHD	4.50 ±0.73	34.30 ±2.29	81.80 ±3.20
	Runtime	2.08 ±1.40	41.31 ±5.00	483.87 ±45.84
NOTEARS	Score	7.00 ±0.46	1073.36 ±414.82	16791.65 ±7201.55
	SHD	12.90 ±1.58	96.20 ±4.58	208.60 ±8.02
	Runtime	2.85 ±0.50	275.92 ±29.76	1107.70 ±39.99
NOFEARS	Score	7.68 ±1.14	936.20 ±331.05	24784.78 ±14187.45
	SHD	11.90 ±1.08	100.00 ±4.62	210.30 ±8.81
	Runtime	2.59 ±0.48	255.10 ±14.23	1124.46 ±42.70