

000
001
002
003

TOP-K STRUCTURE SEARCH WITH SOLUTION PATH

004
005
006
007
008
009
010
011
012
013
014
015
016
017
018
019
020
021
022
023
024
025
026
027
028
029
030
031
032
033
034
035
036
037
038
039
040
041
042
043
044
045
046
047
048
049
050
051
052
053**Anonymous authors**

Paper under double-blind review

ABSTRACT

Structure learning algorithms often output a single estimated graph without offering alternative candidates or a way to capture model uncertainty. This is limiting in finite-sample settings with weak signals or noise, where multiple structures can explain the data equally well. In this work, we propose **Top-K Structure Search with Solution Path**, an algorithm that systematically tracks the evolution of edge weights across a range of values of the ℓ_1 sparsity regularization parameter λ . By scoring candidate structures with the Bayesian Information Criterion (BIC), our method ranks and returns the Top-K most plausible structures. Unlike traditional approaches that yield a single solution, our framework provides a ranked set of candidates, enabling better uncertainty assessment. Experiments on synthetic and real-world datasets demonstrate the effectiveness of our approach in capturing structural variability. This highlights the advantage of leveraging solution paths for structure learning, especially in scenarios where committing to a single graph is unreliable. Our framework offers a complementary perspective on structure learning by considering multiple candidate solutions, thereby mitigating the practical instability of solely relying on a single result.

1 INTRODUCTION

Bayesian structure learning aims to identify plausible dependency structures among variables from observational data. A common limitation is that most algorithms return only a single “best” result, even though multiple structures may explain the data equally well, particularly in finite-sample or noisy settings. This single-graph focus obscures structural uncertainty and can mislead downstream inference, for instance in gene regulatory network reconstruction, where multiple high-scoring structures often arise due to limited samples and measurement noise (Friedman et al., 2000).

To address this, several works have proposed Top-K approaches that return multiple high-scoring structures. Bayesian Model Averaging using the k -best Bayesian network structures was introduced in Tian et al. (2012), showing that averaging over multiple models can significantly improve predictive performance compared to relying on one network. This idea was extended in Chen & Tian (2014), who developed a dynamic programming (DP) approach to compute the k -best equivalence classes of Bayesian networks, where each equivalence class represents a set of networks that are indistinguishable from observational data. While exact and principled, this DP method is computationally intensive and scales poorly with the number of variables. To improve scalability, Chen et al. (2016) proposed an A*-based enumeration of equivalence classes via EC-graphs. While this approach avoids full dynamic programming, the number of equivalence classes grows super-exponentially with the number of variables, so even Top-K enumeration becomes computationally challenging for moderate-sized networks (Chickering, 1996; Robinson, 2006). Together, these works highlight the importance of Top-K structure learning because they demonstrate that multiple high-scoring structures can exist for the same data, and considering them improves predictive performance, captures structural variability, and prevents over-reliance on a single graph, while also underscoring the computational challenges of exhaustive or near-exhaustive search.

In parallel, classical structure learning algorithms have focused on finding a single high-scoring graph, such as a directed acyclic graph (DAG) or a Markov equivalence class (MEC). Constraint-based methods such as PC (Spirtes et al., 2000) use conditional independence tests to prune edges, producing a graph consistent with observed independencies. While effective in the large-sample limit, PC is sensitive to noise and limited data (Tsamardinos et al., 2006). Score-based methods such as GES (Chickering, 2002) search over graphs by greedily adding or deleting edges to maximize a

054 score, such as BIC. These approaches can capture more complex structures but are susceptible to
 055 local optima. More recently, the BOSS algorithm (Andrews et al., 2023) introduced a score-based
 056 search over variable orderings, improving efficiency by focusing on the top-ranked candidate orders
 057 rather than traversing the entire graph space. However, like PC and GES, BOSS typically returns a
 058 single solution, such as a DAG or a MEC, and does not directly capture structural uncertainty.

059 Motivated by these limitations, we propose *Top-K Structure Search with Solution Path*. Our framework
 060 generates multiple candidate structures by tracing the evolution of edge weights along the solution
 061 path of an ℓ_1 -regularized optimization problem. Structural changes occur at critical values of the
 062 regularization parameter λ , and the Top-K graphs are then selected based on their scores. Unlike
 063 dynamic programming or A* search, our method does not require exhaustive exploration of the graph
 064 space. Instead, it follows the solution path to efficiently identify multiple high-scoring structures and
 065 reveals which edges consistently appear across the Top-K candidates and which edges vary, providing
 066 insight into the confidence of different parts of the network.

067 Our focus is on structure learning and the systematic exploration of high-scoring graph skeletons,
 068 [extending Top-K ideas beyond Bayesian formulations to continuous optimization models, where such](#)
 069 [exploration has not been previously studied](#). This reflects our practical goal of ranking candidate
 070 models and capturing structural variability, rather than committing to a single graph.

071 **Contributions.** (1) We propose a principled framework to generate and rank multiple structures
 072 along the solution path induced by varying the ℓ_1 sparsity regularization parameter (λ), using a
 073 gradient-based method to efficiently trace structural changes. (2) We score structures using BIC or
 074 likelihood, and quantify uncertainty by calculating both skeletal graph uncertainty and edge-level
 075 uncertainty. (3) We empirically validate our approach on synthetic and real-world datasets.

076 **Paper organization.** In Section 2, we introduce the necessary preliminaries for our approach. In
 077 Section 3, we present the proposed Top-K Structure Search with Solution Path method in detail.
 078 Section 4 covers our experimental evaluation, including both synthetic experiments and real-world
 079 data experiments. Finally, in Section 5 we discuss key takeaways and outline the limitations and
 080 directions for future work.

082 2 PRELIMINARIES

084 We first start with the formulation of the problem. Let $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ be a directed acyclic graph (DAG)
 085 where the vertex set $\mathbf{V} = \{X_1, \dots, X_d\}$ represents d random variables. The joint random vector is
 086 denoted as $\mathbf{X} = (X_1, \dots, X_d)$ with an associated probability distribution $P_{\mathbf{X}}$. For each variable X_i ,
 087 define $\mathbf{X}_{pa(i)}$ as the set of its parents in \mathcal{G} , meaning all variables X_j for which there exists a directed
 088 edge $X_j \rightarrow X_i$ in \mathbf{E} . Throughout this work, we assume the principle of *causal sufficiency*, implying
 089 that there are no unobserved confounders.

090 We model the data-generating process as a linear structural equation model (SEM):

$$092 \quad \mathbf{X} = B^{\top} \mathbf{X} + \mathbf{N},$$

093 where $B \in \mathbb{R}^{d \times d}$ is the weighted adjacency matrix of the graph \mathcal{G} , and $\mathbf{N} = (N_1, \dots, N_d)$ is a vector
 094 of jointly independent noise variables. Each nonzero entry $B_{ij} \neq 0$ encodes a direct causal effect
 095 from X_j to X_i , corresponding to an edge $X_j \rightarrow X_i$ in \mathcal{G} . The diagonal entries of B are zero, and the
 096 acyclicity of \mathcal{G} ensures that $(I - B^{\top})$ is invertible.

097 Let x be the collection of n i.i.d. samples from the distribution of \mathbf{X} . Given x , the objective of
 098 a structure learning algorithm is to recover the structure of B , and hence the underlying DAG
 099 \mathcal{G} . The goal of the Top-K Solution Path method is to identify a set of K high-scoring candidate
 100 structures, denoted as $\widehat{\mathcal{G}}_1, \widehat{\mathcal{G}}_2, \dots, \widehat{\mathcal{G}}_K$, from observational data. These structures correspond to
 101 plausible graphical models that explain the data well, providing a richer representation of uncertainty
 102 in structure learning compared to a single-point estimate.

104 3 TOP-K SOLUTION PATH METHOD

106 In this section, we present the complete methodology behind the proposed Top-K Solution Path
 107 framework. In Section 3.1, we define the objective function that forms the basis of our optimization.

108 Section 3.2 outlines the gradient-based optimization strategy used to minimize this objective. Section
 109 3.3 details the solution path method, which efficiently traces structural changes as the ℓ_1 sparsity
 110 regularization parameter (λ) varies. In Section 3.4, we describe how the Top-K candidate struc-
 111 tures are selected based on BIC scores. Finally, Section 3.5 introduces our approach for estimating
 112 uncertainty in the discovered structures.

113

114 3.1 OBJECTIVE FUNCTION

115

116 We formulate the structure discovery task inspired by GOLEM (Graph Optimization via Linear
 117 Equations Method) (Ng et al., 2020), as minimizing a regularized objective composed of three parts:
 118 a likelihood term, an ℓ_1 sparsity-inducing penalty, and a soft acyclicity constraint.

119

120 **Likelihood.** The likelihood component is the average negative log-likelihood of a linear additive
 121 Gaussian noise model:

122

$$123 \mathcal{L}(B; x) = \frac{1}{2} \sum_{i=1}^d \log \left(\sum_{k=1}^n \left(x_i^{(k)} - B_i^T x^{(k)} \right)^2 \right) - \log |\det(I - B)| + \frac{d}{2} \log \left(\frac{2\pi e}{n} \right),$$

124

125 where $B \in \mathbb{R}^{d \times d}$ is the weighted adjacency matrix, x the data, and $x^{(k)}$ is the k^{th} data point.

126

127 **Sparsity constraint.** To encourage sparse solutions and prevent overfitting, we include an ℓ_1 -penalty:
 128 $\|B\|_1 = \sum_{i,j} |B_{ij}|$.

129

130 **DAG constraint.** To enforce soft acyclicity, we add a smooth penalty function $h(B) = \text{tr}(e^{B \circ B}) - d$,
 131 where \circ denotes the Hadamard product (Zheng et al., 2018).

132

133 **Final objective.** The complete objective function to be minimized is:

134

$$\min_{B \in \mathbb{R}^{d \times d}} \mathcal{S}(B; x) = \mathcal{L}(B; x) + \lambda \|B\|_1 + \alpha h(B),$$

135

136 where λ is the ℓ_1 sparsity regularization parameter and α is the coefficient that penalizes cycles.

137

138 3.2 GRADIENT-BASED OPTIMIZATION

139

140 To compute the adjacency matrix B for different values of the ℓ_1 sparsity regularization parameter
 141 λ , denoted by $B(\lambda)$, we employ a gradient-based optimization method. **Although the ℓ_1 penalty is**
 142 **non-differentiable at zero, we use standard subgradient updates for the corresponding term. The**
 143 **active-set mechanism serves as an implicit proximal step where coefficients outside the active set**
 144 **remain fixed, while entries whose gradients exceed a threshold are reintroduced into the optimization.**

145

146 We initialize the solution path at λ_{\max} , where the optimal solution is $B = 0$. Then, we gradually
 147 decrease the value of λ in small increments. For each new λ_{t+1} , we initialize from the previous
 148 solution $B(\lambda_t)$ and perform g steps of gradient descent, restricted to the current active set \mathcal{A}_t , to
 149 minimize the objective function.

150

151 The update rule at each gradient step $s \in \{0, \dots, g-1\}$ is:

152

153

$$B^{(s+1)} = B^{(s)} - \eta_t \left[\nabla_B \mathcal{L}(B^{(s)}; x) + \lambda_{t+1} \text{sgn}(B^{(s)}) + \alpha \nabla_B h(B^{(s)}) \right]_{\mathcal{A}_t},$$

$$B^{(0)} = B(\lambda_t), \quad B(\lambda_{t+1}) = B^{(g)}.$$

154

155

156

157 Here, η_t is the adaptive learning rate, and the notation $[\cdot]_{\mathcal{A}_t}$ denotes that the gradient update is applied
 158 only to the elements in the active set \mathcal{A}_t . Any adaptive learning method can be used to choose η_t ,
 159 such as those typically used in gradient-based optimization techniques.

160

161 The explicit gradient expressions are given by:

162

163

164

165

166

167

$$\frac{\partial \mathcal{L}(B; x)}{\partial B_{ij}} = -\frac{\sum_{k=1}^n \left(x_i^{(k)} - B_i^T x^{(k)} \right) x_j^{(k)}}{\sum_{k=1}^n \left(x_i^{(k)} - B_i^T x^{(k)} \right)^2} + \left((I - B)^{-T} \right)_{ij},$$

162
163
164

$$\nabla_B h(B) = (e^{B \circ B})^T \circ 2B.$$

165 3.3 SOLUTION PATH METHOD
166167 In this subsection, we describe the solution path method used to obtain the adjacency matrix B across
168 different values of the sparsity regularization parameter λ . We will cover these key aspects of the
169 method: the initial conditions used for optimization, the identification and evolution of the active set,
170 the tracking of changes in the adjacency matrix B , and the identification of critical points along the
171 solution path. Our method is inspired from other solution path based methods and the effect of Lasso
172 regularization on them (Rosset, 2004; Park & Hastie, 2007; Efron et al., 2004; Tibshirani, 1996).
173174 **Initial Condition.** We initialize the optimization at a sufficiently large sparsity regularization
175 weight, denoted λ_{\max} , for which the optimal solution is $B = 0$. At this value, the strong sparsity
176 penalty forces all entries of B to zero. The initial active set \mathcal{A}_0 , containing the indices with the largest
177 gradient magnitude at $B = 0$, is where the first nonzero weight will enter as λ begins to decrease:
178

179
$$\lambda_0 = \lambda_{\max} = \max_{\substack{i,j \\ i \neq j}} \left| \frac{\partial \mathcal{L}(B; x)}{\partial B_{ij}} \right|_{B=0}, \quad \mathcal{A}_0 = \arg \max_{\substack{i,j \\ i \neq j}} \left| \frac{\partial \mathcal{L}(B; x)}{\partial B_{ij}} \right|_{B=0}.$$

180
181

182 Starting from λ_{\max} , we decrease λ in small steps and track the evolution of non-zero weights in B as
183 they enter the model.
184185 **Active Set Update Rule.** Given the current active set \mathcal{A}_t , we update it to \mathcal{A}_{t+1} in two steps:
186187 (1) Removal: Elements with small magnitude are removed from the active set:
188

189
$$\mathcal{A}'_t = \{(i, j) \in \mathcal{A}_t \mid |B_{ij}| \geq \delta\}, \quad \text{where } \delta > 0.$$

190

191 (2) Addition: Elements not currently in the active set are added if their corresponding gradient
192 magnitude exceeds the regularization threshold:
193

194
$$\mathcal{A}_{t+1} = \mathcal{A}'_t \cup \left\{ (i, j) \notin \mathcal{A}'_t \mid i \neq j, \lambda_{t+1} \leq \left| \frac{\partial \mathcal{L}(B; x)}{\partial B_{ij}} \right|_{B=B(\lambda_{t+1})} \right\}.$$

195
196

197 This procedure enables the active set to adapt dynamically to the current solution and the underlying
198 optimization landscape.
199200 **Solution Path Progression.** We begin the solution path progression at $t = 0$ with the initial
201 conditions $\lambda_0 = \lambda_{\max}$ and the initial active set \mathcal{A}_0 , which is determined based on the sparsity structure
202 at λ_0 . The progression follows a uniform grid search approach over the λ -space, decrementing λ_t by
203 a fixed step size ϵ until λ_t reaches a sufficiently small threshold. The process is outlined as follows:
204205 1. At each step, decrement the regularization parameter: $\lambda_{t+1} = \lambda_t - \epsilon$.
206 2. Update the weight matrix $B(\lambda_{t+1})$ using the gradient method, starting from $B(\lambda_t)$.
207 3. Update the active set \mathcal{A}_{t+1} from \mathcal{A}_t based on the solution $B(\lambda_{t+1})$.
208 4. Increment the iteration counter: $t = t + 1$.
209210 This process is repeated until λ_t becomes sufficiently small, i.e., $\lambda_t \leq \epsilon$, at which point the solution
211 path progression terminates.
212213 **Identifying Critical Points.** As λ decreases, edges in the learned structure progressively appear or
214 disappear, reflecting structural transitions. We define the set of critical points Λ as the values of λ
215 at which any weight $B_{ij}(\lambda)$ changes support—that is, switches from zero to nonzero or vice versa.
Specifically, for any pair (i, j) , a value λ is considered a critical point if:

216

217
$$B_{ij}(\lambda - \zeta) = 0 \quad \text{and} \quad B_{ij}(\lambda + \zeta) \neq 0,$$

218 or

219
$$B_{ij}(\lambda - \zeta) \neq 0 \quad \text{and} \quad B_{ij}(\lambda + \zeta) = 0,$$

220

221 for a sufficiently small $\zeta > 0$. These values capture points of structural change along the solution path.
 222 The set of structures at these critical points, denoted $\widehat{B}(\Lambda)$, forms the candidate pool for selecting the
 223 Top-K structures.

224

225

3.4 TOP-K STRUCTURE SELECTION

226

227 Once we have the set of critical points Λ and the corresponding estimated weight matrices $\widehat{B}(\Lambda)$, we
 228 extract their binary adjacency matrices and perform regression, and then rank them based on their
 229 BIC scores.

230

231 **Binary Adjacency Matrix.** For each estimated weight matrix $\widehat{B}(\lambda)$ at a critical point $\lambda \in \Lambda$, we
 232 extract the corresponding graph structure by applying an element-wise threshold:

233

234
$$\widehat{A}(\lambda) = \begin{cases} 1, & \text{if } |\widehat{B}(\lambda)| \geq \tau \\ 0, & \text{otherwise} \end{cases},$$

235

236 where $\widehat{A}(\lambda)$ is the binary adjacency matrix representing the inferred structure, and $\tau > 0$ is a small
 237 constant controlling sensitivity to very weak edges.

238

239 **Re-estimating Weights via Regression.** To correct for the shrinkage introduced by Lasso regularization,
 240 we re-estimate the weights using ordinary least squares, constrained to the support of the
 241 inferred structure. Specifically, we solve the following optimization problem:

242

243
$$\widetilde{B}(\lambda) = \arg \min_B \|x - B^\top x\|_F^2 \quad \text{subject to} \quad \text{supp}(B) \subseteq \text{supp}(\widehat{A}(\lambda)),$$

244

245 where $x \in \mathbb{R}^{d \times n}$ is the observed data matrix, $\|\cdot\|_F$ denotes the Frobenius norm, and $\widehat{A}(\lambda)$ is the
 246 binary adjacency matrix. The resulting matrix $\widetilde{B}(\lambda)$ contains the re-estimated weights.

247

248 **Computing Likelihood and BIC Scores.** Given the re-estimated weight matrix $\widetilde{B}(\lambda)$, we first
 249 compute the average negative log-likelihood of the data under the model:

250

251
$$\mathcal{L}(\widetilde{B}(\lambda); x) = \frac{1}{2} \sum_{i=1}^d \log \left(\sum_{k=1}^n \left(x_i^{(k)} - \widetilde{B}(\lambda)_i^T x^{(k)} \right)^2 \right) - \log |\det(I - \widetilde{B}(\lambda))| + \frac{d}{2} \log \left(\frac{2\pi e}{n} \right).$$

252

253 The total negative log-likelihood for n samples is then $n\mathcal{L}(\widetilde{B}(\lambda); x)$.

254

255 From this, we calculate the Bayesian Information Criterion (BIC) score as:

256

257
$$\text{BIC}(\widetilde{B}(\lambda)) = 2n \cdot \mathcal{L}(\widetilde{B}(\lambda); x) + \log(n) \cdot |\widehat{E}(\widetilde{B}(\lambda))|,$$

258

259 where $|\widehat{E}(\widetilde{B}(\lambda))|$ is the number of edges in the inferred graph.

260

261 **Sorting and Selecting Top-K Graphs.** Each candidate graph in $\mathcal{G}(\Lambda)$ corresponds to a binary
 262 adjacency matrix $\widehat{A}(\lambda)$ obtained by thresholding the estimated weights $\widehat{B}(\lambda)$ at a critical point $\lambda \in \Lambda$.
 263 To identify the most plausible structures, we select the Top-K graphs with the lowest BIC scores:

264

265

266
$$\Lambda_K = \arg \min_{\substack{\lambda \in \Lambda \\ |\Lambda_K| \leq K}} \text{BIC}(\widetilde{B}(\lambda)), \quad \mathcal{G}_K = \{ \widetilde{B}(\lambda) \mid \lambda \in \Lambda_K \}.$$

267

268

269

270 These graphs represent the best candidates for the underlying structure, balancing fit and complexity.
 271

272 **3.5 UNCERTAINTY QUANTIFICATION**
 273

274 To quantify uncertainty over the selected graph structures, we convert their BIC scores into a
 275 probability-like distribution using a temperature-scaled exponential transformation.
 276

277 **Graph-Level Uncertainty.** Let $\{\hat{\mathcal{G}}_1, \dots, \hat{\mathcal{G}}_K\}$ denote the Top-K graphs corresponding to the weight
 278 matrices $\{\tilde{B}(\lambda_1), \dots, \tilde{B}(\lambda_K)\}$. We define the relative probability of each graph as:
 279

$$280 \quad P(\hat{\mathcal{G}}_k) = \frac{\exp\left(-\frac{1}{2T} \cdot \text{BIC}(\tilde{B}(\lambda_k))\right)}{\sum_{j=1}^K \exp\left(-\frac{1}{2T} \cdot \text{BIC}(\tilde{B}(\lambda_j))\right)}.$$

$$281 \quad 282 \quad 283$$

284 Here, $T > 0$ is a *temperature* parameter that modulates the sharpness of the distribution. A lower
 285 T amplifies differences between BIC scores, favoring the top-scoring graph more strongly, while
 286 a higher T yields a flatter distribution, expressing greater uncertainty. This scaling enables more
 287 flexible and calibrated uncertainty estimates, especially when score differences are either too sharp or
 288 too subtle. One practical heuristic is to set T such that $P(\hat{\mathcal{G}}_K) = \frac{1}{2K}$, ensuring that the top-ranked
 289 graph receives a relatively high probability while still allowing meaningful contribution from all K
 290 candidate graphs.
 291

292 Higher $P(\hat{\mathcal{G}}_k)$ indicates stronger support for graph $\hat{\mathcal{G}}_k$, while more uniform probabilities suggest
 293 greater structural ambiguity.
 294

295 **Edge-Level Uncertainty.** The probability of an edge (i, j) appearing in the structure is defined as:
 296

$$297 \quad P_{\text{edge}}(i, j) = \sum_{k=1}^K P(\hat{\mathcal{G}}_k) \cdot \mathbb{I}\left((i, j) \in \hat{\mathcal{G}}_k\right),$$

$$298 \quad 299$$

300 where $\mathbb{I}(\cdot)$ is an indicator function. This represents a soft confidence score for the existence (but not
 301 direction) of each edge, aggregated across the Top-K graphs.
 302

303 This probabilistic framework offers a principled estimate of graph and edge-level uncertainty without
 304 requiring computationally intensive resampling.
 305

306 **4 EXPERIMENTS**
 307

308 We conduct extensive experiments on both synthetic and real-world data to evaluate the performance
 309 of our method. On synthetic datasets, we systematically vary key parameters such as the number
 310 of samples n , the Top-K value K , graph density ρ , and the number of variables d . We compare
 311 our method against standard structure learning algorithms including GES, PC, BOSS and Top-K
 312 A*. Evaluation is performed by comparing the skeleton of the predicted graph to the skeleton of
 313 the ground truth graph, using metrics such as precision, recall, F1 score, and accuracy. We use the
 314 L-BFGS-B algorithm from `scipy.optimize` for gradient-based optimization. For Top-K A*, we
 315 use a simple cost-based heuristic on synthetic data and a more informed lower-bound heuristic on the
 316 Sachs dataset, as the latter provides better pruning in noisy real-world settings. All experiments are
 317 run on a standard CPU machine, and each individual run typically completes in under a few minutes
 318 for smaller d values. We use violin plots in `seaborn` to visualize the distribution of metrics.
 319

320 **4.1 SYNTHETIC DATA EXPERIMENTS**

321 **Experimental Setup.** We evaluate our method in a challenging setting where edge weights are
 322 sampled from the ranges $[-0.4, -0.1] \cup [0.1, 0.4]$, and Gaussian noise variances are uniformly
 323 sampled from the interval $[0.1, 0.3]$. This setup corresponds to a low signal-to-noise ratio (SNR),
 weak causal effects, and small sample sizes—a regime where many existing algorithms tend to

perform poorly. Our approach is particularly well-suited for this scenario, as it is designed to handle noisy data and subtle dependencies by considering multiple candidate solutions rather than committing to a single estimated structure. For Top-K A* and our method, we generate the Top-K candidate structures and select the one with the highest skeleton accuracy as the representative structure to compare against other algorithms. The number of runs for each experiment is 100, except for the experiments varying the number of variables d , where the number of runs is reduced depending on d for computational reasons.

Effect of Sample Size (n). We evaluate performance across different sample sizes $n \in \{20, 50, 100\}$, focusing on the small-sample regime, while keeping other parameters fixed: $d = 6$, $K = 5$, and $\rho = 0.4$. As shown in Figure 1, our method consistently outperforms others, especially when n is small. While all methods improve as n increases, our gains are most pronounced in low-data settings due to increased robustness.

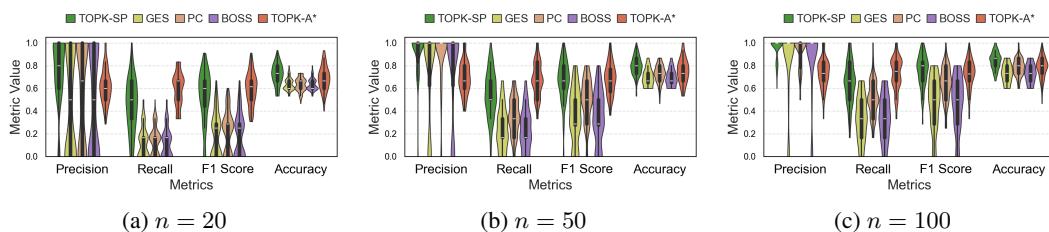


Figure 1: Comparison of methods as sample size n varies.

Effect of number of Top-K structures (K). To evaluate the impact of considering multiple candidate structures, we fix a low sample size $n = 20$ and vary $K \in \{1, 5, 10\}$, keeping other parameters fixed: $d = 6$ and $\rho = 0.4$. As shown in Figure 2, performance improves steadily with larger K , demonstrating the benefit of exploring multiple structures rather than relying on a single estimate—especially in low-data regimes.

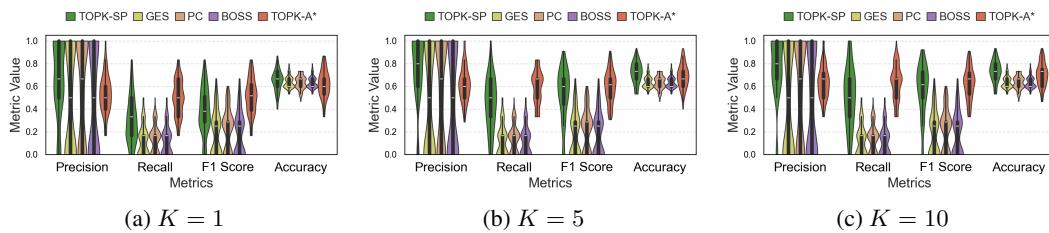


Figure 2: Comparison of methods as the K value varies.

Effect of Graph Density (ρ). We analyze performance across varying graph densities $\rho \in \{0.2, 0.5, 0.8\}$, representing sparse to dense structures, while keeping other parameters fixed: $d = 6$, $n = 100$, and $K = 5$. As shown in Figure 3, our method consistently performs well across all densities and shows strong performance even in dense graphs, where accurately recovering the structure is generally more challenging.

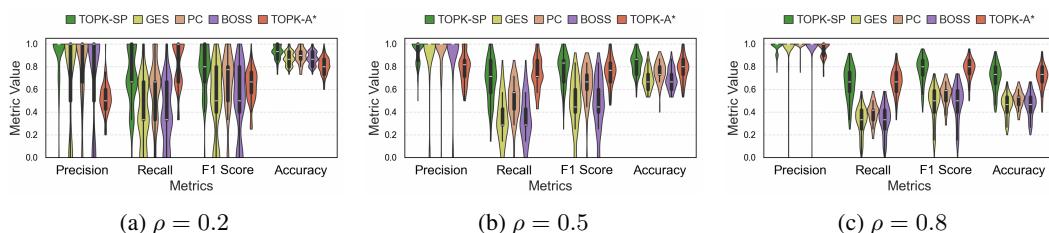
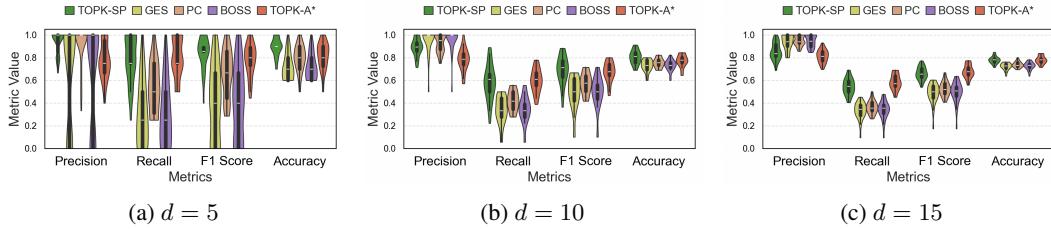
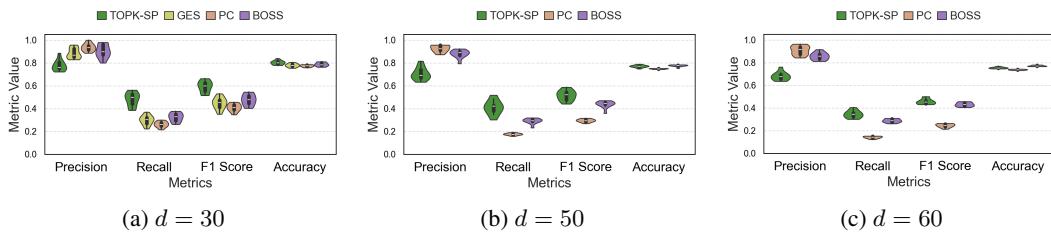


Figure 3: Comparison of methods as graph density ρ varies.

378
 379 **Effect of Number of Variables (d).** For smaller networks with $d \in \{5, 10, 15\}$, we fix the sample
 380 size to $n = 100$, the number of candidate structures to $K = 5$, and the edge density to $\rho = 0.4$. We
 381 perform 30 runs for each of these experiments. As shown in Figure 4, our method performs well in
 382 these smaller problem sizes and consistently outperforms the baseline algorithms.

389 Figure 4: Comparison of methods as the number of variables d varies (for smaller d).
 390

393 For larger networks with $d \in \{30, 50, 60\}$, we keep $n = 100$ and $\rho = 0.3$, while adjusting the number
 394 of candidate structures to $K = 10$ for $d = 30$ and $K = 15$ for $d = 50$ and $d = 60$. To mitigate
 395 the increased computational complexity, we perform 10 runs for each experiment. Since Top-K
 396 A* and GES become computationally expensive, they are omitted for high d . Figure 5 illustrates
 397 that our method maintains strong performance as the dimensionality d increases. While precision is
 398 lower compared to other methods—due to our solution path-based approach capturing more false
 399 positives—the recall improves substantially. This leads to a higher F1 score, and overall accuracy is
 400 also usually better.

408 Figure 5: Comparison of methods as the number of variables d varies (for larger d).
 409

4.2 REAL-WORLD DATA EXPERIMENTS

414 In this section, we evaluate our method and compare it to other algorithms using the Sachs [observational](#)
 415 dataset (Sachs et al., 2005). The Sachs dataset contains $n = 853$ samples and $d = 11$ variables
 416 [with a 17-edge ground truth](#), representing protein signaling pathways in cells. The ground truth
 417 skeleton of the structure is shown in Figure 6, where the black lines represent the true edges.

418 The outputs of PC, GES, and BOSS are identical, all producing the same skeleton shown in Figure 7.
 419 In this figure, green lines represent true positive edges, while dotted grey lines indicate false negatives.
 420 In contrast, our method’s 7th-best skeleton (Figure 8) achieved the highest accuracy and significantly
 421 outperformed others in F1 score, as shown in Figure 10. In Figure 8, light green lines are correct
 422 edges shared by all methods, dark green lines are additional true positives predicted only by our
 423 method, and red lines are false positives. Looking at the structures generated by our Top-K method in
 424 Figure 9, we observe that the Top-2 skeleton is the same as the one predicted by other algorithms.
 425 The Top-7 skeleton has the highest accuracy and F1 score among other candidate structures. This
 426 demonstrates the benefit of considering Top-K structures, as it allows us to explore multiple structures.
 427 Additionally, most of the weights in the network are small, which makes our method more effective.

5 DISCUSSION

431 Our method introduces a Top-K solution path framework for structure learning that leverages continuous
 432 optimization and model diversity. By exploring a family of candidate graphs across the sparsity

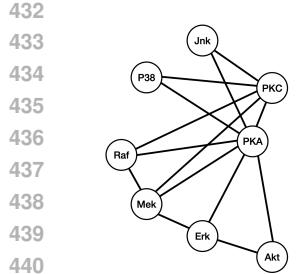


Figure 6: Ground truth skeleton for the Sachs dataset

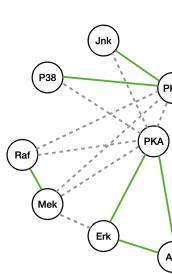


Figure 7: Skeleton output of PC, GES and BOSS

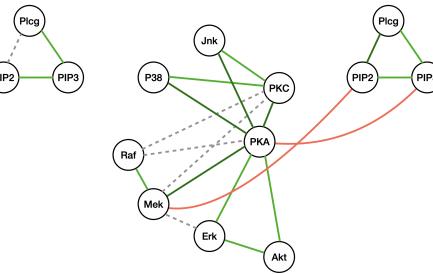


Figure 8: Top-7 Skeleton output of Top-K Method

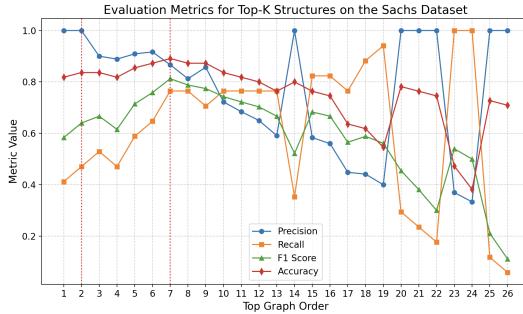


Figure 9: Evaluation metrics for the Top-K Structures on the Sachs dataset

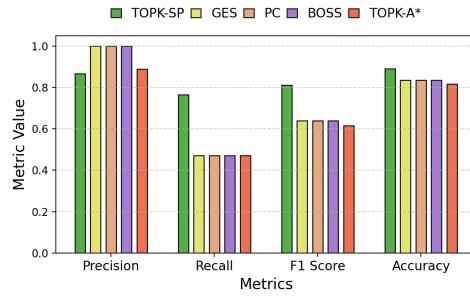


Figure 10: Comparison of methods on the Sachs dataset

regularization parameter λ and selecting the Top- K highest-scoring structures, we achieve robust performance, particularly in low-data, weak causal links, and noisy regimes.

In terms of computational complexity, our approach scales as $\mathcal{O}(gd^3n/\epsilon)$, where g is the number of gradient steps per λ point and ϵ is the granularity of the λ grid. This cost is independent of K , as the Top- K structures are selected from a precomputed candidate pool.

A key strength of our approach lies in its suitability for real-world scenarios where many causal links are weak—common in domains such as biology, economics, and social sciences. These weak edges may be missed by methods focused on a single graph. In contrast, our method retains multiple plausible structures, allowing such weak but meaningful relationships to appear in at least some of the top candidates. This offers a more nuanced view of the underlying dependencies and complements traditional approaches. Moreover, while our method and Top-K A* perform similarly in terms of accuracy and other metrics, Top-K A* quickly becomes computationally infeasible as the number of variables increases, whereas our approach remains tractable even for higher-dimensional settings. This scalability is a central advantage of our method.

Limitations and Future Work. While effective, our method requires setting hyperparameters such as the λ grid resolution ϵ and the number of top solutions K . Finding a principled way to select these values remains an open challenge, and future work could explore adaptive or data-driven tuning schemes. Another promising direction is to develop hybrid models that combine standard single-solution methods with our Top- K framework, which would enable reliable identification of both strong and weak causal links. While our experiments focus on small and moderate-sized graphs, handling very large networks may benefit from additional algorithmic improvements such as parallelization or sparsity-aware heuristics. Importantly, existing Top- K methods based on dynamic programming or A* search quickly become computationally infeasible as d grows, whereas our approach can scale to larger graphs by following the solution path. Incorporating domain knowledge through constraints or priors could further enhance interpretability. Finally, validation on more real-world datasets will help establish broader applicability.

486 REFERENCES
487

488 Bryan Andrews, Joseph Ramsey, Ruben Sanchez Romero, Jazmin Camchong, and Erich Kummerfeld.
489 Fast scalable and accurate discovery of dags using the best order score search and grow shrink
490 trees. *Advances in neural information processing systems*, 36:63945–63956, 2023.

491 Eunice Yuh-Jie Chen, Arthur Choi Choi, and Adnan Darwiche. Enumerating equivalence classes of
492 bayesian networks using ec graphs. In *Artificial Intelligence and Statistics*, pp. 591–599. PMLR,
493 2016.

494 Yetian Chen and Jin Tian. Finding the k-best equivalence classes of bayesian network structures for
495 model averaging. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 1786–1792,
496 2014.

497 David Maxwell Chickering. Learning bayesian networks is np-complete. In *Learning from data:
498 Artificial intelligence and statistics V*, pp. 121–130. Springer, 1996.

499 David Maxwell Chickering. Optimal structure identification with greedy search. *Journal of machine
500 learning research*, 3(Nov):507–554, 2002.

501 Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *The
502 Annals of Statistics*, 32(2):407–499, 2004.

503 Nir Friedman, Michal Linial, Iftach Nachman, and Dana Pe’er. Using bayesian networks to analyze
504 expression data. In *Proceedings of the fourth annual international conference on Computational
505 molecular biology*, pp. 127–135, 2000.

506 Ignavier Ng, AmirEmad Ghassami, and Kun Zhang. On the role of sparsity and dag constraints for
507 learning linear dags. *Advances in Neural Information Processing Systems*, 33:17943–17954, 2020.

508 Mee Young Park and Trevor Hastie. L 1-regularization path algorithm for generalized linear models.
509 *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 69(4):659–677, 2007.

510 Robert W Robinson. Counting unlabeled acyclic digraphs. In *Combinatorial Mathematics V: Pro-
511 ceedings of the Fifth Australian Conference, Held at the Royal Melbourne Institute of Technology,
512 August 24–26, 1976*, pp. 28–43. Springer, 2006.

513 Saharon Rosset. Following curved regularized optimization solution paths. *Advances in Neural
514 Information Processing Systems*, 17, 2004.

515 Karen Sachs, Omar Perez, Dana Pe’er, Douglas A Lauffenburger, and Garry P Nolan. Causal protein-
516 signaling networks derived from multiparameter single-cell data. *Science*, 308(5721):523–529,
517 2005.

518 Peter Spirtes, Clark N Glymour, Richard Scheines, and David Heckerman. *Causation, prediction,
519 and search*. MIT press, 2000.

520 Jin Tian, Ru He, and Lavanya Ram. Bayesian model averaging using the k-best bayesian network
521 structures. *arXiv preprint arXiv:1203.3520*, 2012.

522 Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical
523 Society Series B: Statistical Methodology*, 58(1):267–288, 1996.

524 Ioannis Tsamardinos, Laura E Brown, and Constantin F Aliferis. The max-min hill-climbing bayesian
525 network structure learning algorithm. *Machine learning*, 65:31–78, 2006.

526 Xun Zheng, Bryon Aragam, Pradeep K Ravikumar, and Eric P Xing. Dags with no tears: Continuous
527 optimization for structure learning. *Advances in neural information processing systems*, 31, 2018.

540 **A DETAILED EXAMPLE**
 541

542
 543 We now walk through a concrete example to illustrate the application of our method in detail. The
 544 setting is as follows:

545
 546 • Number of nodes: $d = 5$
 547
 548 • Sample size: $n = 100$
 549
 550 • Graph density: $\rho = 0.4$
 551
 552 • Number of top structures considered: $K = 3$
 553
 554 • Step size: $\epsilon = \lambda_{\max}/100$
 555

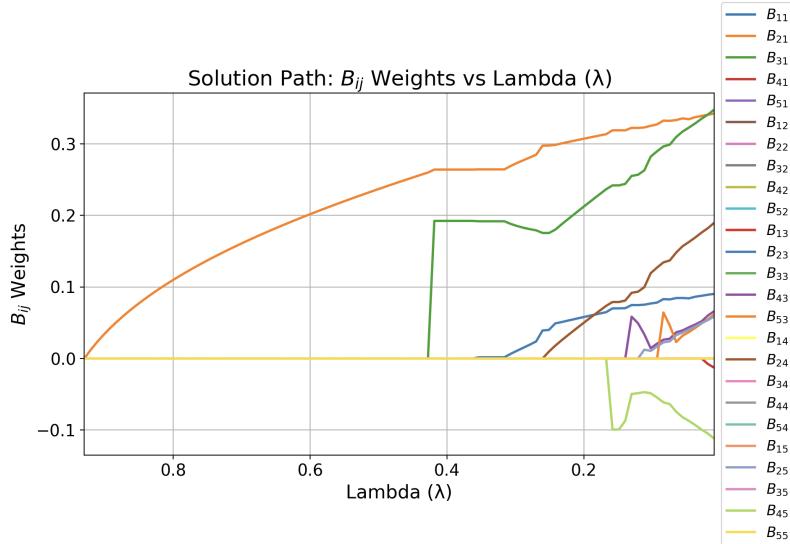
556 The true weighted adjacency matrix B_{true} used to generate the data is:
 557

558
 559
 560
 561

$$B_{\text{true}} = \begin{bmatrix} 0 & 0.25 & 0.26 & 0 & 0 \\ 0 & 0 & 0.19 & 0.16 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

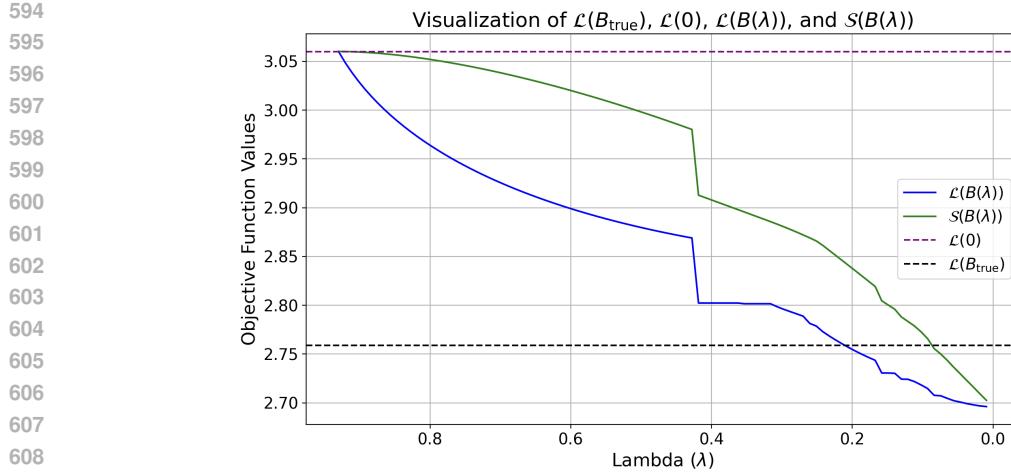
 562
 563
 564

565
 566 This corresponds to a directed acyclic graph (DAG) where node 1 influences nodes 2 and 3, and node
 567 2 influences nodes 3 and 4. The remaining nodes do not have outgoing edges.
 568



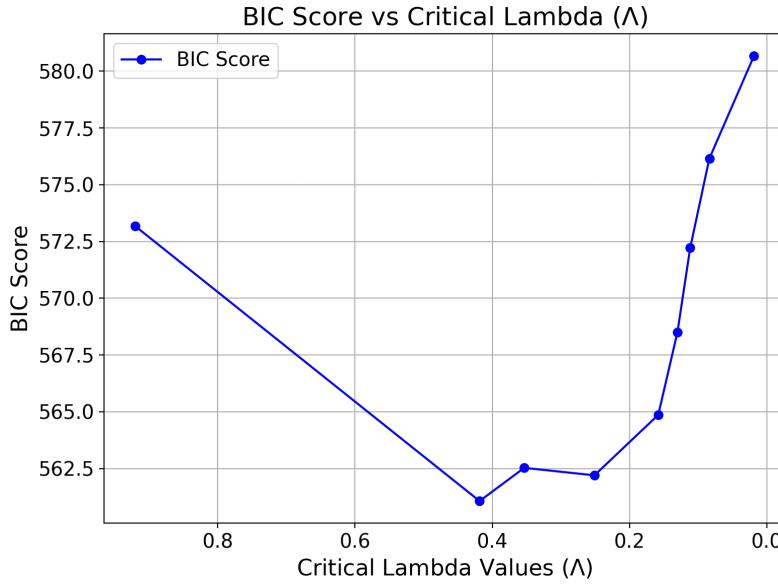
589 Figure 11: Solution path of edge weights B_{ij} versus regularization strength λ .
 590

591 Figure 11 illustrates the solution path of the weights against the sparsity regularization parameter
 592 λ . Initially, for high values of λ , all weights are zero. As λ decreases from λ_{\max} , the regularization
 593 becomes weaker and certain weights begin to emerge—first gradually, and then more rapidly as λ
 continues to drop.

Figure 12: Objective function values as a function of the regularization parameter λ .

614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644

Figure 12 shows how the objective functions are minimized as λ decreases. As the regularization weakens, the model fits the data more closely, reflected in the gradual reduction of the objective values. However, a decreasing objective function value does not necessarily imply recovery of the true underlying structure. This may be due to overfitting, where the model captures noise or spurious relationships in the data rather than the genuine structural connections.

Figure 13: BIC scores of graphs after regression plotted against the critical lambda values (Λ).

645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
789
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
889
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
979
980
981
982
983
984
985
986
987
988
989
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2249
2250
2251
2252
2253
2254
2255
2256
2257
225

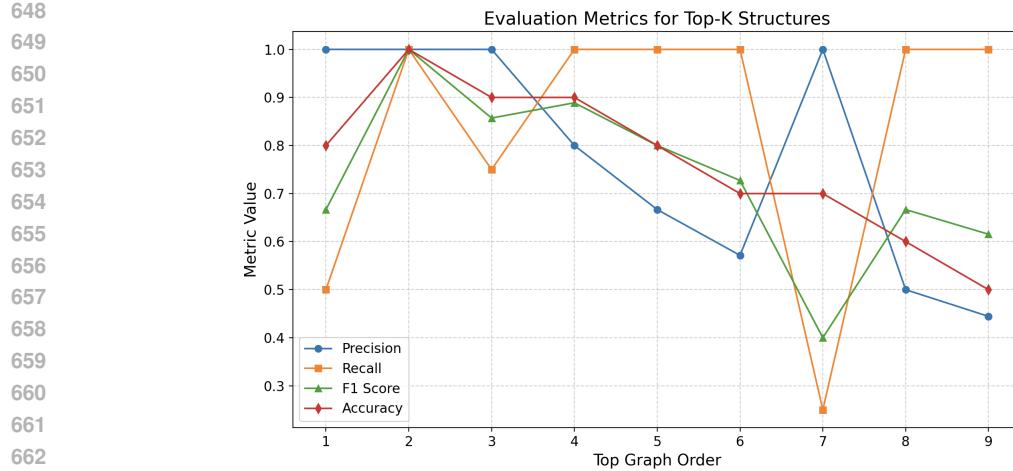


Figure 14: Evaluation metrics for the Top-K structures.

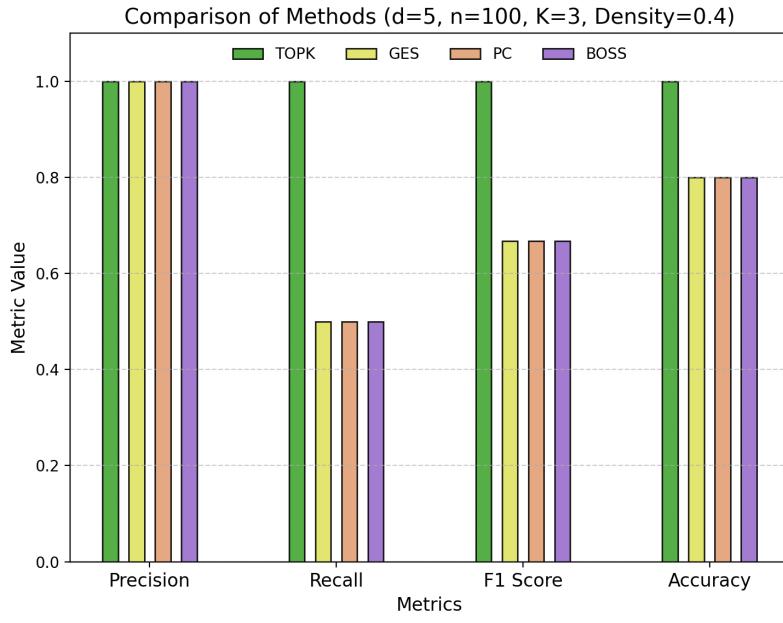


Figure 15: Comparison of predicted structures from other methods versus our Top-K approach.

Figure 15 illustrates that all other methods predict the same structure, which corresponds to the top-1 structure identified by our method. This shows alignment of the top-1 candidate from our approach with the outputs of existing methods.

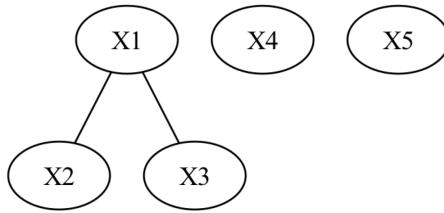


Figure 16: Output structure of the GES, PC, and BOSS algorithms.

In Figure 16, we present the output structure obtained from standard algorithms such as GES, PC, and BOSS. Notably, all three methods recover the same structure, capturing the edges between nodes 1–2 and 1–3 correctly. However, the additional edges between nodes 2–3 and 2–4 present in the true graph are not identified by these methods. This limitation underscores the potential of our approach to recover more subtle structural dependencies that may be overlooked by traditional methods.

By setting $T^* = 0.56$ in our method, we obtain the Top-3 graph probabilities as [0.61, 0.22, 0.17].

The corresponding edge uncertainty matrix is:

$$\begin{bmatrix} 0 & 1.00 & 1.00 & 0 & 0 \\ 0 & 0 & 0.39 & 0.22 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

We observe that the top-1 structure, as well as the predictions from other baseline methods, include only the skeletal edges between nodes (1, 2) and (1, 3). However, our top-2 structure additionally recovers the edges (2, 3) and (2, 4), which are ignored by other approaches. Notably, our uncertainty quantification assigns a 22% probability to the top-2 graph—corresponding to the true structure—and the edge uncertainty matrix shows that these additional edges have non-negligible probabilities (0.39 and 0.22, respectively), suggesting they cannot be completely dismissed. This highlights the value of our approach in not only identifying high-confidence edges but also uncovering plausible yet subtle connections that other methods might overlook. This helps achieve higher recall, sometimes at the cost of lower precision, but overall the F1 score still dominates, demonstrating the robustness of our method.

B TEMPERATURE SCALING FOR MEANINGFUL UNCERTAINTY QUANTIFICATION

Proposition 1. Let $P = \{p_1, p_2, \dots, p_K\}$ be a discrete probability distribution over K outcomes, ordered such that $p_1 \geq p_2 \geq \dots \geq p_K = \tau > 0$, and $\sum_{i=1}^K p_i = 1$. Let

$$H(P) := - \sum_{i=1}^K p_i \ln p_i, \quad H_{\min}(K, \tau) := \min_{\substack{P: p_K = \tau \\ p_1 \geq \dots \geq p_K}} H(P).$$

Then, choosing

$$\tau = \frac{1}{2K}$$

ensures the entropy satisfies the lower bound

$$\frac{H_{\min}(K, \tau)}{\ln K} \geq \frac{1}{2},$$

i.e., at least half of the maximum possible entropy.

Proof. Consider the extremal distribution where one outcome has most of the mass and the remaining $K - 1$ outcomes each have mass τ :

$$p_1 = 1 - (K - 1)\tau, \quad p_2 = \dots = p_K = \tau.$$

756 Substituting $\tau = \frac{1}{cK}$, we have
 757

$$758 \quad p_1 = 1 - \frac{K-1}{cK} = 1 - \frac{1}{c} + \frac{1}{cK}.$$

760 As $K \rightarrow \infty$, $p_1 \rightarrow 1 - \frac{1}{c}$, and the entropy becomes
 761

$$762 \quad H_{\min}(K, \tau) = -p_1 \ln p_1 - (K-1)\tau \ln \tau \\ 763 \quad = -p_1 \ln p_1 + \frac{K-1}{cK}(\ln c + \ln K).$$

765 Dividing by $\ln K$,

$$767 \quad \frac{H_{\min}(K, \tau)}{\ln K} = \frac{-p_1 \ln p_1}{\ln K} + \frac{K-1}{cK} \left(1 + \frac{\ln c}{\ln K}\right).$$

769 Taking $K \rightarrow \infty$, this ratio approaches $\frac{1}{c}$. Thus, setting $c = 2$ gives
 770

$$771 \quad \lim_{K \rightarrow \infty} \frac{H_{\min}(K, \tau)}{\ln K} = \frac{1}{2},$$

774 ensuring the entropy is at least half the maximum possible $\ln K$. This makes $\tau = \frac{1}{2K}$ a natural and
 775 meaningful choice for preserving uncertainty. \square
 776

777 **Temperature Scaling for Graph Uncertainty.** Let $\{\hat{\mathcal{G}}_1, \dots, \hat{\mathcal{G}}_K\}$ denote the top- K candidate
 778 graph structures corresponding to the solutions $\{\tilde{B}(\lambda_1), \dots, \tilde{B}(\lambda_K)\}$, ordered by increasing BIC
 779 score. We define a temperature-scaled probability distribution over these graphs as:

$$781 \quad P(\hat{\mathcal{G}}_k) = \frac{\exp\left(-\frac{1}{2T} \cdot \text{BIC}(\tilde{B}(\lambda_k))\right)}{\sum_{j=1}^K \exp\left(-\frac{1}{2T} \cdot \text{BIC}(\tilde{B}(\lambda_j))\right)}.$$

785 This softmax-like formulation assigns higher weight to graphs with lower BIC while introducing a
 786 tunable smoothness via the temperature parameter $T > 0$. As $T \rightarrow 0$, the distribution concentrates
 787 on the best-scoring graph; as $T \rightarrow \infty$, it approaches a uniform distribution.

788 To ensure meaningful contribution from not only the top-1 graph but also the remaining candidates in
 789 the Top- K set, we enforce a minimal entropy constraint. Specifically, we require the entropy of this
 790 distribution to be at least half of the maximum possible entropy, i.e.,

$$791 \quad H(P) \geq \frac{1}{2} \ln K.$$

793 This choice ensures that the distribution is not overly peaked and that even the K -th graph meaning-
 794 fully contributes to the uncertainty quantification process. It provides a balance between confidence
 795 in top-ranked structures and robustness through structural diversity.

797 Using Proposition 1, we achieve this entropy lower bound by setting the tail probability $\tau = P(\hat{\mathcal{G}}_K)$
 798 to

$$799 \quad \tau = \frac{1}{2K},$$

801 which, in turn, guides the choice of the temperature T via calibration on the BIC values. An empirical
 802 plot of the constant (optimal) c in $\tau = \frac{1}{cK}$ across different values of K is shown in Figure 17, from
 803 which we observe optimal c values between 2 and 4.

804 C DETAILED DISCUSSION REGARDING ALGORITHM

805 C.1 CHOOSING K

808 Choosing the number of Top- K structures to retain along the solution path is important for identifying
 809 a diverse and relevant set of candidate graphs.

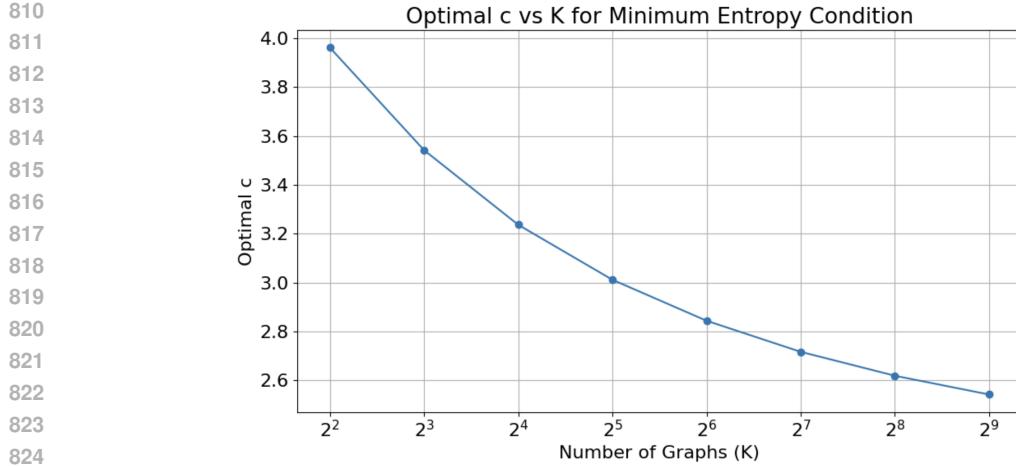


Figure 17: Optimal values of c vs. number of graphs K to achieve desired entropy floor.

- **Effect of Sample Size:** As the sample size increases, the confidence in estimated structures also increases. Therefore, a smaller K may be sufficient, since fewer candidates are likely needed to cover the meaningful structural variations.
- **Effect of Number of Nodes:** As the number of nodes grows, the number of possible edges increases rapidly. A larger K may help to capture a broader set of structural changes and dependencies that emerge due to the increased model complexity.
- **Effect of Graph Density:** Denser graphs tend to have more transitions along the solution path. Increasing K in such cases can help capture more of these edges.
- **Dependence on ϵ :** The value of K does not directly depend on the grid size ϵ , as long as ϵ is sufficiently small to capture all critical points along the solution path. Once the solution path is well-resolved, increasing grid resolution further should not affect the necessary value of K .

C.2 CHOOSING ϵ

The parameter ϵ controls the granularity of the λ -grid along the regularization path. It determines how finely the solution path is sampled and how many structural changes can be detected.

- **Effect of Sample Size:** While more samples improve the accuracy of the estimated edge weights and the resulting structures, they do not directly affect the number of critical points along the solution path. Therefore, the choice of ϵ is largely independent of the sample size.
- **Effect of Number of Nodes and Density:** As the number of nodes or the density of the graph increases, more structural changes are likely to occur along the path. In such settings, a smaller ϵ (i.e., finer grid) may be necessary to accurately capture the critical points where these changes happen.

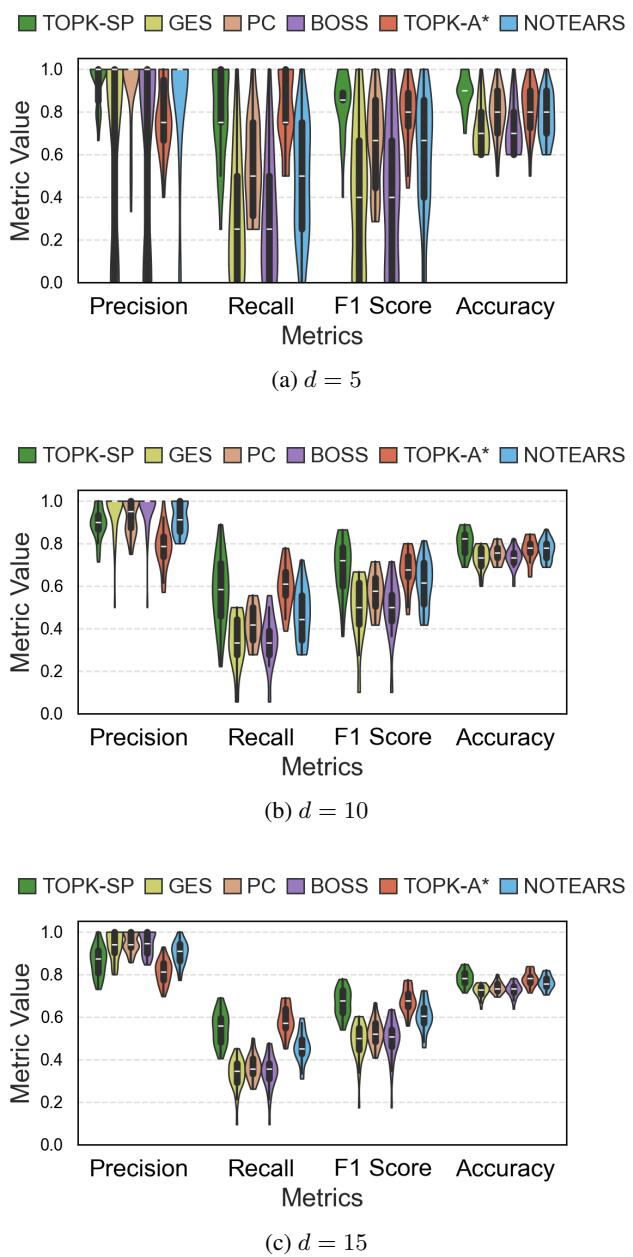
C.3 LIMITATION AND POTENTIAL SOLUTION

While our algorithm excels in recall and F1 score by capturing a wide range of plausible edges, this often comes at the cost of lower precision for a higher number of variables. In contrast, other structure learning algorithms may achieve higher precision but lower recall. This observation motivates a potential hybrid approach: combining the high-precision edges identified by alternative methods with the high-recall structures discovered through our Top-K solution path method. Such a hybrid model could yield better overall structural accuracy and is a promising direction for future work.

864 **D COMPARISON WITH NOTEARS**
865

866
867 We extend our comparison of Top-K Solution Path method against previous algorithms and the
868 continuous optimization method NOTEARS Zheng et al. (2018). We fix the sample size to $n = 100$,
869 the number of candidate structures to $K = 5$, and the edge density to $\rho = 0.4$. We perform 30 runs
870 for each of these experiments. All methods are evaluated on synthetic dataset as before by varying
871 the number of nodes $d \in \{5, 10, 15\}$. For NOTEARS, we use a weight threshold of 0.1 to prune
872 weak edges as the least magnitude of any edge weight is 0.1 in our setting. We also set the ℓ_1 penalty
873 for NOTEARS as $\lambda_1 = 0.03$, which yielded the best performance across all settings.
874

875 As shown in Figure 18, we can see that our algorithm performs relatively well to all the other
876 algorithms in these settings on Recall, F1 score and accuracy.
877

Figure 18: Comparison of all methods (including NOTEARS) as the number of variables d varies.

918 E ALGORITHM
919
920
921
922
923

924	Symbol	Description
926	$x \in \mathbb{R}^{d \times n}$	observational data matrix (d variables, n samples)
927	ϵ	solution path grid size
928	τ	edge binarization threshold
929	δ	active-set tolerance
930	α	acyclicity penalty parameter
931	K	number of top structures to return
932	λ_t	Lasso sparsity regularization parameter at step t
933	$B(\lambda_t)$	the adjacency weight matrix at λ_t
934	\mathcal{A}_t	active set of indices updated at step t
935	Λ	set of critical points where edges appear/disappear
936	Λ_K	subset of Λ corresponding to the Top-K graphs with lowest BIC
937	$\hat{\mathcal{G}}_k$	re-estimated weighted adjacency matrix of the k -th top graph
938	$\hat{A}(\lambda)$	binary adjacency matrix at λ
939	$\tilde{B}(\lambda)$	adjacency matrix weights after regression at λ
940	T	temperature parameter
941	$P(\hat{\mathcal{G}}_k)$	probability of the k -th graph (for uncertainty quantification)
942	$P_{\text{edge}}(i, j)$	probability of edge (i, j) appearing across Top-K graphs

943
944
945
946
947 **Algorithm 1: Top-K Structures with Solution Path Method**

948 **Input:** $x \in \mathbb{R}^{d \times n}, \epsilon, \tau, \delta, \alpha, K$
 949 **Output:** $\{\hat{\mathcal{G}}_1, \dots, \hat{\mathcal{G}}_K\}$
 950 $t \leftarrow 0, \lambda_0 \leftarrow \lambda_{\max} = \max_{i \neq j} \left| \frac{\partial \mathcal{L}(B; x)}{\partial B_{ij}} \right|_{B=0}, \quad \mathcal{A}_0 \leftarrow \arg \max_{i \neq j} \left| \frac{\partial \mathcal{L}(B; x)}{\partial B_{ij}} \right|_{B=0}$
 951 **while** $\lambda_t > \epsilon$ **do**
 952 $\lambda_{t+1} \leftarrow \lambda_t - \epsilon$
 953 $B(\lambda_{t+1}) \leftarrow \text{GradientStep}(B(\lambda_t), \mathcal{A}_t, x, \alpha, \lambda_{t+1})$
 954 $\mathcal{A}_{t+1} \leftarrow \text{UpdateActiveSet}(\mathcal{A}_t, B(\lambda_{t+1}), x, \delta, \lambda_{t+1})$
 955 $t \leftarrow t + 1$
 956 $\Lambda \leftarrow \{\lambda \mid B_{ij}(\lambda) \text{ changes from zero to nonzero or vice versa}\}$
 957 **foreach** $\lambda \in \Lambda$ **do**
 958 $\hat{A}(\lambda) \leftarrow (|B(\lambda)| \geq \tau)$
 959 $\tilde{B}(\lambda) \leftarrow \text{OLSRegression}(x, \hat{A}(\lambda))$
 960 $\text{BIC}(\lambda) \leftarrow 2n \cdot \mathcal{L}(\tilde{B}(\lambda); x) + \log(n) \cdot |\tilde{B}(\lambda)|$
 961 $\Lambda_K \leftarrow \text{Top-}K \text{ values of } \lambda \text{ with lowest BIC}$
 962 $\{\hat{\mathcal{G}}_1, \dots, \hat{\mathcal{G}}_K\} \leftarrow \{\tilde{B}(\lambda) \mid \lambda \in \Lambda_K\}$

963
964
965 **Algorithm 2: GradientStep**

966 **Input:** $B(\lambda_t), \mathcal{A}_t, x, \alpha, \lambda_{t+1}$
 967 **Output:** $B(\lambda_{t+1})$
 968 $B^{(0)} \leftarrow B(\lambda_t)$
 969 **for** $s = 0$ **to** $g - 1$ **do**
 970 $B^{(s+1)} = B^{(s)} - \eta_t [\nabla_B \mathcal{L}(B^{(s)}; x) + \lambda_{t+1} \text{sgn}(B^{(s)}) + \alpha \nabla_B h(B^{(s)})]_{\mathcal{A}_t}$
 971 **return** $B(\lambda_{t+1}) \leftarrow B^{(g)}$

Algorithm 3: UpdateActiveSet

Input: $\mathcal{A}_t, B(\lambda_{t+1}), x, \delta, \lambda_{t+1}$

Output: \mathcal{A}_{t+1}

$\mathcal{A}'_t \leftarrow \{(i, j) \in \mathcal{A}_t \mid |B_{ij}(\lambda_{t+1})| \geq \delta\}$

for all (i, j) with $i \neq j$ **and** $(i, j) \notin \mathcal{A}'_t$ **do**

$G_{ij} \leftarrow \left| \frac{\partial \mathcal{L}(B; x)}{\partial B_{ij}} \right|_{B=B(\lambda_{t+1})}$

if $G_{ij} \geq \lambda_{t+1}$ **then**

$\mathcal{A}'_t \leftarrow \mathcal{A}'_t \cup \{(i, j)\}$

return $\mathcal{A}_{t+1} \leftarrow \mathcal{A}'_t$

Algorithm 4: Uncertainty Quantification from Top-K Graphs

```

Input:  $\{\widehat{\mathcal{G}}_1, \dots, \widehat{\mathcal{G}}_K\}, T$ 
Output:  $\{P(\widehat{\mathcal{G}}_k)\}_{k=1}^K, \{P_{\text{edge}}(i, j)\}_{i,j=1}^d$ 
for  $k = 1$  to  $K$  do
   $w_k \leftarrow \exp\left(-\frac{1}{2T} \text{BIC}(\widehat{\mathcal{G}}_k)\right)$ 
for  $k = 1$  to  $K$  do
   $P(\widehat{\mathcal{G}}_k) \leftarrow \frac{w_k}{\sum_{j=1}^K w_j}$ 
for all  $(i, j)$  with  $1 \leq i, j \leq d$  do
   $P_{\text{edge}}(i, j) \leftarrow \sum_{k=1}^K P(\widehat{\mathcal{G}}_k) \mathbb{I}\left((i, j) \in \widehat{\mathcal{G}}_k\right)$ 

```

F HYPERPARAMETER SENSITIVITY ANALYSIS

F.1 EPSILON (ϵ)

We vary the value of ϵ , which determines the discretization of the solution path through the ratio λ_{\max}/ϵ . In our experiments, we test grid resolutions corresponding to $\lambda_{\max}/\epsilon \in \{5, 10, 20, 40\}$. The remaining settings are fixed to $d = 10$, $n = 100$, $\rho = 0.5$, $K = 3$, and `num_runs` = 30.

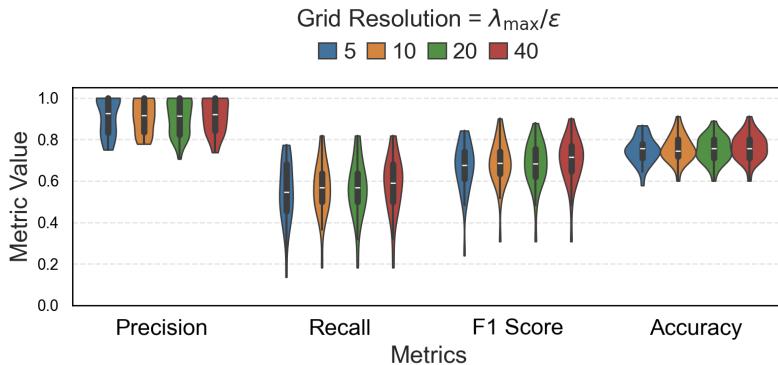
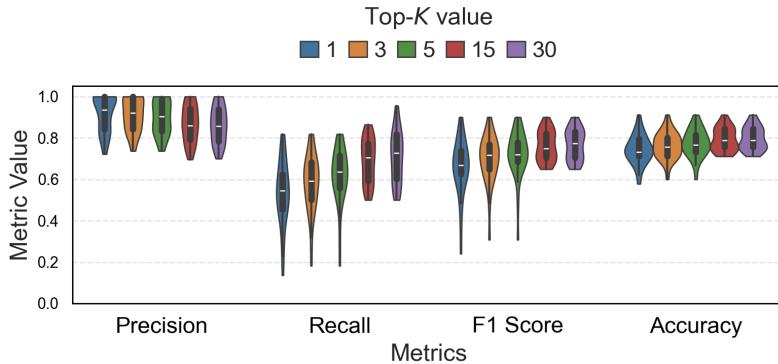


Figure 19: Evaluation metrics for different values of ϵ .

As shown in Figure 19, smaller values of λ_{\max}/ϵ correspond to coarser grids, and we observe that the performance improves slightly as the grid is refined. Beyond a moderate resolution, the metrics saturate, indicating that further refinement offers little additional benefit.

1026
1027F.2 TOP-K VALUE (K)1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044

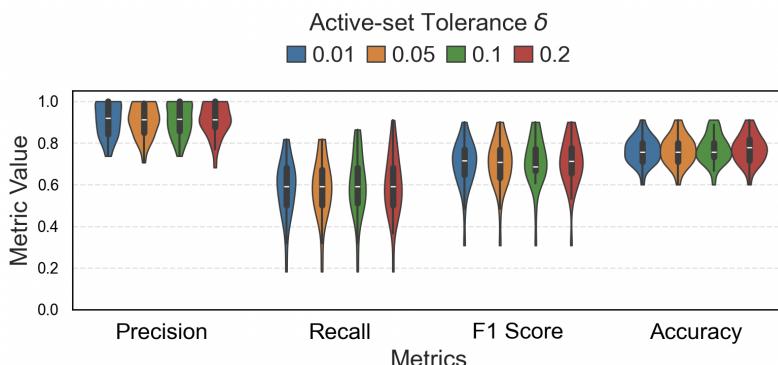
We study the effect of varying the Top- K parameter K , which determines how many of the highest-scoring structures are considered. In our experiments, we evaluate $K \in \{1, 3, 5, 15, 30\}$. All other settings are kept fixed at $d = 10$, $n = 100$, $\rho = 0.5$, and $\epsilon = \lambda_{\max}/40$, with `num_runs` = 30.

Figure 20: Evaluation metrics for different values of K .1045
1046
1047
1048
1049
1050

As shown in Figure 20, as K increases, the evaluation metrics such as F1-score and accuracy tend to improve, but the gains eventually saturate beyond a certain value.

1051
1052
1053F.3 DELTA (δ)1054
1055
1056
1057
1058

We analyze the impact of the parameter δ , which controls the removal of elements from the active set during the Top-K Solution Path optimization. In our experiments, we fix the grid resolution $\lambda_{\max}/\epsilon = 40$ and vary $\delta \in \{0.01, 0.05, 0.1, 0.2\}$. The remaining settings are $d = 10$, $n = 100$, $\rho = 0.5$, $K = 3$, and `num_runs` = 30.

1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074Figure 21: Evaluation metrics for different values of δ .1075
1076
1077
1078
1079

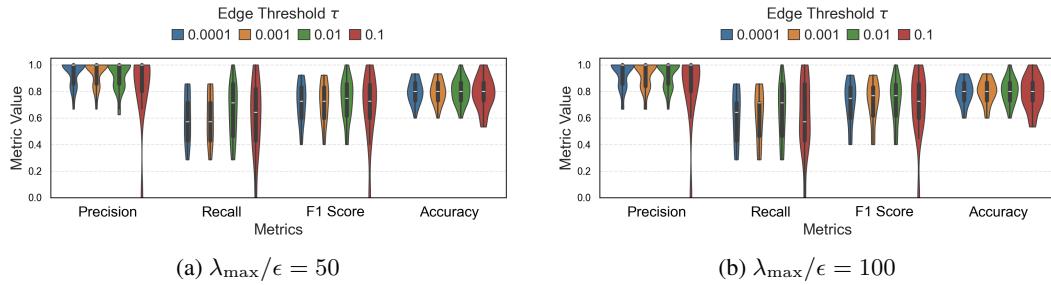
As shown in Figure 21, the performance metrics are largely similar across the tested δ values. This is expected because, with a sufficiently fine grid, the elements that have significant impact on the objective remain in the active set regardless of δ . After the active set removal step that depends on δ , edges that have a sufficient gradient are added back to the active set, ensuring their contribution. Consequently, setting $\delta = 0.01$ provides representative performance without the need to explore multiple values.

1080
1081F.4 TAU (τ)1082
1083
1084
1085

We now study the effect of the threshold parameter τ , which determines when an edge is considered sufficiently active along the solution path. In this experiment, we vary $\tau \in \{0.0001, 0.001, 0.01, 0.1\}$ under two grid resolutions, corresponding to $\lambda_{\max}/\epsilon \in \{50, 100\}$. The remaining settings are fixed to $d = 6, n = 100, \rho = 0.5, K = 3$, and `num_runs` = 30.

1086

1087



1095

1096

1097

1098

1099

As shown in Figure 22, the effect of τ largely depends on the grid resolution. For the coarser grid ($\lambda_{\max}/\epsilon = 50$), a threshold around $\tau = 0.01$ performs well, since weight updates between solution points are relatively large. For the finer grid ($\lambda_{\max}/\epsilon = 100$), these updates become much smaller, making a lower threshold such as $\tau = 0.001$ more appropriate for capturing meaningful changes. Overall, τ should inversely scale with the grid resolution (finer grids require smaller thresholds) reflecting the fact that the smallest detectable coefficient increments shrink as the solution path becomes more finely discretized.

1100

1101

F.5 ALPHA (α)

1102

1103

1104

1105

1106

1107

1108

1109

1110

1111

1112

We evaluate the sensitivity of the acyclicity penalty weight α by varying $\alpha \in \{0.1, 1, 5, 10, 50\}$ while keeping the remaining settings fixed to $d = 8, n = 100, \rho = 0.6, K = 3, \text{num_runs} = 30$, and $\epsilon = \lambda_{\max}/50$.

1113

1114

1115

1116

1117

1118

1119

1120

1121

1122

1123

1124

1125

1126

1127

1128

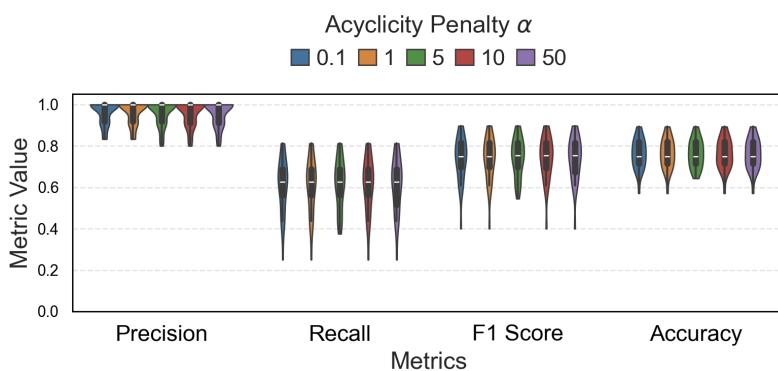
1129

1130

1131

1132

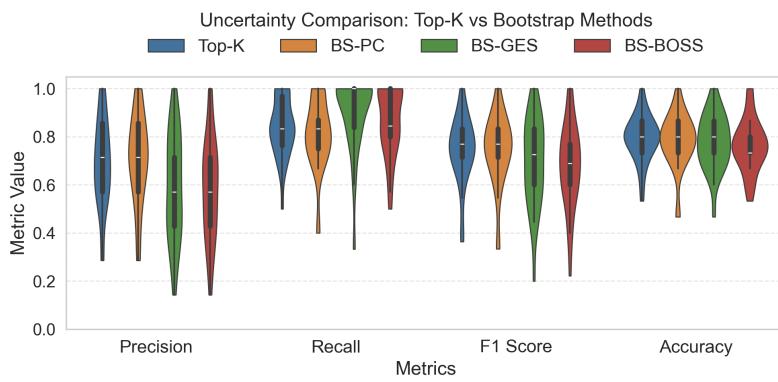
1133

Figure 23: Evaluation metrics for different values of α .

1134

As shown in Figure 23, across the tested values the performance remains largely stable, indicating that the method is not highly sensitive to the choice of α in this range. However, $\alpha = 5$ exhibits slightly more favorable behavior in terms of dispersion in the violin plots, aligning with the commonly used setting in prior work such as GOLEM Ng et al. (2020). We therefore adopt $\alpha = 5$ as a reasonable and robust default.

1134 **G UNCERTAINTY QUANTIFICATION COMPARISON**
1135

1136 We compare the edge-level uncertainty estimates produced by our Top-K solution path algorithm
1137 against a standard bootstrap approach applied to three baseline algorithms: PC, GES, and BOSS. In
1138 our experiments, we use a synthetic dataset with $d = 6$ nodes, $n = 100$ samples, $K = 5$, $\rho = 0.5$,
1139 and a grid resolution $\epsilon = \lambda_{\max}/40$. The comparison is performed over 30 independent runs. For
1140 the Top-K approach, uncertainty quantification is computed by taking the skeleton of each top- K
1141 graph, weighting it by the probability of the corresponding graph, and applying a threshold of 0.2
1142 to determine edge presence. For the bootstrap-based baselines (PC, GES, BOSS), uncertainty is
1143 estimated by generating 50 resampled datasets, computing the skeleton for each run, averaging these
1144 skeletons across bootstrap samples, and then applying the same 0.2 threshold.
1145

1158 **Figure 24: Comparison of evaluation metrics across Top-K SP and bootstrap-based uncertainty**
1159 **quantification approaches (PC, GES, BOSS).**
1160

1162 As shown in Figure 24, the Top-K algorithm achieves strong F1 scores and overall performance that
1163 is comparable to the bootstrap-based PC, GES, and BOSS methods. Importantly, the Top-K approach
1164 requires only a single run of the algorithm, whereas the bootstrap method incurs additional compu-
1165 tational cost proportional to the number of resampled datasets. Consequently, any computational
1166 limitations inherent to the base algorithms are amplified when performing bootstrap resampling. In
1167 contrast, Top-K provides efficient and theoretically grounded uncertainty estimates without repeated
1168 evaluations, making it a computationally attractive alternative.
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187