An AutoML-Powered Architecture for Causal Discovery

Konstantina Biza KBIZA@CSD.UOC.GR
Computer Science Department, University of Crete
Institute of Applied and Computational Mathematics, FORTH
Heraklion, Creece

Sofia Triantafillou SOF.TRIANTAFILLOU@UOC.GR
Department of Mathematics and Applied Mathematics, University of Crete
Institute of Applied and Computational Mathematics, FORTH
Heraklion, Creece

Ioannis Tsamardinos TSAMARD@CSD.UOC.GR
Computer Science Department, University of Crete
Institute of Applied and Computational Mathematics, FORTH
Heraklion, Creece

Editors: Gustau Camps-Valls, Manuele Leonelli and Gherardo Varando

Abstract

Causal discovery is fundamental across a wide range of scientific domains. While numerous algorithms have been proposed in the literature, they often rely on methods that suffer from limitations such as strong parametric assumptions or restrictions on variable types, making them ill-suited for application to complex, real-world datasets. Recent advances in Automated Machine Learning (AutoML) leverage and tune flexible model architectures to process complex non-linear relationships and mixed data types, while automating data preprocessing for real-world applications. In this work, we design an end-to-end AutoML-based architecture for Causal Discovery (MLCD) that fully leverages these capabilities. The proposed framework consists of three main components: (i) selecting the optimal configuration for each node—including data transformations, feature sets, and predictive models and hyperparameters; (ii) constructing the graph skeleton by taking the union of the selected feature sets; and (iii) applying conditional independence tests that employ the selected optimal configurations to compute the p-values. To account for the presence of latent common causes in real-world scenarios, we formulate MLCD for causally insufficient systems, though it can easily adapt to causally sufficient environments as well. We evaluate our method on both synthetic and real-world datasets. Notably, MLCD processes raw data directly, bypassing the preprocessing pipelines required by most causal discovery methods.
Keywords: AutoML, Causal Discovery, Real-world data

1. Introduction

Causal discovery aims to identify causal relationships among variables of interest. While numerous causal discovery methods for observational data have been proposed, they frequently rely on strong assumptions, such as causal sufficiency. In addition, many existing approaches depend on conditional independence tests or scoring functions that may exhibit further limitations, particularly regarding the processing of mixed data types and high computational complexity. To date, the causal discovery methods that simultaneously handle both causal insufficiency and mixed data types remain sparse. Furthermore, applying causal

© 2026 K. Biza, S. Triantafillou & I. Tsamardinos.

discovery to real-world data remains challenging, often requiring manual data preprocessing. While practices like variable discretization and missing value imputation are common, selecting the right combination of these techniques is critical to ensure a reliable output.

Automated Machine Learning (AutoML) focuses on optimizing the predictive modeling pipeline for a target variable. It automates key processes—including data preprocessing, feature engineering, and predictive modeling—and optimizes the selection and combination of these components. By incorporating modern machine learning best practices, AutoML provides a systematic approach to handling real-world data without requiring extensive manual expertise. Due to growing interest in the field, several frameworks are available, offering researchers versatile solutions according to their specific problems.

Despite the success of AutoML in predictive tasks, its potential for causal discovery remains largely unexplored. In this work, we propose an automated system designed to overcome the practical challenges in causal discovery. We mainly focus on the discovery of latent structures and mixed datasets using observational data, since they represent the majority of real-world cases. We propose an end-to-end AutoML-based framework for Causal Discovery (MLCD), which makes the following contributions to the field: (i) It handles diverse data types (continuous, categorical, and mixed) (ii) It accounts for causally insufficient systems. (iii) It automatically detects data types and executes necessary preprocessing steps, including imputation and feature engineering. (iv) It uses an AutoML framework to handle the entire pipeline, including feature selection for skeleton discovery and independence testing for the final graph construction. (v) It maintains computational efficiency on large samples sizes, where other non-parametric methods typically struggle. (vi) It optimizes the causal learning procedure for each node separately, adapting to different functional relationships.

The rest of this article is organized as follows. In Section 2, we present the terminology in causality. In Section 3, we review causal discovery algorithms along with conditional independence tests and scoring functions, focusing on their applicability to mixed data with latent common causes. In Section 4, we provide a motivating example, and in Section 5, we detail our proposed method. We comparatively evaluate our approach on synthetic data in Section 6, and real-world datasets in Section 7. Finally, in Section 8, we discuss the method's limitations, summarize its key contributions and propose future research directions. The code is publicly available on our GitHub page (https://github.com/mensxmachina).

2. Preliminaries

Two types of graphs are commonly used to represent the causal relationships, the Directed Acyclic Graph (DAG) (Pearl, 2009) and the Maximal Ancestral Graph (MAG) (Richardson and Spirtes, 2002). A DAG contains only directed edges, while a MAG contains both directed and bi-directed edges. MAGs admit and represent latent confounding; they are used when common causes (confounders) are not measured. We refer to such cases as causally insufficient systems. Two nodes are called adjacent if there is an edge between them. A node Y is called a collider in a triplet of vertices $\langle X, Y, Z \rangle$ if Y is connected with X and Z, and both edges have arrowheads at Y. If X and Z are not adjacent, then Y is an unshielded collider. Both DAGs and MAGs capture the independence relationships between the observed variables (Pearl, 2009; Spirtes et al., 2001; Richardson and Spirtes, 2002). It is common that different graphs over the same set of variables can entail the

same conditional independencies. These graphs are called Markov Equivalent graphs. A Completed Partially Directed Acyclic Graph (CPDAG) (Chickering, 2002) represents a class of Markov Equivalent DAGs. Similarly, a Partial Ancestral Graph (PAG) (Spirtes et al., 2001) represents a class of Markov Equivalent MAGs. A PAG is often the result of a causal discovery algorithm that allows for causal insufficiency.

A causal graph induces a Markov boundary $\text{Mb}(X)$ for each node X of the graph (Pearl, 2009). The $\text{Mb}(X)$ in a DAG or MAG consists of its adjacent nodes and the nodes that are reachable through a collider path (Borboudakis and Tsamardinos, 2019; Pellet and Elisseeff, 2008b). The Markov boundary is unique for distributions faithful to the graph and it is invariant among all graphs in the same Markov equivalence class. Intuitively, faithfulness means that all and only the conditional independencies present in the distribution are encoded by the graph's structure (see Spirtes et al. (2001), p. 31). Two important properties of the $\text{Mb}(X)$ are that it is the minimal set that renders X conditionally independent of any other node, and it contains all variables that are dependent on X conditioned on all other variables. These properties directly link the Markov boundary with the feature selection problem. Feature selection identifies a set of predictors that are most informative about the outcome of interest. A strongly relevant feature V is a dependent variable with the outcome X given all other variables (Kohavi and John, 1997). In general, it has been shown that the $\text{Mb}(X)$ is the solution to the feature selection problem when trying to predict X under certain broad conditions (Tsamardinos and Aliferis, 2003).

3. Related work

3.1. Causal discovery algorithms

Numerous causal discovery algorithms have been proposed in the literature that assume causal sufficiency. Most of them are primarily optimized for continuous data, and significantly fewer options exist for mixed data types. We refer the reader to (Glymour et al., 2019; Vowels et al., 2022; Zanga et al., 2022) for a comprehensive review. The landscape for causally insufficient systems, where latent confounders may be present, is even more specialized, and outlined briefly below. The FCI algorithm (Spirtes et al., 2001) is the most prominent **constraint-based** method, performing a series of conditional independence (CI) tests to resolve causal relations. Variations of FCI include FCI-stable (Colombo and Maathuis, 2014), CFCI (Ramsey et al., 2006), RFCI (Colombo et al., 2012), FCI-Max, and MGM-FCI (Raghu et al., 2018). L-MARVEL (Akbari et al., 2021) is also a constraint-based approach designed to learn causal structures recursively. **Score-based** algorithms utilize scoring functions to search the space of possible graphs for the model that best fits the observed data. The SALAD (Ng et al., 2024) framework and the BB approach by Rantanen et al. (2021) are two such examples, both tailored for the linear continuous setting. **Hybrid** algorithms combine the above strategies, typically applying CI tests to prune the search space and scoring functions to orient edges. Notable examples include the GFCI (Ogarrio et al., 2016) and its variants (Boss-FCI, GRaSP-FCI, and FCI Targeted-Testing (Ramsey et al., 2025)), as well as the M3HC (Tsirlis et al., 2018), BCCD (Claassen and Heskes, 2012; Sokolova et al., 2014), GSPo (Bernstein et al., 2020), and dcFCI (Ribeiro and Heider, 2026). Causal discovery methods based on **Functional Causal Models** (FCM) include the RCD algorithm (Maeda and Shimizu, 2020), designed for linear non-Gaussian

data; CAM-UV (Maeda and Shimizu, 2021), which addresses nonlinear cases; and ABIC LiNGAM (Morinishi and Shimizu, 2025), which utilizes continuous optimization assuming non-Gaussian data. Causal discovery utilizing **neural networks** (NN) has garnered significant interest. Prominent methods that assume causal insufficiency include NOCADILAC (Kaltenpoth and Vreeken, 2023), which utilizes a variational autoencoder, and DCD (Bhattacharya et al., 2021), which frames causal discovery as a continuous optimization problem. However, both are primarily designed for continuous data. **Score-matching**-based methods use the gradient of the log-density, with AdaScore (Montagna et al., 2025) being a prominent example designed for continuous variables. The applicability of all the above methods to mixed and non-linear data depends heavily on either their internal architecture (e.g., NN- and FCM-based models are often inherently restricted to continuous variables) or the selection of a suitable CI test or scoring function. In Table 1, we report the applicability of the aforementioned algorithms across different data types, based on their currently available implementations. A missing checkmark indicates that application to that data type is restricted by either theoretical constraints or a lack of software support.

Table 1: Causal Discovery Algorithms under Causal Insufficiency

	Algorithm	Con.	Mixed
Con.	FCI & variants	✓	✓
	L-MARVEL	✓	–
Sc.	SALAD	✓	–
	BB	✓	–
Hybrid	GFCI & variants	✓	✓
	BCCD	✓	✓
	dcFCI	✓	✓
	M3HC	✓	–
	GSPo	✓	–

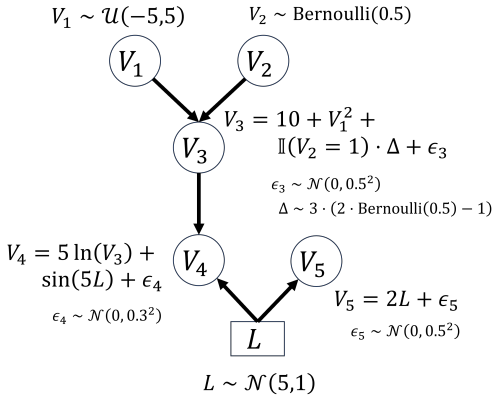
	Algorithm	Con.	Mixed
FCM	RCD	✓	–
	CAM-UV	✓	–
	ABIC LiNGAM	✓	–
NN	NOCADILAC	✓	–
	DCD	✓	–
SM	AdaScore	✓	–

3.2. Conditional Independence Tests and Scores

Conditional independence tests and scoring functions are the primary internal components of constraint-based, score-based, and hybrid causal discovery algorithms. Several methods have been proposed in the literature; however, the majority are suitable only for purely continuous variables (Li and Fan, 2020). The available choices for mixed data are considerably more limited. Classical methods in CI testing and scoring are **parametric** methods that rely on specific distributional assumptions. The Conditional Gaussian (CG) and Degenerate Gaussian (DG) scores (Andrews et al., 2018, 2019) approximate the BIC score for mixed data under Gaussian assumptions, while the Mixed Variable Polynomial (MVP) score is designed to accommodate non-linear relationships (Andrews et al., 2018). These scores can also be utilized for CI testing via Likelihood Ratio Tests (LRT) (e.g., CG-LRT, DG-LRT). Other Likelihood Ratio Tests use regression models and compare their goodness-of-fit (Tsagris et al., 2018). While the above methods are fast, their distributional assumptions are often violated in real-world data. **Mutual-Information**-based tests utilize (conditional) mutual

information as a measure of dependence. The MI-CG is a parametric implementation of such a test (Scutari, 2010), appropriate for conditional Gaussian distributions. Sokolova et al. (2014) proposed to approximate mutual information in the BIC score using Spearman rank correlation, assuming monotonic relationships. Non-parametric methods in this category use nearest neighbors to compute mutual information. This group includes the CMIknnmixed (Popescu et al., 2025) and mCMIknn (Huegle et al., 2023) tests, which are versatile for capturing non-linearities, but scale poorly with sample size. **Kernel-based** methods map variables into high-dimensional feature spaces and compare their relationships using kernel functions. Examples include CI tests such as the Kernel Generalized Variance (KGV) (Bach and Jordan, 2002) and the Kernel Alignment Matrix approach by Handhayani and Cussens (2020), and the Generalized Score (Huang et al., 2018). Similar to nearest neighbors approaches, kernel-based methods can capture complex dependencies, but exhibit high computational complexity with respect to sample size. The **Residual–based** tests examine association on the residuals obtained when regressing Y on $\mathbf{Z}$ and X on $\mathbf{Z}$. The test by Ankan and Textor (2023) was initially proposed for categorical and ordinal variables. However, a recent implementation in `pgmpy` package (Pillai trace), extends the idea to support mixed data based on canonical correlations (Ankan and Textor, 2024).

Overall, many of the aforementioned CI tests and scoring functions have not yet been integrated into standard causal discovery frameworks that operate under causal insufficiency. Most of the algorithms in Table 1 rely on parametric tests and scores, such as CG-LRT and DG-LRT. Consequently, researchers need to manually re-engineer package architectures to pair advanced, non-parametric tests with algorithms for causal insufficiency. Practical software constraints further limit the pool of available methods. Specifically, open-source implementations for certain CI tests are unavailable (e.g., KGV), or they require an extensive hyperparameter search, such as the Kernel Alignment Matrix approach (Popescu et al., 2025). Similarly, the dcFCI algorithm presents software replication constraints, while the MVP-LRT test exhibits execution failures in its current implementation.



(*a*) True DAG

	Adj. Prec.	Adj. Rec.
CG-LRT	0.63 ± 0.03	0.53 ± 0.02
DG-LRT	0.70 ± 0.09	0.33 ± 0.04
Pillai	0.87 ± 0.07	0.30 ± 0.03
CMIknnmixed	0.96 ± 0.03	0.5 ± 0.0
MLCD	0.94 ± 0.03	1.0 ± 0.0

(*b*) Estimated PAG

Figure 1: MLCD successfully captures complex relationships missed by existing approaches.

4. Motivating Example

We consider a synthetic data-generating process defined by the DAG and the functional relationships depicted in Figure 1(*a*). We generate datasets of 5000 samples across 10 repetitions. We employ the FCI-stable algorithm paired with representative tests from each category: CG-LRT and DG-LRT from parametric approaches, Pillai from residual-based methods, and CMIknnmixed from non-parametric mutual information estimators. We leverage Tetrad's implementation for the parametric baselines and our custom implementation of FCI-stable for the remaining tests. To ensure a fair comparison across all evaluated frameworks, all continuous variables were standardized to have zero mean and unit variance. Due to the high computational cost of CMIknnmixed on large sample sizes (where a single test exceeds 7 min), we evaluate it using a subset of 1000 samples. This setup is designed as a stress test to highlight scenarios where these baselines typically struggle due to complex, non-linear functional forms. The parametric tests are generally limited by functional misspecification. The Pillai trace, which relies on canonical correlations, exhibits vulnerability to symmetric, non-monotonic relationships like X^2 and tends to miss the non-linear latent dependency induced by the periodic $\sin(5L)$ term, and the discrete variance shifts in V_3 introduced by the binary variable V_2. While CMIknnmixed can capture the X^2 relationship, both the periodic $\sin(5L)$ term in V_4 and the discrete shifts in V_3 complicate its local neighborhood density estimation. We report the average adjacency precision and recall of the estimated PAGs in Figure 1(*b*). We show that our proposed method, MLCD, outperforms the examined baselines. MLCD inherits the strong functional approximation advantages of AutoML. This allows it to adapt to these diverse non-linear and mixed-type data distributions, ensuring that no true causal edges are missed, while maintaining high precision.

5. AutoML-Powered Causal Discovery

Outline. We introduce MLCD, an end-to-end AutoML-based Causal Discovery framework. MLCD optimizes causal learning for each node independently. The main idea is to leverage recent advances from the AutoML community to optimize two critical steps in causal learning: (i) identify the initial graph skeleton and (ii) perform conditional independence testing to obtain the final causal structure. Our framework consists of two phases: the **Skeleton Construction** and the **Graph Resolution** phase. In the first phase, an AutoML pipeline is applied for each node to determine its optimal configuration. It automatically handles data representation, selects the most informative features, chooses the best model, and tunes its hyperparameters. The selected feature set serves as the node's Markov boundary. The initial skeleton is then derived from the union of all identified Markov boundaries. In the second phase, a series of conditional independence tests are performed to remove any false positive edges and identify the unshielded colliders. For this step, we adapt our recently proposed CI test (Biza et al., 2026), which is based on comparing predictive performances. The selected models and hyperparameters from the first phase are now utilized to optimize the CI testing for each node pair. Finally, we apply the FCI orientation rules (Zhang, 2008) to the resulting skeleton to produce a PAG. The entire framework ensures the discovery process remains data-driven and model-agnostic. In Algorithm 1, we show the pseudocode of MLCD, which we further describe below.

Algorithm 1: MLCD

Input: Dataset D, significance level α
Output: PAG $\mathcal{G}$

1 $\mathcal{G} \leftarrow$ empty graph; $\quad \mathbf{C} \leftarrow \emptyset$
// Skeleton Construction
2 **for** $Y \in \mathbf{V}$ **do**
3 $\quad \widehat{\mathbf{Mb}}_Y, \widehat{C}_Y \leftarrow \texttt{AutoConfig}(D, Y, \alpha)$
4 $\quad$ Add link $X - Y$ in $\mathcal{G}$ for each $X \in \widehat{\mathbf{Mb}}_Y$
5 $\quad \mathbf{C} \leftarrow$ append $\widehat{C}_Y$
6 **end**
// Graph Resolution
7 $\mathcal{G} \leftarrow \texttt{FCI-stable}(\mathcal{G}, \alpha, \mathbf{C})$
8 $\quad$ via $pval \leftarrow \texttt{CIml}_{\text{config}}(D, X, Y, \mathbf{Z}, \widehat{C}_X, \widehat{C}_Y)$ for each pair $X - Y$ in $\mathcal{G}$
9 **return** PAG $\mathcal{G}$

Skeleton Construction. Several works have utilized Markov boundaries in causal structure learning (Margaritis and Thrun, 1999; Pellet and Elisseeff, 2008a,b; Chen et al., 2023; Mokhtarian et al., 2025). Constraint-based algorithms gain significant advantages in both statistical robustness and computational complexity when initialized with a moral rather than a complete graph. The moral graph is an undirected graph where each node is connected to its Markov boundary. This initialization reduces the total number of CI tests, and restricts the size of their conditioning sets (Pellet and Elisseeff, 2008a,b). In this step, we apply AutoML techniques to identify a potential superset of this moral graph. As detailed in lines 2–5 of Algorithm 1, an AutoML pipeline (AutoConfig) is fitted for each node Y to optimize the feature selection and predictive modeling, returning its estimated Markov boundary, $\widehat{\text{Mb}}(Y)$, alongside the optimal model configuration $\widehat{C}_Y$ (predictive model and hyperparameters). To construct the initial skeleton, we connect two nodes if either is detected in the other's estimated Markov boundary—employing an "OR-rule" rather than a strict symmetry constraint. This choice is crucial for handling nonlinear causal mechanisms and exogenous variables, where a parent may easily predict its child, but the reverse predictive task is significantly more challenging. Although this strategy may introduce false-positive edges, the subsequent CI testing phase is designed to prune these out.

AutoConfig procedure. We describe `AutoConfig` in Algorithm 2. We first partition the dataset into training and test sets, fitting the AutoML pipeline on the training data. The pipeline searches over various predictive models and their hyperparameters to identify the most robust fit for the target variable and capture the underlying relationships. The optimal configuration is then identified by evaluating out-of-sample performance on the test set. Since feature importance methods are common across modern AutoML frameworks, we leverage them to perform post-hoc feature selection. Specifically, we employ a permutation-based importance test on the test data. This process repeatedly shuffles the values of each feature and computes the drop in predictive performance. The method then evaluates the statistical significance of these results, returning an importance score and a corresponding

p-value for each feature. A feature is selected if its p-value is below the significance threshold α and its importance score is greater than zero.

Algorithm 2: `AutoConfig`(D, Y, α)

Input: Dataset D, target variable Y, significance level α
Output: Feature set $\widehat{Mb}_Y$, Configuration $\widehat{C}_Y$

$D_{\text{train}}, D_{\text{test}} \leftarrow$ split D with a 70/30 ratio
$\mathcal{P} \leftarrow$ `fit_AutoML`(Y, D_{train})
$\widehat{C}_Y \leftarrow$ `best_model_and_hyperparameters`$(\mathcal{P}, Y, D_{\text{test}})$
$\mathbf{FI}, \mathbf{pvals} \leftarrow$ `feature_importance`$(\mathcal{P}, Y, D_{\text{test}})$
$\widehat{Mb}_Y \leftarrow \{X \in \mathbf{V} \setminus \{Y\} \mid \mathbf{pvals}_X \leq \alpha \quad \text{and} \quad \mathbf{FI}_X > 0\}$
return $\widehat{Mb}_Y$, $\widehat{C}_Y$

Graph Resolution. Given the estimated skeleton G, we perform a series of conditional independence tests on its edges to identify false positive links and unshielded colliders. For this, we employ a modified version of our recently proposed conditional independence test, CIml (Biza et al., 2026). In summary, CIml builds on the idea that if two variables X and Y are conditionally dependent given a set $\mathbf{Z}$, then a predictive model using both X and $\mathbf{Z}$ to predict Y should outperform a model that uses only $\mathbf{Z}$. The null hypothesis, $X \perp\!\!\!\perp Y \mid \mathbf{Z}$, is tested by comparing the predictive performance of two models: model $\mathcal{M}_1 : Y \sim X, \mathbf{Z}$ and model $\mathcal{M}_2 : Y \sim X_p, \mathbf{Z}$, where X_p denotes a random permutation of X. Originally, k-fold cross-validation is applied to select the optimal predictive configuration (predictive model and hyperparameters) for $\mathcal{M}_1$, and the test statistic is calculated as the average difference in the predictive performance between $\mathcal{M}_1$ and $\mathcal{M}_2$ over the k folds. Since CI tests are inherently symmetric, i.e., $X \perp\!\!\!\perp Y \mid \mathbf{Z}$ is equivalent to $Y \perp\!\!\!\perp X \mid \mathbf{Z}$, while supervised learning is asymmetric, we report the combined p-value p_{mm}, which is computed as $p_{mm} = \min\{2\min(p_1, p_2), max(p_1, p_2)\}$ where $p_1 = \text{CIml}(X, Y, \mathbf{Z})$ and $p_2 = \text{CIml}(Y, X, \mathbf{Z})$. In our previous work, we showed that the symmetric CIml is calibrated and achieves higher power under an optimal configuration compared to a standard predictive model choice. However, executing an entire AutoML procedure for every single CI test within a causal discovery algorithm introduces prohibitive computational overhead. To alleviate these computational constraints, we do not re-execute the optimization loop for each individual CI test. Instead, we use the selected configurations $\widehat{C}_X$ and $\widehat{C}_Y$ for X and Y, as identified by the `AutoConfig` (line 8 in Algorithm 1). While these static configurations may not represent the optimal choices for every conditioning subset, maintaining a constant model configuration allows the constraint-based search to scale more efficiently. This computationally efficient adaptation is referred to as CImlconfig, which serves as the core CI test inside FCI-stable in Algorithm 1.

AutoML Integration. We utilize AutoGluon (Erickson et al., 2020) as the primary AutoML engine. While alternative frameworks like FLAML (Wang et al., 2021) or Auto-Sklearn (Feurer et al., 2015) provide comparable functionalities, AutoGluon was selected for the combination of the following features: **Automated Preprocessing and Task Detection.** AutoGluon autonomously manages feature engineering and identifies the task type (i.e., regression or classification). The latter is uniquely advantageous in causal discovery with mixed data. While most existing approaches require manual specification of the data

type for every variable, a process that is often burdensome and error-prone, this automation eliminates that requirement entirely. **Time Management.** Precise duration control ensures that the iterative fitting processes required by our framework remain computationally feasible. **Feature Selection.** AutoGluon supports feature importance calculations alongside p-value computation, derived from the optimal model configuration. This ensures that the post-hoc feature pruning is based on an optimized predictive pipeline, rather than a default model. **Exporting a Single Optimal Configuration.** While AutoGluon is highly regarded for its multi-model ensembling capabilities, our architecture strictly requires the extraction of a single model configuration to apply inside $\texttt{CIml}_{\text{config}}$. To satisfy this constraint, we intentionally disable ensembling and multi-level stacking. Instead, during the skeleton construction phase, we leverage AutoGluon to conduct a hyperparameter optimization search (HPO) within multiple model classes. Finally, we note that the architecture of MLCD is modular and agnostic to the specific AutoML backend. Any alternative AutoML tool can be integrated into the framework, provided it supports automated task detection and can output both the selected features and the optimal predictive configuration.

Time complexity. Let $|\mathbf{V}|$ be the number of nodes, $\mathrm{T}_{\text{autofit}}$ the AutoML training time, and T_{fit} the single configuration training time. MLCD requires $|\mathbf{V}| \cdot \mathrm{T}_{\text{autofit}}$ for the skeleton construction and $N_{ci} \cdot (4 \cdot K \cdot \mathrm{T}_{\text{fit}})$ for the graph resolution, where N_{ci} is the number of CI tests. The term $4 \cdot K \cdot \mathrm{T}_{\text{fit}}$ is the cost of a single CImlconfig test (accounting for the symmetric requirement, the K cross-validation folds, and $\mathcal{M}_1$ and $\mathcal{M}_2$ models). If the first phase yields a complete graph (worst case), N_{ci} reaches the maximum required by the FCI-stable algorithm. Feature importance, prediction, and evaluation times are excluded for clarity.

6. Evaluation on synthetic data

Simulation setup. We simulated Erdős–Rényi random DAGs with mixed data using the Mixed Additive Noise Model (MANM) (Huegle et al., 2021). MANM employs Nonlinear Additive Noise Models of the form: $V_i = \sum_{V_k \in Pa^{con}(V_i)} \beta_{k,i} f_{k,i}(V_k) + N_i$, in case of continuous-to-continuous relationships. In our setup, we define each $f_{k,i}$ to be sampled from the set: $\mathcal{F} = \{x, x^2, \cos(5x), e^{-x^2}\}$ with the linear relationship sampled with a probability of 0.05 and the remaining functions sampled equally. These functions were selected to challenge the evaluated methods. The coefficients $\beta_{k,i}$ are drawn from $\mathcal{U}([-2.0, -0.5] \cup [0.5, 2.0])$. The noise term follows $N_i \sim \mathcal{N}(0, \sigma^2)$, with $\sigma \sim \mathcal{U}(0.1, 0.4)$. We set 40% of the variables to be discrete, with domains ranging from 2 to 5. Mixed causal relationships are modeled using Conditional Linear Gaussian Models. The probability of noise for discrete nodes was set to 0.3. Continuous variables were standardized post-generation to ensure a fair comparison across all evaluated methods. We generated random DAGs and corresponding data across two experimental setups: (a) increasing sample size $n \in \{500, 1000, 5000\}$ with $|\mathbf{V}| = 10$ nodes and $d = 2.2$ average node degree, and (b) increasing average node degree $d \in \{2.2, 3.0, 4.0\}$ with $|\mathbf{V}| = 10$ nodes and $n = 5000$ samples. To evaluate the algorithm's performance under causal insufficiency, $L_p = 20\%$ of the nodes in each graph were designated as latent across all setups. These variables were randomly sampled from the set of common causes (nodes with an out-degree ≥ 2) to induce unobserved confounding.

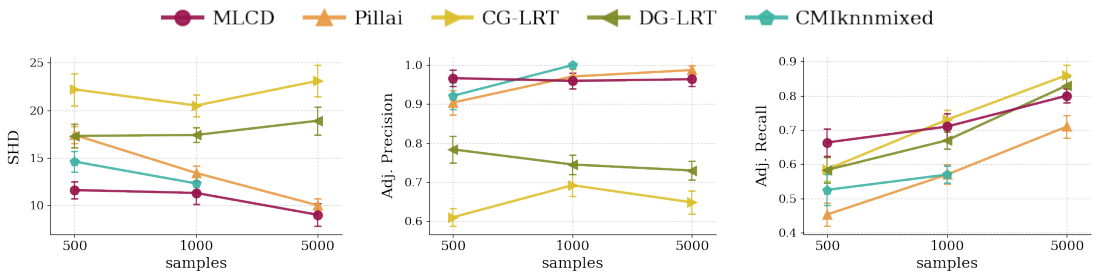


Figure 2: Comparative evaluation over increasing sample size. MLCD outperforms baseline methods, especially when the available data is limited.

MLCD implementation details. Within AutoGluon, we apply a 1000 sec time limit per node, a single 5-fold cross-validated bagging structure, and 15 HPO trials per model class. AutoGluon searches over tree-based (e.g. XGBoost) and KNN models. Neural network architectures are disabled to minimize computational cost. Feature selection utilizes 10 random shuffle sets with $\alpha = 0.1$. For the subsequent graph resolution phase, we set $\alpha = 0.1$ and constrain the maximum conditioning set size to maxk $= 4$. **Comparative Baselines.** We employ the same combinations of FCI-stable and CI tests outlined in our motivating example (CG-LRT, DG-LRT, Pillai, and CMIknnmixed). Due to its high computational cost, CMIknnmixed is omitted from experiments with 5000 samples. For all baseline methods, we use the same parameters as MLCD ($\alpha = 0.1$ and maxk $= 4$). All other hyper-parameters were kept at their default settings. **Evaluation metrics.** We evaluate the estimated PAGs using the Structural Hamming Distance (SHD), as in (Triantafillou and Tsamardinos, 2016), which penalizes mismatched adjacencies as well as edge marks between the estimated and true PAGs. We also compute the Adjacency Precision and Adjacency Recall to evaluate the presence of edges ignoring their orientation. Mean values and standard errors are reported across 10 repetitions for all metrics.

Results. Figure 2 displays the SHD, Adj. Precision, and Adj. Recall over increasing sample sizes for graphs with 10 nodes and an average node degree of 2.2. MLCD outperforms all baselines with respect to SHD, and its superior performance is particularly evident at smaller sample sizes. MLCD, Pillai, and CMIknnmixed all achieve high precision. Crucially, MLCD consistently maintains a higher recall than these two methods across all sample sizes. In contrast, CG-LRT and DG-LRT suffer from low precision despite achieving competitive recall. Figure 3 reports the SHD, Adj. Precision, and Adj. Recall over increasing average node degrees for 10-node DAGs and 5000 samples. MLCD consistently outperforms the baseline methods. Although its performance drops as graph density increases, its precision remains high—on par with Pillai—and it exhibits a higher recall than Pillai. Similar to the sample size experiments, CG-LRT and DG-LRT maintain good recall, but suffer from low precision. Overall, in line with our motivating example, MLCD successfully captures challenging edges where other methods fail, without sacrificing precision. MLCD is also competitive with respect to Arrowhead and Tail Precision and Recall, as shown in Appendix

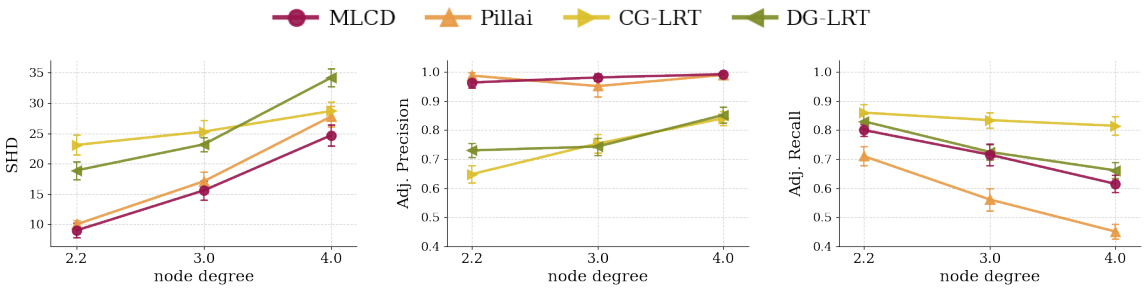


Figure 3: Comparative evaluation across increasing graph density. MLCD remains competitive against the baseline methods.

A.1. **Individual contribution of MLCD's phases.** In Appendix A.2, we show that Skeleton Construction introduces false positives edges, thereby limiting the computational gains of the moral graph initialization. However, it achieves high Adj. Recall, ensuring true edges are preserved for the next step. The Graph Resolution phase then refines the final graph using the CImlconfig test with the selected predictive models from the first phase. **Running times.** At an average node degree of 2.2, CG-LRT and DG-LRT are the most efficient methods ($<$ 1 sec per graph), followed by Pillai, which takes less than 1 min for 5000 samples. MLCD requires $\sim$ 17 to 26 min per graph for 500 to 5000 samples. For 5000 samples, Skeleton Construction needs 8 min per graph, while each CI test requires $\sim$ 5 sec. CMIknnmixed is the most computationally demanding method, requiring 30 min and 80 min for 500 and 1000 samples, respectively.

7. Real-world application

We study the robustness of MLCD on real-world data from the UCI repository, including Maternal Health Risk (7 variables, 1014 samples), AutoMPG (8 variables, 398 samples), and Yeast (9 variables, 1484 samples). MLCD operates on the raw data without prior modification. For the baseline frameworks, we implement a standard preprocessing pipeline including standardization, label encoding, and missing value imputation (mean and most frequent values). To validate the estimated PAGs, we use the predictive performance framework from Out-of-sample Causal Tuning (OCT) (Biza et al., 2022). We split the data into 70% training and 30% test sets. Each algorithm learns a PAG on the training set, which imposes a Markov boundary for each node. Using these Mbs, we train an AutoML pipeline on the training data for each target node and evaluate its predictive performance (mutual information) on the test set. OCT selects the sparsest graph with average performance statistically indistinguishable from the top-performing candidate. OCT evaluates statistical indistinguishability using a permutation test over 1000 repetitions and significance threshold at 0.05. For all methods, we use the same parameters as in Section 6.

In Table 2, we report the estimated number of edges ($|\mathcal{E}|$), the average Mb size ($\overline{\text{Mb}}$), and the average out-of-sample predictive performance ($\overline{\text{MI}}$) for each method and dataset.

Table 2: Evaluation on Real Data

	Maternal Health Risk			**AutoMPG**			**Yeast**		
Method	$\lvert\mathcal{E}\rvert$	$\overline{\mathbf{Mb}}$	$\overline{\mathrm{MI}}$	$\lvert\mathcal{E}\rvert$	$\overline{\mathbf{Mb}}$	$\overline{\mathrm{MI}}$	$\lvert\mathcal{E}\rvert$	$\overline{\mathbf{Mb}}$	$\overline{\mathrm{MI}}$
Pillai	5	2.3	0.94	5	2.8	0.60	5	2.4	0.16
CG-LRT	5	2.3	0.86	10	5	0.88	10	4	0.20
DG-LRT	17	6	1.14	14	7	0.94	13	8	0.24
CMIknnmixed	14	6	1.13	6	2.8	0.68	5	3.3	0.18
MLCD	**10**	**5.1**	**1.11**	**7**	**3.5**	**0.91**	**7**	**3.3**	**0.20**

MLCD is selected by OCT in all cases (shown in bold), with p-values 0.08, 0.18, and 0.12, respectively. However, we note that evaluating using predictive performance provides only indirect evidence of causal accuracy. In addition, OCT's selection may inherently favor MLCD since both approaches rely on measuring predictive performance. The number of estimated edges reveal interesting patterns across the methods. DG-LRT consistently infers denser graphs, which aligns with our findings in the synthetic experiments, whereas Pillai consistently estimates the sparsest graphs. MLCD exhibits an intermediate graph density and maintains a performance statistically indistinguishable from the top-performing candidate (DG-LRT). In Appendix B, we show the estimated PAGs by MLCD.

8. Discussion

In this work, we introduced MLCD, an end-to-end AutoML-based causal discovery framework, suitable for causally insufficient systems. MLCD inherits the soundness and completeness of FCI-based methods (see Appendix C), but also the advantages of AutoML tools: it handles diverse data types, bypasses manual data preprocessing, and captures complex functional relationships even with small sample sizes. Through synthetic experiments, we demonstrate that MLCD outperforms existing approaches. Compared to the FCI-based methods, MLCD (i) avoids global functional assumptions across the network and (ii) is scalable with respect to sample size, unlike other non-parametric approaches (e.g., KNN methods). However, the complexity of MLCD increases alongside variable dimensionality. As a result, MLCD is currently most practical for small graphs or discovering local neighborhoods. Its high computational cost can be justified in cases where we face unknown or complex functional relationships. In addition, the robustness of MLCD is inherently tied to the performance of the chosen AutoML tool and the underlying predictive models. If the selected framework fails to accurately capture the functional relationships or performs poorly in feature selection, the performance of MLCD is expected to decrease. Moreover, while MLCD currently relies on feature importance methods—as they are the most straightforward choices within existing AutoML tools—we acknowledge that this approach may introduce limitations, such as failure to capture feature interactions. Consequently, incorporating more sophisticated feature selection procedures within AutoML is a critical next step for MLCD.

Acknowledgments

Funded by the European Union (EIC TRANSITION OPEN, project number 101291665). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the granting authority (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

References

Sina Akbari, Ehsan Mokhtarian, AmirEmad Ghassami, and Negar Kiyavash. Recursive causal structure learning in the presence of latent variables and selection bias. *Advances in Neural Information Processing Systems*, 34:10119–10130, 2021.

Bryan Andrews, Joseph Ramsey, and Gregory F Cooper. Scoring Bayesian networks of mixed variables. *International journal of data science and analytics*, 6:3–18, 2018.

Bryan Andrews, Joseph Ramsey, and Gregory F Cooper. Learning high-dimensional directed acyclic graphs with mixed data-types. In *The 2019 ACM SIGKDD Workshop on Causal Discovery*, pages 4–21. PMLR, 2019.

Ankur Ankan and Johannes Textor. A simple unified approach to testing high-dimensional conditional independences for categorical and ordinal data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 12180–12188, 2023.

Ankur Ankan and Johannes Textor. pgmpy: A python toolkit for bayesian networks. *Journal of Machine Learning Research*, 25(265):1–8, 2024.

Francis Bach and Michael Jordan. Learning graphical models with mercer kernels. *Advances in Neural Information Processing Systems*, 15, 2002.

Daniel Bernstein, Basil Saeed, Chandler Squires, and Caroline Uhler. Ordering-based causal structure learning in the presence of latent variables. In *International conference on artificial intelligence and statistics*, pages 4098–4108. PMLR, 2020.

Rohit Bhattacharya, Tushar Nagarajan, Daniel Malinsky, and Ilya Shpitser. Differentiable causal discovery under unmeasured confounding. In *International Conference on Artificial Intelligence and Statistics*, pages 2314–2322. PMLR, 2021.

Konstantina Biza, Ioannis Tsamardinos, and Sofia Triantafillou. Out-of-sample tuning for causal discovery. *IEEE Transactions on Neural Networks and Learning Systems*, 35(4): 4963–4973, 2022.

Konstantina Biza, Sofia Triantafillou, and Ioannis Tsamardinos. Conditional independence testing with tuning: An approach based on predictive performance. In Ilias Maglogiannis, Lazaros Iliadis, Michalis Zervakis, and Antonios Papaleonidas, editors, *Artificial Intelligence Applications and Innovations*. Springer Nature Switzerland, 2026.

Giorgos Borboudakis and Ioannis Tsamardinos. Forward-Backward Selection with Early Dropping. *Journal of Machine Learning Research*, 20(8):1–39, 2019.

Wenyu Chen, Mathias Drton, and Ali Shojaie. Causal structural learning via local graphs. *SIAM journal on mathematics of data science*, 5(2):280–305, 2023.

David Maxwell Chickering. Learning equivalence classes of bayesian-network structures. *Journal of Machine Learning Research*, 2:445–498, 2002.

Tom Claassen and Tom Heskes. A bayesian approach to constraint based causal inference. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, UAI'12, page 207–216, Arlington, Virginia, USA, 2012. AUAI Press. ISBN 9780974903989.

Diego Colombo and Marloes H Maathuis. Order-independent constraint-based causal structure learning. *The Journal of Machine Learning Research*, 15(1):3741–3782, 2014.

Diego Colombo, Marloes H Maathuis, Markus Kalisch, and Thomas S Richardson. Learning high-dimensional directed acyclic graphs with latent and selection variables. *The Annals of Statistics*, pages 294–321, 2012.

Nick Erickson, Jonas Mueller, Alexander Shirkov, Hang Zhang, Pedro Larroy, Mu Li, and Alexander Smola. Autogluon-tabular: Robust and accurate automl for structured data. *arXiv preprint arXiv:2003.06505*, 2020.

Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Springenberg, Manuel Blum, and Frank Hutter. Efficient and robust automated machine learning. *Advances in neural information processing systems*, 28, 2015.

Clark Glymour, Kun Zhang, and Peter Spirtes. Review of causal discovery methods based on graphical models. *Frontiers in genetics*, 10:524, 2019.

Teny Handhayani and James Cussens. Kernel-based approach for learning causal graphs from mixed data. In *International Conference on Probabilistic Graphical Models*, pages 221–232. PMLR, 2020.

Biwei Huang, Kun Zhang, Yizhu Lin, Bernhard Schölkopf, and Clark Glymour. Generalized score functions for causal discovery. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1551–1560, 2018.

Johannes Huegle, Christopher Hagedorn, Lukas Boehme, Mats Poerschke, Jonas Umland, and Rainer Schlosser. Manm-cs: Data generation for benchmarking causal structure learning from mixed discrete-continuous and nonlinear data. *WHY-21 at NeurIPS*, 2021 (2):3, 2021.

Johannes Huegle, Christopher Hagedorn, and Rainer Schlosser. A knn-based non-parametric conditional independence test for mixed data and application in causal discovery. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 541–558. Springer, 2023.

David Kaltenpoth and Jilles Vreeken. Nonlinear causal discovery with latent confounders. In *International Conference on Machine Learning*, pages 15639–15654. PMLR, 2023.

Ron Kohavi and George H John. Wrappers for feature subset selection. *Artificial intelligence*, 97(1-2):273–324, 1997.

Chun Li and Xiaodan Fan. On nonparametric conditional independence tests for continuous variables. *Wiley Interdisciplinary Reviews: Computational Statistics*, 12(3):e1489, 2020.

Takashi Nicholas Maeda and Shohei Shimizu. Rcd: Repetitive causal discovery of linear non-gaussian acyclic models with latent confounders. In *International Conference on artificial intelligence and statistics*, pages 735–745. PMLR, 2020.

Takashi Nicholas Maeda and Shohei Shimizu. Causal additive models with unobserved variables. In *Uncertainty in Artificial Intelligence*, pages 97–106. PMLR, 2021.

Dimitris Margaritis and Sebastian Thrun. Bayesian network induction via local neighborhoods. *Advances in neural information processing systems*, 12, 1999.

Ehsan Mokhtarian, Sepehr Elahi, Sina Akbari, and Negar Kiyavash. Recursive causal discovery. *Journal of Machine Learning Research*, 26(61):1–65, 2025.

Francesco Montagna, Philipp Michael Faller, Patrick Blöbaum, Elke Kirschbaum, and Francesco Locatello. Score matching through the roof: linear, nonlinear, and latent variables causal discovery. In Biwei Huang and Mathias Drton, editors, *Proceedings of the Fourth Conference on Causal Learning and Reasoning*, volume 275 of *Proceedings of Machine Learning Research*, pages 552–605. PMLR, 07–09 May 2025.

Yoshimitsu Morinishi and Shohei Shimizu. Differentiable causal discovery of linear non-gaussian acyclic models under unmeasured confounding. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856.

Ignavier Ng, Xinshuai Dong, Haoyue Dai, Biwei Huang, Peter Spirtes, and Kun Zhang. Score-based causal discovery of latent variable causal models. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 37531–37552. PMLR, 21–27 Jul 2024.

Juan Miguel Ogarrio, Peter Spirtes, and Joe Ramsey. A hybrid causal search algorithm for latent variable models. In *Proceedings of the Eighth International Conference on Probabilistic Graphical Models*, volume 52, pages 368–379. PMLR, 06–09 Sep 2016.

Judea Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, USA, 2nd edition, 2009.

Jean-Philippe Pellet and André Elisseeff. Using markov blankets for causal structure learning. *Journal of Machine Learning Research*, 9(7), 2008a.

Jean-Philippe Pellet and André Elisseeff. Finding latent causes in causal networks: an efficient approach based on markov blankets. *Advances in Neural Information Processing Systems*, 21, 2008b.

Oana-Iuliana Popescu, Andreas Gerhardus, Martin Rabel, and Jakob Runge. Nonparametric conditional independence testing for mixed continuous-categorical variables: A novel method and numerical evaluation. In *Proceedings of the Fourth Conference on Causal Learning and Reasoning*, pages 406–450. PMLR, 2025.

Vineet K Raghu, Joseph D Ramsey, Alison Morris, Dimitrios V Manatakis, Peter Sprites, Panos K Chrysanthis, Clark Glymour, and Panayiotis V Benos. Comparison of strategies for scalable causal discovery of latent variable models from mixed data. *International journal of data science and analytics*, 6(1):33–45, 2018.

Joseph Ramsey, Peter Spirtes, and Jiji Zhang. Adjacency-faithfulness and conservative causal inference. In *Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence*, pages 401–408, 2006.

Joseph Ramsey, Bryan Andrews, and Peter Spirtes. Efficient latent variable causal discovery: Combining score search and targeted testing. *arXiv preprint arXiv:2510.04263*, 2025.

Kari Rantanen, Antti Hyttinen, and Matti Järvisalo. Maximal ancestral graph structure learning via exact search. In *Uncertainty in Artificial Intelligence*, pages 1237–1247. PMLR, 2021.

Adèle H Ribeiro and Dominik Heider. dcfci: Robust causal discovery under latent confounding, unfaithfulness, and mixed data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026.

Thomas Richardson and Peter Spirtes. Ancestral graph Markov models. *The Annals of Statistics*, 30(4):962–1030, 2002.

Marco Scutari. Learning bayesian networks with the bnlearn r package. *Journal of statistical software*, 35:1–22, 2010.

Elena Sokolova, Perry Groot, Tom Claassen, and Tom Heskes. Causal discovery from databases with discrete and continuous variables. In *European Workshop on Probabilistic Graphical Models*, pages 442–457. Springer, 2014.

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

Sofia Triantafillou and Ioannis Tsamardinos. Score-based vs constraint-based causal learning in the presence of confounders. In *Cfa@ uai*, pages 59–67, 2016.

Michail Tsagris, Giorgos Borboudakis, Vincenzo Lagani, and Ioannis Tsamardinos. Constraint-based causal discovery with mixed data. *International journal of data science and analytics*, 6:19–30, 2018.

Ioannis Tsamardinos and Constantin F Aliferis. Towards principled feature selection: Relevancy, filters and wrappers. In *International Workshop on Artificial Intelligence and Statistics*, pages 300–307. PMLR, 2003.

Konstantinos Tsirlis, Vincenzo Lagani, Sofia Triantafillou, and Ioannis Tsamardinos. On scoring maximal ancestral graphs with the max–min hill climbing algorithm. *International Journal of Approximate Reasoning*, 102:74–85, 2018.

Matthew J Vowels, Necati Cihan Camgoz, and Richard Bowden. D'ya like dags? a survey on structure learning and causal discovery. *ACM Computing Surveys*, 55(4):1–36, 2022.

Chi Wang, Qingyun Wu, Markus Weimer, and Erkang Zhu. Flaml: A fast and lightweight automl library. *Proceedings of machine learning and systems*, 3:434–447, 2021.

Alessio Zanga, Elif Ozkirimli, and Fabio Stella. A survey on causal discovery: Theory and practice. *International Journal of Approximate Reasoning*, 151:101–129, 2022.

Jiji Zhang. On the completeness of orientation rules for causal discovery in the presence of latent confounders and selection bias. *Artificial Intelligence*, 172(16-17):1873–1896, 2008.

Appendix A. Supplementary results on synthetic data

We report supplementary experiments and results on the synthetic data from Section 6.

A.1. Orientation metrics

Following Ramsey et al. (2025), we compute the Arrowhead and Tail Precision and Recall for common adjacencies. Arrowhead Precision is defined as the number of correctly predicted arrowheads on common adjacencies, divided by the total number of arrowheads predicted on common adjacencies. Arrowhead Recall is defined as the number of correctly predicted arrowheads on common adjacencies, divided by the total number of true arrowheads on common adjacencies. Tail Precision and Recall are defined analogously for tails. Figure 4 reports Arrowhead and Tail Precision, Recall, and F1 scores over increasing sample sizes for graphs with 10 nodes and an average node degree of 2.2 (supplementing Figure 2). Figure 5 reports Arrowhead and Tail Precision, Recall, and F1 scores over increasing average node degrees for 10-node DAGs and 5000 samples (supplementing Figure 3). The performance of MLCD remains competitive across all settings.

A.2. Individual contribution of MLCD's phases

Skeleton Construction. We compare the moral graph obtained by the Skeleton Construction phase against those estimated by the FCI-stable algorithm using the Pillai, CG-LRT, DG-LRT, and CMIknnmixed conditional independence tests. In Figure 6, we report the Adjacency Precision, Recall, and F1 scores over increasing (a) sample sizes for graphs with 10 nodes and an average node degree of 2.2, and (b) average node degrees for 10-node DAGs and 5000 samples. MLCD achieves optimal Adj. Recall in all cases, though it may introduce false positive edges. This behavior is expected due to the "OR-rule". The subsequent CI testing phase in MLCD is designed to prune these false positive edges. Furthermore, we note that the Skeleton Construction phase is utilized not only to derive the moral graph, but also to identify the optimal predictive models for each node that will be used in the Graph Resolution phase.

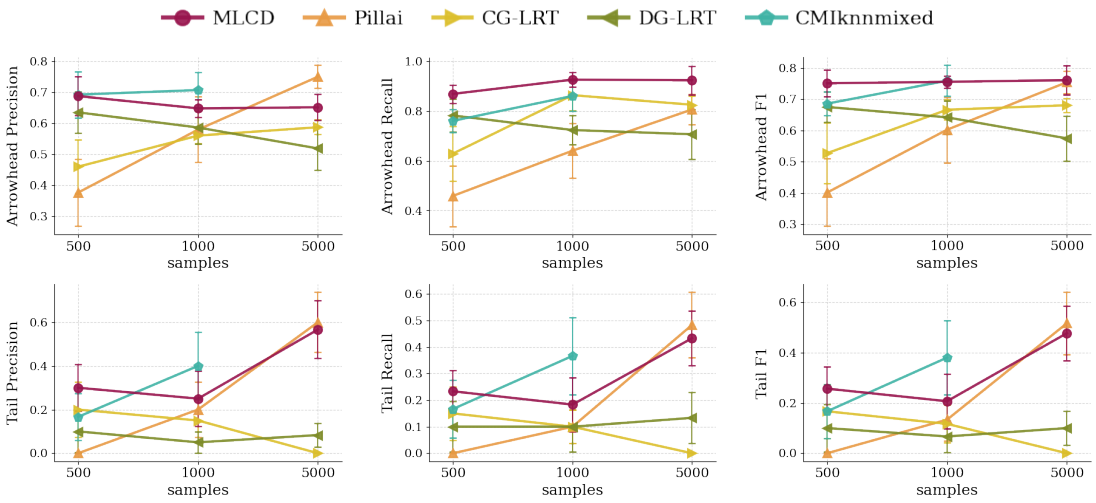


Figure 4: **Orientation Metrics.** Arrowhead and Tail Precision, Recall, and F1 scores over increasing sample size.

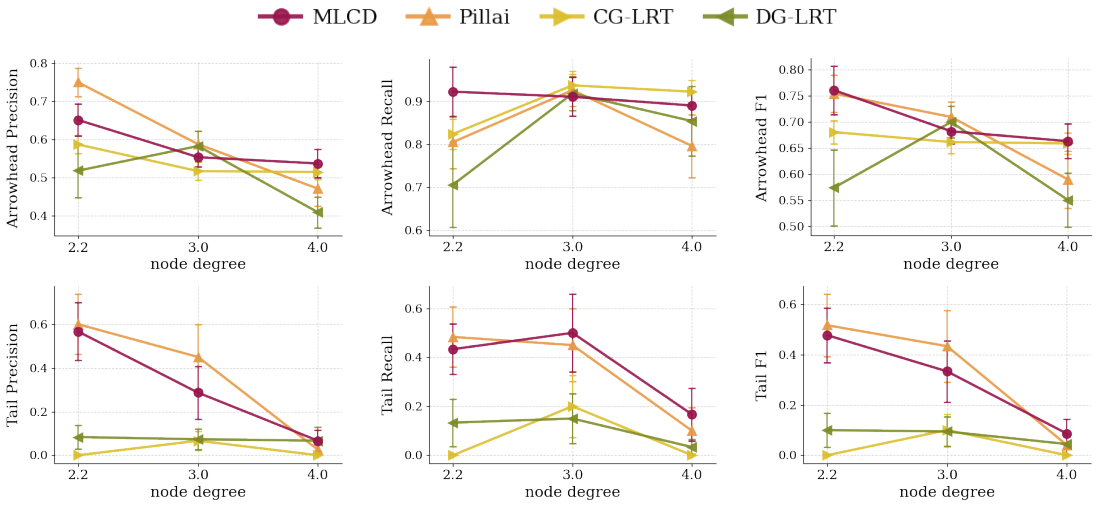
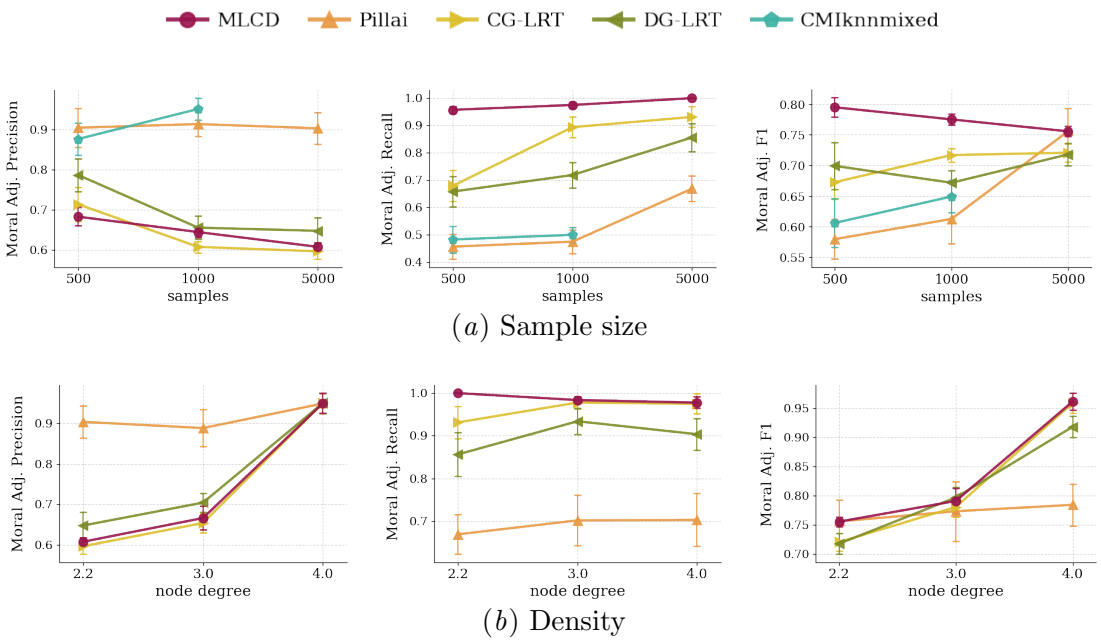


Figure 5: **Orientation Metrics.** Arrowhead and Tail Precision, Recall, and F1 scores over increasing graph density.

Graph Resolution. We evaluate the Graph Resolution phase of MLCD using various conditional independence tests. Specifically, we initialize the FCI-stable algorithm with the estimated moral graph constructed by MLCD and apply the CI tests Pillai, CG-LRT, DG-LRT, and CMIknnmixed. In Figure 7, we report the SHD, Adjacency Precision and



(a) Sample size

(b) Density

Figure 6: **Skeleton Construction phase.** We evaluate the moral graphs with respect to Adjacency Precision, Recall, and F1 scores over increasing (a) sample size and (b) graph density.

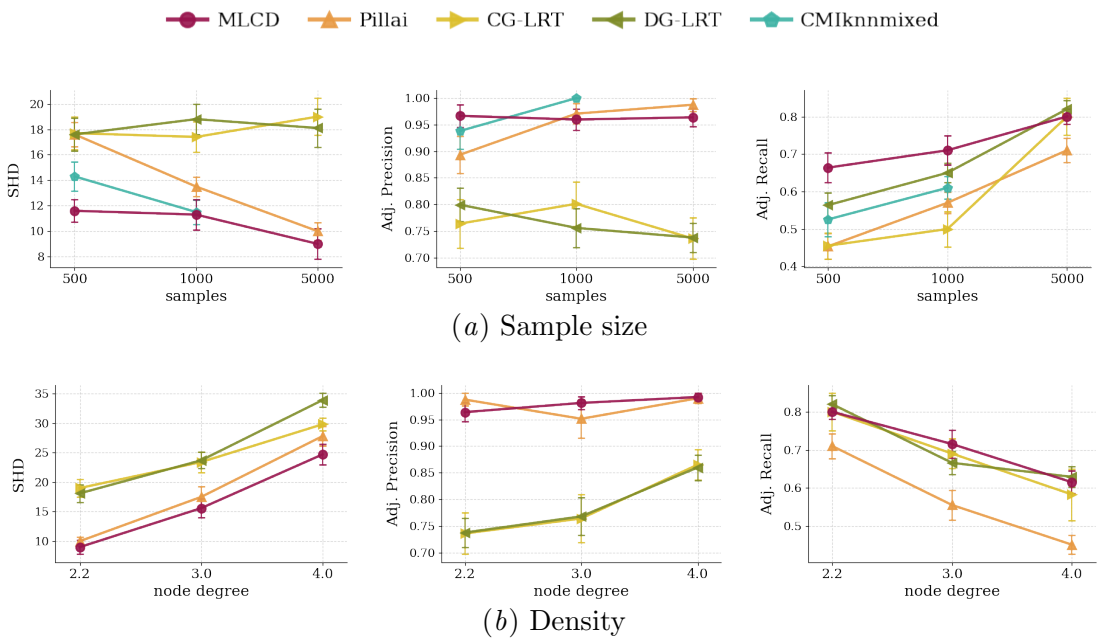
Adjacency Recall for the output PAGs across increasing (a) sample size and (b) graph density. We obtain similar results to those in Figures 2 and 3. MLCD remains competitive, especially for smaller sample sizes.

Appendix B. Evaluation on Real-Data

Figure 8 shows the estimated PAGs produced by MLCD on real-world data: Maternal Health Risk, AutoMPG and Yeast.

Appendix C. Theoretical Soundness

Assuming (i) the Causal Markov Condition and Faithfulness assumptions hold, (ii) we have access to an oracle conditional independence test, and (iii) the Skeleton Construction returns the correct Markov boundaries or a superset of them, then MLCD preserves the theoretical soundness and completeness guarantees of FCI-stable. The true moral graph (and any superset of the true moral graph) contains the true adjacencies that are dependent given any conditioning set, and the links $X - Y$ that can be removed given a separating set $\mathbf{S}_{XY}$. These $X - Y$ can be either moral links (i.e., a link between a node and its Markov boundary) or false positive links. By definition, $\text{Mb}(X)$ renders X conditionally independent of any other node not contained in $\text{Mb}(X)$. As a result, a separating set $\mathbf{S}_{XY}$ for a false positive

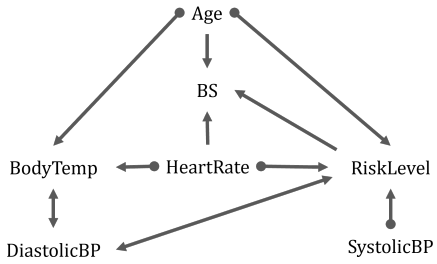


(*a*) Sample size

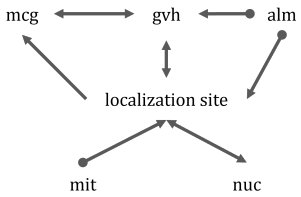
(*b*) Density

Figure 7: **Graph Resolution phase.** The FCI-stable algorithm is initialized with the moral graph estimated by MLCD. We evaluate the output PAGs over increasing (a) sample size and (b) graph density.

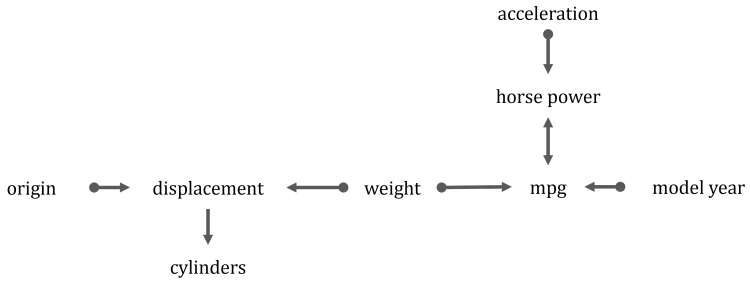
link can be found in the Markov boundaries of X and Y. Similarly, as shown by Pellet and Elisseeff (2008b), the separating set $\mathbf{S}_{XY}$ for a moral link can be also found in the Markov boundaries of X and Y. In addition, the FCI and FCI-stable algorithms are sound and complete; they perform all necessary conditional independence tests over candidate conditioning sets to identify separating sets $\mathbf{S}_{XY}$ and remove the spurious links $X - Y$ (Spirtes et al., 2001; Zhang, 2008; Colombo and Maathuis, 2014). As a result, when FCI (or FCI-stable) is initialized with a superset of the true moral graph, it executes the necessary conditional independence tests and successfully identifies the required separating sets $\mathbf{S}_{XY}$ to remove false positive or moral links.



(a) Maternal Health Risk



(b) Yeast



(c) AutoMPG

Figure 8: The estimated PAGs by MLCD across three real-world data.