AutoCD: Automated Machine Learning for Causal Discovery Algorithms

Gerlise Chan[1] **Tom Claassen**[2] **Holger H. Hoos**[1, 3] **Tom Heskes**[2] **Mitra Baratchi**[1]

[1]Leiden Institute of Advanced Computer Science (LIACS), Leiden University, The Netherlands.
[2]Institute for Computing and Information Sciences (iCIS), Radboud University, The Netherlands.
[3]Chair for AI Methodology (AIM), RWTH Aachen University, Germany.

Abstract This paper studies automated machine learning (AutoML) for causal discovery, the process of uncovering cause-and-effect relationships within data. Causal discovery is an unsupervised learning problem, as the target (the underlying ground truth causal model) is typically unknown. Therefore, the loss functions commonly used as an optimisation objective in AutoML systems developed for supervised learning problems are not applicable. We propose AutoCD, the first AutoML system utilising Bayesian optimisation based on a search space of causal discovery algorithms. In designing AutoCD, we study and compare the applicability of two different loss functions and post-hoc correction strategies. Additionally, based on the analysis of the performance of AutoCD, we propose an improved version called AutoCD$_{PC}$ by warm-starting the search from the PC algorithm. Results from our experiments on datasets simulated from 45 graphical models demonstrate that AutoCD$_{PC}$ performs better than the baselines by ranking the highest (avg. rank 3.69) compared to the best causal tuning baseline (avg. rank 5.21) and the best fine-tuned individual algorithm (avg. rank 4.36).

1 Introduction

Causal discovery is the process of identifying causal relationships in the data. A deeper understanding of causal relationships can assist in developing effective interventions or policies. Research in the field of causal discovery has resulted in various algorithms with different underlying assumptions, showing varying performance across datasets. To achieve high performance on a given dataset, it is essential that users perform both algorithm selection and hyperparameter optimisation.

The field of automated machine learning (AutoML) has previously addressed the combined algorithm selection and hyperparameters optimisation (CASH) problem for supervised learning problems by defining a search space based on existing algorithms and an efficient search strategy [1]. This search strategy evaluates the performance of models using a clearly defined loss function (e.g., classification or regression error). Applying these strategies to causal discovery poses a challenge, due to the typically unknown ground-truth graph and evaluation metrics that do not use this target graph. Consequently, the conventional cross-validation approach and loss functions used in automated supervised learning become inapplicable.

This work focuses on developing the first AutoML system for causal discovery by reformulating the CASH problem. Specifically, we study loss functions from two previously proposed causal tuning methods, namely the stability approach to regularisation selection (StARS) [2] and out-of-sample causal tuning (OCT) [3]. Up to now, these methods have only been tested in conjunction with grid search, which is unsuitable for exploring a larger space comprising many algorithms and hyperparameters. Prior work shows that optimising these two loss functions on a search space of causal discovery algorithms may lead to suboptimal performance requiring a post-hoc correction based on evaluated configurations in the search space to identify a higher-performing configuration [3]. To extend the search space, a more efficient search strategy is needed, such as the Bayesian optimisation (BO) [4]. In addition, the effectiveness of the post-hoc correction needs to be verified,

as BO targets a smaller set of configurations. To address these challenges, the main contributions of our research include:

1. Development of AutoCD, the first AutoML approach and system comprising algorithm selection and hyperparameter optimisation for causal discovery.
2. Proposing variants of AutoCD incorporating post-hoc correction and warm-starting the search from the PC algorithm [5].
3. Evaluation of AutoCD on synthetic datasets from 45 graphical models and a real-world dataset that addresses the progression of mild cognitive impairment and early Alzheimer's disease.

2 Related work

Model selection. Causal discovery algorithms can be tuned using statistical model selection techniques such as the Akaike information criterion (AIC) and Bayes information criterion (BIC) [6]. Maathuis et al. [7] defined an objective function based on a modified BIC criterion to tune the hyperparameter alpha (the cutoff for p-values in conditional independence tests) of PC [5] by trying out 7 alpha values. Biza et al. [3] extended this method to tune a set of hyperparameter configurations. They demonstrated that the computation of the likelihood is difficult for causal models, as real-world data may have arbitrary functional dependencies and noise distributions.

Causal tuning. Liu et al. proposed another causal tuning approach by introducing the stability approach to regularisation selection (StARS) algorithm [2], which employs a stability-based approach [8] founded on subsampling [9] for model selection. This method iteratively refines the hyperparameters of interest until the network instability reaches a user-specified significance threshold, resulting in an optimised model. StARS was further adapted for causal discovery algorithms [10, 3]. Raghu et al. [10] primarily focused on refining two key hyperparameters (alpha for constraint-based algorithms and penalty discount for score-based algorithms). Biza et al. pointed out limitations of StARS in assessing how well the causal model fits the data. This introduces bias and leads to favouring a configuration that consistently makes the same systematic errors on the sub-samples, as this minimises network instability. Building upon their adaptation of StARS, Biza et al. further introduced out-of-sample causal tuning (OCT), the first method employing k-fold cross-validation. OCT transforms the unsupervised learning problem into supervised learning by treating a causal model as a collection of predictive models [3].

AutoML. Existing tuning methods for causal discovery use grid search to explore the search space [7, 10, 3]. Expanding this search space would significantly increase time complexity. Bayesian optimisation (BO) is a more efficient search strategy widely employed method in AutoML for supervised learning tasks [11, 4]. Examples of BO hyperparameter optimisation approaches include sequential model-based algorithm configuration (SMAC) [12], Gaussian processes [13] and tree Parzen estimators (TPE) [14] (for details we refer to the survey paper by Wang et al. [15]). As causal discovery algorithms continue to evolve, the complexity of the respective AutoML search spaces may increase drastically with additional algorithms and hyperparameters. To address this challenge, we aim to develop an AutoML system utilising SMAC due to its proven performance in AutoML systems such as auto-weka [1], auto-sklearn [16], and auto-pytorch [17]. This approach aims to focus on higher-quality configurations compared to those explored by grid search.

AutoML for unsupervised learning. The literature for automated unsupervised learning is much more limited compared to supervised learning. De Souto et al. [18] address the algorithm selection problem for clustering by using a meta-learning approach where knowledge is extracted from datasets with a similar set of dataset meta-features. Adopting this approach, Ferrari et al. [19] suggest new distance-based meta-features and combine ranking methods for algorithm selection. AutoClust proposed by Poulakis et al. [20], is an AutoML framework that automates the clustering task based on cluster validity indices and meta-learning. For algorithm selection, the k-nearest neighbour approach is adopted with majority voting. Furthermore, an optimisation goal is proposed

for hyperparameter tuning that uses a predictive model to learn the mapping between different internal cluster validity indices that do not need ground truth for validation (e.g., Silhouette, CDBW) and the Adjusted Rank Index (ARI) prediction that needs ground truth for validation. Li et al. [21] propose AutoSSL that addresses AutoML for semi-supervised learning. Like unsupervised learning tasks, it employs meta-learning for algorithm selection using the ground truth. Afterwards, hyperparameters of the selected algorithm are tuned using a large margin separation method that assesses and exploits the large margin separation using a predictive model without ground truth. Automating causal discovery exhibits extra challenges compared to other unsupervised learning tasks, as one cannot rely on the existence of ground truth for any form of internal validation.

3 Methods

In this section, we present automated machine learning for causal discovery algorithms (AutoCD), an AutoML system for tuning causal discovery algorithms. We first introduce the formal problem, followed by details of the loss functions and post-hoc correction strategies.

3.1 Automated Causal Discovery

We propose AutoCD, an AutoML system to address the CASH problem [1] for causal discovery. Two differences exist between the CASH problem defined based on supervised learning and causal discovery: (i) the utilisation of k-fold cross-validation and (ii) the loss function. In the context of unsupervised learning, cross-validation is typically not applicable due to the unknown ground truth graph. Moreover, as causal discovery is a descriptive unsupervised learning task, the objective is to understand and describe the dataset rather than making predictions for unseen instances. Cross-validation can be used in predictive tasks with the goal of performing a specific task (e.g., assigning class labels) on unseen instances. The ground truth data needed in cross-validation is typically one target (e.g., a class label) per instance. The ground truth needed for validating causal discovery is, however, the full causal graph explaining the relation between all variables. If such a causal graph was known, there was no point in causal discovery. In absence of ground truth on the target, loss functions are unable to accurately measure the performance of uncovering the underlying causal structure. This is a common problem in automating unsupervised learning tasks where no universally applicable metric exists. Consequently, there is no straightforward way to define a loss function, in contrast to supervised learning where conventional evaluation metrics are optimised. As a result, the reformulated CASH problem for AutoCD is defined as

$$A^*, \boldsymbol{\lambda}^* \in \underset{A^{(i)} \in \mathcal{A}, \boldsymbol{\lambda} \in \boldsymbol{\Lambda}^{(i)}}{\arg\min} \; \mathcal{L}(A^{(i)}_{\boldsymbol{\lambda}}, \mathcal{D}) \,. \tag{1}$$

Here, given an algorithm $A^{(i)}$ from the set of algorithms $\mathcal{A} = \{A^{(1)}, \ldots, A^{(i)}, \ldots, A^{(n)}\}$ with corresponding configuration space $\boldsymbol{\Lambda}^{(i)}$, a hyperparameter configuration is denoted $A^{(i)}_{\boldsymbol{\lambda}}$ for $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}^{(i)}$. The optimal algorithm A^* and optimal hyperparameter settings $\boldsymbol{\lambda}^*$ are determined by optimising the loss function $\mathcal{L}(A^{(i)}_{\boldsymbol{\lambda}}, \mathcal{D})$ for a given dataset $\mathcal{D}$. The CASH problem can be reformulated as hyperparameter optimisation (HPO), by treating algorithm selection as a top-level hyperparameter denoted $\boldsymbol{\lambda}_{alg}$, responsible for selecting an algorithm from the set $\mathcal{A}$. The configuration space changes to $\boldsymbol{\Lambda} = \boldsymbol{\Lambda}^{(1)} \cup \cdots \cup \boldsymbol{\Lambda}^{(n)} \cup \{\boldsymbol{\lambda}_{alg}\}$, where hyperparameter $\boldsymbol{\lambda}_{alg}$ selects algorithm $A^{(i)}$.

3.2 Loss function

To assess the performance of a configuration $A_{\boldsymbol{\lambda}}$ in causal discovery, it is necessary to define a loss function $\mathcal{L}$ that can be assessed without ground truth knowledge. In this work, we study two distinct approaches from the literature: StARS [2, 3] and OCT [3].

StARS (see Algorithm 1 in Appendix A), originally proposed by Liu et al. [2] and modified for causal discovery by Biza et al. [3], utilises data subsampling to quantify the network instability

of a model, which reflects the sensitivity of the graph structure to changes in the data. Given a configuration $A_{\boldsymbol{\lambda}}$, each sub-sample $\mathcal{D}^{(i)}$ is used to estimate a causal graph, compute the number of edges, and compute the probability p_{XY} denoting the presence of an edge (X, Y) between variables X and Y in the graph. The density of the causal graph is obtained by averaging the number of edges over S sub-samples. Additionally, the instability of each edge (X, Y) in the causal graph is computed as $\xi_{XY} \equiv 2 \cdot p_{XY} \cdot (1 - p_{XY})$. The network instability of a configuration $N(A_{\boldsymbol{\lambda}})$ is defined as the average edge instability across all edges in the causal graph.

OCT (see Algorithm 2 in Appendix A), employs k-fold cross-validation to produce a collection of predictive models, each treating a specific variable in the dataset as a target, computing and averaging the performance of the models with mutual information. Given a configuration $A_{\boldsymbol{\lambda}}$, a causal graph is estimated from the training set, and a Markov boundary is computed for each variable X. The Markov boundary is utilised to construct a random forest (RF) model $\mathcal{M}_X$ to predict the variable X using the validation set. The predictive performance of the model predicting the target variable is measured using mutual information, comparing true values of X with the aggregated predictions $\hat{X}$. If X is continuous, the mutual information is defined as $I(X, \hat{X}) = \int_{\hat{x}} \int_{x} p(x, \hat{x}) \cdot \log \frac{p(x,\hat{x})}{p(x) \cdot p(\hat{x})} dx d\hat{x}$, where $p(x)$, $p(\hat{x})$, and $p(x, \hat{x})$ denote the marginal densities of X, $\hat{X}$ and the joint density, respectively. On the other hand, if X is discrete, the mutual information is $I(X, \hat{X}) = \sum_{c_x \in \mathrm{C}} \sum_{c_{\hat{x}} \in \mathrm{C}} P(c_x, c_{\hat{x}}) \cdot \log \frac{P(c_x, c_{\hat{x}})}{P(c_x) \cdot P(c_{\hat{x}})}$, where $c_x, c_{\hat{x}} \in \boldsymbol{C}$ denote the categories of X and $\hat{X}$. The aggregated mutual information $I_{A_{\boldsymbol{\lambda}}}$ is computed as the average mutual information across all variables. HPO methods can optimise these loss functions favouring lower network instability values and higher mutual information values.

3.3 Post-hoc correction

The original StARS and OCT methods utilise penalties. We aim to use these penalties as a post-hoc correction after obtaining a set of evaluated configurations from a BO trial. It should be noted that originally, these penalties apply to all possible configurations in the space. However, when using BO, only a subset of sequentially evaluated configurations are available. Therefore, the effectiveness of these penalties in AutoCD still needs to be verified.

The StARS penalty is shown in Appendix A, Algorithm 3. After obtaining the evaluated configurations $\boldsymbol{A}$ with corresponding density estimation values Q and network instability values N, the network instability values are sorted based on the density estimation values and subsequently reordered as a monotonically increasing sequence. This ordering results in configurations with sparse but stable graphs appearing before those with dense and unstable graphs. Next, a configuration is selected that is not too sparse but also not unstable, using the threshold β. Originally, StARS as implemented by Biza et al. does not optimise for network stability. Instead, the post-hoc correction determines the best-performing configuration among all evaluated configurations. From this penalty, we can gather that the stability of graphs is an important objective. Therefore, in AutoCD we optimise network instability which points to configurations with stable graphs. The effectiveness of the loss function in combination with this penalty needs to be verified.

The OCT penalty is shown in Appendix A, Algorithm 4. After obtaining the evaluated configurations $\boldsymbol{A}$ and the best-performing configuration $A^*_{\boldsymbol{\lambda}}$ with corresponding mutual information values I, predictions $\hat{X}$ and Markov boundary sizes |MB|, the null hypothesis of no difference between the optimal configuration and an arbitrary configuration with threshold β is tested by permutation testing. Eventually, a configuration is selected that has the same mutual information value as the optimal configuration, but a smaller Markov boundary and is, therefore, less complex. Biza et al. introduced this penalty to avoid false positive variables in Markov boundaries and showed that this penalty either improves the performance or maintains the current performance.

3.4 Search space

The AutoCD search space encompasses 11 causal discovery algorithms. These algorithms were sourced from Tetrad [22], a widely used software package, and gCastle [23], which includes recent implementations of causal discovery algorithms. The search space is tree-structured, comprising 74 hyperparameters and including the algorithm selection hyperparameter with a mix of continuous, categorical and conditional hyperparameters. The search space is flexible for future expansion to allow for easy integration of emerging causal discovery algorithms. Table 9 in Appendix B presents an overview of the causal discovery algorithms embedded within AutoCD.

4 Experimental setup

For our experiments we used SMAC [12], a general-purpose algorithm configurator based on BO. SMAC is suitable for searching within tree-structured search spaces with a mixture of different hyperparameter types. We initially want to investigate the impact of data sample size and the time budget available to SMAC, to ensure enough data and time budget is available to achieve the best practically possible results. Next, we study the performance of AutoCD on discrete, continuous, and mixed simulated datasets. In our experiments, the performance of AutoCD is compared against the performance of the baselines and enhanced AutoCD variants. The experiments are run on three compute clusters with Intel Xeon E5-2630 processors @ 2.40GHz.

4.1 Baselines

We consider two groups of baselines: (i) causal tuning methods (StARS and OCT) and (ii) hyperparameter-tuned causal discovery algorithms. The first group originally utilises grid search for a much smaller search space. However, to allocate the same time budget and the same search space as AutoCD, we employed random search for this group. For the second group, we use BO to tune the hyperparameters (hyperparameters are shown in Table 9). For a fair comparison, all methods adhere to a wall-clock time of 1 hour and terminate a trial exceeding 15 minutes to search for a configuration. This termination strategy was chosen, because we observed a considerable amount of configurations taking a long time without producing better results.

- *StARS* [3]: This method has 2 untunable hyperparameters, the number of data sub-samples set to 20 and the threshold set to 0.05, as suggested in Biza et al. [3].
- *OCT* [3]: This method has 3 hyperparameters, the number of folds (cross-validation) set to 10, the number of permutations set to 1000, and the threshold set to 0.05, as suggested by the authors.
- *PC* [5]: This algorithm is included as a baseline due to its popularity and versatile applicability.
- *FGES* [24]: This algorithm shares similarities with PC in terms of broad applicability to diverse data types, making it a suitable baseline for comparative analysis.
- *LiNGAM* [25]: This algorithm is specifically tailored for continuous datasets; it has various extensions to address complexities such as latent confounders.
- *GOLEM* [26]: This algorithm employs a gradient-based approach to learn the underlying causal structure without imposing structural assumptions using a data-driven approach.

These baselines are compared against AutoCD and its variants: AutoCD+, AutoCD$_{PC}$ and AutoCD$_{PC}$+. AutoCD is determined by the best performing version, either AutoCD$_{StARS}$ utilising StARS or AutoCD$_{OCT}$ utilising OCT as loss function. AutoCD+, additionally, applies the penalty as a post-hoc correction to the results of AutoCD. AutoCD$_{PC}$ is an improved AutoCD variant that will be introduced in Section 5.4. AutoCD$_{PC}$+ combines AutoCD+ and AutoCD$_{PC}$.

4.2 Evaluation metrics

We evaluate the causal graphs using two groups of commonly used evaluation metrics: classification-based (false positives, false negatives, causal accuracy) and graph distance-based (structural Hamming distance). False positives (FP) and false negatives (FN) refer to over- and underpredicted

edges, respectively. Causal accuracy (CA) measures the accuracy in discovering the underlying causal structure. It examines the proportion of correctly identified causal relationships relative to the total number of causal connections $CA = TP/(TP + FP + FN)$. Structural Hamming distance (SHD) computes the number of edge insertions, deletions and reversals required to transform the estimated graph into the target graph. For a meaningful comparison across datasets with varying nodes and node degrees, all metrics are normalised by the total number of causal connections $(TP + FP + FN)$, like in CA.

4.3 Datasets

The experiments are conducted using both synthetic and real-world data. While real-world data offers practical insights of AutoCD, the synthetic datasets provide control over the causal structure with a known target graph to validate against. This is a common approach in causal discovery, as the target graph in real-world datasets is unknown or not agreed upon by experts [27].

Synthetic datasets. We employed Py-tetrad [28] to generate random directed acyclic graphs (DAGs), to simulate synthetic datasets with discrete, continuous, and mixed variables. Each graphical model thus generated simulates 25 target graphs and datasets; in total, there are 45 graphical models resulting in 1125 target graphs and datasets. The parameters for the graphical model, such as the data type and the number of nodes, are provided in Appendix C, Table 10.

Real-world dataset. The real-world dataset is obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (`adni.loni.usc.edu`). The ADNI was launched in 2003 as a public-private partnership, led by Principal Investigator Michael W. Weiner, MD. Its primary goal has been to test whether serial magnetic resonance imaging (MRI), positron emission tomography (PET), other biological markers and clinical and neuropsychological assessment can be combined to measure the progression of mild cognitive impairment (MCI) and early Alzheimer's disease (AD) [29]. The "gold standard" graph discussed in Shen et al. [30] is displayed in Appendix D, Table 11 and will be used as the target graph in our experiments. The causal connections in the graph are constructed and established from extensively evaluated literature. This dataset contains 9 variables with records of 1008 patients, details about the dataset are shown in Appendix D, Table 11.

Evaluation protocol. To account for variability in our results, 25 independent runs of SMAC are conducted. The budget and trial budget are reported in Appendix E, Table 12. The configurations are evaluated utilising the metrics described in Section 4.2. The evaluation protocol of AutoCD (designed by Hutter et al. [12] and Thornton et al. [1]) incorporates bootstrapping, where 5 configurations are sampled uniformly at random from the 25 configurations. The best configuration is then reported based on the loss, and the process is repeated 1000 times to create a bootstrap distribution. These parameters are also shown in Appendix E, Table 12.

5 Results

This section presents the findings and results from our experiments. We begin with small-scale exploratory experiments for AutoCD on datasets from one DAG. Following this, various comparative evaluations are conducted using the same synthetic datasets and the ADNI dataset.

5.1 Exploratory experiments

Data sample size. We initially experimented (see Figure 1 here and Figure 6 in Appendix F) with selecting the number of instances in the dataset to ensure the feasibility of finding the causal structure based on the synthetic dataset. Assuming that the best possible configuration is found, with nearly infinite data, the estimated causal graph should mirror the target graph. Any decrease in performance can be ascribed to complex graph structures, insufficient data or a suboptimal configuration. AutoCD is applied to the continuous synthetic datasets with 1000 instances to identify the optimal configuration. The results are evaluated utilising the evaluation protocol with 200, 1000 and 10 000 data samples. We observed a notable change in SHD and CA when comparing

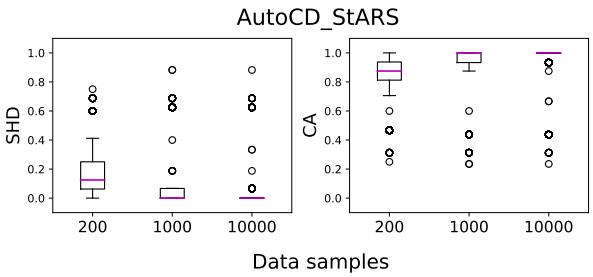

Figure 1: Increasing data sample size on the continuous synthetic dataset with 10 nodes and node degree 3.

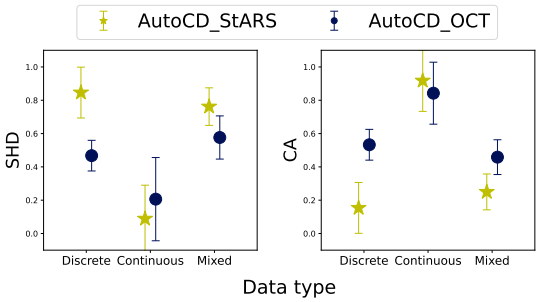

Figure 2: Performance of AutoCD utilising the StARS or OCT loss function on discrete, continuous, and mixed simulated datasets with 10 nodes and node degree 3.

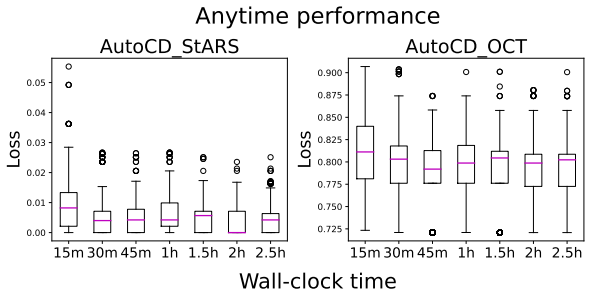

Figure 3: Varying the budget for AutoCD on the mixed synthetic dataset with 10 nodes and node degree 3.

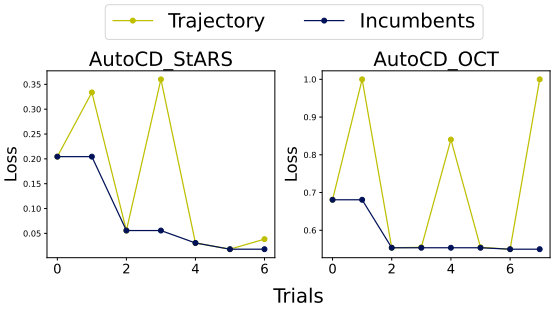

Figure 4: Trajectory of trials and incumbents of the configurator on the continuous synthetic dataset with 10 nodes and node degree 3.

the results obtained for 200 and 10 000 data samples. The mean achieved for 10 000 data samples is lower (SHD) or higher (CA) and the spread is smaller, indicating better performance with less variability. In our subsequent experiments, we utilise a data sample size of 1000 as a compromise between performance and computation time.

Data types. Our next experiment (see Figure 2) aims to investigate the performance of AutoCD on discrete, continuous and mixed synthetic datasets. This distinction is needed, because not all causal discovery algorithms are designed for all data types, with fewer algorithms tailored for mixed datasets. Our findings show that AutoCD utilising either of the loss functions can successfully identify an optimal configuration for the continuous synthetic dataset. However, the high SHD scores and low CA values observed for the discrete and mixed synthetic datasets were unexpected. Prior work showed similar results [3], suggesting that applying post-hoc correction improves the performance on the discrete dataset, which will be addressed in Section 5.4.

Budget size. To determine the optimal time budget for AutoCD (set in SMAC), the experiment is conducted by varying the budgets in minutes (m) and hours (h) within the range [15m, 30m, 45m, 1h, 1.5h, 2h, 2.5h]. At a certain budget, the results plateau, indicating diminishing returns. AutoCD is applied to continuous and mixed synthetic datasets, and the results are evaluated according to the evaluation protocol yielding bootstrap distributions (see Figure 3 and Figure 4). Based on the anytime performance graph (see Figure 3), AutoCD can perform well within a time budget of 1 hour for both loss functions. Based on the results in Figure 4, we can conclude that optimising the loss functions AutoCD$_{StARS}$ and AutoCD$_{OCT}$ leads to better performance in terms of the loss. Considering the trade-off between performance and computation time, in subsequent experiments, all methods and algorithms will adhere to a wall-clock time of 1 hour.

5.2 StARS vs OCT

To determine the most suitable loss function for implementing AutoCD, we perform two experiments to compare the effectiveness of these two loss functions. We use AutoCD$_{StARS}$ and AutoCD$_{OCT}$, to refer to the implementations of AutoCD using the StARS and OCT loss functions, respectively. The experiment uses the synthetic and real-world datasets, comparing the performance of AutoCD$_{StARS}$and AutoCD$_{OCT}$ in terms of SHD, CA, FP and FN. The results are presented in Table 1 (based on aggregating the bootstrap results). These results reveal that, overall, AutoCD$_{OCT}$ achieves better performance than AutoCD$_{StARS}$, except for the continuous case and the ADNI dataset based on SHD. The weaker performance of AutoCD$_{StARS}$ can be explained by the substantial proportion of underpredicted edges, edges that are present in the target but not the predicted graph.

We further compare the average ranking of the methods on all DAGs. Table 2 shows that AutoCD$_{OCT}$ outperforms AutoCD$_{StARS}$ on discrete and mixed synthetic datasets, while AutoCD$_{StARS}$ shows better performance on the continuous synthetic datasets. These ranking results are consistent with the results in Table 1. Therefore, in the next experiments, AutoCD refers to AutoCD$_{OCT}$ when applied to discrete and mixed synthetic datasets, and AutoCD$_{StARS}$ when applied to continuous synthetic datasets.

5.3 AutoCD vs Baselines

The previous analysis helped us determine the loss function for AutoCD based on the data type. Next, we present the results from the experiments outlined in Section 4.

Causal tuning methods. The causal tuning methods compared here are StARS, OCT and AutoCD. Table 3 shows the aggregated bootstrap results on synthetic datasets and the real-world dataset. Both StARS and OCT show inferior performance compared to AutoCD across all datasets in terms of SHD and CA, except for the ADNI dataset, where StARS achieves a higher CA value. The FP and FN rates in Appendix F, Tables 15 and 16 show that AutoCD compared to StARS and OCT obtains the lowest FN rates for 44 out of 46 datasets and the lowest FP rate for 14 out of 15 continuous datasets. The FN rates are consistently higher than the FP rates, indicating that the

Table 1: Performance of AutoCD$_{StARS}$ and AutoCD$_{OCT}$ on synthetic datasets with 10 nodes and node degree 3 and the ADNI dataset. The results in bold indicate better performance tested with the Wilcoxon signed rank test with a significance level of 0.05. The entries show the mean and standard deviation.

	AutoCD$_{StARS}$				AutoCD$_{OCT}$			
	Dis	Con	Mix	ADNI	Dis	Con	Mix	ADNI
SHD	0.85±0.15	**0.09±0.2**	0.76±0.11	**0.83±0.09**	**0.47±0.09**	0.21±0.25	**0.58±0.13**	0.85±0.04
CA	0.15±0.15	**0.92±0.18**	0.25±0.11	0.33±0.03	**0.53±0.09**	0.84±0.19	**0.46±0.1**	**0.34±0.05**
FP	**0.0±0.0**	**0.01±0.02**	**0.0±0.01**	**0.49±0.03**	0.01±0.03	0.14±0.18	0.22±0.23	0.6±0.06
FN	0.85±0.15	0.07±0.16	0.75±0.11	0.18±0.04	**0.45±0.09**	**0.02±0.03**	**0.32±0.22**	**0.05±0.02**

Table 2: Average ranking of loss functions AutoCD$_{StARS}$ *vs* AutoCD$_{OCT}$ in terms of SHD or CA. comparisons are made over 1000 bootstrap samples and all DAGs. The results in bold indicate better performance.

	AutoCD$_{StARS}$			AutoCD$_{OCT}$		
	Dis	Con	Mix	Dis	Con	Mix
SHD	1.99	**1.30**	1.58	**1.01**	1.70	**1.42**
CA	1.99	**1.31**	1.65	**1.01**	1.69	**1.35**

Table 3: Comparing causal tuning methods on synthetic datasets (10 nodes and node degree 3) and the ADNI dataset in terms of SHD and CA. The results in bold indicate better performance according to the Wilcoxon signed rank test (significance level of 0.05).

	StARS		OCT		AutoCD	
	SHD	CA	SHD	CA	SHD	CA
Dis	0.85±0.13	0.15±0.13	0.65±0.15	0.39±0.14	**0.47±0.09**	**0.53±0.09**
Con	0.31±0.34	0.72±0.3	0.68±0.2	0.45±0.19	**0.09±0.2**	**0.92±0.18**
Mix	0.71±0.12	0.33±0.15	0.59±0.11	0.45±0.11	**0.58±0.13**	**0.46±0.1**
ADNI	0.86±0.08	**0.36±0.02**	0.89±0.06	0.34±0.04	**0.85±0.04**	0.34±0.05

estimated causal graphs are missing edges present in the target graph. The relative performance can be seen in Table 5. These results show that AutoCD's overall performance is better than the StARS and OCT baselines.

Individual algorithms. We extended our comparative analysis to include individual algorithms, namely PC, FGES, LiNGAM and GOLEM. The hyperparameters of these algorithms are optimised using AutoCD with a search space including only the respective algorithm. The results shown in Table 4 indicate that AutoCD outperforms individual algorithms on the continuous synthetic dataset and is better on the ADNI dataset in terms of CA. For the discrete synthetic datasets, FGES yields superior performance, while PC achieves better results on the mixed synthetic dataset and the ADNI dataset in terms of SHD. To illustrate the relative performance between the methods, we studied their ranking based on SHD and CA. The results of this analysis are presented in Table 5 (comparing the third, seventh and eighth columns) and demonstrate that AutoCD achieves a better average ranking compared to these two constituting algorithms (i.e., PC and FGES). These results are unexpected, and after inspecting the adequacy of the time budget available to AutoCD, we used these results to propose variants of AutoCD.

5.4 AutoCD vs Variants

Building upon the previous results, we propose several variants of AutoCD. The first of these is called AutoCD+ and includes the post-hoc correction strategies presented in Section 3.3. The second variant, AutoCD$_{PC}$, makes PC, the best-performing method from the previous experiment (see Table 4), the starting point for hyperparameter optimisation. We expect this choice to allow finding a better configuration much faster. The last variant, AutoCD$_{PC}$+, combines the two approaches. The findings in Table 13 show that AutoCD$_{PC}$ performs the best on continuous synthetic datasets (12 out of 15), AutoCD$_{PC}$+ performs the best on discrete synthetic datasets (9 out of 15), and PC performs best on mixed synthetic dataset (13 out of 15). All variants of AutoCD have the same performance on the ADNI dataset, which is better than PC in terms of CA. The relative performance in Table 5 shows that applying the penalty as a post-hoc correction does not improve the performance, but warm-starting the search from PC does, resulting in the best-performing method.

To better understand the reason for this observation, we investigated the trajectory of hyperparameters evaluated during the search. Table 17 in Appendix F shows the losses observed during the optimisation process carried out by SMAC. Comparing the losses of AutoCD$_{PC}$ against PC on the

Table 4: Comparing causal tuning methods and individual algorithms on synthetic datasets with 10 nodes and node degree 3, and the ADNI dataset in terms of SHD and CA. The results in bold indicate better performance using the Wilcoxon signed rank test (significance level of 0.05). The entries show the mean and standard deviation. LiNGAM and GOLEM are only designed for continuous datasets (denoted with n/a).

	AutoCD		PC		FGES		LiNGAM		GOLEM	
	SHD	CA	SHD	CA	SHD	CA	SHD	CA	SHD	CA
Dis	0.47±0.09	0.53±0.09	0.5±0.14	0.54±0.12	**0.39±0.17**	**0.62±0.17**	n/a	n/a	n/a	n/a
Con	**0.09±0.2**	**0.92±0.18**	0.21±0.09	0.79±0.08	0.47±0.23	0.68±0.15	0.85±0.08	0.22±0.06	0.75±0.09	0.43±0.06
Mix	0.58±0.13	0.46±0.1	**0.54±0.1**	**0.51±0.13**	0.67±0.15	0.42±0.11	n/a	n/a	n/a	n/a
ADNI	0.85±0.04	**0.34±0.05**	**0.77±0.03**	0.32±0.02	0.84±0.03	0.29±0.03	n/a	n/a	n/a	n/a

Table 5: Average ranking over all methods. The results are based on SHD and CA , and the comparisons are made over 1000 bootstrap samples and all simulated DAGs. Results shown in bold indicate the best results.

	StARS	OCT	AutoCD	AutoCD+	AutoCD$_{PC}$	AutoCD$_{PC}$+	PC	FGES
SHD	5.33	5.21	3.78	4.12	**3.69**	4.25	4.36	5.25
CA	5.38	5.21	3.80	4.13	**3.72**	4.27	4.33	5.17

Table 6: Performance of AutoCD variants on synthetic datasets (10 nodes and node degree 3), and the ADNI dataset in terms of SHD and CA. The results in bold indicate best performance, according to the Wilcoxon signed rank test with a significance level of 0.05. The entries show the mean and standard deviation.

	AutoCD		AutoCD+		AutoCD$_{PC}$		AutoCD$_{PC}$+		PC	
	SHD	CA	SHD	CA	SHD	CA	SHD	CA	SHD	CA
Dis	0.47±0.09	0.53±0.09	0.47±0.1	0.53±0.1	**0.44±0.11**	**0.56±0.11**	**0.44±0.11**	**0.56±0.11**	0.5±0.14	0.54±0.12
Con	**0.09±0.2**	**0.92±0.18**	0.2±0.3	0.82±0.26	0.12±0.22	0.89±0.19	0.32±0.27	0.69±0.26	0.21±0.09	0.79±0.08
Mix	0.58±0.13	0.46±0.1	0.57±0.12	0.46±0.11	0.61±0.13	0.46±0.12	0.61±0.13	0.47±0.12	**0.54±0.1**	**0.51±0.13**
ADNI	0.85±0.04	**0.34±0.05**	0.85±0.04	**0.34±0.05**	0.85±0.03	**0.34±0.05**	0.85±0.03	**0.34±0.05**	**0.77±0.03**	0.32±0.02

ADNI dataset, equal losses are obtained, which should indicate equal performance. However, according to the evaluation metrics, achieving lower loss values does not necessarily guarantee better performance in uncovering the underlying causal graph. This can be explained by the imperfect loss functions in the unsupervised setting that assess a configuration without the target graph. Furthermore, the target graph may contain errors, even with nearly infinite data, the estimated graph may not allow identifying all causal connections. This points to additional challenges and opportunities for future research in designing new loss functions for AutoCD.

6 Conclusion

In this work, we introduced AutoCD, an AutoML system for causal discovery that encompasses automated algorithm selection and hyperparameter optimisation. To address the suitably reformulated CASH problem for causal discovery, we compared the applicability of two existing loss functions that assess the performance of a given configuration without utilising the ground truth causal graph that is unknown. Moreover, we presented three variants of AutoCD to try to enhance the performance of the basic approach. The first enhancement applies post-hoc correction to the results of AutoCD. This has been proven to increase the performance in combination with grid search [3]. The second enhancement uses PC [5] to warm-start the search of AutoCD. As BO targets a smaller set of configurations, this will guide to higher-performing configurations faster.

We conducted extensive empirical performance analyses to assess our proposed method on synthetically generated datasets and a real-world dataset. The results of these experiments show that (i) AutoCD with an effective search strategy identifies configurations with better performance compared to earlier causal tuning approaches; (ii) AutoCD's overall performance is better than optimised individual causal discovery algorithms; (iii) AutoCD+, a variant of AutoCD that applies a penalty as a post-hoc correction, does not seem effective when used in combination with BO; and (iv) AutoCD$_{PC}$, the improved variant of AutoCD that warm-starts the search from PC, performs overall best in casual tuning. Specifically, AutoCD$_{PC}$ achieves a better average performance rank than the best causal tuning method and the best individual algorithm (3.69 *vs* 5.21 and 4.36, respectively).

Our results show that the best performance in causal discovery cannot be guaranteed. This is due to the imperfect loss functions, calling for future work on designing new loss functions. Moreover, potential errors or noise in the target graph may lead to a more complex problem which we leave for future work.

7 Broader Impact Statement

We introduced a novel AutoML approach for causal discovery, a widely studied and immensely important machine learning task. Our approach identifies high-performing configurations using Bayesian optimisation rather than grid search (as done in existing work). Our new AutoML system makes causal discovery algorithms accessible to a large group of users. After careful reflection, we conclude that this work does not carry any risk of negative impact on society or the environment.

Acknowledgements. This work is part of the project "Physics-aware Spatio-temporal Machine Learning for Earth Observation Data" (with project number OCENW.KLEIN.425) of the research programme Open Competition ENW which is partly financed by the Dutch Research Council (NWO). It was also partially supported by TAILOR, a project funded by EU Horizon 2020 research and innovation programme under GA No 952215.

References

[1] Chris Thornton, Frank Hutter, Holger H. Hoos, and Kevin Leyton-Brown. Auto-WEKA: combined selection and hyperparameter optimization of classification algorithms. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, page 847–855, Chicago, Illinois USA, Aug. 2013. Association for Computing Machinery.

[2] Han Liu, Kathryn Roeder, and Larry Wasserman. Stability approach to regularization selection (StARS) for high dimensional graphical models. In *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc., 2010.

[3] Konstantina Biza, Ioannis Tsamardinos, and Sofia Triantafillou. Out-of-sample tuning for causal discovery. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–11, 2022.

[4] Mitra Baratchi, Can Wang, Steffen Limmer, Jan N van Rijn, Holger Hoos, Thomas Bäck, and Markus Olhofer. Automated machine learning: past, present and future. *Artificial Intelligence Review*, 57(5):1–88, 2024.

[5] Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction, and Search*. The MIT Press, 01 2001.

[6] Jie Ding, Vahid Tarokh, and Yuhong Yang. Model selection techniques: An overview. *IEEE Signal Processing Magazine*, 35(6):16–34, 2018.

[7] Marloes H. Maathuis, Markus Kalisch, and Peter Bühlmann. Estimating high-dimensional intervention effects from observational data. *The Annals of Statistics*, 37(6A):3133 – 3164, 2009.

[8] Nicolai Meinshausen and Peter Bühlmann. Stability Selection. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 72(4):417–473, 08 2010.

[9] Dimitris N. Politis, Joseph P. Romano, and Michael Wolf. Subsampling. *Springer Series in Statistics*, 1999.

[10] Vineet K. Raghu, Allen Poon, and Panayiotis V. Benos. Evaluation of causal structure learning methods on mixed data types. In *Proceedings of 2018 ACM SIGKDD Workshop on Causal Disocvery*, volume 92 of *Proceedings of Machine Learning Research*, pages 48–65, London, UK, Aug. 2018. PMLR.

[11] Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren. *Automated machine learning: methods, systems, challenges*. Springer Nature, 2019.

[12] Frank Hutter, Holger H Hoos, and Kevin Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *Learning and Intelligent Optimization: 5th International Conference*, pages 507–523, Rome, Italy, Jan. 2011. Springer.

[13] Carl Edward Rasmussen. Gaussian processes for machine learning. In *Summer School on Machine Learning*, pages 63–71. Springer, 2003.

[14] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. In *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011.

[15] Xilu Wang, Yaochu Jin, Sebastian Schmitt, and Markus Olhofer. Recent advances in Bayesian optimization. *ACM Comput. Surv.*, 55(13s), Jul. 2023.

[16] Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Springenberg, Manuel Blum, and Frank Hutter. Efficient and robust automated machine learning. In *Advances in Neural Information Processing Systems*, volume 28, pages 2962–2970. Curran Associates, Inc., 2015.

[17] Lucas Zimmer, Marius Lindauer, and Frank Hutter. Auto-Pytorch: Multi-fidelity metalearning for efficient and robust AutoDL. *IEEE transactions on pattern analysis and machine intelligence*, 43(9):3079—3090, Sep. 2021.

[18] Marcilio C. P. de Souto, Ricardo B. C. Prudencio, Rodrigo G. F. Soares, Daniel S. A. de Araujo, Ivan G. Costa, Teresa B. Ludermir, and Alexander Schliep. Ranking and selecting clustering algorithms using a meta-learning approach. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 3729–3735, Hong Kong, China, Jun. 2008.

[19] Daniel Gomes Ferrari and Leandro Nunes de Castro. Clustering algorithm selection by meta-learning systems. *Inf. Sci.*, 301(C):181–194, apr 2015.

[20] Yannis Poulakis, Christos Doulkeridis, and Dimosthenis Kyriazis. AutoClust: A framework for automated clustering based on cluster validity indices. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 1220–1225, Sorrento, Italy, Nov. 2020.

[21] Yu-Feng Li, Hai Wang, Tong Wei, and Wei-Wei Tu. Towards automated semi-supervised learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4237–4244, Hawaii, USA, Jul. 2019.

[22] Joseph D Ramsey, Kun Zhang, Madelyn Glymour, Ruben Sanchez Romero, Biwei Huang, Imme Ebert-Uphoff, Savini Samarasinghe, Elizabeth A Barnes, and Clark Glymour. TETRAD—a toolbox for causal discovery. In *8th international workshop on climate informatics*, page 29, Boulder, Colorado, USA, Nov. 2018.

[23] Keli Zhang, Shengyu Zhu, Marcus Kalander, Ignavier Ng, Junjian Ye, Zhitang Chen, and Lujia Pan. gCastle: A Python toolbox for causal discovery. *CoRR*, abs/2111.15155, 2021.

[24] Joseph Ramsey, Madelyn Glymour, Ruben Sanchez-Romero, and Clark Glymour. A million variables and more: the fast greedy equivalence search algorithm for learning high-dimensional graphical causal models, with an application to functional magnetic resonance images. *International Journal of Data Science and Analytics*, 3(2):121–129, Mar. 2017.

[25] Shohei Shimizu, Patrik O. Hoyer, Aapo Hyvärinen, and Antti Kerminen. A linear non-Gaussian acyclic model for causal discovery. *Journal of Machine Learning Research*, 7(72):2003–2030, 2006.

[26] Ignavier Ng, AmirEmad Ghassami, and Kun Zhang. On the role of sparsity and DAG constraints for learning linear dags. In *Advances in Neural Information Processing Systems*, volume 33, pages 17943–17954. Curran Associates, Inc., 2020.

[27] Lu Cheng, Ruocheng Guo, Raha Moraffah, Paras Sheth, K. Selçuk Candan, and Huan Liu. Evaluation methods and measures for causal learning algorithms. *IEEE Transactions on Artificial Intelligence*, 3(6):924–943, 2022.

[28] Joseph Ramsey and Bryan Andrews. Py-tetrad and rpy-tetrad: A new python interface with r support for tetrad causal search. In *Proceedings of the 2023 Causal Analysis Workshop Series*, volume 223 of *Proceedings of Machine Learning Research*, pages 40–51, Virtual event, Aug. 2023. PMLR.

[29] Michael W. Weiner, Dallas P. Veitch, Paul S. Aisen, Laurel A. Beckett, Nigel J. Cairns, Jesse Cedarbaum, Michael C. Donohue, Robert C. Green, Danielle Harvey, Clifford R. Jack, William Jagust, John C. Morris, Ronald C. Petersen, Andrew J. Saykin, Leslie Shaw, Paul M. Thompson, Arthur W. Toga, and John Q. Trojanowski. Impact of the alzheimer's disease neuroimaging initiative, 2004 to 2014. *Alzheimer's Dementia*, 11(7):865–884, 2015.

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

[31] Joseph D. Ramsey. Improving accuracy of permutation DAG search using best order score search. *CoRR*, abs/2108.10141, 2021.

[32] 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*, UAI'06, page 401–408, Cambridge, MA USA, Jul 2006. AUAI Press.

[33] Kevin Bello, Bryon Aragam, and Pradeep Ravikumar. DAGMA: Learning DAGs via M-matrices and a log-determinant acyclicity characterization. In *Advances in Neural Information Processing Systems*, volume 35, pages 8226–8239. Curran Associates, Inc., 2022.

[34] Shohei Shimizu, T Inazumi, Y Sogawa, Aapo Hyvärinen, Y Kawahara, T Washio, Patrik Hoyer, and K Bollen. Directlingam: A direct method for learning a linear non-Gaussian structural equation model. *Journal of Machine Learning Research*, 12, 2011.

[35] Wai-Yin Lam, Bryan Andrews, and Joseph Ramsey. Greedy relaxations of the sparsest permutation algorithm. In *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*, volume 180 of *Proceedings of Machine Learning Research*, pages 1052–1062, Eindhoven, Netherlands, Aug. 2022. PMLR.

Submission Checklist

1. For all authors...

   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]

   (b) Did you describe the limitations of your work? [Yes] We mentioned that lower loss values do not correspond to better results because of the absence of the target. Additionally, our work only considers the graph model DAG.

   (c) Did you discuss any potential negative societal impacts of your work? [N/A]

   (d) Did you read the ethics review guidelines and ensure that your paper conforms to them? `https://2022.automl.cc/ethics-accessibility/` [Yes]

2. If you ran experiments...

   (a) Did you use the same evaluation protocol for all methods being compared (e.g., same benchmarks, data (sub)sets, available resources)? [Yes] See Section 4

   (b) Did you specify all the necessary details of your evaluation (e.g., data splits, pre-processing, search spaces, hyperparameter tuning)? [Yes] See Section 3 and Section 4.

   (c) Did you repeat your experiments (e.g., across multiple random seeds or splits) to account for the impact of randomness in your methods or data? [Yes] We utilised 25 repetitions and used bootstrapping Section 4.

   (d) Did you report the uncertainty of your results (e.g., the variance across random seeds or splits)? [Yes] We report the standard deviations in Section 5.

   (e) Did you report the statistical significance of your results? [Yes] We used the Wilcoxon signed rank test to test for significance in Section 5.

   (f) Did you use tabular or surrogate benchmarks for in-depth evaluations? [Yes] We generated our synthetic datasets in Section 4.

   (g) Did you compare performance over time and describe how you selected the maximum duration? [Yes] We explored different budgets in Section 4.

   (h) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] We mentioned the resources in Section 4.

   (i) Did you run ablation studies to assess the impact of different components of your approach? [Yes] We improved our proposed method in Section 4.

3. With respect to the code used to obtain your results...

   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results, including all requirements (e.g., `requirements.txt` with explicit versions), random seeds, an instructive `README` with installation, and execution commands (either in the supplemental material or as a URL)? [Yes] It is on GitHub: `https://github.com/Gerlise/AutoCD`

   (b) Did you include a minimal example to replicate results on a small subset of the experiments or on toy data? [Yes] It is included under the example folder. It applies AutoCD on one mixed dataset with 5 nodes and node degree 2.

   (c) Did you ensure sufficient code quality and documentation so that someone else can execute and understand your code? [Yes]

(d) Did you include the raw results of running your experiments with the given code, data, and instructions? [Yes] This link to the raw results: `https://ufile.io/dw5soc5d`

(e) Did you include the code, additional data, and instructions needed to generate the figures and tables in your paper based on the raw results? [Yes]

4. If you used existing assets (e.g., code, data, models)…

(a) Did you cite the creators of used assets? [Yes] We utilises the libraries SMAC [12], Tetrad [22], Py-tetrad [28], and gCastle [23]. The citations appear in the main body of the paper.

(b) Did you discuss whether and how consent was obtained from people whose data you're using/curating if the license requires it? [Yes] We acquired the ADNI dataset and put the acknowledgment in Section 4.

(c) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] The acquired ADNI dataset is already anonymised.

5. If you created/released new assets (e.g., code, data, models)…

(a) Did you mention the license of the new assets (e.g., as part of your code submission)? [No] We did not mention it in the paper, but SMAC uses the BSD (3-Clause) license, Tetrad uses the GNU General Public v2.0 license, Py-tetrad uses the MIT license, and gCastle uses the Apache-2.0 license

(b) Did you include the new assets either in the supplemental material or as a URL (to, e.g., GitHub or Hugging Face)? [Yes]

6. If you used crowdsourcing or conducted research with human subjects…

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

7. If you included theoretical results…

(a) Did you state the full set of assumptions of all theoretical results? [N/A]

(b) Did you include complete proofs of all theoretical results? [N/A]

A Algorithms

This appendix provides the algorithms for the loss functions and penalties.

Algorithm 1: Loss function extracted from StARS [3].

Input: Dataset $\mathcal{D}$, number of subsamples S, configuration $A_{\boldsymbol{\lambda}}$

for $i \in S$ **do**
  $\mathcal{G}^{(i)}_{A_{\boldsymbol{\lambda}}} \leftarrow \text{fit}(A_{\boldsymbol{\lambda}}, \mathcal{D}^{(i)})$
  $Q^{(i)}_{A_{\boldsymbol{\lambda}}} \leftarrow$ number of edges in $\mathcal{G}^{(i)}_{A_{\boldsymbol{\lambda}}}$
end
$Q(A_{\boldsymbol{\lambda}}) \leftarrow$ average $Q^{(i)}_{A_{\boldsymbol{\lambda}}}$ over S
for *each pair of variables X, Y* **do**
  $p_{A_{\boldsymbol{\lambda}},X,Y} \leftarrow$ frequency of edge (X, Y) in $\{\mathcal{G}^{(i)}_{A_{\boldsymbol{\lambda}}}\}_{i \in S}$
  $\xi_{A_{\boldsymbol{\lambda}},X,Y} = 2 \cdot p_{A_{\boldsymbol{\lambda}},X,Y} \cdot (1 - p_{A_{\boldsymbol{\lambda}},X,Y})$
end
$N(A_{\boldsymbol{\lambda}}) \leftarrow$ average $\xi_{A_{\boldsymbol{\lambda}},X,Y}$ over all edges
return $N(A_{\boldsymbol{\lambda}})$

Algorithm 2: Loss function extracted from OCT [3]

Input: Dataset $\mathcal{D}$ over variables $\mathbf{V}$, number of folds K, configuration $A_{\boldsymbol{\lambda}}$

for $i \in K$ **do**
  $\mathcal{G}^{(i)}_{A_{\boldsymbol{\lambda}}} \leftarrow \text{fit}(A_{\boldsymbol{\lambda}}, \mathcal{D}^{(i)}_{train})$
  for $X \in \mathbf{V}$ **do**
    $\text{MB}^{(i)}_{A_{\boldsymbol{\lambda}},X} \leftarrow \text{markovBoundary}(X, \mathcal{G}^{(i)}_{A_{\boldsymbol{\lambda}}})$
    $\mathcal{M}^{(i)}_{A_{\boldsymbol{\lambda}},X} \leftarrow \text{randomForest}(X, \text{MB}^{(i)}_{A_{\boldsymbol{\lambda}},X})$
    $\hat{X}^{(i)}_{A_{\boldsymbol{\lambda}}} \leftarrow \text{predict}(\mathcal{M}^{(i)}_{A_{\boldsymbol{\lambda}},X}, \mathcal{D}^{(i)}_{valid})$
  end
end
for $X \in \mathbf{V}$ **do**
  $|\text{MB}(A_{\boldsymbol{\lambda}})| \leftarrow$ average $|\text{MB}^{(i)}_{A_{\boldsymbol{\lambda}},X}|$ over $\mathbf{V}$
  $\hat{X}(A_{\boldsymbol{\lambda}}) \leftarrow$ average $\hat{X}^{(i)}_{A_{\boldsymbol{\lambda}}}$ over K
  $I_{A_{\boldsymbol{\lambda}},X} \leftarrow \text{mutualInformation}(X, \hat{X}(A_{\boldsymbol{\lambda}}))$
end
$I(A_{\boldsymbol{\lambda}}) \leftarrow$ average $I_{A_{\boldsymbol{\lambda}},X}$ over $\mathbf{V}$
return $I(A_{\boldsymbol{\lambda}})$

Algorithm 3: Penalty from the StARS method [3].

Input: Configurations $\boldsymbol{A}$, density estimation Q, network instability N, threshold β

Rank $N(A_{\boldsymbol{\lambda}})$ by increasing $Q(A_{\boldsymbol{\lambda}})$ $\forall A_{\boldsymbol{\lambda}} \in \boldsymbol{A}$
$N' \leftarrow \text{monotonise}(N)$
$A^*_{\boldsymbol{\lambda}} \leftarrow \underset{A_{\boldsymbol{\lambda}} \in \boldsymbol{A}}{\text{argmax}}\{N'(A_{\boldsymbol{\lambda}}) | N'(A_{\boldsymbol{\lambda}}) \leq \beta\}$
return $A^*_{\boldsymbol{\lambda}}$

Algorithm 4: Penalty from the OCT method [3].

Input: Optimal configuration $A^*_{\boldsymbol{\lambda}}$, configurations $\boldsymbol{A}$, mutual information I, true variables $\mathbf{V}$, predictions $\hat{X}$, Markov boundary sizes |MB|, threshold β, number of permutations P

for $A_{\boldsymbol{\lambda}} \in \boldsymbol{A} \backslash A^*_{\boldsymbol{\lambda}}$ **do**
  $T_{A_{\boldsymbol{\lambda}}} \leftarrow (I(A^*_{\boldsymbol{\lambda}}) - I(A_{\boldsymbol{\lambda}}))$
  for $p = 1, ..., P$ **do**
    for $X \in \mathbf{V}$ **do**
      $\hat{X}'(A^*_{\boldsymbol{\lambda}})\ ,\ \hat{X}'(A_{\boldsymbol{\lambda}}) \leftarrow \text{swap}(\hat{X}(A^*_{\boldsymbol{\lambda}}), \hat{X}(A_{\boldsymbol{\lambda}}))$
      $I'_{A^*_{\boldsymbol{\lambda}},p,X}\ ,\ I'_{A_{\boldsymbol{\lambda}},p,X} \leftarrow$ mutualInformation of $\hat{X}'(A^*_{\boldsymbol{\lambda}})$ and $\hat{X}'(A_{\boldsymbol{\lambda}})$
    end
    $I'(A^*_{\boldsymbol{\lambda}}, p)\ ,\ I'(A_{\boldsymbol{\lambda}}, p) \leftarrow$ average $I'_{A^*_{\boldsymbol{\lambda}},p,X}$ and $I'_{A_{\boldsymbol{\lambda}},p,X}$ over $\mathbf{V}$
    $T'_{A_{\boldsymbol{\lambda}}}(p) \leftarrow (I'(A^*_{\boldsymbol{\lambda}}, p) - I'(A_{\boldsymbol{\lambda}}, p))$
  end
  $p_{val}(A_{\boldsymbol{\lambda}}) \leftarrow |T'_{A_{\boldsymbol{\lambda}}} \geq T_{A_{\boldsymbol{\lambda}}}|/P$
end
$A^*_{\boldsymbol{\lambda}} \leftarrow p_{val}(A_{\boldsymbol{\lambda}}) > \beta$ and $A_{\boldsymbol{\lambda}} = \arg\min_{A_{\boldsymbol{\lambda}}} |\text{MB}|$
return $A^*_{\boldsymbol{\lambda}}$

B The search space of AutoCD

This appendix provides the search space of AutoCD, showing the causal discovery algorithms and the hyperparameters. The hyperparameters can be 'int' (integer), 'cat' (categorical), 'real' (real), 'bin' (binary), and 'cond' (conditional).

Table 7: The AutoCD search space includes 11 causal discovery algorithms sourced from Tetrad and gCastle. The hyperparameters, type, and a short description are given (1/2).

Algorithm	Hyperparameter	Type	Description
AutoCD	`algorithm`	cat	Algorithm selection
BOSS [31]	`score`	cat	Scoring function
	`penalty`	real	Penalty discount
	`structure`	real	Structure prior coefficient
	`data_order`	bin	Use data order or random permutation
	`n_start_thread`	int	Number of random starts and threads
CPC [32]	`alpha`	real	Cutoff for p-values
	`test`	cat	Conditional independence test
	`rule`	cat	Conflict rule
	`b_type`	cond	Basis type (1 = Polynomial, 2 = Cosine)
	`b_num_func`	cond	Number of functions to use in the basis
	`k_type`	cond	Kernel type (1 = Gaussian, 2 = Epinechnikov)
	`k_multiplier`	cond	Capture more or less than the optimal signal
	`k_sample_size`	cond	Minimum sample size for kernel regression
CPCstable [32]	`alpha`	real	Cutoff for p-values
	`test`	cat	Conditional independence test
	`rule`	cat	Conflict rule
	`b_type`	cond	Basis type (1 = Polynomial, 2 = Cosine)
	`b_num_func`	cond	Number of functions to use in the basis
	`k_type`	cond	Kernel type (1 = Gaussian, 2 = Epinechnikov)
	`k_multiplier`	cond	Capture more or less than the optimal signal
	`k_sample_size`	cond	Minimum sample size for kernel regression

Table 8: The AutoCD search space includes 11 causal discovery algorithms sourced from Tetrad and gCastle. The hyperparameters, type, and a short description are given (2/2).

Algorithm	Hyperparameter	Type	Description
Dagma [33]	lambda	real	L1 penalty coefficient
	w_thresh	real	Threshold for entries in W matrix
Direct LiNGAM [34]	score	cat	Scoring function
	penalty	real	Penalty discount
	structure	real	Structure prior coefficient
FGES [24]	score	cat	Scoring function
	penalty	real	Penalty discount
	structure	real	Structure prior
GOLEM [26]	lambda_1	real	L1 penalty coefficient
	lambda_2	real	DAG penalty coefficient
	learning_rate	real	Learning rate of Adam optimiser
	num_iter	int	Number of iterations for training
	graph_thres	real	Threshold for weighted matrix
GrASP [35]	test	cat	Conditional independence test
	score	cat	Scoring function
	alpha	real	Cutoff for p-value
	penalty	real	Penalty discount
	structure	real	Structure prior coefficient
	b_type	cond	Basis type (1 = Polynomial, 2 = Cosine)
	b_num_func	cond	Number of functions to use in the basis
	k_type	cond	Kernel type (1 = Gaussian, 2 = Epinechnikov)
	k_multiplier	cond	Capture more or less than the optimal signal
	k_sample_size	cond	Minimum sample size for kernel regression
	n_starts	int	Number of restarts
ICA-LiNGAM [25]	alpha_ica	real	Threshold for entries in B matrix
	max_iter	real	Maximum iterations for orienting edges
	tolerance	real	Fast ICA tolerance parameter
	b_thresh	real	Threshold for entries in B matrix
PC [5]	alpha	real	Cutoff for p-values
	test	cat	Conditional independence test
	rule	cat	Conflict rule
	b_type	cond	Basis type (1 = Polynomial, 2 = Cosine)
	b_num_func	cond	Number of functions to use in the basis
	k_type	cond	Kernel type (1 = Gaussian, 2 = Epinechnikov)
	k_multiplier	cond	Capture more or less than the optimal signal
	k_sample_size	cond	Minimum sample size for kernel regression
PCstable [5]	alpha	real	Cutoff for p-values
	test	cat	Conditional independence test
	rule	cat	Conflict rule
	b_type	cond	Basis type (1 = Polynomial, 2 = Cosine)
	b_num_func	cond	Number of functions to use in the basis
	k_type	cond	Kernel type (1 = Gaussian, 2 = Epinechnikov)
	k_multiplier	cond	Capture more or less than the optimal signal
	k_sample_size	cond	Minimum sample size for kernel regression

Table 9: The hyperparameters and values considered in AutoCD sourced from Tetrad and gCastle are given.

Hyperparameter	Values
algorithm	{boss, cpc, cpcstable, dagma, direct_lingam, fges, golem, grasp, ica_lingam, pc, pcstable}
score	{bdeu-score, disc-bic-score, sem-bic-score, cg-bic-score, dg-bic-score}
penalty	[0.0, 2.0]
structure	[0.0, 2.0]
data_order	{True, False}
n_start_thread	[1, 10]
alpha	[0.01, 0.05]
test	{chi-square-test, g-square-test, cg-lr-test, dg-lr-test, fisher-z-test, cci-test}
rule	{1, 2, 3}
b_type	{2}
b_num_func	{30}
k_type	{1}
k_multiplier	{1}
k_sample_size	{100}
lambda	[0.01, 0.05]
w_thresh	[0.1, 0.6]
lambda_1	[0.01, 0.05]
lambda_2	[1.0, 5.0]
learning_rate	[0.001, 0.005]
num_iter	[500, 2000]
graph_thres	[0.1, 0.5]
n_starts	[1, 10]
alpha_ica	[1.0, 2.0]
max_iter	[2000.0, 5000.0]
tolerance	[1e-08, 1e-6]
b_thresh	[0.1, 0.6]

C Data generation

This appendix provides the parameters for the graphical models to simulate datasets.

Table 10: Fixed parameters for synthetic data generation.

Parameter	Data type	Nodes	Avg. node degree	Instances	Categories	Discrete %	Seed	Repetitions
Value	{Dis, Con, Mix}	{5, 10, 20, 30, 40}	{2, 3, 4}	1000	[2, 20]	50	[0, 24]	25

D ADNI

This appendix provides the exploratory analysis of the Alzheimer's Disease Neuroimaging Initiative (ADNI) dataset. The ADNI dataset is divided into three trials: ADNI1, ADNI2/GO, and ADNI3. In our experiments, we used the first two and extracted variables fludeoxyglucose PET (FDG), amyloid beta (ABETA), phosphorylated tau (PTAU), apolipoprotein E $\varepsilon 4$ allele (APOE4), AGE, SEX, education (EDU), and diagnosis on Alzheimer's disease (DX). Table 11 shows the statistics of the dataset. Like in Shen et al. [30], records with missing values are removed resulting in 1008 remaining participants. The variable APOE4 is categorical with 0, 1, 2 which indicates the number of APOE4. The variable DX is also categorical with cognitively normal (CN), mild cognitive impairment (MCI), and early Alzheimer's disease (AD). The target graph from Shen et al. [30] is displayed in Figure 5.

Table 11: Statistics of the ADNI datasets.

Demographics	AGE	72.98±7.25
	SEX	0.56±0.50
	EDU	16.13±2.73
Biomarkers	FDG	1.21±0.16
	ABETA	1000.53±456.3
	PTAU	27.40±14.61
Genetics	APOE4	0 (54%)
		1 (36%)
		2 (10%)
Diagnosis	DX	CN (31%)
		MCI (51%)
		AD (18%)

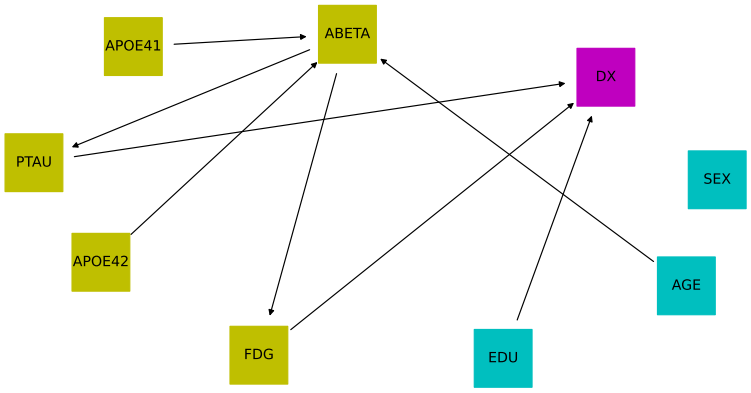

Figure 5: Target graph of the ADNI dataset by Shen et al. [30].

E Evaluation protocol

This appendix provides the fixed parameters for SMAC and the evaluation protocol.

Table 12: Fixed parameters for the SMAC procedure and the evaluation protocol.

	SMAC procedure				Evaluation protocol	
Parameter	Budget	Trial budget	Runs	Seed	Sample size	Samples
Value	60 min.	15 min.	25	[0, 24]	5	1000

F Additional results

This appendix provides additional figures and tables from the results section.

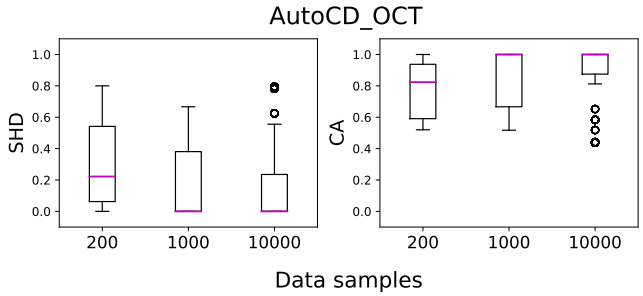

Figure 6: Increasing the data sample size on the continuous simulated dataset with 10 nodes and node degree 3. Increasing the data sample size reveals better performance.

Table 13: Performance of all methods on all synthetic datasets and the ADNI dataset based on SHD. The best values are marked in bold. Empty entries indicate no configuration found. LiNGAM and GOLEM are only designed for continuous datasets (denoted with n/a).

	Causal tuning methods						Individual algorithms			
Dataset	StARS	OCT	AutoCD	AutoCD+	AutoCD$_{PC}$	AutoCD$_{PC}$+	PC	FGES	LiNGAM	GOLEM
con_5_2	0.28±0.33	0.61±0.2	0.07±0.12	0.15±0.2	**0.06±0.11**	0.45±0.24	0.13±0.11	0.37±0.26	0.62±0.17	0.49±0.28
con_5_3	0.29±0.24	0.46±0.25	**0.25±0.22**	0.44±0.2	0.28±0.26	0.46±0.17	0.32±0.18	0.36±0.29	0.64±0.18	0.63±0.1
con_5_4	0.46±0.14	0.4±0.17	0.53±0.17	0.57±0.12	0.52±0.21	0.58±0.17	0.19±0.07	**0.15±0.07**	0.34±0.17	0.43±0.15
con_10_2	0.39±0.31	0.44±0.33	0.04±0.14	0.21±0.28	**0.02±0.04**	0.22±0.25	0.12±0.06	0.4±0.2	0.79±0.09	0.6±0.09
con_10_3	0.31±0.34	0.68±0.2	**0.09±0.2**	0.2±0.3	0.12±0.22	0.32±0.27	0.21±0.09	0.47±0.23	0.85±0.08	0.75±0.09
con_10_4	0.57±0.3	0.6±0.24	**0.06±0.14**	0.61±0.37	**0.06±0.14**	0.58±0.33	0.41±0.07	0.48±0.25	0.76±0.14	0.79±0.04
con_20_2	0.06±0.15	0.14±0.19	**0.0±0.0**	0.04±0.11	**0.0±0.0**	0.25±0.18	0.09±0.07	0.61±0.26	0.88±0.05	0.66±0.08
con_20_3	0.14±0.22	0.3±0.29	**0.03±0.06**	0.06±0.13	**0.03±0.05**	**0.03±0.05**	0.23±0.1	0.25±0.17	0.9±0.04	0.78±0.07
con_20_4	0.11±0.22	0.55±0.31	**0.0±0.01**	0.06±0.17	**0.0±0.01**	0.09±0.24	0.37±0.1	0.53±0.13	0.88±0.03	0.78±0.06
con_30_2	0.03±0.14	0.28±0.29	**0.0±0.01**	0.07±0.13	**0.0±0.01**	0.14±0.16	0.19±0.07	0.56±0.16	0.88±0.03	0.7±0.05
con_30_3	0.11±0.24	0.27±0.24	0.02±0.04	0.06±0.12	**0.01±0.02**	0.05±0.11	0.13±0.04	0.19±0.16	0.91±0.03	
con_30_4	0.06±0.16	0.22±0.22	**0.02±0.08**	0.07±0.19	**0.02±0.08**	0.05±0.17	0.27±0.05	0.35±0.2	0.91±0.04	0.85±0.04
con_40_2	0.01±0.03	0.11±0.15	**0.0±0.01**	0.03±0.07	**0.0±0.01**	0.08±0.1	0.21±0.09	0.36±0.23		
con_40_3	0.05±0.06	0.34±0.29	**0.01±0.05**	0.03±0.06	**0.01±0.04**	0.02±0.06	0.08±0.04	0.12±0.07		
con_40_4	0.07±0.14	0.3±0.31	0.02±0.11	0.08±0.2	**0.01±0.06**	0.04±0.1	0.23±0.03	0.29±0.19		
dis_5_2	0.68±0.3	0.35±0.21	0.13±0.2	0.12±0.2	0.09±0.19	0.1±0.19	0.1±0.19	**0.08±0.14**	n/a	n/a
dis_5_3	0.84±0.18	0.42±0.23	**0.27±0.19**	0.28±0.2	0.29±0.17	**0.27±0.18**	0.33±0.18	0.36±0.14	n/a	n/a
dis_5_4	0.89±0.06	0.7±0.16	0.36±0.2	0.39±0.21	0.27±0.19	0.3±0.2	0.22±0.21	**0.17±0.2**	n/a	n/a
dis_10_2	0.58±0.19	0.66±0.12	0.26±0.12	0.26±0.13	**0.23±0.14**	**0.23±0.14**	0.31±0.18	0.26±0.21	n/a	n/a
dis_10_3	0.85±0.13	0.65±0.15	0.47±0.09	0.47±0.1	0.44±0.11	0.44±0.11	0.5±0.14	**0.39±0.17**	n/a	n/a
dis_10_4	0.83±0.16	0.75±0.14	0.62±0.11	0.62±0.12	0.61±0.11	**0.6±0.11**	0.64±0.15	0.64±0.12	n/a	n/a
dis_20_2	0.38±0.12	0.43±0.16	0.29±0.12	0.3±0.13	0.26±0.13	**0.25±0.12**	0.37±0.07	0.29±0.16	n/a	n/a
dis_20_3	0.75±0.13	0.67±0.08	0.55±0.06	0.56±0.06	0.55±0.05	0.55±0.06	0.55±0.08	**0.54±0.08**	n/a	n/a
dis_20_4	0.82±0.09	0.74±0.06	0.66±0.08	0.66±0.07	**0.62±0.07**	**0.62±0.07**	0.71±0.05	0.68±0.08	n/a	n/a
dis_30_2	0.52±0.16	0.55±0.1	**0.35±0.06**	**0.35±0.07**	**0.35±0.06**	**0.35±0.06**	0.37±0.05	0.39±0.08	n/a	n/a
dis_30_3	0.71±0.09	0.66±0.09	0.55±0.08	**0.54±0.09**	0.56±0.09	0.57±0.1	0.56±0.07	0.57±0.08	n/a	n/a
dis_30_4	0.77±0.06	0.76±0.04	**0.69±0.06**	**0.69±0.06**	**0.69±0.06**	**0.69±0.06**	0.74±0.03	0.7±0.07	n/a	n/a
dis_40_2	0.61±0.13	0.45±0.13	0.37±0.07	0.37±0.07	0.38±0.08	0.38±0.07	0.39±0.06	**0.36±0.11**	n/a	n/a
dis_40_3	0.76±0.07	0.58±0.08	0.57±0.09	0.57±0.09	0.58±0.09	0.58±0.1	0.6±0.08	**0.53±0.07**	n/a	n/a
dis_40_4	0.85±0.08	0.73±0.06	**0.68±0.06**	**0.68±0.06**	0.69±0.07	0.69±0.07	0.74±0.04	0.72±0.06	n/a	n/a
mix_5_2	0.78±0.22	0.57±0.18	0.54±0.13	0.54±0.13	**0.5±0.2**	0.54±0.24	0.54±0.14	0.54±0.16	n/a	n/a
mix_5_3	0.8±0.08	0.62±0.11	0.54±0.09	0.57±0.07	**0.52±0.08**	0.55±0.1	0.6±0.09	0.53±0.12	n/a	n/a
mix_5_4	0.72±0.1	0.56±0.17	0.5±0.1	0.49±0.11	0.45±0.15	0.51±0.09	0.5±0.1	**0.42±0.11**	n/a	n/a
mix_10_2	0.63±0.14	0.57±0.1	0.59±0.17	0.63±0.15	0.62±0.16	0.64±0.11	**0.53±0.12**	0.71±0.1	n/a	n/a
mix_10_3	0.71±0.12	0.59±0.11	0.58±0.13	0.57±0.12	0.61±0.13	0.61±0.13	**0.54±0.1**	0.67±0.15	n/a	n/a
mix_10_4	0.68±0.12	0.59±0.1	0.55±0.09	0.55±0.09	0.57±0.07	0.56±0.07	**0.5±0.08**	0.62±0.09	n/a	n/a
mix_20_2	0.61±0.09	**0.53±0.07**	0.74±0.14	0.74±0.14	0.74±0.14	0.75±0.14	0.62±0.06	0.81±0.12	n/a	n/a
mix_20_3	0.64±0.06	0.64±0.07	0.75±0.1	0.74±0.09	0.78±0.1	0.75±0.1	**0.62±0.06**	0.83±0.08	n/a	n/a
mix_20_4	**0.6±0.09**	0.61±0.11	0.66±0.12	0.66±0.12	0.67±0.12	0.66±0.12	0.61±0.1	0.78±0.09	n/a	n/a
mix_30_2	**0.61±0.09**	0.62±0.05	0.78±0.1	0.77±0.11	0.73±0.12	0.74±0.13	0.66±0.05	0.84±0.12	n/a	n/a
mix_30_3	0.66±0.06	0.66±0.09	0.7±0.11	0.69±0.11	0.7±0.1	0.71±0.11	**0.63±0.03**	0.84±0.08	n/a	n/a
mix_30_4	0.65±0.06	0.61±0.04	0.67±0.1	0.66±0.1	0.66±0.1	0.66±0.09	**0.59±0.06**	0.82±0.06	n/a	n/a
mix_40_2	**0.6±0.07**	0.63±0.06	0.76±0.12	0.75±0.12	0.76±0.12	0.76±0.12	0.71±0.05	0.91±0.05	n/a	n/a
mix_40_3	0.64±0.06	**0.6±0.03**	0.7±0.14	0.7±0.14	0.69±0.12	0.69±0.13	0.63±0.06	0.83±0.12	n/a	n/a
mix_40_4	0.67±0.05	0.66±0.05	0.66±0.09	0.66±0.09	0.65±0.07	0.66±0.07	**0.63±0.04**	0.76±0.1	n/a	n/a
ADNI	0.86±0.08	0.89±0.06	0.85±0.04	0.85±0.04	0.85±0.03	0.85±0.03	**0.77±0.03**	0.84±0.03	n/a	n/a

Table 14: Performance of all methods on all synthetic datasets and the ADNI dataset based on CA. The best values are marked in bold. Empty entries indicate no configuration found. LiNGAM and GOLEM are only designed for continuous datasets (denoted with n/a)

	Causal tuning methods						Individual algorithms			
Dataset	StARS	OCT	AutoCD	AutoCD+	AutoCD$_{PC}$	AutoCD$_{PC}$+	PC	FGES	LiNGAM	GOLEM
con_5_2	0.8±0.22	0.46±0.17	0.93±0.12	0.86±0.18	**0.94±0.11**	0.6±0.19	0.87±0.11	0.72±0.19	0.45±0.16	0.57±0.24
con_5_3	0.74±0.22	0.64±0.18	0.76±0.22	0.58±0.2	0.72±0.26	0.55±0.17	0.77±0.1	**0.79±0.16**	0.51±0.17	0.52±0.12
con_5_4	0.54±0.14	0.6±0.17	0.47±0.17	0.43±0.12	0.48±0.21	0.42±0.17	0.81±0.07	**0.85±0.07**	0.66±0.17	0.57±0.15
con_10_2	0.74±0.23	0.64±0.3	0.96±0.13	0.82±0.26	**0.98±0.04**	0.79±0.25	0.89±0.06	0.62±0.2	0.29±0.05	0.52±0.16
con_10_3	0.72±0.3	0.45±0.19	**0.92±0.18**	0.82±0.26	0.89±0.19	0.69±0.26	0.79±0.08	0.68±0.15	0.22±0.06	0.43±0.06
con_10_4	0.5±0.26	0.58±0.16	0.95±0.11	0.47±0.33	**0.96±0.1**	0.47±0.32	0.61±0.05	0.69±0.16	0.35±0.12	0.42±0.09
con_20_2	0.95±0.09	0.91±0.12	**1.0±0.0**	0.97±0.07	**1.0±0.0**	0.83±0.1	0.91±0.07	0.4±0.26	0.18±0.06	0.47±0.08
con_20_3	0.91±0.15	0.75±0.24	0.97±0.05	0.96±0.09	**0.98±0.04**	**0.98±0.04**	0.79±0.09	0.84±0.1	0.16±0.02	0.37±0.04
con_20_4	0.9±0.18	0.54±0.27	**1.0±0.01**	0.95±0.15	**1.0±0.01**	0.92±0.2	0.65±0.1	0.6±0.11	0.21±0.04	0.37±0.06
con_30_2	0.98±0.08	0.82±0.19	**1.0±0.01**	0.95±0.1	**1.0±0.01**	0.89±0.12	0.82±0.07	0.45±0.16	0.17±0.04	0.4±0.07
con_30_3	0.91±0.19	0.8±0.15	0.98±0.04	0.95±0.11	**0.99±0.02**	0.96±0.1	0.88±0.04	0.85±0.13	0.15±0.03	
con_30_4	0.95±0.13	0.83±0.17	**0.98±0.07**	0.94±0.16	**0.98±0.07**	0.96±0.14	0.73±0.05	0.74±0.16	0.15±0.04	0.33±0.01
con_40_2	0.99±0.02	0.91±0.12	**1.0±0.01**	0.97±0.06	**1.0±0.01**	0.93±0.08	0.79±0.09	0.66±0.23		
con_40_3	0.96±0.05	0.74±0.22	**0.99±0.04**	0.98±0.04	**0.99±0.03**	0.98±0.04	0.92±0.04	0.9±0.05		
con_40_4	0.95±0.11	0.79±0.21	0.98±0.09	0.94±0.16	**0.99±0.05**	0.97±0.08	0.78±0.04	0.78±0.15		
dis_5_2	0.32±0.3	0.65±0.21	0.87±0.2	0.88±0.2	0.91±0.19	0.9±0.19	0.9±0.19	**0.92±0.14**	n/a	n/a
dis_5_3	0.16±0.18	0.59±0.22	**0.76±0.18**	0.75±0.19	0.73±0.16	**0.76±0.16**	0.74±0.15	0.7±0.14	n/a	n/a
dis_5_4	0.11±0.06	0.3±0.16	0.64±0.2	0.61±0.21	0.73±0.19	0.7±0.2	0.78±0.21	**0.83±0.2**	n/a	n/a
dis_10_2	0.42±0.19	0.35±0.12	0.75±0.12	0.75±0.13	**0.77±0.14**	**0.77±0.14**	0.74±0.14	0.75±0.2	n/a	n/a
dis_10_3	0.15±0.13	0.39±0.14	0.53±0.09	0.53±0.1	0.56±0.11	0.56±0.11	0.54±0.12	**0.62±0.17**	n/a	n/a
dis_10_4	0.17±0.16	0.26±0.14	0.39±0.13	0.4±0.14	0.4±0.12	**0.41±0.13**	**0.41±0.15**	0.37±0.13	n/a	n/a
dis_20_2	0.63±0.12	0.58±0.16	0.71±0.12	0.7±0.13	**0.75±0.12**	**0.75±0.12**	0.69±0.09	0.72±0.15	n/a	n/a
dis_20_3	0.26±0.14	0.34±0.09	0.45±0.06	0.45±0.06	0.46±0.05	0.46±0.06	**0.49±0.08**	0.47±0.09	n/a	n/a
dis_20_4	0.18±0.09	0.27±0.06	0.34±0.08	0.34±0.07	**0.38±0.07**	**0.38±0.07**	0.31±0.05	0.32±0.08	n/a	n/a
dis_30_2	0.49±0.16	0.46±0.11	**0.66±0.05**	**0.66±0.05**	**0.66±0.04**	**0.66±0.04**	**0.66±0.04**	0.61±0.09	n/a	n/a
dis_30_3	0.3±0.09	0.36±0.1	0.46±0.08	**0.47±0.1**	0.46±0.08	0.45±0.1	**0.47±0.08**	0.44±0.08	n/a	n/a
dis_30_4	0.25±0.05	0.25±0.05	**0.31±0.06**	**0.31±0.06**	**0.31±0.06**	**0.31±0.06**	0.3±0.03	**0.31±0.07**	n/a	n/a
dis_40_2	0.39±0.14	0.59±0.1	0.64±0.07	0.63±0.07	0.63±0.08	0.63±0.07	**0.65±0.05**	**0.65±0.1**	n/a	n/a
dis_40_3	0.24±0.08	0.44±0.09	0.44±0.09	0.44±0.09	0.42±0.09	0.43±0.09	0.45±0.07	**0.48±0.07**	n/a	n/a
dis_40_4	0.15±0.08	0.28±0.07	**0.33±0.07**	**0.33±0.07**	0.32±0.07	0.32±0.07	0.29±0.04	0.28±0.06	n/a	n/a
mix_5_2	0.24±0.23	0.55±0.26	0.54±0.13	0.54±0.13	0.53±0.19	0.51±0.24	**0.56±0.19**	0.52±0.14	n/a	n/a
mix_5_3	0.25±0.08	0.4±0.1	0.53±0.1	0.51±0.1	0.53±0.1	0.51±0.13	0.48±0.1	**0.61±0.17**	n/a	n/a
mix_5_4	0.28±0.1	0.44±0.17	0.5±0.1	0.51±0.11	0.55±0.15	0.49±0.09	0.5±0.1	**0.58±0.11**	n/a	n/a
mix_10_2	0.41±0.15	0.44±0.1	0.43±0.15	0.4±0.14	0.41±0.15	0.41±0.12	**0.48±0.12**	0.34±0.1	n/a	n/a
mix_10_3	0.33±0.15	0.45±0.11	0.46±0.1	0.46±0.11	0.46±0.12	0.47±0.12	**0.51±0.13**	0.42±0.11	n/a	n/a
mix_10_4	0.35±0.1	0.43±0.09	0.52±0.07	0.51±0.07	0.51±0.06	0.51±0.06	**0.53±0.08**	0.48±0.06	n/a	n/a
mix_20_2	0.41±0.09	**0.48±0.06**	0.3±0.14	0.3±0.14	0.3±0.14	0.29±0.14	0.4±0.05	0.22±0.11	n/a	n/a
mix_20_3	0.38±0.05	0.37±0.09	0.3±0.09	0.32±0.07	0.27±0.09	0.3±0.09	**0.39±0.06**	0.23±0.09	n/a	n/a
mix_20_4	0.43±0.08	**0.44±0.1**	0.4±0.1	0.4±0.09	0.4±0.1	0.4±0.1	0.42±0.08	0.3±0.09	n/a	n/a
mix_30_2	**0.4±0.09**	0.39±0.05	0.25±0.09	0.25±0.1	0.28±0.11	0.28±0.11	0.34±0.05	0.18±0.13	n/a	n/a
mix_30_3	0.35±0.07	0.37±0.07	0.35±0.1	0.35±0.1	0.34±0.09	0.33±0.09	**0.38±0.03**	0.21±0.08	n/a	n/a
mix_30_4	0.37±0.06	**0.42±0.04**	0.37±0.09	0.38±0.09	0.38±0.09	0.38±0.08	**0.42±0.06**	0.23±0.07	n/a	n/a
mix_40_2	**0.41±0.07**	0.4±0.07	0.27±0.12	0.28±0.12	0.27±0.12	0.27±0.12	0.31±0.06	0.11±0.05	n/a	n/a
mix_40_3	0.38±0.08	**0.42±0.03**	0.33±0.13	0.34±0.13	0.35±0.1	0.35±0.11	0.38±0.05	0.21±0.12	n/a	n/a
mix_40_4	0.34±0.06	0.37±0.05	**0.39±0.07**	**0.39±0.07**	**0.39±0.06**	**0.39±0.06**	0.39±0.05	0.3±0.12	n/a	n/a
ADNI	**0.36±0.02**	0.34±0.04	0.34±0.05	0.34±0.05	0.34±0.05	0.34±0.05	0.32±0.02	0.29±0.03	n/a	n/a

Table 15: Performance of all methods on all synthetic datasets and the ADNI dataset based on FP. The best values are marked in bold. Empty entries indicate no configuration found. LiNGAM and GOLEM are only designed for continuous datasets (denoted with n/a)

	Causal tuning methods						Individual algorithms			
Dataset	StARS	OCT	AutoCD	AutoCD+	AutoCD$_{PC}$	AutoCD$_{PC}$+	PC	FGES	LiNGAM	GOLEM
con_5_2	0.06±0.1	0.23±0.15	**0.01±0.04**	0.03±0.07	**0.01±0.04**	0.11±0.13	0.11±0.08	0.25±0.15	0.31±0.12	0.23±0.16
con_5_3	**0.02±0.04**	0.1±0.1	0.03±0.06	0.05±0.06	**0.02±0.05**	0.05±0.06	0.09±0.06	0.12±0.08	0.14±0.11	0.13±0.06
con_5_4	**0.0±0.0**	**0.0±0.0**	**0.0±0.0**	**0.0±0.0**	**0.0±0.0**	**0.0±0.0**	**0.0±0.0**	**0.0±0.0**	**0.0±0.0**	**0.0±0.0**
con_10_2	0.16±0.09	0.21±0.19	0.01±0.03	0.07±0.12	**0.0±0.0**	0.13±0.16	0.09±0.07	0.38±0.2	0.46±0.07	0.27±0.15
con_10_3	0.05±0.05	0.23±0.12	**0.01±0.02**	0.03±0.05	0.02±0.04	0.03±0.04	0.04±0.06	0.27±0.15	0.28±0.1	0.17±0.07
con_10_4	0.05±0.05	0.21±0.14	**0.01±0.03**	0.07±0.06	**0.01±0.04**	0.08±0.06	0.09±0.03	0.25±0.13	0.25±0.07	0.16±0.08
con_20_2	0.04±0.05	0.05±0.08	**0.0±0.0**	0.02±0.06	**0.0±0.0**	0.15±0.09	0.08±0.06	0.6±0.26	0.51±0.07	0.25±0.07
con_20_3	0.03±0.05	0.19±0.21	0.01±0.01	0.02±0.04	0.01±0.01	**0.0±0.01**	0.07±0.04	0.14±0.1	0.4±0.07	0.25±0.06
con_20_4	0.01±0.02	0.37±0.28	**0.0±0.0**	0.01±0.04	**0.0±0.0**	0.02±0.05	0.07±0.04	0.37±0.12	0.33±0.05	0.21±0.05
con_30_2	0.01±0.03	0.13±0.16	**0.0±0.01**	0.04±0.07	**0.0±0.01**	0.08±0.09	0.16±0.07	0.54±0.16	0.52±0.05	0.17±0.07
con_30_3	0.02±0.03	0.13±0.13	0.0±0.01	0.02±0.04	**0.0±0.0**	0.02±0.04	0.04±0.02	0.13±0.13	0.44±0.08	
con_30_4	0.02±0.04	0.13±0.14	**0.01±0.02**	0.02±0.06	**0.01±0.02**	0.01±0.04	0.06±0.02	0.23±0.14	0.34±0.05	0.16±0.13
con_40_2	0.01±0.02	0.09±0.12	**0.0±0.0**	0.02±0.06	**0.0±0.0**	0.07±0.08	0.21±0.1	0.33±0.23		
con_40_3	0.02±0.03	0.2±0.22	**0.0±0.03**	0.01±0.03	**0.0±0.01**	0.01±0.03	0.02±0.02	0.09±0.05		
con_40_4	0.03±0.06	0.18±0.19	0.01±0.04	0.04±0.08	**0.0±0.04**	0.03±0.07	0.04±0.01	0.2±0.14		
dis_5_2	0.02±0.1	0.17±0.18	0.02±0.05	0.02±0.05	**0.0±0.0**	**0.0±0.0**	**0.0±0.0**	0.01±0.03	n/a	n/a
dis_5_3	**0.0±0.0**	0.05±0.08	0.0±0.02	**0.0±0.0**	0.0±0.03	**0.0±0.0**	0.02±0.07	0.13±0.12	n/a	n/a
dis_5_4	**0.0±0.0**	**0.0±0.0**	**0.0±0.0**	**0.0±0.0**	**0.0±0.0**	**0.0±0.0**	**0.0±0.0**	**0.0±0.0**	n/a	n/a
dis_10_2	0.01±0.03	0.01±0.08	0.01±0.03	0.01±0.03	**0.0±0.02**	**0.0±0.02**	0.01±0.02	0.02±0.04	n/a	n/a
dis_10_3	**0.0±0.0**	0.07±0.13	0.01±0.03	0.01±0.03	0.01±0.03	0.01±0.03	0.01±0.02	0.0±0.01	n/a	n/a
dis_10_4	**0.0±0.0**	0.04±0.11	0.01±0.02	0.01±0.02	0.01±0.02	0.01±0.02	0.02±0.02	0.03±0.04	n/a	n/a
dis_20_2	**0.01±0.02**	0.01±0.03	**0.01±0.02**	**0.01±0.02**	**0.01±0.02**	**0.01±0.02**	**0.01±0.02**	0.07±0.1	n/a	n/a
dis_20_3	0.01±0.03	0.01±0.02	0.01±0.02	0.01±0.02	**0.0±0.01**	0.0±0.02	0.01±0.02	0.02±0.05	n/a	n/a
dis_20_4	**0.0±0.01**	0.01±0.01	0.02±0.01	0.02±0.01	0.02±0.01	0.02±0.01	0.02±0.01	**0.0±0.01**	n/a	n/a
dis_30_2	0.02±0.03	**0.01±0.04**	0.03±0.03	0.03±0.03	0.03±0.03	0.03±0.03	0.03±0.02	0.02±0.05	n/a	n/a
dis_30_3	**0.01±0.02**	0.05±0.13	**0.01±0.02**	0.01±0.03	0.02±0.04	0.03±0.04	**0.01±0.02**	0.05±0.07	n/a	n/a
dis_30_4	**0.01±0.01**	0.02±0.04	0.04±0.06	0.04±0.06	0.02±0.04	0.02±0.04	0.02±0.01	0.03±0.05	n/a	n/a
dis_40_2	**0.0±0.01**	0.04±0.04	0.05±0.05	0.05±0.06	0.06±0.06	0.06±0.06	0.04±0.03	0.01±0.04	n/a	n/a
dis_40_3	**0.0±0.01**	0.05±0.08	0.02±0.02	0.02±0.03	0.01±0.02	0.01±0.02	0.03±0.02	0.03±0.04	n/a	n/a
dis_40_4	**0.0±0.01**	0.03±0.02	0.03±0.02	0.03±0.02	0.02±0.01	0.02±0.01	0.04±0.02	0.01±0.03	n/a	n/a
mix_5_2	**0.0±0.0**	0.05±0.08	0.16±0.19	0.15±0.19	0.22±0.17	0.16±0.14	0.06±0.11	0.21±0.2	n/a	n/a
mix_5_3	**0.0±0.0**	0.01±0.03	0.03±0.05	0.02±0.05	0.04±0.06	0.04±0.06	0.02±0.05	0.05±0.07	n/a	n/a
mix_5_4	**0.0±0.0**	**0.0±0.0**	**0.0±0.0**	**0.0±0.0**	**0.0±0.0**	**0.0±0.0**	**0.0±0.0**	**0.0±0.0**	n/a	n/a
mix_10_2	**0.02±0.05**	**0.02±0.05**	0.34±0.29	0.37±0.27	0.35±0.24	0.32±0.15	0.1±0.07	0.48±0.16	n/a	n/a
mix_10_3	**0.01±0.02**	0.04±0.1	0.22±0.23	0.19±0.23	0.21±0.2	0.2±0.2	0.06±0.04	0.42±0.17	n/a	n/a
mix_10_4	**0.0±0.01**	**0.0±0.01**	0.09±0.11	0.09±0.11	0.11±0.11	0.1±0.11	0.05±0.04	0.3±0.16	n/a	n/a
mix_20_2	**0.03±0.05**	0.1±0.11	0.53±0.28	0.53±0.27	0.53±0.26	0.54±0.26	0.34±0.07	0.71±0.19	n/a	n/a
mix_20_3	**0.01±0.03**	0.02±0.03	0.45±0.29	0.41±0.25	0.54±0.23	0.43±0.28	0.22±0.09	0.67±0.19	n/a	n/a
mix_20_4	**0.03±0.02**	0.1±0.16	0.29±0.17	0.28±0.16	0.29±0.17	0.28±0.16	0.13±0.06	0.57±0.18	n/a	n/a
mix_30_2	0.06±0.06	**0.05±0.05**	0.53±0.23	0.51±0.26	0.47±0.24	0.47±0.25	0.38±0.16	0.7±0.3	n/a	n/a
mix_30_3	**0.02±0.03**	0.22±0.18	0.39±0.23	0.38±0.23	0.4±0.22	0.43±0.22	0.3±0.08	0.69±0.19	n/a	n/a
mix_30_4	**0.01±0.01**	0.07±0.05	0.31±0.2	0.29±0.2	0.29±0.2	0.28±0.19	0.18±0.08	0.66±0.14	n/a	n/a
mix_40_2	**0.03±0.06**	0.09±0.15	0.48±0.26	0.45±0.25	0.47±0.25	0.47±0.26	0.35±0.14	0.85±0.09	n/a	n/a
mix_40_3	0.03±0.03	**0.03±0.02**	0.38±0.27	0.37±0.27	0.34±0.21	0.34±0.21	0.27±0.08	0.67±0.25	n/a	n/a
mix_40_4	**0.01±0.02**	0.12±0.16	0.22±0.13	0.22±0.12	0.2±0.1	0.2±0.1	0.15±0.05	0.45±0.26	n/a	n/a
ADNI	**0.49±0.04**	0.55±0.05	0.6±0.06	0.6±0.06	0.61±0.06	0.61±0.06	0.56±0.04	0.67±0.03	n/a	n/a

Table 16: Performance of all methods on all synthetic datasets and the ADNI dataset based on FN. The best values are marked in bold. Empty entries indicate no configuration found. LiNGAM and GOLEM are only designed for continuous datasets (denoted with n/a).

	Causal tuning methods						Individual algorithms			
Dataset	StARS	OCT	AutoCD	AutoCD+	AutoCD$_{PC}$	AutoCD$_{PC}$+	PC	FGES	LiNGAM	GOLEM
con_5_2	0.14±0.16	0.31±0.12	0.06±0.12	0.11±0.14	0.06±0.11	0.29±0.19	**0.02±0.05**	0.04±0.06	0.23±0.13	0.21±0.13
con_5_3	0.24±0.2	0.25±0.16	0.21±0.19	0.37±0.2	0.26±0.26	0.4±0.14	0.13±0.09	**0.09±0.09**	0.35±0.2	0.34±0.1
con_5_4	0.46±0.14	0.4±0.17	0.53±0.17	0.57±0.12	0.52±0.21	0.58±0.17	0.19±0.07	**0.15±0.07**	0.34±0.17	0.43±0.15
con_10_2	0.1±0.16	0.16±0.14	0.03±0.1	0.11±0.17	0.02±0.04	0.08±0.1	0.02±0.04	**0.0±0.0**	0.25±0.09	0.21±0.11
con_10_3	0.23±0.26	0.32±0.22	0.07±0.16	0.15±0.23	0.09±0.17	0.28±0.26	0.16±0.05	**0.06±0.05**	0.51±0.12	0.4±0.1
con_10_4	0.46±0.23	0.21±0.19	**0.03±0.09**	0.46±0.3	**0.03±0.07**	0.45±0.29	0.3±0.04	0.06±0.05	0.4±0.12	0.42±0.09
con_20_2	0.01±0.05	0.04±0.06	**0.0±0.0**	0.0±0.01	**0.0±0.0**	0.02±0.04	0.02±0.02	0.0±0.01	0.31±0.1	0.27±0.1
con_20_3	0.07±0.12	0.06±0.12	**0.02±0.04**	0.03±0.06	**0.02±0.03**	**0.02±0.04**	0.15±0.06	**0.02±0.02**	0.45±0.07	0.39±0.09
con_20_4	0.09±0.17	0.09±0.16	**0.0±0.01**	0.04±0.11	**0.0±0.01**	0.06±0.16	0.28±0.07	0.02±0.01	0.46±0.07	0.42±0.03
con_30_2	0.01±0.07	0.05±0.09	**0.0±0.0**	0.01±0.04	**0.0±0.0**	0.02±0.04	0.02±0.02	0.01±0.01	0.31±0.07	0.43±0.08
con_30_3	0.07±0.17	0.07±0.06	0.02±0.03	0.03±0.09	**0.01±0.02**	0.03±0.08	0.09±0.03	0.02±0.03	0.41±0.08	
con_30_4	0.03±0.11	0.04±0.09	**0.01±0.05**	0.04±0.11	**0.01±0.06**	0.03±0.12	0.21±0.04	0.03±0.02	0.51±0.07	0.51±0.13
con_40_2	0.0±0.01	0.0±0.01	**0.0±0.0**	0.0±0.01	**0.0±0.0**	**0.0±0.0**	0.0±0.01	0.01±0.01		
con_40_3	0.02±0.04	0.06±0.08	0.0±0.02	0.01±0.01	0.0±0.02	**0.0±0.01**	0.06±0.03	0.01±0.01		
con_40_4	0.03±0.07	0.03±0.06	0.01±0.06	0.03±0.12	0.01±0.03	**0.01±0.01**	0.18±0.03	0.02±0.02		
dis_5_2	0.65±0.34	0.18±0.25	0.11±0.2	0.11±0.2	0.09±0.19	0.1±0.19	0.1±0.19	**0.07±0.14**	n/a	n/a
dis_5_3	0.84±0.18	0.36±0.27	0.24±0.18	0.25±0.19	0.27±0.16	0.24±0.16	0.24±0.17	**0.18±0.18**	n/a	n/a
dis_5_4	0.89±0.06	0.7±0.16	0.36±0.2	0.39±0.21	0.27±0.19	0.3±0.2	0.22±0.21	**0.17±0.2**	n/a	n/a
dis_10_2	0.57±0.19	0.64±0.14	0.24±0.12	0.24±0.13	0.23±0.13	**0.22±0.14**	0.26±0.14	0.23±0.2	n/a	n/a
dis_10_3	0.85±0.13	0.54±0.22	0.45±0.09	0.46±0.1	0.43±0.11	0.43±0.11	0.46±0.11	**0.38±0.17**	n/a	n/a
dis_10_4	0.83±0.16	0.7±0.22	0.59±0.14	0.59±0.16	0.59±0.14	**0.58±0.15**	**0.58±0.17**	0.59±0.15	n/a	n/a
dis_20_2	0.36±0.13	0.41±0.16	0.28±0.11	0.29±0.12	0.25±0.11	0.24±0.11	0.3±0.07	**0.21±0.15**	n/a	n/a
dis_20_3	0.73±0.15	0.65±0.1	0.55±0.06	0.55±0.06	0.54±0.05	0.54±0.05	**0.5±0.09**	0.51±0.11	n/a	n/a
dis_20_4	0.82±0.09	0.73±0.06	0.65±0.08	0.64±0.08	**0.6±0.07**	**0.6±0.07**	0.67±0.05	0.68±0.08	n/a	n/a
dis_30_2	0.49±0.17	0.53±0.12	**0.31±0.05**	**0.31±0.05**	**0.31±0.05**	**0.31±0.05**	**0.31±0.04**	0.37±0.1	n/a	n/a
dis_30_3	0.69±0.1	0.59±0.16	0.53±0.1	0.52±0.12	0.52±0.1	0.52±0.11	0.52±0.07	**0.51±0.12**	n/a	n/a
dis_30_4	0.73±0.06	0.73±0.07	**0.65±0.07**	**0.65±0.07**	0.66±0.05	0.67±0.05	0.68±0.04	0.66±0.09	n/a	n/a
dis_40_2	0.61±0.14	0.37±0.12	0.31±0.06	0.31±0.06	0.31±0.06	**0.31±0.05**	0.31±0.04	0.35±0.1	n/a	n/a
dis_40_3	0.75±0.08	0.51±0.15	0.54±0.1	0.54±0.1	0.56±0.1	0.56±0.1	0.52±0.07	**0.49±0.09**	n/a	n/a
dis_40_4	0.85±0.08	0.68±0.08	**0.65±0.06**	**0.65±0.06**	0.66±0.06	0.66±0.06	0.67±0.04	0.71±0.06	n/a	n/a
mix_5_2	0.76±0.23	0.4±0.28	0.3±0.16	0.32±0.17	**0.25±0.19**	0.33±0.21	0.38±0.18	0.27±0.15	n/a	n/a
mix_5_3	0.75±0.08	0.59±0.12	0.45±0.11	0.47±0.12	0.44±0.12	0.45±0.17	0.49±0.11	**0.33±0.21**	n/a	n/a
mix_5_4	0.72±0.1	0.56±0.17	0.5±0.1	0.49±0.11	0.45±0.15	0.51±0.09	0.5±0.1	**0.42±0.11**	n/a	n/a
mix_10_2	0.57±0.17	0.54±0.11	0.23±0.18	0.23±0.19	0.24±0.16	0.28±0.17	0.42±0.11	**0.18±0.16**	n/a	n/a
mix_10_3	0.66±0.14	0.51±0.16	0.32±0.22	0.35±0.22	0.32±0.17	0.33±0.16	0.43±0.12	**0.15±0.12**	n/a	n/a
mix_10_4	0.65±0.1	0.57±0.09	0.39±0.13	0.39±0.13	0.38±0.12	0.38±0.12	0.42±0.07	**0.22±0.15**	n/a	n/a
mix_20_2	0.56±0.11	0.43±0.11	0.17±0.15	0.17±0.14	0.17±0.13	0.16±0.13	0.26±0.09	**0.08±0.08**	n/a	n/a
mix_20_3	0.61±0.06	0.61±0.09	0.25±0.21	0.28±0.19	0.19±0.14	0.27±0.21	0.39±0.09	**0.1±0.12**	n/a	n/a
mix_20_4	0.54±0.07	0.46±0.15	0.31±0.1	0.32±0.09	0.31±0.1	0.31±0.09	0.45±0.08	**0.14±0.1**	n/a	n/a
mix_30_2	0.54±0.11	0.56±0.07	0.22±0.16	0.24±0.17	0.25±0.14	0.24±0.14	0.28±0.13	**0.11±0.17**	n/a	n/a
mix_30_3	0.63±0.08	0.42±0.12	0.27±0.13	0.27±0.13	0.26±0.13	0.24±0.13	0.32±0.06	**0.1±0.12**	n/a	n/a
mix_30_4	0.62±0.07	0.51±0.06	0.32±0.13	0.33±0.13	0.33±0.13	0.34±0.13	0.4±0.06	**0.1±0.08**	n/a	n/a
mix_40_2	0.56±0.07	0.51±0.13	0.26±0.15	0.27±0.14	0.26±0.14	0.25±0.14	0.33±0.1	**0.05±0.05**	n/a	n/a
mix_40_3	0.6±0.09	0.55±0.03	0.29±0.15	0.29±0.15	0.31±0.11	0.31±0.11	0.35±0.06	**0.12±0.13**	n/a	n/a
mix_40_4	0.64±0.07	0.51±0.15	0.39±0.1	0.39±0.09	0.41±0.07	0.41±0.07	0.46±0.06	**0.25±0.15**	n/a	n/a
ADNI	0.15±0.04	0.11±0.02	0.05±0.02	0.05±0.02	0.05±0.01	0.05±0.01	0.12±0.06	**0.04±0.0**	n/a	n/a

This table shows the loss values for all methods. We note that StARS on all datasets and AutoCD variants on continuous datasets use a different loss function (marked in grey). Therefore, the best values marked and in bold compare the loss values for only the discrete and mixed datasets for AutoCD variants, PC, and FGES.

Table 17: Performance of all methods on all synthetic datasets and the ADNI dataset based on Loss. The best values are marked in bold. Empty entries indicate no configuration found and entries in grey are not compared. Empty entries indicate no configuration found. LiNGAM and GOLEM are only designed for continuous datasets (denoted with n/a).

	Causal tuning methods						Individual algorithms			
Dataset	StARS	OCT	AutoCD	AutoCD+	AutoCD$_{PC}$	AutoCD$_{PC}$+	PC	FGES	LiNGAM	GOLEM
con_5_2	0.02±0.02	0.54±0.13	0.0±0.0	0.02±0.02	0.0±0.0	0.03±0.01	0.4±0.12	0.39±0.11	0.51±0.11	0.43±0.1
con_5_3	0.02±0.02	0.46±0.11	0.0±0.0	0.02±0.02	0.0±0.0	0.02±0.02	0.37±0.1	0.35±0.09	0.45±0.08	0.39±0.09
con_5_4	0.02±0.02	0.3±0.11	0.0±0.0	0.01±0.01	0.0±0.0	0.01±0.02	0.13±0.12	0.12±0.12	0.21±0.11	0.19±0.12
con_10_2	0.02±0.01	0.56±0.07	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.44±0.06	0.44±0.06	0.57±0.07	0.48±0.09
con_10_3	0.03±0.01	0.46±0.19	0.0±0.0	0.01±0.01	0.0±0.0	0.02±0.01	0.3±0.11	0.26±0.11	0.49±0.07	0.37±0.11
con_10_4	0.02±0.01	0.25±0.07	0.0±0.0	0.02±0.02	0.0±0.0	0.02±0.02	0.2±0.12	0.12±0.11	0.35±0.13	0.24±0.09
con_20_2	0.01±0.0	0.56±0.06	0.0±0.0	0.0±0.0	0.0±0.0	0.01±0.0	0.5±0.05	0.48±0.05	0.76±0.03	0.56±0.07
con_20_3	0.01±0.01	0.39±0.1	0.0±0.0	0.01±0.01	0.0±0.0	0.0±0.0	0.32±0.08	0.29±0.07	0.65±0.04	0.4±0.08
con_20_4	0.01±0.01	0.23±0.08	0.0±0.0	0.0±0.01	0.0±0.0	0.01±0.01	0.28±0.05	0.16±0.06	.54±0.05	0.29±0.06
con_30_2	0.0±0.0	0.56±0.06	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.53±0.04	0.53±0.04	0.8±0.02	0.73±0.04
con_30_3	0.0±0.0	0.4±0.03	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.37±0.04	0.36±0.03	0.75±0.02	
con_30_4	0.01±0.01	0.29±0.04	0.0±0.0	0.0±0.01	0.0±0.0	0.0±0.01	0.31±0.06	0.25±0.05	0.68±0.03	0.56±0.14
con_40_2	0.0±0.0	0.54±0.03	0.0±0.0	0.0±0.0	0.0±0.0	0.01±0.01	0.54±0.04	0.54±0.03		
con_40_3	0.0±0.0	0.4±0.04	0.0±0.01	0.0±0.0	0.0±0.0	0.0±0.0	0.42±0.03	0.38±0.03		
con_40_4	0.01±0.01	0.28±0.04	0.0±0.01	0.01±0.01	0.0±0.01	0.0±0.01	0.32±0.05	0.27±0.04		
dis_5_2	0.01±0.01	0.87±0.04	**0.82±0.03**	**0.82±0.03**	**0.82±0.03**	**0.82±0.03**	0.83±0.03	**0.82±0.03**	n/a	n/a
dis_5_3	0.01±0.01	0.85±0.06	0.81±0.05	0.81±0.05	0.81±0.04	0.81±0.05	**0.8±0.05**	0.81±0.05	n/a	n/a
dis_5_4	0.0±0.01	0.87±0.05	0.8±0.04	0.8±0.04	0.78±0.05	0.78±0.05	0.78±0.05	**0.78±0.04**	n/a	n/a
dis_10_2	0.01±0.01	0.88±0.04	0.85±0.03	**0.84±0.03**	**0.84±0.03**	**0.84±0.03**	0.85±0.03	0.85±0.03	n/a	n/a
dis_10_3	0.01±0.01	0.89±0.05	**0.84±0.05**	**0.84±0.04**	0.84±0.05	0.84±0.05	0.85±0.05	0.84±0.05	n/a	n/a
dis_10_4	0.01±0.01	0.9±0.04	**0.86±0.04**	**0.86±0.04**	**0.86±0.04**	0.86±0.05	0.87±0.05	0.86±0.05	n/a	n/a
dis_20_2	0.02±0.01	0.9±0.02	**0.87±0.02**	0.88±0.02	**0.87±0.02**	**0.87±0.02**	0.88±0.01	0.88±0.01	n/a	n/a
dis_20_3	0.01±0.01	0.9±0.01	**0.88±0.02**	**0.88±0.02**	**0.88±0.02**	**0.88±0.02**	0.89±0.02	**0.88±0.02**	n/a	n/a
dis_20_4	0.01±0.01	0.91±0.02	**0.89±0.02**	**0.89±0.02**	**0.89±0.02**	**0.89±0.02**	0.9±0.02	**0.89±0.02**	n/a	n/a
dis_30_2	0.01±0.01	0.91±0.01	**0.89±0.01**	**0.89±0.01**	**0.89±0.01**	**0.89±0.01**	**0.89±0.01**	**0.89±0.01**	n/a	n/a
dis_30_3	0.01±0.01	0.91±0.02	0.89±0.02	**0.89±0.01**	**0.89±0.01**	0.89±0.02	**0.89±0.01**	**0.89±0.01**	n/a	n/a
dis_30_4	0.01±0.01	0.91±0.01	**0.9±0.01**	**0.9±0.01**	**0.9±0.01**	**0.9±0.01**	0.91±0.01	**0.9±0.01**	n/a	n/a
dis_40_2	0.0±0.0	0.89±0.01	**0.88±0.01**	**0.88±0.01**	**0.88±0.01**	**0.88±0.01**	**0.88±0.01**	**0.88±0.01**	n/a	n/a
dis_40_3	0.0±0.0	0.9±0.02	0.9±0.01	0.9±0.01	0.9±0.01	0.9±0.01	0.9±0.02	**0.89±0.01**	n/a	n/a
dis_40_4	0.0±0.0	0.92±0.01	**0.91±0.01**	**0.91±0.01**	**0.91±0.01**	**0.91±0.01**	**0.91±0.01**	**0.91±0.01**	n/a	n/a
mix_5_2	0.02±0.01	0.91±0.03	0.89±0.03	0.89±0.03	**0.88±0.03**	0.89±0.02	0.89±0.03	**0.88±0.03**	n/a	n/a
mix_5_3	0.02±0.01	0.84±0.05	0.8±0.03	0.81±0.03	**0.79±0.03**	0.81±0.04	0.8±0.03	0.79±0.04	n/a	n/a
mix_5_4	0.02±0.02	0.81±0.05	**0.75±0.05**	0.77±0.05	**0.75±0.05**	0.76±0.06	**0.75±0.05**	**0.75±0.05**	n/a	n/a
mix_10_2	0.03±0.0	0.88±0.02	0.85±0.02	0.85±0.01	0.84±0.02	0.84±0.02	0.85±0.02	**0.83±0.02**	n/a	n/a
mix_10_3	0.03±0.01	0.82±0.04	**0.79±0.04**	**0.79±0.04**	**0.79±0.04**	**0.79±0.04**	**0.79±0.04**	**0.79±0.04**	n/a	n/a
mix_10_4	0.02±0.01	0.8±0.04	0.77±0.03	0.77±0.03	0.77±0.02	0.77±0.02	0.77±0.02	**0.76±0.03**	n/a	n/a
mix_20_2	0.01±0.01	0.86±0.02	0.84±0.02	0.84±0.02	0.84±0.02	0.84±0.02	0.84±0.02	**0.83±0.02**	n/a	n/a
mix_20_3	0.01±0.01	0.83±0.05	0.79±0.06	0.79±0.05	0.79±0.06	0.79±0.05	0.8±0.04	**0.78±0.05**	n/a	n/a
mix_20_4	0.02±0.0	0.78±0.03	0.77±0.04	0.76±0.04	0.76±0.04	0.76±0.04	0.77±0.03	**0.75±0.03**	n/a	n/a
mix_30_2	0.01±0.01	0.89±0.02	0.87±0.01	0.87±0.01	0.86±0.01	0.86±0.01	0.86±0.01	**0.85±0.01**	n/a	n/a
mix_30_3	0.01±0.01	0.84±0.02	0.83±0.01	0.83±0.01	0.83±0.01	0.83±0.01	0.83±0.01	**0.82±0.01**	n/a	n/a
mix_30_4	0.01±0.01	0.79±0.03	0.77±0.02	0.77±0.02	0.77±0.02	0.77±0.02	0.78±0.02	**0.76±0.02**	n/a	n/a
mix_40_2	0.01±0.01	0.89±0.02	0.87±0.01	0.87±0.01	0.87±0.01	0.87±0.01	0.87±0.02	**0.85±0.02**	n/a	n/a
mix_40_3	0.01±0.0	0.83±0.02	0.81±0.01	0.8±0.01	0.81±0.02	0.8±0.01	0.8±0.02	**0.79±0.01**	n/a	n/a
mix_40_4	0.01±0.01	0.8±0.02	0.78±0.02	0.78±0.02	0.78±0.02	0.78±0.02	0.78±0.02	**0.77±0.02**	n/a	n/a
ADNI	0.1±0.02	0.91±0.0	**0.9±0.0**	**0.9±0.0**	**0.9±0.0**	**0.9±0.0**	**0.9±0.0**	**0.9±0.0**	n/a	n/a