Efficient SAGE Estimation via Causal Structure Learning

Christoph Luther*
University of Vienna
UniVie Doctoral School CS

Gunnar König*
LMU Munich
University of Vienna
Munich Center for ML (MCML)

Moritz Grosse-Wentrup
University of Vienna
Data Science @ Uni Vienna
Vienna CogSciHub

Abstract

The Shapley Additive Global Importance (SAGE) value is a theoretically appealing interpretability method that fairly attributes global importance to a model's features. However, its exact calculation requires the computation of the feature's surplus performance contributions over an exponential number of feature sets. This is computationally expensive, particularly because estimating the surplus contributions requires sampling from conditional distributions. Thus, SAGE approximation algorithms only take a fraction of the feature sets into account. We propose d-SAGE, a method that accelerates SAGE approximation. d-SAGE is motivated by the observation that conditional independencies (CIs) between a feature and the model target imply zero surplus contributions, such that their computation can be skipped. To identify CIs, we leverage causal structure learning (CSL) to infer a graph that encodes (conditional) independencies in the data as d-separations. This is computationally more efficient because the expense of the one-time graph inference and the d-separation queries is negligible compared to the expense of surplus contribution evaluations. Empirically we demonstrate that d-SAGE enables the efficient and accurate estimation of SAGE values.

1 INTRODUCTION

Machine learning (ML) is increasingly deployed in various fields, ranging from the sciences (Reichstein et al., 2019; Schmidt et al., 2019; Luan and Tsai, 2021; Farrell et al., 2018) to high-stakes decisions about individuals (Raghavan et al., 2020; Zeng et al., 2017; Obermeyer and Mullainathan, 2019). Despite impressive successes in predictive performance (Senior et al., 2020; Bhatt et al., 2020), the complexity of ML models makes it difficult to assess their trustworthiness or to gain knowledge about the data generating process. In recent years, the advent of interpretable machine learning has brought about a plethora of methods that provide insight into model and data (Molnar, 2020). Among those, interpretability methods based on the Shapley value from game theory (Shapley, 1953) have gained popularity as they satisfy desirable fairness properties (Štrumbelj and Kononenko, 2014; Datta et al., 2016; Lundberg and Lee, 2017; Sundararajan and Najmi, 2020; Covert et al., 2020).

SAGE values (Covert et al., 2020) apply Shapley values to fairly attribute the model's predictive performance to the features, thereby providing valuable insight into dependencies in the data. They are particularly appealing for scientific inference since they can be linked to properties of the data generating process (Covert et al., 2020; Freiesleben et al., 2022). The building blocks for SAGE values are so-called SAGE value functions $\nu(\mathbf{X}_S)$ that measure the performance contribution of arbitrary subsets of features $\mathbf{X}_S$. Based on these value functions, a feature's importance value ϕ is computed as the average surplus contribution $\nu(\mathbf{X}_{S\cup j}) - \nu(\mathbf{X}_S)$ of the feature X_j over all possible subsets $\mathbf{X}_S$ of the remaining features. This is a computationally demanding procedure due to the number of coalitions $\mathbf{X}_S$ that grows exponentially with the number of features (Covert et al., 2020; Van den Broeck et al., 2022) and the high expense of evaluating ν which stems from the conditional sampling that is required for its estimation. In practice, (Covert et al., 2020) address the exponential number of coalitions by only computing the respective surplus contribution for a randomly sampled subset of the coalitions.[1]

---

[1]Furthermore, Covert et al. (2020) avoid conditional sampling for the evaluation of ν by employing marginal sampling instead. If features are dependent, this leads to extrapolation and does not allow linking the SAGE values to properties of the data generating process (Chen et al., 2020). In this work, we focus on estimating conditional SAGE values.

In this work, we suggest exploiting the dependence structure in the data to speed up the estimation of (conditional sampling based) SAGE values in an approach we coin d-SAGE. More specifically, we show that the surplus contribution $\nu(\mathbf{X}_{S \cup j}) - \nu(\mathbf{X}_S)$ is zero for optimal predictors if the variable of interest is conditionally independent of the model's target given the respective subset of remaining features (i.e., if $X_j \perp Y | \mathbf{X}_S$, Theorem 1). As such, if we know the conditional independencies (CIs) in the data, the respective value function evaluations can be skipped. Since, in general, the dependence structure is unknown, and conditional independence testing is expensive, we leverage research in causal structure learning (CSL) that allows us to greedily learn graphical models which encode the dependence structure in the data.

Overall, the approach is based on the following rationale: The quality of SAGE approximation hinges on the number of evaluations of ν that each require estimating conditional expectations and thus are computationally expensive.[2] d-SAGE relies on the one-time estimation of a causal graph, which in practice can be performed by greedy-search algorithms in polynomial time (Scutari et al., 2019b). The estimated graph then allows to identify CIs using linear-time d-separation queries (Hagberg et al., 2008; Darwiche, 2009). Every found d-separation, in turn, warrants to spare an expensive evaluation of $\nu(\mathbf{X}_{S \cup j}) - \nu(\mathbf{X}_S)$. Since graph learning has to be performed only once and d-separation queries are highly efficient, the runtime of SAGE estimation can be reduced significantly by skipping the computation of $\nu(\mathbf{X}_{S \cup j}) - \nu(\mathbf{X}_S)$ whenever warranted. We show empirically that the saved runtime is approximately equal to the share of CIs.

1.1 Contributions

We propose d-SAGE, the first method that exploits the dependence structure in the data to make SAGE estimation more efficient. More specifically, we find that CIs in the data imply that the respective (expensive) surplus evaluations can be skipped and suggest leveraging greedy CSL for their identification (Section 4). To select a suitable CSL algorithm, we perform a benchmark that, in contrast to previous work, evaluates the algorithms' ability to efficiently identify CIs in the data (Section 5.1). On twelve synthetic datasets, we demonstrate empirically that d-SAGE and the approximation algorithm by Covert et al. (2020) converge towards the same estimates but that d-SAGE is significantly faster. We find that the computational overhead of learning the causal structure is negligible compared to the computational cost of the surplus evaluations, such that the overall runtime reduction is approximately equal to the share

of CIs found in the data (Section 5.2). Consequently, d-SAGE enables the application of SAGE for larger models, especially in sparse settings.

2 RELATED WORK

While there are many attempts to tackle the complexity of Shapley value based methods, most existing work targets speeding up SHAP (Lundberg and Lee, 2017) estimation (Jethani et al., 2021; Covert and Lee, 2021; Li et al., 2020) or is limited to be applied with random forests (Bénard et al., 2022). In contrast, our work is model-agnostic and targets improving SAGE estimation. Moreover, none of the existing work exploits the dependence structure in the data to yield efficiency gains. As such, we see our work as complementary to the approach of Mitchell et al. (2022), who suggest to carefully select permutations.

In recent years, concepts from causality have also been introduced to Shapley value based importance measures to adapt them to answer specific questions or to improve model interpretation. Frye et al. (2020b), for example, introduce *asymmetric Shapley values* that can either shift the explanatory power of all variables along a causal chain towards the root cause (distal approach) or towards immediate causes (proximate approach). Moreover, Heskes et al. (2020) use Pearl's do-calculus to develop *causal Shapley values* and Wang et al. (2021) propose to attach importance to edges in a causal graph instead of explanatory variables, i.e., nodes in the graph. In contrast to the literature, we seek efficiency gains for feature attributions from causal inference research while retaining the principle of SAGE values unaltered.

We do, however, make use of CSL. Scutari et al. (2019a) and Constantinou et al. (2021) provide large-scale benchmark studies of structure learning algorithms. In short, both studies agree on the superiority of score-based structure learning based on greedy search algorithms over constraint-based and hybrid methods. These findings motivate our choice of CSL algorithms for d-separation inference. In contrast to existing work, our benchmark does not focus on recovering the causal structure but on detecting CIs in the data.

3 BACKGROUND

This section serves to familiarise the reader with the basic concepts required to understand this paper. First, we introduce SAGE values for global feature importance. We then explain CSL, which we later use to speed up SAGE estimation.

---

[2]The expense of the computation depends on the type of data for which the conditional expectation shall be computed. Previous work in the field assumes polynomial complexity for the operation (Van den Broeck et al., 2022).

3.1 Shapley Additive Global Importance

The Shapley value, which was initially proposed in game theory (Shapley, 1953), is commonly applied for feature relevance quantification (Štrumbelj and Kononenko, 2014; Datta et al., 2016; Lundberg and Lee, 2017; Sundararajan and Najmi, 2020; Covert et al., 2020). In the study of co-operative games, it serves to fairly attribute the outcome of a game to all participating players. The principle can be applied to assess the relevance of variables for a predictor f, where the predictive performance is the outcome of the game and the variables are the players. Covert et al. (2020) leverage Shapley values to derive a *global* measure of feature importance, i.e. SAGE values. Global in this context means that the importance of a feature across all instances in a sample is assessed. For an arbitrary model $\hat{f}$ using inputs $x_1, ..., x_d$, Covert et al. (2020) define the SAGE value for the j-th feature as:

$$
\phi_j(\nu) = \frac{1}{d!} \sum_{\pi \in \Pi(d)} \big(\nu(\{X_i : \pi(i) \leq \pi(j)\}) \\
- \nu(\{X_i : \pi(i) < \pi(j)\}) \big) \quad (1)
$$

where X_j is the random variable corresponding to feature observation x_j, $\Pi(d)$ is the set of all permutations of indices $\{1, ..., d\}$, π a specific permutation and $\pi(j)$ the position of feature j in permutation π. For the sake of readability, we use the more general notation $\mathbf{X}_S$ instead of $\{X_i : \pi(i) < \pi(j)\}$ as input to the value function ν with $\mathbf{X}_S$ being any set of features and S the collection of indices of the contained features, i.e. $S \subseteq \{1, ..., d\}$ ($\bar{S}$ is its complementary set). $\nu(\mathbf{X}_S)$ is defined as

$$
\nu(\mathbf{X}_S) = \mathbb{E}_{\mathbf{X},Y}[\ell(\hat{f}_\emptyset(\mathbf{X}_\emptyset), Y)] - \mathbb{E}_{\mathbf{X},Y}[\ell(\hat{f}_S(\mathbf{X}_S), Y)],
$$

where $\ell(\cdot)$ is any admissible loss function and $\hat{f}_S(\mathbf{x}_S) = \mathbb{E}_{\mathbf{X}_{\bar{S}}|\mathbf{X}_S}[\hat{f}(\mathbf{X})|\mathbf{X}_S = \mathbf{x}_S]$. Thus, $\nu(\mathbf{X}_S)$ is the reduction in risk induced by adding $\mathbf{X}_S$. Consequently, SAGE values gauge a feature j's importance using the average over the additional reduction in risk of the feature compared to any existing coalition.

SAGE values are particularly appealing as they satisfy six desirable fairness axioms that set them apart from other feature importance measures: *efficiency*, the *dummy property*, *symmetry*, *monotonicity*, *linearity*[3] and invariance to monotone transformations. Despite a thorough mathematical foundation and the fulfilment of mentioned desiderata, SAGE values have a major drawback: They require the evaluation of an exponential number of surplus evaluations, which is computationally infeasible. In practice, only a subset of possible coalitions is evaluated (cf. Section 3.2).

[3] For simplicity we employ the names of these Shapley value properties for the SAGE properties that are described in Appendix D.

To estimate SAGE values, access to the conditional feature distributions is required; More specifically, we need to sample from $P(\mathbf{X}_{\bar{S}}|\mathbf{X}_S)$ to estimate the marginalized prediction $\hat{f}_S(\mathbf{x}_S) = \mathbb{E}_{\mathbf{X}_{\bar{S}}|\mathbf{X}_S}[\hat{f}(\mathbf{X})|\mathbf{X}_S = \mathbf{x}_S]$. However, conditional samplers may not be readily available in practice. Covert et al. (2020) suggest eluding the problem by sampling from $P(\mathbf{X}_{\bar{S}})$ instead (marginal sampling). Albeit easy to implement (and computationally efficient), marginal sampling may generate unrealistic data points $(\mathbf{x}_S, \mathbf{x}_{\bar{S}})$ and thus marginal-sampling based SAGE values are not suitable for inference about the data generating process or to understand the model's behaviour in the observational distribution (Frye et al., 2020a; Chen et al., 2020; Aas et al., 2021; Molnar et al., 2022). Therefore, we focus on conditional SAGE and estimate the conditional distributions if they are not known.

For conditional distribution estimation, a variety of techniques can be employed. For categorical variables, estimating the conditional reduces to standard supervised learning with cross-entropy loss. For linear Gaussian data, it can be estimated analytically from the covariance matrix (Page Jr, 1984). A range of methods exist for continuous settings with nonlinearities (Bishop, 1994; Bashtannyk and Hyndman, 2001; Sohn et al., 2015; Trippe and Turner, 2018; Winkler et al., 2019; Hothorn and Zeileis, 2021). For mixed data, a sequential design can be employed (Blesch et al., 2022).

3.2 Intractability of SAGE and Approximation Algorithm

For the Shapley based interpretability approach SHAP intractability was proven (Van den Broeck et al., 2022). For the exact computation, the surplus contribution for all possible subsets of the remaining features must be evaluated. The number of possible subsets grows exponentially in the number of features.

Exact SAGE estimation also suffers from the exponential number of coalitions. To address the issue, Covert et al. (2020) propose an approximation algorithm that does not take all possible coalitions into account. More specifically, the authors propose to repetitively sample permutations π from the feature indices. Then, for every element of the current permutation, starting with the first one, they successively compute $\Delta_{j|S} := \nu(\mathbf{X}_{S \cup j}) - \nu(\mathbf{X}_S)$ with the set $\mathbf{X}_S$ being all features that come before the feature of interest j in π. The mean of all $\Delta_{j|S}$ values for X_j over the number of repetitions then is its estimated importance $\hat{\phi}_j(\nu)$. The approximation algorithm is unbiased and the variance of the estimate reduces in $O(\frac{1}{n})$ (Covert et al., 2020). However, considering the risk evaluation required for estimating ν, the procedure based on conditional sampling remains computationally demanding.

3.3 Causal Structure Learning

This section deals with the introduction of CSL used to estimate graphs representing d-separations. d-separation is the graphical equivalent to conditional independence in the underlying distribution. Both concepts are indeed equivalent under two standard assumptions: (1) that the Markov property is fulfilled and (2) that the distribution is faithful w.r.t. the graph. Since we merely use graphs to read off d-separations, we leave out a holistic coverage and refer the reader to Darwiche (2009) and Pearl (2009). Here, it shall suffice that we refer to a directed acyclic graph (DAG) whose nodes represent random variables from the underlying distribution and whose edges reflect direct dependencies in the data. Edge directions are further interpreted as cause-effect relations. We now briefly summarise the inference of such graphs from data.

Generally, one distinguishes between *constraint-based* and *score-based* methods. The former use CIs inferred from data as constraints on where to draw edges. The latter explore the space of all possible DAGs over the given variables and assign scores to every visited graph. The output of the algorithm is the highest scoring graph. Since the space of DAGs over a set of variables or nodes grows super-exponentially in the set's cardinality, score-based methods often rely on greedy search techniques. In addition, *hybrid methods* combine both CIs as constraints and scoring of graphs to assess candidates.

In this work, we focus on greedy structure learning that performed best in recent benchmarks (Scutari et al., 2019a; Constantinou et al., 2021). More precisely, we rely on structure inference based on hill-climbing (HC) and TABU search (Russell and Norvig, 2009; Scutari et al., 2019b). Crucially, both algorithms use the Bayesian information criterion (Schwarz, 1978), which satisfies two key properties, *consistency* and *local consistency*[4] (Gámez et al., 2011; Chickering, 2003). Gámez et al. (2011) show that for HC for a dataset of size n and *iid* data, the output graph is a minimal I-Map of the underlying distribution if $n \to \infty$ and the scoring function satisfies *consistency* and *local consistency*. By definition of a minimal I-Map, the set of CIs represented by d-separation in the graph is a subset of the CIs in the distribution. Hence, while there might be independencies in the underlying distribution of the data not represented by d-separation, there are no instances of d-separations that do not correspond to independencies. Note that HC introduces a DAG structure of the output graph but the assumption on the data is just being an *iid* sample. The proof, however, hinges on the assumption of faithfulness. For linear models, though, the probability of faithfulness being violated is shown to be zero if model parameters are randomly drawn from positive densities (cf. Peters et al.

[4]The Bayesian Dirichlet equivalent uniform (BDeu) score satisfies the properties too and is a valid alternative.

(2017), Spirtes et al. (2000)). While there is no similar theoretical result for TABU, the latter is an extension of HC and exhibits similar behaviour in practice (cf. Section 5.1).

4 CAUSAL STRUCTURE LEARNING FOR EFFICIENT SAGE ESTIMATION

SAGE estimation is computationally challenging. For an exact computation, the surplus contribution of the feature of interest j with respect to every possible coalition $\mathbf{X}_S$ of the remaining features must be computed. The surplus contribution is defined as in Section 3.2

$$\Delta_{j|S} = \nu(\mathbf{X}_{S\cup j}) - \nu(\mathbf{X}_S) \tag{2}$$

The number of possible coalitions grows exponentially in the number of features, making the exact computation intractable in high-dimensional settings. SAGE values are therefore estimated by randomly sampling coalitions until the estimates converge (Section 3.2). Nevertheless, estimation remains challenging since evaluating $\Delta_{j|S}$ requires sampling from conditional distributions, and therefore even one evaluation is a significant computational challenge. Thus, in practice, the approximation quality is limited by the number of surplus contributions that can be computed.

We propose d-SAGE, an approach that can identify and skip unnecessary surplus evaluations and thereby allows to improve the approximation quality. The method is based on the observation that $\Delta_{j|S}$ evaluates to zero if X_j is conditionally independent of Y given X_S:

Theorem 1. *For ℓ being cross-entropy loss or the mean-squared error, f^* the respective optimal predictor and ν_{ℓ,f^*} the corresponding SAGE value function, it holds that*

$$X_j \perp Y | \mathbf{X}_S \Rightarrow \nu_{\ell,f^*}(\mathbf{X}_{S\cup j}) - \nu_{\ell,f^*}(\mathbf{X}_S) = 0.$$

Proof (sketch, full proof in A): Covert et al. (2020) show that for the cross entropy loss function with its respective optimal model, the Bayes classifier, Equation 2 equals the conditional mutual information of X_j and Y given $\mathbf{X}_S$, i.e. $I(X_j; Y|\mathbf{X}_S)$. A similar result holds for optimal regression models with the mean squared error (MSE) as loss function. In this case, the surplus contribution is shown to be equal to $\mathbb{E}_{\mathbf{X}_S}[Var(\mathbb{E}[Y|\mathbf{X}_S, X_j]|\mathbf{X}_S)]$ (Covert et al., 2020). For both expressions, one can easily see that they evaluate to zero when X_j is conditionally independent of Y given $\mathbf{X}_S$, i.e. $X_j \perp Y | \mathbf{X}_S$.

As a consequence of Theorem 1, knowledge of the dependence structure in the data allows speeding up the SAGE estimation procedure: evaluations of $\nu(\mathbf{X}_{S\cup j}) - \nu(\mathbf{X}_S)$ can be skipped if $X_j \perp Y | \mathbf{X}_S$.

To identify the CIs in the data, we suggest leveraging greedy procedures that were originally developed to learn

Algorithm 1: Sampling-based Approximation of d-SAGE

Input: Data $\{\mathbf{x}_i, y_i\}_{i=1}^n$ with $\mathbf{x}_i \in \mathbb{R}^d$, model $\hat{f}$, loss function ℓ, number of permutations n_π

Infer **DAG** $\mathcal{G}$ from data $\{\mathbf{x}_i, y_i\}_{i=1}^n$ with structure learning algorithm of choice.

for i *in* $\{1, ..., n_\pi\}$ **do**

    Sample a permutation π

    $S = \emptyset$

    for j *in* $\{1, ..., d\}$ **do**

        if $X_{\pi_j} \not\perp_{\mathcal{G}} Y | X_S$ **then**

            Sample $\mathbf{x}_{\bar{S}}$ from $p(\mathbf{x}_{\bar{S}} | \mathbf{x}_S)$

            Sample $\mathbf{x}_{\overline{S \cup \pi_j}}$ from $p(\mathbf{x}_{\overline{S \cup \pi_j}} | \mathbf{x}_S)$, where π_j is the j-th element of π

            $\hat{\Delta}_{j|S} = \ell(\hat{f}(\mathbf{x}_S, \mathbf{x}_{\bar{S}})) - \ell(\hat{f}(\mathbf{x}_{S \cup \pi_j}, \mathbf{x}_{\overline{S \cup \pi_j}}))$

        else

            $\hat{\Delta}_{j|S} = 0$

        end

        $S = S \cup \pi_j$

    end

end

return $\hat{\phi}_j = \frac{1}{n_\pi} \sum_{i=1}^{n_\pi} \hat{\Delta}_{j|S}$ for $j = 1, ..., d$

Note that we dropped indices of $\hat{\Delta}_{j|S}$ for readability.

---

the causal structure in the data. CSL algorithms allow the estimation of a causal graph in polynomial time (Scutari et al., 2019b). Given that the Markov property and faithfulness are fulfilled, the graph allows reading off (conditional) independencies in the data using linear time d-separation queries (Hagberg et al., 2008; Darwiche, 2009). Our rationale is that the one-time effort of learning the causal graph, as well as the additional linear time d-separation queries, are negligible in comparison to the computational overhead of computing the surplus contributions.[5]

To summarise, d-SAGE estimation introduces two key differences to the original SAGE approximation algorithm. First, a graph $\mathcal{G}$ is fitted over all random variables, the features, and the target. Second, the estimation of $\Delta_{j|S}$ is skipped if the current feature X_j in permutation π is d-separated from the target given the set $\mathbf{X}_S = \{X_i : \pi(i) < \pi(j)\}$. The changes are highlighted in blue in Algorithm 1.

5 EXPERIMENTS

The experiment section is divided into three parts. In the first two parts, we evaluate our method on synthetic data with known ground truth: As we use d-separation

queries in estimated graphs for d-SAGE approximation, we first evaluate the accuracy of d-separations in learned structures with regard to ground truth CIs in the data (Section 5.1). Then we compare d-SAGE to ordinary SAGE value approximation (Section 5.2). In the third part, we demonstrate the usefulness of the method in a real-world application (Section 5.3).[6]

5.1 Benchmark of Causal Structure Learning

Existing structure learning benchmarks evaluate the algorithms regarding how well they can recover the true causal structure (Constantinou et al., 2021; Scutari et al., 2019a). For d-SAGE, however, we are only interested in learning the dependence structure. As such, we assess how well CIs are inferred as d-separations in the estimated graph.

5.1.1 Setup

We evaluate the greedy search algorithms HC and TABU (Scutari et al., 2019b; Russell and Norvig, 2009). We selected these methods based on their superior performance in recent CSL benchmarks (Constantinou et al., 2021; Scutari et al., 2019a). As performance metrics, we employ the F1 score for the detection of d-separations w.r.t. a randomly sampled target Y as well as the respective false discovery rate. More precisely, for every potential d-separation of the form $X_j \perp_{\mathcal{G}} Y | \mathbf{X}_S$, we check whether it had the same status in the ground truth and the estimated graph. To cope with the exponentially large number of d-separations in the higher dimensional graphs (DAG_{sm}, DAG_m and DAG_l) we randomly sampled a node of interest X_j and a conditioning set $\mathbf{X}_S$ one million times instead of iterating over all potential d-separation statements. For both algorithms, we relied on their implementation in *bnlearn* (Scutari, 2010) for R.[7] We consider twelve different synthetic data settings with known ground truth:

DAG$_s$, DAG$_{sm}$, DAG$_m$ and DAG$_l$ We sampled synthetic graphs with a varying number of nodes ($s = 10$, $sm = 20$, $m = 50$ and $l = 100$) and three different densities (average adjacency degrees of 2, 3 and 4). Based on the graphs, we sampled data from the corresponding linear Gaussian data model, where absolute values of edge weights are bounded by 0.5 and 2. We standardised variances to be (approximately) one to avoid that they increase with the topological ordering and counteract a potential bias in the benchmark (Reisach et al., 2021). For the sampling itself, we relied on the the *pcalg* package (Kalisch et al., 2012) implemented in R (R Core Team, 2022).

---

[5]In general, the complexity of conditional sampling depends on the assumptions about the data generating process. In their tractability analysis for SHAP, Van den Broeck et al. (2022) assume polynomial complexity for computing the conditional expectations of the form $\mathbb{E}_{\mathbf{X}_{\bar{S}} | \mathbf{X}_S}[\hat{f}(\mathbf{X}) | \mathbf{X}_S = \mathbf{x}_S]$.

[6]All code is publicly available https://github.com/gcskoenig/csl-experiments/tree/camera-ready.

[7]All graph learning experiments were run on an Intel Core i7-8700K Desktop CPU.

5.1.2 Results

First, we observe TABU, while approximately taking double the time, either performs equally well as or better than HC (cf. Figures 1, 2 and Appendix C). Hence, we restrict this section to results for TABU search, which we also employed for d-SAGE estimation. Figure 1 shows the runtime of graph learning depending on sample size and corresponding F1 scores for d-separation inference for all twelve graphs. The key takeaway is that for the sparsest graph (average adjacency degree 2) the F1 score is greater than 0.88 if $n \geq 10,000$. For the larger graphs, however, there is a slight drop-off in performance, which is expected. Only for the densest graph setting (average adjacency degree 4) and for 50 and 100 nodes, though, a larger sample size, i.e. $n \geq 100,000$, is required to infer d-separations at a reasonable rate. As we will see in Section 5.2, the runtime for graph learning is negligible in the context of d-SAGE estimation.
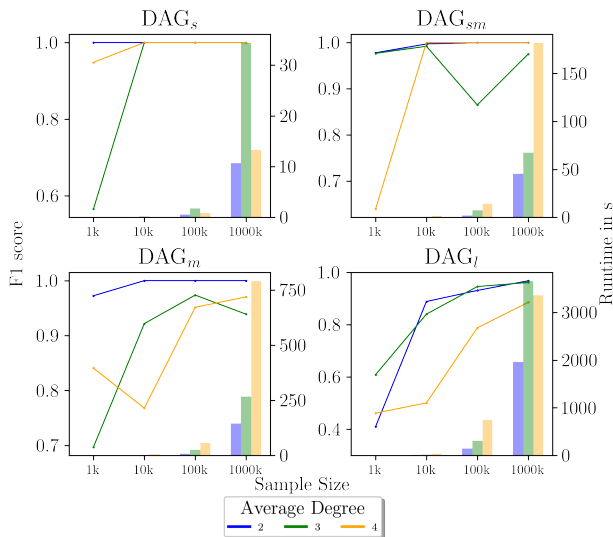

Figure 2: Confusion matrix for true and predicted d-connections ($\not\perp_{\mathcal{G}}$) and d-separations ($\perp_{\mathcal{G}}$) based on TABU search with $n = 10,000$ for all twelve graphs.

Figure 1: F1 scores for d-separation (lines, left y-axes) and runtime of graph learning (bars, right y-axes) using TABU search depending on sample size.

We note that there is no well-defined threshold for the minimal F1 score that would be required for SAGE estimation to benefit from causal structure learning because different error types have distinct consequences. While incorrectly inferred d-separations may lead to biased estimates, non-detected d-separations only reduce the benefit of CSL in terms of reduced runtime. Importantly, our simulation results in Figure 2 show virtually no false discoveries (cases where there is no CI in the underlying distribution but a d-separation is inferred) yet some false-negative instances, which leads to fewer skipped evaluations of $\Delta_{j|S}$ than warranted. This result is in accordance with the reasoning presented in Section 3.3. As such, the use of CSL is a con-
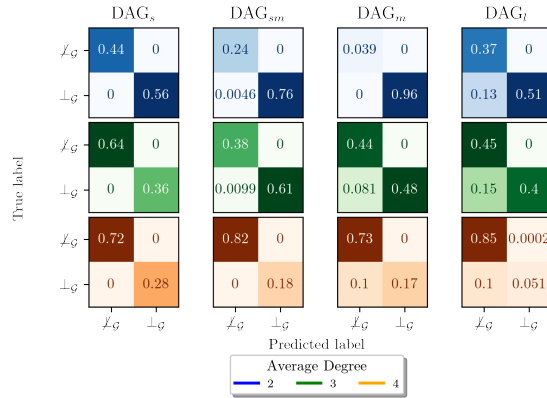
servative approach to the inference of CIs. Note that for the data used in the benchmark, the ground-truth graph is known and the Markov property and faithfulness hold, such that d-separations indeed coincide with statistical independence.

5.2 Evaluating Efficiency and Accuracy of d-SAGE

In the benchmark study in Section 5.1 we highlight the capability of structure learning to efficiently yet conservatively estimate d-separations as equivalents to CIs. We now evaluate d-SAGE regarding its efficiency and its accuracy.

5.2.1 Setup

To evaluate d-SAGE in practice, a linear model (LM) and a random forest (RF) are fitted to each of the twelve datasets (using the *scikit-learn* implementation with default settings (Pedregosa et al., 2011)). As loss function, the mean squared error (MSE) is used for either of them. Hence, the linear model (LM) falls into the category of optimal models required for the theoretical justification. The RF model serves as a sanity check for a high-performing, but not optimal model (cf. Appendix D for the model performances). For a fair comparison, we compare d-SAGE and SAGE based on the exact same feature orderings. This also allowed us to compare the skipped evaluations of $\Delta_{j|S}$, that are set to zero, to their estimated counterparts that should be very close to zero. Overall, we estimated SAGE and d-SAGE values five times for each setup (graph + model). We used the same synthetic datasets for the evaluation as in Section 5.1.
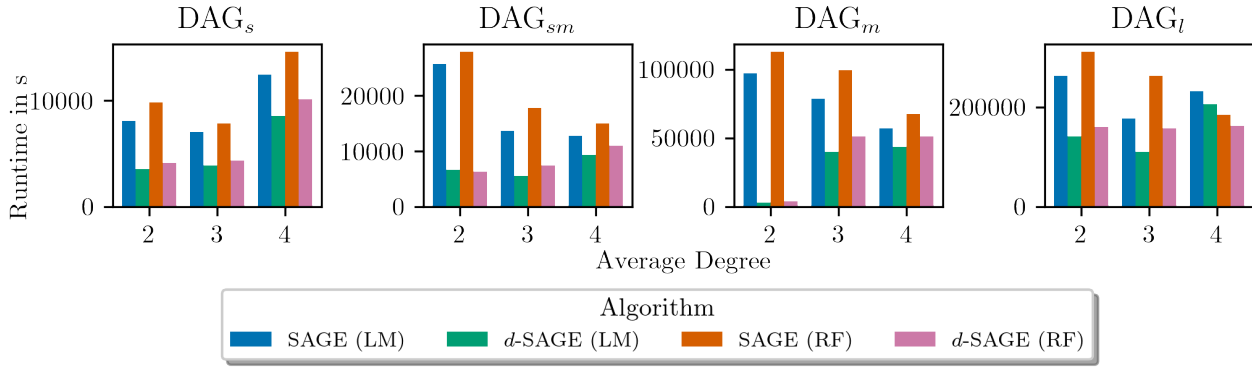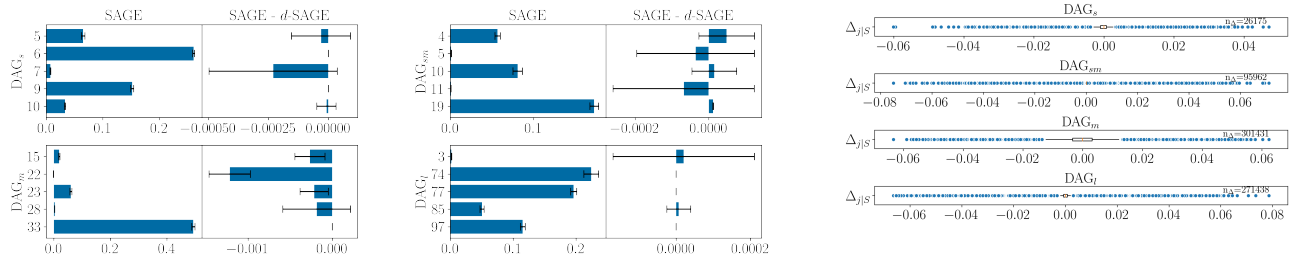
Figure 3: Runtime estimates for SAGE and d-SAGE for all twelve graphs and linear models (LM) as well as random forests (RF) based on $n = 10,000$.

(a) SAGE values and difference between SAGE and d-SAGE for the five largest values.

(b) Boxplots showing the distribution of $\Delta_{j|S}$ for the skipped surplus evaluations.

Figure 4: Results on the approximation quality of d-SAGE based on DAGs with average degree 2 for optimal models (LM). Based on five (d-)SAGE estimates.

5.2.2   Results

We find that d-SAGE indeed speeds up SAGE approximation as expected. More specifically, the estimated runtime[8] decreases by a rate that is approximately equal to the share of CIs w.r.t. the model target (cf. Appendix C) for both model classes across all graphs (cf. Figure 3). Furthermore, d-SAGE manages this speedup without distorting the estimates. Note that we do not include graph learning runtime in Figure 3 since it required between 0.06 seconds (DAG$_s$ with average degree 2) and 39.86 seconds (DAG$_l$ with average degree 4) and hence is negligible in this context.

Linear Model Figure 4 (a) displays the five SAGE values with the largest absolute value for the four graphs with an average degree of two along with the respective differ-

ence between the SAGE and d-SAGE estimates. Overall, the differences are about three orders of magnitude smaller than the original SAGE values, i.e. typically lie beneath one per cent. Even the most pronounced difference for variable 7 in DAG$_s$ only amounts to approximately 2.7 per cent of the SAGE value of approximately 0.007. We find no further striking differences in the remaining SAGE values that identified important features, i.e. those with the largest absolute SAGE values. Features deemed unimportant by SAGE values are detected as such by d-SAGE. Noteworthy, some d-SAGE estimates are equal to zero if the feature of interest is conditionally independent of the target given all (sampled) coalitions. Here, we argue that we bias observational SAGE values towards zero, which for truly independent features is closer (or equal) to the 'true-to-the-data' measure that would be achieved for the optimal predictor and infinite data.

Figure 4 (b) displays every $\Delta_{j|S}$ value, which was derived from a conditionally independent feature that was detected as such and thus set equal to zero in d-SAGE approximation. We see clearly that most values are very close to zero, as mirrored by the narrow boxes, which underlines the use-

---

[8]The complete SAGE estimation was performed on multiple different machines. For a fair evaluation of runtime, we relied on estimates that were performed on the same CPU (Intel Core i7-8700K Desktop CPU): Either approach was conducted using 100 permutations that were the same for SAGE and d-SAGE and runtime multiplied by the factor $\frac{n_\pi}{100}$, where n_π is the number of permutations after which one SAGE run converged. For convergence behaviour see Appendix E.

fulness of our approach.

Random Forest In order to test the sensitivity of the results, we replicated the exact same study using a high-performing but not optimal RF regressor (instead of the optimal LM). While the runtime savings are the same as for the LM, deviations of d-SAGE values from the original estimates are slightly more pronounced (cf. Appendix D). The results indicate that our approach is also useful for close to optimal models.

5.3 Real-world Application

To show the usefulness of d-SAGE in practice, we applied the approach to drug consumption data from the UCI ML repository (Dua and Graff, 2017). The target "Nicotine consumption" was predicted using logistic regression relying on twelve explanatory variables in a dataset with sample size $n = 1885$. Graph fitting was conducted with the TABU search algorithm and took 0.035 seconds. SAGE estimation for five different runs took approximately 12h14min[9]. To derive d-SAGE values, we did not rerun the estimation relying on d-SAGE but simply replaced the respective $\Delta_{j|S}$ that pertained to a d-separation in the fitted graph in the output (that included all such $\Delta_{j|S}$) with zero. We found approximately 38 per cent such $\Delta_{j|S}$ values that can be skipped which warrants an (almost) equally large relative speedup.
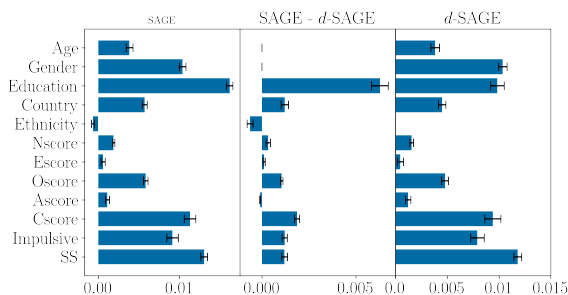
Figure 5: SAGE values, difference between SAGE and d-SAGE and d-SAGE values for drug consumption data. Based on five (d-)SAGE estimates.

Figure 5 shows that d-SAGE values are mostly in accordance with the original SAGE estimate. From the important variables, only *'Education'* has a markedly distinct d-SAGE value as it is reduced by about a third compared to the SAGE estimate. Yet, it is still assigned relatively high importance. The efficacy of d-SAGE in practice is further highlighted by the $\Delta_{j|S}$ values that hover around zero, as shown in Figure 6.

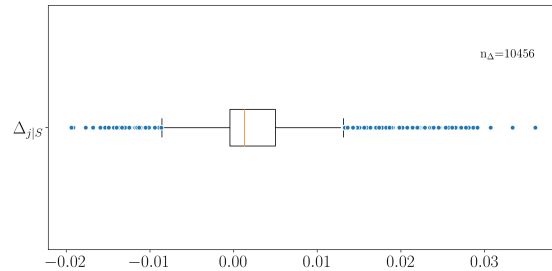[9]All calculations were run on an Intel Core i7-8700K Desktop CPU

Figure 6: Boxplots showing the distribution of $\Delta_{j|S}$ for the skipped surplus evaluations.

6 DISCUSSION

Model optimality and loss Conditional SAGE values are particularly appealing for scientific inference, i.e. to learn about the data (Chen et al., 2020; Covert et al., 2020). Therefore, in general, accurate predictors are required (Molnar et al., 2022). However, the requirement is of increased importance for d-SAGE since if the assumption of model optimality is violated the interpretation may be further biased by skipping the evaluation of non-zero surplus contributions (Theorem 1).

Assumptions for CSL CSL is enabled by causal sufficiency, the Markov property and faithfulness (Peters et al., 2017). The assumptions ensure that all relevant variables are observed, and that CIs in the data coincide with d-separations in the true causal graph (which we assume to be a DAG). We conjecture that violations of these assumptions are not vital for our approach since learning the true causal graph is not the goal. Instead, we are only interested in learning the graph to encode (conditional) independencies present in the observational distribution (irrespective of which causal mechanism they stem from). DAGs learned by HC being a minimal I-Map of the underlying distribution makes it suitable for probabilistic inference of CIs without guarantees of a correct graph or the number of CIs uncovered.

Nevertheless, practitioners should carefully assess the assumptions before applying d-SAGE. In the presence of latent confounders or cyclic assignment, for example, one may consider other concepts, such as m-separation and σ-separation (cf. Bongers et al. (2021)). Moreover, it is advisable to perform sanity checks on whether skipped surplus contributions are actually evaluated to zero.

Use of Score-based CSL The analysis was restricted to the use of score-based CSL because of its efficiency. HC is particularly appealing since it infers a minimal I-Map of the underlying distribution as explained in Section 3.3, and TABU performed well empirically. However, inference of CIs is not limited to those techniques. Graph learning can be performed with an algorithm of choice and under

consideration of the assumptions employed, as explained above. Moreover, the rationale behind our approach is to replace CI testing by CSL. Partial correlation tests, for example, are considerably less efficient than d-separation queries (cf. Appendix F) and thus would require a larger number of CIs to achieve a speedup of SAGE.

7 CONCLUSION

We proposed d-SAGE, a method that exploits the dependence structure in the data to speed up SAGE estimation. More specifically, we observe that conditional independence in the data implies that the corresponding surplus contribution can be directly evaluated to zero. We modify the ordering based SAGE approximation algorithm to first learn the dependence structure in the data using CSL algorithms and to then skip surplus contribution evaluations if the graph encodes a CI. Errors in the learned graph may either slow down convergence (if CIs are not discovered) or bias the result towards zero (in case of false discoveries). However, in our experiments, there were nearly no false discoveries, such that the resulting estimates for features that were not conditionally independent given every coalition essentially converged to the same values as the original SAGE approximation algorithm. Furthermore, the CSL algorithms were able to uncover most CIs, such that we observe significant performance gains. As such, given a fixed computational budget, the efficiency gains of d-SAGE can enable a more accurate estimation of SAGE values than the approximation algorithm proposed by Covert et al. (2020). In future work, it would be interesting to combine d-SAGE with the permutation sampling by Mitchell et al. (2022) and to assess whether the results can be translated to other Shapley based interpretability methods such as SHAP.

Acknowledgements

References

Aas, K., Jullum, M., and Løland, A. (2021). Explaining individual predictions when features are dependent: More accurate approximations to shapley values. *Artificial Intelligence*, 298:103502.

Bashtannyk, D. M. and Hyndman, R. J. (2001). Bandwidth selection for kernel conditional density estimation. *Computational Statistics & Data Analysis*, 36(3):279–298.

Bénard, C., Biau, G., Da Veiga, S., and Scornet, E. (2022). Shaff: Fast and consistent shapley effect estimates via random forests. In *International Conference on Artificial Intelligence and Statistics*, pages 5563–5582. PMLR.

Bhatt, U., Xiang, A., Sharma, S., Weller, A., Taly, A., Jia, Y., Ghosh, J., Puri, R., Moura, J. M., and Eckersley, P. (2020). Explainable machine learning in deployment. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 648–657.

Bishop, C. M. (1994). Mixture density networks. Technical report, Aston University.

Blesch, K., Watson, D. S., and Wright, M. N. (2022). Conditional feature importance for mixed data. *arXiv preprint arXiv:2210.03047*.

Bongers, S., Forré, P., Peters, J., and Mooij, J. M. (2021). Foundations of structural causal models with cycles and latent variables. *The Annals of Statistics*, 49(5):2885 – 2915.

Chen, H., Janizek, J. D., Lundberg, S., and Lee, S.-I. (2020). True to the model or true to the data? *arXiv preprint arXiv:2006.16234*.

Chickering, D. M. (2003). Optimal structure identification with greedy search. *J. Mach. Learn. Res.*, 3(null):507–554.

Constantinou, A. C., Liu, Y., Chobtham, K., Guo, Z., and Kitson, N. K. (2021). Large-scale empirical validation of bayesian network structure learning algorithms with noisy data. *International Journal of Approximate Reasoning*, 131:151–188.

Covert, I. and Lee, S.-I. (2021). Improving kernelshap: Practical shapley value estimation using linear regression. In *International Conference on Artificial Intelligence and Statistics*, pages 3457–3465. PMLR.

Covert, I., Lundberg, S. M., and Lee, S.-I. (2020). Understanding global feature contributions with additive importance measures. *Advances in Neural Information Processing Systems*, 33:17212–17223.

Darwiche, A. (2009). *Modeling and Reasoning with Bayesian Networks*. Cambridge University Press.

Datta, A., Sen, S., and Zick, Y. (2016). Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems. In *2016 IEEE symposium on security and privacy (SP)*, pages 598–617. IEEE.

Dua, D. and Graff, C. (2017). UCI machine learning repository.

Farrell, S., Calafiura, P., Mudigonda, M., Anderson, D., Vlimant, J.-R., Zheng, S., Bendavid, J., Spiropulu, M., Cerati, G., Gray, L., et al. (2018). Novel deep learning methods for track reconstruction. *arXiv preprint arXiv:1810.06111*.

Freiesleben, T., König, G., Molnar, C., and Tejero-Cantero, A. (2022). Scientific inference with interpretable ma-

chine learning: Analyzing models to learn about real-world phenomena. *arXiv preprint arXiv:2206.05487*.

Frye, C., de Mijolla, D., Begley, T., Cowton, L., Stanley, M., and Feige, I. (2020a). Shapley explainability on the data manifold.

Frye, C., Rowat, C., and Feige, I. (2020b). Asymmetric shapley values: Incorporating causal knowledge into model-agnostic explainability. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA. Curran Associates Inc.

Gámez, J., Mateo, J. L., and Puerta, J. (2011). Learning bayesian networks by hill climbing: efficient methods based on progressive restriction of the neighborhood. *Data Mining and Knowledge Discovery*, 22(1):106–148.

Hagberg, A., Swart, P., and S Chult, D. (2008). Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States).

Heskes, T., Sijben, E., Bucur, I. G., and Claassen, T. (2020). Causal shapley values: Exploiting causal knowledge to explain individual predictions of complex models. *Advances in neural information processing systems*, 33:4778–4789.

Hothorn, T. and Zeileis, A. (2021). Predictive distribution modeling using transformation forests. *Journal of Computational and Graphical Statistics*, 30(4):1181–1196.

Jethani, N., Sudarshan, M., Covert, I. C., Lee, S.-I., and Ranganath, R. (2021). Fastshap: Real-time shapley value estimation. In *International Conference on Learning Representations*.

Kalisch, M., Mächler, M., Colombo, D., Maathuis, M. H., and Bühlmann, P. (2012). Causal inference using graphical models with the R package pcalg. *Journal of Statistical Software*, 47(11):1–26.

Li, X., Zhou, Y., Dvornek, N. C., Gu, Y., Ventola, P., and Duncan, J. S. (2020). Efficient shapley explanation for features importance estimation under uncertainty. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 792–801. Springer.

Luan, H. and Tsai, C.-C. (2021). A review of using machine learning approaches for precision education. *Educational Technology & Society*, 24(1):250–266.

Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30.

Mitchell, R., Cooper, J., Frank, E., and Holmes, G. (2022). Sampling permutations for shapley value estimation.

Molnar, C. (2020). *Interpretable machine learning*. Lulu. com.

Molnar, C., König, G., Herbinger, J., Freiesleben, T., Dandl, S., Scholbeck, C. A., Casalicchio, G., Grosse-Wentrup, M., and Bischl, B. (2022). General pitfalls of model-agnostic interpretation methods for machine learning models. In *International Workshop on Extending Explainable AI Beyond Deep Models and Classifiers*, pages 39–68. Springer.

Obermeyer, Z. and Mullainathan, S. (2019). Dissecting racial bias in an algorithm that guides health decisions for 70 million people. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 89–89.

Page Jr, T. J. (1984). Multivariate statistics: A vector space approach. *JMR, Journal of Marketing Research (pre-1986)*, 21(000002):236.

Pearl, J. (2009). *Causality*. Cambridge University Press, 2 edition.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Peters, J., Janzing, D., and Schölkopf, B. (2017). *Elements of Causal Inference: Foundations and Learning Algorithms*. MIT Press, Cambridge, MA, USA.

R Core Team (2022). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Raghavan, M., Barocas, S., Kleinberg, J., and Levy, K. (2020). Mitigating bias in algorithmic hiring: Evaluating claims and practices. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, FAT* '20, page 469–481, New York, NY, USA. Association for Computing Machinery.

Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., Carvalhais, N., et al. (2019). Deep learning and process understanding for data-driven earth system science. *Nature*, 566(7743):195–204.

Reisach, A. G., Seiler, C., and Weichwald, S. (2021). Beware of the simulated dag! causal discovery benchmarks may be easy to game.

Russell, S. and Norvig, P. (2009). *Artificial Intelligence: A Modern Approach*. Prentice Hall, 3 edition.

Schmidt, J., Marques, M. R., Botti, S., and Marques, M. A. (2019). Recent advances and applications of machine learning in solid-state materials science. *npj Computational Materials*, 5(1):1–36.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464.

Scutari, M. (2010). Learning bayesian networks with the bnlearn r package. *Journal of Statistical Software*, 35(3):1–22.

Scutari, M., Graafland, C. E., and Gutiérrez, J. M. (2019a). Who learns better bayesian network structures: Accuracy and speed of structure learning algorithms. *International Journal of Approximate Reasoning*, 115:235–253.

Scutari, M., Vitolo, C., and Tucker, A. (2019b). Learning bayesian networks from big data with greedy search: computational complexity and efficient implementation. *Statistics and Computing*, 29(5):1095–1108.

Senior, A. W., Evans, R., Jumper, J., Kirkpatrick, J., Sifre, L., Green, T., Qin, C., Žídek, A., Nelson, A. W., Bridgland, A., et al. (2020). Improved protein structure prediction using potentials from deep learning. *Nature*, 577(7792):706–710.

Shapley, L. S. (1953). Stochastic games. *Proceedings of the national academy of sciences*, 39(10):1095–1100.

Sohn, K., Lee, H., and Yan, X. (2015). Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28.

Spirtes, P., Glymour, C., and Scheines, R. (2000). *Causation, Prediction, and Search*. MIT press, 2nd edition.

Štrumbelj, E. and Kononenko, I. (2014). Explaining prediction models and individual predictions with feature contributions. *Knowledge and information systems*, 41(3):647–665.

Sundararajan, M. and Najmi, A. (2020). The many shapley values for model explanation. In *International conference on machine learning*, pages 9269–9278. PMLR.

Trippe, B. L. and Turner, R. E. (2018). Conditional density estimation with bayesian normalising flows. *arXiv preprint arXiv:1802.04908*.

Vallat, R. (2018). Pingouin: statistics in python. *Journal of Open Source Software*, 3(31):1026.

Van den Broeck, G., Lykov, A., Schleich, M., and Suciu, D. (2022). On the tractability of shap explanations. *Journal of Artificial Intelligence Research*, 74:851–886.

Wang, J., Wiens, J., and Lundberg, S. (2021). Shapley flow: A graph-based approach to interpreting model predictions. In *International Conference on Artificial Intelligence and Statistics*, pages 721–729. PMLR.

Winkler, C., Worrall, D., Hoogeboom, E., and Welling, M. (2019). Learning likelihoods with conditional normalizing flows. *arXiv preprint arXiv:1912.00042*.

Zeng, J., Ustun, B., and Rudin, C. (2017). Interpretable classification models for recidivism prediction. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 180(3):689–722.

A PROOF OF THEOREM 1

Theorem 1. *For ℓ being cross-entropy loss or the mean-squared error, f^* the respective optimal predictor and ν_{ℓ,f^*} the corresponding SAGE value function, it holds that*

$$X_j \perp Y | \mathbf{X}_S \Rightarrow \nu_{\ell,f^*}(\mathbf{X}_{S \cup j}) - \nu_{\ell,f^*}(\mathbf{X}_S) = 0.$$

Proof. Mean Squared Error: Covert et al. (2020) show that for ℓ being the mean squared error and f^* the corresponding optimal predictor it holds that:

$$\nu(\mathbf{X}_{S \cup j}) - \nu(\mathbf{X}_S) = \mathbb{E}[\text{Var}(Y|\mathbf{X}_S)] - \mathbb{E}[\text{Var}(Y|\mathbf{X}_{S \cup j})]$$

Under conditional independence $Y \perp X_j | \mathbf{X}_S$ it follows that

$$\mathbb{E}[\text{Var}(Y|\mathbf{X}_{S \cup j})] = \mathbb{E}[\mathbb{E}[\text{Var}(Y|\mathbf{X}_{S \cup j})|\mathbf{X}_S]]$$
$$= \mathbb{E}[\text{Var}(Y|\mathbf{X}_S)]$$

and consequently $Y \perp X_j | \mathbf{X}_S \Rightarrow \nu(\mathbf{X}_{S \cup j}) - \nu(\mathbf{X}_S) = 0$.

Cross Entropy: Covert et al. (2020) show that given cross entropy as loss and the corresponding loss optimal predictor f^* it holds that:

$$\nu(\mathbf{X}_{S \cup j}) - \nu(\mathbf{X}_S) = I(Y; X_j | \mathbf{X}_S)$$

Mutual information $I(Y; X_j | \mathbf{X}_S)$ is zero if and only if $Y \perp X_j | \mathbf{X}_S$. Consequently $\nu(\mathbf{X}_{S \cup j}) - \nu(\mathbf{X}_S) = 0 \Leftrightarrow X_j \perp Y | \mathbf{X}_S$.
$$\square$$

B SAGE VALUE PROPERTIES

As mentioned in Section 3, SAGE values satisfy certain fairness properties that are deduced from those valid for Shapley values (Covert et al., 2020). While not explicitly named after the Shapley value properties (*efficiency*, the *dummy property*, *symmetry*, *monotonicity*, *linearity*) we employ these terms for the SAGE properties for simplicity:

1. *Efficiency*: $\sum_{j=1}^{d} \phi_j(\nu) = \nu(\mathbf{X})$, where $\mathbf{X}$ is the set of all features.

2. *Dummy property*: $\phi_j(\nu) = 0$ if $X_j \perp \hat{f}(\mathbf{X}) | \mathbf{X}_S$ for all $S \subseteq \{1, ..., d\} \setminus j$.

3. *Symmetry*: $\nu(\mathbf{X}_{S \cup j}) = \nu(\mathbf{X}_{S \cup i})$ for two variables X_j and X_i with a deterministic relationship.

4. *Monotonicity*: For two target variables Y, Y' and corresponding models $\hat{f}$, $\hat{f}'$: $\phi_j(\nu_{\hat{f}}) \geq \phi_j(\nu_{\hat{f}'})$ if $\nu_{\hat{f}}(\mathbf{X}_{S \cup j}) - \nu_{\hat{f}}(\mathbf{X}_S) \geq \nu_{\hat{f}'}(\mathbf{X}_{S \cup j}) - \nu_{\hat{f}'}(\mathbf{X}_S)$ for all $S \subseteq \{1, ..., d\} \setminus j$.

5. From *Linearity*: $\phi_j(\nu) = \mathbb{E}_{\mathbf{X},Y}[\phi_j(\nu_{\hat{f},x,y})]$, where $\phi_j(\nu_{\hat{f},x,y})$ is the Shapley value of the game $\nu_{\hat{f},x,y}(\mathbf{X}_S) = \ell(\hat{f}_\emptyset(\mathbf{X}_\emptyset), y) - \ell(\hat{f}_S(\mathbf{X}_S), y)$

6. SAGE values are invariant to invertible mappings applied to the input, e.g. they are the same for original input data and and their log values.

C GRAPH BENCHMARK

In this section, we provide detailed information about the graphs employed in Section 5, the graph learning algorithms and the graph benchmark. Additionally, we present results derived from the HC algorithm for CSL.

C.1 Overview of Graphs

In Table 1 we provide an overview of all twelve graphs used in Section 5, the randomly sampled target, the adjacency degree of the target and the share of d-separations w.r.t. the target. This gives further insight into the relation of graph sparsity, degree of target and share of d-separations. The latter can be regarded as the potential relative runtime decrease for SAGE approximation.

Table 1: Overview of all twelve graphs used in Section 5, the randomly sampled target, the adjacency degree of the target and the share of d-separations w.r.t. the target.

GRAPH (AVG. DEGREE)	TARGET	DEGREE OF TARGET	SHARE OF $\perp_{\mathcal{G}}$
$\text{DAG}_s(2)$	8	2	0.556
$\text{DAG}_s(3)$	1	2	0.357
$\text{DAG}_s(4)$	1	4	0.283
$\text{DAG}_{sm}(2)$	17	1	0.765
$\text{DAG}_{sm}(3)$	2	1	0.623
$\text{DAG}_{sm}(4)$	16	4	0.185
$\text{DAG}_m(2)$	4	1	0.961
$\text{DAG}_m(3)$	32	5	0.556
$\text{DAG}_m(4)$	2	3	0.274
$\text{DAG}_l(2)$	4	3	0.632
$\text{DAG}_l(3)$	66	3	0.552
$\text{DAG}_l(4)$	66	7	0.151

Table 2 shows the hyperparameter settings used for CSL relying on the *bnlearn* package (Scutari, 2010) for R (R Core Team, 2022).

Table 2: Hyperparameters Used for Graph Learning

ALGORITHM	HYPERPARAMETERS
HC	Max. iterations ∞, max. in-degree: ∞; score: BIC
TABU	Size of list: 10; Max. iterations ∞, max. in-degree: ∞; score: BIC

C.2 MC Sampling for d-separation Inference

In Algorithm 2 we explicate how we inferred the number of true positive, false positive, true negative and false negative d-separations within an estimated graph and especially how we dealt with the exponential number of potential conditioning sets for the larger graphs.

Algorithm 2: Monte Carlo Sampling for d-separation Inference

Input: True graph $\mathcal{G}^*$ and estimated graph $\mathcal{G}$ over node set $\{X_1, X_2, ...X_d, Y\}$ with target node Y; Number of MC samples n_{mc}
Output: True positives, true negatives, false positives and false negatives for inferred d-separations in $\mathcal{G}$: TP, TN, FP, FN
Set $TP = TN = FP = FN = 0$
for $m = 1, ..., n_{mc}$ **do**

    Randomly draw a node X_j from $\{X_1, X_2, ...X_d\}$

    Randomly draw size n_s of conditioning set $\mathbf{X}_S$ from discrete probability distribution $P(n_s = i) = \frac{\binom{d-1}{i}}{2^{d-1}}, i \in \{0, ..., d-1\}$
    Randomly draw elements $X_i, i = 1, ...n_s$, from $\{X_1, X_2, ...X_d\} \setminus X_j$ without replacement and set $\mathbf{X}_S = \{X_i\}_{i=1,...,n_s}$
    if $X_j \perp_{\mathcal{G}^*} Y|X_S$ **then**

        if $X_j \perp_{\mathcal{G}} Y|X_S$ **then**
            TP = TP+1
        else
            FN = FN+1
        end

    else

        if $X_j \not\perp_{\mathcal{G}} Y|X_S$ **then**
            TN = TN+1
        else
            FP = FP+1
        end

    end

end
Return: TP, TN, FP, FN

C.3 Results - HC

In Figure 7 we show the results of the graph learning benchmark for HC in contrast to those from Section 5. As HC never performed better but for some experiments worse than TABU, we chose the latter for the use in d-SAGE.

(a) F1 scores for d-separation (lines, left y-axes) and runtime of graph learning (bars, right y-axes) using HC depending on sample size.

(b) Confusion matrix for true and predicted d-connections ($\not\perp_{\mathcal{G}}$) and d-separations ($\perp_{\mathcal{G}}$) based on HC with $n = 10,000$ for all twelve graphs.
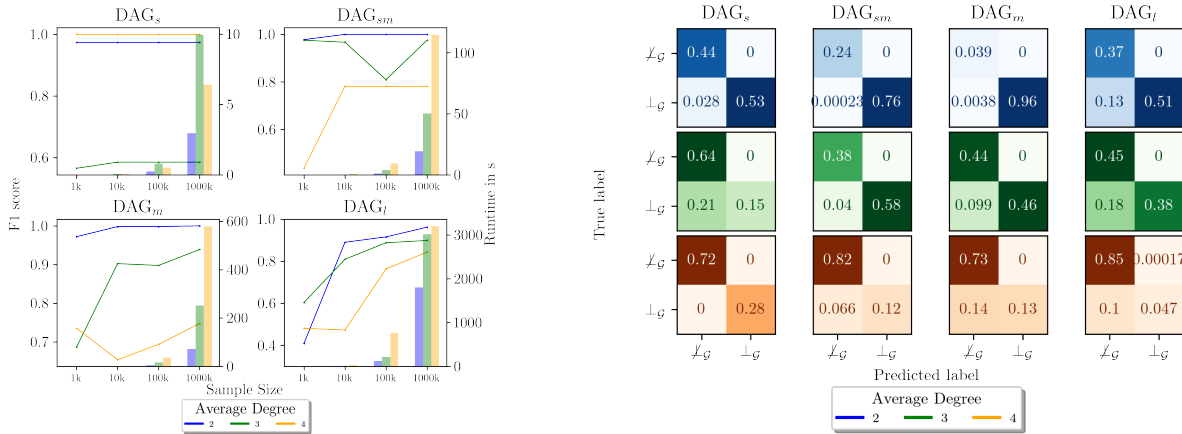
Figure 7: Results from graph learning benchmark for HC algorithm.

D SAGE - EXPERIMENTS

In this section, we briefly explain the experiment setup and afterwards present missing results. For our analysis, we fitted two models, LM and RF, for every dataset relying on the same targets that were sampled randomly for the analysis of d-separations in a graph. We relied on $n_{train} = 8000$ for model fitting and $n_{test} = 2000$ for model evaluation (the same $n = 10000$ data points as used for graph fitting and SAGE inference). We then used the data to estimate SAGE and d-SAGE five times, i.e. we were provided five approximations of $(d\text{-})$SAGE for every graph and model, which were then used to provide error bounds. The $\Delta_{j|S}$ plots rely on skipped evaluations of each of these runs.

In Table 3 we provide performance measures of the models and in Appendix D.1 the plots pertaining to experiments not shown in Section 5 are displayed. Note that Table 3 highlights that RF performs slightly worse than the optimal LM throughout all settings and with regard to the MSE and R^2.

Table 3: Details of Linear Models (LMs) and Random Forests (RF); Random Forests based on 100 Tree Estimators.

DATA (AVERAGE DEGREE)	$\mathbf{n}_{train}; \mathbf{n}_{test}$	$\mathbf{MSE}_{LM}$	$\mathbf{R}^2_{LM}$	$\mathbf{MSE}_{RF}$	$\mathbf{R}^2_{RF}$
$DAG_s(2)$	8000; 2000	0.541	0.495	0.572	0.466
$DAG_{sm}(2)$	8000; 2000	0.035	0.963	0.038	0.960
$DAG_m(2)$	8000; 2000	0.474	0.522	0.498	0.498
$DAG_l(2)$	8000; 2000	0.070	0.930	0.103	0.897
$DAG_s(3)$	8000; 2000	0.382	0.616	0.480	0.517
$DAG_{sm}(3)$	8000; 2000	0.072	0.926	0.078	0.921
$DAG_m(3)$	8000; 2000	0.089	0.914	0.174	0.832
$DAG_l(3)$	8000; 2000	0.065	0.938	0.082	0.922
$DAG_s(4)$	8000; 2000	0.101	0.902	0.161	0.843
$DAG_{sm}(4)$	8000; 2000	0.075	0.925	0.086	0.914
$DAG_m(4)$	8000; 2000	0.163	0.840	0.194	0.810
$DAG_l(4)$	8000; 2000	0.004	0.996	0.059	0.943

D.1 Results - SAGE and d-SAGE

In this section we provide the same results as in Section 5 for all missing setups and both models, LM and RF as well as the top fifteen values for the setup presented in Section 5. Overall, we can confirm our findings in the different settings.
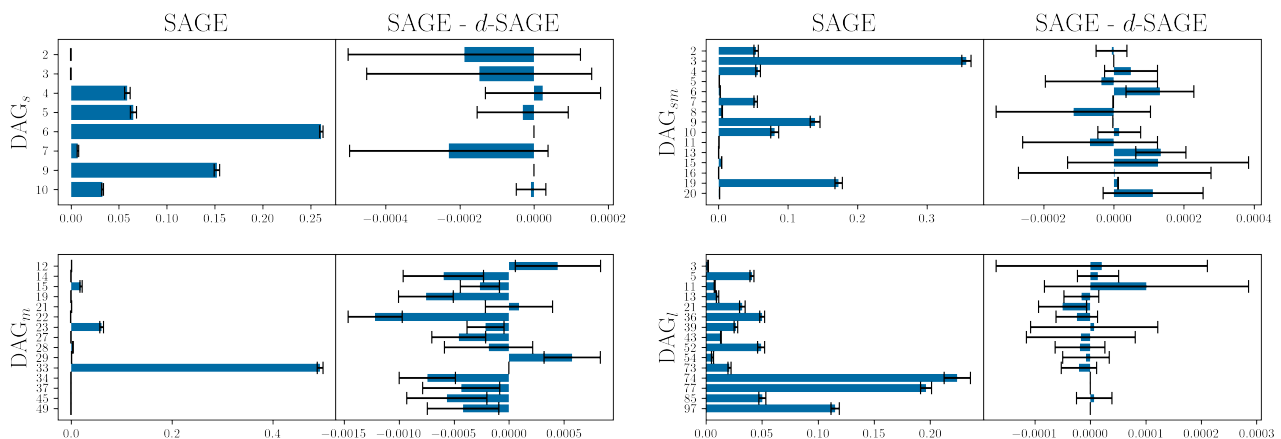

Figure 8: SAGE values and difference between SAGE and d-SAGE for the fifteen (all for DAG_s) largest values for optimal models for DAGs with average degree two.
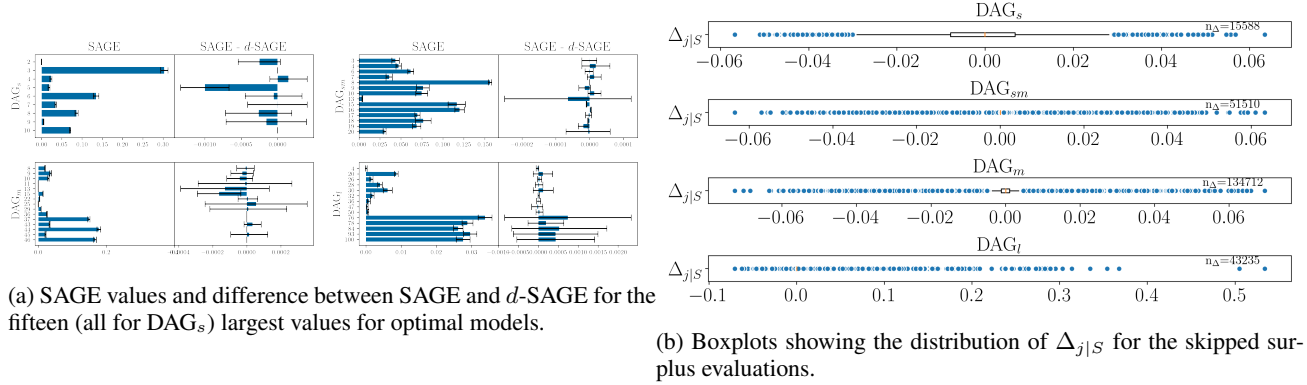
(a) SAGE values and difference between SAGE and d-SAGE for the fifteen (all for DAG$_s$) largest values for optimal models.

(b) Boxplots showing the distribution of $\Delta_{j|S}$ for the skipped surplus evaluations.

Figure 9: Results on the estimation quality for d-SAGE based on each DAG with average degree three and the LM.

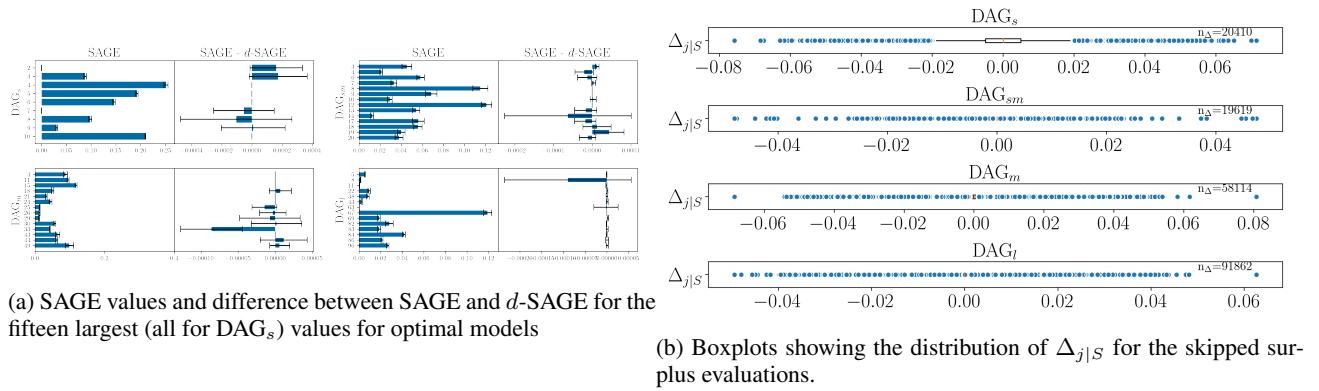(a) SAGE values and difference between SAGE and d-SAGE for the fifteen largest (all for DAG$_s$) values for optimal models

(b) Boxplots showing the distribution of $\Delta_{j|S}$ for the skipped surplus evaluations.

Figure 10: Results on the estimation quality for d-SAGE based on each DAG with average degree four and the LM.

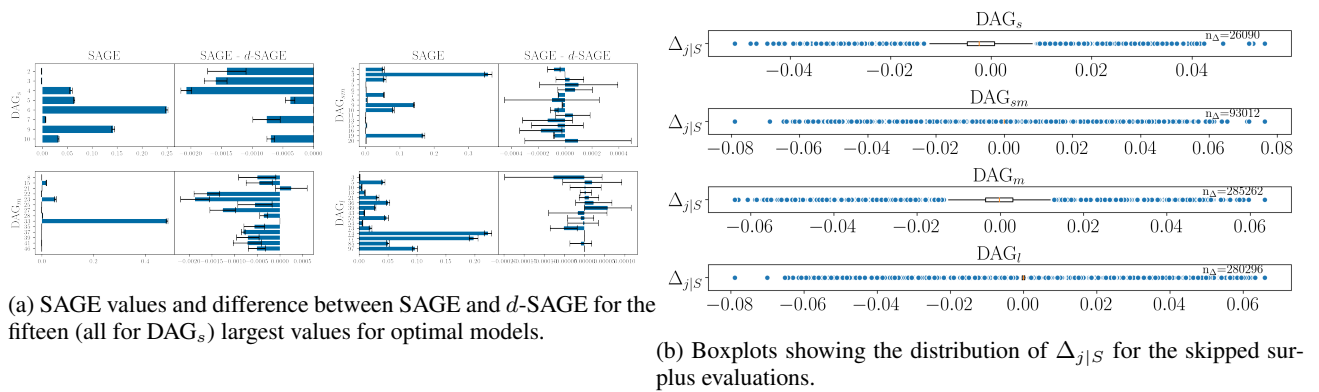(a) SAGE values and difference between SAGE and d-SAGE for the fifteen (all for DAG$_s$) largest values for optimal models.

(b) Boxplots showing the distribution of $\Delta_{j|S}$ for the skipped surplus evaluations.

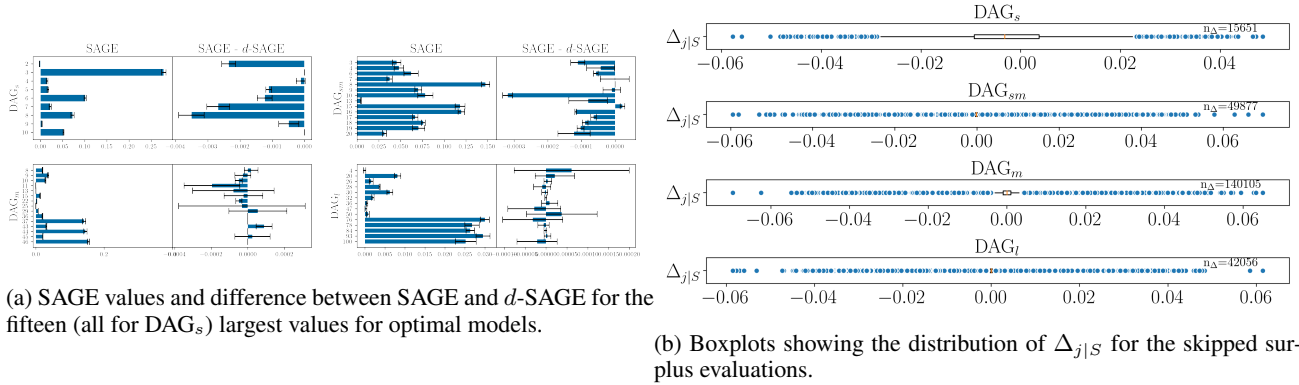Figure 11: Results on the estimation quality for d-SAGE based on each DAGs with average degree two and the RF.

(a) SAGE values and difference between SAGE and d-SAGE for the fifteen (all for DAG$_s$) largest values for optimal models.

(b) Boxplots showing the distribution of $\Delta_{j|S}$ for the skipped surplus evaluations.

Figure 12: Results on the estimation quality for d-SAGE based on each DAG with average degree three and the RF.

(a) SAGE values and difference between SAGE and d-SAGE for the fifteen (all for DAG$_s$) largest values for optimal models.

(b) Boxplots showing the distribution of $\Delta_{j|S}$ for the skipped surplus evaluations.
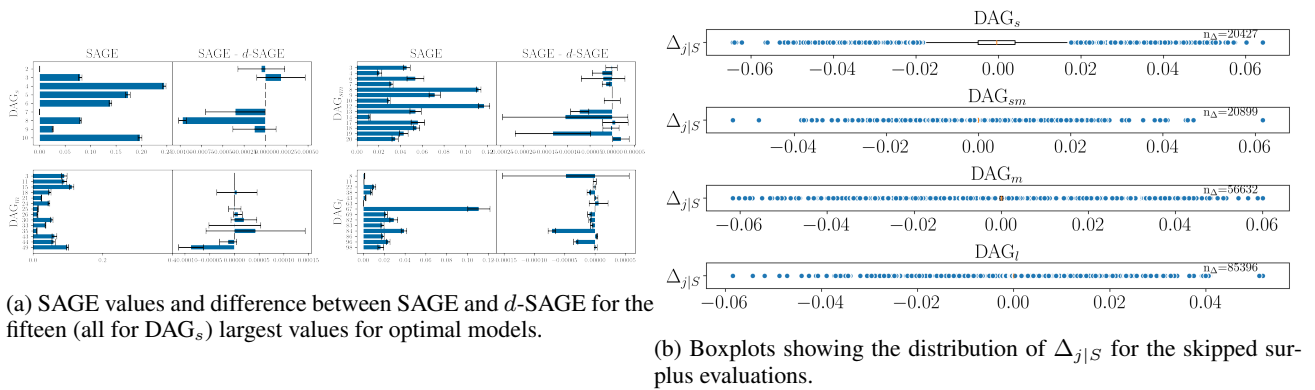
Figure 13: Results on the estimation quality for d-SAGE based on each DAG with average degree four and the RF.

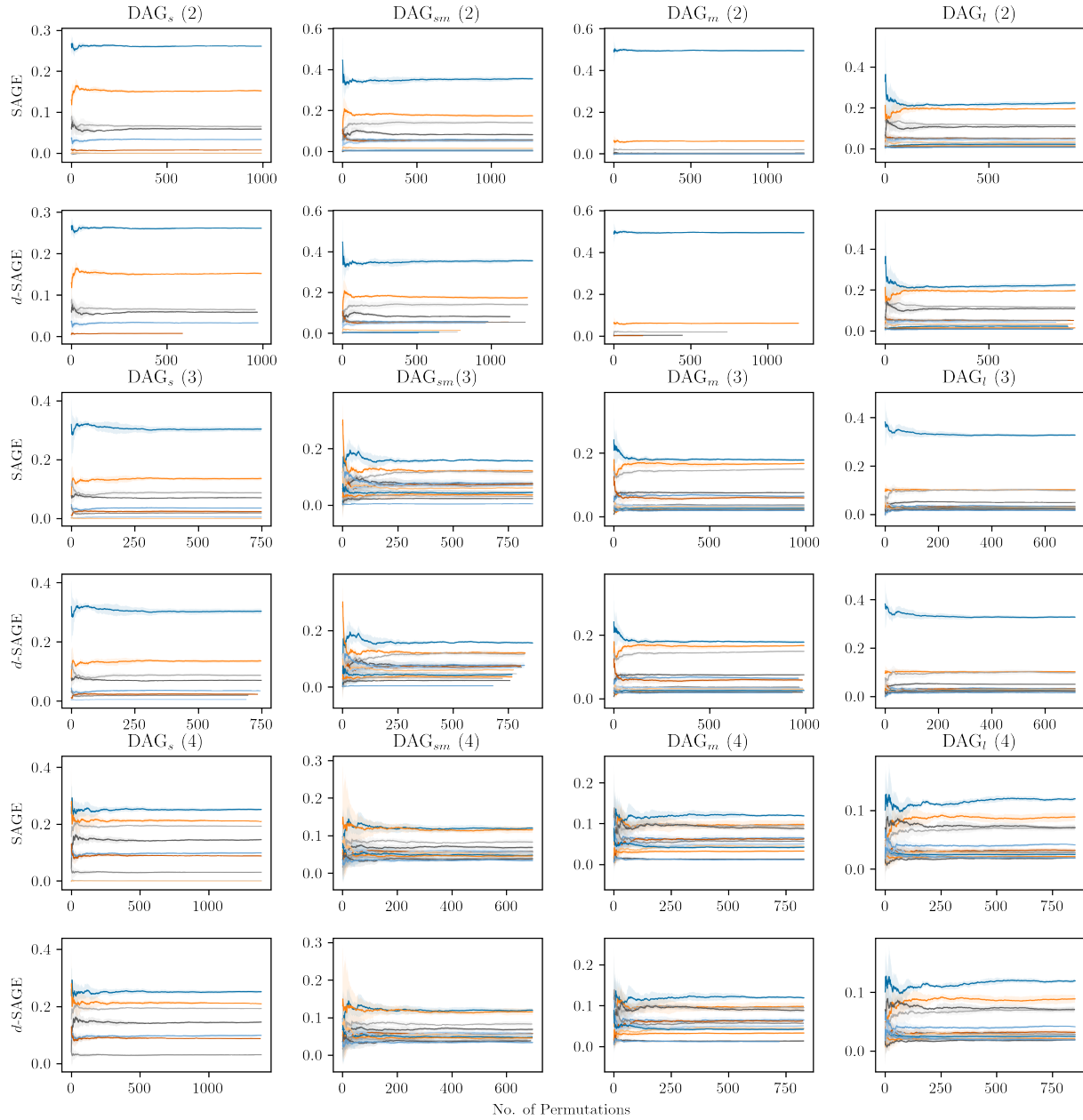# E   CONVERGENCE PLOTS

Figure 14: Convergence of largest fifteen SAGE and d-SAGE values for optimal models (LM) for every DAG (average adjacency degree). Each colour represents the same feature in SAGE and d-SAGE plots for a given graph (if present in both). Legend omitted for readability.

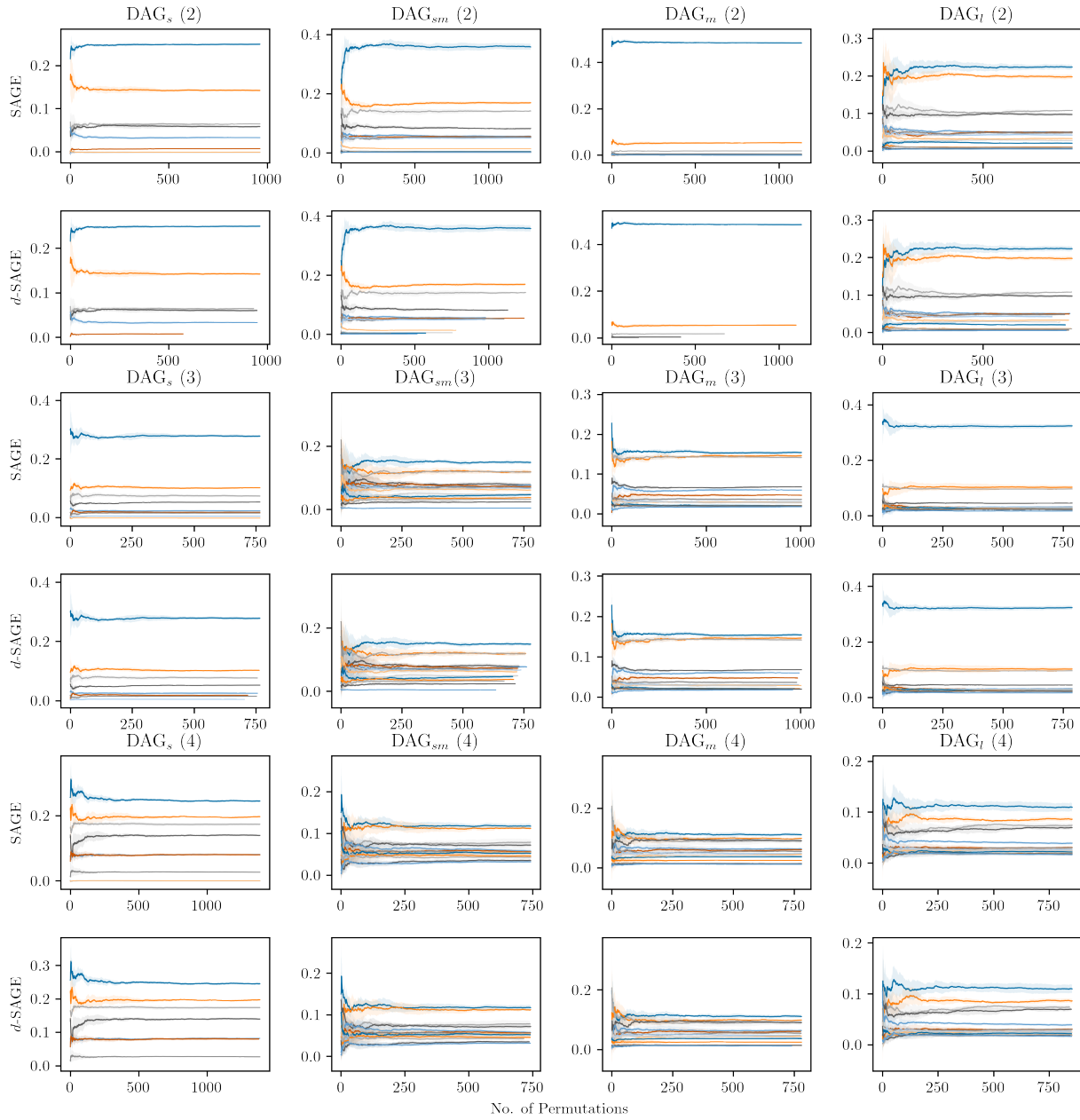**Christoph Luther*, Gunnar König*, Moritz Grosse-Wentrup**

Figure 15: Convergence of largest fifteen SAGE and d-SAGE values for random forest models (RF) for every DAG (average adjacency degree). Each colour represents the same feature in SAGE and d-SAGE plots for a given graph (if present in both). Legend omitted for readability.

Figure 16: Convergence of bottom fifteen SAGE and d-SAGE values for optimal models (LM) and random forest models (RF) for DAG_m and DAG_l (average adjacency degree). Each colour represents the same feature in SAGE and d-SAGE plots for a given graph (if present in both). Legend omitted for readability.
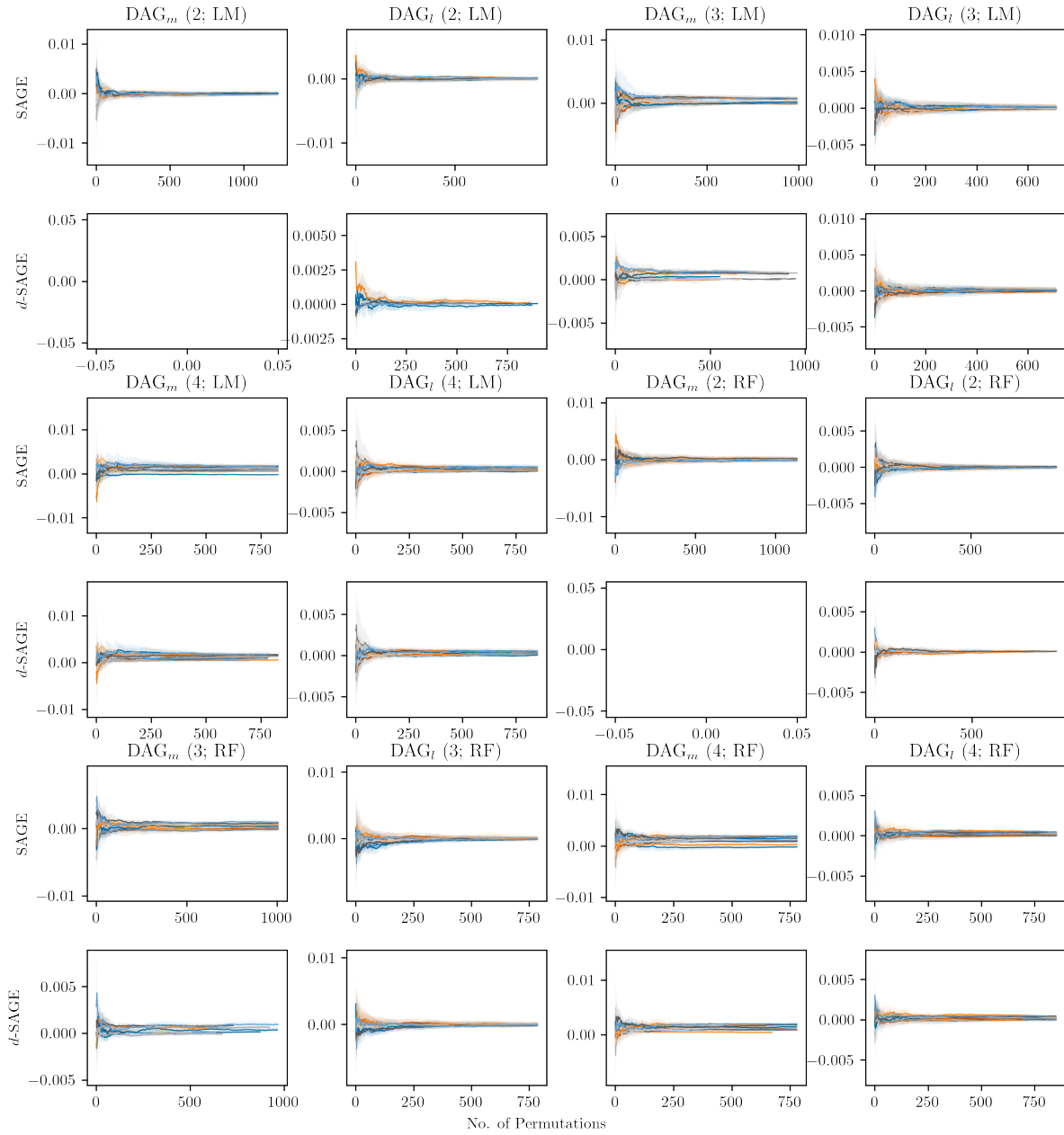
E.1 Convergence of SAGE Values

The approximation algorithm is designed such that convergence for all values is required to stop. Hence, some values are converged but still computed. However, the benefit of d-SAGE depends on the share of CIs and not the number of permutations required for convergence, and hence, even a fewer number of permutations would lead to a similar speedup. Missing lines in the convergence plots belong to conditionally independent features (given every sampled coalition), which highlights the ability of (d-)SAGE for post-hoc feature selection. An example of faster converging d-SAGE values is displayed by the comparison of SAGE and d-SAGE for $\mathrm{DAG}_m(2)$ in Figure 14, where the small values (slightly above zero) converge faster for d-SAGE.

F PARTIAL CORRELATION TESTS v d-SEPARATION QUERIES

To highlight the benefit of CSL over statistical independence tests, we compared the runtime of linear time d-separation queries (in graphs inferred by TABU) from the NetworkX package for Python (Hagberg et al., 2008) to that of partial correlation tests for linear Gaussian data from the Pingouin package (Vallat, 2018). Results are based on 100 permutations. Table 4 clearly shows that partial correlation tests are typically more accurate at the cost of much higher runtime in comparison to d-separation queries (+ graph learning).

Table 4: Partial correlation tests v d-separation queries based on $n = 10,000$ and 100 permutations; Graph learning based on TABU; ACC = Accuracy.

DATA	TIME (d-separation)	TIME (TABU)	TIME (CIs)	ACC (d-separation)	ACC (CIs)
DAG_s (2)	0.13s	0.06s	46.82s	1.000	1.000
DAG_{sm} (2)	0.39s	0.22s	166.75s	0.996	0.999
DAG_m (2)	1.95s	1.11s	1058.80s	1.000	1.000
DAG_l (2)	15.48s	12.02s	4344.81s	0.863	0.934
DAG_s (3)	0.13s	0.18s	47.15s	1.0	1.0
DAG_{sm} (3)	0.41s	0.71s	166.28s	0.996	0.992
DAG_m (3)	2.12s	2.51s	1089.00s	0.908	0.983
DAG_l (3)	16.85s	18.22s	4299.47s	0.857	0.941
DAG_s (4)	0.14s	0.09s	47.18s	1.0	0.998
DAG_{sm} (4)	0.42s	1.37s	163.48s	1.0	0.988
DAG_m (4)	2.33s	5.65s	1093.50s	0.845	0.940
DAG_l (4)	20.74s	39.86s	4312.16s	0.902	0.916