Prequential MDL for Causal Structure Learning with Neural Networks

Jörg Bornschein **Silvia Chiappa** **Alan Malek** **Rosemary Nan Ke**
DeepMind, London
{bornschein,csilvia,alanmalek,nke}@deepmind.com

Abstract

Learning the structure of Bayesian networks and causal relationships from observations is a common goal in several areas of science and technology. We show that the prequential minimum description length principle (MDL) can be used to derive a practical scoring function for Bayesian networks when flexible and overparametrized neural networks are used to model the conditional probability distributions between observed variables. MDL represents an embodiment of Occam's Razor and we obtain plausible and parsimonious graph structures without relying on sparsity inducing priors or other regularizers which must be tuned. Empirically we demonstrate competitive results on synthetic and real-world data. The score often recovers the correct structure even in the presence of strongly nonlinear relationships between variables; a scenario were prior approaches struggle and usually fail. Furthermore we discuss how the the prequential score relates to recent work that infers causal structure from the speed of adaptation when the observations come from a source undergoing distributional shift.

1 Introduction

Bayesian networks are a powerful probabilistic framework based on a graphical representation of statistical relationships between random variables. Inferring the Bayesian network structure that best represents a dataset not only allows to use the network to perform probabilistic, and possibly causal, reasoning but can also provide substantial illumination about the domain under consideration. This paper considers the problem of structure learning in settings in which modern, possibly overparametrized, neural networks are used to model the Bayesian network conditional distributions.

Recent effort on structure learning with modern neural networks has focused on improving scalability w.r.t. the number of variables by relaxing the discrete search problem over structures to a continuous optimization problem [Zheng et al., 2018, Yu et al., 2019, Zheng et al., 2020]. Whilst enabling the use of large structures, the regularized maximum-likelihood score used to rank structures makes these methods prone to overfitting random fluctuations and sensitive to the regularizer.

We propose an approach to ranking structures based on the minimum description length (MDL) principle. Motivated by fundamental ideas in data-compression, information theory, as well as philosophical notions like Occam's razor, the MDL principle posits that models which lead to compact and parsimonious descriptions of the data are more plausible. In the context of structure learning, this criterion induces a preference for more compact and simpler structures as more plausible explanations of the data generation mechanism. Many traditional scores, such as AIC [Akaike, 1973], BIC [Schwarz, 1978], marginal likelihood [Heckerman et al., 1995], and more recent scores [Silander et al., 2018], can be seen as implementations of the MDL principle. However, some of these scores are approximations that, especially when applied to overparametrized neural networks, can lead to poor empirical performance [Silander et al., 2008]. In addition, many such scores can only be

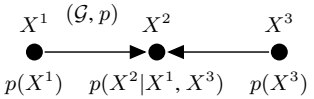
35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

applied to simple model families, and therefore might not be suitable to modelling complex nonlinear relationships in the data.

We propose using the prequential plug-in score, which evaluates conditional distributions by their sequential predictive performance, and gives an approach to ranking structures that balances fit to the data with overfitting without the need for an explicit sparsity inducing priors or regularizer. We provide a specific method for implementing the score with modern neural networks. We demonstrate on both artificial and real-world data that our method often retrieves the data-generating structure and is robust to neural network hyperparameter selection.

2 Structure Learning with Prequential MDL

Our approach to learning the structure of a Bayesian network is to rank structures by measuring the complexity of the associated *conditional probability distributions* (CPDs) through the *minimum description length* (MDL) principle [Grünwald, 2004, Grünwald, 2007]. In particular, we propose the use of the prequential plug-in score and an implementation of this score for the case in which modern neural networks are used to model the CPDs. Before describing our method in detail, we give an introduction into Bayesian networks and structure learning with MDL.



Bayesian Networks (BNs). A *Bayesian network* [Pearl, 1988, 2000, Cowell et al., 2007, Koller and Friedman, 2009] is a *directed acyclic graph* (DAG) $\mathcal{G}$ whose nodes $X^1, \ldots, X^D$ represent random variables and links express statistical dependencies among them. Each node X^d is associated with CPD $p(X^d \,|\, \text{pa}(X^d))$, where $\text{pa}(X^d)$ denote the *parents* of X^d, namely the nodes with a link into X^d. The joint distribution of all nodes is given by the product of all CPDs, i.e. $p(X^1, \ldots, X^D \,|\, \mathcal{G}) = \prod_{d=1}^{D} p(X^d \,|\, \text{pa}(X^d))$. We make the common assumption that each CPD $p(X^d \,|\, \text{pa}(X^d))$ is parametrized by a separate set of parameters θ^d. The set of BNs that encode the same set of conditional independence assumptions forms a *Markov equivalence class*. A BN can be given a causal semantic by interpreting a link between two nodes as expressing causal rather than statistical dependence.

Score-based Structure Learning with MDL. Let p^* be a joint distribution over D random variables with joint domain $\mathcal{X}$, and let $\mathcal{D} = \{x_i := (x_i^1, \ldots x_i^D)\}_{i=1}^n$ be a dataset of n i.i.d. samples from p^*. The goal of *structure learning* is to infer the DAG $\mathcal{G}$, referred to as *structure*, or the Markov equivalence class that best represents $\mathcal{D}$.

We focus on score-based approaches that rank structures w.r.t. some scoring metric [Heckerman, 1999, Drton and Maathuis, 2017, Glymour et al., 2019]. A naïve score is the *maximum log-likelihood* $\max_{\theta \in \Theta} \log p(\mathcal{D} \,|\, \theta, \mathcal{G})$, which ignores model complexity and results in a preference for dense and complex structures that do not generalize well. A simple approach to account for model complexity is to add to the log-likelihood a regularization term that can depend on the dimension of the parameters $\dim(\theta)$ and on the size of the dataset n—the two most common penalty terms are $-\dim(\theta)$ (AIC) and $-0.5 \log(n) \dim(\theta)$ (BIC). A more sophisticated approach is to instead integrate out θ, which gives the *log-marginal likelihood* $\log \int_\Theta p(\mathcal{D} \,|\, \theta, \mathcal{G}) p(\theta \,|\, \mathcal{G}) d\theta$. Both these approaches can be described within the unifying framework of MDL.

The MDL framework is based on the principle that the model that yields the shortest description of the data is also the most plausible. In the context of structure learning, the MDL principle prescribes that we pick the model class $\mathcal{M}_\mathcal{G} = \{p(\cdot|\theta, \mathcal{G}) : \theta \in \Theta\}$ from the set $\{\mathcal{M}_\mathcal{G} : \mathcal{G} \in G\}$ which leads to the most compact representation of the dataset $\mathcal{D}$ with a code derived from $\mathcal{M}_\mathcal{G}$. Considering the one-to-one relationship between code-lengths and probability distributions this means selecting the model class under which the data has the highest likelihood. From this perspective maximum log-likelihood $\log p(\mathcal{D}|\hat{\theta}^{\text{MLE}}(\mathcal{D}), \mathcal{G})$, where $\hat{\theta}^{\text{MLE}}(\mathcal{D}) := \arg\max_\theta \log\, p(\mathcal{D}|\theta, \mathcal{G})$ alone cannot be the basis for a code because it does not normalize to one, it is thus not a distribution over $\mathcal{D}$ which precludes the existence of a code with the corresponding code-length. Maximum log-likelihood can only be used as a basis for a code if $\hat{\theta}^{\text{MLE}}$ is known a-priori.

Instead, the MDL literature suggests using the *universal distribution* $\overline{p}(\cdot|\mathcal{G})$ of $\mathcal{M}_\mathcal{G}$. It is defined to be the distribution that as closely as possible tracks the code-length of the maximum log-likelihood

solution within $\mathcal{M}_\mathcal{G}$ on all possible data $\mathcal{Z} \in \mathcal{X}^n$:

$$\overline{p}(\cdot \mid \mathcal{G}) := \arg\min_{q} \max_{\mathcal{Z} \in \mathcal{X}^n} \left(-\log q(\mathcal{Z}) - \left(-\log p(\mathcal{Z} \mid \hat{\theta}^{\text{MLE}}(\mathcal{Z}), \mathcal{G}) \right) \right). \tag{1}$$

The MDL structure selection rule is

$$\mathcal{G}_{\text{MDL}}(\mathcal{D}) := \arg\max_{\mathcal{G} \in G} \log \overline{p}(\mathcal{D} \mid \mathcal{G}) \tag{2}$$

because the model class with the most compact representation corresponds to the model class with highest $\overline{p}(\mathcal{D}|\mathcal{G})$ (see Grünwald and Roos [2019]). The universal distribution is a probability distribution that, in some sense, summarizes how well $\mathcal{M}_\mathcal{G}$ fits data: it places large probability on $\mathcal{D}$ only if there is a distribution $p(\cdot|\theta, \mathcal{G}) \in \mathcal{M}_\mathcal{G}$ that places large probability on $\mathcal{D}$. The requirement that $\overline{p}(\cdot|\mathcal{G})$ must normalize to one naturally induces complexity regularization. For a model class that is very expressive (e.g. $\mathcal{G}$ is fully connected) $\log\, p(\mathcal{Z}|\hat{\theta}^{\text{MLE}}(\mathcal{Z}), \mathcal{G})$ is large for many values of $\mathcal{Z}$, and therefore the universal distribution must spread its mass across much of $\mathcal{X}^n$. This implies that $\log \overline{p}(\mathcal{D}|\mathcal{G})$ for the observed dataset $\mathcal{D}$ cannot be high. On the other hand, for a model class that is not as expressive (e.g, $\mathcal{G}$ includes only few links), $\log p(\mathcal{Z}|\hat{\theta}^{\text{MLE}}(\mathcal{Z}), \mathcal{G})$ is large only for data that are compatible with its graph structure and the universal distribution can have much higher log-likelihood on such data. Using the universal distribution for structure selection leads to favoring structures that have expressiveness for data similar to $\mathcal{D}$ but do not waste expressiveness on dissimilar data.

2.1 Prequential Plug-in Score

Equation 1 provides a prescriptive definition of the universal distribution required by Eq. (2) to compute the score $\log \overline{p}(\mathcal{D} \mid \mathcal{G})$. Several constructive definitions have been proposed to closely approximate Eq. (1). These are, following the MDL literature, also referred to as universal distributions. We propose approximating $\log \overline{p}(\mathcal{D} \mid \mathcal{G})$ with the *prequential plug-in score* from the prequential plug-in universal distribution, defined as

$$\log p_{\text{preq}}(\mathcal{D} \mid \mathcal{G}) := \log \prod_{i=1}^{n} p(x_i \mid \hat{\theta}(x_{<i}), \mathcal{G}),$$

where $\hat{\theta}(x_{<i}) \in \Theta$ indicates a consistent parameters estimate given $x_{<i} := (x_1, \ldots, x_{i-1})$.

The prequential plug-in score is based on the idea of evaluating a model by its sequential predictive performance and therefore by its generalization capabilities [Dawid and Vovk, 1999]. The prequential approach in the context of MDL has been proposed by Grünwald [2004], Poland and Hutter [2005], Grünwald [2007]. There are advantages in using the prequential plug-in score w.r.t. other scores derived from popular and well-studied universal distributions, such as the log-marginal likelihood (also called *Bayesian score*) $\log p_{\text{Bayes}}(\mathcal{D} \mid \mathcal{G}) := \log \int_\Theta p(\mathcal{D} \mid \theta, \mathcal{G}) p(\theta \mid \mathcal{G}) d\theta$, the *log-normalized maximum likelihood* $\log p_{\text{NML}}(\mathcal{D} \mid \mathcal{G}) := \log p(\mathcal{D} \mid \hat{\theta}^{\text{MLE}}(\mathcal{D}), \mathcal{G}) - \log \int_{\mathcal{Z} \in \mathcal{X}^n} p(\mathcal{Z} \mid \hat{\theta}^{\text{MLE}}(\mathcal{Z}), \mathcal{G}) d\mathcal{Z}$ [Rissanen, 1996], or other approximations. The prequential plug-in score is better suited to neural networks than the Bayesian score, as it does not require integration over the parameters. Whilst it might appear that these two scores imply different preferences for model selection, they are equivalent for several, and often natural, choices of $p(\theta)$ and $\hat{\theta}(x_{<i})$ (see Sect. 3.1). The log-normalized maximum likelihood is widely used in theoretical treatments of MDL. However, the normalization term over all possible observed data makes this score often intractable or not defined. The well-known AIC/BIC scores can also be cast as approximations to $\log \overline{p}(\mathcal{D}|\mathcal{G})$ [Lam and Bacchus, 1994], but both are known to have poor empirical performance [Silander et al., 2008] as they can be quite loose.

Decomposability over CPDs. The assumption that each CPD is modelled by a separate set of parameters enables us to write the prequential plug-in score as

$$\log p_{\text{preq}}(\mathcal{D} \mid \mathcal{G}) = \sum_{d=1}^{D} \sum_{i=1}^{n} \log p(x_i^d \mid \text{pa}(x_i^d), \hat{\theta}^d(x_{<i})), \tag{3}$$

where $\text{pa}(x_i^d)$ indicates the observed values of $\text{pa}(X^d)$ for observation x_i and $\hat{\theta}^d(x_{<i})$ the parameters learned using $\{(x_j^d, \text{pa}(x_j^d))\}_{j=1}^{i-1}$. This decomposition allows a computationally more efficient ranking of structures—for example there are 29,280 DAGs with 5 nodes but only 80 underlying CPDs.

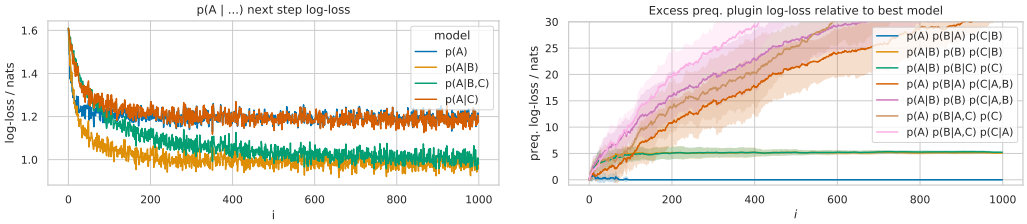


Figure 1: Prequential scoring with tabular CPDs on synthetic data generated using $\mathcal{G}^* = A \to B \to C$. **(a)**: Next-step log-loss for variable A given all possible combinations of the other variables as parents. **(b)**: Excess prequential log-loss for all possible DAGs relative to $\mathcal{G}^*$. Uncertainty bands show standard deviation over 1,000 permutations of the data.

2.2 Implementation of the Prequential Plug-In Score with Neural Networks

The computation of the prequential plug-in score (3) requires evaluating $\log p(x_i^d|\text{pa}(x_i^d), \hat{\theta}^d(x_{<i}))$ $\forall i = 1, \ldots, n$. When the CPDs are modelled by neural networks, we must train the networks to convergence on many subsets of $x_{<i}$ using a stochastic gradient-based optimizer. Modern, usually overparametrized, neural networks 1) may overfit severely for small i, and 2) training them from scratch for each i can be computationally infeasible, while at the same time it is difficult to use them in online settings where the training set constantly grows (a topic of active research). For example, using the model parameters from training on $x_{<i}$ as a starting point for learning model parameters from $x_{<i+j}$ often leads to significantly reduced generalization [Ash and Adams, 2020].

To overcome the second obstacle, we propose to use the approach described by Blier and Ollivier [2018], Bornschein et al. [2020], specifically to choose a set of increasing split points $\{s_k\}_{k=1}^K$, with $s_k \in [2, \ldots, n]$, $s_K{=}n{+}1$ and compute the score of the data between two split points $x_{s_k}^d, \ldots, x_{s_{k+1}-1}^d$ with a neural network trained from scratch on $x_{<s_k}$ to convergence, which corresponds to the approximation

$$\sum_{i=1}^{n} \log p(x_i^d|\text{pa}(x_i^d), \hat{\theta}(x_{<i})) \approx \sum_{k=1}^{K-1} \sum_{j=s_k}^{s_{k+1}-1} \log p(x_j^d|\text{pa}(x_j^d), \hat{\theta}^d(x_{<s_k})).$$

In the experiments, we chose the split points to be exponentially spaced and performed $K{-}1$ independent training and evaluation runs, usually in parallel.

To overcome the first obstacle, we propose to use a simple confidence calibration approach introduced by Guo et al. [2017] to independently calibrate every CPD on every training run. First, consider a network with a softmax output layer for categorical prediction. Conceptually, we could perform post-calibration by first training the network to convergence and then, with all parameters frozen, replacing the output layer softmax(h) with the calibrated output layer softmax$(\beta \cdot h)$, where β is a scalar parameter chosen to minimize the loss on validation data. In practice, we optimize β by gradient descent in parallel with the other model parameters. We alternate computing ten gradient steps for θ, calculated from the training set and using the uncalibrated network (with final layer softmax(h)), with a single gradient step on β, calculated from the validation set using the calibrated network (with final layer softmax$(\beta \cdot h)$). This simple calibration procedure has proven to be surprisingly effective at avoiding overfitting when training large neural networks on small datasets [Bornschein et al., 2020]. To the best of our knowledge, an analogous method to calibrate continuous-valued neural network outputs does not exist. Thus, we approximate networks for a continuous random variable by networks for a categorical random variable on the quantized values.

3 Experiments

We demonstrate the effectiveness of our approach, which we refer to as *prequential scoring*, on a variety of synthetic datasets and on a real-world dataset.

3.1 Prequential Scoring with Tabular CPDs

Whilst our motivation for introducing the prequential plug-in score is its suitability when neural networks are used to model the CPDs, we first build intuition by considering the case of categorical data with conditional probability tables, i.e. with $p(X^d = k \,|\, \text{pa}(X^d) = l, \theta^d) = \theta^d_{k,l}$, which does not require approximating the score.

We generated synthetic data $\mathcal{D} = \{x_i\}_{i=1}^n$ by using the DAG $\mathcal{G}^* = A \to B \to C$ with each variable taking five possible values, and by drawing the parameters for each ground-truth CPD and value of parents from a Dirichlet distribution with $\alpha^* = 1$. We then computed the *next-step log-loss* $-\log\, p(x_i^d \,|\, \text{pa}(x_i^d), \hat{\theta}^d(x_{<i}))$, $\forall i = 1, \ldots, n$ and $\forall d = 1, \ldots, D$ for all possible parents sets $\text{pa}(x_i^d)$, using the $\alpha = 0.5$-regularized MLE estimator $\hat{\theta}^d_{k,l}(x_{<i}) = {}^{N^d_{k,l} + \alpha}/_{\sum_m (N^d_{m,l} + \alpha)}$, where $N^d_{k,l}$ denotes the number of times that, in $x_{<i}$, X^d and $\text{pa}(X^d)$ take values k and l respectively.

Fig. 1(a) displays the next-step log-loss $\forall i = 1, \ldots, n$ for variable A given all possible parents sets, averaged over $1,000$ different permutations of the datapoints to make the plot less noisy. This average can be seen as an approximation to the *generalization log-loss* $-\mathbb{E}_{\tilde{x} \sim p^*} \log p(\tilde{x}^d \,|\, \text{pa}(\tilde{x}^d), \hat{\theta}^d(x_{<i}))$ (i.e. the negative log-likelihood on held-out data). The plot shows that conditioning A on B generally gives the best result. Additionally conditioning on C reaches the same performance when sufficient training data is available, but results in worse performance in the small-data regime. In other words, if we were to train on e.g. $1,000$ datapoints and use the generalization log-loss to select a model, we would not be able to reliably select $p(A \,|\, B)$ over $p(A \,|\, B, C)$. The generalization log-loss does not account for model complexity and might lead us to select models that are more complex than necessary. However, with only 100 training examples this loss does give a clear signal that we should prefer $p(A \,|\, B)$, because the over conditioned $p(A \,|\, B, C)$ has significantly worse performance in the small-data regime. These observations suggest that we should select models in the small-data regime, but finding the right regime could be difficult. Fig. 1(a) indicates that the regime is between ≈ 50 and ≈ 200 for $p(A|\cdot)$, but that is not known a-priori. Additionally, the optimal regime to perform model selection for e.g. $p(B|\cdot)$ might be different. By summing the next-step log-losses up to i, the *prequential log-loss* $-\log\, p_{\text{preq}}(x_{\leq i} \,|\, \mathcal{G})$ accumulates and persists the differences from the small-data regimes. Fig. 1(b) shows the prequential log-loss $\forall i = 1, \ldots, n$ for all DAGs relative to the best one. The ground-truth DAG $A \to B \to C$ is identified as most plausible, followed by $A \leftarrow B \to C$ and $A \leftarrow B \leftarrow C$; all three are in the same Markov equivalence class. The fully connected DAGs, which reach about same next-step log-loss after being trained on $\approx 1,000$ datapoints, accumulate more than 50 nats additional loss compared to $A \to B \to C$. Notice that with our choice for the parameters estimator the prequential log-loss becomes equivalent to the log-marginal likelihood with $\theta^1, \ldots, \theta^D$ independent random variables with Dirichlet distributions (see Appendix F).

3.2 Prequential Scoring with Neural Networks

We evaluate prequential scoring with neural networks on several synthetic datasets and on the Sachs real-world dataset [Sachs et al., 2005]. We primarily compare with the DAG-GNN method introduced by Yu et al. [2019], which represents one of the more competitive modern methods to structure learning with neural networks. Additionally we compare with the PC algorithm, a constraint-based method using linear-regression based Pearson correlation as independence test [Spirtes et al., 2000][1].

Architecture and Hyperparameters. In all experiments, we modelled the CPDs with neural networks consisting of 3 fully connected layers of 512 hidden units, ReLU activation functions, and dropout with probability of 0.5 on all the hidden units. We applied a random Fourier transformation to the data obtained by sampling 512 random frequencies from a Gaussian distribution $\mathcal{N}(0, 10^2)$, as this has been shown to improve the performance in neural networks with low-dimensional inputs [Tancik et al., 2020]. To use the softmax confidence calibration described in Sect. 2.2, we mapped the predicted values into the interval $[-1, 1]$ with $\tanh$ and then discretized them according to a uniform 128-values grid. For optimization, we used Adam [Kingma and Ba, 2015] with a batch size of 128; for each point s_k, we independently choose the learning rate to be either $1{\cdot}10^{-4}$ or $3{\cdot}10^{-4}$ depending on which one resulted in a lower calibrated log-loss. We performed 25,000 gradient steps but used early-stopping if the calibrated log-loss increased, which led to considerable compute savings as many

[1] Constraint-based approaches use independence tests to infer the existence of links between pairs of variables and require faithfulness—see Appendix B for a discussion on this assumption.

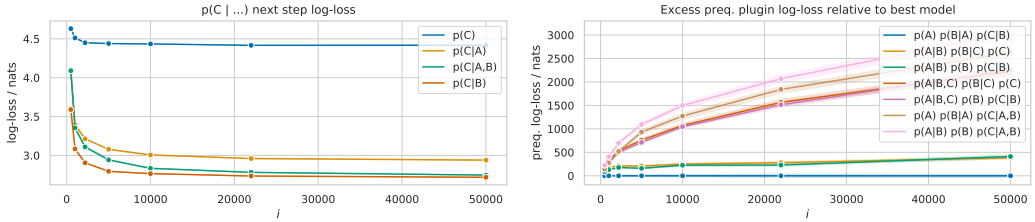


Figure 2: Prequential scoring with neural networks on synthetic data generated from $\mathcal{G}^* = A \to B \to C$. **(a):** Next-step log-loss for variable C given all possible combinations of the other variables as parents. **(b):** Excess prequential log-loss for all possible DAGs relative to $\mathcal{G}^*$. Uncertainty bands show standard deviation over 5 random seeds.

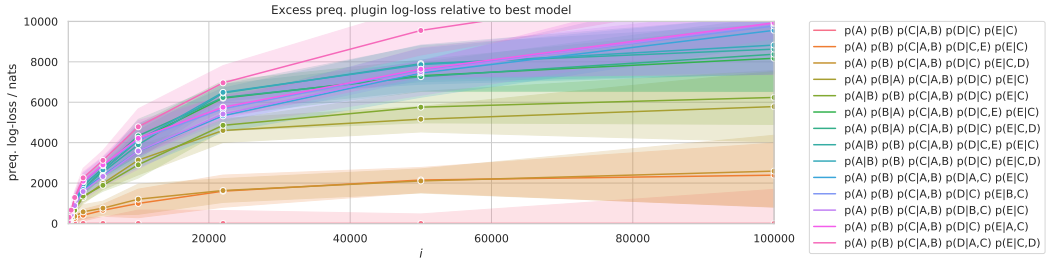


Figure 3: Prequential scoring with neural networks on synthetic data generated from $\mathcal{G}^* = A \to C \leftarrow B, C \to D, C \to E$. Excess prequential log-loss for the 14 most likely DAGs relative to the the most likely one. Uncertainty bands show standard deviation over 3 random seeds for neural network initialization and mini-batching.

training runs on small subsets of the data converge after only a few hundred or thousand gradient steps. All experiments were carried out on CPUs without accelerators as the networks were relatively small, and a typical run to convergence took between 5 and 15 minutes. We performed such training runs for each potential CPD and for $K \approx 6$ log-equidistantly spaced split-points s_k. For example, with 5 observed variables we ran 960 independent training runs corresponding to all combinations of the 80 CPDs, 6 split-points, and 2 learning-rates. We collected and accumulated the results and performed an exhaustive search over all DAGs to find the most likely one given the data. Changing the depth and width of the networks did not impact the rankings of the structures, provided that the models had sufficient capacity. Similarly, changing the optimizer to RMSprop [Graves, 2014] or Momentum SGD [Qian, 1999] had minimal effect. This robustness, together with the fact that prequential scoring has no hyperparameter for regulating the sparsity of the inferred graphs, allowed us to use the same hyperparameter settings throughout all the experiments.

Case Studies for 3 and 5-Node DAGs. We first evaluated prequential scoring on data generated from hand-crafted generation mechanisms. Below, we describe the results from two mechanisms (a third mechanism is reported in Appendix C). Fig. 2 show the results obtained on data generated from the DAG $\mathcal{G}^* = A \to B \to C$ with $A \sim \mathcal{N}(0, 1)$, $B = \sin(A + \epsilon_B)$, $C = \sin(B + \epsilon_C)$, and $\epsilon_B, \epsilon_C \sim \mathcal{N}(0, 0.1^2)$. We observe the same scaling behaviour of Sect. 3.1 for both the next-step log-loss and the prequential log-loss, and the retrieval of $\mathcal{G}^*$ with an almost 500 nat margin. Fig. 3 shows the results obtained on data generated from $\mathcal{G}^* = A \to C \leftarrow B, C \to D, C \to E$ with $A \sim \mathcal{N}(0, 1)$, $B \sim \mathcal{N}(0, 1)$, $C = \sin(2AB + \epsilon_C)$, $D = \sin(C) + \epsilon_D$, $E = \sin(3C + \epsilon_E)$, and $\epsilon_C, \epsilon_D, \epsilon_E \sim \mathcal{N}(0, 0.1^2)$. As above, we observe that the next-step log-loss accumulated on small and medium-sized subsets of the dataset (smaller than roughly 20,000) is crucial to getting a discerning signal for DAG selection. Once again, prequential scoring identifies $\mathcal{G}^*$ by a significant margin of 2,000 nats.

Data from Yu et al. [2019]. To test prequential scoring on a larger gamut of distributions, we turned to the data generating mechanism introduced by Yu et al. [2019] for validating DAG-GNN. Specifically, we generated datasets by the fixed points X of the equations *a)* $X = A^\top \cos(X + \mathbf{1}) + Z$

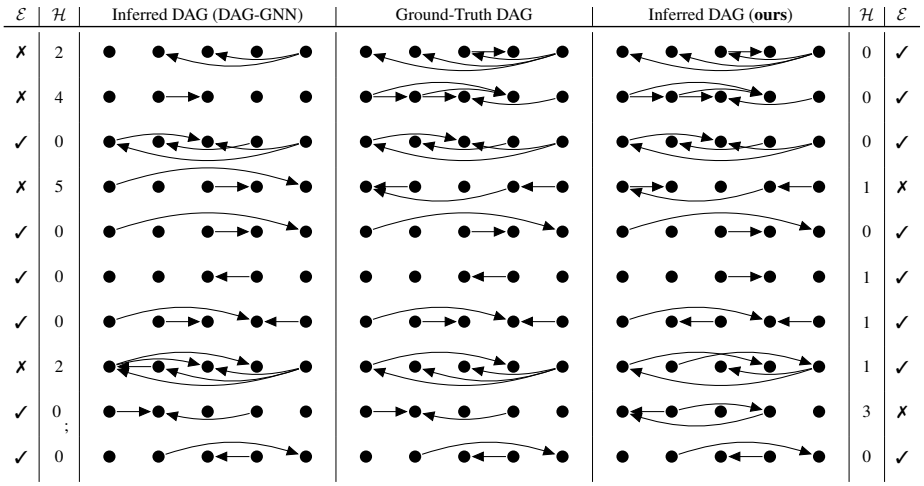


Table 1: Results on nonlinearities from Yu et al. [2019]. We list the ground-truth DAGs and the DAGs inferred by DAG-GNN and by prequential scoring. We also report whether the inferred DAGs are in the Markov equivalence class $\mathcal{E}$ of the ground-truth DAGs and the structural Hamming distance $\mathcal{H}$.

and *b)* $X = 2\sin(A^\top(X + 0.5 \cdot \mathbf{1})) + A^\top(X + 0.5 \cdot \mathbf{1}) + Z$, where $\mathbf{1}$ denotes the all-ones vector and Z a standard normal variable. Due to the limited scalability of the exhaustive DAG search step required in prequential scoring, we restricted ourselves to 5×5 adjacency matrices A. We generated 5 datasets using *a)* and 5 datasets using *b)*.

The structures inferred by prequential scoring and DAG-GNN are given in Table 1. Prequential scoring tends to recover DAGs with lower structural Hamming distance $\mathcal{H}$ to the ground-truth DAG (0.7 vs 1.3 on average) and that are more frequently in the same Markov equivalence class $\mathcal{E}$ (8 of 10 vs 6 of 10). The PC algorithm also performs reasonably well by inferring DAGs within the correct Markov equivalence class in 5 cases (details are given in Appendix D). Inspection of the data revealed that the vast majority of relationships between the observed variables can be well approximated with simple linear functions; as such, it is not surprising that the PC algorithm performs well even though it uses linear-regression based Pearson correlation as independence test.

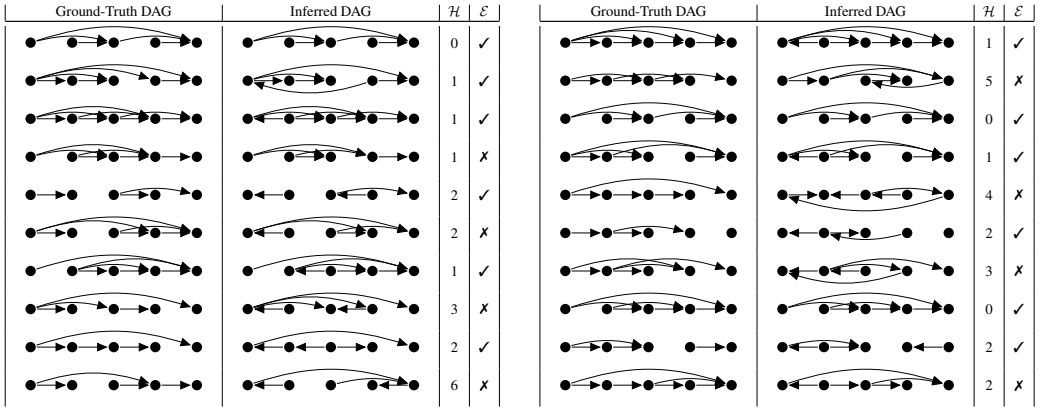


Table 2: Results obtained with prequential scoring on 20 randomly generated five-node graphs, with CPDs corresponding to compositions of polynomials and trigonometric functions (see Table 3 in Appendix E). We list the ground-truth DAGs, the inferred DAGs, the structural Hamming distance $\mathcal{H}$, and whether the inferred DAGs are in the Markov equivalence class $\mathcal{E}$ of the ground-truth DAGs.

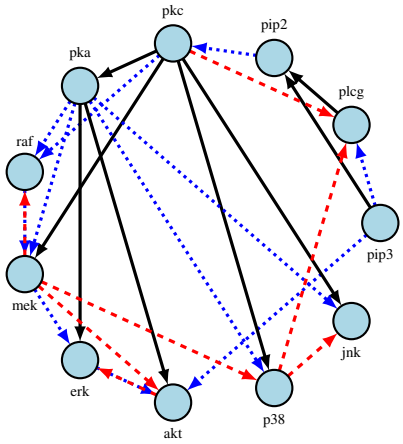
Complex Nonlinearities from Polynomials and Trigonometric Functions. To investigate the performance of prequential scoring for setting in which the CPDs need to model highly nonlinear relationships, we created a potpourri of synthetic data by first sampling random 5-node DAGs using

the GNP algorithm [Batagelj and Brandes, 2005] with link probability 0.25 and then annotating the links with random compound functions of polynomials, trigonometric functions, reciprocals and random noise. We created a set of 20 such data generation mechanisms which are listed in Table 3 of Appendix E. For example, a typical generation mechanism is $A = \sin(30\,\epsilon_A)$, $B = \sin(2A) + \epsilon_B$, $C = \sin(B^3 - A + \epsilon_C)$, $D = (C + \epsilon_D)^3$, $E = sgn(A)(|A| + 0.1)^{-1} + \epsilon_E$, with $\epsilon_A, \epsilon_B, \epsilon_C, \epsilon_D, \epsilon_E \sim \mathcal{N}(0, 0.1^2)$.

In Table 2 we list the ground-truth DAGs, the DAGs inferred by prequential scoring, the structural Hamming distance $\mathcal{H}$, and whether the inferred DAGs are in the Markov equivalence class $\mathcal{E}$ of the ground-truth DAGs. We observe that prequential scoring infers DAGs with an average $\mathcal{H}$ of 1.9 and recovers a member of $\mathcal{E}$ in 12 of the 20 cases.

We applied DAG-GNN and PC to the same 20 data datasets and were not able to obtain reliable and reproducible results. For DAG-GNN with default hyperparameters, the inferred structure varied significantly for different initial seeds and different sizes of the dataset. We also noticed a high sensitivity to the sparsity regularization hyperparameter. For the constraint-based method the results appear random. We did however not expect reliable results because the independence test is not designed to work on strongly nonlinear data.

Protein Signaling Network. As real-world dataset we considered the Sachs dataset for the discovery of a protein signaling network [Sachs et al., 2005], a benchmark dataset for structure learning with experimental annotations accepted by the biology research community. The data contains continuous measurements of expression levels of multiple phosphorylated proteins and phospholipid components in human immune system cells, and the network provides the ordering of the connections between pathway components. Based on $n = 7,466$ samples of $D = 11$ cell types, Sachs et al. [2005] estimated 20 links in the graph. In addition to observational samples, the dataset contains interventional samples obtained by activating or inhibiting expression levels at particular nodes. We handled interventional samples following the principles behind intervention in causal Bayesian networks [Pearl, 2000, Pearl et al., 2016]: if sample x_j^d was marked as the result of an intervention, we did not consider it for the learning and evaluation of the d-th CPD (see Appendix G for a description and a visualization of the effect that interventional data can have on prequential scoring).



We computed the prequential plug-in score from three runs with different random seeds for neural network initialization and mini-batching. For scalability reasons, we were only able to consider CPDs with at most 4 parents, and we used a heuristic hill-climbing method [Heckerman et al., 1995] to search the space of DAGs instead of an exhaustive search.

	$\mathcal{H}$	# links
NOTEARS	22	16
DAG-GNN	19	18
prequential scoring	16	15
prequential scoring (PWA)	18.4	16.8

Above, we show the DAG $\mathcal{G}^*$ that has been accepted by the biology research community as the best known solution overlayed with the DAG inferred by prequential scoring. Shared links are solid in black; otherwise, links discovered by prequential scoring are dotted blue and the DAG links only in $\mathcal{G}^*$ are dashed in red. The table on the left reports the structural Hamming distance $\mathcal{H}$ and the number of links found by prequential scoring, DAG-GNN and NOTEARS [Zheng et al., 2018]. Prequential scoring performs favourably and finds a graph with lower structural Hamming distance $\mathcal{H}$ to the ground-truth compared to DAG-GNN and NOTEARS.

However, prequential scoring identified a number of DAGs with good prequential plug-in scores and partially overlapping uncertainty bands. Using Bayes rule we can approximate the posterior $p(\mathcal{G}|\mathcal{D})$ and compute the posterior weighted average (PWA) Hamming distance $\sum_{\mathcal{G}} p(\mathcal{G}|\mathcal{D})\mathcal{H}(\mathcal{G}, \mathcal{G}^*)$ and the posterior weighted number of links. To approximate the posterior we considered all graphs $\mathcal{G}$ visited by the hill climbing algorithm. The results are reported in the table. The posterior was dominated by a few hundred graphs $\mathcal{G}$ with good scores.

4 Discussion

This paper considered the problem of learning the structure of a Bayesian network in settings in which modern, possibly overparametrized, neural networks might be used to model its conditional distributions. We proposed the use of the prequential plug-in score as a MDL-based model selection criterion that does not require any explicit sparsity regularization, and provided a specific implementation using neural networks that shows good performance, leads to sparse, parsimonious structural inferences, and often recovers the structure underlying the data generation mechanism.

Previous literature on MDL-based approaches to structure learning has focused on analytically tractable model families, such as tabular distributions or distributions with conjugate priors [Grünwald, 2007]. For categorical random variables, Silander et al. [2008] derived a factorized approximation to the normalized maximum likelihood by exploiting the conditional distribution structure. This work was further extended [Silander et al., 2018] to focus on model selection procedures that assign the same score to every model in a Markov equivalence class. However, it is not clear how to generalize these techniques to continuous random variables or complex models.

Outside the context of structure learning, MDL-inspired model selection for neural networks has primarily focused on using variational approximations or approximations based on AIC or BIC [Hinton and Van Camp, 1993, MacKay, 2003]. Lehtokangas et al. [1993] were perhaps the first to use a prequential plug-in approach, though the structure learned was the capacity of the neural network. Blier and Ollivier [2018] pioneered using the prequential plug-in distribution for modern scale neural networks architectures and essentially argued that prequential coding leads to much shorter description lengths than state-of-the-art variational approximations. They used the block-wise estimates described in Sect. 2.2 but without the confidence calibration; as a result, they had to switch between different model classes to avoid overfitting. They thus calculated prequential MDL estimates for a particular switching pattern, not for a model class. Bornschein et al. [2020] extended the block-wise estimate with calibration and obtained MDL estimates for modern overparametrized neural networks without limiting their capacity.

Recent efforts on structure learning with modern neural networks has focused on improving scalability by framing structure search as a continuous optimization problem with regularized maximum-likelihood as a scoring metric (see Zheng et al. [2018], Yu et al. [2019], Zheng et al. [2020], Pamfil et al. [2020], and Vowels et al. [2021] for a review). Scalability is an important aspect that we did not consider. As a consequence, our experiments were only feasible with a small number of variables. While this might seem like a step backwards compared to recent work, we believe that it is important to investigate new scoring metrics without the confounding effect of approximating the search procedure. Proposals for scaling prequential scoring to a higher number of variables include classical approximation methods developed for Bayesian scores [Heckerman, 1999], techniques like dynamic programming [Malone et al., 2011], branch and bound [de Campos and Ji, 2011], mathematical programming [Jaakkola et al., 2010, Cussens, 2011], and continuous optimization approaches [Zheng et al., 2018, Yu et al., 2019, Zheng et al., 2020, Pamfil et al., 2020].

Our approach effectively uses the generalization performance when trained on limited data as a model selection criterion. As such it is related to recent work that uses adaptation speed, i.e. how quickly models adapt to changes in the data generating process, to infer causal structures [Ke et al., 2019, Bengio et al., 2020] (see Le Priol et al. [2020] for a theoretical justification of this principle). The prequential MDL perspective offers an alternative and potentially simpler argument based on sample-efficiency instead of gradient-step efficiency to justify such an approach. And, as we have shown, this perspective is not only theoretically well-motivated but also applicable to i.i.d. data.

References

Hirotogu Akaike. *Information Theory and an Extension of the Maximum Likelihood Principle*, pages 199–213. Springer New York, 1973.

Ankur Ankan and Abinash Panda. pgmpy: Probabilistic graphical models using Python. In *Python in Science Conference*, 2015.

Jordan T. Ash and Ryan P. Adams. On warm-starting neural network training. In *Advances in Neural Information Processing Systems*, pages 3884–3894, 2020.

Vladimir Batagelj and Ulrik Brandes. Efficient generation of large random networks. *Physical Review E*, 71(3):036113, 2005.

Yoshua Bengio, Tristan Deleu, Nasim Rahaman, Nan Rosemary Ke, Sebastien Lachapelle, Olexa Bilaniuk, Anirudh Goyal, and Christopher Pal. A meta-transfer objective for learning to disentangle causal mechanisms. In *International Conference on Learning Representations*, 2020.

Léonard Blier and Yann Ollivier. The description length of deep learning models. In *Advances in Neural Information Processing Systems*, pages 2216–2226, 2018.

Jörg Bornschein, Francesco Visin, and Simon Osindero. Small data, big decisions: Model selection in the small-data regime. In *International Conference on Machine Learning*, pages 1035–1044, 2020.

Thomas M. Cover. *Elements of Information Theory*. John Wiley & Sons, 1999.

Robert G. Cowell, A. Philip Dawid, Steffen Lauritzen, and David J. Spiegelhalter. *Probabilistic Networks and Expert Systems, Exact Computational Methods for Bayesian Networks*. Springer-Verlag, 2007.

James Cussens. Bayesian network learning with cutting planes. In *Uncertainty in Artificial Intelligence*, page 153–160, 2011.

A. Philip Dawid and Vladimir G. Vovk. Prequential probability: principles and properties. *Bernoulli*, 5(1):125–162, 1999.

Cassio P. de Campos and Qiang Ji. Efficient structure learning of Bayesian networks using constraints. *Journal of Machine Learning Research*, 12:663–689, 2011.

Mathias Drton and Marloes H. Maathuis. Structure learning in graphical modeling. *Annual Review of Statistics and Its Application*, 4(1):365–393, 2017.

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

Alex Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2014.

Peter Grünwald. A tutorial introduction to the minimum description length principle. *arXiv preprint arXiv:0406077*, 2004.

Peter Grünwald. *The Minimum Description Length Principle*. MIT Press, 2007.

Peter Grünwald and Teemu Roos. Minimum description length revisited. *International journal of mathematics for industry*, 11(01):1930001, 2019.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330, 2017.

David Heckerman. A tutorial on learning with Bayesian networks. In Michael I. Jordan, editor, *Learning in Graphical Models*. MIT Press, 1999.

David Heckerman, Dan Geiger, and David M. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20:197–243, 1995.

Geoffrey E. Hinton and Drew Van Camp. Keeping the neural networks simple by minimizing the description length of the weights. In *Conference on Computational Learning Theory*, pages 5–13, 1993.

Tommi Jaakkola, David Sontag, Amir Globerson, and Marina Meila. Learning Bayesian network structure using LP relaxations. In *International Conference on Artificial Intelligence and Statistics*, pages 358–365, 2010.

Nan Rosemary Ke, Olexa Bilaniuk, Anirudh Goyal, Stefan Bauer, Hugo Larochelle, Bernhard Schölkopf, Michael C Mozer, Chris Pal, and Yoshua Bengio. Learning neural causal models from unknown interventions. *arXiv preprint arXiv:1910.01075*, 2019.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

Timo Koski and John Noble. A review of Bayesian networks and structure learning. *Mathematica Applicanda*, 40, 2012.

Way Lam and Fahiem Bacchus. Learning Bayesian belief networks an approach based on the MDL principle. *Computational Intelligence*, 10, 1994.

Rémi Le Priol, Reza Babanezhad Harikandeh, Yoshua Bengio, and Simon Lacoste-Julien. An analysis of the adaptation speed of causal models. *arXiv preprint arXiv:2005.09136*, 2020.

Mikko Lehtokangas, Jukka Saarinen, Pentti Huuhtanen, and Kimmo Kaski. Neural network optimization tool based on predictive MDL principle for time series prediction. In *IEEE Conference on Tools with AI*, pages 338–342, 1993.

Ahmed Mabrouk, Christophe Gonzales, Karine Jabet-Chevalier, and Eric Chojnacki. An efficient Bayesian network structure learning algorithm in the presence of deterministic relations. *Frontiers in Artificial Intelligence and Applications*, 263:567–572, 2014.

David J. C. MacKay. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2003.

Brandon Malone, Changhe Yuan, and Eric A. Hansen. Memory-efficient dynamic programming for learning optimal Bayesian networks. In *AAAI Conference on Artificial Intelligence*, page 1057–1062, 2011.

Roxana Pamfil, Nisara Sriwattanaworachai, Shaan Desai, Philip Pilgerstorfer, Konstantinos Georgatzis, Paul Beaumont, and Bryon Aragam. DYNOTEARS: Structure learning from time-series data. In *International Conference on Artificial Intelligence and Statistics*, pages 1595–1605, 2020.

Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., 1988.

Judea Pearl. *Causality: Models, Reasoning and Inference*. Springer, 2000.

Judea Pearl, Madelyn Glymour, and Nicholas P. Jewell. *Causal Inference in Statistics: A Primer*. John Wiley & Sons, 2016.

Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of Causal Inference: Foundations and Learning Algorithms*. MIT Press, 2017.

Jan Poland and Marcus Hutter. Asymptotics of discrete MDL for online prediction. *IEEE Transactions on Information Theory*, 51(11):3780–3795, 2005.

Ning Qian. On the momentum term in gradient descent learning algorithms. *Neural Networks*, 12(1): 145–151, 1999.

Jorma J. Rissanen. Fisher information and stochastic complexity. *IEEE Transactions on Information Theory*, 42(1):40–47, 1996.

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

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

Tomi Silander, Teemu Roos, Petri Kontkanen, and Petri Myllymäki. Factorized normalized maximum likelihood criterion for learning Bayesian network structures. In *European Workshop on Probabilistic Graphical Models*, pages 257–264, 2008.

Tomi Silander, Janne Leppä-aho, Elias Jääsaari, and Teemu Roos. Quotient normalized maximum likelihood criterion for learning Bayesian network structures. In *International Conference on Artificial Intelligence and Statistics*, pages 948–957, 2018.

Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction, and Search*. MIT Press, 2000.

Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems*, 2020.

Erdogan Taskesen. bnlearn. `https://github.com/erdogant/bnlearn`, 2019.

Matthew J. Vowels, Necati Cihan Camgoz, and Richard Bowden. D'ya like DAGs? A survey on structure learning and causal discovery. *arXiv preprint arXiv:2103.02582*, 2021.

Yue Yu, Jie Chen, Tian Gao, and Mo Yu. DAG-GNN: DAG structure learning with graph neural networks. In *International Conference on Machine Learning*, pages 7154–7163, 2019.

Xun Zheng, Bryon Aragam, Pradeep Ravikumar, and Eric P. Xing. DAGs with NO TEARS: Continuous optimization for structure learning. In *Advances in Neural Information Processing Systems*, pages 9472–9483, 2018.

Xun Zheng, Chen Dan, Bryon Aragam, Pradeep Ravikumar, and Eric P. Xing. Learning sparse nonparametric DAGs. In *International Conference on Artificial Intelligence and Statistics*, pages 3414–3425, 2020.