VICAUSE: Simultaneous missing value imputation and causal discovery

Pablo Morales-Alvarez [1] **Angus Lamb** [2] **Simon Woodhead** [3]
Simon Peyton Jones [2] **Miltiadis Allamanis** [2] **Cheng Zhang** [2]

Abstract

Missing values constitute an important challenge in real-world machine learning for both prediction and causal discovery tasks. However, only few methods in causal discovery can handle missing data in an efficient way, while existing imputation methods are agnostic to causality. In this work we propose VICAUSE, a novel approach to simultaneously tackle missing value imputation and causal discovery efficiently with deep learning. Particularly, we propose a generative model with a structured latent space and a graph neural network-based architecture, scaling to large number of variables. Moreover, our method can discover relationship between groups of variables which is useful in many real-world applications. VICAUSE shows improved performance compared to popular and recent approaches in both missing value imputation and causal discovery.

1. Introduction

The causal structure of data is a powerful source of information for real-world decision making, and it can improve and complement other learning tasks. However, historically causality and machine learning research have evolved separately. The last years have witnessed a growing interest in integrating causality techniques to specific tasks such as classification (Fatemi et al., 2021; Kyono et al., 2020), time-series prediction (Kipf et al., 2018; Löwe et al., 2020), and model explainability (Schwab & Karlen, 2019). One of the main challenges in real-world machine learning is the presence of missing data (Mohan & Pearl, 2014; Nabi et al., 2020; Rubin, 1976). We hypothesize that causal discovery can help the task of missing value imputation, since the relationships between variables are paramount for such task. To address this, several important aspects need to be considered. First, the number of possible causal graphs grows super-exponentially with the number of variables (Peters et al., 2017), so that scalable approaches to causal discovery are required. Second, we aim to perform causal discovery in the presence of missing values, which is not considered in standard approaches (Peters et al., 2017; Strobl et al., 2018; Tu et al., 2019a). Third, we seek to model complex relationships between variables, so that flexible deep learning models are required.

Moreover, many real-world applications require discovering causal relationships between *groups* of variables, which is challenging. Standard algorithms find relationships between the fully observed individual variables. However, this is neither efficient nor meaningful enough when dealing with datasets that contain a large number of variables (Wang & Drton, 2019). Instead, the variables may be grouped in a smaller number of semantically coherent pre-defined groups. One setting in which such a need arises is in the education domain (Wang et al., 2021; 2020). Education data can contain student responses to thousands of individual questions, where each question belongs to a broader topic. It is insightful to find relationships between topics instead of individual questions to help teachers adjust the curriculum. For instance, if there exists a causal relationship from one topic to another, the former should be taught earlier in the curriculum. Also, educational data is inherently sparse, since it is not feasible to ask every question to every student.

In this work we propose VICAUSE (missing value imputation with causal discovery), a novel approach to simultaneously tackle missing data imputation and causal discovery (Sec. 2). VICAUSE provides two outputs in one framework. This is accomplished by inferring a generative model that leverages a structured latent space and a decoder based on Graph Neural Networks (GNN) (Gilmer et al., 2017; Scarselli et al., 2009). Namely, the structured latent space endows each variable with its own latent subspace, and the interactions between the subspaces are regulated by a GNN whose behavior depends on the graph of causal relationships, see Fig. 1(a). VICAUSE satisfies all the desired properties: it leverages continuous optimization of the causal structure to achieve scalability (Zheng et al., 2018; 2020), it can be used in the presence of missing data, and it makes use of deep learning architectures for increased

[1]University of Granada (contributed during an internship in Microsoft) [2]Microsoft Research [3]Eedi. Correspondence to: Cheng Zhang <cheng.zhang@microsoft.com>.

Workshop on the Neglected Assumptions in Causal Inference (NACI) at the 38th International Conference on Machine Learning, 2021

flexibility. Moreover, the causal structure can be learned at different levels of granularity when the variables are organised in groups (Sec. 2.4). We evaluate VICAUSE in both synthetic (Appx. B) and two real-world applications (Sec. 3.1 and 3.2) that cover different types of variables (continuous or discrete), number of variables (i.e., graph sizes), and granularity of the causal discovery (i.e. variable-level or group-level causal structure discovery). VICAUSE shows improved performance in both missing data imputation and causal discovery when compared to popular and recent approaches for each task.

2. Model Description

2.1. Problem setting

Our goal is to develop a model that jointly learns to impute missing values and finds causal relationships between variables. The input to VICAUSE is a $N \times D$ training set $\mathbf{X} = \{\mathbf{x}_n\}_{n=1}^N$ with N data points and D variables, which may contain missing values. The observed and unobserved training values are denoted $\mathbf{X}_O$ and $\mathbf{X}_U$, respectively. In this work, we assume data are either missing completely at random (MCAR) or missing at random (MAR). The output of VICAUSE is i) a model that is able to impute missing values for a previously unseen test sample $\tilde{\mathbf{x}} \in \mathbb{R}^D$, and ii) a directed graph representing the causal relationships between the D variables. The graph is represented by its adjacency matrix $\mathbf{G}$, i.e. a $D \times D$ matrix whose element $\mathbf{G}_{ij}$ is 1 if there exists a causal relationship from the i-th variable to the j-th, and is 0 otherwise.

VICAUSE aims to discover the underlying causal relationships given partially observed data, and the learned model can also be used to impute missing data for test samples. We use a score-based approach for causal discovery. Inspired by (Heckerman et al., 2006), our score is defined as the posterior probability of $\mathbf{G}$ given the partially observed training data, subject to the constraint that $\mathbf{G}$ forms a directed acyclic graph (DAG). Thus, our objective is:

$$\mathbf{G}_\star = \arg\max_{\mathbf{G}\in\text{DAGs}} \mathrm{p}(\mathbf{X}_O|\mathbf{G})\mathrm{p}(\mathbf{G}). \tag{1}$$

To optimize over the causal structure with the DAG constraint in Eq. 1, we resort to recent continuous optimization techniques (Kyono et al., 2020; Zheng et al., 2018; 2020). Namely, it has been shown that $\mathbf{G}$ represents a DAG if and only if the non-negative quantity $\mathcal{R}(\mathbf{G}) = \mathrm{tr}(e^{\mathbf{G}\odot\mathbf{G}}) - D - 1$ equals zero (Zheng et al., 2018). To leverage this DAG-ness characterisation, we follow (Kyono et al., 2020; Yu et al., 2019) and introduce a regulariser based on $\mathcal{R}(\mathbf{G})$ to favour the DAG-ness of the solution, i.e.

$$\mathbf{G}_\star = \arg\max_{\mathbf{G}} \left(\mathrm{p}(\mathbf{X}_O|\mathbf{G})\mathrm{p}(\mathbf{G}) - \lambda\mathcal{R}(\mathbf{G})\right). \tag{2}$$

The model used to compute the score needs to handle partial observations. In addition, with the learned model we can impute missing values given any observations. Thus, given a test sample $\widetilde{\mathbf{x}} \in \mathbb{R}^D$ with partially observed variables, we can estimate the distribution over $\widetilde{\mathbf{x}}_U$ (the unobserved values) given $\widetilde{\mathbf{x}}_O$ (the observed ones) using the learned model (Eq. 10).

2.2. Generative model and amortized inference

Generative model. We assume that the observations in $\mathbf{X}$ are generated given the relationships $\mathbf{G}$ and exogenous noise $\mathbf{Z}$. Fig. 1(b) illustrates this generative process.Thus, we can write the probabilistic model as

$$\mathrm{p}\left(\mathbf{X}, \mathbf{Z}, \mathbf{G}\right) = \mathrm{p}(\mathbf{G}) \textstyle\prod_n \mathrm{p}(\mathbf{x}_n|\mathbf{Z}_n, \mathbf{G})\mathrm{p}(\mathbf{Z}_n). \tag{3}$$

We use deep learning, in particular a Graph Neural Network (GNN) for f_θ, to provide a highly flexible model of the generative process.

Amortized variational inference. The true posterior distribution over $\mathbf{Z}$ and $\mathbf{G}$ cannot be obtained in closed form in Eq. 3, since we use a deep learning architecture. Therefore, we resort to efficient amortized variational inference as in (Kingma & Welling, 2013; Kingma et al., 2019; Zhang et al., 2018). Here, we consider a fully factorized variational distribution $\mathrm{q}(\mathbf{Z}, \mathbf{G}) = \mathrm{q}(\mathbf{G}) \prod_{n=1}^N \mathrm{q}_\phi(\mathbf{Z}_n|\mathbf{x}_n)$, where $\mathrm{q}_\phi(\mathbf{Z}_n|\mathbf{x}_n)$ is a Gaussian whose mean and (diagonal) covariance matrix are given by the *encoder*. For $\mathrm{q}(\mathbf{G})$ we consider the product of independent Bernoulli distributions over the edges, that is, each edge is present with a probability $\mathbf{G}_{ij} \in (0, 1)$ to be estimated. With this formulation, the evidence lower bound (ELBO) is

$$\begin{aligned}\mathrm{ELBO} = \textstyle\sum_n \big\{&\mathbb{E}_{\mathrm{q}_\phi(\mathbf{Z}_n|\mathbf{x}_n)\mathrm{q}(\mathbf{G})} \log \mathrm{p}(\mathbf{x}_n|\mathbf{Z}_n, \mathbf{G}) \\ &- \mathrm{KL}[\mathrm{q}_\phi(\mathbf{Z}_n|\mathbf{x}_n)||\mathrm{p}(\mathbf{Z}_n)] - \mathrm{KL}[\mathrm{q}(\mathbf{G})||\mathrm{p}(\mathbf{G}))]\end{aligned} \tag{4}$$

Next, we dive into our choice of generator, which uses a GNN to regulate the interactions between the variables. Then, we focus on the inference network, which respects the variable-wise structure of the latent space.

Generator. The generator (also known as decoder) takes $\mathbf{Z}_n$ and $\mathbf{G}$ as input, and outputs the reconstructed $\hat{\mathbf{x}}_n = f_\theta(\mathbf{Z}_n, \mathbf{G})$, where θ are the decoder parameters. We partition the exogenous noise $\mathbf{Z}_n$ into D parts, where $\mathbf{z}_{n,d}$ is the exogenous noise for each variable $d = 1, \ldots, D$. Notice that this defines a variable-wise structured latent space, see Fig. 6(a) in the Appendix for an illustration. Intuitively, the decoder regulates the interactions between variables with a GNN whose behavior is determined by the relationships in $\mathbf{G}$. Specifically, this is done in two steps: GNN message passing layers and a final read-out layer yielding the reconstructed sample.

GNN message passing in the generator. In message passing, the information flows between nodes in T consecutive node-to-edge (n2e) and edge-to-node (e2n) operations

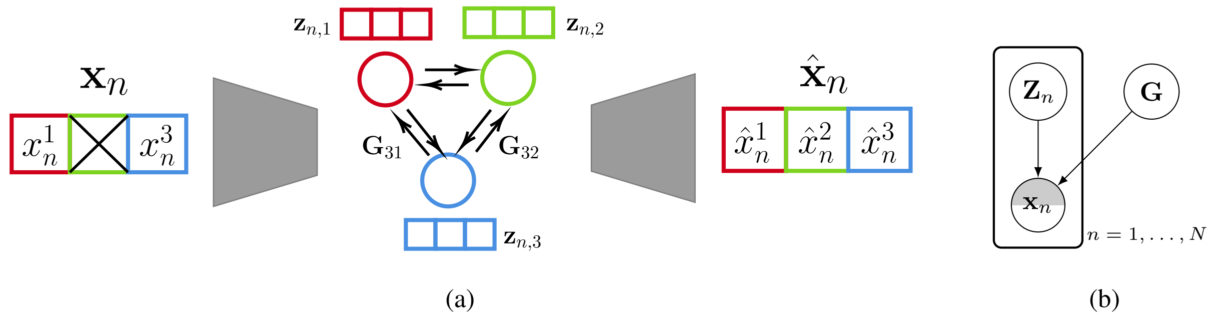

Figure 1: (a) Graphic representation of VICAUSE. (b) Probabilistic graphical model for VICAUSE.

(Gilmer et al., 2017; Kipf et al., 2018). At the t-th step, each edge $i \to j$ has a representation (or embedding) $\mathbf{h}_{i\to j}$ summarizing the information sent from node i to node j. Since we are interested in the imputation task, where we may want to predict the value of the parents from their children only, we also introduce the *backward* embedding. This is denoted $\mathbf{h}^b_{i\to j}$, and codifies the information that the $i \to j$ edge lets flow from the j-th to the i-th node (for symmetry, the "standard" embedding is called here *forward* embedding and denoted $\mathbf{h}^f_{i\to j}$). Specifically, the n2e and e2n operations used in VICAUSE are

$$\text{n2e}: \quad \mathbf{h}^{(t),f}_{i\to j} = \text{MLP}^f\left(\left[\mathbf{z}^{(t-1)}_i, \mathbf{z}^{(t-1)}_j\right]\right), \tag{5}$$

$$\mathbf{h}^{(t),b}_{i\to j} = \text{MLP}^b\left(\left[\mathbf{z}^{(t-1)}_i, \mathbf{z}^{(t-1)}_j\right]\right), \tag{6}$$

$$\text{e2n}: \quad \mathbf{z}^{(t)}_i = \text{MLP}^{e2n}\left(\sum\nolimits_{k\neq i} \mathbf{G}_{ik} \cdot \left\{\mathbf{h}^{(t),f}_{k\to i} + \mathbf{h}^{(t),b}_{i\to k}\right\}\right). \tag{7}$$

Here, t refers to the t-th iteration of message passing (that is, $\mathbf{Z}^{(0)} = \mathbf{Z}_n$, notice that we omit the subindex n for simplicity). Finally, MLP^f, MLP^b and MLP^{e2n} are MLPs to be estimated. Interestingly, Eqs. (5)-(7) link together the imputation and causal discovery tasks, since the information flow between two nodes (i.e. variables) is proportional to the weight of the corresponding edge.

Read-out layer in the generator. After T iterations of GNN message passing, we have $\mathbf{Z}^{(T)}$. We then apply a final function that maps $\mathbf{Z}^{(T)}$ to the reconstructed $\hat{\mathbf{x}}$, i.e. $\hat{\mathbf{x}} = (g(\mathbf{z}^T_1), \ldots, g(\mathbf{z}^T_D))$, with g given by an MLP. Notice that the decoder parameters $\boldsymbol{\theta}$ include the parameters of four neural networks: MLP^f, MLP^b, MLP^{e2n} and g.

Inference network. As in standard VAEs, the encoder maps a sample $\mathbf{x}_n$ to its latent representation $\mathbf{Z}_n$. In VICAUSE, we additionally ensure that the encoder respects the structure of the latent space. As discussed before, $\mathbf{Z}_n$ is partitioned in D parts, one for each variable. To obtain the mean and variance of $\mathbf{Z}_n$, we utilize a multi-head approach with shared parameter $\phi = \{\phi_\mu, \phi_\sigma\}$ for all the variables:

$$\boldsymbol{\mu}_n = \left(\mu_{\phi_\mu}(x_{n,1}), \ldots, \mu_{\phi_\mu}(x_{n,D})\right)^\intercal, \tag{8}$$
$$\boldsymbol{\sigma}_n = \left(\sigma_{\phi_\sigma}(x_{n,1}), \ldots, \sigma_{\phi_\sigma}(x_{n,D})\right)^\intercal.$$

Here, μ_{ϕ_μ} and σ_{ϕ_σ} are given by neural networks. When missing values are present, we replace them with a constant as in (Nazabal et al., 2020). A graphic representation on how the encoder respects the structure of the latent space is provided in the Appendix, Fig. 7(a).

2.3. Training VICAUSE

Given the model in Sec. 2.2, we have the final objective to minimize w.r.t. θ, ϕ and $\mathbf{G}$:

$$\mathcal{L}_{\text{VICAUSE}}(\theta, \phi, \mathbf{G}) = -\text{ELBO} + \lambda \mathbb{E}_{\text{q}(\mathbf{G})}\left[\mathcal{R}(\mathbf{G})\right], \tag{9}$$

where ELBO is given by Eq. 4 and the DAG regulariser $\mathcal{R}(\mathbf{G})$ is defined in Sec. 2.1.

Evaluating the training loss $\mathcal{L}_{\text{VICAUSE}}$. VICAUSE can work with any type of data. The log-likelihood term (the first term in Eq. 4) is defined according to the data type. We use a Gaussian likelihood for continuous variables and a Bernoulli likelihood for binary ones. The standard reparametrization trick is used to sample $\mathbf{Z}_n$ from the Gaussian distribution $\text{q}_\phi(\mathbf{Z}_n|\mathbf{x}_n)$ (Kingma & Welling, 2013; Kingma et al., 2019). To backpropagate the gradients through the discrete variable $\mathbf{G}$, we resort to the Gumbel-softmax trick to sample from $\text{q}(\mathbf{G})$ (Jang et al., 2017; Maddison et al., 2017). The $\text{KL}[\text{q}_\phi(\mathbf{Z}_n|\mathbf{x}_n)||\text{p}(\mathbf{Z}_n)]$ term can be obtained in closed-form, since both are Gaussian distributions. The $\text{KL}[\text{q}(\mathbf{G})||\text{p}(\mathbf{G}))]$ term can also be obtained in closed-form, since both are the product of independent Bernoulli distributions over the edges. Notice that this term allows for specifying prior knowledge on the causal structure (e.g. sparsity). Finally, the DAG-loss regulariser in Eq. 9 can be computed by evaluating the function $\mathcal{R}$ on a Gumbel-softmax sample from $\text{q}(\mathbf{G})$. To make the model adapt to different sparsity levels in the training data $\mathbf{X}$, during training we drop a random percentage of the observed values. The full training procedure for VICAUSE is summarised in Algorithm 1.

Two-step training. Although important for the imputation task, the use of both forward and backward MLPs introduces a symmetry that hampers the correct identification of the causal direction. Namely, if the forward and backward MLPs are similar models, then $A \to B$ and $B \to A$ produce

Algorithm 1 Training VICAUSE.

Input : Training dataset $\mathbf{X}$, possibly with missing values.
foreach *batch of samples* $\{\mathbf{x}_n\}_{n\in B}$ **do**
- Drop a percentage of the data for each sample $\mathbf{x}_n$.
- Encode $\mathbf{x}_n$ through the reparametrization trick to sample $\mathbf{Z}_n \sim \mathcal{N}(\mu_\phi(\mathbf{x}_n), \sigma^2_\phi(\mathbf{x}_n))$.
- Use the Gumbel-softmax to sample $\mathbf{G}$ from the posterior $\mathrm{q}(\mathbf{G})$.
- Use the decoder to calculate the reconstructed sample $\hat{\mathbf{x}}_n = f_\theta(\mathbf{Z}_n, \mathbf{G})$.
- Calculate the training loss $\mathcal{L}_{\text{VICAUSE}}$ (Eq. 9).
- Gradient step w.r.t. ϕ (encoder parameters), θ (decoder parameters) and $\mathbf{G}$ (posterior edge probabilities).

Output : Encoder parameters ϕ, decoder parameters θ, and posterior probabilities over the edges $\mathbf{G}$.

exactly the same information flow when the two MLPs are swapped. To overcome this issue, we propose a two-step training scheme. In the first stage, the backward MLP is disabled so that the symmetry is broken and the algorithm can learn the causal structure. In the second stage, we fix the graph structure (i.e. the variational parameter $\mathbf{G}$) and continue to train the model with backward MLP. This two-stage training process allows VICAUSE to leverage the backward MLP for the imputation task without interfering with the causal discovery.

Revisiting the learning objectives. The optimal graph of relationships, which was denoted $\mathbf{G}_\star$ in Sec. 2.1, is given by the posterior graph of probabilities $\mathbf{G}$ (it gives the best score as it maximizes the posterior). Similar to (Ma et al., 2019; Nazabal et al., 2020), the trained model can impute missing values for a test instance $\widetilde{\mathbf{x}}$ as

$$\begin{aligned}\mathrm{p}(\widetilde{\mathbf{x}}_U|\widetilde{\mathbf{x}}_O, \mathbf{X}) &= \textstyle\int \mathrm{p}(\widetilde{\mathbf{x}}_U|\mathbf{Z},\mathbf{G})\mathrm{q}_\phi(\mathbf{Z}|\widetilde{\mathbf{x}})\mathrm{q}(\mathbf{G})\mathrm{d}\mathbf{Z}\mathrm{d}\mathbf{G} \\ &= \mathbb{E}_{\mathrm{q}_\phi(\mathbf{Z}|\widetilde{\mathbf{x}})\mathrm{q}(\mathbf{G})}\mathrm{p}(\widetilde{\mathbf{x}}_U|\mathbf{Z},\mathbf{G}). \end{aligned} \tag{10}$$

Therefore, the distribution over $\widetilde{\mathbf{x}}_U$ (the missing values) is obtained with $\widetilde{\mathbf{x}}$ as input.

2.4. Extension to relationship discovery between groups of variables

Thus far, we have assumed that the relationship between individual variables are of interest. As discussed in Sec. 1, finding the relationships between *groups* of variables is needed in many real-world applications. Here, we extend VICAUSE to discover relationships between (pre-defined) groups of variables.

Problem definition. We assume that the D variables in $\mathbf{X}$ are organized in $M \ll D$ groups. For each group $m = 1, \ldots, M$, we write $\mathcal{I}_m$ for the variables associated to that group (i.e. $\mathcal{I}_m = \{4, 5, 6\}$ means that the m-th group contains the 4th, 5th and 6th variables). The goal is to learn to impute missing values for test samples $\widetilde{\mathbf{x}} \in \mathbb{R}^D$, and learn causal relationships between the M groups of variables. In particular, the shape of the learned parameter $\mathbf{G}$ is now $M \times M$. Also, the structured latent representation $\mathbf{Z}$ is split in M parts, each one corresponding to a different group, see Fig. 6(b) in the Appendix for an illustration.

VICAUSE for groups. The formulation of Sec. 2.2 can be naturally generalised to this setting. The generative model is analogous, but each node must be thought now as a group of variables (instead of a single variable). The main difference lies in the mappings that connect the sample $\mathbf{x}_n$ and its latent representation $\mathbf{Z}_n$. Specifically, there are two such mappings: the encoder and the read-out layer in the decoder. Unlike before (Eq. 8), the same neural network cannot be used now for all the latent subspaces, since different groups of variables may have different dimensionalities (namely, the m-th group has a dimensionality of $|\mathcal{I}_m|$, i.e. the number of variables in that group). To overcome this, we propose to use a group-specific neural network for each latent subspace. Specifically, the encoder computes the mean of the latent variable as

$$\boldsymbol{\mu}_n = \left(\mu^1_{\phi_1}(\boldsymbol{\chi}_1), \ldots, \mu^M_{\phi_M}(\boldsymbol{\chi}_M)\right)^\intercal, \tag{11}$$

where $\boldsymbol{\chi}_m$ includes all the variables in the m-th group (i.e., $\boldsymbol{\chi}_m = [x_i]_{i\in\mathcal{I}_m}$), and $\mu^1_{\phi_1}, \ldots, \mu^M_{\phi_M}$ are M different MLPs. The expressions for the variance and for the read-out layer within the decoder are analogous. Fig. 7(b) in the Appendix shows a graphical representation of Eq. 11. The rest of training for VICAUSE is identical to the case of variables, recall Algorithm 1.

3. Experiments

We evaluate the performance of VICAUSE in a synthetic experiment where the data generation process is controlled (Appx. B), a semi-synthetic problem (simulated data from a real-world problem) with many more variables (Sec. 3.1), and the real-world problem that motivated the development of the group-level extension (Sec. 3.2).

Baselines. For the causal discovery task, we consider five baselines. PC (Spirtes et al., 2000) and GES (Chickering, 2002) are the most popular methods in constrained-based and score-based causal discovery approaches, respectively. We also consider three recent algorithms based on continuous optimization and deep learning: NOTEARS (Zheng et al., 2018), the non-linear (NL) extension of NOTEARS (Zheng et al., 2020), and DAG-GNN (Yu et al., 2019). Unlike VICAUSE, these causality baselines cannot deal with missing values in the training data. Therefore, in Appx. B and Sec. 3.1 we work with fully observed training data. In contrast, the real-world data in Sec. 3.2 comes with partially observed training data, and the goal is to discover group-wise relationships. Thus the causality baselines cannot be used there, as they deal with variable-wise relationship only. For the missing data imputation task, we also consider five baselines. Mean Imputing and Major-

	Accuracy	AUROC	AUPR
Majority vote	0.9268±0.0003	0.5304±0.0003	0.3366±0.0025
Mean imputing	0.9268±0.0003	0.8529±0.0012	0.3262±0.0034
MICE	0.9469±0.0007	0.9319±0.0010	0.6513±0.0046
Missforest	0.9305±0.0004	0.8915±0.0093	0.5227±0.0033
PVAE	0.9415±0.0003	0.9270±0.0007	0.5934±0.0046
VICAUSE	**0.9471±0.0006**	**0.9392±0.0008**	**0.6597±0.0053**

Table 1: Imputation results for neuropathic pain dataset (mean and standard error over five runs).

ity Vote are popular techniques used as reference, Missforest (Stekhoven & Bühlmann, 2012) and MICE (Buuren & Groothuis-Oudshoorn, 2010) are two of the most widely-used imputation algorithms, and PVAE (Ma et al., 2019) is a recent algorithm based on amortized inference.

Metrics. Imputation performance is evaluated with standard metrics such as RMSE (for continuous variables) and accuracy (for categorical variables). For categorical variables, we also provide the area under the ROC and the Precision-Recall curves (AUROC and AUPR, respectively), which are especially useful for imbalanced data (such as that in Sec. 3.1). Regarding causal discovery, we consider both *adjacency* and *orientation* metrics as is common practice (Glymour et al., 2019; Tu et al., 2019a). Whereas the former do not take into account the direction of the edges, the latter do. For each metric (adjacency and orientation) we compute recall, precision and F_1-score. We also provide *causal accuracy*, a popular metric introduced in Claassen & Heskes (2012) that takes into account edge orientation.

3.1. Neuropathic pain dataset

Motivation and dataset description. This experiment extends the previous one in three directions. First, the relationships used are not synthetic, but instead come from a well-studied medical setting (Tu et al., 2019b). Second, the number of variables considered is 222 — significantly larger than before. Third, the variables are binary, rather than continuous. The dataset contains records of different patients regarding the diagnosis of symptoms associated to neuropathic pain. The train and test sets have 1000 and 500 patients respectively, for which 222 binary variables have been measured (the value is 1 if the symptom is present for the patient and 0 otherwise). The data was generated with the Neuropathic Pain Diagnosis Simulator, whose properties have been evaluated from the medical and statistical perspectives (Tu et al., 2019b).

Imputation performance. VICAUSE shows competitive or superior performance when compared to the baselines, see Table 1. Notice that AUROC and AUPR allow for an appropriate threshold-free assessment in this imbalanced scenario. Indeed, as expected from medical data, the majority of values are 0 (no symptoms); here it is around 92% of them in the test set. Interestingly, it is precisely in AUPR where the differences between VICAUSE and the rest of baselines are larger (except for MICE, whose performance is very similar to that of VICAUSE in this dataset).

Causality results. As in the synthetic experiment, VICAUSE outperforms the causal discovery baselines, see Table 2. Notice that NOTEARS (NL) is slightly better in terms of adjacency-precision, i.e. the edges that it predicts are slightly more reliable. However, this is at the expense of a significantly lower capacity to detect true edges, see the recall and the trade-off between both (F_1-score).

3.2. Eedi topics dataset

Motivation and dataset description. This experiment extends the previous ones in three directions. First, we tackle an important real-world problem in the field of AI-powered educational systems (Wang et al., 2021; 2020). Second, we are interested in relationships between groups of variables (instead of individual variables). Third, the training data is very sparse, with 25.9% observed values. The dataset contains the responses given by 6147 students to 948 mathematics questions. The 948 variables are binary (1 if the student provided the correct answer and 0 otherwise). These 948 questions target very specific mathematical concepts, and they are grouped within a more meaningful hierarchy of *topics*, see Fig. 4. Here we apply the extension introduced in Sec. 2.4 to find relationships between groups (the topics). Specifically, we group the topics at the third level of the topic hierarchy (Fig. 4 in the appendix), resulting in 57 nodes in the GNN.

Imputation results. VICAUSE achieves competitive or superior performance when compared to the baselines (Table 4). Although the dataset is relatively balanced (54% of the values are 1), we provide AUROC and AUPR for completeness. Notice that this setting is more challenging than the previous ones, since we learn relationships between groups of variables (topics). Indeed, whereas the group extension in Sec. 2.4 allows for more meaningful relationships, the information flow happens at a less granular level. Interestingly, even in this case VICAUSE obtains similar or improved imputation results compared to the baselines.

Causal discovery results between groups. Most of the baselines used so far cannot be applied here because i) they cannot learn relationships between groups of variables and ii) they cannot deal with partially observed training data. DAG-GNN is the only one that can be adapted to satisfy both properties. For the first one, we adapt DAG-GNN following the same strategy as in VICAUSE, i.e. replacing missing values with a constant value. For the latter, notice that DAG-GNN can be used for vector-valued variables according to the original formulation (Yu et al., 2019). However, all of them need to have the same dimensionality. To cope with arbitrary groups, we apply the group-specific

	Adjacency			Orientation			Causal Accuracy
	Recall	Precision	F_1-score	Recall	Precision	F_1-score	
PC	0.046±0.001	0.375±0.006	0.082±0.001	0.024±0.001	0.199±0.011	0.044±0.002	0.058±0.003
GES	0.110±0.001	0.436±0.008	0.176±0.002	0.082±0.001	0.323±0.009	0.131±0.003	0.121±0.001
NOTEARS (L)	0.006±0.000	0.011±0.001	0.008±0.000	0.001±0.000	0.001±0.001	0.001±0.000	0.001±0.000
NOTEARS (NL)	0.011±0.001	**0.644±0.025**	0.022±0.002	0.006±0.001	0.354±0.018	0.012±0.001	0.006±0.001
DAG-GNN	0.129±0.028	0.272±0.101	0.128±0.027	0.051±0.010	0.126±0.059	0.050±0.007	0.051±0.010
VICAUSE	**0.261±0.006**	0.637±0.009	**0.370±0.005**	**0.236±0.007**	**0.573±0.005**	**0.334±0.006**	**0.245±0.006**

Table 2: Causal discovery results for neuropathic pain dataset (mean and std error over five runs).

VICAUSE	Number	Algebra	Geometry
Number	30	4	3
Algebra	2	6	0
Geometry	0	0	5

DAG-GNN	Number	Algebra	Geometry
Number	8	3	6
Algebra	1	5	2
Geometry	14	7	11

Random	Number	Algebra	Geometry
Number	7	4	6
Algebra	8	1	6
Geometry	6	3	9

Table 3: Distribution of the relationships across level 1 topics (Number, Algebra, and Geometry). The item (i, j) refers to edges in the direction $i \to j$. The proportion of relationships inside level 1 topics is 82%, 42% and 34% for VICAUSE, DAG-GNN and *Random*, respectively.

	Accuracy	AUROC	AUPR
Majority vote	0.6260±0.0000	0.6208±0.0000	0.7465±0.0000
Mean imputing	0.6260±0.0000	0.6753±0.0000	0.6906±0.0000
MICE	0.6794±0.0005	0.7453±0.0007	0.7483±0.0010
Missforest	0.6849±0.0005	0.7219±0.0007	0.7478±0.0008
PVAE	0.7138±0.0005	**0.7852±0.0001**	**0.8204±0.0002**
VICAUSE	**0.7147±0.0007**	0.7815±0.0008	0.8179±0.0006

Table 4: Imputation results for Eedi topics dataset (mean and standard error over five runs).

	Adjacency		Orientation	
	Expt 1	Expt 2	Expt 1	Expt 2
Random	2.04	2.08	1.44	1.40
DAG-GNN	2.04	2.32	1.68	1.68
VICAUSE	**3.60**	**3.70**	**2.76**	**2.60**

Table 5: Average expert evaluation of the topic relationships. Cohen's κ inter-annotator agreement is 0.72 for adjacency and 0.76 for orientation (substantial agreement).

mappings described in Sec. 2.4 (Eq. 11). Finally, to have an additional reference, we also compare with randomly generated relationships, which we will refer to as *Random*.

Importantly, this is a real-world dataset with no ground truth on the true relationships. Therefore, we asked two experts (experienced high school teachers working with the Eedi dataset) to assess the validity of the relationships found by VICAUSE, DAG-GNN and *Random*. For each relationship, they evaluated the adjacency (whether it is sensible to connect the two topics) and the orientation (whether the first one is a prerequisite for the second one). They provided an integer value from 1 (strongly disagree) to 5 (strongly agree), i.e. the higher the better. The complete list of relationships and expert evaluations for VICAUSE, DAG-GNN and *Random* can be found in the appendix, see Table 11, Table 12 and Table 13, respectively. As a summary, Table 5 shows here the average evaluations: we see that the relationships discovered by VICAUSE score much more highly across both metrics than the baseline models.

Another interesting aspect is how the relationships found between level-3 topics are distributed across higher-level topics (recall Fig. 4). Intuitively, it is expected that most of the relationships happen *inside* higher-level topics (e.g. Number-related concepts are more probably related to each other than to Geometry-related ones). Table 3 shows such a distribution for the compared methods. Indeed, notice that there are more inside-topic relationships for VICAUSE (82%) and DAG-GNN (42%) than for *Random* (34%). An analogous analysis for the 25 level-2 topics is provided in the appendix, see Table 14 (VICAUSE), Table 15 (DAG-GNN), and Table 16 (*Random*).

4. Conclusions

We introduced VICAUSE, a novel approach that simultaneously performs causal discovery and learns to impute missing values. Both tasks are performed jointly: imputation is informed by the discovered relationships and vice-versa. This is achieved through a structured latent space and a GNN-based decoder. Namely, each variable has its own latent subspace, and the interactions between the latent subspaces are governed by the GNN through a (global) graph of relationships. Moreover, motivated by a real-world problem, VICAUSE is extended to learn the causal relationships among groups of variables (rather than variables themselves). VICAUSE fosters further research. In terms of causality, it would be interesting to carry out a theoretical analysis on identifiablilty, sample complexity etc. In terms of missing values imputation, our work assumes that the data are missing at random. It would be interesting to explore how VICAUSE can be extended to missing not at random scenario.

References

Buuren, S. v. and Groothuis-Oudshoorn, K. mice: Multivariate imputation by chained equations in r. *Journal of statistical software*, pp. 1–68, 2010.

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

Chickering, D. M. and Meek, C. Selective greedy equivalence search: finding optimal bayesian networks using a polynomial number of score evaluations. In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, pp. 211–219, 2015.

Chickering, M. Statistically efficient greedy equivalence search. In *Conference on Uncertainty in Artificial Intelligence*, pp. 241–249. PMLR, 2020.

Claassen, T. and Heskes, T. A bayesian approach to constraint based causal inference. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, pp. 207–216, 2012.

Dempster, A. P., Laird, N. M., and Rubin, D. B. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

Fatemi, B., Asri, L. E., and Kazemi, S. M. Slaps: Self-supervision improves structure learning for graph neural networks. *arXiv preprint arXiv:2102.05034*, 2021.

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. Neural message passing for quantum chemistry. In *International Conference on Machine Learning*, pp. 1263–1272. PMLR, 2017.

Glymour, C., Zhang, K., and Spirtes, P. Review of causal discovery methods based on graphical models. *Frontiers in genetics*, 10:524, 2019.

Heckerman, D., Meek, C., and Cooper, G. A bayesian approach to causal discovery. In *Innovations in Machine Learning*, pp. 1–28. Springer, 2006.

Hoyer, P., Janzing, D., Mooij, J. M., Peters, J., and Schölkopf, B. Nonlinear causal discovery with additive noise models. *Advances in neural information processing systems*, 21:689–696, 2008.

Jang, E., Gu, S., and Poole, B. Categorical reparameterization with gumbel-softmax. In *International conference on learning representations*, 2017.

Kingma, D. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Kingma, D. P., Welling, M., et al. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019.

Kipf, T., Fetaya, E., Wang, K.-C., Welling, M., and Zemel, R. Neural relational inference for interacting systems. In *International Conference on Machine Learning*, pp. 2688–2697. PMLR, 2018.

Kyono, T., Zhang, Y., and van der Schaar, M. Castle: Regularization via auxiliary causal graph discovery. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1501–1512. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/1068bceb19323fe72b2b344ccf85c254-Paper.pdf.

Löwe, S., Madras, D., Zemel, R., and Welling, M. Amortized causal discovery: Learning to infer causal graphs from time-series data. *arXiv preprint arXiv:2006.10833*, 2020.

Ma, C., Tschiatschek, S., Palla, K., Hernandez-Lobato, J. M., Nowozin, S., and Zhang, C. EDDI: Efficient dynamic discovery of high-value information with partial VAE. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 4234–4243. PMLR, 09–15 Jun 2019. URL http://proceedings.mlr.press/v97/ma19c.html.

Maddison, C. J., Mnih, A., and Teh, Y. W. The concrete distribution: A continuous relaxation of discrete random variables. In *International conference on learning representations*, 2017.

Mohan, K. and Pearl, J. On the testability of models with missing data. In *Artificial Intelligence and Statistics*, pp. 643–650. PMLR, 2014.

Monti, R. P., Zhang, K., and Hyvärinen, A. Causal discovery with general non-linear relationships using non-linear ica. In *Uncertainty in Artificial Intelligence*, pp. 186–195. PMLR, 2020.

Nabi, R., Bhattacharya, R., and Shpitser, I. Full law identification in graphical models of missing data: Completeness results. In *International Conference on Machine Learning*, pp. 7153–7163. PMLR, 2020.

Nazabal, A., Olmos, P. M., Ghahramani, Z., and Valera, I. Handling incomplete heterogeneous data using vaes. *Pattern Recognition*, 107:107501, 2020.

Peters, J., Janzing, D., and Schölkopf, B. *Elements of causal inference: foundations and learning algorithms*. The MIT Press, 2017.

Ramsey, J., Glymour, M., Sanchez-Romero, R., and Glymour, C. A million variables and more: the fast greedy equivalence search algorithm for learning high-dimensional graphical causal models, with an application to functional magnetic resonance images. *International journal of data science and analytics*, 3(2):121–129, 2017.

Rubin, D. B. Inference and missing data. *Biometrika*, 63(3): 581–592, 1976.

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. The graph neural network model. *IEEE Transactions on Neural Networks*, 1(20):61–80, 2009.

Scheffer, J. Dealing with missing data. In *Research Letters in the Information and Mathematical Sciences*, pp. 153–160, 2002.

Schwab, P. and Karlen, W. CXPlain: Causal Explanations for Model Interpretation under Uncertainty. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Shimizu, S., Hoyer, P. O., Hyvärinen, A., Kerminen, A., and Jordan, M. A linear non-gaussian acyclic model for causal discovery. *Journal of Machine Learning Research*, 7(10), 2006.

Spirtes, P. and Glymour, C. An algorithm for fast recovery of sparse causal graphs. *Social science computer review*, 9(1):62–72, 1991.

Spirtes, P., Glymour, C. N., Scheines, R., and Heckerman, D. *Causation, prediction, and search*. MIT press, 2000.

Stekhoven, D. J. and Bühlmann, P. Missforest—non-parametric missing value imputation for mixed-type data. *Bioinformatics*, 28(1):112–118, 2012.

Strobl, E. V., Visweswaran, S., and Spirtes, P. L. Fast causal inference with non-random missingness by test-wise deletion. *International journal of data science and analytics*, 6(1):47–62, 2018.

Tu, R., Zhang, C., Ackermann, P., Mohan, K., Kjellström, H., and Zhang, K. Causal discovery in the presence of missing data. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1762–1770. PMLR, 2019a.

Tu, R., Zhang, K., Bertilson, B. C., Kjellström, H., and Zhang, C. Neuropathic pain diagnosis simulator for causal discovery algorithm evaluation. In *33rd Conference on Neural Information Processing Systems (NeurIPS), DEC 08-14, 2019, Vancouver, Canada*, volume 32. Neural Information Processing Systems (NIPS), 2019b.

Wang, Y. S. and Drton, M. High-dimensional causal discovery under non-Gaussianity. *Biometrika*, 107 (1):41–59, 10 2019. ISSN 0006-3444. doi: 10.1093/biomet/asz055. URL https://doi.org/10.1093/biomet/asz055.

Wang, Z., Tschiatschek, S., Woodhead, S., Hernández-Lobato, J. M., Jones, S. P., Baraniuk, R. G., and Zhang, C. Educational question mining at scale: Prediction, analysis and personalization. *arXiv preprint arXiv:2003.05980*, 2020.

Wang, Z., Lamb, A., Saveliev, E., Cameron, P., Zaykov, Y., Hernandez-Lobato, J. M., Turner, R. E., Baraniuk, R. G., Barton, C., Jones, S. P., et al. Results and insights from diagnostic questions: The neurips 2020 education challenge. *arXiv preprint arXiv:2104.04034*, 2021.

Wu, G., Domke, J., and Sanner, S. Conditional inference in pre-trained variational autoencoders via cross-coding. *arXiv preprint arXiv:1805.07785*, 2018.

Yu, Y., Chen, J., Gao, T., and Yu, M. Dag-gnn: Dag structure learning with graph neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

Zhang, C., Bütepage, J., Kjellström, H., and Mandt, S. Advances in variational inference. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):2008–2026, 2018.

Zhang, K. and Hyvärinen, A. On the identifiability of the post-nonlinear causal model. In *25th Conference on Uncertainty in Artificial Intelligence (UAI 2009)*, pp. 647–655. AUAI Press, 2009.

Zheng, X., Aragam, B., Ravikumar, P., and Xing, E. P. DAGs with NO TEARS: Continuous Optimization for Structure Learning. In *Advances in Neural Information Processing Systems*, 2018.

Zheng, X., Dan, C., Aragam, B., Ravikumar, P., and Xing, E. P. Learning sparse nonparametric DAGs. In *International Conference on Artificial Intelligence and Statistics*, 2020.

A. Related Work

Since VICAUSE tackles missing value imputation and causal discovery simultaneously, we review the related work from both fields. Moreover, we review recent works that utilize causality to improve the performance of another deep learning task, similar to VICAUSE.

Causal discovery. Randomized controlled trials are often not possible in real-world. Causal discovery aims to find causal relationships between variables from historical data without additional experiments (Glymour et al., 2019). There are mainly three type of methods: constraint-based, score-based and functional causal models. Constraint-based ones exploit (conditional) independence tests to find the underlying causal structure, such as PC (Spirtes & Glymour, 1991) and Fast Causal Inference (FCI) (Spirtes et al., 2000). They have recently been extended to handle partially observed data through test-wise deletion and adjustments (Strobl et al., 2018; Tu et al., 2019a). Score-based methods find the causal structure by optimizing a scoring function, such as Greedy Equivalence Search (GES) (Chickering, 2002) and extensions (Chickering & Meek, 2015; Chickering, 2020; Ramsey et al., 2017; Zheng et al., 2018; 2020). In functional causal models, the effect variable is represented as a function of the direct causes and some noise term, with different assumptions on the functional form and the noise (Hoyer et al., 2008; Monti et al., 2020; Shimizu et al., 2006; Zhang & Hyvärinen, 2009). Traditional methods do not scale to large number of variables. Recently, continuous optimization of causal structures has become very popular within score-based methods (Zheng et al., 2018; 2020). In particular, continuous optimization has been combined with GNNs to improve the performance of structural equation models (SEMs) (Yu et al., 2019). VICAUSE also considers non-linear relationships through a GNN architecture. However, since it jointly learns to impute missing values, VICAUSE leverages a general GNN architecture based on message passing, which is not an extension of linear SEMs as in (Yu et al., 2019). Moreover, VICAUSE treats the graph of relationships in a fully probabilistic manner, handles missing values in the training data, and can deal with groups of variables of different sizes.

Causal deep learning. Continuous optimization of causal structures has been used to boost performance in classification. In CASTLE (Kyono et al., 2020), structure learning is introduced as a regulariser for a deep learning classification model. This regulariser reconstructs only the most relevant causal features, leading to improved out-of-sample predictions. In SLAPS (Fatemi et al., 2021), the classification objective is supplemented with a self-supervised task that learns a graph of interactions between variables through a GNN. However, these works are focused on the supervised classification task, and they did not advance the performance of causal discovery methods. Causal discovery has also been used within models that predict the dynamics of interacting systems with deep neural networks (Kipf et al., 2018; Löwe et al., 2020). Unlike VICAUSE, these approaches are developed for time series with Granger causality.

Missing values imputation. The relevance of missing data in real-world problems has motivated a long history of research (Dempster et al., 1977; Rubin, 1976). A popular approach is to estimate the missing values based on the observed ones through different techniques (Scheffer, 2002). Here, we find popular methods such as missforest (Stekhoven & Bühlmann, 2012), which relies on Random Forest, and MICE (Buuren & Groothuis-Oudshoorn, 2010), which is based on Bayesian Ridge Regression. Also, the efficiency of amortized inference in generative models has motivated its use for missing values imputation. This is explored in Wu et al. (2018), although fully observed training data is required. This limitation is addressed in both Nazabal et al. (2020), where a simple zero-imputation strategy is used for partially observed data, and Ma et al. (2019), where a permutation invariant set encoder is utilized to directly handle missing values. VICAUSE also leverages amortized inference, although the imputation is informed by the discovered causal relationships through a GNN.

B. Synthetic experiment

We simulate fifteen synthetic datasets. To understand how the number of variables affects VICAUSE, we use $D = 5, 7, 9$ variables (five datasets for each value of D). For each simulated dataset, we first sample the true causal structure $\mathbf{G}$, see Fig. 2(a) for an example. Then, the dataset samples are obtained. Each variable is computed from its parents through a non-linear mapping based on the $\sin$ function, see Sec. C.1 and Fig. 5 in the appendix for additional details and a visualisation of the generated data, respectively. For each dataset, we simulate 5000 training and 1000 test samples.

	RMSE
Majority vote	0.5442±0.0032
Mean imputing	0.2206±0.0061
MICE	0.1361±0.0046
Missforest	0.1313±0.0025
PVAE	0.1407±0.0043
VICAUSE	**0.1196±0.0024**

Table 6: Imputation results for the synthetic experiment. Mean and standard error over fifteen datasets.

Imputation performance. VICAUSE outperforms the baselines in terms of imputation, and this is consistent across all datasets with different number of variables,

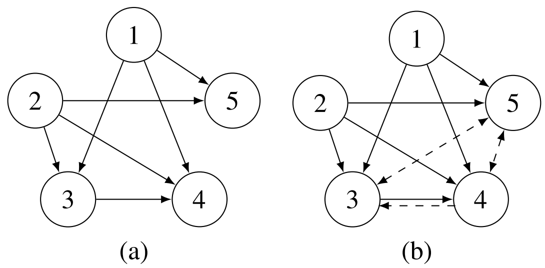

Figure 2: (a): Causal structure simulated for one of the synthetic datasets with 5 variables. (b): Graph predicted by VICAUSE (when the one on the left is used as the true one). VICAUSE predicts all the true relationships plus some additional ones (dashed edges).

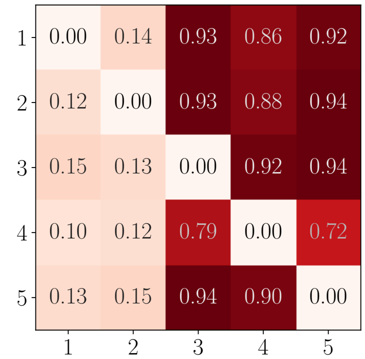

Figure 3: Probability of edges obtained by VICAUSE in the synthetic experiment. By using a 0.5 threshold, we get the predicted graph in Fig. 2(b). The item (i, j) refers to the probability of the edge $i \to j$.

	Adjacency			Orientation			Causal accuracy
	Recall	Precision	F_1-score	Recall	Precision	F_1-score	
PC	0.422±0.056	0.634±0.067	0.495±0.056	0.218±0.046	0.328±0.061	0.257±0.051	0.33±0.046
GES	0.452±0.044	0.569±0.036	0.491±0.038	0.249±0.046	0.305±0.053	0.270±0.049	0.364±0.045
NOTEARS (L)	0.193±0.028	0.443±0.059	0.265±0.036	0.149±0.023	0.367±0.060	0.209±0.032	0.149±0.023
NOTEARS (NL)	0.328±0.039	0.489±0.051	0.387±0.044	0.277±0.032	**0.417±0.043**	0.327±0.035	0.277±0.032
DAG-GNN	0.443±0.064	0.509±0.062	0.464±0.061	0.352±0.050	0.415±0.052	0.373±0.049	0.352±0.050
VICAUSE	**0.843±0.043**	**0.679±0.037**	**0.740±0.033**	**0.520±0.067**	0.414±0.058	**0.454±0.060**	**0.726±0.069**

Table 7: Causal discovery results for synthetic experiment (mean and std error over fifteen datasets).

see Table 6. The results split by number of variables are in the appendix, Table 8. Therefore, in addition to predicting the relationships between variables, VICAUSE exploits this information to obtain enhanced imputation.

Causal discovery performance. VICAUSE obtains better performance than the causality baselines, see Table 7. The results split by number of variables are in the appendix, Table 10. Notice that NOTEARS (NL) is slightly better in terms of orientation-precision, i.e. the orientation of the edges that it predicts is slightly more reliable. However, this is at the expense of a significantly lower capacity to detect true edges, see the recall and the trade-off between both (F_1-score). In this small synthetic experiment, it is possible to visually inspect the predicted graph. Fig. 3 shows the posterior probability of each edge (i.e. the estimated matrix $\mathbf{G}$) for the simulated dataset that uses the true graph in Fig. 2(a). By using a threshold of 0.5, we obtain the predicted graph in Fig. 2(b). We observe that all the true edges are captured by VICAUSE, which also predicts some additional edges. Some of them can be explained by looking at the relationships between the corresponding variables, recall Fig. 5 in the appendix. For instance, the one connecting the third and fifth variables can be explained through the linear relationship that exist between both variables.

C. Experimental details

Here we specify the complete experimental details for full reproducibility. We first provide all the details for the synthetic experiment (Sec. C.1). Then we explain the differences for the neuropathic pain and the Eedi topics experiments in Sec. C.2 and Sec. C.3, respectively.

C.1. Synthetic experiment

Data generation process. We first sample the underlying true causal structure. An edge from variable i to variable j is sampled with probability 0.5 if $i < j$, and probability 0 if $i \geq j$ (this ensures that the true causal structure is a DAG, which is just a standard scenario, and not a requirement for any of the compared algorithms). Then, we generate the data points. Root nodes (i.e. nodes with no parents, like variables 1 and 2 in Fig. 2(a) in the paper) are sampled from $\mathcal{N}(0, 1)$. Any other node v_i is obtained from its parents $\mathrm{Pa}(i)$ as $v_i = \sum_{j \in \mathrm{Pa}(i)} \sin(3v_j) + \varepsilon$, where $\varepsilon \to \mathcal{N}(0, 0.01)$ is a Gaussian noise. We use the $\sin$ function to induce non-linear relationships between variables. Notice that the 3-times factor inside the $\sin$

encourages that the whole period of the sin function is used (to favor non-linearity). As an example of the data generation process, Fig. 5 shows the pair plot for the dataset generated from the graph in Fig. 2(a) in the paper.

Model parameters. We start by specifying the parameters associated to the generative process. We use a prior probability $p_{ij} = 0.05$ in $\mathrm{p}(\mathbf{G})$ for all the edges. This favours sparse graphs, and can be adjusted depending on the problem at hand. The prior $\mathrm{p}(\mathbf{Z})$ is a standard Gaussian distribution, i.e. $\sigma_z^2 = 1$. This provides a standard regularisation for the latent space. The output noise is set to $\sigma_x^2 = 0.02$, which favours the accurate reconstruction of samples. As for the decoder, we perform $T = 3$ iterations of GNN message passing. All the MLPs in the decoder (i.e. MLP^f, MLP^b, MLP^{e2n} and g) have two linear layers with ReLU non-linearity. The dimensionality of the hidden layer, which is the dimensionality of each latent subspace, is 256. Regarding the encoder, it is given by a multi-head neural network that defines the mean and standard deviation of the latent representation. The neural network is a MLP with two standard linear layers with ReLu non-linearity. The dimension of the hidden layer is also 256. When using groups, there are as many such MLPs as groups. Finally, recall that the variational posterior $\mathrm{q}(\mathbf{G})$ is the product of independent Bernoulli distributions over the edges, with a probability $\mathbf{G}_{ij}$ to be estimated for each edge. These values are all initialised to $\mathbf{G}_{ij} = 0.5$.

Training hyperparameters. We use Adam optimizer with learning rate 0.001. We train during 300 epochs with a batch size of 100 samples. Each one of the two stages described in the two-step training takes half of the epochs (recall Sec. 2.3 in the paper). The percentage of data dropped during training for each instance is sampled from a uniform distribution. When doing the reparametrization trick (i.e. when sampling from $\mathbf{Z}_n$), we obtain 1 sample during training (100 samples in test time). For the Gumbel-softmax sample, we use a temperature $\tau = 0.5$. The rest of hyperparameters are the standard ones in `torch.nn.functional.gumbel_softmax`, in particular we use soft samples. To compute the DAG regulariser $\mathcal{R}(\mathbf{G})$, we use the exponential matrix implementation in `torch.matrix_exp`. This is in contrast to previous approaches, which resort to approximations (Zheng et al., 2018; Yu et al., 2019). When applying the encoder, missing values in the training data are replaced with the value 0 (continuous variables).

Baselines details. Regarding the causality baselines, we ran both PC and GES with the Causal Command tool offered by the Center for Causal Discovery `https://www.ccd.pitt.edu/tools/`. We used the default parameters in each case (i.e. disc-bic-score for GES and cg-lr-test for PC). NOTEARS (L), NOTEARS (NL) and DAG-GNN were run with the code provided by the authors in GitHub: `https://github.com/xunzheng/notears` (NOTEARS (L) and NOTEARS (NL)) and `https://github.com/fishmoon1234/DAG-GNN` (DAG-GNN). In all cases, we used the default parameters proposed by the authors. Regarding the imputation baselines, Majority Vote and Mean Imputing were implemented in Python. MICE and Missforest were used from Scikit-learn library with default parameters `https://scikit-learn.org/stable/modules/generated/sklearn.impute.IterativeImputer.html#sklearn.impute.IterativeImputer`. For PVAE, we use the authors implementation with their proposed parameters, see `https://github.com/microsoft/EDDI`.

Other experimental details. VICAUSE is implemented in PyTorch. The code is available in the supplementary material. The experiments were run using a local Tesla K80 GPU and a compute cluster provided by Azure Machine Learning platform with NVIDIA Tesla V100 GPU.

C.2. Neuropathic pain experiment

Data generation process. We use the Neuropathic Pain Diagnosis Simulator in `https://github.com/TURuibo/Neuropathic-Pain-Diagnosis-Simulator`. We simulate five datasets with 1500 samples, and split each one randomly in 1000 training and 500 test samples. These five datasets are used for the five independent runs in Sec. 3.1 in the paper.

Model and training hyperparameters. Most of the hyperparameters are identical to the synthetic experiment. However, in this case we have to deal with 222 variables, many more than before. In particular, the number of possible edges is 49062. Therefore, we reduce the dimensionality of each latent subspace to 32, the batch size to 25, and the amount of test samples for $\mathbf{Z}_n$ to 10 (in training we still use 1 as before). Moreover, we reduce the initial posterior probability for each edge to 0.2. The reason is that, for 0.5 initialization, the DAG regulariser $\mathcal{R}(\mathbf{G})$ evaluates to extremely high and unstable values for the 222×222 matrix. Since this is a more complex problem (no synthetic generation), we run the algorithm for 1000 epochs. When applying the encoder, missing values in the training data are replaced with the value 0.5 (binary variables).

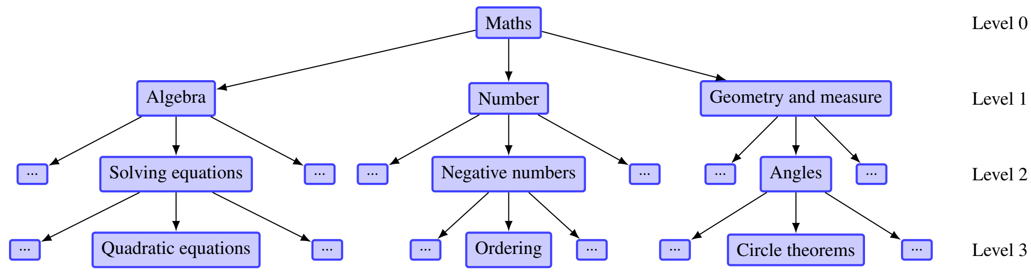

Figure 4: Hierarchy of topics in the Eedi topics dataset. All the questions are related to maths, which is the level 0 topic. The number of topics at levels 1, 2 and 3 is 3, 25 and 57, respectively. Each question is associated to only one topic at level 3 (and therefore to only one topic at any higher level).

	Number of variables			Average
	5	7	9	
Majority vote	0.5507±0.0056	0.5391±0.0050	0.5427±0.0050	0.5442±0.0032
Mean imputing	0.2351±0.0104	0.2124±0.0112	0.2143±0.0064	0.2206±0.0061
MICE	0.1352±0.0044	0.1501±0.0095	0.1230±0.0025	0.1361±0.0046
Missforest	0.1279±0.0040	0.1403±0.0030	0.1258±0.0022	0.1313±0.0025
PVAE	0.1324±0.0048	0.1536±0.0095	0.1360±0.0019	0.1407±0.0043
VICAUSE	**0.1146±0.0026**	**0.1251±0.0055**	**0.1191±0.0015**	**0.1196±0.0024**

Table 8: Imputation results for the synthetic experiment in terms of RMSE (not aggregating by number of variables, $D = 5, 7, 9$). The values are the mean and standard error over five different simulations.

C.3. Eedi topics experiment

Data generation process. The real-world Eedi topics dataset contains 6147 samples. We use a random 80%-10%-10% train-validation-test split. The validation set is used to perform Bayesian Optimization (BO) as described below. The five runs reported in the experimental section come from different initializations for the model parameters.

Model and training hyperparameters. Here, we follow the same specifications as in the neuropathic pain dataset. The only difference is that we perform BO for three hyperparameters: the dimensionality of the latent subspaces, the number of GNN message passing iterations, and the learning rate. The possible choices for each hyperparameter are $\{5, 10, 15, 20, 25, 30, 35, 40, 45, 50\}$, $\{3, 5, 8, 10, 12, 14, 16, 18, 20\}$, and $\{10^{-4}, 10^{-3}, 10^{-2}\}$ respectively. We perform 39 runs of BO with the hyperdrive package in Azure Machine Learning platform `https://docs.microsoft.com/en-us/python/api/azureml-train-core/azureml.train.hyperdrive?view=azure-ml-py`. We use validation accuracy as the target metric. The best configuration obtained through BO was 15, 8 and 10^{-4}, respectively.

Baselines details. As explained in Sec. 3.2 in the paper, in this experiment DAG-GNN is adapted to deal with missing values and groups of arbitrary size. For the former, we adapt the DAG-GNN code to replace missing values with 0.5 constant value, as in VICAUSE. For the latter, we also follow VICAUSE and use as many different neural networks as groups (recall Sec. 2.4 in the paper), all of them with the same architecture as the one used in the original code (`https://github.com/fishmoon1234/DAG-GNN`).

Other experimental details. The list of relationships found by VICAUSE (Table 11) and DAG-GNN (Table 12) aggregates the relationships obtained in the five independent runs. This is done by setting a threshold of 0.35 on the posterior probability of edge (which is initialized to 0.2) and considering the union for the different runs. This resulted in 50 relationships for VICAUSE and 57 for DAG-GNN. For *Random*, we simulated 50 random relationships. Also, the probability reported in the first column of Table 11 is the average of the probabilities obtained for that relationship in the five different runs.

D. Complementary results and figures

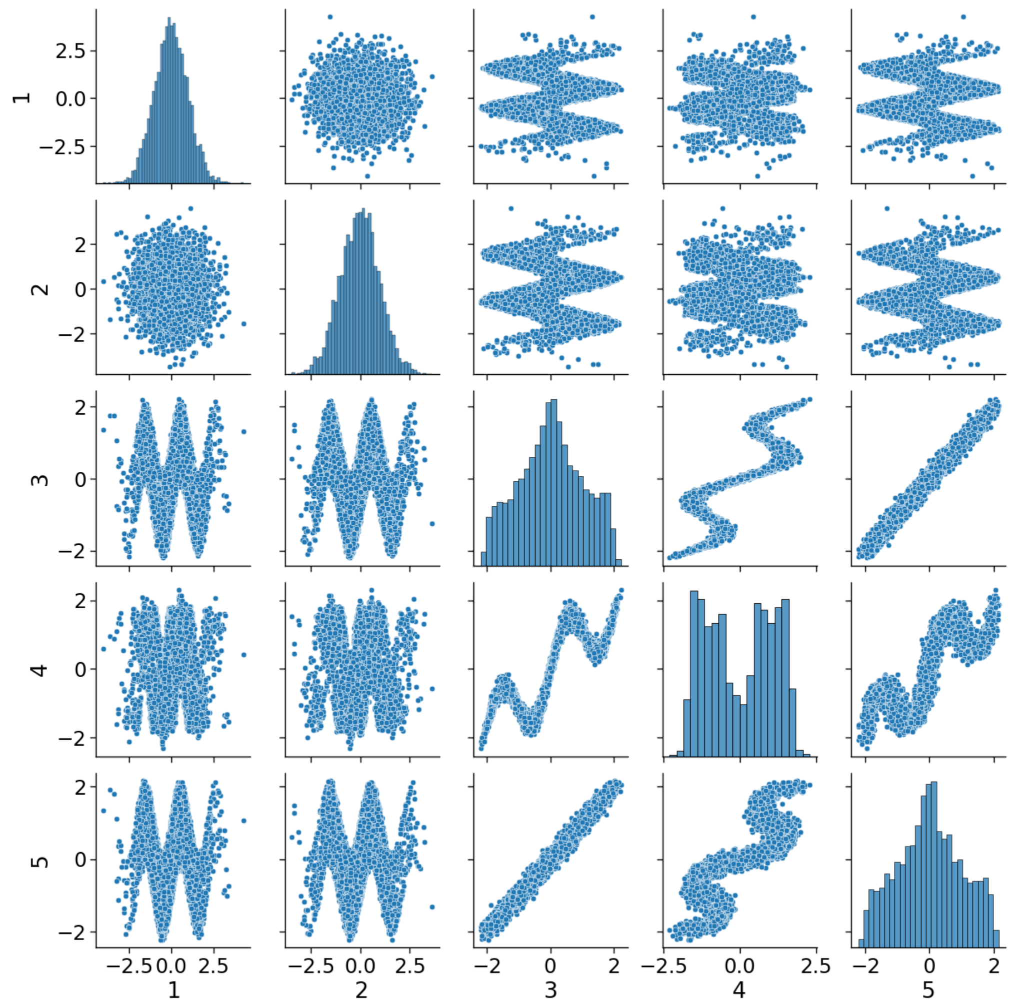

Figure 5: Pair-plot for the dataset generated from the graph in Fig. 2(a) in the paper. We observe different type of relationships between variables, including non-linear ones.

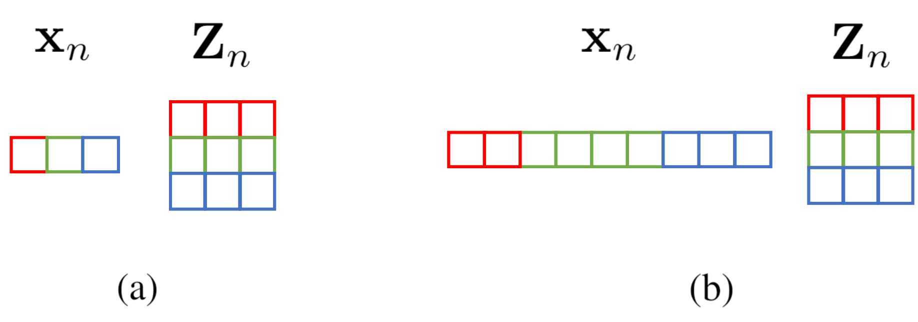

Figure 6: Structured latent space. (a) At the level of variables. Each variable in $\mathbf{x}_n$ (each color) has its own latent subspace, which is given by a row in $\mathbf{Z}_n$. (b) At the level of groups of variables. Here, each group of variables (each color) has its own latent subspace, which is given by a row in $\mathbf{Z}_n$.

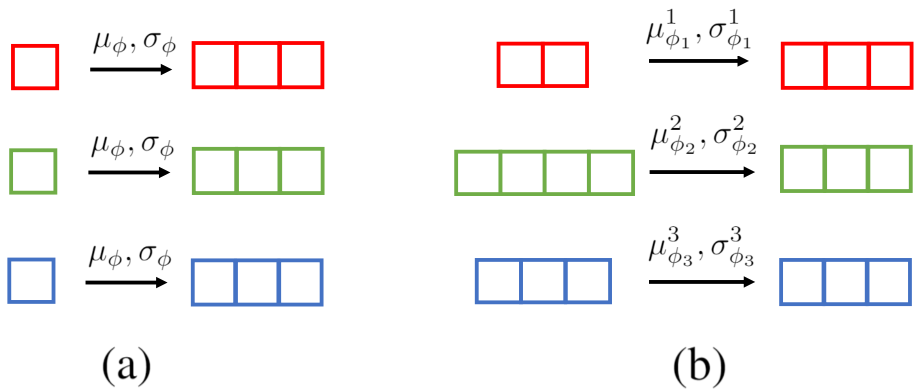

Figure 7: The encoder respects the structure of the latent space. (a) At the level of variables. All the variables use the same encoding functions. (b) At the level of groups of variables. Each group of variables uses different encoding functions.

Index	Topic name
1	Decimals
2	Factors, Multiples and Primes
3	Fractions, Decimals and Percentage Equivalence
4	Fractions
5	Indices, Powers and Roots
6	Negative Numbers
7	Straight Line Graphs
8	Inequalities
9	Sequences
10	Writing and Simplifying Expressions
11	Angles
12	Circles
13	Co-ordinates
14	Construction, Loci and Scale Drawing
15	Symmetry
16	Units of Measurement
17	Volume and Surface Area
18	Basic Arithmetic
19	Factorising
20	Solving Equations
21	Formula
22	2D Names and Properties of Shapes
23	Perimeter and Area
24	Similarity and Congruency
25	Transformations

Table 9: Mapping between indexes for row/column names in Table 14 and Table 16 and the actual level-2 topic names.

		Adjacency			Orientation			Causal Accuracy
		Recall	Precision	F_1-score	Recall	Precision	F_1-score	
5	PC	0.464±0.099	0.610±0.117	0.526±0.107	0.364±0.098	0.490±0.127	0.416±0.111	0.436±0.076
	GES	0.414±0.067	0.507±0.071	0.446±0.065	0.257±0.103	0.327±0.117	0.285±0.110	0.368±0.072
	NOTEARS (L)	0.186±0.052	0.400±0.089	0.247±0.063	0.119±0.049	0.300±0.110	0.167±0.065	0.119±0.049
	NOTEARS (NL)	0.331±0.057	0.470±0.078	0.384±0.065	0.264±0.047	0.370±0.053	0.304±0.049	0.264±0.047
	DAG-GNN	0.381±0.130	0.433±0.121	0.399±0.127	0.231±0.067	0.283±0.073	0.249±0.068	0.231±0.067
	VICAUSE	0.971±0.026	0.598±0.059	0.730±0.047	0.574±0.111	0.356±0.085	0.432±0.093	0.971±0.026
7	PC	0.396±0.110	0.639±0.154	0.468±0.112	0.113±0.043	0.193±0.083	0.134±0.050	0.324±0.088
	GES	0.429±0.087	0.647±0.042	0.501±0.076	0.208±0.067	0.279±0.081	0.235±0.073	0.345±0.091
	NOTEARS (L)	0.222±0.059	0.526±0.124	0.309±0.078	0.176±0.041	0.436±0.109	0.248±0.058	0.176±0.041
	NOTEARS (NL)	0.315±0.094	0.513±0.119	0.382±0.104	0.269±0.074	0.453±0.105	0.330±0.084	0.269±0.074
	DAG-GNN	0.396±0.109	0.539±0.123	0.446±0.111	0.318±0.082	0.445±0.102	0.361±0.085	0.318±0.082
	VICAUSE	0.813±0.088	0.694±0.057	0.725±0.053	0.559±0.134	0.447±0.070	0.480±0.089	0.701±0.103
9	PC	0.406±0.072	0.654±0.053	0.491±0.060	0.176±0.020	0.302±0.045	0.219±0.024	0.229±0.041
	GES	0.514±0.065	0.553±0.050	0.525±0.049	0.282±0.057	0.308±0.068	0.291±0.061	0.379±0.069
	NOTEARS (L)	0.172±0.026	0.403±0.076	0.238±0.036	0.151±0.023	0.366±0.082	0.211±0.035	0.151±0.023
	NOTEARS (NL)	0.338±0.042	0.485±0.053	0.394±0.045	0.297±0.034	0.429±0.044	0.347±0.036	0.297±0.034
	DAG-GNN	0.551±0.067	0.554±0.053	0.547±0.057	0.508±0.061	0.516±0.054	0.508±0.055	0.508±0.061
	VICAUSE	0.705±0.061	0.615±0.042	0.652±0.044	0.356±0.092	0.297±0.065	0.322±0.076	0.526±0.081

Table 10: Causality results for the synthetic experiment (not aggregating by number of variables, $D = 5, 7, 9$). The values are the mean and standard error over five different simulations.

Prob	Topic 1 (from)	Topic 2 (to)	Adj1	Ori1	Adj2	Ori2
0.44	Adding and Subtracting Negative Numbers [Negative Numbers] [Number]	Ordering Negative Numbers [Negative Numbers] [Number]	5	1	5	1
0.38	Mental Multiplication and Division [Basic Arithmetic] [Number]	Multiples and Lowest Common Multiple [Factors, Multiples and Primes] [Number]	5	5	5	5
0.37	Mental Multiplication and Division [Basic Arithmetic] [Number]	Factors and Highest Common Factor [Factors, Multiples and Primes] [Number]	5	5	5	5
0.37	Adding and Subtracting Negative Numbers [Negative Numbers] [Number]	Multiples and Lowest Common Multiple [Factors, Multiples and Primes] [Number]	2	2	2	1
0.36	Adding and Subtracting Negative Numbers [Negative Numbers] [Number]	Factors and Highest Common Factor [Factors, Multiples and Primes] [Number]	2	1	2	1
0.35	Mental Multiplication and Division [Basic Arithmetic] [Number]	Place Value [Basic Arithmetic] [Number]	4	2	4	2
0.35	Mental Multiplication and Division [Basic Arithmetic] [Number]	BIDMAS [Basic Arithmetic] [Number]	5	5	5	5
0.35	Adding and Subtracting Negative Numbers [Negative Numbers] [Number]	Simplifying Expressions by Collecting Like Terms [Writing and Simplifying Expressions] [Algebra]	5	5	5	5
0.35	Adding and Subtracting Negative Numbers [Negative Numbers] [Number]	BIDMAS [Basic Arithmetic] [Number]	4	4	4	3
0.35	Adding and Subtracting Negative Numbers [Negative Numbers] [Number]	Multiplying and Dividing Negative Numbers [Negative Numbers] [Number]	4	4	5	4
0.35	Mental Multiplication and Division [Basic Arithmetic] [Number]	Squares, Cubes, etc [Indices, Powers and Roots] [Number]	5	5	5	5
0.34	Factors and Highest Common Factor [Factors, Multiples and Primes] [Number]	Mental Multiplication and Division [Basic Arithmetic] [Number]	5	1	5	1
0.34	Basic Angle Facts (straight line, opposite, around a point, etc) [Angles] [Geometry and Measure]	Angle Facts with Parallel Lines [Angles] [Geometry and Measure]	4	4	4	4
0.34	Multiplying and Dividing Negative Numbers [Negative Numbers] [Number]	Adding and Subtracting Negative Numbers [Negative Numbers] [Number]	4	2	5	2
0.34	Writing Expressions [Writing and Simplifying Expressions] [Algebra]	Simplifying Expressions by Collecting Like Terms [Writing and Simplifying Expressions] [Algebra]	5	2	5	2
0.34	Adding and Subtracting Negative Numbers [Negative Numbers] [Number]	Squares, Cubes, etc [Indices, Powers and Roots] [Number]	2	2	2	2
0.33	Ordering Negative Numbers [Negative Numbers] [Number]	Adding and Subtracting Negative Numbers [Negative Numbers] [Number]	5	5	5	5
0.33	Basic Angle Facts (straight line, opposite, around a point, etc) [Angles] [Geometry and Measure]	Measuring Angles [Angles] [Geometry and Measure]	3	2	5	2
0.33	Simplifying Expressions by Collecting Like Terms [Writing and Simplifying Expressions] [Algebra]	Writing Expressions [Writing and Simplifying Expressions] [Algebra]	4	4	4	4
0.33	Measuring Angles [Angles] [Geometry and Measure]	Basic Angle Facts (straight line, opposite, around a point, etc) [Angles] [Geometry and Measure]	3	3	5	3
0.33	Adding and Subtracting Negative Numbers [Negative Numbers] [Number]	Place Value [Basic Arithmetic] [Number]	4	1	4	1
0.33	Adding and Subtracting Negative Numbers [Negative Numbers] [Number]	Prime Numbers and Prime Factors [Factors, Multiples and Primes] [Number]	2	2	2	1
0.33	Multiplying and Dividing Negative Numbers [Negative Numbers] [Number]	BIDMAS [Basic Arithmetic] [Number]	4	4	4	4
0.32	Factors and Highest Common Factor [Factors, Multiples and Primes] [Number]	BIDMAS [Basic Arithmetic] [Number]	3	2	3	2
0.32	Mental Multiplication and Division [Basic Arithmetic] [Number]	Prime Numbers and Prime Factors [Factors, Multiples and Primes] [Number]	5	5	5	5
0.32	Adding and Subtracting Negative Numbers [Negative Numbers] [Number]	Mental Multiplication and Division [Basic Arithmetic] [Number]	2	1	2	1
0.32	Factors and Highest Common Factor [Factors, Multiples and Primes] [Number]	Multiples and Lowest Common Multiple [Factors, Multiples and Primes] [Number]	3	3	3	3
0.32	Linear Equations [Solving Equations] [Algebra]	Substitution into Formula [Formula] [Algebra]	4	2	4	2
0.32	Factors and Highest Common Factor [Factors, Multiples and Primes] [Number]	Squares, Cubes, etc [Indices, Powers and Roots] [Number]	3	2	3	2
0.32	Angle Facts with Parallel Lines [Angles] [Geometry and Measure]	Basic Angle Facts (straight line, opposite, around a point, etc) [Angles] [Geometry and Measure]	4	2	4	2
0.32	Simplifying Expressions by Collecting Like Terms [Writing and Simplifying Expressions] [Algebra]	Substitution into Formula [Formula] [Algebra]	2	2	2	2
0.32	Writing Expressions [Writing and Simplifying Expressions] [Algebra]	Substitution into Formula [Formula] [Algebra]	4	3	4	3
0.32	Mental Multiplication and Division [Basic Arithmetic] [Number]	Time [Units of Measurement] [Geometry and Measure]	4	4	4	4
0.32	Multiplying and Dividing Negative Numbers [Negative Numbers] [Number]	Ordering Negative Numbers [Negative Numbers] [Number]	4	2	4	2
0.32	Adding and Subtracting Negative Numbers [Negative Numbers] [Number]	Substitution into Formula [Formula] [Algebra]	5	5	5	5
0.32	Multiplying and Dividing Negative Numbers [Negative Numbers] [Number]	Prime Numbers and Prime Factors [Factors, Multiples and Primes] [Number]	2	1	2	1
0.31	Factors and Highest Common Factor [Factors, Multiples and Primes] [Number]	Prime Numbers and Prime Factors [Factors, Multiples and Primes] [Number]	5	5	5	5
0.31	Basic Angle Facts (straight line, opposite, around a point, etc) [Angles] [Geometry and Measure]	Types, Naming and Estimating [Angles] [Geometry and Measure]	4	2	5	2
0.31	Ordering Negative Numbers [Negative Numbers] [Number]	Multiplying and Dividing Negative Numbers [Negative Numbers] [Number]	4	4	4	4
0.31	Substitution into Formula [Formula] [Algebra]	Writing Expressions [Writing and Simplifying Expressions] [Algebra]	4	3	4	3
0.31	Adding and Subtracting Negative Numbers [Negative Numbers] [Number]	Writing Expressions [Writing and Simplifying Expressions] [Algebra]	2	2	2	1
0.31	BIDMAS [Basic Arithmetic] [Number]	Place Value [Basic Arithmetic] [Number]	4	2	4	1
0.31	Multiples and Lowest Common Multiple [Factors, Multiples and Primes] [Number]	Mental Multiplication and Division [Basic Arithmetic] [Number]	4	2	4	2
0.31	Multiplying and Dividing Negative Numbers [Negative Numbers] [Number]	Factors and Highest Common Factor [Factors, Multiples and Primes] [Number]	4	2	4	2
0.30	Simplifying Expressions by Collecting Like Terms [Writing and Simplifying Expressions] [Algebra]	Multiplying and Dividing Negative Numbers [Negative Numbers] [Number]	2	2	1	1
0.30	Adding and Subtracting Negative Numbers [Negative Numbers] [Number]	Time [Units of Measurement] [Geometry and Measure]	2	2	2	2
0.30	Ordering Negative Numbers [Negative Numbers] [Number]	Substitution into Formula [Formula] [Algebra]	3	3	2	2
0.30	Adding and Subtracting Negative Numbers [Negative Numbers] [Number]	Angles in Polygons [Angles] [Geometry and Measure]	1	1	1	1
0.30	Factors and Highest Common Factor [Factors, Multiples and Primes] [Number]	Place Value [Basic Arithmetic] [Number]	3	2	3	1
0.28	Simplifying Expressions by Collecting Like Terms [Writing and Simplifying Expressions] [Algebra]	Mental Multiplication and Division [Basic Arithmetic] [Number]	2	1	2	1

Table 11: Full list of relationships found by VICAUSE in the Eedi topics dataset. Each row refers to one relationship (one edge). From left to right, the columns are the posterior probability of the edge, the sending node (topic), the receiving node (topic), and the adjacency and orientation evaluations from each expert. For each topic, the brackets contain its parent level 2 and level 1 topics.

Topic 1 (From)	Topic 2 (To)	Adj1	Ori1	Adj2	Ori2
Missing Lengths [Perimeter and Area] [Geometry and Measure]	Midpoint Between Two Co-ordinates [Co-ordinates] [Algebra]	4	4	4	5
Construct Triangle [Construction, Loci and Scale Drawing] [Geometry and Measure]	Place Value [Basic Arithmetic] [Number]	1	1	1	1
Squares, Cubes, etc [Indices, Powers and Roots] [Number]	Volume of Prisms [Volume and Surface Area] [Geometry and Measure]	4	5	5	4
Converting between Fractions and Percentages [Fractions, Decimals and Percentage Equivalence] [Number]	Volume of Prisms [Volume and Surface Area] [Geometry and Measure]	1	1	1	1
Angles in Triangles [Angles] [Geometry and Measure]	Parts of a Circle [Circles] [Geometry and Measure]	1	1	1	1
Types, Naming and Estimating [Angles] [Geometry and Measure]	Angle Facts with Parallel Lines [Angles] [Geometry and Measure]	4	5	5	5
Mental Multiplication and Division [Basic Arithmetic] [Number]	Measuring Angles [Angles] [Geometry and Measure]	1	1	1	1
Angles in Polygons [Angles] [Geometry and Measure]	Compound Area [Perimeter and Area] [Geometry and Measure]	1	1	1	1
Squares, Cubes, etc [Indices, Powers and Roots] [Number]	Solving Linear Inequalities [Inequalities] [Algebra]	2	1	3	1
Construct Triangle [Construction, Loci and Scale Drawing] [Geometry and Measure]	Solving Linear Inequalities [Inequalities] [Algebra]	1	1	1	1
Written Multiplication [Basic Arithmetic] [Number]	Translation and Vectors [Transformations] [Geometry and Measure]	1	1	1	1
Enlargement [Transformations] [Geometry and Measure]	Reflection [Transformations] [Geometry and Measure]	5	2	5	3
Rotation [Transformations] [Geometry and Measure]	Reflection [Transformations] [Geometry and Measure]	4	3	5	2
Construct Angle and Line Bisectors [Construction, Loci and Scale Drawing] [Geometry and Measure]	Length Scale Factors in Similar Shapes [Similarity and Congruency] [Geometry and Measure]	1	1	2	1
Angles in Triangles [Angles] [Geometry and Measure]	Properties of Quadrilaterals [2D Names and Properties of Shapes] [Geometry and Measure]	4	3	5	3
Naming Co-ordinates in 2D [Co-ordinates] [Algebra]	Properties of Quadrilaterals [2D Names and Properties of Shapes] [Geometry and Measure]	1	1	3	1
Adding and Subtracting Negative Numbers [Negative Numbers] [Number]	Properties of Quadrilaterals [2D Names and Properties of Shapes] [Geometry and Measure]	1	1	1	1
Construct Angle and Line Bisectors [Construction, Loci and Scale Drawing] [Geometry and Measure]	Properties of Quadrilaterals [2D Names and Properties of Shapes] [Geometry and Measure]	1	1	1	1
Written Multiplication [Basic Arithmetic] [Number]	Perimeter [Perimeter and Area] [Geometry and Measure]	2	1	2	1
Basic Angle Facts (straight line, opposite, around a point, etc) [Angles] [Geometry and Measure]	Perimeter [Perimeter and Area] [Geometry and Measure]	2	1	4	1
Naming Co-ordinates in 2D [Co-ordinates] [Algebra]	Area of Simple Shapes [Perimeter and Area] [Geometry and Measure]	1	1	1	1
Types, Naming and Estimating [Angles] [Geometry and Measure]	Writing Expressions [Writing and Simplifying Expressions] [Algebra]	1	1	1	1
Substitution into Formula [Formula] [Algebra]	Writing Expressions [Writing and Simplifying Expressions] [Algebra]	4	2	3	1
Naming Co-ordinates in 2D [Co-ordinates] [Algebra]	Linear Equations [Solving Equations] [Algebra]	1	1	1	1
Multiples and Lowest Common Multiple [Factors, Multiples and Primes] [Number]	Factorising into a Single Bracket [Factorising] [Algebra]	4	5	5	4
Linear Equations [Solving Equations] [Algebra]	Factorising into a Single Bracket [Factorising] [Algebra]	4	3	5	3
Converting between Fractions and Decimals [Fractions, Decimals and Percentage Equivalence] [Number]	BIDMAS [Basic Arithmetic] [Number]	1	1	1	1
Reflection [Transformations] [Geometry and Measure]	Place Value [Basic Arithmetic] [Number]	1	1	1	1
Length, Area and Volume Scale Factors [Similarity and Congruency] [Geometry and Measure]	Mental Multiplication and Division [Basic Arithmetic] [Number]	5	1	4	1
Naming Co-ordinates in 2D [Co-ordinates] [Algebra]	Midpoint Between Two Co-ordinates [Co-ordinates] [Algebra]	5	5	5	5
Enlargement [Transformations] [Geometry and Measure]	Time [Units of Measurement] [Geometry and Measure]	1	1	1	1
Rotational Symmetry [Symmetry] [Geometry and Measure]	Midpoint Between Two Co-ordinates [Co-ordinates] [Algebra]	1	1	2	1
Factors and Highest Common Factor [Factors, Multiples and Primes] [Number]	Horizontal and Vertical Lines [Straight Line Graphs] [Algebra]	1	1	1	1
Angles in Triangles [Angles] [Geometry and Measure]	Simplifying Expressions by Collecting Like Terms [Writing and Simplifying Expressions] [Algebra]	1	1	1	1
Naming Co-ordinates in 2D [Co-ordinates] [Algebra]	Simplifying Expressions by Collecting Like Terms [Writing and Simplifying Expressions] [Algebra]	1	1	1	1
Rotational Symmetry [Symmetry] [Geometry and Measure]	Simplifying Expressions by Collecting Like Terms [Writing and Simplifying Expressions] [Algebra]	1	1	1	1
Types, Naming and Estimating [Angles] [Geometry and Measure]	Simplifying Expressions by Collecting Like Terms [Writing and Simplifying Expressions] [Algebra]	1	1	1	1
Equivalent Fractions [Fractions] [Number]	Converting Mixed Number and Improper Fractions [Fractions] [Number]	5	5	5	5
Multiplying and Dividing with Decimals [Decimals] [Number]	Prime Numbers and Prime Factors [Factors, Multiples and Primes] [Number]	1	1	1	1
Construct Angle and Line Bisectors [Construction, Loci and Scale Drawing] [Geometry and Measure]	Prime Numbers and Prime Factors [Factors, Multiples and Primes] [Number]	1	1	1	1
Construct Angle [Construction, Loci and Scale Drawing] [Geometry and Measure]	Prime Numbers and Prime Factors [Factors, Multiples and Primes] [Number]	1	1	1	1
Types, Naming and Estimating [Angles] [Geometry and Measure]	Prime Numbers and Prime Factors [Factors, Multiples and Primes] [Number]	1	1	1	1
Angle Facts with Parallel Lines [Angles] [Geometry and Measure]	Factors and Highest Common Factor [Factors, Multiples and Primes] [Number]	1	1	1	1
Measuring Angles [Angles] [Geometry and Measure]	Factors and Highest Common Factor [Factors, Multiples and Primes] [Number]	1	1	1	1
Simplifying Expressions by Collecting Like Terms [Writing and Simplifying Expressions] [Algebra]	Adding and Subtracting Negative Numbers [Negative Numbers] [Number]	5	1	4	5
Squares, Cubes, etc [Indices, Powers and Roots] [Number]	Adding and Subtracting Negative Numbers [Negative Numbers] [Number]	1	1	3	1
Multiplying and Dividing Negative Numbers [Negative Numbers] [Number]	Adding and Subtracting Negative Numbers [Negative Numbers] [Number]	5	2	5	1
Ordering Negative Numbers [Negative Numbers] [Number]	Adding and Subtracting Negative Numbers [Negative Numbers] [Number]	5	5	5	5
Rotation [Transformations] [Geometry and Measure]	Adding and Subtracting Negative Numbers [Negative Numbers] [Number]	1	1	3	1
Reflection [Transformations] [Geometry and Measure]	Adding and Subtracting Negative Numbers [Negative Numbers] [Number]	1	1	3	1
Perimeter [Perimeter and Area] [Geometry and Measure]	Adding and Subtracting Negative Numbers [Negative Numbers] [Number]	1	1	1	1
Types, Naming and Estimating [Angles] [Geometry and Measure]	Adding and Subtracting Negative Numbers [Negative Numbers] [Number]	1	1	1	1
Converting between Fractions and Percentages [Fractions, Decimals and Percentage Equivalence] [Number]	Ordering Negative Numbers [Negative Numbers] [Number]	1	1	1	1
Construct Angle and Line Bisectors [Construction, Loci and Scale Drawing] [Geometry and Measure]	Ordering Negative Numbers [Negative Numbers] [Number]	1	1	1	1
Perimeter [Perimeter and Area] [Geometry and Measure]	Ordering Negative Numbers [Negative Numbers] [Number]	1	1	1	1
Construct Angle and Line Bisectors [Construction, Loci and Scale Drawing] [Geometry and Measure]	Time [Units of Measurement] [Geometry and Measure]	1	1	1	1
Written Multiplication [Basic Arithmetic] [Number]	BIDMAS [Basic Arithmetic] [Number]	5	4	5	3

Table 12: Full list of relationships found by DAG-GNN in the Eedi topics dataset. Each row refers to one relationship (one edge). From left to right, the columns are the sending node (topic), the receiving node (topic), and the adjacency and orientation evaluations from each expert. For each topic, the brackets contain its parent level 2 and level 1 topics.

Topic 1 (From)	Topic 2 (To)	Adj1	Ori1	Adj2	Ori2
Midpoint Between Two Co-ordinates [Co-ordinates] [Algebra]	Angles in Triangles [Angles] [Geometry and Measure]	1	1	1	1
Solving Linear Inequalities [Inequalities] [Algebra]	Enlargement [Transformations] [Geometry and Measure]	1	1	1	1
Squares, Cubes, etc [Indices, Powers and Roots] [Number]	Written Multiplication [Basic Arithmetic] [Number]	4	1	5	1
Substitution into Formula [Formula] [Algebra]	Written Multiplication [Basic Arithmetic] [Number]	4	1	3	1
Linear Sequences (nth term) [Sequences] [Algebra]	Mental Multiplication and Division [Basic Arithmetic] [Number]	5	1	5	2
Measuring Angles [Angles] [Geometry and Measure]	Construct Angle [Construction, Loci and Scale Drawing] [Geometry and Measure]	5	5	5	5
Dividing Fractions [Fractions] [Number]	Volume of Prisms [Volume and Surface Area] [Geometry and Measure]	2	2	2	2
Multiplying and Dividing Negative Numbers [Negative Numbers] [Number]	Parts of a Circle [Circles] [Geometry and Measure]	1	1	1	1
Types, Naming and Estimating [Angles] [Geometry and Measure]	Parts of a Circle [Circles] [Geometry and Measure]	2	2	2	1
Angles in Polygons [Angles] [Geometry and Measure]	Basic Angle Facts (straight line, opposite, around a point, etc) [Angles] [Geometry and Measure]	5	1	5	1
Angles in Polygons [Angles] [Geometry and Measure]	Compound Area [Perimeter and Area] [Geometry and Measure]	1	1	1	1
Length, Area and Volume Scale Factors [Similarity and Congruency] [Geometry and Measure]	Linear Sequences (nth term) [Sequences] [Algebra]	1	1	2	1
Substitution into Formula [Formula] [Algebra]	Rotation [Transformations] [Geometry and Measure]	1	1	1	1
Adding and Subtracting Negative Numbers [Negative Numbers] [Number]	Gradient Between Two Co-ordinates [Co-ordinates] [Algebra]	5	5	5	5
Compound Area [Perimeter and Area] [Geometry and Measure]	Reflection [Transformations] [Geometry and Measure]	1	1	1	1
BIDMAS [Basic Arithmetic] [Number]	Reflection [Transformations] [Geometry and Measure]	1	1	1	1
Adding and Subtracting Negative Numbers [Negative Numbers] [Number]	Properties of Quadrilaterals [2D Names and Properties of Shapes] [Geometry and Measure]	1	1	1	1
Compound Area [Perimeter and Area] [Geometry and Measure]	Properties of Quadrilaterals [2D Names and Properties of Shapes] [Geometry and Measure]	3	1	3	1
Rotational Symmetry [Symmetry] [Geometry and Measure]	Perimeter [Perimeter and Area] [Geometry and Measure]	3	1	3	1
Converting between Fractions and Percentages [Fractions, Decimals and Percentage Equivalence] [Number]	Area of Simple Shapes [Perimeter and Area] [Geometry and Measure]	1	1	1	1
Angles in Triangles [Angles] [Geometry and Measure]	Types, Naming and Estimating [Angles] [Geometry and Measure]	4	3	5	2
Length Scale Factors in Similar Shapes [Similarity and Congruency] [Geometry and Measure]	Types, Naming and Estimating [Angles] [Geometry and Measure]	1	1	1	1
Factorising into a Single Bracket [Factorising] [Algebra]	Types, Naming and Estimating [Angles] [Geometry and Measure]	1	1	1	1
Enlargement [Transformations] [Geometry and Measure]	BIDMAS [Basic Arithmetic] [Number]	1	1	1	1
Linear Sequences (nth term) [Sequences] [Algebra]	Time [Units of Measurement] [Geometry and Measure]	1	1	1	1
Horizontal and Vertical Lines [Straight Line Graphs] [Algebra]	Adding and Subtracting Negative Numbers [Negative Numbers] [Number]	1	1	1	1
Area of Simple Shapes [Perimeter and Area] [Geometry and Measure]	Multiplying and Dividing Negative Numbers [Negative Numbers] [Number]	1	1	1	1
Writing Expressions [Writing and Simplifying Expressions] [Algebra]	Factors and Highest Common Factor [Factors, Multiples and Primes] [Number]	1	1	1	1
Squares, Cubes, etc [Indices, Powers and Roots] [Number]	Midpoint Between Two Co-ordinates [Co-ordinates] [Algebra]	1	1	1	1
Writing Expressions [Writing and Simplifying Expressions] [Algebra]	Naming Co-ordinates in 2D [Co-ordinates] [Algebra]	1	1	1	1
BIDMAS [Basic Arithmetic] [Number]	Line Symmetry [Symmetry] [Geometry and Measure]	1	1	1	1
Simplifying Expressions by Collecting Like Terms [Writing and Simplifying Expressions] [Algebra]	Length, Area and Volume Scale Factors [Similarity and Congruency] [Geometry and Measure]	1	1	1	1
Converting Mixed Number and Improper Fractions [Fractions] [Number]	Horizontal and Vertical Lines [Straight Line Graphs] [Algebra]	1	1	1	1
Construct Angle and Line Bisectors [Construction, Loci and Scale Drawing] [Geometry and Measure]	Horizontal and Vertical Lines [Straight Line Graphs] [Algebra]	1	1	1	1
Multiplying and Dividing with Decimals [Decimals] [Number]	Simplifying Expressions by Collecting Like Terms [Writing and Simplifying Expressions] [Algebra]	1	1	1	1
Reflection [Transformations] [Geometry and Measure]	Simplifying Expressions by Collecting Like Terms [Writing and Simplifying Expressions] [Algebra]	1	1	1	1
Substitution into Formula [Formula] [Algebra]	Dividing Fractions [Fractions] [Number]	4	1	3	1
Factorising into a Single Bracket [Factorising] [Algebra]	Dividing Fractions [Fractions] [Number]	2	1	2	1
Fractions of an Amount [Fractions] [Number]	Multiplying Fractions [Fractions] [Number]	5	4	5	2
Time [Units of Measurement] [Geometry and Measure]	Converting Mixed Number and Improper Fractions [Fractions] [Number]	4	1	4	1
Length Scale Factors in Similar Shapes [Similarity and Congruency] [Geometry and Measure]	Converting Mixed Number and Improper Fractions [Fractions] [Number]	4	1	5	1
Place Value [Basic Arithmetic] [Number]	Equivalent Fractions [Fractions] [Number]	4	4	3	5
Reflection [Transformations] [Geometry and Measure]	Equivalent Fractions [Fractions] [Number]	1	1	1	1
Writing Expressions [Writing and Simplifying Expressions] [Algebra]	Fractions of an Amount [Fractions] [Number]	1	1	1	1
Dividing Fractions [Fractions] [Number]	Prime Numbers and Prime Factors [Factors, Multiples and Primes] [Number]	1	1	1	1
Adding and Subtracting Fractions [Fractions] [Number]	Prime Numbers and Prime Factors [Factors, Multiples and Primes] [Number]	1	1	1	1
Simplifying Expressions by Collecting Like Terms [Writing and Simplifying Expressions] [Algebra]	Multiples and Lowest Common Multiple [Factors, Multiples and Primes] [Number]	1	1	1	1
Adding and Subtracting Fractions [Fractions] [Number]	Multiples and Lowest Common Multiple [Factors, Multiples and Primes] [Number]	1	1	2	1
Mental Multiplication and Division [Basic Arithmetic] [Number]	Multiples and Lowest Common Multiple [Factors, Multiples and Primes] [Number]	5	5	5	5
Length Scale Factors in Similar Shapes [Similarity and Congruency] [Geometry and Measure]	BIDMAS [Basic Arithmetic] [Number]	1	1	1	1

Table 13: Full list of relationships found by *Random* in the Eedi topics dataset. Each row refers to one relationship (one edge). From left to right, the columns are the sending node (topic), the receiving node (topic), and the adjacency and orientation evaluations from each expert. For each topic, the brackets contain its parent level 2 and level 1 topics.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	2	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	5	0	0	1	6	0	0	0	2	1	0	0	0	0	1	0	4	0	0	2	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	1	0	0	0	2	0	0	0	0	0	0	0	1	0	0	2	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	3	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	3	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
21	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 14: How the 50 relationships found by VICAUSE are distributed across level 2 topics. The item (i, j) refers to edges in the direction $i \to j$. There are 18 relationships inside level 2 topics (36%). See Table 9 for a mapping between indexes shown here in row/column names and the actual level-2 topic names.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
3	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
4	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
6	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	3	0	0	0	1	0	0	0	3	1	1	0	0	0	0	0	0	0	0	0	1	2	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	1	0	1	1	0	0
14	0	2	0	0	0	1	0	1	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0	1	0
15	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0	1
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	2	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
25	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	2

Table 15: How the 57 relationships found by DAG-GNN are distributed across level 2 topics. The item (i, j) refers to edges in the direction $i \to j$. There are 8 relationships inside level 2 topics (14%). See Table 9 for a mapping between indexes shown here in row/column names and the actual level-2 topic names.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
4	0	3	0	1	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0
7	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
10	0	2	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0
11	0	0	0	0	0	0	0	0	0	0	2	1	0	1	0	0	0	0	0	0	0	0	1	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
16	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1
19	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
24	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0
25	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

Table 16: How the 50 relationships found by *Random* are distributed across level 2 topics. The item (i, j) refers to edges in the direction $i \rightarrow j$. There are 3 relationships inside level 2 topics (6%). See Table 9 for a mapping between indexes shown here in row/column names and the actual level-2 topic names.