Novel multi-omics deconfounding variational autoencoders can obtain meaningful disease subtyping

Zuqi Li[1,†], Sonja Katz[2,3,†], Vitor A.P. Martins dos Santos[4,5], David W. Fardo[6], Peter Claes[7,8,9], Edoardo Saccenti[3], Kristel Van Steen[1,10], Gennady V. Roshchupkin[2,11,*]

[1]BIO3 - Laboratory for Systems Medicine, Department of Human Genetics, KU Leuven, Leuven, Belgium.

[2]Department of Radiology and Nuclear Medicine, Erasmus MC, Rotterdam, The Netherlands.

[3]Laboratory of Systems and Synthetic Biology, Wageningen University & Research, Wageningen WE, The Netherlands.

[4]Laboratory of Bioprocess Engineering, Wageningen University & Research, Wageningen WE, The Netherlands.

[5]LifeGlimmer GmbH, Berlin, Germany.

[6]University of Kentucky, Lexington, The United States.

[7]Department of Human Genetics, KU Leuven, Leuven, Belgium.

[8]Medical Imaging Research Center, University Hospitals Leuven, Leuven, Belgium.

[9]Department of Electrical Engineering, ESAT-PSI, KU Leuven, Leuven, Belgium

[10]BIO3 - Laboratory for Systems Genetics, GIGA-R Medical Genomics, University of Liège, Liège, Belgium.

[11]Department of Epidemiology, Erasmus MC, Rotterdam, The Netherlands.

zuqi.li@kuleuven.be, sonja.katz@wur.nl

†Authors contributed equally *Corresponding author

Abstract

In the field of precision medicine, the use of multi-omics data for patient stratification holds great promise for delivering tailored treatments based on comprehensive individual biological profiles. The clinical potential of (multi-)omics data, however, faces significant limitations due to the presence of confounding factors in the data, such as noise from experimental procedures or other irrelevant biological signals. As confounding factors in the data potentially bias patient clustering, deconfounding deep learning frameworks, such as autoencoders, have been developed. Despite encouraging initial outcomes, these frameworks have seen limited validation when applied to clustering tasks using multi-omics data. Based on different deconfounding strategies, we propose four novel multi-omics variational autoencoder frameworks for clustering, capable of reducing confounding effects while preserving the integrity of true biological patterns. We therefore simulate artificial confounders of different effects (linear, non-linear and categorical) using gene expression and DNA methylation data from the TCGA pan-cancer study. We find the conditional multi-omics variational autoencoder to be clearly superior to other models in terms of stability, deconfounding potential, and in retrieving biologically-driven clustering structures. Conversely, the use of adversarial training for deconfounding proves challenging in terms of model optimization, leading to a poor deconfounding performance. Our study finds profound differences between autoencoder-based frameworks for clustering multi-omics data in the presence of confounders. The knowledge obtained from our experiments may aid in selecting an appropriate framework for multi-omics studies, ultimately facilitating better patient stratification for precision medicine.

Keywords: deep learning, autoencoder, multi-omics, genomics, confounders, deconfounding, bias, dimensionality reduction, patient stratification, clustering

1 Introduction

Unsupervised learning, in particular clustering, focuses on subgrouping individuals based on their intrinsic data structures, therefore playing an essential role in tasks like disease subtyping and patient stratification. In the realm of biology and medicine, where

large-scale multi-omics data, including genomics, transcriptomics, and epigenomics, is prevalent, deep learning models can enhance clustering algorithms. Their ability to reduce the dimensionality of complex data allows clustering algorithms to more effectively explore the heterogeneity between patients. Underscoring the utility of deep learning models, in particular autoencoders, in terms of data integration, dimensionality reduction, and handling a multitude of heterogeneous input data, Simidjievski et al. recently benchmarked various variational autoencoder models for multi-omics data [1].

Despite patient stratification with deep learning methods gaining traction in genomic data applications, they are often susceptible to external influences that are unrelated to the condition of interest. One severe limitation is the entanglement of biologically meaningful signals with variables unrelated to the inherent structure that one is interested in, i.e. technical artifacts, random noise from measurements, or other biological factors such as sex, ethnicity, and age (Figure 1a). These factors, referred to as confounders in the context of unsupervised learning, may cause clustering algorithms to form subgroups based on irrelevant signals, which may ultimately lead to spurious biological conclusions [2, 3].

Conventional strategies to combat confounders involve training linear regression per feature against the confounder and take the residual part during pre-processing [4] or adjustments like pruning predictive dimensions after model training [5]. Conditional variational autoencoders (cVAE) have been used to create normative models considering confounding variables, such as age, for neurological disorders [6]. Dincer et al. proposed adversarial training to derive expression embeddings devoid of confounding effects [7], expanded upon by the single-cell Generative Adversarial Network (scGAN) for batch effect removal [8]. Conversely, Liu et al. used a regularization term in the autoencoder's loss function to minimize correlation between latent embeddings and confounding bias [9]. Despite their methodological diversity, these methods have only been validated to work effectively on data from a single omics source and not particularly for patient clustering.

To address this gap, we developed four different multi-omics deconfounding approaches based on VAE, including i) removal of latent features correlated with confounders ii) a conditional variational autoencoder [6] iii) adversarial training [7, 8], and iv) adding a regularisation term to the loss function [9]. To objectively evaluate whether our

models can remove out-of-interest signals and find a patient clustering unbiased by confounding signals, we validated our models on gene expression and DNA methylation pan-cancer data from The Cancer Genome Atlas (TCGA) program which we augmented with artificial confounding effects. In total, we simulated four different confounding effects, including linear, non-linear, categorical, and a combination thereof, resembling realistic confounders such as age (linar, non-linear) [10, 11, 12] , BMI (non-linear) [13], or batch effects (categorical) [2, 7].

Our study uncovered profound differences between the four proposed deconfounding strategies in terms of versatility and their ability to retrieve an un-confounded patient clustering. Based on the results of our experiments, we were able to dissect strengths and limitations of each approach and provide readers with suggestions on selecting an appropriate framework fitting their purposes.

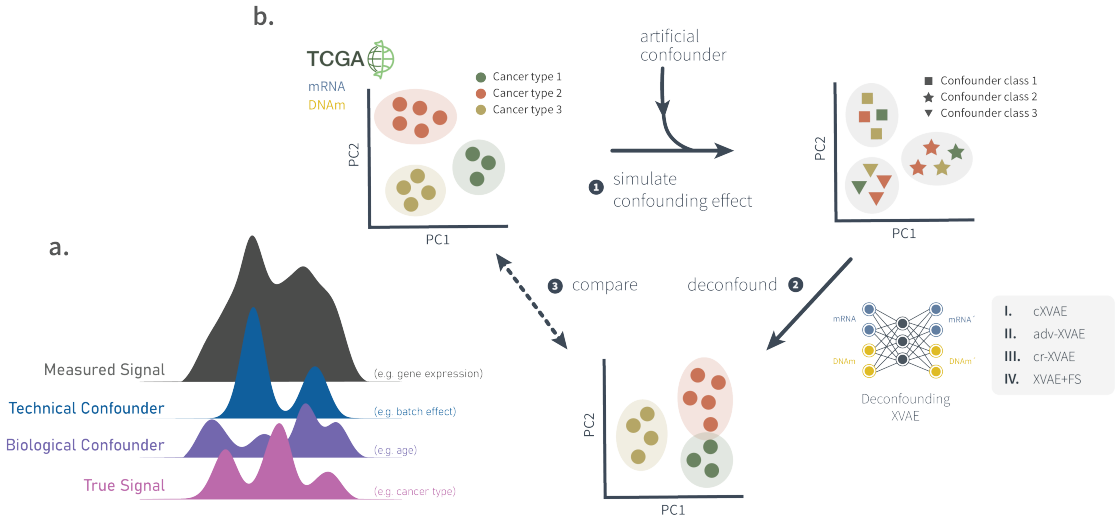

Figure 1: **a.** A simplified graphical representation of a measured signal (gray) which is a mix of independent sources such as the true signal (pink), a biological confounder (purple), and a technical confounder (blue). Note the difficulty of extracting the true signal from the measured additive signals. **b.** Graphical summary of the work conducted in this study.

A simplified graphical model of measured expression shown as mixture of a true signal, confounders of biological and non-biological origin and random noise. Note that this model shows neither possible connections between a true signal and confounders nor connections among confounders

2 Materials and methods

2.1 Data collection & preprocessing

This study utilized data collected within The Cancer Genome Atlas project (TCGA) [14]. Gene expressions (mRNA) of 4333 patients and DNA methylations (DNAm) of 2940 patients from six different cancer types, including BRCA, THCA, BLCA, LUSC, HNSC, and KIRC, were downloaded using the R package TCGAbiolinks [15]. The subsequent filtering step removed patients (i) with only a single data type available, (ii) with missing clinical metadata, (iii) of "american indian" or "alaska native" descent, and (iv) unknown tumor stage, resulting in a total of 2547 patients. The preprocessing of mRNA and DNAm data included the removal of probes (i) not shared across all cancer types, (ii) with missing values, and (iii) with 0 variance across all included patients, resulting in 58456 mRNA and 232088 DNAm variables. To reduce the number of input features to our models, we only considered the 2000 most variable probes for each data type, resulting in a final data set of 2547 patients and 4000 features. This reduction is with the same spirit of other clustering works on TCGA data [16].

2.2 Simulation of confounders

In an attempt to imitate common confounding scenarios in real-life clustering applications we simulated linear, squared, categorical confounders, and a combination thereof, resembling e.g. ageing [10, 11, 12], BMI [13], or batch effects [2, 7]. These confounders hinder the true or biologically meaningful clustering by intrinsically affecting the data structure in an unwanted way and possibly leading to a confounded clustering. Here we denote the mRNA data as $X_1 \in \mathbb{R}^{n \times p}$ and DNAm data as $X_2 \in \mathbb{R}^{n \times q}$, where n, p, q are the number of patients, gene expressions, and DNA methylations, respectively. We first rescaled every mRNA and DNAm feature to the range $[0, 1]$ to avoid large ratios between the raw feature and the confounding effect. A visualisation of all confounding effects can be found in the Supplementary Methods 6.2.

2.2.1 Linear effect

Firstly, we uniformly generated a random numeric confounder $c \in \mathbb{R}^n$ with discrete values $\{0, 1, 2, 3, 4, 5\}$, leading to a confounder clustering of six classes. Its linear effect

on each individual is $c+5$ and a random weight for each feature was multiplied with it:

$$X_1' = X_1 + E_1^{\text{linear}} = X_1 + (c+5) \otimes w_1 \tag{1}$$

$$X_2' = X_2 + E_2^{\text{linear}} = X_2 + (c+5) \otimes w_2 \tag{2}$$

, where $\otimes$ denotes the outer product between two vectors, and $w_1 \in \mathbb{R}^p \sim U(0, 0.1)$, $w_2 \in \mathbb{R}^q \sim U(0, 0.2)$. We chose the uniform distribution of w_1 to range from 0 to 0.1 so that the total linear effect would range from 0 to 1, having the same scale as w_1. We increased the upper bound of w_2 to 0.2 due to our observation that X_2 is less sensitive to linear confounders.

2.2.2 Non-linear effect

Non-linear effects were simulated in a similar way to linear effects. However, to mimic a non-linear confounder, as observed in, e.g. the significant quadratic association between body mass index and colon cancer risk [13], we considered adding an element-wise squared confounding effect c^2 on the features:

$$X_1' = X_1 + E_1^{\text{square}} = X_1 + c^2 \otimes w_1 \tag{3}$$

$$X_2' = X_2 + E_2^{\text{square}} = X_2 + c^2 \otimes w_2 \tag{4}$$

, where $w_1 \sim U(0, 0.04)$, $w_2 \sim U(0, 0.04)$. The distribution of w_1 and w_2 was also determined based on the scale of X_1 and X_2.

2.2.3 Categorical effect

A categorical confounding effect was achieved by shifting patients with the same confounder class to a distinctive direction in the feature space. More specifically, we first sampled six shifting vectors from $U(0, 1)$ corresponding to six different confounder classes, and patients were randomly assigned to each of the six categories. As a result, $C_1 \in \mathbb{R}^{n \times p}$ denotes the concatenation of shifting vectors of every patient for gene expression, while $C_2 \in \mathbb{R}^{n \times q}$ for DNA methylation, both are matrices. The categorical confounder is therefore the membership of all individuals in the six classes. A typical

example of categorical confounders for clustering could be batch effects caused by collecting data from different centers [2, 7]. Then the confounded features were created via:

$$X_1' = X_1 + E_1^{\text{categ}} = X_1 + \text{diag}(W) \cdot C_1 \tag{5}$$

$$X_2' = X_2 + E_2^{\text{categ}} = X_2 + \text{diag}(W) \cdot C_2 \tag{6}$$

, where $\text{diag}(\cdot)$ converts a vector into its corresponding diagonal matrix. Different from the case of a numeric confounder, the weight vector $w \in \mathbb{R}^n \sim U(0,1)$ of the categorical confounder indicates to what extent every patient was shifted so that patients would have various strength of association with their confounded class.

2.2.4 Combined effect

Real-life data analyses are likely affected by multiple confounders of different kinds, for instance, many cancer studies correct for age, age squared, education, etc. jointly in their models [11, 12]. Here we simulated a combined confounding effect of linear, non-linear, and categorical confounders as described below:

$$X_1' = X_1 + E_1^{\text{linear}} + E_1^{\text{square}} + E_1^{\text{categ}} \tag{7}$$

$$X_2' = X_2 + E_2^{\text{linear}} + E_2^{\text{square}} + E_2^{\text{categ}} \tag{8}$$

, where $E_1^{\text{linear}}, E_2^{\text{linear}}, E_1^{\text{square}}, E_2^{\text{square}}, E_1^{\text{categ}}, E_2^{\text{categ}}$ represent the second term in Formula (1-6), respectively.

2.3 Variational autoencoder for data integration (XVAE)

A variety of different VAE architectures exist for the purpose of data integration, as extensively compared by Simidjievski et al. [1]. In this study, we utilize one architecture recommended by the respective authors, namely the X-shaped Variational Autoencoder (XVAE) (Figure 2). This architecture merges the heterogeneous input data sources into a combined latent representation z by learning to reconstruct each source individually from the common representation. Here we consider only two data types of a single datapoint x_1 and x_2, and the loss function of XVAE is as follow:

$$L_{\text{XVAE}}(\phi, \theta; x_1, x_2) = -E_{z \sim q_\phi(z|x_1,x_2)}[\log p_\theta(x_1, x_2|z)] + \beta * \text{MMD}(q_\phi(z|x_1, x_2)||p(z)) \tag{9}$$

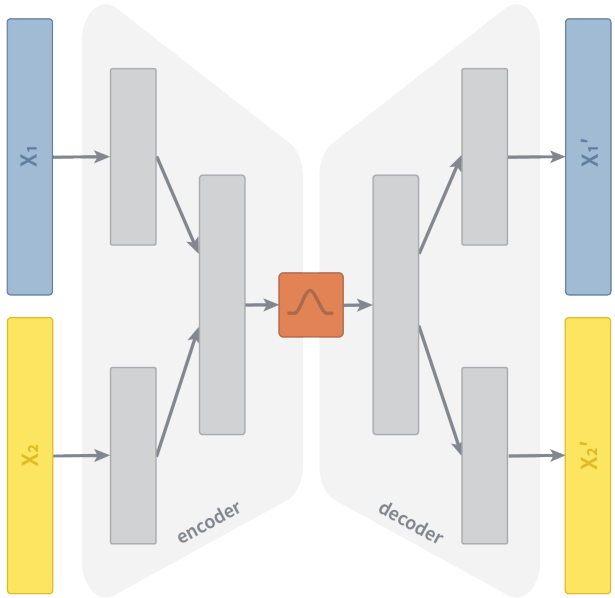

Figure 2: Schematic representation of an X-shaped Variational Autoencoder (XVAE). The two input layers (X_1, X_2) denote the two omics dimension used in this study, namely gene expression and DNA methylation. The encoder consists of contiguous hidden layers, each with fewer nodes. We design the encoder of XVAE with a total of 2 layers prior to the latent embedding. In the first hidden layer, the dimension of each input entity is reduced individually. In the second hidden layer, input entities get fused into a combined layer. The latent embedding (red) represents the bottleneck of the XVAE with the minimum number of nodes. The decoder reversely mirrors the layer structure of the encoder, with the final layer featuring the same number of nodes as the input layer as it attempts to reconstruct ($X_1{}'$, $X_2{}'$) the original input from the latent embedding.

, where $q_\phi(z|x_1, x_2)$ encodes the latent space as a probability distribution over the input variables (parameterised by ϕ) and $p_\theta(x_1, x_2|z)$ encodes the reconstruction of input variables as a probability distribution over the latent space (parameterised by θ). Following the originally proposed implementation, we use maximum mean discrepancy (MMD) as a regularization term to constrain the latent distribution q_ϕ to be a standard Gaussian distribution, balanced by the constant beta (β), which is set to 1 for all experiments.

For the activation function, we choose the Leaky Rectified Linear Activation Function (LeakyReLU) [17] in all layers except the output layer, which employs a sigmoid activa-

tion function. We are able to use the sigmoid function due to the normalisation of every mRNA and DNAm feature to the range $[0, 1]$ during preprocessing. To enhance training stability, batch normalisation layers are added to the encoder. We train every model for a fixed period of 150 epochs using an Adam optimiser [18] and a batch size of 64. To derive an optimal architecture for the XVAE base model, we carry out a hyperparameter search assessing the combination of (1) the number of nodes in hidden layers and latent embedding, (2) the rate of dropout layers, and (3) the weight initialisation. A more detailed description on autoencoders can be found in the Supplementary Methods 6.3.

2.4 Multi-omics deconfounding models

In this section we will first describe the use of linear regression for multi-omics confounder correction, which we deem the "baseline model" due to its wide popularity. Then, we outline the four XVAE-based deconfounding models proposed in this study. Throughout this section we denote the confounder value of a single data point as c.

2.4.1 LR+PCA: baseline model

Under the assumption that the effects of one or multiple confounders are linearly additive to the true signal of a feature, we build a linear model for the confounders against each mRNA or DNAm feature and then take the residual as the adjusted feature. In theory, the residual stands for the part of the feature that cannot be linearly explained by the confounders, i.e. it is deconfounded. Subsequently, the adjusted features from the two data types are concatenated and their dimensionality is reduced via PCA. We select the top 50 PCs explaining the most variance of data to keep the size of the embedding identical because it is the same number as the embedding size of every XVAE-based model. The 50 PCs explaining the most variance of data were considered for the final clustering, for which KMeans with 10 random initialisations was applied.

2.4.2 cXVAE: conditional XVAE

Conditional variational autoencoder (cVAE) [19] is a semi-supervised variation of VAE, which originally aims to fit the distribution of the high-dimensional output as a generative model conditioned on auxiliary variables. Lawry et al. proposed to achieve

deconfounding through a cVAE incorporating confounding variable information as auxiliary variables [6]. We extend this initial idea to be able to handle multi-omics data by replacing the originally proposed VAE with our XVAE model, resulting in a conditional X-shaped variational autoencoder architecture (cXVAE) (Figure 3A). We tested the integration of confounders at different levels of the cXAVE, including the input layer, the hidden layer that fuses multiple inputs, and the embedding. More details on cXVAE implementations can be found in the Supplementary Methods 6.4.

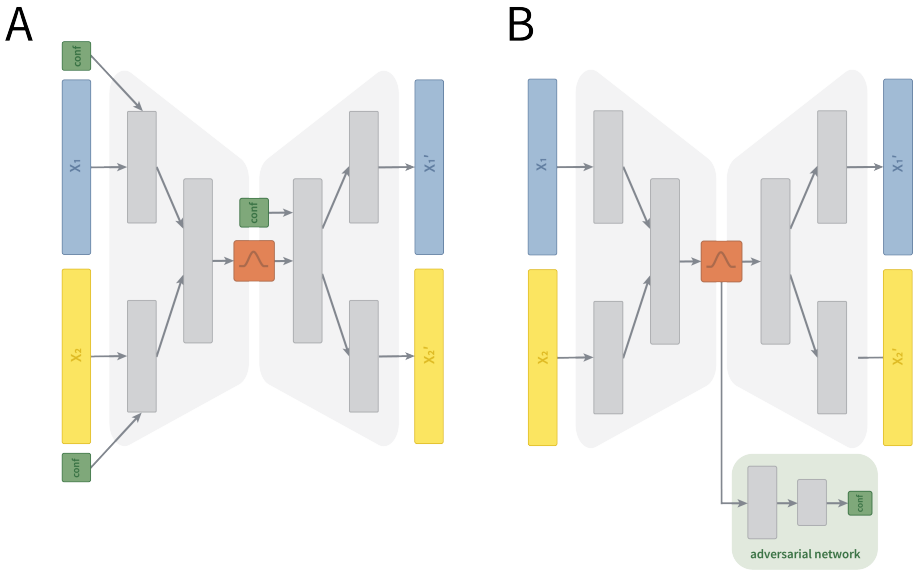

Figure 3: Schematic representation of (A) conditional variational autoencoder (cVAE) and (B) adversarial deconfounding XVAE (adv-XVAE). (A) Depicts the cXVAE implementation termed *input + embed* due to the addition of confounders (green) in the first layer of the encoder and decoder. (B) Depicts the adv-XVAE implementation termed *multiclass* due to the usage of only a single supervised adverarial network (light green) trained to predict confounders (green) using a multiclass prediction loss. X_1 and X_2 are the two omics dimension, namely gene expression and DNA methylation, while $X_1{}'$ and $X_2{}'$ denote their respective reconstruction. More details and visualisations of other implementation can be found in the Supplementary Material.

2.4.3 adv-XVAE: XVAE with adversarial training

The adversarial deconfounding autoencoder proposed by Dincer et al. [7] follows the idea of training two networks simultaneously - an autoencoder to generate a low dimensional embedding and an adversary multi-layer perceptron (MLP) trained to predict the confounder from said embedding (Figure 3B). By adversarially training the two networks, i.e. the autoencoder aims to generate an embedding which can not be used

for confounder prediction by the MLP, it aims at generating embeddings that can encode biological information without encoding any confounding signal. As the original framework can only handle a single data type, we adapt it to work with multi-omics input by replacing its autoencoder with XVAE architecture. As an adversarial network we implement a simple MLP with two layers. The first layer takes the generated embedding of the XVAE model as input and applies ReLU activation to condense it to 10 features. The output of the second layer features the number of confounders to be predicted, and uses the activation function fitting for predicting respective confounder; i.e. uses ReLU in case of continuous confounders, a sigmoid activation function in case of binary confounders, and softmax activation for multiple categorical confounders. Details on training procedure of the adv-XVAE can be found in Supplementary Methods 6.4.1.

2.4.4 cr-XVAE: XVAE with deconfounding regularization

Augmenting the loss function of deep learning models is an effective way to impose restrictions on the model or enforce learning of specific patterns. As an example, studies focused on disentangling the often highly correlated latent space of autoencoders impose constraints on the correlation between latent features by adding a penalty term to the loss function [9]. Inspired by this idea, we formulate a deconfounding regularization term aiming to reduce the degree of correlation between latent features and confounders. The regularized loss function becomes:

$$L_{\mathrm{cr-XVAE}}(\phi, \theta) = L_{\mathrm{XVAE}}(\phi, \theta) + f(z, c) \tag{10}$$

, where $f(z, c)$ denotes the joint association between latent features and confounders. More specifically, we choose two different association measurements, Pearson correlation and mutual information. Because Pearson correlation ranges from -1 (negatively correlated) to 1 (positively correlated) and both indicate strong relationship, we regularize only the magnitude of correlation by two methods, taking its absolute value or squared value. Because the confounder distribution needed for mutual information is normally unknown, we implement two methods to approximately compute mutual information as loss function, with differentiable histogram or kernel density estimate.

2.4.5 XVAE+FS: feature selection by removing correlated latent features

The removal of latent features correlated with confounders comes from the idea of *post hoc* interpretation of latent features [20]. To identify confounded latent features, we calculate the Pearson correlation between each latent variable and the confounder. For determining the threshold indicating which latent features are being removed from further analyses, we test two different approaches:

1. *p-value cutoff* - the Scipy implementation of the Pearson correlation yields a p-value indicating the probability that the distributions underlying the samples are uncorrelated. Latent features with a p-value < 0.05 are excluded from analyses.

2. *absolute correlation coefficient* - Pearson correlation measures the linear relationship between two variables. Latent features exhibiting an absolute Pearson correlation of more than 0.3 (weak correlation) or 0.5 (strong correlation) are excluded.

2.5 Consensus clustering

Different from the baseline linear regression model which adopts KMeans on the deconfounded features for clustering, we apply consensus clustering on the latent features of each VAE-based deconfounding model. Consensus clustering takes the advantage of random sampling in a VAE and it constructs a consensus matrix $A \in \mathbb{R}^{n \times n}$ from the individual clustering of each embedding sampled from the latent distributions [21]. We perform consensus clustering on the embeddings of the entire sample set, generating 50 embedding matrices, on each of which a k-means clustering is conducted. The values in the consensus matrix indicate the fraction of times that two data points are assigned to the same cluster in those 50 clustering solutions. Subsequently, each value is divided by the total number of times (50), resulting in the range [0,1], where 0 means the two corresponding samples are never clustered together while 1 means they are always in the same cluster. Finally, a spectral clustering is performed on the consensus matrix A to derive a stable clustering of the patients.

2.6 Evaluation metrics

We apply each of the aforementioned models to the artifically confounded multi-omics dataset described in section 2.1 and 2.2. Every model is evaluated in terms of their reconstruction accuracy, measured as the relative reconstruction error of inputs, their clustering stability, evaluated by the dispersion score of consensus clustering (CC), and deconfounding capabilities for clustering, estimated by calculating the adjusted Rand index (ARI) for true (cancer types) and confounder labels.

2.6.1 XVAE reconstruction monitoring

Model training is monitored through inspection of the validation loss. To evaluate reconstruction quality of the trained XVAE model, we compute the L2 relative error (RE) between the original input (x) and reconstructed data ($x^{'}$) for (i) each data type individually:

$$\text{RE} = \frac{\sqrt{\sum_{i=1}^{n} \|x_i - x_i^{'}\|^2}}{\sqrt{\sum_{i=1}^{n} \|x_i\|^2}} \tag{11}$$

,as well as (ii) for the combined data types:

$$\text{RE} = \frac{\sum_{m=1}^{2} \sqrt{\sum_{i=1}^{n} \|x_{mi} - x_{mi}^{'}\|^2}}{\sum_{m=1}^{2} \sqrt{\sum_{i=1}^{n} \|x_{mi}\|^2}} \tag{12}$$

, where $m = 1, 2$ indicates the two data types.

2.6.2 Clustering stability

Before assessing how well each model can derive a meaningful clustering, we want to first check if a model can stably cluster the samples. To achieve this goal, the dispersion score measures the internal stability of consensus clustering based on its consensus matrix A:

$$\text{Dispersion} = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} (A_{ij} - 0.5)^2 * 4}{n^2} \tag{13}$$

It ranges from 0 to 1, where 1 shows a perfect stability that every value in A is either 0 or 1, i.e. no confusion among clusterings, and the lower the less consensus or more confusion among clusterings.

2.6.3 Deconfounding performance

Since we are focusing on the deconfounding ability of models for clustering, we compute ARI as a classic external metric that assesses the similarity between two clusterings, the clustering derived from a model and a known label. Its value ranges from 0 to 1, and the higher the more similar. Here we compare our clustering with two different labels, the true one, namely cancer types, and the confounder. An ideal model should deconfound the features sufficiently while keeping the meaningful information for obtaining the true clustering. In other words, we expect a model with high ARI with the true label and low ARI with the confounder label. The association between true patient label and clusters obtained when modelling the original (unconfounded) data represents the best achievable clustering, with ARI value converging towards 1.

Similar to ARI, we compute another external clustering metric, the normalized mutual information (NMI), which measures the dependence between two clusterings. As it only shows complementary information to ARI, we record the NMI of every clustering model in Supplementary Table 1.

2.7 Implementation

For better stability and generalization, we train each model 50 times using i) randomly sampled training and validation sets with a ratio of 80:20 and ii) different seed of randomization. All of the models described in this study are built in Pytorch Lightning [22] and trained using the GPU units RTX 2080 Ti 11GB.

3 Results

cXVAE outperforms other deconfounding strategies in the presence of a single confounder

We simulated different types of confounding effects - linear, non-linear (squared), and categorical - on the multi-omics TCGA pan-cancer dataset to benchmark a total of four deconfounding frameworks, namely XVAE with Pearson correlation feature selection (XVAE+FS), conditional XVAE (cXVAE), adversarial training with XVAE (adv-XVAE), and confounder-regularised XVAE (cr-XAVE) (see Methods for more details).

We additionally included two baseline models to compare with: 1) confounder correction with linear regression (LR+PCA) and 2) vanilla XVAE without any deconfounding (XVAE). To estimate the robustness of each method, each model was trained on 50 iterations of randomly sampled training and validation data (80:20 split) and random seed initialization.

All proposed deconfounding approaches were able to correct for a linear confounder, as denoted by the high ARI for true clustering and low ARI for confounder clustering (Table 1). Performances started to decline for non-linear confounding problems, with cXVAE clearly outperforming other strategies. For non-linear confounders we noted large ARI for confounder clustering across all strategies and simulation setups. This illustrates, that while good clustering performance for true labels were achieved, the full removal of unwanted signal was not easily achievable for all the models. Categorical confounding was perceived to be the most difficult, with all models except cXVAE exhibiting a high decrease in true clustering performance. Notably, cr-XVAE and XVAE+FS were able to remove artificial confounders completely, however at the cost of simultaneously removing true clustering signal. adv-XVAE, which in theory should be a strategy well suited to deal with categorical problems, fails to consistently remove the categorical confounding effect. In general we noted a decline of reconstruction accuracy of models with increasing complexity of the confounder simulations.

Figure 4 visualizes the deconfounding behaviour of cXVAE for categorical confounding. With an exception to THCA (thyroid carcinoma), all classes were strongly confounded prior to cXVAE application (Figure 4, left). After model training, confounding classes are homogeneously mixed and clustering occurs with respect to true cancer types (Figure 4, right). In summary, across all confounder simulations, cXVAE clearly outperformed other deconfounding strategies in terms of clustering accuracy, deconfounding power, and model robustness. The ARI on true clustering obtained by cXVAE in all three scenarios reached around 0.7, which is very close to the performance of the vanilla XVAE on unconfounded data (0.731, see details in Supplementary Table 2).

A more detailed summary of the performances of each model can be found in Supplementary Table 1.

While Table 1 depicts the best performing implementation of each deconfounding model, we tested a number of possible implementations (see Methods), which we observed to

have a notable impact on model performance (Supplementary Table 2). Therefore, we provide design recommendations for each deconfounding strategy in the Supplementary Results 6.

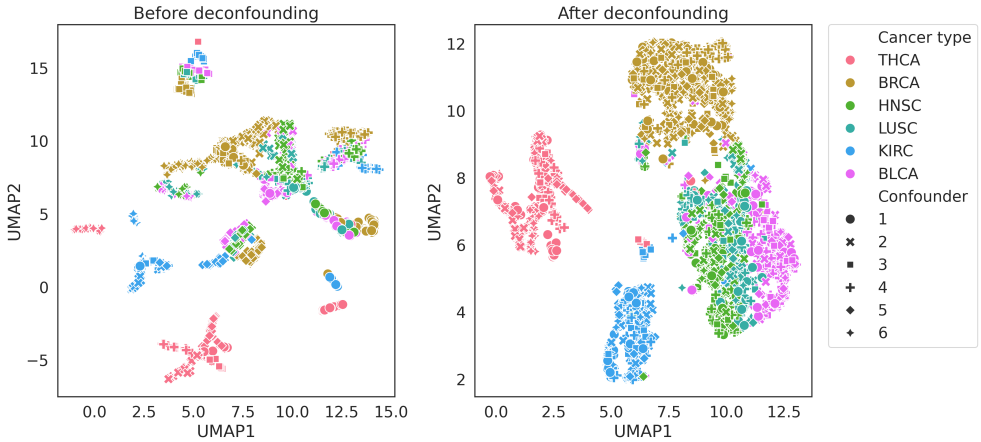

Figure 4: **Deconfounding behaviour of cXVAE**. Dimensionality reduction (UMAP) plot of categorically confounded data before and after application of cXVAE. Marker colors indicate the true label labels (i.e. TCGA cancer types), while marker shapes indicate the six different simulated confounders.

cXVAE is easily extendable to multiple confounder scenarios

In a realistic setting datasets can be confounded by multiple confounders with different biasing effects. In an attempt to investigate how well deconfounding strategies can handle more than one confounder, we simulated the parallel presence of three confounders of different effect, namely linear, non-linear, and categorical (Table 2).

In line with our observations with the single confounder simulations, cXVAE outperformed other models in terms of true clustering accuracy and de-confounding efficiency. While also other strategies like XVAE+FS, cr-XVAE, or LR+PCA were able to successfully remove all three simulated effects, they achieved this at the cost of true signal. adv-XVAE failed to fully remove confounders, while also showing very low true clustering accuracy and can therefore be considered unsuitable for the task. We also noted that the decline in reconstruction accuracy with increasingly complex confounding situations is even more pronounced in multiple confounder settings.

4 Discussion

In this study we explored which autoencoder-based deconfounding strategies i) can be extended to accommodate multi-omics data while still effectively adjusting for confounders, ii) can handle the presence of multiple confounders, and iii) derive an confounder-free patient sub-grouping. We proposed four conceptually different deconfounding approaches based on autoencoders and achieved a fair comparison between them by validating them using a dataset augemented with artificailly simulated confounding effects. This way, we were able to estimate which approaches are able to remove unwanted signal while preserving true patient clustering. Our study revealed significant differences between the performance of deconfounding methods in terms of true clustering accuracy and efficiency in confounder removal.

cXVAE is able to retrieve biology-driven clustering from confounded data

To illustrate the deconfounding capabilities of cXVAE, the model that outperformed others across all four evaluation metrics in various confounding scenarios, we examined the clustering results obtained from the TCGA dataset involving categorical confounders (Figure 4, right). The UMAP plot of latent features clearly showed that BRCA (breast invasive carcinoma), THCA (thyroid carcinoma), and KIRC (kidney renal clear cell carcinoma) were well clustered by cXVAE, while BLCA (bladder urothelial carcinoma), LUSC (lung squamous cell carcinoma), and HNSC (head and neck squamous cell carcinoma) were still entangled. In summary, we found this behaviour to be in line with the pathological and physiological differences between these cancer types. BLCA arises from urothelial cells in the transitional epithelium, which can change from cuboidal to squamous form when stretched. Furthermore, squamous differentiation is by far the most common histological variant of urothelial carcinoma [23], indicating a close relationship between urothelial carcinoma and squamous cell carcinoma. Apart from BLCA, the overlap in clustering of LUSC and HNSC can be directly explained by their common origin of squamous cells, while BRCA, THCA, and KIRC are all carcinoma related to glandular cells [24]. Supporting the validity of our obtained cXVAE clustering, other multi-omics pan-cancer studies utilising stacked variational autoencoders [25], penal-

ized matrix factorization [26], or supervised VAE [27] have retrieved similar cancer type clustering.

cXVAE stands out as versatile and accurate deconfounding approach

We found cXVAE to outperform other methods regardless the type of confounder simulated (linear, non-linear, categorical) and number of confounders present (single, multiple). Besides one of the earliest implemention of conditional autoencoders for biological data by Lawry et al. [6], also other studies have used conditional autoencoders to disentangle confounders in fMRI [28] or microRNA [29] data. The use of cXVAE for deconfounding, however, also comes with some limitations. For instance, we observed that cXVAE models showed the lowest reconstruction accuracy across all approaches tested, highlighting that the addition of more input information creates a challenge for the correct modelling of the data.

Adversarial training is challenging to optimise

While adversarial training represents a flexible deconfounding approach with its separate adversary network, we found it difficult to hyperparameter optimise the model to generate consistent results. We tested a number of implementations including varying the number of training batches during adversarial training or updating only part of the autoencoder as proposed in [8], however failed to improve the stability of the model. During training, we have found it challenging to balance deconfounding and reconstruction terms. This effect was even more pronounced in the case of multiple confounders as individual adversarial networks were trained for each confounder, each adding its own term to the objective function (see Supplementary Table 2).

We also found adversarial networks lacking of robustness due to a flaw in the proposed objective function. The flaw shows itself when adversarial networks are able to easily predict the confounders, as pre-training losses quickly converge to zero. The adversarial training stage that follows pre-training subsequently backpropagated negligibly small losses, effectively rendering the deconfounding ineffective. Novel studies address this weaknesses in the loss function by replacing the adversarial prediction loss with a sta-

tistical correlation loss [30]. While deconfounding through adversarial training remains an attractive approach due to its ability to handle any type of confounder, also others have reported difficulties in tuning the model [31].

cr-XVAE can adapt different measurements for confounding effect

Despite the wide usage of regularization terms in the loss function to, e.g. sparsify weights [32] or incorporate graphic information [33], our study focused on the deconfounding utility of regularizing the autoencoder. We implemented Pearson correlation and mutual information as the regularization term but other association measures could also be adopted, e.g. Spearman correlation and cosine similarity. Even though Pearson correlation only detects the linear association between latent features and confounders, it works moderately on non-linear and categorical confounding effects, suggesting that the dimensionality reduction of the encoder part helps reform the confounding effect to become linear. On the other hand, mutual information measures higher-level dependences compared to correlation. Its suboptimal performance could be partially explained by the approximation of the feature distribution, for which we resorted to both differentiable histogram and kernel density estimations. In the case of multiple confounders, it is also possible to weigh their associations differently to balance deconfounding strength.

Deconfounding approaches should be motivated by the nature of confounders

Our results revealed that methods differentiate themselves regarding the trade-off they achieve between removing confounding signal and maintaining true patient clustering. According to this, we categorised all surveyed deconfounding approaches in either *aggressive* or *preserving* methods.

Aggressive methods, e.g. removing latent features correlated with confounders, or enforcing deconfounding in the objective function, tend to efficiently remove confounders, however at the cost of biological signal. While this stringent approach resulted in intermediate true clustering accuracies in our study, the remaining true signal will likely contain strong true positives findings.

Preserving methods, including cXVAE and adversarial training, tend to not remove confounders at all costs, which generally yielded higher true clustering accuracies, but also a higher fraction of deconfounding signal remained in the data.

We argue that aggressive methods are best applied in the case of technical confounders, like batch effects, in which not removing all of the confounding signal may easily lead to spurious biological conclusions [2, 3]. Preserving methods, on the other hand, are interesting for removing biological confounders, e.g. sex or age effects, as applying a too stringent method may lead to removing too much (weak) biological signal of interest. However, this comes at the risk of findings containing more false positives, so thorough validation of results is advised.

Notably, we observed that simply training an autoencoder on confounded data may already lead to deconfounding, which is grounded in the inability of the autoencoder to fully model the confounder. Nevertheless, the obscuring effect of confounders on the true clustering can not be removed by a vanilla autoencoder, underlying the need for extraneous deconfounding approaches.

Alternative clustering strategies

In our study, clustering is done as an additional step on the latent features to guarantee the optimal performance of the deconfounding autoencoder. However, other researchers have pointed out the inconsistency between validation loss and downstream clustering performance [34] and proposed to iteratively train the clustering in a joint loss function with the autoencoder [35, 36]. We chose the standalone clustering strategy over the iterative one because 1) we want to avoid having too many terms in the loss function to confuse the training process and 2) the iterative clustering relies on a good initialization and is computationally expensive. Out of the various clustering methods, consensus clustering was selected for its high compatibility with VAE. This compatibility originates in the fact that consensus clustering can take the ensemble of clusterings computed on each sampling from the latent distribution. In theory, consensus clustering can facilitate more stable clusters than traditional methods and it has the potential to automatically determine the optimal number of clusters based on the consensus score [37].

Study limitations

The foundation of our study is the simulation of artificial confounders representing biological or technical out-of-interest variations. While the biggest advantage of simulations lies in the fact that they enable the comparison to ground-truth labels (i.e. cancer types), it is close to impossible to simulate the complexity of realistic confounding effects, so one needs to be aware that simulations always represent simplifications of real observable effects.

Despite treating all pan-cancer classes equally during confounder simulations, we noticed difficulties in confounding THCA and KIRC. Already in the original data set we observed strong and distinct clustering for these classes. This distinct pattern may ground in the large biological differences between these and other included cancer types. To preserve biological meaning, we therefore restrained from artificially over-tuning confounders for these classes.

5 Conclusion

In this study, we presented four VAE-based multi-omics clustering models and their variations, following different deconfounding strategies. Their clustering and deconfounding performance was evaluated and compared with baseline models on the multi-omics pan-cancer dataset from TCGA with artificailly generated confoundingn effects. The results showed that our novel models, cXVAE in particular, can effectively deal with the confounding effects simulated on the TCGA dataset and obtain the biologically meaningful clustering. We demonstrate that the multi-omics deconfounding VAE models explored in this study have big potential in delivering accurate patient subgrouping or disesase subtyping, ultimately enabling more personalised healthcare.

Data and code availability

Our code is available at https://github.com/ZuqiLi/Multi-view-Deconfounding-VAE. The simulated data generated in the course of this study is available at Zenodo (https://doi.org/10.5281/zenodo.10458941).

Acknowledgements

The authors thank the supporters of this study, namely the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No. 860895 TranSYS. Furthermore we thank the members of the Computational Population Biology group at Erasmus Medical Center for their critical and creative input to this work.

Author contributions statement

Z.L. and S.K. designed, planned, carried out the practical aspects of this study, and wrote the manuscript. G.V.R. offered mentoring and scientific input throughout the work process. G.V.R., E.S., K.v.S., and V.MdS. provided critical revision of the manuscript. All authors approve of the final manuscript.

Competing interests

The authors declare no competing interests.

Funding

Z.L. and S.K. received funding under the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No. 860895 TranSYS.

References

[1] Nikola Simidjievski, Cristian Bodnar, Ifrah Tariq, Paul Scherer, Helena Andres-Terre, Zohreh Shams, Mateja Jamnik, and Pietro Liò. Variational autoencoders for cancer data integration: Design principles and computational practice. *bioRxiv*, page 719542, 2019.

[2] Wilson Wen Bin Goh, Wei Wang, and Limsoon Wong. Why Batch Effects Matter in Omics Data, and How to Avoid Them. *Trends in Biotechnology*, 35(6):498–507, June 2017.

[3] Jelena Čuklina, Patrick G. A. Pedrioli, and Ruedi Aebersold. Review of Batch Effects Prevention, Diagnostics, and Correction Approaches. In Rune Matthiesen, editor, *Mass Spectrometry Data Analysis in Proteomics*, Methods in Molecular Biology, pages 373–387. Springer, New York, NY, 2020.

[4] M.A. Pourhoseingholi, A.R. Baghestani, and M. Vahedi. How to control confounding effects by statistical analysis. *Gastroenterol Hepatol Bed Bench*, 5:79–83.

[5] Adityanarayanan Radhakrishnan, Sam F Friedman, Shaan Khurshid, Kenney Ng, Puneet Batra, Steven A Lubitz, Anthony A Philippakis, and Caroline Uhler. Cross-modal autoencoder framework learns holistic representations of cardiovascular state. *Nature Communications*, 14(1):2436, 2023.

[6] Ana Lawry Aguila, James Chapman, Mohammed Janahi, and Andre Altmann. Conditional VAEs for Confound Removal and Normative Modelling of Neurodegenerative Diseases. In Linwei Wang, Qi Dou, P. Thomas Fletcher, Stefanie Speidel, and Shuo Li, editors, *Medical Image Computing and Computer Assisted Intervention – MICCAI 2022*, Lecture Notes in Computer Science, pages 430–440, Cham, 2022. Springer Nature Switzerland.

[7] Ayse B Dincer, Joseph D Janizek, and Su-In Lee. Adversarial deconfounding autoencoder for learning robust gene expression embeddings. *Bioinformatics*, 36(Suppl 2):i573–i582, December 2020.

[8] Mojtaba Bahrami, Malosree Maitra, Corina Nagy, Gustavo Turecki, Hamid R Rabiee, and Yue Li. Deep feature extraction of single-cell transcriptomes by generative adversarial network. *Bioinformatics*, 37(10):1345–1351, June 2021.

[9] X. Liu, B. Li, E.E. Bron, W.J. Niessen, E.B. Wolvius, and G.V. Roshchupkin. Projection-wise disentangling for fair and interpretable representation learning: Application to 3d facial shape analysis. In M. Bruijne, editor, *Medical Image Computing and Computer Assisted Intervention – MICCAI 2021. MICCAI 2021. Lecture Notes in Computer Science(*, volume 12905. Springer.

[10] Lucas Paulo de Lima Camillo, Louis R. Lapierre, and Ritambhara Singh. A pan-tissue DNA-methylation epigenetic clock based on deep learning. *npj Aging*, 8(1):1–15, April 2022.

[11] Huakang Tu, Chi Pang Wen, Shan Pou Tsai, Wong-Ho Chow, Christopher Wen, Yuanqing Ye, Hua Zhao, Min Kuang Tsai, Maosheng Huang, Colin P. Dinney, Chwen Keng Tsao, and Xifeng Wu. Cancer risk associated with chronic diseases and disease markers: prospective cohort study. *BMJ (Clinical research ed.)*, 360:k134, January 2018.

[12] Christiana Kartsonaki, Yuanjie Pang, Iona Millwood, Ling Yang, Yu Guo, Robin Walters, Jun Lv, Michael Hill, Canqing Yu, Yiping Chen, Xiaofang Chen, Eric O'Neill, Junshi Chen, Ruth C. Travis, Robert Clarke, Liming Li, Zhengming Chen, and Michael V. Holmes. Circulating proteins and risk of pancreatic cancer: a case-subcohort study among Chinese adults. *International Journal of Epidemiology*, 51(3):817–829, June 2022.

[13] Andrew O. Odegaard, Woon Puay Koh, Mimi C. Yu, and Jian Min Yuan. Body mass index and risk of colorectal cancer in Chinese Singaporeans: The Singapore Chinese Health Study. *Cancer*, 117(16):3841–3849, August 2011.

[14] John N. Weinstein, Eric A. Collisson, Gordon B. Mills, Kenna M. Shaw, Brad A. Ozenberger, Kyle Ellrott, Ilya Shmulevich, Chris Sander, and Joshua M. Stuart. The Cancer Genome Atlas Pan-Cancer Analysis Project. *Nature genetics*, 45(10):1113–1120, October 2013.

[15] Antonio Colaprico, Tiago C. Silva, Catharina Olsen, Luciano Garofano, Claudia Cava, Davide Garolini, Thais S. Sabedot, Tathiane M. Malta, Stefano M. Pagnotta, Isabella Castiglioni, Michele Ceccarelli, Gianluca Bontempi, and Houtan Noushmehr. TCGAbiolinks: An R/Bioconductor package for integrative analysis of TCGA data. *Nucleic Acids Research*, 44(8):e71, May 2016.

[16] Fengju Chen, Yiqun Zhang, Dominick Bossé, Aly-Khan A. Lalani, A. Ari Hakimi, James J. Hsieh, Toni K. Choueiri, Don L. Gibbons, Michael Ittmann, and Chad J. Creighton. Pan-urologic cancer genomic subtypes that transcend tissue of origin. *Nature Communications*, 8(1):199, August 2017.

[17] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015.

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[19] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[20] Ya Ju Fan. Autoencoder node saliency: Selecting relevant latent representations. *Pattern Recognition*, 88:643–653, April 2019.

[21] V.Y. Kiselev, K. Kirschner, M.T. Schaub, T. Andrews, A. Yiu, T. Chandra, K.N. Natarajan, W. Reik, M. Barahona, A.R. Green, and M. Hemberg. Sc3: consensus clustering of single-cell rna-seq data. *Nat Methods*. Epub 2017 Mar 27. PMID: 28346451; PMCID: PMC5410170.

[22] William Falcon and The PyTorch Lightning team. PyTorch Lightning, March 2019.

[23] Ashish M Kamat, Noah M Hahn, Jason A Efstathiou, Seth P Lerner, Per-Uno Malmström, Woonyoung Choi, Charles C Guo, Yair Lotan, and Wassim Kassouf. Bladder cancer. *The Lancet*, 388(10061):2796–2810, 2016.

[24] Tamarisk K. Horne and Marianne J. Cronje. Cancer Tissue Classification, Associated Therapeutic Implications and PDT as an Alternative. *Anticancer Research*, 37(6):2785–2807, June 2017.

[25] Bora Uyar, Jonathan Ronen, Vedran Franke, Gaetano Gargiulo, and Altuna Akalin. Multi-omics and deep learning provide a multifaceted view of cancer. *bioRxiv*, pages 2021–09, 2021.

[26] Agustín González-Reymúndez and Ana I. Vázquez. Multi-omic signatures identify pan-cancer classes of tumors beyond tissue of origin. *Scientific Reports*, 10(1):8341, May 2020.

[27] Xiaoyu Zhang, Jingqing Zhang, Kai Sun, Xian Yang, Chengliang Dai, and Yike Guo. Integrated multi-omics analysis using variational autoencoders: Application to pan-cancer classification. In *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 765–769, 2019.

[28] Xuetong Wang, Rong Zhou, Kanhao Zhao, Alex Leow, Yu Zhang, and Lifang He. Normative Modeling Via Conditional Variational Autoencoder and Adversarial Learning to Identify Brain Dysfunction in Alzheimer's Disease. In *2023 IEEE 20th International Symposium on Biomedical Imaging (ISBI)*, pages 1–4, April 2023.

[29] Tianwei Yu. AIME: Autoencoder-based integrative multi-omics data embedding that allows for confounder adjustments. *PLOS Computational Biology*, 18(1):e1009826, January 2022.

[30] Ehsan Adeli, Qingyu Zhao, Adolf Pfefferbaum, Edith V Sullivan, Li Fei-Fei, Juan Carlos Niebles, and Kilian M Pohl. Representation learning with statistical independence to mitigate bias. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2513–2523, 2021.

[31] Piotr Bojanowski, Armand Joulin, David Lopez-Paz, and Arthur Szlam. Optimizing the latent space of generative networks. *arXiv preprint arXiv:1707.05776*, 2017.

[32] Christos Louizos, Max Welling, and Diederik P. Kingma. Learning sparse neural networks through l_0 regularization. In *International Conference on Learning Representations*, 2018.

[33] Hongwei Wang, Fuzheng Zhang, Mengdi Zhang, Jure Leskovec, Miao Zhao, Wenjie Li, and Zhongyuan Wang. Knowledge-aware graph neural networks with label smoothness regularization for recommender systems. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, page 968–977, New York, NY, USA, 2019. Association for Computing Machinery.

[34] Mostafa Eltager, Tamim Abdelaal, Mohammed Charrout, Ahmed Mahfouz, Marcel J T Reinders, and Stavros Makrodimitris. Benchmarking variational AutoEncoders on cancer transcriptomics data. *PLOS ONE*.

[35] A.R. Owens, C.E. McInerney, and K.M. Prise. Novel deep learning-based solution for identification of prognostic subgroups in liver cancer (hepatocellular carcinoma. *BMC Bioinformatics*, 22:563.

[36] H. Muhammad. Unsupervised subtyping of cholangiocarcinoma using a deep clustering convolutional autoencoder. In D. Shen, editor, *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019. MICCAI 2019. Lecture Notes in Computer Science(*, volume 11764. Springer.

[37] S. Monti, P. Tamayo, and J. Mesirov. Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data. *Machine Learning*, 52:91–118.

[38] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

linear

	Reconstruction error	CC dispersion	Clustering performance (ARI)	
			True label (cancer type)	Confounder
LR+PCA	-	-	0.692	0.001
XVAE	0.246 (± 0.004)	0.844 (± 0.045)	0.506 (± 0.116)	0.151 (± 0.057)
XVAE+FS	0.245 (± 0.004)	0.860 (± 0.028)	0.571 (± 0.092)	0.008 (± 0.007)
cXVAE	**0.234 (± 0.003)**	**0.935 (± 0.023)**	**0.712 (±0.055)**	**0.001 (± 0.001)**
adv-XVAE	0.245 (± 0.004)	0.901 (± 0.032)	0.568 (± 0.070)	0.093 (± 0.051)
cr-XVAE	0.244 (± 0.003)	0.873 (± 0.028)	0.598 (± 0.074)	0.004 (± 0.004)

non-linear

	Reconstruction error	CC dispersion	Clustering performance (ARI)	
			True label (cancer type)	Confounder
LR+PCA	-	-	0.391	0.215
XVAE	0.236 (± 0.003)	0.826 (± 0.042)	0.307 (± 0.141)	0.297 (± 0.071)
XVAE+FS	0.236 (± 0.003)	0.805 (± 0.040)	0.411 (± 0.142)	0.138 (±0.078)
cXVAE	**0.227 (± 0.002)**	**0.908 (± 0.031)**	**0.646 (± 0.079)**	**0.076 (± 0.074)**
adv-XVAE	0.238 (± 0.004)	0.942 (± 0.025)	0.568 (± 0.049)	0.194 (± 0.006)
cr-XVAE	0.235 (± 0.002)	0.852 (± 0.043)	0.478 (± 0.129)	0.154 (± 0.042)

categorical

	Reconstruction error	CC dispersion	Clustering performance (ARI)	
			True label (cancer type)	Confounder
LR+PCA	-	-	0.150	0.071
XVAE	0.216 (± 0.003)	0.762 (± 0.055)	0.330 (± 0.125)	0.048 (± 0.088)
XVAE+FS	0.216 (± 0.003)	0.787 (± 0.040)	0.361 (± 0.100)	0.010 (± 0.023)
cXVAE	**0.210 (± 0.002)**	**0.911 (± 0.033)**	**0.664 (± 0.070)**	**0.001 (± 0.001)**
adv-XVAE	0.217 (± 0.002)	0.764 (± 0.058)	0.240 (± 0.188)	0.156 (± 0.084)
cr-XVAE	0.216 (± 0.003)	0.813 (± 0.034)	0.368 (± 0.101)	0.001 (± 0.001)

Table 1: **Overview performances of deconfounding strategy for single confounder simulations**. Values are displayed as mean ± standard deviation of 50 runs with different parameter initialisation and randomly sampled training and validation data. Models on the first column indicate the following deconfounding strategies and implementations thereof: linear regression followed by principal component analysis and KMeans clustering (LR+PCA), vanilla XVAE without any deconfounding (XVAE), XVAE with feature selection in the form of removing correlated latent features (XVAE+FS, correlation cutoff = 0.5), conditional XVAE (cXVAE, input + embedding), adversarial training with XVAE (adv-XVAE, multiclass MLP), confounder-regularised XVAE (cr-XAVE, squared correlation regularisation). Reconstruction error: relative error in the reconstruction of X1 and X2 weighted eqally; CC dispersion: consensus clustering agreement over 50 iterations; True clustering: adjusted rand index (ARI) of consensus clustering derived clusters with True label labels; Confounder clustering: ARI of consensus clustering derived clusters with simulated confounder labels.

			multiple confounders			
			Clustering performance (ARI)			
	Reconstruction error	CC dispersion	True label (cancer type)	Linear confounder	Squared confounder	Categorical confounder
LR+PCA	-	-	0.215	0.001	0.014	0.001
XVAE	0.161 (± 0.003)	0.725 (± 0.043)	0.216 (± 0.089)	0.015 (± 0.019)	0.140 (± 0.043)	0.067 (± 0.048)
XVAE+FS	0.161 (± 0.003)	0.731 (± 0.037)	0.265 (± 0.085)	0.007 (± 0.009)	0.019 (± 0.030)	0.109 (± 0.057)
cXVAE	**0.146 (± 0.002)**	**0.905 (± 0.022)**	**0.634 (± 0.042)**	**0.001 (± 0.001)**	**0.001 (± 0.001)**	**0.001 (± 0.001)**
adv-XVAE	0.158 (± 0.004)	0.753 (± 0.066)	0.225 (± 0.120)	0.016 (± 0.023)	0.107 (± 0.052)	0.106 (± 0.051)
cr-XVAE	0.161 (± 0.003)	0.764 (± 0.031)	0.369 (± 0.064)	0.003 (± 0.002)	0.007 (± 0.010)	0.001 (± 0.001)

Table 2: **Overview performances of deconfounding strategy in the presence of multiple confounders**. Values are displayed as mean ± standard deviation of 50 runs with different parameter initialisation and randomly sampled training and validation data. For a detailed description of columns and models, please refer to Table 1.

6 Supplementary Results

6.1 Differences in model implementations have significant impact on deconfounding performance

Each of the deconfounding strategies compared in this study have a number of possible implementations for both single (Supplementary Table 2) and multiple (Supplementary Table 3) confounder simulations, which we observed to have a major impact on model performances. Hence, in this section, we sequentially discuss each strategy, highlighting the most notable differences between implementations and providing readers with recommendations on designing similar frameworks.

XVAE with feature selection (XVAE+FS) - the central question when conducting feature selection by removing latent features correlated with unwanted confounders is the correlation cutoff employed. We compared the use of two different correlation cutoffs (Pearson correlation > 0.5, correlation > 0.3) and using a significant p-value as cutoff (p-value <0.05). While using absolute correlations as threshold removed up to 30% of latent features, using the p-value commonly removed >90% of features, rendering the latent space too sparse to use.

Conditional XVAE - cXVAE may vary regarding which layer of the autoencoder confounders are added, and whether they are added in both the encoder and decoder of the autoencoder. We found that in general it does not affect the deconfounding performance of the network in which layer of the encoder confounders are added (Supplementary Table 2, *input + embedding*, *fused + embedding*). We hypothesised that it may suffice to only add confounders to the encoder, however we found that a decoder lacking confounders (*input*) performs significantly worse. Conversely, adding the conditional variable to only the decoder results in good deconfounding performances (*embed*).

Adversarial training (adv-XVAE) - while the possibilities to design adversarial training strategies are countless we found it challenging to hyperparameter tune and balance the regularisation terms to simulatenously achieve good reconstruction and satisfactory deconfounding. Bahrami et al. [8] proposed a variation in which a simplified loss is backpropagated solely through the encoder of the network, to reinforce train-

ing (*scGAN*). However, also this modified version could not deliver consistent results. We attempted to extend the original model to accommodate multiple confounders by incorporating one adversarial network for each confounder present (*multinet MLP*). Contrary to expectations however, this implementations failed to improve adversarial training performance for the multiple confounder simulations (Supplementary Table 3).

Confounder regularised XVAE (cr-XVAE) - the regularisation term added for our deconfounding regularisation strategy largely determines the deconfounding properties. As a general guideline, Pearson correlation accounts for linear associations whereas mutual information can capture higher-level dependence. Because we are only interested in the strength of correlation, not the sign, the correlation coefficient was converted into absolute value or squared, leading to slightly different performances (Supplementary Table 2). Meanwhile, we conducted two different implementations for mutual information as loss function to approximate latent feature distribution, one based on differentiable histogram and the other on kernel density estimate. For both methods, the regularisation quality relies on how good the approximation is.

Supplementary Methods

6.2 Visualization of confounder simulations

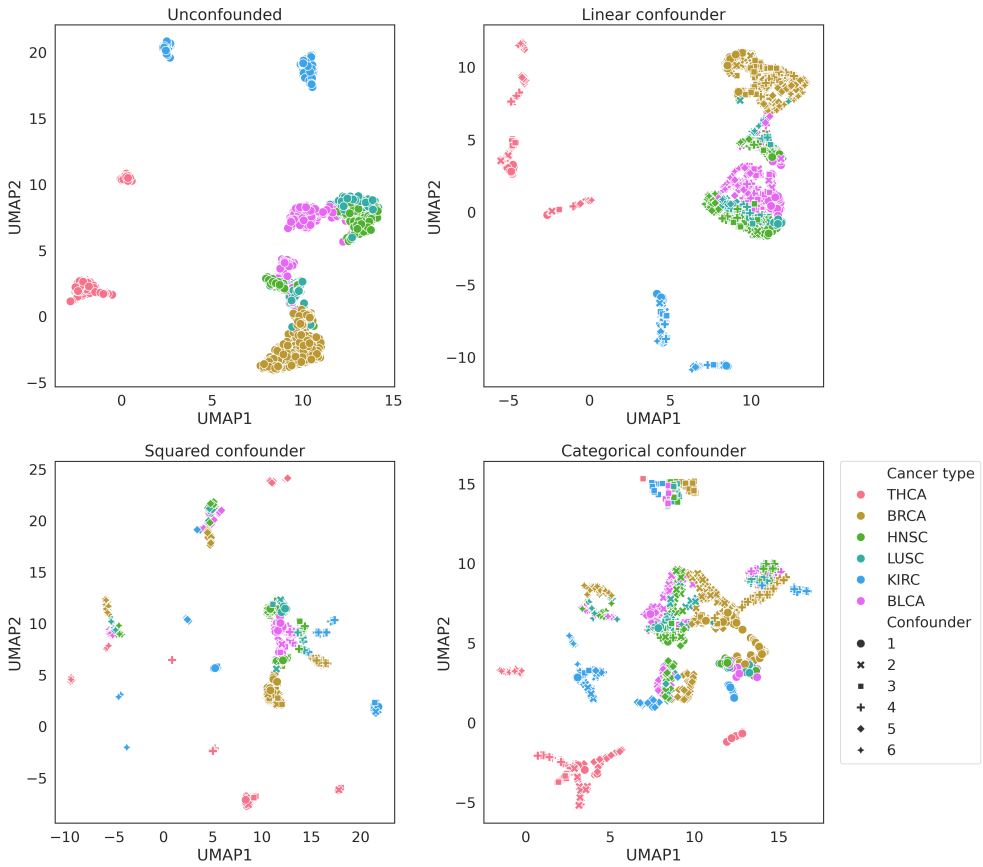

Figure 5: Dimensionality reduction (UMAP) plots of unconfounded (upper left), linear confounded (upper right), non-linear confounded (lower left), and categorical confounded (lower right) data. Colors indicate the true labels (i.e. TCGA cancer types), while shapes indicate the six different simulated confounders

6.3 Variational autoencoder for data integration (XVAE)

Autoencoders (AE) are unsupervised neural networks consisting of an encoder and a decoder part. While the encoder (E_ϕ, parameterised by ϕ) maps the high dimensional input datapoint (x) into a lower dimensional latent embedding (z), the decoder (D_θ, parameterised by θ) attempts to reconstruct the original input from said embedding ($x' = D_\theta(E_\phi(x))$).

The network is then trained by trying to minimize the error between original input and

reconstruction, quanitfied by the mean squared error:

$$L_{\mathrm{AE}}(\phi, \theta) = \|x - x'\|^2 = \|x - D_\theta(z)\|^2 = \|x - D_\theta(E_\phi(x))\|^2 \tag{14}$$

Standard autoencoders have been demonstrated shortcomings in terms of generative abilities and the tendency to overfit input data due to their lack of regularization. These limitations are addressed by a popular variant of autoencoders, termed variational autoencoders (VAE), which encodes input variables as a probability distribution ($q_\phi(z|x)$) over the latent space rather than as a fixed value [38]. For the sake of easy computation and interpretation, the typical choice is to assume the latent distribution $q_\phi(z|x)$ as a standard Gaussian distribution The unsupervised encoder-decoder structure allows (variational) autoencoders to act as dimensionality reduction and clustering tools, capable of accommodating input data from different sources in a joint low-dimensional embedding, ultimately making them a popular tool for data integration. Following the originally proposed implementation, we used a combined loss function of the Mean Squared Error (MSE) for scoring the reconstruction loss and Maximum Mean Discrepancy (MMD) as regularization term, balanced by the constant beta (β), which was set to 1 for all experiments:

$$L_{\mathrm{VAE}}(\phi, \theta) = -E_{z \sim q_\phi(z|x)}[\log p_\theta(x|z)] + \beta * \mathrm{MMD}(q_\phi(z|x)||p(z)) \tag{15}$$

6.4 cXVAE: Conditional XVAE

While traditionally auxilary variables are added in the encoder and decoder, in theory this can be varied. To investigate the impact on deconfounding attributable to when confounders are added to the model, we created several variations of the cXVAE architecture (Supplementary Figure 6):

1. *input* - in the encoder confounders are concatenated with the input vector; no addition of confounders in the decoder

2. *embed* - in the decoder confounders are concatenated with the latent embedding; no addition of confounders in the encoder

3. *input + embedding* - in the encoder confounders are concatenated with the input

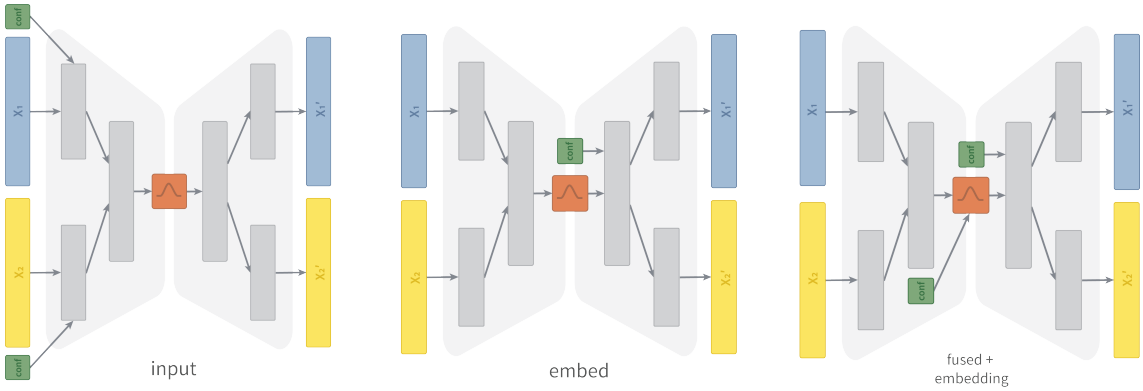

Figure 6: Overview on the different cXVAE assessed in this study.

vector; in the decoder confounders are concatenated with the latent embedding

4. *fused + embedding* - in the encoder confounders are concatenated in the second hidden layer, where the fusion of data types occurs; in the decoder confounders are concatenated with the latent embedding

6.4.1 adv-XVAE: XVAE with adversarial training

The training procedure can be broken down in three steps:

(i) Pre-train the XVAE: we pre-trained the XVAE model for 5 epochs optimising equation 9.

(ii) Pre-train the adversarial network: we used the latent embedding initialized in (i) to pre-trained the adversarial network for 5 epochs using a loss function fitting the numerical type of confounder; mean squared error for continuous, binary cross entropy for binary, and crossEntropy for categorical confounders, i.e. $L_{\text{adv}}(h_\nu(x), c)$ where h_ν denotes the adversarial network parameterized by ν.

(iii) Joint adversarial training: joint training is unique in its attempt to minimize the XVAE loss (equation 9) while simultaneously maximizing the adversarial network loss through a combined loss function:

$$L_{\text{adv}-\text{XVAE}}(\phi, \theta, \nu) = L_{\text{XVAE}}(\phi, \theta) - \lambda L_{\text{adv}}(h_v(x), c)] \tag{16}$$

Training is carried out in a ping-pong fashion, where firstly all weights of the

adversarial network are frozen and the autoencoder trained for one epoch optimizing equation 15, followed by the freezing of autoencoder weights and training of the adversary network for an entire epoch to minimize the adversarial network prediction loss L_{adv}.

Depending on how the adversarial architecture is designed in the case of multiple confounders present, we differentiated two major variations the above described base model: multiclass MLP and multinet MLP.

In *multiclass MLP* adversarial networks confounders are concatenated to a combined label, effectively creating a (mixed) multiclass classification problem. To successfully predict labels of different numeric nature, instead of a single final layer, multiclass MLP feature up to three final layers, each with different activation functions (ReLU for numerical confounders, Sigmoid for binary categorical confounders, softmax for categorical confounders with more than one class).

Multinet MLP, on the other hand, possesses an individual adversarial MLP for each confounder. The adversarial loss L_{adv} becomes the sum of losses of these individual networks.

Instead of augmenting the architecture of their *scGAN* model, Bahrami et al. [8] proposed changes to the training procedure. While following the same initial strategy of separately pre-training the autoencoder and adversarial MLP, they simplify the overall objective function (equation 16) by omitting to autoencoder loss and instead only train the encoder by maximizing the adversarial prediction loss L_{adv}.

7 Supplementary Tables

linear confounder

		reconstruction error			CC	Internal clustering		True clustering		Confounder clustering	
		X1	X2	X1,X2	dispersion	Silhouette	DB index	ARI	NMI	ARI	NMI
XVAE	mean	0.303	0.222	0.246	0.844	0.120	2.114	0.506	0.594	1.51E-01	2.02E-01
	std	0.010	0.003	0.004	0.045	0.013	0.118	0.116	0.093	5.71E-02	7.13E-02
XVAE + FS	mean	0.302	0.222	0.245	0.860	0.131	2.076	0.571	0.686	7.70E-03	1.14E-02
	std	0.010	0.003	0.004	0.028	0.011	0.146	0.092	0.053	6.79E-03	7.44E-03
cXVAE	mean	0.286	0.212	0.234	0.935	0.133	2.243	0.712	0.786	4.82E-04	3.11E-03
	std	0.008	0.001	0.003	0.023	0.011	0.092	0.055	0.037	5.37E-04	7.02E-04
adv-XVAE	mean	0.302	0.221	0.245	0.901	0.143	2.013	0.568	0.666	9.34E-02	1.28E-01
	std	0.010	0.003	0.004	0.032	0.011	0.099	0.070	0.053	5.14E-02	6.60E-02
cr-XVAE	mean	0.300	0.221	0.244	0.873	0.129	2.133	0.598	0.715	4.18E-03	6.97E-03
	std	0.007	0.002	0.003	0.028	0.010	0.123	0.074	0.046	3.91E-03	3.97E-03

non-linear confounder

		reconstruction error			CC	Internal clustering		True clustering		Confounder clustering	
		X1	X2	X1,X2	dispersion	Silhouette	DB index	ARI	NMI	ARI	NMI
XVAE	mean	0.235	0.236	0.236	0.826	0.119	2.078	0.307	0.424	2.97E-01	3.71E-01
	std	0.005	0.002	0.003	0.042	0.011	0.099	0.141	0.120	7.07E-02	8.17E-02
XVAE + FS	mean	0.235	0.236	0.236	0.805	0.114	2.120	0.411	0.526	1.38E-01	1.84E-01
	std	0.005	0.002	0.003	0.040	0.011	0.111	0.142	0.123	7.80E-02	9.75E-02
cXVAE	mean	0.223	0.229	0.227	0.908	0.124	2.185	0.646	0.713	7.59E-02	1.06E-01
	std	0.005	0.001	0.002	0.031	0.012	0.070	0.079	0.076	7.36E-02	9.87E-02
adv-XVAE	mean	0.240	0.237	0.238	0.942	0.147	1.918	0.568	0.635	0.194	0.249
	std	0.010	0.002	0.004	0.025	0.012	0.090	0.049	0.032	0.006	0.007
cr-XVAE	mean	0.235	0.235	0.235	0.852	0.121	2.092	0.478	0.575	1.54E-01	2.08E-01
	std	0.005	0.001	0.002	0.043	0.011	0.094	0.129	0.092	4.15E-02	5.24E-02

Table 1: **Overview performances of deconfounding strategy for single confounder simulations**. Values are displayed as mean (mean) and standard deviation (std) of 50 runs with different initialisation. Models on the first column indicate the following deconfounding strategies and implementations thereof: vanilla XVAE without any deconfounding (XVAE), XVAE with feature selection in the form of removing correlated latent features (XVAE+FS, correlation cutoff = 0.5), conditional XVAE (cXVAE, input + embedding), adversarial training with XVAE (adv-XVAE, multiclass MLP), confounder-regularised XVAE (cr-XAVE, squared correlation regularisation). Reconstruction error: relative error in the reconstruction of X1 and/or X2; CC dispersion: consensus clustering agreement over 50 iterations; Internal clustering: ....; True clustering: adjusted rand index (ARI) and Normalized Mutual Information (NMI) of consensus clustering derived clusters with ground-truth labels; Confounder clustering: ARI and NMI of consensus clustering derived clusters with simulated confounder labels.

categorical confounder

		reconstruction error			CC	Internal clustering		True clustering		Confounder clustering	
		X1	X2	X1,X2	dispersion	Silhouette	DB index	ARI	NMI	ARI	NMI
XVAE	mean	0.205	0.224	0.216	0.762	0.085	2.379	0.330	0.431	4.85E-02	9.12E-02
	std	0.006	0.002	0.003	0.055	0.008	0.073	0.125	0.148	8.80E-02	1.44E-01
XVAE + FS	mean	0.205	0.224	0.216	0.787	0.084	2.327	0.361	0.473	9.53E-03	2.02E-02
	std	0.006	0.002	0.003	0.040	0.008	0.087	0.100	0.099	2.31E-02	4.50E-02
cXVAE	mean	0.197	0.218	0.210	0.911	0.117	2.311	0.664	0.738	4.67E-04	3.17E-03
	std	0.003	0.001	0.002	0.033	0.011	0.084	0.070	0.060	9.62E-04	1.34E-03
adv-XVAE	mean	0.205	0.224	0.217	0.764	0.121	2.168	0.240	0.273	1.56E-01	3.12E-01
	std	0.003	0.001	0.002	0.058	0.008	0.050	0.188	0.203	8.40E-02	1.60E-01
cr-XVAE	mean	0.204	0.224	0.216	0.813	0.109	2.077	0.368	0.490	-9.12E-05	2.55E-03
	std	0.005	0.002	0.003	0.034	0.011	0.091	0.101	0.098	6.09E-04	7.96E-04

Table 2: **Overview performances of deconfounding strategy for single confounder simulations** - continued

Table 2: **Performance overview of different implementations of each deconfounding strategy for single confounders**

linear confounder

	Implementation	Reconstruction error			CC	Clustering performance					
						Internal		True label		Confounder	
		X1	X2	X1,X2	dispersion	Silhouette	DB index	ARI	NMI	ARI	NMI
baseline	XVAE (unconfounded)	0.438	0.278	0.31	0.957	0.128	2.17	0.731	0.821	1.14E-04	2.60E-03
	KMeans	-	-	-	-	0.132	2.14	0.212	0.381	3.53E-01	5.15E-01
	PCA(50) + KMeans	-	-	-	-	0.189	1.66	0.211	0.381	3.54E-01	5.17E-01
	LR + PCA + KMeans	-	-	-	-	0.251	1.80	0.692	0.763	1.50E-04	2.66E-03
	XVAE	0.294	0.22	0.241	0.807	0.11	2.25	0.411	0.522	2.08E-01	2.93E-01
XVAE + FS	XVAE (corr 0.3)	0.294	0.220	0.241	0.843	0.133	2.12	0.561	0.634	6.54E-03	1.13E-02
	XVAE (corr 0.5)	0.294	0.220	0.241	0.868	0.124	2.22	0.617	0.710	4.73E-03	8.51E-03
	XVAE (pVal 0.05)	0.294	0.220	0.241	0.653	0.160	1.66	0.261	0.354	3.38E-02	4.96E-02
cXVAE	input	0.29	0.217	0.238	0.907	0.111	2.06	0.429	0.582	1.84E-01	2.23E-01
	inputEmbed	0.281	0.212	0.233	0.895	0.147	2.21	0.715	0.768	4.72E-04	3.13E-03
	embed	0.285	0.215	0.235	0.931	0.146	2.20	0.702	0.769	4.87E-04	3.06E-03
	fusedEmbed	0.282	0.212	0.233	0.933	0.127	2.26	0.718	0.764	5.41E-04	3.15E-03
adv-XVAE	multiclass	0.289	0.217	0.238	0.920	0.142	2.16	0.632	0.723	3.46E-03	6.98E-03
	multiclass (1batch)	0.358	0.272	0.297	0.867	0.150	1.79	0.368	0.509	1.62E-01	3.05E-01
	multiclass (scGAN)	0.287	0.214	0.235	0.940	0.125	2.26	0.739	0.821	2.23E-04	2.93E-03
	multinet	0.289	0.217	0.238	0.920	0.142	2.16	0.632	0.723	3.46E-03	6.98E-03
cr-XVAE	corrSq	0.302	0.223	0.246	0.871	0.133	2.02	0.549	0.673	7.17E-03	9.80E-03
	corrAbs	0.292	0.218	0.239	0.921	0.134	2.23	0.692	0.779	1.93E-04	2.95E-03
	MIhist	0.295	0.219	0.241	0.838	0.089	2.21	0.273	0.377	2.88E-01	3.59E-01
	MIKDE	0.294	0.22	0.241	0.814	0.115	2.24	0.513	0.671	3.66E-03	6.00E-03

non-linear confounder

	Implementation	Reconstruction error			CC	Clustering performance					
						Internal		True label		Confounder	
		X1	X2	X1,X2	dispersion	Silhouette	DB index	ARI	NMI	ARI	NMI
baseline	XVAE (unconfounded)	0.438	0.278	0.31	0.957	0.128	2.17	0.731	0.821	1.14E-04	2.60E-03
	KMeans	-	-	-	-	0.160	1.82	0.251	0.411	3.25E-01	4.90E-01
	PCA(50) + KMeans	-	-	-	-	0.222	1.44	0.252	0.412	3.24E-01	4.89E-01
	LR + PCA + KMeans	-	-	-	-	0.242	1.48	0.391	0.567	2.15E-01	3.65E-01
	XVAE	0.227	0.234	0.232	0.846	0.123	2	0.153	0.293	4.00E-01	4.88E-01
XVAE + FS	XVAE (corr 0.3)	0.227	0.234	0.232	0.725	0.113	2.02	0.322	0.485	9.10E-02	1.16E-01
	XVAE (corr 0.5)	0.227	0.234	0.232	0.784	0.113	2.09	0.464	0.592	4.40E-02	5.30E-02
	XVAE (pVal 0.05)	0.227	0.234	0.232	0.721	0.192	1.44	0.226	0.308	1.16E-01	1.90E-01
cXVAE	input	0.224	0.233	0.23	0.956	0.146	1.96	0.477	0.598	1.95E-01	2.56E-01
	inputEmbed	0.217	0.23	0.225	0.867	0.145	2.18	0.758	0.788	8.64E-05	2.64E-03
	embed	0.216	0.229	0.224	0.832	0.114	2.22	0.498	0.576	1.34E-01	1.82E-01
	fusedEmbed	0.219	0.229	0.224	0.909	0.128	2.08	0.612	0.671	1.69E-01	2.20E-01
adv-XVAE	multiclass	0.229	0.235	0.233	0.951	0.144	1.930	0.559	0.637	1.97E-01	2.52E-01
	multiclass (1batch)	0.282	0.293	0.289	0.877	0.164	1.700	0.401	0.532	1.61E-01	3.27E-01
	multiclass (scGAN)	0.223	0.230	0.228	0.900	0.107	2.290	0.543	0.638	1.81E-01	2.37E-01
	multinet	0.229	0.245	0.233	0.951	0.144	1.930	0.559	0.637	1.97E-01	2.52E-01
cr-XVAE	corrSq	0.227	0.234	0.232	0.857	0.126	2.08	0.361	0.51	1.34E-01	1.93E-01
	corrAbs	0.226	0.234	0.232	0.818	0.118	2.03	0.334	0.496	1.67E-01	2.24E-01
	MIhist	0.228	0.236	0.233	0.867	0.126	2.01	0.336	0.493	1.91E-01	2.59E-01
	MIKDE	0.227	0.234	0.231	0.852	0.117	2.10	0.337	0.487	1.92E-01	2.53E-01

Table 2: **Performance overview of different implementations of each deconfounding strategy for single confounders** - continued

categorical confounder

	Implementation	Reconstruction error			CC	Clustering performance					
						Internal		True label		Confounder	
		X1	X2	X1,X2	dispersion	Silhouette	DB index	ARI	NMI	ARI	NMI
baseline	XVAE (unconfounded)	0.438	0.278	0.31	0.957	0.128	2.17	0.731	0.821	1.14E-04	2.60E-03
	KMeans	-	-	-	-	0.150	1.91	0.0610	0.117	2.18E-01	3.71E-01
	PCA(50) + KMeans	-	-	-	-	0.197	1.62	0.0631	0.120	2.15E-01	3.68E-01
	LR + PCA + KMeans	-	-	-	-	0.186	1.71	0.150	0.269	7.09E-02	1.52E-01
	XVAE	0.203	0.225	0.217	0.755	0.0819	2.32	0.335	0.476	3.31E-03	6.50E-03
XVAE + FS	XVAE (corr 0.3)	0.203	0.225	0.217	0.727	0.117	1.98	0.212	0.307	7.05E-03	1.23E-02
	XVAE (corr 0.5)	0.203	0.225	0.217	0.761	0.0842	2.31	0.339	0.482	4.00E-03	7.38E-03
	XVAE (pVal 0.05)	-	-	-	-	-	-	-	-	-	-
cXVAE	input	0.197	0.217	0.209	0.66	0.106	2.33	0.0961	0.129	2.36E-01	4.14E-01
	inputEmbed	0.195	0.218	0.209	0.873	0.0877	2.43	0.443	0.53	1.47E-03	5.22E-03
	embed	0.195	0.216	0.208	0.9	0.109	2.43	0.657	0.742	8.35E-04	3.65E-03
	fusedEmbed	0.191	0.216	0.206	0.829	0.101	2.38	0.575	0.680	3.76E-04	2.96E-03
adv-XVAE	multiclass	0.204	0.225	0.217	0.707	0.123	2.15	0.254	0.298	1.29E-01	2.48E-01
	multiclass (1batch)	0.263	0.286	0.277	0.777	0.146	2.05	0.140	0.193	1.36E-01	3.37E-01
	multiclass (scGAN)	0.196	0.218	0.210	0.681	0.111	2.41	0.0115	0.0102	3.58E-01	5.92E-01
	multinet	-	-	-	-	-	-	-	-	-	-
cr-XVAE	corrSq	0.200	0.222	0.213	0.744	0.079	2.380	0.206	0.314	3.78E-02	9.69E-02
	corrAbs	0.201	0.223	0.214	0.704	0.093	2.410	0.205	0.279	1.31E-01	2.51E-01
	MIhist	0.204	0.223	0.215	0.755	0.090	2.310	0.393	0.532	2.09E-03	6.24E-03
	MIKDE	0.202	0.225	0.216	0.815	0.091	2.280	0.319	0.469	2.15E-03	5.81E-03

multiple confounders

		reconstruction error			CC	Clustering performance									
						Internal		True label		Linear conf.		Non-linear conf.		Categorical conf.	
		X1	X2	X1,X2	dispersion	Silhouette	DB index	ARI	NMI	ARI	NMI	ARI	NMI	ARI	NMI
XVAE	mean	0.145	0.174	0.161	0.725	0.077	2.441	0.216	0.269	1.50E-02	2.57E-02	1.40E-01	1.97E-01	6.72E-02	1.33E-01
	std	0.004	0.002	0.003	0.043	0.006	0.058	0.089	0.100	1.93E-02	3.03E-02	4.33E-02	4.71E-02	4.77E-02	8.20E-02
XVAE + FS	mean	0.145	0.174	0.161	0.731	0.089	2.341	0.265	0.327	7.17E-03	1.20E-02	1.91E-02	2.78E-02	1.09E-01	2.03E-01
	std	0.004	0.002	0.003	0.037	0.006	0.064	0.085	0.104	8.63E-03	1.06E-02	2.99E-02	3.97E-02	5.75E-02	9.60E-02
cXVAE	mean	0.130	0.159	0.146	0.905	0.126	2.243	0.634	0.713	1.25E-04	2.69E-03	-3.67E-04	2.18E-03	4.62E-04	3.35E-03
	std	0.003	0.002	0.002	0.022	0.009	0.065	0.042	0.031	3.83E-04	5.10E-04	4.68E-04	5.78E-04	5.01E-04	7.26E-04
adv-XVAE	mean	0.141	0.172	0.158	0.753	0.106	2.207	0.225	0.262	1.56E-02	2.58E-02	1.07E-01	1.51E-01	1.06E-01	2.22E-01
	std	0.003	0.005	0.004	0.066	0.008	0.054	0.120	0.130	2.28E-02	3.48E-02	5.23E-02	7.37E-02	5.14E-02	1.04E-01
cr-XVAE	mean	0.145	0.174	0.161	0.764	0.095	2.234	0.369	0.490	2.67E-03	6.91E-03	6.88E-03	1.71E-02	4.87E-04	3.47E-03
	std	0.005	0.002	0.003	0.031	0.007	0.074	0.064	0.064	1.74E-03	3.58E-03	1.05E-02	2.44E-02	5.02E-04	7.75E-04

Table 3: **Overview performances of deconfounding strategy for multiple confounder simulations**