
Moment Matching Deep Contrastive Latent Variable Models

Ethan Weinberger
University of Washington

Nicasia Beebe-Wang
University of Washington

Su-In Lee
University of Washington

Abstract

In the contrastive analysis (CA) setting, machine learning practitioners are specifically interested in discovering patterns that are enriched in a *target* dataset as compared to a *background* dataset generated from sources of variation irrelevant to the task at hand. For example, a biomedical data analyst may seek to understand variations in genomic data only present among patients with a given disease as opposed to those also present in healthy control subjects. Such scenarios have motivated the development of contrastive latent variable models to isolate variations unique to these target datasets from those shared across the target and background datasets, with current state of the art models based on the variational autoencoder (VAE) framework. However, previously proposed models do not explicitly enforce the constraints on latent variables underlying CA, potentially leading to the undesirable leakage of information between the two sets of latent variables. Here we propose the moment matching contrastive VAE (MM-cVAE), a reformulation of the VAE for CA that uses the maximum mean discrepancy to explicitly enforce two crucial latent variable constraints underlying CA. On three challenging CA tasks we find that our method outperforms the previous state-of-the-art both qualitatively and on a set of quantitative metrics.

1 INTRODUCTION

In unsupervised learning settings, the goal is to find interesting patterns in a dataset. Here the defini-

tion of “interesting” is subjective and dependent on the specifics of a given problem domain. In many domains, data analysts are specifically interested in variations that are enriched in one dataset, referred to as the *target*, as compared to a second related dataset, referred to as the *background*. Target and background dataset pairs arise naturally in many settings. For example, data from healthy controls versus a diseased population, from pre-intervention and post-intervention groups, or signal-free versus signal-containing recordings all form natural background and target pairs (Abid et al., 2018). In each of these scenarios, the target dataset likely shares some set of uninteresting nuisance variations with the background dataset, such as population level variations or sensor noise. Unfortunately, when these nuisance variations explain the majority of the overall variance in a target dataset, standard unsupervised learning algorithms struggle to recover the patterns of interest in the data.

Isolating these salient variations present only in a target dataset is the subject of *contrastive analysis* (CA; (Zou et al., 2013)), and a number of algorithms have been proposed that extend unsupervised learning methods to the CA setting. For example, Abid et al. (2018) developed contrastive principal components analysis (cPCA), which finds a set of contrastive principal components that maximize variance in the target dataset while minimizing variance in the background. Moreover, a number of recent works (Li et al., 2020; Severson et al., 2019; Abid and Zou, 2019; Ruiz et al., 2019; Jones et al., 2021) have focused on developing methods for learning contrastive latent variable models. In such models, data points are generated using two sets of latent variables: one set, called the *background variables*, is used to generate both target and background data points, while the other, called the *salient variables*, is used exclusively for generating target data points.

In particular, to recover nonlinear relationships between latent variables and observed samples, a number of algorithms have been proposed that adapt the variational autoencoder (VAE) for CA and isolate salient

Proceedings of the 25th International Conference on Artificial Intelligence and Statistics (AISTATS) 2022, Valencia, Spain. PMLR: Volume 151. Copyright 2022 by the author(s).

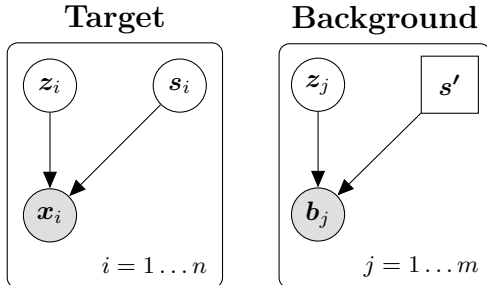


Figure 1: Generative processes for target and background samples. Background latent variables z are common to the two processes. A set of salient latent variables s are also used in the target generative process. For the background dataset these salient variables are fixed at s' . Observed values are shaded, and square nodes denote constant values.

factors of variation in target datasets (Abid and Zou, 2019; Severson et al., 2019; Ruiz et al., 2019). However, none of these previous methods explicitly enforce the constraints on latent variables assumed by CA during training, thereby possibly learning latent representations that are ineffective for CA tasks.

In this work we explore the impact of enforcing these constraints during model training. The specific contributions of this work are the following:

1. We identify two constraints underlying the CA framework that are ignored by previously proposed deep latent variable models for CA.
2. We propose the moment matching contrastive variational autoencoder (MM-cVAE), a new deep latent variable model for CA that is trained to learn representations that satisfy these assumptions.
3. We demonstrate on three challenging datasets that MM-cVAE’s representations more closely adhere to the latent variable constraints of CA; moreover, we find that adhering to these constraints results in representations that perform better on downstream CA tasks.

2 CONTRASTIVE LATENT VARIABLE MODELING

Recall that the goal of CA is to isolate variations only present in a target dataset from those present in both the target and background datasets. To capture this idea, we make use of a contrastive latent variable model defined as follows: let $X = \{\mathbf{x}_i\}_{i=1}^n$ denote our target dataset consisting of samples of interest. We

assume that our \mathbf{x}_i are drawn i.i.d. from a random process that depends on two sets of latent variables z_i and s_i distributed according to known prior distributions. Our observed \mathbf{x}_i are then drawn from a conditional distribution p_θ parameterized by unknown parameters θ :

$$\mathbf{x}_i \sim p_\theta(\mathbf{x}|z_i, s_i).$$

Here s refers to a set of *salient* latent variables of interest, while z refers to a set of *background* latent variables that are irrelevant to a given analysis. Our goal in the CA setting is to identify and isolate the salient latent variables for further analysis. Unfortunately, when trained on a target dataset alone, standard unsupervised learning algorithms will not necessarily isolate these salient latent variables. For example, standard VAEs are prone to learning latent representations in which salient latent factors are strongly entangled with background factors (Abid and Zou, 2019; Ruiz et al., 2019; Severson et al., 2019). Moreover, if the background latent factors explain most of the overall variance in the target dataset, a VAE may fail to capture the salient factors entirely.

Following previous work in the CA literature, we can attempt to solve our problem by introducing a background dataset $B = \{\mathbf{b}_j\}_{j=1}^m$ for which we assume that samples are generated only from the irrelevant latent factors with the salient variables fixed to some constant “reference vector” s' representing a lack of information (e.g., $\mathbf{0}$). We note that both our target and background samples are assumed to be drawn from the same generative process $p_\theta(\cdot)$. Moreover, we do not assume that we have the same number of samples in the target and background datasets (i.e., $n \neq m$), and we do not assume that there is any special relationship between samples i and j from the target and background datasets, respectively. We depict our data generating process in Figure 1. Target and background dataset pairs arise naturally in many experimental settings. For example, suppose we wanted to analyze data from patients receiving an experimental treatment in a clinical trial to better understand differences in how the treatment impacts the patients who receive it. In this case, a suitable background dataset would consist of data from control patients drawn from a similar population but administered a placebo instead of a real treatment, and our salient latent factors of interest would correspond to variations present *only* in patients that received the real treatment.

Our problem is thus to learn the parameters θ used to generate samples as well as another set of parameters ϕ for performing inference of the latent features. In

particular, we seek to learn two sets of parameters ϕ_s and ϕ_z to infer salient and background latent variable values, respectively.

3 RELATED WORK

Since the introduction of the VAE (Kingma and Welling, 2014), many works have focused on constraining the representations learned by VAEs to capture factors of variation that are more meaningful to the user. For example, learning disentangled representations (Bengio et al., 2013), in which each feature encodes a semantically meaningful attribute and is invariant to the other features, is the subject of much previous work (Higgins et al., 2017; Kim and Mnih, 2018; Chen et al., 2018; Mathieu et al., 2019; Lopez et al., 2018; Kumar et al., 2018; Burgess et al., 2018). Other works have focused on developing semi-supervised variants of the VAE, in which latent factors are encouraged to encode specific, pre-labeled factors of interest (Kingma et al., 2014; Paige et al., 2017). In contrast, in this work we specifically focus on learning representations that disentangle salient latent variables (which only underlie target data points) and background latent variables (which underlie both target and background data points).

Uncovering such salient variations is the focus of contrastive analysis, originally introduced by Zou et al. (2013). The CA framework has previously been used to extend a number of unsupervised learning methods. In the initial formulation, Zou et al. (2013) adapted mixture models for CA. Previous work (Abid et al., 2018) adapted principal component analysis to the contrastive setting so that a set of salient principal components are learned that summarize variations of interest in a target dataset while ignoring uninteresting variations. Other recent works (Li et al., 2020; Abid and Zou, 2019; Severson et al., 2019; Ruiz et al., 2019; Jones et al., 2021) have since developed probabilistic latent variable models for CA.

In particular, recent work has focused on adapting VAEs to the CA setting to handle situations in which the relationships between observed data points and latent variable values are highly nonlinear. Severson et al. (2019) were the first to do so; however, their formulation assumes that target data points can be decomposed as the sum of nonlinear transformations of target and salient latent variable values, i.e., $\mathbf{x}_i = f_\theta(\mathbf{s}_i) + g_\theta(\mathbf{z}_i)$, where \mathbf{s}_i and \mathbf{z}_i are salient and background latent variable values, respectively, for \mathbf{x}_i and f_θ and g_θ are nonlinear transformations. Such an assumption is highly restrictive, and likely to be violated by complex real-world datasets. Subsequently, two less restrictive models, the cVAE (Abid and Zou,

2019) and sRB-VAE (Ruiz et al., 2019), were proposed for CA tasks.

Shortcomings of previous methods: Despite their advantages over the original VAE for CA proposed by Severson et al. (2019), cVAE and sRB-VAE still have notable shortcomings; namely, their formulations do not fully reflect the constraints on latent variables underlying CA. Specifically, neither model explicitly **(1)** encourages the distribution of background latent factors to be similar across target and background samples or **(2)** enforces that inferred salient latent variable values for background samples are close to the informationless reference vector \mathbf{s}' . Such choices may enable information to leak between the salient and background latent spaces (i.e., target-dataset-specific variations may be present in the shared latent space and vice-versa). Such leakage violates the assumptions of the CA framework and may lead to lower quality representations for downstream CA tasks. Moreover, rather than explicitly enforcing these assumptions, cVAE and sRB-VAE add new KL divergence terms to the original VAE objective to encourage disentanglement of salient and background latent factors. Such divergence terms cannot be calculated directly, and they are instead approximated using the density-ratio trick (Kim and Mnih, 2018). However, doing so requires using adversary networks during model training, increasing training time and potentially leading to instabilities during training. In contrast, our MM-cVAE model explicitly enforces the assumptions of contrastive latent variable models during training by using a non-parametric kernel-based measure of independence. Moreover, we find that doing so results in high quality representations without needing the extra KL divergence loss terms used by previous models and the adversary networks needed to approximate them.

4 METHOD

Here we present the moment matching contrastive VAE (MM-cVAE, Figure 2), our extension of the VAE to the latent variable model introduced in Section 2. Similar to the cVAE (Abid and Zou, 2019), we use two probabilistic encoders $q_{\phi_z}(\mathbf{z}|\mathbf{x})$ and $q_{\phi_s}(\mathbf{s}|\mathbf{x})$ to approximate the posteriors over our two sets of latent variables \mathbf{z} and \mathbf{s} , respectively, and we use a single generator network $p_\theta(\cdot)$ that takes in a concatenation of \mathbf{z} and \mathbf{s} to reconstruct our data. For target samples, we concatenate the salient and background latent variables \mathbf{z}_x and \mathbf{s}_x as-is when reconstructing data points. However, for background samples, we concatenate \mathbf{z}_b with a constant \mathbf{s}' to reflect the assumptions of our latent variable model. In our experiments we chose $\mathbf{s}' = \mathbf{0}$.

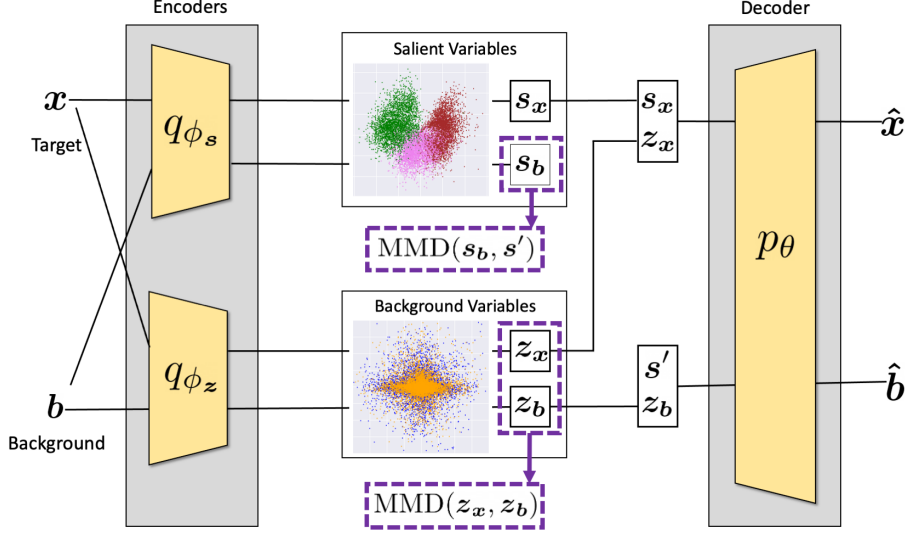


Figure 2: Network architecture for MM-cVAE. Target and background samples are fed through encoder networks q_{ϕ_s} and q_{ϕ_z} to obtain values for the salient and background latent variables, respectively. To explicitly enforce the assumptions of contrastive latent variable modeling, we penalize the Maximum Mean Discrepancy (MMD) between background latent features for background and target samples as well as the MMD between salient features for background samples and a delta distribution centered at an informationless reference vector s' . Blue and orange indicate whether background latent variable values are from a target or background sample, respectively. Green, pink, and brown indicate different classes of target samples.

We can then obtain variational lower bounds for the likelihoods of individual data points. For a given point \mathbf{x}_i in our target dataset we have the following bound:¹

$$\begin{aligned} \mathcal{L}_{\mathbf{x}}(\mathbf{x}_i) &= \mathbb{E}_{q_{\phi_z}(\mathbf{z})q_{\phi_s}(\mathbf{s})}[\log p_\theta(\mathbf{x}_i|\mathbf{z}, \mathbf{s})] \\ &\quad - D_{KL}(q_{\phi_z}(\mathbf{z}|\mathbf{x}_i) || p_{\mathbf{x}}(\mathbf{z})) \\ &\quad - D_{KL}(q_{\phi_s}(\mathbf{s}|\mathbf{x}_i) || p_{\mathbf{x}}(\mathbf{s})), \end{aligned}$$

where $p_{\mathbf{x}}(\mathbf{z})$ and $p_{\mathbf{x}}(\mathbf{s})$ are the prior distributions for our background and salient variables respectively for target samples. We let these two prior distributions be multivariate isotropic Gaussians, i.e., $p_{\mathbf{x}}(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, I)$ and $p_{\mathbf{x}}(\mathbf{s}) = \mathcal{N}(\mathbf{s}; \mathbf{0}, I)$. For points \mathbf{b}_j in our background dataset we similarly have

$$\begin{aligned} \mathcal{L}_{\mathbf{b}}(\mathbf{b}_j) &= \mathbb{E}_{q_{\phi_z}(\mathbf{z})}[\log p_\theta(\mathbf{b}_j|\mathbf{z}, \mathbf{s}')] \\ &\quad - D_{KL}(q_{\phi_z}(\mathbf{z}|\mathbf{b}_j) || p_{\mathbf{b}}(\mathbf{z})) \\ &\quad - D_{KL}(q_{\phi_s}(\mathbf{s}|\mathbf{b}_j) || p_{\mathbf{b}}(\mathbf{s})), \end{aligned}$$

where $p_{\mathbf{b}}(\mathbf{z})$ and $p_{\mathbf{b}}(\mathbf{s})$ are the prior distributions for our background and salient variables respectively for background samples. As with $p_{\mathbf{x}}(\mathbf{z})$, we let $p_{\mathbf{b}}(\mathbf{z})$

be an isotropic Gaussian. However, we set $p_{\mathbf{b}}(\mathbf{s}) = \delta\{\mathbf{s} = \mathbf{s}'\}$, where $\delta\{\mathbf{s} = \mathbf{s}'\}$ denotes a Dirac distribution centered at \mathbf{s}' , to reflect our assumption that salient latent variables are not used to generate background data points. Unfortunately, this prior distribution makes the second KL divergence term in our expression for $\mathcal{L}_{\mathbf{b}}$ non-differentiable. Prior work (Abid and Zou, 2019; Ruiz et al., 2019) has sidestepped this differentiability issue by ignoring this term altogether during optimization. However, doing so introduces a path for information to leak between the background and salient latent spaces. Rather than ignoring this term altogether, we can instead introduce a differentiable surrogate loss term that encourages the inferred salient latent variable values for background points to be close to \mathbf{s}' .

Here we choose to use the maximum mean discrepancy (MMD) (Gretton et al., 2006) as our surrogate. The MMD is a nonparametric kernel-based two sample test statistic originally proposed for testing whether two datasets $X \in \mathbb{R}^{n \times d}$ and $Y \in \mathbb{R}^{m \times d}$ were drawn from the same distribution. For our purposes it is sufficient to know that the MMD has a differentiable closed-form empirical estimator

¹Originally derived in Abid and Zou (2019). For completeness, we include a full derivation of this bound in the supplement.

$$\begin{aligned} \mathcal{L}_{\text{MMD}}(X, Y) &= \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n k(\mathbf{x}_i, \mathbf{x}_j) \\ &\quad - \frac{1}{nm} \sum_{i=1}^n \sum_{j=1}^m k(\mathbf{x}_i, \mathbf{y}_j) + \frac{1}{m^2} \sum_{i=1}^m \sum_{j=1}^m k(\mathbf{y}_i, \mathbf{y}_j). \end{aligned}$$

If k is a universal² kernel function, such as the Gaussian kernel $k(\mathbf{x}, \mathbf{y}) = \exp(-\gamma \|\mathbf{x} - \mathbf{y}\|^2)$, then minimizing the MMD is equivalent to minimizing a distance between all moments of X and Y (Li et al., 2015), and the MMD equals 0 asymptotically if and only if X and Y were drawn from the same distribution (Gretton et al., 2006, 2012). MMD-based penalties using the Gaussian kernel have been successfully used by a number of works to enforce invariance constraints when training deep generative models (Shaham et al., 2017; Louizos et al., 2016; Lopez et al., 2018; Li et al., 2015; Dziugaite et al., 2015; Zhao et al., 2019), and we refer the reader to the supplement for a more detailed treatment of the MMD. Substituting an empirical estimate of the MMD for the undifferentiable divergence term in our equation for $\mathcal{L}_{\mathbf{b}}$ and letting $\mathcal{L}'_{\mathbf{b}}(B)$ denote the new likelihood bound for our entire background dataset B we have,

$$\begin{aligned} \mathcal{L}'_{\mathbf{b}}(B) &= \sum_{j=1}^m \left(\mathbb{E}_{q_{\phi_{\mathbf{z}}}(\mathbf{z})} [\log p_{\theta}(\mathbf{b}_j | \mathbf{z}, \mathbf{s}')] \right. \\ &\quad \left. - D_{KL}(q_{\phi_{\mathbf{z}}}(\mathbf{z} | \mathbf{b}_j) \parallel p_{\mathbf{b}}(\mathbf{z})) \right) \\ &\quad - \lambda_1 \cdot \mathcal{L}_{\text{MMD}}(\hat{q}_{\phi_{\mathbf{s}, \mathbf{b}}}(\mathbf{s}), \delta\{\mathbf{s} = \mathbf{s}'\}), \end{aligned}$$

where $\hat{q}_{\phi_{\mathbf{s}, \mathbf{b}}}(\mathbf{s})$ denotes the empirical distribution of inferred salient variable values for background data points and λ_1 is a hyperparameter controlling regularization strength.

An additional assumption of our contrastive latent variable model is that the distribution of common latent variables \mathbf{z} is identical across our background and target datasets. Previous work (Abid and Zou, 2019; Ruiz et al., 2019) ignores this constraint, adding yet another potential avenue for information to leak between the salient and background latent spaces. Here we explicitly enforce this assumption by once again making use of the MMD. Let $\hat{q}_{\phi_{\mathbf{z}, \mathbf{a}}}(\mathbf{z})$ and $\hat{q}_{\phi_{\mathbf{z}, \mathbf{b}}}(\mathbf{z})$ denote the empirical distributions of shared latent factors for our target and background datasets,

²A kernel function $k(x, \cdot)$ is called universal if k is continuous for all x and the RKHS induced by k is dense in $\mathcal{C}(\mathcal{X})$, where $\mathcal{C}(\mathcal{X})$ denotes the space of continuous bounded functions on the compact domain \mathcal{X} .

respectively. We then add a penalty term $-\lambda_2 \cdot \mathcal{L}_{\text{MMD}}(\hat{q}_{\phi_{\mathbf{z}, \mathbf{a}}}(\mathbf{z}), \hat{q}_{\phi_{\mathbf{z}, \mathbf{b}}}(\mathbf{z}))$ to our objective function, where λ_2 is another hyperparameter controlling regularization strength. Letting $\mathcal{L}_{\mathbf{x}}(X) = \sum_{i=1}^n \mathcal{L}_{\mathbf{x}}(\mathbf{x}_i)$, we then have our final objective

$$\max_{\theta, \phi} \mathcal{L}_{\mathbf{x}}(X) + \mathcal{L}'_{\mathbf{b}}(B) - \lambda_2 \cdot \mathcal{L}_{\text{MMD}}(\hat{q}_{\phi_{\mathbf{z}, \mathbf{a}}}(\mathbf{z}), \hat{q}_{\phi_{\mathbf{z}, \mathbf{b}}}(\mathbf{z})),$$

which we can optimize using a variant of minibatch gradient descent. Due to the use of the MMD in our objective function, we call our approach the moment matching contrastive VAE (MM-cVAE).

5 EXPERIMENTS

To confirm that MM-cVAE’s loss function results in higher-quality representations for CA, we experimented with three collections of target and background datasets (Section 5.1). For each of these collections, points in the target dataset have ground truth class labels. However, distinguishing between these classes may be difficult from latent variable values inferred using a non-contrastive approach (e.g. a standard VAE). In our experiments, we compared our MM-cVAE with cVAE (Abid and Zou, 2019) and sRB-VAE (Ruiz et al., 2019), the previous state of the art deep latent variable models for CA. For each dataset, we evaluated our models’ performance quantitatively and qualitatively (Section 5.2). We refer the reader to the supplement for additional details on dataset preprocessing, model training, and hyperparameter tuning. The code used to preprocess our datasets as well as to train and evaluate our models is available at <https://github.com/suinleelab/MM-cVAE>.

5.1 Datasets

CelebA images with accessories. We created target and background datasets from subsets of the well-known CelebA (Liu et al., 2015) dataset. Our target dataset comprised images of celebrities wearing either glasses or caps, while our background dataset contained images of celebrities wearing neither accessory. Our goal here was to learn a salient representation of the target dataset that separated celebrities wearing glasses versus those with caps.

Leukemia treatment response. Here we combined datasets from Zheng et al. (2017) containing gene expression measurements from two patients with acute myeloid leukemia (AML) before and receiving a blood cell transplant. Our goal here is to learn a representation that separates pre- and post-transplant measurements. As a background dataset we used expression

Table 1: Quantitative evaluation comparing how well MM-cVAE and two baseline models cVAE (Abid and Zou, 2019) and sRB-VAE (Ruiz et al., 2019) satisfy the assumptions of contrastive latent variable models; i.e., that the distributions of background latent variables z_x and z_b should be identical and salient variable values for background samples s_b should be close to a model’s informationless reference vector. To measure distributional similarity, we train logistic regression models to classify points based on their origins and report accuracy on a held out test set. We also report silhouette scores to quantify how well clustered points are based on their origins. Lower values correspond to better satisfying the assumptions.

Dataset	Model	Difference in distribution for z_x vs. z_b (Lower is better)		Difference in distribution for s_b vs. s' (Lower is better)	
		Logistic (%)	Silhouette	Logistic (%)	Silhouette
CelebA	sRB-VAE	0.72 ± 0.01	0.02 ± 0.01	0.83 ± 0.02	0.35 ± 0.01
	cVAE	0.68 ± 0.01	0.02 ± 0.00	0.87 ± 0.06	0.36 ± 0.01
	MM-cVAE	0.51 ± 0.01	0.00 ± 0.01	0.73 ± 0.10	0.22 ± 0.05
AML	sRB-VAE	0.84 ± 0.01	0.05 ± 0.01	0.92 ± 0.05	0.44 ± 0.06
	cVAE	0.84 ± 0.03	0.05 ± 0.01	0.92 ± 0.05	0.40 ± 0.04
	MM-cVAE	0.74 ± 0.00	0.00 ± 0.01	0.53 ± 0.00	0.04 ± 0.01
Epithelial Cells	sRB-VAE	0.75 ± 0.08	0.01 ± 0.01	0.95 ± 0.07	0.47 ± 0.08
	cVAE	0.69 ± 0.04	0.01 ± 0.01	0.84 ± 0.04	0.35 ± 0.01
	MM-cVAE	0.56 ± 0.01	0.00 ± 0.01	0.51 ± 0.01	0.09 ± 0.02

measurements from two healthy control patients that were collected as part of the same study.

Epithelial cell infection response. We constructed our target dataset by combining two sets of gene expression measurements from Haber et al. (2017). These datasets consist of gene expression measurements of intestinal epithelial cells from mice infected with either *Salmonella* or *Heligmosomoides polygyrus* (*H. poly*). Here our goal is to separate cells by infection type. As a background dataset we used measurements collected from healthy cells released by the same authors.

5.2 Assessing Adherence to the Assumptions of CA

We began our evaluation by quantitatively assessing how well MM-cVAE and our baseline models’ learned representations satisfy the assumptions of contrastive latent variable modeling. First, we evaluated how well each model satisfies the assumption that the shared background latent variables for background and target samples have the same distribution (i.e., the assumption that z_b and z_x have the same distribution). To do so, we trained logistic regression classifiers on each models’ background latent variable values to classify whether a data point came from a background sample or target sample. We also used the Silhouette Score (SS) to quantify how well z_x and z_b separated into distinct clusters. In short: the SS varies between -1 and 1, with larger positive values corresponding to better

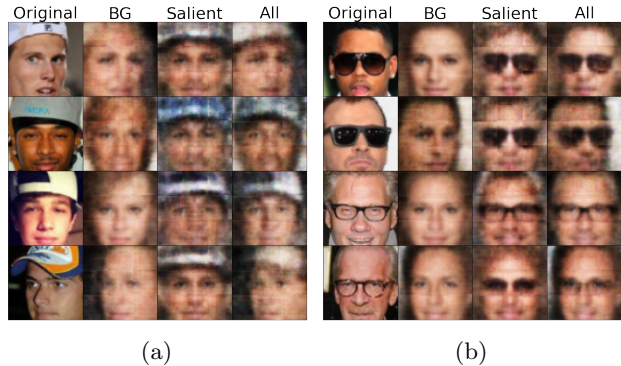


Figure 3: Sample of qualitative results on the CelebA with accessories task for images with hats (a) and glasses (b). For both sets of samples we present original images (left), reconstructions using only background (BG) variables (center left), reconstructions using only salient variables (center right), and reconstructions using all latent variables (right). For both collections of images, the salient latent variables encode the accessory of interest (i.e., hats or glasses) along with other seemingly random facial features. On the other hand, the background variables encode the remaining attributes (e.g., pose, lighting etc.)

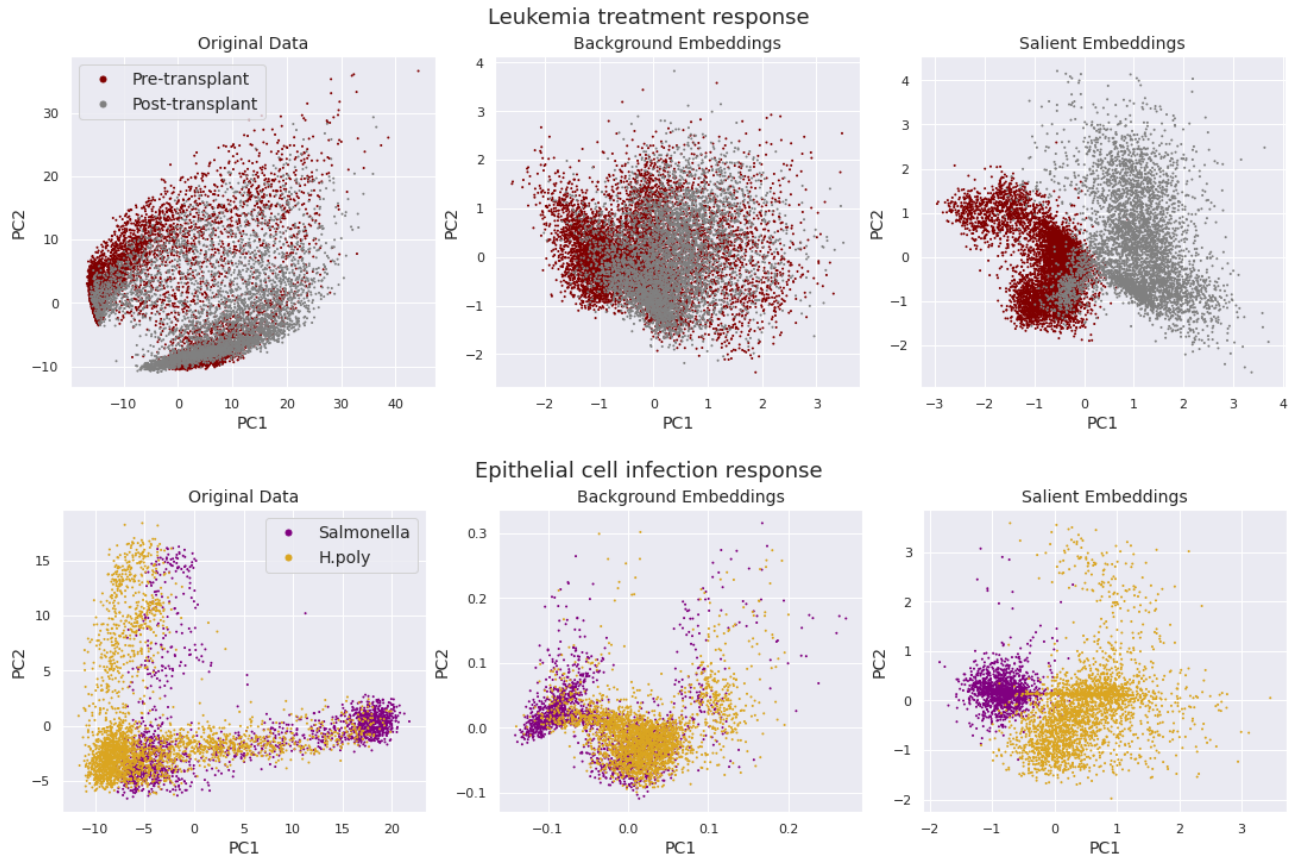


Figure 4: Plots of the first two principal components for our target RNA-seq datasets pre-embedding (left), embedded in the shared latent space (center) and in the salient embedding space (right). For both target datasets, the two subgroups are difficult to distinguish in the spaces spanned by the first two principal components for the original data and the background embeddings. However, the subgroups are more easily separable in the salient embedding space.

separated clusters and values near 0 indicating overlapping clusters; we refer the reader to the supplement for additional details on the SS. A model that satisfies this assumption, producing similar distributions of \mathbf{z}_x and \mathbf{z}_b , should result in low values for both of these metrics. Using a similar methodology we also quantified how well each model satisfied the assumption that inferred salient latent variable values for background samples \mathbf{s}_b should be similar to models’ reference vectors \mathbf{s}' . That is, we trained logistic regression classifiers to distinguish between models’ values of \mathbf{s}_b and copies of \mathbf{s}' , and we used the SS to quantify how well each model’s values of \mathbf{s}_b and copies of \mathbf{s}' formed distinct clusters. Once again, lower values correspond to better satisfying the assumption.

We report the values of these metrics for both assumptions in Table 1. We find that MM-cVAE consistently outperforms our baseline models on these metrics, often considerably. These results indicate that MM-cVAE’s loss function leads it to learn representa-

tions that align more closely with the assumptions of contrastive latent variable modeling than do those of previously proposed models. Given these results, we next confirmed that learning representations that better conform to the assumptions of contrastive latent variable modeling indeed leads to better performance on downstream CA tasks.

5.3 Assessing Latent Representation Quality

We next assessed the quality of each model’s representations for downstream CA tasks. We began our assessment by qualitatively evaluating these representations. For the CelebA dataset, we did so as follows. For a target image \mathbf{x} , we first inferred \mathbf{x} ’s background latent variable values by feeding it into q_{ϕ_b} . Next, we concatenated these inferred background features with a zero vector in place of the salient variable values, and we then passed these values through the decoder network to generate a new image. We also performed the same procedure using salient variable values in-

ferred by q_{ϕ_s} and a zero vector in place of the background variable values. If a model has learned representations that are well-suited for CA, we would expect it to generate faces with the accessory present in the original target image (i.e., glasses or a hat) and arbitrary settings for other attributes when decoding using only salient variable values; similarly, we would expect the model to generate faces with similar characteristics (e.g., pose, lighting) as the original target image but with no accessory present when decoding using only background variables. Since a neural network may transform an input of $\mathbf{0}$ to a non-zero output, in these experiments we constrained our decoder network’s bias terms to all be zero. We report results from this procedure in Figure 3.

From visually inspecting the images generated using only background variables, we find that our background latent variables appear to encode sensible values for non-accessory-related attributes while not encoding information on the accessories themselves. Moreover, by inspecting the images generated from salient variables, we find that our salient variables consistently encode the correct accessory while simultaneously encoding arbitrary values for other attributes. On the other hand, images produced with our baseline models using the same procedure (supplement) indicate that these models were less effective at separating salient and target factors of variation, with information appearing to leak between the two latent spaces (e.g., accessories appear in images generated solely from background variables). Taken together, these results indicate that our training procedure does indeed result in representations that are better suited for CA than those of previously proposed models.

We next turned our attention to the two gene expression datasets. For these datasets, we could not easily assess the quality of generated samples by visually inspecting them as we did with CelebA. Thus, we adopted a different procedure. For both datasets we used q_{ϕ_b} and q_{ϕ_s} to infer values for our background and salient variables for our target data points. We then performed principal component analysis separately on the inferred background and salient latent variable values, and we show plots of the first two principal components for both in Figure 4. From our plots, we can observe strong mixing between the two known subgroups for both datasets in the background latent variable space. This phenomenon indicates that our background latent variable spaces are encoding shared features across the groups, as desired. On the other hand, we find clear separation between subgroups in the plots of our salient latent variables. We performed the same procedure using our baseline models, and we provide the results in the supplement. Once again,

we find that our baselines are less effective at disentangling salient and background latent variables, with stronger (undesirable) separation between subgroups in the background latent variable space and mixing between subgroups in the salient space; these phenomena are especially pronounced for the Epithelial cell dataset. These results further indicate that MM-cVAE learns representations that are better suited for CA.

We next quantitatively evaluated how well each model’s salient representations for target points \mathbf{s}_x separated classes within each target dataset (e.g., glasses vs. caps in CelebA). To do so, we used the same metrics as in Table 1. That is, we trained logistic regression classifiers to classify points based on ground-truth class labels and used silhouette scores to measure how well clustered points from different classes were. As we want classes to be well separated in the salient embedding space, *higher* values here correspond to better performance. We also computed these metrics for the common latent factors \mathbf{z}_x . Because our background latent factors should have similar distributions across the classes within each target dataset, *lower* values for our metrics on \mathbf{z}_x correspond to better performance. We report these results in Table 2. We find that MM-cVAE once again consistently outperforms both baseline models across all our metrics. These results provide further evidence that MM-cVAE’s stronger adherence to the assumptions underlying CA indeed leads to improved performance on downstream CA tasks.

5.4 Assessing Sample Quality

In addition to latent representation quality, we also assessed the quality of samples generated by each model. In particular, we sought to understand whether MM-cVAE’s better disentanglement of salient and background variables came at the expense of sample quality. For the CelebA dataset we computed the Fréchet Inception distance (FID; (Heusel et al., 2017)) for each model, and we report our results in Supplementary Table 1. We computed the FID separately for generated background samples and for generated target samples. Since FID scores for all contrastive models were relatively high - indicating low sample quality across all contrastive methods tested - we also evaluated a non-contrastive VAE with otherwise identical architecture. Its scores were also high, indicating that the high scores were due to our relatively simple architecture (as opposed to e.g. VQ-VAE (van den Oord et al., 2017)), rather than the contrastive setup. We also evaluated sample quality for our gene expression datasets using MMD in place of FID (Supplementary Table 2). MM-cVAE performed the best for most met-

Table 2: Quantitative comparison of MM-cVAE and two baseline models cVAE (Abid and Zou, 2019) and sRB-VAE (Ruiz et al., 2019). To measure the quality of a model’s representations, we train logistic regression classifiers to predict target data points’ classes from their background and salient representations. We also use the silhouette score to measure how well separated points from different classes are. Higher (lower) values on salient (background) representations correspond to better performance.

Dataset (Task in target dataset)	Model	Target Dataset Class Separation			
		s_x (Higher is better)		z_x (Lower is better)	
		Logistic (%)	Silhouette	Logistic (%)	Silhouette
CelebA (Glasses vs. caps)	sRB-VAE	0.79 ± 0.01	0.11 ± 0.01	0.83 ± 0.00	0.05 ± 0.01
	cVAE	0.75 ± 0.05	0.06 ± 0.02	0.81 ± 0.01	0.04 ± 0.01
	MM-cVAE	0.87 ± 0.04	0.17 ± 0.03	0.58 ± 0.02	0.00 ± 0.01
AML (Pre- vs. post-transplant patients)	sRB-VAE	0.94 ± 0.02	0.23 ± 0.04	0.87 ± 0.03	0.11 ± 0.01
	cVAE	0.88 ± 0.05	0.18 ± 0.02	0.77 ± 0.04	0.06 ± 0.01
	MM-cVAE	0.95 ± 0.01	0.28 ± 0.03	0.76 ± 0.02	0.04 ± 0.01
Epithelial Cells (Mice infected with <i>Salmonella</i> vs. <i>H. poly</i>)	sRB-VAE	0.84 ± 0.03	0.21 ± 0.05	0.88 ± 0.03	0.11 ± 0.01
	cVAE	0.83 ± 0.10	0.10 ± 0.05	0.90 ± 0.04	0.08 ± 0.02
	MM-cVAE	0.95 ± 0.03	0.33 ± 0.08	0.78 ± 0.02	0.03 ± 0.01

rics, suggesting MM-cVAE achieves better representation quality while maintaining similar sample quality to previously proposed models.

6 DISCUSSION

In this work we demonstrated that previous VAE-based models designed for CA learn representations that often in fact strongly violate the assumptions of CA. In response to this issue, we proposed MM-cVAE. In contrast to previous work, MM-cVAE explicitly enforces the assumptions of CA by incorporating two penalties based on the maximum mean discrepancy into the training process. Moreover, by doing so, we avoid the need to use complex adversarial training procedures used by previously proposed models to disentangle salient and background latent variables.

Through both qualitative and quantitative evaluations on the CelebA dataset and two real-world biological datasets, we find that explicitly enforcing these penalties led to higher quality embeddings for CA tasks, effectively stratifying different classes of target samples and relegating irrelevant latent factors to a separate background embedding space.

Acknowledgements

We would like to thank Ian Covert, members of the AIMS Lab at the University of Washington, and our AISTATS reviewers for feedback that greatly improved this work. We would also like to thank the Theis Lab at Helmholtz Munich for their development of high-quality open-source software for analyzing single-cell

RNA-sequencing data (`scanpy`), which facilitated this work. This work was funded by the National Science Foundation [DGE-2140004, DBI-1552309, and DBI-1759487] and the National Institutes of Health [R35 GM 128638, and R01 NIA AG 061132].

References

- A. Abid and J. Zou. Contrastive variational autoencoder enhances salient features. *arXiv preprint arXiv:1902.04601*, 2019.
- A. Abid, M. J. Zhang, V. K. Bagaria, and J. Zou. Exploring patterns enriched in a dataset with contrastive principal component analysis. *Nature communications*, 9(1):1–7, 2018.
- Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.
- C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Waters, G. Desjardins, and A. Lerchner. Understanding disentangling in β -vae. *arXiv preprint arXiv:1804.03599*, 2018.
- T. Q. Chen, X. Li, R. B. Grosse, and D. K. Duvenaud. Isolating sources of disentanglement in variational autoencoders. In *NeurIPS*, 2018.
- G. K. Dziugaite, D. M. Roy, and Z. Ghahramani. Training generative neural networks via maximum mean discrepancy optimization. In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, pages 258–267, 2015.
- A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf,

- and A. Smola. A kernel method for the two-sample problem. *Advances in neural information processing systems*, 19:513–520, 2006.
- A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.
- A. L. Haber, M. Biton, N. Rogel, R. H. Herbst, K. Shekhar, C. Smillie, G. Burgin, T. M. Delorey, M. R. Howitt, Y. Katz, et al. A single-cell survey of the small intestinal epithelium. *Nature*, 551(7680):333–339, 2017.
- M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.
- I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24–26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL <https://openreview.net/forum?id=Sy2fzU9gl>.
- A. Jones, F. W. Townes, D. Li, and B. E. Engelhardt. Contrastive latent variable modeling with application to case-control sequencing experiments. *arXiv preprint arXiv:2102.06731*, 2021.
- H. Kim and A. Mnih. Disentangling by factorising. In *International Conference on Machine Learning*, pages 2649–2658. PMLR, 2018.
- D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- D. P. Kingma and M. Welling. Auto-encoding variational bayes. In Y. Bengio and Y. LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14–16, 2014, Conference Track Proceedings*, 2014. URL <http://arxiv.org/abs/1312.6114>.
- D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling. Semi-supervised learning with deep generative models. In *Advances in neural information processing systems*, pages 3581–3589, 2014.
- A. Kumar, P. Sattigeri, and A. Balakrishnan. Variational inference of disentangled latent concepts from unlabeled observations. In *International Conference on Learning Representations*, 2018.
- D. Li, A. Jones, and B. Engelhardt. Probabilistic contrastive principal component analysis. *arXiv preprint arXiv:2012.07977*, 2020.
- Y. Li, K. Swersky, and R. Zemel. Generative moment matching networks. In *International Conference on Machine Learning*, pages 1718–1727. PMLR, 2015.
- Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015.
- R. Lopez, J. Regier, M. I. Jordan, and N. Yosef. Information constraints on auto-encoding variational bayes. *Advances in Neural Information Processing Systems*, 31, 2018.
- C. Louizos, K. Swersky, Y. Li, M. Welling, and R. S. Zemel. The variational fair autoencoder. In Y. Bengio and Y. LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2–4, 2016, Conference Track Proceedings*, 2016. URL <http://arxiv.org/abs/1511.00830>.
- E. Mathieu, T. Rainforth, N. Siddharth, and Y. W. Teh. Disentangling disentanglement in variational autoencoders. In *International Conference on Machine Learning*, pages 4402–4412. PMLR, 2019.
- B. Paige, J.-W. van de Meent, A. Desmaison, N. Goodman, P. Kohli, F. Wood, P. Torr, et al. Learning disentangled representations with semi-supervised deep generative models. *Advances in Neural Information Processing Systems*, 30:5925–5935, 2017.
- A. Ruiz, O. Martinez, X. Binefa, and J. Verbeek. Learning disentangled representations with reference-based variational autoencoders. *arXiv preprint arXiv:1901.08534*, 2019.
- K. A. Severson, S. Ghosh, and K. Ng. Unsupervised learning with contrastive latent variable models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4862–4869, 2019.
- U. Shaham, K. P. Stanton, J. Zhao, H. Li, K. Raddassi, R. Montgomery, and Y. Kluger. Removal of batch effects using distribution-matching residual networks. *Bioinformatics*, 33(16):2539–2546, 2017.
- A. van den Oord, O. Vinyals, and K. Kavukcuoglu. Neural discrete representation learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6309–6318, 2017.
- S. Zhao, J. Song, and S. Ermon. Infovae: Balancing learning and inference in variational autoencoders. In *Proceedings of the aaai conference on artificial intelligence*, volume 33, pages 5885–5892, 2019.
- G. X. Zheng, J. M. Terry, P. Belgrader, P. Ryvkin, Z. W. Bent, R. Wilson, S. B. Ziraldo, T. D. Wheeler, G. P. McDermott, J. Zhu, et al. Massively parallel digital transcriptional profiling of single cells. *Nature communications*, 8(1):1–12, 2017.

J. Y. Zou, D. Hsu, D. C. Parkes, and R. P. Adams.
Contrastive learning using spectral methods. *Proceedings of Advances in Neural Information Processing Systems*, 2013.

Supplementary Material: Moment Matching Deep Contrastive Latent Variable Models

A Variational Lower Bound Derivations

Here we present derivations for the likelihood lower bounds presented in the main text. For our target data points, we have

$$\begin{aligned}\log p_\theta(\mathbf{x}_i) &= \log \int p_\theta(\mathbf{x}_i, \mathbf{s}, \mathbf{z}) d\mathbf{s} d\mathbf{z} \\ &= \log \int \frac{p_\theta(\mathbf{x}_i, \mathbf{s}, \mathbf{z}) \cdot q_\phi(\mathbf{x}_i | \mathbf{s}, \mathbf{z})}{q_\phi(\mathbf{x}_i | \mathbf{s}, \mathbf{z})} d\mathbf{s} d\mathbf{z} \\ &\geq \int q_\phi(\mathbf{x}_i | \mathbf{s}, \mathbf{z}) \log \frac{p_\theta(\mathbf{x}_i, \mathbf{s}, \mathbf{z})}{q_\phi(\mathbf{x}_i | \mathbf{s}, \mathbf{z})} d\mathbf{s} d\mathbf{z} \\ &= \int q_\phi(\mathbf{x}_i | \mathbf{s}, \mathbf{z}) \log \frac{p_\theta(\mathbf{x}_i | \mathbf{s}, \mathbf{z}) \cdot p(\mathbf{s}, \mathbf{z})}{q_\phi(\mathbf{x}_i | \mathbf{s}, \mathbf{z})} d\mathbf{s} d\mathbf{z} \\ &= \int \left(q_\phi(\mathbf{x}_i | \mathbf{s}, \mathbf{z}) \log p_\theta(\mathbf{x}_i | \mathbf{s}, \mathbf{z}) + \log \frac{p(\mathbf{s}, \mathbf{z})}{q_\phi(\mathbf{x}_i | \mathbf{s}, \mathbf{z})} \right) d\mathbf{s} d\mathbf{z} \\ &= \mathbb{E}_{q_\phi(\mathbf{s}, \mathbf{z})} [\log p_\theta(\mathbf{x}_i | \mathbf{s}, \mathbf{z})] - D_{KL}(q_\phi(\mathbf{s}, \mathbf{z} | \mathbf{x}_i) || p(\mathbf{s}, \mathbf{z})) \\ &= \mathbb{E}_{q_\phi(\mathbf{s}) q_\phi(\mathbf{z})} [\log p_\theta(\mathbf{x}_i | \mathbf{s}, \mathbf{z})] - D_{KL}(q_\phi(\mathbf{s} | \mathbf{x}_i) || p(\mathbf{s})) - D_{KL}(q_\phi(\mathbf{z} | \mathbf{x}_i) || p(\mathbf{z})) \\ &\triangleq \mathcal{L}_\mathbf{x}(\mathbf{x}_i)\end{aligned}$$

where we obtain the inequality in line three of our derivation using Jensen’s inequality, and our last line follows from the assumed independence of our salient and target latent values \mathbf{s} and \mathbf{z} . Here we take our prior distributions $p(\mathbf{s})$ and $p(\mathbf{z})$ to be isotropic Gaussians. $\mathcal{L}_\mathbf{b}$ follows essentially the same derivation, with $p(\mathbf{z})$ chosen to be an isotropic Gaussian and $p(\mathbf{s})$ a delta distribution centered at some reference vector \mathbf{s}' .

B Dataset Preprocessing

Here we provide preprocessing details for each of the datasets used in Section 5.1. Jupyter notebooks with implementations of our preprocessing workflows can be found in the supplementary materials.

Epithelial Cell. We began by downloading `GSE92332_SalmHelm.UMIcounts.txt` from <https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE92332>. Each column in this file corresponds to a single cell sample, while each row corresponds to the expression level of a given gene. For each cell, we extracted its condition (i.e., healthy control or disease status) from that cell’s label in the first row of the file. In particular, in our experiments we used cells labelled as `Control`, `Salmonella`, and `Hpoly.Day10`; cells with other labels were discarded.

Following standard practices in single cell RNA-seq analyses, we first normalized the counts such that each cell had a total count of 10000 after normalization. We then took the logarithm of these normalized counts + 1. For these two normalization steps we used the `normalize_total` and `log1p` functions implemented in the `scanpy` Python package. Finally, we filtered our data to only have the features selected by the `highly_variable_genes` function in the `scanpy` Python package.

Leukemia Treatment Response. We downloaded a set of scRNA-seq gene expression measurements from a leukemia patient pre and post-transplant from <https://cf.10xgenomics.com/samples/>

cell-exp/1.1.0/aml027_pre_transplant/aml027_pre_transplant_filtered_gene_bc_matrices.tar.gz and https://cf.10xgenomics.com/samples/cell-exp/1.1.0/aml027_post_transplant/aml027_post_transplant_filtered_gene_bc_matrices.tar.gz, respectively. Similar measurements were downloaded for a second patient at https://cf.10xgenomics.com/samples/cell-exp/1.1.0/aml035_pre_transplant/aml035_pre_transplant_filtered_gene_bc_matrices.tar.gz and https://cf.10xgenomics.com/samples/cell-exp/1.1.0/aml035_post_transplant/aml035_post_transplant_filtered_gene_bc_matrices.tar.gz. A set of measurements from a healthy control patient were also downloaded from https://cf.10xgenomics.com/samples/cell-exp/1.1.0/frozen_bmmc_healthy_donor1/frozen_bmmc_healthy_donor1_filtered_gene_bc_matrices.tar.gz.

We then preprocessed our data in a similar manner as done in previous work (Li et al., 2020). That is, we took the log of expression counts + 1, removed any gene features or cell samples consisting entirely of zeros, and filtered down the list of gene features for each file to those that were shared among all the files. Finally, we filtered our data down to the 1000 highest variance gene features.

CelebA. We cropped each image and scaled it to be 64x64 pixels. We then constructed our target dataset as follows. First, we filtered our images to those with a 1 value for the binary **Eyeglasses** attribute and a 0 value for the binary **Wearing_Hat** attribute (i.e., the people in these images are wearing glasses and not hats). We then randomly choose 5000 of these images. Next, we filtered our images to those with a 1 value for the binary **Wearing_Hat** attribute and a 0 value for the binary **Eyeglasses** attribute (i.e., the people in these images are wearing hats and not glasses). We then selected 5000 of these images, and combined them with the previously selected images to form our target dataset. For our background dataset we randomly selected 10000 images with both binary attributes having a 0 value.

C Implementation Details and Hyperparameter Tuning

All models were implemented using PyTorch with the PyTorch Lightning framework.³ All models were optimized using ADAM (Kingma and Ba, 2014) with a learning rate of 0.001, $\beta_1 = 0.9$ and $\beta_2 = 0.999$. Code implementing all models is available as part of the supplementary materials.

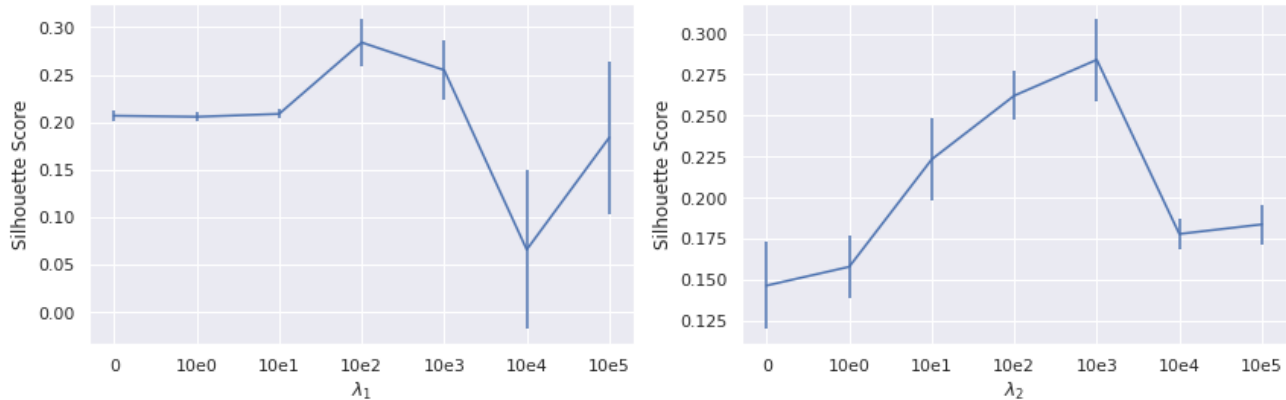
Network Architectures: For our experiments with the two genomic datasets, each model’s encoder networks had a single fully connected hidden layer with 400 units. A ReLU activation function was used at the hidden layer. These hidden layers then connected to salient/background latent spaces with 5 and 10 nodes, respectively. Our decoder networks consisted of the same architecture in reverse. For cVAE, we used separate encoder networks for background and salient latent variables while we used a single network for sRB-VAE as described in the models’ original papers. For MM-cVAE we took the same approach as cVAE (i.e., using separate encoder networks). For all models we used a single decoder network to map from the latent variables to the original input space.

For the CelebA experiments, our encoder networks instead consisted of four convolutional blocks. Each block contained a convolutional layer with a 4x4 kernel and 2x2 stride with the padding parameter set to maintain the original input size. This convolutional layer was then followed by a LeakyReLU activation function with a negative slope of 0.2 and a batch normalization layer. The number of output channels in each convolutional layer were 32, 64, 128 and 256, respectively. After the convolutional blocks, the outputs were connected via fully connected layer to the salient/background latent spaces. For each model we used a salient latent space size of 5 and background latent space of size 15. Our decoder networks consisted of the same convolutional blocks architecture in reverse.

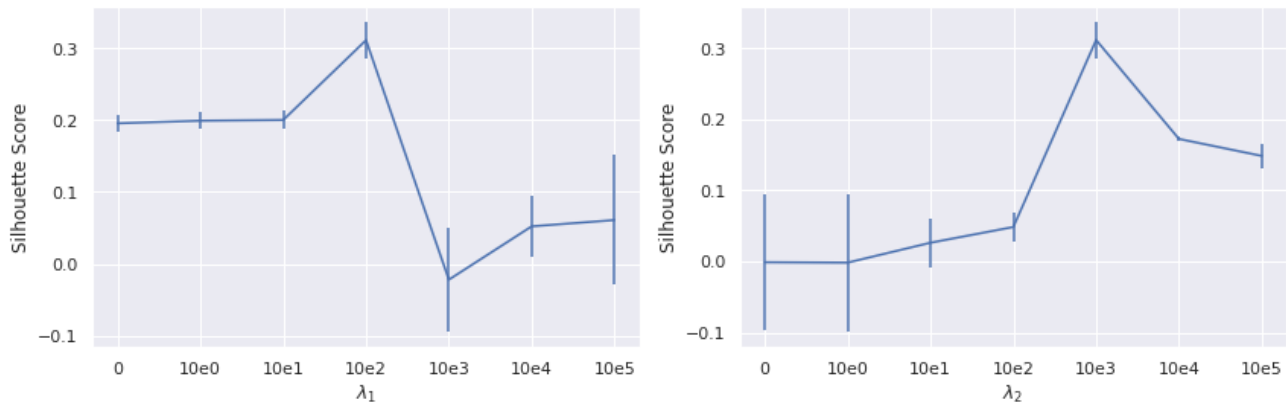
Hyperparameter choices for MM-cVAE: To tune MM-cVAE’s first regularization parameter λ_1 (i.e., the hyperparameter controlling the strength of the KL surrogate term), we set its value so that the resulting loss term would be similar in magnitude to that of the other KL term $\sum_j D_{KL}(q_{\phi_z}(\mathbf{z}|\mathbf{b}_j) || p_{\mathbf{b}}(\mathbf{z}))$ for background data points in a given minibatch. We found that setting $\lambda_1 = 10e2$ accomplished this goal in all our experiments.

We set λ_2 ’s value to keep the scale of our second penalty similar to that of the lower bound, as done with another MMD-based penalty proposed in previous work (Louizos et al., 2016). We found that setting $\lambda_2 = 10e3$

³<https://github.com/PyTorchLightning/pytorch-lightning>



Supplementary Figure 1: Sensitivity analysis on for MM-cVAE’s salient latent space for the AML gene expression dataset.



Supplementary Figure 2: Sensitivity analysis on for MM-cVAE’s salient latent space for the mice epithelial cell gene expression dataset.

accomplished this goal for all our experiments.

We further explored the impact of these hyperparameters by performing a sensitivity analysis of MM-cVAE’s performance with respect to λ_1 and λ_2 on salient representations for the two gene expression datasets (Supplementary Figure 1, Supplementary Figure 2). At low values of λ_1 and λ_2 , our MMD loss terms are dominated by other terms and do not have a strong effect on performance. We found that increases in both parameters up until the values described previously improved representation quality as the MMD loss terms were no longer ignored during training. Once these terms began dominating the loss, representation quality suffered.

D The Maximum Mean Discrepancy

Suppose we have two datasets $X = \{x_i\}_{i=1}^N \sim P_0$ and $Y = \{y_j\}_{j=1}^M \sim P_1$, and we wish to test whether the datasets were drawn from the same distribution; i.e., whether $P_0 = P_1$. One way to do so is via a test statistic known as the maximum mean discrepancy (MMD) (Gretton et al., 2012). The MMD is based on the following intuitive idea: if the empirical statistics $\psi(\cdot)$ of the two datasets are similar, then the datasets were likely to have been drawn from the same distribution. Formally, an empirical estimator for the MMD is defined as

$$\begin{aligned} \widehat{\text{MMD}}(X, Y) &= \left\| \frac{1}{N} \sum_{i=1}^N \psi(x_i) - \frac{1}{M} \sum_{j=1}^M \psi(y_j) \right\|^2 \\ &= \frac{1}{N^2} \sum_{i=1}^N \sum_{i'=1}^N \psi(x_i)^T \psi(x_{i'}) - \frac{2}{NM} \sum_{i=1}^N \sum_{j=1}^M \psi(x_i)^T \psi(y_j) + \frac{1}{N^2} \sum_{j=1}^M \sum_{j'=1}^M \psi(y_j)^T \psi(y_{j'}) \end{aligned}$$

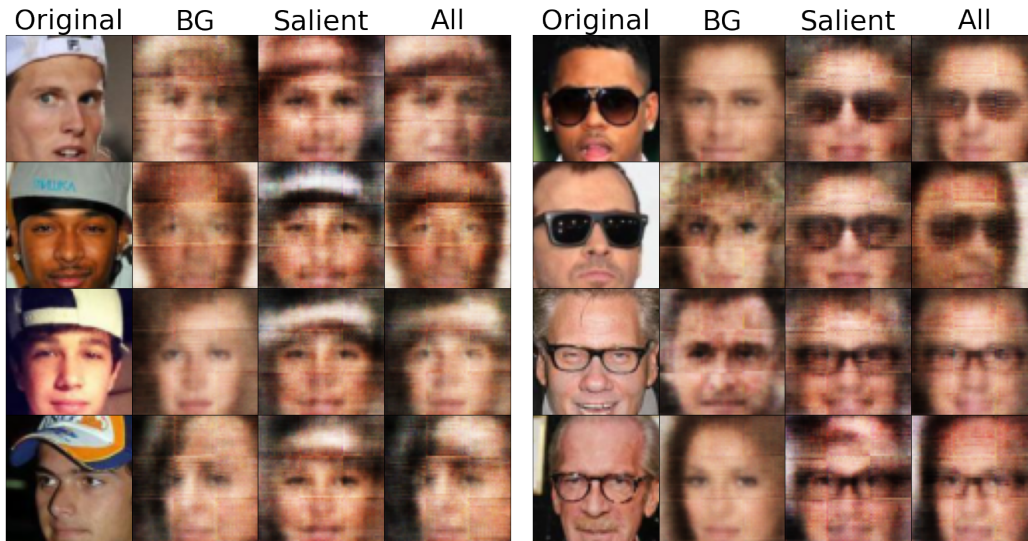
Because each term in the previous equation only depends on the inner products of the ψ vectors, we can rewrite it using the kernel trick to obtain

$$\widehat{\text{MMD}}(X, Y) = \frac{1}{N^2} \sum_{i=1}^N \sum_{i'=1}^N k(x_i, x_{i'}) - \frac{2}{NM} \sum_{i=1}^N \sum_{j=1}^M k(x_i, y_j) + \frac{1}{N^2} \sum_{j=1}^M \sum_{j'=1}^M k(y_j, y_{j'})$$

For a universal⁴ kernel such as the Gaussian kernel $k(x, y) = e^{-\gamma \|x-y\|^2}$, we can use a Taylor expansion to obtain an explicit feature map ψ that contains an infinite number of terms and which covers all orders of statistics. Minimizing the MMD for such a kernel function is equivalent to minimizing the distance between all moments of P_0 and P_1 , and it has been shown (Gretton et al., 2006) that asymptotically $\widehat{\text{MMD}}(X, Y) = 0$ if and only if $P_0 = P_1$.

E Qualitative Results for Baseline Models

E.1 CelebA



(a)

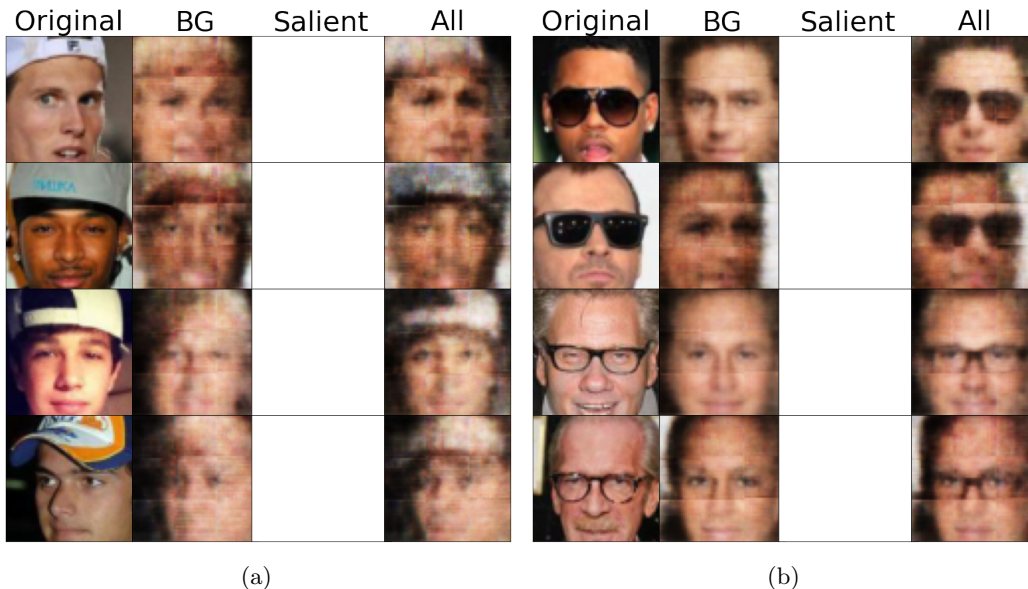
(b)

Supplementary Figure 3: Images generated with cVAE using the same procedure as in Figure 3 in the main text. From a visual inspection, we can see that salient and uninteresting factors of variation are not as well disentangled for these iamges as they were with MM-cVAE.

We first present qualitative results for the CelebA dataset for cVAE (Supplementary Figure 3), generated using the same procedure described in the main text for MM-cVAE. From these results, it appears that cVAE fails

⁴A kernel function $k(x, \cdot)$ is called universal if k is continuous for all x and the RKHS induced by k is dense in $\mathcal{C}(\mathcal{X})$, where $\mathcal{C}(\mathcal{X})$ denotes the space of continuous bounded functions on the compact domain \mathcal{X} .

to disentangle salient and background latent factors as effectively as MM-cVAE. For example, a clear outline of a hat is present in the topmost image in the background column in Supplementary Figure 3a. Similarly, dark glasses-like circles are present near the eyes in the BG column for the image in the row second from the bottom of Supplementary Figure 3b.



Supplementary Figure 4: Images generated with sRB-VAE using a slightly modified version of the procedure as in Figure 3 in the main text. Once again, we can see that salient and uninteresting factors of variation are not as well disentangled for these images as they were with MM-cVAE.

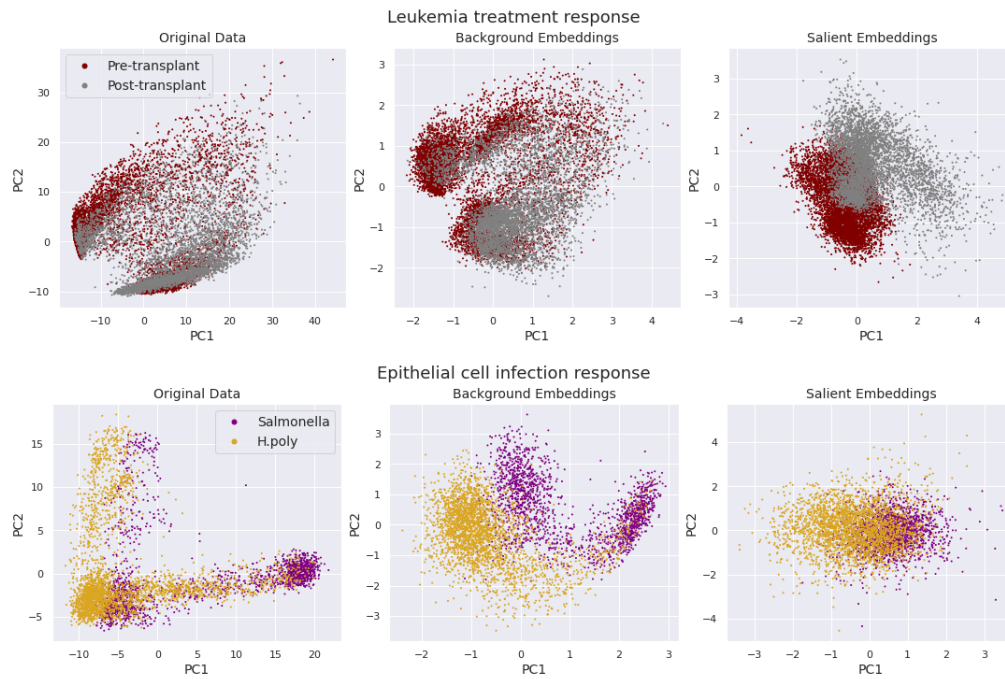
We next generated images using sRB-VAE (Supplementary Figure 4). Due to the details of the sRB-VAE architecture, we were forced to make some minor modifications to our image generation procedure. Rather than use a combination of zero vectors and zero bias terms to indicate the absence of features, sRB-VAE uses a learned vector of constants \mathbf{s}' to represent uninformative salient variable values for background data points. Thus, to generate images using only background variables, we concatenated images' inferred background variable values with this reference vector before decoding. From these generated images, we can see that sRB-VAE too seems to struggle with disentangling salient and background latent factors. For example, images generated using background latent variable values for celebrities with hats (Supplementary Figure 4a) all appear to have outlines of hats. Similarly, we can see glasses start to form in some images generated using background variables from celebrities with glasses. Specifically, we see sunglasses-like dark circles in our image in the second row of Supplementary Figure 4b, and faint glasses frames forming in the fourth row image.

Unfortunately, sRB-VAE does not provide a corresponding vector of uninformative values for background latent variables, preventing us from generating samples using only salient latent variable values as we did with cVAE and MM-cVAE.

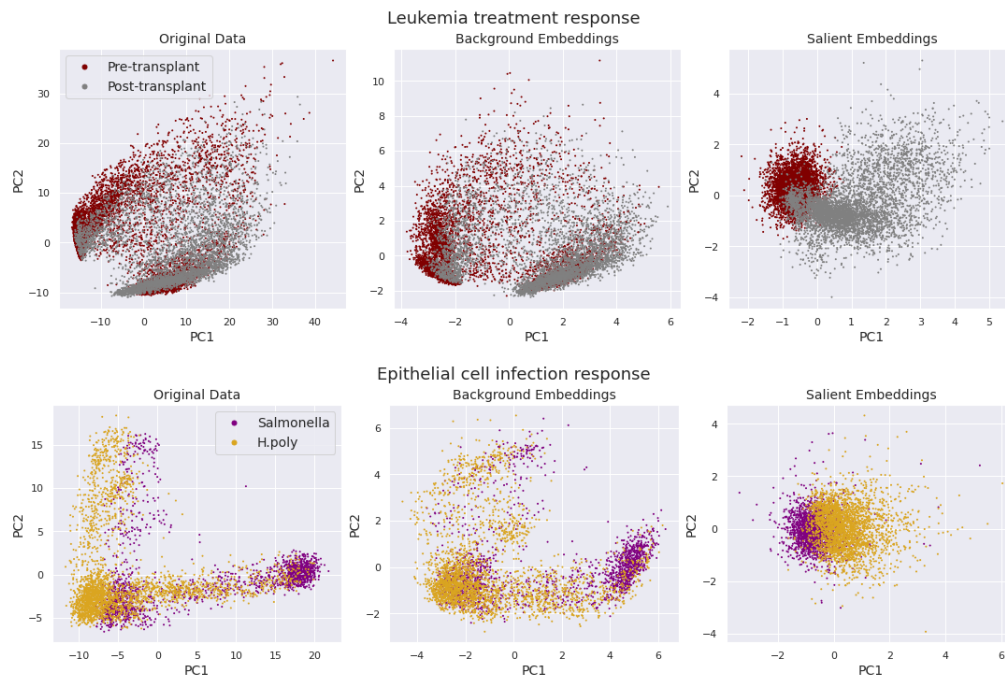
E.2 RNA-Seq data

In Supplementary Figure 5 we provided qualitative results for cVAE on our two RNA-seq datasets generated using the same procedure as in the main text for MM-cVAE. Here we can see that subgroups are less well-separated in the space spanned by the first two principal components of our salient latent variable values for both datasets than they were for MM-cVAE. This effect is especially pronounced for the Epithelial cell data. Moreover, we find a strong (undesirable) separation between subgroups in the space spanned by the first two PCs for background latent variable values for the Epithelial cell dataset.

In Supplementary Figure 6 we provide the corresponding results for sRB-VAE. We similarly find that subgroups are less well separated in the salient latent space for sRB-VAE than they were with MM-cVAE.



Supplementary Figure 5: Qualitative RNA seq results for cVAE. Subgroups are not as well separated in the first two PCs of the salient embedding spaces for both datasets as they were with MM-cVAE; this is especially noticeable for the Epithelial cell dataset. Moreover, we see clear separation between subgroups in the first two PCs of the background latent variables for Epithelial cells.



Supplementary Figure 6: Qualitative RNA seq results for sRB-VAE. While subgroups are (correctly) well mixed in the background latent space, we find more (undesirable) mixing in the salient space than we did with MM-cVAE.

F Sample Quality Results

Here we provide the quantitative results measuring sample quality referenced in the main text.

Supplementary Table 1: Frechet Inception Distance scores for MM-cVAE and baseline models on the CelebA dataset.

Model	Background	Target
sRB-VAE	335.18 \pm 14.04	299.70 \pm 13.69
cVAE	203.05 \pm 4.33	223.91 \pm 8.83
MM-cVAE	197.05 \pm 6.44	224.73 \pm 5.46
VAE	174.39 \pm 15.23	240.38 \pm 3.55

Supplementary Table 2: Sampled quality as measured by the maximum mean discrepancy for the two RNA-seq datasets for MM-cVAE and baseline models.

Dataset	Model	Sample Quality (MMD)	
		Background	Target
AML (Pre- vs. post-transplant patients)	sRB-VAE	0.37 \pm 0.02	0.22 \pm 0.02
	cVAE	0.20 \pm 0.01	0.10 \pm 0.01
	MM-cVAE	0.14 \pm 0.01	0.08 \pm 0.01
Epithelial Cells (Mice infected with <i>Salmonella</i> vs. <i>H. poly</i>)	sRB-VAE	0.20 \pm 0.02	0.22 \pm 0.01
	cVAE	0.15 \pm 0.01	0.13 \pm 0.01
	MM-cVAE	0.11 \pm 0.01	0.13 \pm 0.01

G Details on Quantitative Metrics

G.1 Logistic Regression Score

For a given set of embeddings, we trained a logistic regression classifier on 80% of the embeddings, holding out the remaining 20% for evaluation. In our evaluation, we used scikit-learn’s logistic regression solver with the `max_iter` parameter set to 1000 and all other parameters left at their default values. We report the accuracy numbers generated by the `score` function of our classifier Python objects.

G.2 Silhouette Score (SS)

For a given data point i , the SS is defined as $SS(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$, where $a(i)$ is the average distance from i to other points with the same class label, and $b(i)$ is the average distance to points in the next nearest cluster. For a dataset, the SS is the mean $SS(i)$ for all of the points in that dataset. To calculate the SS, we used the `silhouette_score` function in the scikit-learn Python library.