Adversarial Canonical Correlation Analysis

Benjamin Dutton

Abstract  Canonical Correlation Analysis (CCA) is a statistical technique used to extract common information from multiple data sources (views). It has been used in various representation learning problems, such as dimensionality reduction, word embedding, and clustering. Recent work has given CCA probabilistic footing in a deep learning context and uses a variational lower bound for the data log likelihood to estimate model parameters. Alternatively, adversarial techniques have arisen in recent years as a powerful alternative to variational Bayesian methods in autoencoders. In this work, we explore straightforward adversarial alternatives to recent work in Deep Variational CCA (VCCA and VCCA-Private) we call ACCA and ACCA-Private and show how these approaches offer a stronger and more flexible way to match the approximate posteriors coming from encoders to much larger classes of priors than the VCCA and VCCA-Private models. This allows new priors for what constitutes a good representation, such as disentangling underlying factors of variation, to be more directly pursued. We offer further analysis on the multi-level disentangling properties of VCCA-Private and ACCA-Private through the use of a newly designed dataset we call Tangled MNIST. We also design a validation criteria for these models that is theoretically grounded, task-agnostic, and works well in practice. Lastly, we fill a minor research gap by deriving an additional variational lower bound for VCCA that allows the representation to use view-specific information from both input views.

Keywords  Multiview Learning, Representation Learning, Adversarial Learning, Variational Bayesian Methods

1 Introduction

In multi-view learning (MVL) problems, multiple data sources or views are available at training time and the assumption is made that there is a high degree of information overlap between them. These data sources can be artificially induced
by simply partitioning features into sets or can correspond to distinct physical, real world sensors monitoring a common information source [21], such as multiple cameras pointing at a common object. MVL is a generalization of the multi-modal learning problem, which describes learning scenarios where different sensors utilize different modalities, such as images, audio or text. In the most comprehensive survey on multi-view learning methods to date [25], Xu et. al. formalize the common assumptions of MVL by identifying two principles universal to all MVL algorithms:

1. **Consensus principle**: methods operating on different views should largely agree on what they find in each view
2. **Complementary principle**: there may exist some information in each view, not found in other views, that can be exploited for learning tasks

Historically, research in multi-view representation learning (MVRL) [18] has largely focused on the first principle by seeking to exploit the information overlap across views. For instance, Multiview Subspace Learning, largely based on Canonical Correlation Analysis (CCA) [11], seeks to maximize agreement between information extracted between views, measured in terms of correlation. Similar approaches to MVRL seek to also maximize agreement, but use different criteria such as distance or alternative similarity measures [18]. Together, these form alignment-based approaches to MVRL [18].

Recently, [23] introduced two probabilistic forms of CCA, one based on the probabilistic graphical model (PGM) of [3] but that uses deep networks instead of linear models they call VCCA and, of more interest to us, one that segregates view-specific latent variables from view-common in a PGM they call VCCA-Private. We believe that models such as VCCA-Private better capture the universal assumptions made in MVL because they segregate view-common latent variables to capture the information overlap between views (addressing the consensus principle) from view-specific latent variables to capture information particular to each view (addressing the complementary principle). This allows full information preservation to be sought but in an orderly, disentangled manner which is a promising bias [4] in representation learning at the moment [6,20,13,5].

In [23], Wang et al. also bring together two veins of CCA-based MVRL research: deep learning approaches to CCA and probabilistic interpretations of CCA. The difficulty with using the models they introduce (both with and without Private information), however, is that there is no closed-form maximum likelihood estimate for the network parameters. Like [14] do with Variational Autoencoders (VAE), Wang et al. overcome this by deriving a variational lower bound (also known as evidence lower bound or ELBO) that decomposes the loss function into components that seek to a) maximize the log probability of reconstructions and b) minimize the KL-divergence between the aggregated posterior coming from encoders with chosen priors.

However, there are limitations to maximizing the ELBO for a model of this structure: the KL-divergence between the prior(s) and aggregated posterior(s) needs to be a known, differentiable function and there are, arguably, better ways to match aggregated posteriors to priors, such as by using adversaries [7,19] or by using Maximum Mean Discrepancy (MMD) [8,26].

Adversarial Autoencoders (AAE) [19] took the existing VAE model and showed how discriminators could be used in place of the KL-divergence terms to allow multiple, straightforward extensions to the model: a) the use of a much larger
class of priors that do not require known, differentiable KL divergence terms b) the integration of known label information in a partially supervised setup as well as c) unsupervised clustering d) semi-supervised extensions and e) dimensionality-reduction extensions.

In this work, we introduce Adversarial Canonical Correlation Analysis (ACCA) in two forms which match the model assumptions of VCCA and VCCA-Private, that we correspondingly call ACCA and ACCA-Private. These models address a gap in the multi-view representation learning (MVRL) research landscape and offer similar extensions to the VCCA model that AAE offers to VAE, showing how adversaries can be used in place of differentiable KL divergence terms to match approximate posteriors to priors. Although we believe that all of the extensions introduced in AAE are possible and straightforward to use with ACCA, in this work we focus largely on the use of arbitrary priors and the goodness of fit to those priors that adversaries can provide. We leave other extensions to future work. In this work, we introduce Adversarial Canonical Correlation Analysis (ACCA) in two forms which match the model assumptions of VCCA and VCCA-Private, that we correspondingly call ACCA and ACCA-Private. These models address a gap in the multi-view representation learning (MVRL) research landscape and offer similar extensions to the VCCA model that AAE offers to VAE, showing how adversaries can be used in place of differentiable KL divergence terms to match approximate posteriors to priors. Although we believe that all of the extensions introduced in AAE are possible and straightforward to use with ACCA, in this work we focus largely on the use of arbitrary priors and the goodness of fit to those priors that adversaries can provide. We leave other extensions to future work.

We also aim to highlight and motivate a new perspective of analysis largely missing from MVRL that is becoming prominent in other areas of Representation Learning research. We believe this perspective on the proper aims of representation learning offers new theoretical insights inline with the universal principles of [25]: namely, that the purpose of representation learning is to disentangle underlying factors of variation. Conveniently, the VCCA-Private model of [23] and ACCA-Private we propose here allows this hypothesis to be explored. We aim for this to be the first step in multi-view disentangling as a promising research direction, as it has become in general representation learning research.

1.1 Our contributions and where they can be found

1.1.1 We design the ACCA and ACCA-Private algorithms and task-agnostic validation criteria

In section three, we present the ACCA and ACCA-Private models and training algorithms, along with their task-agnostic validation criteria.

1.1.2 We show that ACCA and ACCA-Private provide increased flexibility in choosing priors over VCCA and VCCA-Private, allowing new biases to be pursued in multi-view representations

This follows from work on adversarial density estimation [7] and the flexibility that adversarial approaches offer at matching posteriors to priors. For work in this area
specific to autoencoders, see [19]. We demonstrate this flexibility experimentally in section 4.4.

1.1.3 We fill a minor research gap in VCCA by deriving variational lower bounds in terms of both views

This is derived in section 2.4.2.

1.1.4 We demonstrate that ACCA acts as a stronger regularizer on the posterior than VCCA, which comes at the expense of overall information content if the network is not powerful enough but allows better fit of posteriors to priors

We show this experimentally in section 4.2. For information that is clearly not Gaussian (categorical class information), the variational approach of VCCA fits the rough shape of the posterior to the prior, but allows large fissures, demonstrating a compromise between the discrete underlying aggregate posterior and the continuous prior. The adversarial approach of ACCA, on the other hand, forces the gaps to close (because those gaps in the posterior get recognized easily by the discriminator), leading to a closer match between the posterior and prior, even though it comes at the expense of the overall information content. This demonstrates that the adversarial approach to matching the distributions can be thought of as a stronger regularizer than the variational approach of VCCA, which can be overcome by increasing the capacity of the network (which we show in the latter half of section 4.2).

1.1.5 We perform new analysis on VCCA and VCCA-Private from a view-level disentangling perspective

In sections 4.2 through 4.4, we focus on highlighting where underlying factors of variation can be found. For VCCA and ACCA, in 4.2, we show how the factors of variation (class, style, angle of rotation for each of the views) is distributed across the dimensions of the latent representation. In section 4.3, for VCCA-Private and ACCA-Private, we show how this information is distributed across the three latent representations in addition to the individual dimensions in each. Similarly, in section 4.4, we show how class information gets distributed across the posterior. We take this perspective because we think that the disentangling properties of VCCA and ACCA - both across views and within each view are the most exciting future direction in multiview representation learning at the moment and that these frameworks offer a good theoretical foundation for such research.

1.1.6 We construct a new dataset, Tangled MNIST, which is more appropriate for evaluating multi-view representation learning algorithms than existing benchmark datasets, such as Noisy MNIST

Because we prioritize the disentangling perspective in this work, we move away from the Noisy MNIST dataset of [22,23]. We are interested in disentangling in multiview contexts from a layered perspective within a VCCA-Private or ACCA-Private framework, where there are three latent variables - one particular to each
view and one common across views. We call these $z_x$, $z_y$, and $z_c$ to indicate the view-$x$ specific latent variable, view-$y$ specific latent variable, and the common latent variable (respectively). Information particular to one view or the other (therefore, not common) should show up in that view’s corresponding latent variable. And, information that is common should show up in $z_c$. That is the first level of disentanglement. The second level occurs within the dimensions of the latent random variables, exactly like current disentangling research in single view contexts. To analyze the algorithms from this perspective, we need known factors of variation for each of the latent variables - some factors specific to each view and some that are common. The existing benchmark dataset does not contain factors of variation particular to view $y$ - they use independent noise for each dimension of the view, which is incompressible and we have no hope of recovering it in $z_y$. We propose the new dataset, Tangled MNIST, in section 4.1.

2 Background

2.1 Standard Canonical Correlation Analysis

Canonical Correlation Analysis (CCA) [11] forms the basis for much research in MVRL [18], including our work. In CCA, vectors $w_{x,0}$ and $w_{y,0}$ are sought for views $X = [x_1, ..., x_N]$ and $Y = [y_1, ..., y_N]$ that maximize the correlation between linear projections $A_0 = w_{x,0}^X X$ and $B_0 = w_{y,0}^Y Y$. Additional vectors $w_{x,i}$ and $w_{y,i}$ can be sought, subject to the restriction that they are uncorrelated with earlier vectors. In matrix form, when $m$ projections are sought, all the projection vectors can be combined into matrices $W_x = [w_{x,0}, ..., w_{x,m}]$ and $W_y = [w_{y,0}, ..., w_{y,m}]$ and CCA rewritten as [9]:

$$\max_{W_x, W_y} \text{Tr}(W_x^T C_{x,y} W_y)$$

subject to

$$W_x^T C_{x,x} W_x = I,$$
$$W_y^T C_{y,y} W_y = I,$$
$$w_{x,i}^T C_{x,y} w_{y,j} = 0,$$
for $i, j \in \{1, ..., m\}, i \neq j$ (1)

where $C_{x,x}$, $C_{y,y}$, and $C_{x,y}$ are the covariance matrices of $x$, $y$ and between $x$ and $y$, respectively.

2.2 Nonlinear Canonical Correlation Analysis

One of the main limitations of standard CCA is the reliance on simple linear projections. When written in the form above, it is easy to see how kernel matrices could replace the covariance matrices, using the ”kernel trick” [10] and provide a nonlinear extension to the linear projection functions [11][17][9].

Neural network extensions to CCA similarly allow complex, nonlinear hypothesis spaces for each view’s projection function. In [15], Lai and Fyfe demonstrate a way to use a neural network to maximize the correlation between individual
projections for each view, but the network is simple and the function is still linear. In [14], they extend their work to multiple projections and introduce simple nonlinearities through the use of activation functions.

In [12], Hsieh made two breakthroughs relevant for this work. They used multilayer perceptrons for each projection function (which relies on a negative correlation loss function) and added networks that try to reconstruct the inputs from the projections, essentially situating CCA in a multiview autoencoder framework, an important precedent for this work.

The DCCA [2] model of Andrew et al. was arguably the first to explore the use of deep networks. They do not use reconstructions, but allow multiple projections and seek to maximize the total correlation between the outputs. Unfortunately, the loss function requires an expensive correlation calculation across a batch or the entire dataset. They derive the gradient for the loss function but recommend using the entire dataset instead of batches after experimenting with using batches, limiting the utility of the approach.

In [22], Wang et al. devise three variations of DCCA. All three models use deep networks to generate projections and use the same correlation loss term, but they also include additional decoder networks that seek to reconstruct each view from the projections. The first model, DCCAE, uses the same loss function and constraints (uncorrelation and normalization) as DCCA, but includes reconstruction terms in the loss function. The second model, CorrAE, removes the uncorrelation constraint of DCCAE and uses the sum of scalar correlations between the projections. The third model, DistAE, replaces the correlation term from DCCA with a distance-based criteria. The authors motivate this using the work of [9], who show that CCA can be understood as minimizing the distances between the projections as long as they meet the uncorrelation and normalization (whitening) constraints. It is also worth noting, as discussed in [18], that distance-based techniques are another type of alignment-based approaches to MVRL, so they share some theoretical grounding.

2.3 Probabilistic Canonical Correlation Analysis

One limitation of the modern deep CCA methods described above is their reliance on expensive correlation losses that must be computed over, at a minimum, batches. There is another vein of CCA-based MVRL research, however, based ultimately on the probabilistic interpretation of CCA found in [3] (PCCA) that has allowed deep networks to bypass this restriction. With PCCA, Bach and Jordan offered a latent variable probabilistic interpretation of standard CCA using the PGM found in Figure 1a, with $z$ as the latent variable for observed views $x$ and $y$. This model has a factorization for $p(x, y, z)$ of $p(x, y, z) = p(z)p(x|z)p(y|z)$. They make the following distribution assumptions:

\[
\begin{align*}
p(z) &= \mathcal{N}(0, I_d) \\
p(x|z) &= \mathcal{N}(W_x z + \mu_x, \phi_x) \\
p(y|z) &= \mathcal{N}(W_y z + \mu_y, \phi_y)
\end{align*}
\]
Fig. 1: The probabilistic CCA model (a) introduced by [3] established CCA on probabilistic footing, but found maximum likelihoods for distributions assuming only simple linear projection functions. The VCCA model (b) of [23] replaces these linear projection functions with deep neural networks, making closed form maximum likelihood estimates impossible. Instead, they derive a variational lower bound for the data log likelihood requiring the use of an encoder network $q_{\phi}(z|x)$ to approximate the single view posterior $p(z|x)$. In (c), we show the network structure using the variational lower bound we derive in section 2.4.2 which allows $z$ to contain information from both views.

Where $\phi_x$ and $\phi_y$ are positive semidefinite matrices, $\min\{m_x, m_y\} \geq d \geq 1$, $x \in \mathbb{R}^{m_x}$, $y \in \mathbb{R}^{m_y}$, and $z \in \mathbb{R}^d$. They show that the maximum likelihood estimates for this model lead to the standard CCA solution.

2.4 Variational Canonical Correlation Analysis

While providing good probabilistic footing for CCA, PCCA suffers from the same linearity limitation for the projection functions as CCA. Wang et al. overcome this in [23] with models VCCA and VCCA-Private. Together, they form the primary basis for our work so we devote this section to understanding them.

VCCA uses the same graphical model as PCCA shown in Figure 1a but radically changes the distribution assumptions by replacing the linear projection functions with deep neural networks, $g_x(z; \theta_x)$ and $g_y(z; \theta_y)$, where $\theta_x$ parameterizes the neural network for view $x$, $\theta_y$ parameterizes the neural network for view $y$, and when $\theta$ is used without a subscript, it refers to the set of network parameters for both models combined. The new distribution assumptions VCCA makes are then:

\[
\begin{align*}
p(z) &= \mathcal{N}(0, I) \\
p(x|z) &= \mathcal{N}(g_x(z; \theta_x), I) \\
p(y|z) &= \mathcal{N}(g_y(z; \theta_y), I)
\end{align*}
\]

The resulting model, while significantly more expressive, makes straightforward maximum likelihood estimation of the model parameters impossible. To address this, they use the approach Kingma et al. [14] take with variational autoencoders.
(a) The graphical model of VCCA-Private.
(b) The network structure of VCCA-Private with $z$ a function of just one view.
(c) The network structure of VCCA-Private where $z$ is a function of both views. New connection highlighted for emphasis.

Fig. 2: The VCCA-Private model of [23], with the graphical model of the joint distribution in (a) and the network structure (b) employed to maximize the data likelihood via maximizing the variational lower bound of the data likelihood.

(VAE) and situate the model within an autoencoder framework (see Figure 1b) and use the encoder network(s) to help maximize a variational or evidence lower bound (ELBO) on the data log likelihood coming from the generative model provided by the decoder.

There is one other limitation, though, they address that is of particular interest to us. CCA naturally exploits the consensus principle (discussed in our introduction) of [25] because both observed views $x$ and $y$ rely on a common latent variable, $z$. While $z$ can contain view-specific information for each view (after using the variational lower bound we derive above in terms of both views), it is not clear in what manner and certain VCCA architectures explicitly prevent this possibility (to be discussed shortly).

Disentangled representation learning, on the other hand, has become a promising research direction in representation learning [4,6,20,13,5] where information from underlying factors of variation are isolated to individual latent dimensions. There is no reason in VCCA to think that these view-specific factors of variation will be isolated to individual dimensions of $z$. VCCA-Private provides a form of view-disentanglement by isolating view-specific latent variables ($z_x$ and $z_y$ for views $x$ and $y$, respectively) from view-common ($z_c$). Although no effort is made to isolate individual factors of variation in dimensions of these variables, it is a good first step in our opinion. See Figure 2a for the underlying PGM of VCCA-Private and Figure 2b for the network structure. This approach to multi-view disentanglement matches the theory of [25] and prevents view-specific information from being tangled with view-common in $z_c$, allowing more informed uses from learned representations.

2.4.1 VCCA, Single View Encoder

In [23], Wang et al. derive an ELBO for VCCA and use an encoding distribution for $z$, $q_\phi(z|x)$ defined in terms of just one view as an estimate of the posterior. In other words, $q_\phi(z|x)$ is used as an estimate of $p(z|x)$ instead of using $q_\phi(z|x,y)$ as
an estimate of $p(z|x, y)$ in the ELBO. Practically speaking, this means that the encoder for $z$ can only, in theory, be expressed as a function of one view. They argue that this allows certain uses, such as still functioning when only one view is available at test time. They additionally explore using a convex combination of the losses from encoders over each view independently.

The ELBO they derive is as follows, which we denote $\mathcal{L}_{VCCA}$ (all following expectations are with respect to the data distribution unless otherwise specified):

$$
\log p_\theta(x, y) \\
\geq \frac{-D_{KL}(q_\phi(z|x) || p(z))}{\text{Distribution matching term}} + \mathbb{E}_{q_\phi(z|x)} \left( \log p_\theta(x | z) + \log p_\theta(y | z) \right) \quad \text{Reconstruction loss terms}
$$

$$
= \mathcal{L}_{VCCA}(x, y; \theta, \phi)
$$

Note that this derivation is general to any choice of distribution for $p(z)$ and $q_\phi(z|x)$, with the only requirement being that the KL divergence between the two distributions is closed form and differentiable.

2.4.2 VCCA, Two View Encoder

As the authors discuss [23], in scenarios where only one view is available at test time, it is useful to have the encoder network be a function of only one view. However, this places an important restriction on the information $z$ can contain in VCCA. While it can contain the common information between the views and $x$-specific information, it cannot contain $y$-specific information due to the structural limitations of the derivation. While we believe the best approach is to disentangle each of these into separate latent variables, as is done in VCCA-Private and the ACCA-Private method we propose in this work, here we fill a minor research gap by deriving a variational lower bound for VCCA in terms of both views, using $q_\phi(z|x, y)$:

$$
\log p_\theta(x, y) = \log p_\theta(x, y) \int q_\phi(z|x, y) dz
\geq \int \log p_\theta(x, y) q_\phi(z|x, y) dz \\
= \int \log p_\theta(x, y) q_\phi(z|x, y) dz + \log \frac{p_\theta(x, y, z)}{q_\phi(z|x, y)} dz \\
= \text{Distribution matching term} + \mathbb{E}_{q_\phi(z|x, y)} \left( \log \frac{p_\theta(x, y, z)}{q_\phi(z|x, y)} \right) \\
\geq \int q_\phi(z|x, y) \log \left( \frac{p_\theta(x, y, z)}{q_\phi(z|x, y)} \right) dz \\
= \int q_\phi(z|x, y) \log \left( \frac{p_\theta(x|z)p_\theta(y|z)p(z)}{q_\phi(z|x, y)} \right) dz \\
= \text{Distribution matching term} + \mathbb{E}_{q_\phi(z|x, y)} \left( \log \frac{p_\theta(x|z)p_\theta(y|z)}{q_\phi(z|x, y)} \right)
$$

$$
= \mathcal{L}_{VCCA}(x, y; \theta, \phi)
$$
2.4.3 VCCA-Private

In addition to VCCA, Wang et al. introduce a second model they call VCCA-Private which, we argue, makes much more realistic multi-view assumptions. The PGM for this model can be found in Figure 2a. VCCA-Private introduces two new latent variables, one for each view: $z_x$ and $z_y$. $z_x$ is particular to view $x$ and $z_y$ is particular to view $y$ while $z$ is still shared by both. We think this model better gets at the heart of the assumptions made in multi-view data [25] described in the introduction: consensus information should reside in $z$ and complementary information should reside in $z_x$ and $z_y$.

\[
p(z) = \mathcal{N}(0, I)
\]

\[
p(x|z, z_x) = \mathcal{N}(g_x(z, z_x; \theta_x), I)
\]

\[
p(y|z, z_y) = \mathcal{N}(g_y(z, z_y; \theta_y), I)
\]

Wang et al. similarly derive the variational lower bound for this model and arrive at

\[
\log p_\theta(x, y) \geq -D_{KL}(q_\phi(z|x)||p(z)) - D_{KL}(q_\phi(z_x|x)||p(z_x)) - D_{KL}(q_\phi(z_y|y)||p(z_y))
\]

\[\text{Distribution matching terms}\]

\[+ \mathbb{E}_{q_\phi(z|x), q_\phi(z_x|x)} \log p_\theta(x|z, z_x) + \mathbb{E}_{q_\phi(z|x), q_\phi(z_y|y)} \log p_\theta(y|z, z_y)
\]

\[\text{Reconstruction loss terms}\]

\[= \mathcal{L}_{Private}(x, y; \theta, \phi)\]

Readers familiar with variational autoencoders [14] will likely recognize the form of these lower bounds as consisting of two parts. The KL divergence terms match user-chosen priors to aggregated posteriors coming from encoder networks and the expected data log likelihood terms maximize the log probability of the observed data. In practice the KL divergence terms require well behaved, differentiable expressions and the expected log probability of the reconstructions can be replaced with a reconstruction loss term (usually assumptions are made about the distributions to make these equivalent).

The goal, then, of training is twofold: fit the aggregated posteriors to the user-chosen priors and minimize reconstruction error. In [19], Makhzani et al. show how the first goal can be better achieved using adversaries and introduce Adversarial Autoencoders (AAE). Why? First, they allow a much larger class of priors to be chosen from. The restriction is no longer that a known, differentiable expression for the KL divergence between the aggregated posterior and prior is known. Instead, the priors must simply be able to be sampled from. Second, adversaries arguably do a better job of matching priors to posteriors than KL divergence does. So, the primary research gap we aim to address in our work can be summarized as this: as AAE is to VAE, ACCA is to VCCA and ACCA-Private is to VCCA-Private.

Before moving on to a description of the model and how it is trained, we should mention the work of [24] since it is both inspired by VCCA and uses adversaries. However, it is not a direct extension of VCCA to using adversaries. It
instead employs multiple adversarial autoencoders in addition to multiple cross-view autoencoders. The aim of this work is to fill the gap described above.

3 Adversarial Canonical Correlation Analysis

In this section we present our ACCA model

3.1 ACCA Overview

Before explaining the structure and mechanics of ACCA, we start with the simple adversarial autonencoder model of Figure 3a. In this model, inputs $x$ are encoded to $z$ using an encoder network $q_\phi(z|x)$ with parameters $\phi$. We choose a distribution $p(z)$ from which we can sample and we wish to match $q_\phi(z|x)$ to $p(z)$ as closely as possible, as a function of the network parameters $\phi$. A decoder network $p_\theta(x|z)$ parameterized by $\theta$ maps $z$ to $\hat{x}$ and we wish to minimize the difference between $x$ and $\hat{x}$.

In VCCA, $z_i \sim \mathcal{N}(\mu(x_i), \sigma(x_i))$ with $\mu(x_i)$ and $\sigma(x_i)$ the actual outputs of the encoder. In other words, the encoder outputs the parameters of the Gaussian that $z_i$ is sampled from, providing an additional source of variability to $z_i$, beyond the stochasticity of $x_i$. In AAE [19], Makhzani et al. explore this approach to the encoders, in addition to two others: deterministic and universal approximator posterior. In the deterministic approach, $z_i$ is a deterministic function of $x_i$ and the only source of randomness comes from $x_i$. In the universal approximator posterior approach, multiple samples from a noise distribution are added to the input $x_i$, of the encoder and then averaged out. They find that there is no noticeable difference between each version of $q_\phi(z|x)$ and use the deterministic approach. We do the same for all encoders: $q_\phi(z|x,y)$ in ACCA and $q_\phi(z|x)$, $q_\phi(z_x|x)$ and $q_\phi(z_y|y)$ in ACCA-Private.

3.2 Training ACCA

The adversarial game being played between the encoder $q_\phi(z|x)$ and the discriminator networks $D_{z;\psi}$ (parameterized by $\psi$) is used to match $q_\phi(z|x)$ to $p(z)$ and is a competition between the discriminator (which aims to differentiate between the two distributions) and the encoder (which aims to fool the discriminator, in addition to providing embeddings that allow good reconstructions though this requirement is outside the game and comes from the reconstruction loss). The adversarial game between the encoder and discriminator can be written as:

$$\min_{\phi} \max_{\psi} \mathbb{E}_{z \sim p(z)}[\log D_{z;\psi}(z)] + \mathbb{E}_{x \sim p_{\text{data}}}[\log(1 - D_{z;\psi}(q_\phi(z|x,y)))]$$

This game, which replaces the KL divergence terms of the VCCA loss, requires two separate training passes: a pass to update the discriminator parameters $\psi$ and a pass to update the encoder parameters $\phi$. A third and final pass to update all the autoencoder parameters, $\phi$ and $\theta$, uses reconstruction loss only and corresponds to the data log likelihood terms of the VCCA ELBO. We describe each of the three passes next.
3.2.1 Identifying the Frauds: Discriminator Update

In this pass, a batch $X_k$ is constructed containing $n$ samples from $X$ and then passed through the encoder to yield a batch of latent values which we denote $NEG_k$. Similarly, we draw $n$ samples from $p(z)$ placed into a batch called $POS_k$. The batch $NEG_k$ is considered the negative batch to the discriminator $D_{z;\psi}$ because it is trying to recognize elements of that batch as imposters and samples from $p(z)$ (the $POS_k$ batch) as the true or positive class.

$NEG_k$ and $POS_k$ are concatenated and fed to $D_{z;\psi}$, which is a deep neural network with one output neuron followed by a sigmoid activation function. Binary cross entropy loss is used with elements of $NEG_k$ using class label $y = 0$ and elements of $POS_k$ using class label $y = 1$. The discriminator loss is then:

$$
L_{disc} = -\frac{1}{2n} \sum_{i=1}^{i=n} y_i \log D_{z;\psi}(z_i) + (1 - y_i) \log(1 - D_{z;\psi}(z_i))
$$

and the discriminator parameters $\psi$ alone (not the encoder parameters $\phi$ even though the encoder was used in the forward pass) are updated as $\psi := \psi - \lambda \frac{dL_{disc}}{d\psi}$.

3.2.2 Fooling the Discriminator: Encoder Update

In this pass, the encoder has the opposite objective as the discriminator did in the prior pass: its aim is to fool the discriminator. For this pass, a negative batch $NEG_k$ is constructed in the same manner. However, the labels for the batch are switched to $y = 1$ and instead of updating the discriminator parameters, $\psi$, we update the parameters of the encoder, $\phi$, using loss

$$
L_{gen} = \frac{1}{2n} \sum_{i=1}^{i=n} y_i \log D_{z;\psi}(z_i)
$$

and encoder parameters $\phi$ alone (not the discriminator parameters $\psi$ even though the network was used in the forward pass) are updated as follows $\phi := \phi - \lambda \frac{dL_{gen}}{d\phi}$.

Notice the change in sign in the loss function. The encoder’s aim is to maximize the likelihood of fooling the discriminator into mistaking negative batches as positive.

3.2.3 Maximizing Data Log Likelihood (Minimizing Reconstruction Loss): Full Autoencoder Update

The adversarial game described above served only to replace the KL divergence terms of the VCCA ELBO and does not in any way seek to maximize the data log likelihood terms of the ELBO. So, to address those data log likelihood terms, a third and final pass is made through the full autoencoder. Batch $X_k$ is sent through both $q_\phi(z|x)$ and $p_\theta(x|z)$ to yield $\hat{X}_k$. Reconstruction loss is then calculated as:

$$
L_{recon} = \||x - \hat{x}||_k
$$

with $k \in \{1, 2\}$. 
and both encoder $\phi$ and decoder $\theta$ parameters are updated as $\phi := \phi - \lambda \frac{dL_{\text{recon}}}{d\phi}$ and $\theta := \theta - \lambda \frac{dL_{\text{recon}}}{d\theta}$.

### 3.3 ACCA Validation Criteria

One of the challenges with representation learning algorithms, in general, is how to design a principled validation criteria that is not overly task-biased. In [23], Wang et. al use classification accuracy from a linear SVM as the validation criteria for VCCA and VCCA-Private, arguing that it is a common use case for a representation learned on that dataset and the limited expressive power of the classifier does not itself aid the representation. We seek a less task-specific bias. Because VCCA and VCCA-Private have a single loss value and representation learning is an unsupervised task, we use the best loss value during training as the stopping criteria.

For ACCA, because training is more complicated and involves three separate training passes with their own losses and adversarial training is inherently less stable, we explored a few different choices. Binary cross entropy is used on both the discriminator and generator pass of training ACCA and ACCA-private, where $D(z_i)$ is the output of the discriminator representing the probably that $z_i$ comes from prior $p(z)$. It can be written as

$$D \text{Loss}(\text{batch}) = -\frac{1}{N} \sum_{i=1}^{N} y_i \log(D(z_i)) + (1 - y_i) \log(1 - D(z_i))$$

(2)

Where $y_i \in \{0, 1\}$ are labels used to indicate whether the sample came from $p(z)$ or $q(z|x)$, respectively (or whatever latent variable the discriminator acts on). The generator loss is similar but uses the opposite labels, containing only samples from an encoding distribution, (e.g. $q(z|x)$), but using $y_i = 0$:
\[ \text{Gloss(batch)} = -\frac{1}{N} \sum_{i=1}^{N} y_i \log(D(z_i)) + (1 - y_i) \log(1 - D(z_i)) \]  
\[ = -\frac{1}{N} \sum_{i=1}^{N} \log(1 - D(z_i)) \]  
(3)

In theory, \( p(z) \) is matched to \( q(z|x) \) when the discriminator loss and generator loss are both equal at random chance, which reduces to \(-\log(0.5) \approx 0.693147\). And, all things being equal, we want to minimize reconstruction error. Because of this, we use the following as the validation criteria on ACCA and find it to be a good validation criteria that balances goodness of fit to priors as well as general information content in the representations:

\[ \text{ValACCA} = | -\log(0.5) - \text{DLoss}| + | -\log(0.5) - \text{GLoss}| + \text{ReconLoss} \]  
(4)

3.4 ACCA-Private Training

Training of ACCA-Private proceeds in the same manner as ACCA except multiple embeddings are computed for each batch, corresponding to \( z, z_x, \) and \( z_y \). These happen in parallel during the first two passes since each of the encoders and discriminators are independent of each other. The last stage, where reconstruction loss is calculated, is entirely feedforward but not independent: the decoder for \( x \) relies on \( z \) and \( z_x \) and the decoder for \( y \) relies on \( z \) and \( z_y \).

3.5 ACCA-Private Validation Criteria

For ACCA-Private, the criteria is the same except \( \text{DLoss} \) and \( \text{GLoss} \) are the total discriminator and generator losses, respectively. That is:

\[ \text{DLoss} = \frac{| -\log(0.5) - \text{Dloss}_{z_x}| + | -\log(0.5) - \text{Dloss}_{z_y}| + | -\log(0.5) - \text{Dloss}_{z_y}|}{3} \]  
(5)

and

\[ \text{GLoss} = \frac{| -\log(0.5) - \text{Gloss}_{z_x}| + | -\log(0.5) - \text{Gloss}_{z_y}| + | -\log(0.5) - \text{Gloss}_{z_y}|}{3} \]  
(6)

4 The Tangled MNIST Dataset

Wang et al. [22] introduced a dataset they call Noisy MNIST in order to study the behavior of DCCAE, CorrAE, and DistAE. It is also used in [24] to study VCCA and VCCA-Private. The dataset is constructed as follows: first, scale MNIST to \([0, 1]\). Then, for each element of MNIST, rotate the image a random amount by
(a) The generative model for Noisy MNIST. noise
is incompressible (784 dimensions, the same number as the images) additive Gaussian noise.

(b) The generative model for Tangled MNIST. noise
is replaced with rot, a one-dimensional factor of variation.

(c) Samples from Noisy MNIST and Tangled MNIST. Rows 1 and 2 are samples from view x and y of Noisy MNIST, respectively. Rows 3 and 4 are samples from view x and y of Tangled MNIST, respectively. In both datasets, each view is a random sample from the same class and view x is rotated a random angle of rotation sampled uniformly from \((-\frac{\pi}{4}, \frac{\pi}{4})\). They differ in view y. In Noisy MNIST, incompressible independent noise is added to each pixel, which makes quantitative analysis from a disentangling perspective challenging. In Tangled MNIST, we replace the incompressible noise with an independent angle of rotation sampled uniformly from the same set as view x, \((-\frac{\pi}{4}, \frac{\pi}{4})\). This yields a low dimensional underlying factor of variation for view y which we have hope to capture in learned representations.

Fig. 4: The Noisy MNIST and Tangled MNIST datasets.

sampling the angle of rotation uniformly from \((-\frac{\pi}{4}, \frac{\pi}{4})\). Consider this view x. To generate view y, first randomly choose another MNIST digit of the same class. Then, for each pixel of the second image, add independent noise sampled uniformly from [0, 1] and then truncate the resulting values back to [0, 1].

There is one major limitation to using this dataset for multi-view representation learning analysis: because the noise is independent per dimension in view y, it is incompressible and, in models such as VCCA-private and ACCA-private, we have no hope of recovering it in \(z_y\) unless it has enough dimensions to capture the number of independent noise dimensions in \(y\) (784 in MNIST). Because we believe that the PGM introduced by VCCA-Private, which contains view-specific information, sets an important precedent for MVRL research, we think it is important to use a dataset that has some known information for each of \(z\), \(z_x\), and \(z_y\) to contain of lower dimensions to make analysis more tractable. For Noisy MNIST, class information is common between views and should reside in \(z\) and the angle of rotation for view \(x\) is specific to that view, so should reside in \(z_y\). Because of this, we introduce a new dataset that is a minor variation to Noisy MNIST that will allow us to explore some of the disentangling properties of VCCA, VCCA-Private, ACCA, and ACCA-Private that we call Tangled MNIST. It is constructed in the
same manner as Noisy MNIST except that no noise is added to view $y$. Instead, view $y$ is rotated by an independent angle of rotation sampled from $(-\frac{\pi}{4}, \frac{\pi}{4})$. Because it is sampled independently from the angle of rotation for view $x$, this information should reside in $z_y$, giving known information to investigate in each of $z$ (class information), $z_x$ (angle of rotation for view $x$), and $z_y$ (angle of rotation for view $y$).

In Tangled MNIST, there are 5 independent factors of variation, 3 of which are known and 2 of which are theoretical (the style dimensions which are of unknown dimensionality):

1. class (common)
2. $x$ angle of rotation
3. $y$ angle of rotation
4. $x$ style (intra-class coordinates for $x$)
5. $y$ style (intra-class coordinates for $y$)

The generative model for Noisy MNIST is shown in Figure 4a. The underlying factors of variation are organized by whether they are view-common or view-specific. The generative model for Tangled MNIST is likewise shown in Figure 4c. Observe that the only difference between the two figures is that noise$_y$ from Noisy MNIST is replaced with rot$_y$ in Tangled MNIST.

Because of these advantages, all the following experiments were run on Tangled MNIST.

5 Experiments

For the sake of reproducibility, all code for the experiments and figures generated in this section can be found at https://github.com/bcdutton/AdversarialCanonicalCorrelationAnalysis. The goals for this section are to demonstrate the performance of ACCA and ACCA-Private at

5.1 Experiment 1: ACCA and VCCA on Tangled MNIST

We begin the experiments by first comparing the general performance and training behavior of VCCA and ACCA. Tangled MNIST has at least 5 independent factors of variation, as discussed in the previous section. So, we first experiment using $z$-dim=5. We use the same network architectures as [23] except that we do not use dropout: four hidden layers in both the encoder and decoder with 1024 units each followed by ReLU activation functions except on the layer leading to $z$ (where no activation function is used) and the final decoder layer (which uses sigmoid activation functions since the training data is all in $[0, 1]$).

In all the following experiments, all models were trained for 100 epochs.

VCCA and ACCA with $z$-dim=5 were trained on Tangled MNIST over 100 epochs using the validation criteria just described. The training loss curves can be found in Figures 6a and 7a, respectively. From the ACCA loss curves, it seems as though the discriminator and generator converge quickly to random with occasional resufflings. To understand the information content in the representation,
we train three linear SVMs each epoch and predict the three known factors of variation: \texttt{class}, \texttt{rot}_x, and \texttt{rot}_y. In Figures 6b and 7b you can see how the information content for \( z \) changes over the course of training.

In figures 11a and 11b you can see random generations produced by sampling from \( p(z) \) and sending the result through the decoder. In Figures 11c and 11d, 5 random reconstructions are shown from each of the trained models. Columns 1 and 3 from each of those plots are views \( x \) and \( y \) from out-of-sample data from Tangled MNIST and columns 2 and 4 are their respective reconstructions.

Unfortunately, we have little understanding in these dimensions of the differences in quality between how VCCA and ACCA fit \( p(z) \) to \( q(z|x,y) \). To explore this, we drop \( z \)-dim to 2. At this dimension, there are more independent factors of variation to store than there are dimensions so we expect some significant loss in representational power, so we increase the capacity of the decoders by adding two additional layers. The loss and information curves can be found in Figure 8.

In Figure 10a all of the training data is projected into \( z \)-space. Each column has the same embeddings, but is colored by information type. In column 1, VCCA embeddings are colored by \texttt{class}. In column 2, they are colored by \texttt{rot}_x and in column 3, they are colored by \texttt{rot}_y. The same thing is done with ACCA in Figure 10b. In Figure 10c the log densities on the ACCA and VCCA embeddings are estimated using kernel density estimation with a Gaussian kernel of bandwidth 0.2.

The information content in \( z \) can be further explored by iterating over the highest density region of \( p(z) \). We place a grid with step size 0.25 over \((-4, 4) \times (-4, 4) \) and decoding the center of each cell into generations for views \( x \) and \( y \). The view \( x \) and \( y \) results for VCCA and can be found in Figures 12a and 12b, respectively. The results for ACCA for views \( x \) and \( y \) can be found in Figures 12c and 12d, respectively.

We believe it is clear from these experiments that the discriminator of ACCA acts as a stronger regularizer to the autoencoder during training and does a better job of fitting \( q(z|x,y) \) to \( p(z) \). The better fit to \( p(z) \) allows ACCA to produce better generations with fewer low density regions in \( q(z|x,y) \) where the decoder produces gibberish. However, the stronger regularization does sometimes come at the expense of information content, as can be seen from the information curves. VCCA consistently predicts class and angles of rotation better than ACCA.

5.2 Experiment 2: ACCA-Private and VCCA-Private on Tangled MNIST

In this section, we explore the properties of ACCA-Private and VCCA-Private on Tangled MNIST. We start off using \( z \)-dim=\( z_x \)-dim=\( z_y \)-dim=2. Each model is trained for 100 epochs and chosen using the validation criteria of the previous section.

In Figure 13a we visualize the embeddings for \( z \) (row 1), \( z_x \) (row 2), and \( z_y \) (row 3) on the test data of Tangled MNIST. In column 1, the embeddings are colored by class information, in column 2, they are colored by \texttt{rot}_x information, and in column 3, they are colored by \texttt{rot}_y information. As can be seen from the diagram, \( z \) has discriminate class information with very little rotational information present from either view. It is impossible for \( z \) to contain information about view \( y \) since it is a function of view \( x \) only. However, there is very little discernible rotational information for view \( x \), either, indicating that this information was discarded.
While both $z_x$ and $z_y$ include some class information, they are dominated by discernible information about their view’s angle of rotation. They also contain no information about the other view (since they are not functions of that view). Similar phenomenon can be observed for ACCA in Figure 14a: $z$ contains no discernible information about angle of rotation for either view. And, while $z_x$ and $z_y$ contain some class information, they are dominated by information on their own view’s angle of rotation. We can also observe in Figures 13b and 14b that, as discussed last section, VCCA uses a weaker regularizer on the aggregate posteriors $q(z|x)$, $q(z_x|x)$, and $q(z_y|y)$ than ACCA.

Further insight into how information is distributed across the representations for both VCCA-Private and ACCA-Private can be found in Figures 15b and 15d, respectively. Here we track the information content of each of the latent representations during training. Both models store roughly the same amount of information about each view in each representation, with ACCA-Private having roughly 8% better accuracy at predicting class from the view-specific latent variables (32% to 24%), showing the information spill over. We can also see from their loss curves in Figures 15a and 15c that training is more challenging for ACCA and requires more epochs to stabilize, which is can also be seen from the information curves: it takes ACCA-Private roughly 30 epochs for the information content of each variable to stabilize.

Reconstructions for each model are of roughly the same quality, as can be seen in Figure 17a and Figure 17b. Columns 1 and 3 are view $x$ and $y$, respectively, and columns 2 and 4 are their corresponding reconstructions. Interestingly, we rarely observe large errors in angle of rotation - they are almost always stylistic within the class.

When we increase the latent dimensions to 4 (i.e. $z$-dim=$z_x$-dim=$z_y$-dim=4), we observe less chaotic FOV reshuffling during training, as can be seen in Figure 16a and Figure 16b.

5.3 Arbitrary Priors

One of the main advantages that ACCA and ACCA-Private offer over VCCA and VCCA-Private is the ability to match their aggregate posteriors to relatively arbitrary priors. In this experiment, we construct a complicated prior by wrapping a uniform distribution over an S-manifold in 3-d. Samples from the prior can be seen in Figure 5a (note: coloring is chosen here simply to highlight the shape of the manifold).

We train ACCA using $z$-dim=3 on Tangled MNIST with the S-manifold distribution acting as $p(z)$. The best model is chosen using the validation criteria discussed earlier for ACCA in experiment 1. The loss and information curves are shown in Figure 18a and Figure 18b below. All three known factors of variation are present in $z$, with $\text{rot}_y$ the least present.

This can more directly be seen from the embeddings on Tangled MNIST test data. In Figures 5c, 5d, and 5e the embeddings are colored by class, $\text{rot}_x$, and $\text{rot}_y$, respectively.
(a) S-manifold prior. A uniform 2d distribution is wrapped over the S-manifold in 3d, class. Colors shown here are chosen to highlight the shape of the manifold only.

(b) Embeddings from ACCA colored by class.

(c) Embeddings from ACCA colored by view $x$’s angle of rotation.

(d) Embeddings from ACCA colored by view $y$’s angle of rotation.

Fig. 5: $p(z)$ (a) and embeddings (b-d) from ACCA colored by known factors of variation in Tangled MNIST.

Complicated priors can also be used with ACCA-Private, as shown in Figure 19 below, where $z$-dim=3, $z_x$-dim=2, $z_y$-dim=2, $p(z_x) = p(z_y) = \mathcal{N}(0, I)$, and $p(z)$ is the S-manifold distribution above. The embeddings (colored by their corresponding FOV information) are shown.

6 Conclusion

In this work, we show how adversaries can be used in multi-view representation learnings using the PGMs of VCCA and VCCA-Private. These approaches

1. Act as more powerful regularizers matching aggregate posteriors to priors more effectively

2. Allow a broader class of more complicated distributions to be used without complicated KL Divergence derivations
We also explore the multi-level disentangling properties of VCCA-Private and ACCA-Private, where disentangling factors of variation can be understood as occurring at two levels: at the set level, factors of variation common to both classes should, in theory, reside only in $z$ while information particular to view $x$ should reside in $h_x$ and information particular to view $y$ should reside in $h_y$. We observe that, while this is largely the case, there is some significant bleed over with common information spilling over into the view-specific representations. We also design a validation criteria for ACCA and ACCA-Private that works well in practice, bypassing the need for manual investigation during training. Lastly, we derive variational lower bounds for VCCA using $q(z|x,y)$, allowing $z$ to store both view’s view-specific information and not just $x$.

References

1. AKAHO, S.: A kernel method for canonical correlation analysis. In: Proceedings of the International Meeting of the Psychometric Society (IMPS2001). Springer-Verlag (2001)
2. Andrew, G., Arora, R., Bilmes, J., Livescu, K.: Deep canonical correlation analysis. In: International conference on machine learning, pp. 1247–1255 (2013)
3. Bach, F.R., Jordan, M.I.: A probabilistic interpretation of canonical correlation analysis (2005)
4. Bengio, Y., Courville, A., Vincent, P.: Representation learning: A review and new perspectives. IEEE transactions on pattern analysis and machine intelligence 35(8), 1798–1828 (2013)
5. Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., Abbeel, P.: Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In: Advances in neural information processing systems, pp. 2172–2180 (2016)
6. Eastwood, C., Williams, C.K.: A framework for the quantitative evaluation of disentangled representations (2018)
7. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in neural information processing systems, pp. 2672–2680 (2014)
8. Gretton, A., Borgwardt, K., Rasch, M., Schölkopf, B., Smola, A.J.: A kernel method for the two-sample problem. In: Advances in neural information processing systems, pp. 513–520 (2007)
9. Hardoon, D.R., Szedmak, S., Shawe-Taylor, J.: Canonical correlation analysis: An overview with application to learning methods. Neural computation 16(12), 2639–2664 (2004)
10. Hofmann, T., Schölkopf, B., Smola, A.J.: Kernel methods in machine learning. The annals of statistics pp. 1171–1220 (2008)
11. HOTELLING, H.: Relations between two sets of variates. Biometrika 28(3-4), 321–377 (1936)
12. Hsieh, W.W.: Nonlinear canonical correlation analysis by neural networks. Neural Networks 13(10), 1095–1105 (2000)
13. Kim, H., Mnih, A.: Disentangling by factorising. arXiv preprint arXiv:1802.05983 (2018)
14. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
15. Lai, P.L., Fyfe, C.: Canonical correlation analysis using artificial neural networks. In: ESANN, pp. 363–368. Citeseer (1998)
16. Lai, P.L., Fyfe, C.: A neural implementation of canonical correlation analysis. Neural Networks 12(10), 1391–1397 (1999)
17. Lai, P.L., Fyfe, C.: Kernel and nonlinear canonical correlation analysis. International Journal of Neural Systems 10(05), 365–377 (2000)
18. Li, Y., Yang, M., Zhang, Z.M.: A survey of multi-view representation learning. IEEE Transactions on Knowledge and Data Engineering (2018)
19. Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., Frey, B.: Adversarial autoencoders. arXiv preprint arXiv:1511.05644 (2015)
20. van Steenkiste, S., Locatello, F., Schmidhuber, J., Bachem, O.: Are disentangled representations helpful for abstract visual reasoning? arXiv preprint arXiv:1905.12506 (2019)
7 Appendix

7.1 Additional Plots from Section 4.2

Fig. 6: VCCA loss and information curves during training on Tangled MNIST for 100 epochs with z-dim=5. Information content is with respect to each of the three known factors of variation: class, rot_x, and rot_y.
(a) Losses during training

Fig. 7: ACCA loss and information curves during training on Tangled MNIST for 100 epochs with \( z\)-dim=5. Information content is with respect to each of the three known factors of variation: class, \( \text{rot}_x \), and \( \text{rot}_y \).
Fig. 8: VCCA loss and information curves during training on Tangled MNIST for 100 epochs with $z$-dim=2.
(a) Losses during training

(b) Information content in $z$ during training

Fig. 9: ACCA loss and information curves during training on Tangled MNIST for 100 epochs with $z$-dim=2.
Fig. 10: VCCA (a) and ACCA (b) embeddings, colored by information type. Class information is in column 1, view x rotation angle in column 2, and view y angle in column 3. In (c), log probability densities are shown for the prior $p(z) = \mathcal{N}(0, I)$ on the left and estimated using kernel density estimation with a Gaussian kernel with bandwidth = 0.2.
Fig. 11: VCCA and ACCA generations and reconstructions on Tangled MNIST with $z$-dim=5. For the reconstructions, columns 1 and 3 are views $x$ and $y$, respectively. Columns 2 and 4 are reconstructions $\hat{x}$ and $\hat{y}$ of views $x$ and $y$, respectively.

Fig. 12: Here we walk $z$ over grid $(-4,4)/times(-4,4)$ with step size 0.25 and decode the center of each cell into generations for views $x$ and $y$. Subfigures (a) and (b) show resulting generations for views $x$ and $y$, respectively, for VCCA. Subfigures (c) and (d) show resulting generations for views $x$ and $y$, respectively, for ACCA.
7.2 Additional Plots from Section 4.3

(a) caption

(b) VCCA-Private with $z - \text{dim} = z_x - \text{dim} = z_y - \text{dim} = 2$
Fig. 14: ACCA-Private with $z - \text{dim} = z_x - \text{dim} = z_y - \text{dim} = 2$
Fig. 15: Training and prediction curves for different models.
7.3 Additional Plots from Section 4.4

(a) Loss curves during training

![Loss curves during training](image1)

(b) Information curves during training

![Information curves during training](image2)

Fig. 18: Loss (a) and information (b) curves for ACCA with $z$-dim=3, trained on Tangled MNIST for 100 epochs with the S-manifold prior.

Fig. 19: ACCA-Private with S prior for $z$ and Gaussian for $z_x$ and $z_y$