Improving Latent Representations via Explicit Disentanglement

Joey Hejna* jhejna@berkeley.edu
Ashwin Vangipuram* avangipuram@berkeley.edu
Kara Liu* karamarieliu@berkeley.edu

* UC Berkeley, Equal contribution

Abstract

A promising approach for improving both the interpretability and usefulness of latent representations for downstream tasks is disentanglement. Though recent work in variational frameworks has found success implicitly encouraging disentanglement for learning better representations, such methods do not take advantage of the readily available priors in most data. We propose three methods for explicit disentanglement and evaluate their ability to learn better representations with different datasets. Additionally, we investigate how we can leverage explicit disentanglement for learning representations under biased data.

1 Introduction

Recently, generative models have shown great promise for building representations in images. However, the internal representations these 'black-box' models learn often leave much to be desired in terms of interpretability. A promising approach for improving both the interpretability and usefulness of latent representations for downstream tasks is *disentanglement*. As stated by [15], we define *disentanglement* as the representation when altering one latent dimension mainly affects one factor of variation while leaving other factors relatively unaffected. We further separate *explicit disentanglement* from *implicit*. Explicit disentanglement allows for the user to know *a priori* what factor each latent code corresponds to, while the implicit representation can only reveal the factor after learning the latent space. To motivate the need for disentanglement, note that humans are naturally skilled at finding causal relationships between variables and outcomes, and are adept at generalizing these variable-outcome relationships to other situations. Similarly, building models that can explicitly disentangle latents would provide valuable insights into causal analysis and generalization to downstream tasks, among others.

Though recent work in variational learning has found success implicitly encouraging disentanglement [5, 10], such methods do not take advantage of the readily available priors in most data. Additionally, all methods of implicit disentanglement are impractical, in that they require the user to find the correspondence between different components of the latent to factors in the learned representation. While these brute force techniques have functioned in the past, they are likely to fail for larger representations and provide no guarantees of learning the desired factors of variation.

On the other hand, self-supervised methods such as SimCLR [4] show that strong priors can be learned via relationships induced by data augmentation. For example, image cropping, skewing, and rotation allow for sufficient representation learning, which can bridge the gap to fully supervised techniques. Though image manipulation naturally provides information for learning spatial factors, pseudo-labels for other factors of variation can be found in usually unused meta-data (ie. the timestamp of a photo) or through labels provided by classifiers.

For the above reasons, we study learning explicit representations in the self-supervised or semi-supervised setting where information on desired factors in data is readily generated or known. Using available information on these factors, we integrate disentanglement directly into the learning objective. We propse three methods for explicit disentanglement and evaluate their ability to learn better better representations. Additionally, we investigate how we can leverage explicit disentanglement for learning representations under biased data.

2 Related Works

In the field of unsupervised learning, there have been several early works investigating implicit disentanglement of latent representations. Early works [14] sought to capture high-level latent representations without understanding exactly what the latent codes corresponded to. More recently, building off of the Variational Autoencoder (VAE) generative model [11], $\beta-$VAE [7] alters the objective by placing a higher weight ($\beta > 1$) on the KL divergence between the posterior and the prior, which encourages the independence of latent dimensions at the expense of reconstruction quality. To handle this trade-off, [10] proposed FactorVAE, an alteration of $\beta-$VAE that adds a term to the VAE objective which encourages minimizing $KL[q(z)||p(z)]$. This pushes for a factorial prior without losing information about x in z.

InfoGAN [5] takes the Generative Adversarial Network (GAN) generative model [6] one step further by incorporating the mutual information between a specific subset of latents s and the original observation x into the minimax objective. This allows for interpolations across the one latent dimension s_i to correspond to a certain feature found in the dataset. However, in addition to GANs notorious difficulty to train, both InfoGAN and FactorVAE only model implicit disentanglement of the latent code. These models force the user to seek out the factor each latent corresponds to and can be impractical in more complex datasets.

Other recent works have explored explicit spatial disentanglement. Spatial VAE [2] directly incorpoates translation and rotation into the optimization loop, while Affine VAE [3] incorporates affine transformations. However, both methods are limited to factors of variation that can be explicitly derived from manipulating the input image and are thus less general. Work most similar to ours is Cycle VAE [8]; however the authors do not enforce a distribution over specified factors (prohibiting sampling) and only investigate class identity as a factor.

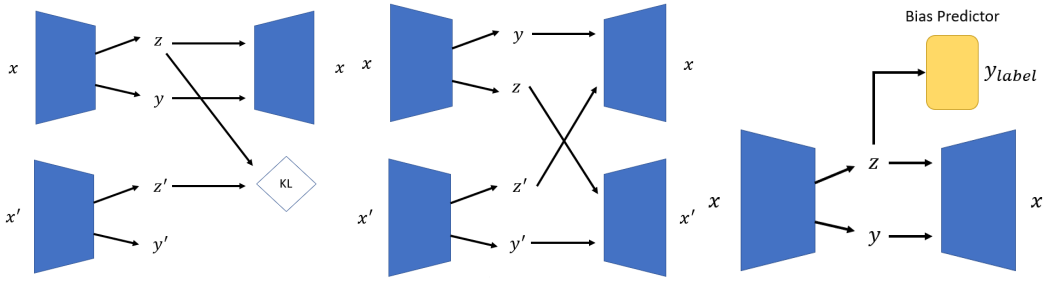
3 Background

3.1 Variational Autoencoders

The variational autoencoder (VAE) learns a mapping from a known low-dimensional distrubtion z (usually $N(0, I)$) to a high-dimensional distribution $x \sim p_{data}$ by maximizing the likelihood of $p_\theta(x|z)$ under p_{data}. On its own, $p_\theta(x|z)$ fails to learn due to the intractable size of the data space X. Thus, VAEs use importance sampling with an approximate posterior $q_\phi(z|x)$. To increase expressivity of the model, both the encoder $q_\phi(z|x)$ and decoder $p_\theta(x|z)$ are parameterized by neural networks. We then optimize the objective using the variational lower bound, leading to the optimization problem: $\min_{\theta,\phi} -E_{z\sim q_\phi(z|x)}[\log p_\theta(x|z)] + KL[q_\phi(z|x)||P(z)]$, which we can solve by using the reparamaterization trick [11]. For more detail on VAEs, we refer the reader to [11].

3.2 Factored Latents

We now consider the explicitly factorized setting for disentanglement. Rather than modeling x as a function of latents z, we postulate that each data point x is derived from both a latent z and explicit factors y coming from some factor distribution. Note that this representation extends to multiple an arbitrary number of factors $y_1, y_2, ..., y_k$. Our decoder now models $p_\theta(x|z, y)$ and our encoder models $q_\phi(z, y|x)$. Mathematically, we model the disentanglement of our representation by assuming the independence of z and y conditioned on x, $P(Z, Y|X) = P(Z|X)P(Y|X)$. We can thus interpret each example from data distribution $x \sim p_{data}$ as being derived from features we care about z and additional factors y. Our methods seek to maximize this independence during learning. We assume that there exist strong enough priors in p_{data} to select reasonable disentanglement factors y.



(a) Similarity Loss (b) Forward Cycle Loss (c) Factor Prediction

Figure 1: Graphical depictions of our methods for explicit disentanglement.

4 Method

We now propose three techniques for encouraging the independence of z and y in the factored setting: a similarity loss, a forward cycle loss, and a predictor loss. We then consider predictor loss in the scenario of learning under a biased factor distribution y.

4.1 Similarity Loss

The independence condition in section 3.2 is equivalent to $P(Z|X,Y) = P(Z|X)$. We can interpret this to mean that regardless of the factor variable y, our latent z should be be the same. We can rewrite this as $P(Z|X,Y=y) = P(Z|X,Y=y')$, where y' is considered to be a different variant of the original factor y. To construct y', we produce x' from x by performing an augmentation on x that, using prior knowledge on the data distribution, we know will not change the underlying representation. We can enforce this independence between the latents and factors during training by encouraging $q_\phi(z|x) = q_\phi(z|x')$ to hold. We do this by adding a KL regularization term $KL[q_\phi(z|x)||q_\phi(z|x')]$ to the original VAE formulation. The new learning objective is given below, where λ is a hyper-parameter determining the intensity of the regularization:

$$\min_{\theta,\phi} -E_{z,y\sim q_\phi}[\log p_\theta(x|z,y)] + KL[q_\phi(z,y|x)||P(z,y)] + \lambda KL[q_\phi(z|x)||q_\phi(z|x')] \tag{1}$$

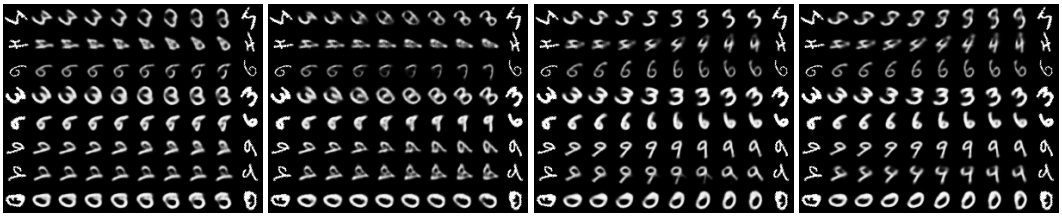
4.2 Forward Cycle Loss

As in similarity loss we use augmented data points to promote the conditional independence of z and y. Rather than directly constraining the latents, z, we constrain the reconstruction by using z' and y to reconstruct x and use z and y' to reconstruct x'. This way, the VAE learns that z and z' must remain invariant for images x and x' that differ in only factors y and y' and not in other aspects. Probabilisticaly, this can be viewed as promoting y to have no effect on z. The new objective is given below:

$$\begin{aligned}\min_{\theta,\phi} &- E_{z',y}[\log p_\theta(x|z',y)] + KL[q_\phi(z,y|x)||P(z,y)] \\ &- E_{z,y'}[\log p_\theta(x'|z,y')] + KL[q_\phi(z',y'|x')||P(z',y')]\end{aligned} \tag{2}$$

4.3 Factor Prediction

We now consider scenarios where it is impossible to generated paired data points x and x', but it is still possible to predict factors y from the data points. Maximizing the independence between Z and Y can be viewed as the information theoretic objective $\min I(Z;Y|X)$ since conditioned on x, z should provide no information about y. By minimizing the mutual information, the conditional entropy $H(Y|Z,X)$ approaches that of $H(Y|X)$. Thus, we can view disentanglement in a predictive framework where we want z to provide no useful information for predicting the factors y. We thus learn a classifier $g(z)$ that attempts to predict y from the latent representation z. By learning an encoding $q_\phi(z,y|x)$ such that $g(z)$ is unable to provide useful predictions of y, we incentivize z to be independent of y. In practice, we do this via mini-max optimization. First, we quantize continuous factors into a fixed number of discrete labels. While the classifier $g(z)$ attempts to predict the correct



(a) β-VAE (β = 2) (b) Factor Prediction (c) Sim. Loss ($\lambda = 0.6$) (d) Forward Cycle Loss

Figure 2: Visual disentanglement results on the Rotated MNIST dataset. Originals on the left and right of each figure are rotated by -50 and 50 degrees respectively. Interpolations are done only the single most varied latent between images, corresponding to the rotation factor latent in our methods.

factor y, the encoder $q_\phi(z, y|x)$ tries to maximize the entropy of the discrete distribution over factors output by $g(z)$.

$$\min_{\theta,\phi} \max_{w} -E_{z,y\sim q_\phi}[\log p_\theta(x|z,y)] + KL[q_\phi(z,y|x)||P(z,y)] - E_{z\sim q_\phi}[\mathcal{L}_{CE}(g_w(z), y_{\text{gt}})] \quad (3)$$

In the above objective, $\mathcal{L}_{CE}$ denotes categorical cross entropy loss and y_{gt} are the ground truth labels for y. One example where this approach is particularly useful is when attempting to learn complete representations from biased samples. A dataset is biased in a factor y when the factor y is highly correlated with other latent attributes in the data. However, for downstream tasks we may want our learned representation to generalize to these unseen combinations of factors. The predictor $g(z)$ can then be interpreted as predicting the bias of data, and our optimization attempts to remove bias from the latent component of the data z.

5 Experiment Results

5.1 Rotated MNIST

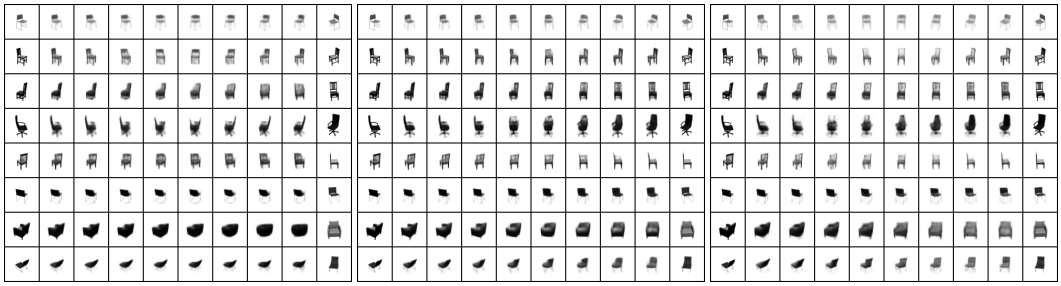
We take images from the MNIST dataset [12] and randomly rotate digits uniformly from negative 60 degrees to positive 60 degrees. We seek to disentangle the latent factor of rotation from the rest of the representation. To generate paired data points for similarity loss and forward cycle loss, we pair the same digit with different rotations together. For factor prediction, our self-supervised labels are binary, indicating if the digit was rotated to the left or to the right. Across all methods, our encoder and decoder both consist of two fully connected hidden layers of size 512 with $tanh$ activations. We use a total latent size of 11, with 10 dimensions for regular latents z and one dimension for the rotation factor y. Visual results for each of our methods and regular VAE baseline can be found in Figure 2, where we only vary a single latent variable while interpolating across a rotated digit. We observe that β-VAE is completely unable to disentangle the rotation factor, and ends up barely changing the digit. While the factor prediction improves upon this somewhat by general rotating the contents of the image, it fails to completely disassociate class from rotation. The extra supervision provided by explicit pairing helps similarity loss and forward cycle loss perform extremely well. In addition, we quantitatively analyze our results by training linear classifiers on top of the latent vectors z in order to predict the class of the digit. As seen in Table 1, a more disentangled and meaningful latent representation yields a higher classification accuracy.

5.2 3D Chairs

The 3D chairs dataset [1] consists of renderings of 1393 unique chairs from varying angles and perspectives. We employ the same network architecture and hyper-parameters from FactorVAE [10], with a few minor changes. Namely, we use a 29 dimensional latent z and three additional dimensions for the factor y designed to represent zoom, azimuth, and altitude in combination. Pairs were generated by sampling two images of the same chair. Visual results against $\beta-$VAE can be found in Figure 3, and additional results against Factor VAE are in the Appendix. We qualitatively find that all the baselines are unable to disentangle chair identity from viewpoint effectively.

Classifcation Acc on:	Rotated MNIST	Colored MNIST
Full Encoder	92%	43.28%
VAE	38%	48.68%
$\beta-$VAE ($\beta = 2$)	40%	57.20%
Predictor VAE	58%	59.29%
VAE + Similarity Loss	82%	-
VAE + Forward Cycle Loss	84%	-

Table 1: Classification Accuracy. For all methods, we freeze the encoder of a pretrained model and train a linear layer/s to learn the correct class over 3 epochs. Full Encoder means the model was trained from scratch (for 10 epochs) on just classification and allowed to adjust all parameters.



(a) β-VAE ($\beta = 2$) (b) Similarity Loss ($\lambda = 0.5$) (c) Forward Cycle Loss

Figure 3: Visual disentanglement results for the 3D chairs dataset. In each figure the image on the left gives the original, while the image on the right gives a different view of the same chair. We interpolate only over the three latent dimensions that vary the most, which correspond to the predefined factor dimensions in b) and c).

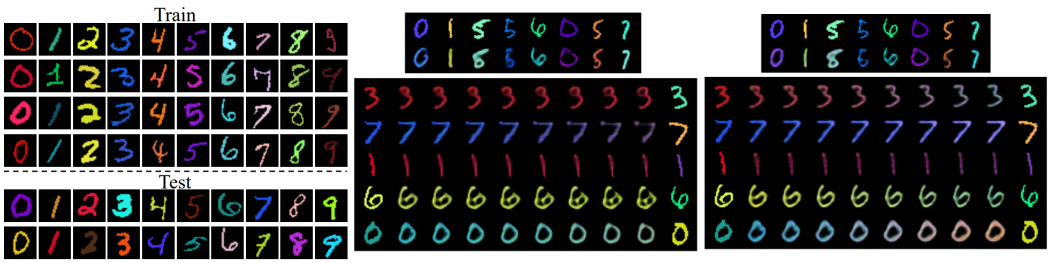
5.3 Colored MNIST

We consider a colorized version of the MNIST dataset [9] that injects bias. Rather than all digits being uniformly being colored, each digit in the training set is assigned a unique mean color. During training, digits are colorized by taking their class's mean color and adding a small Gaussian perturbation. As a result, while a red six is in the training data, no blue six is the training dataset. We then seek to disentangle the color factor y from the representation even though the dataset is extremely biased, allowing us to learn a representation that performs well on the unbiased test set where colors are assigned uniformly across digits.

We employ an encoder architecture with three convolutional layers of size 32 with 3x3 kernels and stride 2 followed by a fully-connected layer of size 128. The decoder is the reverse with a Tanh at the end. We chose a latent dimension of size 12 with a factor dimension of three, one for each color channel. We derive predictive labels for color by quantizing each color channel into four values. We find that factor prediction is better able to separate color information from class than regular VAE as seen in Figure 4, though reconstruction quality is similar. Quantitatively, we boost classification performance on the test set from 43.28% to 59.29% as seen in Table 1. However, we additionally find that β-VAE performs well on this task and include these results in the Appendix.

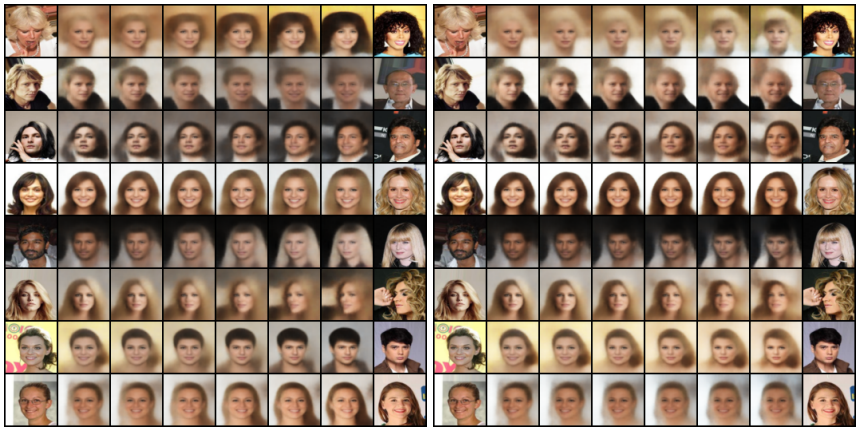
5.4 CelebA

We use a modification of the similarity regularization in 4.1 for the CelebA dataset [13]. We want the latents z to exactly represent the 40 different face attributes ("glasses", "bangs", "mustache" etc.) and have the y factors correspond to the remaining noise in the faces. Instead of using augmentations, we take two images from the dataset x, x' and find the common attributes between them c, represented as a 40 dimensional binary array. The KL regularization for the similarity loss is now only over the subset of features that both images share, $\lambda KL[q_\phi(z \odot c|x')||q_\phi(z' \odot c|x')]$. Our z is 40-dimensional, y is 10-dimensional, and we use $\lambda = 5$. The encoder and decoder architectures are the same as what we used in Colored MNIST.



(a) Dataset Visualization (b) VAE (c) Factor Prediction

Figure 4: Graphical results on the colored MNIST dataset. Subfigure a) depicts how the dataset used for training is color biased. In subfigures b) and c) the top portion give reconstructions on images from the test set while the bottom section gives interpolations over images in the test set.



(a) Regular linear interpolations (b) Interpolations over only feature latents

Figure 5: Comparison of interpolations for CelebA using the modified similarity loss

We provide standard linear interpolations between images, as well as a specific interpolation that sets the factors $y = 0$ and thus only interpolate over the features z. As seen in Figure 5, the linear interpolations blend everything about the images together, whereas the feature interpolation keeps the style of the leftmost image and tries to adopt some of the major features of the right image, like hair or facial structure. We hope to extend this and construct a model to explicitly sample whichever features are wanted.

6 Conclusions and Future Work

In this work, we proposed three methods–similarity loss, forward cycle loss, and factor prediction–that explicitly disentangle the latent space. We also study factor prediction in the context of biased data. Qualitatively, our three approaches led to smooth interpolations over a latent dimension/s that corresponded to changing a pre-specified visual feature. Quantitatively, we showed our approaches lead to better representation learning through increased performance on downstream classification tasks.

The primary limitation of limitations of our approach(s) is the required specification of factors before training via some form of weak supervision (pairing, augmentation, etc.). When incorporating the latent meaning with downstream tasks like RL, it is unclear how to augment the data such that a higher-order or possibly unknown concept can be learned.

In the future, we think it would be interesting to investigate the usage of explicit disentanglement on more challenging datasets, such as ImageNet. We also wish to investigate how to combine implicit and explicit techniques for disentanglement.

References

[1] Mathieu Aubry, Daniel Maturana, Alexei Efros, Bryan Russell, and Josef Sivic. Seeing 3d chairs: exemplar part-based 2d-3d alignment using a large dataset of cad models. In *CVPR*, 2014.

[2] Tristan Bepler, Ellen D. Zhong, Kotaro Kelley, Edward Brignole, and Bonnie Berger. Explicitly disentangling image content from translation and rotation with spatial-vae, 2019.

[3] Rene Bidart and Alexander Wong. Affine variational autoencoders: An efficient approach for improving generalization and robustness to distribution shift, 2019.

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations, 2020.

[5] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *NIPS*, 2016.

[6] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[7] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework, 2016.

[8] Ananya Harsh Jha, Saket Anand, Maneesh Singh, and V. S. R. Veeravasarapu. Disentangling factors of variation with cycle-consistent variational auto-encoders, 2018.

[9] Byungju Kim, Hyunwoo Kim, Kyungsu Kim, Sungjin Kim, and Junmo Kim. Learning not to learn: Training deep neural networks with biased data, June 2019.

[10] Hyunjik Kim and Andriy Mnih. Disentangling by factorising, 2018.

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

[12] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

[13] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[14] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *ICML*, 2008.

[15] Pascal Vincent Yoshua Bengio, A Courville. Representation learning: A review and new perspectives, 2013. 35 (8):1798–1828.

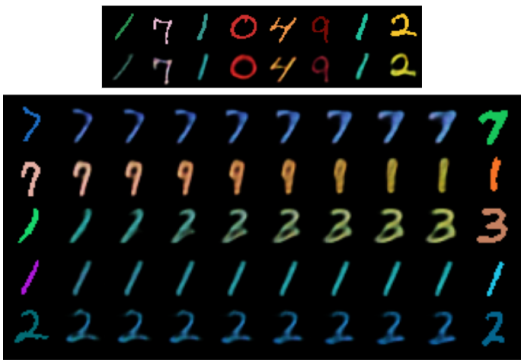
7 Appendix

7.1 Code release

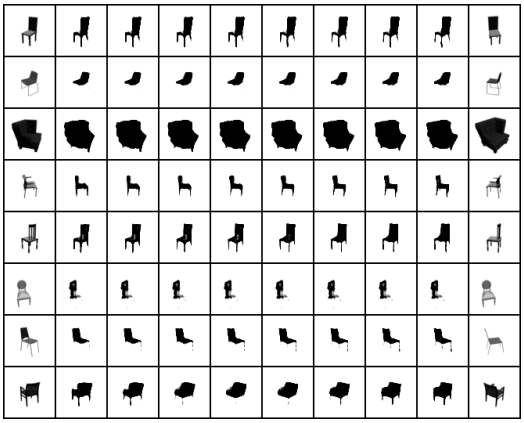
You can find all of the code used in our implementations at `https://github.com/jhejna/ul_gen`.

7.2 Additional Results

This section contains some additional results and larger versions of images already in the main body of the paper.

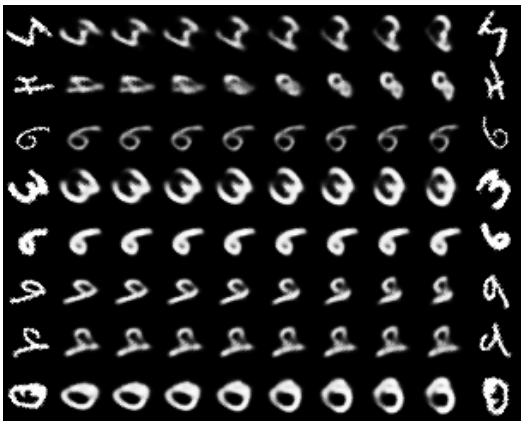


(a) Beta VAE interpolations and reconstructions on the colored MNIST test dataset.

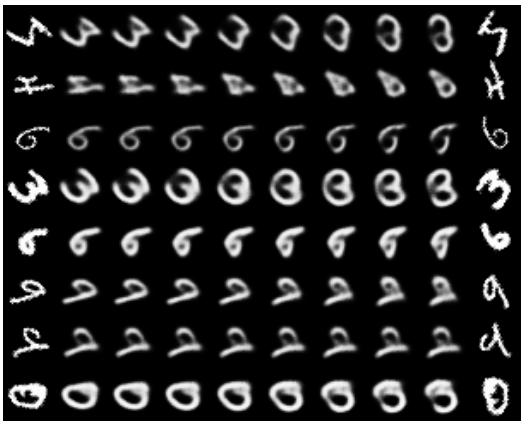


(b) FactorVAE interpolations on the 3d chairs dataset after 150k steps. Model and parameters followed [10].

Figure 6: Additional Baselines

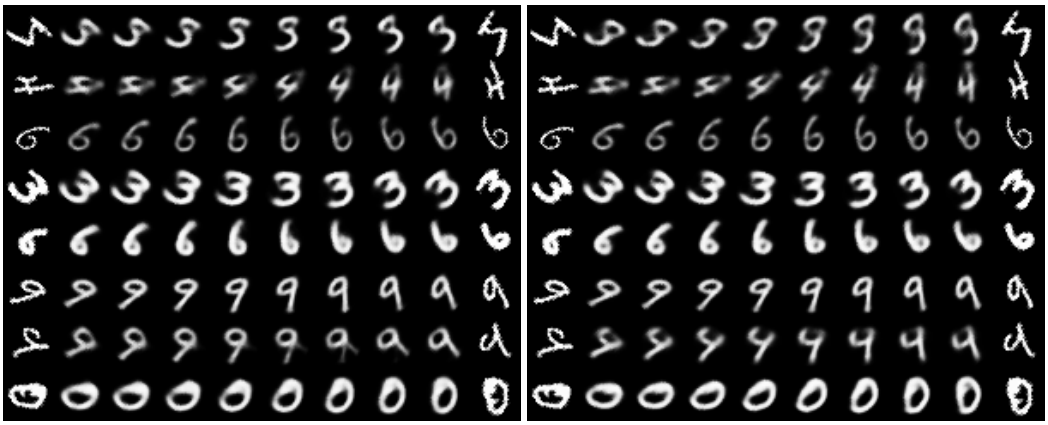


(a) Regular VAE



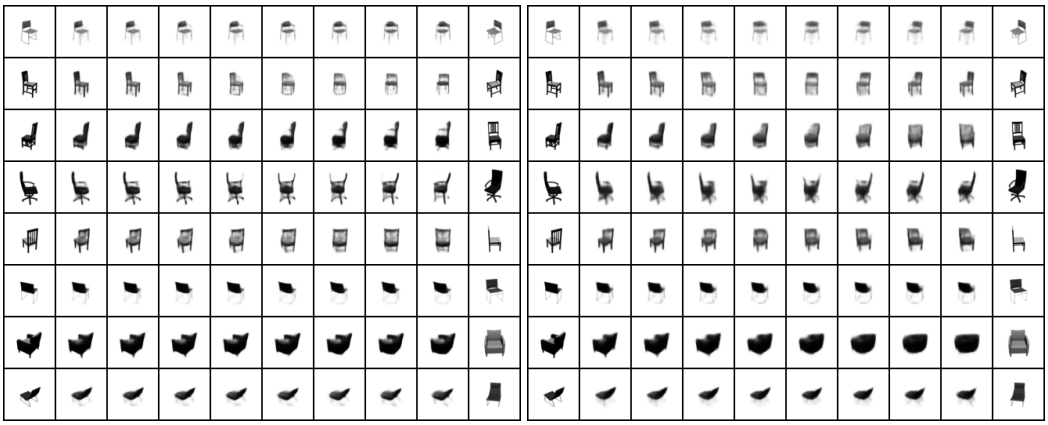
(b) β-VAE ($\beta = 2$)

Figure 7: Larger versions of baselines for the Rotated MNIST dataset



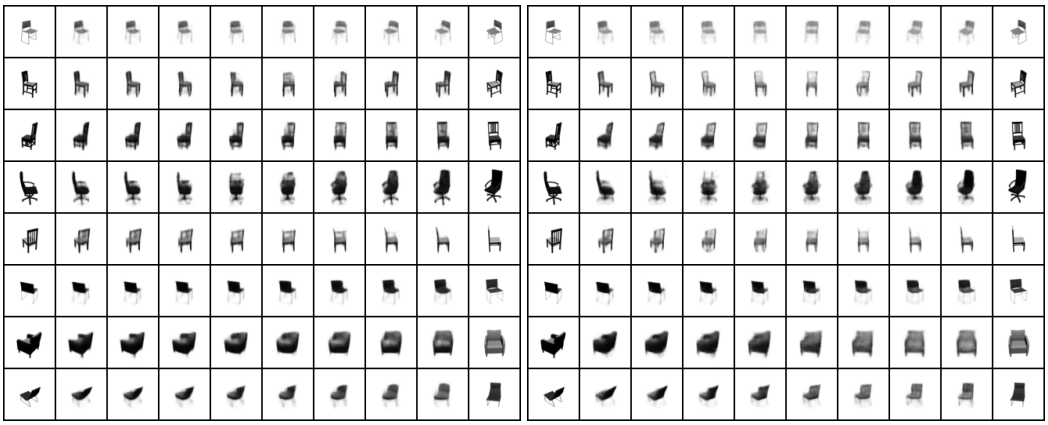
(a) Similarity Loss (b) Forward Cycle Loss

Figure 8: Larger versions explicit factorization models on the Rotated MNIST



(a) Regular VAE (b) β-VAE (β = 2)

Figure 9: Larger versions of baselines for the 3D Chairs dataset



(a) Similarity Loss (b) Forward Cycle Loss

Figure 10: Larger versions explicit factorization models on the 3D chairs dataset.