PROTOVAE: USING PROTOTYPICAL NETWORKS FOR UNSUPERVISED DISENTANGLEMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Generative modeling and self-supervised learning have in recent years made great strides towards learning from data in a completely *unsupervised* way. There is still however an open area of investigation into guiding a neural network to encode the data into representations that are interpretable or explainable. The problem of unsupervised *disentanglement* is of particular importance as it offers to encode information about the different factors of variation or semantic concepts into structurally disjoint latent spaces. Recent work has argued that this task is possible given correctly chosen inductive biases. To this end, we introduce a novel deep generative model, ProtoVAE, that leverages a deep metric learning Prototypical network trained using self-supervision to constrain the actions of a latent space to follow local isometry as well as isolate the changes to be distinct, consistent and noticeable. Our model is unsupervised and requires no *a priori* knowledge of the dataset, including the number of factors. We evaluate our proposed model on the benchmark dSprites, 3DShapes, and MPI3D disentanglement datasets, showing state of the art results against similar methods via qualitative traversals in the latent space, as well as quantitative disentanglement metrics. We further qualitatively demonstrate the effectiveness of our model on the real-world CelebA dataset.

1 INTRODUCTION

One theory of the success of deep learning models for supervised learning revolves around their ability to learn mappings from the input space to a lower dimensional abstract representation space which are best predictive of the corresponding labels (Tishby & Zaslavsky, 2015). However, additional constraints are necessary for these representations to be robust to noise and adversarial examples, transferable to different domain settings and distribution shifts and interpretable for explainable predictions. As a promising solution to this, the networks can be constrained to encode each distinctive property of the data distribution into disjoint subspaces of the latent representation and then use these representations for downstream tasks. This process of encoding the distinctive properties, or *factors of variations*, into disjoint subspaces is known as *disentanglement*.

From a generative perspective, higher dimensional data is assumed to be a non-linear mapping of these factors of variation, where each factor assumes different values to generate specific examples in the higher dimensional data manifold. The generative definition of disentanglement, first proposed in Bengio (2013), proposes to discover these different factors from the data, encoding each factor in a separate subspace or dimension of the learned latent representation. Locatello et al. (2019) intuitively characterizes representations which encode the factors, as disentangled, if a change in a factor produces a change only in the corresponding dimensions of the representation that encode the factor. Alternatively, a change in one dimension of the representation changes only the corresponding factor of variation in the data while being invariant to changes in other factors. These disentangled representations are not only interpretable and give valuable insights into the data distribution but are also more robust for various downstream tasks (Bengio, 2013; Schoelkopf et al., 2012) which might depend only on a subset of factors (Suter et al., 2019).

The problem of *unsupervised* disentanglement is particularly difficult as it proposes to discover the factors of variations without knowing the kind of the factors of variation nor even the *number* of factors. This makes the task much more challenging, and necessitates inductive biases in the model

design or the data for a one to one mapping between the representations and the factors of variations (Locatello et al., 2019).

It has also been argued recently in Zhang et al. (2021); Wilson & Izmailov (2020) that a major fundamental property of deep neural networks is the *inductive bias*, or the structural knowledge we impart into the network before training. Networks with inductive biases that more closely reflect the data are better able to capture and learn the true underlying distribution.

While it has been claimed, in recent work, that unsupervised disentanglement is possible, if suitable inductive biases are introduced into the network and learning process (Higgins et al., 2018; Locatello et al., 2019), so far finding a good bias is an open question. Here, we posit the most critical of these is that the model can be enforced to map changes in the latent space to changes in the data space that resemble changes in the factors of variations. For truly disentangled representations, the changes in a disentangled factor of representation should be *unique* and *consistent* with the changes in the representation space. In addition, this mapping from latents to data space should be *locally isometric:* The recent work of Horan et al. (2021) demonstrated empirically that the concept of local isometry was a good inductive bias for unsupervised disentanglement, and it can aid generative models in discovering a "natural" decomposition of data into factors of variation. We take this observation as a starting point, and propose building a novel deep generative neural network model for unsupervised disentanglement that is trained end-to-end with this constraint in mind. If we want to learn these factors in an unsupervised manner the model must have some component that can generate these kinds of changes in the data, as well as a component that is able to discern the changes.

To address both of these constraints, the local isometry and the unique and consistent changes, we develop a novel deep generative model, ProtoVAE, consisting of a Prototypical Network and Variational Autoencoder (VAE), which affords both the mode-covering properties of a VAE and the interpretability of GANs. Prototypical Networks Snell et al. (2017) learn a metric space where examples of the same class cluster near the class prototypes and the prototypes of different classes are embedded into separate regions in metric space. To learn these representations in an unsupervised way, we train the prototypical network using self-supervision as described below to constrain the VAE to encode the information corresponding to different factors in separate subspaces. This is achieved via transforming the encoded latent space of the data in different ways and subsequently decoded, which are paired with the original data and passed into the prototypical network. The prototypical network constrains this mapping from the representation space to the data space to be consistent with the desired properties discussed above. These transformations in the latent representation space are done via the approach of *interventions* (Sec 2.3). We call the changes in the data space induced by interventions in the latent space *actions*. The prototypical network's main role is to detect and isolate unique changes in the discovered generative factors resulting from intervention. This process allows us to perform disentanglement without knowledge of the underlying data distribution.

Our core contributions are such:

- Design a self-supervised data generation mechanism using a VAE that creates new samples via a process of intervention to train a metric learning prototypical network.
- Design and implement a novel model, ProtoVAE, which combines a VAE and prototypical network to perform disentanglement without any prior knowledge of the underlying data.
- Empirically evaluate ProtoVAE on standard benchmark DSprites, 3DShapes, MPI3D, and CelebA datasets, showing state of the art results.

2 PROTOVAE

In the following section, we describe in detail our proposed model, ProtoVAE, the component subnetworks. Variational Autoencoders and Prototypical Networks, and the concept of *interventions* and the self-supervised dataset mechanism.

2.1 MODEL OVERVIEW

Our proposed method first encodes the data into lower dimensional informative representations using a VAE Kingma & Welling (2014); Rezende et al. (2014). The information bottleneck has been shown to be effective for extracting informative representations needed to reconstruct the data faithfully. We impose structural constraints on the latent space through a Prototypical Network which is trained in conjunction with the VAE. Concretely, we intervene on the individual subspaces of the latent representation such that the actions of these interventions are clustered in the prototypical metric space, with each cluster identifiable with one independent subspace. The data from the interventions is created via a self-supervised process described in Appendix Algorithm 1. Due to the way the self-supervision dataset is created, the prototypical network enforces each independent subspace of the representation to consistently change only one property in the data space. Further, it enforces each subspace changes a different property in the data space. These constraints on the actions of the latent space ensure that each subspace transforms the data to make a distinct and noticeable change corresponding to the factors of variation. We further augment the prototypical network with a separate output head that enforces local isometry, discussed below.

For the actions of the interventions on the individual subspaces to correspond to the factors of variations, we constrain the corresponding generated data space to lie in the true data distribution. This constraint can be effectively enforced in the latent representation by minimizing the distance between the distribution of the latent representations of the VAE and the intervened representations. We do so by training a Generative Adversarial Network (GAN) Goodfellow et al. (2014), in the latent space of the VAE which effectively minimizes the total correlation Kim & Mnih (2019) between the independent subspaces of the representation.

2.2 VARIATIONAL AUTOENCODER AND INTERVENTIONS

The first component of our model is a VAE which consists of an encoder $q_\phi : \mathbb{R}^D \rightarrow \mathbb{R}^d$ that maps the input data x to a lower dimensional representation $\mathbf{z}$ and a decoder $p_\theta : \mathbb{R}^d \rightarrow \mathbb{R}^D$ which reconstructs the data $\hat{x}$ from the representations $\mathbf{z}$. The encoder and the decoder are trained together to optimize the evidence lower bound (ELBO) of the data log-likelihood as in equation 13.

$$\max_{\theta,\phi} \mathcal{L}_V(\theta,\phi) = \mathbb{E}_{q(x)}[\mathbb{E}_{q_\phi(\mathbf{z}|x)}[\log p_\theta(x|\mathbf{z})] - \beta KL(q_\phi(\mathbf{z}|x)||p(\mathbf{z}))] \tag{1}$$

Here $q(x)$ is the empirical data distribution. Optimizing the first term of equation 13 ensures that the latent representation encodes all the information needed to faithfully reconstruct the data, from the representation alone, and hence encodes the different factors of variations. The KL divergence term creates an information bottleneck which enforces optimal, compact encoding of the data by enforcing the posterior distribution to be similar to the independent, non-informative prior distribution. For more details of the workings of the VAE please refer to section A.1.

2.3 SELF-SUPERVISED DATA GENERATION FOR PROTOTYPICAL NETWORKS

To enforce each dimension of the learned representation space z to encode information about a single interpretable factor of variation, different from the factors encoded by the other dimensions, we impose constraints on the actions of intervening the latent representation using the Prototypical Network. To train the prototypical network we generate self-supervised support and query sets, by *intervening* on the values of the latent space and generating data from the intervened representations. The sets consist of a pairs of reconstructed data from the original representations, concatenated with the generated data from the intervened representations as input with the target label being the index of the dimension of the representation that was intervened on. See Appendix Section B.1, Algorithm 1 for the full self-supervised data creation algorithm in detail. The algorithm takes as input a batch of data examples and outputs a set of pairs of images corresponding to reconstructions of changes in the latent space. The the support and the query sets are used to train the prototypical network as described below while the isometric constraint labels are used to train the additional head.

The comparison of the *pair* of outputs allows the network to focus on the *change* or *difference* that was brought about by the intervened dimension, and makes the central focus of the prototypical losses this change. By focusing on this difference, the properties of the data that were changed should be easily discerned from the ones that were invariant to the intervention *irrespective of the absolute value of the properties*.

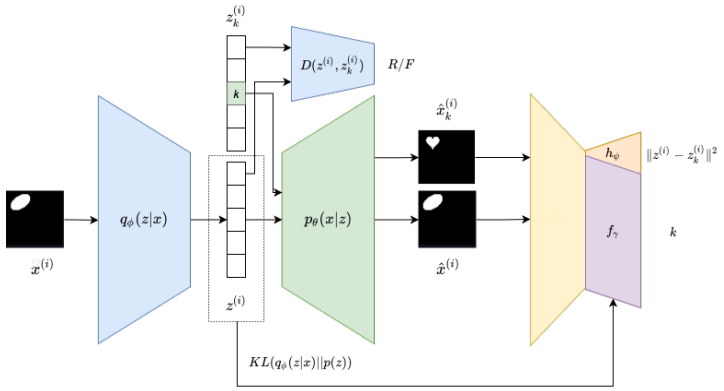


Figure 1: Architecture of our model consisting of a VAE, a latent space discriminator and a Prototypical network. The latent encoding $\mathbf{z}$ from the encoder of the VAE, is changed at a particular dimension k to get the intervened latent variable $\hat{z}_k$. A discriminator is trained to distinguish between z and $\hat{z}_k$ and the encoder is constrained to fool the discriminator. $\mathbf{z}$ and $\hat{\mathbf{z}}_k$ are passed to the decoder to produce the reconstructed image $\hat{x}$ and the generated image $\hat{x}_k$. The original and the generated image are concatenated to form the pair $(\hat{x}, \hat{x}_k)$, which is then passed to the prototypical network. This network is trained to predict both the changed dimension k and the magnitude of the change $\|\mathbf{z} - \hat{\mathbf{z}}_k\|^2$.

2.4 Prototypical Network

Our prototypical network (Snell et al., 2017) component looks at the *pair* of input data generated by Algorithm 1, and tries to discern the dimension of the latent representation, or the factor encoded, that was changed. To effectively predict the changed factor, the change between the pair of data must be easily identifiable with the corresponding dimension of z. Our proposed prototypical network takes in elements of S, which consists of pairs of original and self-generated data from Alg. 1 as input, and computes an m-dimensional representation through the embedding function $f_\gamma : \mathbb{R}^D \times \mathbb{R}^D \rightarrow \mathbb{R}^m$. Note this space is different from the embedding space $\mathbf{z}$ of the VAE. In this m-dimensional space the prototypical network clusters the support set into d clusters, with one cluster for each intervened dimension, and computes a prototype representation for the cluster as in equation 14:

$$c_k = \frac{1}{|S_k|} \sum_{s_k^{(i)} \in S_k} f_\gamma(s_k^{(i)}) \tag{2}$$

For each dimension of the latent representation to consistently encode a single factor of variation *across the entire dataset* and change the corresponding property in the data space upon intervention, we introduce the *consistency loss*. The prototypical network is trained to group the support set into clusters, with each cluster identifiable with a dimension of the latent representation by construction of the support set. The embedding of each query example in the prototypical space must be closer to the prototype embedding calculated by actions of the same intervened dimension. While the support points are used to calculate the prototypes, the loss is computed by calculating the distance of the query examples Q embedding to the prototypes of the target class. (See Appendix Sec. A.2 for more detail).

This consistency loss, calculated for each example in the query set is calculated as in equation 3:

$$\min_{\gamma,\phi,\theta} \mathcal{L}_C(\gamma, \phi, \theta) = -\frac{1}{|Q|} \sum_{q_k^{(i)} \in Q} \log p_\gamma(t = k|q_k^{(i)}) \cdot KL(q_\phi(z_k|x)||p(z)) \tag{3}$$

where $p_\gamma(t = k|q_k^{(i)})$ is calculated as follows according to the Euclidean distance metric d:

$$p_\gamma(t = k|x_q) = \frac{\exp\left(-d(f_\gamma(q_k^{(i)}), c_k)\right)}{\sum_{k'} \exp\left(-d(f_\gamma(q_k^{(i)}), c_{k'})\right)} \tag{4}$$

The loss for every query example is multiplied by the KL-divergence of the target dimension, averaged for the batch of examples, such that the loss for each dimension is scaled by the amount of information the dimension encodes. For the dimensions that do not encode any information, and the actions of which after interventions do not produce any changes in the data, their corresponding losses are scaled by zero.

In addition to consistency, we want each dimension to encode a factor different from the other factors encoded by the other dimensions. Thus for a particular example, the action of each dimension must produce a different change such that the change is identifiable with the intervened dimension. For each example's representation when intervened on by different dimensions of the representation must produce different changes, such that a query example when acted upon by one of the dimensions can easily identify the dimension that was changed. This is also important practically, as the network is forced to make similar changes to an example when the same dimension is intervened on by different amounts. To this end, we add the *uniqueness* loss (equation 5), where the prototypes are actions of the different dimensions on a particular example and the query example is the action of a randomly chosen dimension on that example.

$$\min_{\gamma,\phi,\theta} \mathcal{L}_U(\gamma, \phi, \theta) = \frac{1}{|Q|} \sum_{q_k^{(i)} \in Q} -\log q_\gamma(t = k|q_k^{(i)}) \cdot KL(q_\phi(z_k|x)||p(z)) \tag{5}$$

With q_γ calculated as follows:

$$q_\gamma(t = k|q_k^{(i)}) = \frac{\exp\left(-d(f_\gamma(q_k^{(i)}), f_\gamma(s_k^{(i)}))\right)}{\sum_{k'} \exp\left(-d(f_\gamma(q_k^{(i)}), f_\gamma(s_{k'}^{(i)}))\right)} \tag{6}$$

The consistency loss and the uniqueness loss are added together to get a combined prototypical loss:

$$L_P(\gamma, \phi, \theta) = L_C + L_U \tag{7}$$

As an addition inductive bias, as proposed in Horan et al. (2021), we constrain the actions to be locally isometric (Donoho & Grimes, 2003) to the interventions in the latent space. We add a separate output head which learns by how much the latent z changed, constraining these actions to be isometric with the interventions. This serves as an imperative inductive bias for unsupervised disentanglement.

The additional head $h_\psi : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}^d$, when given a pair of data, is trained to predict the difference in the values for the all the dimensions of latent variables through the loss function in equation 8:

$$\min_{\psi,\theta,\phi} \mathcal{L}_I(\psi, \theta, \phi) = \|h_\psi((\hat{x}^{(i)}, \hat{x}_k^{(i)})) - (z^{(i)} - \hat{z}_k^{(i)})\|^2 \tag{8}$$

In implementation, f_γ and h_ψ share all hidden convolutional layers and differ only in final the fully connected layer.

2.5 Latent Space GAN

For the decoder to map the intervened latent representations to the true data distribution, we enforce the original latent distribution and the latent distribution after interventions to be close to each other. This constraint is necessary to ensure that the changes made to the data after interventions is natural and correspond to the factors of variations.

To address this in our framework, we train a separate discriminator network in the latent space to distinguish the intervened latent representations from the original representations. This discriminator is similar to the one proposed in Kim & Mnih (2019) to minimize the total correlation term $\mathrm{KL}(q(z)||\prod_i q(z_i))$:

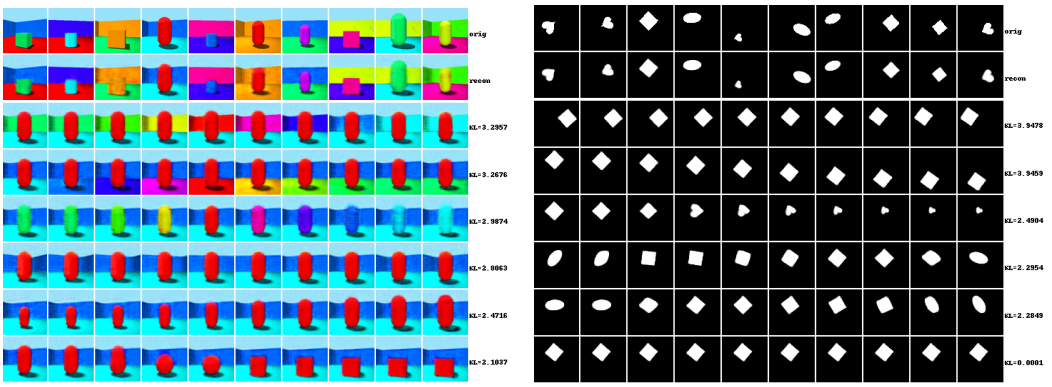


Figure 2: A comparison of latent traversals in latent space for the 3DShapes and Dsprites dataset. *Left:* 3DShapes, *Right:* Dsprites. Proto-VAE produces smooth, disentangled latent representations. Row 1 and 2 are some sample original images, and their reconstructions generated by our model, respectively. Rows 3 downward are the traversals for each latent element, as detailed below. For 3DShapes, we actually see a near-perfect traversal across all of the known factors of variation.

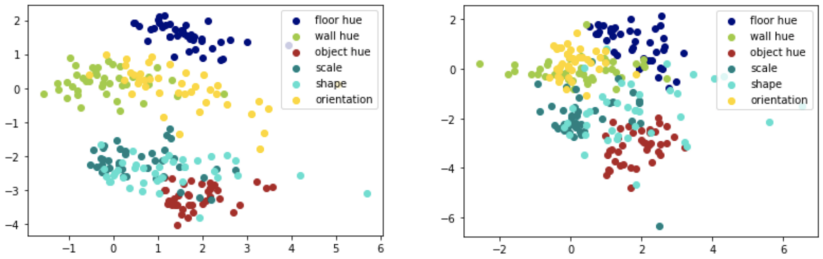


Figure 3: Output of the prototypical network with embedding dim $m = 2$ when the input is real pairs of data with a single factor with differing values amongst the pair on the *3DShapes* dataset. Each color corresponds to a different uniquely changed factor of variation. The network clusters the changes correctly on the pairs from the original dataset. This suggests that the prototypical network is clustering pairs of images based on the changed factor of variation. *Left*: $\lambda = 10$. *Right*: $\lambda = 5$.

$$\min_{w} \mathcal{L}_D(w) = -\mathbb{E}_{\hat{z}}[\log(D_w(\hat{z}))] + \mathbb{E}_z[\log(1 - D_w(z))] \tag{9}$$

Note here that the real samples for the discriminator are the intervened representations and the fake samples are the original representations. The encoder is trained to fool this discriminator by minimizing the loss in equation 10, by making the real representations indistinguishable from the intervened representations.

$$\min_{\phi} \mathcal{L}_E(\phi) = \mathbb{E}_z[\log(1 - D_w(z))] \tag{10}$$

The final objective (equation 11) of our method is a weighted sum of the different losses, and is optimized by the network parameters of the VAE and the prototypical network corresponding to each loss:

$$\min_{\phi,\theta,\gamma,\psi} \mathcal{L} = -\mathcal{L}_V(\phi,\theta) + \alpha\mathcal{L}_E(\phi) + \lambda\mathcal{L}_P(\gamma,\phi,\theta) + \kappa\mathcal{L}_I(\psi,\phi,\theta) \tag{11}$$

The weights of the discriminator w are optimized to minimize the objective $\mathcal{L}_D(w)$. As the training progresses some dimensions of the latent representation collapse to the prior distribution and encode no information. To ensure that interventions in these dimensions do not contribute to the losses, we scale equations 5 and 3 with the KL divergence values of the intervened latent dimension.

3 EMPIRICAL EVALUATION

To empirically evaluate our method, we perform both quantitative and qualitative evaluation on two synthetic datasets and one real dataset with known factors of variation and qualitative evaluation on CelebA dataset Liu et al. (2015). The two synthetic and one real datasets are generated from independent ground truth factors of variation; DSprites Matthey et al. (2017) binary 64 x 64 images with 5 factors of variation: 3 shapes, 6 scales, 40 orientations, 32 x-positions and 32 y-positions; 3D Shapes Burgess & Kim (2018) 64 x 64 x 3 color images with 6 factors of variations: 4 shapes, 8 scales, 15 orientations, 10 floor colors, 10 wall colors and 10 object colors; MPI3D real Gondal et al. (2019) 64 x 64 x 3 color images with 7 factors of variations: 6 colors, 6 shapes, 2 sizes, 3 camera heights, 3 background colors, 40 horizontal axis, 40 vertical axis, and one real world dataset; CelebA. The details of the architecture for the different components and the corresponding hyperparameters are listed in the supplementary material (Table 2) and (Table 4) respectively.

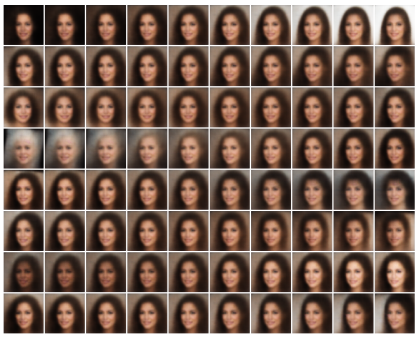
DATASETS	DSPRITES			3DSHAPES		
MODEL	FVAE	DCI	MIG	FVAE	DCI	MIG
β-VAE	$0.51 \pm .10$	$0.23 \pm .10$	$0.15 \pm .10$	$0.81 \pm .10$	$0.44 \pm .17$	$0.28 \pm .18$
ANNVAE	$0.70 \pm .10$	$0.28 \pm .10$	$0.23 \pm .10$	$0.84 \pm .09$	$0.46 \pm .16$	$0.31 \pm .15$
β-TCVAE	$0.68 \pm .10$	$0.35 \pm .06$	$0.17 \pm .09$	$0.88 \pm .07$	$0.63 \pm .10$	$0.40 \pm .18$
FVAE	$0.74 \pm .06$	$0.38 \pm .10$	$0.28 \pm .09$	$0.81 \pm .06$	$0.47 \pm .12$	$0.33 \pm .14$
GR-FVAE	$\mathbf{0.75 \pm .08}$	$0.41 \pm .07$	$0.31 \pm .06$	$0.79 \pm .06$	$0.49 \pm .06$	$0.43 \pm .11$
PROTOVAE	$0.70 \pm .06$	$\mathbf{0.51 \pm .04}$	$\mathbf{0.37 \pm .09}$	$\mathbf{0.90 \pm .06}$	$\mathbf{0.84 \pm .07}$	$\mathbf{0.71 \pm .11}$

Table 1: Various disentanglement metrics evaluated across a number of state of the art methods for the DSprites and 3Dshapes dataset. For all metrics, **higher is better.** The results for the other models are obtained using the hyperparameter settings and experimental conditions as described in Locatello et al. (2019). The scores for all the models were averaged across ten runs with different random seeds, with standard deviation shown as $\pm$. Gr-FVAE is the GroupifyVAE variant applied to the FactorVAE, as this is the closest to our model and the results for which are taken from Yang et al. (2021). The highest values in a column are written in bold. As we see, the ProtoVAE outperforms the state of the art on a majority of the metrics. Proto-VAE hyperparameters for DSprites and 3Dshapes results shown are $\{\alpha = 10, \lambda = 10, \kappa = 10\}$ and $\{\alpha = 20, \lambda = 20, \kappa = 20\}$

We qualitatively evaluate our model by performing traversals in the learned latent space, by fixing all dimensions of the representation except one and varying the values linearly in a fixed range $[-2, 2]$ for that dimension. A model has better disentanglement if the changes made in the data space while traversing a dimension are similar to the changes a factor of variation makes in the data space.

Our model both finds the correct number of factors and encodes them separately without any specific hyper-parameter tuning. From Figure 2 we can see that our method ProtoVAE, produces disentangled traversals covering both the number of factors as well as the entirety of the range of the values for the Dsprites and the 3DShapes dataset. The latent traversals on the MPI3D dataset can be found in Appendix 6. Our method effectively separates the factors thus disentangling the learned latent representation without compromising on the reconstruction quality as seen from row 2. Owing to the unsupervised nature our method struggles to exactly disentangle the non-isometric discrete factor of shape in the DSprites dataset. For the 3DShapes dataset, in our traversals in 2, we achieved near perfect disentanglement, completely unsupervised. In the Appendix D, we show the performance of the ProtoVAE for a subset of the Dsprites dataset with only a few factors. We show traversals from models FactorVAE and $\beta-$VAE, along with our model, with only a few isometric factors for comparison. Our proposed ProtoVAE is the only model that does not conflate two factors and encodes factors which are interpretable.

Furthermore, we quantitatively evaluate the learned representation by calculating state-of-the-art disentanglement metrics. We choose metrics from each of the three kinds of metrics described in (Zaidi et al., 2021); Intervention-based FactorVAE Kim & Mnih (2019), Predictor-based Disentanglement-Completeness-Informativeness (DCI) (Eastwood & Williams, 2018) and Information-based Mutual Information Gap (MIG) (Chen et al., 2019). The metrics were implemented as proposed in Locatello et al. (2019) with the same hyperparameters. We refer the interested readers to Zaidi et al. (2021)



Model	FVAE	DCI	MIG
β-VAE	$.41 \pm .05$	$.23 \pm .04$	$.06 \pm .03$
AnnVAE	$.29 \pm .04$	$.12 \pm .02$	$.07 \pm .07$
β-TCVAE	$.45 \pm .06$	$.27 \pm .03$	$.16 \pm .03$
FVAE	$.40 \pm .04$	$.30 \pm .03$	$\mathbf{.23 \pm .03}$
DisCo	$.39 \pm .07$	$.29 \pm .02$	$.07 \pm .03$
ProtoVAE	$\mathbf{.46 \pm .04}$	$\mathbf{.38 \pm .05}$	$\mathbf{.25 \pm .11}$

Figure 4: *Left:* Latent traversals on the CelebA dataset ProtoVAE successfully captures ground-truth factors of variation on real-world data. From top to bottom: background color, hairstyle, head angle, age, hairstyle, hair color, skin color, face profile. *Right:* Quantitative comparative metrics on the MPI3D dataset. ProtoVAE performs comparatively or better consistently across multiple metrics on a difficult real disentanglement dataset. See Appendix Fig. 6 for latent traversals on MPI3D. Proto-VAE hyperparameters for results shown are $\{\alpha = 10, \lambda = 2, \kappa = 2\}$. The numbers for DisCo have been borrowed from their paper Ren et al. (2021) for the VAE-based methods.

for an intuitive understanding of the metrics. We highlight here that our model achieves a higher DCI metric and a higher corresponding *completeness* and *informativeness* metric, which reflects the mode covering capabilities of the learned representation.

From table 1 we see that on the DSprites dataset, our method outperforms the state of the art models in a majority of the metrics. Similar performance of our model on the 3DShapes dataset as seen in table. Also, the variance in the metrics for the different runs is significantly lower than of the previous methods, thus ensuring a more robust way to disentangle representations. For the real disentanglement dataset of MPI3D (Gondal et al., 2019) consisting of a camera taking photos of an object attached to a jointed arm, we see that our model consistently either matches or outperforms the state of the art. From the baselines, especially important is the FactorVAE model, which is the base model upon which we add our contributions for the ProtoVAE model and hence use it as a comparison to demonstrate the effectiveness of our contributions.

On the CelebA dataset, we find that across multiple runs, our model is able to find the same "natural" decompositions that correspond to human-interpretable factors of variation consistently (Fig. 4). We notice that the model is not constrained to completely encode one factor per latent dimension and the model might encode different ranges of a factor in different latents; we see this occur for example when it encodes half of the azimuth in one latent, and half in another. However, as we can see, for the most part, each latent dimension contains information only about one factor of variation and even in the unsupervised regime our model still encodes natural decompositions.

We visualize the embedding space of a trained prototypical network using our method in Fig. 3. We see that when input to the prototypical network is pairs of images from the dataset, with one ground truth factor differing in value between the pair, the prototypical network effectively clusters the pairs based on the differing factor. This clustering aligns with the labels based on the intervened dimensions during training and thus points to the effectiveness of the prototypical network for encouraging disentanglement.

We also performed quantitative and qualitative ablation studies on the 3DShapes dataset by changing the values of α, λ and κ to understand the effectiveness of each of the components and losses we introduce. The results of these ablations can be found in the Appendix, Figures 5 and 6. Furthermore, we also perform ablation studies on the effect of dimension m of the metric space of the prototypical network on the metric scores. We also show in the Appendix some limitation cases where the representations of the model did not axis align with a few factors but was rotated with respect to those factors.

4 RELATED WORKS

Many state-of-the-art unsupervised disentanglement methods extend the VAE objective function to impose additional constraints on the structure of the latent space to match the assumed independent factor distribution. β-VAE (Higgins et al., 2017) and AnnealedVAE (Burgess et al., 2018) heavily penalize the KL divergence term thus forcing the learned posterior distribution $q_\phi(z|x)$ to be independent like the prior. Factor-VAE (Kim & Mnih, 2019) and β-TCVAE (Chen et al., 2019) penalize the total correlation of the aggregated posterior $q_\phi(z)$. $TC = KL(q(z)||\prod_{i=1}^{K} q(z_i))$ where the aggregated posterior is calculated as $q_\phi(z) = \mathbb{E}_{p(x)}[q(z|x_i)] = \frac{1}{N}\sum_{i=1}^{N} q_\phi(z|x_i)$ using adversarial and statistical techniques respectively. DIP-VAE (Kumar et al., 2018) forces the covariance matrix of the aggregated posterior $q(z)$ to be close to the identity matrix by method of moment matching. The changes that we described in the latent space are defined as intervention by Suter et al. (2019) to study the robustness of the learned representations under the Independent Mechanisms (IM) (Schoelkopf et al., 2012) assumption. Most closely related to our work are VAE models that learn to disentangle by altering the latent code. In Jha et al. (2018), the authors use a VAE or AE with a split double latent code, with a cycle consistent loss, but required that attribute labels be known *a priori*, which was also a requirement in Feng et al. (2018) which learned by swapping out chunks of the latent code, and Szabó et al. (2017), which used the labels as a constraint to find unique disentanglement. The authors in Chen et al. (2020) also used cycle-consistent loss, but again required labeling. In Hu et al. (2018) the authors attempted unsupervised disentanglement with regular (non-variational) Autoencoder models, by stacking one after another, our model instead uses a prototypical neural network. In Lezama (2018) the authors derive a novel Jacobian loss combined with a student-teacher iterative training algorithm with an Autoencoder model. In Park et al. (2020) the authors develop a latent-manipulating model aimed at human-interactive image manipulation tasks.

Yang et al. (2021) use the group based definition by Higgins et al. (2018) and a cycle consistency loss to define the elements of a group. Our work differs significantly as we do not re-encode the data or the intervention-observations. Zhu et al. (2021) encode the the latent space of a VAE using the commutative Lie group and enforce constraints on the latent space. A recent work Ren et al. (2021) propose to learn disentangled representations from pre-trained models using contrastive methods.

The most prominent work from the GAN family is InfoGAN (Chen et al., 2016) which learns disentangled, semantically meaningful representations by maximizing a lower bound on the intractable mutual information between the conditioning latent variables c and the generated samples $G(z, c)$. InfoGAN-CR (Lin et al., 2020) and Zhu et al. (2020) add a contrastive regularizer to the InfoGAN model to further encourage disentanglement. (Liu et al., 2020) add orthogonal regularization to encourage independent representations. However, all the GAN methods suffer from the limitation that they require *a priori* the number of factors to be discovered, in addition to the number of values for all the discrete factors. For fairness of comparison, we thus only compare against methods that do not require these priors.

5 CONCLUSION AND FUTURE WORK

In this work, we proposed a novel model consisting of a VAE and a Prototypical Network for learning disentangled representations in a completely unsupervised way, inspired by recent discovery of sufficient inductive biases. We impose constraints on the structure of the representations learned by training the model in an self-supervised manner. Moreover, our model has the mode covering properties of the VAE-based methods as seen by the metrics and affords the interpretability of the GAN-based models by discovering "natural" factors of variations. Our proposed method is able to outperform other state of the art networks on a number of metrics on two prominent disentanglement datasets. For future work, our method can be easily adapted to be trained in a weakly supervised regime with pairs of data differing in known number of factors being the prototypes for the prototypical network. The results can be possibly improved by intervening on multiple dimensions of the representations simultaneously. The importance of methods that can disentangle data *without labels* is critical as data is plentiful and the resulting representations give interpretable insights into the variations in the data distribution. Our hope is this work adds evidence that self-supervised methods are important in this endeavor.

REFERENCES

Yoshua Bengio. Deep learning of representations: Looking forward, 2013.

Chris Burgess and Hyunjik Kim. 3d shapes dataset. https://github.com/deepmind/3dshapes-dataset/, 2018.

Christopher P. Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in β-vae, 2018.

Jiawei Chen, Janusz Konrad, and Prakash Ishwar. A cyclically-trained adversarial network for invariant representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 782–783, 2020.

Ricky T. Q. Chen, Xuechen Li, Roger Grosse, and David Duvenaud. Isolating sources of disentanglement in variational autoencoders, 2019.

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

David L. Donoho and Carrie Grimes. Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *Proceedings of the National Academy of Sciences*, 100(10):5591–5596, 2003. ISSN 0027-8424. doi: 10.1073/pnas.1031596100. URL https://www.pnas.org/content/100/10/5591.

Cian Eastwood and Christopher K. I. Williams. A framework for the quantitative evaluation of disentangled representations. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=By-7dz-AZ.

Zunlei Feng, Xinchao Wang, Chenglong Ke, An-Xiang Zeng, Dacheng Tao, and Mingli Song. Dual swap disentangling. *Advances in neural information processing systems*, 31, 2018.

Muhammad Waleed Gondal, Manuel Wuthrich, Djordje Miladinovic, Francesco Locatello, Martin Breidt, Valentin Volchkov, Joel Akpo, Olivier Bachem, Bernhard Schölkopf, and Stefan Bauer. On the transfer of inductive bias from simulation to the real world: a new disentanglement dataset. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/d97d404b6119214e4a7018391195240a-Paper.pdf.

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

I. Higgins, Loïc Matthey, A. Pal, C. Burgess, Xavier Glorot, M. Botvinick, S. Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2017.

Irina Higgins, David Amos, David Pfau, Sebastien Racaniere, Loic Matthey, Danilo Rezende, and Alexander Lerchner. Towards a definition of disentangled representations, 2018.

Daniella Horan, Eitan Richardson, and Yair Weiss. When is unsupervised disentanglement possible? *Advances in Neural Information Processing Systems*, 34, 2021.

Qiyang Hu, Attila Szabó, Tiziano Portenier, Matthias Zwicker, and Paolo Favaro. Disentangling factors of variation by mixing them, 2018.

Ananya Harsh Jha, Saket Anand, Maneesh Singh, and VSR Veeravasarapu. Disentangling factors of variation with cycle-consistent variational auto-encoders. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 805–820, 2018.

Hyunjik Kim and Andriy Mnih. Disentangling by factorising, 2019.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2014.

Abhishek Kumar, Prasanna Sattigeri, and Avinash Balakrishnan. Variational inference of disentangled latent concepts from unlabeled observations. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=H1kG7GZAW.

José Lezama. Overcoming the disentanglement vs reconstruction trade-off via jacobian supervision. In *International Conference on Learning Representations*, 2018.

Zinan Lin, Kiran Koshy Thekumparampil, Giulia Fanti, and Sewoong Oh. Infogan-cr and model-centrality: Self-supervised model training and selection for disentangling gans, 2020.

Bingchen Liu, Yizhe Zhu, Zuohui Fu, Gerard de Melo, and Ahmed Elgammal. Oogan: Disentangling gan with one-hot sampling and orthogonal regularization, 2020.

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.

Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *International Conference on Machine Learning*, pp. 4114–4124, 2019.

Loic Matthey, Irina Higgins, Demis Hassabis, and Alexander Lerchner. dsprites: Disentanglement testing sprites dataset. https://github.com/deepmind/dsprites-dataset/, 2017.

Taesung Park, Jun-Yan Zhu, Oliver Wang, Jingwan Lu, Eli Shechtman, Alexei Efros, and Richard Zhang. Swapping autoencoder for deep image manipulation. *Advances in Neural Information Processing Systems*, 33:7198–7211, 2020.

Xuanchi Ren, Tao Yang, Yuwang Wang, and Wenjun Zeng. Do generative models know disentanglement? contrastive learning is all you need. *arXiv preprint arXiv:2102.10543*, 2021.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models, 2014.

Bernhard Schoelkopf, Dominik Janzing, Jonas Peters, Eleni Sgouritsa, Kun Zhang, and Joris Mooij. On causal and anticausal learning, 2012.

Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning, 2017.

Raphael Suter, ore Miladinović, Bernhard Schölkopf, and Stefan Bauer. Robustly disentangled causal mechanisms: Validating deep representations for interventional robustness, 2019.

Attila Szabó, Qiyang Hu, Tiziano Portenier, Matthias Zwicker, and Paolo Favaro. Challenges in disentangling independent factors of variation. *arXiv preprint arXiv:1711.02245*, 2017.

Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle, 2015.

Andrew G Wilson and Pavel Izmailov. Bayesian deep learning and a probabilistic perspective of generalization. *Advances in neural information processing systems*, 33:4697–4708, 2020.

Tao Yang, Xuanchi Ren, Yuwang Wang, Wenjun Zeng, and Nanning Zheng. Towards building a group-based unsupervised representation disentanglement framework. In *International Conference on Learning Representations*, 2021.

Julian Zaidi, Jonathan Boilard, Ghyslain Gagnon, and Marc-André Carbonneau. Measuring disentanglement: A review of metrics, 2021.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

Xinqi Zhu, Chang Xu, and Dacheng Tao. Learning disentangled representations with latent variation predictability, 2020.

Xinqi Zhu, Chang Xu, and Dacheng Tao. Commutative lie group vae for disentanglement learning. In *International Conference on Machine Learning*, pp. 12924–12934. PMLR, 2021.

A APPENDIX

A.1 VARIATIONAL AUTOENCODERS

Variational Autoencoders (VAEs) Kingma & Welling (2014); Rezende et al. (2014) are generative models, which consist of an encoder and decoder, instantiated as deep neural networks, which aim to maximize the data log-likelihood calculated according to the generative model $p_\theta(x|z)$ with $z \sim p(z)$:

$$\log p(x) = \log\left(\int_z p_\theta(x|z)\, p(z) dz\right) \tag{12}$$

where z is a lower dimensional latent variable than x and is sampled from a known prior. However, calculating this exact log-likelihood is intractable due to higher dimensional latent space, hence, VAEs instead maximize a lower-bound by approximating the true posterior distribution $p_\theta(z|x)$ with tractable distribution $q_\phi(z|x)$ learned by the encoder:

$$\max_{\theta,\phi} \mathcal{L}_1(\theta,\phi) = \mathbb{E}_{q(x)}[\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - \beta D_{KL}(q_\phi(z|x)||p(z))] \tag{13}$$

Where $q(x)$ is the empirical distribution of the dataset given by $q(x) = \frac{1}{N}\sum_{i=1}^{N}\delta(x^{(i)})$ and $D_{KL}(q_\phi(z|x)||p(z)) = \mathbb{E}_{q_\phi(z|x)} \log \frac{q_\phi(z|x)}{p(z)}$.

θ and ϕ parameterize the distributions learned by the decoder and encoder of a VAE respectively. The top-down generator $p_\theta(x|z)$ and a bottom up encoder network $q_\phi(z|x)$ and trained jointly to maximize lower bound of the empirical distribution of the training data, when passed through the information bottleneck z Burgess et al. (2018).

Given a training example $x^{(i)} : i \in \{1, \cdots, N\}$, where N is the total number of training points in the dataset, the encoder network maps $x^{(i)}$ to a lower dimensional representation $z^{(i)}$, which is sampled from the encoder distribution $q_\phi(z|x)$,and the decoder network maps the representation back to the higher dimensional data space $\hat{x}^{(i)}$. In the objective function in 13, first term measures the difference between the example $x^{(i)}$ and the reconstructed example $\hat{x}^{(i)}$ and the second term measures the distance between the approximate posterior distribution $q_\phi(z|x)$ of the learned representation and the assumed prior distribution $p(z) \sim \mathcal{N}(0, I)$. VAEs, and variants thereof, have shown to disentangle data by imposing structural constraints in the latent space and owing to the information bottleneck principle Burgess et al. (2018). Our network builds on this base capability of the VAE.

A.2 PROTOTYPICAL NETWORKS

Prototypical Networks Snell et al. (2017), originally used for the few-shot supervised classification regime, are used to learn a metric embedding space where examples belonging to the same class cluster around a prototype representation of the class. The base prototypical network is based around a supervised loss, where the labels are known *a priori* and the network is used to project the data into a new clustered metric space. The network is trained by training episodes consisting of labeled support and query sets, where the support sets are used to find the class prototypes and the query sets are used to evaluate the loss. Mathematically, they are given a labeled support set $S = \{S_1, \cdots S_K\}$, where K is the number of classes, and where each set $S_k = \{(x_k^{(1)}, t_k^{(1)}), \cdots, (x_k^{(N)}, t_k^{(N)})\}$ contains $x_k^{(i)} \in \mathbb{R}^D$ such that $t_k^{(i)} = k$. Prototypical networks compute an M-dimensional representation $c_k \in \mathbb{R}^m$, or *prototype*, of each class, $k \in \{1, \cdots, K\}$, through an embedding function $f_\gamma : \mathbb{R}^D \to \mathbb{R}^m$, usually a deep neural network with parameters γ.

Each prototype c_k is the mean vector of the embedded support points belonging to its class k:

$$c_k = \frac{1}{|S_k|} \sum_{(x_k^{(i)}, t_k^{(i)}) \in S_k} f_\gamma(x_k^{(i)}) \tag{14}$$

For a given query point x_q, classification is performed by calculating the distance of the query point's embedding $f_\gamma(x_q)$ with the prototypes of the different classes and assigning the label of the closest prototype. While any distance metric $d : \mathbb{R}^M \times \mathbb{R}^M \to [0, +\infty)$ can be used, the Euclidean distance

metrics is used in practice which corresponds to the spherical Gaussian density. A distribution over the classes is calculated by applying a softmax function on these distances:

$$p_\gamma(t = k|x_q) = \frac{\exp\left(-d(f_\gamma(x_q), c_k)\right)}{\sum_{k'} \exp\left(-d(f_\gamma(x_q), c_{k'})\right)} \tag{15}$$

The weights of the embedding function are updated by minimizing the negative log-probability of the true class:

$$\min_\gamma \mathcal{L}_2(\gamma) = -\log p_\gamma(t = k|x_q) \tag{16}$$

The above loss is minimized when examples belonging to the same class cluster around the class prototype sufficiently far from the prototypes of the other clusters. For our work, we convert this into an unsupervised training regime by creating labels on the fly in a self-supervised manner, as described in the text.

A.3 Disentangled Representations via Group Theory

We base our work on the proposed formalized definition proposed by Higgins et al. (2018), which connect the disentangled representations to symmetry transformations using group theory. They define the notion of symmetry transformations as transformations which affect only certain aspects or properties of the data while leaving the rest unchanged. The set of these symmetry transformations constitute a group G and their corresponding effect on some abstract space are defined as the *actions* of the symmetry group $\cdot : G \times X \rightarrow X$. Furthermore, they define actions as disentangled with respect to some group decomposition $G = G_1 \times \cdots \times G_n$ if there exist a decomposition of the abstract space $X = X_1 \times \cdots X_n$ and actions $\cdot_i : G_i \times X_i \rightarrow X_i$ such that

$$(g_1, \cdots g_n) \cdot (v_1, \cdots v_n) = (g_1 \cdot v_1, \cdots g_n \cdot v_n) \tag{17}$$

Using these definitions Higgins et al. (2018) characterize a representations as disentangled with respect to a particular decomposition of a symmetry group into subgroups if it decomposes into independent subspaces, where each subspace is affected by only the action of a single subgroup.

Importantly, this definition also brings in an *agent* as a critical aspect of the definition. Disentanglement is no longer disembodied here: an agent takes a specific action (a symmetry transformation), and observes the result of the transformation. Formally, we can say that agent's representation of the data is disentangled with respect to a particular group decomposition if: (1) An action $G \times Z \rightarrow Z$ exists, (2) there exists a map $f : W \rightarrow Z$ which is equivariant between actions on W and Z, and (3) each element of the decomposition of the representation $Z = Z_1 \times ... \times Z_n$ is affected by only one transformation G_i.

B Implementation details

In this section we give more details and descriptions of our proposed ProtoVAE architecture which consists of. The visual depiction of the model is shown in Fig. 1. The layers and layer sizes are given in Table 2. The hyperparameters used are given in Table 4. The architecture of the latent discriminator a Multi-Layer Perceptron (MLP) with 6 layers with 1000 hidden dimension per layer and 2 output logits as in Kim & Mnih (2019). We normalize the data between [0,1] as input to the encoder and the output of the decoder is the mean of a Bernoulli distribution $p_\theta(x|z)$. The reconstruction cost in the first term of 13 is calculated as the negative cross-entropy between the output of the decoder and the input to the encoder. Code available at: https://github.com/protovae/protovae.

B.1 SELF-SUPERVISED DATA GENERATION

Algorithm 1: Self-Supervised Dataset Creation for Prototypical Network Learning

input : Data examples $\{x^{(i)}\}_{i=1}^{B}$, Batch size B, latent dimension d
given : *Encoder:* $q_\theta(z|x^{(i)})$, *Decoder:* $p_\theta(x|z^{(i)})$;
output: Support Set: $S = \{S_1, S_2, ..., S_d\}$ s.t. $S_i = \{s_i^1, \cdots, s_i^B\}$, Query Set: $Q = \{q_l^1, \cdots, q_l^B\}, l \in_R [d]$, Isometry Loss Labels: $I = \{I_j; \forall j \in [d]\}$

Encode data examples into latent space $\{z^{(i)} \sim q_\theta(z|x^{(i)})\}_{i=1}^{B}$ with dimension d
Generate each Support Set $S_j \;\; \forall j \in d$:
for $j = 1$ **to** d **do**
  $\zeta \leftarrow$ random permutation of $[B]$
  $\hat{z}_j^{(i)} \leftarrow z_j^{(\zeta(i))} \quad \forall i \in [B]$: swap the jth dimension of example (i) with that of $k_{(k \neq i)}$
  $\hat{\mathbf{z}}_j^{(i)} \leftarrow (z_1^{(i)}, \cdots \hat{z}_j^{(i)}, \cdots z_d^{(i)}) \quad \forall i \in [B]$
  $\hat{x}_j^{(i)} \sim p_\theta(x|\hat{\mathbf{z}}_j^{(i)}) \quad \forall i \in [B]$ generate data from intervened latent
  $s_j^{(i)} \leftarrow (\hat{x}^{(i)}, \hat{x}_j^{(i)}) \quad \forall i \in [B]$ assign examples to an element of a support set with label j
  $S_j \leftarrow \{s_j^1, \cdots, s_j^B\}$
  $I_j \leftarrow \{\|\mathbf{z}^{(i)} - \hat{\mathbf{z}}_j^{(i)}\|^2 : \quad \forall i \in [B]\}$
end
$S \leftarrow \{S_1, \cdots, S_d\}, I \leftarrow \{I_1, \cdots, I_d\}$
Generate Query Set Q:
for $i = 1$ **to** B **do**
  $\chi \leftarrow$ random permutation of $[B]$
  $k \in_R [d]$ dimension sampled randomly from set of dimensions
  $\hat{z}_k^{(i)} \leftarrow z_k^{(\chi(i))}$
  $\hat{\mathbf{z}}_k^{(i)} \leftarrow (z_1^{(i)}, \cdots \hat{z}_k^{(i)}, \cdots z_d^{(i)})$
  $\hat{x}_k^{(i)} \sim p_\theta(x|\hat{\mathbf{z}}_k^{(i)})$
  $q^{(i)} \leftarrow \{(\hat{x}^{(i)}, \hat{x}_k^{(i)}), k\}$
end
$Q \leftarrow \{q^{(1)}, \cdots, q^{(B)}\}$

B.2 ARCHITECTURE DETAILS

ENCODER	DECODER	PROTONET
INPUT: CHANNELS X 64 × 64	INPUT: $\in \mathbb{R}^{10}$	INPUT: 2 X CHANNELS X 64 × 64
4 × 4 CONV. 32 LRELU. STRIDE 2	FC. 128 LRELU.	4 × 4 CONV. 32 LRELU. STRIDE 2
4 × 4 CONV. 32 LRELU. STRIDE 2	FC. 4 × 4 × 64 LRELU.	4 × 4 CONV. 32 LRELU. STRIDE 2
4 × 4 CONV. 64 LRELU. STRIDE 2	4 × 4 UPCONV. 64 LRELU. STRIDE 2	4 × 4 CONV. 64 LRELU. STRIDE 2
4 × 4 CONV. 64 LRELU. STRIDE 2	4 × 4 UPCONV. 64 LRELU. STRIDE 2	4 × 4 CONV. 64 LRELU. STRIDE 2
FC. 128 LRELU.	4 × 4 UPCONV. 64 LRELU. STRIDE 2	FC. 128 LRELU.
FC. 2 × 10.	4 × 4 UPCONV. 1 SIGMOID. STRIDE 2	FC. 2 , FC. 10

Table 2: Network Architecture of the VAE Encoder, VAE Decoder, and the Prototypical Network.

B.3 NETWORK HYPERPARAMETERS

We detail the hyperparameters used for each dataset. Table 4 shows the full list of values for each component.

PARAMETER (DSPRITES)	VALUE
LATENT SPACE DIMENSION	10
METRIC SPACE DIMENSION	3
OPTIMIZER	ADAM
OPTIMIZER (BETA1)	0.9
OPTIMIZER (BETA2)	0.999
LEARNING RATE (VAE)	$1e$-4
LEARNING RATE (GAN)	$1e$-4
LEARNING RATE (PROTO)	$1e$-4
BATCH SIZE	128
EPOCHS (TRAIN)	50
α (TOTAL CORRELATION (GAN)	10
λ (PROTOTYPICAL LOSS)	10
κ (ISOMETRY LOSS)	10

PARAMETER (3DSHAPES, CELEBA)	VALUE
LATENT SPACE DIMENSION	10
METRIC SPACE DIMENSION	2
OPTIMIZER	ADAM
OPTIMIZER (BETA1)	0.9
OPTIMIZER (BETA2)	0.999
LEARNING RATE (VAE)	$1e$-04
LEARNING RATE (GAN)	$5e$-05
LEARNING RATE (PROTO)	$1e$-04
BATCH SIZE	128
EPOCHS (TRAIN)	30
α (TOTAL CORRELATION (GAN)	20
λ (PROTOTYPICAL LOSS)	20
κ (ISOMETRY LOSS)	20

Table 3: Hyperparameters values for the different components of the network for the different datasets.

PARAMETER (MPI3D)	VALUE
LATENT SPACE DIMENSION	10
METRIC SPACE DIMENSION	3
OPTIMIZER	ADAM
OPTIMIZER (BETA1)	0.9
OPTIMIZER (BETA2)	0.999
LEARNING RATE (VAE)	$1e$-4
LEARNING RATE (GAN)	$5e$-5
LEARNING RATE (PROTO)	$1e$-4
BATCH SIZE	128
EPOCHS (TRAIN)	30
α (TOTAL CORRELATION (GAN)	10
λ (PROTOTYPICAL LOSS)	2
κ (ISOMETRY LOSS)	2

Table 4: Hyperparameters values for the different components of the network for the MPI3D dataset

C ABLATION STUDIES (METRICS AND IMAGES)

?? We demonstrate qualitative and quantitative results showing the effectiveness of the different hyperparameters for disentanglement by varying them across a range while keeping the others the same. We perform these ablations studies on the 3D shapes dataset with hyperparameters given in Table 4 except the one hyperparameter which is varied. in Table 5 we vary the dimension of the metric space of the prototypical network and see it's effect on the various metrics. In Table 5 we vary the strength of the total correlation loss as measured by the latent space GAN to train the VAE by varying the hyperparameter α. In Table 6 we perform similar experiments by varying the hyperparameter λ which scales the combined loss of the consistency and uniqueness loss and the regularization strength of the Isometry loss which is governed by the hyperparameter κ.

Qualitatively we show the latent traversals of results where learned latent representations are not axis aligned with the ground-truth factors of variations. In Figure 7 we show the effects of having a lower value of λ and see that even though each dimension encodes information related to one factor, the complete range of the factor is encoded by multiple dimensions. In Figure 8 we show the effect of having a lower regularization for the Isometry loss (κ) and see that we learn a subset of factors which are rotated in space and are not axis aligned with the ground truth factors.

In addition, we perform an ablation study by removing the GAN discriminator and seeing the result of just the VAE and prototypical network in Fig. 9. We found that without the GAN, a lower dimensional prototype metric space is needed (we use dimension $d = 1$), but that the network is

METRIC	$m = 2$	$m = 5$	$m = 8$
FVAE	$.90 \pm .06$	$.88 \pm .04$	$.87 \pm .04$
DCI	$.84 \pm .07$	$.77 \pm .05$	$.74 \pm .05$
MIG	$.71 \pm .11$	$.62 \pm .08$	$.57 \pm .06$
β-VAE	$.92 \pm .05$	$.91 \pm .05$	$.91 \pm .04$

METRIC	$\alpha = 0$	$\alpha = 10$	$\alpha = 20$
FVAE	$.93 \pm .04$	$.85 \pm .03$	$.88 \pm .05$
DCI	$.78 \pm .05$	$.81 \pm .06$	$.81 \pm .07$
MIG	$.59 \pm .07$	$.63 \pm .04$	$.65 \pm .08$
β-VAE	$.92 \pm .06$	$.88 \pm .04$	$.90 \pm .05$

Table 5: Ablation study showing the performance on different metrics for our proposed ProtoVAE, with different values of the prototype dimension space m (*left*) and α (*right*) on the 3D shapes dataset. In some instances, these versions outperform the results posted in the main paper. We see that smaller values of the prototypical network dimension d actually perform better as it imposes stronger constraints on the VAE. We also see that $\alpha = 0$ performs well for the β VAE and the FactorVAE metrics but does not perform as well on the MIG and the DCI scores as the Latent Space GAN helps in confining the encoding of the factors into a single dimension whereas our losses do not do this as effectively.

METRIC	$\kappa = 0$	$\kappa = 10$	$\kappa = 40$
FVAE	$.68 \pm .04$	$.84 \pm .03$	$.72 \pm .04$
DCI	$.57 \pm .06$	$.72 \pm .04$	$.69 \pm .05$
MIG	$.43 \pm .06$	$.56 \pm .08$	$.58 \pm .08$
β-VAE	$.80 \pm .05$	$.84 \pm .06$	$.88 \pm .06$

METRIC	$\lambda = 0$	$\lambda = 10$	$\lambda = 20$
FVAE	$.84 \pm .06$	$.82 \pm .04$	$.88 \pm .05$
DCI	$.64 \pm .05$	$.71 \pm .02$	$.81 \pm .07$
MIG	$.58 \pm .06$	$.57 \pm .06$	$.65 \pm .08$
β-VAE	$.88 \pm .04$	$.88 \pm .05$	$.90 \pm .05$

Table 6: Ablation study showing the performance on different metrics for our proposed ProtoVAE, with different values of the hyperparameter κ (*left*) and λ (*right*) on the 3D shapes dataset.

still able to disentangle well. We leave for future work more investigations into removing the GAN component entirely.

Another promising future work would be to impose constraints on the structure of the metric space to obtain more separable clusters. Moreover, curriculum learning could be used while performing interventions to ensure that the intervened dimension is changed by a significant value and then to slowly decrease this value as training proceeds.

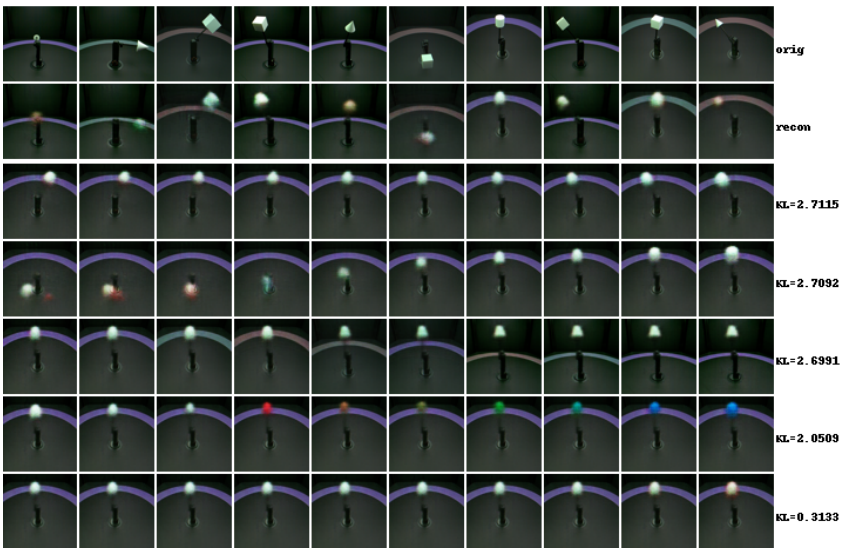


Figure 5: Latent traversals on the MPI3D real world disentanglement dataset. The data is collected via a camera that observes a jointed arm with known changed ground truth factors of variation. From top to bottom: original data, reconstruction, arm angle left/right, arm angle top/bottom, background height, arm end color, size.

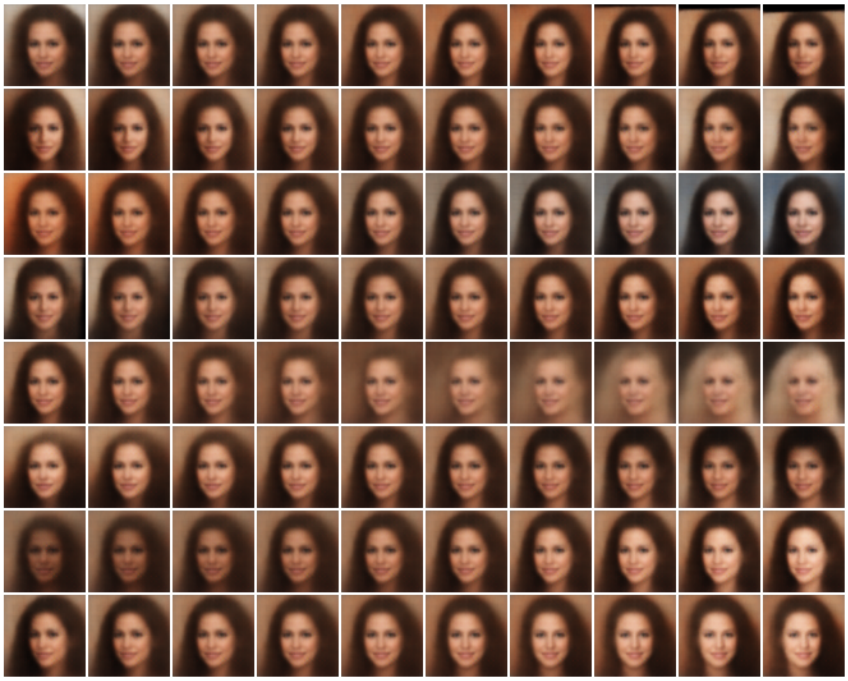


Figure 6: More latent traversals on the CelebA dataset. We see that the traversals are similar to the ones discovered in Fig. 4 of the main paper.

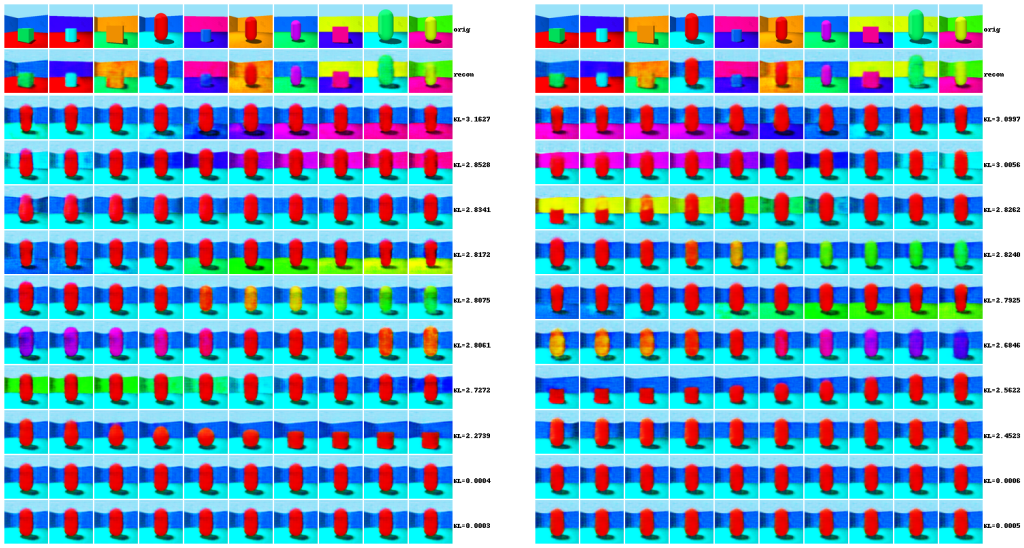


Figure 7: Results on *3dShapes* with a lower value of λ, showing theoretically correct disentanglement that disagrees with a "natural" decomposition - namely, that the factors need not be completely capture by one latent, and may be split into several; for example, here, color change need not cover the entire spectrum in one traversal. In row 7, 8, *left* and row 6,8 *right*, we can see that color traversal was split into two latents. This still follows the group theoretical definition of disentangled representations. And given a high value of κ, the mapping between the dimensions and the corresponding traversals is isometric.

C.1 Prototypical Network Embeddings

We take a closer look at the workings of the prototypical network component, and analyze its output. To do so, we trained a network using prototypical metric dimension $d = 2$, so that we can plot the

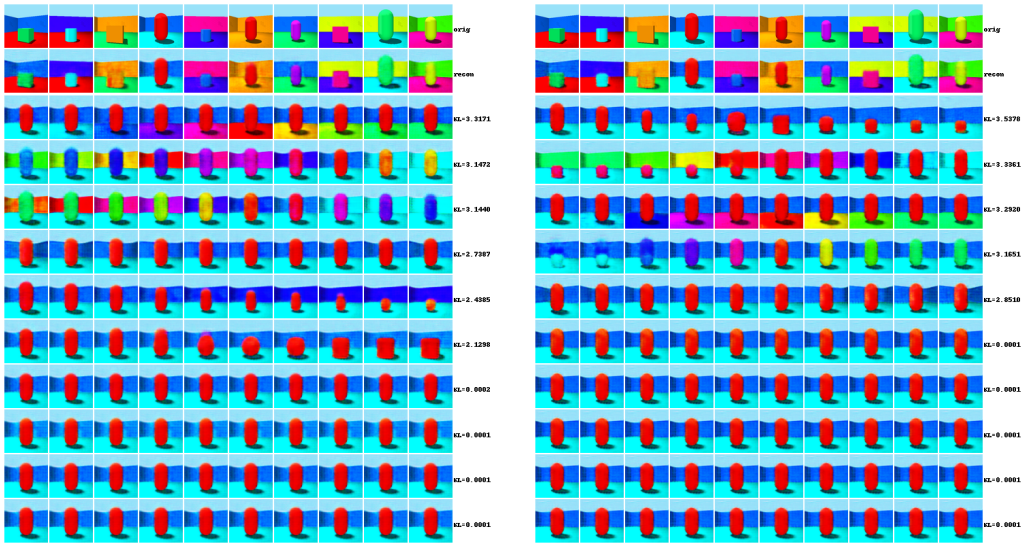


Figure 8: Similar to Fig. 7, a non-unique decomposition can be found by a 'rotation' of that factors such that facets of each factor are mixed together. We can see this here with the wall and shape color being intermingled. For this run, had a lower value of the Isometry loss scaling (κ). This shows that this loss was important in ensuring a more natural decomposition.

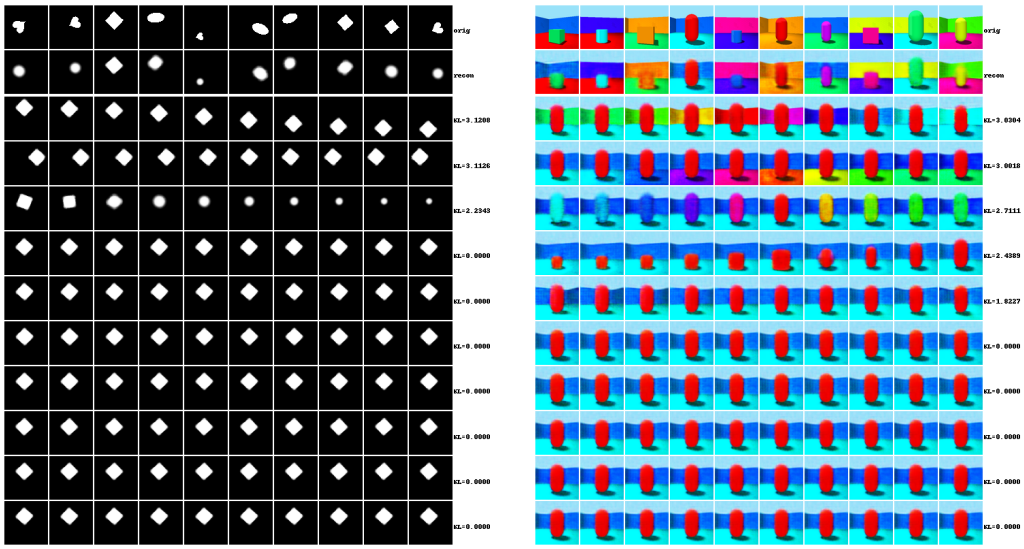


Figure 9: Results of the ablation study removing the GAN discriminator and just training a VAE with the prototypical network, on a low dimensional prototypical network metric space $d = 1$; we find that removing the GAN requires a lower dimensional metric space. However, even without the GAN discriminator we find that the network performs remarkably well at disentangling the factors of variation into natural decompositions. We also found that we needed higher values of λ and κ for better disentanglement. Due to this however, with the GAN removed not all factors are captured and the reconstructions are slightly blurrier. We leave to future work further development of this variant without the GAN discriminator.

resulting points in 2D space. We pass into the prototypical network network two data samples from the *3DShapes* dataset that differ in one ground-truth factor of variation (mimicking the intervention process, but with ground truth examples), and plotted the corresponding projection into the $\mathbb{R}^2$ space. This tests to see if the trained prototypical network will generalize the clustering to ground truth pairs of data which differ in only one factor as shown in Fig. 12. If the prototypical network has been trained correctly and aids in disentanglement of the factors, it would cluster changes to the

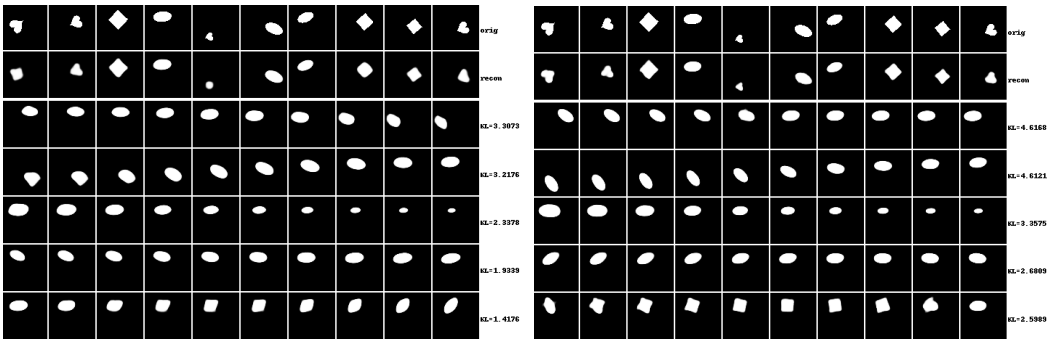


Figure 10: Results of latent traversals for the DSprites dataset for two baseline methods of β-VAE and FactorVAE. *left* The latent traversals of representations learned by a β-VAE with the same hyperparameters as listed in 4 and $\beta = 4$. *right* The latent traversals for the FactorVAE model with $\gamma = 40$. We see that our method qualitatively outperforms both these methods on the DSprites dataset with better reconstruction quality.

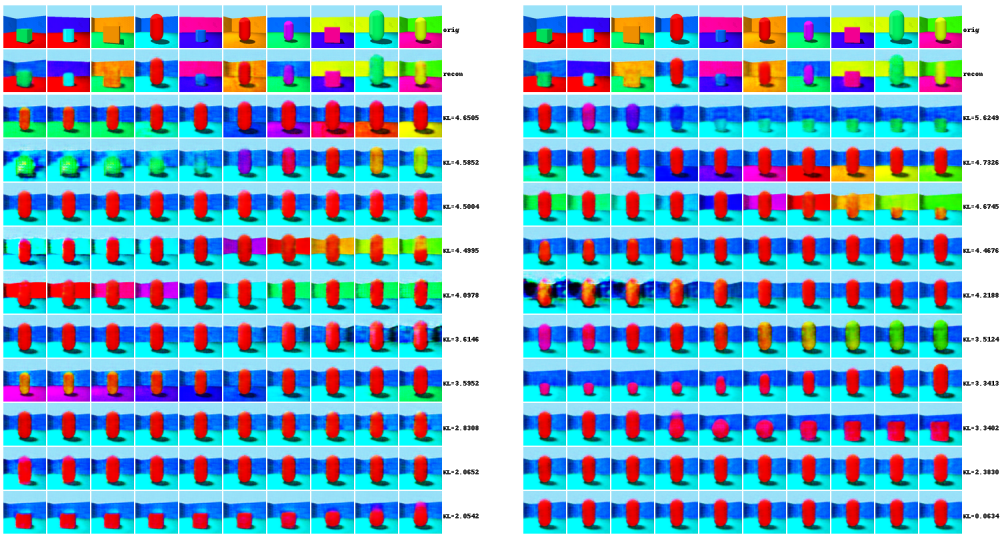


Figure 11: Results of latent traversals for the 3Dshapes dataset for the baseline method of FactorVAE. *left* The latent traversals of representations learned by the FactorVAE for $\gamma = 20$ with the same hyperparameters as listed in 4. *right* The latent traversals for the FactorVAE model with $\gamma = 40$. We see that our method qualitatively outperforms both these methods on the 3Dshapes dataset with better reconstruction quality and more disentangled factors encoded in the different latents.

same latent in similar clusters in $\mathbb{R}^2$. Notably, here we supply the prototypical network directly with *two real dataset images*, which is not the same distribution that the prototypical network is trained on. During training the prototypical network is trained on a pair of one reconstruction of a real sample and one intervention-observation. Furthermore, we also see the effect of the regularization effect of the λ parameter on the clustering for a model with the other parameters the same. In Fig. 3, we see the clusters formed by a trained model on the 3D shapes dataset.

This suggests that during training the prototypical network clusters the pair of reconstructed and intervention-observation according to the index of the dimension of the representation that was intervened on to create that pair. This in turn forces the decoder of the VAE to make consistent and unique changes to the data based on the intervened dimension. Furthermore, it is apparent that clusters of some factors are well separated whereas clusters are closer in the metric space based on the entity that they affect.

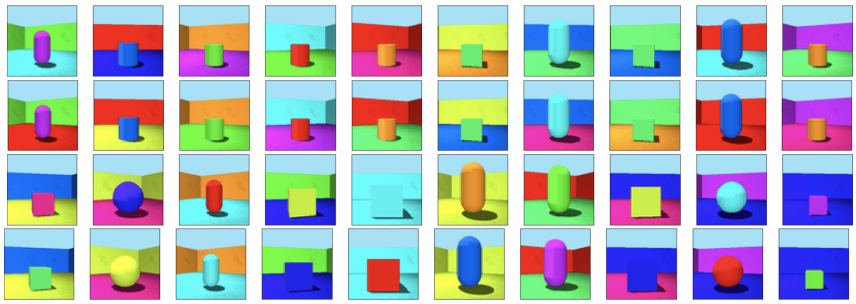


Figure 12: Inputs to the prototypical network. The first two rows represent pairs (each pair is a column) where the floor hue is the only factor that is differing amongst the pair of images. Similarly the bottom two rows represent pairs where the object hue is the differing factor.

D COMPARISON OF MODELS WITH FEW KNOWN FACTORS

To show the effectiveness of the Prototypical network in disentangling the factors of variation we perform ablation studies with smaller number of factors of variation, with the latent space being equal to the number of factors. We show the comparison of our method with the β-VAE and the Factor-VAE that we trained ourselves for fairness of comparison. As an ablation, we created subset of synthetic datasets from DSprites containing only two factors of variation: one with only position x and position y, and one with the orientation (o) and position y factors of variations. We show the reconstructions and the latent traversals of our method in the in Figure D and Figure D for the position x and position y factors and the orientation and position y factors respectively. From the figure we can see that the ProtoVAE effectively disentangles the different factors into separate dimensions of the latent representation. We can see that for the subdataset consisting of x and y, the FactorVAE and β-VAE correctly disentangle the two factors (Fig. D and Fig. D), as does our proposed ProtoVAE. However, with y and orientation, we see that the FactorVAE (Fig. D) and the β-VAE (Fig. D do not fully "naturally" disentangle the factors, leading to mixed representations in the latents.

For this, we used a prototypical metric dimension of $d = 1$. We found that the GAN is only necessary for larger metric dimension space, such as the ones we use for the main paper results. In Figure D and Figure D we show the Factor-VAEs performance and in Figure D and Figure D we show the β-VAEs performance on the same datasets. We see that the ProtoVAE outperforms these methods.

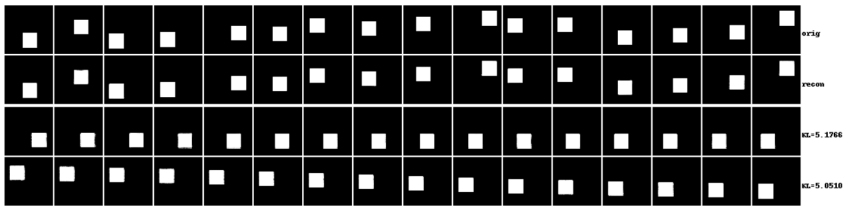


Figure 13: ProtoVAE (x & y):Traversal results for $d = 1$ ($\alpha = 0, \lambda = 10, \kappa = 10$)without the latent discriminator, on the sub-dataset from DSprites created by only varying x and y (same shape, orientation, and size.) The first row consists of examples from the dataset, the second row consists of the corresponding reconstructions. The third row shows the latent traversals for the 1st dimensions which encodes the position x factor and the 2nd row which encodes the position y factor.

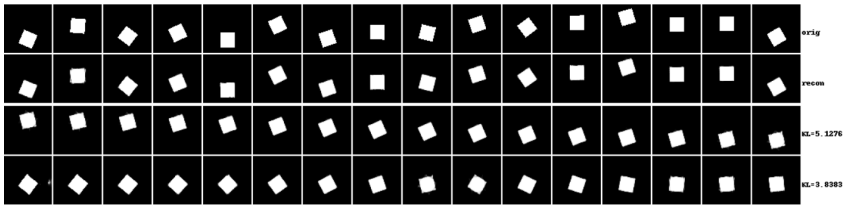


Figure 14: ProtoVAE (orientation & y-position)Traversal results for $d = 1$ without the latent discriminator, on the sub-dataset from DSprites created by only varying y and orientation (same shape, position x, and size.) The third row shows that the dimension 1 of the latent representation encodes the position y factor and dimension 2 encodes the orientation factor. Unlike FactorVAE and β-VAE (Fig. D, Fig. D), our proposed model fully disentangles the factors into separate latents without rotation.

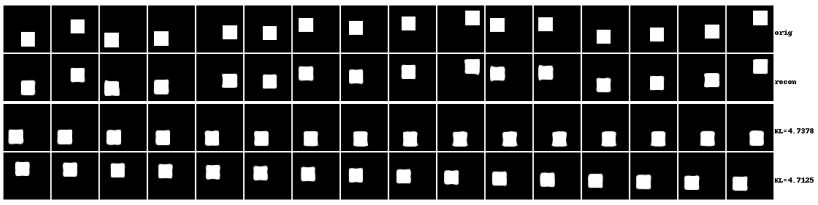


Figure 15: FactorVAE (x position & y position): Latent traversal results for FactorVAE with hyperparameter value 10. The FactorVAE correctly disentangles these two factors without any mixing.

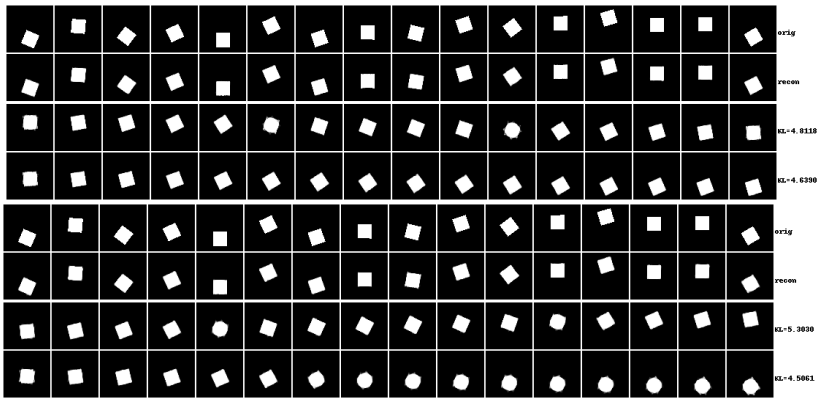


Figure 16: FactorVAE (orietnation & y-position): Traversal results for FactorVAE on the orientation and y-position dataset, showing only partial disentanglement, or a rotation of the factors. *Top four rows*: Hyperparameter value for the total correlation loss is 5. *Bottom four rows*: The value of the hyperparameter is 10. We see that a rotational subspace of the factors is learned. In particular, we can see in row 3 *top*, the y position and orientation is mixed.

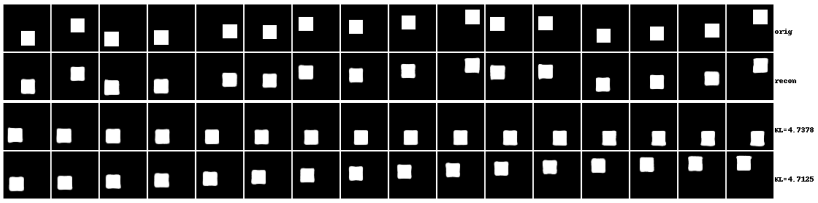


Figure 17: β-VAE on the subdataset of (x-position & y-position): Traversal results with $\beta = 10$ We see that the β-VAE did train successfully to disentangle x and y.

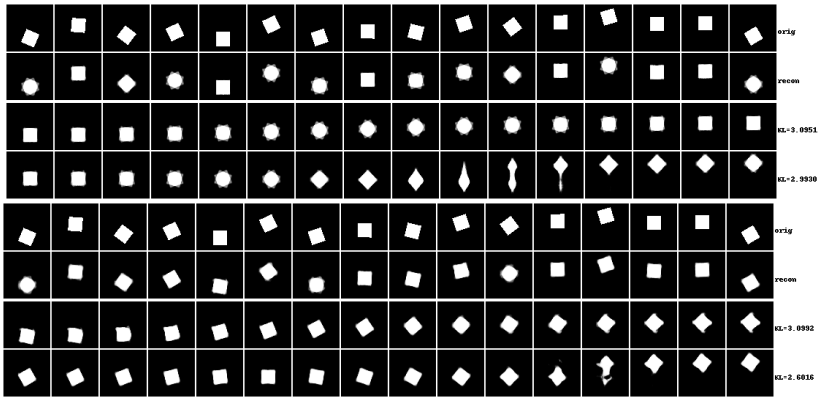


Figure 18: β-VAE on orientation and y-position: Traversal results for β-VAE on the orientation and position y dataset. *Top four rows*: $\beta = 5$. *Bottom four rows*: $\beta = 10$. We can see the network performs more poorly on these factors than our proposed model (row 4, 8) in generating smooth traversals.