Learning Disentangled Representations with Identifiable Diffusion Models

Term Paper
Advanced Medical Machine Learning Seminar 2024

Constantin Kühne and Till Zemann

Hasso Plattner Institute for Digital Engineering
{constantin.kuehne, till.zemann}@student.hpi.uni-potsdam.de

Abstract. This work introduces a novel framework that combines the Denoising Diffusion Probabilistic Model (DDPM) with an identifiable Variational Autoencoder (iVAE) for disentangled representation learning. We propose a new sampling strategy for training the model and improve the previous model by adding a deeper encoder and classifier-free guidance. Our approach leverages the strengths of diffusion models in capturing data distributions across scales and the iVAE's capability to learn identifiable latent representations. Through extensive quantitative experimentation using various disentanglement metrics to compare our model to baseline models, and a qualitative evaluation on the Shapes3D dataset, we demonstrate our method's ability to generate disentangled representations.

Keywords: Disentanglement · Identifiable VAE · DDPM · Nonlinear ICA

1. Introduction

In recent years, disentangled representation learning has emerged as a pivotal area of research. This approach aims to decompose high-dimensional data such as images into their underlying low-dimensional generative factors, which increases the interpretability [22] and generalizability [2], as well as the predictive accuracy for downstream tasks, such as classification [14] and segmentation [6].

We propose a new model for disentangled representation learning that merges the Denoising Diffusion Probabilistic Model (DDPM) [10] with the identifiable VAE (iVAE) [15]. The iVAE conditions the latent representation prior $p_\psi(z)$ on additional observed variables to ensure identifiability up to invertible component-wise transformations [12]. In the context of disentangled representation learning, identifiability means that the model not only learns the marginal distribution of the data but also learns the true joint distribution of the hidden generative factors and observed data.

We hypothesize that incorporating diffusion processes within an iVAE architecture will improve the disentanglement of latent representations. By extending the architecture with a score-based DDPM, we tap into the ability of diffusion models to methodically learn features – initially learning broad, global variations in the early stages of diffusion, followed by refining and capturing more fine-grained variations in the later diffusion steps. This systematic approach allows for a nuanced representation of data variance across various scales.

Furthermore, we identify several strategies that potentially can improve the iVAE-DDPM framework and evaluate them on the Shapes3D dataset [4]. First, we integrate a ResNet backbone, which adds to the capability to extract abstract features from the data, leading to improved disentanglement. Although this seems to be a simple addition, it is a non-trivial improvement, as the datasets are not very complex and should be solvable with a simpler architecture. Additionally, we address the sampling distribution by correcting the gaussian sampling to be uniform, which matches the dataset factor distributions. Moreover, to further refine the DDPM generations, we incorporate classifier-free guidance within the reverse diffusion process, enabling the model to better follow the provided conditioning. We also extend the iVAE model to handle categorical latent variables, thus matching the discrete latent distributions in the Shapes3D dataset with the goal of better learning the latent distribution. Furthermore, to have an effective latent space regularization, we apply KL balancing, which allows us to optimize both directions of the divergence between the conditional prior encoding and iVAE encoding. During the prototyping phase, we experiment using synthetic data with a simple linear rotation matrix transformation, which provides a controlled environment for validating our sampling strategy and other improvements. Lastly, we run many lengthy experiments on the Shapes3D dataset to identify which of the proposed modifications have a positive impact on the model's ability to learn disentangled representations.

1.1. Definition of Disentanglement

In a disentangled representation [9] created by a model, the different generative factors of data, such as the color or shape of an object in an image, correspond to distinct latent dimensions and remain invariant to changes in all other latent variables.

More formally, disentanglement describes the concept where observational data are transcribed into a lower-dimensional space in which there is a one-to-one correspondence between latent variables and generative factors.

1.2. Measuring Disentanglement

Discrete MIG: The Discrete Mutual Information Gap (MIG) [5] that is computed over factors is one of many quantitative metrics designed to assess the level of disentanglement in the learned representations of a model. Formally, the MIG

is defined for a given set of latent representations, μ, and a set of ground truth generative factors, y, as follows:

$$\text{MIG} = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{H(y_k)} \left(\max_i I(\mu_i, y_k) - \max_{j \neq i} I(\mu_j, y_k) \right) \tag{1}$$

where $I(\mu_i y_k)$ represents the mutual information between the discretized i-th latent dimension and the k-th generative factor. Furthermore, $H(y_k)$ is the entropy of the k-th generative factor, and K is the total number of generative factors. The MIG formula captures the difference between the highest mutual information of a latent variable with a given factor and the second-highest mutual information of any other latent variable with that factor. By normalizing this difference with the entropy of the generative factor, $H(y_k)$, we account for the inherent uncertainty in the factor itself, thereby ensuring that the score reflects the disentanglement relative to the informativeness of the factor.

The MIG score inherently favors scenarios where each latent dimension correlates with a unique generative factor. Thus, models with high MIG scores are interpreted as having achieved disentanglement, whereas low scores suggest entanglement – where factors are mixed within dimensions.

1.3. Related Work

Our work is mainly based on two papers. Firstly, Abstreiter et al. [1] introduce diffusion-based representation learning (DRL), a new framework for unsupervised representation learning using diffusion-based generative models with the denoising score-matching objective and an additional encoder. The approach shows significant improvements in downstream tasks such as semi-supervised image classification and offers a new method for representation learning without supervision that competes with traditional autoencoders and contrastive learning methods.

Secondly, Khemakhem et al. [15] present a novel approach to independent component analysis (ICA) using a variant of the Variational Autoencoder (VAE) termed the identifiable VAE (iVAE). The paper's core contribution is the theoretical framework that allows for the identifiability of latent variables in the nonlinear ICA setting.

2. Methodology

2.1. Architecture

The foundation of our model is an identifiable Variational Autoencoder (iVAE) [15], which includes an encoder that learns $p_\theta(z|x)$ and a decoder that learns $p_\omega(\hat{x}|z)$. The encoder maps the original image to a latent space representation, while the decoder reconstructs the image from this latent space. What sets the iVAE apart from regular VAEs is an additional encoder that learns a prior $p_\psi(z|u)$ conditioned

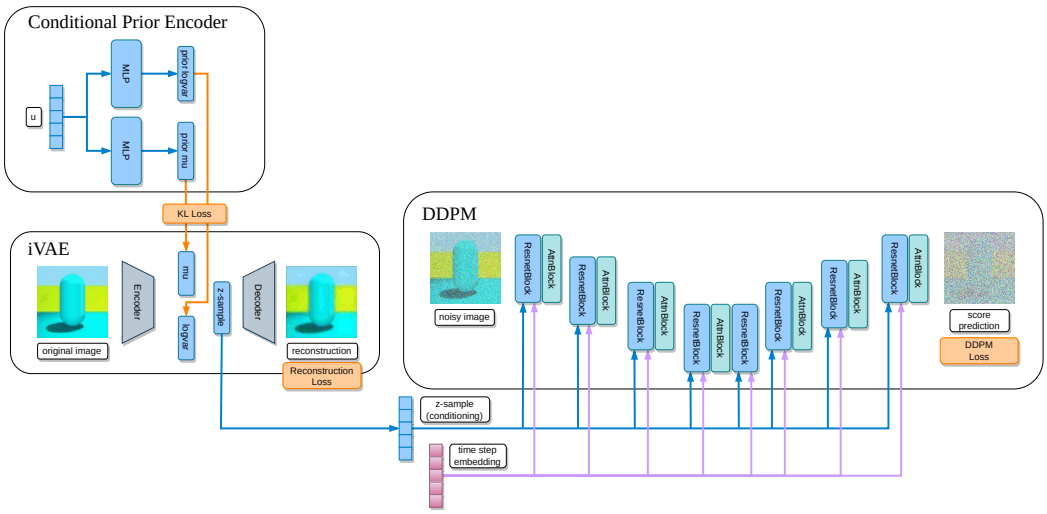


Figure 1: Model architecture and overview where the losses (illustrated in orange) are applied.

on a vector u with additional information, consisting of the variance and optionally the mean, of the label y. If the condition is met that the dimensions of u modulate the variance of the generative factors, then using the prior as a regularization target for z enables the iVAE to learn identifiable latent representations. These representations can reconstruct the generative factors by applying invertible component-wise operations (e.g. permutations and rotations). The iVAE model provides theoretical assurances that the model is able to approximately solve non-linear independent component analysis (ICA), meaning that it can learn the true joint distribution of the hidden generative factors and observed data.

Building upon the iVAE, we integrate a score-based DDPM [21] (see Figure 1). Unlike using VAEs as generative models that solely rely on a latent space for generation, DDPMs gradually denoise data by approximately solving a reverse stochastic differential equation (SDE) to model the data distribution $p_\omega(x|z)$. For this work, we use a variance-preserving SDE (VP SDE) that generally produces good results for images [19].

By integrating a DDPM with the iVAE, we benefit from the DDPMs ability to capture the complex data distribution at various scales of detail. The DDPM part of our architecture consists of several stacked residual blocks within a U-Net [18] that are conditioned on the latent vector obtained from the iVAE encoder, providing a conditioned diffusion process that is capable of generating samples with higher fidelity compared to the iVAE decoder.

2.2. Loss

Our framework's optimization objective is composed of three weighted loss functions: the score-based DDPM loss (3), a KL-Loss between the conditional prior and

the encoded samples, and a reconstruction loss between the iVAE reconstruction $\hat{x}$ and original sample x. The combined loss is given by:

$$\mathcal{L} = \mathcal{L}_{\text{DDPM}} + \lambda_{\text{rec}}\mathcal{L}_{\text{rec}} + \lambda_{\text{KL}}\mathcal{L}_{\text{KL}} \tag{2}$$

The hyperparameters λ_{rec} and λ_{DDPM} weight the contributions of the iVAE in comparison to the DDPM part of the model. Increasing both λ_{rec} and λ_{DDPM} will place a greater emphasis on the iVAE losses, lowering the contribution of the DDPM loss.

The first term of the combined objective consists of the weighted score-matching DDPM loss [20, 21]:

$$\mathcal{L}_{\text{DDPM}} = \mathbb{E}_{t,x_0,x_t}\left[\lambda(t)\left|\left|\nabla_x \log p_t(x) - s_\phi(x_t, z, t)\right|\right|_2^2\right] \tag{3}$$

where we train the predicted score $s_\phi(x_t, z, t) = \text{DDPM}_\phi(x_t, z, t)$ to match the actual score $\nabla_x \log p_t(x)$, with $\lambda(t) = \sigma^2(t)$ being a time-dependent weighting factor of the score-matching loss at time t to account for increasing variance in the diffusion process. Using $\sigma^2(t)$ as the weighting function results in the KL-divergence objective presented by Song et al. [20].

Next, we have two weighted losses that control the weight put on iVAE reconstructions and the KL divergence between the mean and standard deviation priors and latent code. The reconstruction loss is given by $\mathcal{L}_{\text{rec}} = \mathbb{E}\left[||x - \hat{x}||_2^2\right]$. Lastly, we have the KL loss that is tasked with pulling the posterior towards the prior: $\mathcal{L}_{\text{KL}} = \mathbb{E}\left[KL\left(q_\theta(z|x)||p_\psi(z)\right)\right]$.

2.3. ResNet Architecture

Due to unsatisfactory results with smaller models in the previous iteration of this research project, we explore whether the model capacity could be a hindering factor. In this section, we list the shallow models that were used previously and also compare our different architectures with deeper backbone models. This is a simple yet crucial addition, because for comparing different architectures (the VAE, iVAE and iVAE + DDPM), the results are not meaningful if the models do not have the capacity to extract good features from the data.

Baseline: For the encoder and decoder shown in Figure 1, we use two convolutional neural networks (CNNs) with a mirrored structure. The encoder consists of four convolutional layers followed by two linear layers. All layers use ReLU as their activation functions.

ResNet Encoder and Decoder: Additionally, we alter the Encoder and Decoder to use a ResNet-18 or ResNet-50 architecture [8] and compare them to the smaller baseline CNNs.

2.4. Sampling

Due to a modelling error in the first stage of this project, the normal distribution of ground truth factors z that was assumed beforehand and then used for sampling data in the training scripts does not match their actual uniform distribution in the Shapes3D and ColorDSprites datasets that we use for model evaluations, see Figure 9. As a result, a subset of images was oversampled and another disjoint subset of images is undersampled, which leads to an unreliable evaluation pipeline. Additionally, quantizing the sampled z values results in clipping of very small and large values, which increases the probability of labels at the edges (the smallest and largest label) being sampled and thus also contributes to the sampling imbalance.

To fix this, we change the modelled distribution of z to a uniform distribution. Since the distribution of z depends on its variance σ_z^2 and mean μ_z, and the distribution of z is given by our desired uniform distribution, we can first sample z and then calculate the posterior over σ_z^2 and μ_z. Using the posterior, we can then sample σ_z^2 during training. The mean μ_z is fixed to zero to simplify the setup. If we were to naively sample it, z could be rewritten as a sum of two uniform distributions $z = \mu_z + \epsilon$ with $\mu_z, \epsilon \sim U$, which would result in a trapezoidal distribution for z. On the other hand, first sampling z and then calculating the distribution for the mean and variance of the uniform distribution that z was sampled from would yield a complicated joint distribution with interactions between the mean and variance. This becomes clear when using d-seperation on the small Bayesian network depicted in Figure 2, which shows that μ_z and σ_z^2 have a converging connection to z and are therefore not independent when conditioning on z.

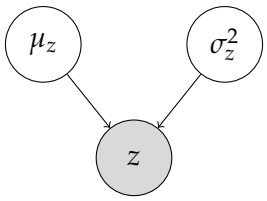


Figure 2: Bayesian network with random variables z, μ_z and σ_z^2. The variable z is shaded to indicate it is observed, while its parameters μ_z and σ_z^2 are unobserved. In our implementation, μ_z is fixed to zero, leaving only σ_z^2 unobserved.

Thus, to retrieve the desired uniform distribution of z, we can fix its mean to zero and sample its variance from a non-standard posterior distribution (which is a truncated Beta distribution in our implementation). Intuitively, if the sampled z has a small absolute value, the variance σ_z^2 that yields the highest likelihood $P(z|\sigma_z^2)$ will also have a small value, because making the interval where the PDF f_z has support for z smaller will result in a higher probability for all z within the interval due to its uniform PDF, see Equation 4). More specifically, $P(\sigma_z^2|z)$ has support in

the interval $[\frac{z^3}{3}, \frac{1}{12}]$. This is derived in Appendix A.1.

1. **Likelihood** $P(z|\sigma_z^2)$:
We model z as uniformly distributed over $U(a_z, b_z)$ with bounds $a_z = -0.5, b_z = 0.5$ and probability density function

$$f_z(a_z, b_z) = \begin{cases} \frac{1}{b_z - a_z} & \text{if } z \in [a_z, b_z] \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

In other words the likelihood of observing a specific z conditioned on the variance random variable σ_z^2 is modelled as a uniform distribution. The bounds of the uniform distribution can be calculated from the variance (this is derived in Appendix A.1):

$$b_z = -a_z = \sqrt{3\sigma_z^2} \tag{5}$$

It follows that $b_z - a_z = 2\sqrt{3\sigma_z^2}$. Thus, the likelihood $P(z|\sigma_z^2)$ is:

$$P(z|\sigma_z^2) = \begin{cases} \frac{1}{2\sqrt{3\sigma_z^2}} & \text{if } |z| \leq \frac{2\sqrt{3\sigma_z^2}}{2} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

Examples for the likelihood distribution with two different values for the observed variable σ_z^2 are depicted in Figure 3.

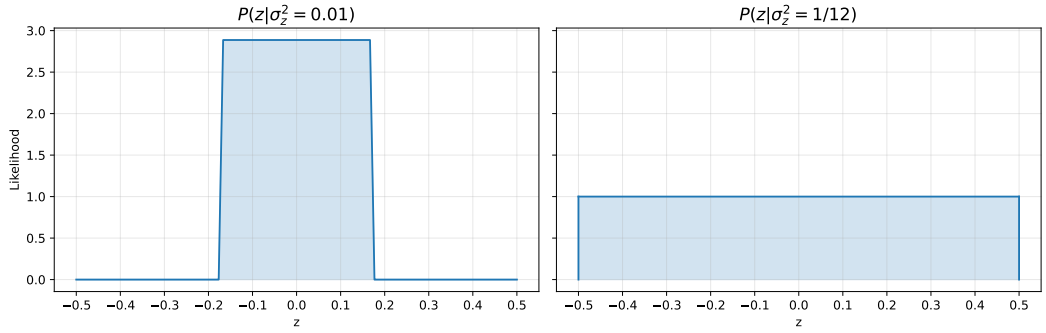


Figure 3: Likelihood of z given two different values of σ_z^2.

2. **Posterior** $P(\sigma_z^2|z)$:
Using Bayes' theorem, we compute the posterior distribution $P(\sigma_z^2|z)$ by combining the likelihood $P(z|\sigma_z^2)$ with the prior distribution on σ_z^2:

$$P(\sigma_z^2|z) \propto P(z|\sigma_z^2) \cdot P(\sigma_z^2)$$

The prior $P(\sigma_z^2)$ is a uniform distribution over the interval $[\epsilon, \frac{1}{12}]$ with $0 < \epsilon <= \frac{1}{12}$. We select $\epsilon = 0.01$. Thus, the unnormalized posterior is:

$$P(\sigma_z^2|z) \propto \underbrace{\frac{1}{2\sqrt{3\sigma_z^2}} \cdot \mathbf{1}_{|z|\leq\sqrt{3\sigma_z^2}}}_{\text{Likelihood } P(z|\sigma_z^2)} \cdot \underbrace{\frac{1}{\frac{1}{12}-\epsilon} \cdot \mathbf{1}_{\sigma_z^2\in[\epsilon,\frac{1}{12}]}}_{\text{Prior } P(\sigma_z^2)} \tag{7}$$

The posterior is then normalized over all values of σ_z^2. Two examples for the resulting posterior distribution can be found in Figure 4. Using this distribution, we sample $\sigma_z^2 \sim P(\sigma_z^2|z)$ during training and give it to the models' prior-encoder as auxiliary information about the latent factors.

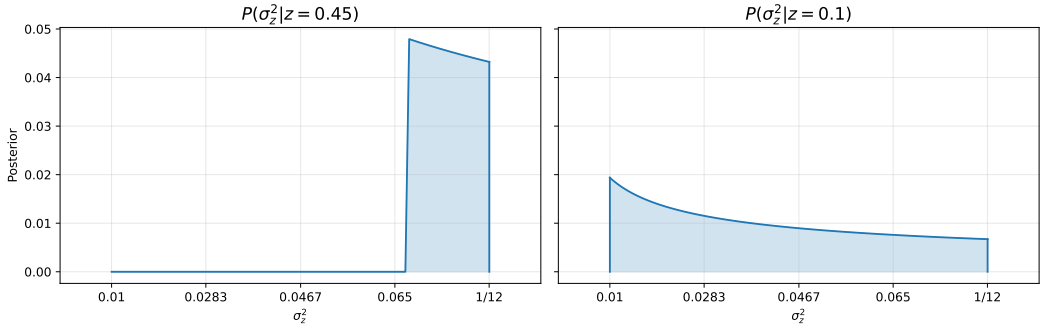


Figure 4: Posterior over σ_z^2 given two different values of z.

In summary, our fixed sampling process first samples the ground truth factors from a uniform distribution with mean zero and fixed variance. Then a posterior over possible variances that could have generated the uniform distribution from which z was sampled is calculated. One variance is sampled from it (using a discrete approximation with 100 bins) and used as additional input to the iVAE. This framework allows us to match the datasets uniform distribution of ground truth factors while keeping the model input non-deterministic.

2.5. Classifier-Free Guidance

To guide the reverse process towards a conditioning z, we make use of classifier-free guidance [11]. This lets us reconstruct images with the DDPM that are more aligned with the encoded input. During training, we train the DDPM with 20% unconditional samples (in this case z is set to all zeros), and 80% conditional samples with z being the sampled iVAE latent vector. These values were proposed in the original paper and shown to empirically perform well. Training one model with and without conditioning lets us avoid having to train two seperate models.

Afterwards, we can extract a classifier from the trained generative U-Net:

$$\nabla_x \log p_\gamma(x|z) = (1-\gamma)\nabla_x \log p(x) + \gamma \nabla_x \log p(x|z) \tag{8}$$
$$= \nabla_x \log p(x) + \gamma \left(\nabla_x \log p(x|z) - \nabla_x \log p(x)\right) \tag{9}$$

Using this, we can guide the score during the reverse process:

$$s_\phi(x_t, z, t, \gamma) = s_\phi(x_t, t) + \gamma \left(s_\phi(x_t, y, t) - s_\phi(x_t, t)\right) \tag{10}$$

There exists a tradeoff between sample diversity and guidance. However, since we can deterministically infer the image from the conditioning vector in the Shapes3D and DSprites datasets, we opt for a high guidance weight $\gamma = 4$. This results in more guidance and low sample variance, which is desirable for our use-case. The difference between sampling with and without guidance is shown in Figure 7.

2.6. Categorical iVAE

We created a Categorical-iVAE model to test whether modelling the discrete ground truth factors as discrete instead of continuous latents could help with disentanglement. This model builds upon the iVAE architecture and incorporates categorical (multinomial) distributions to sample z. To make this change, the iVAE-encoder is slightly adjusted to output logits for the categorical distributions instead of a mean and log-variance vector. Using the logits, we sample a 10-dimensional latent vector with 10 possible discrete outcomes per distribution. Thus, the predicted logits are a $10x10$ matrix. Next, we convert the logits to probabilities using the Softmax function. To avoid possible training destabilization through deterministic distributions, these probabilities are slightly adjusted towards a uniform distribution using linear interpolation. The mixup rate is 1% uniform and 99% predicted probabilites, following Hafner et al. [7].

To enable backpropagation through the sampling step, we employ a straight-through gradient estimator [3], which simply scales the gradient by the probability of each class. This is a biased estimate of the gradient, but works well in practice [7]. Furthermore, it can be easily implemented in modern Deep Learning frameworks. An example PyTorch [17] implementation is shown in Listing 1.

Listing 1: Straight-through Gradient Estimator PyTorch Implementation

```
# Create and sample from the probability distribution
categorical = torch.distributions.OneHotCategorical(probs=probs)
sample = categorical.sample()

# Compute the straight-through gradient estimator
grad = categorical.probs - categorical.probs.detach()
sample = sample + grad  # sample now has the gradient of probs
```

Another advantage is that it does not have any tunable parameters in comparison to the prevalent alternative for backpropagation through categorical sampling, the Gumbel-Softmax trick [13].

Regularization is handled similarly to the iVAE by using the Kullback-Leibler (KL) Divergence. More specifically, the probabilities for the prior categorical distributions are generated by the prior-encoder, which takes auxiliary information $u = [\mu_z, \sigma_z^2]$ as input. These probabilities are used to regularize the encoders' categorical distribution to penalize deviations from the prior. As a result, the regularized latent space is encouraged towards disentanglement of the latent representations.

Finally, the loss function used to train the model combines the reconstruction loss and the KL divergence as a weighted sum.

2.7. KL Balancing

In our model, we regularize the latent representations by aligning them with the conditioned prior, but a standard KL divergence minimization might not be the optimal loss formulation due to its asymmetry, which could lead to over-regularization towards the prior. To address this, we implement KL balancing, as proposed by Hafner et al. [7]. We minimize both directions $KL\left(q_\theta(z|x)||p_\psi(z)\right)$ and $KL\left(p_\psi(z)||q_\theta(z|x)\right)$ using different weight coefficients in the weighted loss. This enables us to balance how much the latent representations are regularized towards the prior and how predictable the prior is. Therefore, we might be able to prioritize learning a better prior instead of over-regularizing the latent representations towards an undertrained prior.

3. Evaluation

3.1. Prototyping Dataset

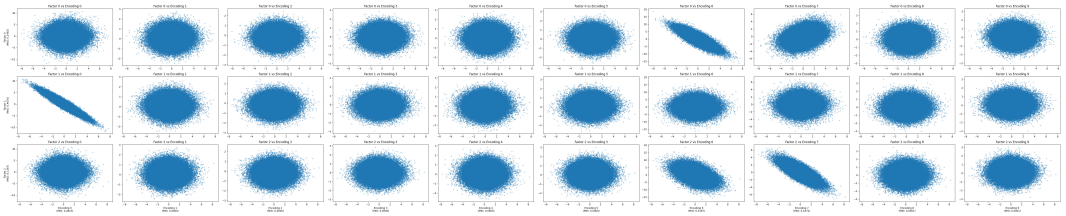
Data Generation We generate synthetic data by sampling latent factors z from a uniform distribution over $[0, 1]$. To generate the input data x, we apply a randomly initialized rotation matrix on the factors. The rotation matrix provides a linear transformation that is simple to learn and thus useful for quickly validating whether the model can capture the linear dependency.

More specifically, we sample the rotation matrix from the special orthogonal group $SO(3)$, which is the group of all orthorgonal matrices with a determinant of one. A matrix in this group can be used to rotate a vector around an axis that passes through the origin.

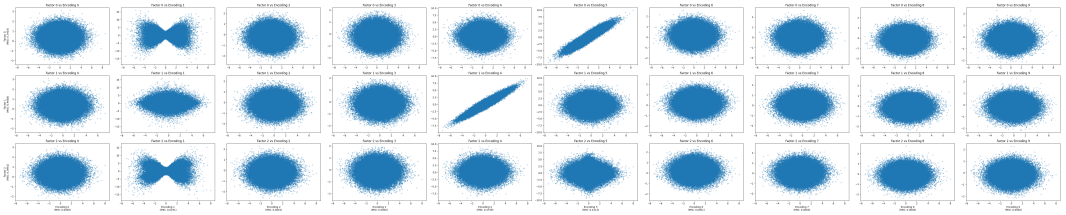
Evaluation To evaluate the performance of our model, we use the discrete MIG and MCC Pearson metrics to assess how well the learned factors align with the ground truth factors. Figure 5 shows scatter plots of the learned factors versus

the ground truth factors. These plots illustrate how well each pair of a factor and encoding dimension aligns. This visualizes the model's ability to capture the underlying structure and to disentangle its latent space.

A perfectly disentangled plot such as the example illustrated in Figure 5a has one scatter-plot with all points on a straight line per factor. In this case, all other subplots would show gaussian distributions, which indicate that a factor is not encoded in this latent dimension. In other words, this pair has zero mutual information. However, if multiple subplots show a correlation between a ground truth factor and the respective latent dimension, the model's representation would be entangled. An example for this case is depicted in Figure 5b. Additionally, the MIG and MCC Pearson values overlaid on the figure indicates the degree of disentanglement achieved by the model.



(a) Scatter plots showing the relationship between *disentangled* learned encodings and ground truth factors.



(b) Scatter plots showing the relationship between *entangled* learned encodings and ground truth factors.

Figure 5: Comparison of scatter plots for disentangled (a) and entangled (b) encodings, with MIG values indicating the degree of disentanglement.

In summary, the synthetic dataset and rotation matrix approach enable controlled experimentation and validation of our sampling strategy, while the evaluation plots provide a clear visualization of the captured variation and degree of disentanglement.

3.2. Dataset

We use the Shapes3D dataset [4] for many experiments and results. This dataset consists of 480,000 RGB images with a size of 64 x 64 pixels. They are images of

various objects that stand on a floor and before a background. Each image contains six generative factors. These describe the image's floor color, background color, and background orientation as well as the object's color, scale/size, and shape (see Figure 8). Some of the generative factors are of categorical nature, for example, there are four kinds of shapes (see Figure 9). Other factors, such as the color of the floor, object, and background are categorical because of the dataset's finite size but can also be interpreted as continuous dimensions.
Furthermore, we use the DSprites dataset [16]. There are 737,280 black and white images with a size of 64 x 64 pixels in the dataset. It has six generative factors, namely the color, shape, scale, orientation, x-position and y-position. They are 2d images of a white shape on a black background.

3.3. Experiments

We conduct a range of quantitative and qualitative experiments, including hyperparameter tuning through grid search and Bayesian optimization (see Table 1). Additionally, we assess the replicability of the best-performing hyperparameters by training models with different seeds. Our experiments also involve fixing the weights of the iVAE and using the pretrained weights of a bad-performing iVAE and our best iVAE to test whether the conditioning of the DDPM on the iVAE latent vector z is working as expected. Lastly, we qualitatively validate that the generations of the iVAE decoder align with the generations of the DDPM.

Using the described hyperparameter tuning protocols, we first rank trained models based on the discrete MIG score (1) and qualitatively evaluate the disentanglement of image generations by the iVAE decoder and DDPM when interpolating each dimension of the iVAE's latent space. Following this procedure, the best hyperparameters for the loss coefficients are $\lambda_{\text{KL}} = 0.0001$ and $\lambda_{\text{rec}} = 0.1$. Additionally, not changing the mean of the conditioning vector u yields better results.

Tuned Parameters		
Parameter	**Search space**	**Best value**
λ_{KL}	[1e-5, 0.1] (log uniform)	1e-4
λ_{rec}	[0.01, 5] (uniform)	0.1
dataset_modulate_mean	{true, false}	false

Fixed Parameters	
Parameter	**Value**
batch size	32
learning rate	2e-4
number of epochs	10

Table 1: Parameters and search space used for hyperparameter tuning for the models.

3.4. Results

To test our hypothesis that integrating the diffusion process improves the disentanglement of latent representations, we conduct a comprehensive empirical study (see Table 2) that demonstrates our method's ability to learn disentangled representations, using various disentanglement metrics on the Shapes3D dataset [4]. Additionally, we validate the results by comparing our model's qualitative generations (see Figures 10 to 12) to baselines (see Figures 13 and 14) when interpolating individual dimensions of the latent space.

The experimental results indicate that our approach can, in certain instances, outperform or match traditional VAEs and iVAEs in terms of disentanglement, as seen in the comparative metrics over the hyperparameter sweep, which is provided in Table 2. This is particularly notable in the context of the Spearman Mean Correlation Coefficient (MCC) with a mean of 0.23 and a maximum of 0.51 and Disentanglement metric with a mean of 0.13 and a maximum of 0.39, where our iVAE+DDPM framework demonstrates the ability to separate the latent factors associated with the different generative aspects of the dataset better than the other models.

Despite the VAE and iVAE sometimes showing superior performance to the iVAE+DDPM in the quantitative results, the addition of the DDPM has enhanced the model's ability to generate more nuanced and detailed interpolation. This synergy between the iVAE's structured latent space and the DDPM's generative capabilities has led to more refined and coherent image generations, as showcased in Figure 10 in comparison to Figures 13 and 14.

Additionally, from the interpolations and latent representations of our best-performing model with a CNN backbone (see Figures 10 to 12) we observe that it almost perfectly learns three latent variables, namely the object color, the background color as well as the floor color. We can also see that the model encodes the object shape which it does not learn as well as the other factors, as can be seen in the metrics with a rather low Discrete MIG score of only 0.059 in comparison to the other factors (see Dim 1 in Figure 12), and also visually in the interpolations in Figure 10. We therefore conclude that learning the color is much easier for this model than the other latent variables. In particular, it missed two factors, the object size, and the background orientation completely. Lastly, when looking at Figure 12, one can count the modes that indicate how many discrete values of the different latent variables the model has learned. From the visual inspection, we infer that the model probably did not find all of the distinct values for every factor.

Another important finding from the sweep over multiple random seeds to test whether the best-performing hyperparameters of the iVAE+DDPM model with a CNN backbone can consistently achieve the same performance (see Table 9) is that the iVAE+DDPM does not reach the same results on average, indicating that the

proposed model is not very robust yet.

When using and fixing the pretrained weights of the iVAE of the best- and a bad-performing iVAE+DDPM model with a CNN backbone to train new models, we can observe that the interpolations of the newly trained models are of the same quality. Which in return means that the conditioning of the DDPM on the latent vector z of the iVAE is working. This is also further solidified by qualitatively comparing the interpolations of the iVAE and DDPM which match for good models (e.g. compare Figure 10 and Figure 11).

Metrics	**VAE**	**iVAE**	**iVAE + DDPM**
Discrete MIG	$0.06 \pm 0.04/0.14$	$\mathbf{0.19 \pm 0.07}/0.34$	$0.16 \pm 0.10/\mathbf{0.39}$
Spearman MCC	$0.20 \pm 0.13/0.49$	$0.19 \pm 0.09/0.39$	$\mathbf{0.23 \pm 0.14/0.51}$
Disentanglement	$0.09 \pm 0.09/0.26$	$0.11 \pm 0.10/0.30$	$\mathbf{0.13 \pm 0.11/0.39}$
Modularity	$\mathbf{0.51 \pm 0.25/0.87}$	$0.41 \pm 0.12/0.63$	$0.45 \pm 0.23/0.77$

Table 2: Disentanglemnt metrics on the Shapes3D dataset showing the mean, standard deviation, and max per metric (in that order) for a hyperparameter grid search to compare the robustness across hyperparameters of our model with the VAE and iVAE baselines.

Additionally, we explore the impact of different architectural choices, sampling methods, and Kullback-Leibler (KL) divergence balancing on the disentanglement performance.

Firstly, we conduct a hyperparameter grid search over the VAE, iVAE and iVAE + DDPM models with a ResNet-50 as its backbone. This can be seen in Table 3. The results show that using a ResNet-50 instead of a CNN for the encoder and decoder increases all metrics for disentanglement substantially, doubling or even tripling them in some cases compared to the results reported in Table 2. Besides the discrete MIG where the iVAE shows a much higher mean than the other model types with 0.42, the experiment shows similar results between the different model types. Qualitatively, the interpolations (seen in Figure 17) created show a clear disentangled representation for two generative factors, namely the background rotation in dimension 6 and the object's size in dimension 9. We can also see that in dimensions 1 and 2 the shape of the object is encoded. The discrete shapes can also be seen nicely in the distribution plot of the z vector in Figure 15 where we can see 3 definite peaks in dimension 1 and also 3 in dimension 2. This means shapes are encoded more than once as there a only 4 discrete shapes in total (as can be seen from Figure 9). In addition, we observe from the comparison of Figure 17 and Figure 16 that the DDPM correctly is conditioned on the iVAE as it also changes the background rotation in its 6^{th} dimension, the object's size in dimension 9 and the object's shape in the dimension 1 and 2. Lastly, from the t-SNE dimensionality

reduction of the encoded ground truth data (seen in Figure 6), one can discern semantically meaningful spatial ordering, for example, the images with the same background color are close together in the plot. This means the model learns useful embeddings for the task of disentangled representation, as it learns to distinguish between different discrete classes of the generative factors.

Metrics	**VAE**	**iVAE**	**iVAE + DDPM**
Discrete MIG	$0.32 \pm 0.14/0.47$	$\mathbf{0.42} \pm 0.14/\mathbf{0.68}$	$0.31 \pm 0.10/0.49$
Spearman MCC	$\mathbf{0.70} \pm 0.13/0.83$	$\mathbf{0.70} \pm 0.11/\mathbf{0.85}$	$0.68 \pm 0.08/0.78$
Disentanglement	$\mathbf{0.75} \pm 0.15/\mathbf{0.99}$	$0.74 \pm 0.13/0.94$	$0.68 \pm 0.09/0.80$
Modularity	$\mathbf{0.97} \pm 0.02/\mathbf{0.99}$	$0.94 \pm 0.04/0.98$	$0.95 \pm 0.02/0.98$

Table 3: Disentanglement metrics on the Shapes3D dataset showing the mean, standard deviation, and max per metric (in that order) for a hyperparameter grid search to compare the robustness across hyperparameters of our model with the VAE and iVAE baselines. These results are with a ResNet-50 as encoder and decoder (as described in 2.3).

Secondly, we compare a ResNet-18 to a ResNet-50 as architectures for the encoder and decoder for an iVAE in Table 4. One can see very close results for both model architectures. Only the discrete MIG metric again shows a very high mean of 0.42 for an iVAE with a ResNet-50 compared to the mean of an iVAE with a ResNet-18 of 0.33. All other metrics have higher means and max values with the ResNet-18, but only by a small margin. The reason for the high mean in the discrete MIG metric is not clear to us and we expect it to be the cause of good initialization for the runs and not a clear improvement caused by the ResNet-50. Therefore, in further experiments, we use a ResNet-18 as encoder and decoder, as the runtime reduces from around 60 to 17.5 hours.

Metrics	**iVAE ResNet-18**	**iVAE ResNet-50**
Discrete MIG	$0.33 \pm 0.09/0.56$	$\mathbf{0.42} \pm 0.14/\mathbf{0.68}$
Spearman MCC	$\mathbf{0.71} \pm 0.09/\mathbf{0.87}$	$0.70 \pm 0.11/0.85$
Disentanglement	$\mathbf{0.77} \pm 0.15/\mathbf{1.00}$	$0.74 \pm 0.13/0.94$
Modularity	$\mathbf{0.95} \pm 0.02/\mathbf{1.00}$	$0.94 \pm 0.04/0.98$

Table 4: Disentanglement metrics on the Shapes3D dataset showing the mean, standard deviation, and max per metric (in that order) for a hyperparameter grid search to compare the robustness across hyperparameters of the different ResNet model sizes for the encoder and decoder (as described in 2.3).

We also compare the sampling with a normal distribution assumption to the sampling with the uniform distribution assumption as described in 2.4 in Table 5. Here we see a definite improvement of the uniform assumption sampling over the normal assumption sampling for an iVAE with a CNN encoder and decoder for nearly all metrics. The discrete MIG mean improves from 0.09 to 0.26 with the max value jumping from 0.13 to 0.43. Also, the Spearman MCC mean increases by 0.05 points, as well as the max value by 0.03. Another small improvement can be seen in the disentanglement metric. It increases in its mean from 0.11 to 0.15. Only the mean of the modularity metric decreases from 0.40 to 0.36.

Metrics	**Normal Sampling**	**Uniform Sampling**
Discrete MIG	$0.09 \pm 0.03/0.13$	$\mathbf{0.26} \pm 0.12/\mathbf{0.43}$
Spearman MCC	$0.22 \pm 0.06/0.35$	$\mathbf{0.27} \pm 0.07/\mathbf{0.38}$
Disentanglement	$0.11 \pm 0.10/0.26$	$\mathbf{0.15} \pm 0.09/\mathbf{0.29}$
Modularity	$\mathbf{0.40} \pm 0.12/\mathbf{0.63}$	$0.36 \pm 0.07/0.49$

Table 5: Disentanglement metrics on the Shapes3D dataset showing the mean, standard deviation, and max per metric (in that order) for a hyperparameter grid search to compare the robustness across hyperparameters of an iVAE model with sampling based on the assumption of a normal distribution and a uniform distribution (as described in 2.4). These results are with a CNN as encoder and decoder (as described in 2.3).

We train several Categorical VAE models (as described in 2.6) in a grid search over its hyperparameters and compare the results to an iVAE in Table 6. They both have a ResNet-18 ass encoder and decoder. In all but one metric we can see worse results by the Categorical VAE in relation to the iVAE. Only the modularity score can match the iVAE's in the mean.

Metrics	**CatVAE**	**iVAE**
Discrete MIG	$0.09 \pm 0.04/0.15$	$\mathbf{0.33} \pm 0.09/\mathbf{0.56}$
Spearman MCC	$0.42 \pm 0.07/0.59$	$\mathbf{0.71} \pm 0.09/\mathbf{0.87}$
Disentanglement	$0.38 \pm 0.11/0.56$	$\mathbf{0.77} \pm 0.15/\mathbf{1.00}$
Modularity	$\mathbf{0.95} \pm 0.02/0.98$	$\mathbf{0.95} \pm 0.02/\mathbf{1.00}$

Table 6: Disentanglement metrics on the Shapes3D dataset showing the mean, standard deviation, and max per metric (in that order) for a hyperparameter grid search to compare the robustness across hyperparameters of the CatVAE model. These results are with a ResNet-18 as encoder and decoder (as described in 2.3).

Next, to estimate the implementation of kl balancing (as described in 2.7), we run an experiment over combinations of posterior-prior and prior-posterior weights. This means we extend the kl loss to calculate the kl divergence in both possible directions and afterwards weigh them based on a predefined factor. Over this factor, we create a hyperparameter sweep in which each combination of weights is tested for the values $[0.005, 0.0025, 5e^{-4}, 2.5e^{-4}, 5e^{-5}, 2.5e^{-5}]$. The results can be seen in Table 7, where we show the value range for the posterior-prior weights and report the mean over the same value range for the prior-posterior weights. One can see that the posterior-prior weights $2.5e^{-4}$, $5e^{-5}$ and $2.5e^{-5}$ show the best results. In comparison to no kl balancing in Table 2, one can see slight improvements in the Spearman MCC metric as well as in the disentanglement metric.

kl	Discrete MIG	Spearman MCC	Disentanglement	Modularity
0.005	$0.18 \pm 0.09/0.27$	$0.23 \pm 0.07/0.36$	$0.12 \pm 0.11/0.32$	$0.33 \pm 0.17/0.65$
0.0025	$0.11 \pm 0.14/0.29$	$0.15 \pm 0.16/0.33$	$0.10 \pm 0.11/0.22$	$0.28 \pm 0.21/0.59$
$5e^{-4}$	$0.18 \pm 0.10/0.28$	$0.21 \pm 0.10/0.29$	$0.14 \pm 0.09/0.23$	$0.37 \pm 0.14/0.58$
$2.5e^{-4}$	$\mathbf{0.21} \pm 0.12/\mathbf{0.33}$	$0.21 \pm 0.08/0.32$	$0.16 \pm 0.09/0.24$	$0.42 \pm 0.05/0.49$
$5e^{-5}$	$0.17 \pm 0.08/0.28$	$0.28 \pm 0.09/0.38$	$\mathbf{0.21} \pm 0.15/\mathbf{0.42}$	$0.43 \pm 0.18/0.59$
$2.5e^{-5}$	$0.19 \pm 0.07/0.25$	$\mathbf{0.30} \pm 0.10/\mathbf{0.40}$	$0.20 \pm 0.02/0.23$	$\mathbf{0.51} \pm 0.11/\mathbf{0.67}$

Table 7: Disentanglement metrics on the Shapes3D dataset showing a hyperparameter grid search to compare the robustness across hyperparameters for Kullback-Leibler divergence balancing where we change the weight of the divergences of the posterior-prior and prior-posterior directions (as described in 2.7). The "kl" column in the table describes the different values used for the weights of the posterior-prior direction. The results show the mean, std over the same value range for the prior-posterior coefficient. The results are based on an iVAE model with sampling based on the assumption of a uniform distribution (as described in 2.4). These results are with a CNN as encoder and decoder (as described in 2.3).

Moreover, we qualitatively assess the effects of classifier-free guidance (as described in Section 2.5) on the interpolations of the DDPM. We can see more similar interpolations to the ground truth when encoding a ground truth image with the encoder and then creating the image from these encodings when using guidance. One can see that for example in the last row of Figure 7 where the "DDPM with Guidance" correctly creates an image with the object's color being pink rather than the "DDPM without Guidance" which generates an image where the object is red.

Lastly, we also tried the same framework on another dataset in particular the DSprites dataset [16]. Here we observe similar mean values for the discrete MIG of 0.31 and the Modularity score of 0.92 by comparing Table 8 and the ResNet-18

Metrics	iVAE + DDPM
Discrete MIG	$0.31 \pm 0.04/0.35$
Spearman MCC	$0.64 \pm 0.04/0.67$
Disentanglement	$0.63 \pm 0.02/0.66$
Modularity	$0.92 \pm 0.04/0.95$

Table 8: Disentanglement metrics on the DSprites dataset showing the mean, standard deviation, and max per metric (in that order) for a hyperparameter grid search to see the robustness across hyperparameters of an iVAE + DDPM model with sampling-based on the assumption of a uniform distribution (as described in 2.4). These results are with a ResNet-18 as encoder and decoder (as described in 2.3).

values of Table 4. For the DSprites dataset, we see lower Speaman MCC values and Disentanglement values. The divergence is significant, but would still mean that one can see clear disentanglement in the interpolations of the Shapes3D dataset. This is not the case for the DSprites dataset, as can be seen in Figure 18. There is no clear disentanglement between generative factors one can observe in the created images of the DDPM. Therefore, we derive that our model architecture is still not robust enough to generalize to other data when trained on this data with the found best hyperparameters of the Shapes3D dataset.

4. Limitations and Future Work

There are multiple promising directions left open that can improve the performance of our proposed approach. To facilitate future work on this project, we lay out the shortcomings of our model and propose a few possible ways of addressing them.

Time-Step Sampling Strategies for DDPM Training In our experiments, setting the iVAE-decoder reconstruction loss coefficient to zero resulted in poor performance, suggesting that the DDPM alone is insufficient for training the iVAE-encoder. Our hypothesis is that the DDPM loses sensitivity to conditioning at later time steps, as the input image at these stages has the same amount of information as the conditioned vector once the rough features are learned. This issue is especially apparent in the simple Shapes3D dataset. To investigate this, future work could focus on training the DDPM more intensively on earlier time steps, where the conditioning might have more impact, i.e. by using a Beta-distribution over time steps instead of the typically applied uniform distribution. This revised sampling strategy could enable the model to better utilize the conditioning information and thus have better gradients that train the iVAE encoder.

Application to Datasets Requiring Fine-Grained Conditioning Another research direction related to the first point is the application of our model to new datasets

that require more complicated or fine-grained conditioning. This would help with the evaluation of the hypothesis that the DDPMs is not sensitive to the conditioning vector if the noisy input image encodes very similar information. If this hypothesis is true, a more complicated dataset would result in better metrics when incorporating the DDPM relative to just using the iVAE.

Explainable AI (XAI) Methods A third possible way of investigating the insensitivity hypothesis is to employ XAI methods to visualize or quantify the sensitivity on the conditioning vector. The outcomes of this would be helpful for informing further model design choices.

Alternative Conditioning Methods While our current DDPM has a very simple additive conditioning on latent vectors from the iVAE, future work could explore more advanced conditioning strategies. One potential approach is the use of cross-attention, which could allow the DDPM to better incorporate the conditioning information by formulating explicit queries that would be matched with key-value pairs resulting from the conditioning vector. We suspect that more explicitly aligning the generative process with the latent factors would have a positive impact on the model's conditioning sensitivity, and thus would improve the overall goal of learning more disentangled representations.

5. Conclusion

In this work, we introduced a novel framework that integrates the DDPM with an iVAE for disentangled representation learning. This framework combines the DDPM's capacity to model complex data distributions and to learn features across different scales and the iVAE's ability to achieve latent space identifiability. Through quantitative and qualitative evaluations on the Shapes3D dataset, we validated different improvements of our model, which at this stage have not surpassed the baseline methods. However, in our future work section, we identified several points that could further improve the proposed framework. The major paths include exploring more complicated datasets, as well as investigating and refining the DDPM conditioning strategy for better alignment with the encoded representation. With the improvements identified in this work and the future directions, we aim to make this framework viable for disentangled representation learning in complex domains.

References

[1] K. Abstreiter, S. Mittal, S. Bauer, B. Schölkopf, and A. Mehrjou. *Diffusion-Based Representation Learning*. 2022. arXiv: 2105.14257 [cs.LG].

[2] Y. Bengio, A. C. Courville, and P. Vincent. "Unsupervised Feature Learning and Deep Learning: A Review and New Perspectives". In: *CoRR* abs/1206.5538 (2012). arXiv: 1206.5538.

[3] Y. Bengio, N. Léonard, and A. Courville. *Estimating or Propagating Gradients Through Stochastic Neurons for Conditional Computation*. 2013. arXiv: 1308.3432 [cs.LG].

[4] C. Burgess and H. Kim. *3D Shapes Dataset*. https://github.com/google-deepmind/3d-shapes/. 2018.

[5] R. T. Q. Chen, X. Li, R. Grosse, and D. Duvenaud. *Isolating Sources of Disentanglement in Variational Autoencoders*. 2019. arXiv: 1802.04942 [cs.LG].

[6] S. Chu, D. Kim, and B. Han. *Learning Debiased and Disentangled Representations for Semantic Segmentation*. 2021. arXiv: 2111.00531 [cs.CV].

[7] D. Hafner, T. Lillicrap, M. Norouzi, and J. Ba. *Mastering Atari with Discrete World Models*. 2022. arXiv: 2010.02193 [cs.LG].

[8] K. He, X. Zhang, S. Ren, and J. Sun. "Deep Residual Learning for Image Recognition". In: *CoRR* abs/1512.03385 (2015). arXiv: 1512.03385.

[9] I. Higgins, D. Amos, D. Pfau, S. Racaniere, L. Matthey, D. Rezende, and A. Lerchner. *Towards a Definition of Disentangled Representations*. 2018. arXiv: 1812.02230 [cs.LG].

[10] J. Ho, A. Jain, and P. Abbeel. *Denoising Diffusion Probabilistic Models*. 2020. arXiv: 2006.11239 [cs.LG].

[11] J. Ho and T. Salimans. *Classifier-Free Diffusion Guidance*. 2022. arXiv: 2207.12598 [cs.LG].

[12] A. Hyvarinen, I. Khemakhem, and H. Morioka. *Nonlinear Independent Component Analysis for Principled Disentanglement in Unsupervised Deep Learning*. 2023. arXiv: 2303.16535 [cs.LG].

[13] E. Jang, S. Gu, and B. Poole. *Categorical Reparameterization with Gumbel-Softmax*. 2017. arXiv: 1611.01144 [stat.ML].

[14] J. Jia, F. He, N. Gao, X. Chen, and K. Huang. *Learning Disentangled Label Representations for Multi-label Classification*. 2022. arXiv: 2212.01461 [cs.CV].

[15] I. Khemakhem, D. P. Kingma, R. P. Monti, and A. Hyvärinen. *Variational Autoencoders and Nonlinear ICA: A Unifying Framework*. 2020. arXiv: 1907.04809 [stat.ML].

[16] L. Matthey, I. Higgins, D. Hassabis, and A. Lerchner. *dSprites: Disentanglement testing Sprites dataset*. https://github.com/google-deepmind/dsprites-dataset/. 2017.

[17] A. Paszke et al. *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. 2019. arXiv: 1912.01703 [cs.LG].

[18] O. Ronneberger, P. Fischer, and T. Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. 2015. arXiv: 1505.04597 [cs.CV].

[19] Y. Song. *Generative Modeling by Estimating Gradients of the Data Distribution*. https://yang-song.net/blog/2021/score/. May 2021.

[20] Y. Song, C. Durkan, I. Murray, and S. Ermon. *Maximum Likelihood Training of Score-Based Diffusion Models*. 2021. arXiv: 2101.09258 [stat.ML].

[21] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. "Score-Based Generative Modeling through Stochastic Differential Equations". In: *CoRR* abs/2011.13456 (2020). arXiv: 2011.13456.

[22] X. Zhu, C. Xu, and D. Tao. *Where and What? Examining Interpretable Disentangled Representations*. 2021. arXiv: 2104.05622 [cs.CV].

A. Appendix

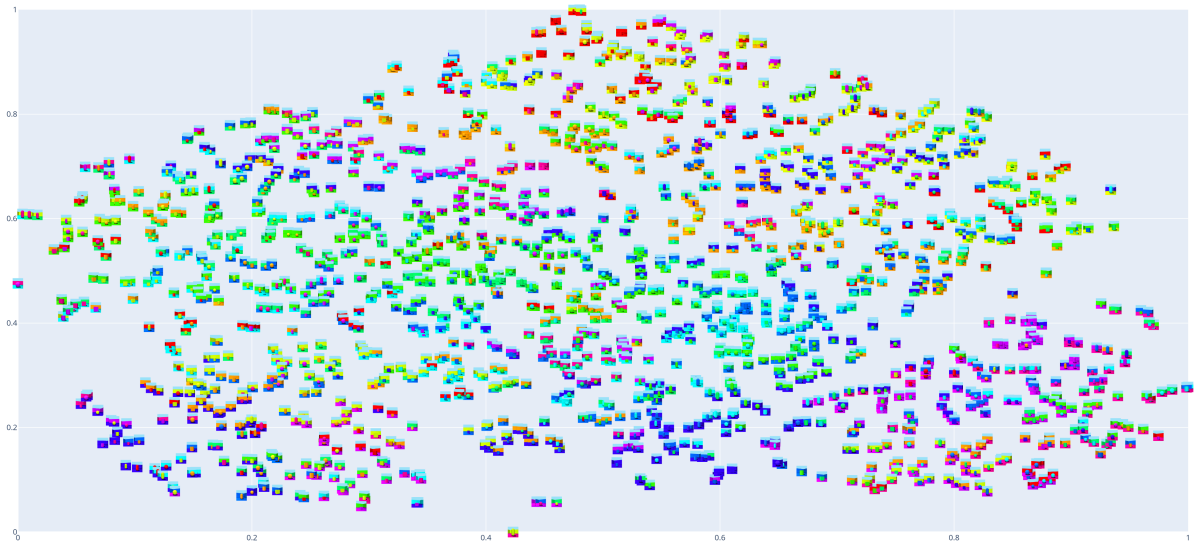


Figure 6: Scatter plot of randomly sampled images from the Shapes3D dataset. The images are clustered in a 2d space using **t-SNE dimensionality reduction on top of the iVAE-encoder embeddings**. The latent representations are able to separate different attribute values of the input images in the resulting clustering.

Metrics	**Best qualitative iVAE+DDPM**	**Different Seeds (N=5)**
Discrete MIG	0.39	$0.26 \pm 0.09/0.34$
Spearman MCC	0.21	$0.11 \pm 0.08/0.21$
Disentanglement	0.21	$0.13 \pm 0.11/0.25$
Modularity	0.39	$0.35 \pm 0.11/0.47$

Table 9: Comparison of the best-performing iVAE+DDPM model with a CNN backbone (in terms of qualitative generations) on the Shapes3D dataset and aggregate values over different random seeds using the same hyperparameters to test the model's robustness. For the sweep over random seeds, the mean, standard deviation, and maximum values are reported for each metric.

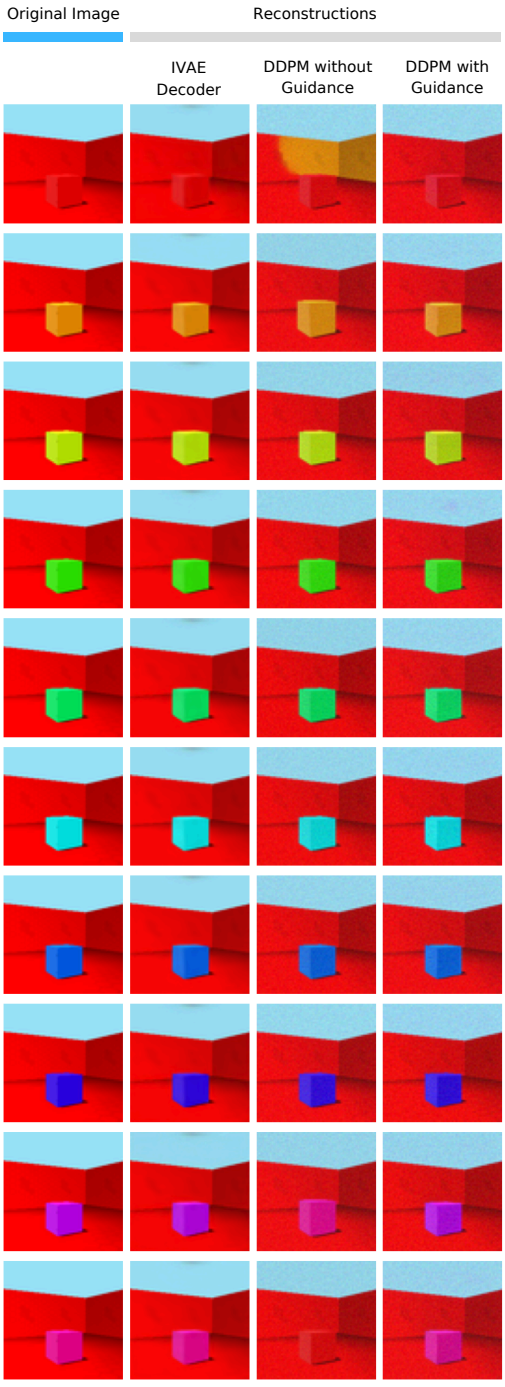


Figure 7: Comparison of the original image with reconstructions from the iVAE-decoder, DDPM without guidance and DDPM with a guidance weight of $\gamma = 4$. The guidance helps the DDPM better align with the conditioning in this case.

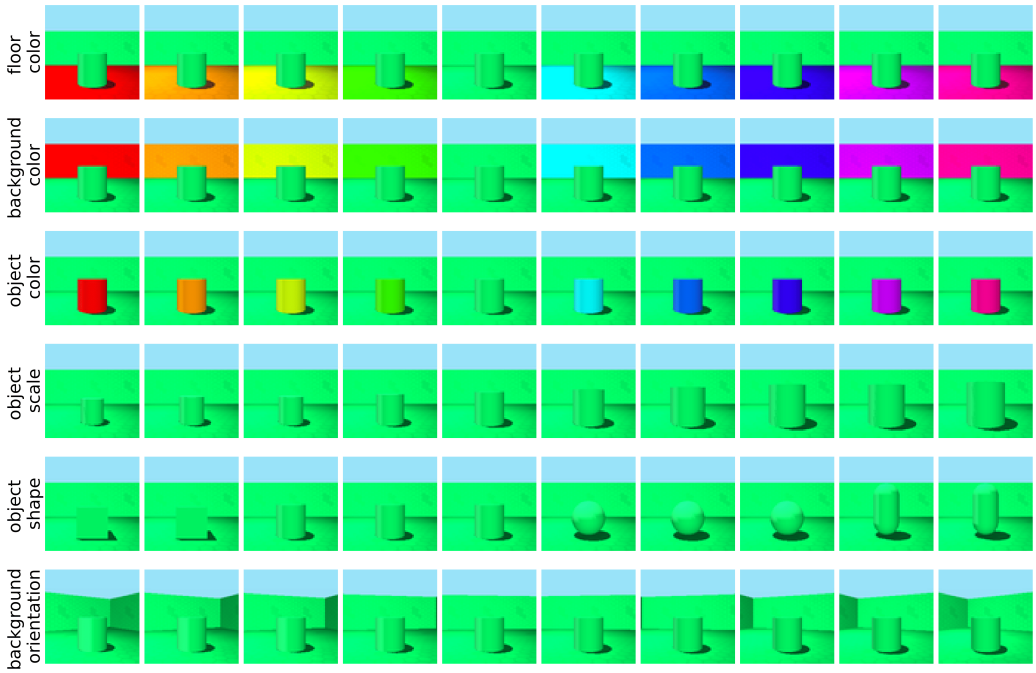


Figure 8: Example images from the Shapes3D dataset visualizing interpolation across generative factors.

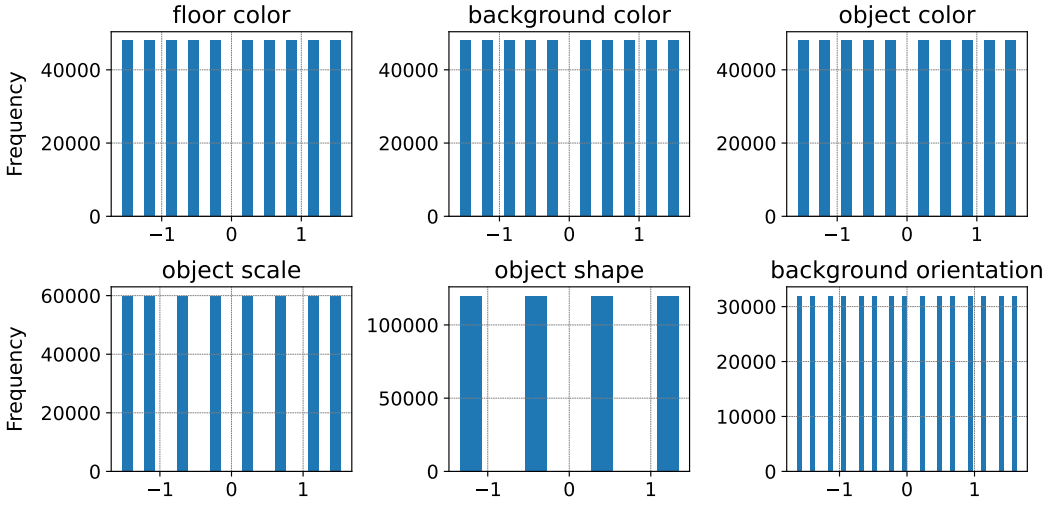


Figure 9: Distributions of the Shapes3D dataset generative factors.

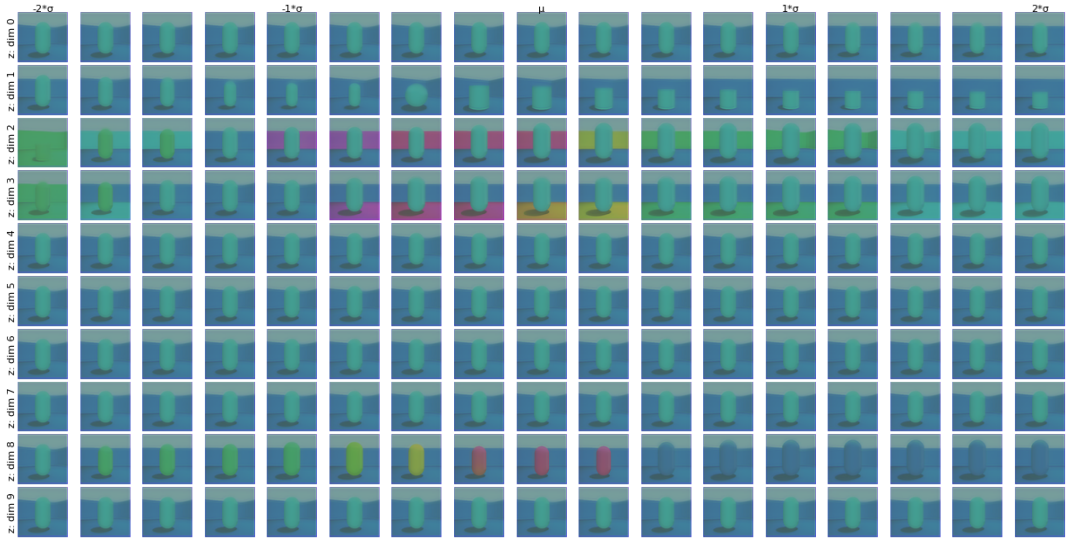


Figure 10: Conditional DDPM generations of the Shapes3D dataset (using the best iVAE+DDPM model with CNN backbone) for feature-space interpolations of one latent dimension at a time, showcasing disentangled representations.

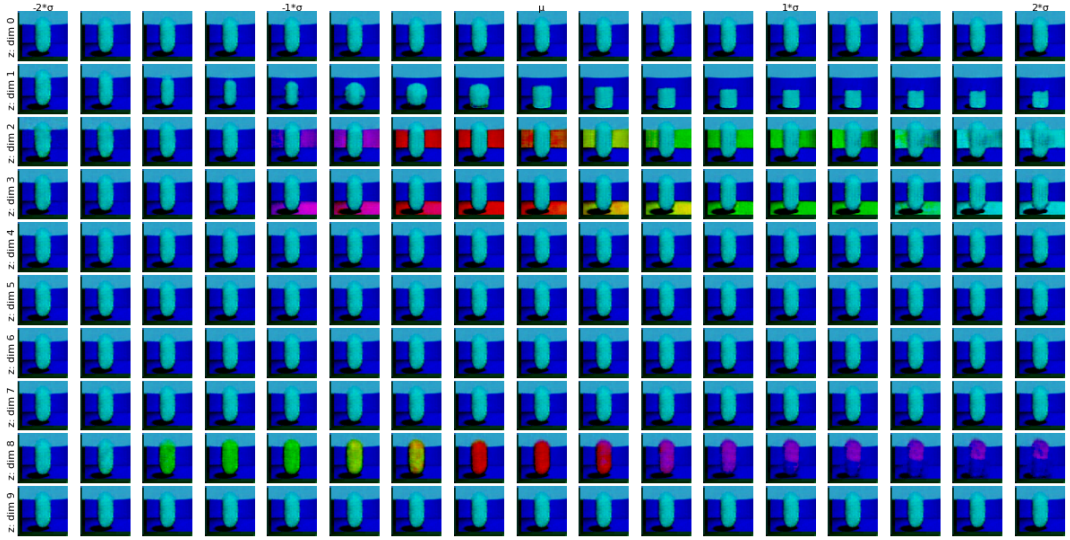


Figure 11: Conditional iVAE generations of the Shapes3D dataset (using the best iVAE+DDPM model with CNN backbone) for feature-space interpolations of one latent dimension at a time, showcasing disentangled representations.

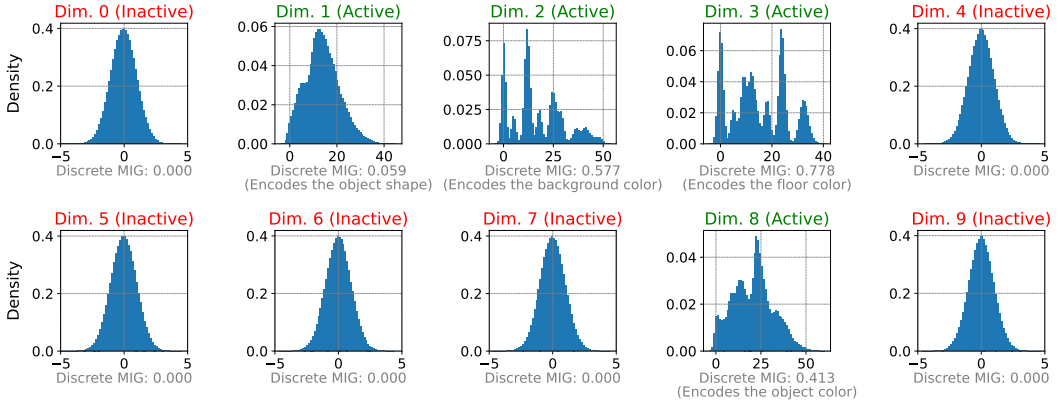


Figure 12: Distributions of the learned Shapes3D dataset z encodings obtained by passing every image in the Shapes3D dataset through the iVAE encoder with a CNN backbone and sampling a latent vector. The model has learned 4 out of 6 ground truth factors, encoded in dimensions 1,2,3, and 8. All other latent dimensions do not encode any information and thus show the normally distributed samples of z from the iVAE encoder.

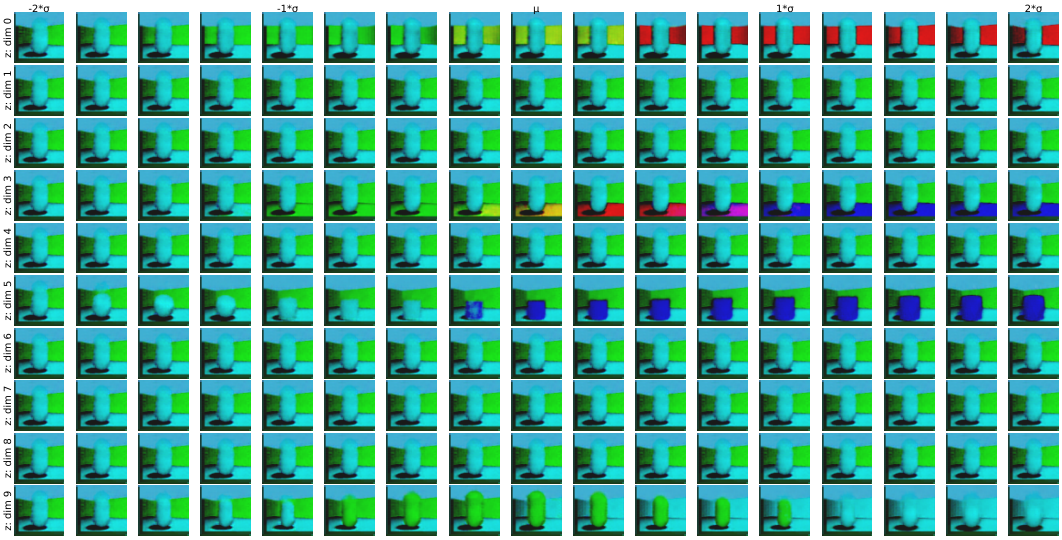


Figure 13: Conditional generations of the Shapes3D dataset using the best iVAE baseline model with a CNN backbone for feature-space interpolations of one latent dimension at a time, showcasing disentangled representations.

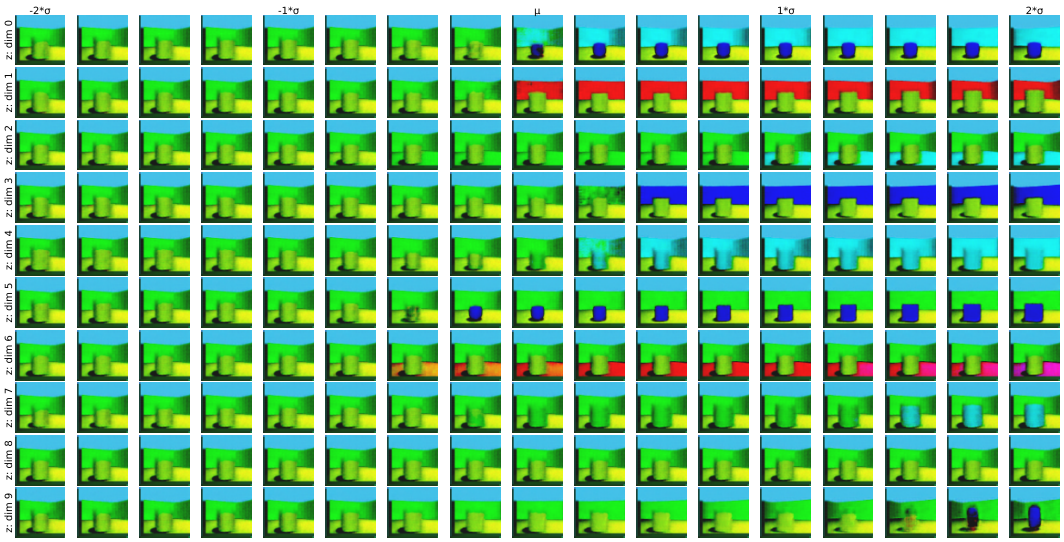


Figure 14: Conditional generations of the Shapes3D dataset using the best VAE baseline model with a CNN backbone for feature-space interpolations of one latent dimension at a time, showcasing disentangled representations.

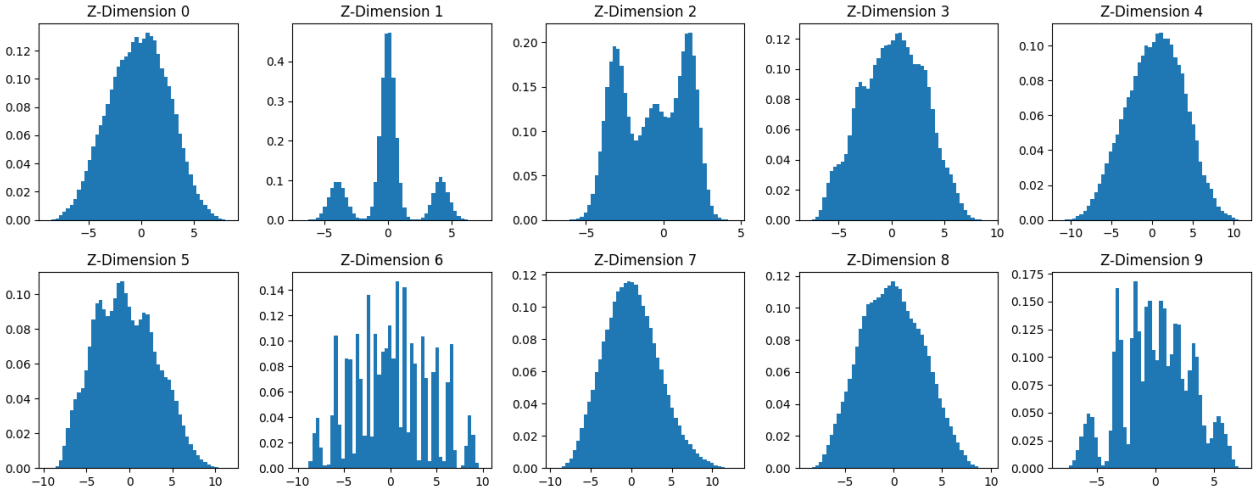


Figure 15: Distributions of the learned Shapes3D z encodings obtained by passing every image in the Shapes3D dataset through the iVAE encoder with a ResNet-50 backbone and sampling a latent vector. The model has learned some ground truth factors, especially encoded in dimensions 1,2,3,6 and 9. All other latent dimensions do not encode much information and thus show nearly normally distributed samples of z from the iVAE encoder.

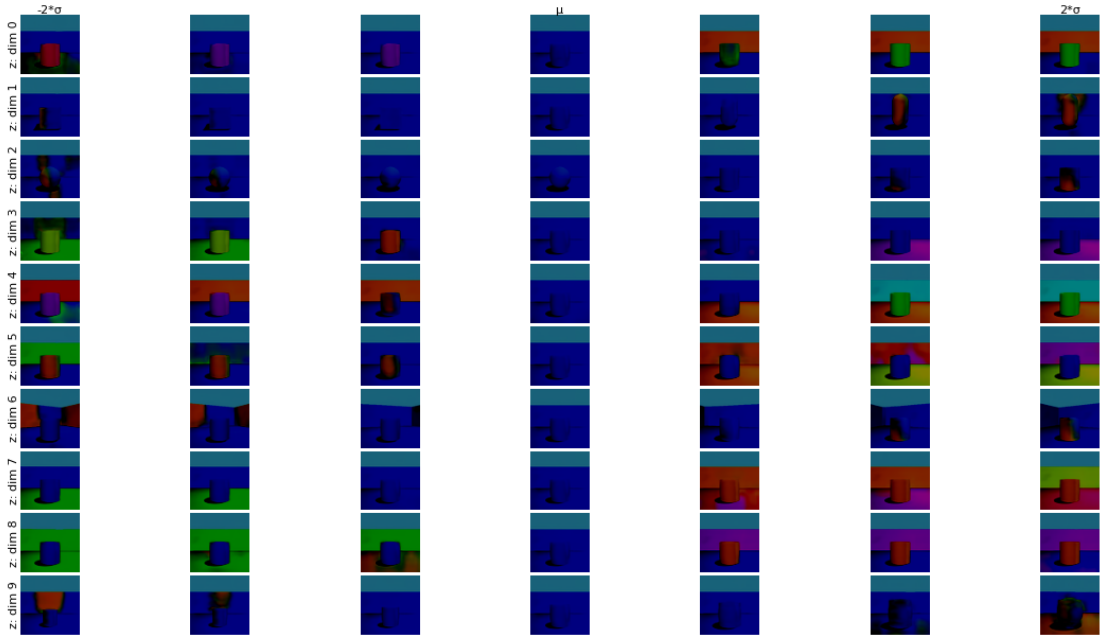


Figure 16: Conditional iVAE generations of the Shapes3D dataset (using the best iVAE+DDPM model with ResNet-50 backbone) for feature-space interpolations of one latent dimension at a time, showcasing disentangled representations.

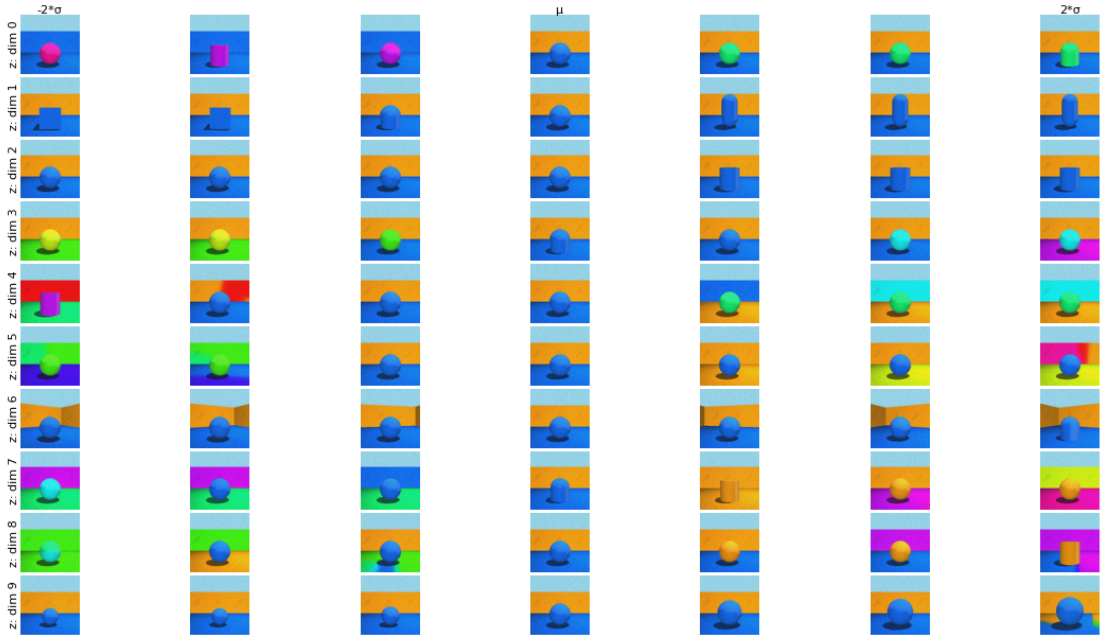


Figure 17: Conditional DDPM generations of the Shapes3D dataset (using the best iVAE+DDPM model with ResNet-50 backbone) for feature-space interpolations of one latent dimension at a time, showcasing disentangled representations.

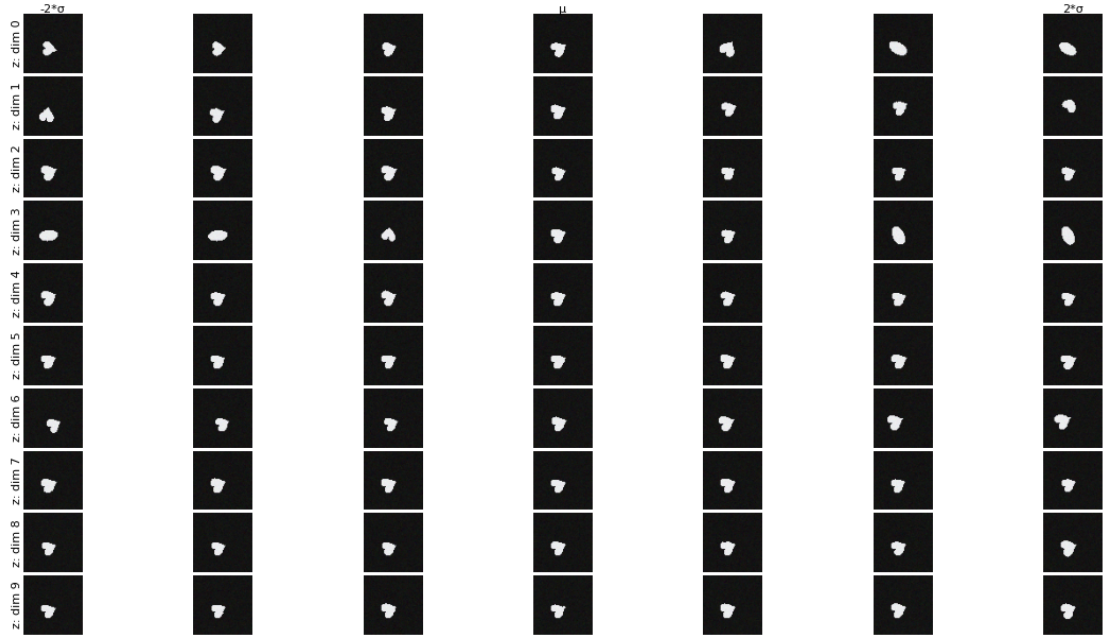


Figure 18: Conditional DDPM generations of the DSprites dataset (using the best iVAE+DDPM model with ResNet-18 backbone) for feature-space interpolations of one latent dimension at a time, showcasing disentangled representations.

A.1. Posterior Auxiliary Derivation

We derive the posterior over the variance σ_z^2 given that the factors z match the desired uniform distribution. Let $z \sim U(a_z, b_z)$. The *PDF* of z is given by:

$$f_z(a_z, b_z) = \begin{cases} \frac{1}{b_z - a_z} & \text{if } z \in [a_z, b_z] \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

and the expected value μ_z of z is:

$$\mu_z = \frac{a_z + b_z}{2}. \tag{12}$$

Assuming $\mu_z = 0$, we have a symmetric distribution with $a_z = -b_z$, which can be written in a compact form as $U(-b_z, b_z)$.

The *variance* σ_z^2 of the uniform distribution $U(a_z, b_z)$ is:

$$\sigma_z^2 = \frac{(b_z - a_z)^2}{12}. \tag{13}$$

Since $a_z = -b_z$, this simplifies to:

$$\sigma_z^2 = \frac{4b^2}{12} = \frac{b^2}{3}. \tag{14}$$

Thus, b_z can be expressed as:

$$b_z = \sqrt{3\sigma_z^2} = \sqrt{3}\sigma_z. \tag{15}$$

Support interval of posterior over σ_z^2: Using the posterior $P(\sigma_z^2|z)$, we ensure that the distribution of σ_z^2 is correctly sampled during training. In the following section, we derive the bounds of the interval where the auxiliary inputs σ_z^2 to the prior-encoder have non-zero values.

The minimum variance that has support in the posterior fulfills $|a_z| = |b_z| = |z|$, because this yields the smallest interval that still includes z. For this reason and because we are using a uniform PDF (see equation 11), this interval also has the highest posterior probability. To calculate where this point occurs, we use the fact that $|a_z| = |b_z| = |z|$:

$$\sigma_{min,z}^2 = \frac{(b_z - a_z)^2}{12} = \frac{4z^2}{12} = \frac{z^2}{3} \tag{16}$$

The maximum variance that has support is fixed to $1/12$ in our experiments, so that the uniform distribution that it creates has support in the desired interval $[-0.5, 0.5]$. We can verify this by calculating the variance for the interval bounds:

$$\sigma_{max,z}^2 = \frac{(-0.5 - (0.5))^2}{12} = \frac{1}{12} \tag{17}$$

Using the previous results, the posterior over σ_z^2 has support in $[\frac{z^2}{3}, \frac{1}{12}]$.