A Approximating the Inference Marginal

We will here derive a Monte Carlo estimator for the entropy of the marginal $q_\phi(\boldsymbol{z})$ of the inference model

$$H_\phi[\boldsymbol{z}] = -\mathbb{E}_{q_\phi(\boldsymbol{z})}\left[\log q_\phi(\boldsymbol{z})\right]. \tag{7}$$

As with other terms in the objective, we can approximate this expectation by sampling $\boldsymbol{z}^b \sim q_\phi(\boldsymbol{z})$ using,

$$\boldsymbol{x}^b \sim q(\boldsymbol{x}), \qquad b = 1, \dots, B, \tag{8}$$

$$\boldsymbol{z}^b \sim q_\phi(\boldsymbol{z} \mid \boldsymbol{x}^b). \tag{9}$$

We now additionally need to approximate the values,

$$\log q_\phi(\boldsymbol{z}^b) = \log\left[\frac{1}{N}\sum_{n=1}^{N} q_\phi(\boldsymbol{z}^b \mid \boldsymbol{x}^n)\right]. \tag{10}$$

We will do so by pulling the term for which $\boldsymbol{x}^n = \boldsymbol{x}^b$ out of the sum

$$q_\phi(\boldsymbol{z}^b) = \frac{1}{N} q_\phi(\boldsymbol{z}^b \mid \boldsymbol{x}^b) + \frac{1}{N}\sum_{\boldsymbol{x}^n \neq \boldsymbol{x}^b} q_\phi(\boldsymbol{z}^b \mid \boldsymbol{x}^n).$$

As also noted by Chen et al. [2018], the intuition behind this decomposition is that $q_\phi(\boldsymbol{z}^b \mid \boldsymbol{x}^b)$ will in general be much larger than $q_\phi(\boldsymbol{z}^b \mid \boldsymbol{x}^n)$.

We can approximate the second term using a Monte Carlo estimate from samples $\boldsymbol{x}^{(b,c)} \sim q(\boldsymbol{x} \mid \boldsymbol{x} \neq \boldsymbol{x}^b)$,

$$\frac{1}{N-1}\sum_{\boldsymbol{x}^n \neq \boldsymbol{x}^b} q_\phi(\boldsymbol{z}^b \mid \boldsymbol{x}^n) \simeq \frac{1}{C}\sum_{c=1}^{C} q_\phi(\boldsymbol{z}^b \mid \boldsymbol{x}^{(b,c)}).$$

Note here that we have written $1/(N-1)$ instead of $1/N$ in order to ensure that the sum defines an expected value over the distribution $q(\boldsymbol{x} \mid \boldsymbol{x} \neq \boldsymbol{x}^b)$.

In practice, we can replace the samples $\boldsymbol{x}^{(b,c)}$ with the samples $b' \neq b$ from the original batch, which yields an estimator over $C = B - 1$ samples

$$\hat{q}(\boldsymbol{z}^b) = \frac{1}{N} q_\phi(\boldsymbol{z}^b \mid \boldsymbol{x}^b) + \frac{N-1}{N(B-1)}\sum_{b' \neq b} q_\phi(\boldsymbol{z}^b \mid \boldsymbol{x}^{b'}).$$

Note that this estimator is unbiased, which is to say that

$$\mathbb{E}[\hat{q}(\boldsymbol{z}^b)] = q(\boldsymbol{z}^b). \tag{11}$$

In order to compute the entropy, we now define an estimator $\hat{H}_\phi(\boldsymbol{z})$, which defines a upper bound on $H_\phi(\boldsymbol{z})$

$$\hat{H}_\phi[\boldsymbol{z}] \simeq -\frac{1}{B}\sum_{b=1}^{B} \log \hat{q}_\phi(\boldsymbol{z}^b) \geq H_\phi[\boldsymbol{z}]. \tag{12}$$

The upper bound relationship follows from Jensen's inequality which states that

$$\mathbb{E}[\log \hat{q}_\phi(\boldsymbol{z})] \leq \log \mathbb{E}[\hat{q}_\phi(\boldsymbol{z})] = \log q_\phi(\boldsymbol{z}). \tag{13}$$

A.1 Mutual Information between label y and representation z

We quantize each individual dimension $\boldsymbol{z}_d$ into 10 bins based on the CDF of the empirical distribution. In other words, dimension $\boldsymbol{z}_d$ is divided in a way that each bin contains 10% of the training data. We then compute the mutual information $I(\boldsymbol{x}; \boldsymbol{z}_d)$ as:

$$I(\boldsymbol{z} \in \text{bin}_i, \boldsymbol{y} = k) = q(z \in \text{bin}_i, \boldsymbol{y} = k)\left[\log \frac{q(z \in \text{bin}_i, \boldsymbol{y} = k)}{q(z \in \text{bin}_i) q(\boldsymbol{y} = k)}\right]$$

For the case where $\boldsymbol{z}$ is a concrete variable, we use the following formulation:

$$
\begin{aligned}
I(\boldsymbol{z}=l,\boldsymbol{y}=k) &= q(\boldsymbol{z}=l,\boldsymbol{y}=k)\left[\log\frac{q(\boldsymbol{z}=l,\boldsymbol{y}=k)}{q(\boldsymbol{z}=l)q(\boldsymbol{y}=k)}\right] \\
q(\boldsymbol{z}=l,\boldsymbol{y}=k) &= q(\boldsymbol{y}=k)q(\boldsymbol{z}=l|\boldsymbol{y}=k) \\
&= \frac{N_k}{N}q(\boldsymbol{z}=l|\boldsymbol{y}=k) \\
q(\boldsymbol{z}=l|\boldsymbol{y}=k) &= \sum_{\boldsymbol{x}} q(\boldsymbol{z}=l,\boldsymbol{x}|\boldsymbol{y}=k) \\
&= \sum_{\boldsymbol{x}} q(\boldsymbol{z}=l|\boldsymbol{x},\boldsymbol{y}=k)q(\boldsymbol{x}|\boldsymbol{y}=k) \\
&= \frac{1}{N_k}\sum_{x} q(\boldsymbol{z}=l|\boldsymbol{x},\boldsymbol{y}=k) \\
q(\boldsymbol{z}=l) &= \sum_{\boldsymbol{x}} q(\boldsymbol{z}=l,\boldsymbol{x}) \\
&= \sum_{\boldsymbol{x}} q(\boldsymbol{z}=l|\boldsymbol{x})q(\boldsymbol{x}) \\
&= \frac{1}{N}\sum_{\boldsymbol{x}} q(\boldsymbol{z}=l|\boldsymbol{x})
\end{aligned}
$$

Finally, for the overall mutual information we have:

$$
I(\boldsymbol{z},\boldsymbol{y}) = \sum_{l}\sum_{k} I(\boldsymbol{z}=l,\boldsymbol{y}=k)
$$

B Latent Traversals

Figure 8: Interpretable factors in CelebA for a HFVAE ($\beta = 5.0$, $\gamma = 3.0$) and a β-VAE ($\beta = 8.0$)

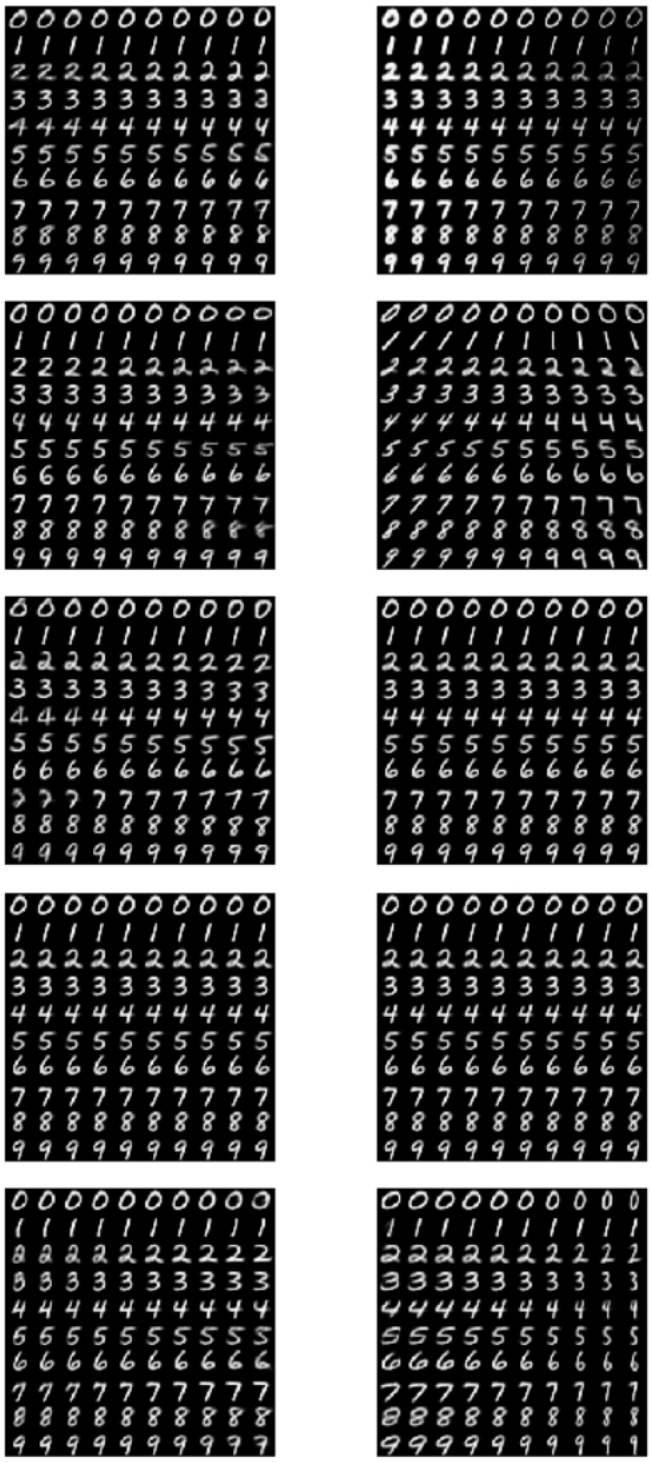


Figure 9: Qualitative results for disentanglement in MNIST dataset. In each case, one particular z_d is varying from -3 to 3 while the others are fixed at 0. For this particular set of traversals, we used 10% supervision in order to extract the digit more reliably, therefore visualizing all 'style' features present in MNIST.

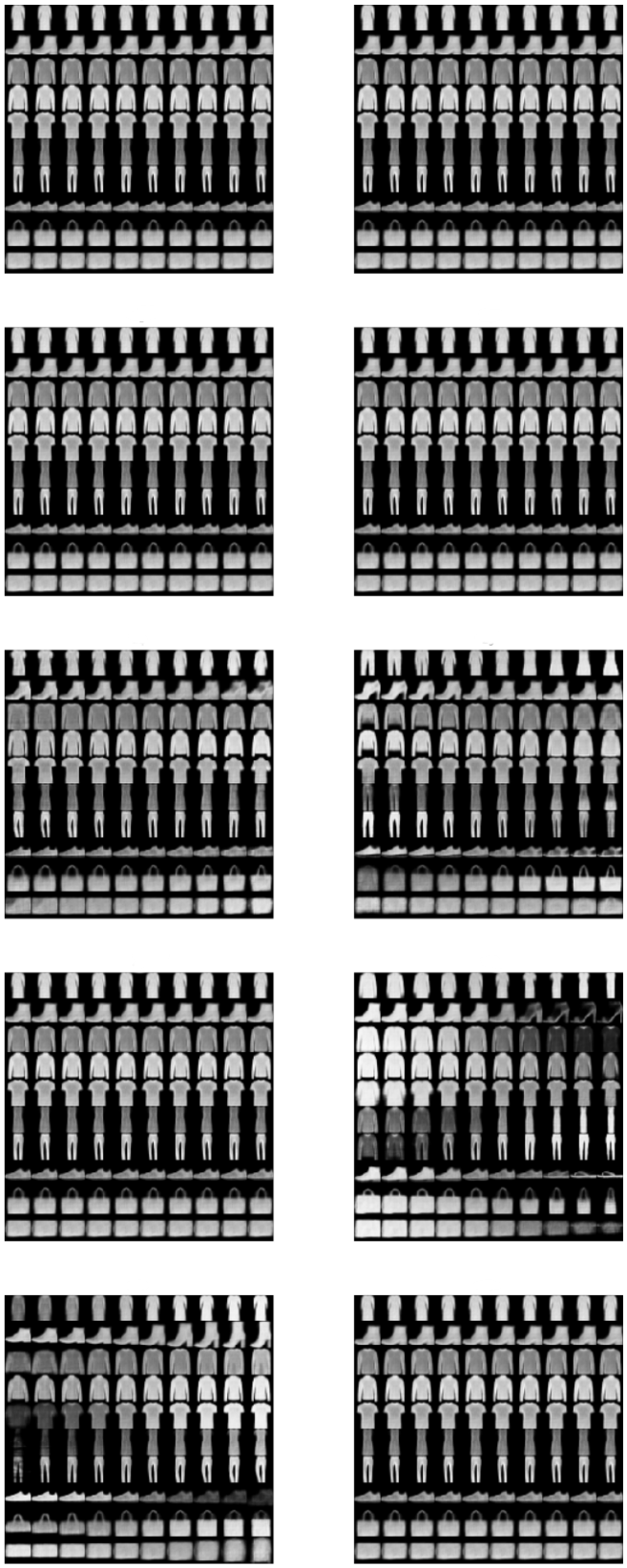


Figure 10: Qualitative results for disentanglement in F-MNIST dataset. In each case, one particular z_d is varying from -3 to 3 while the others are fixed at 0

C Model Architectures

We considered 4 datasets:

dSprites [Higgins et al., 2016]: 737,280 binary 64×64 images of 2D shapes with ground truth factors,

MNIST [LeCun et al., 2010]: 60000 gray-scale 28×28 images of handwritten digits,

F-MNIST [Xiao et al., 2017]: 60000 gray-scale 28×28 images of clothing items divided in 10 classes,

CelebA [Liu et al., 2015]: 202,599 RGB $64 \times 64 \times 3$ images of celebrity faces.

As mentioned in the main text, we used two hidden variables for each of the datasets. One variable is modeled as a normal distribution which representing continuous (denoted as z_c), and one modeled as a Concrete distribution to detect categories (denoted as z_d). We used Adam optimizer with learning rate $1e$-*3* and the default settings.

Table 4: Encoder and Decoder architecture for MNIST and F-MNIST datasets.

Encoder	**Decoder**
Input 28×28 grayscale image	Input $\boldsymbol{z} = \textit{Concat}\left[z_c \in \mathbb{R}^{10},\ z_d \in (0,1)^{10}\right]$
FC. 400 ReLU	FC. 200 ReLU
FC. 2×200 ReLU, FC. 10 (z_d)	FC. 400 ReLU
FC. 2×10 (z_c)	FC. 28×28 Sigmoid

Table 5: Encoder and Decoder architecture for dSprites data.

Encoder	**Decoder**
Input 64×64 binary image	Input $\boldsymbol{z} = \textit{Concat}\left[z_c \in \mathbb{R}^{10},\ z_d \in (0,1)^{3}\right]$
FC. 1200 ReLU	FC. 400 Tanh
FC. 1200 ReLU	FC. 1200 Tanh
FC. 2×400 ReLU, FC. 3 (z_d)	FC. 1200 Tanh
FC. 2×10 (z_c)	FC. 64×64 Sigmoid

Table 6: Encoder and Decoder architecture for CelebA data.

Encoder	**Decoder**
Input 64×64 RGB image	Input $\boldsymbol{z} = \textit{Concat}\left[z_c \in \mathbb{R}^{20},\ z_d \in \{0,1\}^{10}\right]$
4×4 conv, 32 BatchNorm ReLU, stride 2	FC. 256 ReLU
4×4 conv, 32 BatchNorm ReLU, stride 2	FC. $(4 \times 4 \times 64)$ Tanh
4×4 conv, 64 BatchNorm ReLU, stride 2	4×4 upconv, 64 BatchNorm ReLU, stride 2
4×4 conv, 64 BatchNorm ReLU, stride 2	4×4 upconv, 32 BatchNorm ReLU, stride 2
FC. 2×256 ReLU, 2 FC. (z_d)	4×4 upconv, 32 BatchNorm ReLU, stride 2
FC. 2×20 ReLU	4×4 upconv, 3, stride 2

Figure 11: Learned topics in the 20NewsGroups dataset using the HFVAE and the VAE objective. The middle column shows frequent words for the 3 most informative topics of the VAE, and the 3 *most correlated* topics in the HFVAE. The left column lists their corresponding mutual information with $\boldsymbol{x}$ and the topic coherence score. The right column shows the correlations between topics. The HFVAE learns 2 groups of topics that are internally uncorrelated (top-left and bottom-right quadrants), whilst uncovering sparse correlations between groups (top-right and bottom-left quadrants).

D Disentangled Representation for Text

D.1 Model Architectures

We consider the following dataset:

20NewsGroups : 11314 newsgroup documents which are partitioned in 20 categories. We used bag-of-words representation where vocabulary size is 2000, after removing stopwords using Mallet stopwords list.

With HFVAE objective, we used two hidden variables (denoted as z_{c1} and z_{c2}) with 25 dimensions each. In ProdLDA, we used Adam optimizer with $\beta_1 = 0.99$, $\beta_2 = 0.999$, and learning rate $1e\text{-}3$; In NVDM, we used Adam optimizer with learning rate $5e\text{-}5$ and default settings.

Encoder	**Decoder**
Input 1×2000 document	Input $z_{c1} \in \mathbb{R}^{25}$, $z_{c2} \in \mathbb{R}^{25}$
FC. 100 Softplus	Softmax Dropout
FC. 100 Softplus Dropout	FC. 2000 BatchNorm Softmax
FC. 2×25 BatchNorm (z_{c1})	
FC. 2×25 BatchNorm (z_{c2})	

Table 7: Encoder and Decoder architecture in ProdLDA.

Encoder	**Decoder**
Input 1×2000 document	Input $z_{c1} \in \mathbb{R}^{25}$, $z_{c2} \in \mathbb{R}^{25}$
FC. 500 ReLU	FC. 2000 Softmax
FC. 2×25 (z_{c1})	
FC. 2×25 (z_{c2})	

Table 8: Encoder and Decoder architecture in NVDM.

D.2 Neural variational document model

We train a standard NVDM with a 50-dimensional latent variable using the normal VAE objective. We compare this baseline to a HFVAE with two 25-dimensional latent variables, trained with $\beta = 7$, and $\gamma = 4$—*allowing* correlations within a group

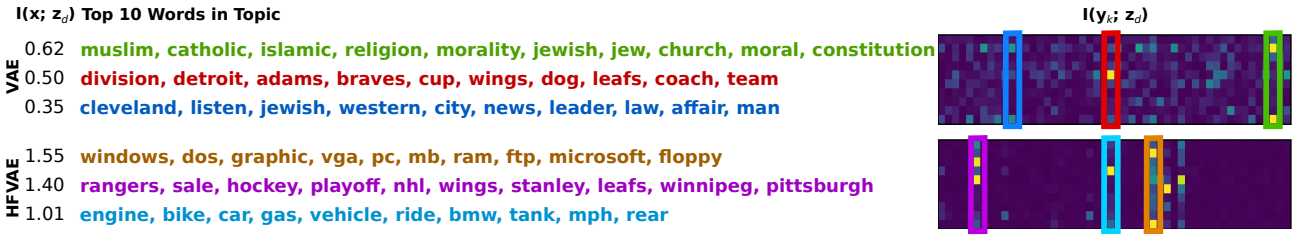


Figure 12: Learned topics in 20NewsGroups dataset using HFVAE objective and VAE objective. The middle column shows frequent words for the 3 most informative dimensions of the latent space. The left column lists their corresponding mutual information with $\boldsymbol{x}$. The right column shows the mutual information between the latent code and binary indicator variable for the document category.

but *preventing* correlations across groups.

Figure 12 (right) shows the mutual information between the latent code and binary indicator variables for the document category. We see that the latent dimensions of the HFVAE (columns) achieve a higher degree of disentanglement as is evident from the fact that indicator labels (shown as rows) correlate generally with only one latent feature (shown in columns). Note that a single feature can capture two distinct topics in this model (of which only one is shown), which correspond to negative and positive weights in the likelihood model.

D.3 Binary Indicator Variables for Document Category

In 20NewsGroups dataset, we derived 10 binary variables where each indicates whether a document belongs to this specific topic. Since some of the newsgroups are closely related (e.g. religion vs politics) while others are not related at all (e.g. science vs sports), we regarded highly related categories as one single topic. Then we computed the mutual information between each binary indicator variable b_l and individual dimension z_d (see Appendix A.1), which is shown in Figure 12.

Table 9: Topics after grouping highly related categories.

Grouped Topics	**Original Categories**
Atheism	alt.atheism
Computer	comp.graphics comp.os.ms-windows.misc comp.sys.ibm.pc.hardware comp.sys.mac.hardware comp.windows.x
Forsale	misc.forsale
Autos	rec.autos rec.motorcycles
Sports	rec.sport.baseball rec.sport.hockey
Encryption	sci.crypt
Electronics	sci.electronics
Medical	sci.med
Space	sci.space
Politics and Religion	talk.politics.misc talk.politics.guns talk.politics.mideast talk.religion.misc soc.religion.christian

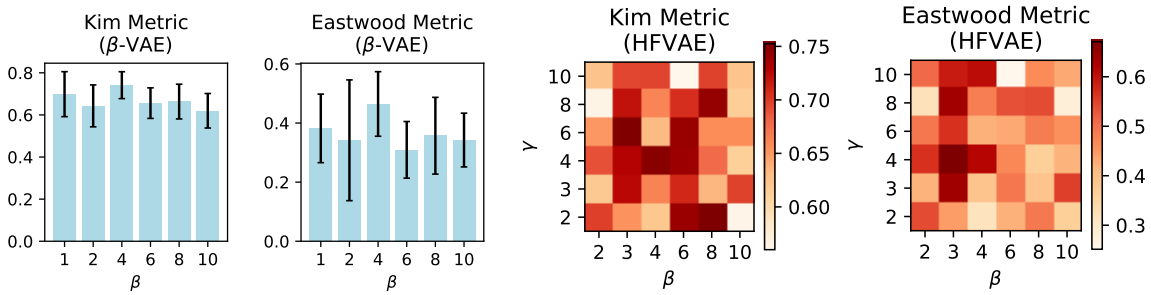
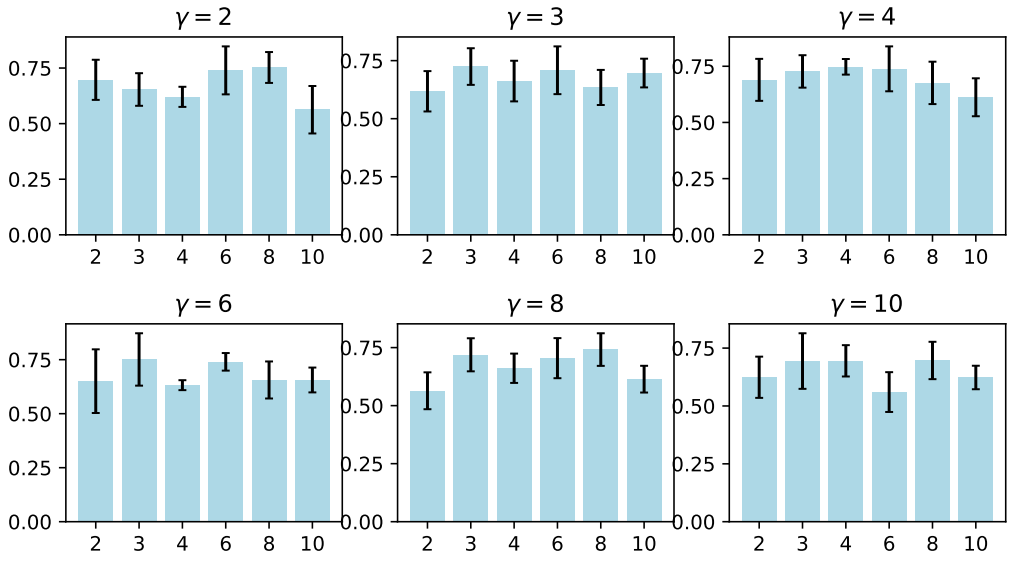


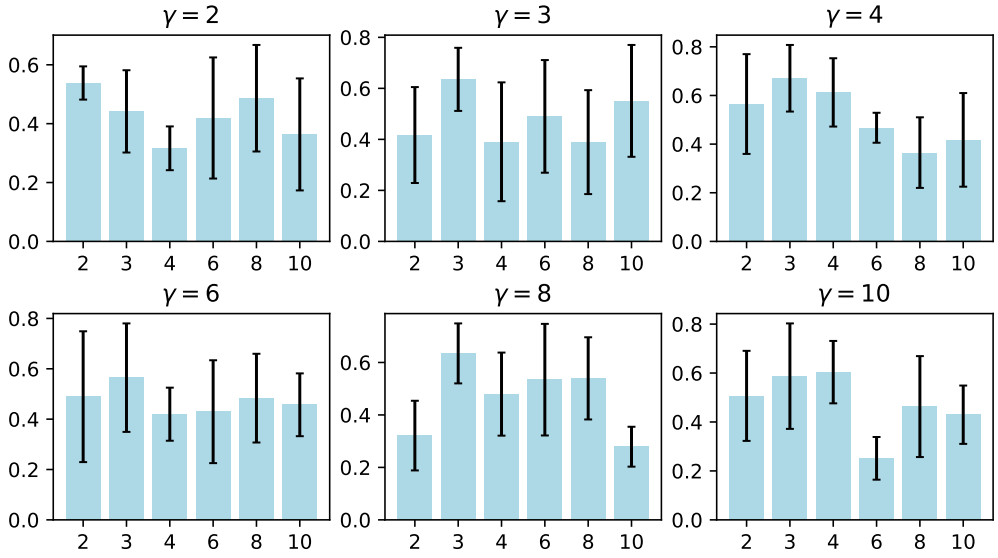
Figure 13: On the left, we plot the histogram of Kim and Eastwood scores for β-VAE with β hyperparameters. On the right, we plot the heatmap of the score means for HFVAE with different hyperparameter values.

E Hyperparameter Analysis Based on Disentanglement Metrics

In order to have better understanding of how well HFVAE performs compared to other approaches with respect to encoder's disentanglement capability, we ran all models: β-VAE, β-TCVAE and HFVAE wth 10 random restarts with a variety of hyperparameter values. For β-VAE, we tried $beta$s in the range $[1, 2, 4, 6, 8, 10]$, and for HFVAE with tried γ, β values in the range of $[2, 3, 4, 6, 8, 10]$. The prior we choose for all models consist of a 10-dimensional Gaussian with a diagonal covariance, and a Concrete variable of length 3 (number of shapes in dSprite). The results can be observed in Figures 13 and 14. In general, we found that the most influential factor in archiving a good disentanglement score is the starting random seed rather than the hyperparameter choice or the model. We note that the instability in our experiments is higher compared to previous work, as the prior we used also consist of a Concrete variable, thus the encoder has more options in terms of encoding different information in different variables/dimensions.



(a) Kim metric



(b) Eastwood metric

Figure 14: On the top, we show the histograms of Kim metric values for a range of different γ and β. On the bottom, we show the same but for the Eastwood metric.