Disentangled Representation Learning with β-VAE

Ahsan Mir

December 10, 2024

Abstract

This report presents an implementation and evaluation of disentangled representation learning using the β-VAE framework on the dSprites dataset. We systematically vary the β hyperparameter and compare the resulting models against a baseline VAE. We visualize the latent traversals, evaluate metrics such as Mutual Information Gap (MIG), Z-diff, and Modularity, and discuss the implications of increasing β on both disentanglement and reconstruction quality. Finally, we highlight the limitations of β-VAEs in capturing fine-grained details and fully separating certain generative factors.

1 Introduction

The objective of disentangled representation learning (DRL) is to obtain a latent space where individual latent dimensions correspond to meaningful generative factors of the data. This is of particular importance in interpretable and controllable generation tasks. In this task, we analyze β-VAE [1], a popular method that introduces a hyperparameter β to the VAE loss function to encourage disentanglement.

In a standard Variational Autoencoder (VAE) [2, 3], the loss function is:

$$\mathcal{L}_{\text{VAE}}(\theta, \phi) = \mathbb{E}_{q_\phi(z|x)}[-\log p_\theta(x|z)] + KL(q_\phi(z|x)||p(z)), \tag{1}$$

where θ and ϕ are the decoder and encoder parameters, respectively, and $p(z)$ is typically chosen as a standard normal prior $N(0, I)$.

The β-VAE modifies this objective to:

$$\mathcal{L}_{\beta\text{-VAE}}(\theta, \phi) = \mathbb{E}_{q_\phi(z|x)}[-\log p_\theta(x|z)] + \beta \cdot KL(q_\phi(z|x)||p(z)), \tag{2}$$

where $\beta > 1$ places greater emphasis on the latent channel capacity constraint and can lead to improved disentanglement at the expense of reconstruction fidelity.

1.1 Dataset

We use the dSprites dataset [4], which consists of 2D shapes (squares, ellipses, and hearts) with variations in five generative factors:

- Shape (3 possible values)
- Scale (6 values)

- Orientation (40 values)
- Position X (32 values)
- Position Y (32 values)

We chose dSprites because it provides clean, well-controlled generative factors and a large number of images (64x64 resolution) that are well-suited for testing disentanglement.

1.2 Models

We train a baseline VAE ($\beta = 1$) and multiple β-VAEs by varying $\beta \in \{0.5, 1, 2.5, 5, 7.5, 10, 50\}$. Each model uses a convolutional encoder-decoder architecture with a latent dimension $z = 10$. The models are trained for 10 epochs each.

2 Methodology

2.1 Dataset Preparation

We used a subset of 50,000 images randomly sampled from the full dSprites dataset. Each image is a single-channel 64x64 pixel binary image. The dataset is downloaded and loaded into memory-mapped arrays to ensure efficient training.

2.2 Model Configurations

We implement an encoder to map x to $q_\phi(z|x)$ and a decoder $p_\theta(x|z)$ to reconstruct x. For training, we use the Adam optimizer with a learning rate of 10^{-3}, a batch size of 32, and train each model for 10 epochs.

2.3 Assumptions and Task Setup

We assume that increasing β will improve disentanglement by penalizing the KL term more strongly, pushing latent codes to be more factorized. However, this may also deteriorate reconstruction quality. After training, we:

1. Visualize latent traversals for qualitative assessment of disentanglement.
2. Compute quantitative metrics (MIG, Z-diff, Modularity) to objectively measure disentanglement.
3. Compare the baseline VAE ($\beta = 1$) with other β values to analyze trends.

3 Results

3.1 Loss Curves

Figures 1, 2, and 3 show the KL divergence, reconstruction, and total loss curves over 10 epochs for all β values. As β increases, the KL term is more heavily penalized, leading to:

- Higher KL divergence loss penalties for larger β values (Figure 1).
- Higher reconstruction losses, especially at very large β (e.g., $\beta = 50$), as the model becomes less focused on accurately reconstructing (Figure 2).
- Overall total loss balances these two terms, but for extremely large β, the total loss remains high (Figure 3).

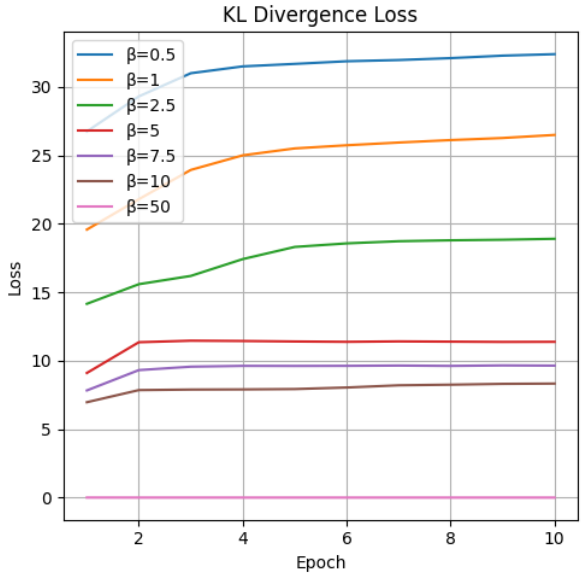

Figure 1: KL Divergence Loss for all β values over 10 epochs. Higher β values yield higher KL penalty.

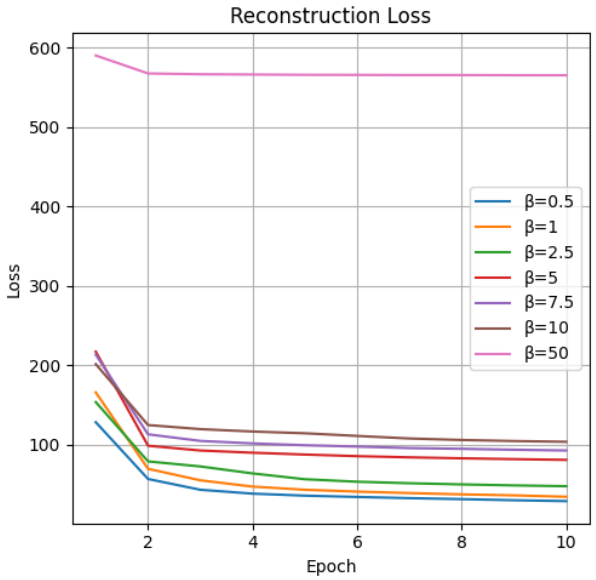

Figure 2: Reconstruction Loss for all β values. As β increases, the model focuses less on reconstruction quality, increasing this loss.

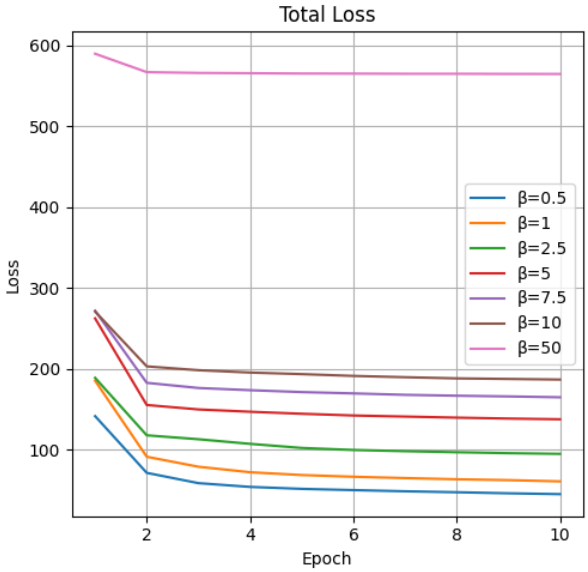

Figure 3: Total Loss for all β values. For moderate β values, the total loss decreases, while very high β (e.g., 50) leads to stagnation at a high loss value.

3.2 Visualizing Disentangled Latent Factors

Figure 4 shows latent traversals for $\beta = 5$. In this visualization, we pick a single image and vary one latent dimension at a time while fixing the others. Each row corresponds to a latent dimension, and each column corresponds to a sampled value from a continuous range. By examining the changes in the generated images, we can identify which latent dimensions control which generative factors:

- Some latent dimensions may correlate with orientation (e.g., the shape rotates as we move along that dimension).
- Others might correspond to shape changes or scale modifications.
- Position factors may appear if the object's location shifts across the canvas.

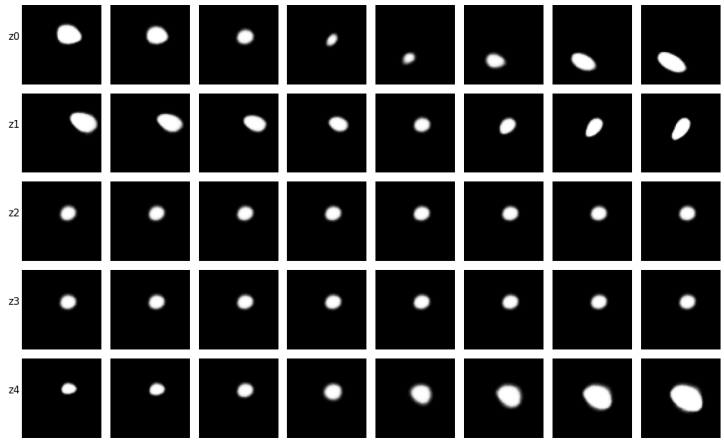

Figure 4: Latent traversals for $\beta = 5$. Each row corresponds to a latent dimension z_i, and each column is a step in that dimension. We observe changes in orientation, shape, or position as we move along individual latent axes.

3.3 Disentanglement Metrics

We evaluate three metrics:

1. **Mutual Information Gap (MIG):** Measures how well each latent dimension captures a single generative factor by comparing the top two mutual information scores. A higher MIG means better disentanglement.

2. **Z-diff:** Measures how changes in latent variables correspond to changes in generative factors by correlation of latent differences and factor differences. Higher Z-diff indicates more factor-specific latent dimensions.

3. **Modularity:** Checks how each latent dimension correlates with one factor, penalizing when a latent variable correlates strongly with multiple factors. Higher modularity implies a more modular (disentangled) representation.

Figure 5 shows how these metrics change with β. The baseline ($\beta = 1$) is marked with a red dashed line.

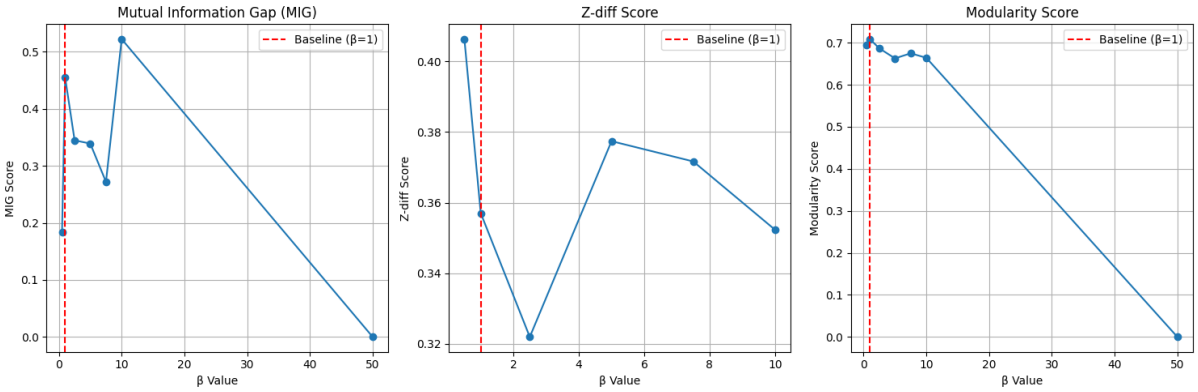

Figure 5: MIG, Z-diff, and Modularity scores for various β values. The red dashed line shows the baseline at $\beta = 1$. We see a non-monotonic relationship where certain intermediate β values yield better MIG scores, while very large β (e.g., 50) collapses performance.

3.4 Correlation Matrices

To gain a deeper insight, we plot the MIG, Z-diff, and Modularity matrices for the baseline model ($\beta = 1$) and a top-performing β (e.g., $\beta = 10$). These matrices show which latent variables correlate most strongly with which generative factors.

For $\beta = 1$, the correlation is more spread out. For $\beta = 10$, we observe more distinct patterns of specialization (one latent dimension might control shape, another orientation, etc.), though not perfectly disentangled. Figures 6 and 7 illustrate these points.

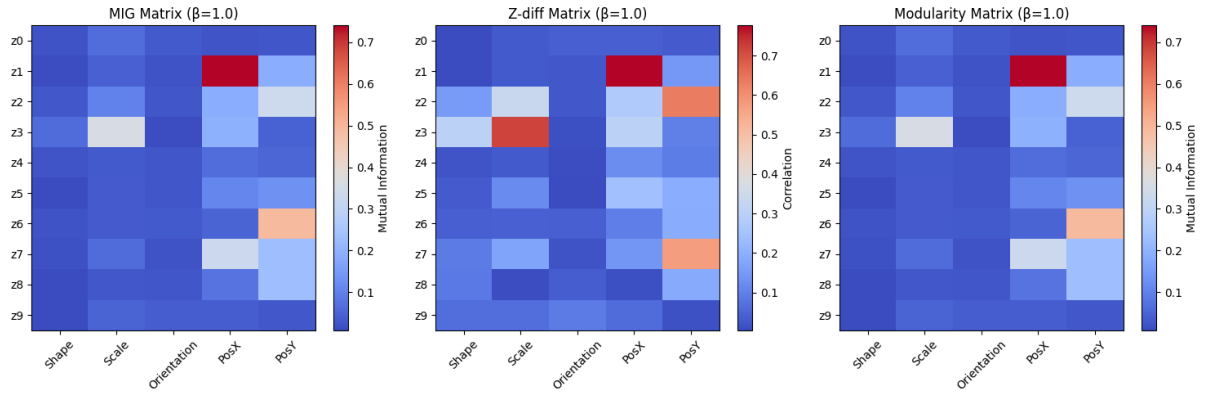

Figure 6: MIG, Z-diff, and Modularity matrices for $\beta = 1$ (baseline). We note that certain factors (e.g., orientation) show moderately high mutual information with one or two latent dimensions, but the entanglement is still relatively broad.

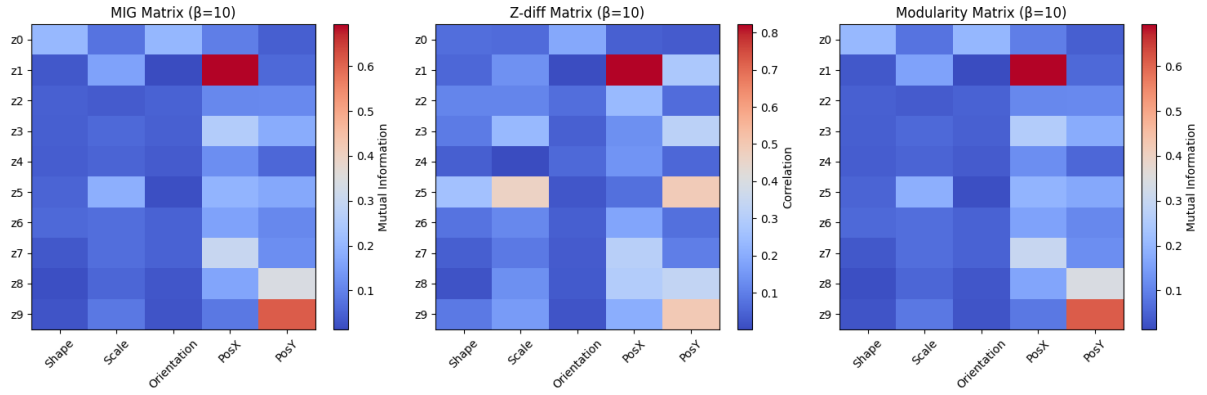

Figure 7: MIG, Z-diff, and Modularity matrices for $\beta = 10$. Here we see clearer patterns where some latent variables correlate more distinctly with individual factors compared to the baseline.

3.5 Summary of Disentanglement Metrics

Table 1 summarizes the MIG, Z-diff, and Modularity scores for all tested β values. Notably:

- $\beta = 0.5$ and $\beta = 10$ achieve relatively good MIG scores.
- Extremely high $\beta = 50$ collapses the representation, resulting in near-zero MIG and Modularity, and undefined Z-diff.

Table 1: Summary of Disentanglement Metrics for each β value

β Value	MIG Score	Z-diff Score	Modularity Score
0.5	0.1844	0.4063	0.6956
1.0	0.4547	0.3570	0.7087
2.5	0.3448	0.3219	0.6867
5.0	0.3391	0.3774	0.6628
7.5	0.2720	0.3717	0.6749
10.0	0.5224	0.3523	0.6642
50.0	0.0000	nan	0.0000

4 Discussion

4.1 Effect of Increasing β on Disentanglement and Reconstruction

As β increases from 1, we generally see improved disentanglement (e.g., MIG increases at certain intermediate β values). This is because we impose a stronger KL divergence penalty. This forces the model to use fewer latent dimensions and to push latent distributions closer to the prior. This leads to more "compressed" representations and can harm reconstruction fidelity because the model cannot afford to perfectly reconstruct each detail. We observe this while pushing β too far (e.g., $\beta = 50$) which leads to a near-collapse in reconstruction quality and no meaningful disentanglement. This suggests there is an optimal range of β values that balance the trade-off between encouraging factorization and maintaining fidelity.

4.2 Qualitative and Quantitative Trends

Qualitatively, latent traversals show that for moderate β values (5 or 10), distinct latent dimensions control different aspects of the image (e.g., orientation, position). Quantitatively, MIG, Z-diff, and Modularity reflect these observations, peaking at intermediate β values and decreasing when β is too small or too large.

4.3 Limitations of β-VAE

Despite its strengths, the β-VAE approach has notable limitations:

- Loss of fine-grained details in reconstructions, especially as β increases. This is because the model prioritizes global structures (factors of variation) rather than pixel-perfect fidelity.
- Certain factors might remain entangled if they are inherently more complex or do not align well with the Gaussian latent space assumptions. Therefore, even with a high β, there's no absolute guarantee that each latent dimension corresponds neatly to a single generative factor.
- Sensitivity to hyperparameters: selecting a proper β is non-trivial and may depend heavily on the dataset and desired outcome. The learned disentanglement can

depend on the network architecture, optimization parameters, random seeds, and the complexity of the dataset. On more complex datasets like CelebA, β-VAEs may not disentangle as clearly as on dSprites.

5 Conclusion

We implemented a β-VAE on the dSprites dataset and systematically varied β. We found that moderate increases in β beyond 1 can improve disentanglement as measured by MIG and Z-diff, but too large a β severely degrades both reconstruction and disentanglement. Our analysis also highlights that while β-VAE can learn more factorized representations than a standard VAE, it is not guaranteed to disentangle all factors nor preserve fine details in the output images. Essentially, increasing β improves disentanglement but at a cost of reconstruction quality. There is a trade-off that must be resolved depending on the priorities of the application.

References

[1] Higgins, I. et al. "beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework." ICLR, 2017.

[2] Kingma, D.P. and Welling, M. "Auto-Encoding Variational Bayes." ICLR, 2014.

[3] Rezende, D.J., Mohamed, S., and Wierstra, D. "Stochastic Backpropagation and Approximate Inference in Deep Generative Models." ICML, 2014.

[4] Matthey, L. et al. "dSprites: Disentanglement testing Sprites dataset." DeepMind, 2017.