# β-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework

Presented by Mohamed Mohamed

# My Contributions

- Codebase builds upon *WonKwang Lee and Tony Metger* [1] , where I introduce key improvements for usability, reproducibility, and performance

# My Contributions

https://github.com/lesupermomo/B-VAE Pytorch reproduction of two papers:

1. *β-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework, Higgins et al., ICLR, 2017* [2]
2. *Understanding disentangling in β-VAE, Burgess et al., arxiv:1804.03599, 2018* [3]

|------ Dataset Management
|    |---- Integrated scripts to automatically download datasets (3DChairs and CelebA) with a single command.
|
|------ New Features and bug fixes
|    |---- Fixed a bug related to ploting the posterior mean.
|    |---- Updated the plot gif method to use `imageio` insteaf of `grid2gif` to enhance scalability of the codebase.
|
|------ Code Compatibility and Optimization
|    |---- Updated the codebase to be compatible with the latest version of PyTorch, enhancing scalability and performance.
|    |---- Refactored the code to remove unnecessary methods and ensure smooth execution
|        regardless of the working directory from which `main.py` is called.
|
|------ GPU Selection
|    |---- Enabled GPU selection to allow running experiments on specified GPUs, providing flexibility for multi-GPU environments.
|
|------ Project Organization
|    |---- Created a dedicated `scripts/` directory to streamline the setup and execution of experiments.
|    |---- Added a `requirements.txt` file that includes all project dependencies for easy environment setup.
|
|------ Documentation and Reproducibility
|    |---- Developed detailed documentation on how to set up the environment and reproduce experimental results.

# Introduction - VAE [4]

- VAEs provide an automated discovery of interpretable factorised latent representation [4]
- Sampling from the latent space produces meaningful synthetic data, based on the latent parameters

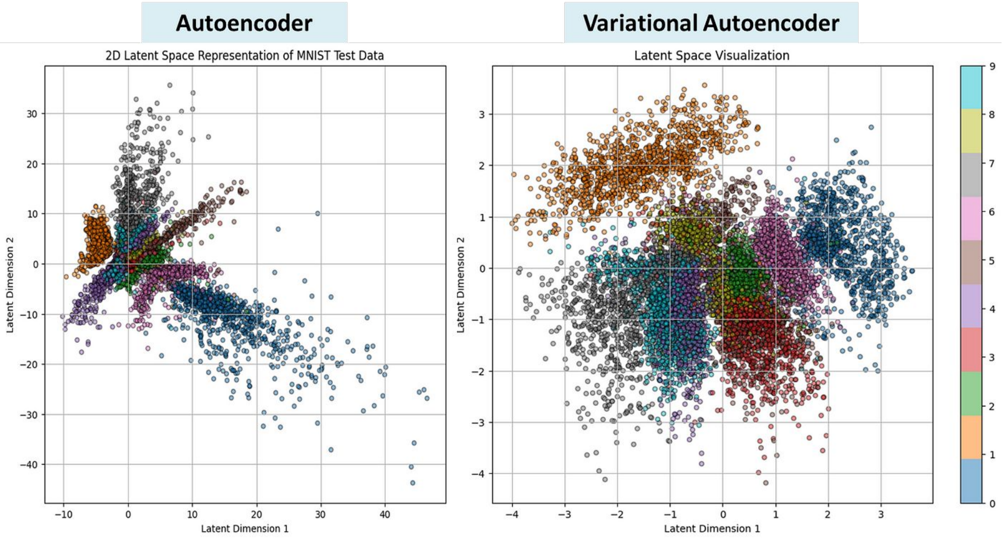
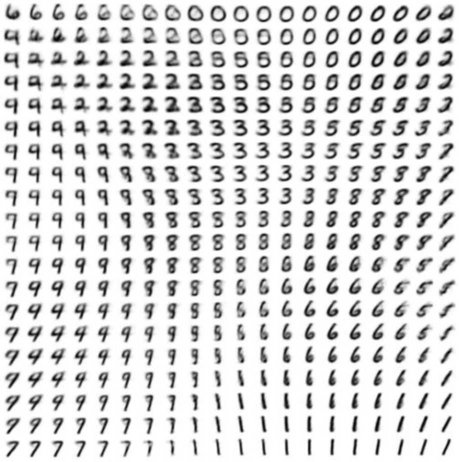



Figure 1: Autoencoder vs VAE latent space



(b) Learned MNIST manifold

Figure 2: : Visualisations of learned data manifold for generative models with two-dimensional latent space [4]

Figure 1: https://medium.com/@hugmanskj/hands-on-understanding-and-implementing-variational-autoencoders-1013b658d216
Figure 2: [4] Kingma, Diederik P. "Auto-encoding variational bayes." arXiv preprint arXiv:1312.6114 (2013)

# VAE - Theory: Defining the Marginal Likelihood [4]

- The likelihood of our data can be defined as the marginalization over the joint prob. Dist. w.r.t the latent variables.

$$p(\boldsymbol{x}) = \int p(\boldsymbol{x}, \boldsymbol{z}) d\boldsymbol{z}$$

is intractable since we would need to integrate over all latent variables Z.

$$p(\boldsymbol{x}) = \frac{p(\boldsymbol{x}, \boldsymbol{z})}{p(\boldsymbol{z}|\boldsymbol{x})}$$

the true posterior ***p(z|x)*** is also intractable

$$p(\boldsymbol{z}|\boldsymbol{x}) = \frac{p(\boldsymbol{x}, \boldsymbol{z})}{p(\boldsymbol{x})}$$

$$q_\phi(\mathbf{z}|\mathbf{x})$$

They introduce the surrogate which is an approximation to the intractable true posterior

# VAE - Theory: Defining the Marginal Likelihood [4]

$$\log p_\theta(\boldsymbol{x}) = \log p_\theta(\boldsymbol{x})$$

$$= \log p_\theta(\boldsymbol{x}) \int q_\varphi(\mathbf{z}|\boldsymbol{x})d\mathbf{z}$$ Multiply by 1

$$= \int \log p_\theta(\boldsymbol{x})\, q_\varphi(\mathbf{z}|\boldsymbol{x})d\mathbf{z}$$ Bring inside the integral

$$= E_{q_\varphi(\mathbf{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x})]$$ Definition of expectation

$$= E_{q_\varphi(\mathbf{z}|\boldsymbol{x})}\left[\log \frac{p_\theta(\boldsymbol{x},\mathbf{z})}{p_\theta(\mathbf{z}|\boldsymbol{x})}\right]$$ Apply the equation $p_\theta(\boldsymbol{x}) = \frac{p_\theta(\boldsymbol{x},\mathbf{z})}{p_\theta(\mathbf{z}|\boldsymbol{x})}$

$$= E_{q_\varphi(\mathbf{z}|\boldsymbol{x})}\left[\log \frac{p_\theta(\boldsymbol{x},\mathbf{z})q_\varphi(\mathbf{z}|\boldsymbol{x})}{p_\theta(\mathbf{z}|\boldsymbol{x})q_\varphi(\mathbf{z}|\boldsymbol{x})}\right]$$ Multiply by 1

$$= E_{q_\varphi(\mathbf{z}|\boldsymbol{x})}\left[\log \frac{p_\theta(\boldsymbol{x},\mathbf{z})}{q_\varphi(\mathbf{z}|\boldsymbol{x})}\right] + E_{q_\varphi(\mathbf{z}|\boldsymbol{x})}\left[\log \frac{q_\varphi(\mathbf{z}|\boldsymbol{x})}{p_\theta(\mathbf{z}|\boldsymbol{x})}\right]$$ Split the expectation

$$\log p_\theta(\boldsymbol{x}) = \underbrace{E_{q_\varphi(\mathbf{z}|\boldsymbol{x})}\left[\log \frac{p_\theta(\boldsymbol{x},\mathbf{z})}{q_\varphi(\mathbf{z}|\boldsymbol{x})}\right]}_{\text{ELBO}} + \underbrace{D_{KL}\left(q_\varphi(\mathbf{z}|\boldsymbol{x})\|p_\theta(\mathbf{z}|\boldsymbol{x})\right)}_{\geq 0}$$ Definition of KL divergence

[4] Kingma, Diederik P. "Auto-encoding variational bayes." arXiv preprint arXiv:1312.6114 (2013)

# VAE - Theory: Understanding the likelihood [4]

$$\log p_\theta(\boldsymbol{x}) = \underbrace{E_{q_\varphi(\mathbf{z}|\boldsymbol{x})}\left[\log \frac{p_\theta(\boldsymbol{x},\mathbf{z})}{q_\varphi(\mathbf{z}|\boldsymbol{x})}\right]}_{\textbf{ELBO}} + \underbrace{D_{KL}\left(q_\varphi(\mathbf{z}|\boldsymbol{x}) \| p_\theta(\mathbf{z}|\boldsymbol{x})\right)}_{\geq 0}$$

$$\log p_\theta(\boldsymbol{x}) \geq E_{q_\varphi(\mathbf{z}|\boldsymbol{x})}\left[\log \frac{p_\theta(\boldsymbol{x},\mathbf{z})}{q_\varphi(\mathbf{z}|\boldsymbol{x})}\right] = E_{q_\varphi(\mathbf{z}|\boldsymbol{x})}\left[\log \frac{p_\theta(\boldsymbol{x}|\mathbf{z})p(\mathbf{z})}{q_\varphi(\mathbf{z}|\boldsymbol{x})}\right]$$

$$= E_{q_\varphi(\mathbf{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}|\mathbf{z})] + E_{q_\varphi(\mathbf{z}|\boldsymbol{x})}\left[\log \frac{p(\mathbf{z})}{q_\varphi(\mathbf{z}|\boldsymbol{x})}\right]$$

$$= E_{q_\varphi(\mathbf{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}|\mathbf{z})] - D_{KL}\left(q_\varphi(\mathbf{z}|\boldsymbol{x}) \| p_\theta(\mathbf{z})\right)$$

By Maximizing the ELBO

- LHS: Maximizing the reconstruction likelihood of the decoder
- RHS: Minimizing the KL term enforces our prior belief on the latent variables

[4] Kingma, Diederik P. "Auto-encoding variational bayes." arXiv preprint arXiv:1312.6114 (2013)

# β-VAE [2]

$$\log p_\theta(\boldsymbol{x}) \geq E_{q_\varphi(\mathbf{z}|\boldsymbol{x})}\left[\log \frac{p_\theta(\boldsymbol{x}, \mathbf{z})}{q_\varphi(\mathbf{z}|\boldsymbol{x})}\right] = E_{q_\varphi(\mathbf{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}|\mathbf{z})] - D_{KL}\left(q_\varphi(\mathbf{z}|\boldsymbol{x}) \| p_\theta(\mathbf{z})\right)$$

$$\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \beta\, D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))$$

β-VAE is a Variational Autoencoder with a special emphasis to discover interpretable disentangled representations. [2]

- When $\beta > 1$ : It applies a strong constraint on the latent bottleneck and limits the representation capacity of z.
  - the model is pushed to learn a more efficient latent representation of the data, which is disentangled if the data contains at least some underlying factors of variation that are independent [2]
- Limitation: Tradeoff between the reconstruction and compact latent representations

# Datasets

- dSprites: Contains 2D shapes procedurally generated from 6 ground truth independent latent factors. These factors are color, shape, scale, rotation, x and y positions of a sprite. [2]
- 3DChairs: Contains rendered images of around 1000 different three-dimensional chair models. [5]
- CelebA: 200k images of celebrities [6]
- Faces: 3D face model database created for pose and illumination invariant face recognition [7]

# Results (ours)

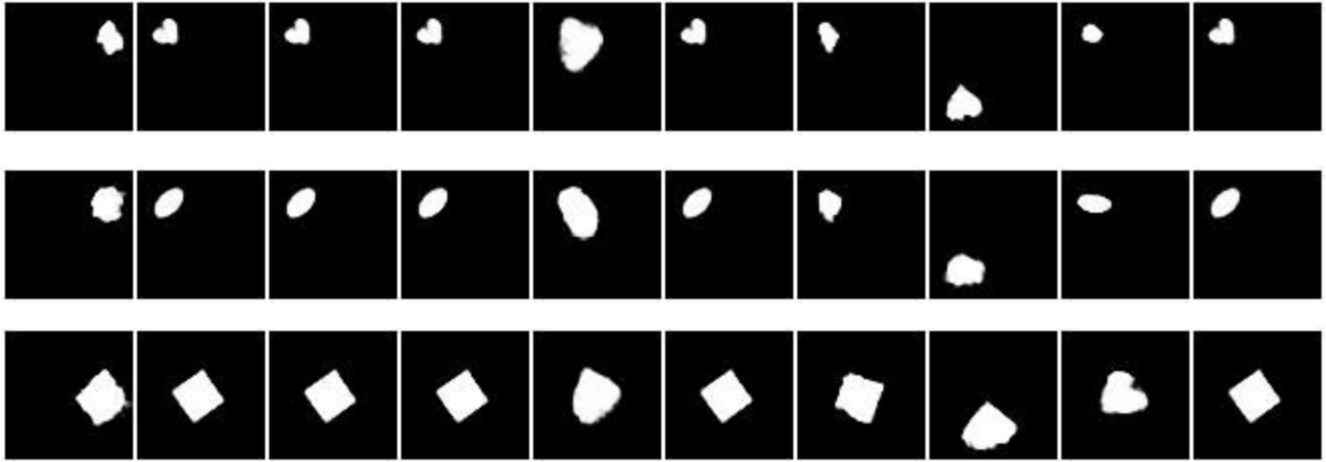
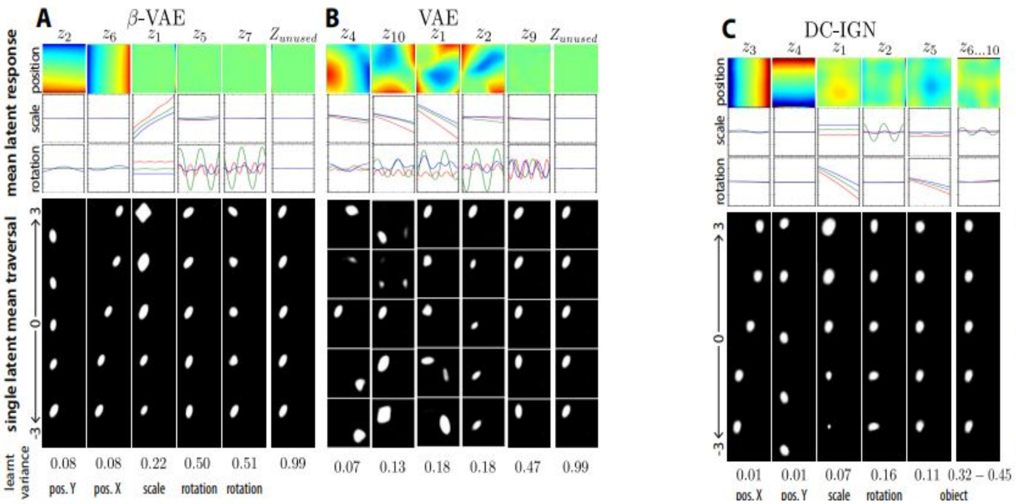


Figure 3: Effect of modifying different latent parameters

- 1 -> X position
- 8 -> Y position
- 5 -> scale
- 7 -> Rotation
- 9 -> shape/rotation
- The rest is unused

# Results (Theirs)




[2] Figure 4: Representations learnt by different models

- Row 1 shows the effect of varying the position
- Row 2 shows the effect of varying the scale
- Row 3 shows the effect of varying the rotation
- Row 4-8 show reconstructions resulting from the traversal of each latent

# Results (ours)

Figure 5: Representations learnt for 3D chairs z=10 β=4

Figure 6: Representations learnt for CelebA  z=10 β=10

# Results (theirs)

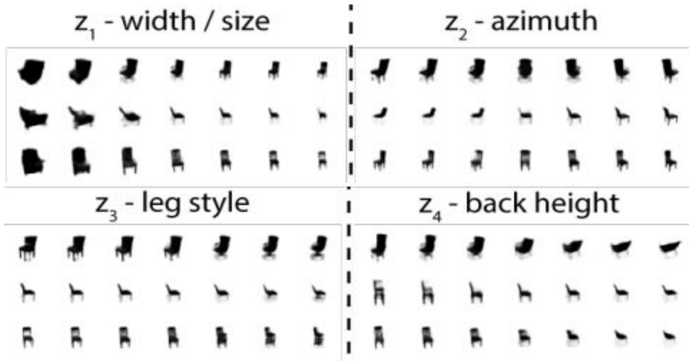



Figure 7: Representations learnt for 3D chairs

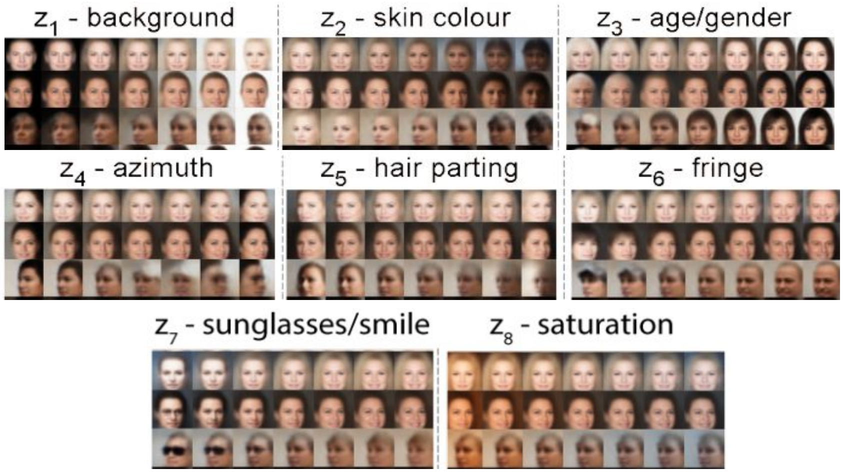



Figure 8: Representations learnt for CelebA

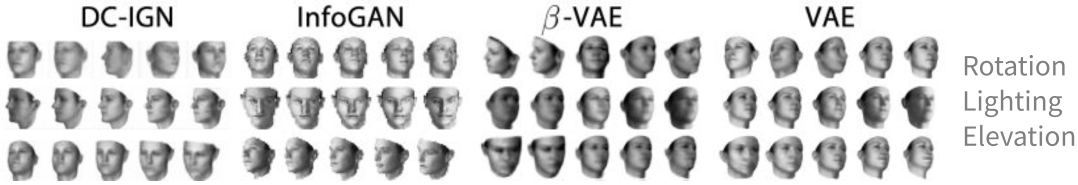



Figure 9: Representations learnt for 3D faces

# Disentanglement Metric

The goal of the metric is to have generative factors that are *interpretable* and *independent*. [2]

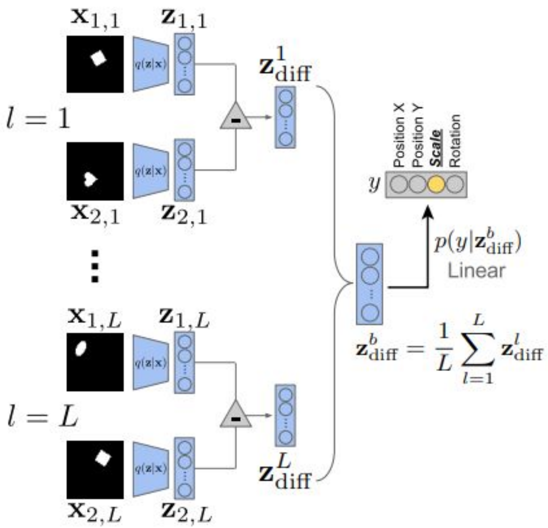



Figure 10: Disentanglement Metric

1. Fix a Target Generative Factor:
   Select a generative factor (e.g., scale) and generate two sets of samples with y fixed while other factors vary.

2. Infer Latent Representations:
   Use the encoder to map images into latent representations. Compute the absolute difference between the two latent representations for each sample.

3. Measure Variance in Latents:
   Calculate the average absolute difference. Latent dimensions corresponding to y should exhibit minimal variance.

4. Predict the Generative Factor:
   Train a low-capacity linear classifier to predict y based on $z_{diff}$. A disentangled representation simplifies this task.

5. Final Metric Score:
   The classifier's accuracy serves as the disentanglement metric. Higher accuracy reflects better alignment of latent variables with generative factors.

[2] β-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework, Higgins et al., ICLR, 2017

# Disentanglement Metric Results (theirs)

| Model | Disentanglement metric score |
|---|---|
| *Ground truth* | *100%* |
| Raw pixels | $45.75 \pm 0.8\%$ |
| PCA | $84.9 \pm 0.4\%$ |
| ICA | $42.03 \pm 10.6\%$ |
| DC-IGN | $\mathbf{99.3 \pm 0.1\%}$ |
| InfoGAN | $73.5 \pm 0.9\%$ |
| VAE untrained | $44.14 \pm 2.5\%$ |
| VAE | $61.58 \pm 0.5\%$ |
| **$\beta$-VAE** | $\mathbf{99.23 \pm 0.1\%}$ |

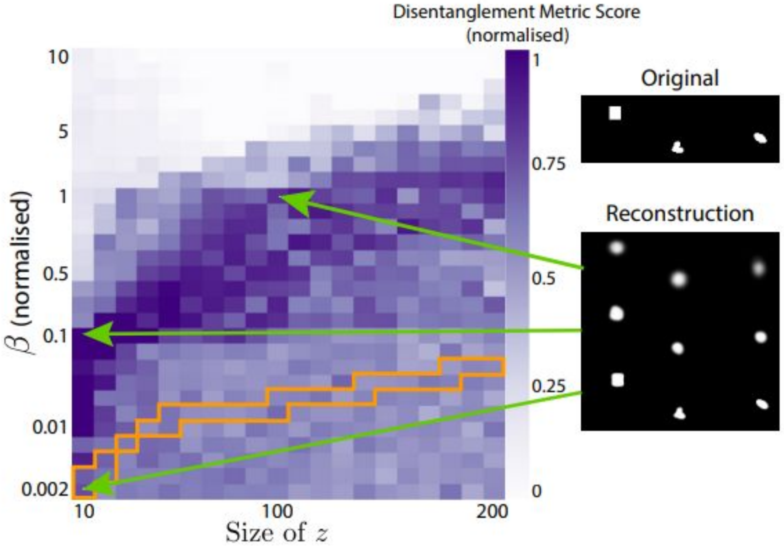



Figure 11: Disentanglement metric classification accuracy for 2D shapes dataset

# Understanding disentangling in β-VAE [3]

Limitations of β-VAE: disentangled representations tradeoff with reconstruction accuracy

The paper sheds light on why β-VAE disentangles from an information theory perspective, and uses the insights to suggest practical improvements to the training procedure[3]

The $\beta$-VAE objective is closely related to the information bottleneck principle

$$\max[I(Z;Y) - \beta I(X;Z)]$$

Small $\beta$: Higher mutual information $I(X;Z)$ as the model retains more information
Large $\beta$: Lower mutual information $I(X;Z)$ as the model enforces more compression and disentanglement.

[3] Understanding disentangling in β-VAE, Burgess et al., arxiv:1804.03599, 2018

# Understanding disentangling in β-VAE [3]

$$\mathcal{L}(\theta, \phi; \mathbf{x}(\mathbf{f}), \mathbf{z}, C) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{f})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \gamma \left| D_{KL}\big(q_\phi(\mathbf{z}|\mathbf{f}) \parallel p(\mathbf{z})\big) - C \right|$$

- The generative factors ***f*** are distinct (e.g., scale, shape) and vary in importance depending on the dataset.
- When the model's capacity is very **low**, it prioritizes representing the **most important generative factor** to optimize the reconstruction objective.
- As the model's capacity increases, it **progressively recovers and represents additional generative factors**.
- The authors propose a controlled capacity increase strategy:
  - Early stages of training: Focus on encouraging disentanglement by limiting capacity.
  - Later stages of training: Emphasize reconstruction by increasing capacity.
- This strategy ensures the model achieves both disentangled features and high-quality reconstructions, **as features are less likely to be reallocated during the high-capacity phase**.

# Understanding disentangling in β-VAE Results [3]

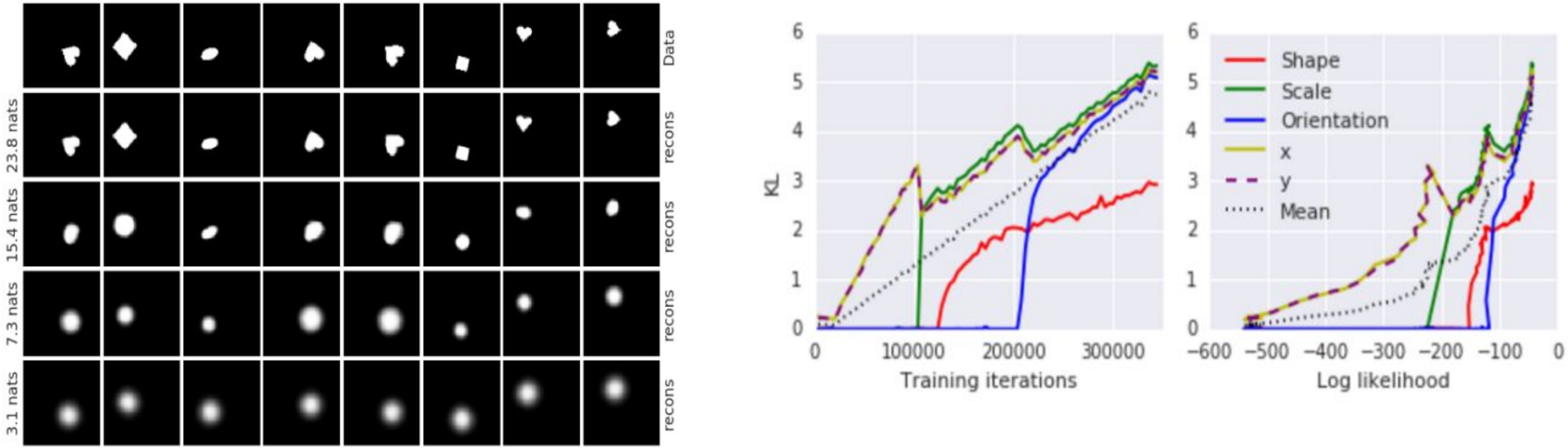



Figure 12: Disentanglement metric classification accuracy for 2D shapes dataset

[3] Understanding disentangling in β-VAE, Burgess et al., arxiv:1804.03599, 2018

# References

[1] Lee, W., & Metger, T. (2018). 1Konny/beta-VAE: Pytorch implementation of β-vae. GitHub. https://github.com/1Konny/Beta-VAE
[2] *β-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework, Higgins et al., ICLR, 2017*
[3] *Understanding disentangling in β-VAE, Burgess et al., arxiv:1804.03599, 2018*
[4] Kingma, Diederik P. "Auto-encoding variational bayes." arXiv preprint arXiv:1312.6114 (2013)
[5] Aubry, Mathieu, et al. "Seeing 3d chairs: exemplar part-based 2d-3d alignment using a large dataset of cad models." Proceedings of the IEEE conference on computer vision and pattern recognition. 2014.
[7] Paysan, Pascal, et al. "A 3D face model for pose and illumination invariant face recognition." 2009 sixth IEEE international conference on advanced video and signal based surveillance. Ieee, 2009.

# Discussion

- In which scenario is the loss function for the β-VAE equivalent to that of the VAE?
- What analogies can we make with respect to β-VAE and PCA?
    - Is there an ordering necessary for the latent representations?
- What other methods can you think of for ensuring disentangled latent representations?
- How does limiting the constraint on the latent space and the KL divergence ensure disentanglement
- Are there specific domains or types of data where β-VAE is particularly effective or ineffective?