GAN + VAE

MATH 270

Pol Diéguez Pastallé, Aniruddh Venkatesan, Prastik Mohanraj

November 3rd, 2025

Generative Adversarial Networks (GANs)

Train *two* networks: **Generators** and **Discriminators**

- Generator: Goal is to learn how to generate images matching training data (fake images)
- Discriminator: Goal is to learn how to distinguish between images coming from training data (real) and images generated by the generator

Generator produces a new image, and is ‘rewarded’ if the discriminator classifies it as ‘real’.

Discriminator is trained on **both** generated images and real images; penalized when it makes a mistake (i.e classifies real image as fake or vice versa).

GANs cont’d

This is a **zero sum game:** whenever the discriminator is penalized, the generator is rewarded.

Goal: Equilibrium is achieved - the generator creates images that look so real that the discriminator has a 50% chance of categorizing the image correctly (i.e a random guess).

$$\begin{aligned}\mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) &= \mathbb{E}_{y\sim p_{\text{data}}(y)}[\log D_Y(y)] \\ &+ \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log(1 - D_Y(G(x))]\end{aligned}$$

Weaknesses of GANs

Training two models leads to some more difficulties:

- Vanishing Gradients: The discriminator learns how to tell real vs fake ‘too fast’
- Mode Collapse: The generator only generates a specific type of data from dataset
- Training Instability: Training process can be unstable, and does not converge, leading to cycles/loops

Radford et. al (2015) on DCGANs

“Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks”

- Introduce new kind of GAN, called Deep Convolutional Generative Adversarial Networks (DCGANs)
    - Change architecture of GANs, making them stable to train

- Latent space of generator learns meaningful semantic representation
    - Leads to ‘vector arithmetic’ for inputs

Architectural Improvements

Four main architectural improvements:

- Replace all pooling layers with **strided convolutions in the discriminator** and **fractional-strided convolutions in the generator.**
- Use **Batch-Normalization** in both the **generator and discriminator.**
- Removing fully connected hidden layers to make both networks fully convolutional
- Use ReLU activation in the generator (with tanh output) and **LeakyReLU** activation in the discriminator

Example architecture

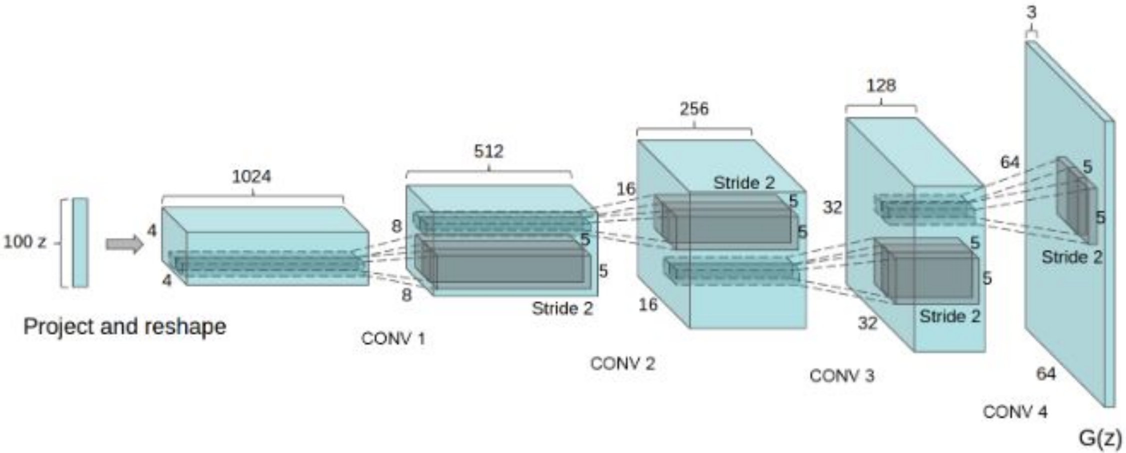


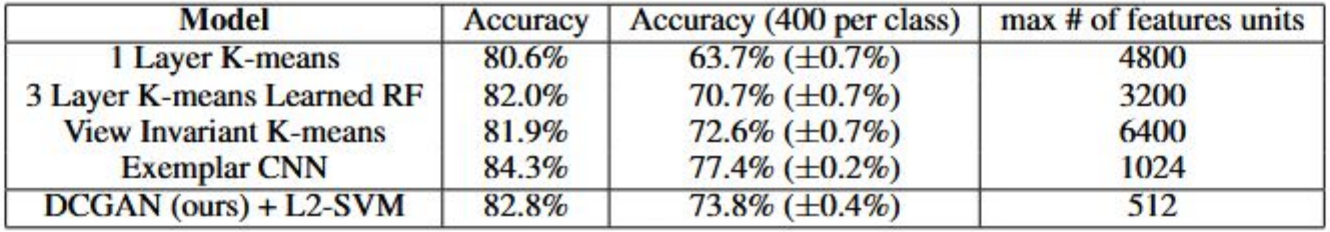
Figure: DCGAN architecture from Radford et. al.

Example DCGAN architecture

Experiments/Results

Common way to assess unsupervised learning models: apply them as feature extractors on supervised datasets, observe performance of linear models fitted on these features.

Trained on ImageNet-1k database, used to classify images on CIFAR-10



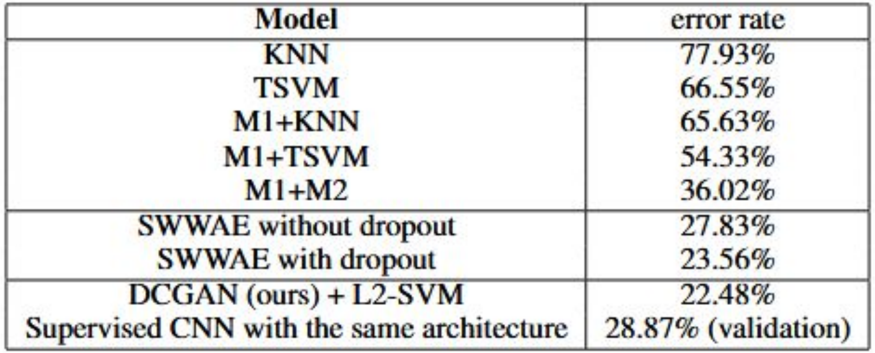
Model	Accuracy	Accuracy (400 per class)	max # of features units
1 Layer K-means	80.6%	63.7% (±0.7%)	4800
3 Layer K-means Learned RF	82.0%	70.7% (±0.7%)	3200
View Invariant K-means	81.9%	72.6% (±0.7%)	6400
Exemplar CNN	84.3%	77.4% (±0.2%)	1024
DCGAN (ours) + L2-SVM	82.8%	73.8% (±0.4%)	512

Figure: Taken from Radford et. al.

Experiments/Results, cont’d.

Classifying SVHN (Street View House Numbers) using DCGANs as Feature extractors

Table 2: SVHN classification with 1000 labels



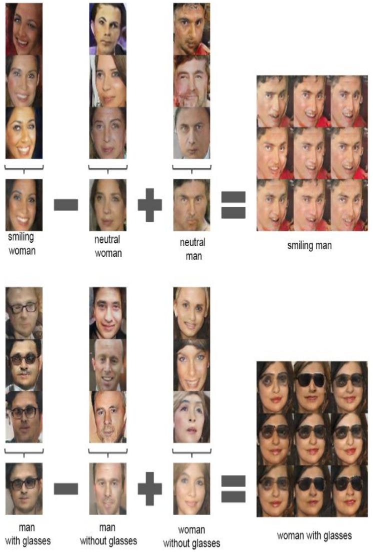
Model	error rate
KNN	77.93%
TSVM	66.55%
M1+KNN	65.63%
M1+TSVM	54.33%
M1+M2	36.02%
SWWAE without dropout	27.83%
SWWAE with dropout	23.56%
DCGAN (ours) + L2-SVM	22.48%
Supervised CNN with the same architecture	28.87% (validation)

Key Takeaways and Contributions

Semantic Vector Arithmetic in Latent Spaces

- Idea: “King - Man + Woman ‘=’ Queen”

Latent Space of Generator is more structured than previously expected.



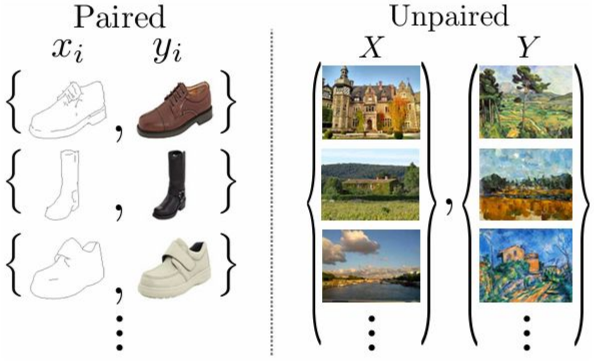
Zhu et. al. (2017) on CycleGANs

“Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks”

How do we deal with training on data where a paired training set is not available?

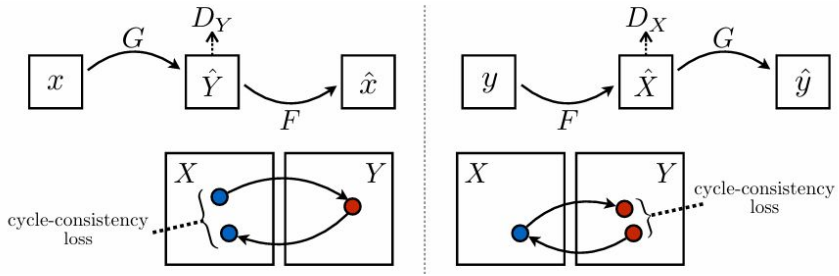
Train a G:X→Y, s.t. ŷ = G(x), x ∈ X,
is indistinguishable from images y ∈ Y

In practice, this does not guarantee that x and y are paired in a meaningful way (mode collapse)



Approach

Add a cycle-consistency loss that encourages F(G(x)) ≈ x and G(F(y)) ≈ y



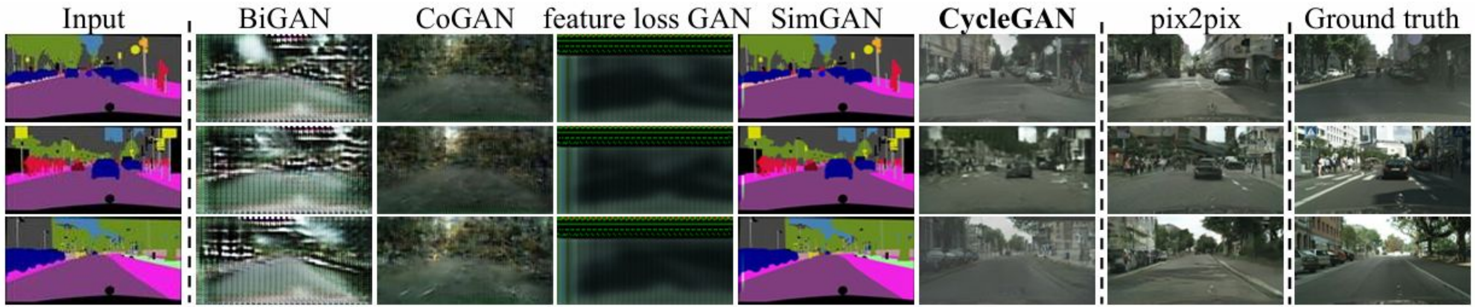
$$\begin{aligned}\mathcal{L}_{\text{cyc}}(G,F) =& \mathbb{E}_{x\sim p_{\text{data}}(x)}[\|F(G(x))-x\|_1] \\ &+\mathbb{E}_{y\sim p_{\text{data}}(y)}[\|G(F(y))-y\|_1].\end{aligned}$$

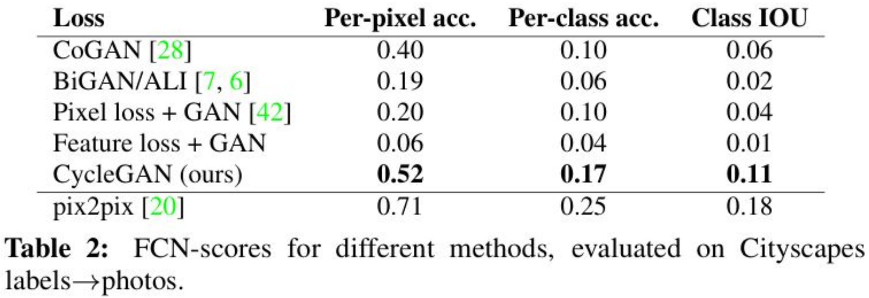
$$\begin{aligned}\mathcal{L}_{\text{GAN}}(G,D_Y,X,Y) =& \mathbb{E}_{y\sim p_{\text{data}}(y)}[\log D_Y(y)] \\ &+ \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log(1-D_Y(G(x))]\end{aligned}$$

$$\begin{aligned}\mathcal{L}(G,F,D_X,D_Y) =& \mathcal{L}_{\text{GAN}}(G,D_Y,X,Y) \\ &+ \mathcal{L}_{\text{GAN}}(F,D_X,Y,X) \\ &+ \lambda\mathcal{L}_{\text{cyc}}(G,F),\end{aligned}$$

$$G^*,F^* = \arg\min_{G,F}\max_{D_x,D_Y}\mathcal{L}(G,F,D_X,D_Y)$$

Results compared to different methods



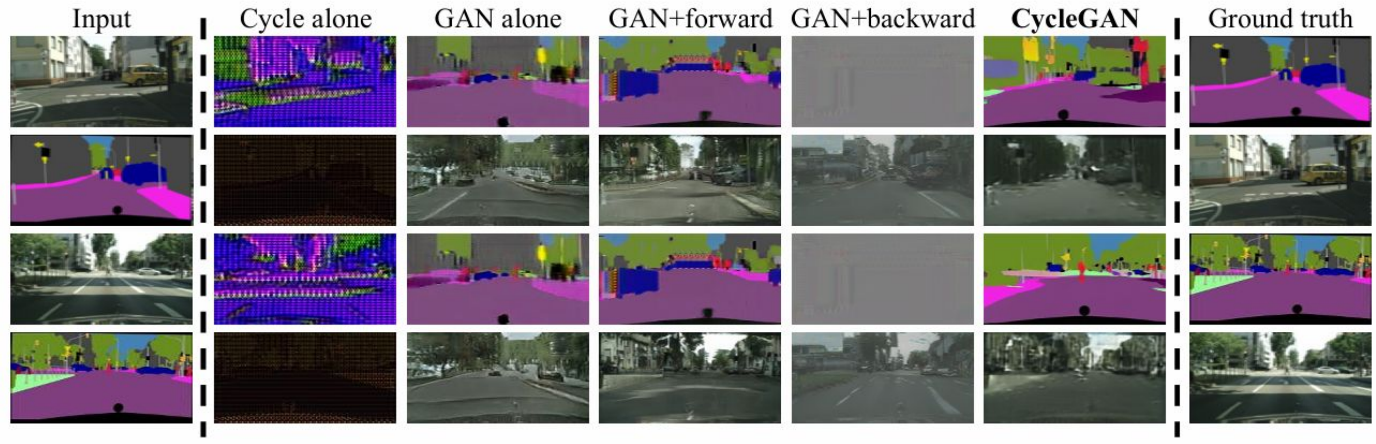


Loss	Per-pixel acc.	Per-class acc.	Class IOU
CoGAN [28]	0.40	0.10	0.06
BiGAN/ALI [7, 6]	0.19	0.06	0.02
Pixel loss + GAN [42]	0.20	0.10	0.04
Feature loss + GAN	0.06	0.04	0.01
CycleGAN (ours)	**0.52**	**0.17**	**0.11**
pix2pix [20]	0.71	0.25	0.18

Table 2: FCN-scores for different methods, evaluated on Cityscapes labels→photos.

- Pix2pix is trained on paired data, whereas the other methods are trained on unpaired data

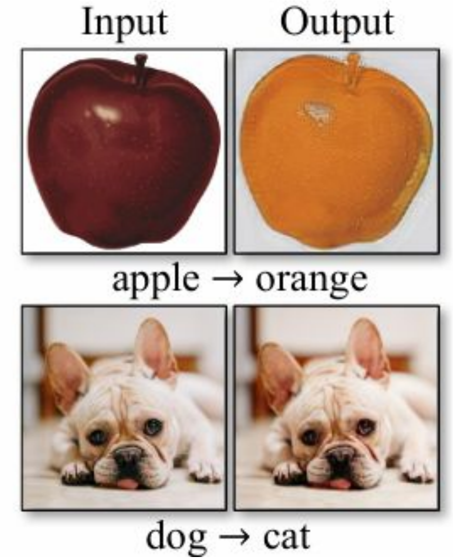
Results with different variants of CycleGAN



- Cycle alone and GAN+backward don’t produce images similar to target domain
- GAN alone and GAN+forward suffer mode collapse

Weakness

Results are good on texture changes, but do not succeed on geometric changes (e.g. Apple → Orange)



apple → orange

dog → cat

Variational Autoencoders (VAEs)

Generative Models and the Latent Space

- **Generative Models:** Aim to generate synthetic samples that look like they came from the unknown data distribution P.

- **Autoencoders (AE):** Learn a compressed representation in a latent space (Z) by approximating the identity function $X \rightarrow Z \rightarrow X$.

- **VAEs:** The latent space representation (Z) is explicitly probabilistic, allowing for smooth, meaningful generation via sampling.

Variational Autoencoders: The Architecture

- **The Encoder ($q_{\Phi}(z \mid x)$):** Maps input data x to a conditional probability distribution over the latent space Z.

- **The Decoder ($p_{\theta}(x \mid z)$):** Takes a sampled latent vector z and maps it back to the data space X.

- **Reparameterization Trick:** To allow gradient flow through the sampling process (backpropagation), the latent vector z is computed as $z = μ + σε$, where $ε \sim N(0, Id)$ is a standard Gaussian noise vector.

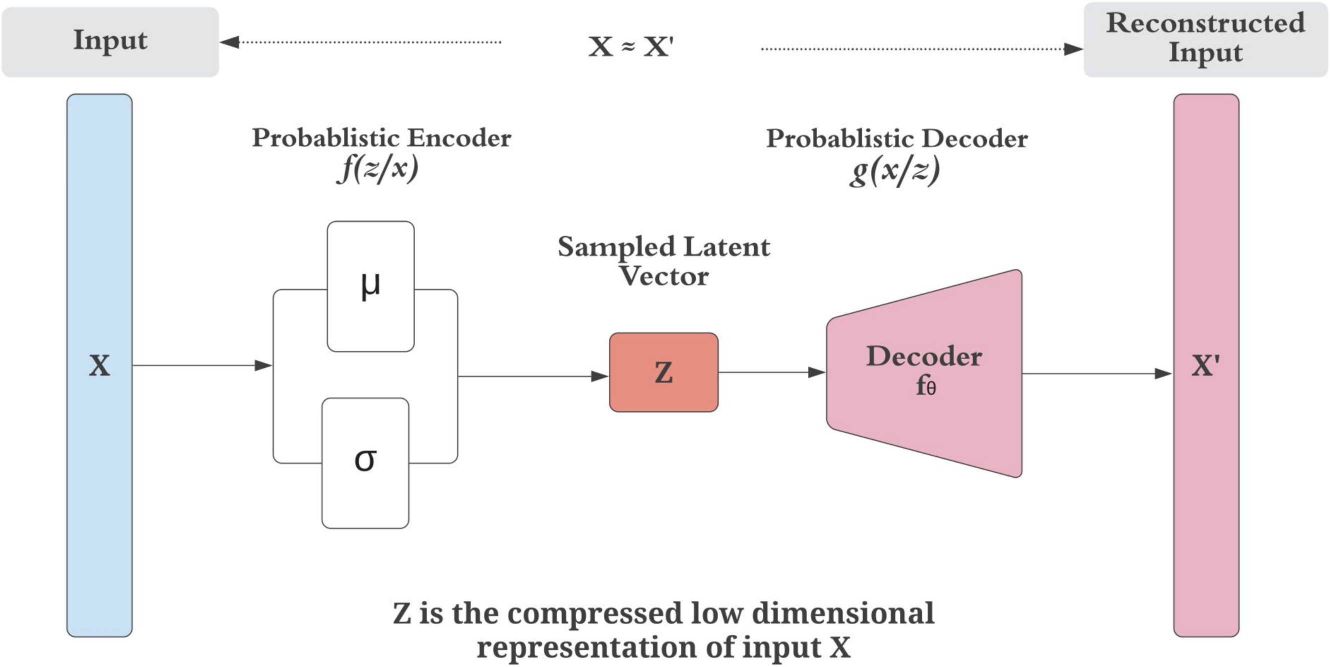


Fig. 5, Joshi et al. (2022), *Meta-Learning, Fast Adaptation, and Latent Representation for Head Pose Estimation*

The VAE Objective: Evidence Lower Bound (ELBO)

- **Goal:** Maximize the Evidence Lower Bound (ELBO), a proxy for maximizing the data log-likelihood (log $p_\theta(x)$).
- **ELBO Loss Function:**

$$\mathrm{ELBO}(x|p_\theta, q_\phi) = \mathbb{E}_{q_\phi}[\log p_\theta(x|z)] - \mathrm{KL}(q_\phi(z|x) \mid\mid p(z))$$

- **Term 1: Reconstruction Loss:** Ensures the decoder can accurately reproduce the input *x*.
- **Term 2: Regularization Loss:** The Kullback–Leibler (KL) Divergence forces the learned posterior to stay close to a simple prior *p*(*z*) (typically a standard Gaussian N(0, Id)), ensuring a smooth and continuous latent space.

β-VAEs (Higgins et al., 2016)

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

The Need for Disentanglement

- **The Problem with Standard VAEs:** VAEs often suffer from posterior collapse, where the encoder ignores the input *x* and the decoder always generates the same image, *or* they learn an entangled latent representation.
- **The Goal:** Learning a truly interpretable factorized representation of independent generative factors (e.g., position, scale, color) without supervision.
- **Disentangled Representation:** A single latent unit is sensitive to a single generative factor while being invariant to all others (e.g., one latent unit controls only rotation, another controls only color).

The β-VAE Formulation

- **The Modification:** The β-VAE introduces a single adjustable hyperparameter β > 1 that modulates the balance between the two loss terms.
- **Loss Function:**

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

- **Effect of β:** By setting β > 1, the model increases the weight on the regularization term (the KL Divergence) relative to the reconstruction loss.

β's Mechanism: Constrained Latent Capacity

- **Information Bottleneck:** β > 1 imposes a stricter limit on the latent information channel capacity.

- **Increased Pressure:** By heavily penalizing the KL Divergence term, the encoder is forced to find the most statistically independent and salient features in the data to encode (it must use the limited latent capacity efficiently).

- **The Trade-off:** High β leads to better disentanglement and a smoother latent space, but can come at the cost of reduced reconstruction quality (blurry images) because of the limited capacity.

Qualitative Results: Latent Traversal

- **Superior Performance:** β-VAE with appropriately tuned $\beta > 1$ qualitatively outperforms VAE ($\beta = 1$) and other state-of-the-art models (InfoGAN, DC-IGN).

- **Key Demonstration: Latent Traversal Plots.**
  - Systematically varying a single latent unit while keeping all others fixed.
  - The generated image shows a smooth, controlled change in only one visual concept (e.g., object rotation, scale, or azimuth).

- **Datasets:** Disentanglement demonstrated across complex datasets like CelebA (faces), 3D Faces, and 3D Chairs.

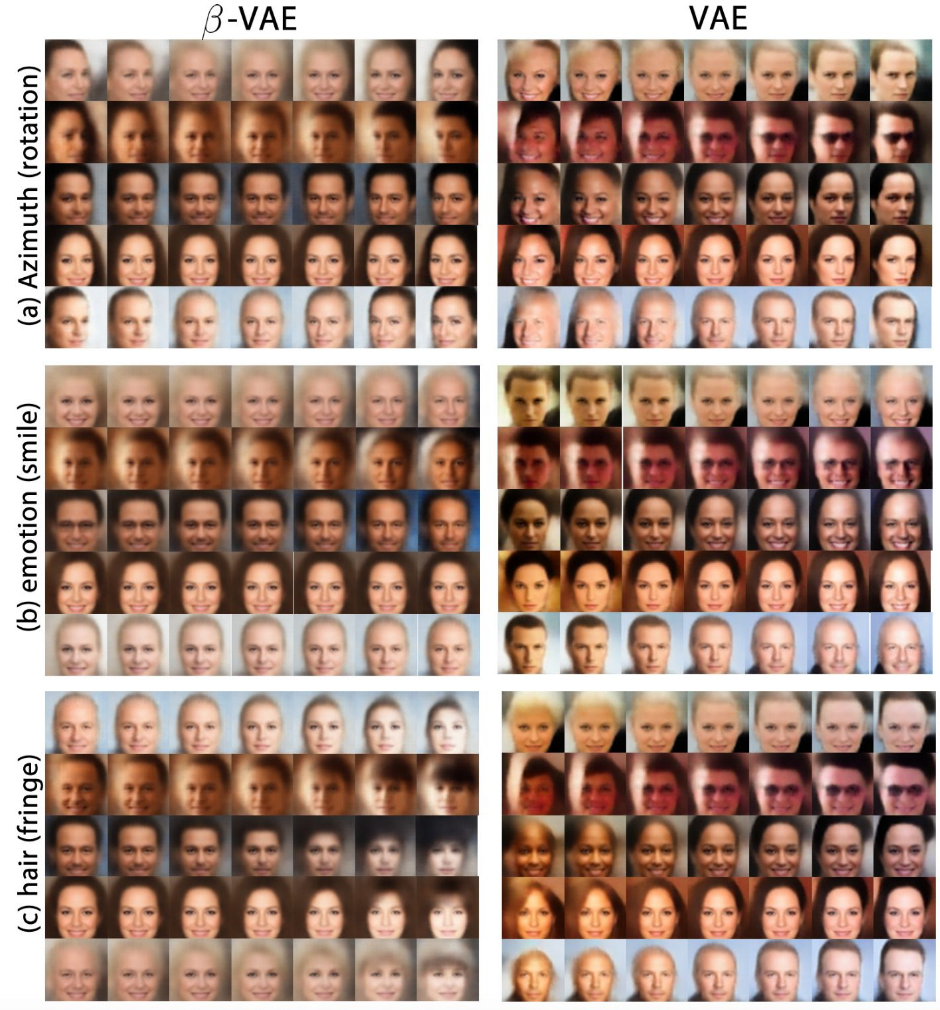


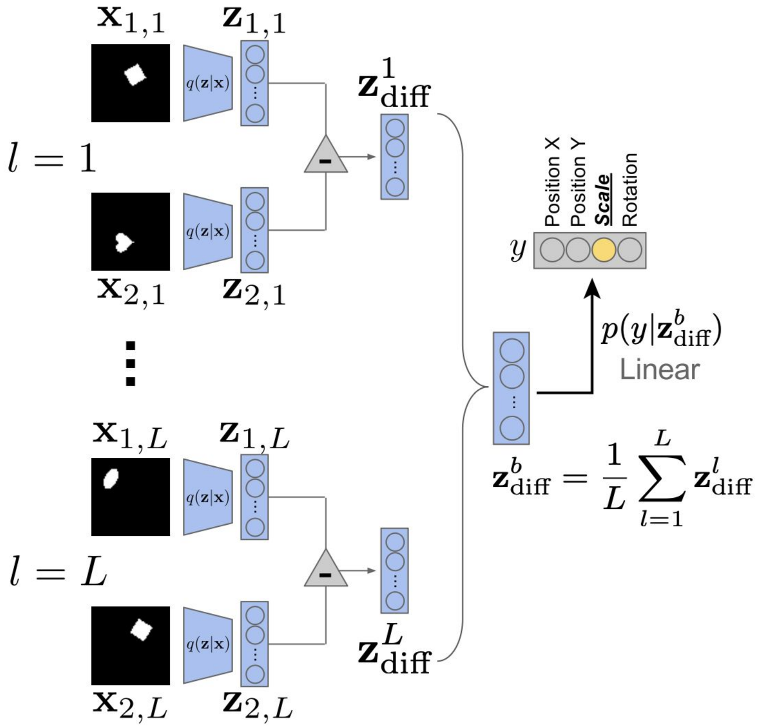
Fig. 1, Higgins et al.

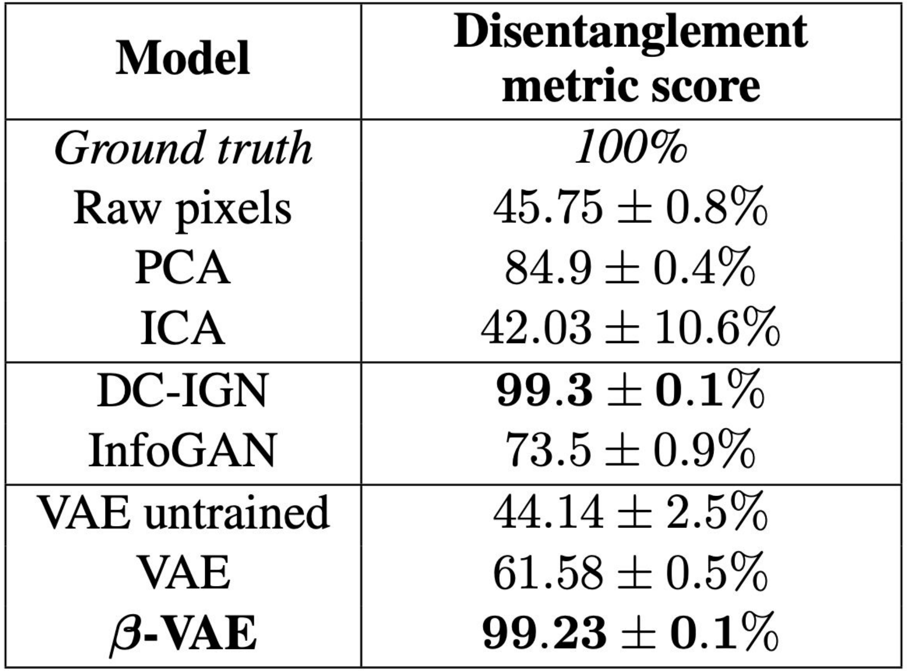
A Quantitative Metric for Disentanglement

- **The Challenge:** Disentanglement is typically measured by heuristic visual inspection (latent traversal), which is subjective and non-scalable.

- **The Method:** After the β-VAE is trained, a simple linear classifier is trained to predict the ground-truth value of each generative factor from the latent representation.

- **Interpretation:** If a single latent unit perfectly predicts a single ground-truth factor, that factor is considered linearly separable and thus well-disentangled. The metric is the classification accuracy.





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	$\mathbf{99.3} \pm \mathbf{0.1}\%$
InfoGAN	$73.5 \pm 0.9\%$
VAE untrained	$44.14 \pm 2.5\%$
VAE	$61.58 \pm 0.5\%$
β-VAE	$\mathbf{99.23} \pm \mathbf{0.1}\%$

Figs. 5 + 6, Higgins et al.

Conclusion: Key Takeaways

- **β-VAE** builds on the **VAE** framework to successfully enforce a disentangled and interpretable latent representation.

- **Mechanism:** The single hyperparameter $\beta > 1$ acts as an effective constraint on the latent channel capacity, forcing the network to learn only the most independent and relevant factors.

- **Impact:** Disentangled representations are a crucial step toward general AI, boosting capabilities for knowledge transfer, zero-shot inference, and reasoning.

Colab and References

Colabs:

- DCGAN
- VAE

Papers:

- Radford et. al, Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. ICLR 2015.
- Zhu et. al, Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. 2017 IEEE International Conference on Computer Vision (ICCV).
- Higgins et. al, β-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. ICLR 2016.

Thanks for listening!

Questions?