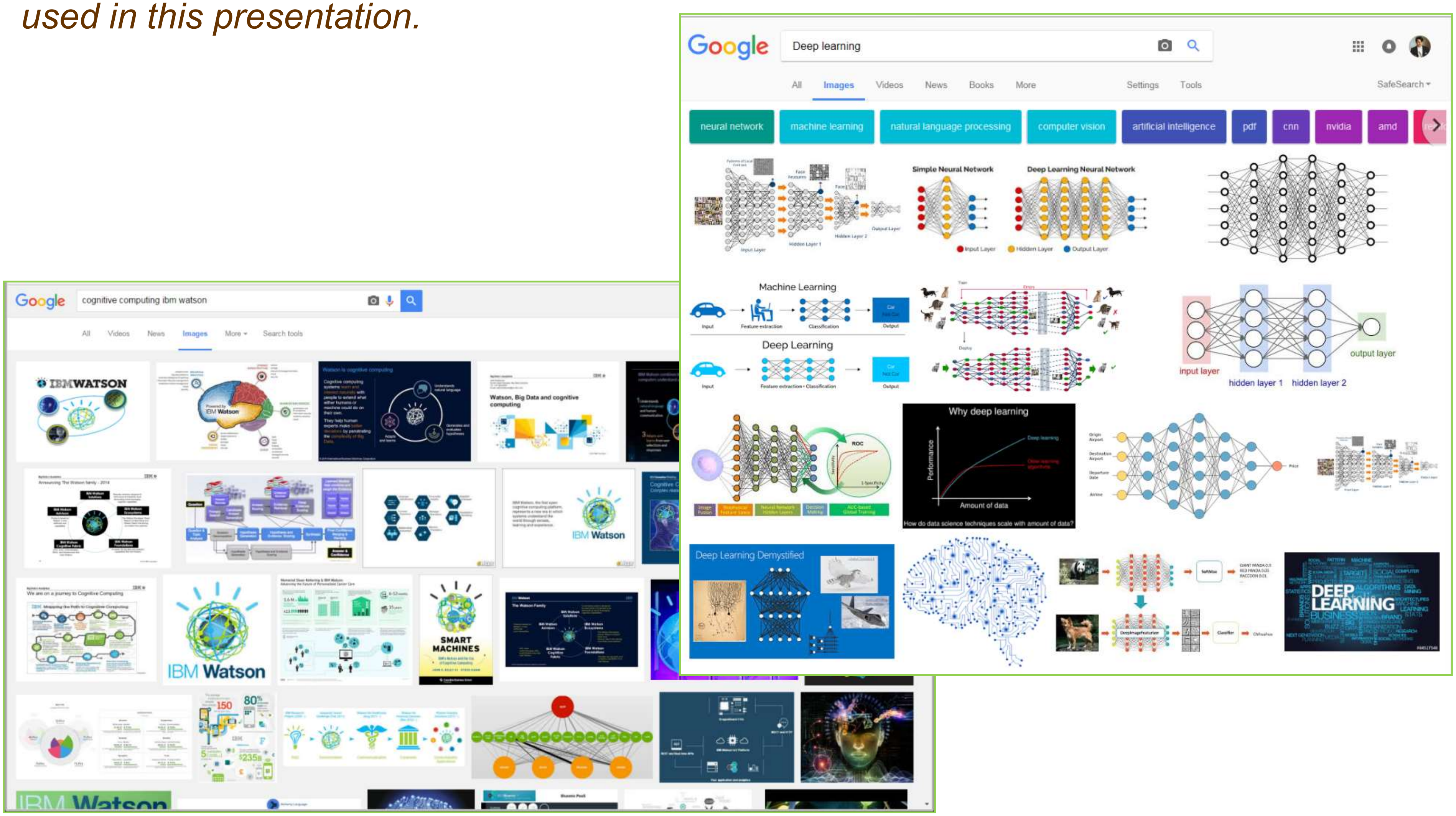
Lecture 9: Generative models

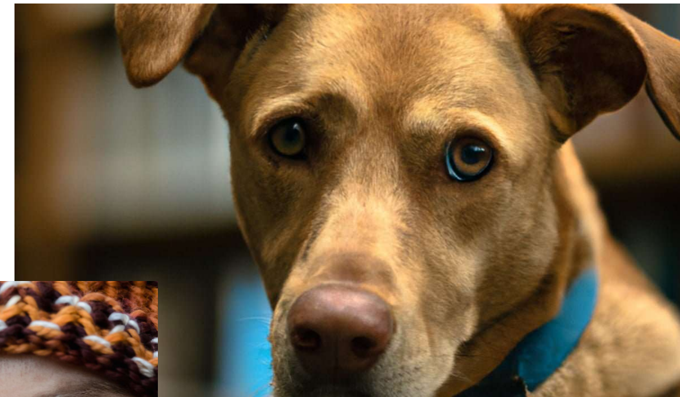
TIES4911 Deep-Learning for Cognitive Computing for Developers
Spring 2024

by:
Dr. Oleksiy Khriyenko
IT Faculty
University of Jyväskylä

Acknowledgement

I am grateful to all the creators/owners of the images that I found from Google and have used in this presentation.





Different classes of problems

Supervised vs. **Unsupervised**

Supervised

Inputs: *(x , y)*

data → x, label → y

Goal: Learn *function to map x* to *y*

- Regression
- Classification
- Object detection
- Sentiment segmentation
- Feature learning (with labels)
- etc.

Unsupervised

Inputs: *(x)*

data → x, no label

Goal: Learn some *hidden* or *underlying structure* of the data

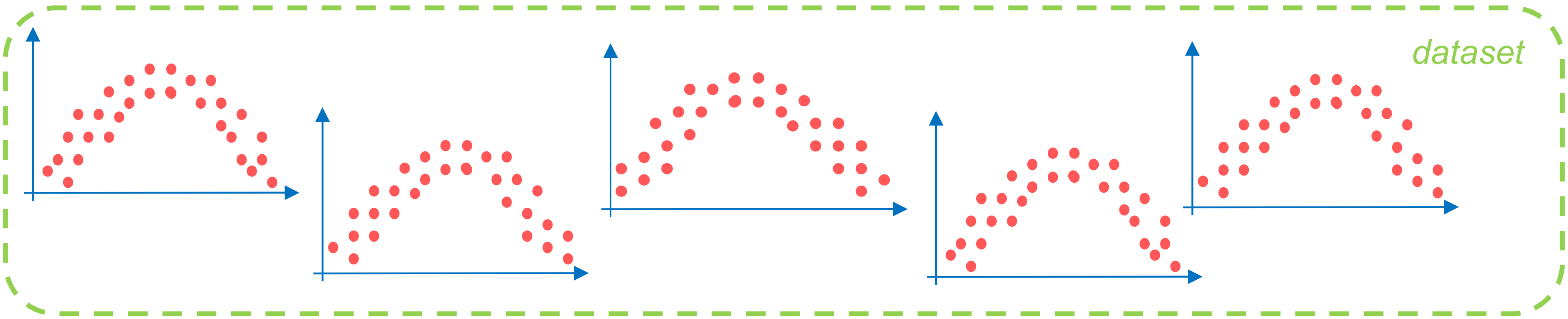
- Feature extraction/learning (without labels)
- Clustering
- Dimensionality reduction
- Density estimation
- etc.

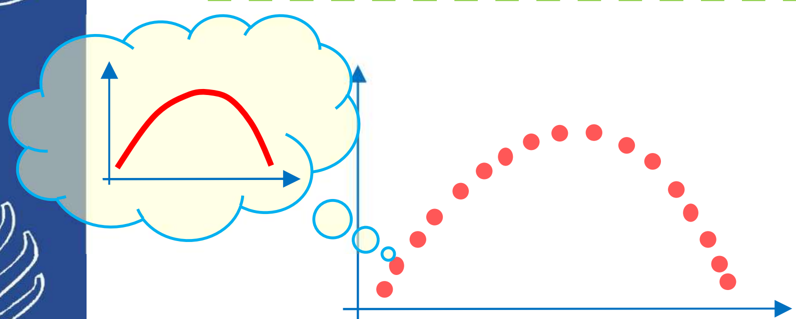
Models…

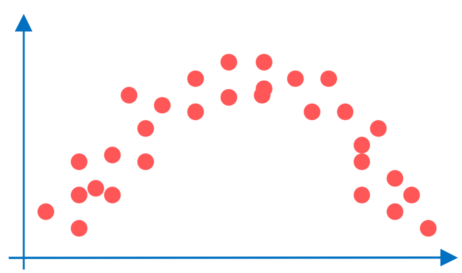
Discriminative vs. **Generative**

It tells us what the data is…
It discriminates, differentiates, classifies…

It generates a new data: new images, new video, new texts, new music, etc.







Infinite number of possible options...

Learns a *boundary between the classes…*
p(y|x) = probability of y given x

Learns a *distribution of individual classes…*
p(x) = probability distribution of x
p(x|y) = probability of x given y (conditional model)

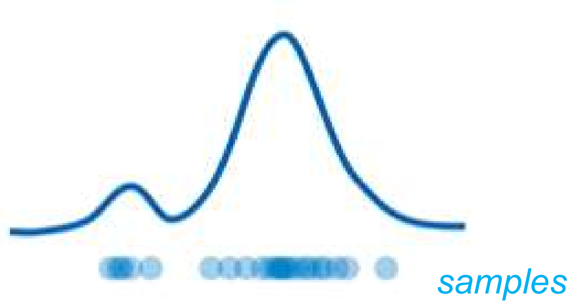
Goals of generative modeling

Generative modeling…

… taking training samples from some distribution as an input, learn a model that represents that distribution.

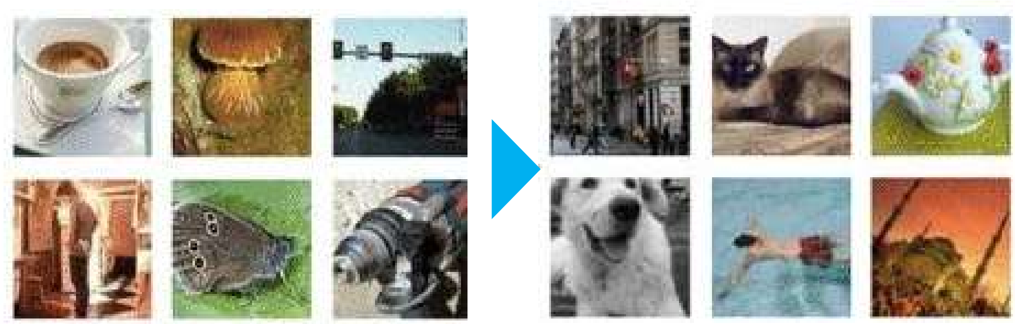
Density Estimation

Describes where the data was drawn from…

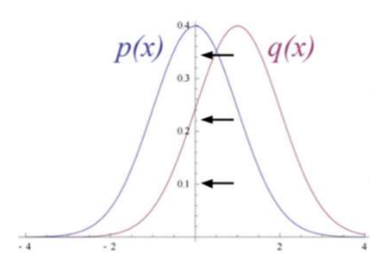


Samples Generation

Learn (model) probability distribution similar to the true distribution that describes how the data was generated…

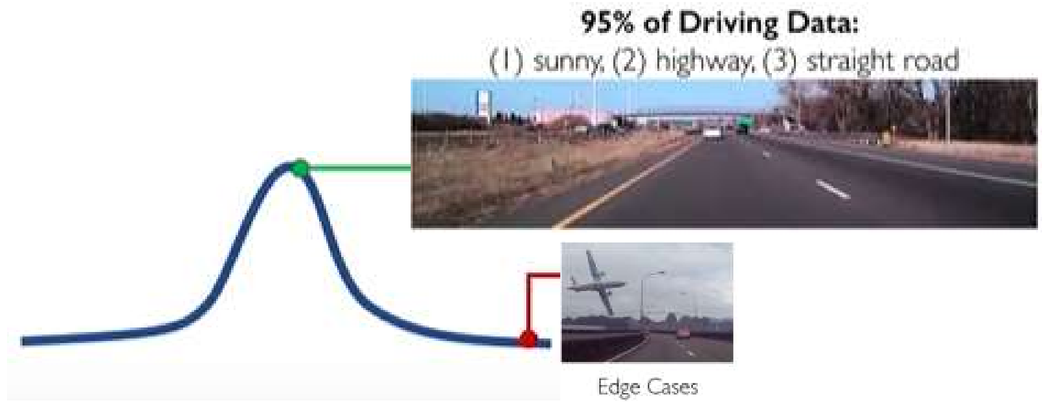


Training data $\sim P_{data}(x)$ Generated data $\sim P_{model}(x)$

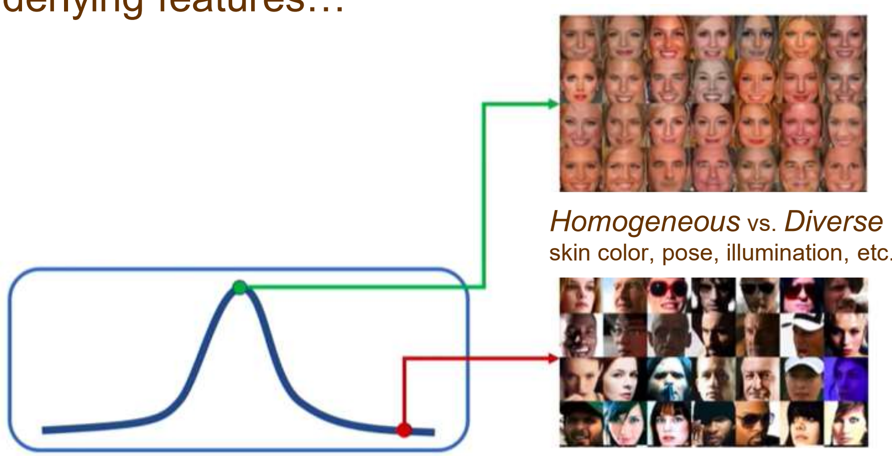


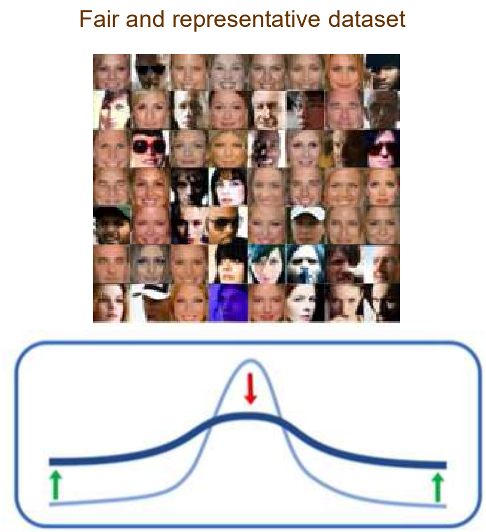
Generative modeling

Outlier detection… leveraged generative model enables outliers detection in the distribution. Use of outliers during the training helps to improve the model enable to detect something new or rare…

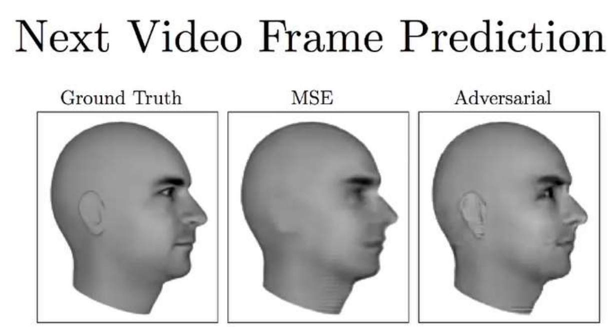


Debiasing… helps to create fairly representative dataset by uncovering underlying features…

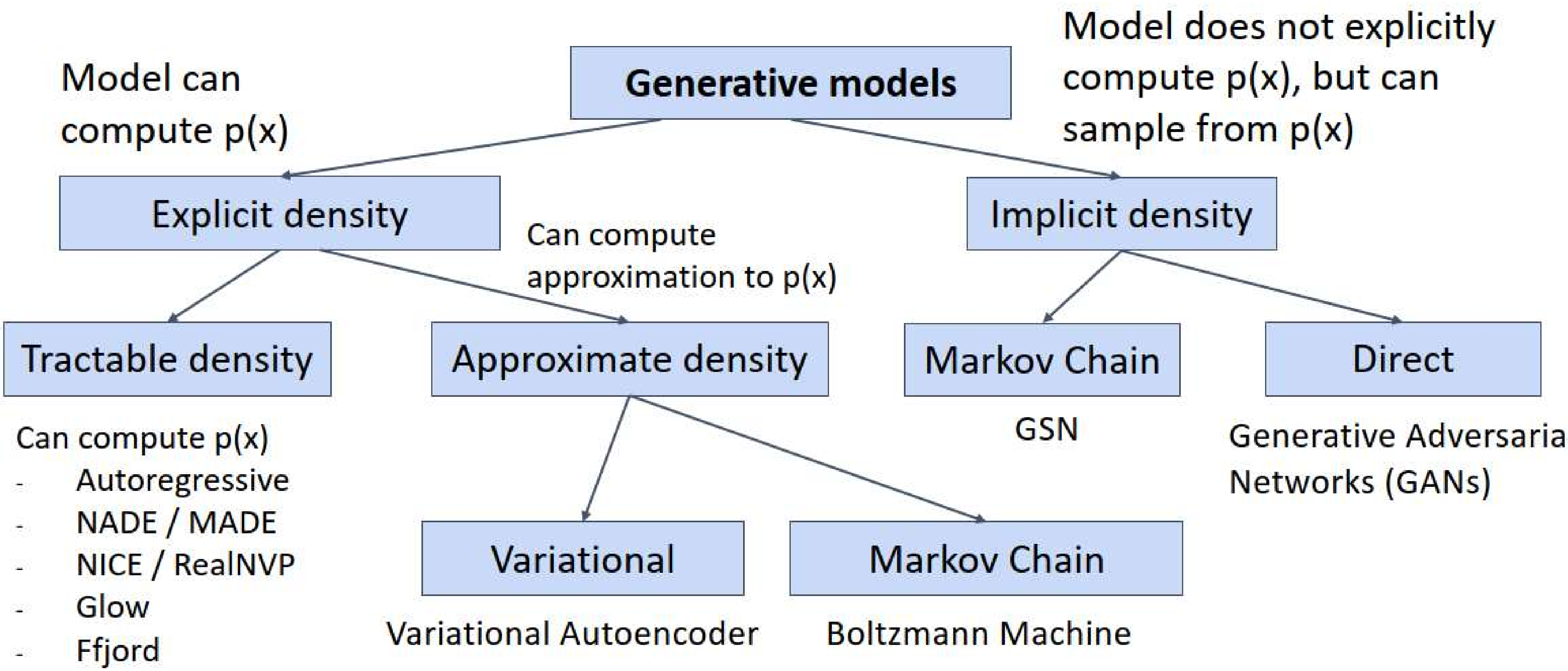




New data generation… helps to simulate possible futures for planning or simulated Reinforcement Learning, allows to fill the gap of missing data, supports in realistic generation tasks, etc.



Taxonomy of Generative Models



Relevant links:
https://www.youtube.com/watch?v=5WoItGTWV54
https://www.youtube.com/watch?v=9JpdAg6uMXs
https://channel9.msdn.com/Events/Neural-Information-Processing-Systems-Conference/Neural-Information-Processing-Systems-Conference-NIPS-2016/Generative-Adversarial-Networks

Autoregressive Generative Modeling

PixelRNN and *PixelCNN* are explicit density models for fully visible belief networks that use chain rule to decompose likelihood of an image x into product of 1d distribution, and then maximize likelihood of training data…

Links: https://arxiv.org/abs/1601.06759
https://arxiv.org/abs/1606.05328

this complex distribution over pixel values is expressed via a neural network…

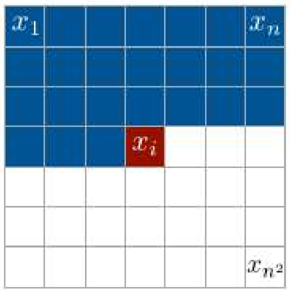
$$p(x) = \prod_{i=1}^{n} p(x_i | x_1, ..., x_{i-1})$$

Likelihood of image x

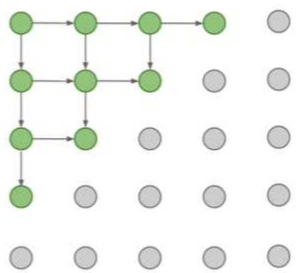
Probability of i'th pixel value given all previous pixels

PixelRNN

- Generates image pixels starting from corner
- Dependency on previous pixels is modeled using an RNN (LSTM)



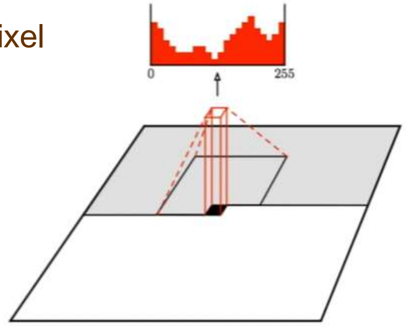
or



- Drawback: ***sequential generation is slow***

PixelCNN

- Generates image pixels starting from corner
- Dependency on previous pixels is modeled using a CNN over context region
- Softmax loss at each pixel



- Training is faster than PixelRNN (convolution parallelization)
- ***Generation*** must still proceed sequentially, therefore it is ***still slow***

Relevant links:
https://towardsdatascience.com/auto-regressive-generative-models-pixelrnn-pixelcnn-32d192911173
http://proceedings.mlr.press/v70/kolesnikov17a/kolesnikov17a.pdf

Autoregressive Generative Modeling

PixelRNN

ImageNet 32x32

PixelCNN

Coral Reef

Sorrel horse

Pros:
- Explicitly compute likelihood P(x)
- Explicit likelihood of training data gives good evaluation metric
- Good samples

Con:
- Sequential generation is slow

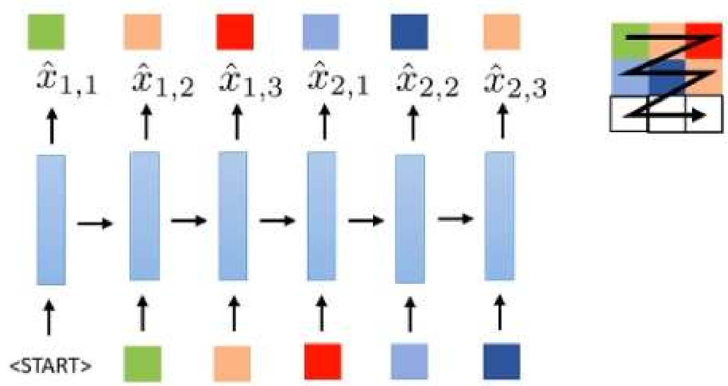
Improvement for PixelCNN:
- Gated convolutional layers
- Short-cut connections
- Discretized logistic loss
- Multi-scale
- Training tricks
- Etc.
- See: (Salimans et.al., 2017)

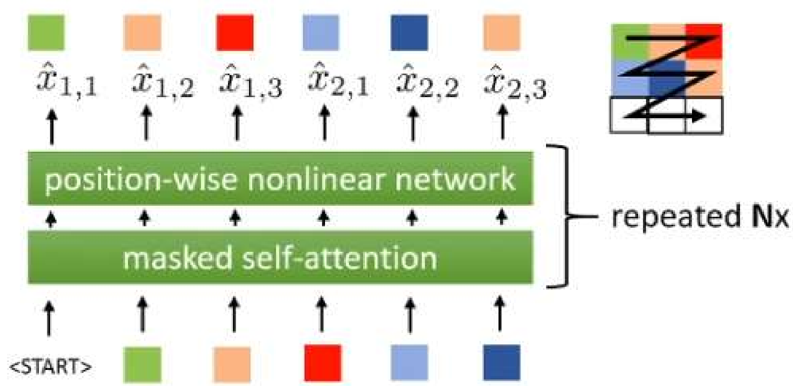
https://arxiv.org/abs/1701.05517
https://github.com/openai/pixel-cnn

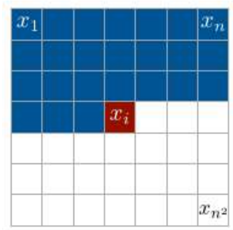
Autoregressive Generative Modeling

PixelTransformer replaces sequential model with Transformer Decoder style architecture with masked self-attention and position-wise nonlinear network. Without positioning embedding, self-attention model consider all the pixels equally close to each other.

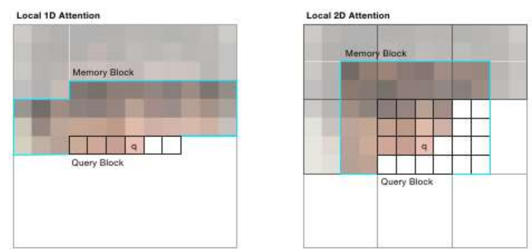
Links: https://arxiv.org/abs/1802.05751



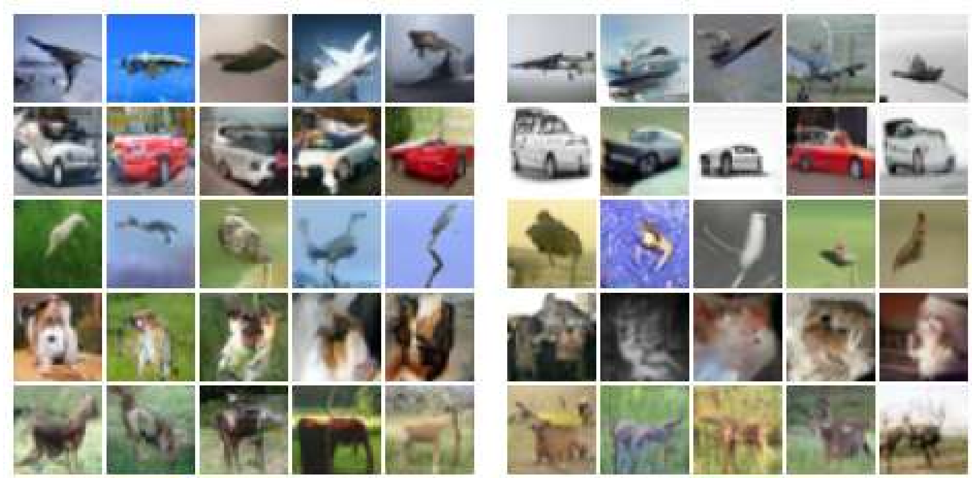




Without positioning embedding, self-attention model consider all the pixels equally close to each other. For example, in PixelRNN, pixel above is considered as very far pixel from target one.



For big images, number of pixels is huge and computation become very expensive. Solution is to compute attention based on smaller set of nearest pixels (similar to PixelCNN).



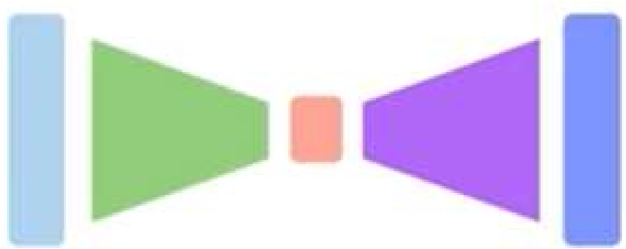
Relevant links:
https://www.youtube.com/watch?v=y380v-Mtvzo

Deep Generative Modeling

Latent variable models…

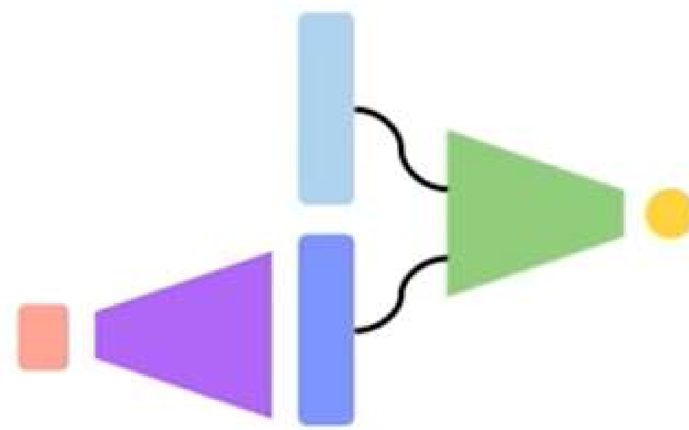
Autoencoders and Variational Autoencoders (VAEs)

Learn **lower-dimensional** latent space and **sample** to generate input reconstructions



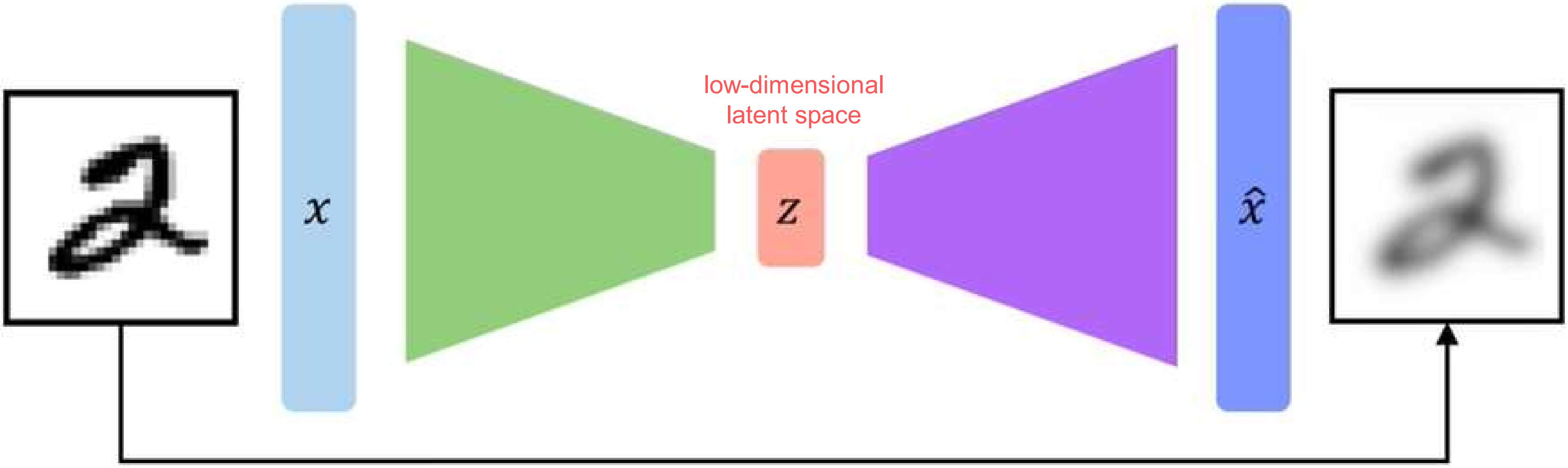
Generative Adversarial Networks (GANs)

Competing **generator** and **discriminator** networks

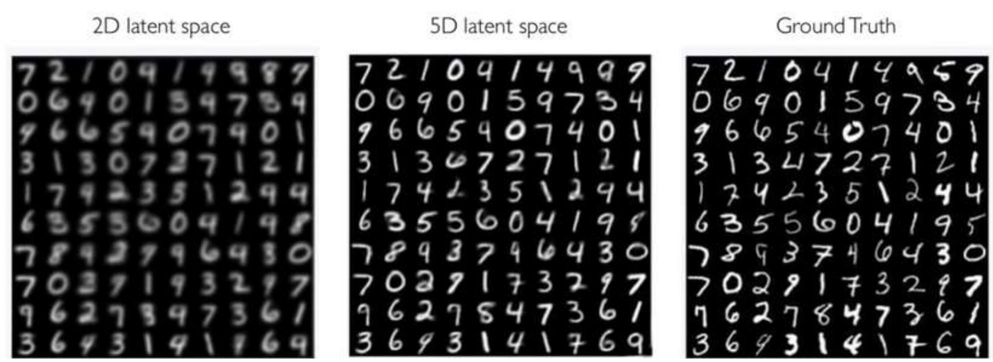


Autoencoders

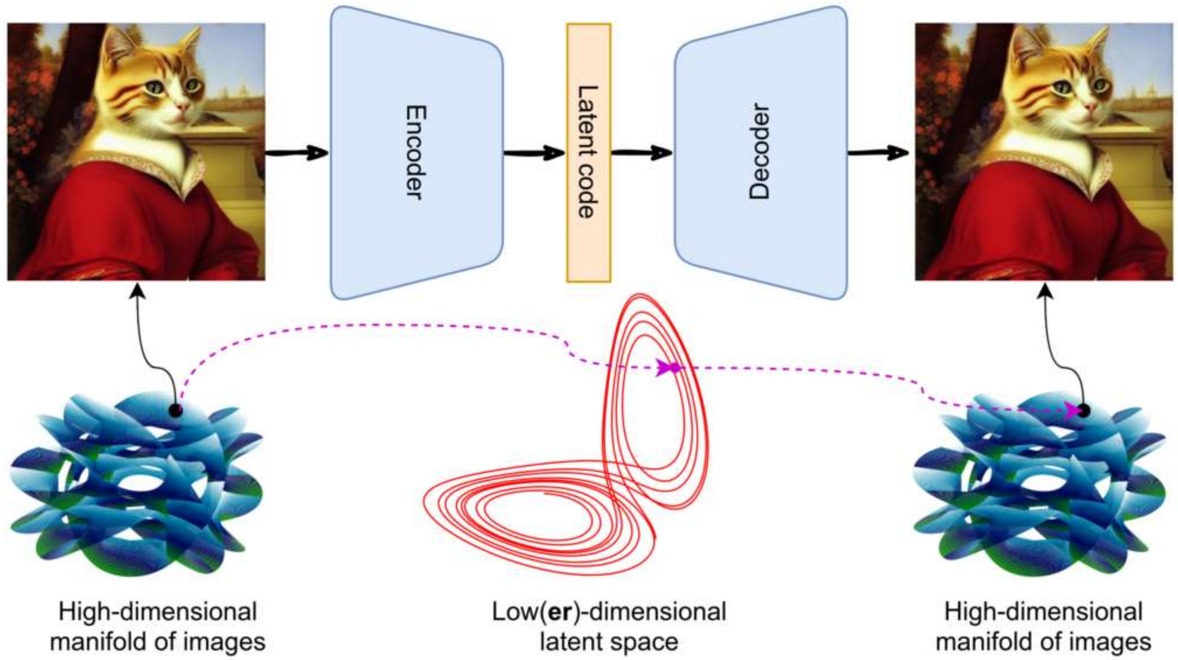
Learning ***a lower-dimensional feature representation*** from unlabeled training data…

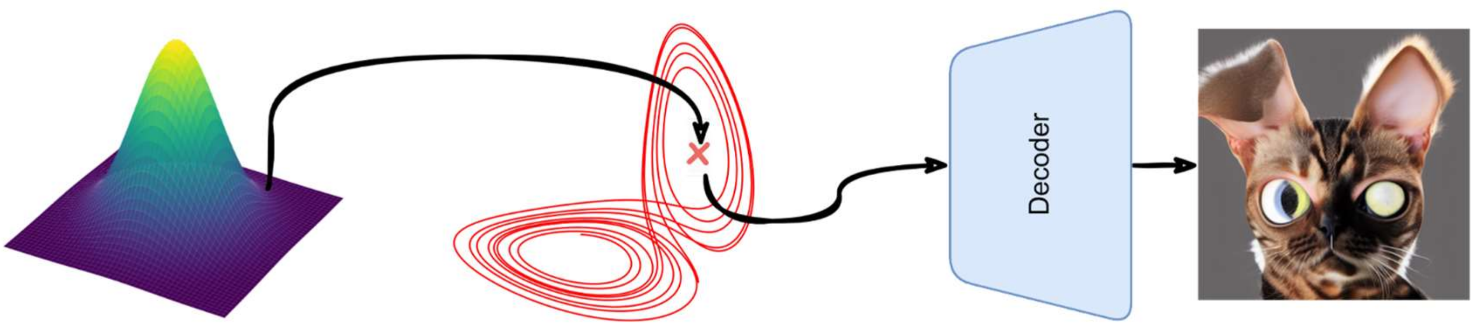


$$\mathcal{L}(x, \hat{x}) = \|x - \hat{x}\|^2$$



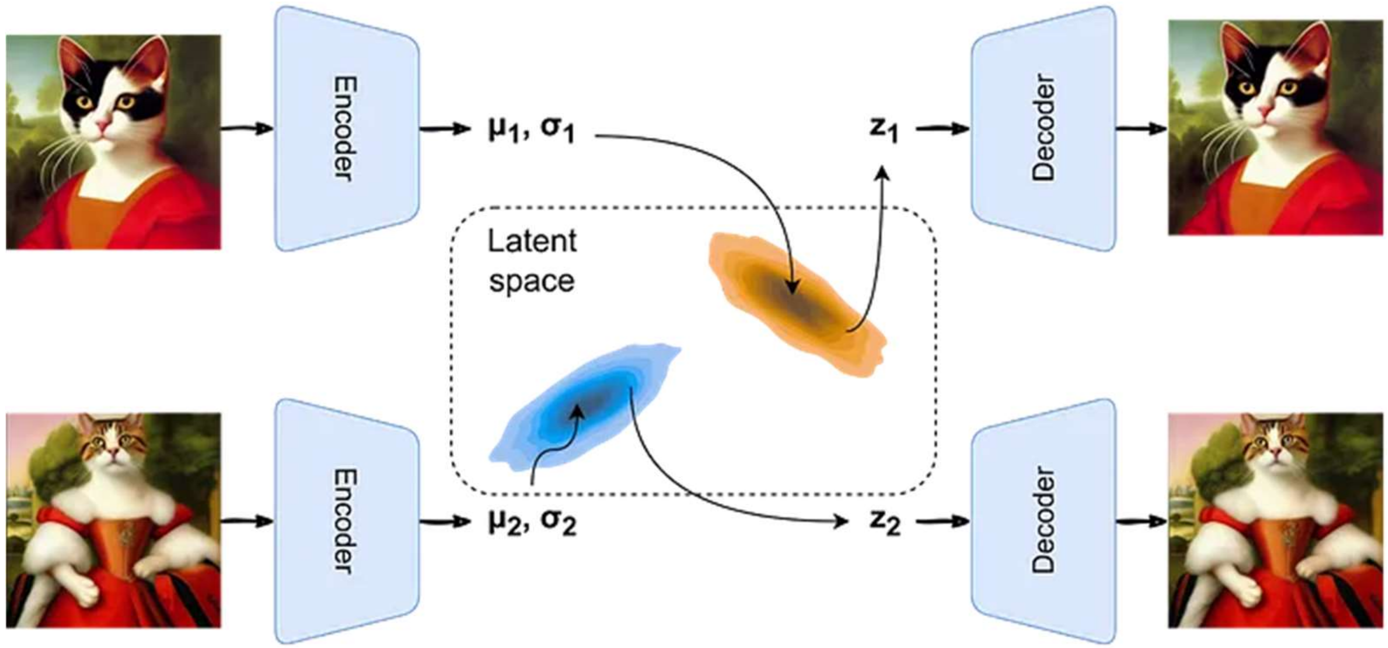
Autoencoders

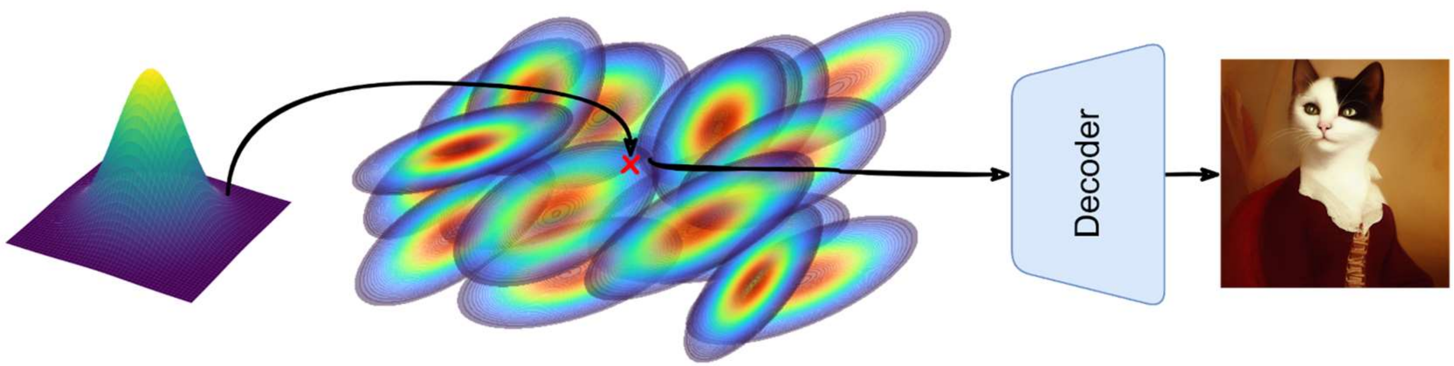




Relevant links:
https://synthesis.ai/2023/02/07/generative-ai-i-variational-autoencoders/

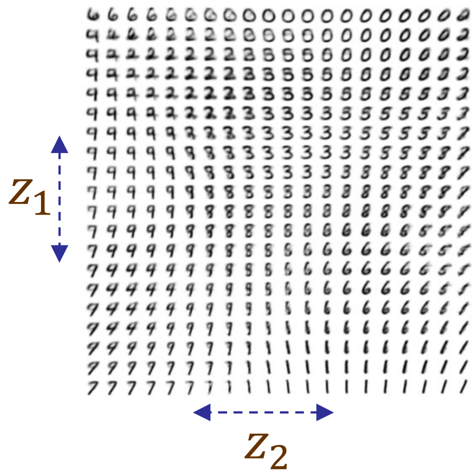
Variational Autoencoders (VAEs)

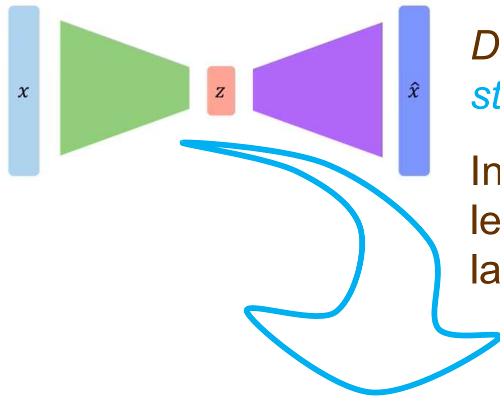




Relevant links:
https://synthesis.ai/2023/02/07/generative-ai-i-variational-autoencoders/

Variational Autoencoders (VAEs)



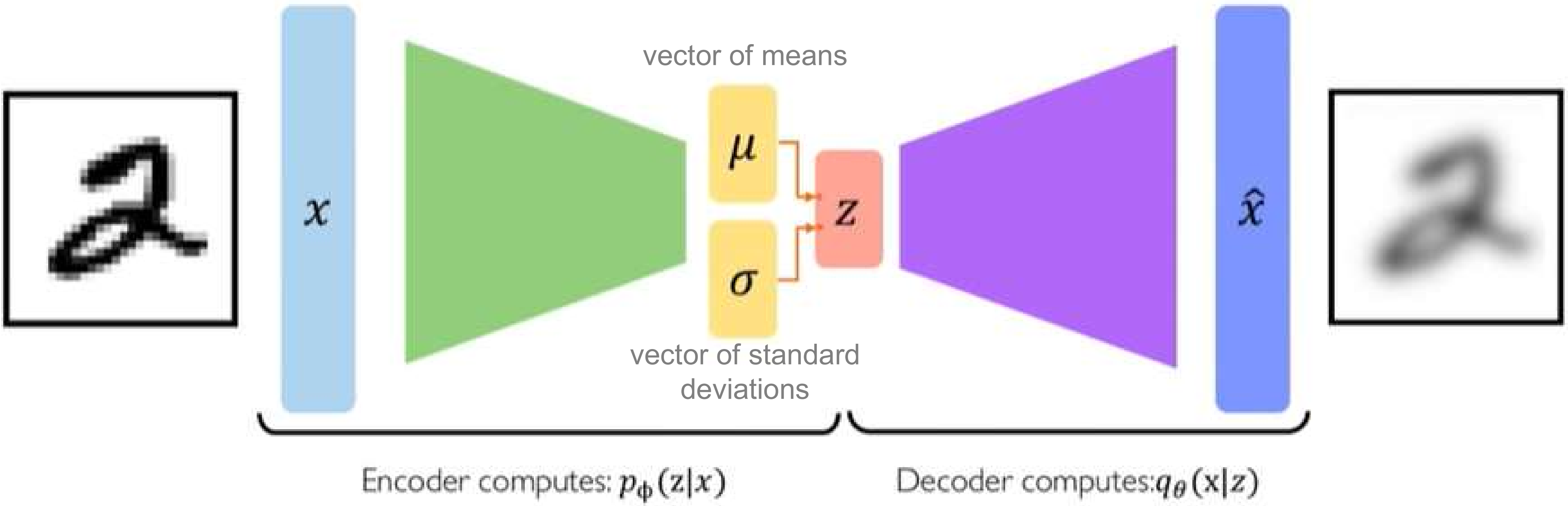


Deterministic bottleneck is replaced with *stochastic sampling* operation…

Instead of learning latent variables directly, learn *mean* and *standard deviation* for each latent variable separately…

Smooth representation of latent space

… both these vectors (μ and σ) describe probability distribution associated with each of the latent variables…



$$\mathcal{L}(\phi, \theta) = (\text{reconstruction loss}) + (\text{regularization term})$$

Variational Autoencoders (VAEs)

$$\mathcal{L}(\phi,\theta) = \underbrace{(\text{reconstruction loss})}_{Loss_{Reco}} + \underbrace{(\text{regularization term})}_{Loss_{KL}}$$

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

helps to reduce overfitting, encourages encodings to be distributed evenly around the center of the latent space and penalize the network when it tries to cluster points in specific regions memorizing the data.

$$D\left(q_\phi(z|x) \parallel p(z)\right) = -\frac{1}{2}\sum_{j=0}^{k-1}(\sigma_j + \mu_j^2 - 1 - \log\sigma_j)$$

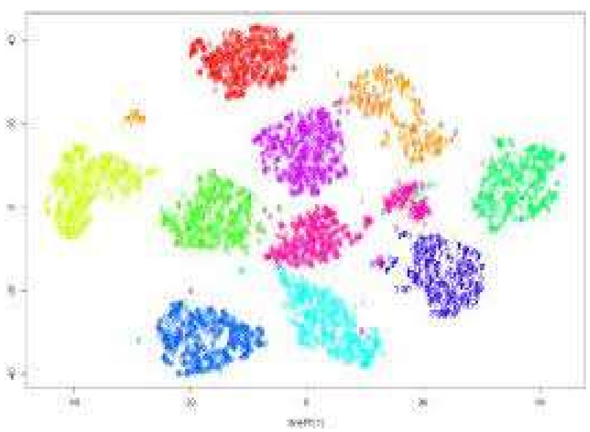
$Loss_{KL}$ - *KL (Kullback-Leibler) divergence between two distributions*

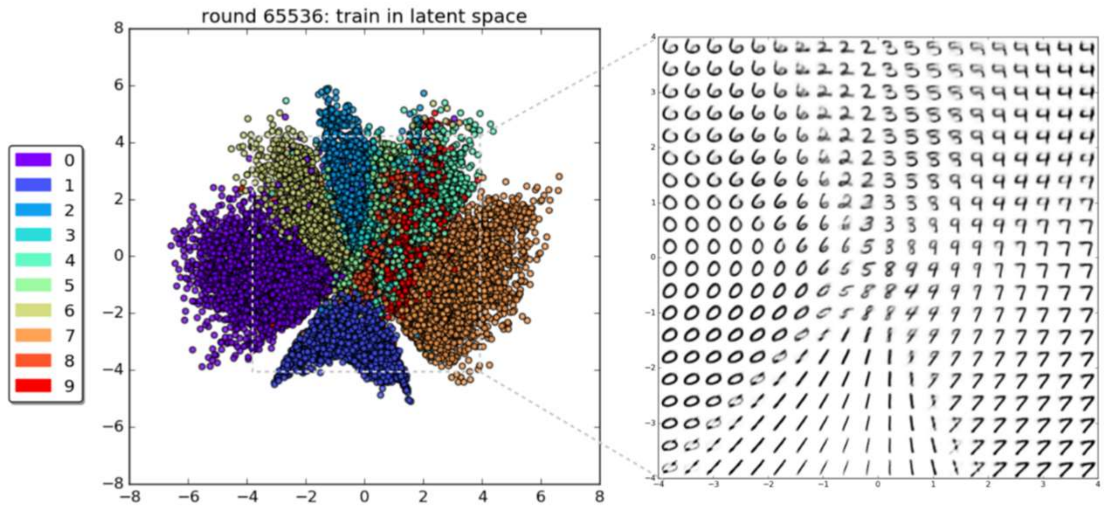
$$p(z) = \mathcal{N}(\mu = 0, \sigma^2 = 1)$$

Normal Gaussian

Regularization with Normal prior helps enforce information gradient in the latent space. $z \sim N(\mu, \sigma^2)$

It allows continuity (close points in latent space lead to similar decoded content) and completeness (decoded content is meaningful).





Not Regularized:
- *Small variances causes pointed distribution*
- *Different means lead to discontinuities*

Regularized:
- *Regularized variances*
- *Center means*

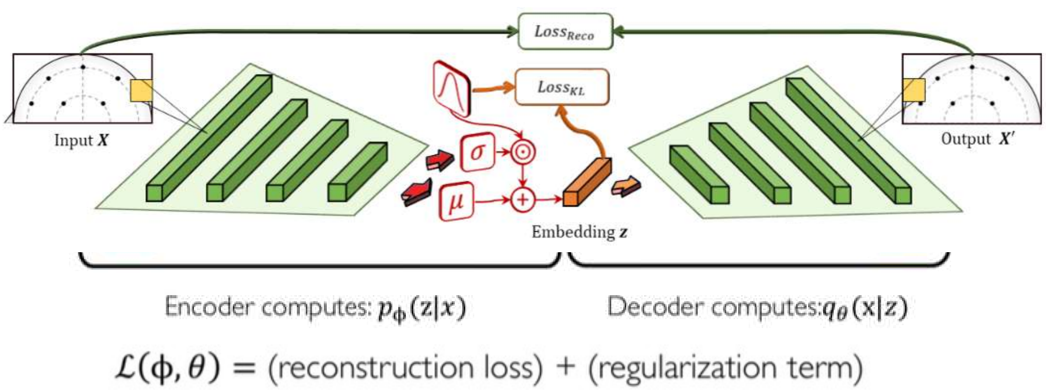
Variational Autoencoders (VAEs)

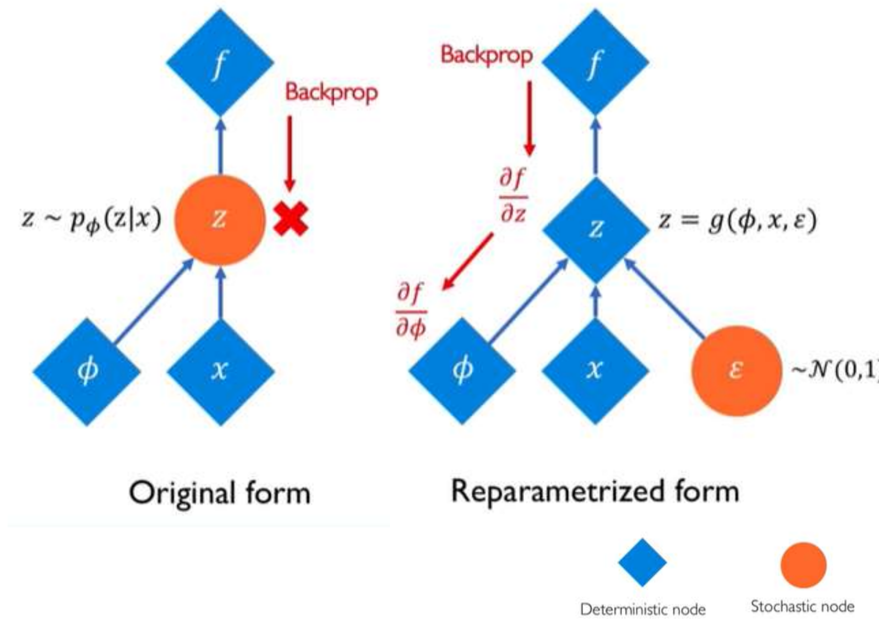
Problem: it is not possible to backpropagate gradients through sampling layer due to the *stochastic* nature of it *(z is a result of stochastic sampling operation)*…

It is impossible to integrate over all z ! $p_\theta(x) = \int p_\theta(x,z)dz = \boxed{\int} p_\theta(x|z)p_\theta(z)\boxed{dz}$

Solution: *re-parametrization of the sampling layer* $z \sim N(\mu, \sigma^2) \quad \Longrightarrow \quad z = \mu + \sigma \odot \varepsilon$

, where μ and σ are fixed vectors, and ε is random scaling constant drawn from the prior distribution ($\varepsilon \sim N(0,1)$)

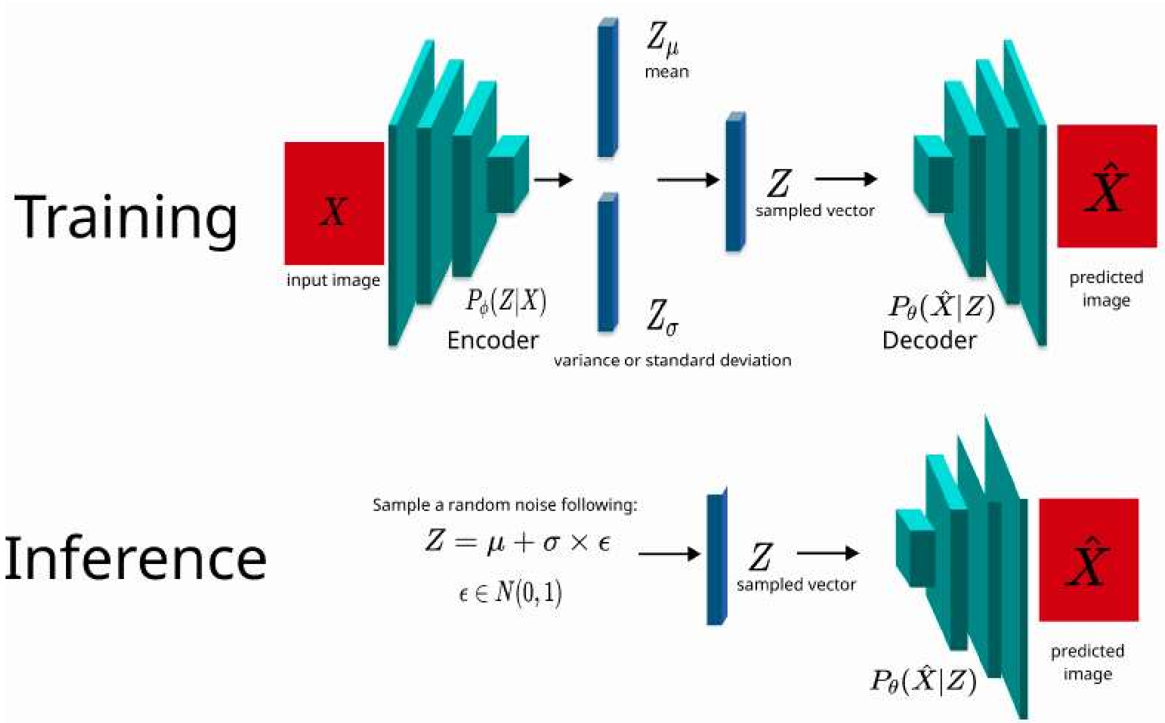




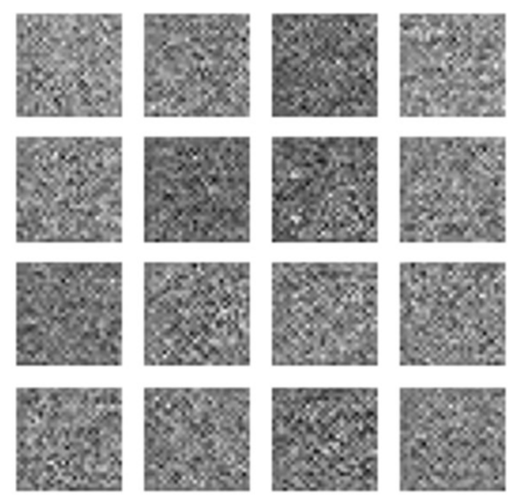
Relevant links:
https://www.youtube.com/watch?v=5WoItGTWV54
http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture13.pdf
https://arxiv.org/abs/1312.6114
https://www.youtube.com/watch?v=igP03FXZqgo
https://www.youtube.com/watch?v=AX5v5med3Rw
https://www.youtube.com/watch?v=9KTrUea1apo
https://www.youtube.com/watch?v=dptTrfzSwb8
https://www.youtube.com/watch?v=PedRXuVcObg
https://www.youtube.com/watch?v=DamPMgZrnSc

Variational Autoencoders (VAEs)

Convolutional Variational Autoencoder (CVAE)



CVAE
after 100 epochs

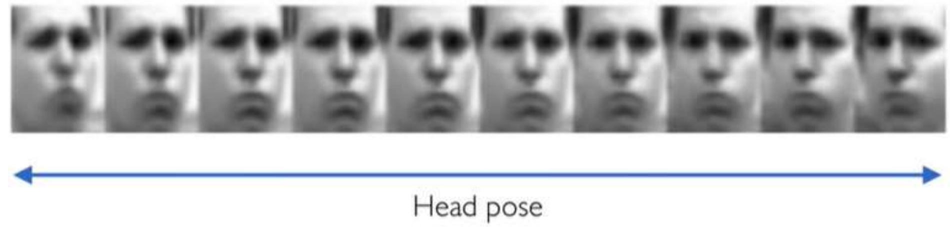


Relevant links:
https://www.tensorflow.org/tutorials/generative/cvae

Variational Autoencoders (VAEs)

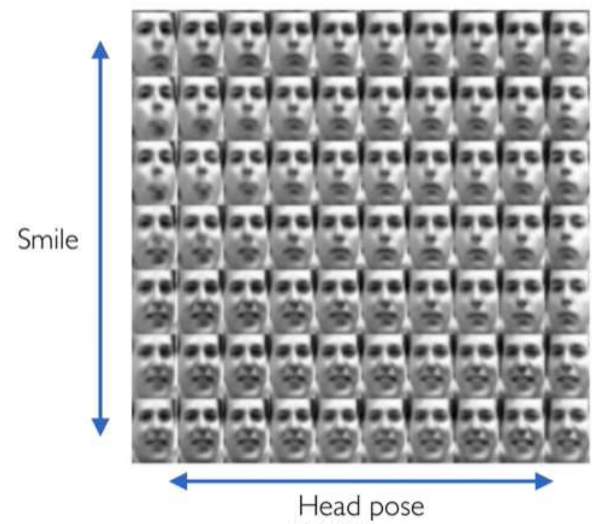
Different dimensions of z encodes different *interpretable latent* feature…

Increasing/decreasing a single latent variable (keeping all other variables fixed), we may manipulate through particular feature (e.g. pose of a head)…



With *disentanglement* we would like to learn the most richest and compact representation, we need the latent variables to be uncorrelated and independent from each other as possible.

Beta-VAE is a type of variational autoencoder that seeks to discovered disentangled latent factors. It modifies VAEs with an adjustable hyperparameter β that balances latent channel capacity and independence constraints with reconstruction accuracy.

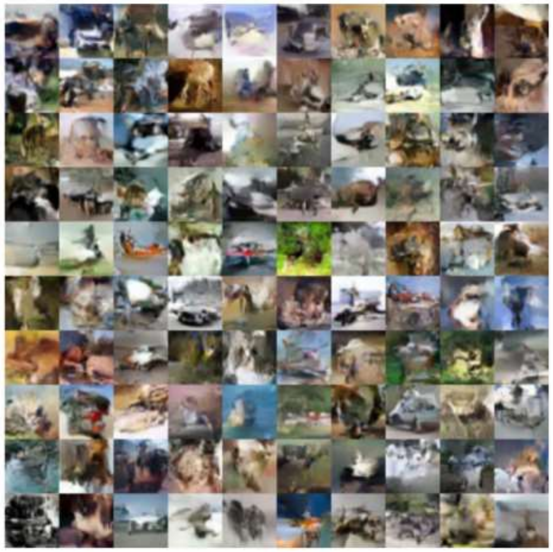


$$\mathcal{F}(\theta, \phi, \beta; \mathbf{x}, \mathbf{z}) \geq \mathcal{L}(\theta, \phi; \mathbf{x}, \mathbf{z}, \beta) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \beta\, D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}) || p(\mathbf{z}))$$

Links: https://openreview.net/pdf?id=Sy2fzU9gl

Relevant links:
https://www.youtube.com/watch?v=5WoItGTWV54
http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture13.pdf
https://arxiv.org/abs/1312.6114
https://paperswithcode.com/method/beta-vae

Variational Autoencoders (VAEs)



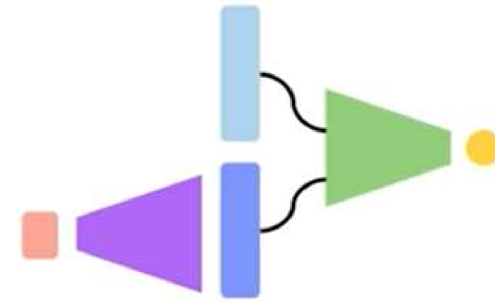
32x32 CIFAR-10

Labeled Faces in the Wild

Autoregressive models:	***Variational models:***
- *Directly maximize P(data)*	- *Maximize lower-bound on P(data)*
- *High-quality generated images*	- *Generated images are often blurry*
- *Slow on image generation*	- *Very fast image generation*
- *No explicit latent codes*	- *Learn rich latent codes*

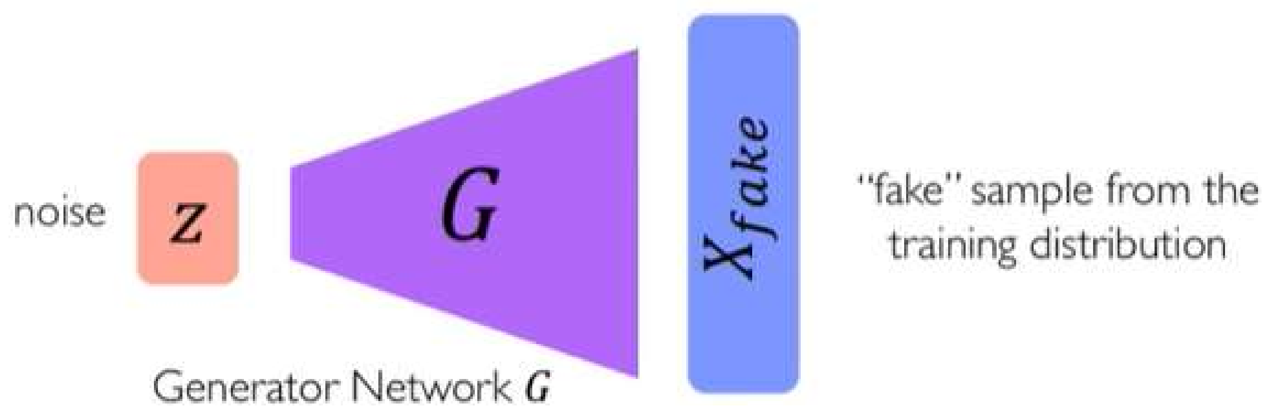
Relevant links:
https://www.youtube.com/watch?v=5WoItGTWV54
https://www.youtube.com/watch?v=FMuvUZXMzKM
http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture13.pdf
https://arxiv.org/abs/1312.6114

Generative Adversarial Networks (GANs)

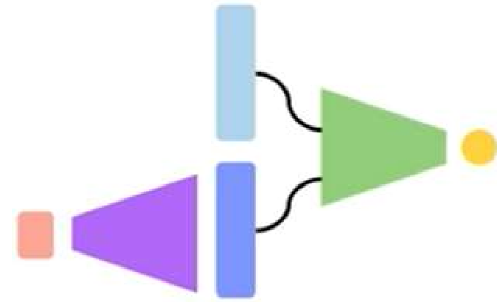


Unlike Autoregressive Models that directly maximize likelihood of training data, and VAEs that introduce a latent space and explicitly model density (distribution underlying some data) maximizing a lower bound, *GANs do not model a distribution directly*, *but instead allow us to generate new instances from it* (meaning that we sampling from the really complex distribution that might be very difficult to learned directly).

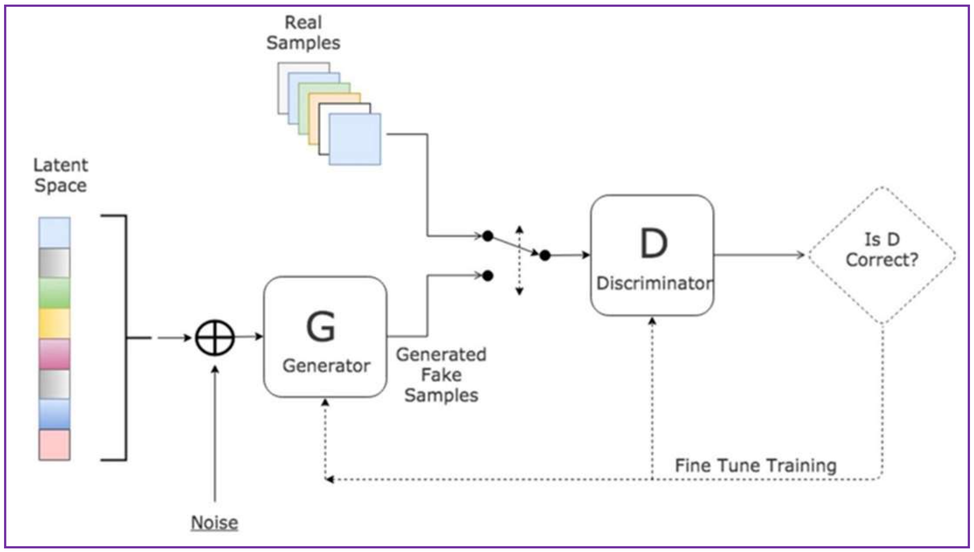
Since, it is not obvious what to sample from complex distribution, *GANs just sample from simple random noise and learn a transformation to the training distribution*...



GANs



Generative Adversarial Networks (GANs) are deep net architectures *(introduced by Ian Goodfellow et al., 2014)* comprised of two nets, competing one against the other (thus the "*adversarial*").



$$\min_{\theta_g}\max_{\theta_d}\left[\mathbb{E}_{x\sim p_{data}}\log D_{\theta_d}(x)+\mathbb{E}_{z\sim p(z)}\log\left(1-D_{\theta_d}\left(G_{\theta_g}(z)\right)\right)\right]$$

Discriminator wants to maximize objective s.t. $D(x)$ close to 1, $D(G(z))$ close to 0.
Generator wants to minimize objective s.t. $D(G(z))$ close to 1.

The *Generative model* attempts to produce fake data (real looking image) that looks so real that the *Discriminative model* cannot tell it is fake. In turn, The Discriminative model is learning to not get fooled by the Generative model and has the task of determining whether a given image looks natural (an image from the dataset) or looks like it has been artificially created. After the models have played the minimax game, we supposed to get:

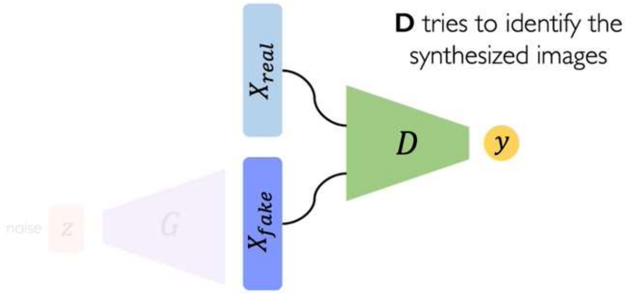
- good quality generator that can generate as many artificial real looking samples as we want;
- good enough discriminator that is aware of the "internal representation of the data" (because it has been trained to understand the differences between real images from the dataset and artificially created ones) and can be used as a feature extractor for a CNN.

Further extensions of GAN are *DCGAN*, *Sequence-GAN*, *LSTM-GAN*, etc.

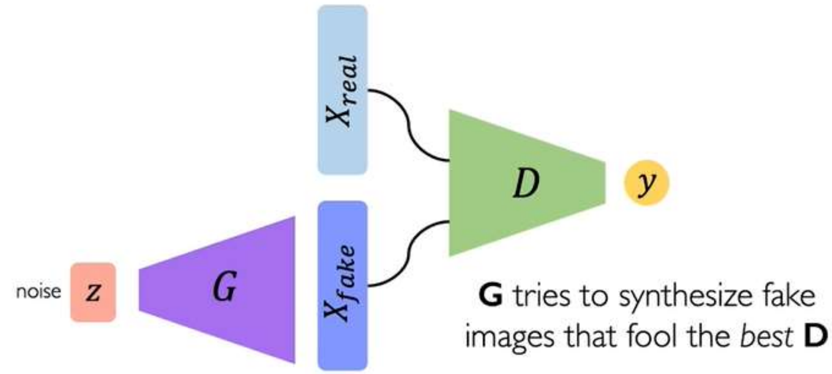
GAN Zoo *https://github.com/hindupuravinash/the-gan-zoo*

Relevant links:
https://arxiv.org/pdf/1406.2661v1.pdf ; https://arxiv.org/pdf/1506.05751.pdf ; https://arxiv.org/pdf/1701.00160.pdf
https://channel9.msdn.com/Events/Neural-Information-Processing-Systems-Conference/Neural-Information-Processing-Systems-Conference-NIPS-2016/Generative-Adversarial-Networks
https://www.analyticsvidhya.com/blog/2017/06/introductory-generative-adversarial-networks-gans/
https://blog.statsbot.co/generative-adversarial-networks-gans-engine-and-applications-f96291965b47

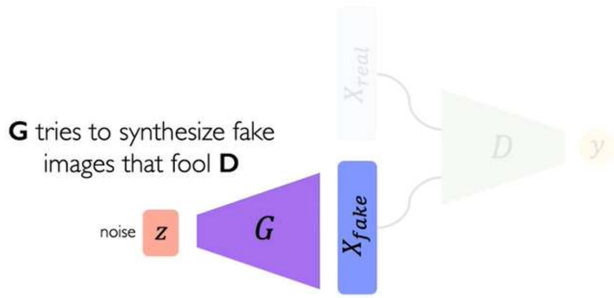
GANs



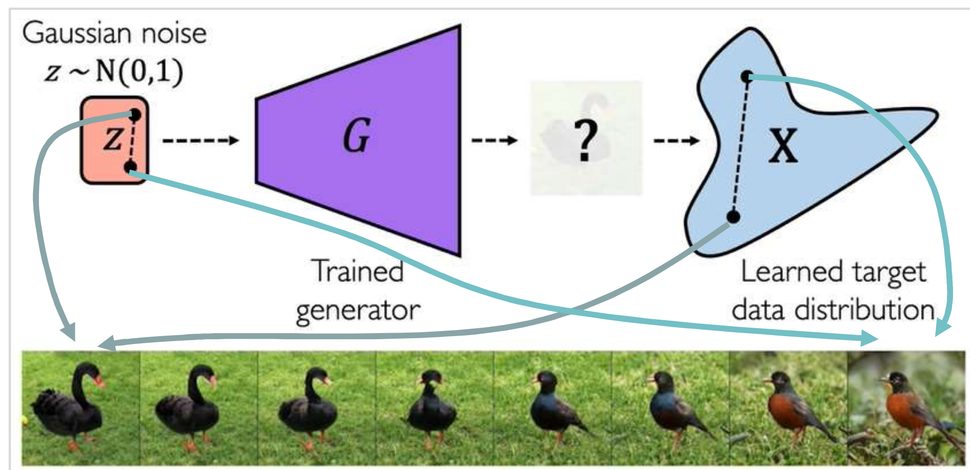
$$\arg\max_{D} \mathbb{E}_{\mathbf{z},\mathbf{x}}[\ \underbrace{\log D(G(\mathbf{z}))}_{\text{Fake}} + \underbrace{\log (1 - D(\mathbf{x}))}_{\text{Real}}\ ]$$



$$\arg\min_{G} \max_{D} \mathbb{E}_{\mathbf{z},\mathbf{x}}[\ \log D(G(\mathbf{z})) + \log (1 - D(\mathbf{x}))\ ]$$



$$\arg\min_{G} \mathbb{E}_{\mathbf{z},\mathbf{x}}[\ \log D(G(\mathbf{z})) + \log (1 - D(\mathbf{x}))\ ]$$

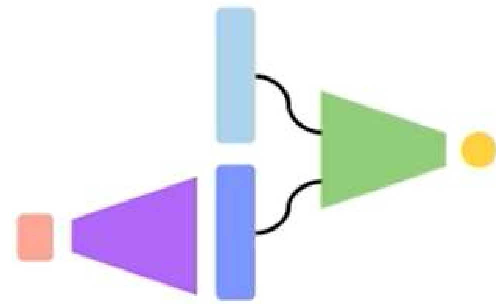


GAN Lab *https://poloclub.github.io/ganlab/*

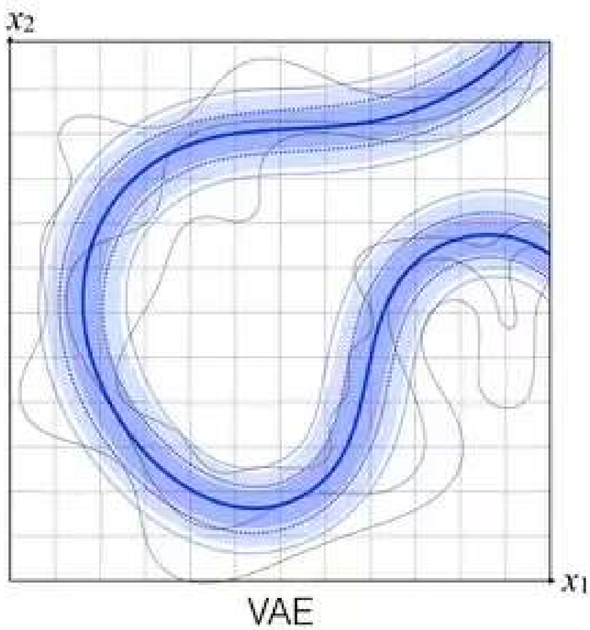
Relevant links:

https://arxiv.org/pdf/1406.2661v1.pdf ; https://arxiv.org/pdf/1506.05751.pdf ; https://arxiv.org/pdf/1701.00160.pdf
http://introtodeeplearning.com/slides/6S191_MIT_DeepLearning_L4.pdf
https://www.youtube.com/watch?v=ZQCe3oN9gKI

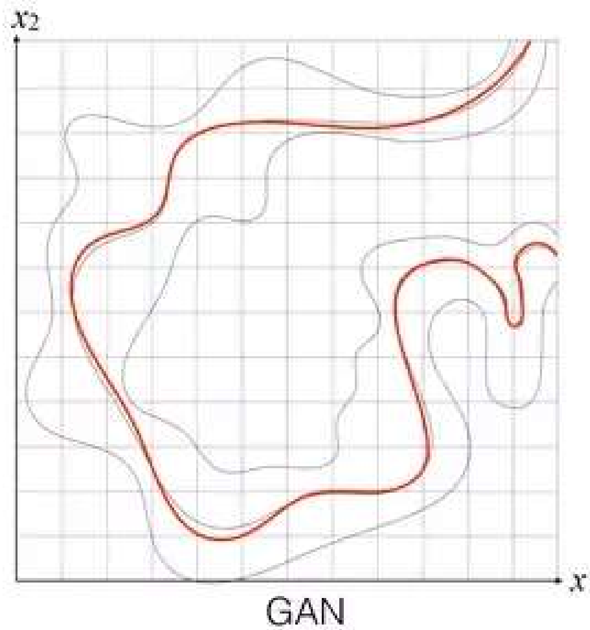
GANs

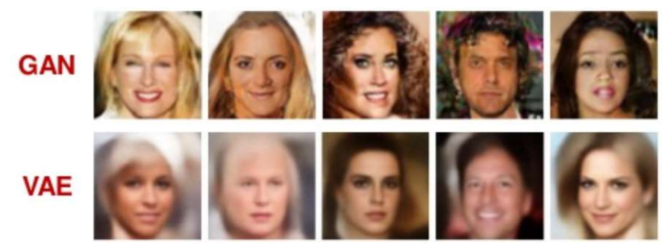


VAE vs GAN



more traditional max-likelihood approach





- objective is to ***reconstruct real data***
- uses pixel-to-pixel loss
- output images are ***more blurred***
- ***lower diversity*** and ***higher stability***

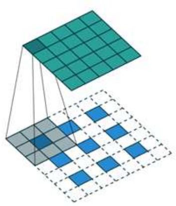
- objective is to ***generate new data***
- Generator aims to fool the Discriminator
- Discriminator aims distinguish generated data from real
- output images are ***sharper***
- ***higher diversity*** and ***lower stability***

GANs

Deep Convolutional Generative Adversarial Networks (DCGAN)

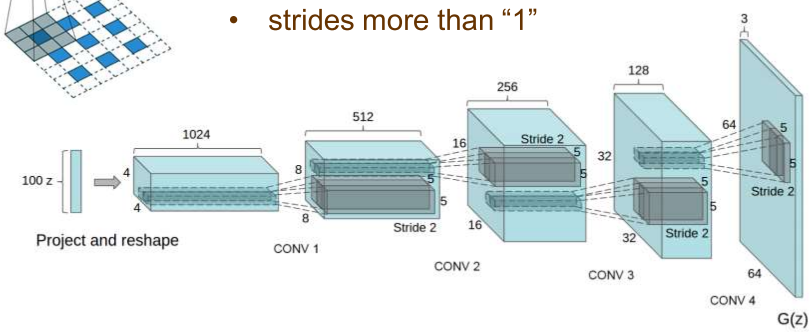
Generating bedroom images and Face arithmetic with DCGANs…
Radford et al. 2015: Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks
https://arxiv.org/abs/1511.06434

Interpolation…
We may observe image transformations while interpolation between points in latent space.

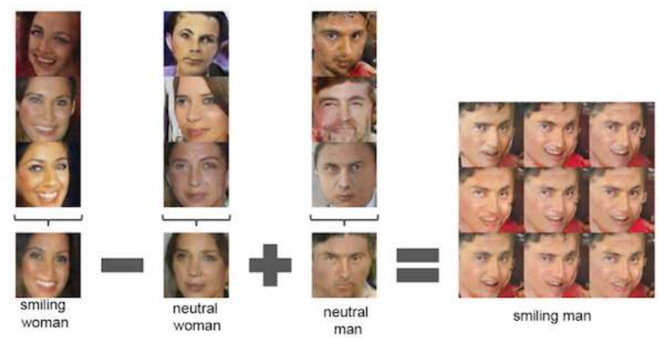


DCGAN Architecture:

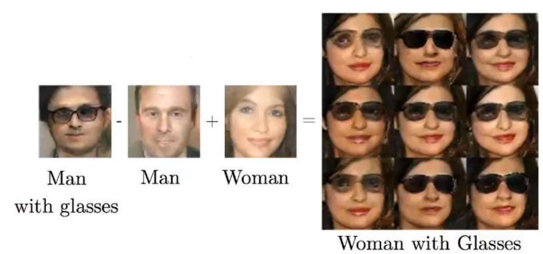
- most "deconvs" are batch normalized
- strides more than "1"



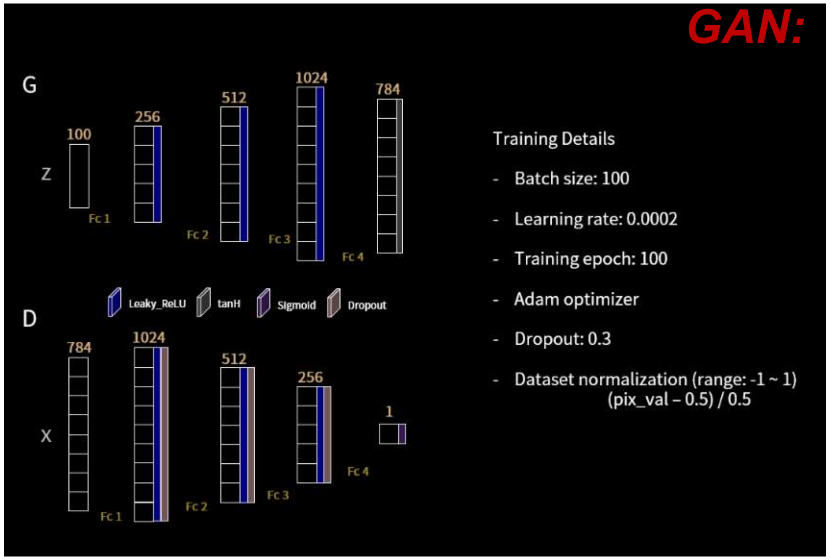
Vector Space Arithmetic…
Algebra in the latent space also corresponds to semantics, similarly to the word embedding in language models (e.g. queen – woman ~ king).

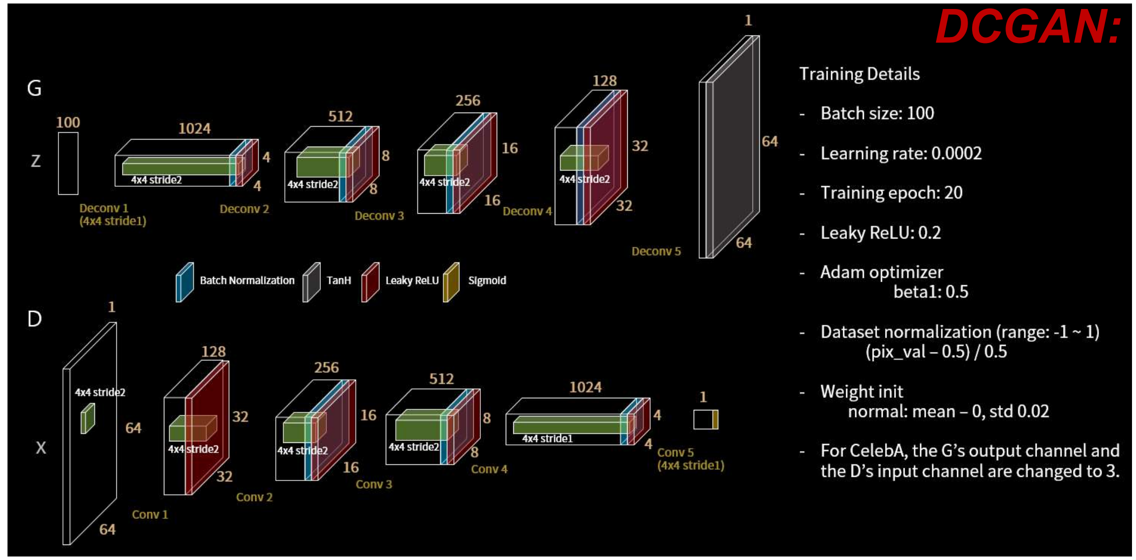


Relevant links:
https://arxiv.org/abs/1511.06434
https://github.com/carpedm20/DCGAN-tensorflow ; https://github.com/openai/improved-gan
https://bamos.github.io/2016/08/09/deep-completion
https://medium.com/@ramyahrgowda/dcgan-implementation-in-keras-explained-e1918fc930ea

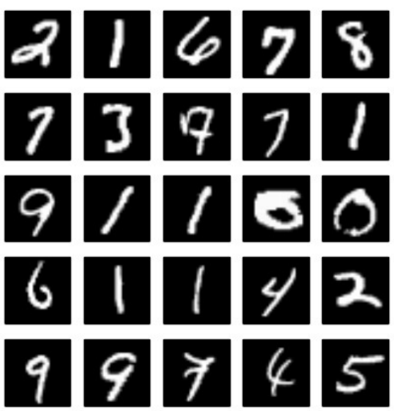


GANs

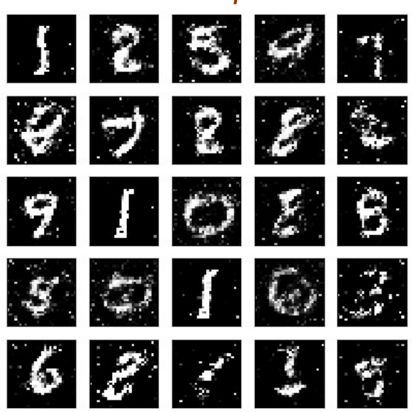




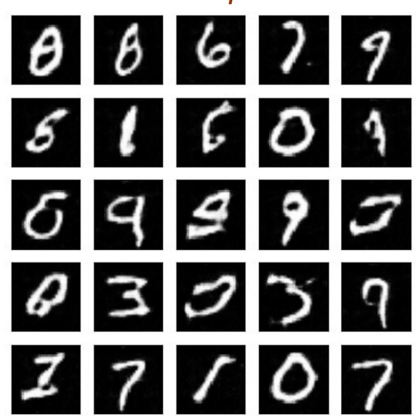
MNIST
original



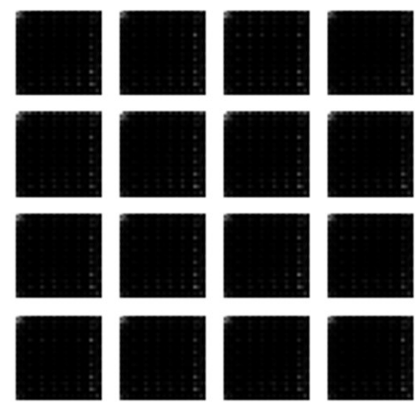
GAN
after 100 epochs



DCGAN
after 20 epochs



DCGAN

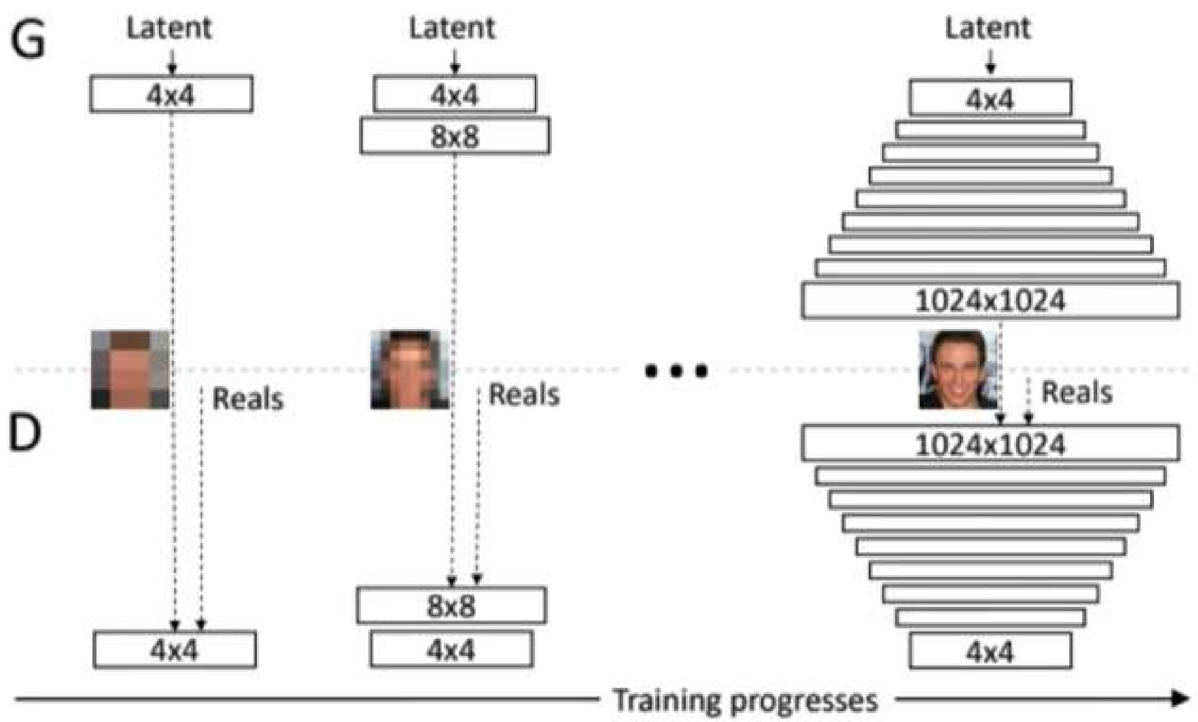


Relevant links:
https://github.com/znxlwm/tensorflow-MNIST-GAN-DCGAN
https://www.tensorflow.org/tutorials/generative/dcgan

GANs

Progressive growing of GANs (NVIDIA) *as a new training methodology for generative adversarial networks* *(Karras et.al., 2018).*
The key idea is to grow both the generator and discriminator progressively. Starting from a low resolution, add new layers that model increasingly fine details as training progresses. This approach allows the generation of large high-quality images, such as 1024×1024 photorealistic faces of celebrities that do not exist.

Links: https://arxiv.org/abs/1710.10196
https://github.com/tkarras/progressive_growing_of_gans
https://www.youtube.com/watch?v=G06dEcZ-QTg

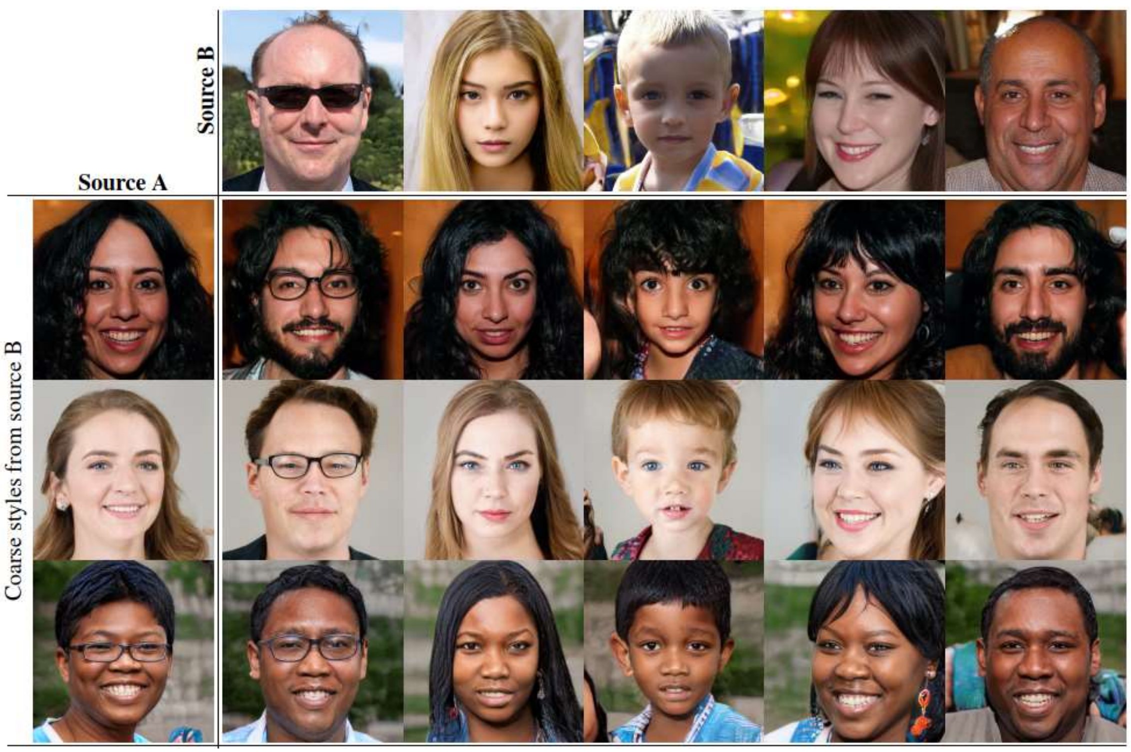


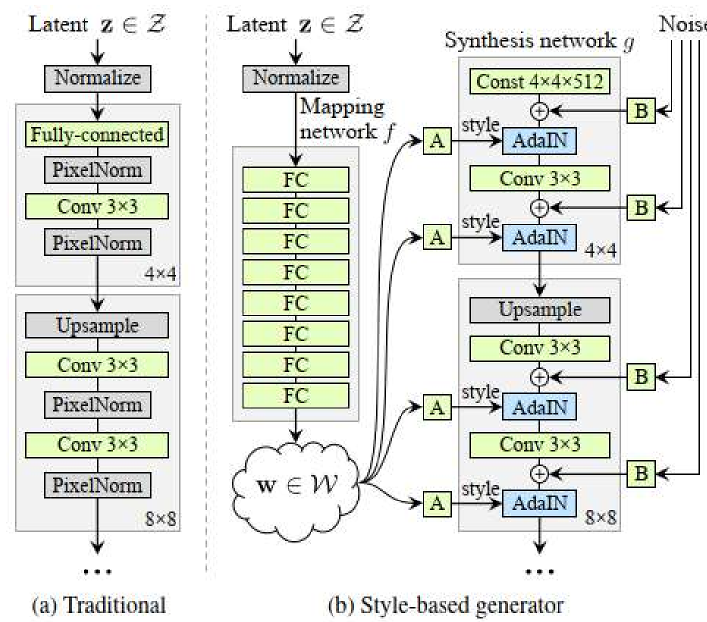
Relevant links:
https://machinelearningmastery.com/introduction-to-progressive-growing-generative-adversarial-networks/

GANs

StyleGAN - a Style-Based Generator Architecture for GANs *propose an alternative generator architecture for generative adversarial networks, borrowing from style transfer literature. It could be considered as a* ***combination of progressive growing and style transfer****. The new architecture leads to an automatically learned, unsupervised separation of high-level attributes (e.g., pose and identity when trained on human faces) and stochastic variation in the generated images (e.g., freckles, hair), and it enables intuitive, scale-specific control of the synthesis* *(Karras et.al., 2019).*

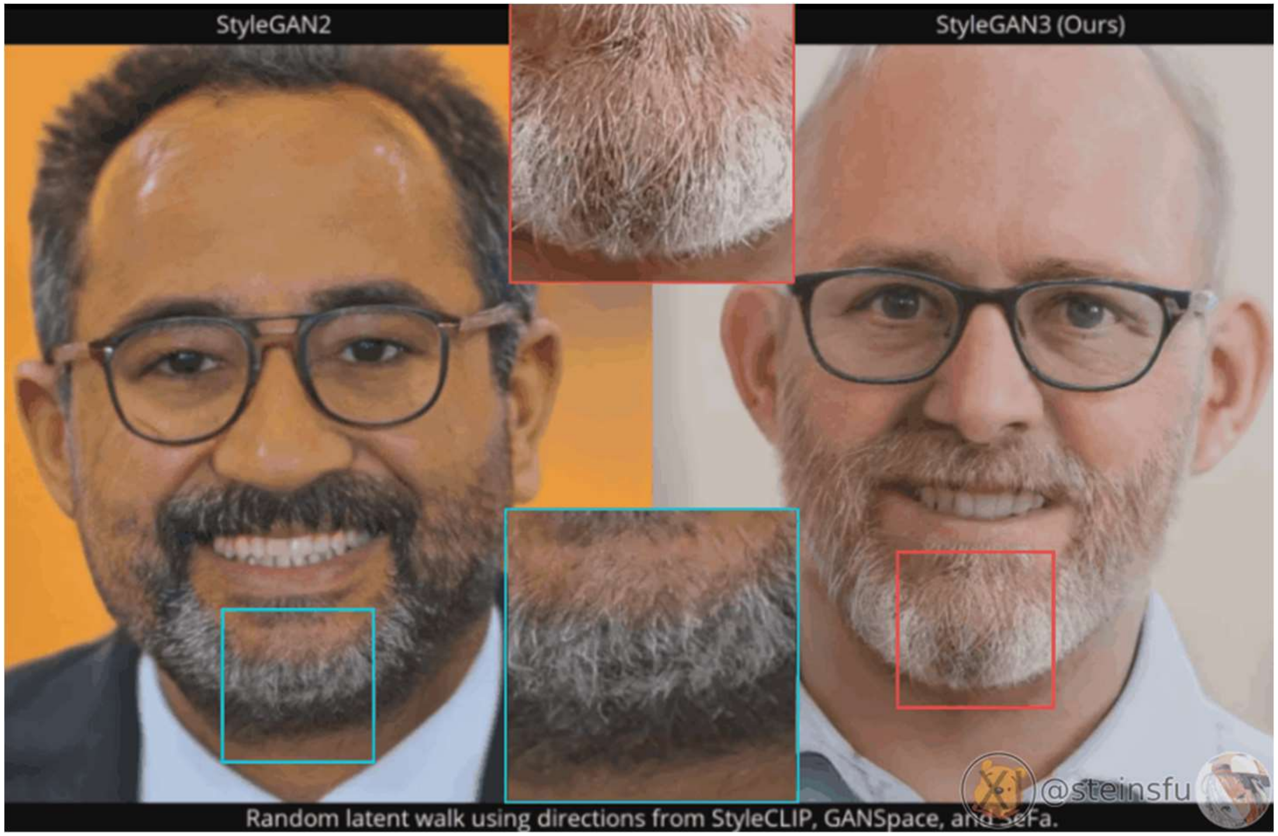
Links: https://arxiv.org/abs/1812.04948
https://www.youtube.com/watch?v=dCKbRCUyop8





GANs

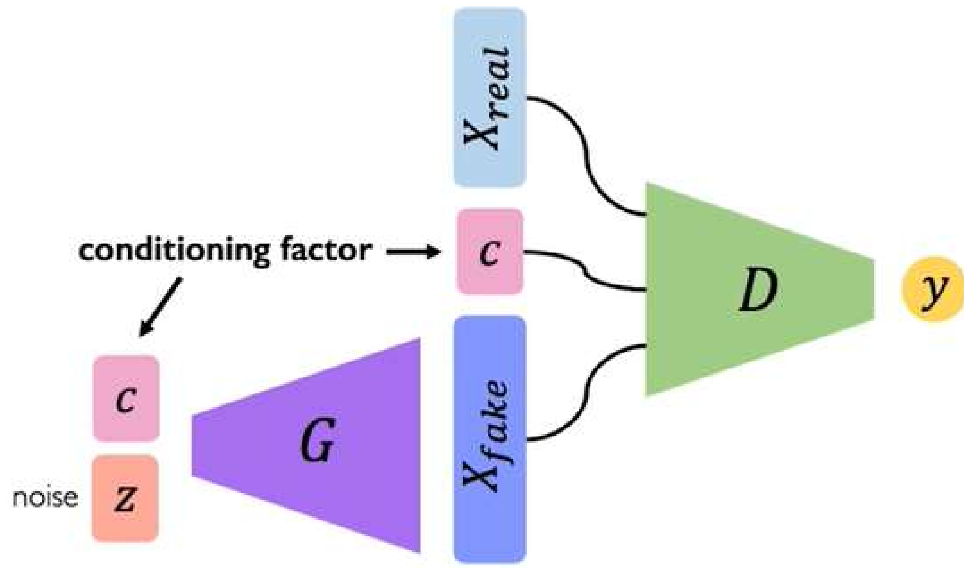
… StyleGAN2, StyleGAN2-ADA, StyleGAN3 …



Relevant links:
https://medium.com/@steinsfu/stylegan-vs-stylegan2-vs-stylegan2-ada-vs-stylegan3-c5e201329c8a
https://medium.com/@steinsfu/stylegan3-clearly-explained-793edbcc8048
https://nvlabs.github.io/stylegan3/
https://github.com/NVlabs/stylegan3
https://catalog.ngc.nvidia.com/orgs/nvidia/teams/research/models/stylegan3
https://blog.paperspace.com/stylegan3-gradient-notebooks/

GANs

Conditional GANs



GANs

BigGAN: Large Scale GAN Training for High Fidelity Natural Image Synthesis

(Brock et.al., 2019).

Authors studied the instabilities specific to large scale images and find that applying orthogonal regularization to the generator renders it amenable to a simple "truncation trick," allowing fine control over the trade-off between sample fidelity and variety by reducing the variance of the Generator's input. Their modifications lead to models which set the new state of the art in class-conditional image synthesis.

Links: https://arxiv.org/abs/1809.11096
https://paperswithcode.com/method/biggan

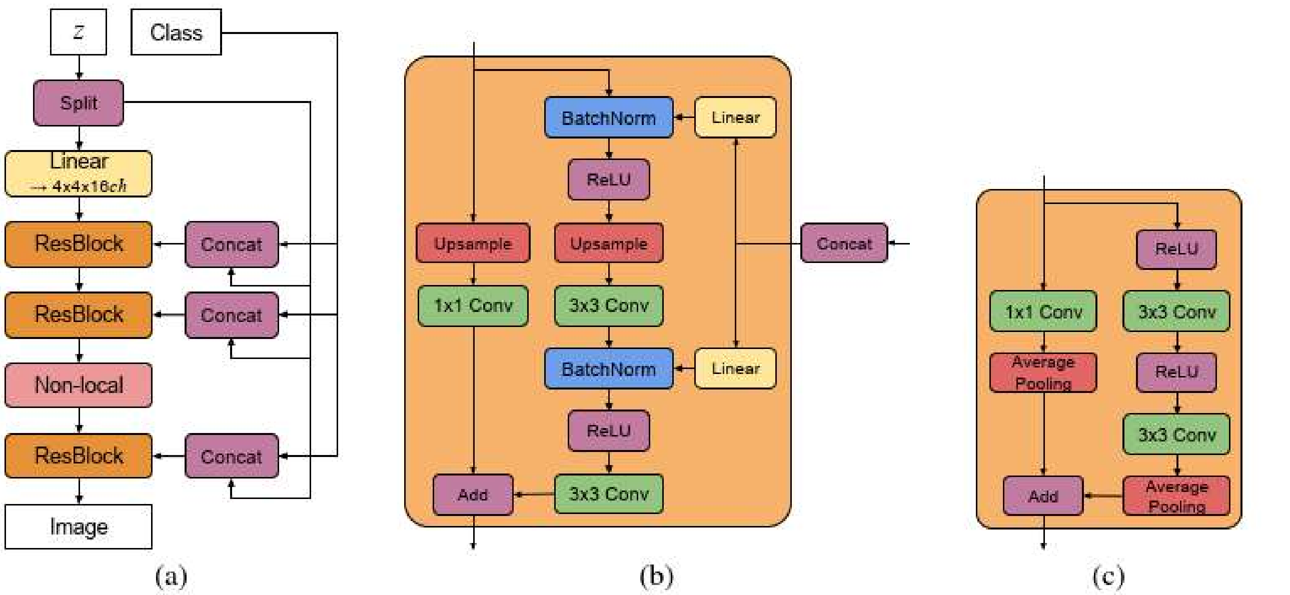


Figure 15: (a) A typical architectural layout for BigGAN's G; details are in the following tables. (b) A Residual Block (*ResBlock up*) in BigGAN's G. (c) A Residual Block (*ResBlock down*) in BigGAN's D.

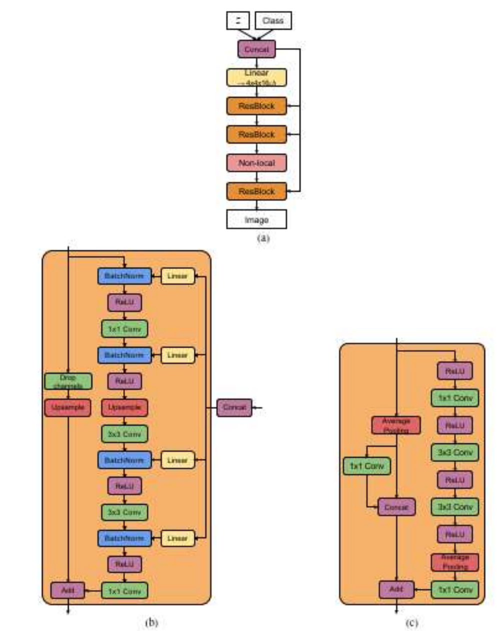


Figure 16: (a) A typical architectural layout for BigGAN-deep's G; details are in the following tables. (b) A Residual Block (*ResBlock up*) in BigGAN-deep's G. (c) A Residual Block (*ResBlock down*) in BigGAN-deep's D. A *ResBlock* (without *up* or *down*) in BigGAN-deep does not include the *Upsample* or *Average Pooling* layers, and has identity skip connections.

GANs

BigGAN: Large Scale GAN Training for High Fidelity Natural Image Synthesis

(Brock et.al., 2019).

Authors studied the instabilities specific to large scale images and find that applying orthogonal regularization to the generator renders it amenable to a simple "truncation trick," allowing fine control over the trade-off between sample fidelity and variety by reducing the variance of the Generator's input. Their modifications lead to models which set the new state of the art in class-conditional image synthesis.

Links: https://arxiv.org/abs/1809.11096
https://paperswithcode.com/method/biggan

Figure 1: Class-conditional samples generated by our model.

GANs

Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network *(Ledig et.al., 2017).*

Links: https://arxiv.org/pdf/1609.04802.pdf

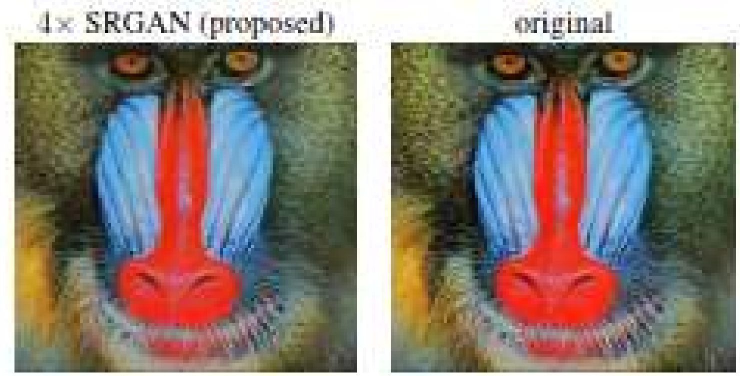


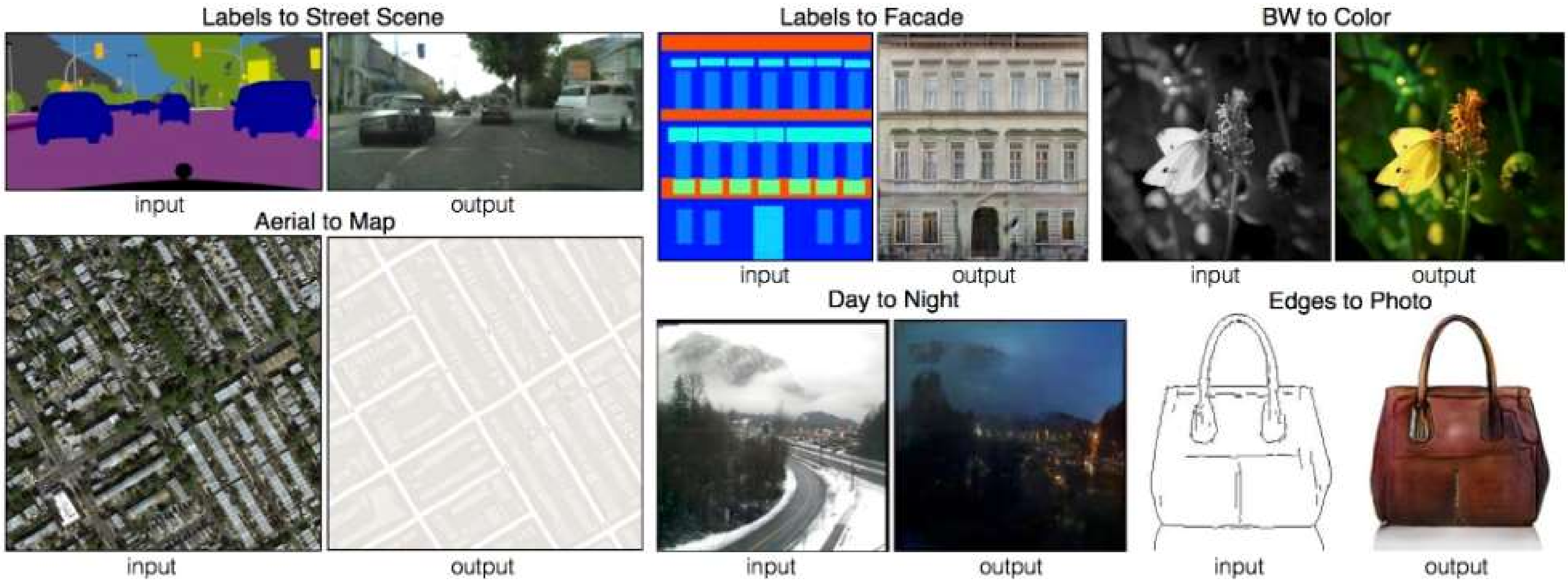
Figure 1: Super-resolved image (left) is almost indistinguishable from original (right). [4× upscaling]

Figure 2: From left to right: bicubic interpolation, deep residual network optimized for MSE, deep residual generative adversarial network optimized for a loss more sensitive to human perception, original HR image. Corresponding PSNR and SSIM are shown in brackets. [4× upscaling]

GANs

Image-to-Image Translation with Conditional Adversarial Nets (Pix2Pix) *as a general-purpose solution to image-to-image translation problems* *(Isola et.al., 2017)*

Links: https://phillipi.github.io/pix2pix/
https://arxiv.org/abs/1611.07004

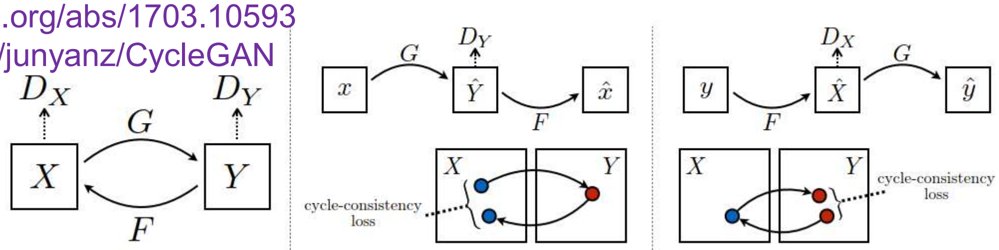


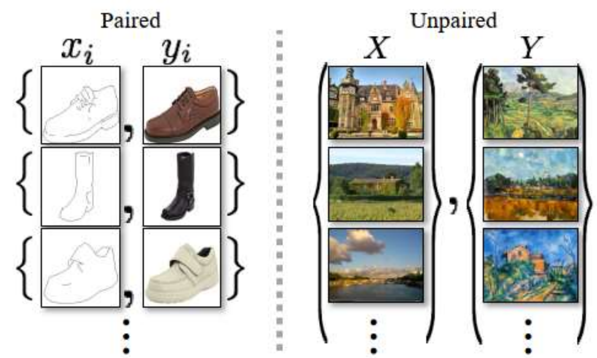
Relevant links:
https://www.tensorflow.org/tutorials/generative/pix2pix

GANs

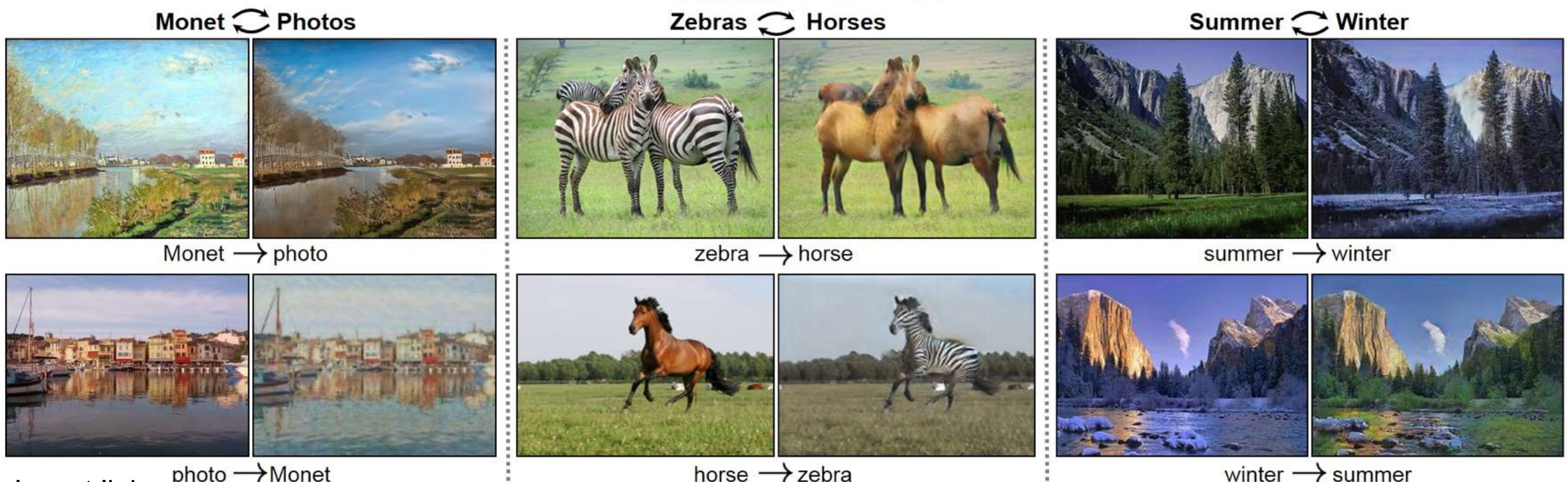
CycleGAN (image-to-image translation using cycle-consistent adversarial network) *presents an approach for learning to translate an image from a source domain X to a target domain Y in the absence of paired examples* (Zhu et.al., 2018).

Links: https://arxiv.org/abs/1703.10593
https://github.com/junyanz/CycleGAN





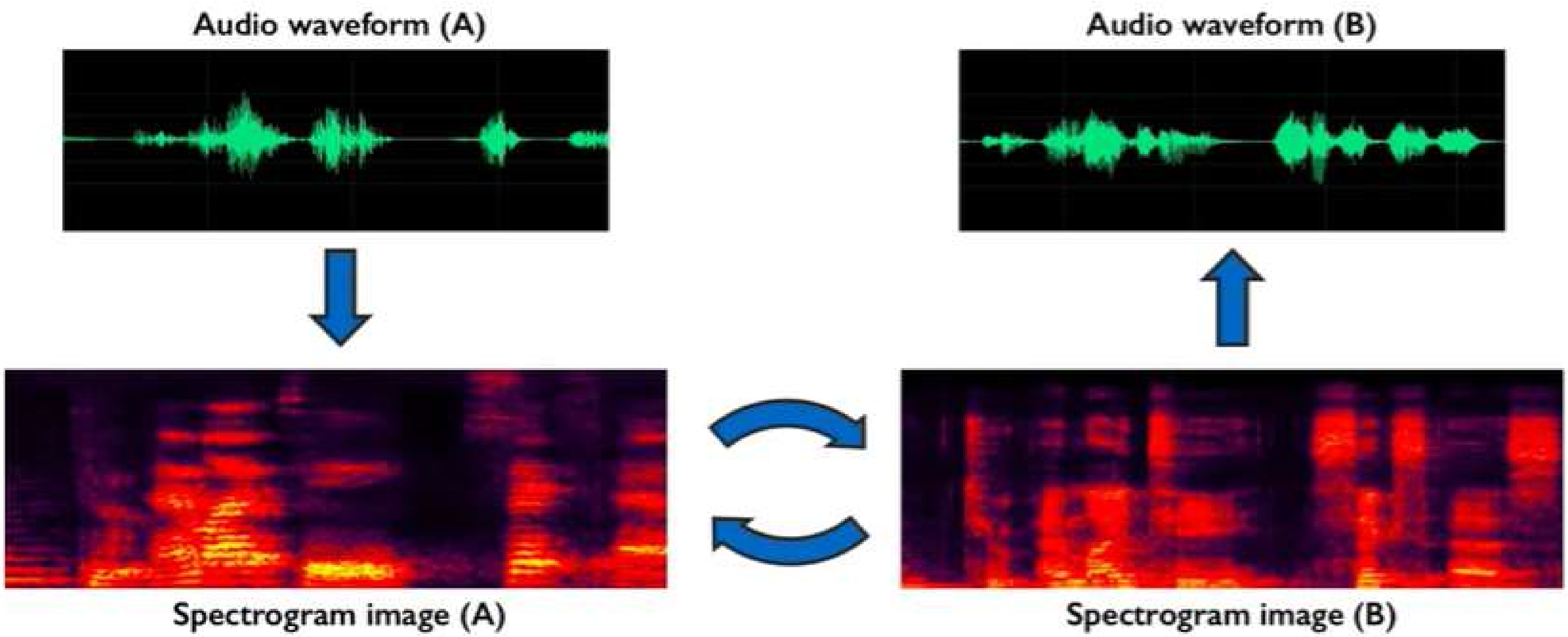
$$\mathcal{L}_{\text{cyc}}(G, F) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\|F(G(x)) - x\|_1] + \mathbb{E}_{y \sim p_{\text{data}}(y)}[\|G(F(y)) - y\|_1].$$



Relevant links:
https://towardsdatascience.com/cyclegan-how-machine-learning-learns-unpaired-image-to-image-translation-3fa8d9a6aa1d
https://neptune.ai/blog/6-gan-architectures
https://www.tensorflow.org/tutorials/generative/cyclegan

GANs

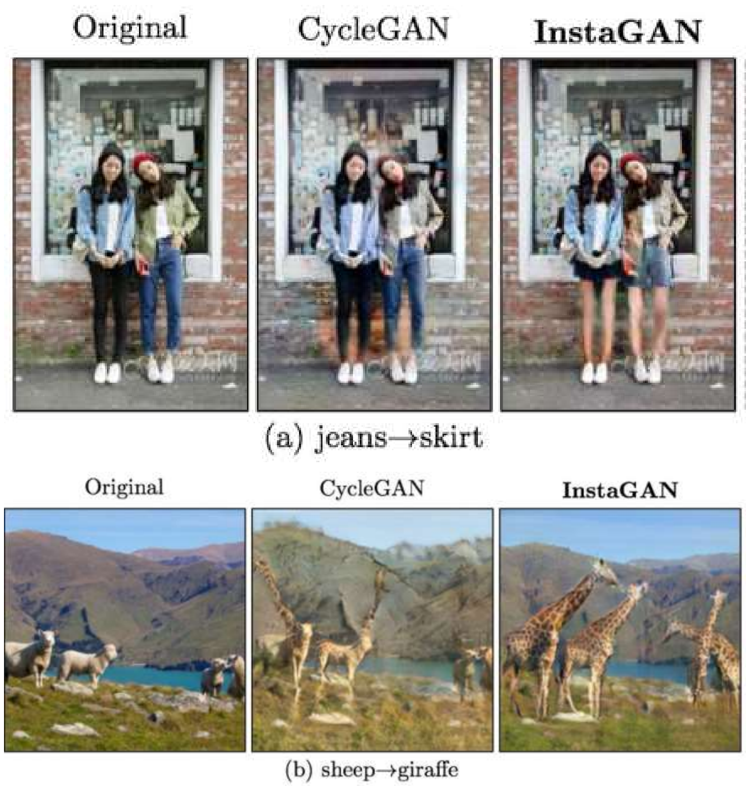
CycleGAN for Speech Transformation. *Having bunch of audio samples of two voices, we can learn to transform representations of voices appearance.*

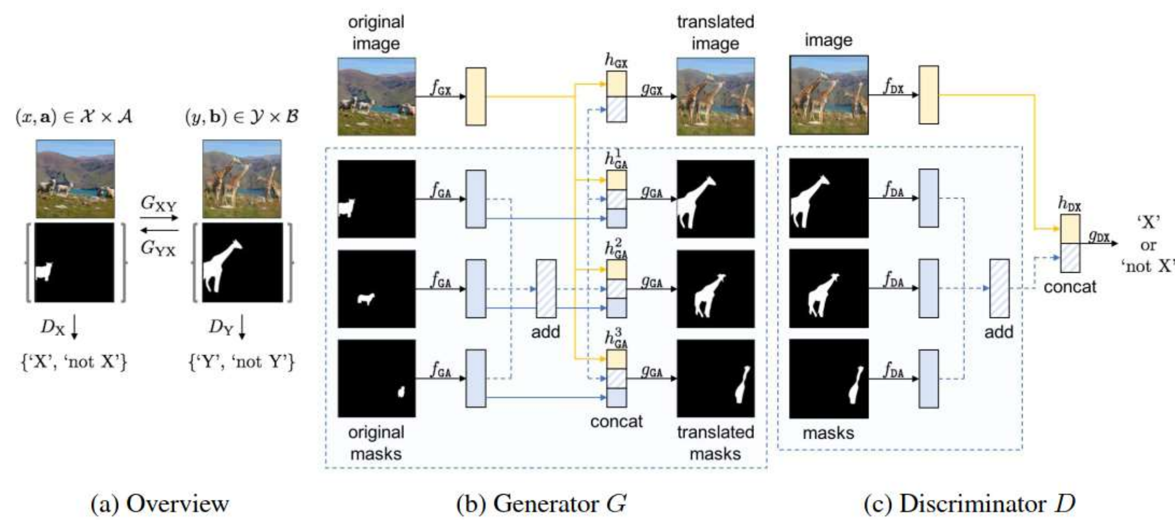


GANs

InstaGAN (instance-aware image-to-image translation) *proposes a novel method that incorporates the instance information (e.g., object segmentation masks) and improves multi-instance transfiguration. The proposed method translates both an image and the corresponding set of instance attributes while maintaining the permutation invariance property of the instances (Mo et.al., 2019).*

Links: https://arxiv.org/abs/1812.10889





GANs

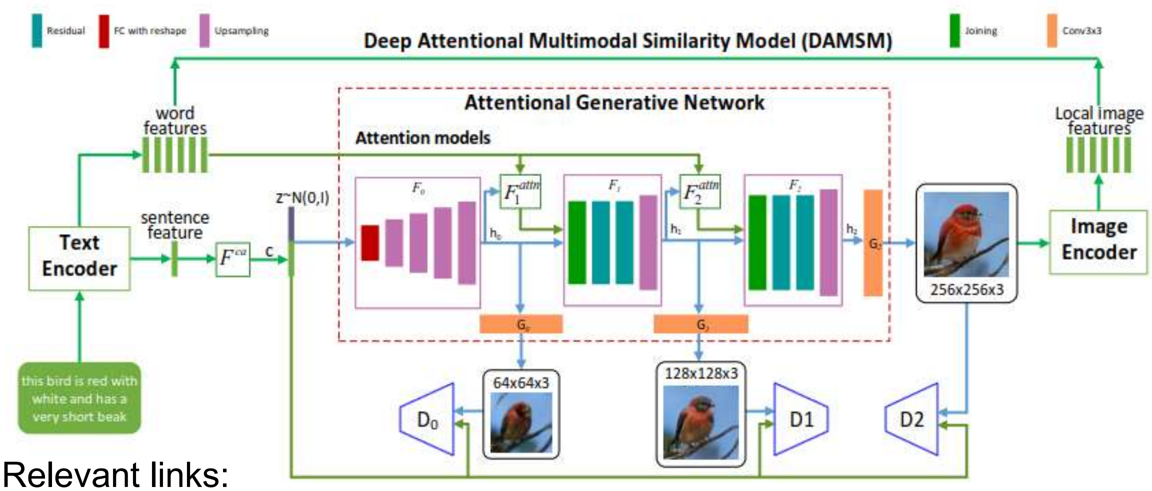
Attentional Generative Adversarial Networks (AttnGAN): Text-to-Image convertor

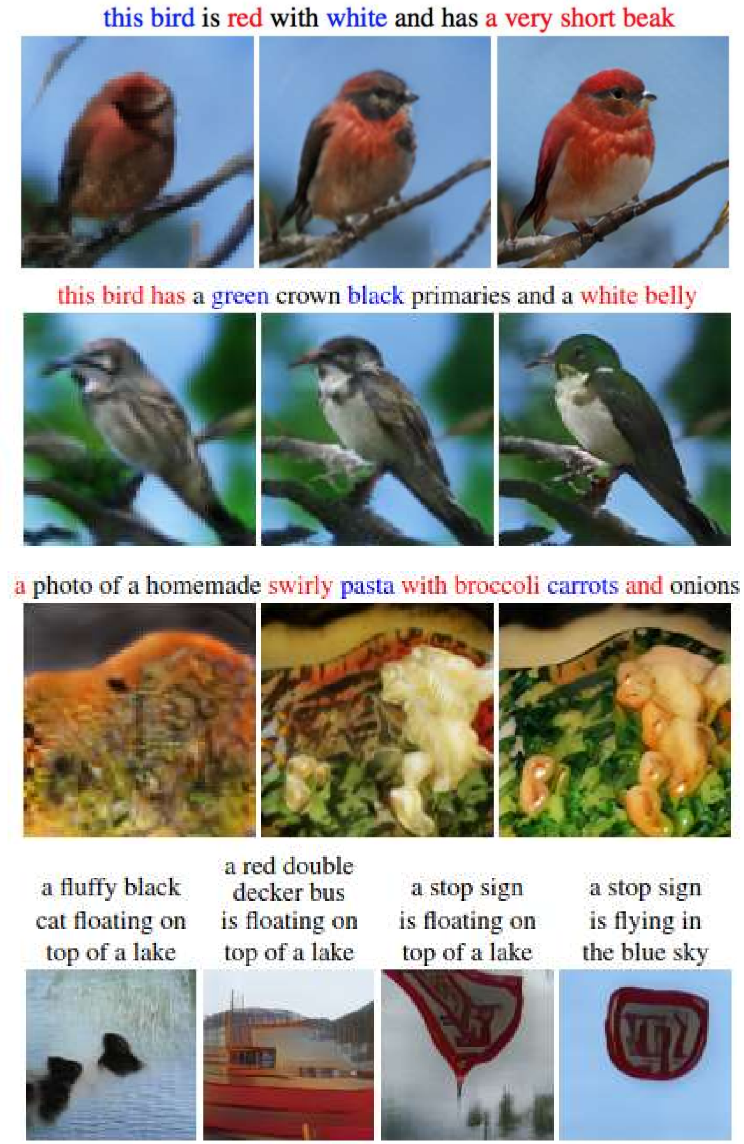
(by Tao Xu et.at., 2017)

"…Here, the pictures are created by the computer, pixel by pixel, from scratch," Microsoft researcher *Xiaodong He* said in a report on the project. "*These birds may not exist in the real world — they are just an aspect of our computer's imagination of birds*."

AttnGAN begins with a crude, low-res image, and then improves it over multiple steps to come up with a final image…

- starts off by generating an image from (random noise + a summation of the caption's token-embeddings);
- uses a combination of *Attention* & *GAN* at every stage, to iteratively add details to the image through highlighting words (words weighted vector) that need more detail (e.g. from "*bird, this, has, belly, white*" towards "*black, green, white, this, bird*", etc.)



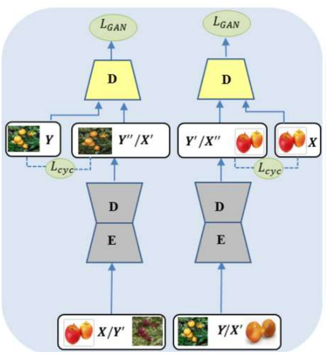


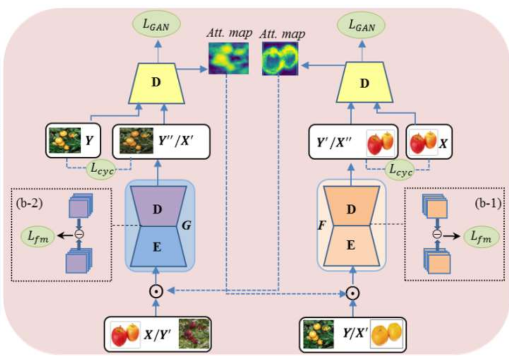
Relevant links:
https://arxiv.org/pdf/1711.10485.pdf
https://codeburst.io/understanding-attngan-text-to-image-convertor-a79f415a4e89
https://www.geekwire.com/2018/artistic-microsoft-bot-draws-whatever-tell-pixel-pixel/

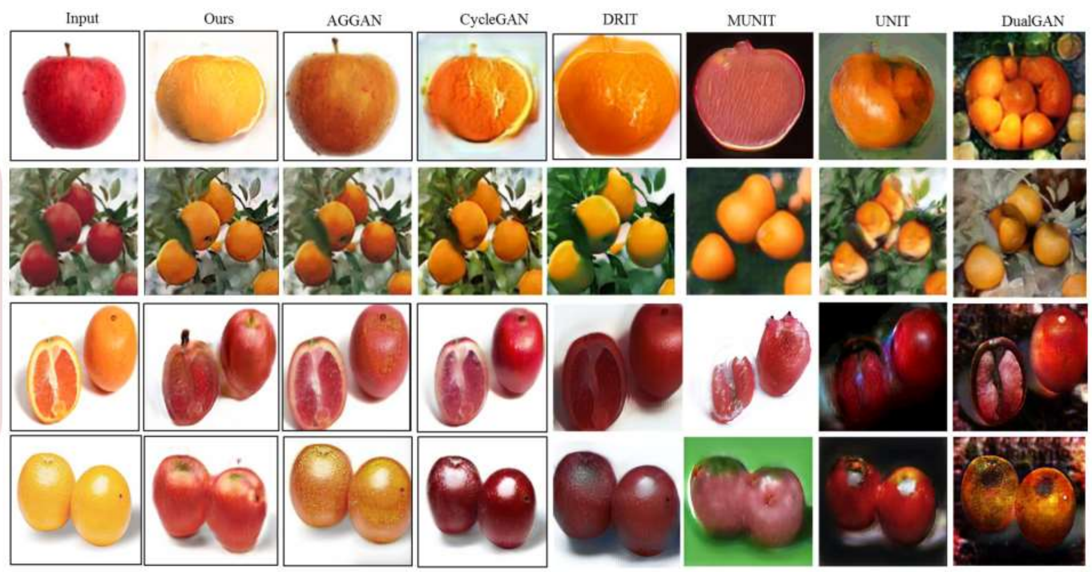
GANs

SPA-GAN: Spatial Attention GAN for Image-to-Image Translation *introduces the attention mechanism directly to the generative adversarial network (GAN) architecture and propose a novel spatial attention GAN model (SPA-GAN) for image-to-image translation tasks. SPA-GAN computes the attention in its discriminator and use it to help the generator focus more on the most discriminative regions between the source and target domains, leading to more realistic output images (Emami et.al., 2019).*

Links: https://arxiv.org/pdf/1908.06616

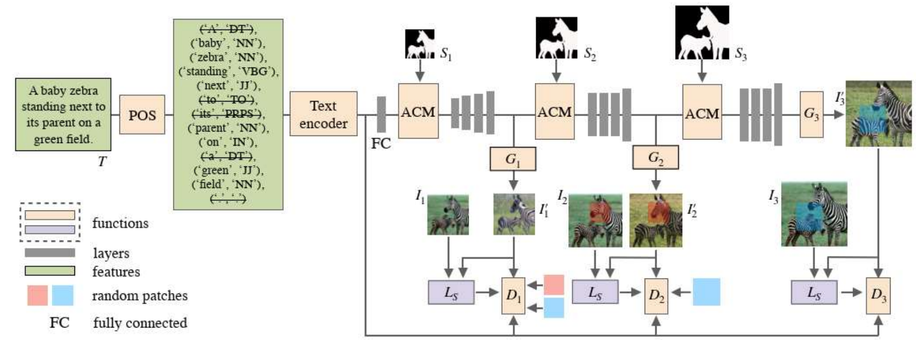






GANs

Image-to-Image Translation with Text Guidance *embeds controllable factors, i.e., natural language descriptions, into image-to-image translation with generative adversarial networks, which allows text descriptions to determine the visual attributes of synthetic images* (Li et.al., 2020).



Links: https://arxiv.org/pdf/2002.05235

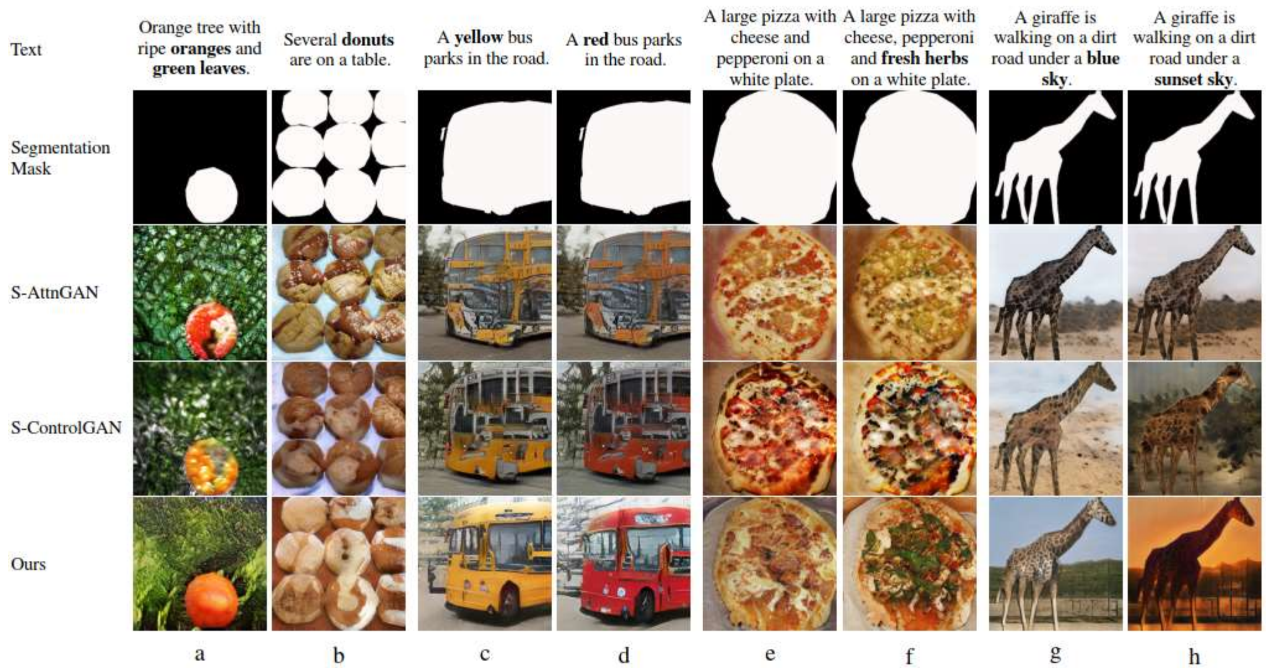


Figure 3. Qualitative comparison of three methods on the COCO dataset. (1) *a* and *b* represent the generation of objects belonging to different categories on similar segmentation masks; (2) *c* and *d* illustrate the controllable ability of internal visual attributes of objects; (3) *e* and *f* show the capability of adding new visual attributes on synthetic images while preserving other text-unmodified contents; and (4) *g* and *h* show that the model can also control the global style of the generated results.

GANs

Semantic Image Synthesis with Spatially-Adaptive Normalization *propose spatially-adaptive normalization, a simple but effective layer for synthesizing photorealistic images given an input semantic layout. (Park et.al., 2019).*

Links: https://arxiv.org/pdf/1903.07291.pdf

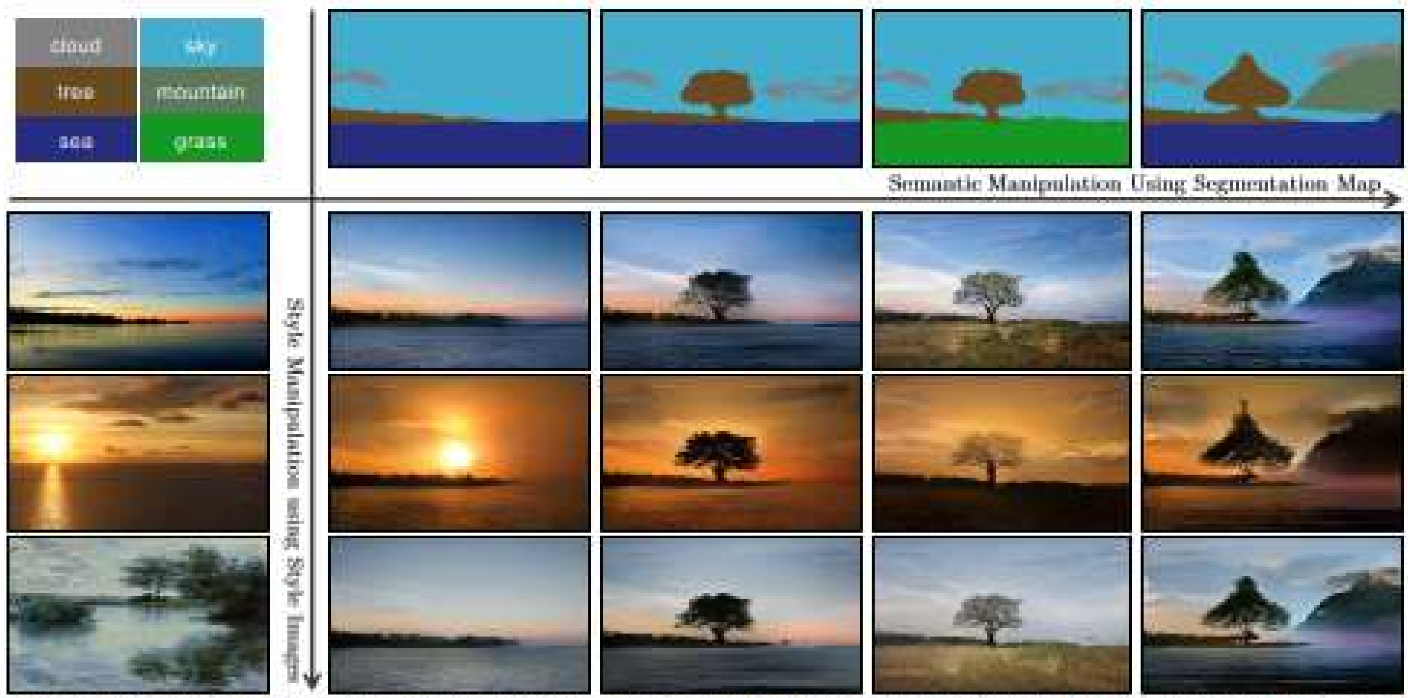
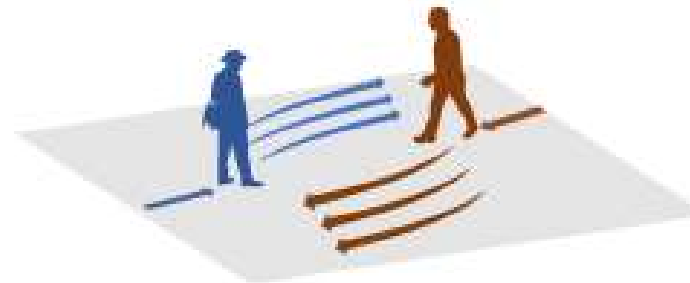


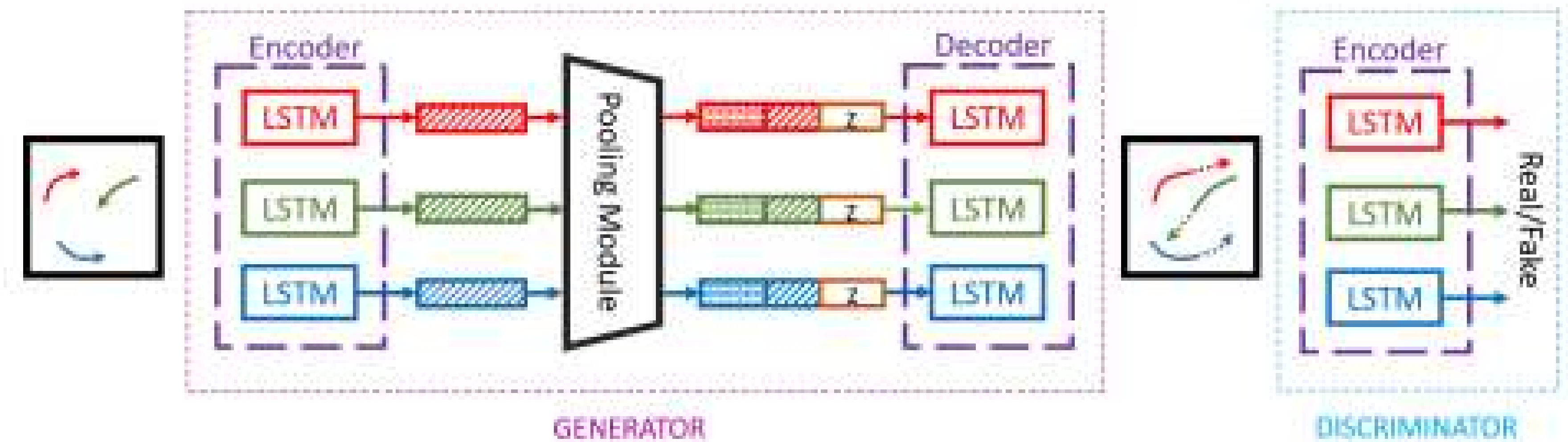
Figure 1: Our model allows user control over both semantic and style as synthesizing an image. The semantic (e.g., the existence of a tree) is controlled via a label map (the top row), while the style is controlled via the reference style image (the leftmost column). Please visit our website for interactive image synthesis demos.

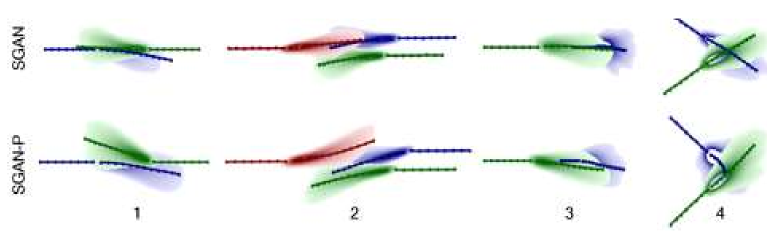
GANs

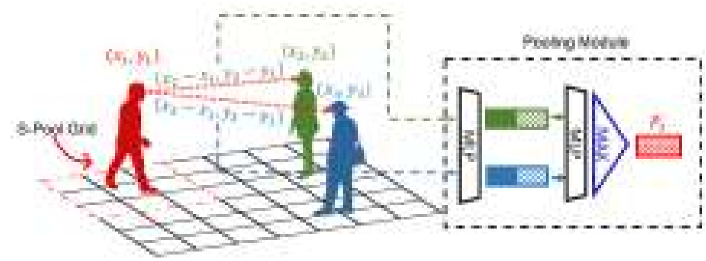


Social GAN: Socially Acceptable Trajectories with Generative Adversarial Networks *(Gupta et.al., 2018).*

Links: https://arxiv.org/pdf/1803.10892.pdf







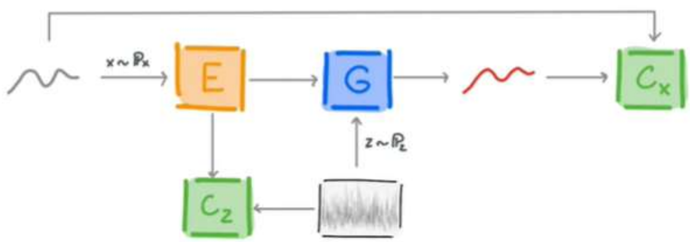
GANs

TadGAN: Time Series Anomaly Detection Using Generative Adversarial Networks

(Geiger et.al., 2020).

Links: https://arxiv.org/pdf/2009.07769.pdf

$$\min_{\{\mathcal{E},\mathcal{G}\}} \max_{\{\mathcal{C}_x\in\mathbf{C}_x,\mathcal{C}_z\in\mathbf{C}_z\}} V_X(\mathcal{C}_x,\mathcal{G})+V_Z(\mathcal{C}_z,\mathcal{E})+V_{L2}(\mathcal{E},\mathcal{G})$$



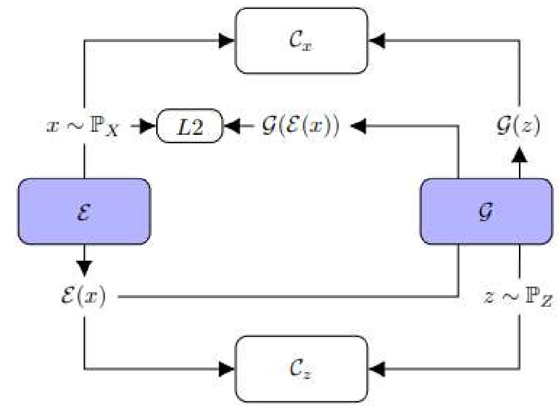
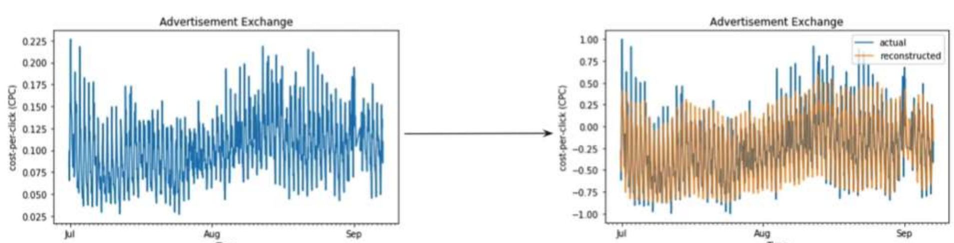
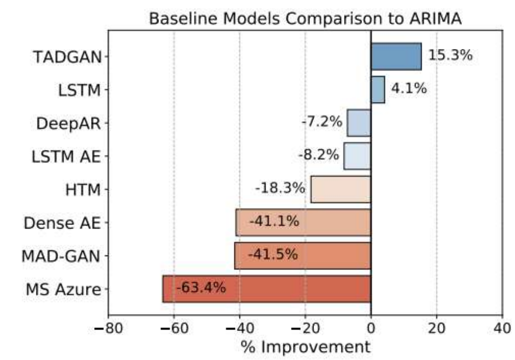


Fig. 2. Model architecture: *Generator* $\mathcal{E}$ is serving as an Encoder which maps the time series sequences into the latent space, while *Generator* $\mathcal{G}$ is serving as a Decoder that transforms the latent space into the reconstructed time series. *Critic* $\mathcal{C}_x$ is to distinguish between real time series sequences from X and the generated time series sequences from $\mathcal{G}(z)$, whereas *Critic* $\mathcal{C}_z$ measures the goodness of the mapping into the latent space.



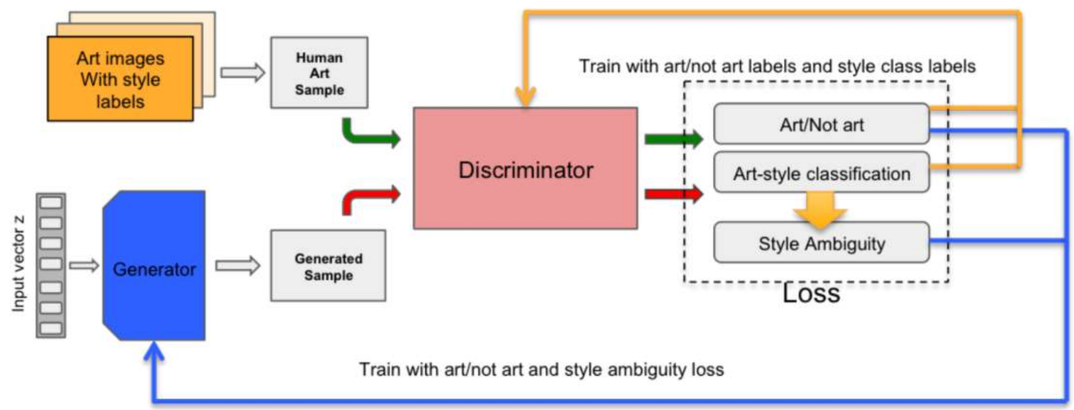


Relevant links:
https://github.com/gusty1g/TadGAN
https://www.youtube.com/watch?v=jIDj2dhU99k

GANs

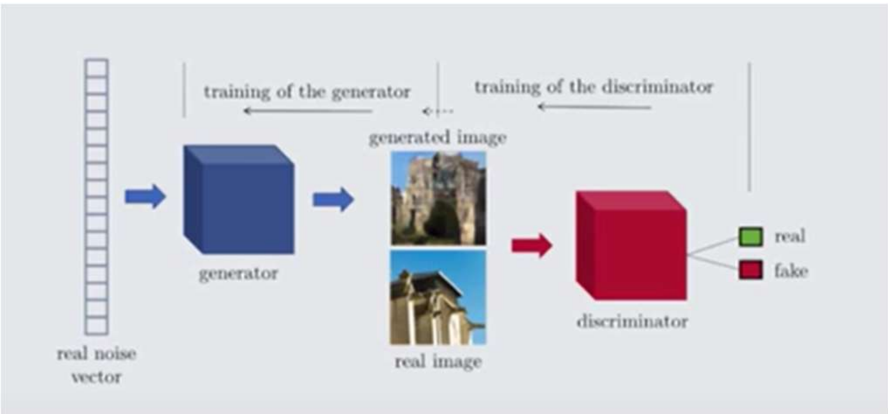
Creative Adversarial Networks (CANs)

Elgammal et al. ICCC 2017: CAN: Creative Adversarial Networks, Generating "Art" by Learning About Styles and Deviating from Style Norms
https://arxiv.org/abs/1706.07068



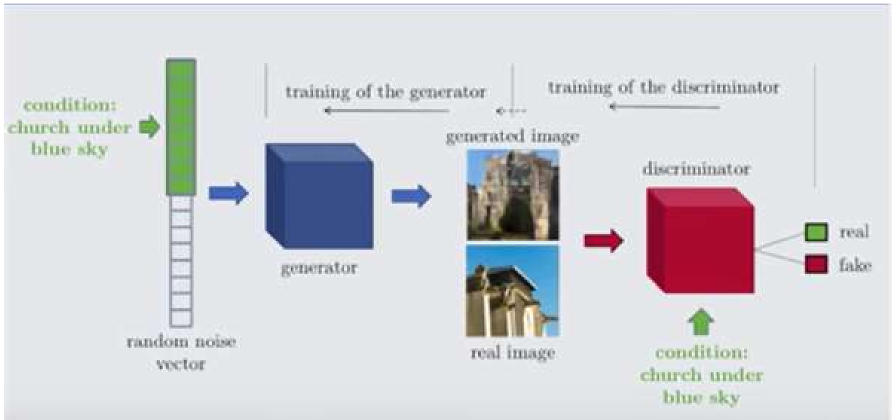
GANs

*more **Image Synthesis** approaches…*

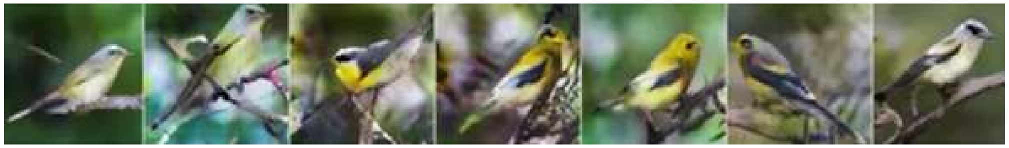


with Adversarial Networks

Denton et al. 2015: Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks
https://arxiv.org/pdf/1506.05751.pdf



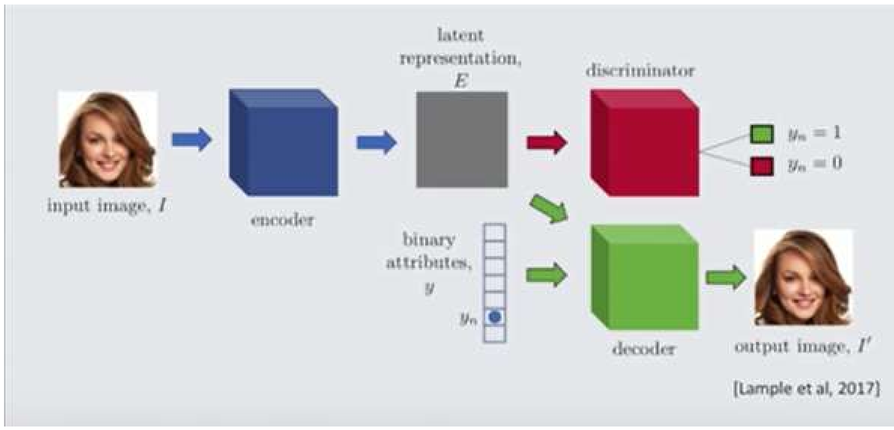
with Conditional Adversarial Networks



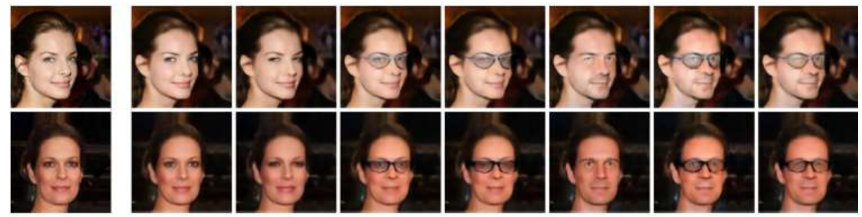
Zhang et al. 2018: StackGAN++: Realistic Image Synthesis with Stacked Generative Adversarial Networks
https://arxiv.org/pdf/1710.10916.pdf
Zhang et al. 2017: StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks
https://arxiv.org/pdf/1612.03242.pdf

GANs

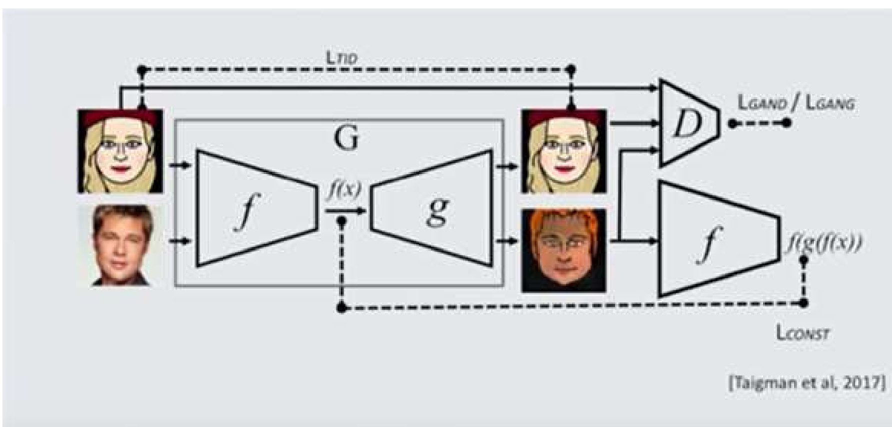
more ***Image Synthesis*** *approaches…*



Disentangling image attributes with Fader Networks
(change gander, add objects, change age, etc.)



Lample et al. 2017: Fader Networks: Manipulating Images by Sliding Attributes
https://arxiv.org/abs/1706.00409



Domain adaptation and creation of completely new styles with Adversarial Nets

Taigman et al. 2017: Unsupervised Cross-domain Image Generation
https://research.fb.com/publications/unsupervised-cross-domain-image-generation/
https://research.fb.com/wp-content/uploads/2017/04/unsupervised-cross-domain_camera_ready0.pdf

GANs

Spectral Normalization for Generative Adversarial Networks *propose a novel weight normalization technique called spectral normalization to stabilize the training of the discriminator. Proposed normalization technique is computationally light and easy to incorporate into existing implementations. Efficacy of spectral normalization was tested on CIFAR10, STL-10, and ILSVRC2012 dataset. (Miyato et.al., 2018).*

Links: https://arxiv.org/pdf/1802.05957.pdf

Figure 7: 128x128 pixel images generated by SN-GANs trained on ILSVRC2012 dataset. The inception score is 21.1±.35.

GANs

Relevant links:
https://github.com/hindupuravinash/the-gan-zoo
https://neptune.ai/blog/6-gan-architectures
https://machinelearningmastery.com/tour-of-generative-adversarial-network-models/
https://arxiv.org/abs/1801.04406
https://github.com/LMescheder/GAN_stability
www.youtube.com/watch?v=RdC4XeExDeY

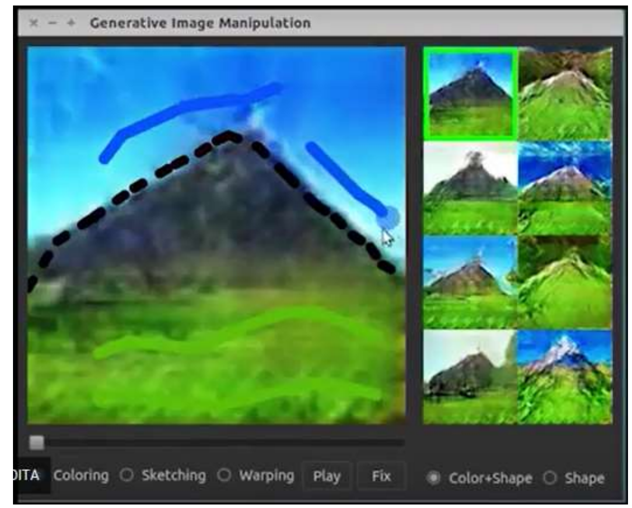
GANs

Neural Face *is an Artificial Intelligence which uses Deep Convolutional Generative Adversarial Networks (DCGAN) (developed by Facebook AI Research) to generate face images…*

Demo: https://carpedm20.github.io/faces/

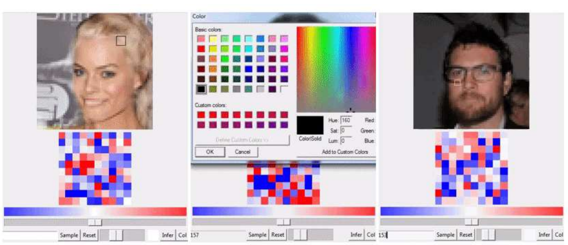
Yearbook Face Editor

Demo: http://codeparade.net/faces/



interactive Generative Adversarial Networks (iGANs) *(by Zhu et al., 2016) is an interactive application that tries to produce the most similar realistic image based on user drawn a rough sketch of an image…*

Video: https://www.youtube.com/watch?v=115YPEdsWI8

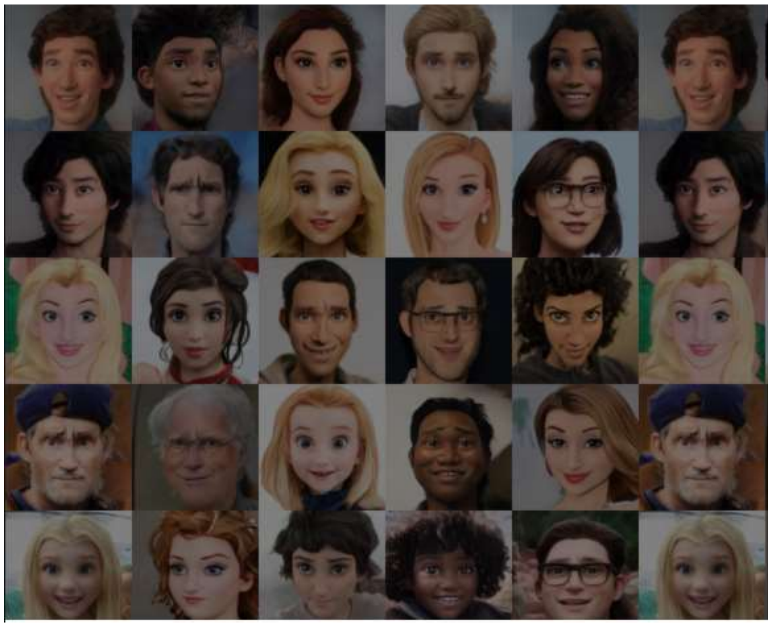


Neural Photo Editing with Introspective Adversarial Networks *(Andrew Brock et.al. 2017) presents Neural Photo Editor - an interface that leverages the power of generative neural networks to make large, semantically coherent changes to existing images…*

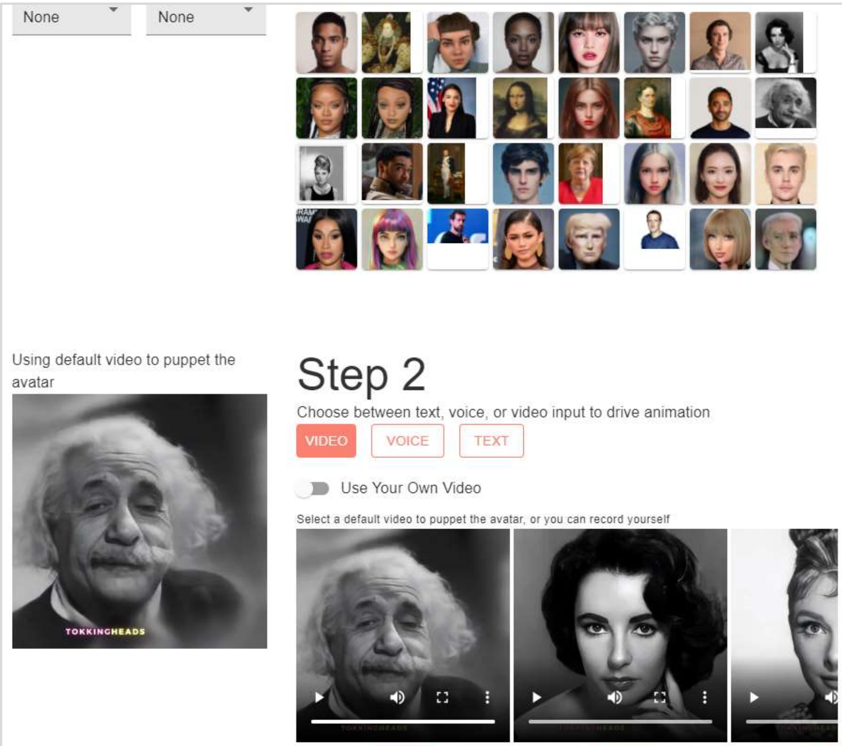
Links: https://openreview.net/forum?id=HkNKFiGex
https://www.youtube.com/watch?time_continue=2&v=FDELBFSeqQs&feature=emb_logo
https://github.com/ajbrock/Neural-Photo-Editor

Generative AI

Toonify! https://toonify.photos/



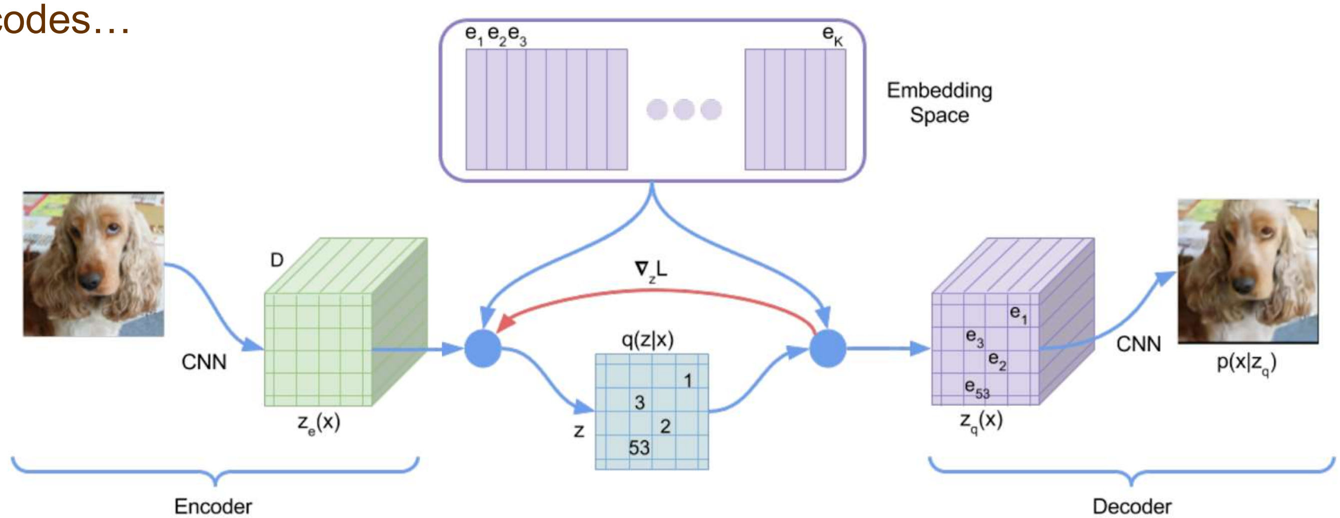
Tokkingheads https://tokkingheads.com/



Relevant links:
https://towardsdatascience.com/animating-yourself-as-a-disney-character-with-ai-78af337d4081

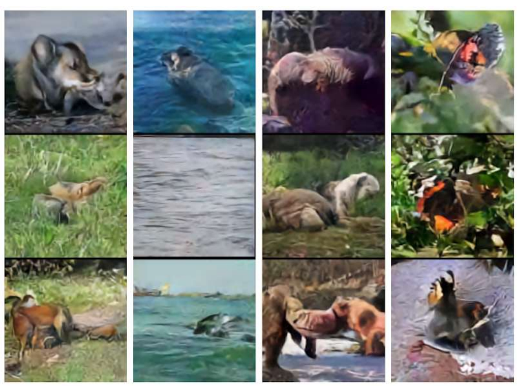
Discrete Latent Spaces

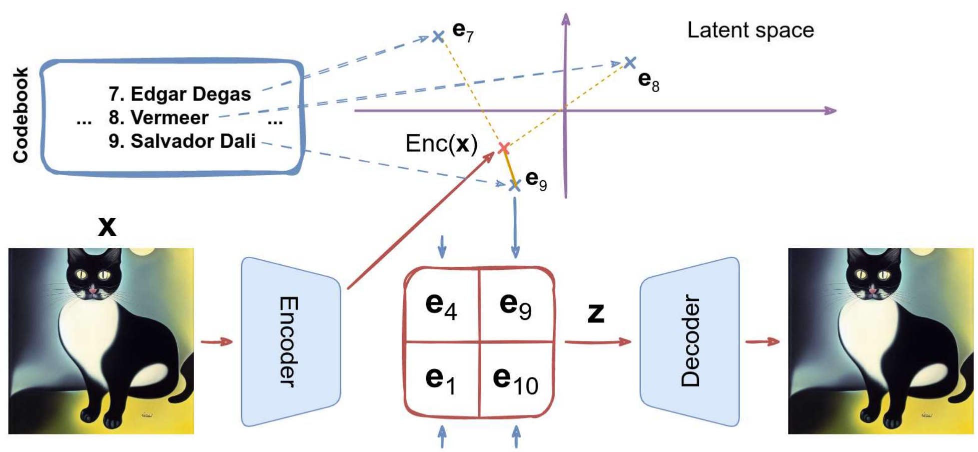
Vector-Quantized VAE (VQ-VAE) - finds a finite vocabulary (codebook) and encodes images as fixed sets (tensors) of discrete codes…



A realistic size of the latent code tensor is something like 32×32 with, say, 8192 codebook vectors (the numbers are taken from the original DALL-E model). Thus, there are 8192^(32×32) = 240960 possibilities, while the number of atoms in the Universe is less than 2^300.

The original VQ-VAE, trained on ImageNet with a separate PixelCNN trained to generate latent codes.

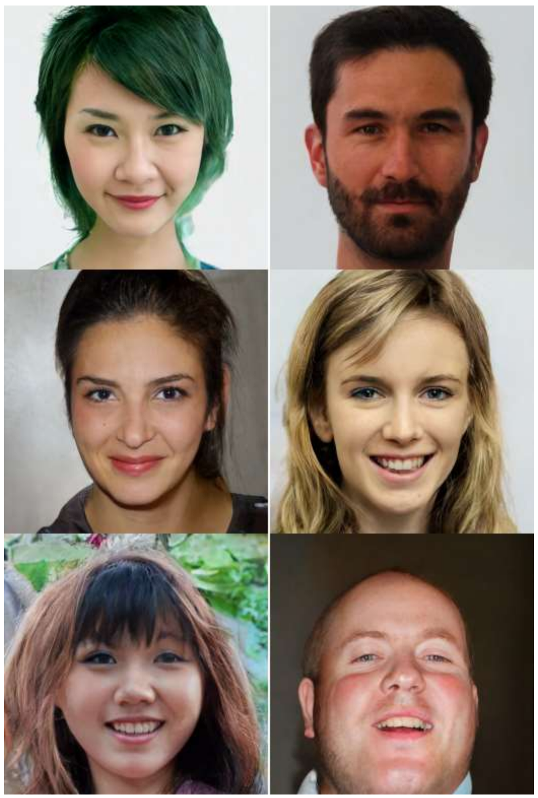


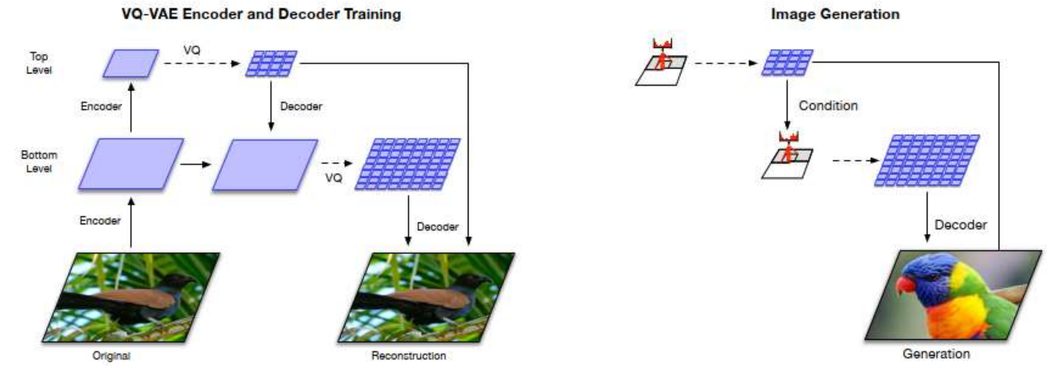


Relevant links:
https://synthesis.ai/2023/03/21/generative-ai-ii-discrete-latent-spaces/
https://arxiv.org/abs/1711.00937

Discrete Latent Spaces

Vector-Quantized Variational Autoencoder (VQ-VAE2) - combination of VAE and Autoregressive models…





(a) Overview of the architecture of our hierarchical VQ-VAE. The encoders and decoders consist of deep neural networks. The input to the model is a 256×256 image that is compressed to quantized latent maps of size 64×64 and 32×32 for the *bottom* and *top* levels, respectively. The decoder reconstructs the image from the two latent maps.

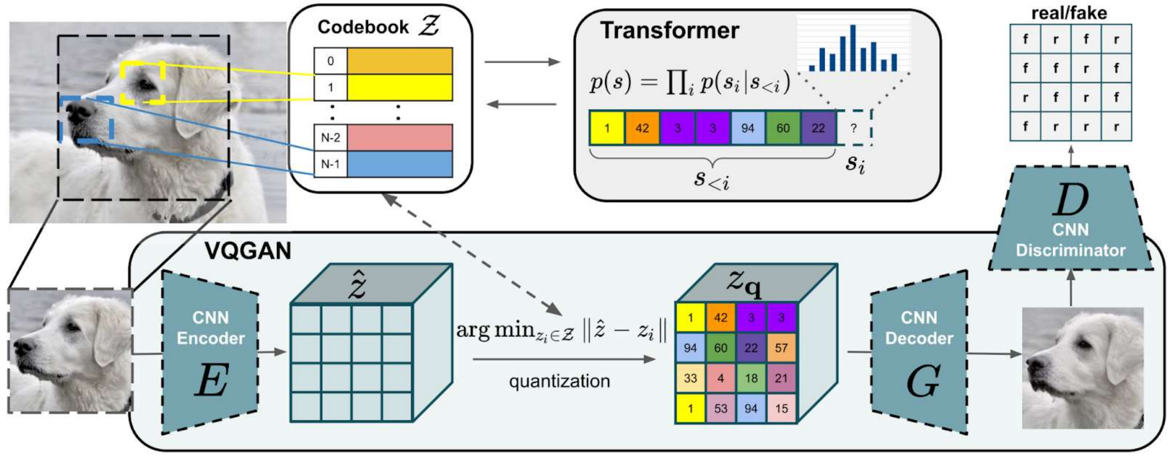
(b) Multi-stage image generation. The top-level PixelCNN prior is conditioned on the class label, the bottom level PixelCNN is conditioned on the class label as well as the first level code. Thanks to the feed-forward decoder, the mapping between latents to pixels is fast. (The example image with a parrot is generated with this model).

Relevant links:
https://arxiv.org/abs/1906.00446
https://paperswithcode.com/method/vq-vae-2

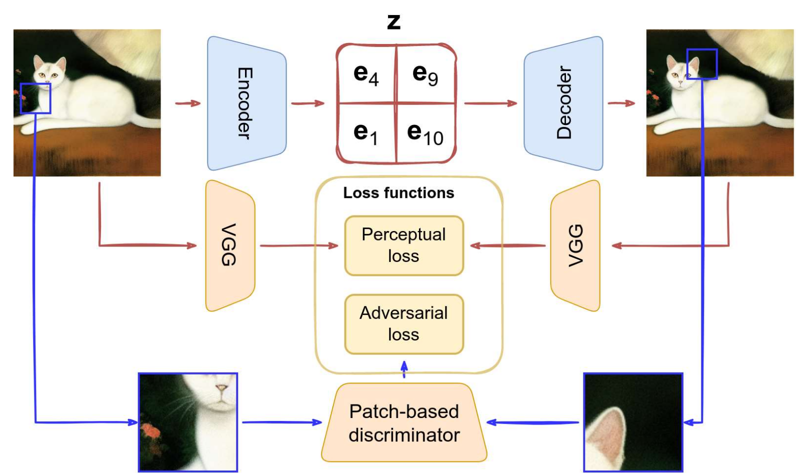
Discrete Latent Spaces

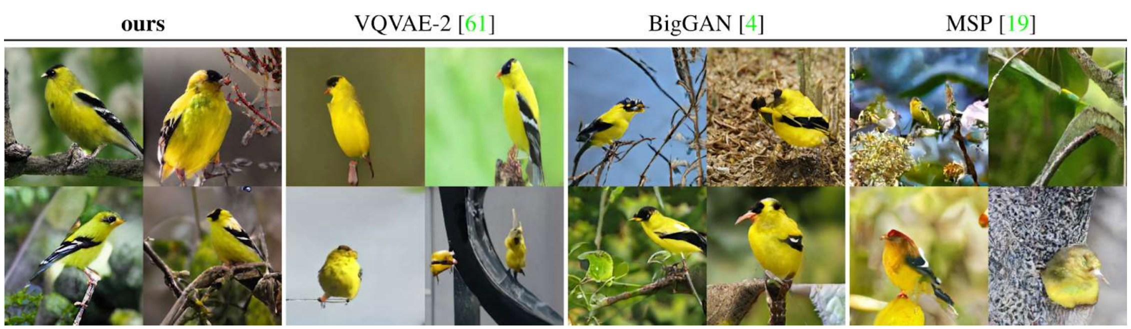
Vector-Quantized GAN (VQ-GAN) naturally uses a Transformer as the autoregressive model to generate the codes and keeps the autoencoder part similar to VQ-VAE. (Esser et al., 2020)

VQ-GAN could not only produce better images on the basic ImageNet, but it could scale to far higher resolutions (e.g. generating a landscape from a semantic layout, i.e., from a rough segmentation map).



To learn a very rich and expressive codebook, VQ-GAN adds a patch-based discriminator that aims to distinguish between (small patches of) real and reconstructed images (instead of using just a straightforward reconstruction loss), and the loss becomes a perceptual loss, i.e., the difference between features extracted by some standard convolutional network. Thus, the discriminator takes care of the local structure of the generate image, and the perceptual loss deals with the actual content.





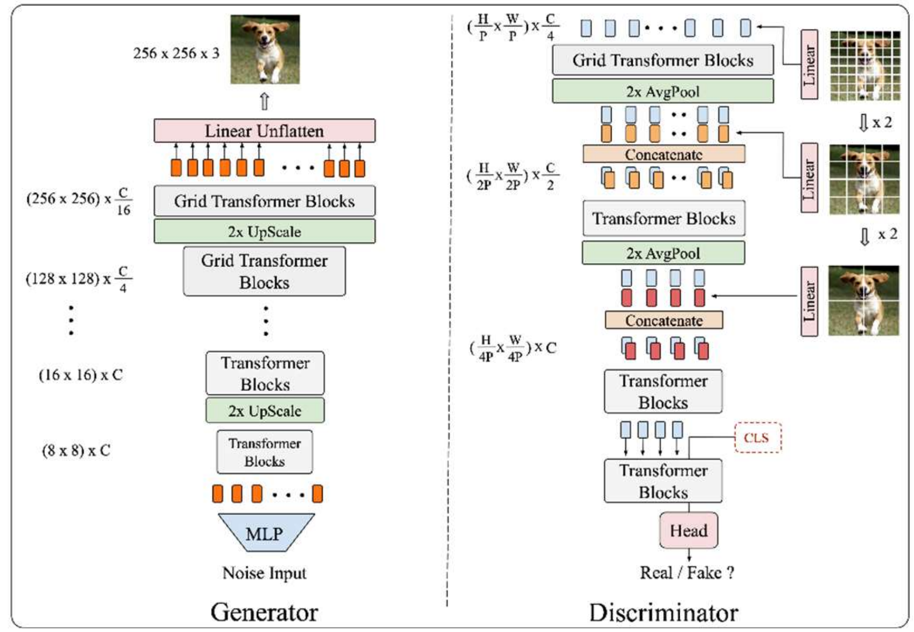
Relevant links:
https://synthesis.ai/2023/03/21/generative-ai-ii-discrete-latent-spaces/
https://arxiv.org/abs/2012.09841

GANs

TransGAN:
Two Pure Transformers Can Make One Strong GAN, and That Can Scale Up

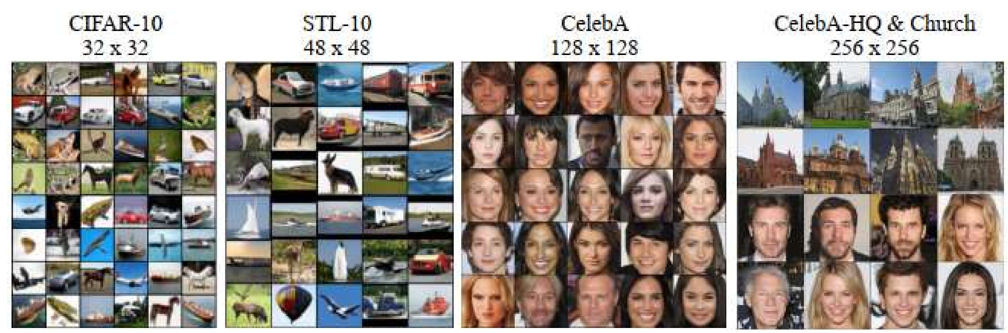
(Jiang et.al., 2021).

Authors conduct the first pilot study in building a GAN completely free of convolutions, using only pure transformer-based architectures. TransGAN, consists of a memory-friendly transformer-based generator that progressively increases feature resolution, and correspondingly a multi-scale discriminator to capture simultaneously semantic contexts and low-level textures. Authors introduce the new module of grid self-attention for alleviating the memory bottleneck further, in order to scale up TransGAN to high-resolution generation, as well as, develop a unique training recipe including a series of techniques that can mitigate the training instability issues of TransGAN, such as data augmentation, modified normalization, and relative position encoding.



Links: https://arxiv.org/abs/2102.07074
https://github.com/VITA-Group/TransGAN

(a) Synthesized Image (b) Interpolation on Latent Space



Relevant links:
https://www.youtube.com/watch?v=R5DiLFOMZrc

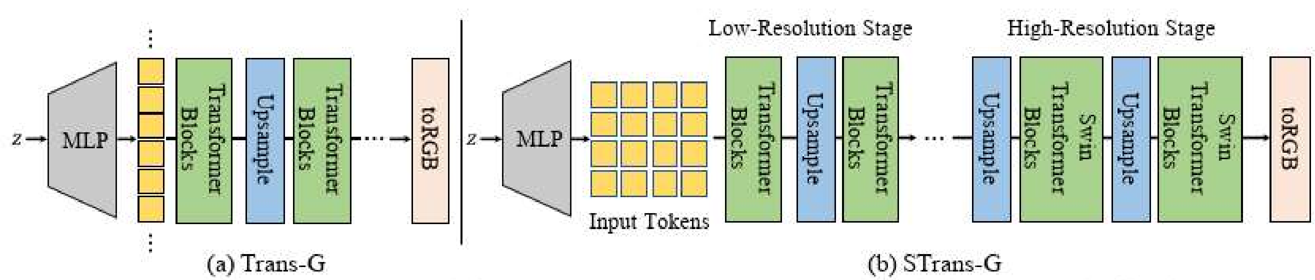
GANs

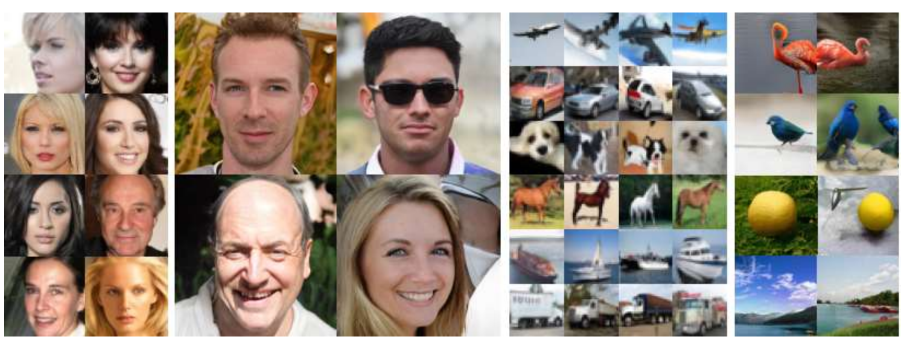
STransGAN: An Empirical Study on Transformer in GANs
The Nuts and Bolts of Adopting Transformer in GANs

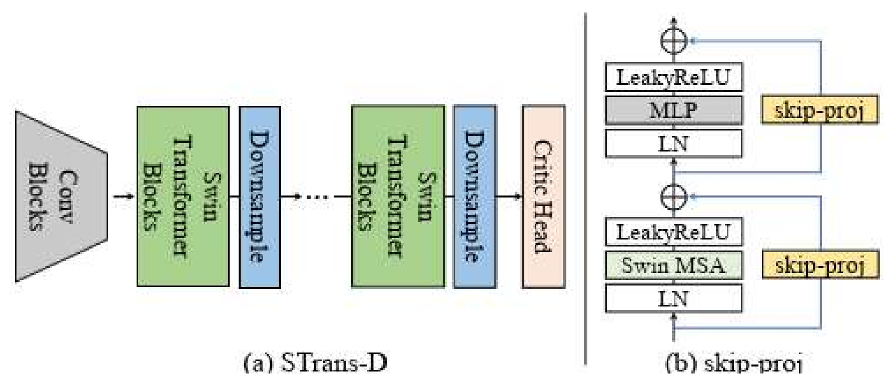
(Xu et.al., 2021).

Authors conduct a comprehensive empirical study to investigate the properties of Transformer in GAN for highfidelity image synthesis. Analysis highlights and reaffirms the importance of feature locality in image generation, although the merits of the locality are well known in the classification task. They have found the residual connections in self-attention layers harmful for learning Transformer-based discriminators and conditional generators and proposed effective ways to mitigate the negative impacts. Study leads to a new alternative design of Transformers in GAN, a convolutional neural network (CNN)-free generator termed as STrans-G, which achieves competitive results in both unconditional and conditional image generations. The Transformer-based discriminator, STrans-D, also significantly reduces its gap against the CNN-based discriminators.

Links: https://arxiv.org/abs/2110.13107
https://nbei.github.io/stransgan.html

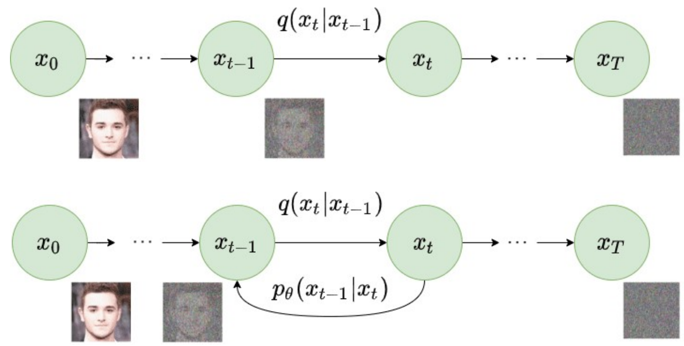


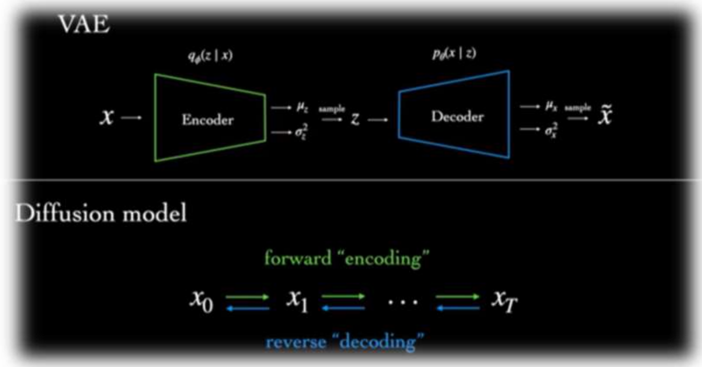




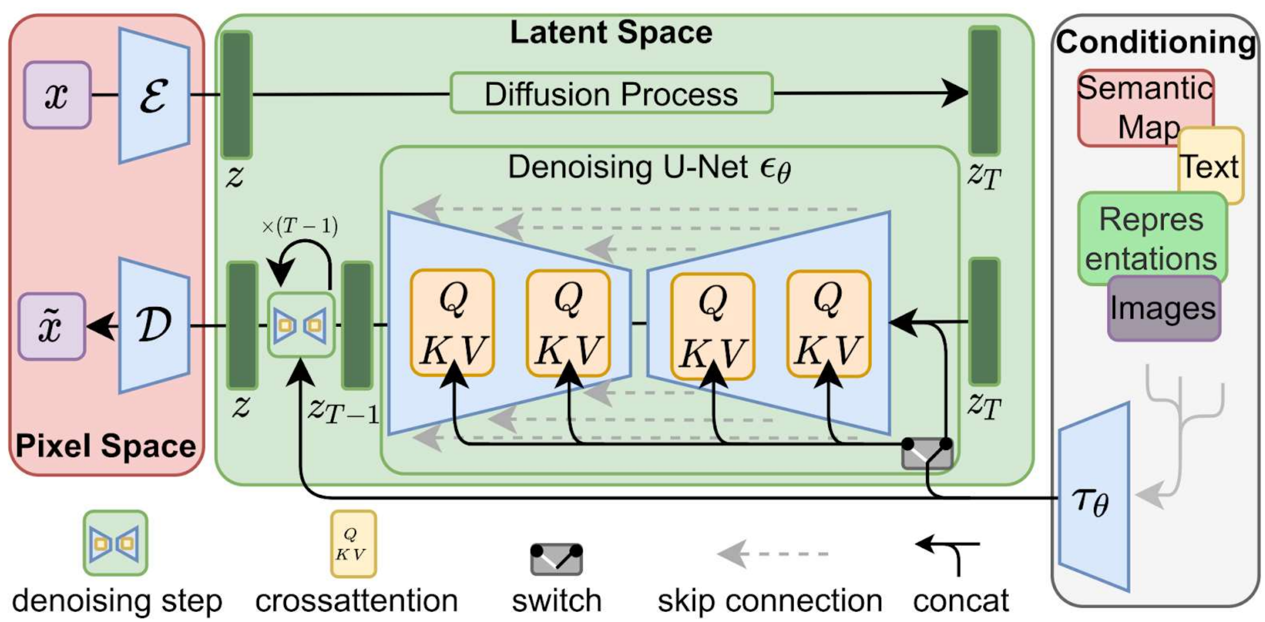
Diffusion models

Diffusion models are fundamentally different from all the previous generative methods. Intuitively, they aim to decompose the image generation process (sampling) in many small "denoising" steps - e.g. *Denoising Diffusion Probabilistic Models (DDPM)* (Sohl-Dickstein et al, 2015)(Ho. et al, 2020)





Stable Diffusion model represents a model where diffusion and denoising processes take place in the *latent space* of autoencoder for images that does mapping of the codes into the pixel space (e.g. *VQ-VAE* or *VQGAN*).

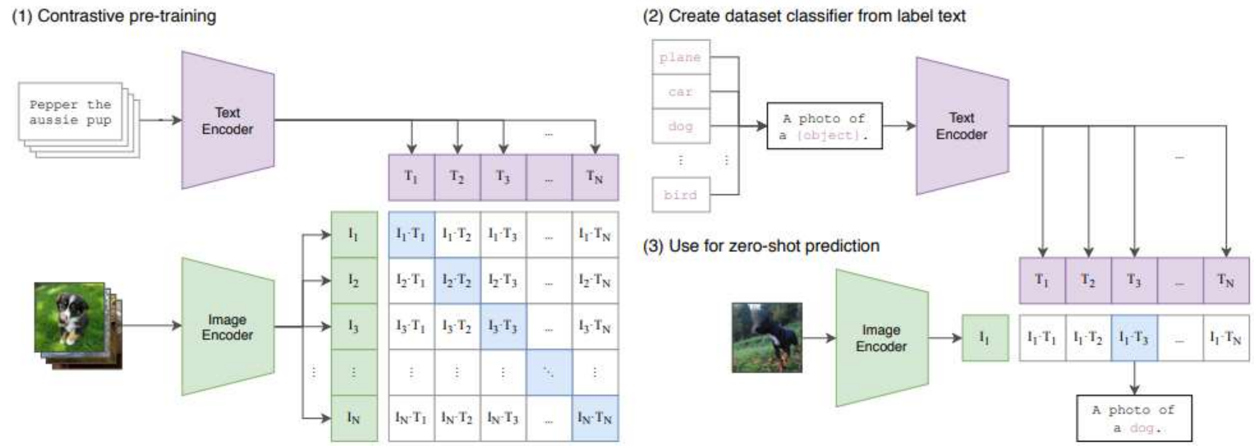


Relevant links:

https://theaisummer.com/diffusion-models/
https://www.youtube.com/watch?v=fbLgFrlTnGU
https://www.youtube.com/watch?v=hVk7Py1c24Q
https://arxiv.org/abs/1503.03585
https://arxiv.org/abs/2006.11239
https://arxiv.org/abs/2106.15282
https://arxiv.org/abs/2011.13456
https://arxiv.org/abs/2112.10752
https://ommer-lab.com/research/latent-diffusion-models/

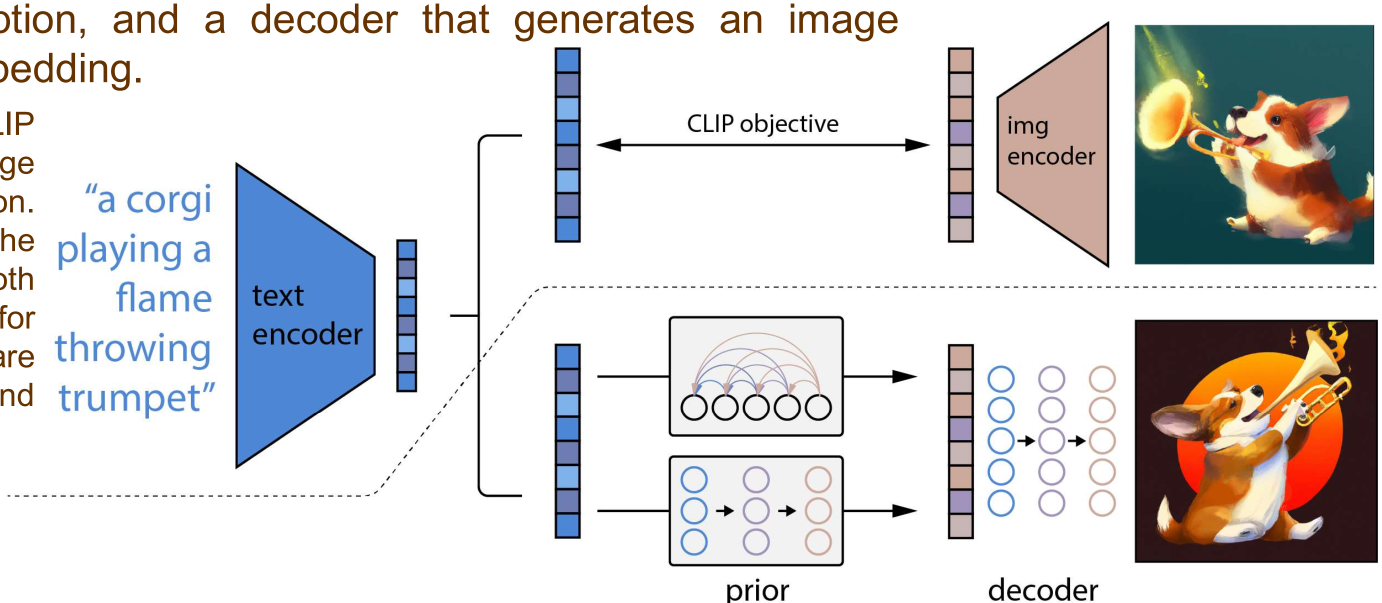
Diffusion models

Contrastive Language-Image Pre-training (CLIP) demonstrates that the simple pre-training task of predicting which caption goes with which image is an efficient and scalable way to learn SOTA image representations from scratch on a dataset of 400 million (image, text) pairs collected from the internet. After pre-training, natural language is used to reference learned visual concepts (or describe new ones) enabling zero-shot transfer of the model to downstream tasks. (Radford et al, 2021)



unCLIP (DALL-E2) *Hierarchical Text-Conditional Image Generation with CLIP Latents* is a two-stage model: a prior that generates a CLIP image embedding given a text caption, and a decoder that generates an image conditioned on the image embedding.

The joint embedding space of CLIP enables language-guided image manipulations in a zero-shot fashion. Authors used diffusion models for the decoder and experiment with both autoregressive and diffusion models for the prior, finding that the latter are computationally more efficient and produce higher-quality samples.



Relevant links:
https://arxiv.org/abs/2103.00020
https://arxiv.org/abs/2204.06125

Diffusion models

Diffusion Transformers (DiTs) explore a new class of diffusion models based on the transformer architecture. Authors train latent diffusion models of images, replacing the commonly-used U-Net backbone with a transformer that operates on latent patches. (Peebles and Xie, 2023)

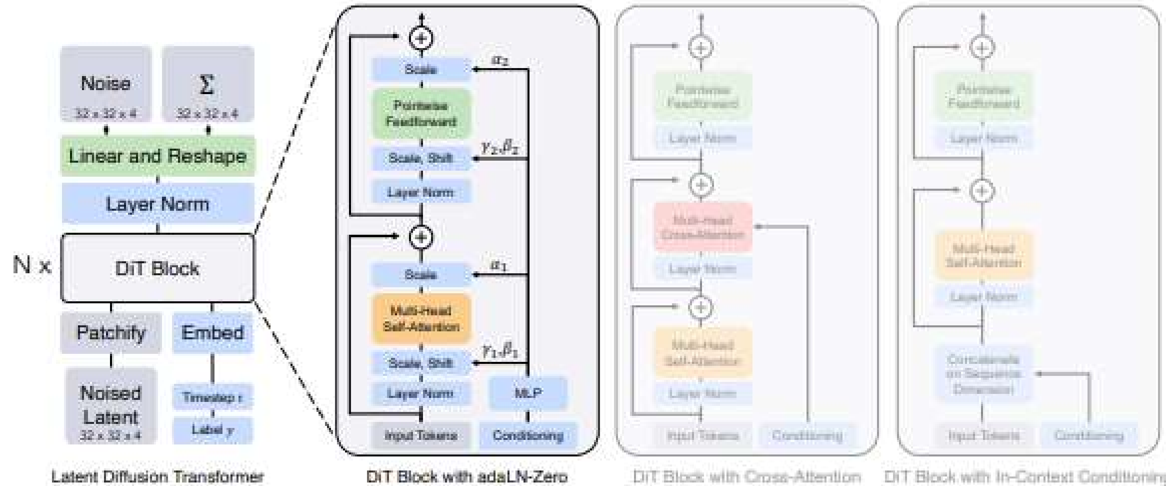


Figure 3. **The Diffusion Transformer (DiT) architecture.** *Left:* We train conditional latent DiT models. The input latent is decomposed into patches and processed by several DiT blocks. *Right:* Details of our DiT blocks. We experiment with variants of standard transformer

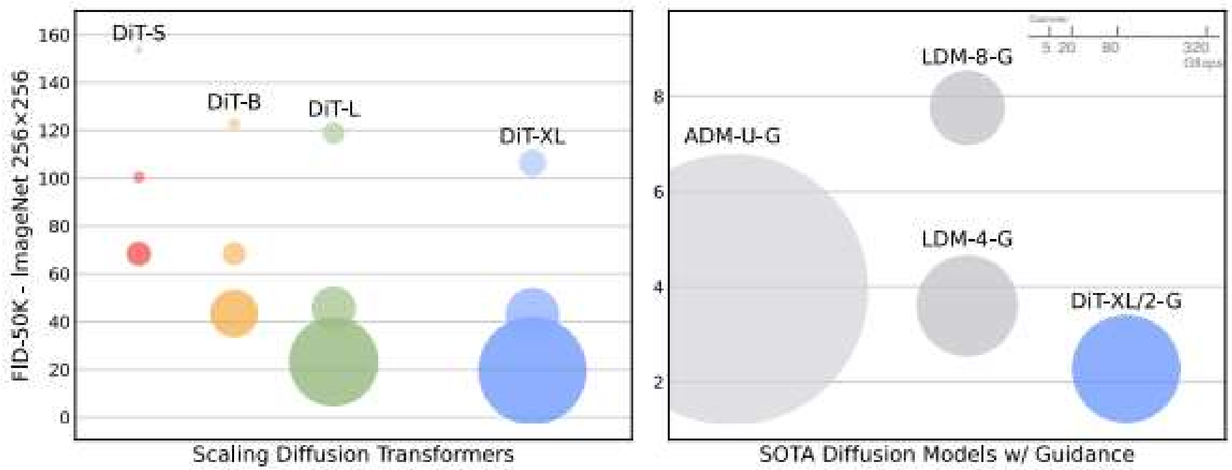


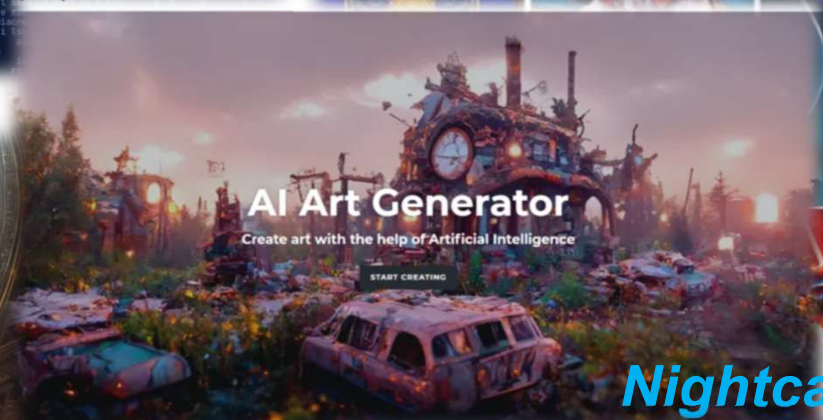
Figure 2. **ImageNet generation with Diffusion Transformers (DiTs).** Bubble area indicates the flops of the diffusion model. *Left:* FID-50K (lower is better) of our DiT models at 400K training iterations. Performance steadily improves in FID as model flops increase. *Right:* Our best model, DiT-XL/2, is compute-efficient and outperforms all prior U-Net-based diffusion models, like ADM and LDM.

Relevant links:
https://arxiv.org/abs/2212.09748
https://github.com/chuanyangjin/fast-DiT

Generative AI

Midjourney www.midjourney.com

DALL-E
https://openai.com/dall-e-2
https://openai.com/dall-e-3



Nightcafe https://nightcafe.studio/

Stable Diffusion https://stability.ai/
https://stablediffusionweb.com/
https://huggingface.co/stabilityai/stable-diffusion-2-1
https://clipdrop.co/stable-diffusion-reimagine

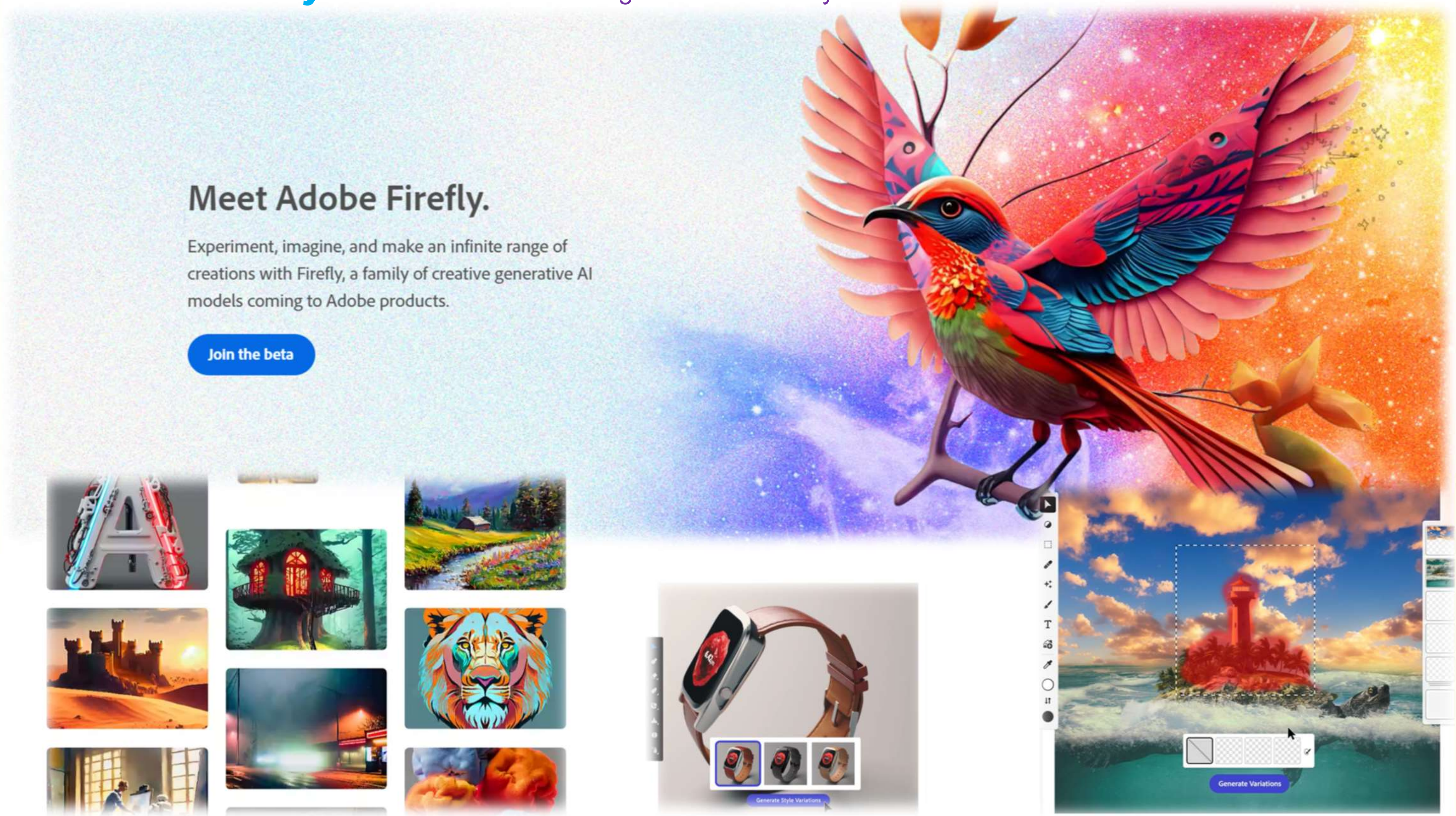
Lexica https://lexica.art

Metaphysic https://metaphysic.ai/

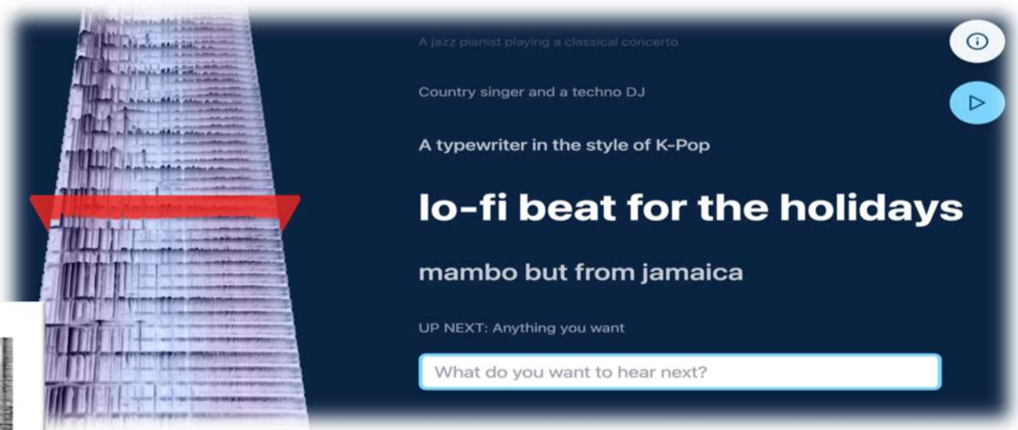
Relevant links:
https://beincrypto.com/learn/ai-image-generators/

Generative AI

Adobe Firefly www.adobe.com/sensei/generative-ai/firefly.html



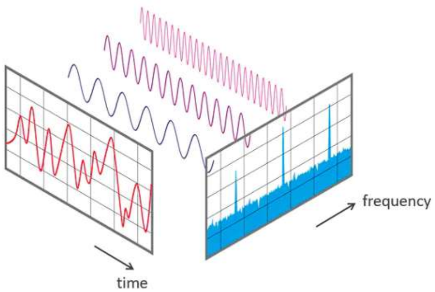
Generative AI

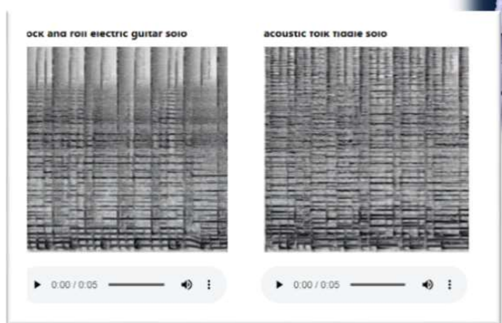


Riffusion https://www.riffusion.com/about

https://huggingface.co/riffusion/riffusion-model-v1

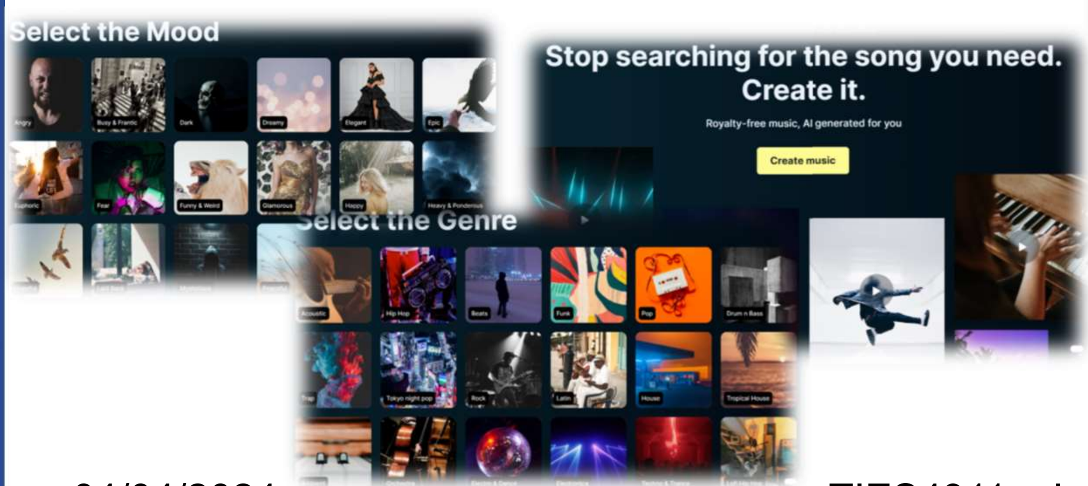
A fine-tuned "Stable Diffusion" model to generate images of spectrograms that are further converted to an audio…





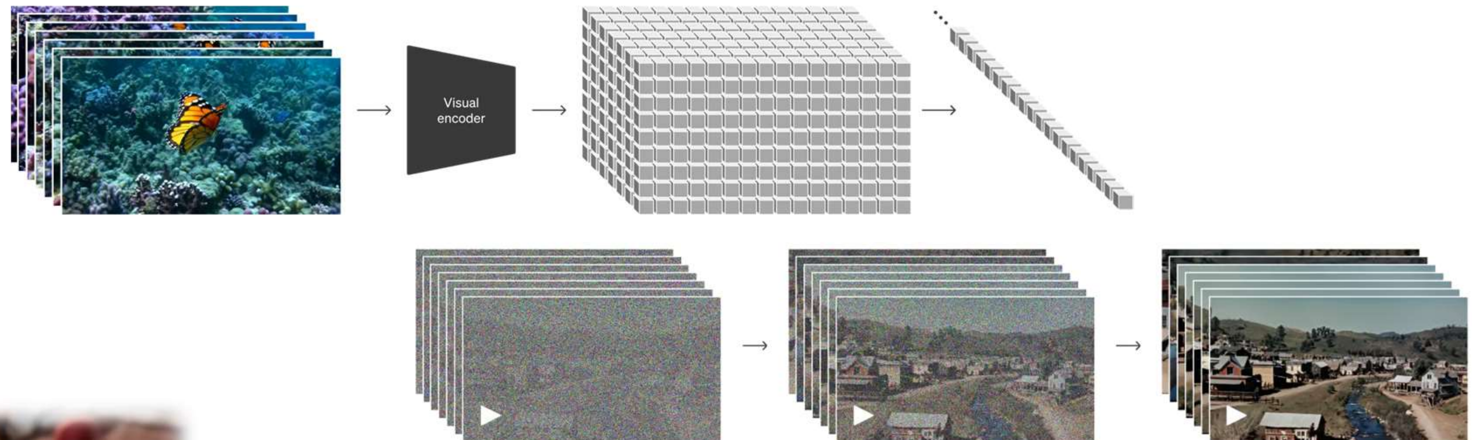
AIVA https://www.aiva.ai/

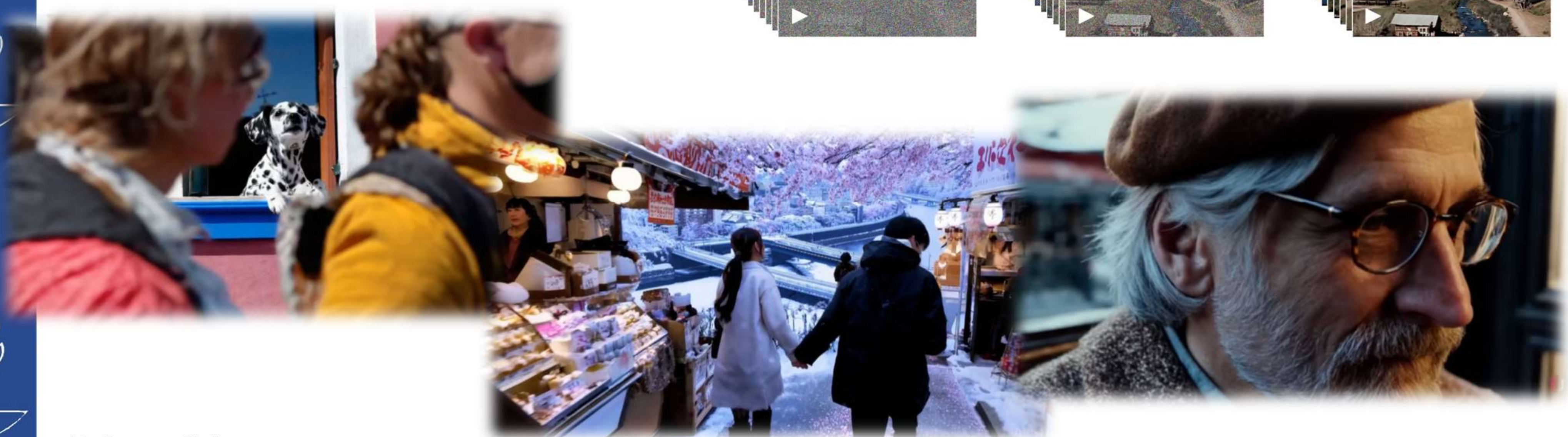
Soundraw https://soundraw.io/



Generative AI

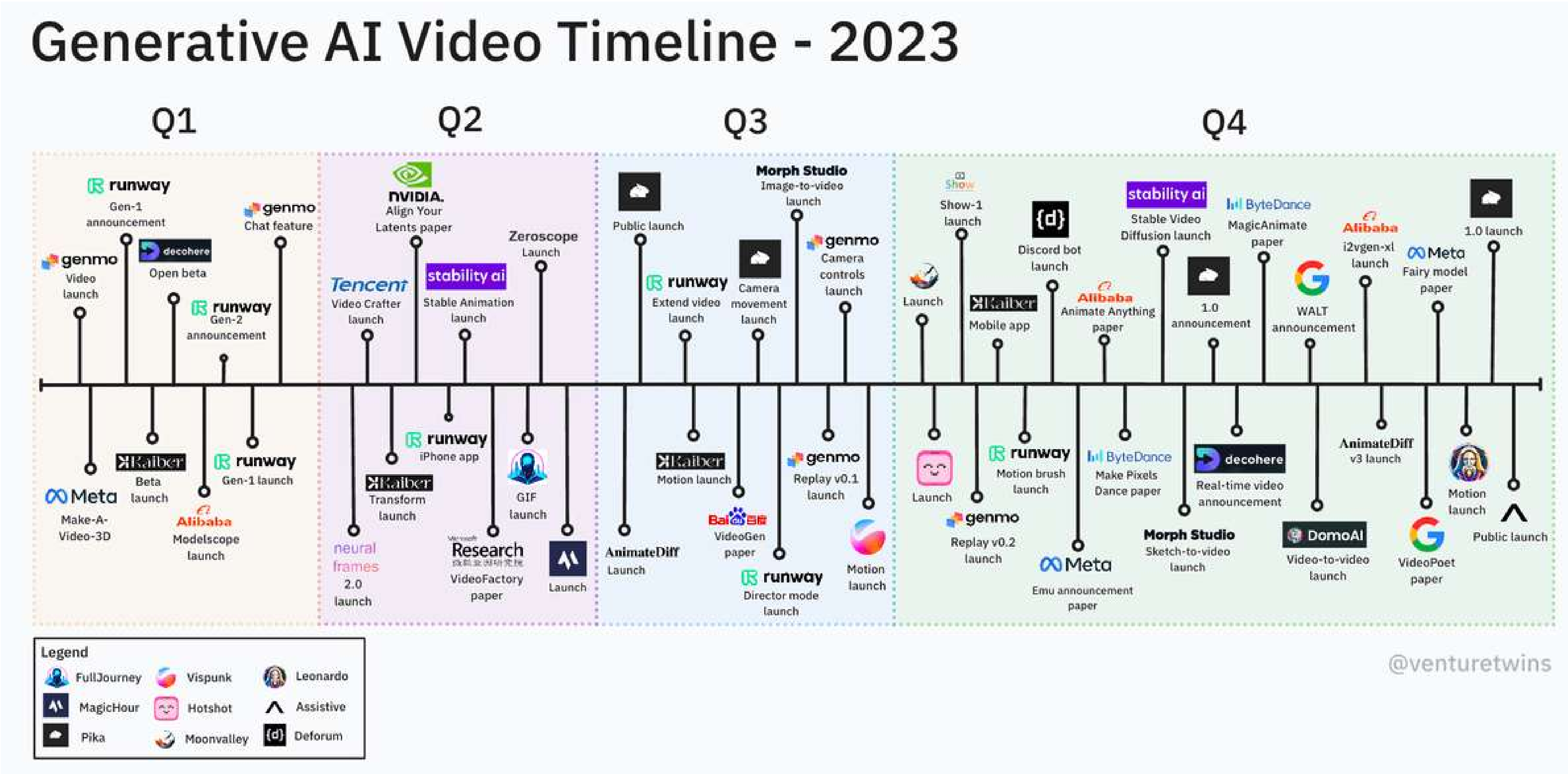
Sora *is a video diffusion model (in particular - diffusion transformer); given input noisy patches (and conditioning information like text prompts), it's trained to predict the original "clean" patches.*
https://openai.com/sora





Relevant links:
https://openai.com/research/video-generation-models-as-world-simulators
https://www.youtube.com/watch?v=hVk7Py1c24Q

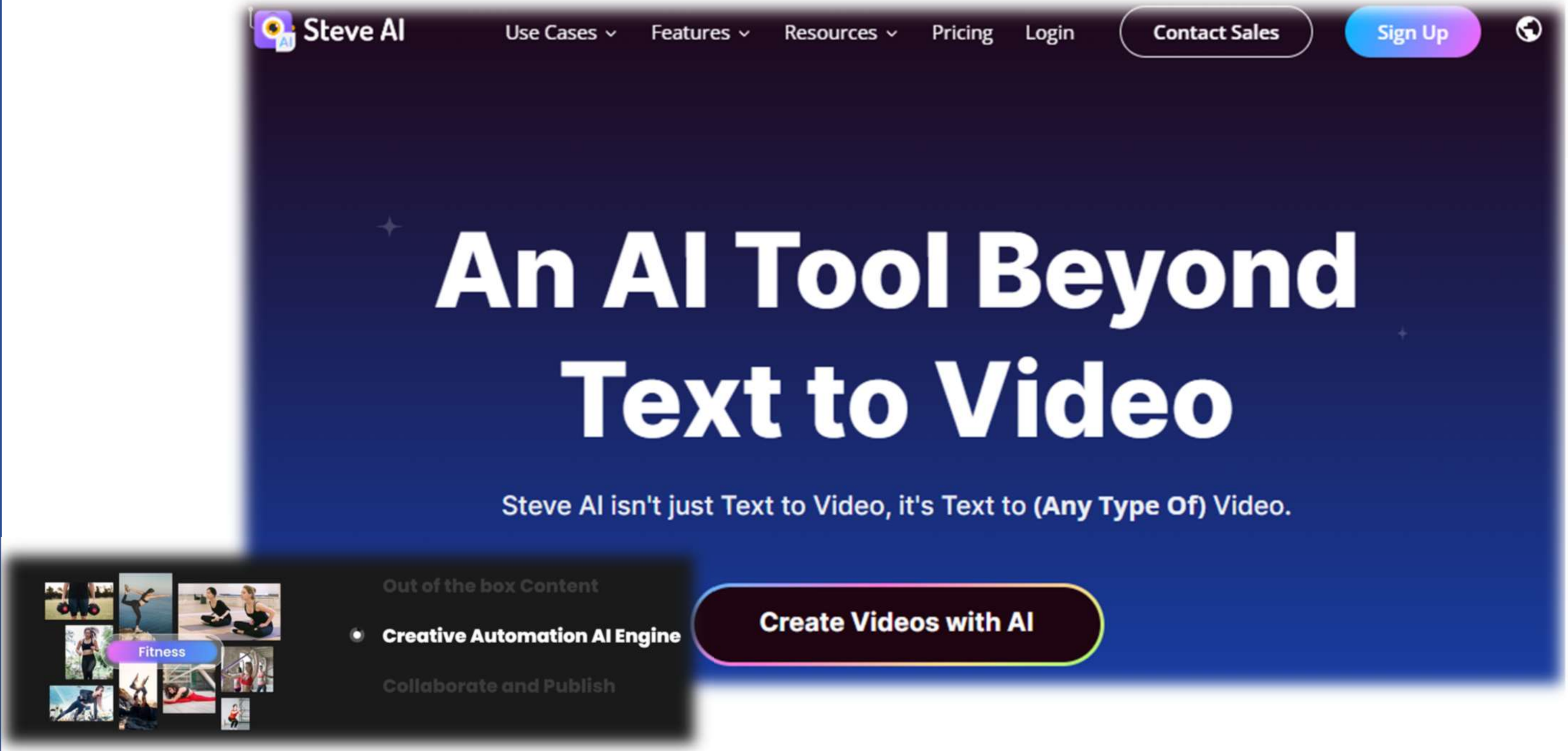
Generative AI



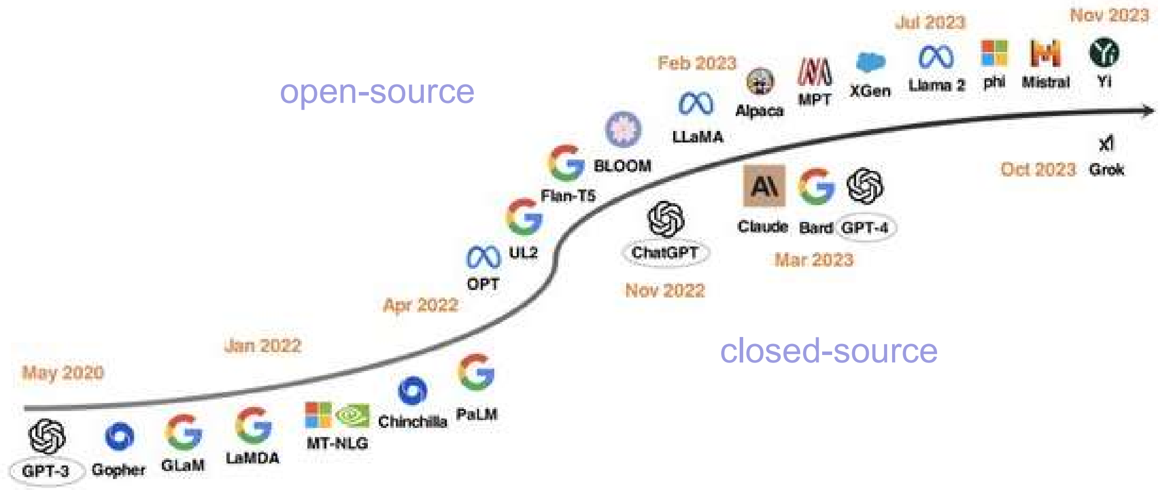
Relevant links:
https://briansolis.com/2024/01/generative-insights-in-ai-january-5-2024/

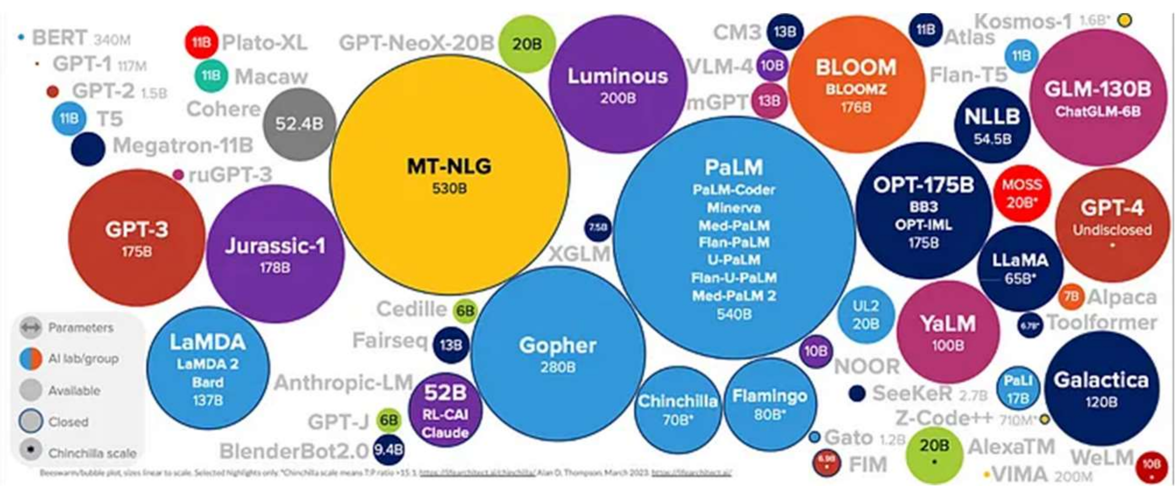
Generative AI

Steve AI https://www.steve.ai/



Generative AI: LLM





LLM	Developer	Popular apps that use it	# of parameters	Access
GPT	OpenAI	Microsoft, Duolingo, Stripe, Zapier, Dropbox, ChatGPT	175 billion+	API
Gemini	Google	Some queries on Bard	Nano: 1.8 & 3.25 billion; others unknown	API
PaLM 2	Google	Google Bard, Docs, Gmail, and other Google apps	340 billion	API
Llama 2	Meta	Undisclosed	7, 13, and 70 billion	Open source
Vicuna	LMSYS Org	Chatbot Arena	7, 13, and 33 billion	Open source
Claude 2	Anthropic	Slack, Notion, Zoom	Unknown	API
Stable Beluga	Stability AI	Undisclosed	7, 13, and 70 billion	Open source
StableLM	Stability AI	Undisclosed	7, 13, and 70 billion	Open source
Coral	Cohere	HyperWrite, Jasper, Notion, LongShot	Unknown	API
Falcon	Technology Innovation Institute	Undisclosed	1.3, 7.5, 40, and 180 billion	Open source
MPT	Mosaic	Undisclosed	7 and 30 billion	Open source
Mixtral 8x7B	Mistral AI	Undisclosed	46.7 billion	Open source
XGen-7B	Salesforce	Undisclosed	7 billion	Open source
Grok	xAI	Grok Chatbot	Unknown	Chatbot

Relevant links:
https://zapier.com/blog/best-llm/
https://www.revelo.com/blog/best-large-language-models
https://medium.com/@kentsui/large-language-model-2023-review-and-2024-outlook-cbd5211cf49b

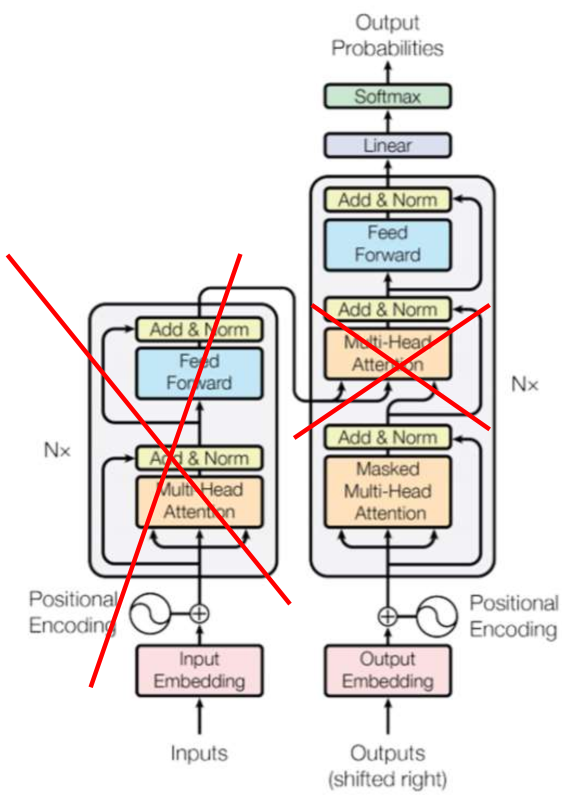
Generative AI

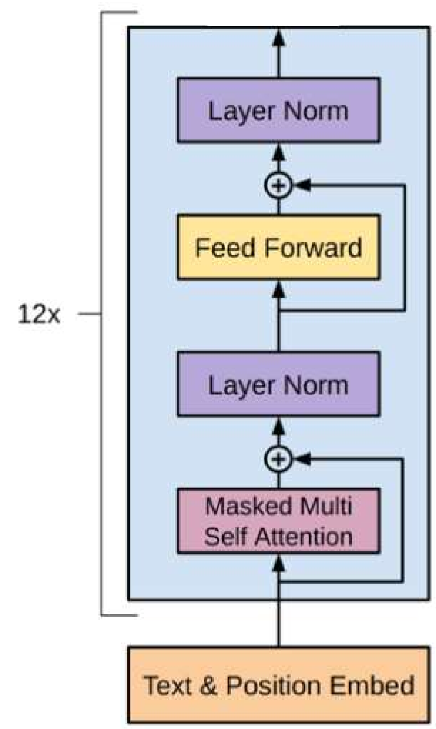
GPT-1

- *Dataset 5GB*
- *Size of the model – 117M parameters*
- *12 Layers*
- *Vocabulary size is 40K tokens*
- *Context (512 tokens)*

Improving Language Understanding by Generative Pre-Training

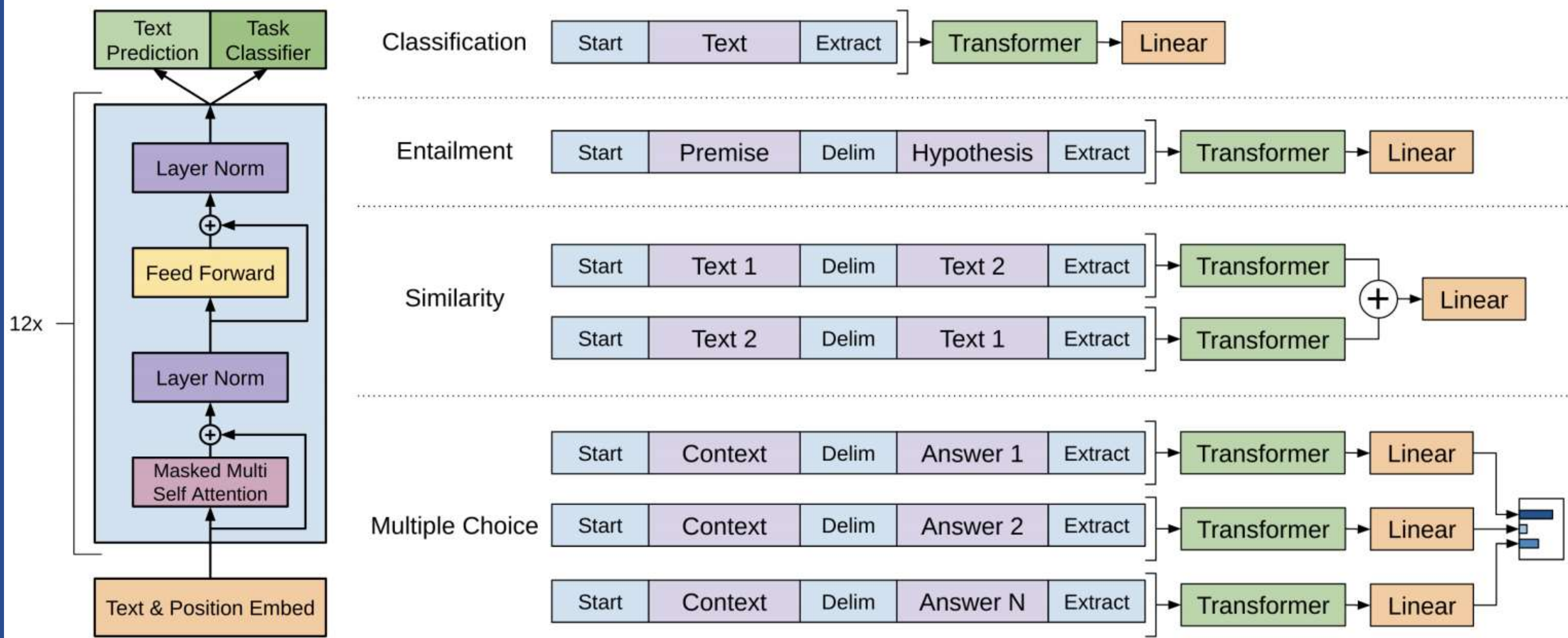
Alec Radford, OpenAI, alec@openai.com
Karthik Narasimhan, OpenAI, karthikn@openai.com
Tim Salimans, OpenAI, tim@openai.com
Ilya Sutskever, OpenAI, ilyasu@openai.com





Generative AI

GPT-1

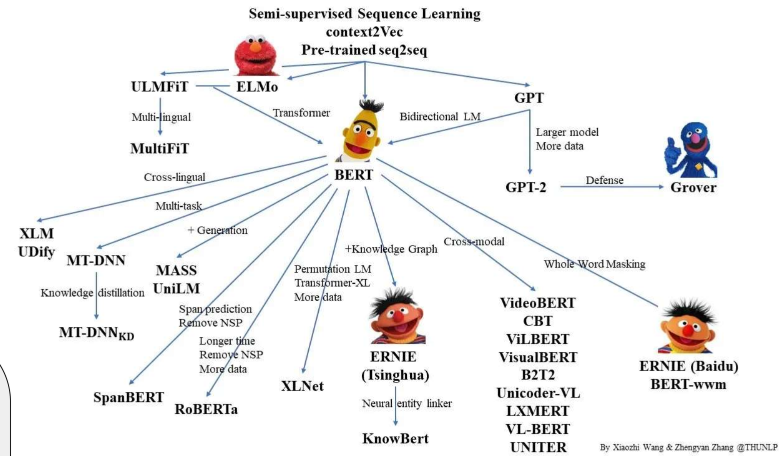


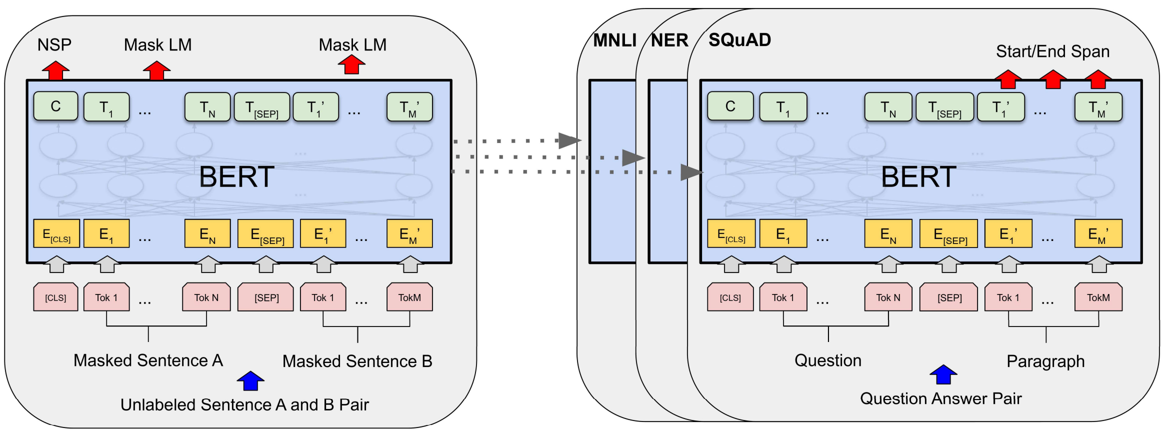
Relevant links:
https://paperswithcode.com/method/gpt

Generative AI

BERT (Bidirectional Encoder Representations from Transformers)

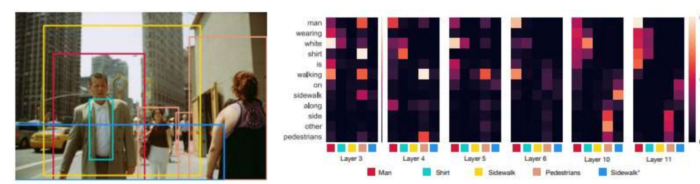
is a language model based on the (encoder-only) transformer architecture, notable for its dramatic improvement over previous state of the art models. It was introduced in October 2018 by researchers at Google. A 2020 literature survey concluded that "in a little over a year, BERT has become a ubiquitous baseline in Natural Language Processing (NLP) experiments counting over 150 research publications analyzing and improving the model."

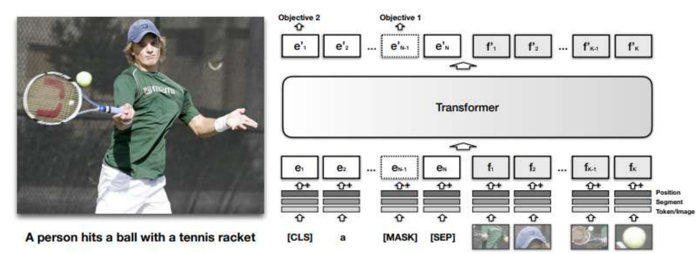




Pre-training

Fine-Tuning





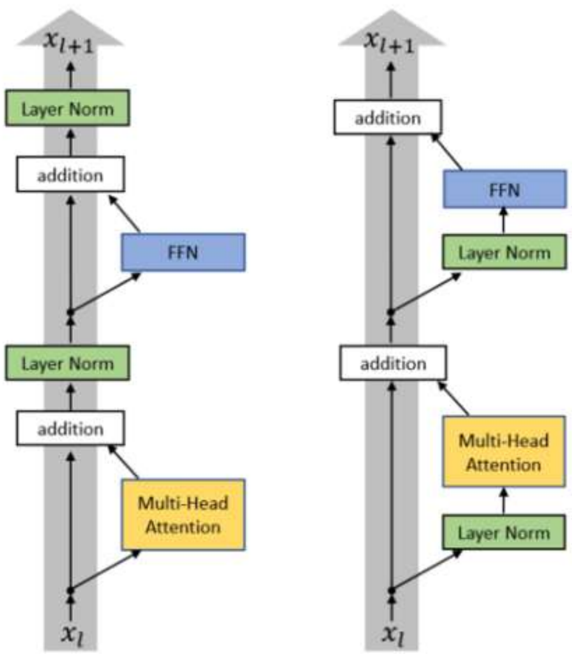
Relevant links:
https://paperswithcode.com/method/bert
https://neptune.ai/blog/bert-and-the-transformer-architecture
https://arxiv.org/abs/1908.03557

Generative AI

GPT-2

- *Larger dataset*
- *Larger size of the model*
- *Vocabulary size is 45K tokens*
- *Extended context (1024 tokens)*

Parameters	Layers	d_{model}
117M	12	768
345M	24	1024
762M	36	1280
1542M	48	1600



(a) Post-LN

(b) Pre-LN

GPT-1:

```
def block(x, scope, train=False, scale=False):
    with tf.variable_scope(scope):
        nx = shape_list(x)[-1]
        a = attn(x, 'attn', nx, n_head, train=train, scale=scale)
        n = norm(x+a, 'ln_1')
        m = mlp(n, 'mlp', nx*4, train=train)
        h = norm(n+m, 'ln_2')
        return h
```

GPT-2:

```
def block(x, scope, *, past, hparams):
    with tf.variable_scope(scope):
        nx = x.shape[-1].value
        a, present = attn(norm(x, 'ln_1'), 'attn', nx, past=past, hparams=hparams)
        x = x + a
        m = mlp(norm(x, 'ln_2'), 'mlp', nx*4, hparams=hparams)
        x = x + m
        return x, present
```

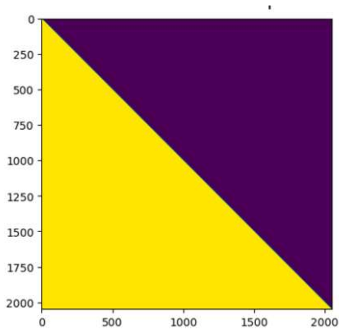
Relevant links:
https://www.catalyzex.com/paper/arxiv:2002.04745

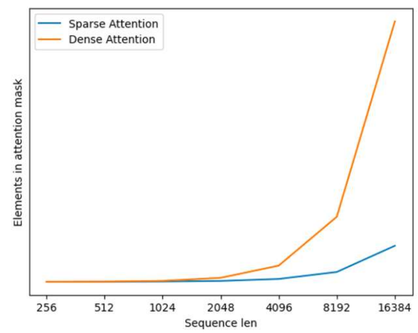
Generative AI

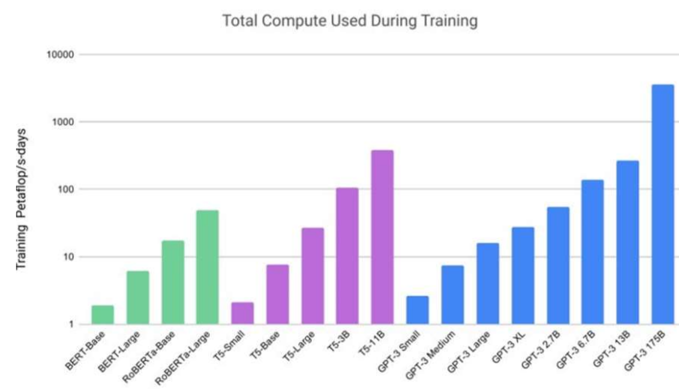
GPT-3

- *Larger size of the model (175B)*
- *Number of layers is 96*
- *Extended context (2048 tokens)*
- *Embedding size is 12288*
- *Number of heads is 96*
- *More computation (10x)*
- *More data (300B tokens)*
- *From Dense to Sparse attention map*

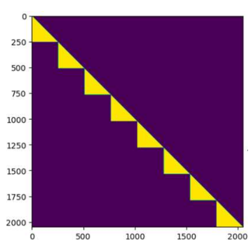
GPT-1	GPT-2	GPT-3
117M	1.5B	175B
12	48	96
512	1024	2048
768	1600	12288
12 (d_{head} = 64)	25 (d_{head} = 64)	96 (d_{head} = 128)

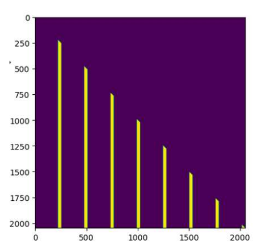


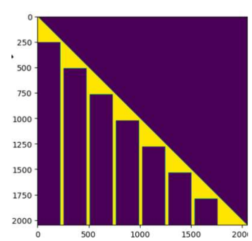




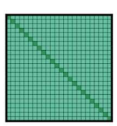
Model	Total train compute (PF-days)	Total train compute (flops)
T5-3B	1.04E+02	9.00E+21
T5-11B	3.82E+02	3.30E+22
GPT-3 13B	2.68E+02	2.31E+22
GPT-3 175B	3.64E+03	3.14E+23



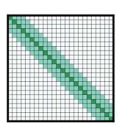




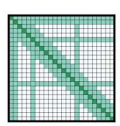
Dataset	Quantity (tokens)	Weight in training mix	Epochs elapsed when training for 300B tokens
Common Crawl (filtered)	410 billion	60%	0.44
WebText2	19 billion	22%	2.9
Books1	12 billion	8%	1.9
Books2	55 billion	8%	0.43
Wikipedia	3 billion	3%	3.4



(a) Full n^2 attention



(b) Sliding window attention



(d) Global+sliding window

However, we may still use a dense attention without modifications if we apply gradient checkpointing during the training !!!… Plus, use of Flash Attention also speed up the training process.

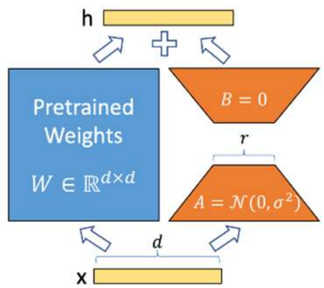
LLM Fine-Tuning Techniques

Full fine-tuning *results in a new version of the model with updated weights. Just like pre-training, full fine tuning requires enough memory and compute budget to store and process all the gradients, optimizers and other components that are being updated during training.*

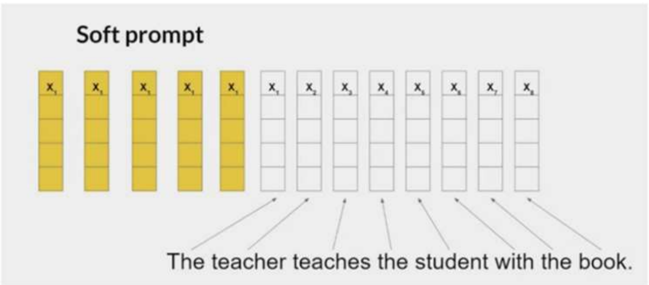
Parameter efficient fine-tuning *, in contrast to full fine-tuning, only update a small subset of parameters.*

https://huggingface.co/docs/peft/en/conceptual_guides/adapter , https://huggingface.co/docs/peft/en/conceptual_guides/prompting

- ***LoRA*** *(Low-rank Adaptation) is a parameter-efficient fine-tuning technique that falls into the re-parameterization category.*
  https://arxiv.org/pdf/2309.15223.pdf



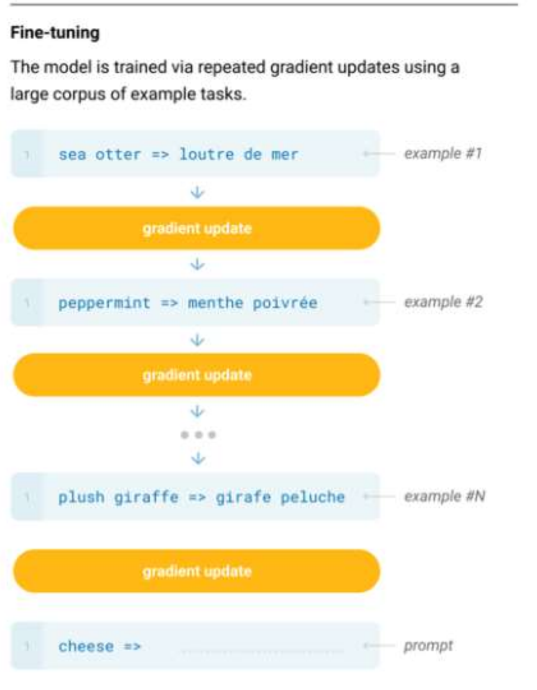
- ***Soft prompting*** *With prompt tuning, you add additional trainable tokens to your prompt and leave it up to the supervised learning process to determine their optimal values. The set of trainable tokens is called a soft prompt, and it gets prepended to embedding vectors that represent your input text.*



Reinforcement learning by human feedback (RLHF) *resulting in a model that is better aligned with human preferences. Use RLHF to make sure that the model produces outputs that maximize usefulness and relevance to the input prompt. Perhaps most importantly, RLHF can help minimize the potential for harm. Train the model to give caveats that acknowledge their limitations and to avoid toxic language and topics.*

In-context Learning

Traditional fine-tuning (not used for GPT-3)



Zero-shot

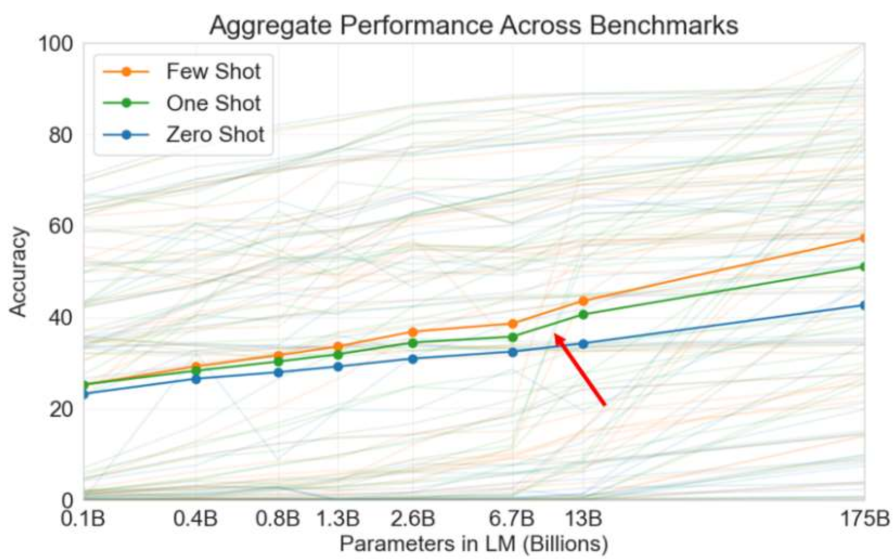
The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

One-shot

In addition to the task description, the model sees a single example of the task. No gradient updates are performed.

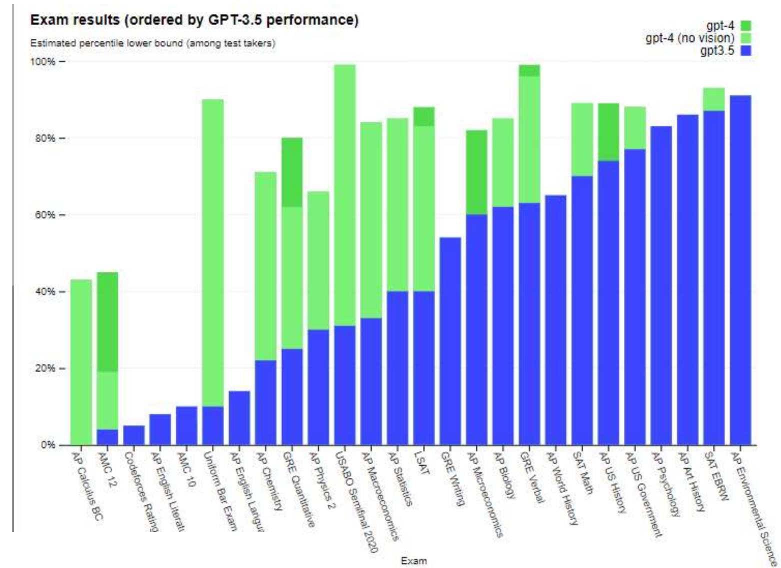
Few-shot

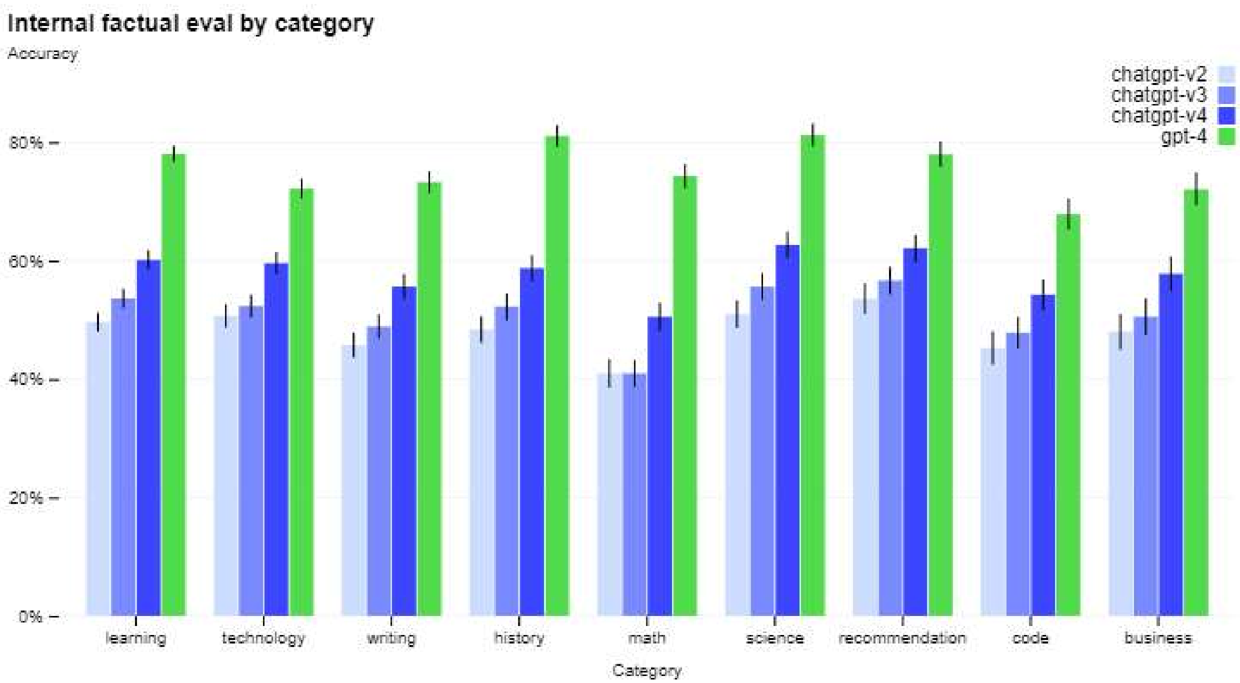
In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.



Generative AI

GPT-4 *is OpenAI’s most advanced system, producing safer and more useful responses. It is a large multimodal model (accepting image and text inputs, emitting text outputs) that, while less capable than humans in many real-world scenarios, exhibits human-level performance on various professional and academic benchmarks.*
https://openai.com/gpt-4
https://openai.com/research/gpt-4





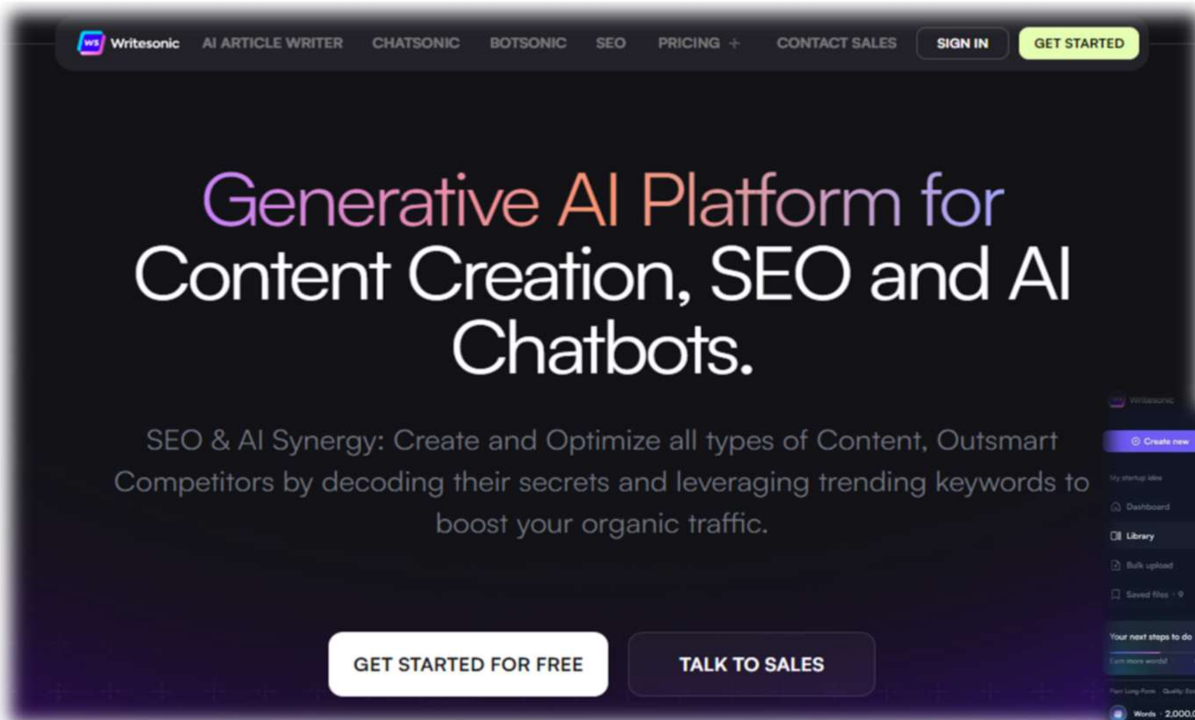
- *Use of external tools and services*
- *Code writing and execution*
- *GPTs and their store*
- *…*

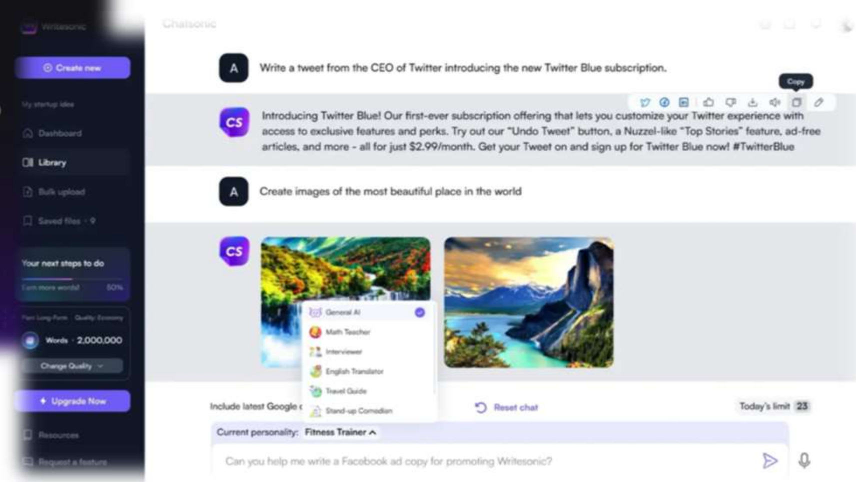
Generative AI

Writesonic / ChatSonic

https://writesonic.com https://writesonic.com/chat

A revolutionary AI like Chat GPT - ChatSonic (now with GPT-4 capabilities), the conversational AI chatbot addresses the limitations of ChatGPT, turning out to be the best Chat GPT alternative.

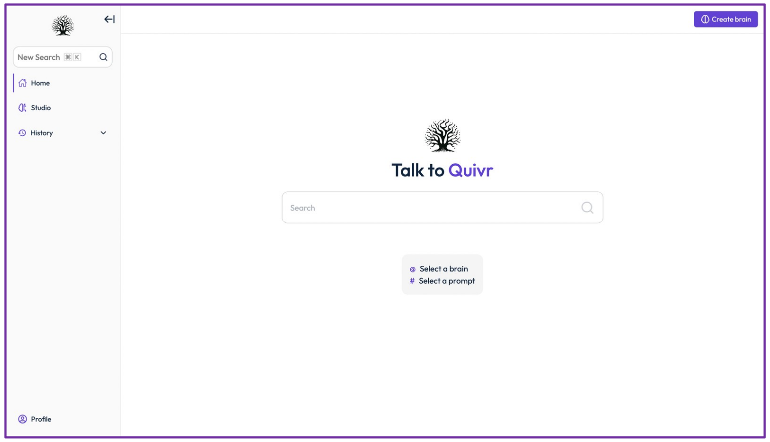


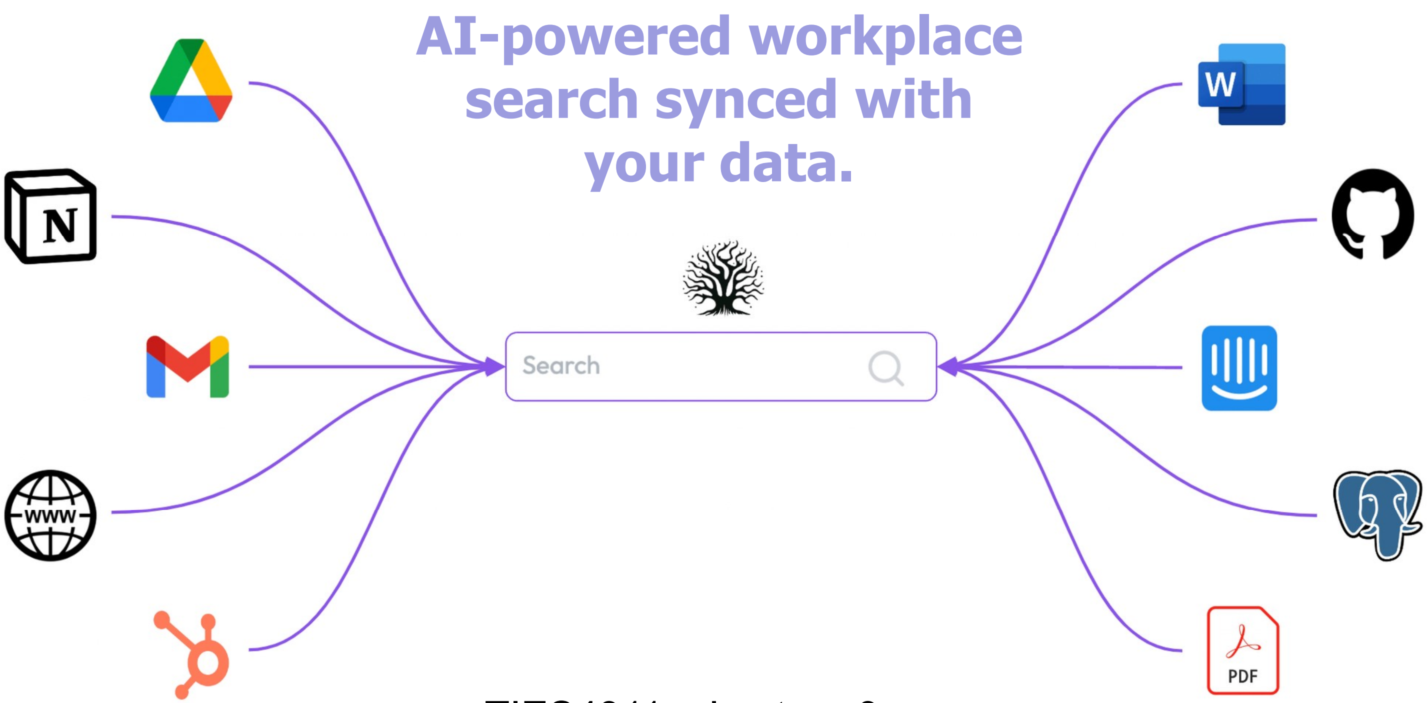


Relevant links:
https://openai.com/blog/chatgpt
https://openai.com/blog/chatgpt-plus
https://chat.openai.com

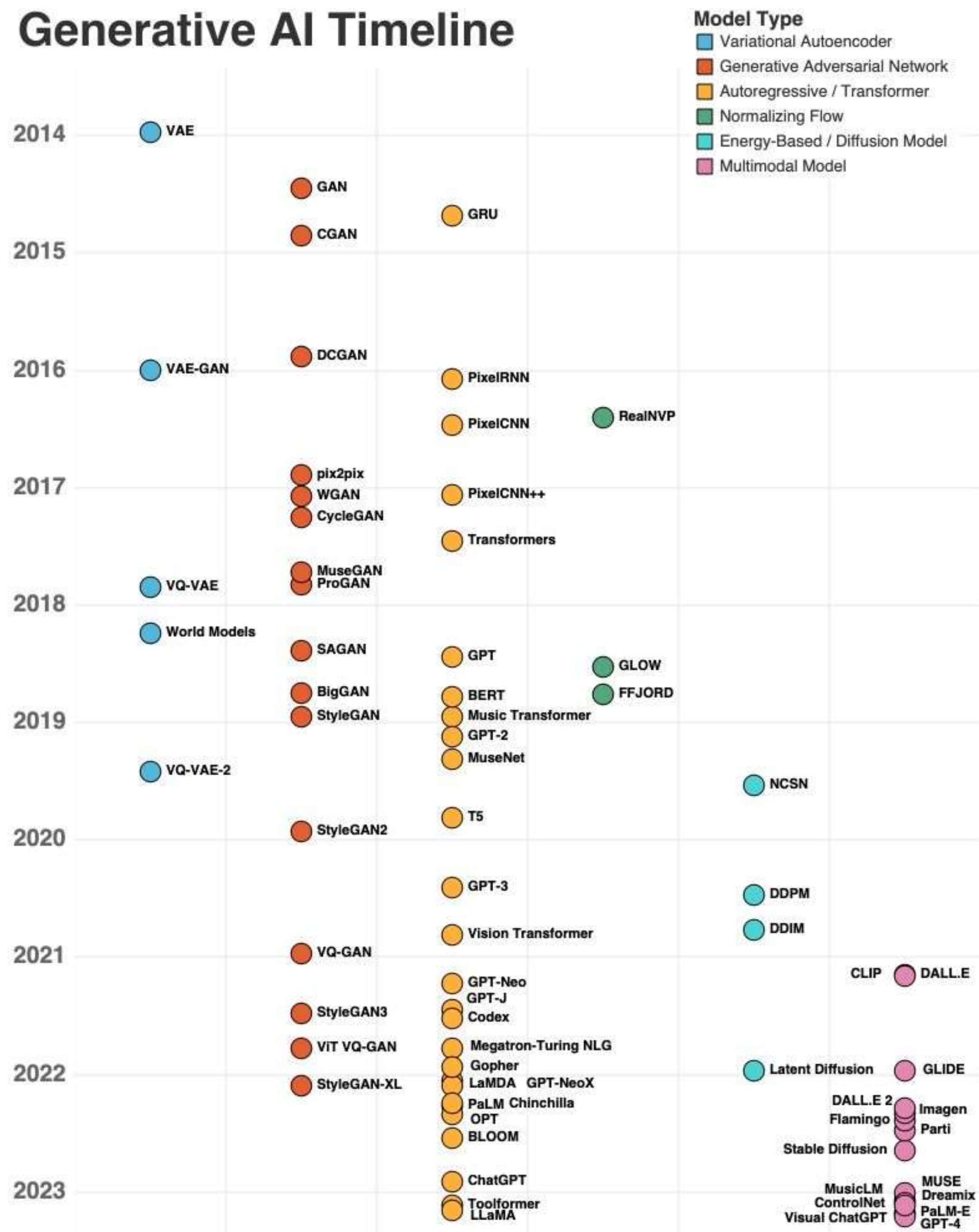
LLM-powered Search

Quivr *is an open source chat-powered second brains to build a unified search engine across all your documents, tools, and databases.* https://www.quivr.app





Generative AI



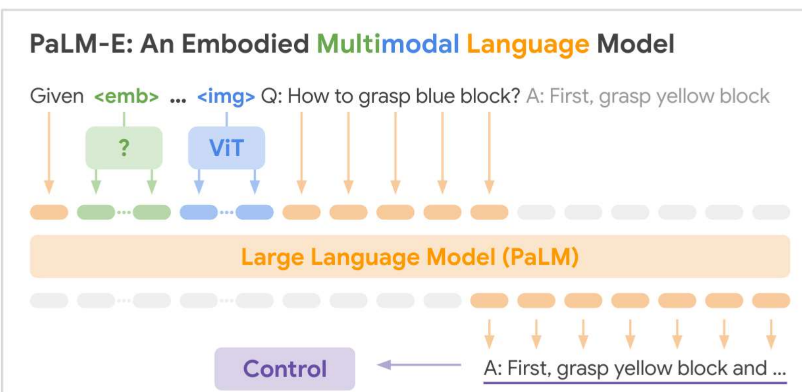
Relevant links:

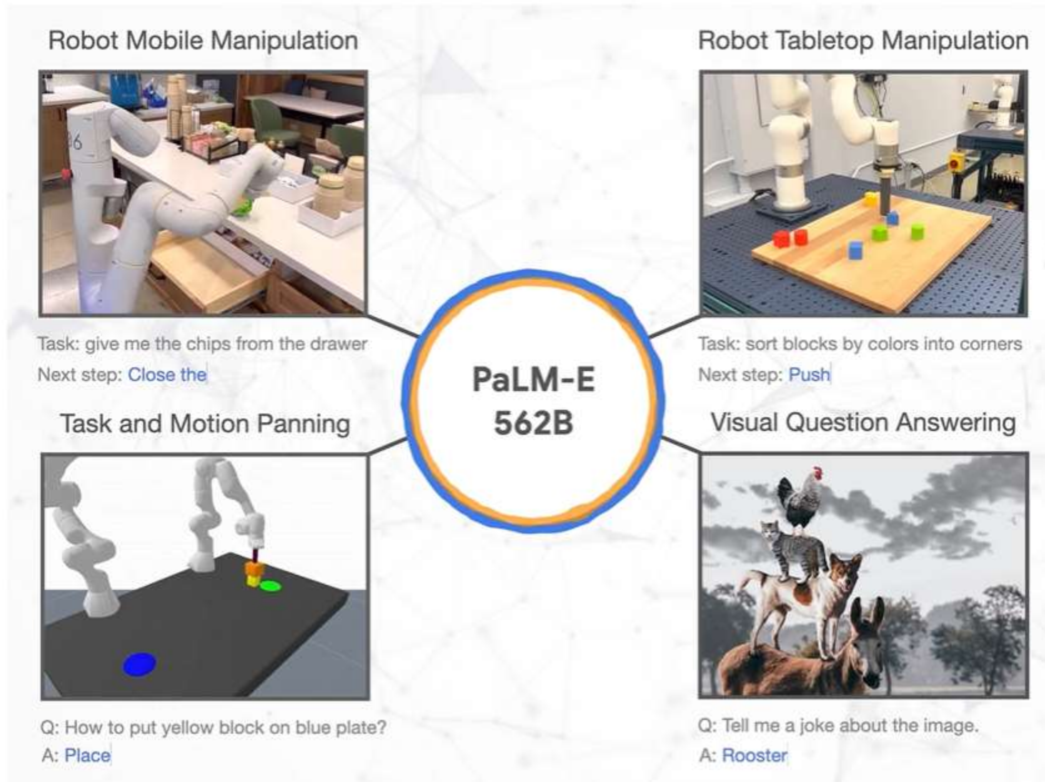
https://www.linkedin.com/posts/davidtfoster_datascience-machinelearning-ai-activity-7044233450295316480-nd31/

Multimodal models

LLaVA (Large Language-and-Vision Assistant) *is an end-to-end trained large multimodal model that connects a vision encoder and LLM (Vicuna) for general-purpose visual and language understanding, achieving impressive chat capabilities mimicking spirits of the multimodal GPT-4 and setting a new state-of-the-art accuracy on Science QA.* https://llava-vl.github.io/

PaLM-E *is an embodied multimodal language model. It is a new generalist robotics model that transfers knowledge from varied visual and language domains to a robotics system. PaLM-E combines our most recent large language model, PaLM, together with one of our most advanced vision models, ViT-22B.* https://blog.research.google/2023/03/palm-e-embodied-multimodal-language.html





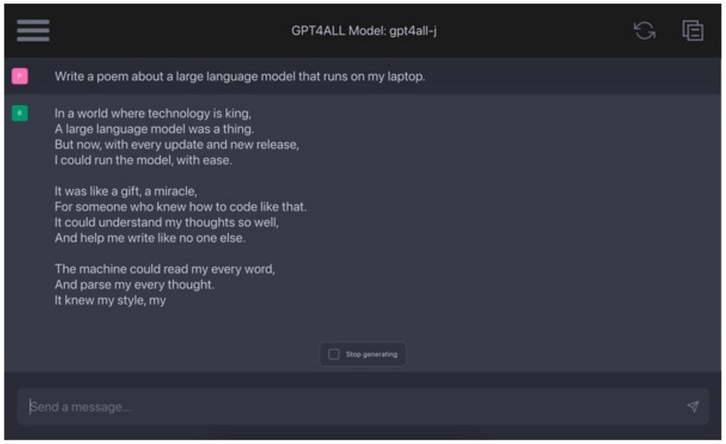
Multimodal models

FlashAttention-2: *Faster attention with better parallelism and work partitioning.*
https://www.together.ai/blog/tri-dao-flash-attention

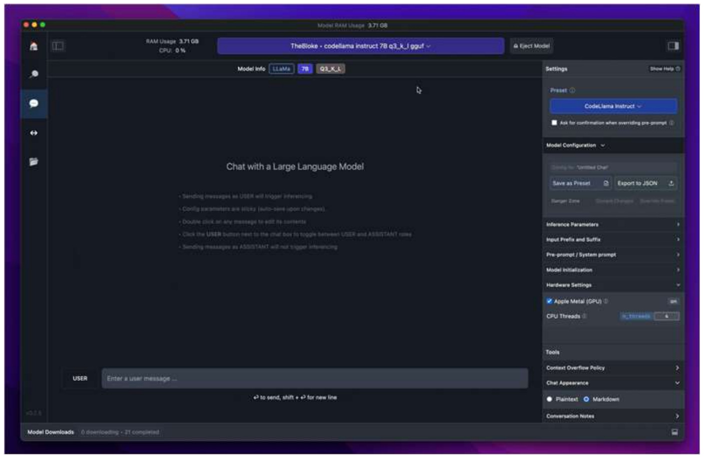
"LLMs allow humans to talk to AI, and FlashAttention is critical to allow for the longest possible context lengths while maintaining an interactive experience. It is amazing to see how FlashAttention-2 not only has doubled performance with the help of NVIDIA CUTLASS and CuTe on A100, but now is four times the original performance when using H100 without any additional code changes," said Vijay Thakkar, Senior Compute Architect at NVIDIA. *"We look forward to working with researchers to further optimize and help bring the next generation LLMs to the world."*

LLM related tools

GPT4ALL - *a free-to-use, locally running, privacy-aware chatbot. No GPU or internet required. GPT4All is an ecosystem to train and deploy powerful and customized large language models that run locally on consumer grade CPUs. The goal is simple - be the best instruction tuned assistant-style language model that any person or enterprise can freely use, distribute and build on. A GPT4All model is a 3GB - 8GB file that you can download and plug into the GPT4All open-source ecosystem software. Nomic AI supports and maintains this software ecosystem to enforce quality and security alongside spearheading the effort to allow any person or enterprise to easily train and deploy their own on-edge large language models.*
https://gpt4all.io/index.html , https://github.com/nomic-ai/gpt4all



LM Studio *supports to discover, download, and run local LLMs. With LM Studio, you can run LLMs on your laptop, entirely offline, use models through the in-app Chat UI or an OpenAI compatible local server, download any compatible model files from HuggingFace 🤗 repositories, discover new & noteworthy LLMs in the app's home page. LM Studio supports any ggml Llama, MPT, and StarCoder model on Hugging Face (Llama 2, Orca, Vicuna, Nous Hermes, WizardCoder, MPT, etc.).*
https://lmstudio.ai/

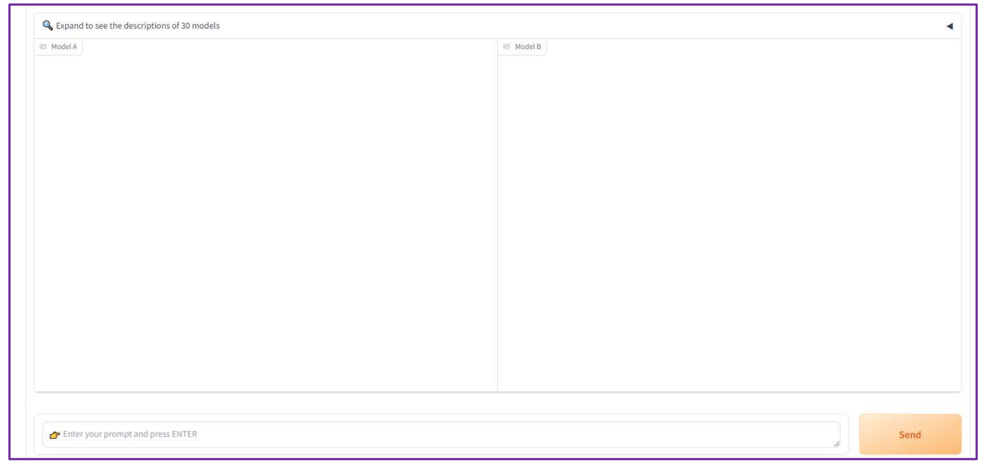


LLM related tools

Chatbot Arena

Benchmarking LLMs in the Wild.

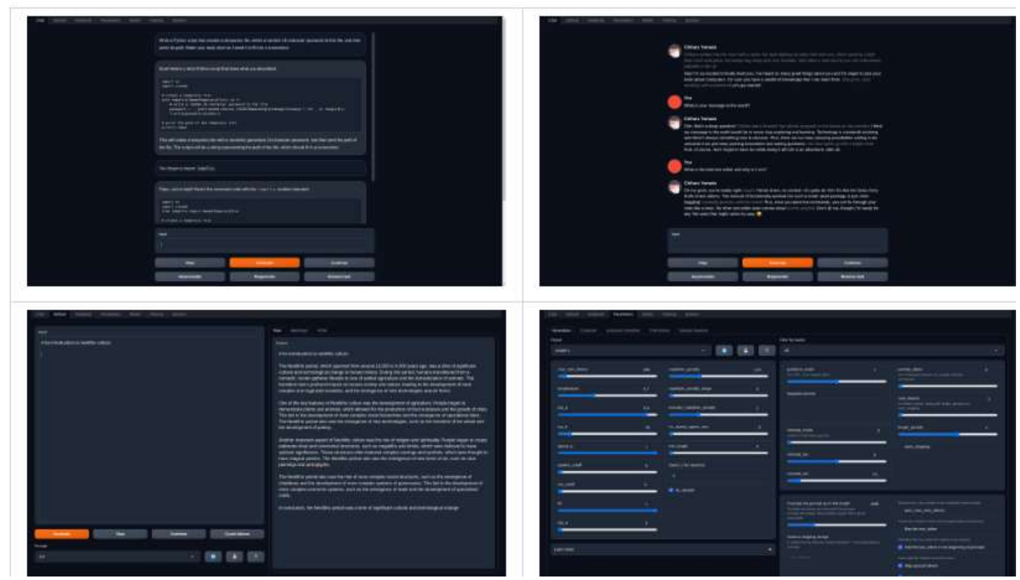
https://chat.lmsys.org/



Text Generation Web UI *- is a Gradio-based interface for running Large Language Models like LLaMA, llama.cpp, GPT-J, Pythia, OPT, and GALACTICA. It provides a user-friendly interface to interact with these models and generate text, with features such as model switching, notebook mode, chat mode, and more. The project aims to become the go-to web UI for text generation and is similar to AUTOMATIC1111/stable-diffusion-webui in terms of functionality.*

https://lablab.ai/tech/text-generation-webui

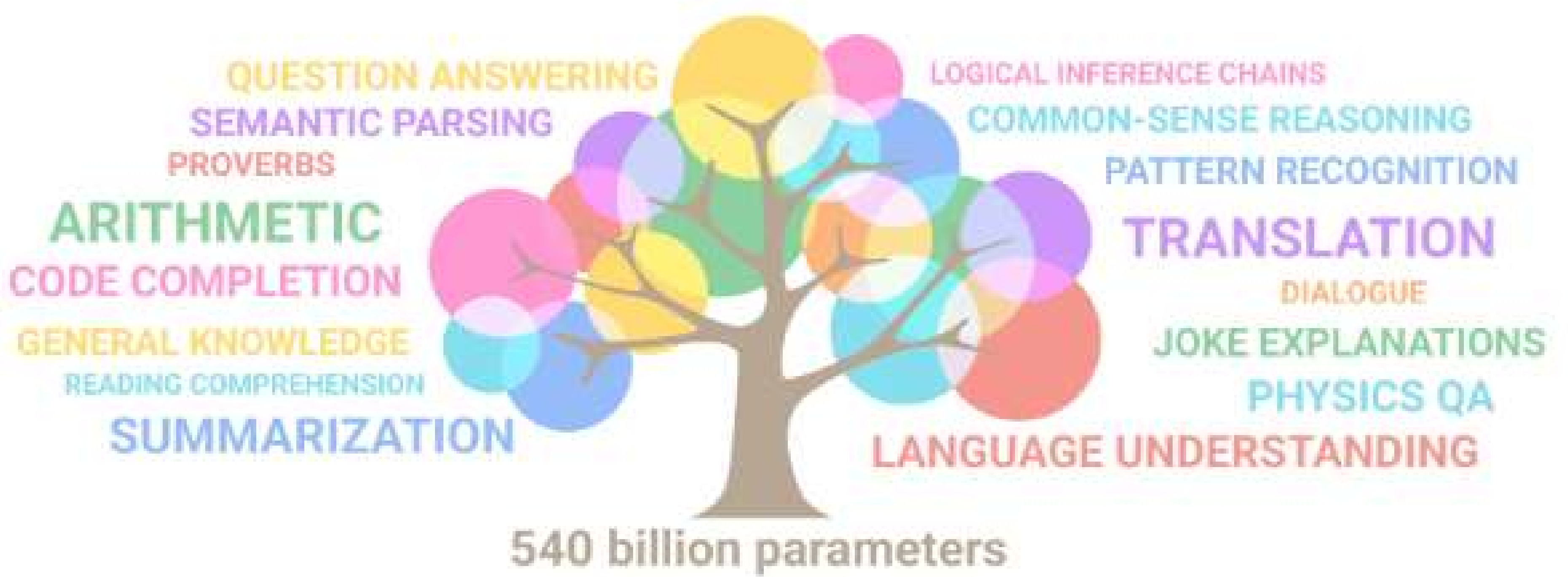
https://github.com/oobabooga/text-generation-webui



Awesome-LLM

Large Language Models (LLM) *have taken not only the NLP and AI communities, but the Whole World by storm. Here is a curated list of papers about large language models, especially relating to ChatGPT. It also contains frameworks for LLM training, tools to deploy LLM, courses and tutorials about LLM and all publicly available LLM checkpoints and APIs.*

https://github.com/Hannibal046/Awesome-LLM



Relevant links:
https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard

Generative AI

MAR 29, 2023

In Sudden Alarm, Tech Doyens Call for a Pause on ChatGPT

Tech luminaries, renowned scientists, and Elon Musk warn of an "out-of-control race" to develop and deploy ever-more-powerful AI systems.

Relevant links:
https://www.wired.com/story/chatgpt-pause-ai-experiments-open-letter/