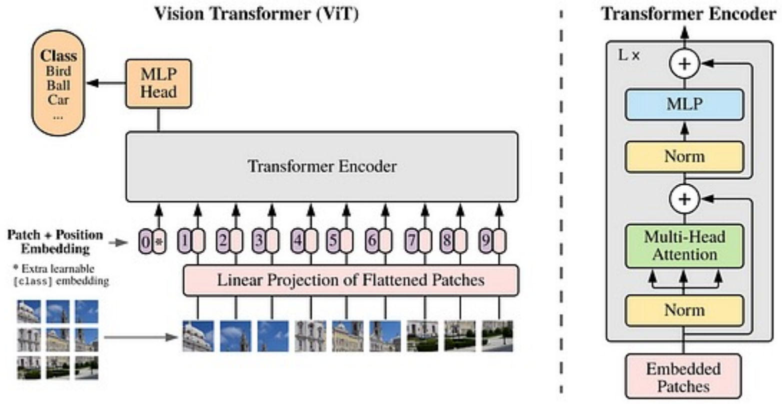
GAN and Diffusion with Transformer

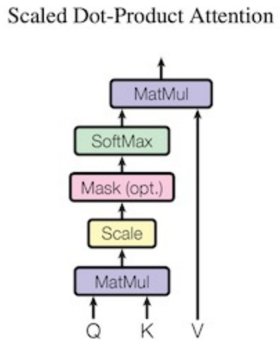
Stanley Liang

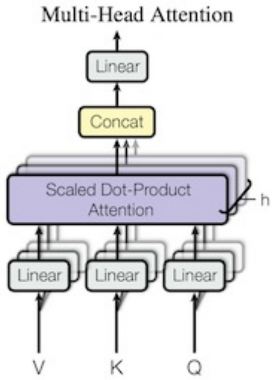
Research Fellow, NLM

Recap

- The transformer architecture can extend to computer vision tasks
- Unlike the convolutional approach
  - Feed the NN with a sequence of image patches
  - Convert flattened patches into class embedding + position
  - Multi-head Self Attention
    - Key - content details, relationships, crucial features in recognition
    - Query – patch content, similarity, significance in the whole image
    - Value – patch information to other patches, capture & express importance of features
    - $Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$

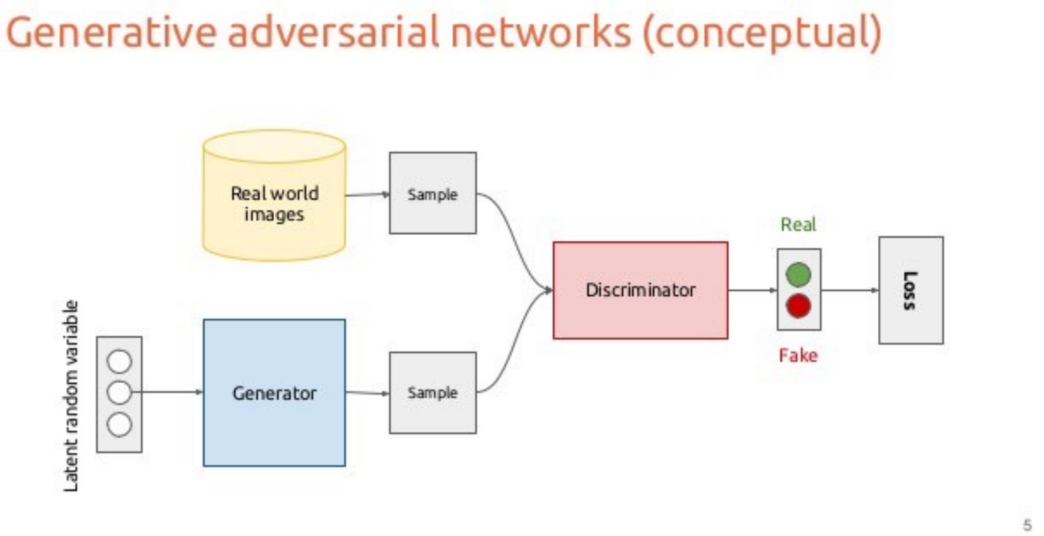


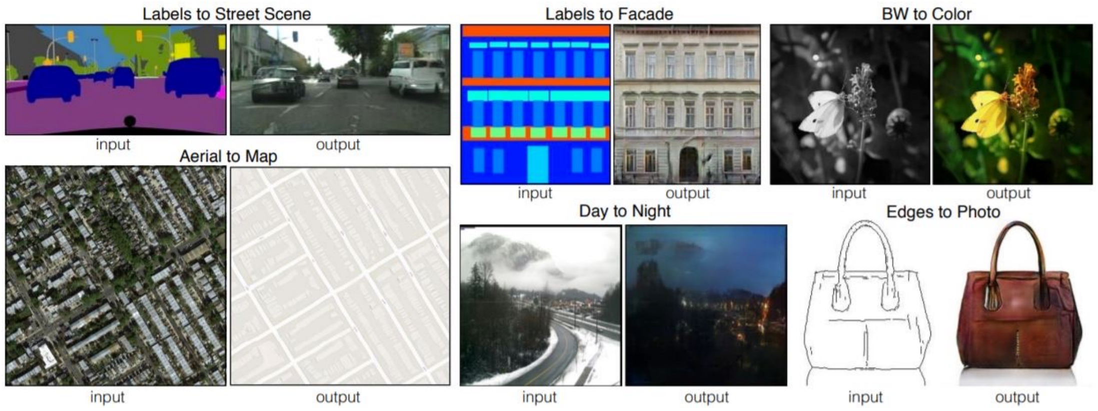




GAN – Generative Adversarial Network

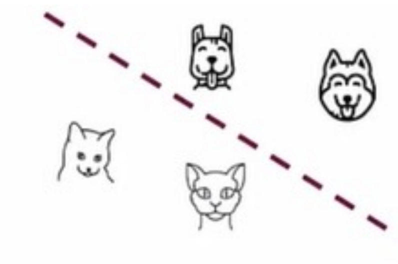
- GAN is a deep neural network framework to generate plausible data samples given a distribution domain
- GAN trains a generative model by framing the problem as a supervised learning problem with two sub-models
  - The generator model produces new samples
  - The discriminator model classifies whether a sample is truly from the domain (real), or a generated one (fake)
- The two models are trained together in an adversarial manner, or zero-sum game
  - Ideal status – the discriminator classifies about 50% as fake, meaning the generated images can fool the discriminator





Generative models

- Generative vs Discriminative
- Discriminative



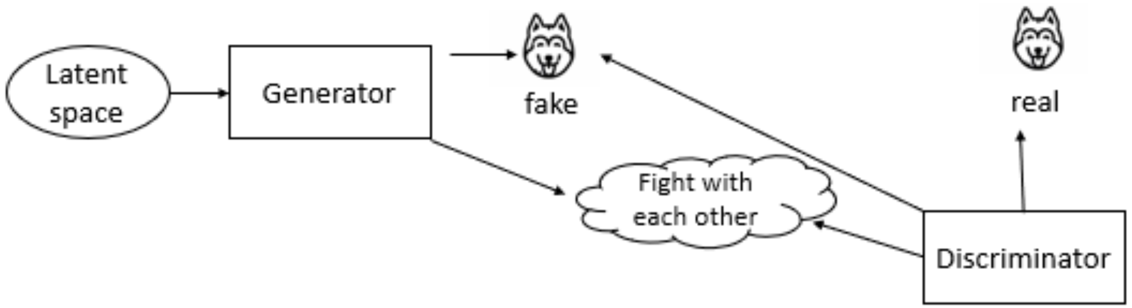
- $X \rightarrow Y$ by ML, Arg max P(Y|X)
- Generative

- $\xi_{noise}, Y_{class} \rightarrow X_{features}$
- $P(X|Y)$

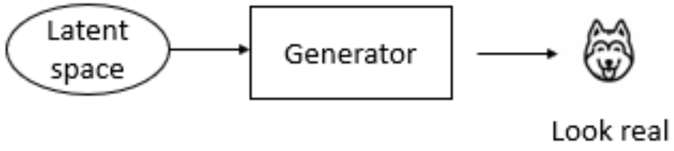
- Variational Autoencoders (VAE)

- After training

- Generative adversarial network (GAN)

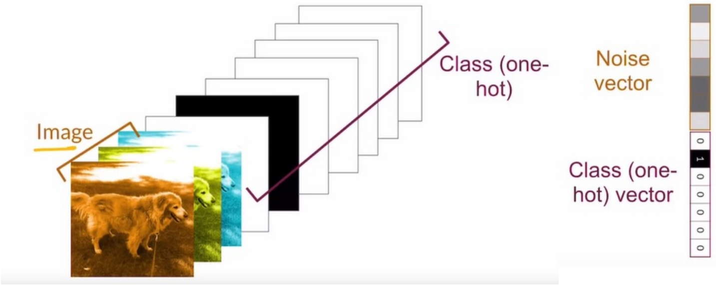


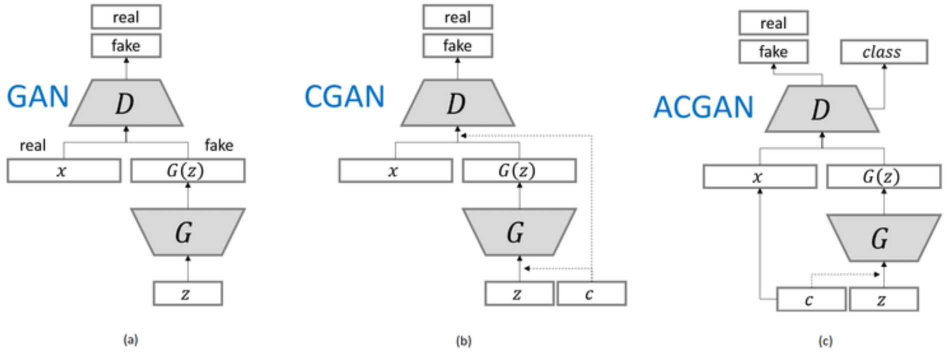
- After training



Conditional GAN

- Conditional GAN or cGAN generates sample to a designated class
- Unconditional GAN generates sample to a random class
- cGAN requires labeled training data
- Label encoding
  - Extra class vector
  - Extra dimensions to the input matrices
  - Use shallow NN to encode a feature map as the labels
- Issues of cGAN
  - Complexity of feature encoding
  - Difficult to optimize
  - Require large training datasets





Wasserstein GAN

- Mode collapse
  - Generator produces a particular plausible output which classified as real by the discriminator
  - Discriminator finds the best idea is to always reject this type of outputs
  - Generator over-optimizes for a particular discriminator
  - Discriminator never manages to learn its way out of the trap.

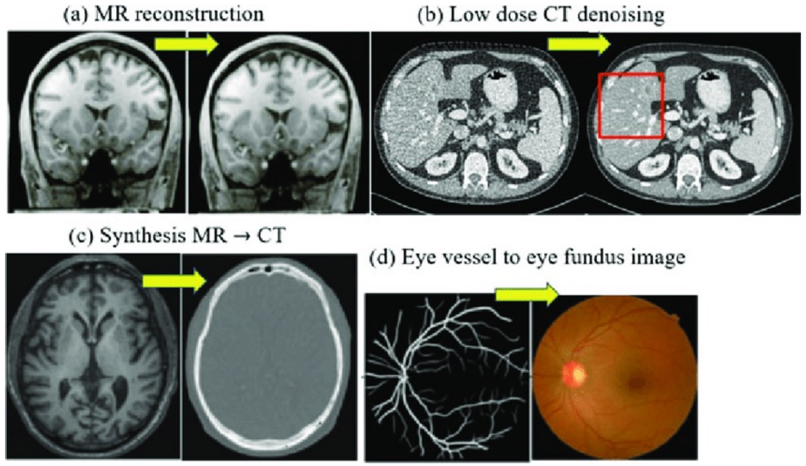
- Wasserstein use a new discriminator called critic to measure the dissimilarity of the two distributions by earth mover's distance (EMD)

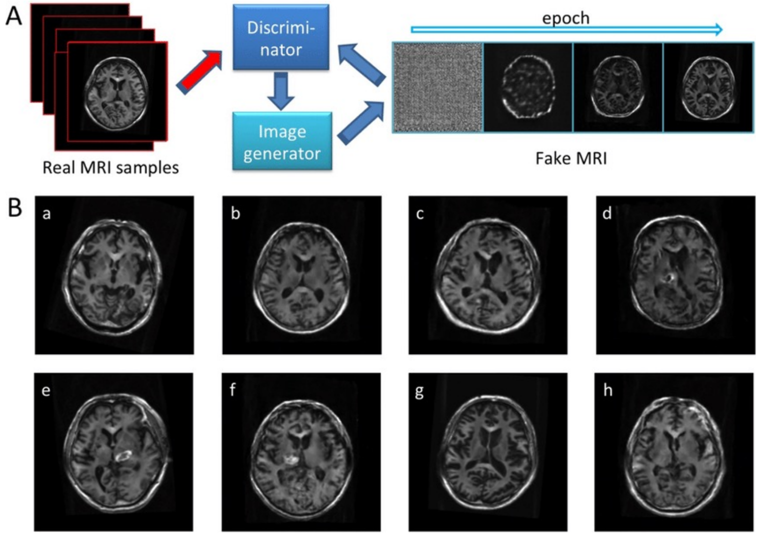
$$\mathbb{W}(\boldsymbol{P}_r, \boldsymbol{P}_g) = \sup_{\|f\|_L \leq 1} \mathbb{E}_{x\sim\mathbb{P}_r}(f(x)) - \mathbb{E}_{x\sim\mathbb{P}_g}(f(x))$$

- The Wasserstein loss no longer measures the probability of an image being real or fake, by the distance between the synthetic distribution and the real distribution
- The loss is differentiable in full range
- EMD computing
  - Lipschitz L1 norm
  - Weight clipping
  - Gradient penalty

Image Synthesis by GAN

- The generator produces plausible image example to a belonging to the designated domain
- Synthetic images usually look more real
- Evaluation metrics
  1. Subjective judgment
  2. Fidelity – synthetic images vs real images
  3. Diversity – generated images should not be identical
  4. Inception score, Frechet-Inception distance (FID), Kernel-Inception distance (KID), etc..





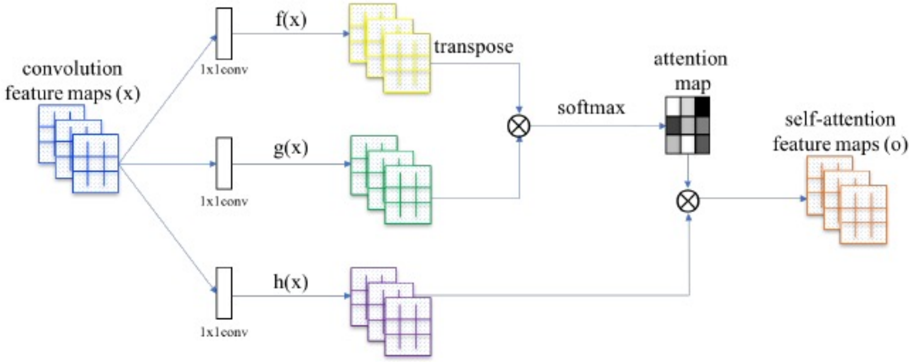
Self-Attention Generative Adversarial Networks (SAGAN)

- Proposed by GAN inventor in 2019
- Attention-driven, long-range dependency
- Discriminator can detect consistency of highly detailed features
- Spectral normalization for generator
- Objective loss: hinge loss

$$L_D = -\mathbb{E}_{(x,y)\sim p_{data}}[\min(0, -1 + D(x,y))] - \mathbb{E}_{z\sim p_z, y\sim p_{data}}[\min(0, -1 - D(G(z), y))],$$
$$L_G = -\mathbb{E}_{z\sim p_z, y\sim p_{data}} D(G(z), y),$$

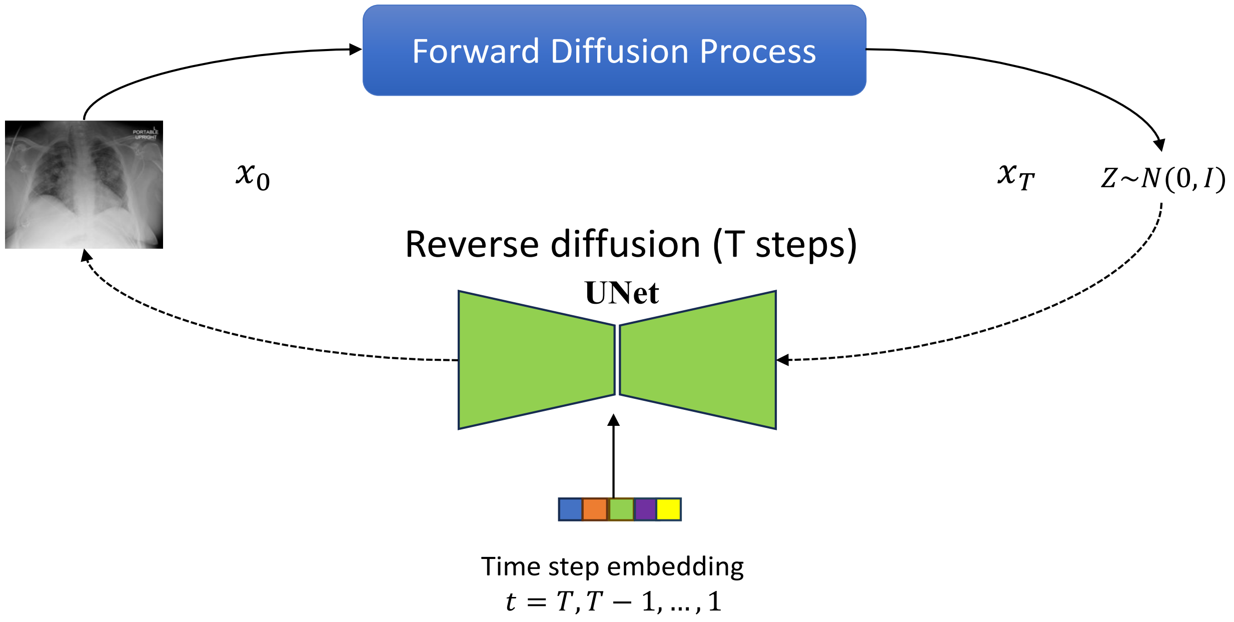
- Spectral Normalization (SN): prevent parameter magnitudes from escalating
- Two-Timescale Update Rule (TTUR)
- Compensate slow learning by regularization

Model	Inception Score	FID
AC-GAN [31]	28.5	/
SNGAN-projection [17]	36.8	27.62*
SAGAN	**52.52**	**18.65**

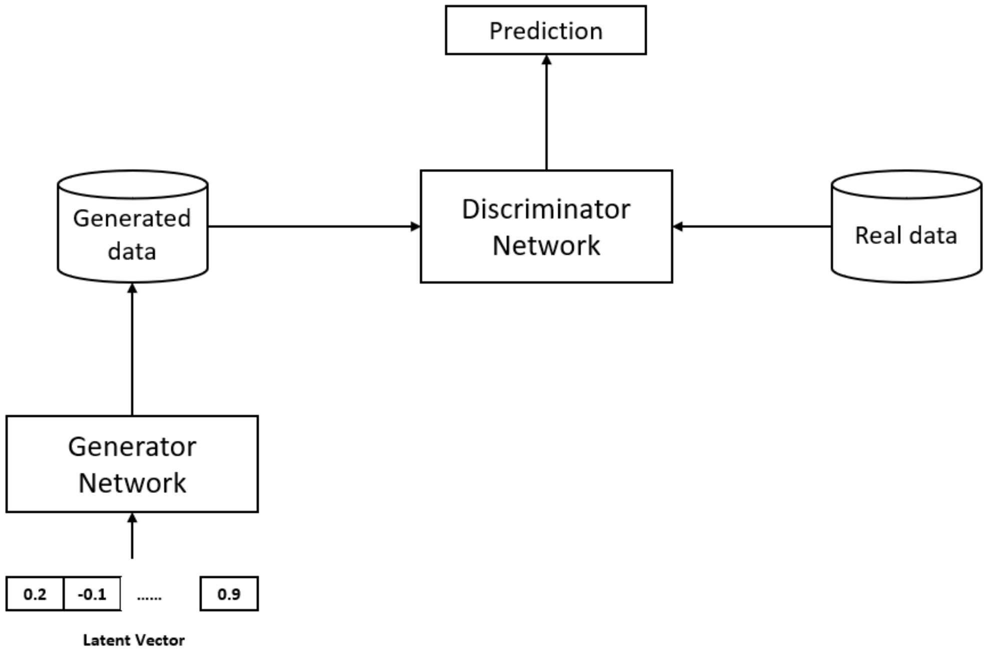


Denoising diffusion versus GAN

$$q(x_t|x_0) = \sqrt{\bar{a}_t}x_0 + \sqrt{1-\bar{a}_t}\epsilon,\ \epsilon \sim N(0,1), \text{let } \bar{\alpha} = \prod_0^T a_t$$



$$p_\theta(x_{t-1}|x_t) \coloneqq N(x_{t-1}; \mu_\theta; (x_t, t)\ \epsilon_\theta(x_t, t))$$



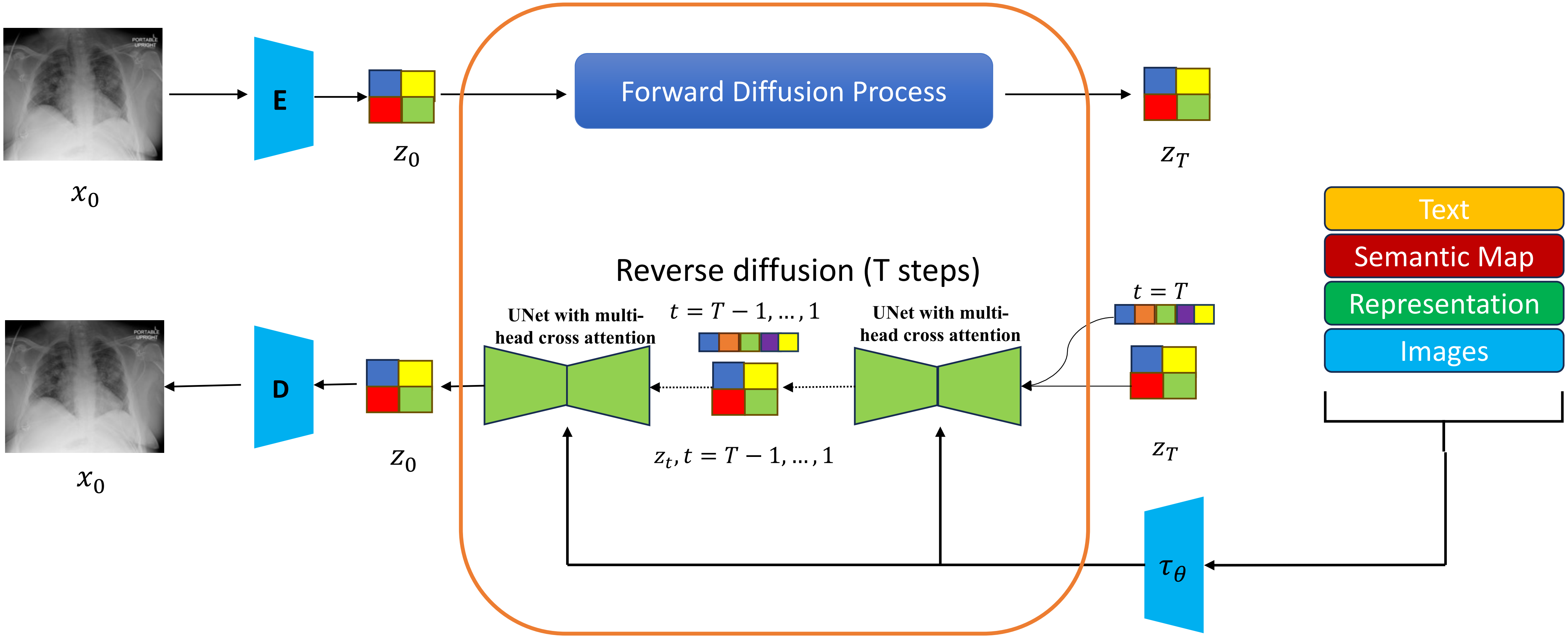
Loss Objective

$$L_{KL} = \mathbb{E}_{t,x_0,\epsilon}\left[\left\|\epsilon - \epsilon_\theta\left(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon\right)\right\|^2\right]$$

$$\textit{Generator loss}: \nabla_{\theta_g} \frac{1}{m}\sum_{i=1}^{m} \log(1 - D(G(z^{(i)})))$$

$$\textit{Discriminator loss}: \nabla_{\theta_g} \frac{1}{m}\sum_{i=1}^{m}[\log D(x^{(i)}) + \log\left(1 - D\left(G(z^{(i)})\right)\right)]$$

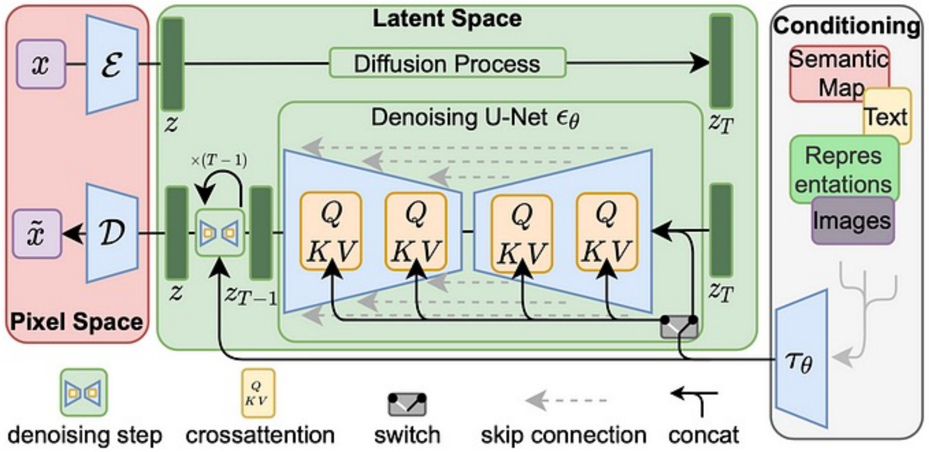
Latent Diffusion Model (LDM)



$$L_{LDM} = \mathbb{E}_{t,z_0,\epsilon,y}\left[\left\|\epsilon - \epsilon_\theta\left(z_{t,} t, \tau_\theta(y)\right)\right\|^2\right]$$

U-Net with attention

- Conventional U-net uses convolution blocks in both up/down sampling
- U-net with self-attention enhances the learning for both local and global context
- Patching + position embedding
- UP / down blocks, bottlenet
  - Convolution + residual
  - Residual connection
  - normalization layer
  - self-attention layer



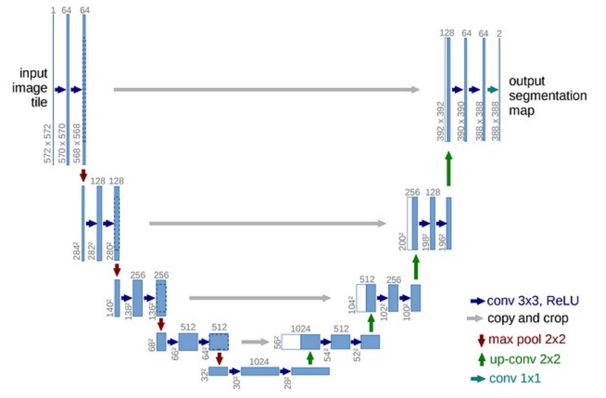


Fig. 1. U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations.

Explainable AI (XAI) to explore transformers

- XAI shows how the ML algorithms make decisions
- Grad-CAM (Gradient-weighted Class Activation Mapping)
- Mean of attention distance
- Attention heatmap