Deep Learning for Vision & Language

Text-to-Scene Models

RICE UNIVERSITY

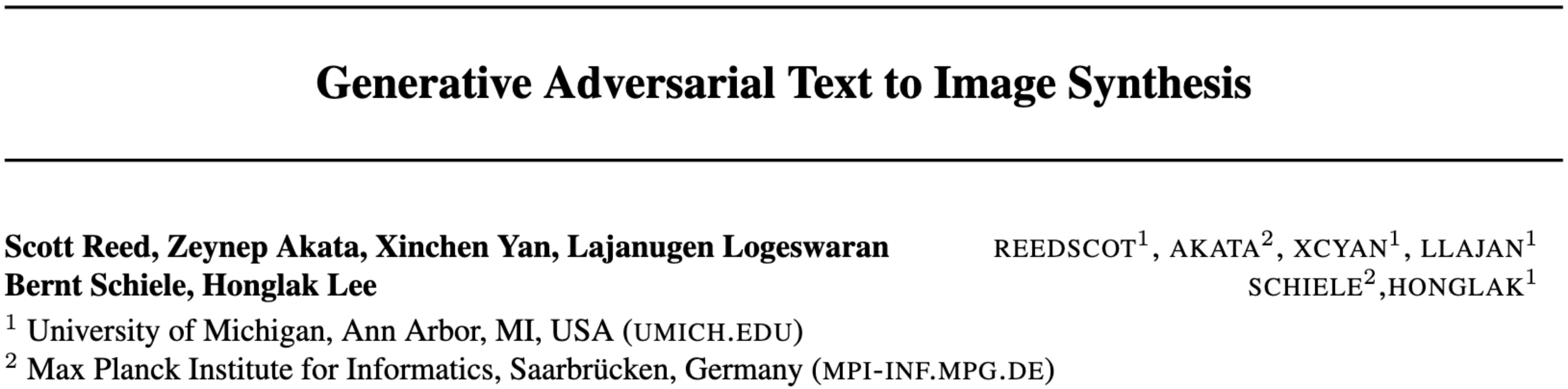
Last Class

- Conditional GANs
- AutoEncoder Models (AEs, VAEs)

Today:

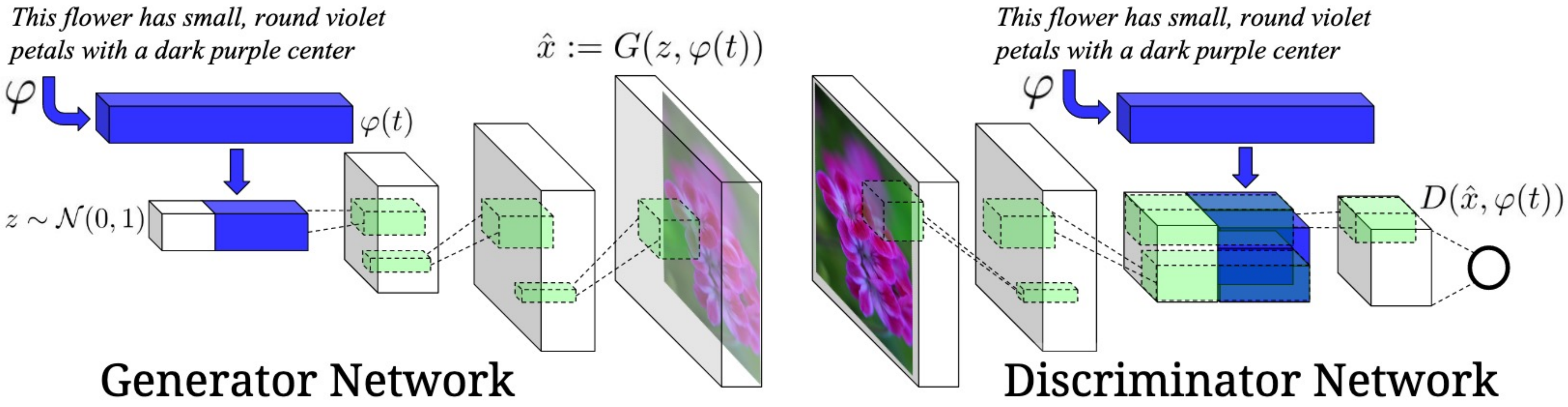
- Text to image Models
- Sequence-to-sequence based text-to-image models
- Detour: Style Transfer – Input Feature Optimization.
- Reverse Diffusion Models

Conditional GANs / Text-conditioned


Generative Adversarial Text to Image Synthesis

Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran REEDSCOT[1], AKATA[2], XCYAN[1], LLAJAN[1]
Bernt Schiele, Honglak Lee SCHIELE[2],HONGLAK[1]
[1] University of Michigan, Ann Arbor, MI, USA (UMICH.EDU)
[2] Max Planck Institute for Informatics, Saarbrücken, Germany (MPI-INF.MPG.DE)

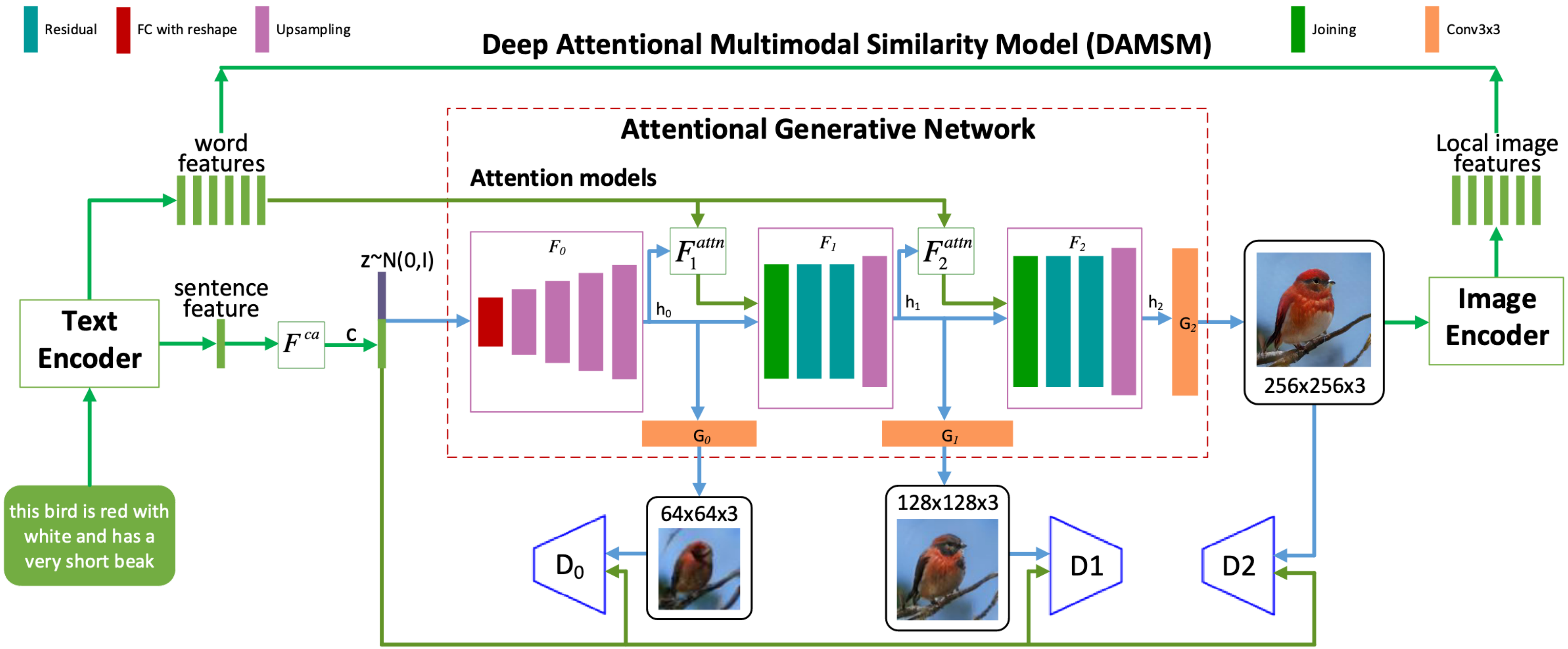
Conditional GANs / Text-conditioned



Conditional GANs / Text-conditioned

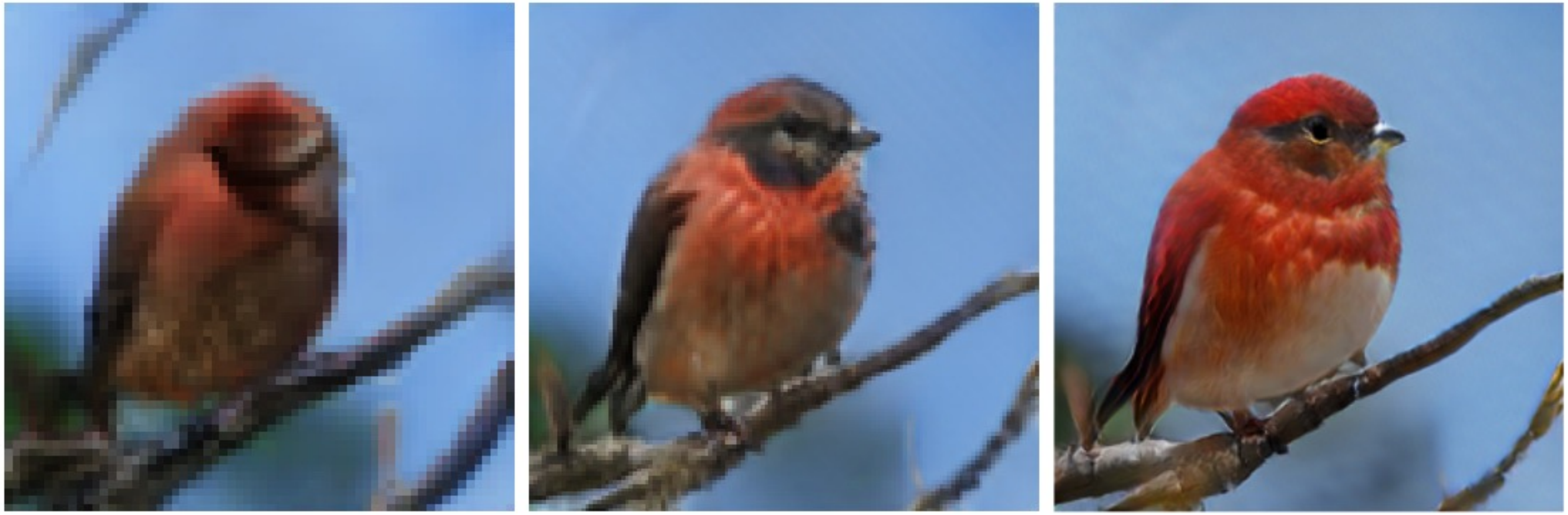
this small bird has a pink breast and crown, and black primaries and secondaries.

Conditional GANs / Text-conditioned

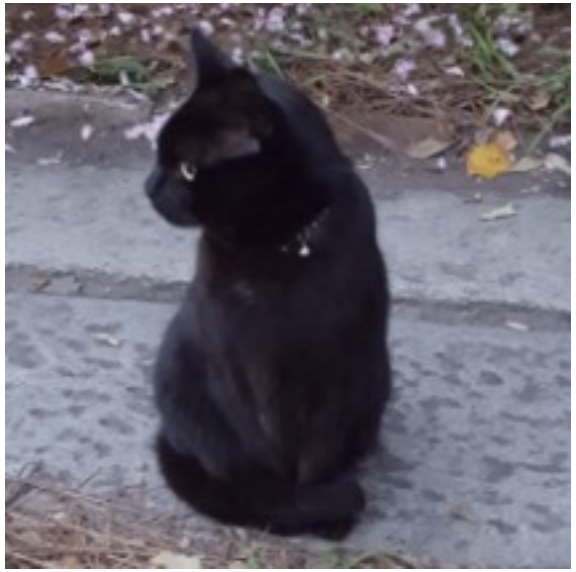


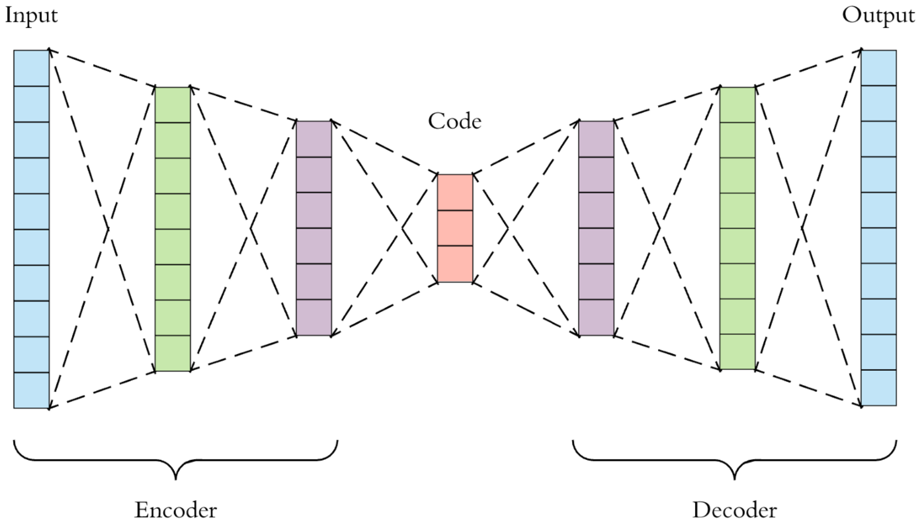
Conditional GANs / Text-conditioned

this bird is red with white and has a very short beak

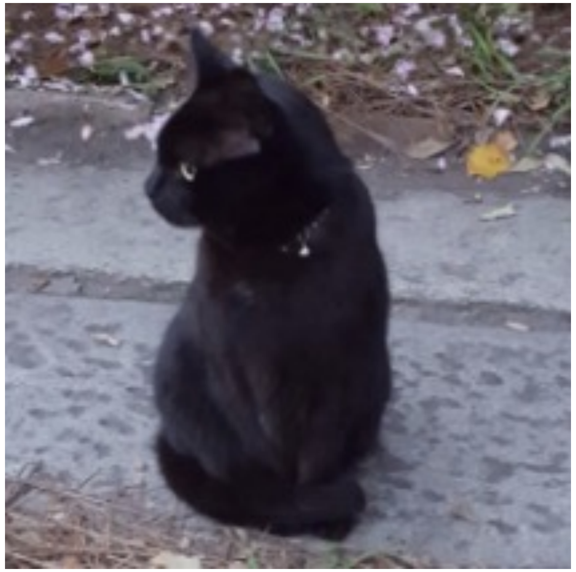


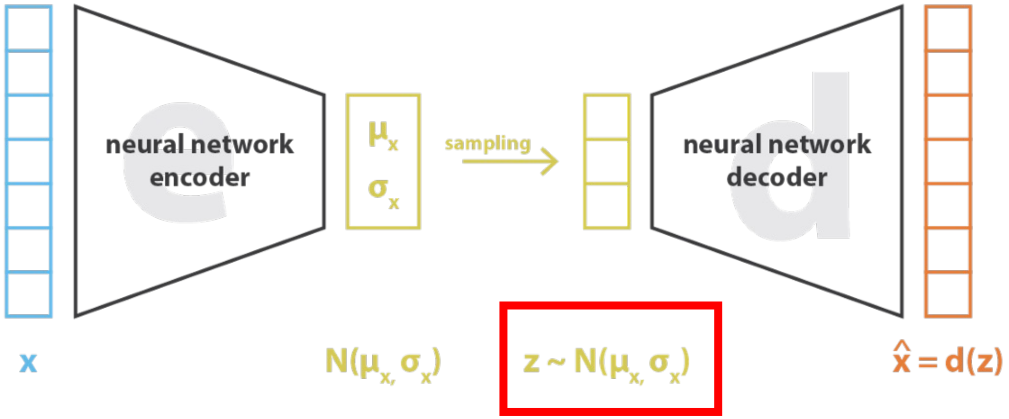
AutoEncoder Models





Variational AutoEncoder (VAE)

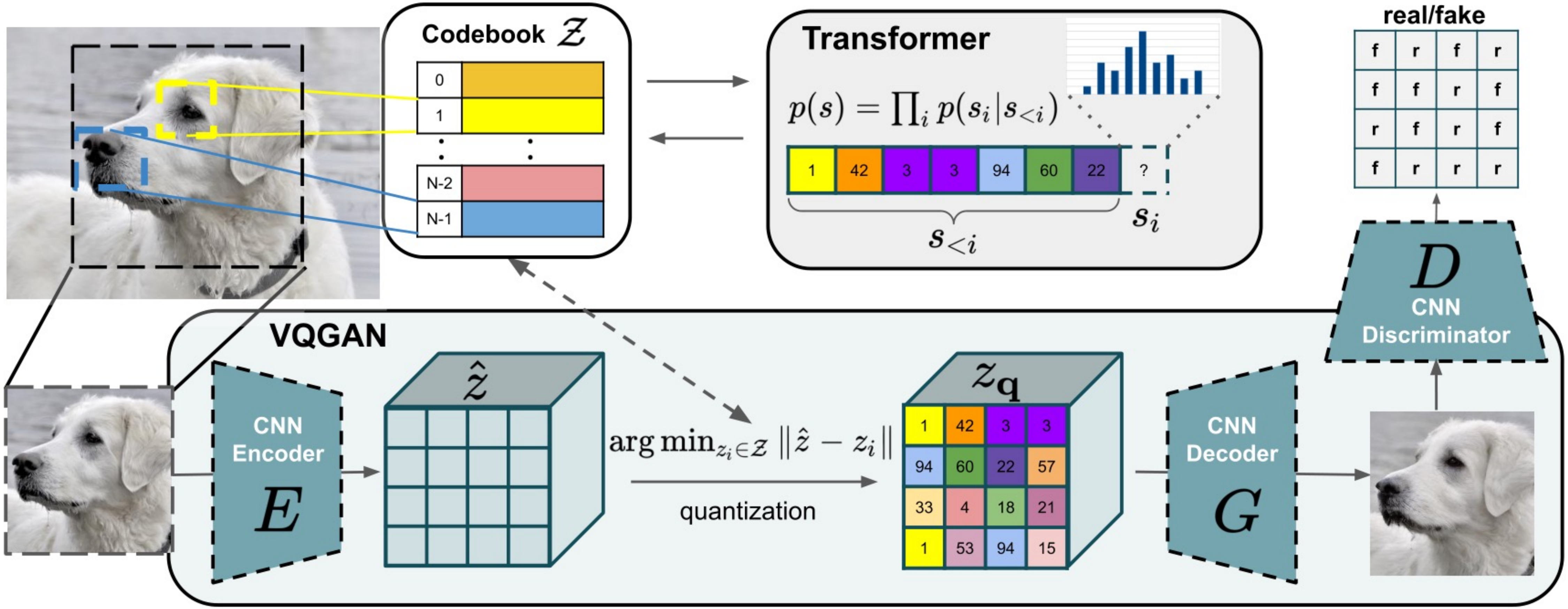




$$\text{loss} = \| x - \hat{x} \|^2 + \text{KL}[\, N(\mu_x, \sigma_x), N(0, I) \,]$$

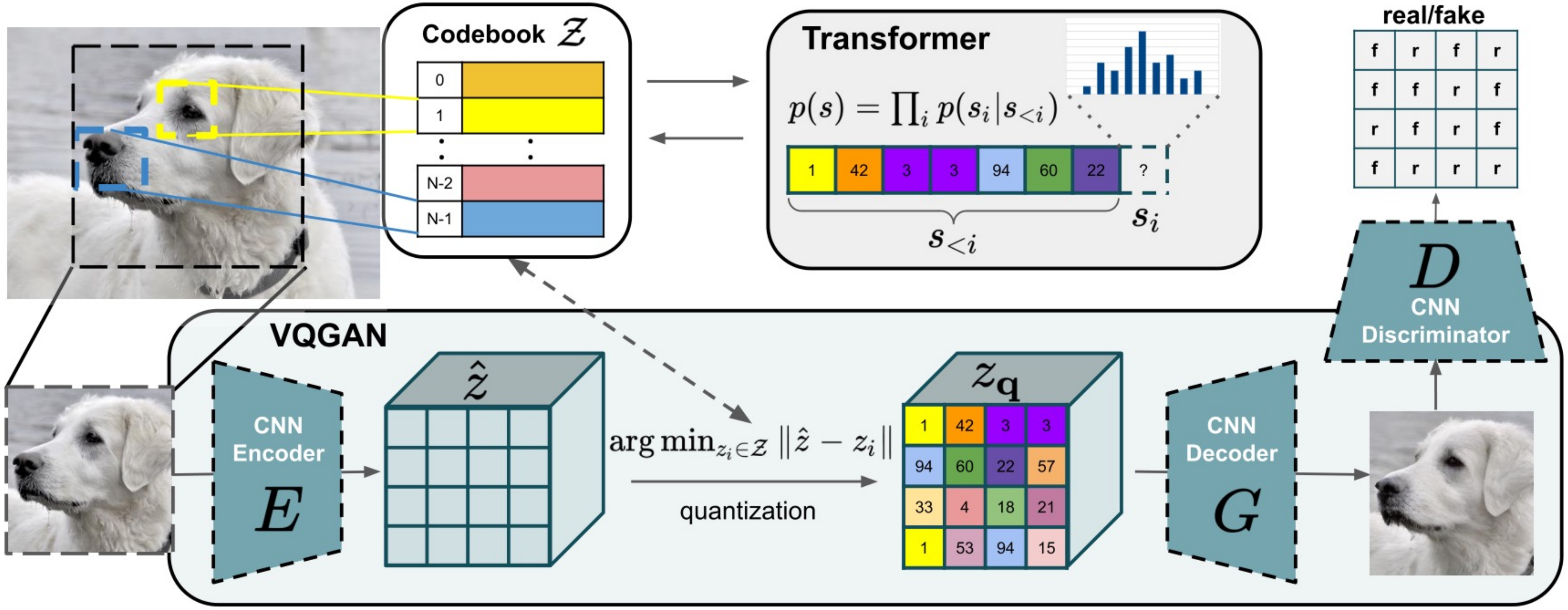
https://pytorch.org/docs/stable/generated/torch.normal.html
https://ai.stackexchange.com/questions/30176/are-mean-and-standard-deviation-in-variational-autoencoders-unique

Vector Quantized - GAN



https://arxiv.org/abs/2012.09841 https://arxiv.org/pdf/2110.04627.pdf

Vector Quantized GAN (VQGAN)



$$\mathcal{Q}^* = \arg\min_{E,G,\mathcal{Z}} \max_D \mathbb{E}_{x\sim p(x)} \Big[\mathcal{L}_{\mathrm{VQ}}(E,G,\mathcal{Z}) + \lambda \mathcal{L}_{\mathrm{GAN}}(\{E,G,\mathcal{Z}\},D) \Big]$$

$$\mathcal{L}_{\mathrm{VQ}}(E,G,\mathcal{Z}) = \|x - \hat{x}\|^2 + \|\mathrm{sg}[E(x)] - z_{\mathbf{q}}\|_2^2 + \|\mathrm{sg}[z_{\mathbf{q}}] - E(x)\|_2^2.$$

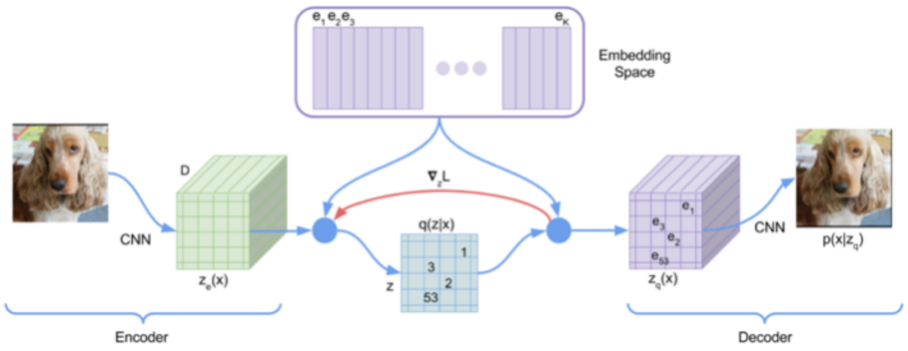
$$\mathcal{L}_{\mathrm{GAN}}(\{E,G,\mathcal{Z}\},D) = [\log D(x) + \log(1 - D(\hat{x}))]$$

https://arxiv.org/abs/2012.09841 https://arxiv.org/pdf/2110.04627.pdf

DALL-E (v1)

Step 1:

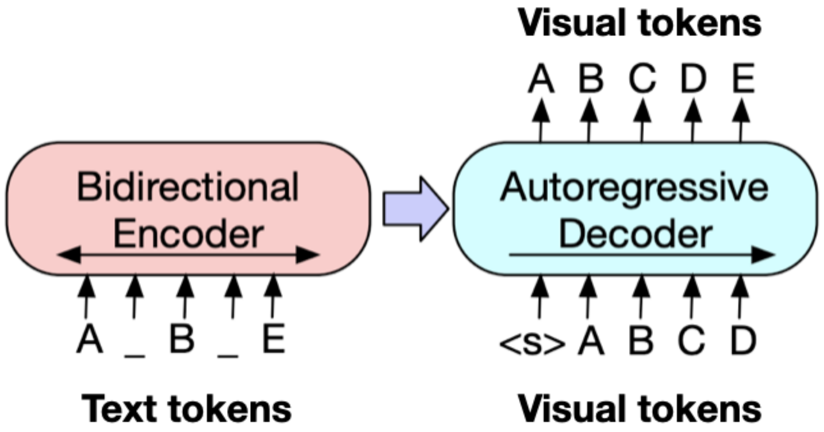
Learn Discrete Dictionary of Visual Tokens



VQVAE — Oord, Vinyals, Kavukcuoglu, 2017
VQGAN — Esser, Rombach, Ommer, 2021
dVAE - DALL-E — Ramesh et al 2021

Step 2:

Build a scene as a composition of discrete visual tokens

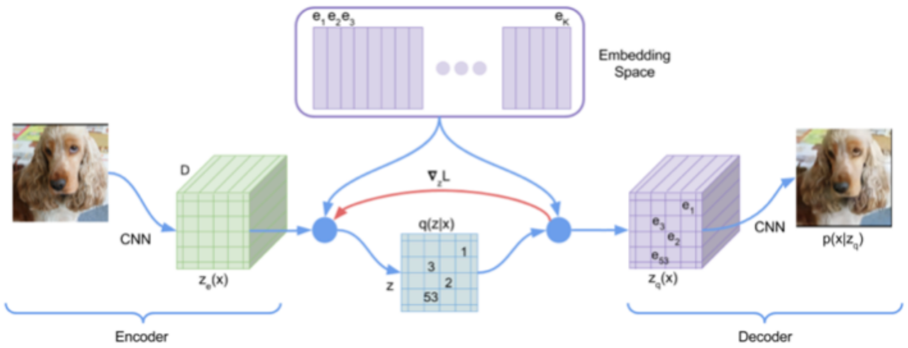


BART, GPT-3, etc

DALL-E (v1)

Step 1:

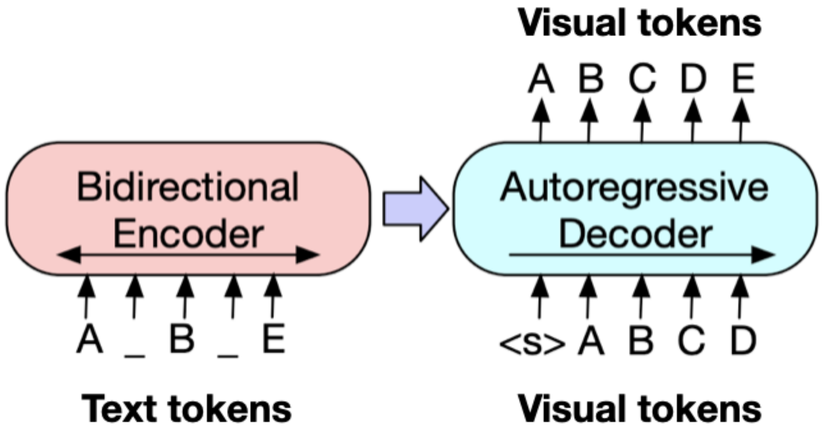
Learn Discrete Dictionary of Visual Tokens



VQVAE — Oord, Vinyals, Kavukcuoglu, 2017
VQGAN — Esser, Rombach, Ommer, 2021
dVAE - DALL-E — Ramesh et al 2021

Step 2:

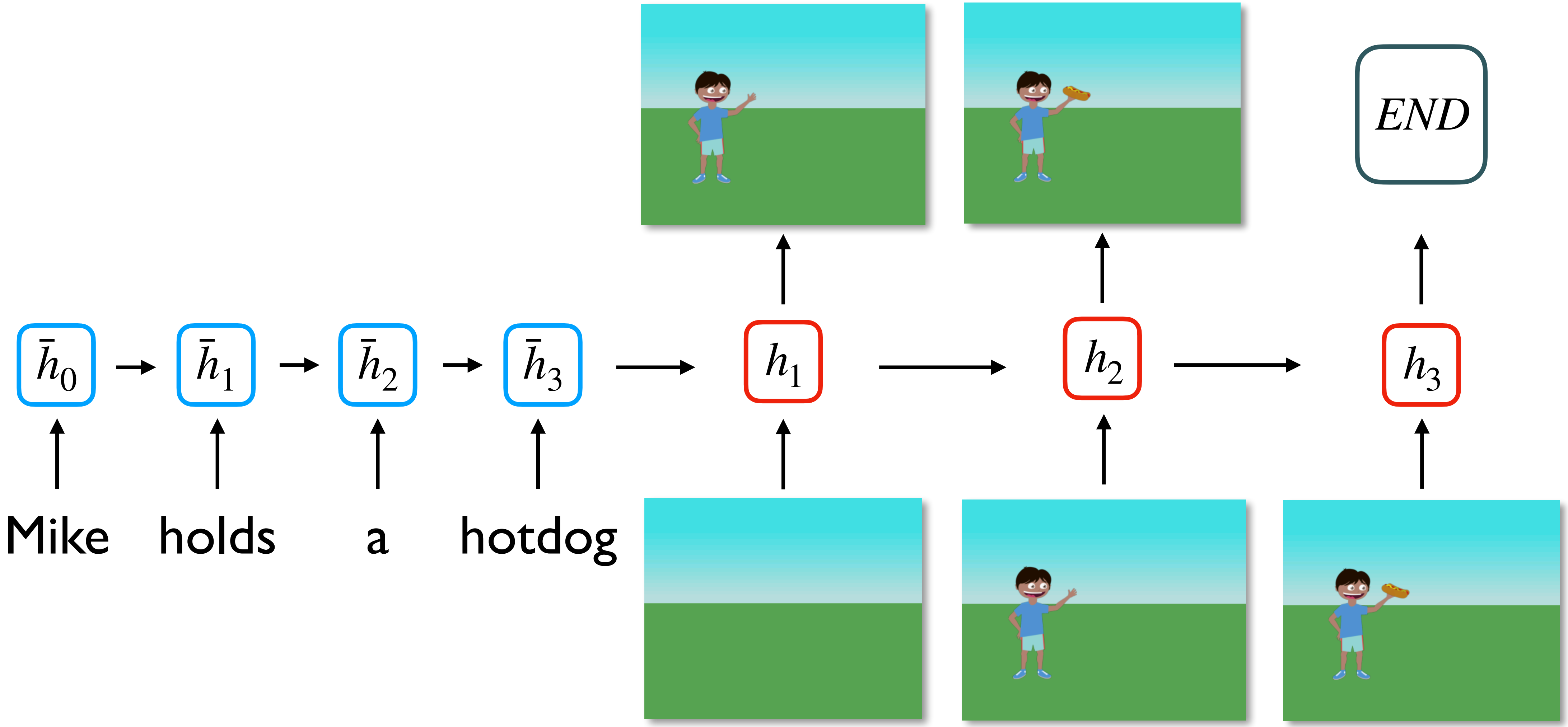
Build a scene as a composition of discrete visual tokens



BART, GPT-3, etc

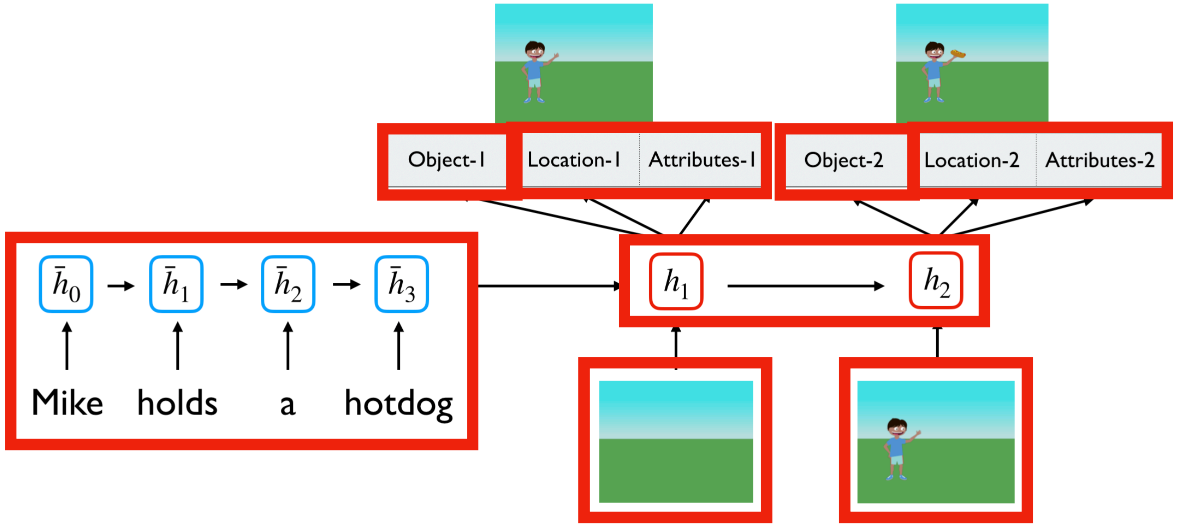
an armchair in the shape of an avocado. . . .

Text to Scene as Machine Translation!

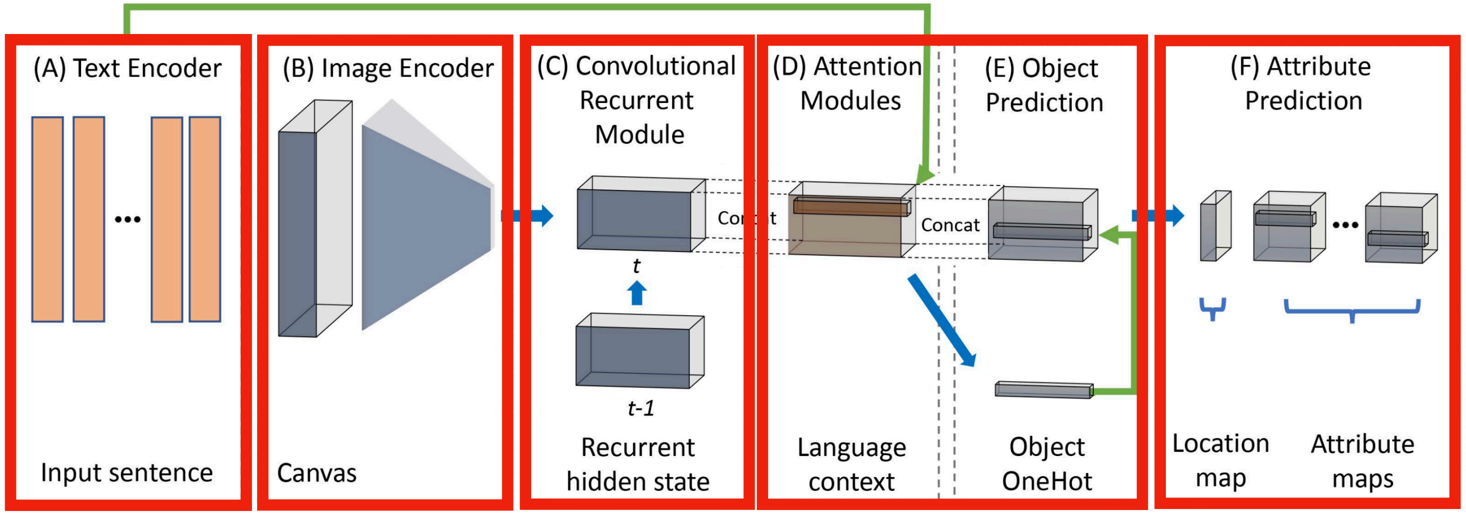


Text2Scene: Generating Compositional Scenes from Textual Descriptions
Fuwen Tan, Song Feng, Vicente Ordonez. Intl. Conference on Computer Vision and Pattern Recognition. **CVPR 2019.**
Long Beach, California. June 2019.(~Oral presentation + Best Paper Finalist -- top 1% of submissions)

The actual model



$h_i^E = \mathrm{BiGRU}(x_i, h_{i-1}^E, h_{i+1}^E$ $\Omega(B$ $h_t^D = \mathrm{ConvGRU}(\Omega($ $p(o_t) \propto \Theta^o([u_t^o; o_t$ $p(l_t, \{R_t^k\}) = \Theta^a([u_t^a; o_t; c_t^a])$



Objective

objects

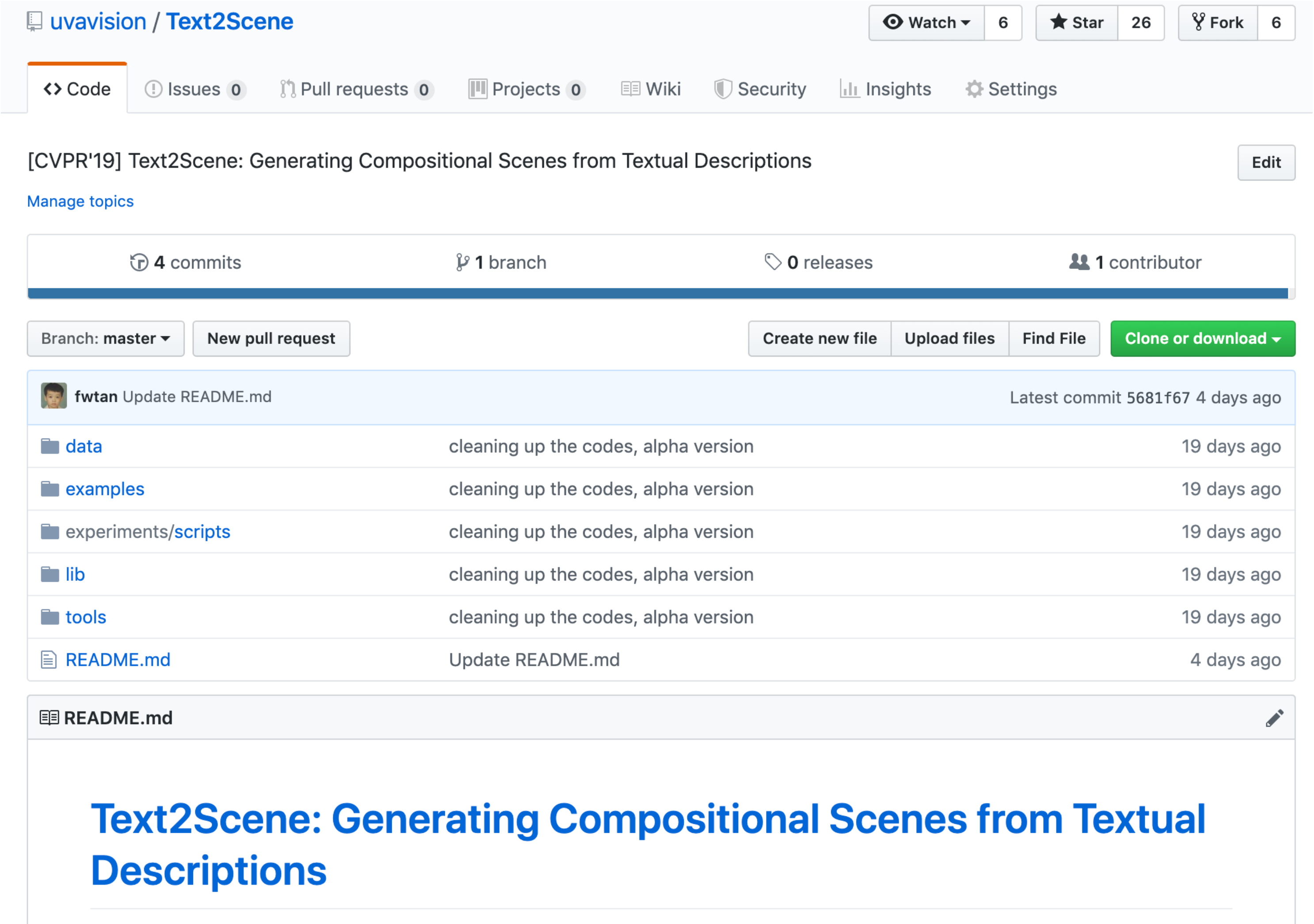
locations

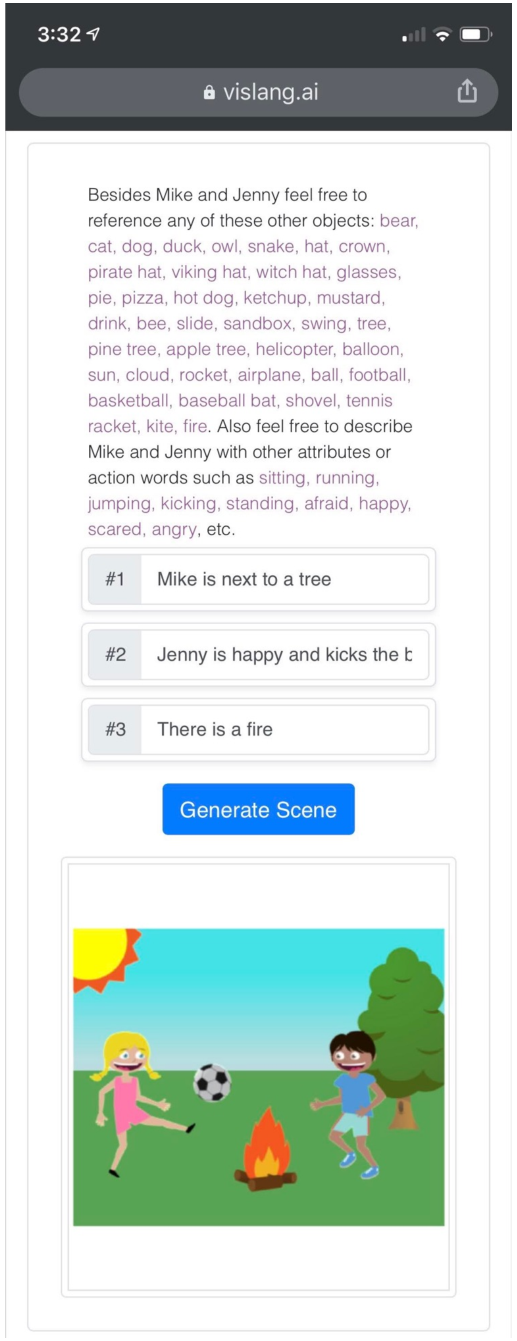
$$L = -w_o \sum_t \log p(o_t) - w_l \sum_t \log p(l_t) - \sum_k w_k \sum_t \log p(R_t^k) + w_a^O L_{attn}^O + w_a^A L_{attn}^A,$$

attributes

Encourage attention weights to fully use the input text.

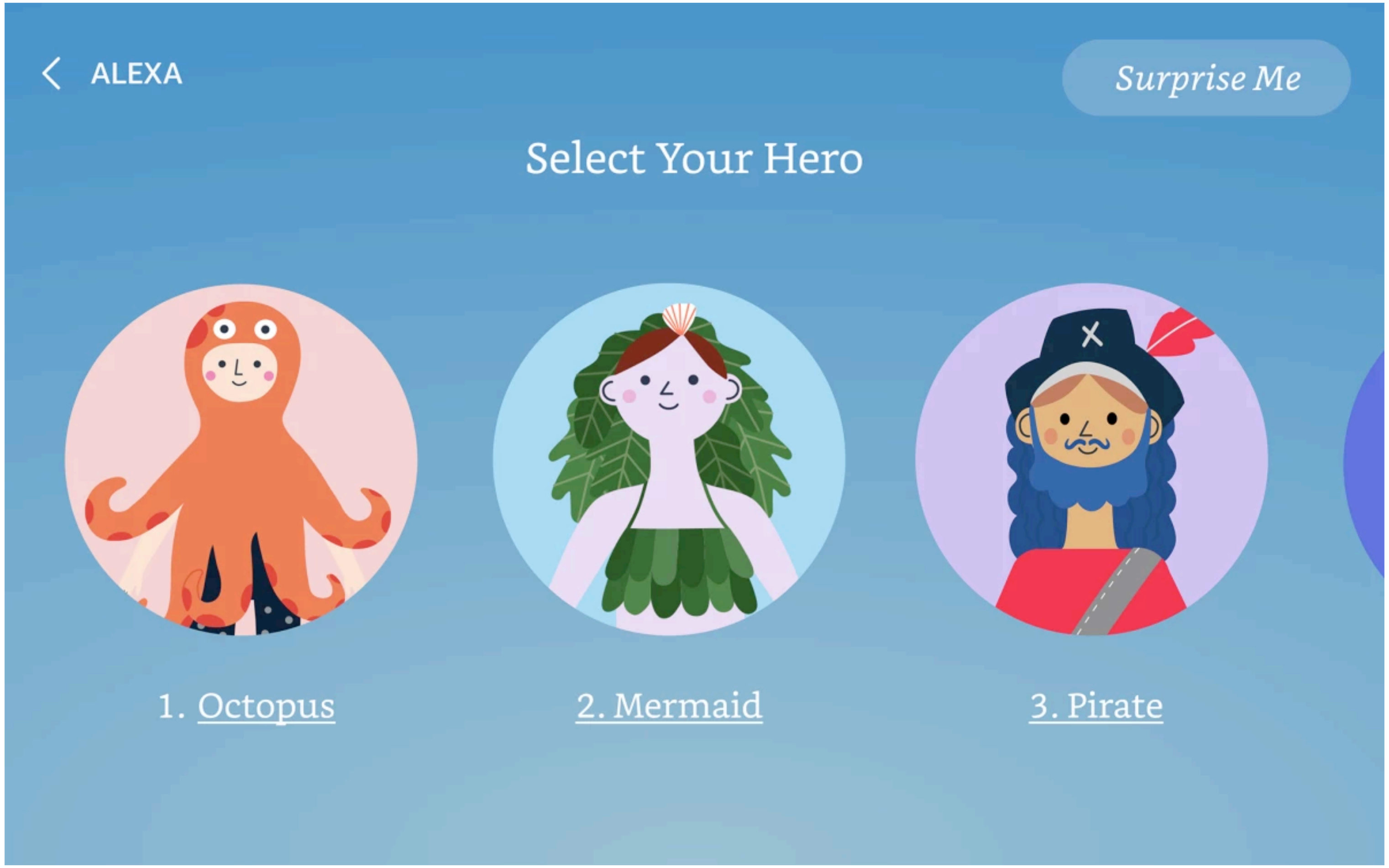
$$L_{attn} = \sum_i [1 - \sum_t \alpha_{t,i}]^2$$





https://www.vislang.ai/text2scene

Amazon Alexa AI



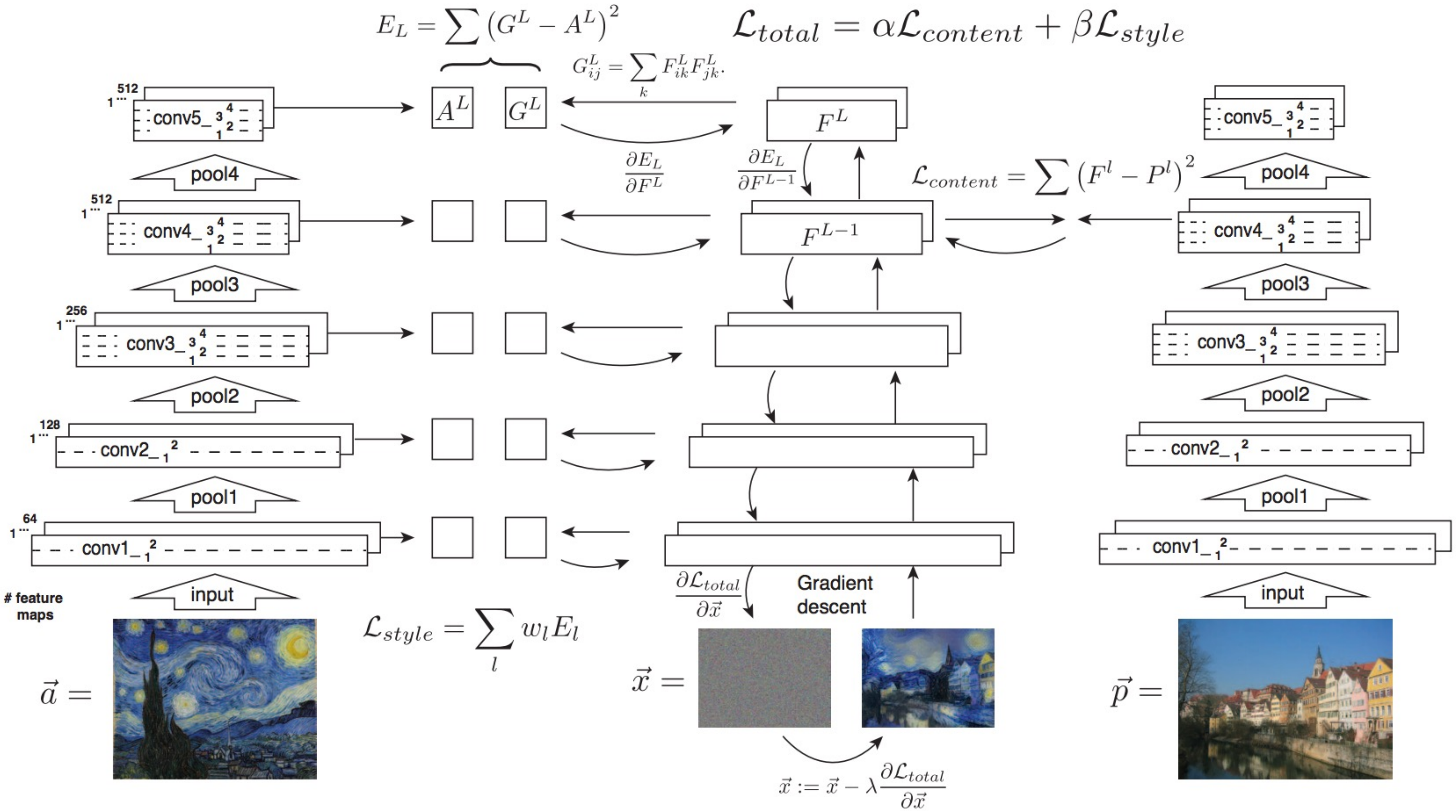
https://www.amazon.science/blog/the-science-behind-alexas-new-interactive-story-creation-experience

Amazon Alexa AI

https://www.amazon.science/blog/the-science-behind-alexas-new-interactive-story-creation-experience

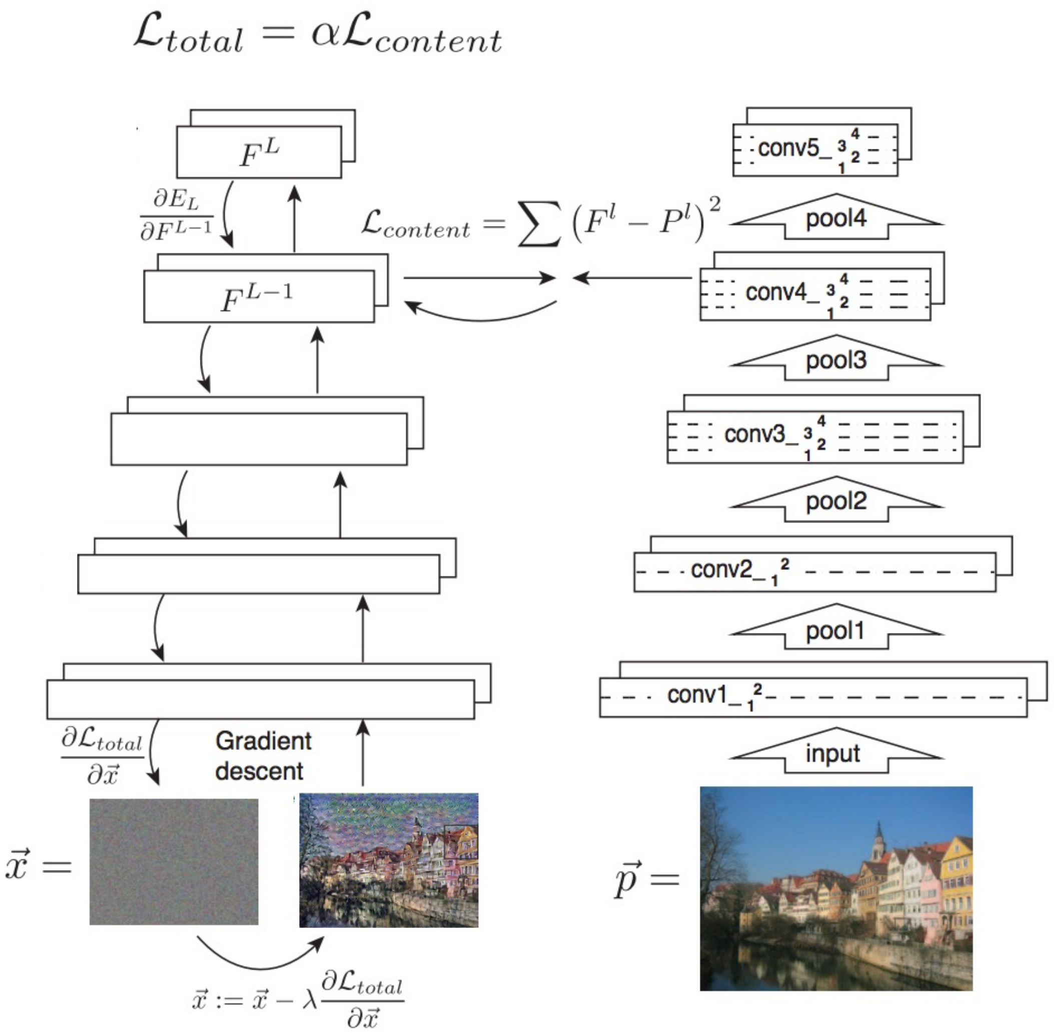
More on the Idea of Feature Space Optimization

Gatys et. al. Image Style Transfer Using Convolutional Neural Networks. CVPR 2016


$E_L = \sum (G^L - A^L)^2$
$\mathcal{L}_{total} = \alpha \mathcal{L}_{content} + \beta \mathcal{L}_{style}$
$G^L_{ij} = \sum_k F^L_{ik} F^L_{jk}.$
A^L
G^L
F^L
$\frac{\partial E_L}{\partial F^L}$
$\frac{\partial E_L}{\partial F^{L-1}}$
$\mathcal{L}_{content} = \sum (F^l - P^l)^2$
F^{L-1}
conv5_1 2 3 4
pool4
conv4_1 2 3 4
pool3
conv3_1 2 3 4
pool2
conv2_1 2
pool1
conv1_1 2
input
512
512
256
128
64
feature maps
$\frac{\partial \mathcal{L}_{total}}{\partial \vec{x}}$
Gradient descent
$\mathcal{L}_{style} = \sum_l w_l E_l$
$\vec{a} =$
$\vec{x} =$
$\vec{p} =$
$\vec{x} := \vec{x} - \lambda \frac{\partial \mathcal{L}_{total}}{\partial \vec{x}}$

Gatys et. al. Image Style Transfer Using Convolutional Neural Networks. CVPR 2016

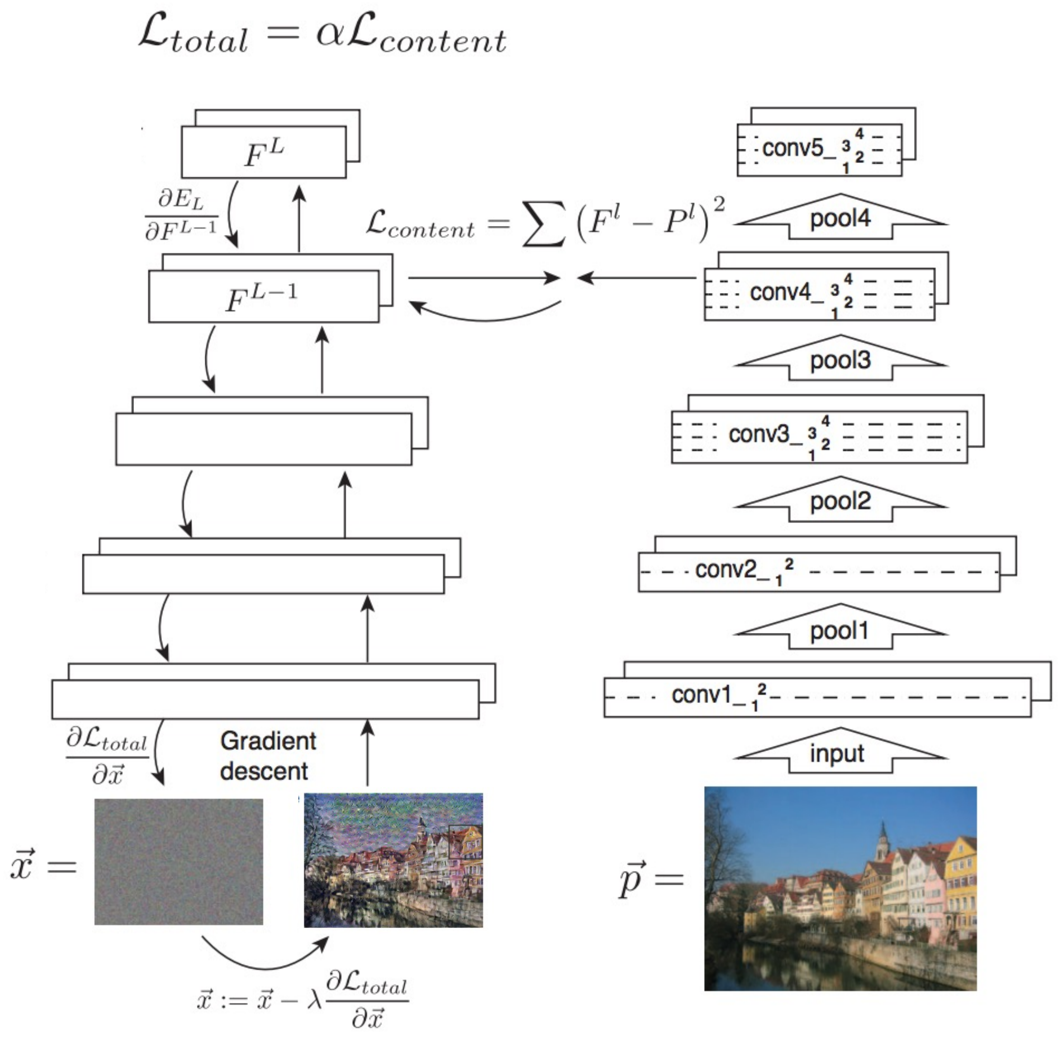
Idea 1: Image Reconstruction from Features



Gatys et. al. Image Style Transfer Using Convolutional Neural Networks. CVPR 2016

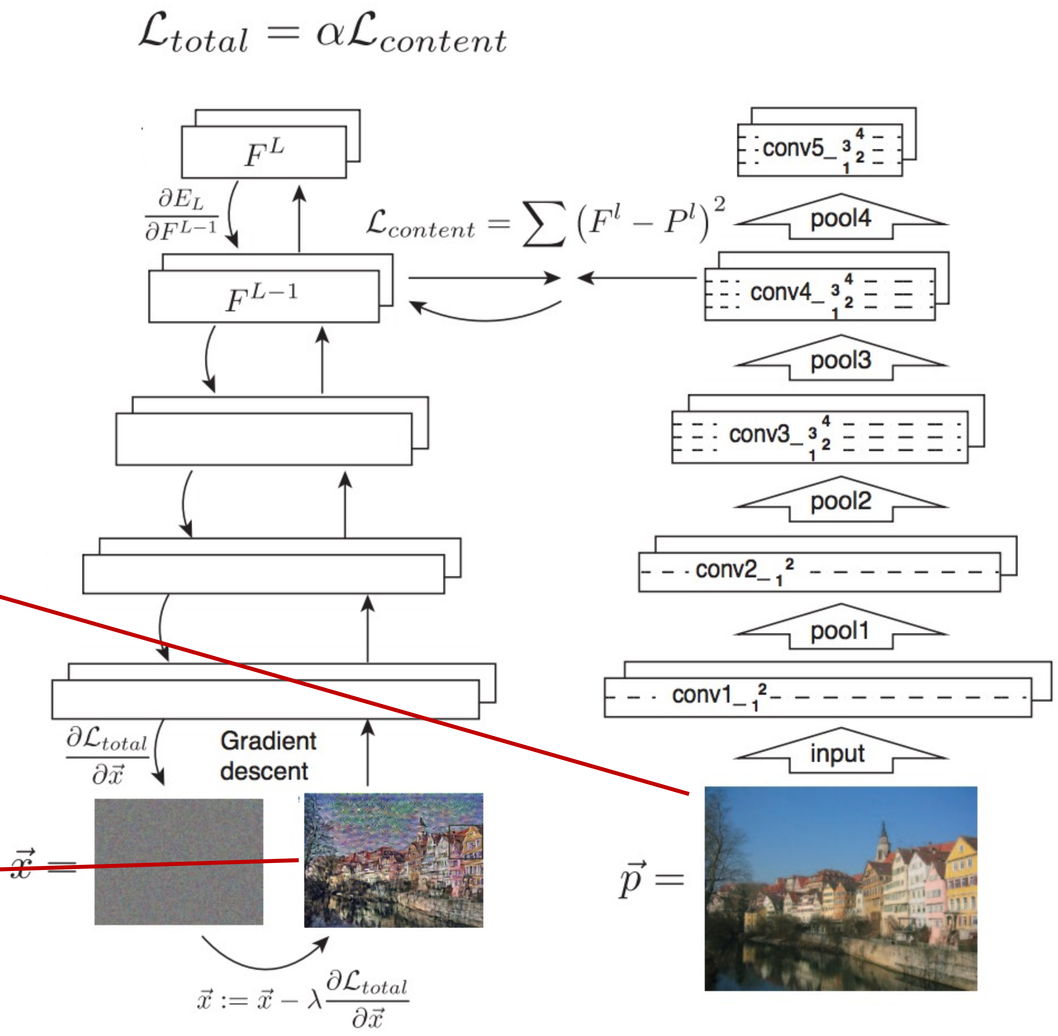
Idea 1: Image Reconstruction from Features

$$\mathcal{L}_{content} = \sum (F^l - P^l)^2$$



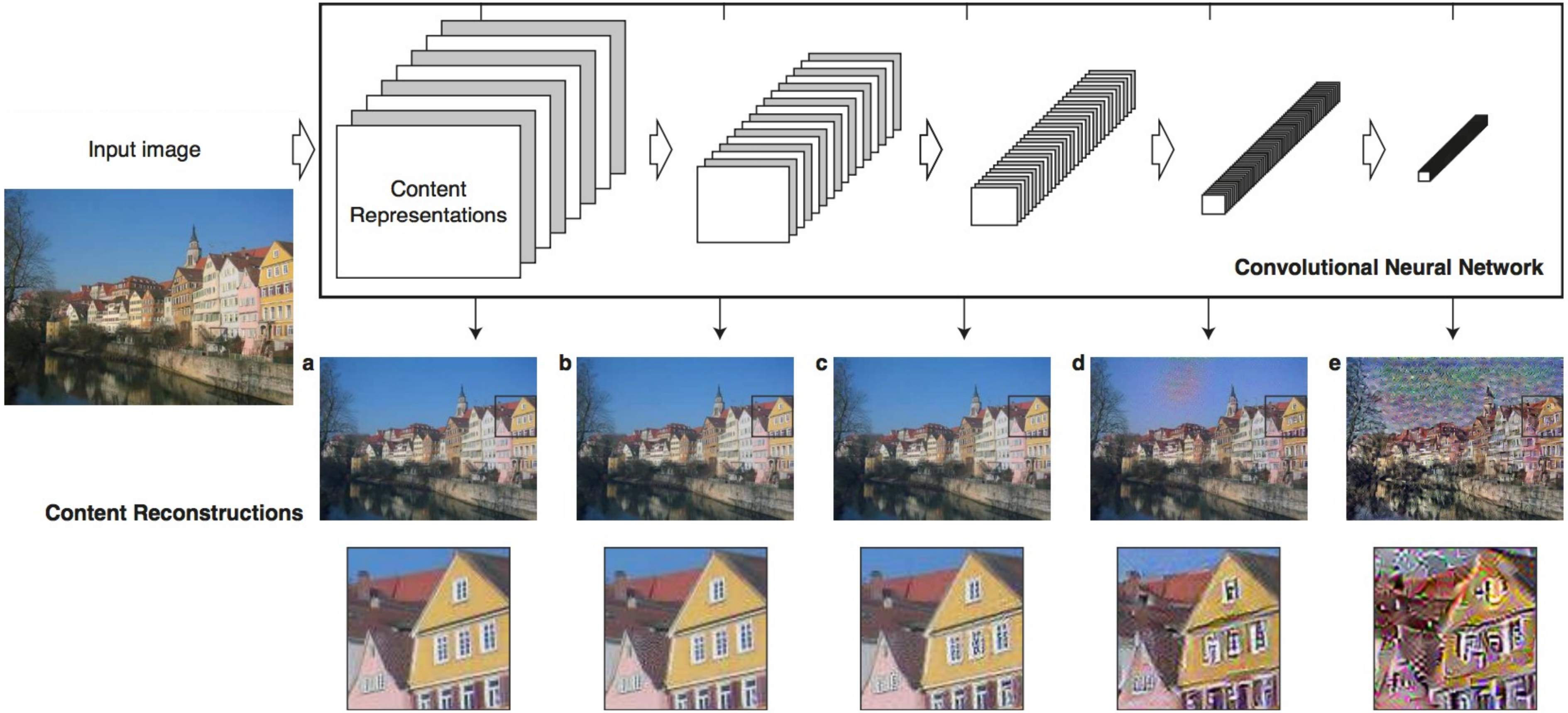
Gatys et. al. Image Style Transfer Using Convolutional Neural Networks. CVPR 2016

Idea 1: Image Reconstruction from Features



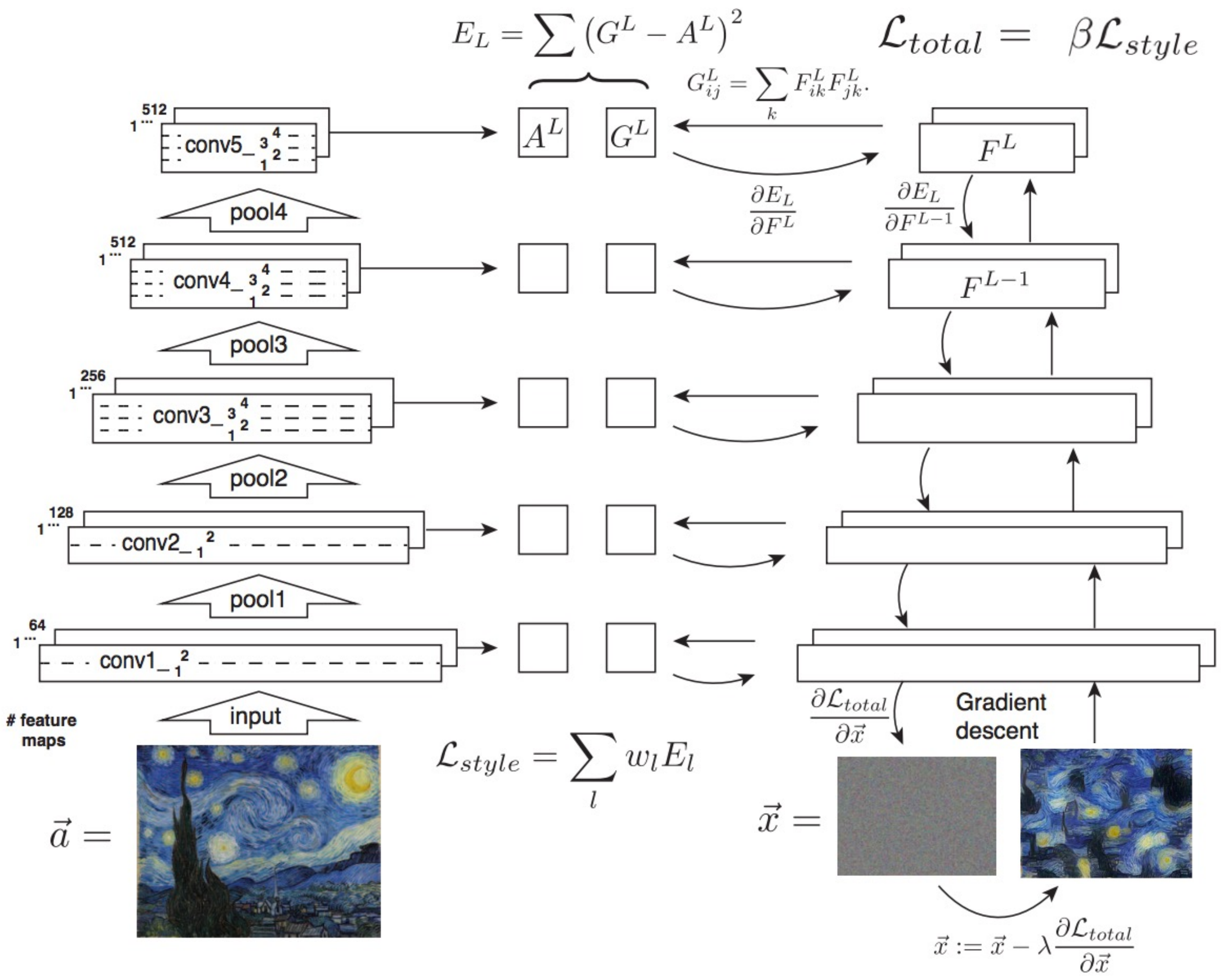
Gatys et. al. Image Style Transfer Using Convolutional Neural Networks. CVPR 2016

$$\mathcal{L}_{content} = \sum \left(F^l - P^l\right)^2$$

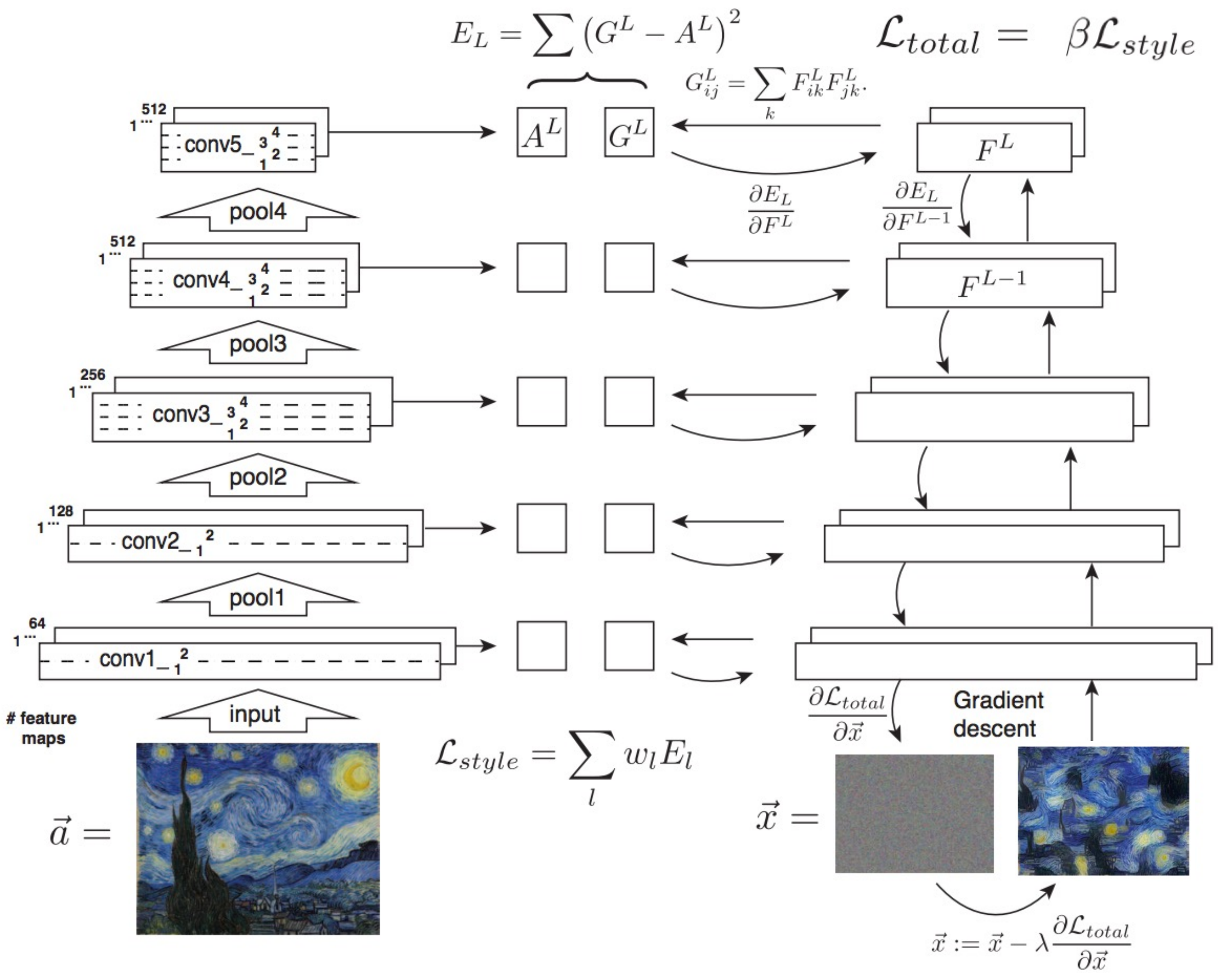


Gatys et. al. Image Style Transfer Using Convolutional Neural Networks. CVPR 2016

Idea 2: Backpropagation of Style



Gatys et. al. Image Style Transfer Using Convolutional Neural Networks. CVPR 2016



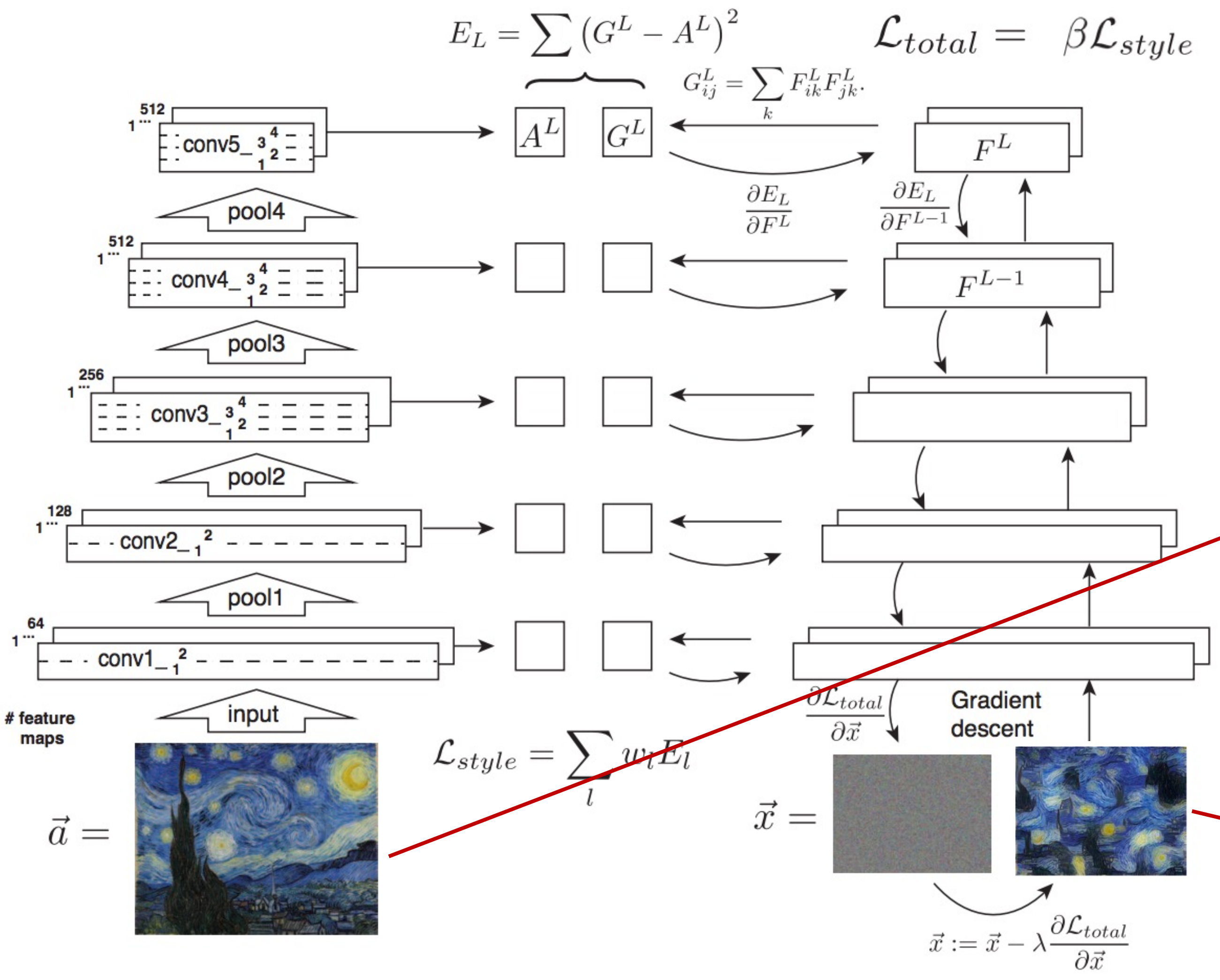
Idea 2: Backpropagation of Style

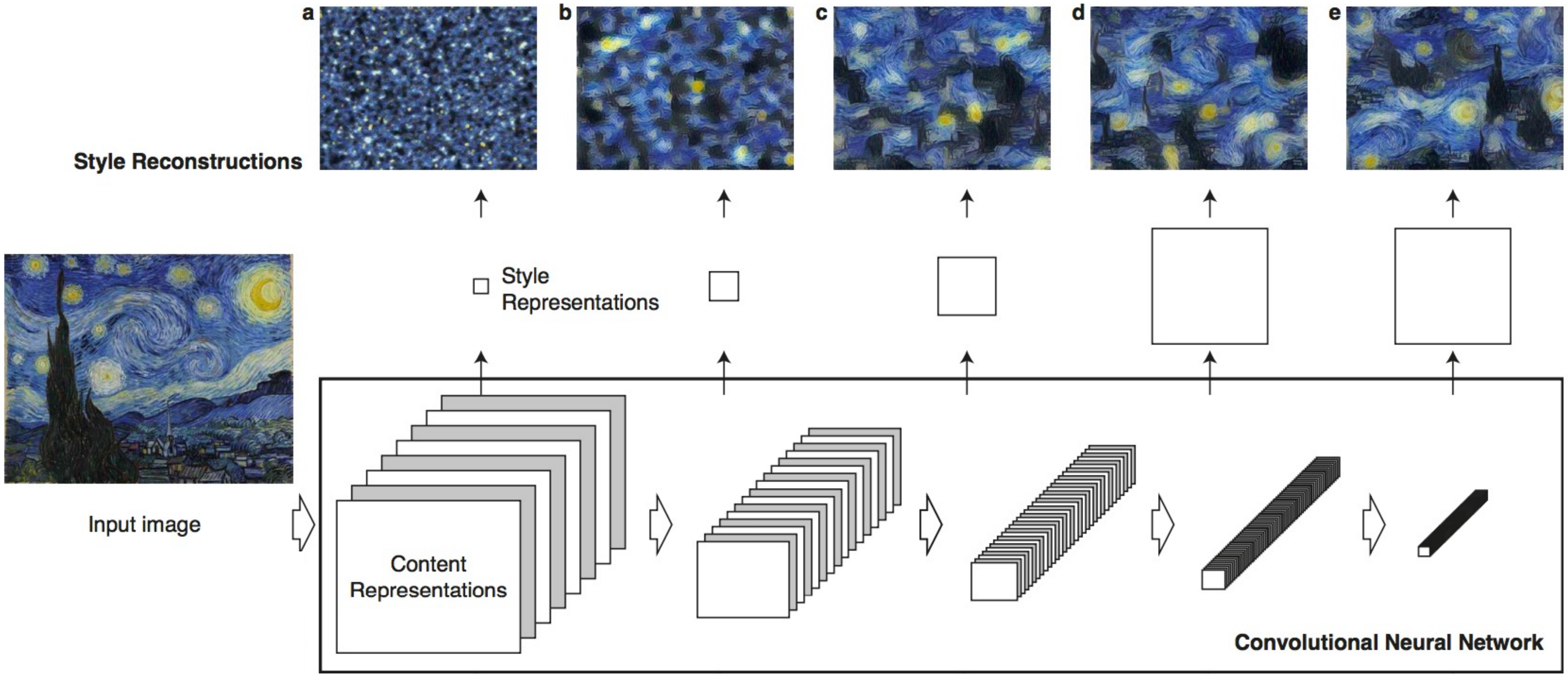
$$G^l_{ij} = \sum_k F^l_{ik} F^l_{jk}.$$

$$E_L = \sum (G^L - A^L)^2$$

Gatys et. al. Image Style Transfer Using Convolutional Neural Networks. CVPR 2016

Idea 2: Backpropagation of Style



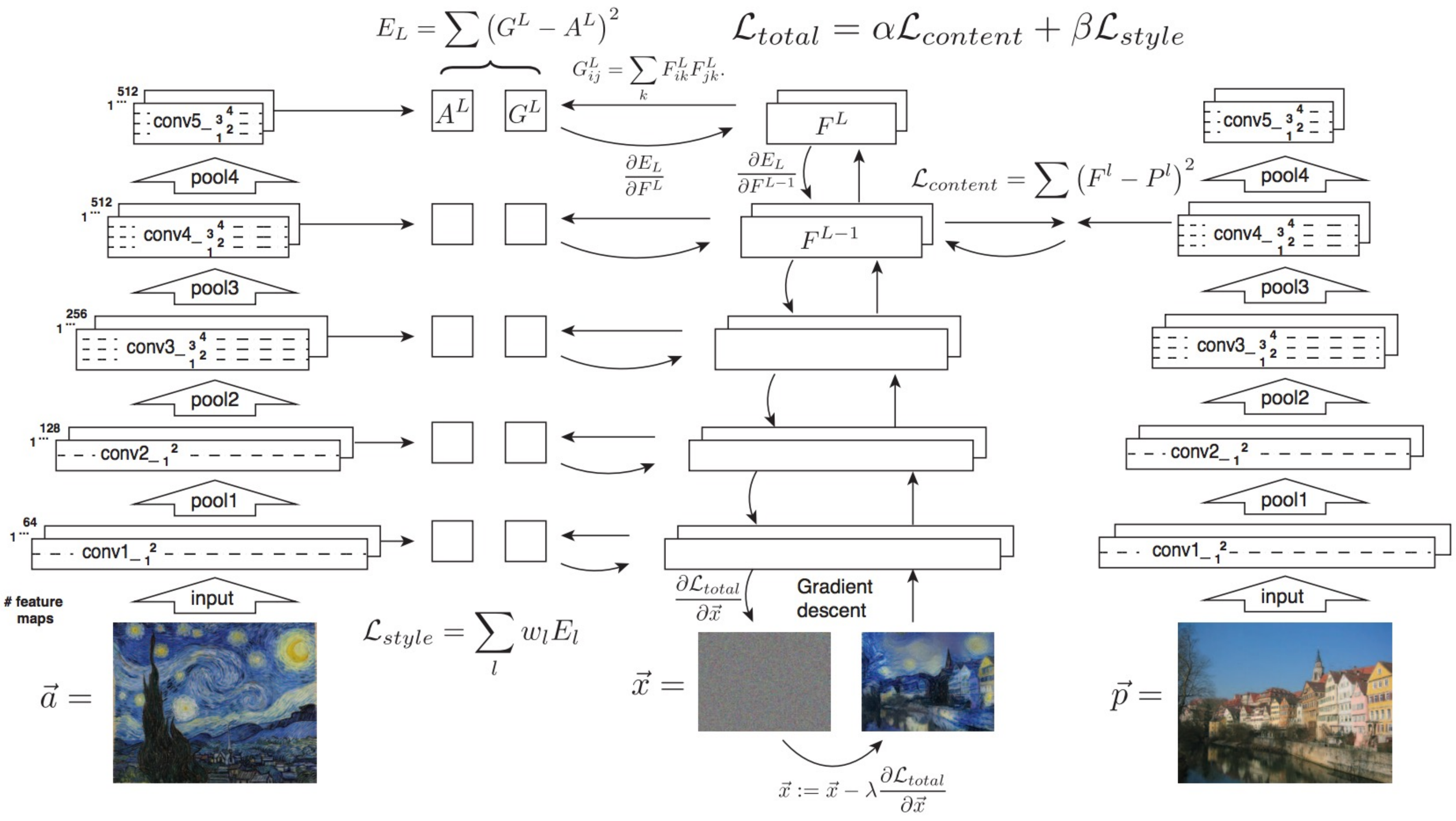


$$G^l_{ij} = \sum_k F^l_{ik} F^l_{jk}.$$

$$E_L = \sum (G^L - A^L)^2$$

Gatys et. al. Image Style Transfer Using Convolutional Neural Networks. CVPR 2016

Gatys et. al. Image Style Transfer Using Convolutional Neural Networks. CVPR 2016

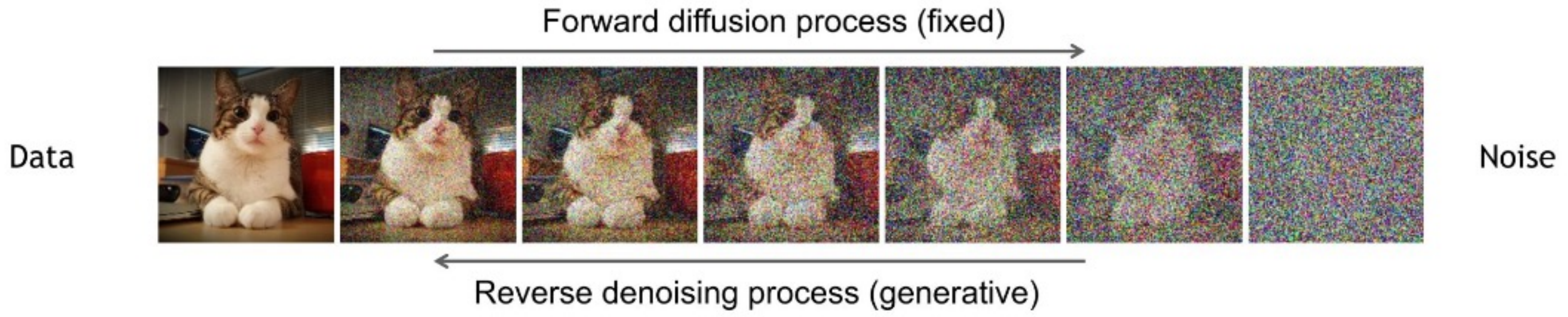


Gatys et. al. Image Style Transfer Using Convolutional Neural Networks. CVPR 2016

Denoising Diffusion Probabilistic Models (DDPM)

Forward diffusion: Markov chain of diffusion steps to slowly add gaussian noise to data

Reverse diffusion: A model is trained to generate data from noise by iterative denoising



Denoising Diffusion Probabilistic Models

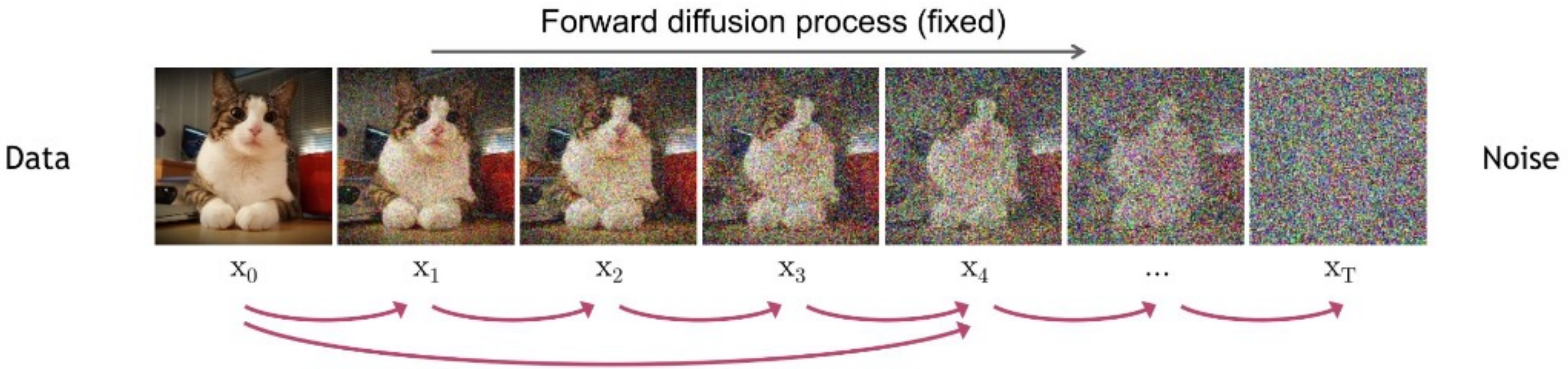
Jonathan Ho
UC Berkeley
jonathanho@berkeley.edu

Ajay Jain
UC Berkeley
ajayj@berkeley.edu

Pieter Abbeel
UC Berkeley
pabbeel@cs.berkeley.edu

Slides compiled by my student Aman Shrivastava

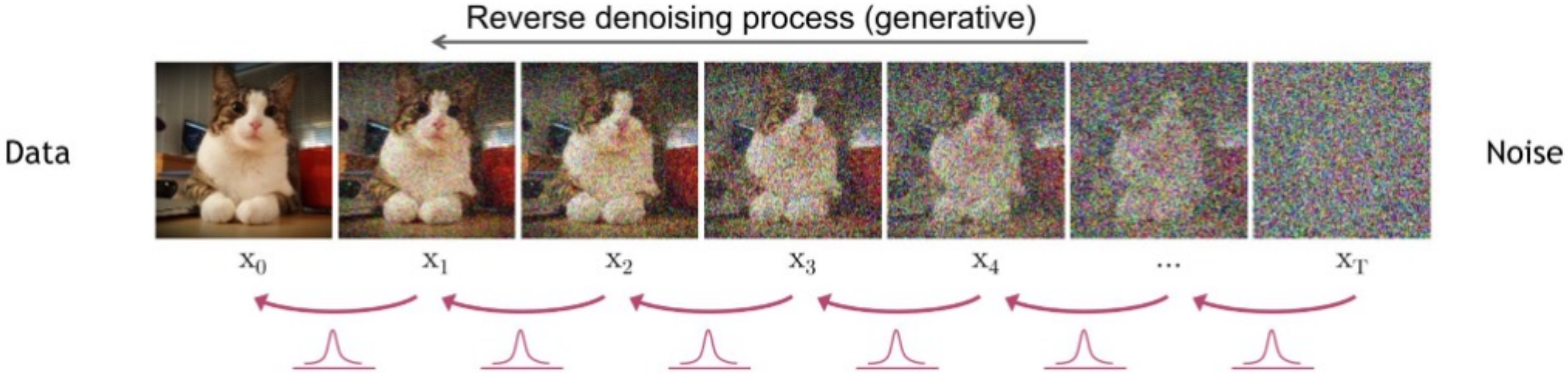
DDPM | Forward diffusion



We add a small amount of gaussian noise to a sample $\mathbf{x_0}$ in **T** timesteps to produces noised samples, $\{\mathbf{x_1}, \mathbf{x_2}, \dots, \mathbf{x_T}\}$. The steps are controlled by the noise schedule as follows:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}) \quad q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1})$$

DDPM | Reverse Diffusion

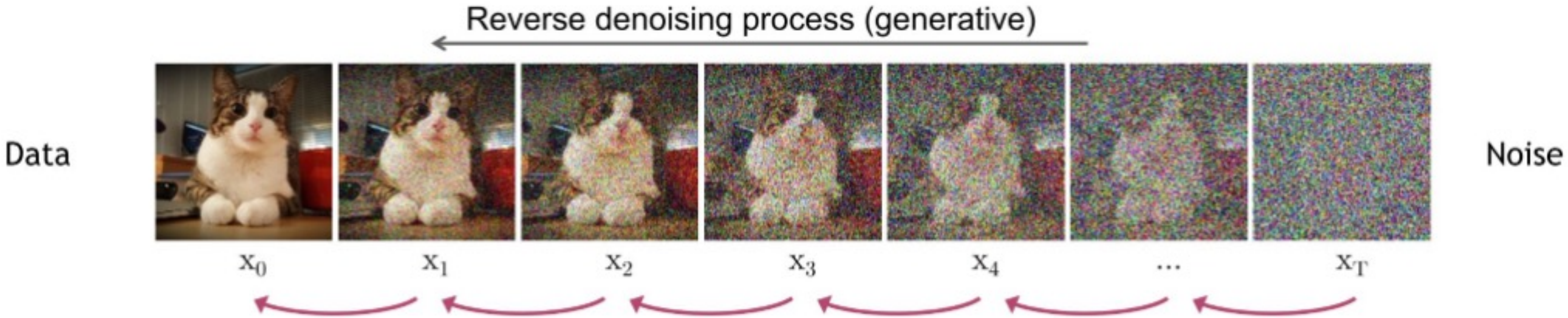


We learn a neural network model **(p_θ)** to approximate these conditional probabilities **$q(x_{(t-1)} | x_t)$** in order to run the reverse diffusion process as follows:

$$p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) \quad p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$$

Slides compiled by my student Aman Shrivastava

How do we train?



Algorithm 1 Training
repeat
$\quad \mathbf{x}_0 \sim q(\mathbf{x}_0)$
$\quad t \sim \mathrm{Uniform}(\{1, \dots, T\})$
$\quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
$\quad$ Take gradient descent step on $\nabla_\theta \left\lVert \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar\alpha_t}\mathbf{x}_0 + \sqrt{1-\bar\alpha_t}\boldsymbol{\epsilon}, t) \right\rVert^2$
until converged

Algorithm 2 Sampling
$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = T, \dots, 1$ **do**
$\quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
$\quad \mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar\alpha_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$
end for
return $\mathbf{x}_0$

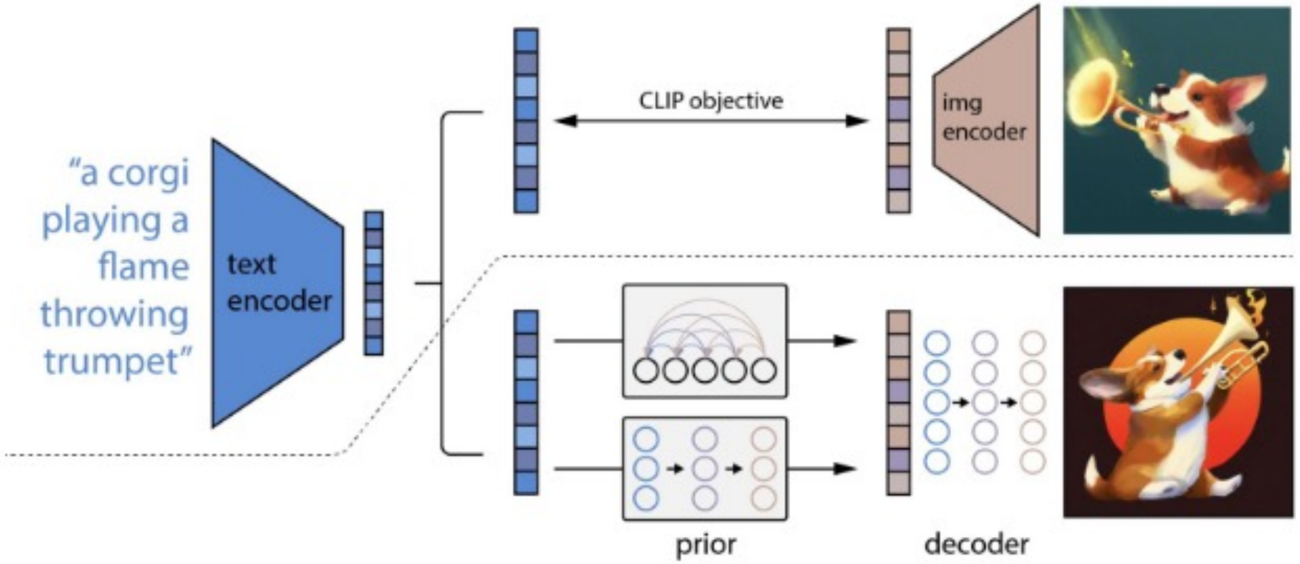
Slides compiled by my student Aman Shrivastava

DALL.E 2 | Open AI

Conditioning on CLIP-embeddings

- Helps capture multimodal representations

- The bi-partite latent enables several text-controlled image manipulation tasks



Ramesh, Aditya, et al. "Hierarchical text-conditional image generation with clip latents." *arXiv preprint arXiv:2204.06125* (2022).

DALL.E 2 | OpenAI

- 1kx1k text-conditioned image generation
- Uses a **prior** to produce CLIP embeddings conditioned on the text-caption
- Uses a **decoder** to produce images conditioned on the CLIP embeddings

a shiba inu wearing a beret and black turtleneck

a close up of a handpalm with leaves growing from it

panda mad scientist mixing sparkling chemicals, artstation

a corgi's head depicted as an explosion of a nebula

Slides compiled by my student Aman Shrivastava

Questions