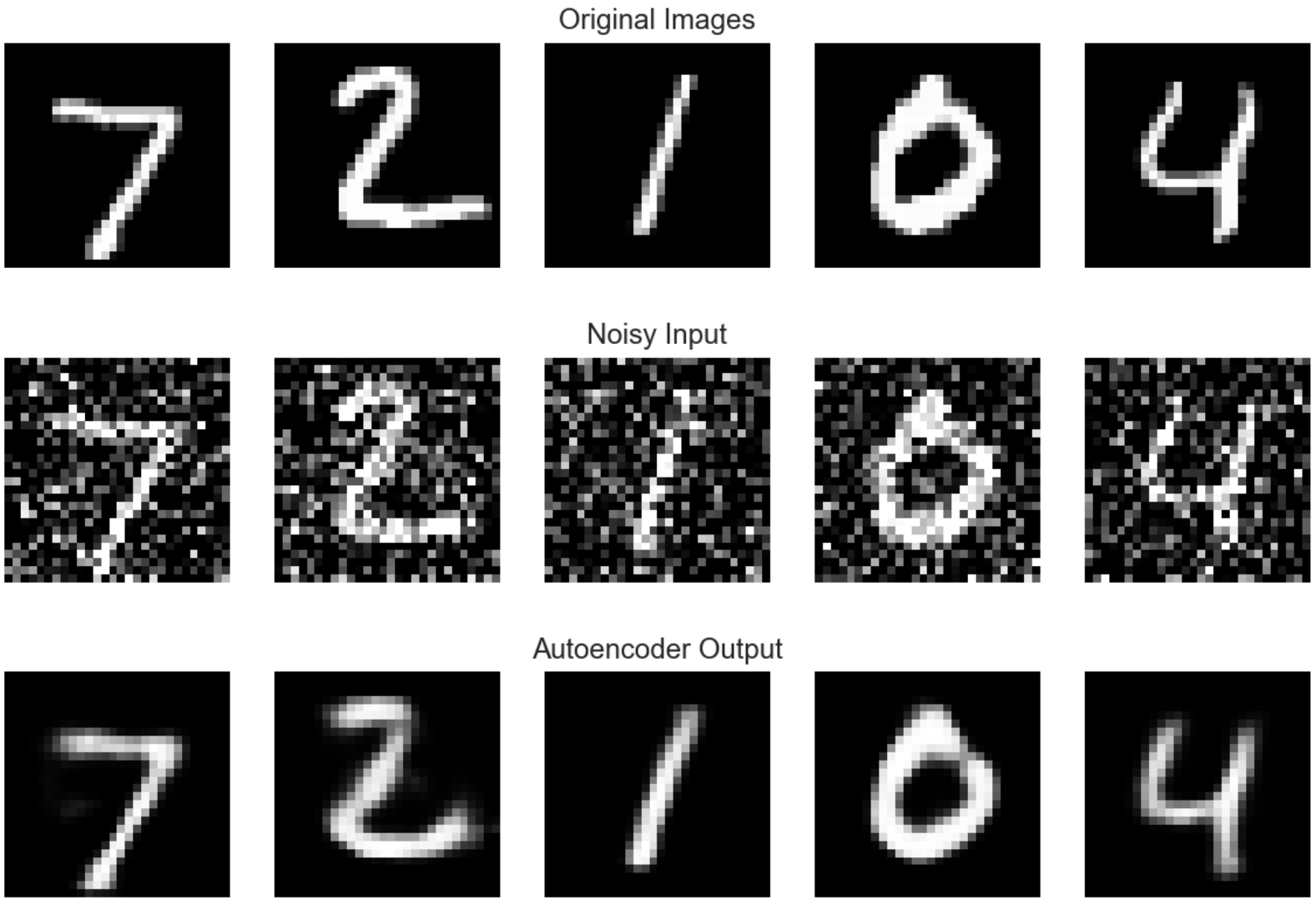
COMPSCI 689
Lecture 24: Multi-Modal Transcoders

Benjamin M. Marlin

College of Information and Computer Sciences
University of Massachusetts Amherst

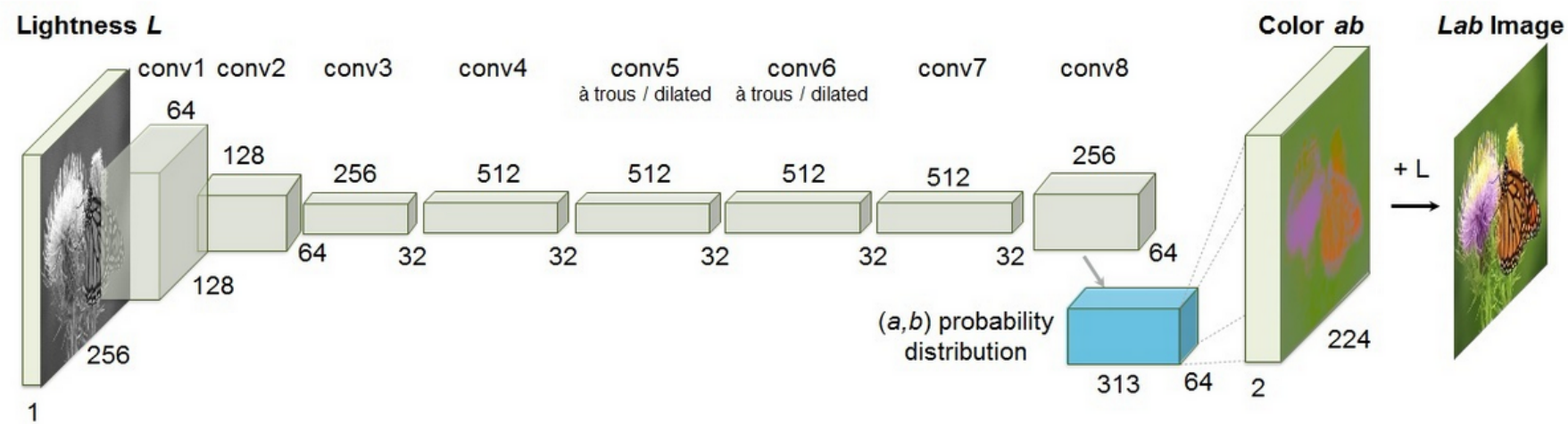
Slides by Benjamin M. Marlin (marlin@cs.umass.edu).

Deep Denoising Autoencoder for Images



Deep Autoencoder for Image Colorization

Figure 13: The CAE is trained to colorize the image



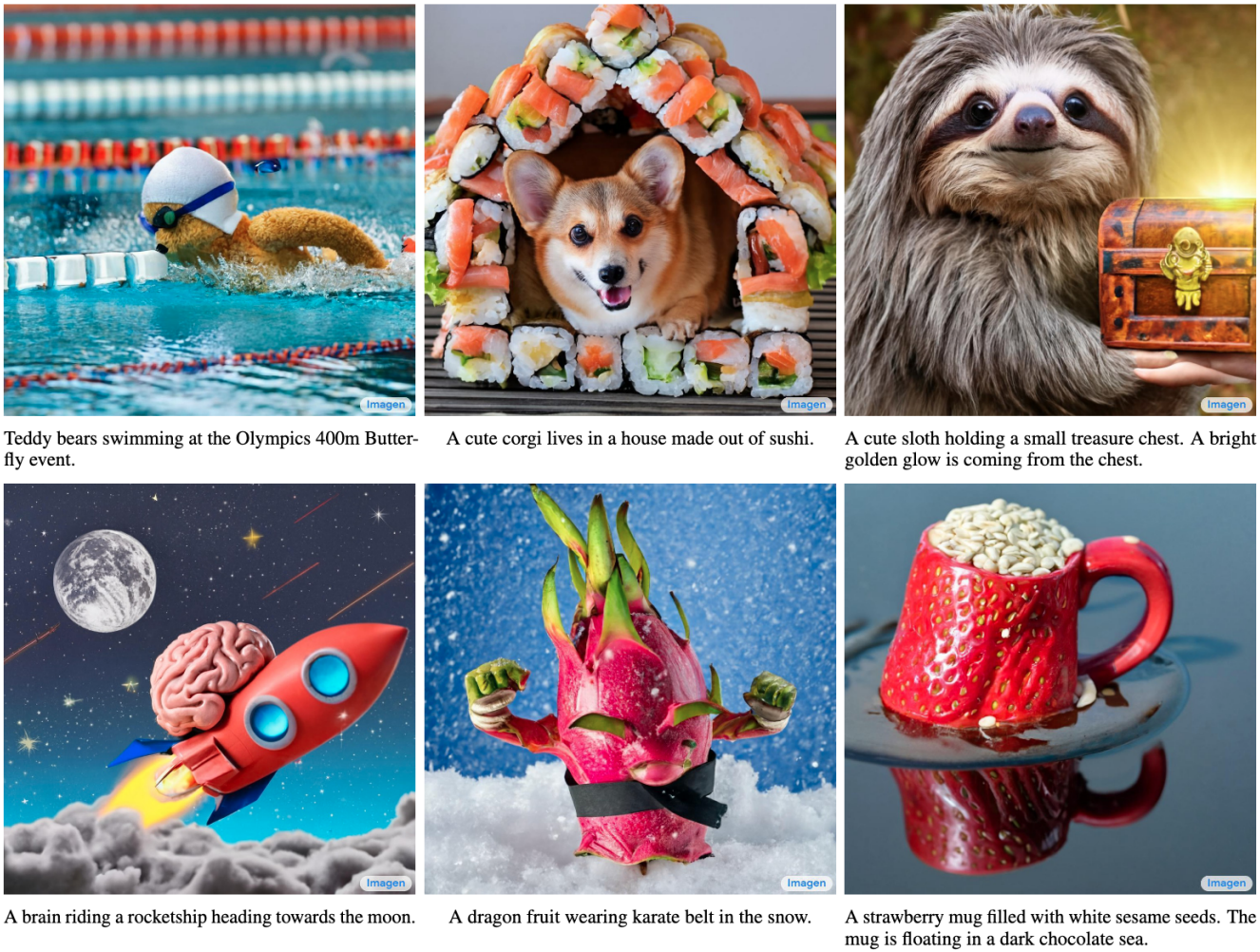
Class Conditioned Image Generation

Image Credit: `https://arxiv.org/pdf/2006.06676.pdf`
(Augmented GANS, 2020)

Text Conditioned Image Generation

Image Credit: `https://arxiv.org/pdf/2204.06125.pdf`
(Dall-e 2 Paper, April 2022)

Text Conditioned Image Generation



Teddy bears swimming at the Olympics 400m Butterfly event.

A cute corgi lives in a house made out of sushi.

A cute sloth holding a small treasure chest. A bright golden glow is coming from the chest.

A brain riding a rocketship heading towards the moon.

A dragon fruit wearing karate belt in the snow.

A strawberry mug filled with white sesame seeds. The mug is floating in a dark chocolate sea.

Image Credit: `https://arxiv.org/pdf/2205.11487.pdf`
(Imagen Paper, May 2022)

Text Conditioned Video Generation

Wooden figurine surfing on a surfboard in space.

Balloon full of water exploding in extreme slow motion.

Melting pistachio ice cream dripping down the cone.

Image Credit: `https://arxiv.org/pdf/2210.02303.pdf`
Videos: `https://imagen.research.google/video`
(Imagen Video Paper, Oct 2022)

Dall-e 2 Architecture Overview

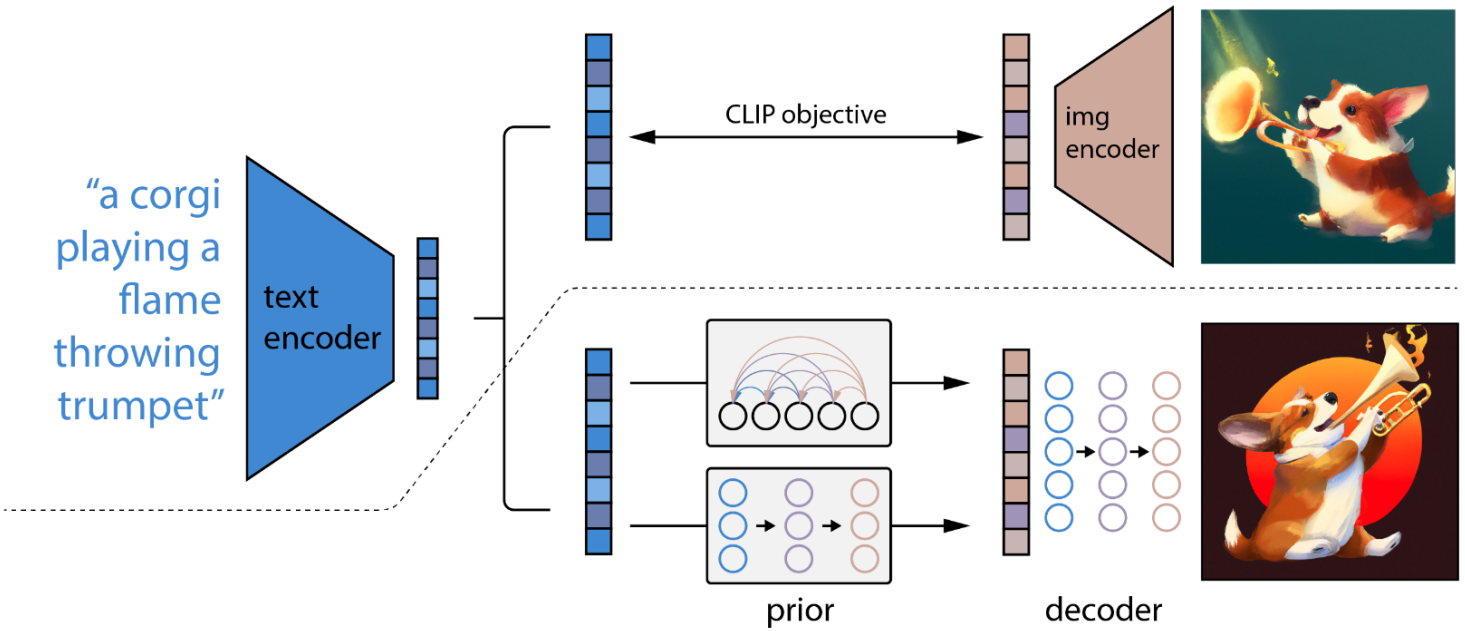


Image Credit: `https://arxiv.org/pdf/2204.06125.pdf`

Dall-e 2 Architecture Details

- Dall-e 2 uses a transformer-based model called CLIP as the basis for the text prompt to image latent code component. CLIP learns to embed text and images from data consisting of images with captions.
- Dall-e 2 first encodes the text into a latent text code, then transforms the latent text code into a latent image code using a diffusion model.
- Dall-e 2 then decodes the latent image code into a 64×64 image using a diffusion-based model called GLIDE.
- Lastly, Dall-e 2 progressively upsamples the initial image into 256×256 and 1024×1024 representations using a diffusion-based upsampler.

Imagen Architecture Overview

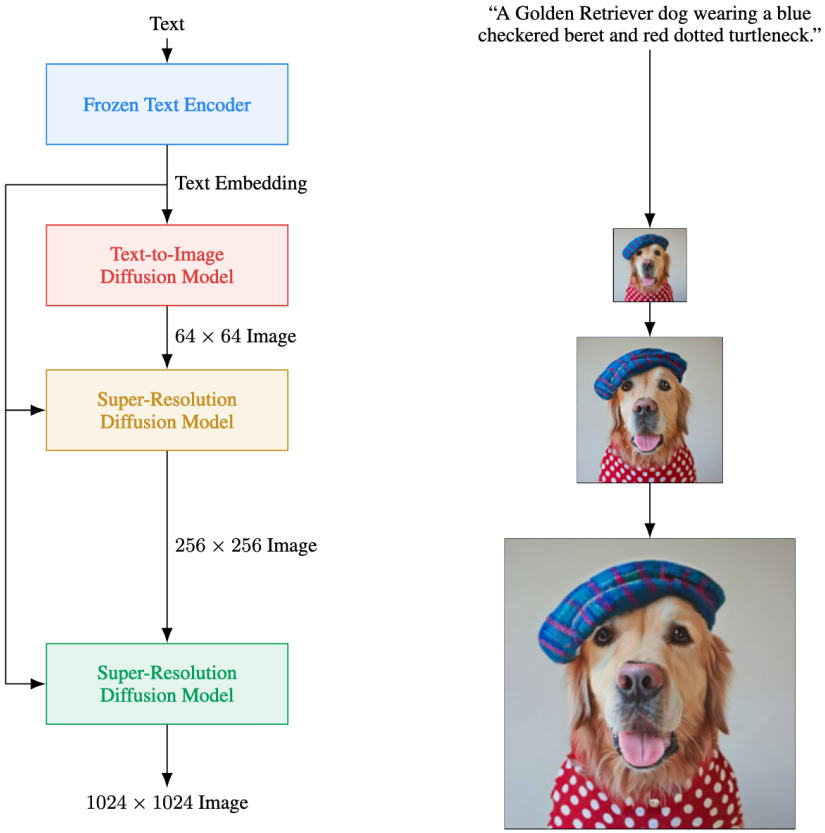


Image Credit: `https://arxiv.org/pdf/2205.11487.pdf`

Imagen 2 Architecture Details

- Imagen uses a transformer-based model called T5-XXL to produce an embedding of the text prompt. This model is trained on text data only and is then frozen.
- Imagen then decodes the the text embedding into a 64×64 image using a diffusion-based model.
- Imagen then progressively upsamples the initial image into 256×256 and 1024×1024 representations using a diffusion-based upsampler.

The CLIP Model

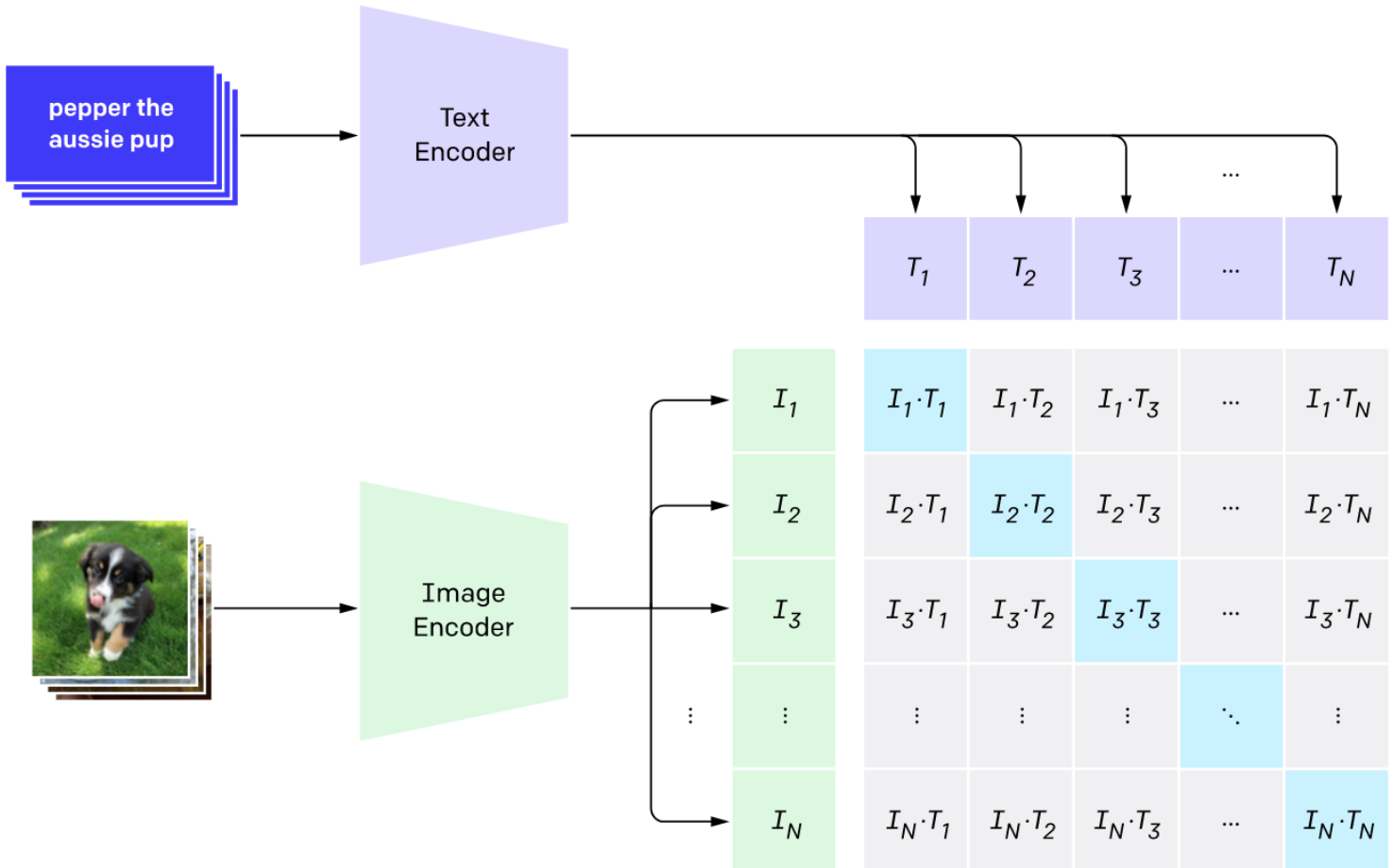


Image Credit: `https://arxiv.org/pdf/2103.00020.pdf`

CLIP Training Details

- Given a batch of N (image, text) pairs, CLIP is trained to predict which of the $N \times N$ possible (image, text) pairings across a batch actually occurred.
- CLIP learns a two embedding spaces by jointly training an image encoder and text encoder to maximize the cosine similarity of the image and text embeddings of the N real pairs in the batch while minimizing the cosine similarity of the embeddings of the incorrect pairs.

CLIP Architecture Details

- The CLIP paper describes using several different ResNet architectures as the image encoder. ResNet is a descendent of the AlexNet and LetNet5 convolutional models.
- The CLIP paper also describes using a vision transformer model as the image encoder. Visions transformers use the basic transformer architecture, but applied to blocks of pixels in a image instead of token embeddings.
- The CLIP text encoder is a straight transformer architecture with 12 layers, 63M parameters, a paired byte encoding and a maximum sequence length of 76 bytes (152 characters).

T5 Architecture Details

- The T5-XXL model is a standard transformer trained on the “Colossal Clean Crawled Corpus" (20TB).
- The model is trained on the standard “predict the next word" task.
- The model uses a 24 layer encoder and decoder and 1024 dimensional embeddings. The model has 11B parameters.

Diffusion Generators

- Both Dall-e 2 and Imagen use diffusion-based generators to generate 64×64 dimensional images from latent image embeddings.
- An un-guided diffusion model can be interpreted as a specialized type of non-linear factor analysis model $P(\mathbf{Z} = \mathbf{z}) = \mathcal{N}(\mathbf{z}; 0, I)$, $P(\mathbf{X} = \mathbf{x}, \mathbf{Z} = \mathbf{z}) = \mathcal{N}(\mathbf{x}; f_\theta(\mathbf{z}), \Psi)$.
- Like all non-linear factor analysis models, generation works by sampling a random vector $\mathbf{z} \sim \mathcal{N}(\mathbf{z}; 0, I)$, and then running that vector through a generator $f_\theta(\mathbf{z})$.

Diffusion Generators

- A diffusion model is trained to invert a probabilistic process where increasing amounts of noise are added to an input until the distribution of the input is transformed to $\mathcal{N}(\mathbf{z}; 0, I)$.
- This process looks like $q(\mathbf{z}_t|\mathbf{x}) = \mathcal{N}(\mathbf{z}_t; \alpha_t \mathbf{x}, \sigma_t^2)$.
- $0 \leq t \leq 1$ indexes the nosing process. α_t and σ_t follow a nosing schedule such that at $t = 1$, the distribution of $q(\mathbf{z}_t|\mathbf{x}) = \mathcal{N}(\mathbf{z}; 0, I)$.
- Samples z_{ts} are computed using reparameterization: $z_{ts} = \alpha_t \mathbf{x} + \epsilon_s \sigma_t$ where $\epsilon_s \sim N(0, I)$.

Conditional Diffusion Generators

- The diffusion generator is then trained to de-noise the noisy samples $\mathbf{z}_t$ over a uniform distribution of values of t.
- Under some additional assumptions, the learning problem simplifies to:

$$\mathbf{E}_{p(\epsilon,t)}\left[\sum_{n=1}^{N}\|f_\theta(\mathbf{z}_t)-\mathbf{x}_n\|_2^2\right]$$

- This looks similar to a denoising auto-encoder objective function, but it is derived as an approximation to a VAE objective function.
- Different specific model structures $f_\theta(\mathbf{z}_t)$ can be used. U-Net, a type of autoencoder with residual connections is commonly used.

Conditional Diffusion Generators

- In practice when decoding from a purely random sample, an iterative approach is used that samples progressively less noisey images.
- Examples: `https://ai.googleblog.com/2021/07/high-fidelity-image-generation-using.html`

Conditional Diffusion Generators

- A basic diffusion generator can generate low resolution photo realistic images, but there is no control over what is generated.
- To guide the generation toward a specific image, the generator can be conditioned on additional context information $\mathbf{c}_t$. The generator becomes the function $f_\theta(\mathbf{z}_t, \mathbf{c}_t)$.
- In both Imagen uses the T5-XXL prompt embedding as the context information. Dall-e 2 uses the CLIP image latent code as the context information.

Upsampling with Diffusion Generators

- Both Dall-e 2 and Imagen use diffusion-based upsampling to increase the resolution of the initially generated images.
- Diffusion-based upsampling is solved as a conditional diffusion generation problem where the context consists of both the text prompt, and the previously generated lower-dimensional image.
- Examples: `https://ai.googleblog.com/2021/07/high-fidelity-image-generation-using.html`