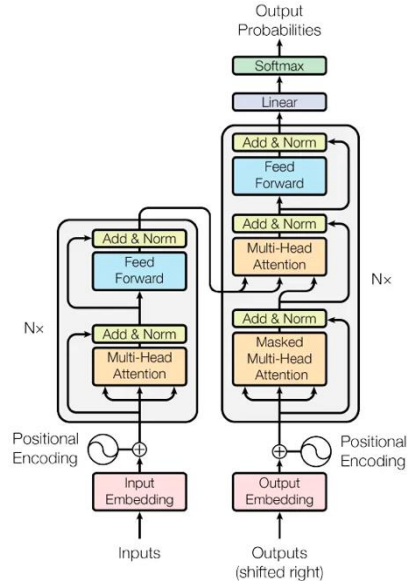
Generative Models (Stable Diffusion and Dall-E)

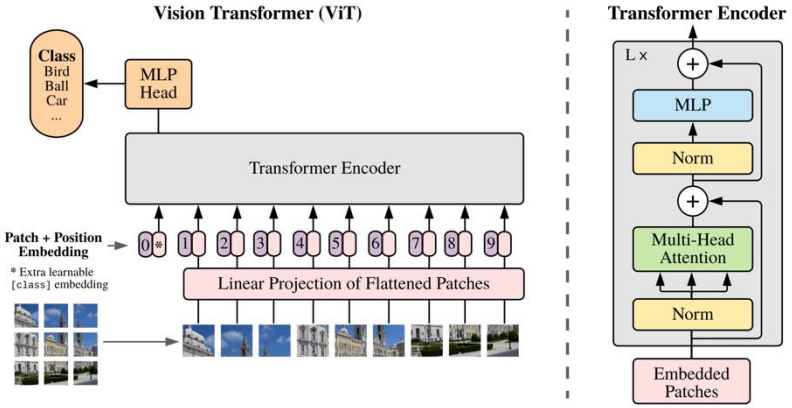
Transfomers for Robotics, Lecture 4, Nikolaus Correll

So far…

- Self-attention has replaced recurrent models as it is easier to train (parallel) and numerically stabler (vanishing gradients)
- Transformers use self-attention for text (lecture 2), images (lecture 3), and multi-modal input (lecture 4)
- Today: text-conditioned image generation



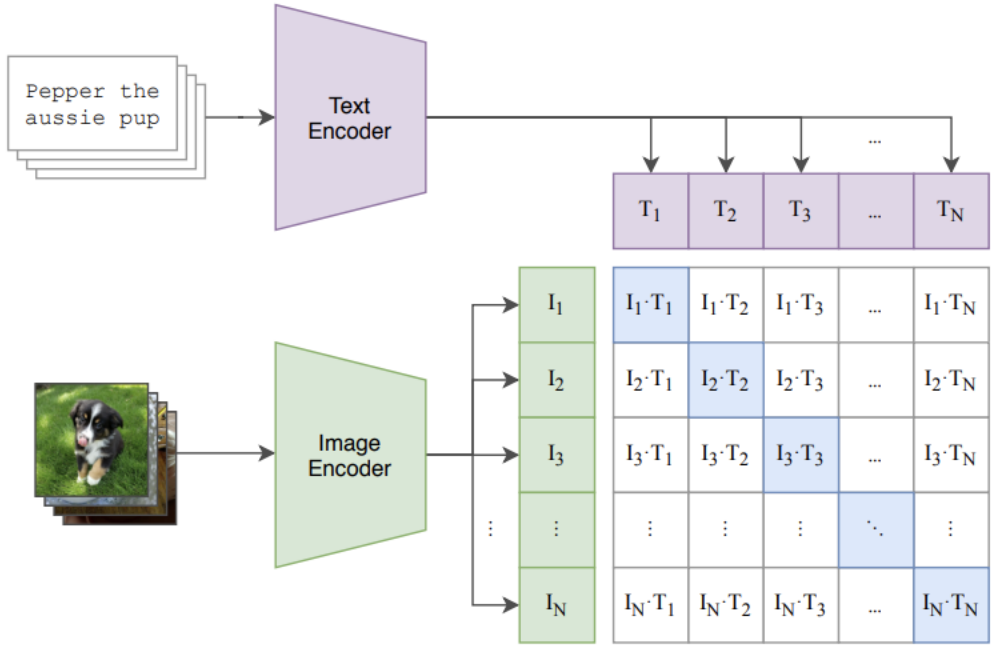
Encoder-Decoder for ChatGPT



Encoder for image classification

Last week: Contrastive Image-Language Pretraining

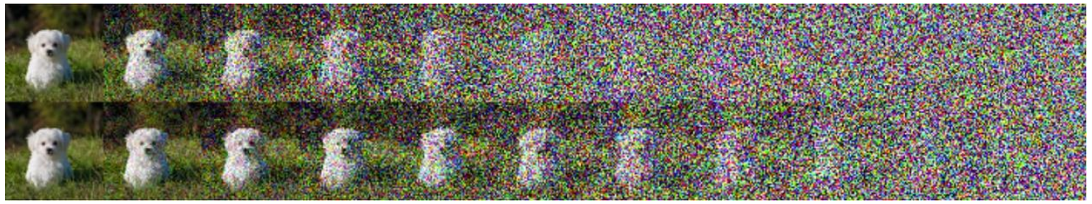
- Instead of hand-curated image-text pairs (300M for training ViT), use 400M text-image pairings from the internet
- Train Text and Image Encoder together
- **Maximize cosine similarity between correct pairs**
- Unprecedented zero-shot performance



Today

- Generative Models to create images
- Using text to condition image generation
- Wide variety of possible approaches
  - Transformer + Auto-encoder (Dall-e)
  - Transformer + Diffusion + Auto-encoder (Stable Diffusion, Dall-e 2)
- Trend: diffusion emerges as dominant technique in generative models

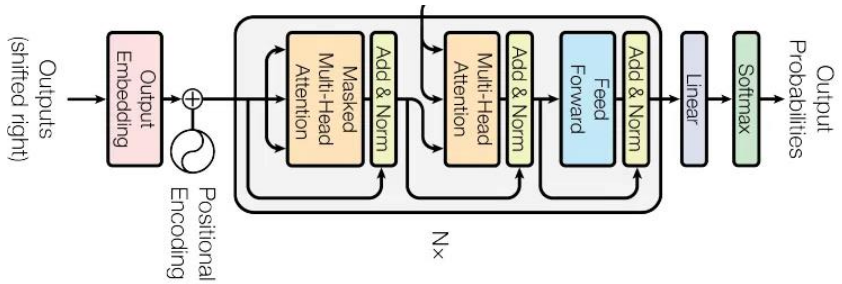
An armchair in the shape of an Avocado (Dall-E, 2021)

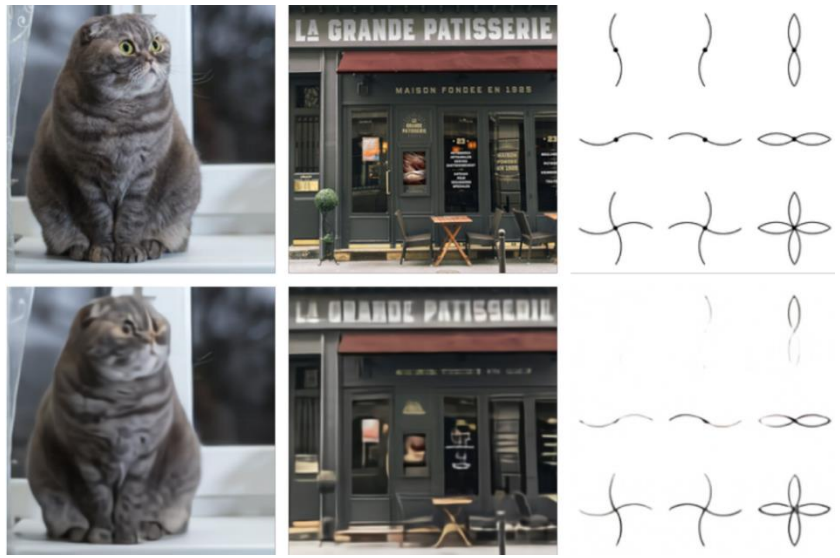


Adding noise at different schedules

DALL-E

- Auto-regressive transformer to generate image tokens from text/image input
- 250M training image/text pairs
- Two step-training:
  - Use auto-encoder to reduce 256x256x3 images to 32x32 latent space
  - 256 BPE text tokens + 1024 image tokens to train 12B parameter transformer
- Text is padded with a learned padding token – better performance



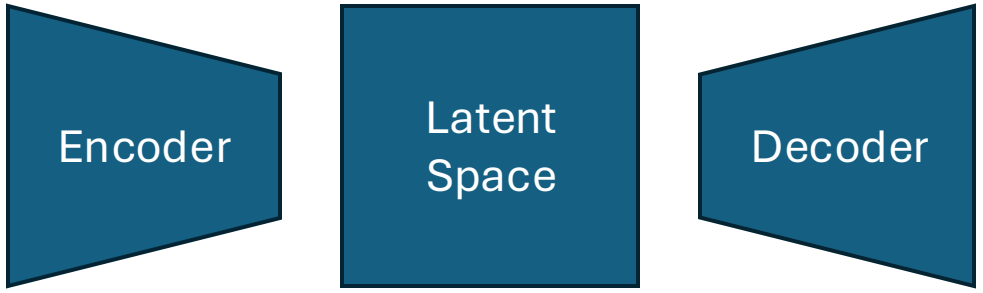


Auto-encoded images (bottom row)

Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M. and Sutskever, I., 2021, July. Zero-shot text-to-image generation. In *International conference on machine learning* (pp. 8821-8831). Pmlr.

The Auto-Encoder is trained to reproduce input

512x512



64x64

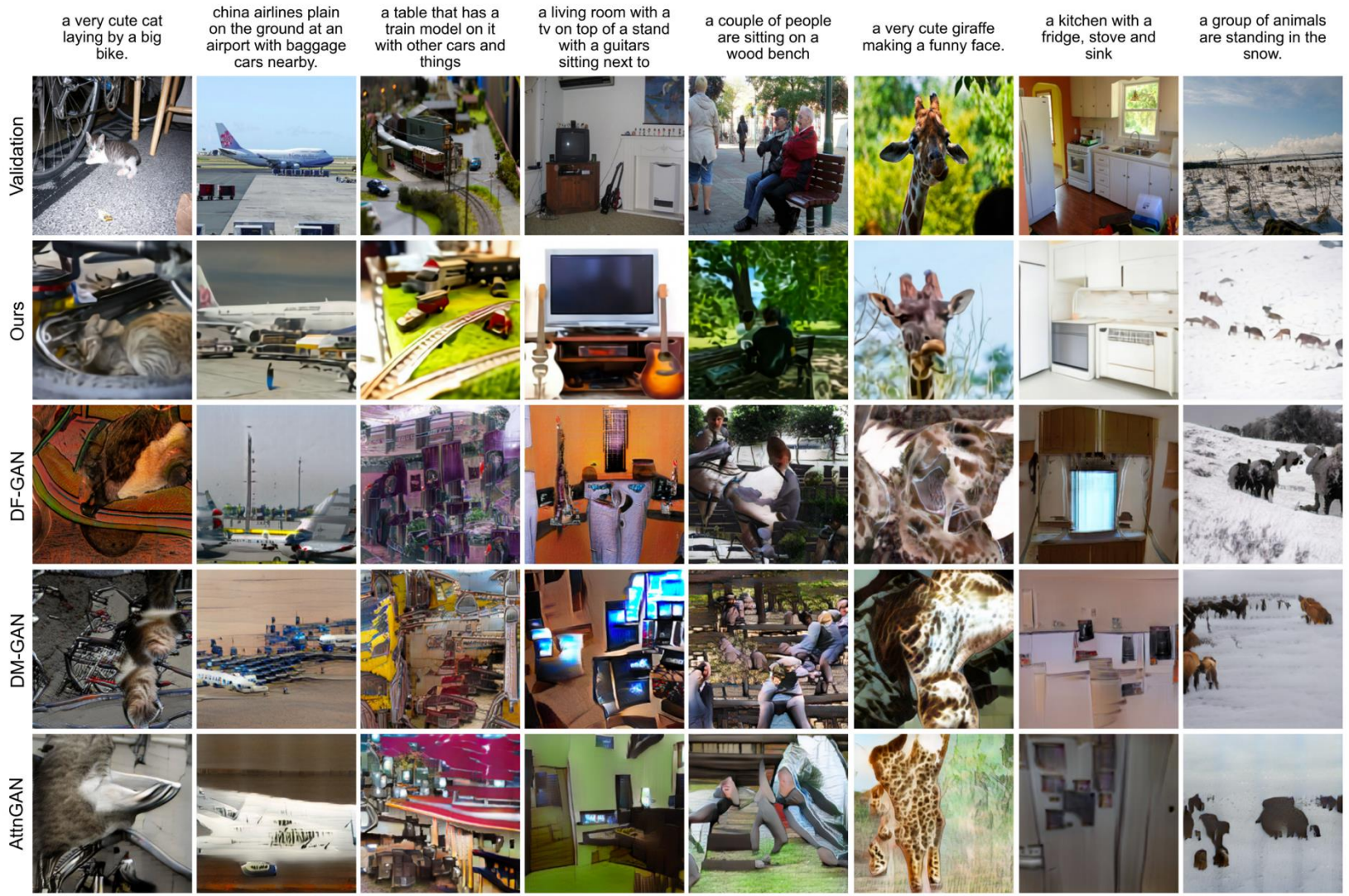
512x512

```
vae = AutoencoderKL.from_pretrained("CompVis/stable-diffusion-v1-4", subfolder="vae", use_auth_token=True)
```

83M parameters

320MB

Dall-E Results



Zero-shot image-to-image translation

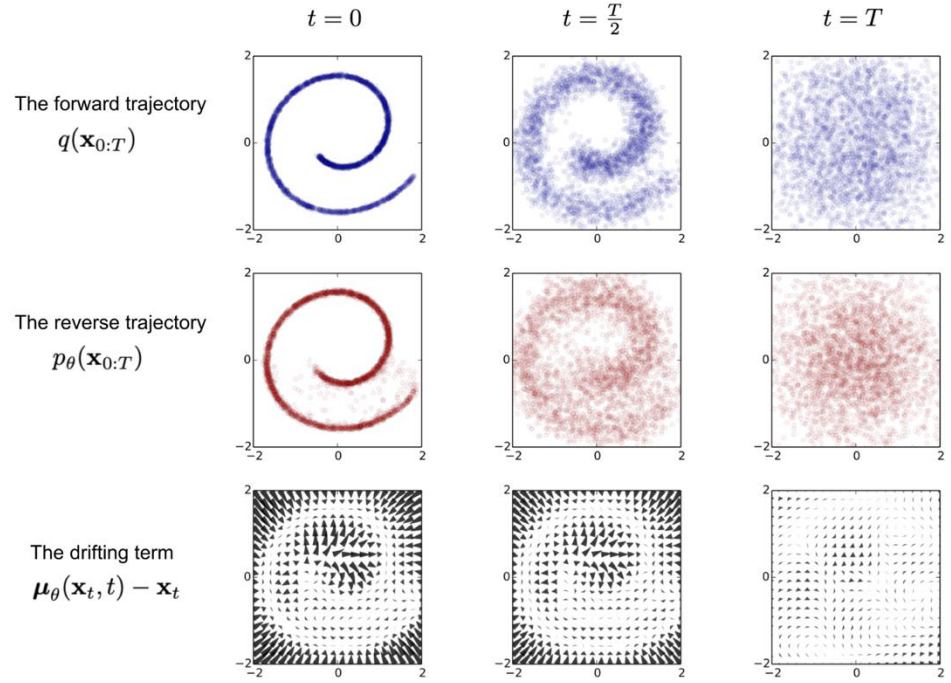
(d) “the exact same cat on the top colored red on the bottom”

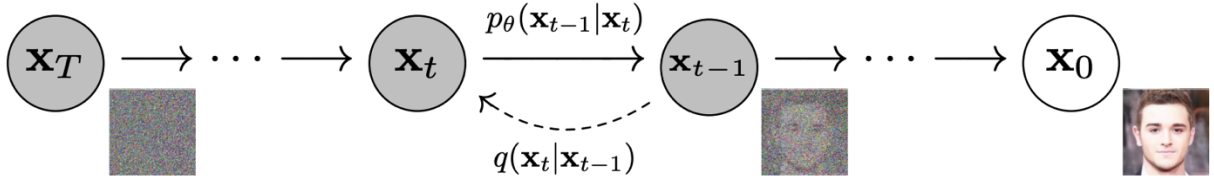
(e) “2 panel image of the exact same cat. on the top, a photo of the cat. on the bottom, the cat with sunglasses.”

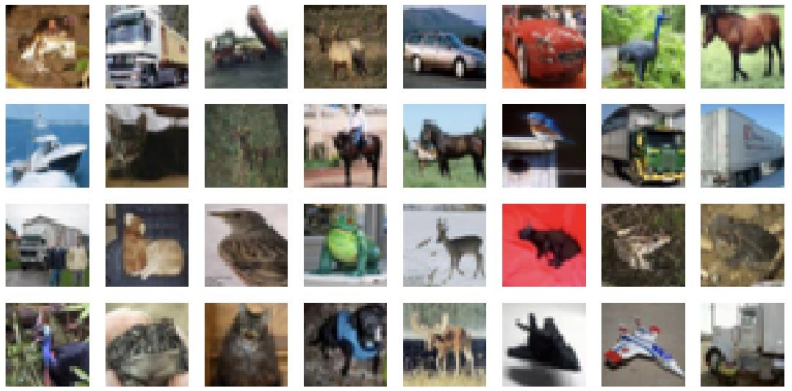
(f) “the exact same cat on the top as a postage stamp on the bottom”

Top half of the image is provided with the prompt

New, old idea: Diffusion





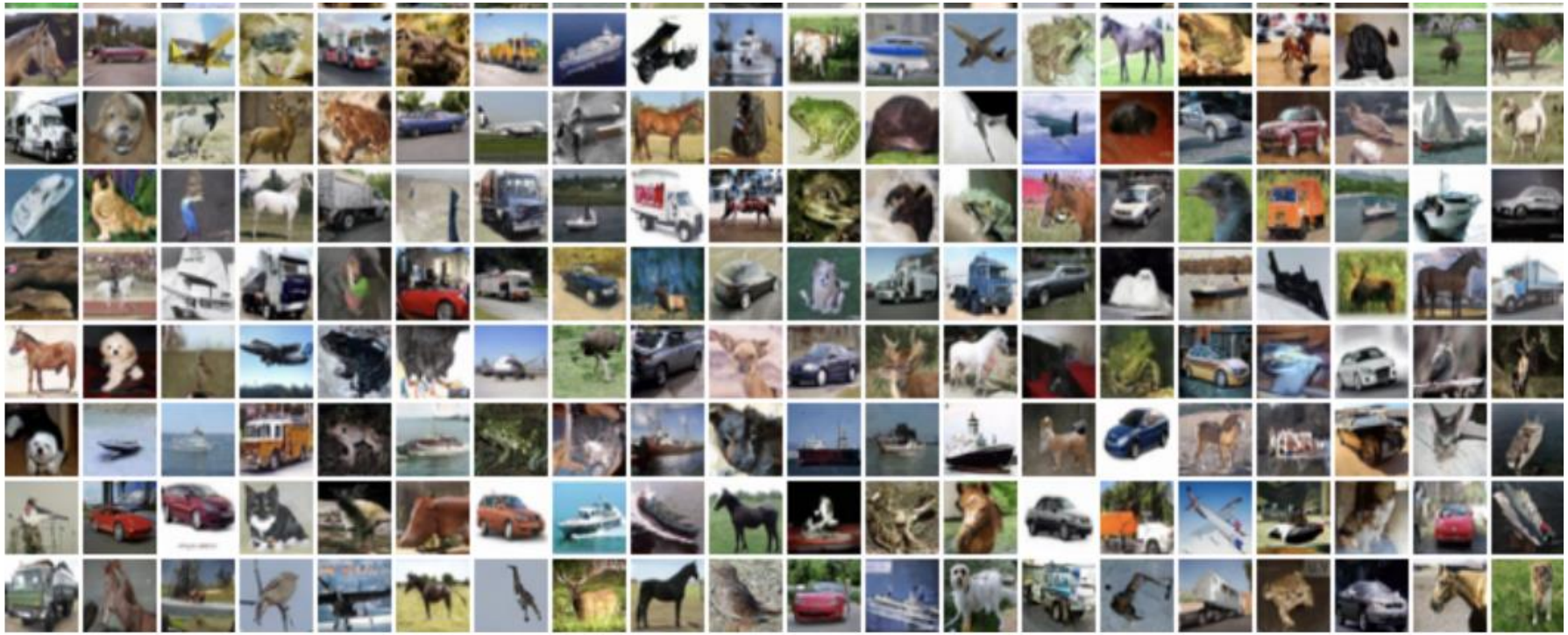


Training data (CIFAR)

Sample output

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N. and Ganguli, S., 2015, June. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning* (pp. 2256-2265). PMLR.

High-quality sample generation

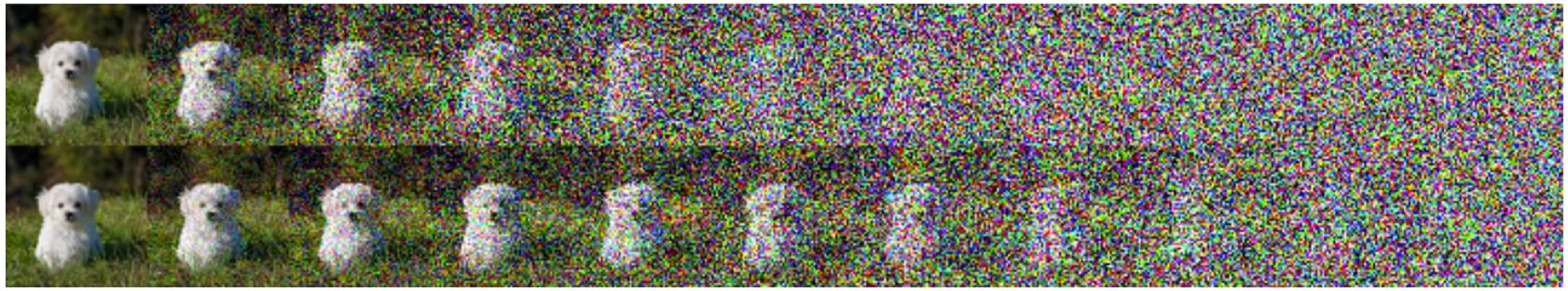


Better results using improved loss function when training

Ho, J., Jain, A. and Abbeel, P., 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, *33*, pp.6840-6851.

Forward Diffusion Scheduler

X_0 X_T



Model	ImageNet	CIFAR
Glow (Kingma & Dhariwal, 2018)	3.81	3.35
Flow++ (Ho et al., 2019)	3.69	3.08
PixelCNN (van den Oord et al., 2016c)	3.57	3.14
SPN (Menick & Kalchbrenner, 2018)	3.52	-
NVAE (Vahdat & Kautz, 2020)	-	2.91
Very Deep VAE (Child, 2020)	3.52	2.87
PixelSNAIL (Chen et al., 2018)	3.52	2.85
Image Transformer (Parmar et al., 2018)	3.48	2.90
Sparse Transformer (Child et al., 2019)	3.44	**2.80**
Routing Transformer (Roy et al., 2020)	**3.43**	-
DDPM (Ho et al., 2020)	3.77	3.70
DDPM (cont flow) (Song et al., 2020b)	-	2.99
Improved DDPM (ours)	**3.53**	**2.94**

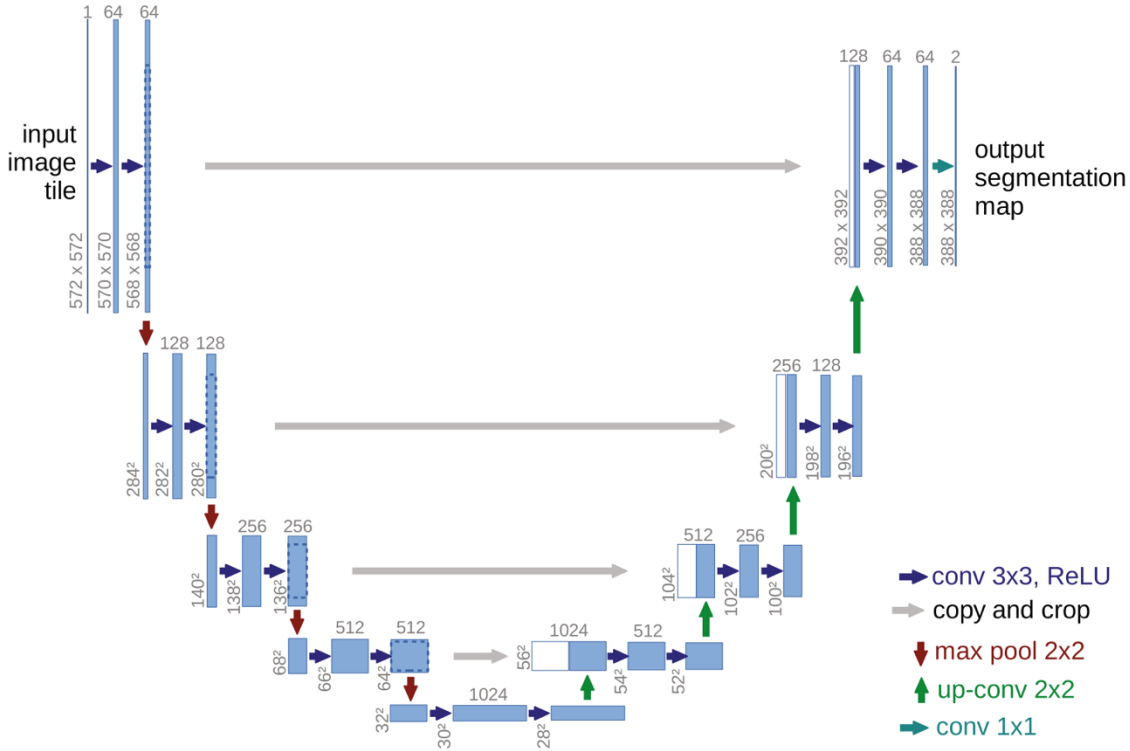
Cosine schedule creates lesser noise at the beginning and end, improving NLL

$$\beta_t = 1 - \frac{\bar{\alpha}_t}{\bar{\alpha}_{t-1}}, \quad \bar{\alpha}_t = \frac{f(t)}{f(0)}, \quad f(t) = \cos\left(\frac{t/T + s}{1+s} \cdot \frac{\pi}{2}\right)^2$$

Nichol, A.Q. and Dhariwal, P., 2021, July. Improved denoising diffusion probabilistic models. In *International conference on machine learning* (pp. 8162-8171). PMLR.

U-Net

- Downsampling, upsampling convolutional neural network
- Feature channels double at every down-sampling step
- Image information goes through “Bottleneck”
- Skip connections inform reconstruction
- Original application: pixel-wise segmentation



Ronneberger, O., Fischer, P. and Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18* (pp. 234-241). Springer International Publishing.

U-Net results

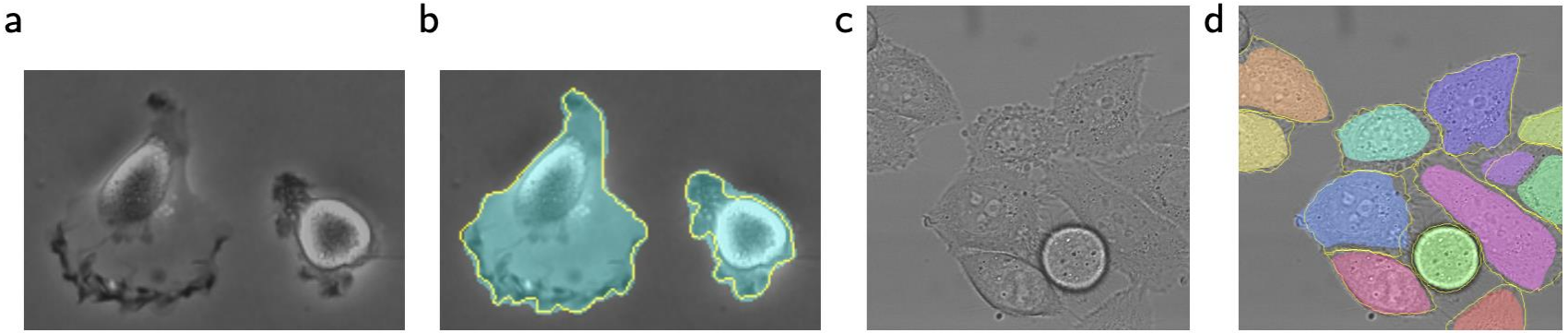
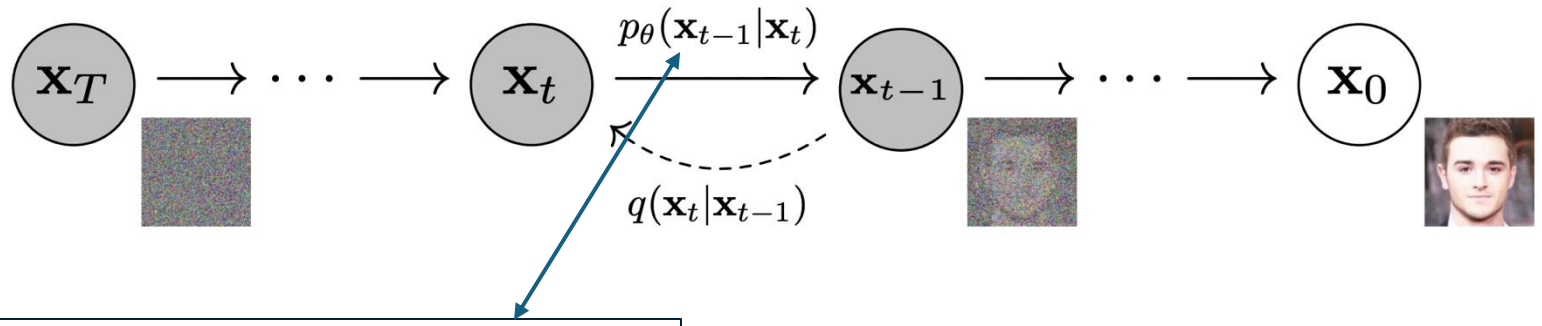
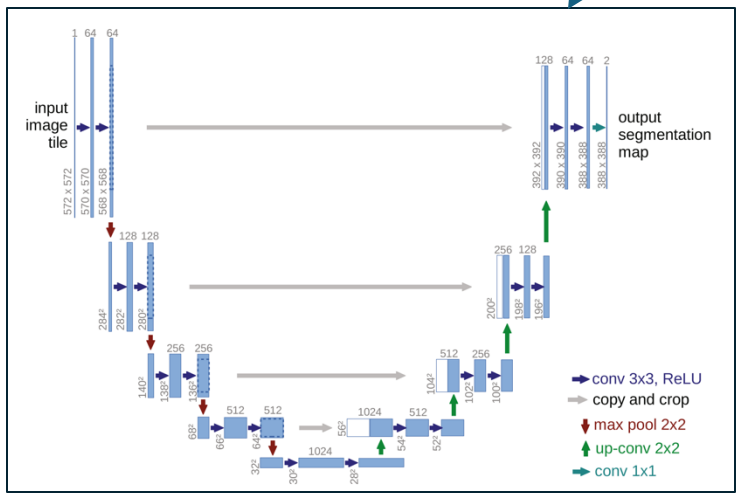


Fig. 4. Result on the ISBI cell tracking challenge. (**a**) part of an input image of the "PhC-U373" data set. (**b**) Segmentation result (cyan mask) with manual ground truth (yellow border) (**c**) input image of the "DIC-HeLa" data set. (**d**) Segmentation result (random colored masks) with manual ground truth (yellow border).

Ronneberger, O., Fischer, P. and Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18* (pp. 234-241). Springer International Publishing.

Learning the Noise





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

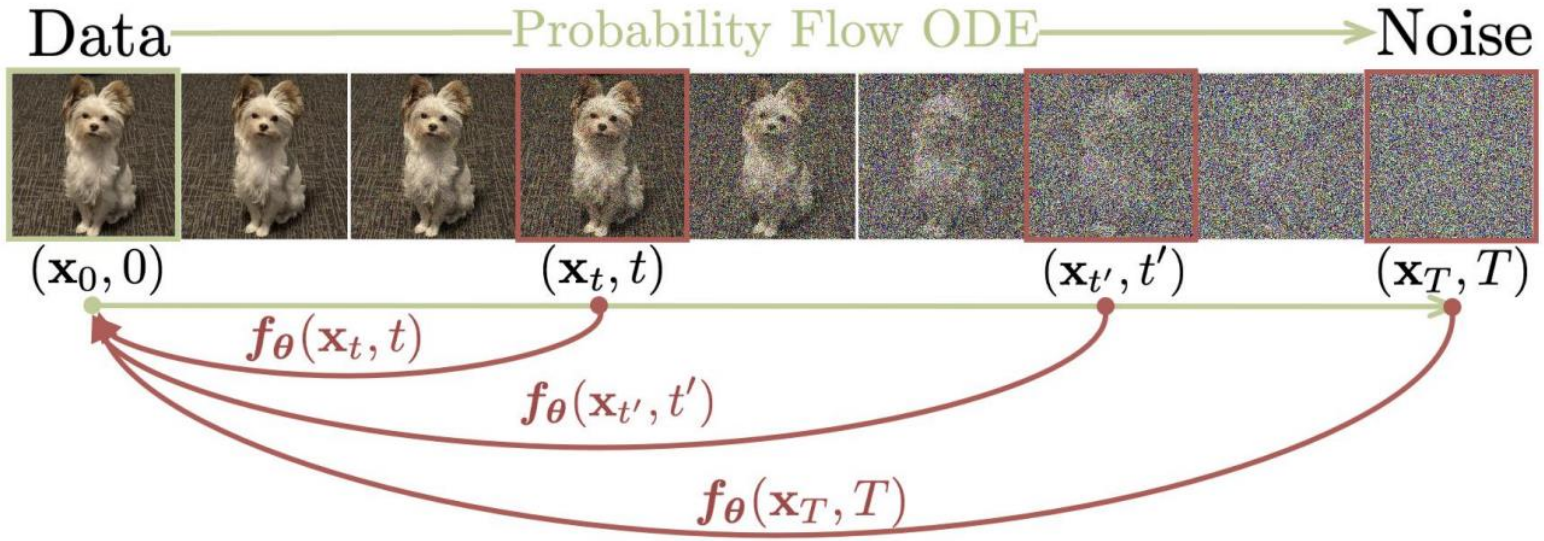
$$\alpha_t := 1 - \beta_t \text{ and } \bar{\alpha}_t := \prod_{s=1}^{t} \alpha_s$$

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})$$

Ho, J., Jain, A. and Abbeel, P., 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, *33*, pp.6840-6851.

Consistency Models

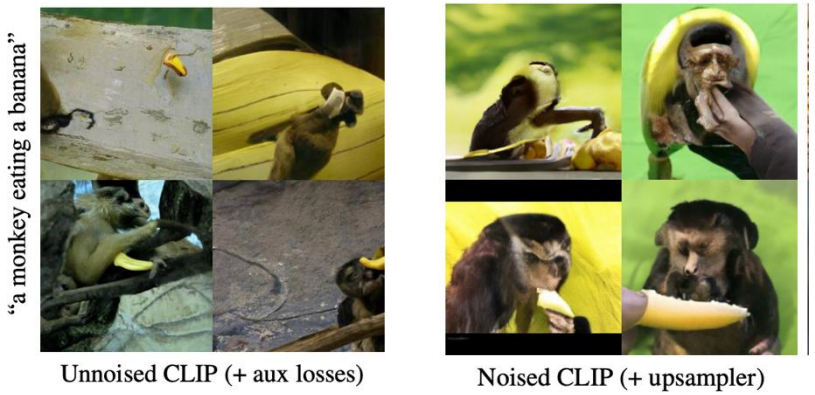
- Instead of learning noise step by step, train the U-Net on larger steps
- Result: much faster image generation



Song, Y., Dhariwal, P., Chen, M. and Sutskever, I., 2023. Consistency models. *arXiv preprint arXiv:2303.01469*.

Guiding Diffusion Toward Desired Results (GLIDE)

- Conditioned Diffusion: add a text embedding to the convolutional layers (bias)
- Classifier guidance: use an image classifier at the output
- Elegant solution: use CLIP contrastive loss between image-in-progress and desired caption
- Classifier-free guidance: randomly drop text embedding during training, combine un-conditioned and conditioned image at inference -> *highest quality in human evaluations*




GLIDE

Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I. and Chen, M., 2021. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*.

Guided Language to Image Diffusion for Generation and Editing

Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I. and Chen, M., 2021. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*.

Guided diffusion

- Guided diffusion: image x_t given class y

$$\hat{\mu}_\theta(x_t|y) = \mu_\theta(x_t|y) + s \cdot \Sigma_\theta(x_t|y)\nabla_{x_t} \log p_\phi(y|x_t)$$

Dhariwal, P. and Nichol, A., 2021. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, *34*, pp.8780-8794.

Increasing *s* (guidance scale) increases quality at cost of diversity

- Classifier free guidance: drop class embedding randomly during training, replace by 0, blend class-conditioned and random image at inference

$$\hat{\epsilon}_\theta(x_t|y) = \epsilon_\theta(x_t|\emptyset) + s \cdot (\epsilon_\theta(x_t|y) - \epsilon_\theta(x_t|\emptyset))$$

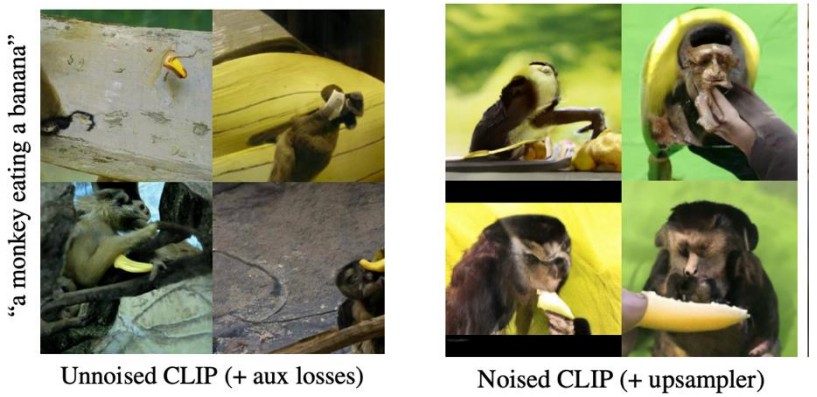
Ho, J. and Salimans, T. Classifier-free diffusion guidance. In NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications, 2021.

CLIP-Guided diffusion

- CLIP-guided diffusion: image x_t given caption c

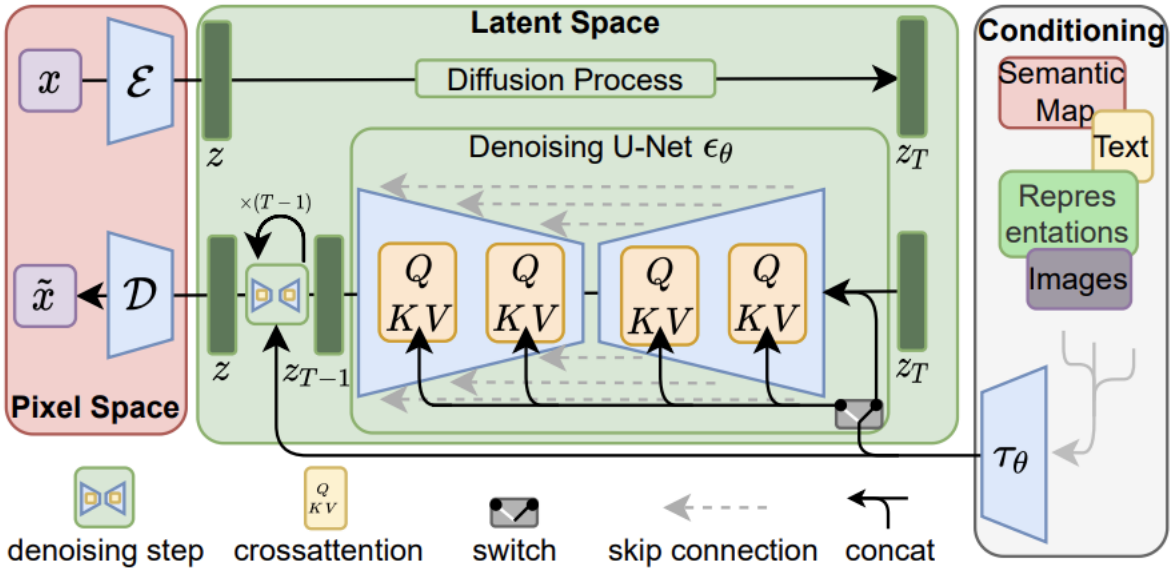
$$\hat{\mu}_\theta(x_t|c) = \mu_\theta(x_t|c) + s \cdot \Sigma_\theta(x_t|c)\nabla_{x_t}\left(f(x_t) \cdot g(c)\right)$$

- Need to train CLIP model on noisy versions of the image for best results



Latent Diffusion Models

- Idea: perform diffusion in the reduced latent space of an auto-encoder
- U-Net with diffusion is passed into cross-attention with conditioning tokens
- Faster training, smaller models



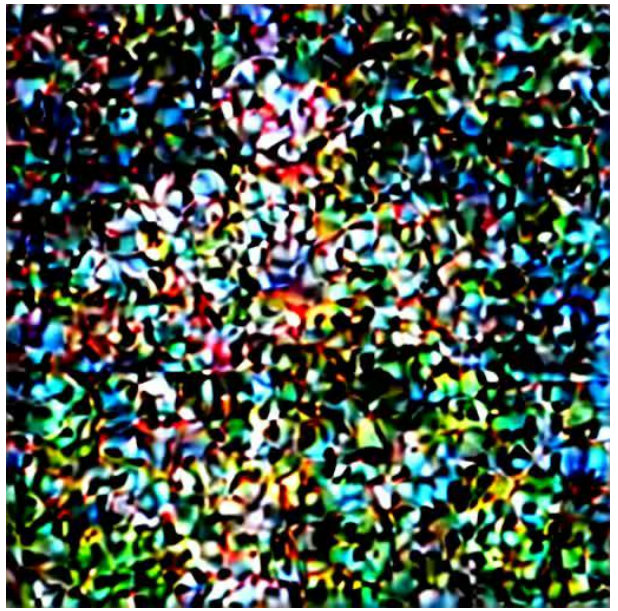
Rombach, R., Blattmann, A., Lorenz, D., Esser, P. and Ommer, B., 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 10684-10695).

Stable Diffusion

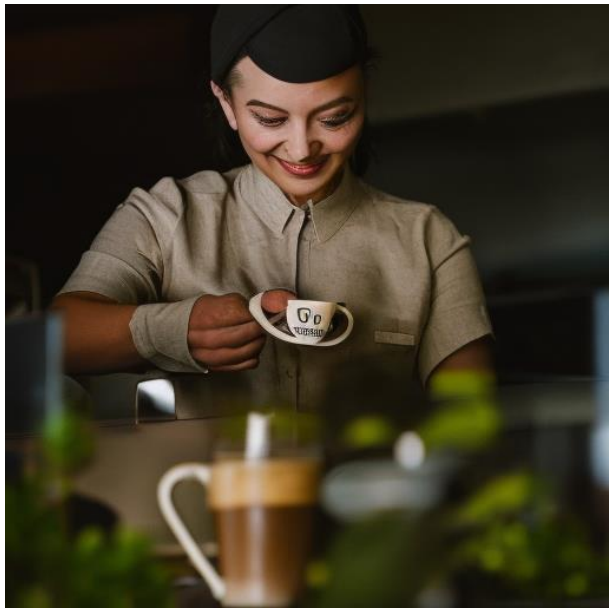
- Prompt: `a barista handing over a coffee`
- Start with random noise
- A trained U-Net predicts the noise that must have been added to an image of the prompt in a latent space
- Cross-attention with CLIP embedding of promp
- An Auto-Encoder scales the image up

Stable Diffusion from Seed (2/5th)

Step 0



Step 20

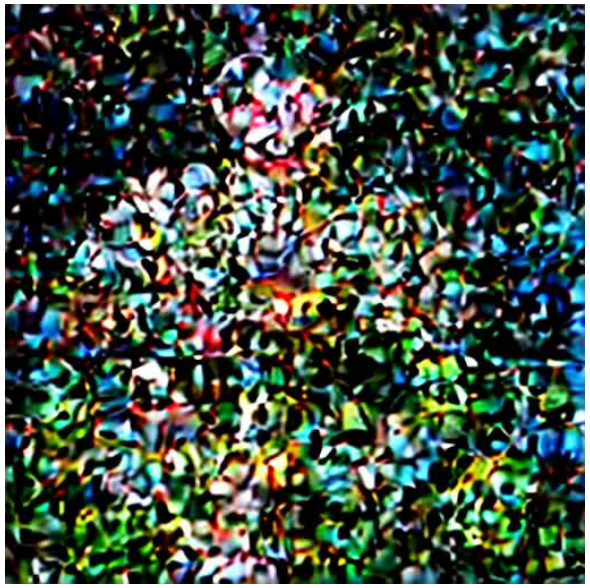


Step 50

a barista handing over a coffee

Stable Diffusion from Seed (4/8th)

Step 0



Step 40

Step 80

a barista handing over a coffee

Exercise

- Inspect sample code for Stable Diffusion
  https://colab.research.google.com/drive/1dlgggNa5Mz8sEAGU0wFCHhGLFooW_pf1
- Stable diffusion uses
  - CLIP text encoders
  - CLIP image encoders
  - An auto-encoder for dimensionality reduction
  - A U-Net to estimate the noise that needs to be removed
- **Create a data flow diagram of stable diffusion**

Stable Diffusion

3x512x512

Text prompt

”A barista handing over a coffee”

CLIP Tokenizer

CLIP Text Encoder

“”

CLIP Tokenizer

CLIP Text Encoder

Concatenation

key

value

query

Cross-Attention

U-Net

-

Decoder

*

noise variance

timestep

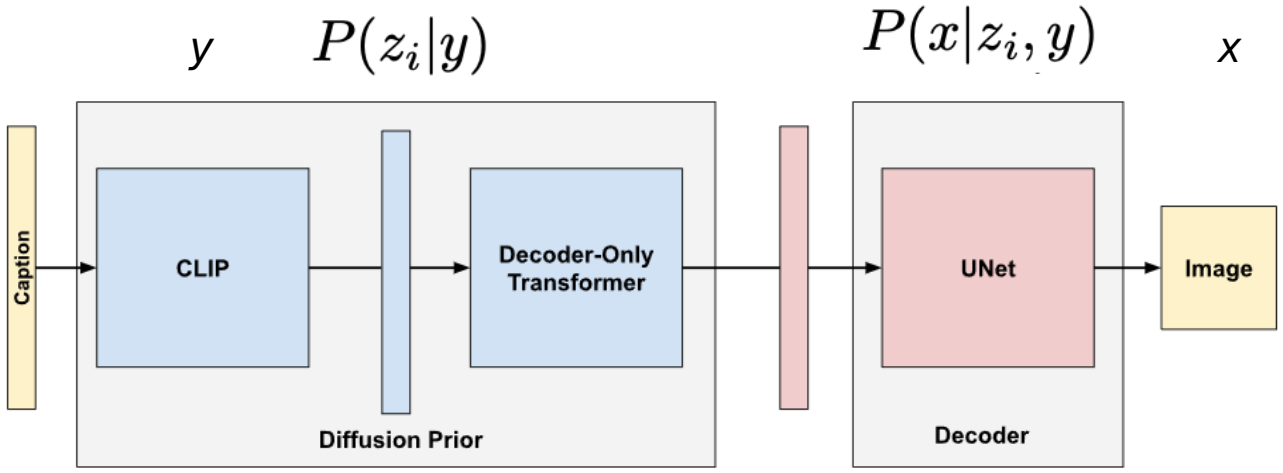
Scheduler

4x64x64

Conditioning Diffusion During Inference

Dall-E 2

- Idea: unCLIP an image embedding -> turn it back into a picture
- First step: turn CLIP text embedding into image embedding (“prior”)
- Use U-Net decoder
- Builds up on Glide



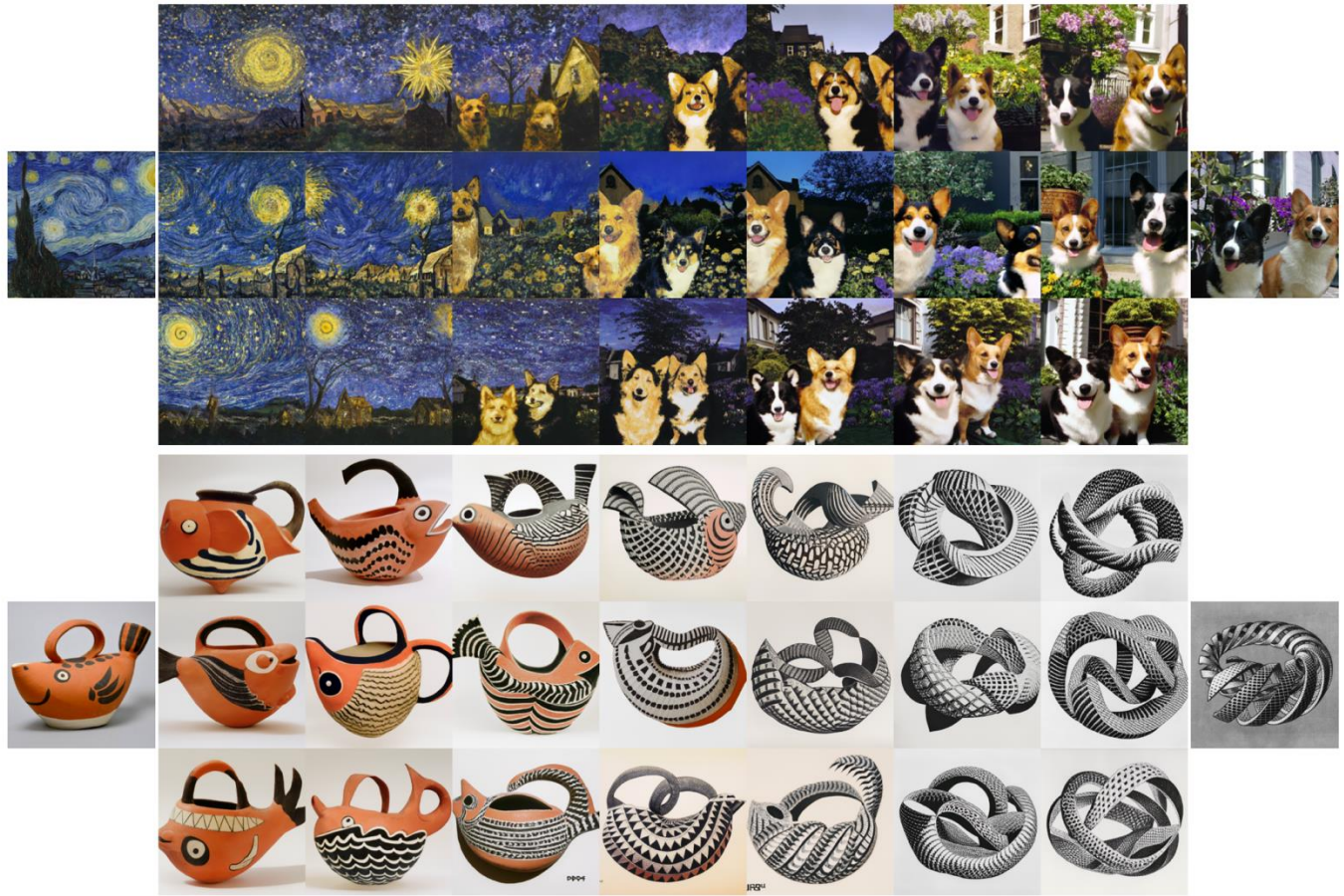
© Matthew Nguyen

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C. and Chen, M., 2022. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, *1*(2), p.3.

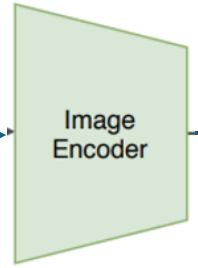
Prior is trained on PCA of Image Embeddings

Figure 7: Visualization of reconstructions of CLIP latents from progressively more PCA dimensions (20, 30, 40, 80, 120, 160, 200, 320 dimensions), with the original source image on the far right. The lower dimensions preserve coarse-grained semantic information, whereas the higher dimensions encode finer-grained details about the exact form of the objects in the scene.

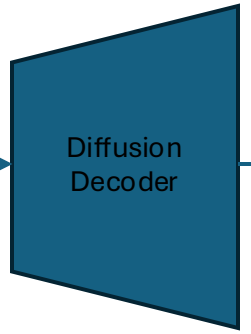
Blending image embeddings



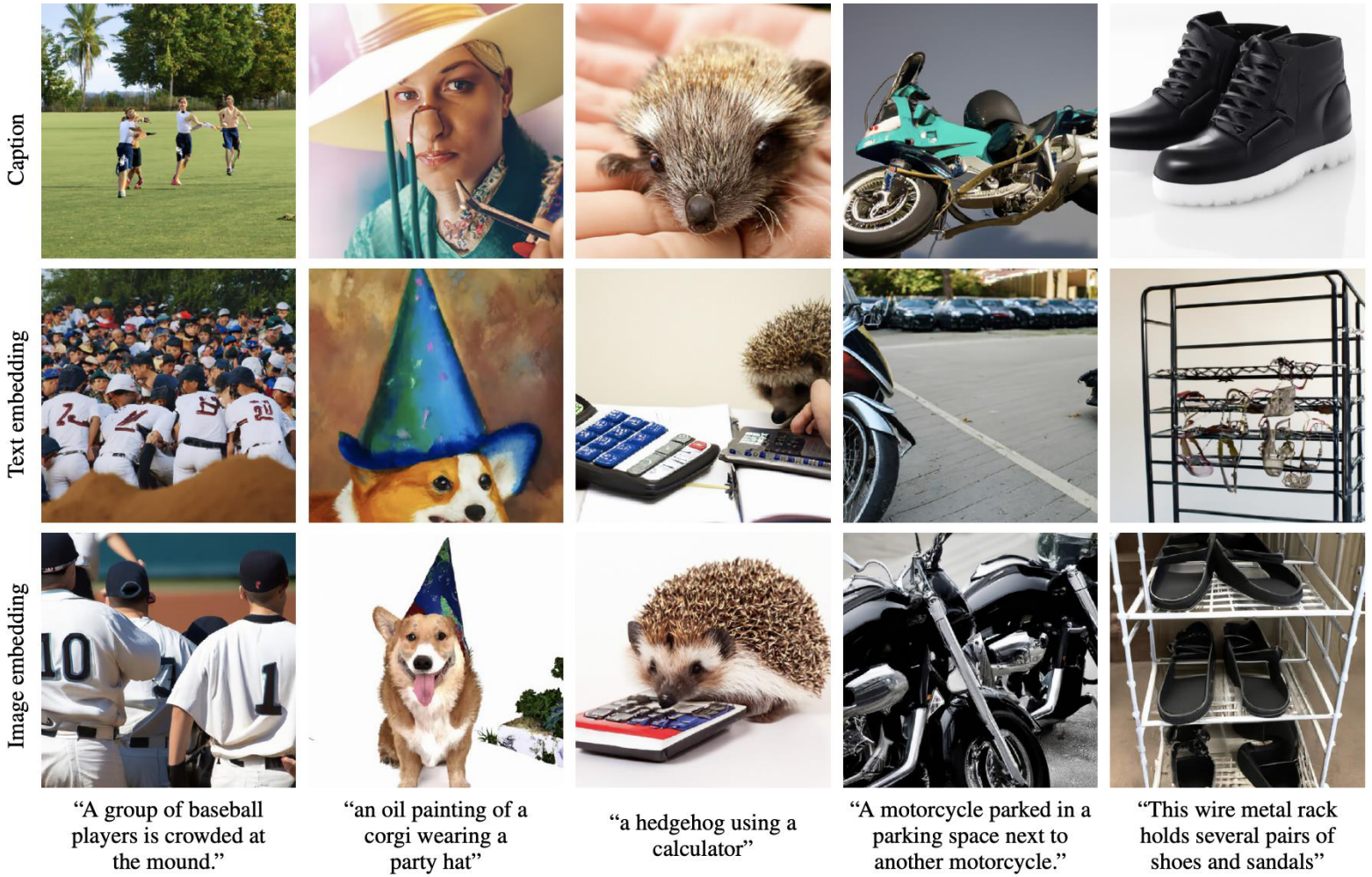
Encoder-Decoder



CLIP

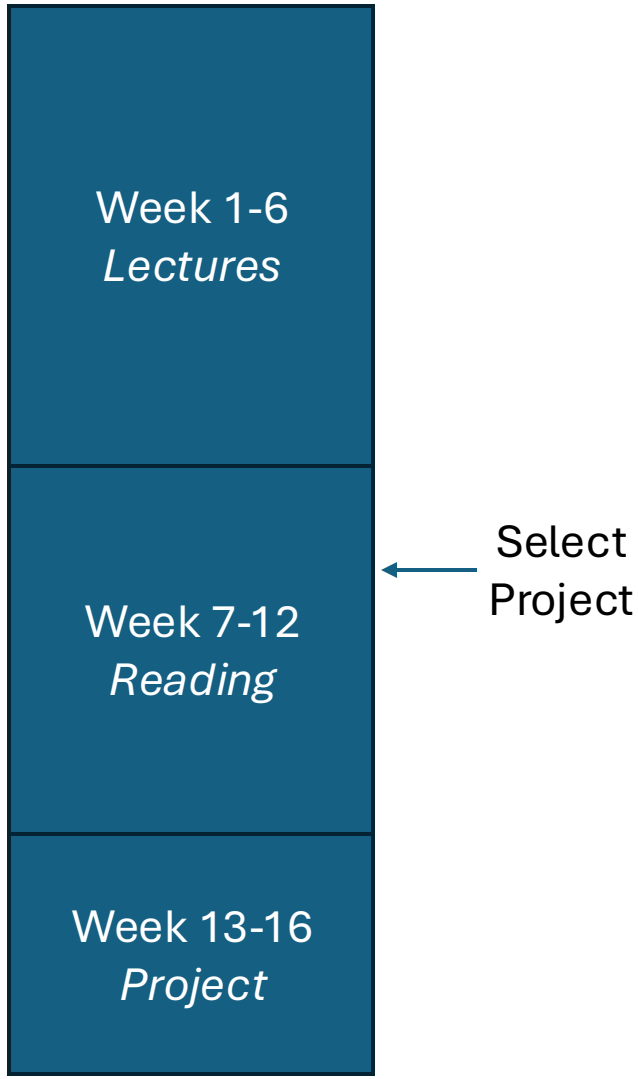


Text vs Image Embeddings



Next task

- Select a paper
  Diffusion, Octo, OpenVLA, TinyVLA, CoRL Proceedings, ….
- Summarize it in a blog article
- Present in class
- Three papers a week, five weeks, starting next week



Paper Line-up

- Monday, October 7th, 2024
  - Paper 1: Yutong, Diffusion Policy Shuran Song
  - Paper 2: Jay, OpenVLA
  - Paper 3: Max Conway, TinyVLA
- Monday, October 14th, 2024
  - Paper 4: Aritra, **TBD**
  - Paper 5: Xuji, **TBD**
  - Paper 6: Stefan, **TBD**
- Monday, October 21th, 2024
  - Paper 7: Himanshu, **TBD**
  - Paper 8: James, **TBD**
  - Paper 9: William, **TBD**
- Monday, October 28th, 2024
  - Paper 10: Carson, **TBD**
  - Paper 11: Peter, **TBD**
  - Paper 12: Naren, **TBD**
- Monday, November 4th, 2024
  - Paper 13: Andy, Swin Transfomers
  - Paper 14: Lekai, Model Merge on LLM
  - Paper 15: -

Notes

- Diffusion has spatial inductive bias that leads to better quality data with lesser training -> same trend in robotics (TinyVLA vs. OpenVLA e.g.)
- Latent-space reduction simplifies both training and generation, what are appropriate latent spaces in robotics?
  - Use an auto-encoder to compress all the RT-X trajectories (done?)
  - Convert Cartesian trajectories into joint-space and use forward kinematics
- U-Nets are well suited to predict noise that has been added to an image because they can map one image to another. What is the best model to deal with trajectories? DVAEs, transformers, …