UNSW
THE UNIVERSITY OF NEW SOUTH WALES

Generative Artificial Intelligence

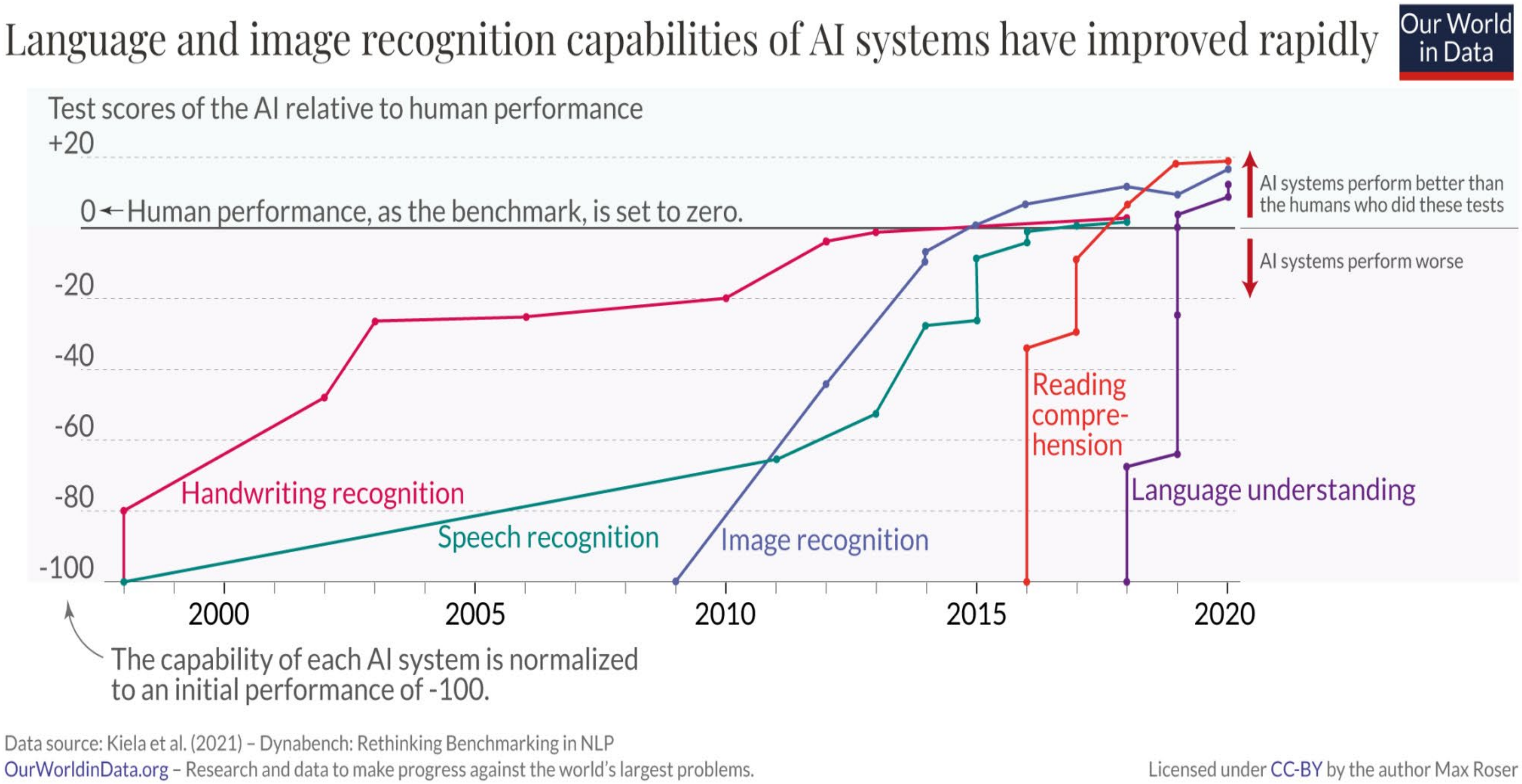
Never Stand Still | Faculty of Engineering | COMP9444 10a

Sonit Singh
School of Computer Science and Engineering
Faculty of Engineering
The University of New South Wales, Sydney, Australia
sonit.singh@unsw.edu.au

Key-ideas covered throughout the course

- Convolutional Neural Networks (CNNs)
  - Building blocks of a typical CNN architecture
  - LeNet, AlexNet, VGG-Net, ResNet, GoogLeNet, DenseNet, …
- Recurrent Neural Networks (RNNs)
  - Vanilla RNNs
  - Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU)
  - Transformer model
  - Bidirectional Encoder Representations from Transformers (BERT)
- Reinforcement Learning
  - Value function learning (TD-Learning, Q-Learning)
  - Policy learning
  - Actor-Critic
- Autoencoders, Variational Autoencoder (VAE)
- Generative Adversarial Networks (GANs)

The AI context



Source: Our World in Data https://ourworldindata.org

What is Generative Artificial Intelligence?

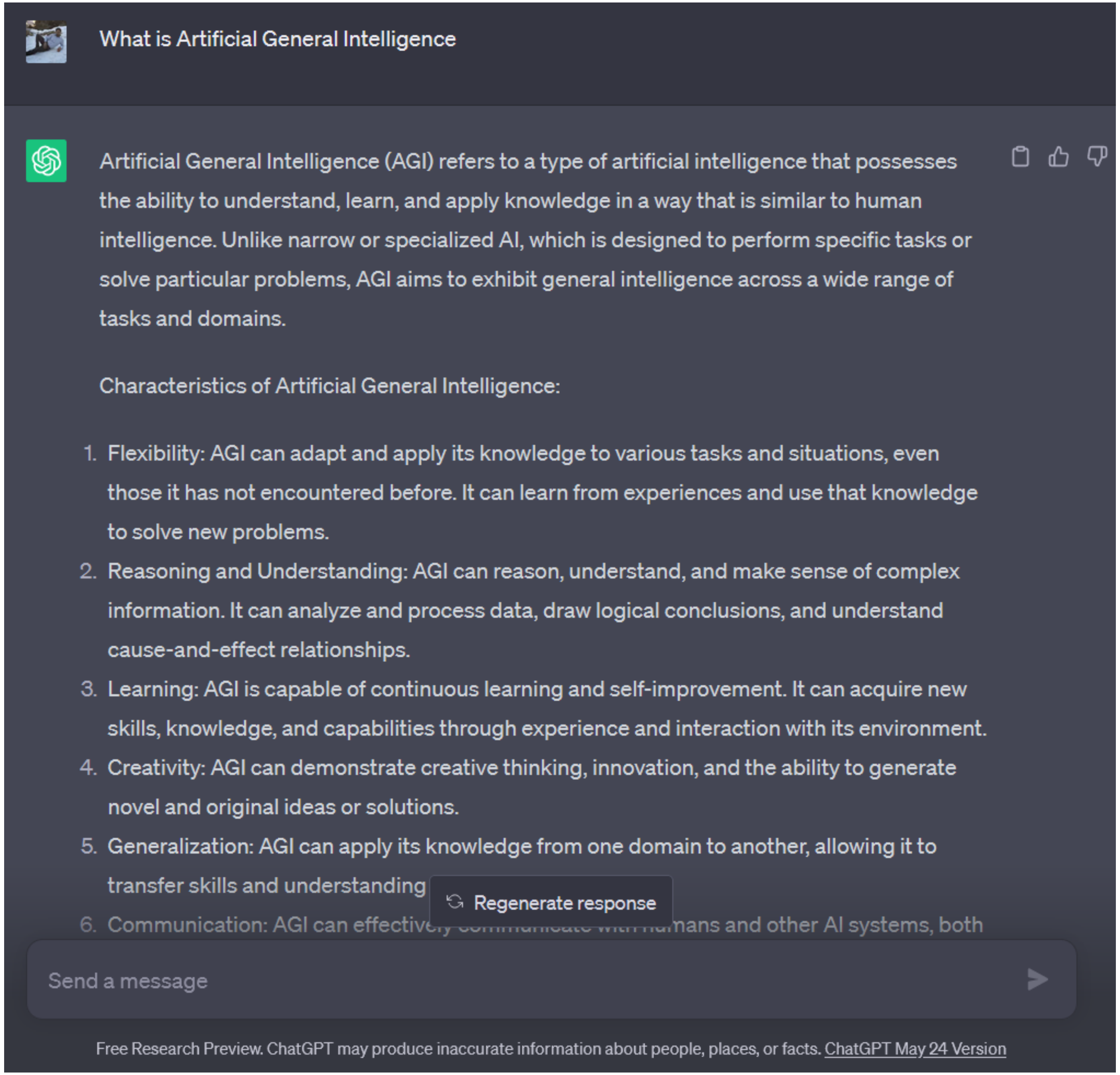
- Set of algorithms that can generate:
  - Images (GANs)
  - Text (Chat-GPT)
  - Audio (OpenAI Whisper)
  - Videos (Synthesia)
  - Code (GitHub CoPilot)
  - Simulations (CARLA – Open-source simulator for autonomous driving research)

- Guided by:
  - Training sample
  - User specifications

- Has huge potential to drastically change the way we approach content creation.

Generative AI tools

- Chat-GPT



Credit: https://chat.openai.com/

Generative AI tools

- Stable Diffusion

stability.ai

Stable Diffusion XL

Create and inspire using the worlds fastest growing open source AI platform.

With Stable Diffusion XL, you can create descriptive images with shorter prompts and generate words within images. The model is a significant advancement in image generation capabilities, offering enhanced image composition and face generation that results in stunning visuals and realistic aesthetics.

Stable Diffusion XL is currently in beta on DreamStudio and other leading imaging applications. Like all of Stability AI's foundation models, Stable Diffusion XL will be released as open source for optimal accessibility in the near future.

Prompt: Glimpses of a herd of wild elephants crossing a savanna

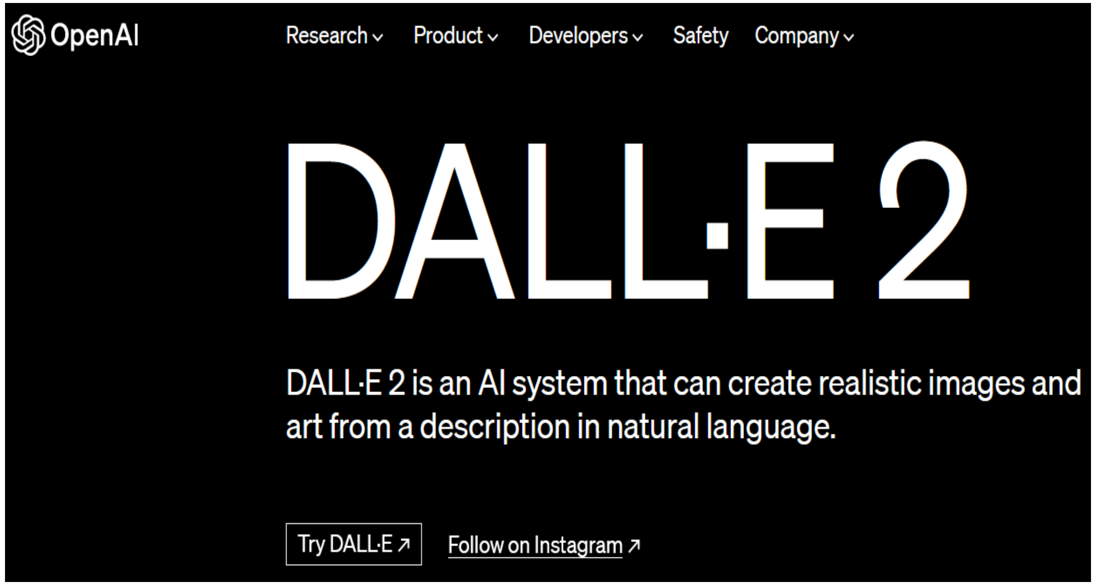
Prompt: Vintage hot rod with custom flame paint job

Prompt: Ancient, mysterious temple in a mountain range, surrounded by misty clouds and tall peaks

Prompt: Beautiful waterfall in a lush jungle, with sunlight shining through the trees

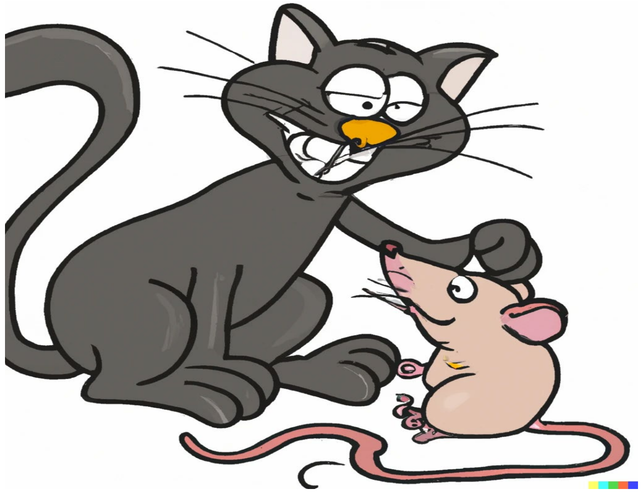
Generative AI tools

- DALLE-2



Prompt: A hand-drawn sailboat circled by birds on the sea at sunrise

Prompt: A photograph of a sunflower with sunglasses on in the middle of the flower in a field on a bright sunny day



Prompt: A cartoon of a cat catching a mouse

Prompt: 3D render of a cute tropical fish in an aquarium on a dark blue background, digital art

Generative AI tools

- MusicLM

ChatGPT

- It is a powerful text-generating dialogue system that can generate humanlike responses to inputs from users.
- Based on Generative Pre-trained Transformer (GPT) architecture
- It is trained on vast data from the internet
- It can accomplish a variety of NLP tasks such as translation, answering questions, sentence completion, etc.

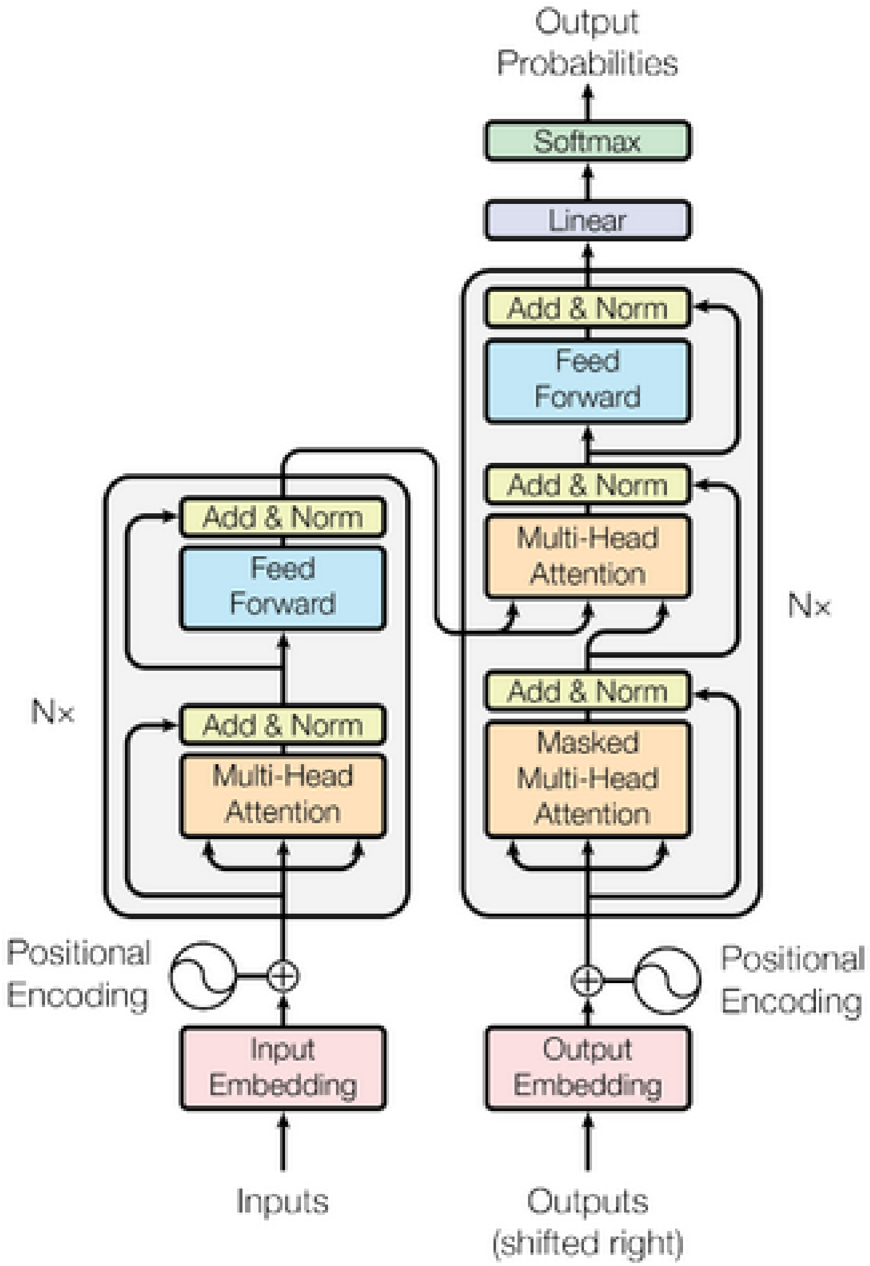
- Let's check GPT-4 capabilities

  https://openai.com/gpt-4

Source: https://openai.com/gpt-4

Transformers

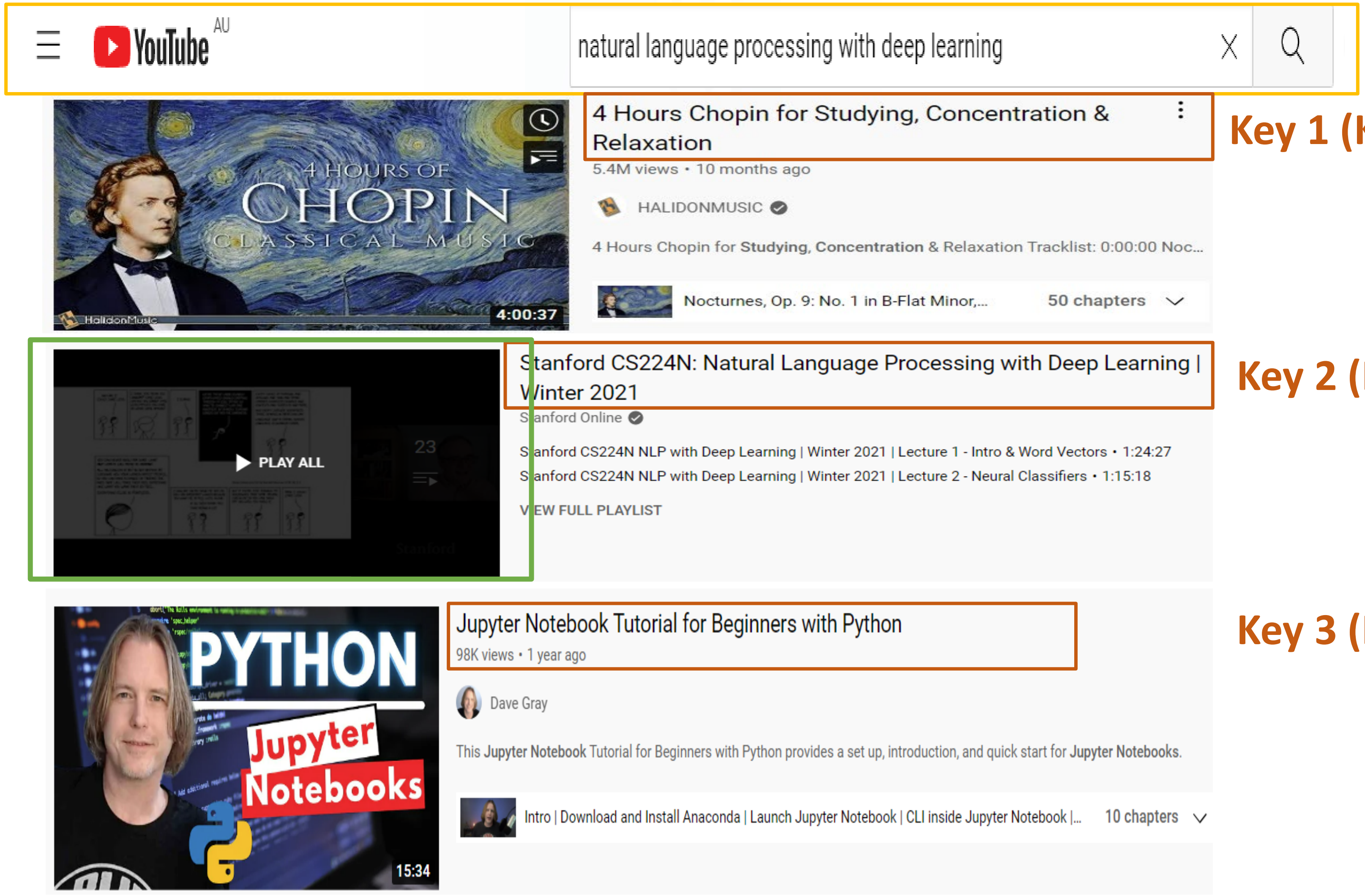
- Attention is All You Need
- Novel architecture relies entirely on **self-attention** to compute representations of its input and output without using sequential RNNs or convolutions.
- Aim is to solve seq2seq tasks while handling long-range dependencies



"Griezmann's announcement comes as a bit of a shock. After enduring the drama surrounding his potential last summer, many thought he was committed to Atletico for more than a year, but the Frenchman seems to have changed his mind."

Self-attention in transformers

- Understanding self-attention with Search
  - Query (Q)
  - Key (K)
  - Value (V)



Query (Q)

Key 1 (K1)

Value (V)

Key 2 (K2)

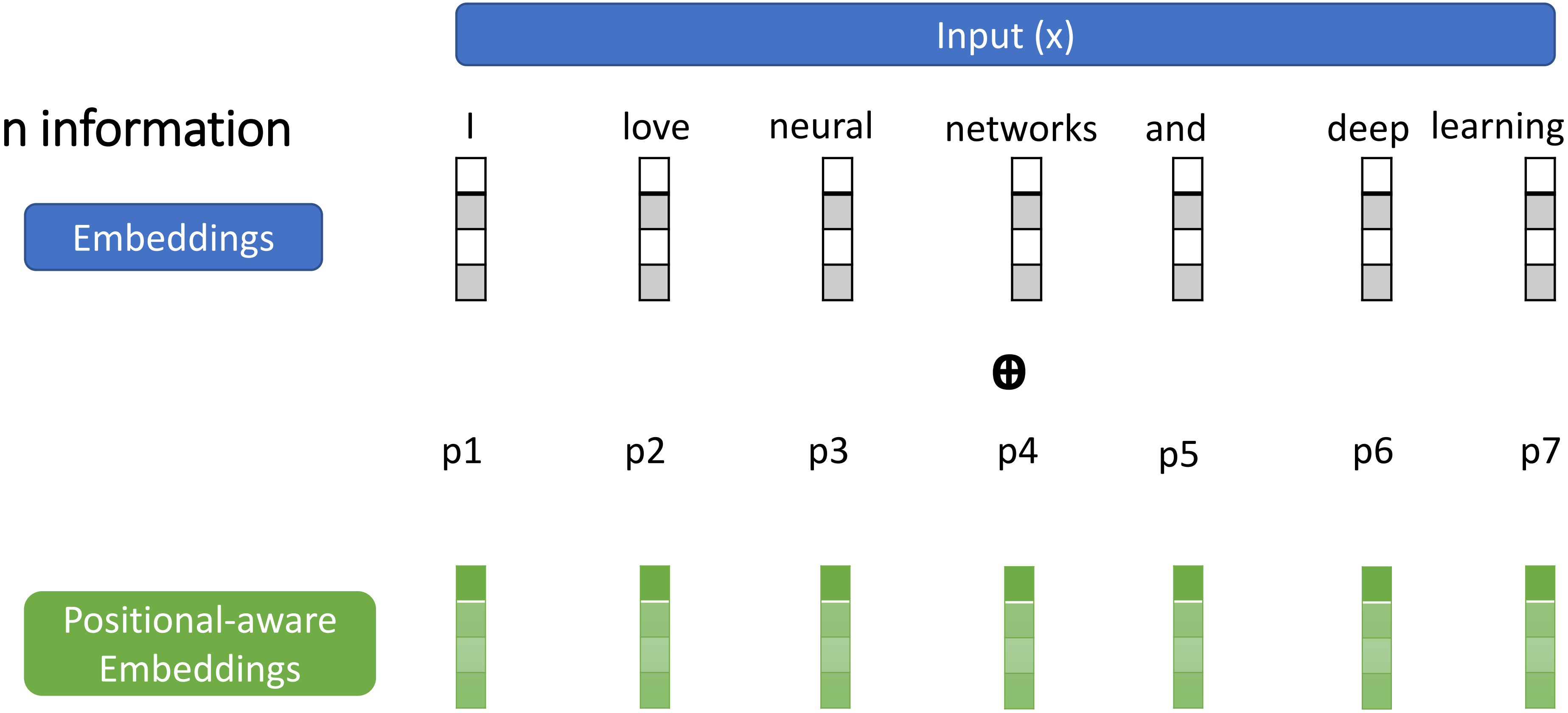
Key 3 (K3)

Self-attention in transformers

Self-attention:

Identify and attend to most important features in input

Step 1: Encode position information



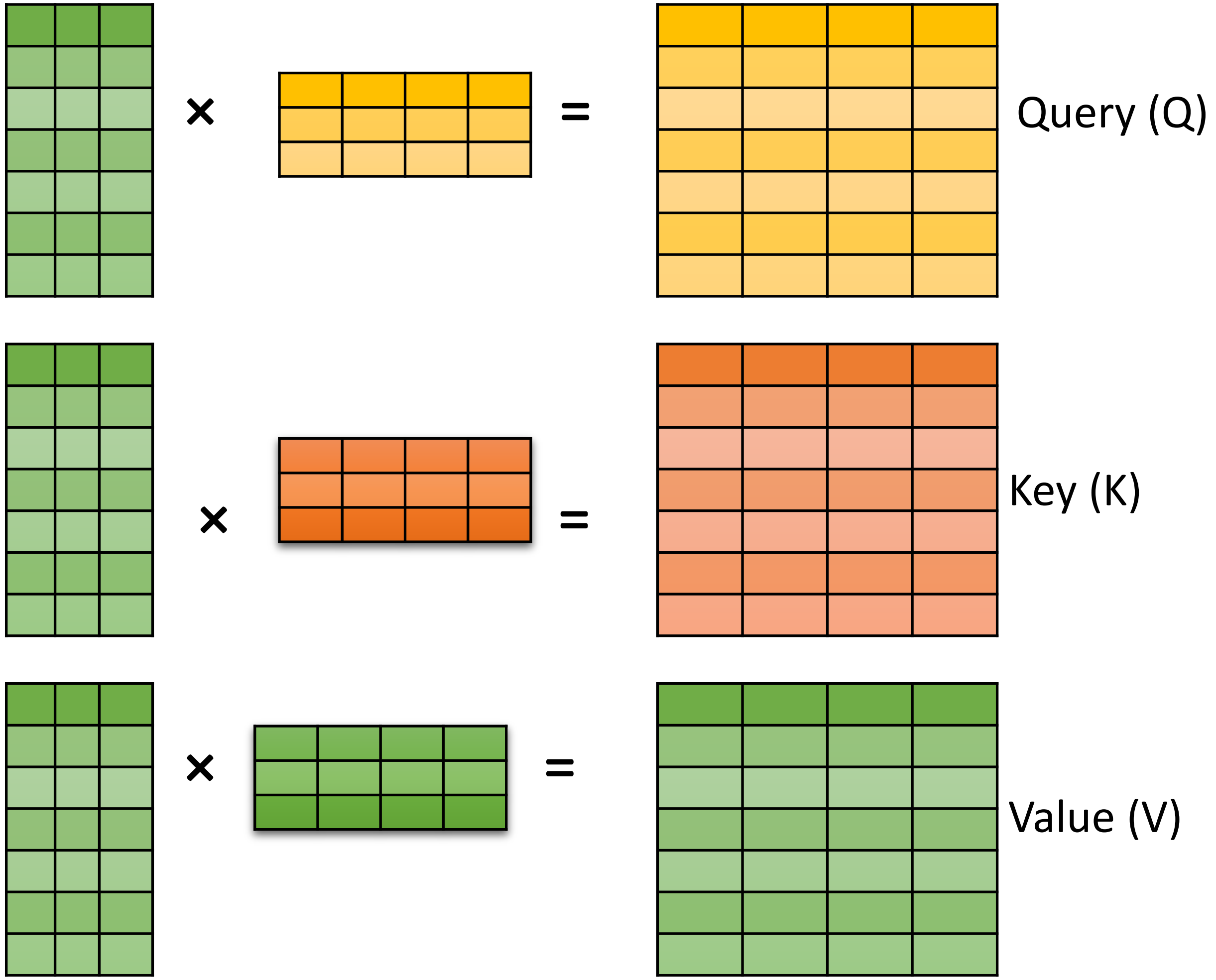
Self-attention in transformers

Self-attention:

Identify and attend to most important features in input

Step 1: Encode position information

Step 2: Extract query (Q), key (K), and value (V)



Self-attention in transformers

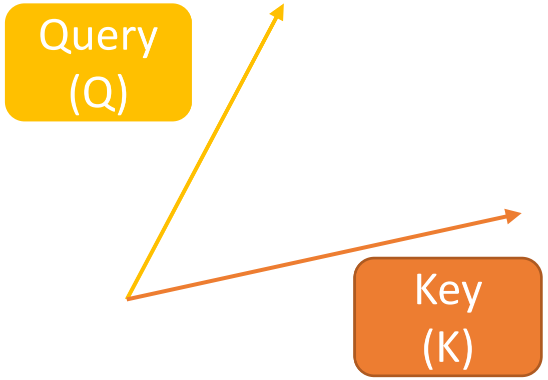
Self-attention:

Identify and attend to most important features in input

Step 1: Encode position information

Step 2: Extract query (Q), key (K), and value (V)

Step 3: Compute attention weighting



$$\frac{Q.\,K^{T}}{\text{Scaling}}$$

Self-attention in transformers

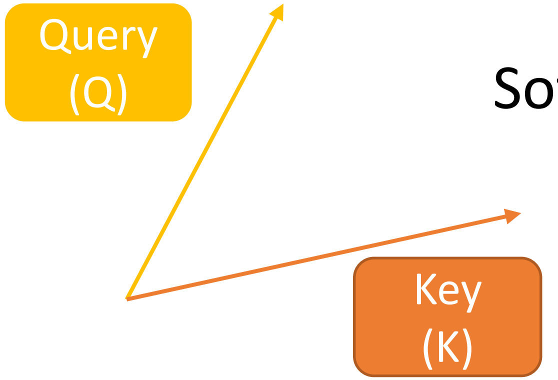
Self-attention:

Identify and attend to most important features in input

Step 1: Encode position information

Step 2: Extract query (Q), key (K), and value (V)

Step 3: Compute attention weighting



$$\text{Softmax}\left\{\frac{Q.\,K^{T}}{\text{Scaling}}\right\}$$

Self-attention in transformers

Self-attention:

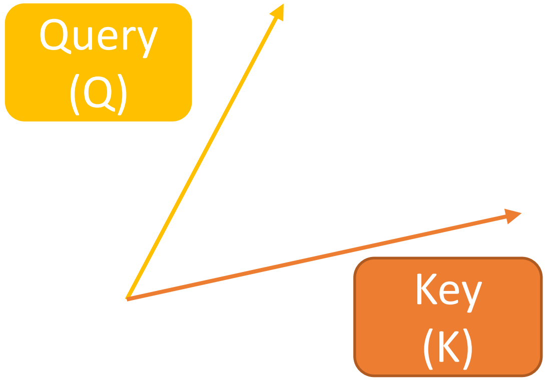
Identify and attend to most important features in input

Step 1: Encode position information

Step 2: Extract query (Q), key (K), and value (V)

Step 3: Compute attention weighting

Step 4: Extract features with high attention



$$\text{Softmax}\left\{\frac{Q.\ K^T}{\text{Scaling}}\right\} \times V = A\,(Q, K, V)$$

Self-attention in transformers

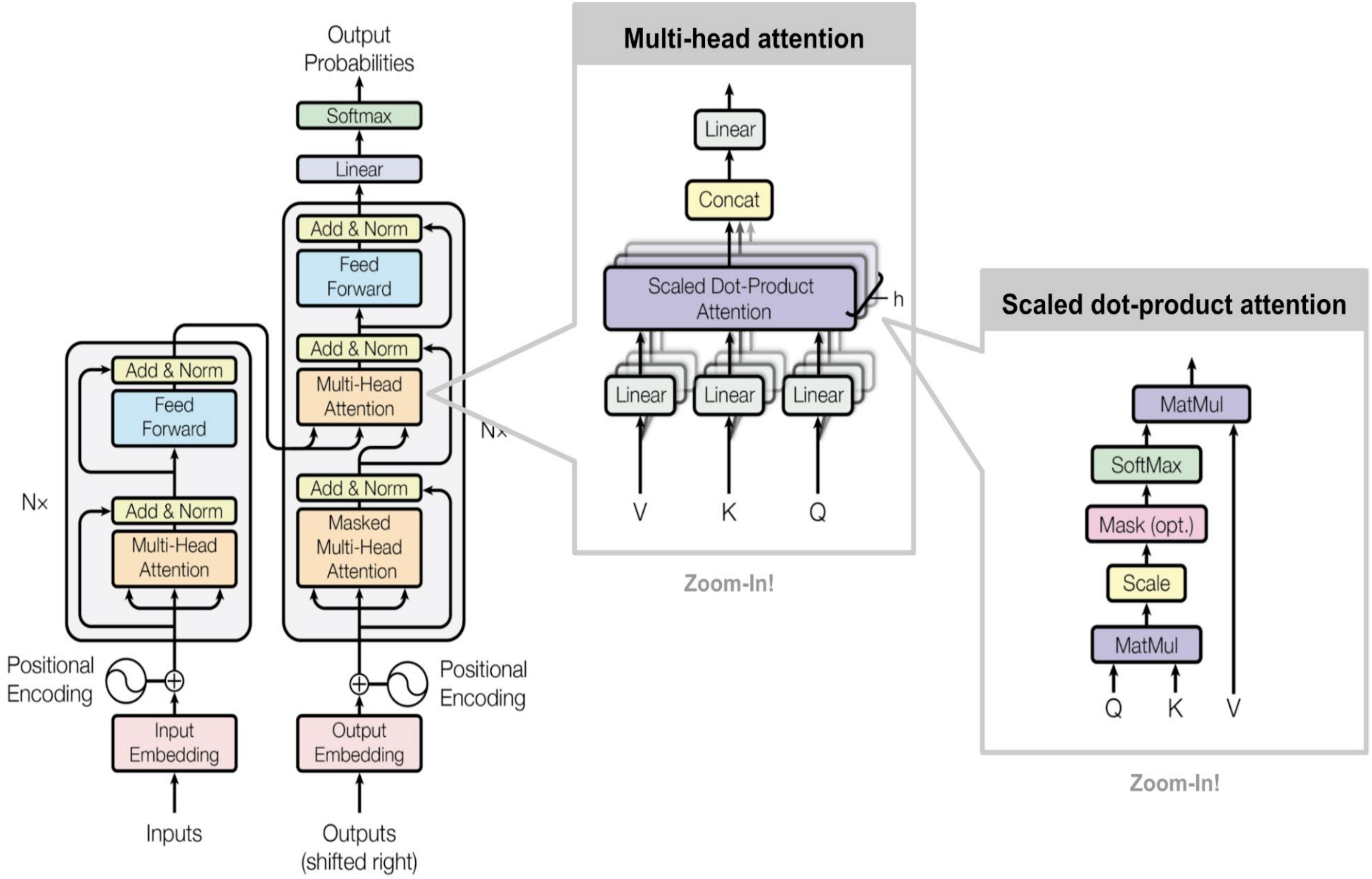
Self-attention:

Step 1: Encode position information

Step 2: Extract query (Q), key (K), and value (V

Step 3: Compute attention weighting

Step 4: Extract features with high attention

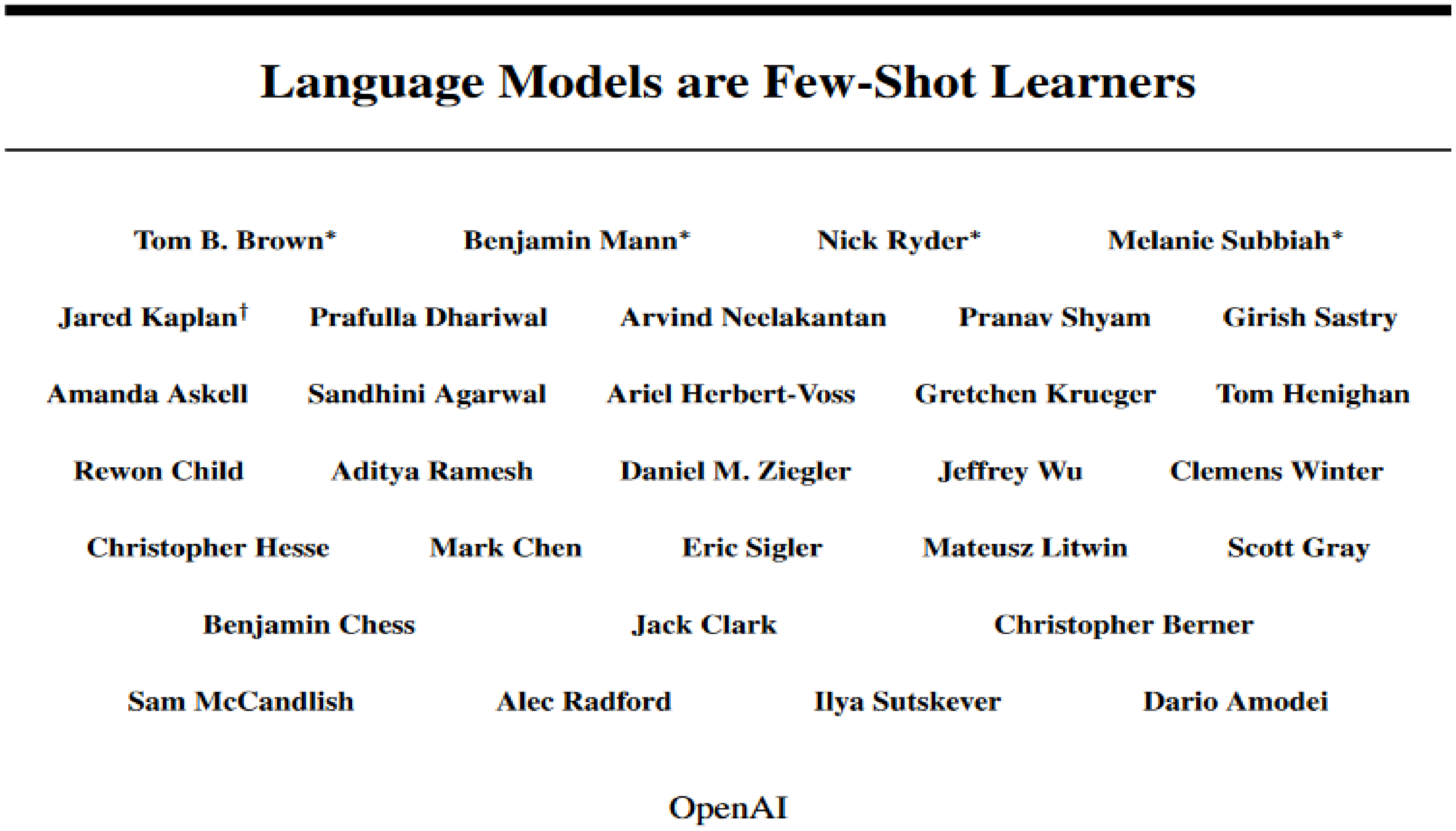


GPT-3

- Key idea: Improve task-agnostic few-shot performance
- Evaluation on various NLP tasks under few-shot learning, one-shot learning, and zero-shot learning demonstrates GPT-3 promising results
- Technical details:
  - An autoregressive language model
  - GPT-3 has 96 layers with each layer having 96 attention heads
  - trained on datasets with 500 billion tokens
  - Word embedding size of 12888
  - Context window size is of 2048 tokens
  - Uses alternating dense and locally banded sparse attention patterns
- Compute:
  - Trained on more than 576 GB of text data including common crawl, Books, and Wikipedia
  - About 175 billion parameters
  - costs OpenAI around $4.6 million

Credit: Brown et al., (2020). Language models are few-shot learners.
Priya Shree. Medium: The journey of OpenAI GPT models. https://medium.com/walmartglobaltech/the-journey-of-open-ai-gpt-models-32d95b7b7fb2

GPT-3



Credit: Brown et al., (2020). Language models are few-shot learners.

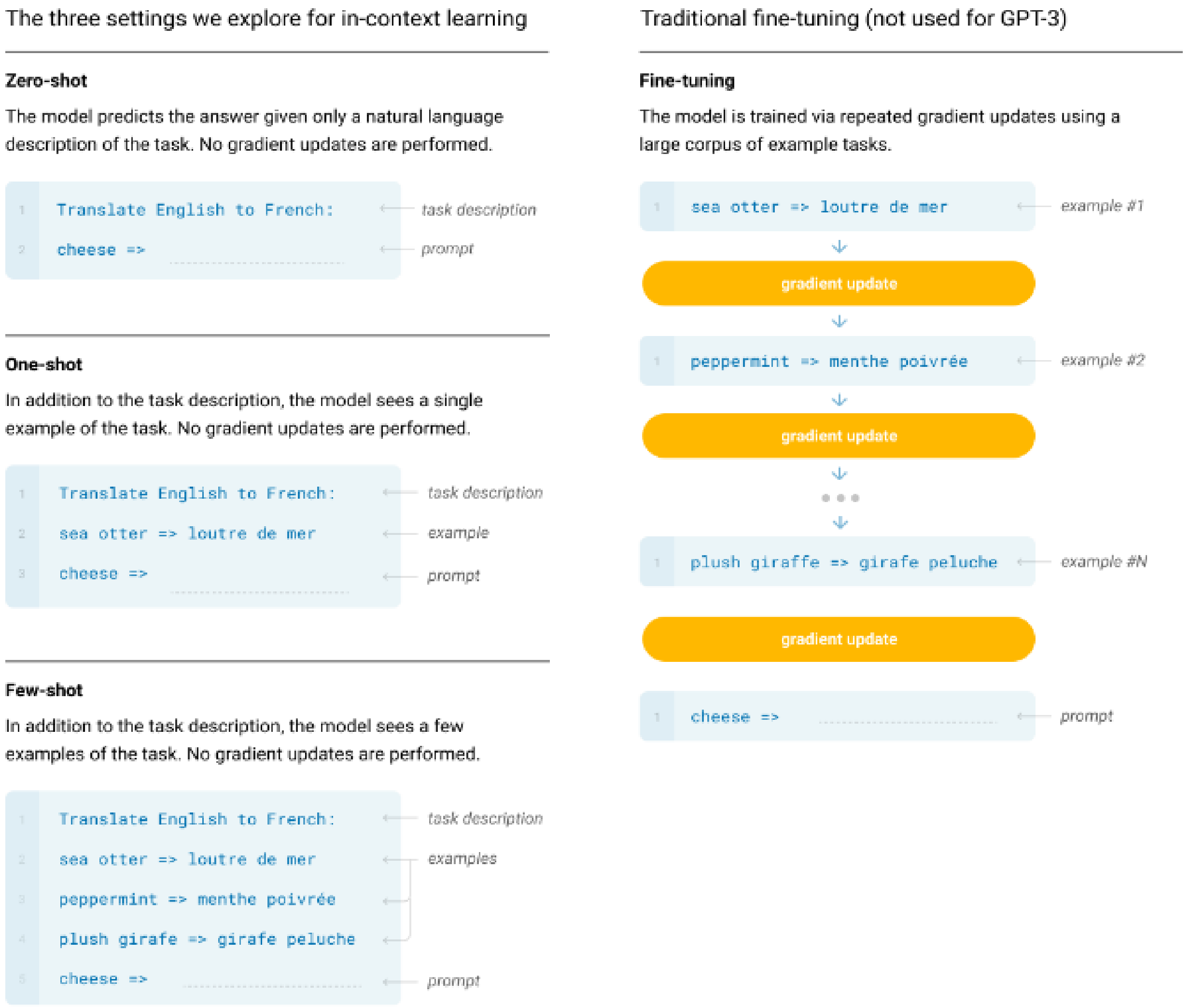
GPT-3

- Fine-Tuning: Updating the weights of a pre-trained model by training on a supervised dataset specific to the desired task.
- Few-Shot Learning: Refers to the setting where the model is given a few demonstrations of the task at inference time as conditioning, but no weight updates are allowed.
- One-Shot Learning: In this setting, only one demonstration is allowed
- Zero-Shot Learning: In this setting, a model can learn to recognize things that it hasn’t explicitly seen before during training. The idea behind is how humans can naturally find similarities between data classes, in the same way, training the machines to identify.

Credit: Brown et al., (2020). Language models are few-shot learners.

GPT-3

Zero-shot, one-shot and few-shot, contrasted with traditional fine-tuning



Credit: Brown et al., (2020). Language models are few-shot learners.

GPT-3

- Sizes, architectures, and learning hyper-parameters of GPT-3 models

Model Name	n_{params}	n_{layers}	d_{model}	n_{heads}	d_{head}	Batch Size	Learning Rate
GPT-3 Small	125M	12	768	12	64	0.5M	6.0×10^{-4}
GPT-3 Medium	350M	24	1024	16	64	0.5M	3.0×10^{-4}
GPT-3 Large	760M	24	1536	16	96	0.5M	2.5×10^{-4}
GPT-3 XL	1.3B	24	2048	24	128	1M	2.0×10^{-4}
GPT-3 2.7B	2.7B	32	2560	32	80	1M	1.6×10^{-4}
GPT-3 6.7B	6.7B	32	4096	32	128	2M	1.2×10^{-4}
GPT-3 13B	13.0B	40	5140	40	128	2M	1.0×10^{-4}
GPT-3 175B or "GPT-3"	175.0B	96	12288	96	128	3.2M	0.6×10^{-4}

- Datasets used to train GPT-3

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

Credit: Brown et al., (2020). Language models are few-shot learners.

GPT-3 Tasks

- **Language modelling:** Calculated zero-shot perplexity on the Penn Tree Bank (PTB) dataset
- **LAMBADA:** The LAMBADA dataset tests the modelling of long-range dependencies in text – the model is asked to predict the last word of sentences which require reading a paragraph of context.
- The **StoryCloze 2016 dataset**, which involves selecting the correct ending sentence for five-sentence long stories.
- The **HelloSwag dataset** involves picking the best ending to a story or set of instructions.
- **Closed Book Question Answering**: Measuring GPT-3's ability to answer questions about broad factual knowledge.
- **Translation**: Translation English, French, German, Romanian
- **Reading Comprehension**
- **SuperGLUE** (prominent evaluation framework for research towards language understanding)
- **Natural Language Inference** (NLI)
- **SAT Analogies**
- **Synthetic and Qualitative tasks** (addition, subtraction, ...)
- ... and more ...

Credit: Brown et al., (2020). Language models are few-shot learners.

GPT-3 Results

➢ Results on language modelling, Cloze, and completion Tasks

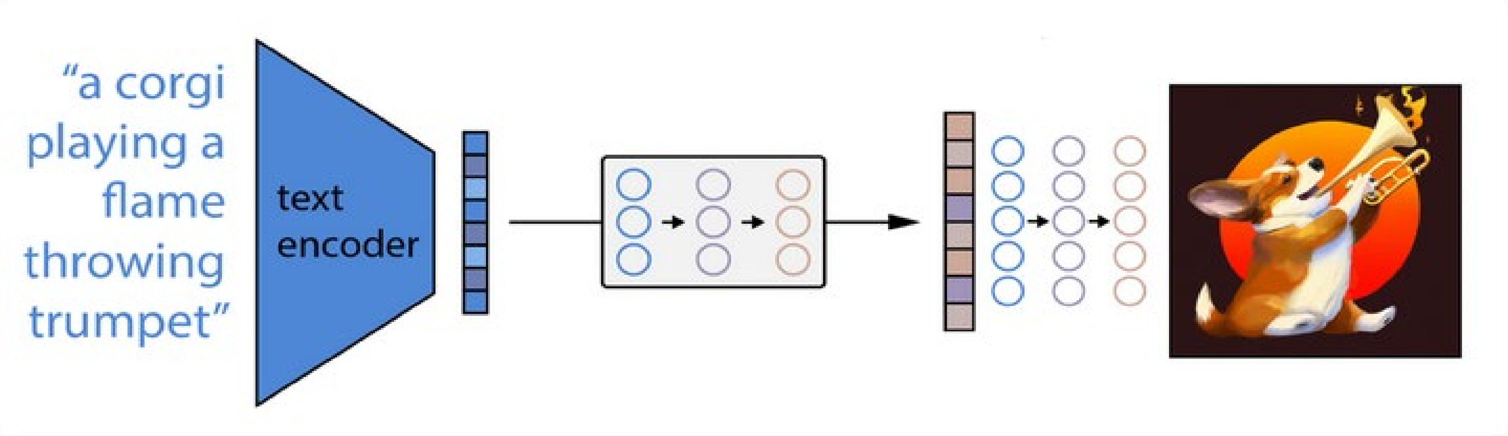
Setting	LAMBADA (acc)	LAMBADA (ppl)	StoryCloze (acc)	HellaSwag (acc)
SOTA	68.0[a]	8.63[b]	**91.8**[c]	**85.6**[d]
GPT-3 Zero-Shot	**76.2**	**3.00**	83.2	78.9
GPT-3 One-Shot	**72.5**	**3.35**	84.7	78.1
GPT-3 Few-Shot	**86.4**	**1.92**	87.7	79.3

➢ Results on three open-domain QA tasks

Setting	NaturalQS	WebQS	TriviaQA
RAG (Fine-tuned, Open-Domain) [LPP+20]	**44.5**	**45.5**	**68.0**
T5-11B+SSM (Fine-tuned, Closed-Book) [RRS20]	36.6	44.7	60.5
T5-11B (Fine-tuned, Closed-Book)	34.5	37.4	50.1
GPT-3 Zero-Shot	14.6	14.4	64.3
GPT-3 One-Shot	23.0	25.3	**68.0**
GPT-3 Few-Shot	29.9	41.5	**71.2**

DALL-E 2

- Capabilities:
  - can generate images from text
  - can insert new features or styles in images to modify them
- Based on **CLIP** (Contrastive Language-Image Pre-training)



a propaganda poster depicting a cat dressed as french emperor napoleon holding a piece of cheese

Vision Transformer (ViT)

- Transformers showed great performance on variety of NLP tasks.
- How to use Transformers (self-attention) for vision tasks?
- Transformer applied to image patches- similar to NLP, but with patches instead of words

Published as a conference paper at ICLR 2021

AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE

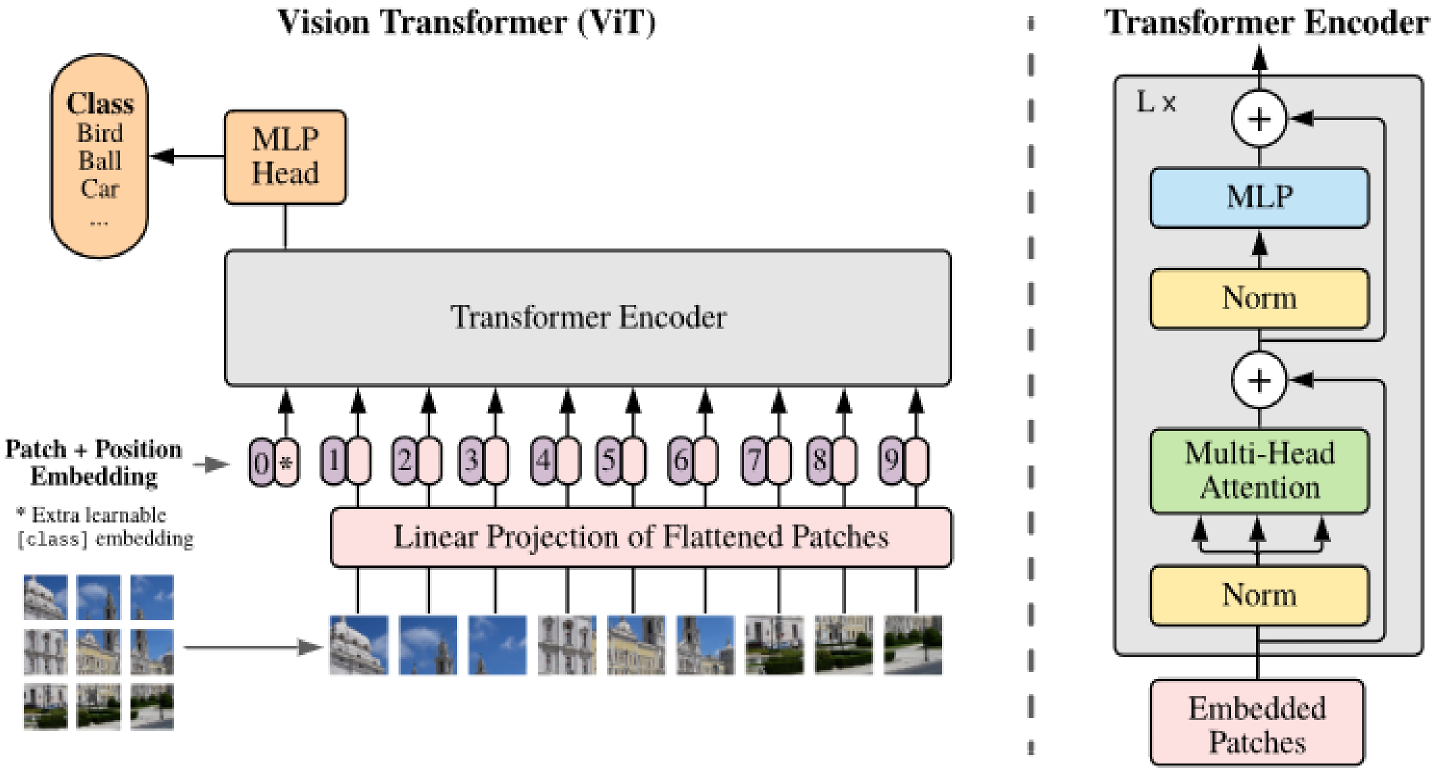
Alexey Dosovitskiy[*,†], **Lucas Beyer**[*], **Alexander Kolesnikov**[*], **Dirk Weissenborn**[*], **Xiaohua Zhai**[*], **Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby**[*,†]
[*]equal technical contribution, [†]equal advising
Google Research, Brain Team
{adosovitskiy, neilhoulsby}@google.com

ABSTRACT

While the Transformer architecture has become the de-facto standard for natural language processing tasks, its applications to computer vision remain limited. In vision, attention is either applied in conjunction with convolutional networks, or used to replace certain components of convolutional networks while keeping their overall structure in place. We show that this reliance on CNNs is not necessary and a pure transformer applied directly to sequences of image patches can perform very well on image classification tasks. When pre-trained on large amounts of data and transferred to multiple mid-sized or small image recognition benchmarks (ImageNet, CIFAR-100, VTAB, etc.), Vision Transformer (ViT) attains excellent results compared to state-of-the-art convolutional networks while requiring substantially fewer computational resources to train.[1]

Vision Transformer (ViT) method

- Split an image into fixed-size patches, linearly embed each of them, add position embeddings, and feed the resulting sequence of vectors to a standard Transformer encoder.



Source: Dosovitskiy et al., An image is worth 16 x 16 words: Transformers for image recognition at scale. ICLR. 2021.

Vision Transformer (ViT) experimental setup

*Approximate number of classes and images

- Pre-training datasets
  - ImageNet ILSVRC-2012: 1000 classes, 1.3 million images
  - ImageNet-21k: 21000 classes, 14 million images
  - JFT: 18000 classes, 303 million images
- Fine-tuning datasets
  - ImageNet- both original validation labels and Reassessed Labels (ReaL)
  - CIFAR-10/100: 60000 images each
  - Oxford-IIIT Pets: 7400 images
  - Oxford Flowers-102: 7100 images
  - VTAB: natural, specialized, structured
- Pre-processing
  - Pre-training: image cropped, random horizontal mirroring, resize to 224 x 224
  - Fine-tuning: images resized to 448 x 448, random crop of 384 x 384
  - Random horizontal flips

Vision Transformer (ViT)

- ViT variants

Model	Layers	Hidden size D	MLP size	Heads	Params
ViT-Base	12	768	3072	12	86M
ViT-Large	24	1024	4096	16	307M
ViT-Huge	32	1280	5120	16	632M

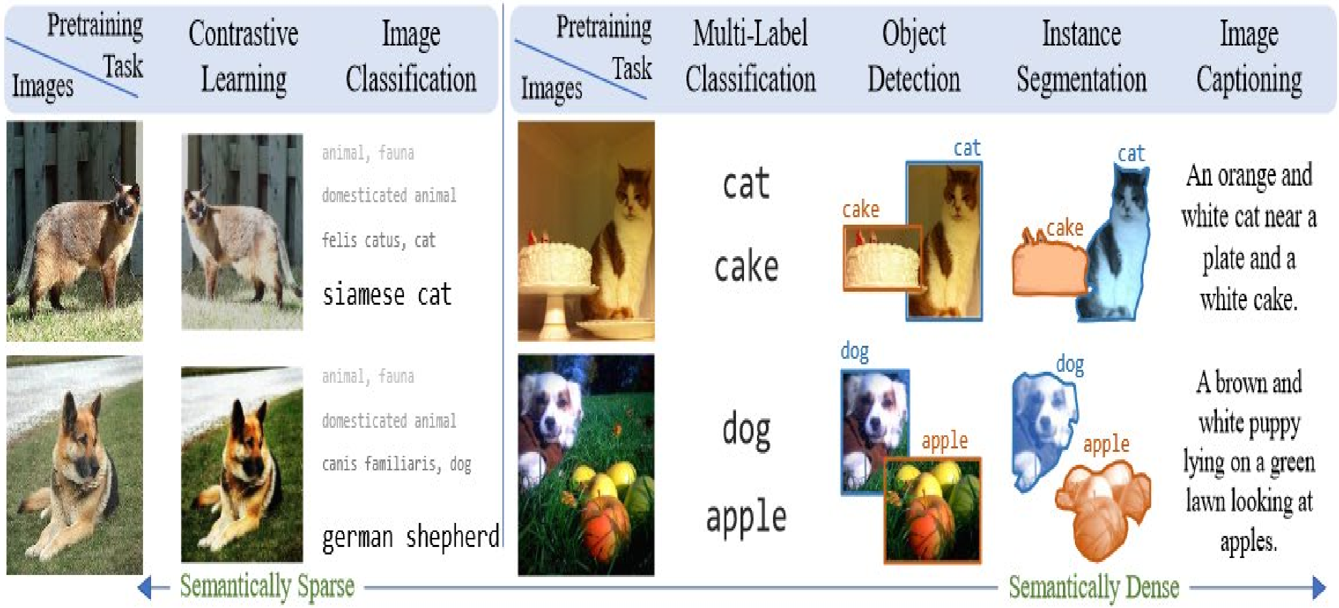
- Results

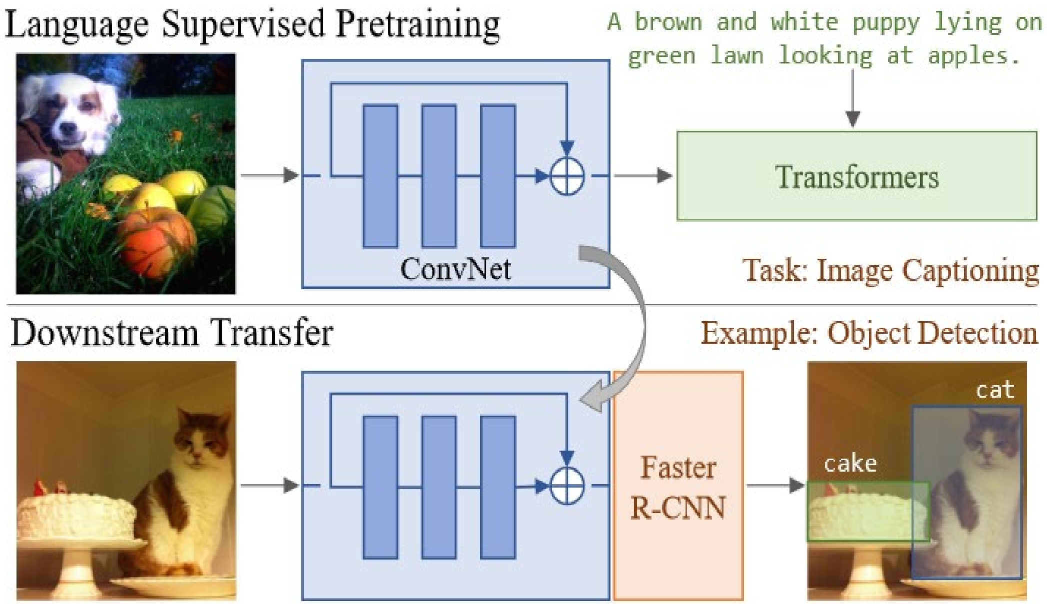
	Ours-JFT (ViT-H/14)	Ours-JFT (ViT-L/16)	Ours-I21k (ViT-L/16)	BiT-L (ResNet152x4)	Noisy Student (EfficientNet-L2)
ImageNet	**88.55** $\pm$ 0.04	87.76 $\pm$ 0.03	85.30 $\pm$ 0.02	87.54 $\pm$ 0.02	88.4/88.5*
ImageNet ReaL	**90.72** $\pm$ 0.05	90.54 $\pm$ 0.03	88.62 $\pm$ 0.05	90.54	90.55
CIFAR-10	**99.50** $\pm$ 0.06	99.42 $\pm$ 0.03	99.15 $\pm$ 0.03	99.37 $\pm$ 0.06	–
CIFAR-100	**94.55** $\pm$ 0.04	93.90 $\pm$ 0.05	93.25 $\pm$ 0.05	93.51 $\pm$ 0.08	–
Oxford-IIIT Pets	**97.56** $\pm$ 0.03	97.32 $\pm$ 0.11	94.67 $\pm$ 0.15	96.62 $\pm$ 0.23	–
Oxford Flowers-102	99.68 $\pm$ 0.02	**99.74** $\pm$ 0.00	99.61 $\pm$ 0.02	99.63 $\pm$ 0.03	–
VTAB (19 tasks)	**77.63** $\pm$ 0.23	76.28 $\pm$ 0.46	72.72 $\pm$ 0.21	76.29 $\pm$ 1.70	–
TPUv3-core-days	2.5k	0.68k	0.23k	9.9k	12.3k

Source: Dosovitskiy et al., An image is worth 16 x 16 words: Transformers for image recognition at scale. ICLR. 2021.

CLIP uses prior work of VirTex and ConVIRT

- VirTex: Learning Visual Representations from Textual Annotations
- Prevailing paradigm: Pretrain a CNN and then transfer the learned features to downstream tasks.
- Using textual features to learn visual features require fewer images than other approaches.

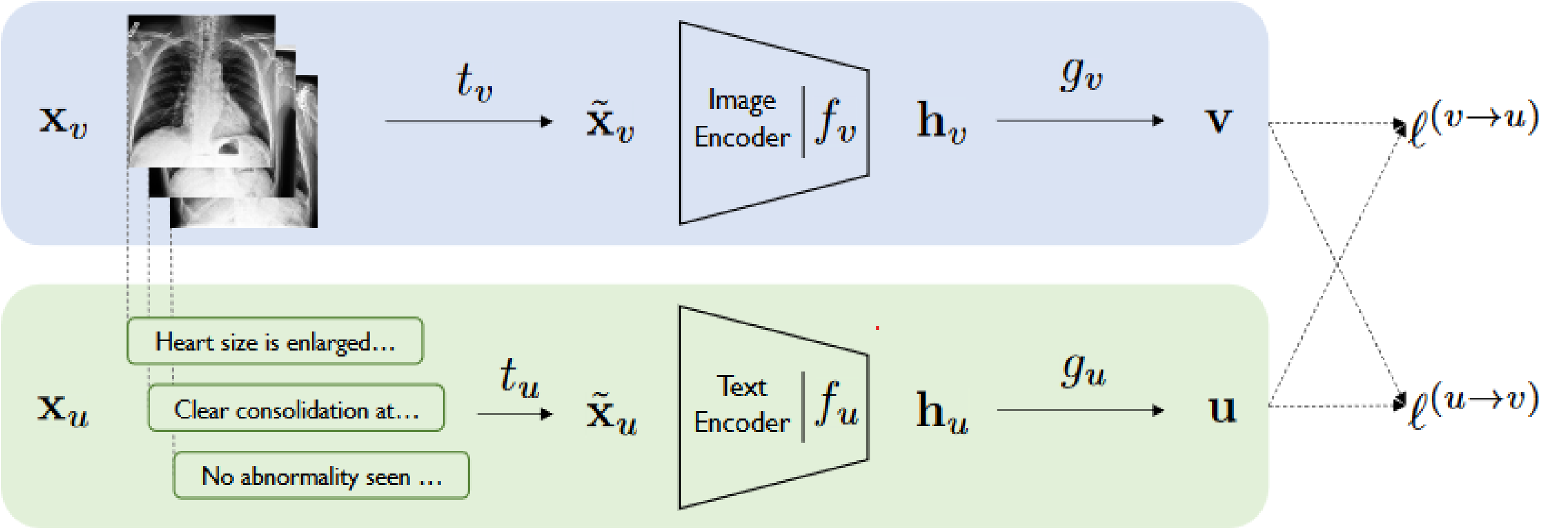




Source: Desai and Johnson. VirTeX: Learning visual representations from textual annotations. 2021.
Zhang et al., Contrastive learning of medical visual representations from paired images and text. 2022.

CLIP uses prior work of VirTex and ConVIRT

- ConVIRT: Contrastive Learning of Medical Visual Representations from paired images and text
- Proposes unsupervised strategy to learn visual representations by exploiting naturally occurring paired descriptive text (both in natural and medical domain)
- Maximizes the agreement between the true image-text representation pairs with bidirectional losses



CLIP (Contrastive Language-Image Pre-training)

- CLIP learns visual concepts from natural language supervision
- Training process:
  1. All images and their associated captions are passed through respective encoders (image and text) to map images and text into n-dimensional vector
  2. Compute cosine similarity for each (image, text) pair
  3. During training, maximize the cosine similarity between correct encoded (image, text) pairs and minimize the cosine similarity between incorrect encoded (image, text) pairs.

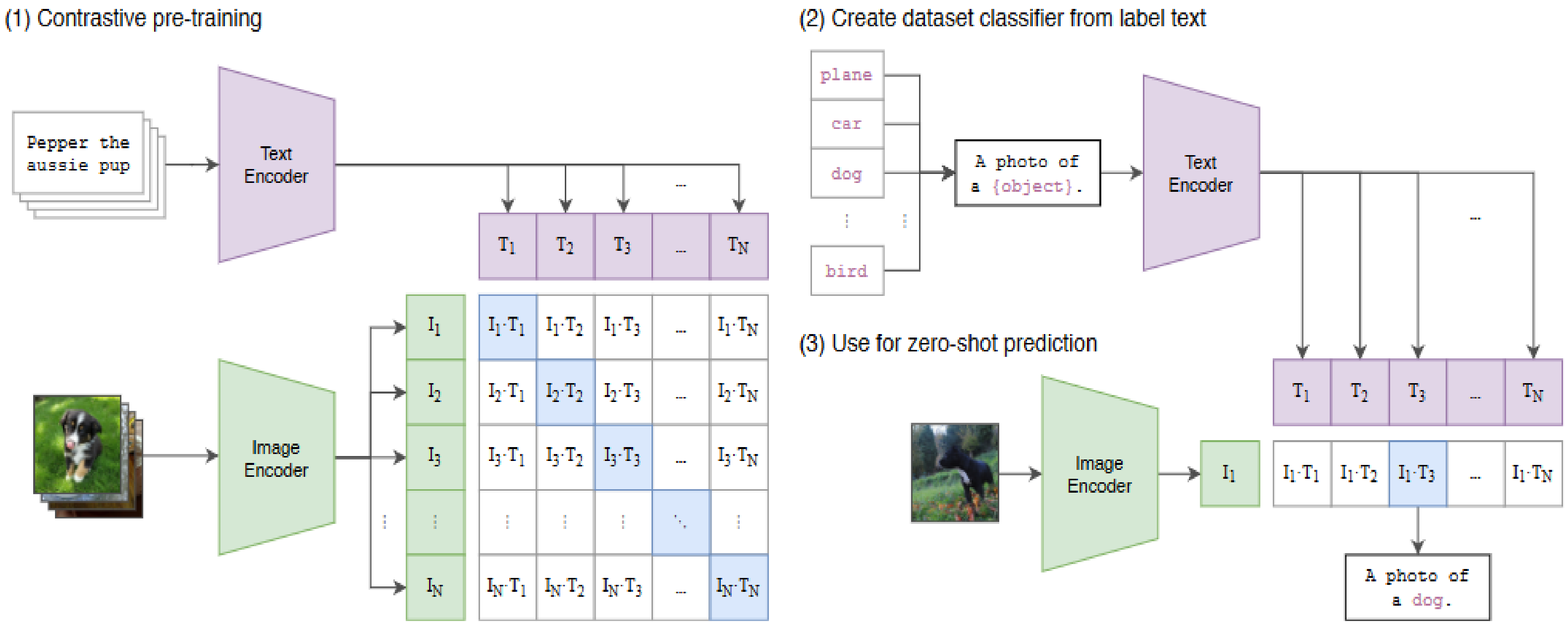
- Let's visualize the training process

https://www.assemblyai.com/blog/how-dall-e-2-actually-works/

Source: Radford et al., Learning Transferable Visual Models from Natural Language Supervision

CLIP model

- CLIP jointly trains an image encoder and a text encoder to predict the correct pairings of a batch of (image, text) training examples.
- At test time the learned text encoder synthesizes a zero-shot linear classifier by embedding the names or descriptions of the target dataset's classes.

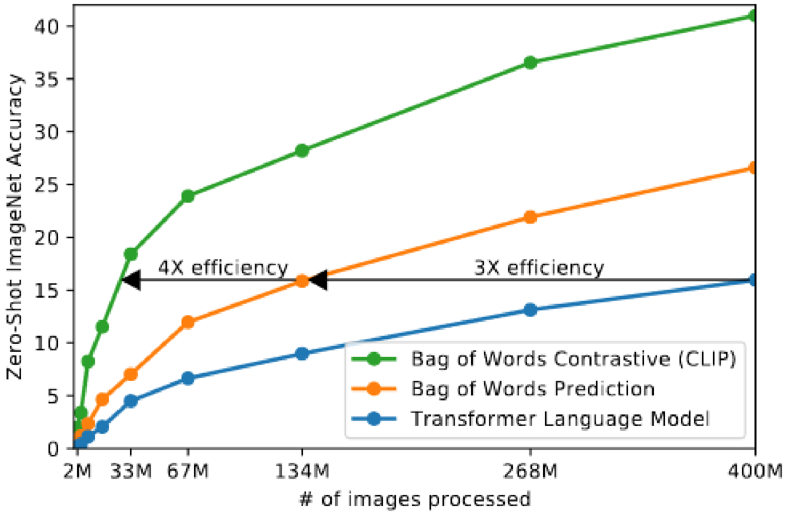


Pseudocode of core implementation of CLIP

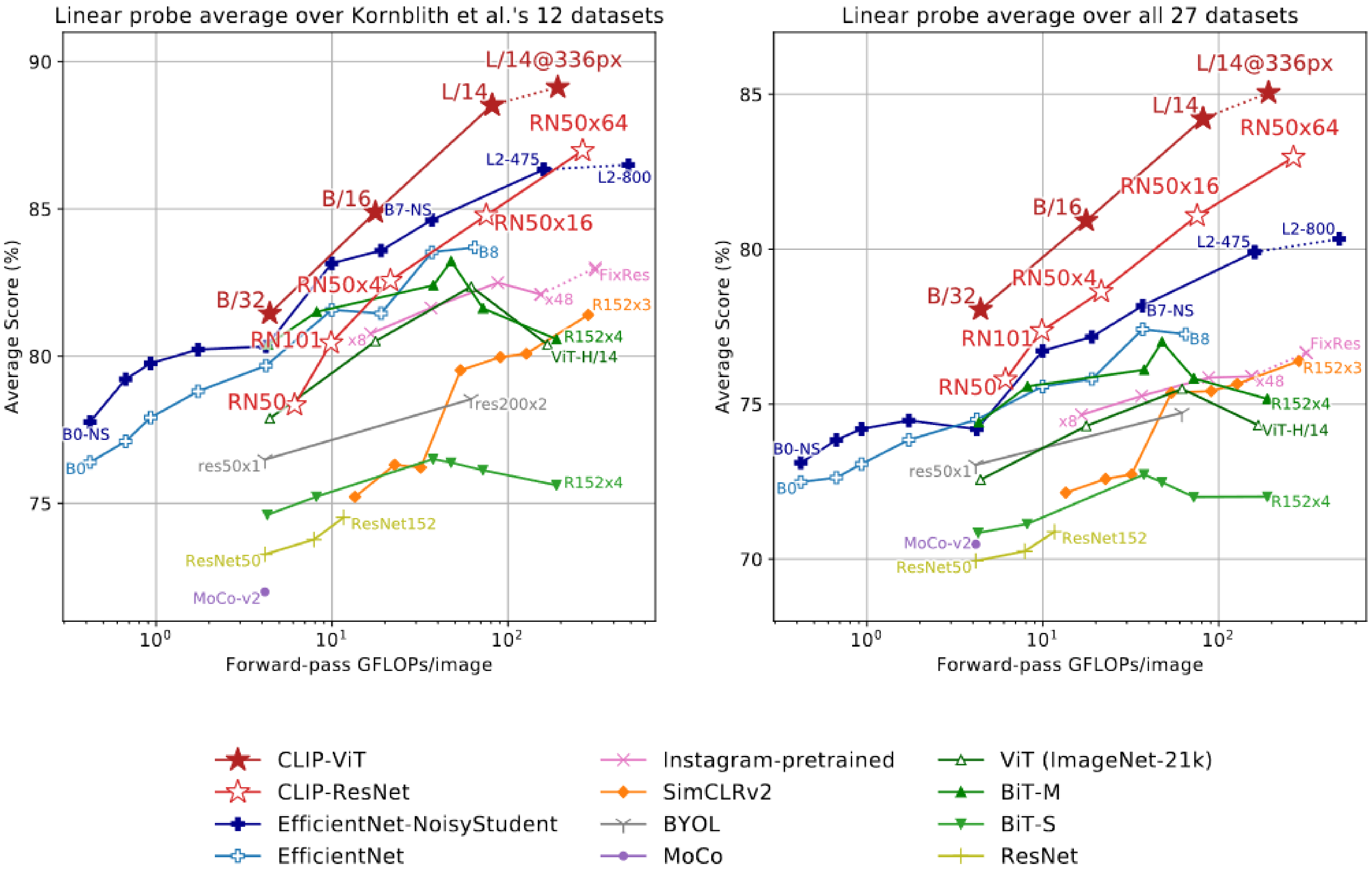
```
# image_encoder - ResNet or Vision Transformer
# text_encoder  - CBOW or Text Transformer
# I[n, h, w, c] - minibatch of aligned images
# T[n, l]       - minibatch of aligned texts
# W_i[d_i, d_e] - learned proj of image to embed
# W_t[d_t, d_e] - learned proj of text to embed
# t             - learned temperature parameter

# extract feature representations of each modality
I_f = image_encoder(I) #[n, d_i]
T_f = text_encoder(T)  #[n, d_t]

# joint multimodal embedding [n, d_e]
I_e = l2_normalize(np.dot(I_f, W_i), axis=1)
T_e = l2_normalize(np.dot(T_f, W_t), axis=1)

# scaled pairwise cosine similarities [n, n]
logits = np.dot(I_e, T_e.T) * np.exp(t)

# symmetric loss function
labels = np.arange(n)
loss_i = cross_entropy_loss(logits, labels, axis=0)
loss_t = cross_entropy_loss(logits, labels, axis=1)
loss   = (loss_i + loss_t)/2
```

CLIP results

CLIP is much more efficient at zero-shot transfer than image caption baseline



Linear probe performance of CLIP models in comparison to state-of-the-art computer vision models



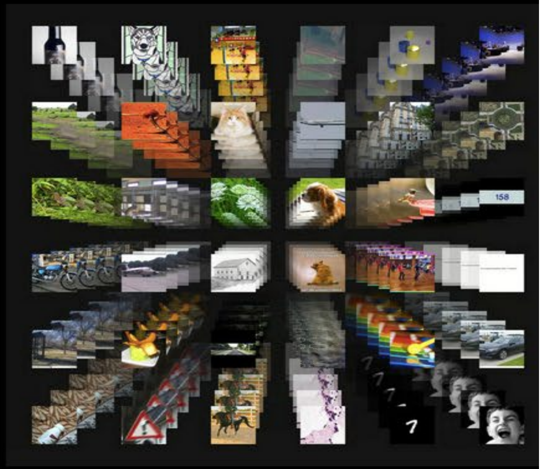
DALLE-2 Demo

- ➢ Important papers relevant to understand DALLE-2

Hierarchical text-conditional image generation with CLIP latents
Apr 13, 2022

DALL·E: Creating images from text
Jan 5, 2021

DALL·E 2 pre-training mitigations
Jun 28, 2022



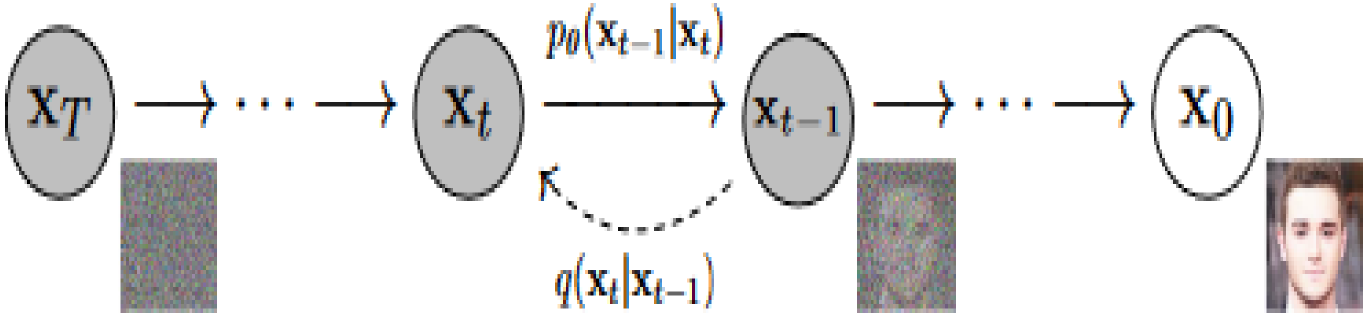
CLIP: Connecting text and images
Jan 5, 2021

- ➢ Let’s get an overview of DALLE-2 and see it in action

https://openai.com/dall-e-2

Diffusion models (DM)

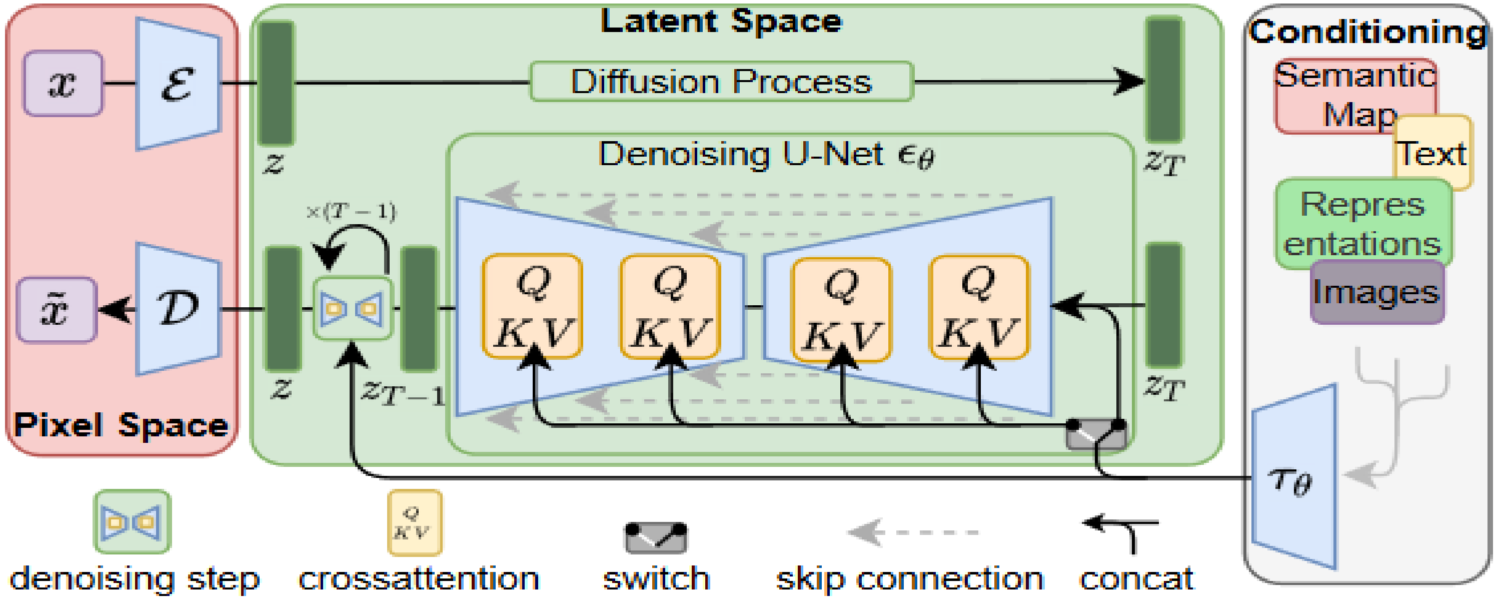
- Generative models that can generate diverse high-resolution images given a text prompt.
- Inspired by thermodynamics
- Diffusion models learn to generate data by reversing a gradual noising process
- DM learns to navigate along the parameterized Markov chain, gradually removing the noise over a series of timesteps to reverse the process of generating image from random noise.
- **Key idea:** Learning to generate images by iterative denoising



Stable Diffusion

Many components integrated together to generate high-resolution images

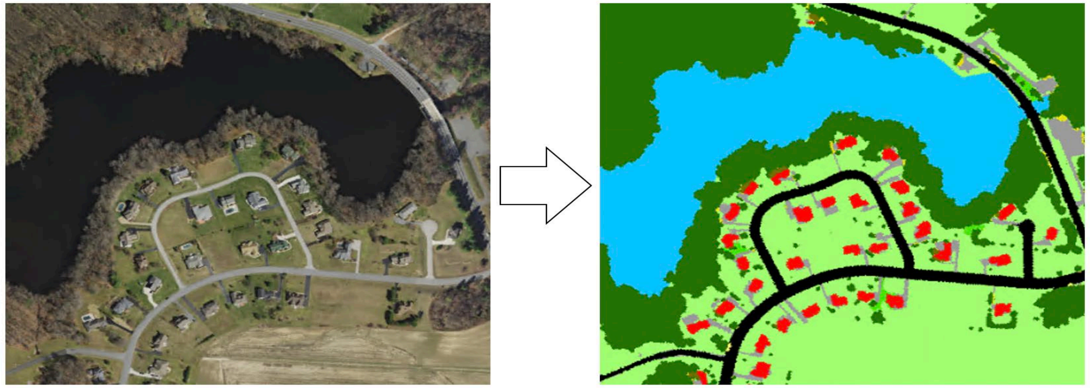
- **Forward/reverse diffusion** -> method of learning to generate new stuff
- **Image-text representations** -> method to link images and text (e.g., CLIP method)
- **Autoencoder** -> method to compress images (important to speed-up process)
- **U-Net + Attention** -> methods to add in good inductive biases (to generate novel images)



Source: Rombach et al., High-Resolution Image Synthesis with Latent Diffusion Models. CVPR 2022.
Stable Diffusion slides by Binxu Wang

U-Net for image segmentation

- U-net learns segmentation in an end-to-end setting
- Proven to be very powerful segmentation tool in scenarios with limited annotated data
- Doesn’t contain any fully connected layers



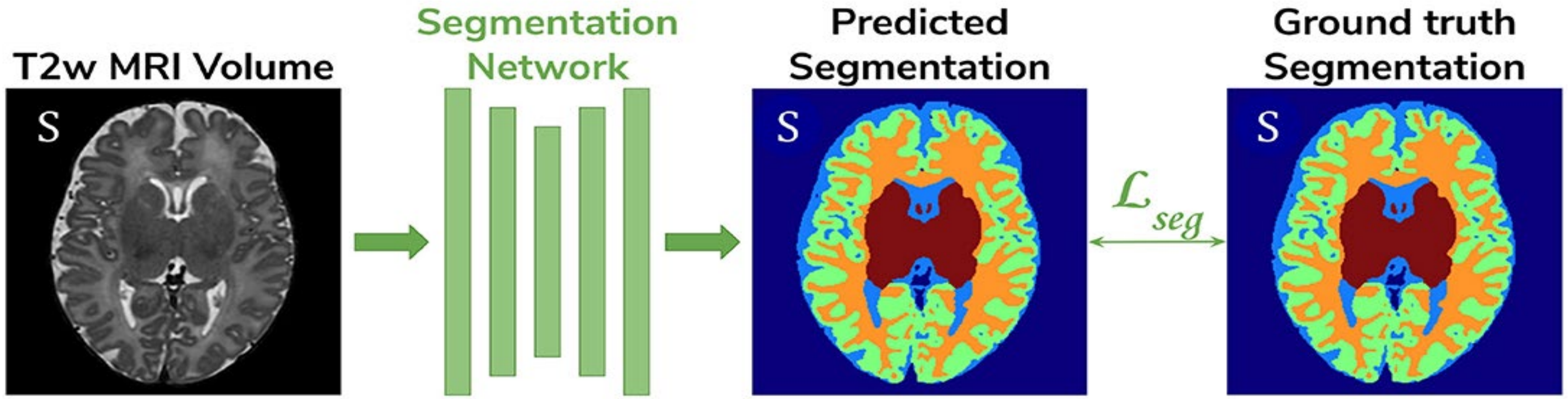
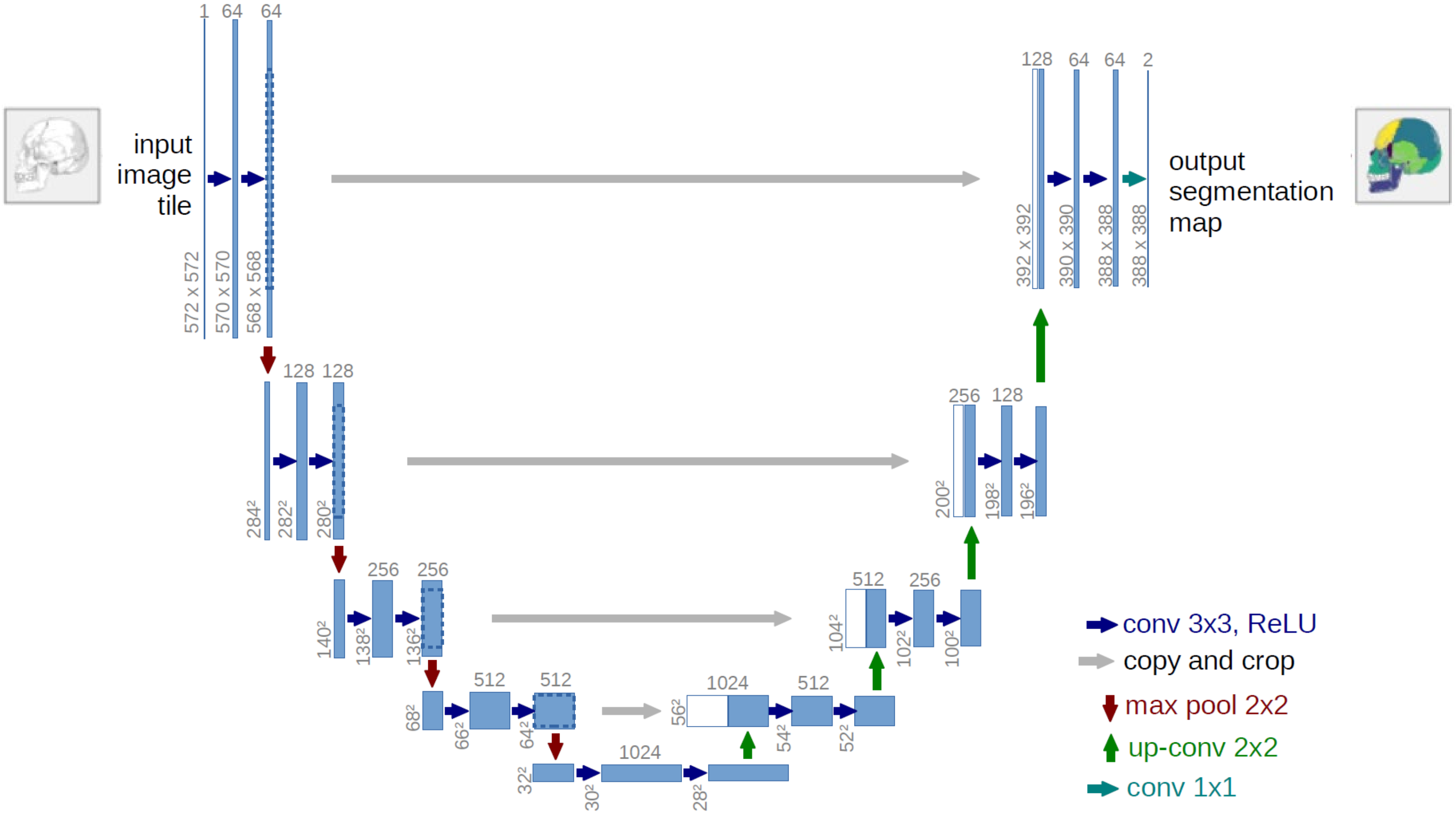


Image Credit: https://developers.arcgis.com/python/guide/how-unet-works/
https://www.frontiersin.org/articles/10.3389/fnins.2021.662005/full

U-net Architecture

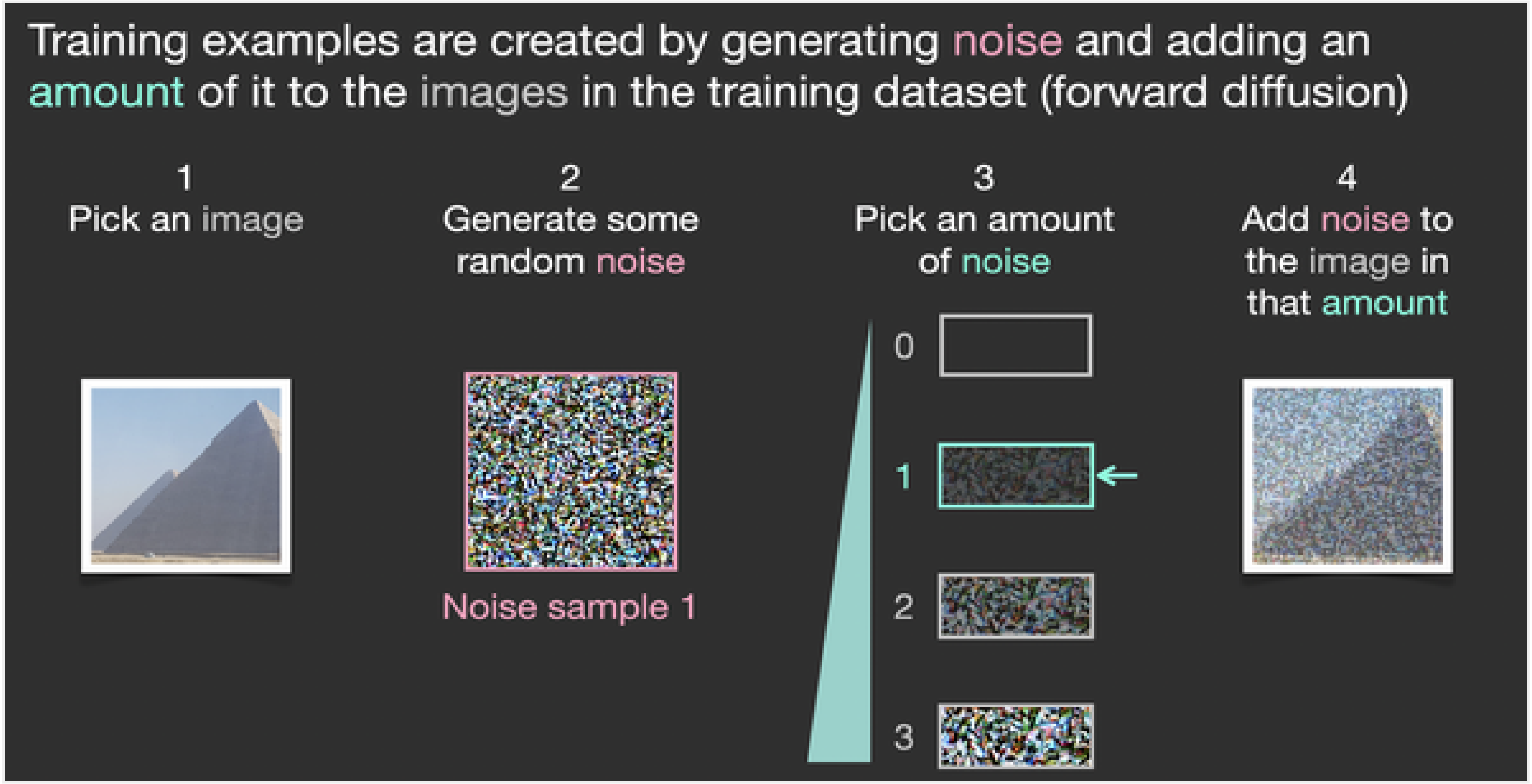


Source: Ronneberger et al. (2015). U-net: Convolutional Networks for Biomedical Image Segmentation. MICCAI 2015.

How diffusion works

Acknowledgement: Jay Alammar https://jalammar.github.io/illustrated-stable-diffusion/

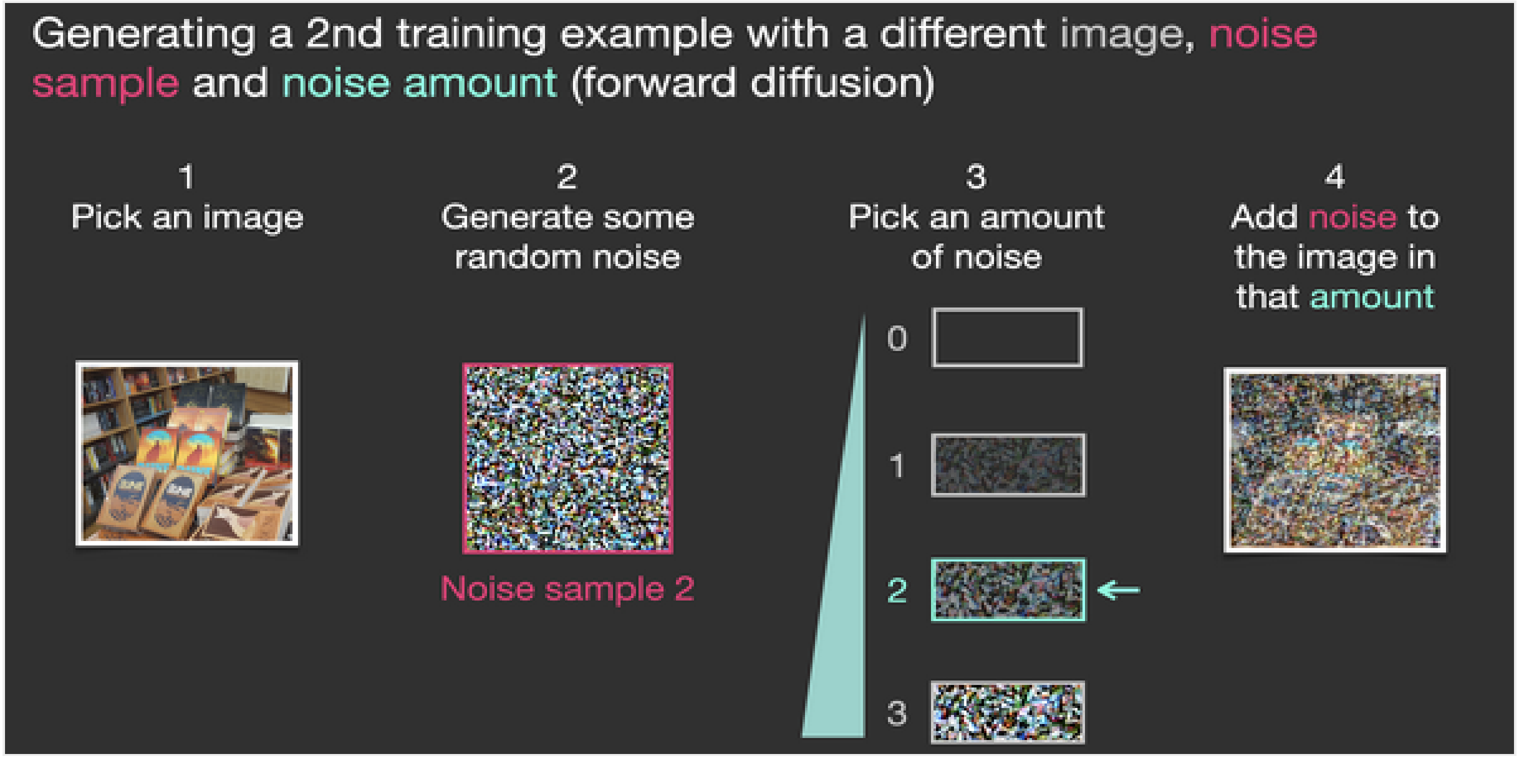
- Say we have an image, we generate some noise, and add it to the image



How diffusion works

Acknowledgement: Jay Alammar https://jalammar.github.io/illustrated-stable-diffusion/

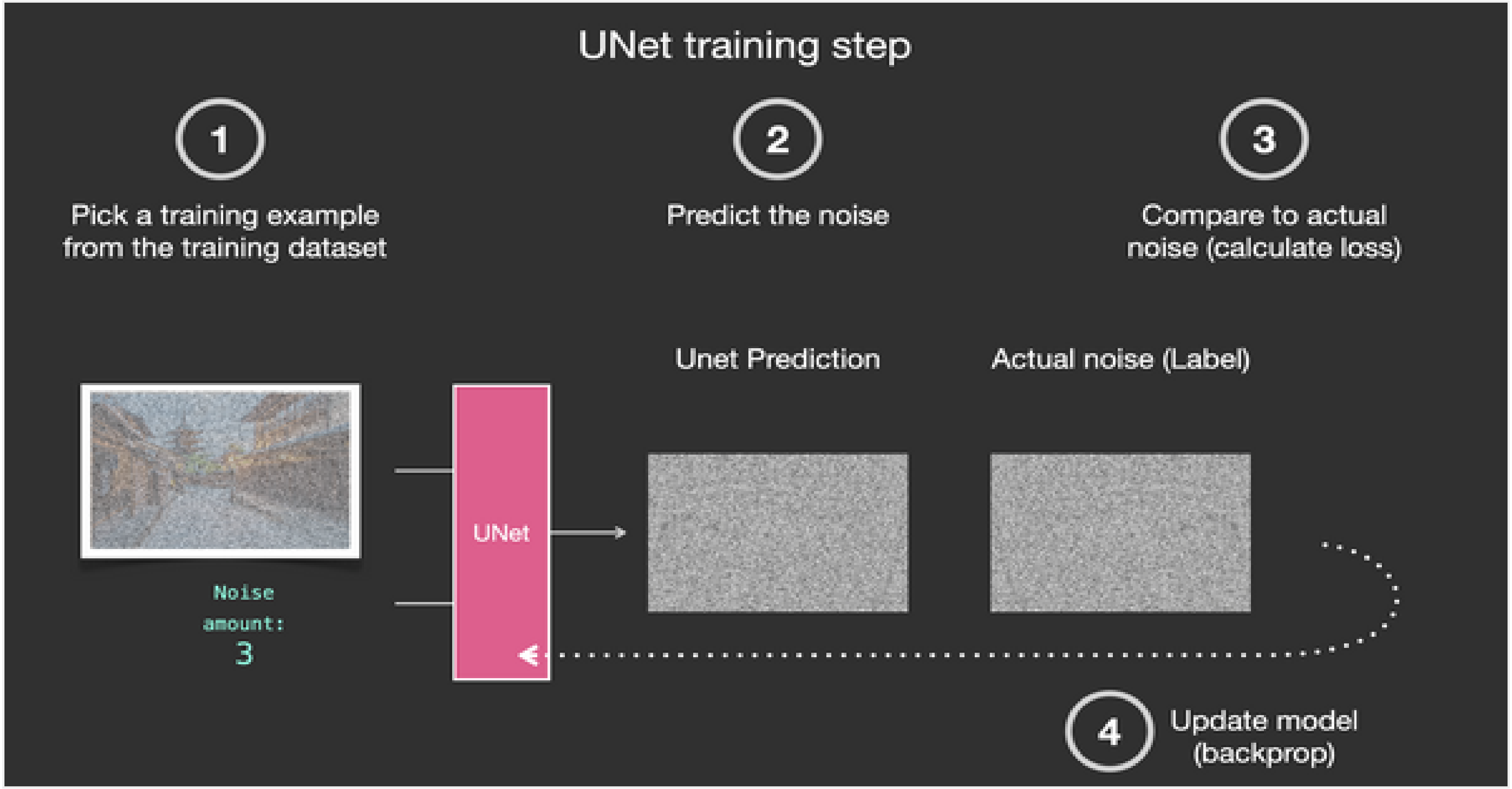
- Create lots of training examples like the same



How diffusion works

Acknowledgement: Jay Alammar https://jalammar.github.io/illustrated-stable-diffusion/

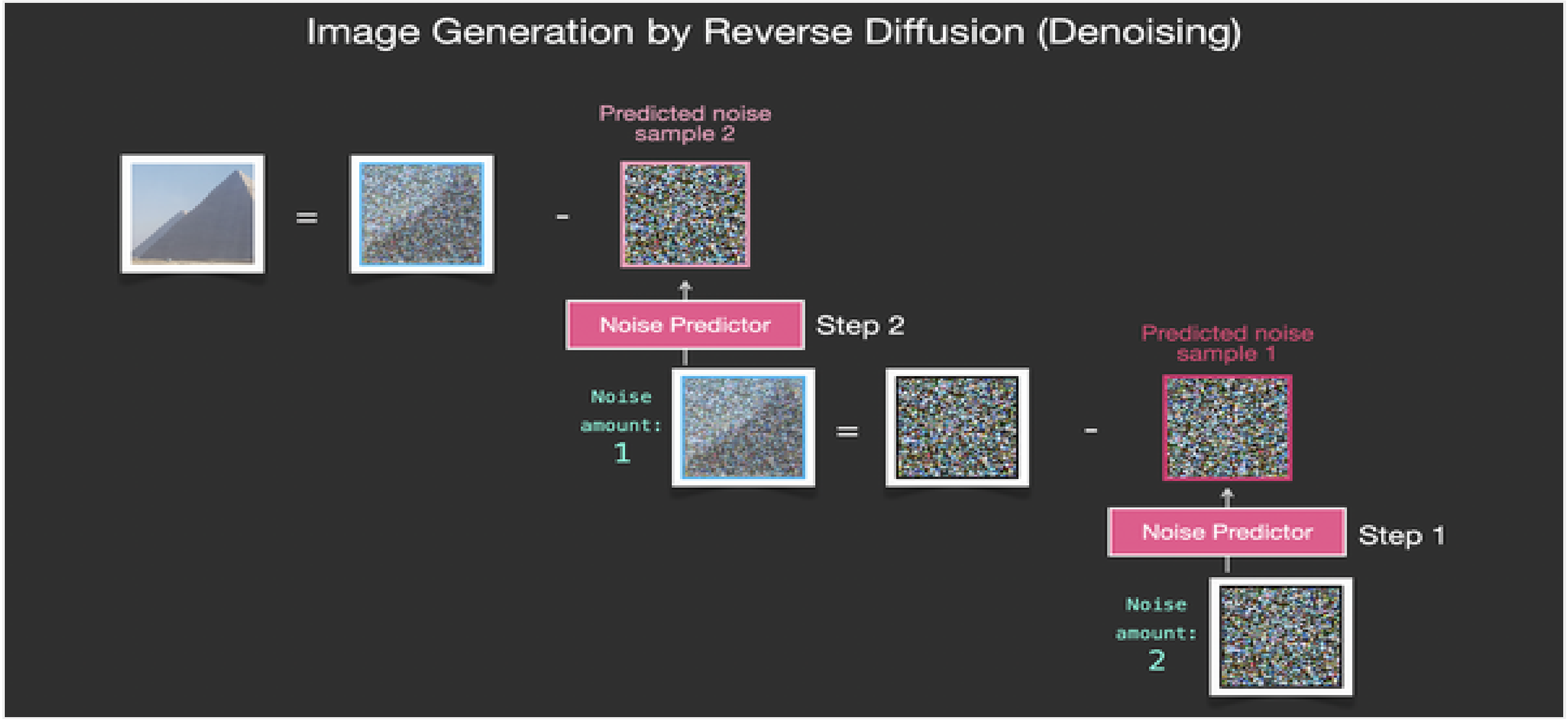
- With dataset, train the noise predictor



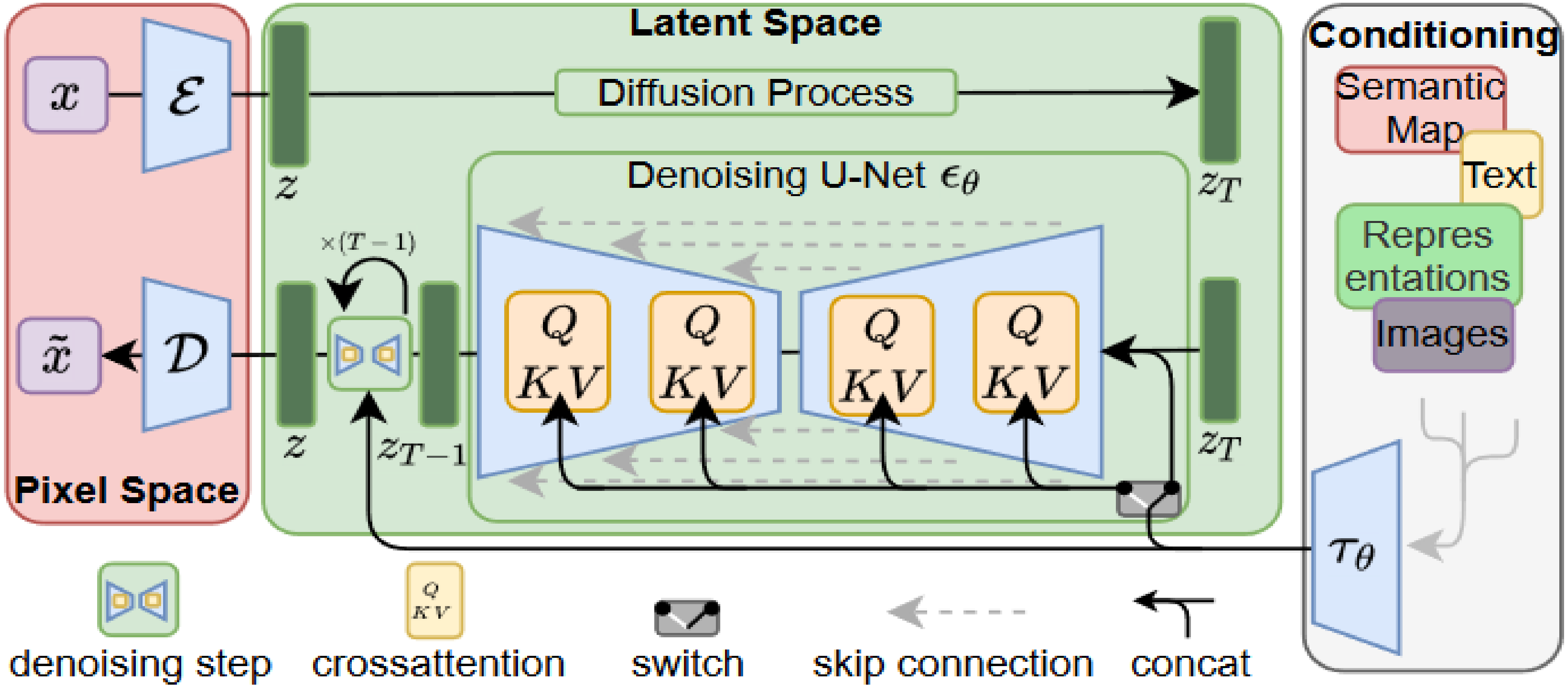
How diffusion works

Acknowledgement: Jay Alammar https://jalammar.github.io/illustrated-stable-diffusion/

- The trained noise predictor can take a noisy image, and with number of denoising steps, is able to predict slice of noise
- The predicted slice of noise is subtracted to get an image that is closer to images model was originally trained on (distribution of pixels)



Latent Diffusion Model/Stable Diffusion

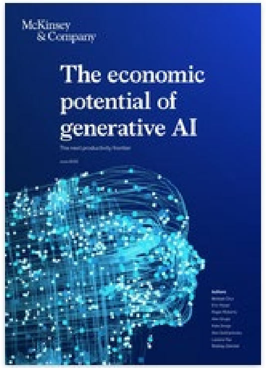


Source: Rombach et al., High-Resolution Image Synthesis with Latent Diffusion Models. CVPR 2022.
Stable Diffusion slides by Binxu Wang

Source: Our world in Data https://ourworldindata.org/

Economic potential of Generative AI

Tweet: from Sam Altman, CEO of OpenAI

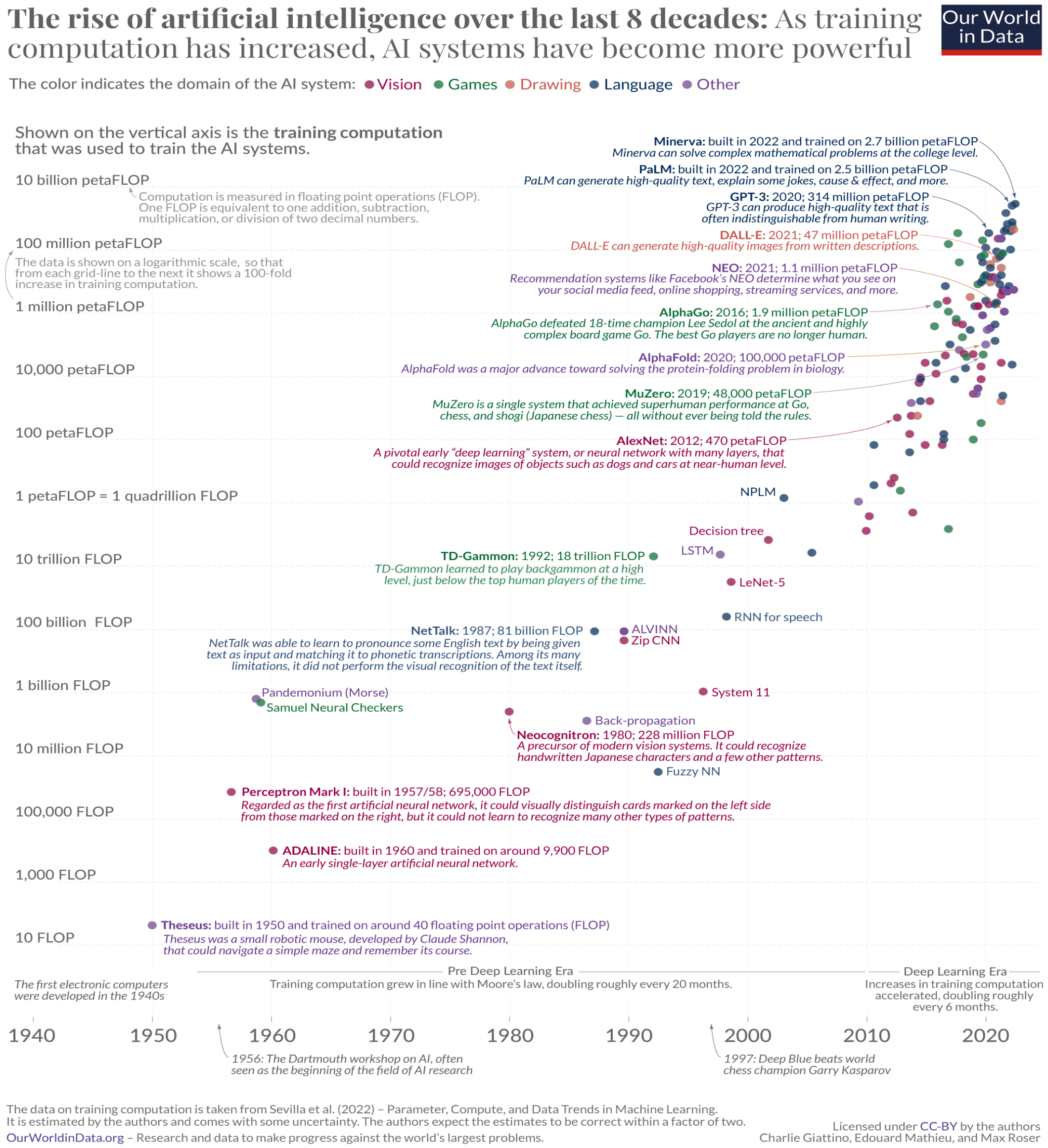


McKinsey & Company (2023): "Generative AI is poised to unleash the next wave of productivity. …"

Goldman Sachs (2023): "we estimate that one-fourth of current work tasks could be automated by AI in the US … with particularly high exposures in administrative (46%) and legal (44%) professionals and low exposures in physically-intensive professions such as construction (6%) and maintenance (4%).

Source: https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/the-economic-potential-of-generative-ai-the-next-productivity-frontier#introduction
https://www.goldmansachs.com/intelligence/pages/generative-ai-could-raise-global-gdp-by-7-percent.html

Rise of AI over the last 8 decades



Source: Our World in Data https://ourworldindata.org/

Summary

- Generative AI has huge potential in varied applications, especially content creation.
- Generative AI can be used to:
  - formulate first drafts of promotional material
  - draft procedural correspondence
  - generate itineraries for trips
  - generate synthetic data
  - code generation (code completion, bug fixing, code style checking, code refactoring)
  - content creation for courses
  - creating designs for fashion designers
  - generating explanations for loan denials
  - multilingual customer support
  - generating automated email replies for customer support
  - Job description generation
  - draft content creation for writers
- Healthy use of generative AI will likely improve productivity across various industries.

References

[1] Ho et al., Denoising Diffusion Probabilistic Models. NeurIPS 2020.

https://arxiv.org/pdf/2006.11239.pdf?ref=assemblyai.com

[2] Ramesh et al., Hierarchical Text-Conditional Image Generation with CLIP Latents. 2022.

https://arxiv.org/pdf/2204.06125.pdf

[3] Desai and Johnson. VirTex: Learning Visual Representations from Textual Annotations

https://arxiv.org/pdf/2006.06666.pdf

[4] Zhang et al., Contrastive Learning of Medical Visual Representations from Paired Images and Text. 2022.

https://arxiv.org/pdf/2010.00747.pdf

[5] Radford et al., Learning Transferable Visual Models From Natural Language Supervision. 2021.

https://arxiv.org/pdf/2103.00020.pdf

[6] Vaswani et al., Attention is All You Need. NeurIPS 2017.

https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[7] Dosovitskiy et al., An Image is Worth 16 x 16 words: Transformers from image recognition at scale. ICLR 2021.

https://openreview.net/pdf?id=YicbFdNTTy

[8] Rombach et al., High-Resolution Image Synthesis with Latent Diffusion Models. CVPR 2022.

https://openaccess.thecvf.com/content/CVPR2022/papers/Rombach_High-Resolution_Image_Synthesis_With_Latent_Diffusion_Models_CVPR_2022_paper.pdf

[9] Ronneberger et al., U-Net: Convolutional Networks for Biomedical Image Segmentation. MICCAI 2015.

https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/

[10] Jay Alammar. The Illustrated Stable Diffusion

https://jalammar.github.io/illustrated-stable-diffusion/

Questions?

Please complete myExperience survey

https://myexperience.unsw.edu.au/unsw/