Carnegie Mellon University

Intermediate Deep Learning

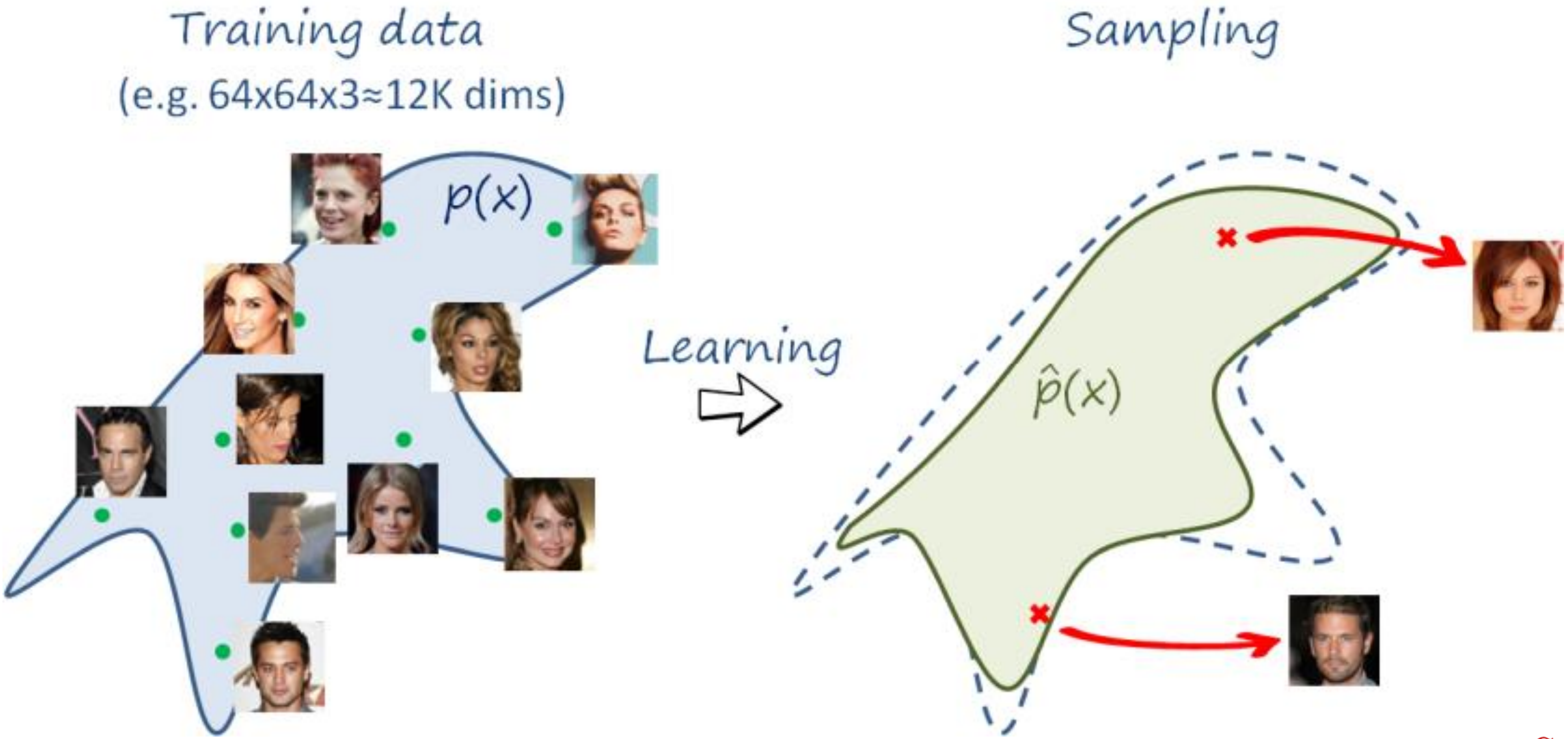
Spring 2025, Deep Learning for Engineers
March 11, 2025, First Session

Amir Barati Farimani
Assistant Professor of Mechanical Engineering and Bio-Engineering
Carnegie Mellon University

Welcome to Mini 4

GPT

What is Generation?



Text To Image Generation

DALL. E2

“a teddy bear on a skateboard in times square”

“Hierarchical Text-Conditional Image Generation with CLIP Latents” Ramesh et al.,2022

Imagen

A group of teddy bears in suit in a corporate office celebrating the birthday of their friend. There is a pizza cake on the desk.

“Photorealistic Tex-to-Image Diffusion Models with Deep Language Understanding”, Saharia et al.,2022

Text To Video Generation

Sora (2024)

Prompt: Photorealistic closeup video of two pirate ships battling each other as they sail inside a cup of coffee.

Prompt: A movie trailer featuring the adventures of the 30 year old space man wearing a red wool knitted motorcycle helmet, blue sky, salt desert, cinematic style, shot on 35mm film, vivid colors.

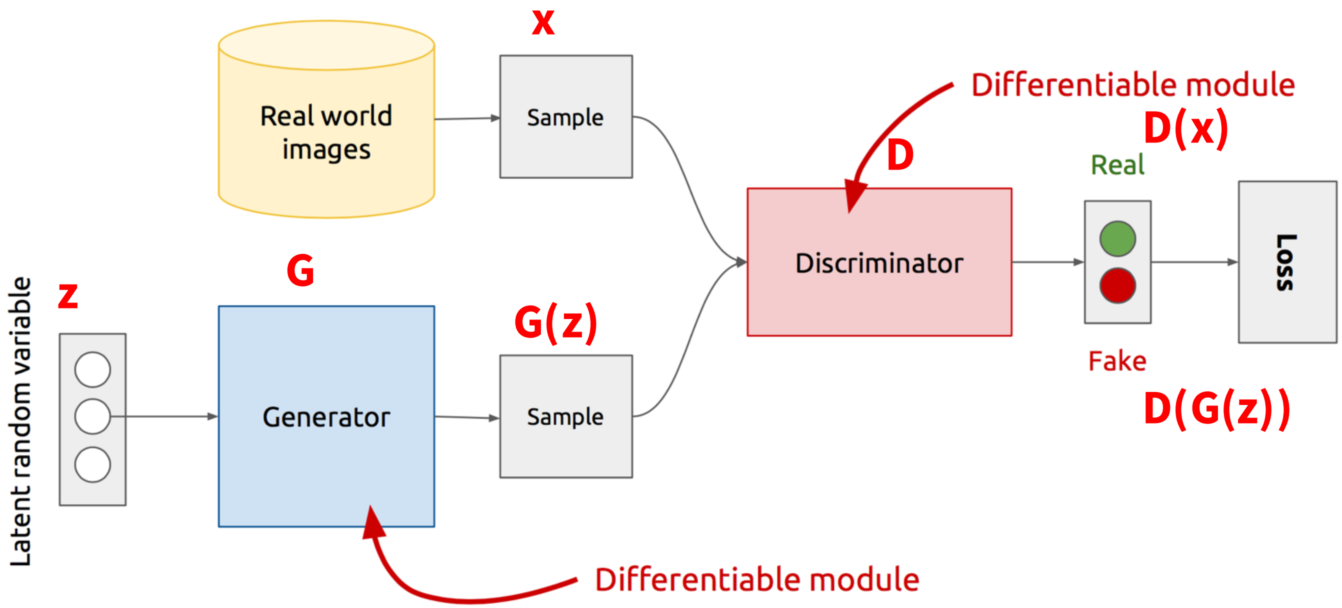
Carnegie Mellon University

G of GPT

Generative Adversarial Networks

G of GPT

Generative Adversarial Networks



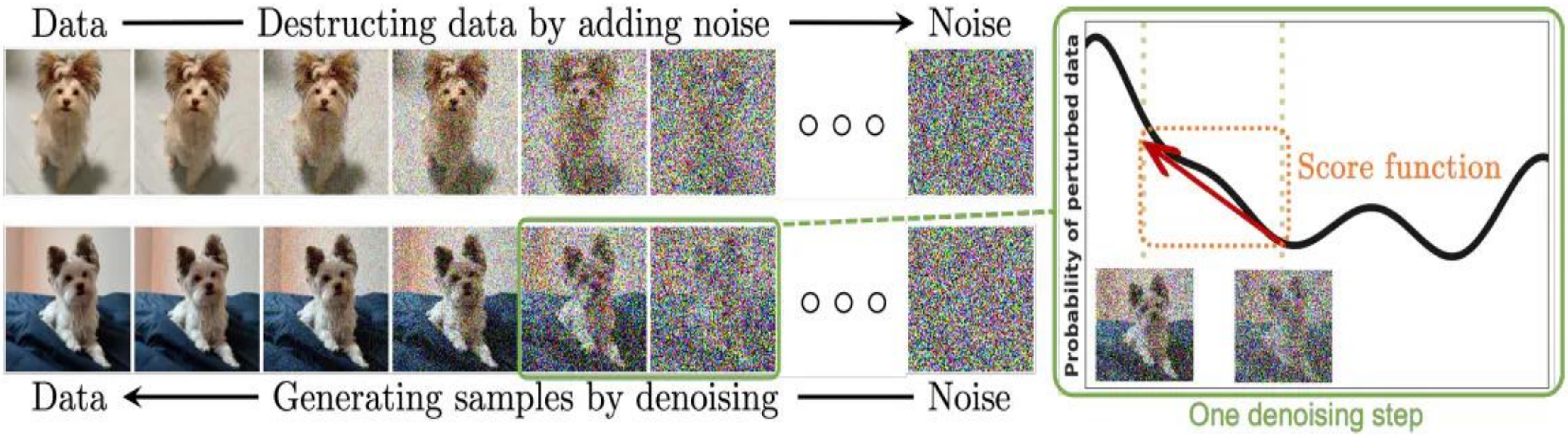
G of GPT

Diffusion Models

Carnegie Mellon University

G of GPT

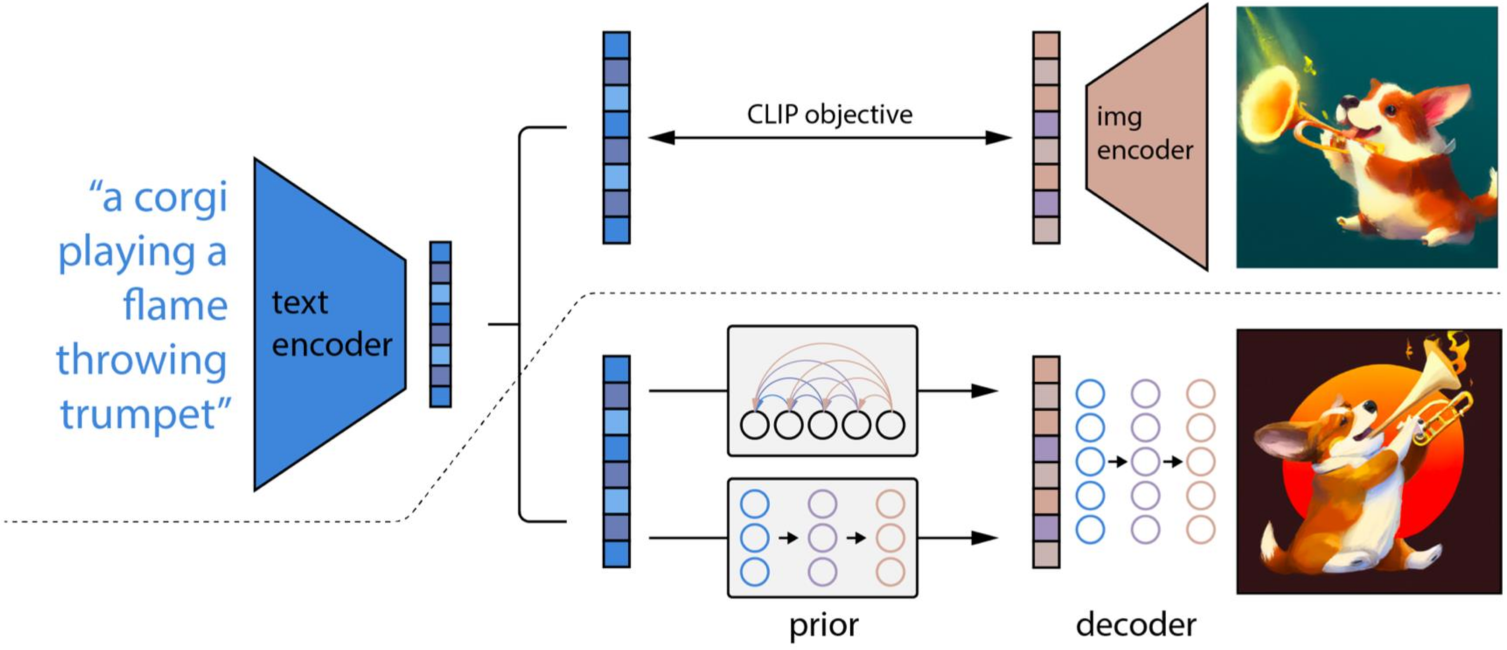
Diffusion Models



Carnegie Mellon University

G of GPT

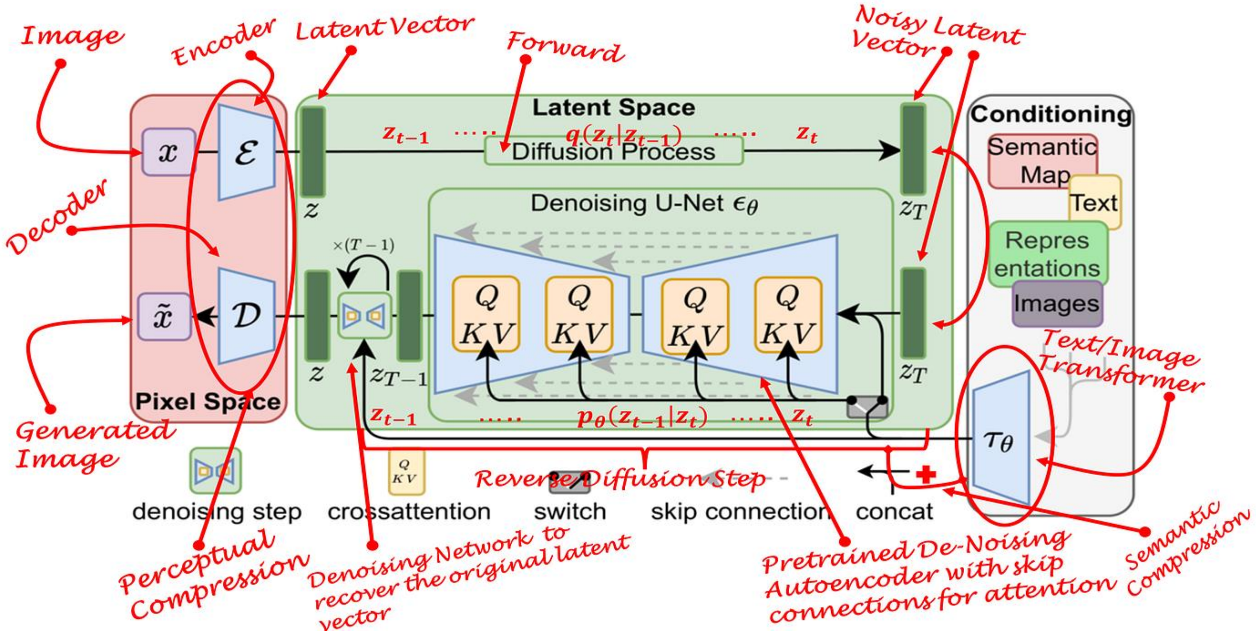
Diffusion Models



Carnegie Mellon University

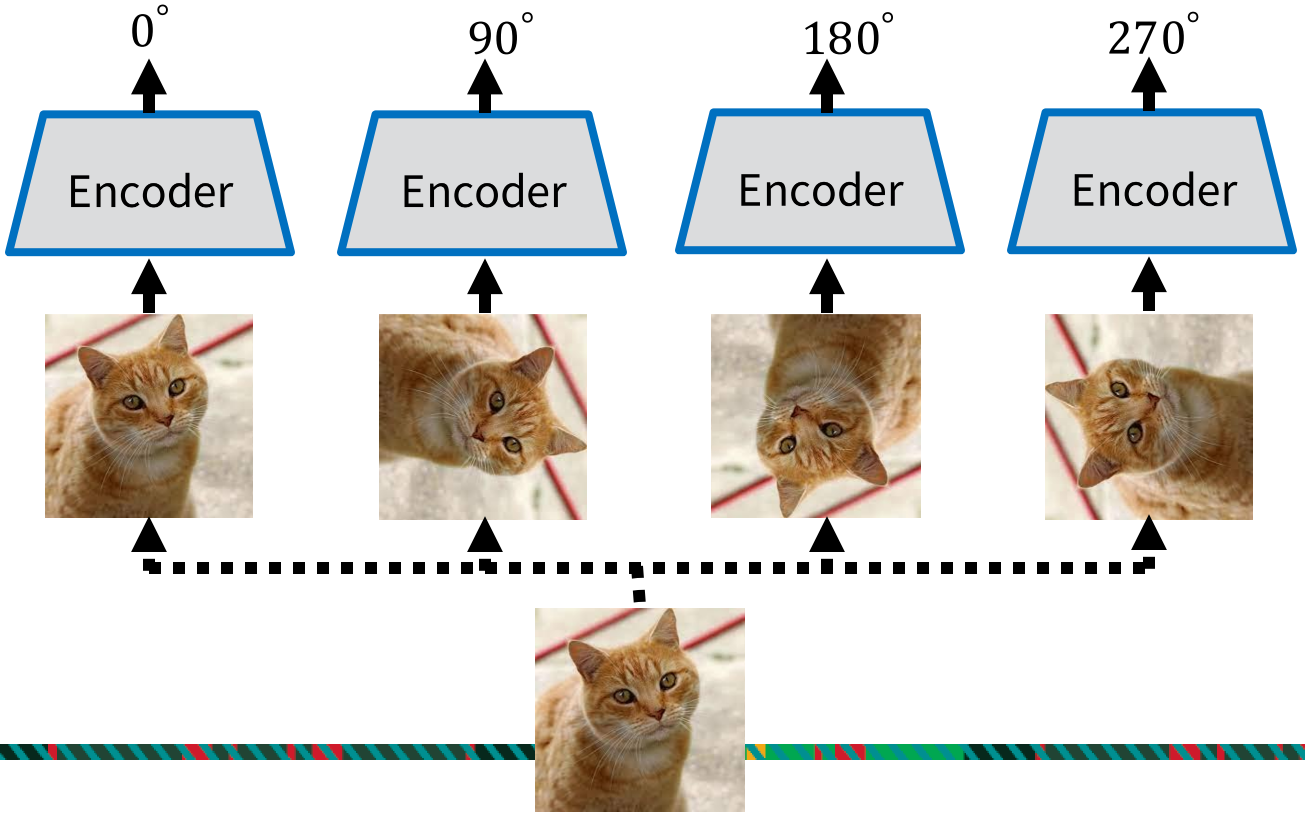
G of GPT

Diffusion Models



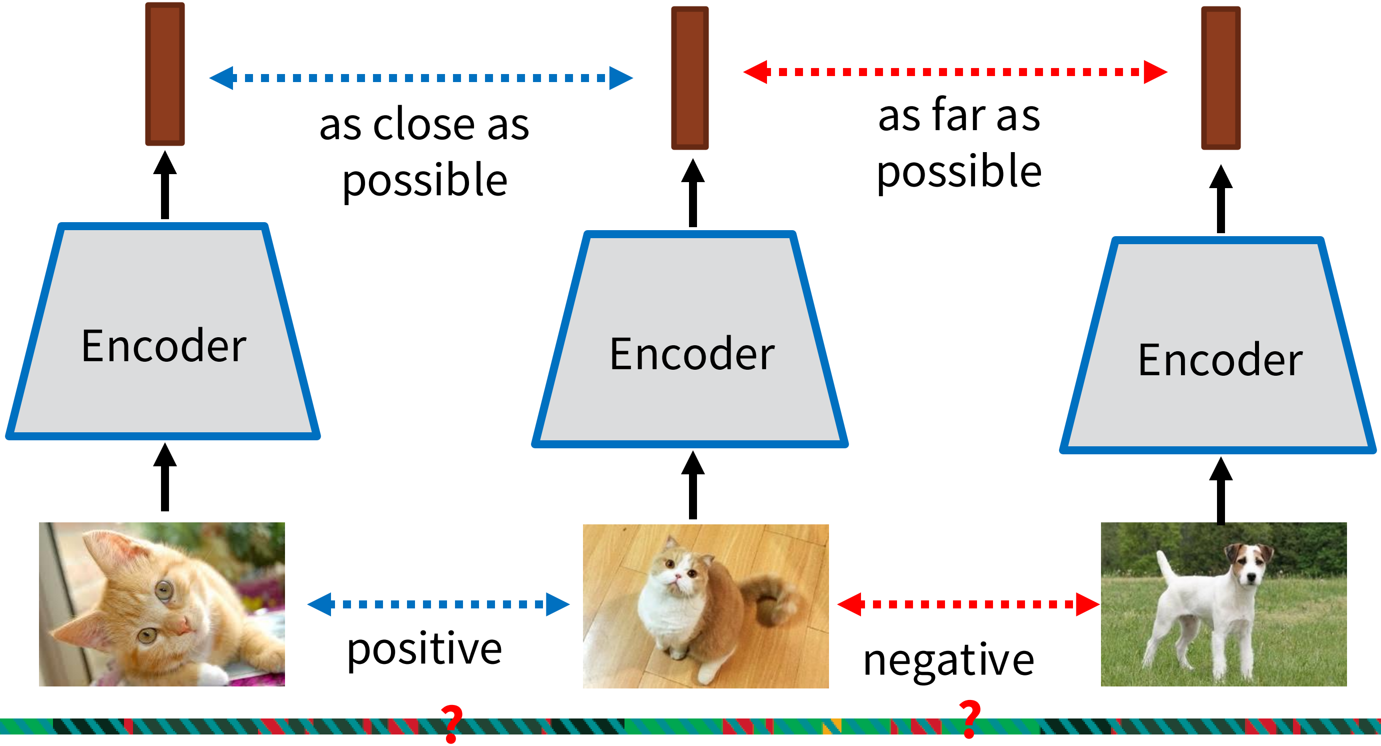
P of GPT

Pretraining: Predictive Learning



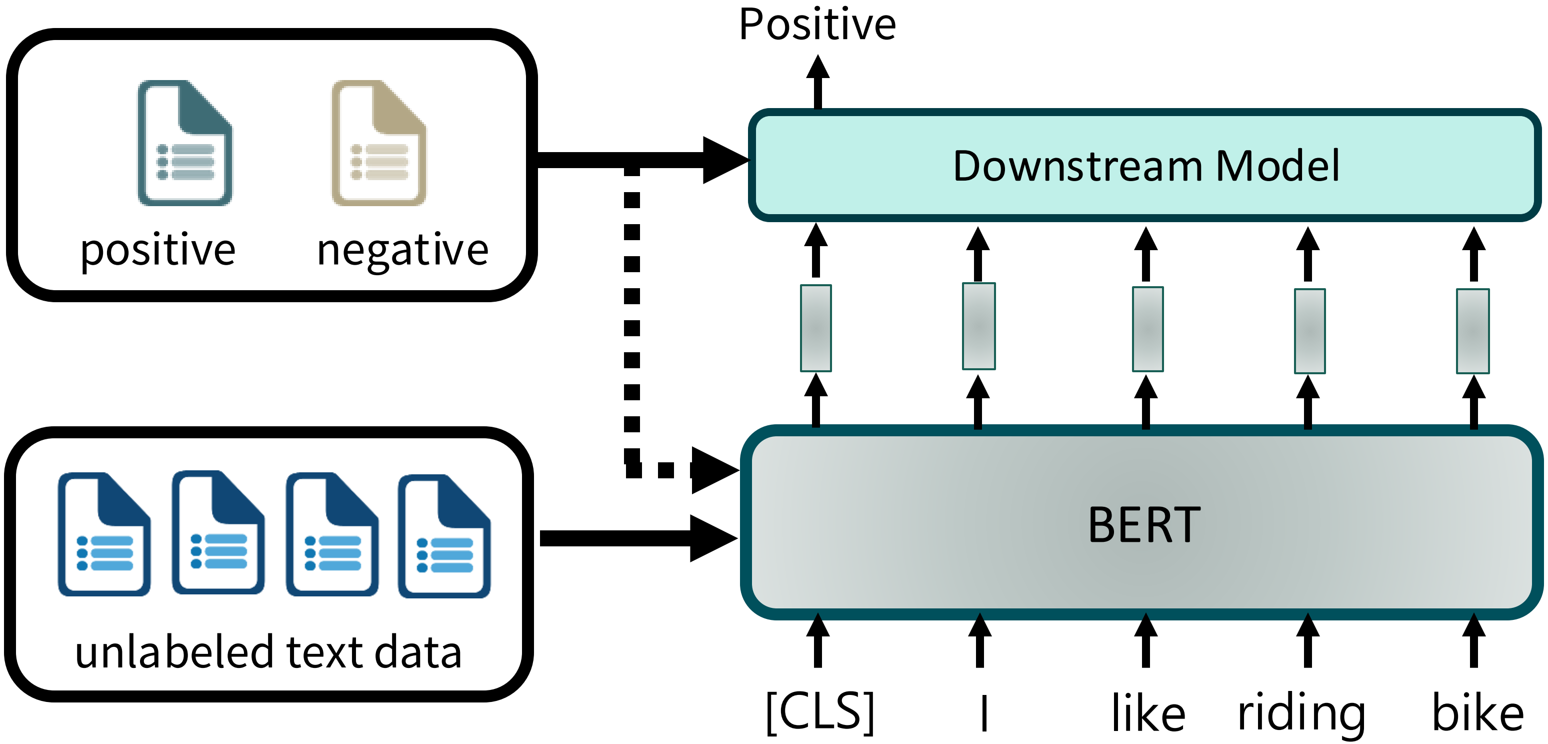
P of GPT

Pretraining: Contrastive Learning



Carnegie Mellon University

P of GPT
Pretraining: in Large Language Models



Carnegie Mellon University

T of GPT

Transformers: Attention is all you need!

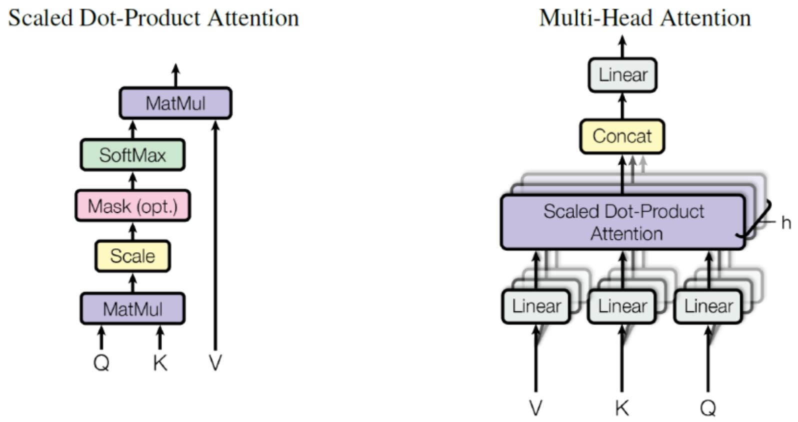
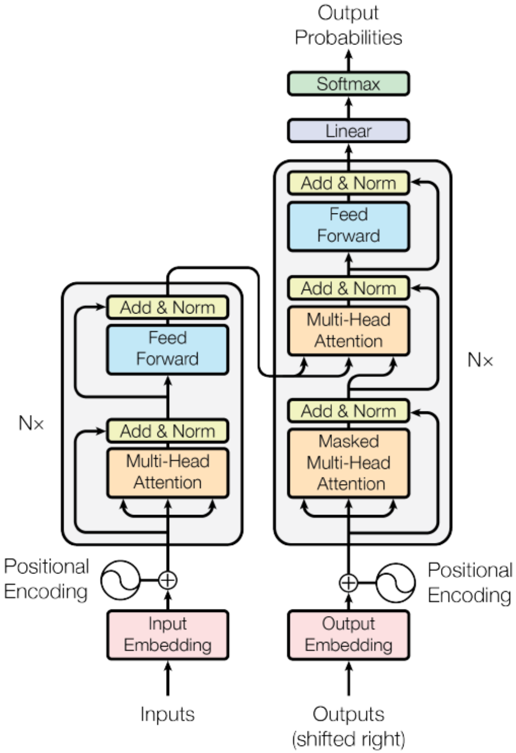


Figure 2: (left) Scaled Dot-Product Attention. (right) Multi-Head Attention consists of several attention layers running in parallel.



T of GPT
Transformers: Attention

Attention Visualizations

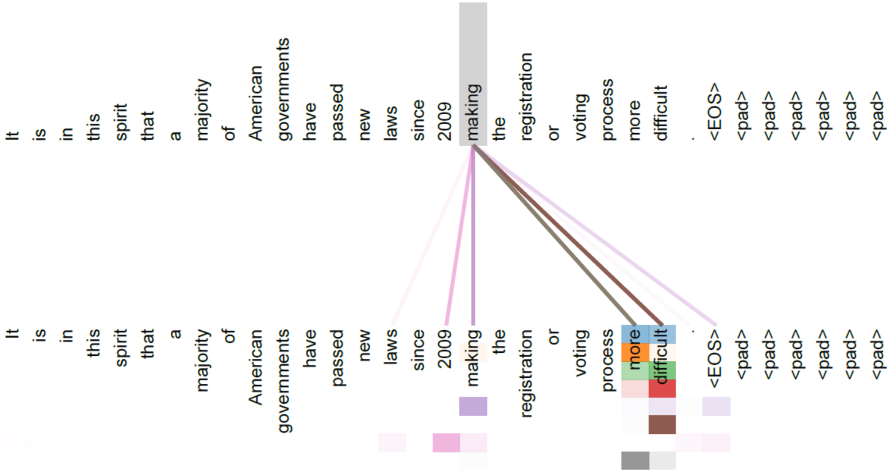
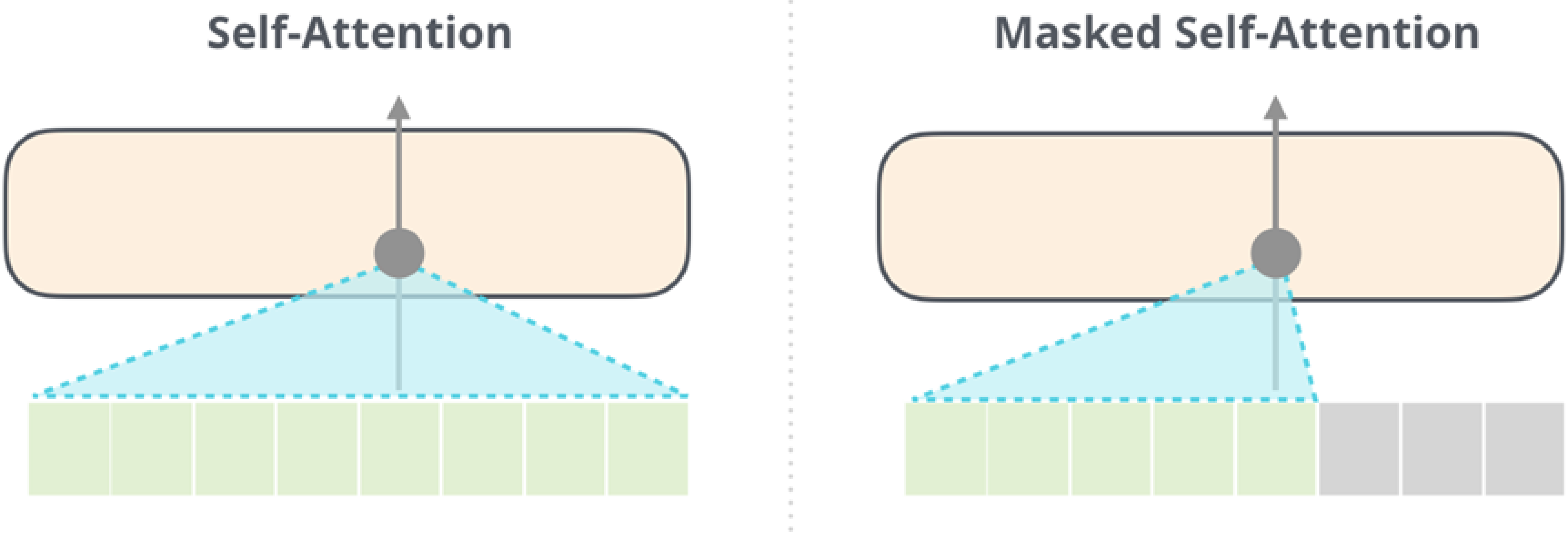


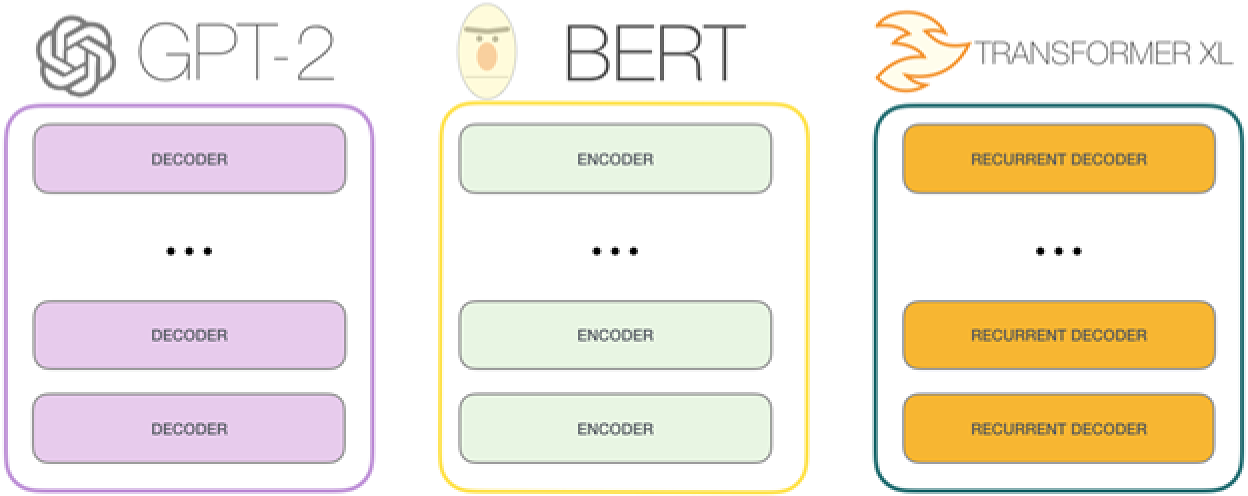
Figure 3: An example of the attention mechanism following long-distance dependencies in the encoder self-attention in layer 5 of 6. Many of the attention heads attend to a distant dependency of the verb 'making', completing the phrase 'making...more difficult'. Attentions here shown only for the word 'making'. Different colors represent different heads. Best viewed in color.

T of GPT
Transformers: Self-Attention

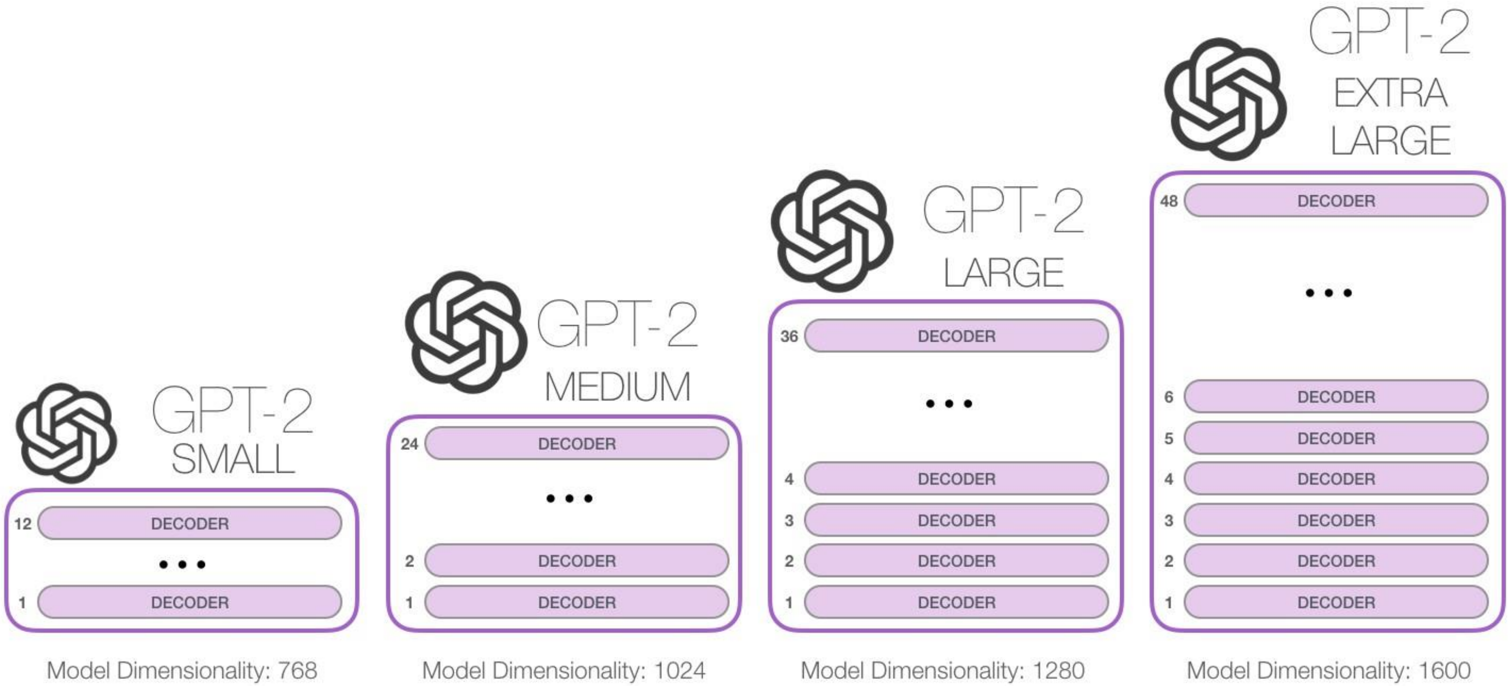


Carnegie Mellon University

Foundational Models



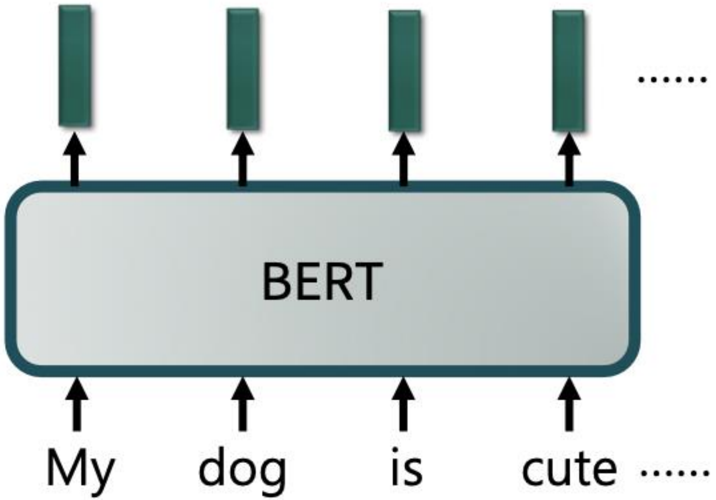
Foundational Models

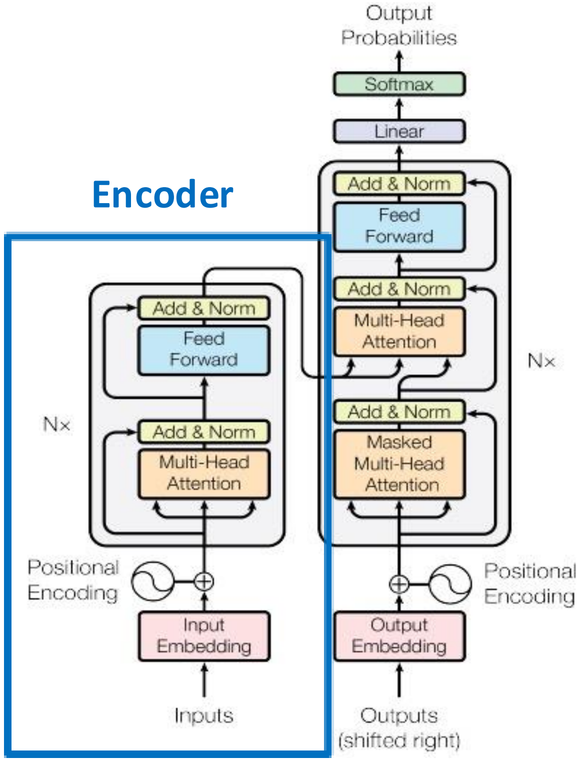


Foundational Models

BERT = ENCODER OF TRANSFORMER

Learned from a large amount of text without annotation





Neural Networks Recap

1. **MLP** (Perceptron, Non-linear separability, Capacity, Depth vs number of Neurons)
2. **Universal Function Approximation**
3. **Empirical Risk Minimization** (Changing Integral to Summation with samples)
4. **Neural Networks Ingredients** (inputs, outputs, Loss functions, architectures)
5. **Optimization (**Gradient Descent**) and Backpropogation** (Chain rules & automatic differentiation)
6. **Design of F (x; w): Regularization** (weight Initialization, Drop out, Data Augmentation, etc.)
7. **Convolutional Neural Networks** (Automatic Feature Learners)

Carnegie Mellon University

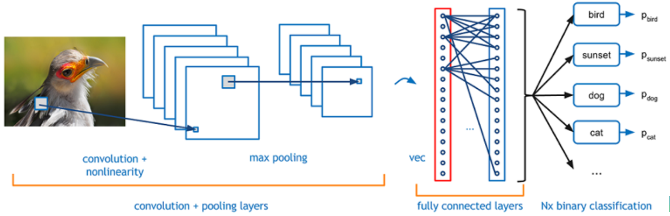
CNN Recap

4. How to build a scanner for feature learning? what should be the properties of this scanner?

1. It should be numbers (a matrix) because it should be machine readable
2. It should be learnable
3. It should be flexible in size and dimension
4. Should be pluggable to Neural Networks

5. Can we design the scanners based on the learning tasks?

Yes, and we should. Because the mode of data might be different (sound, image, video) and features are needed based on the task to make a good model, the scanner should ONLY learn the relevant features connecting them to the output



Carnegie Mellon University

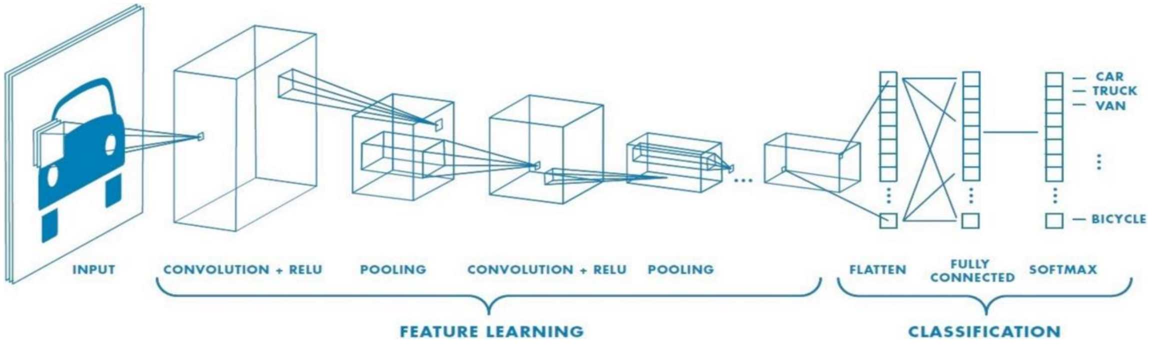
CNN Recap

6. How can the scanners learn?

Inspired by iterative optimization and backpropagation in neural networks, we can iteratively learn the initialized weights of scanners (remember these are numbers)

7. How can we plug in the scanners into Neural networks?

By flattening the output of the last convolved map and passing it to the FC layer, we can forward propagate, and we can backpropagate to learn the parameters of a filter (scanner)



CNN Recap

8. What are the components of CNN and why they are necessary?

Components of CNNs are (Convolution, Non-linearity, Pooling (subsampling)). Convolution operation is for learning the filters and scanners. Non-linearity is for having more robust representation and pooling is for making the network translation invariant and focus on the important features

9. What are the good consequences of CNN layer?

1. Learning spatial features, 2. Weight sharing and reduction in the number of parameters, 3. Translation invariant representation learning