Bonus Slide

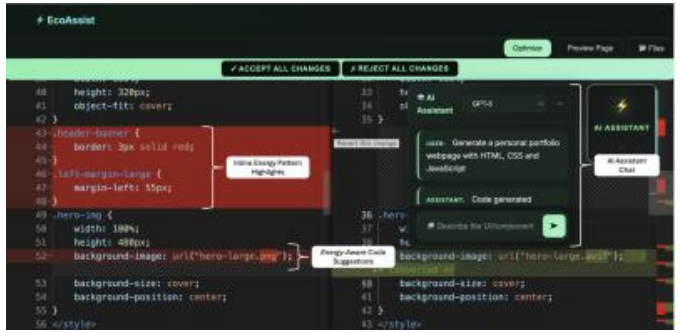
Human-Computer Interaction Institute (HCII)

CMU's Human-Computer Interaction Institute (HCII) was founded in 1993 as an interdisciplinary effort. The department studies how people interact with computational devices and how computers integrate into society.



[Increasing Mobility with Lightweight Exoskeletons]

This field draws on concepts and techniques from psychology and design in addition to computer science. Specialties include context-aware computing, social computing, and user experience design.



[AI Tool Shows Developers Real-time Sustainability Guidelines]

Unit 4: Demystifying CS

Generative AI

15-110 – Friday 10/30

Announcements

- Project2 due next Monday

Learning Goals

- Differentiate between **Artificial Intelligence**, **Machine Learning**, and **Generative AI** as nested but distinct fields.

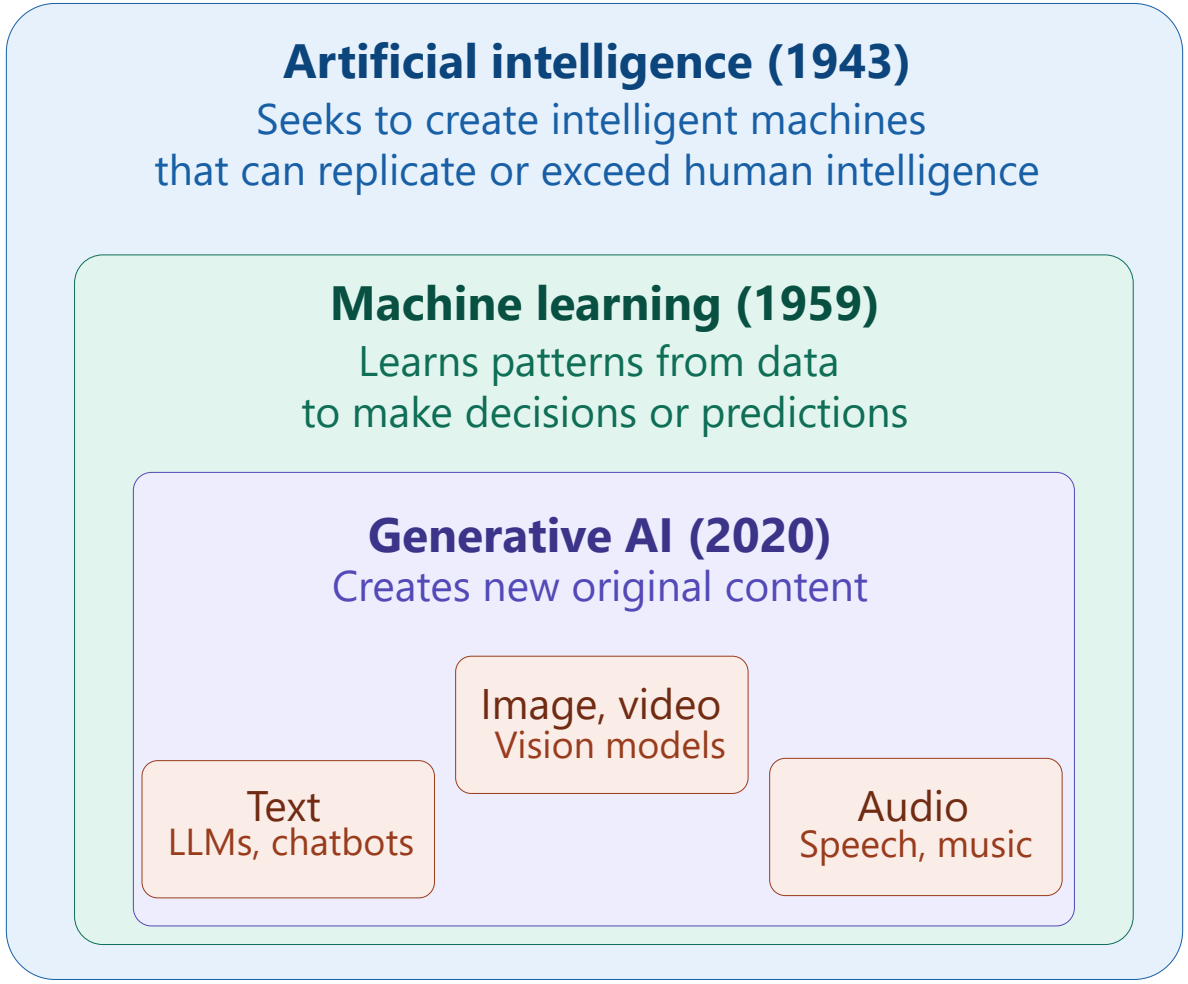
- Recognize how **transformers** and **diffusion models** are used by **generative artificial intelligence** to produce new text or media based on a prompt.

The AI Family Tree

We have talked a bit about GenAI and ML at this point.

While they are closely related, they are not the same thing.

Today we will focus on how these different fields relate to each other and dig a bit deeper into some of them.



Artificial Intelligence (AI)

What is Artificial Intelligence?

Artificial Intelligence (AI) is a branch of computer science that studies techniques which allow computers to do things that, when humans do them, are considered *evidence of intelligence*.

However, it's extremely hard to build a machine with **general intelligence -** that is, a machine that can do everything a human can do. This is complicated because it involves many difficult tasks (visual and auditory perception, language understanding, reasoning, planning, and more).

Many modern AI applications are **specialized**; they do one specific task, and they do it very well. We call an AI application trained for a specific task an **agent**.

Examples of AI Agents

We've built AI agents that can play games, run robots, animate children's drawings, and do a variety of other tasks.

The term "agent" is used loosely in AI; sometimes it just means a program, sometimes it means a system that senses and acts in an environment.

Some agents work alone. Others are more complicated systems where multiple agents coordinate to get a complex job done.

What do these agents have in common? Each agent we build is specialized and has a specific **goal** (to be good at one task or a related handful of tasks).



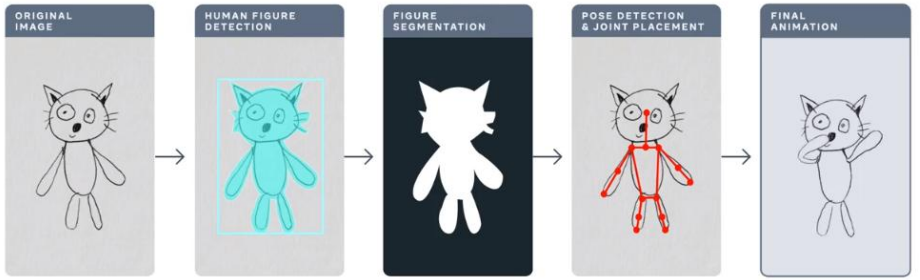


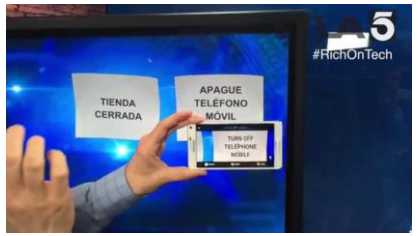


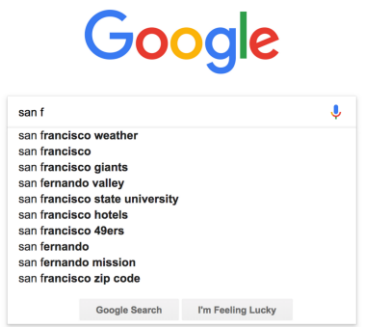


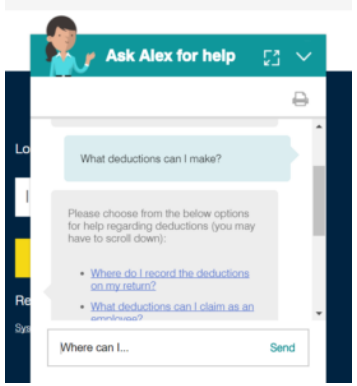












Example: IBM Watson

In 2011 IBM's AI agent Watson was designed to play (and win!) the game Jeopardy. Its **goal** was to answer Jeopardy problems with a question.



It combined multiple AI techniques (NLP, ML, knowledge representation) to parse natural-language clues and search vast unstructured data sources in seconds.

Watson is a great example of narrow intelligence that looks impressively broad.

Game AIs

In games, we usually have a clear, measurable goal, which makes them a useful sandbox for testing AI techniques before applying them to more complex real-life problems.

Deep Blue (1997) beat world chess champion Garry Kasparov using search and handcrafted rules (not learning from data).

AlphaGo (2016) beat Go champion Lee Sedol using deep learning + reinforcement learning. The game was long thought too complex to solve with AI (Go has more possible positions than atoms in the universe).



DeepBlue beat chess grandmaster Garry Kasparov



AlphaGo beats 9-dan ranked Go champion Lee Sedol

Fun Fact: AI Agents from CMU (Robots!)

- Carnegie Mellon's Snake Robot: https://www.youtube.com/watch?v=ajhZiGYq3Pw

- Robust Legged Robot: https://www.youtube.com/watch?v=N70CqROzwxI







Generative AI (GenAI)

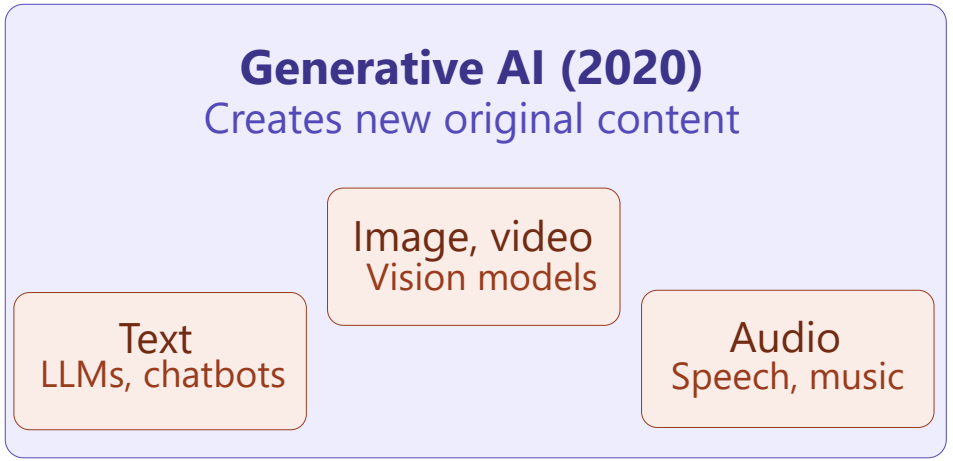
Intro to GenAI

Traditional AI and ML are typically built to make decisions, predictions, or classifications from existing data (e.g., "is this email spam?" or "what's the next best chess move?").

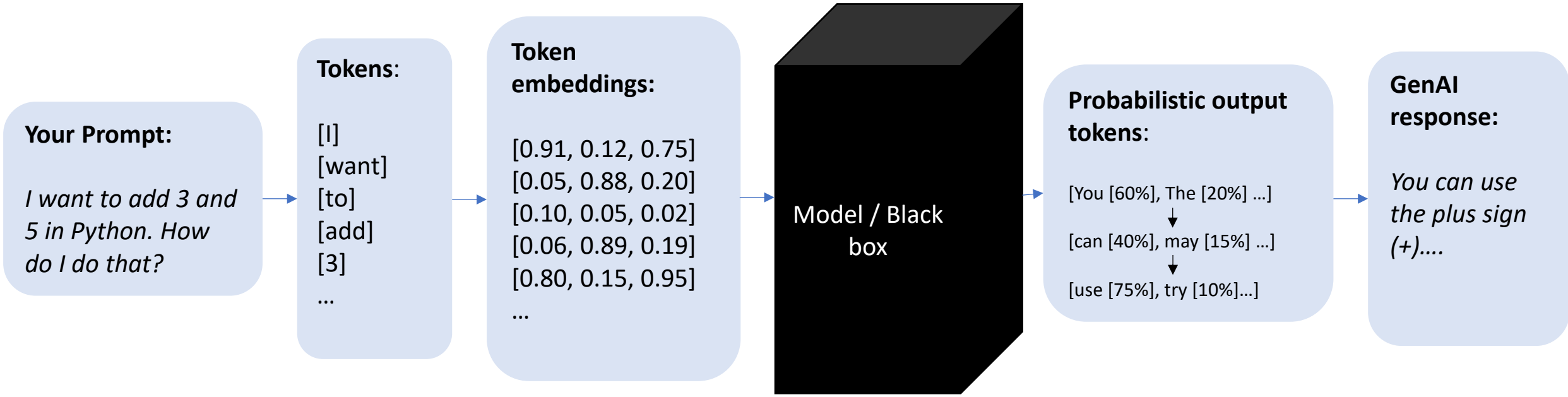
Generative AI is built to create **new content** such as text, images, audio, or video. It can also create new code by learning patterns from large datasets.

What is GenAI capable of creating?

- **Text generation**: writing, summarizing, answering questions, etc.
- **Code generation**: writing, explaining, improving code (we've seen this!)
- **Image editing**: modifying existing images, including tasks like reconstruction and inpainting
- **Text-to-image generation**: given a text description, create an image that depicts the prompt
- **Music generation**: composing original audio from a text prompt or style reference
- **Video generation**: creating (short) video clips from a text prompt or a still image



How does GenAI generate new content?



Training: GenAI models learn from massive datasets (billions of examples of text, images, or audio) to pick up statistical patterns: which words, pixels, or sounds tend to follow or relate to one another.

Generating: instead of just classifying or labeling something, the model uses those learned patterns to predict and produce *new* content, one piece at a time (e.g., one word, pixel, or sound frame after another).

How does GenAI create new content?

Text & code (**transformers**):

- Predict the next most likely word (token) given every token that came before it, one token at a time
- This is the architecture behind ChatGPT, Claude, and Copilot

Images, audio & video (**diffusion models**):

- Starts with random noise and gradually "denoises" it step by step, guided by the text prompt, until a coherent image (or sound/video) emerges.

Common thread: no "rules" are hand-coded in GenAI; the model learns purely from data and generates new examples that fit those learned patterns.

Transformers

What is a transformer?

A transformer is an architecture that can do **next-token prediction**; given everything before it, predict the most likely next word. It does this in three steps:



Text converts to tokens; tokens convert to embeddings (lists of numbers)

Which other words matter?

Create a probability distribution over the next token to be predicted

What does self-attention mean?

When the model reads a word, it doesn't look at that word alone; it looks back at every other word in the sentence and asks, "how much should I be paying attention to *you* right now?".

Example: "The cat jumped on the couch because it was **soft**." vs

"The cat jumped on the couch because it was **tired**."

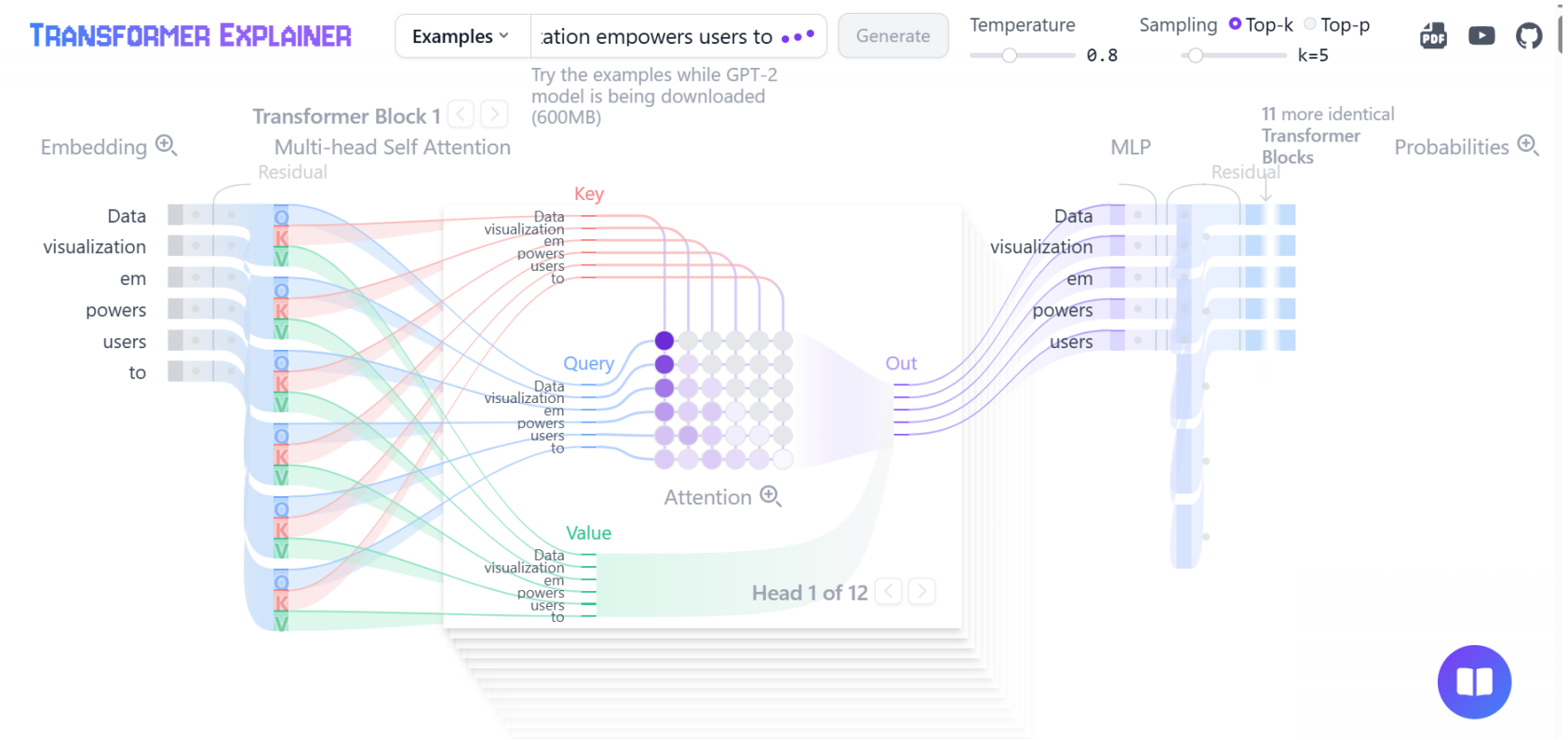
Same sentence, only the last word changes!

"Soft" is a word we associate with couches → the model boosts its attention on "couch".

"Tired" is a word we associate with living things → the model boosts its attention on "cat".

Example Transformer

Let's explore: https://poloclub.github.io/transformer-explainer/



What are LLMs?

Large language models (LLMs) are transformer-based models trained on massive amounts of text (much of the public internet, books, code, etc.).

They take the transformer architecture and scale it up, often adding extra fine-tuning steps (like reinforcement learning from human feedback) to make outputs more helpful, accurate, and safe.

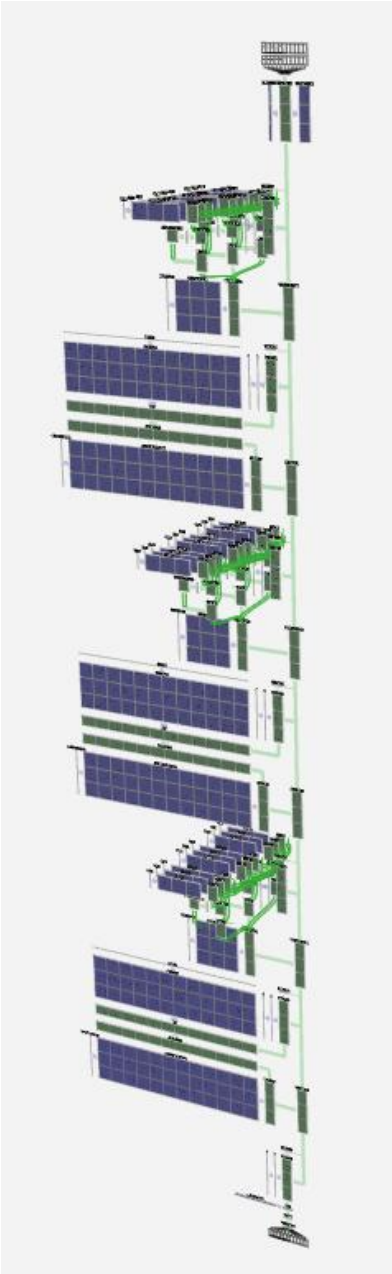
Because they're trained on such broad, general text, LLMs can handle many language tasks (i.e., writing, summarizing, translating, coding, answering questions) without being built for just one.

- That's a real shift from the **narrow, single-task systems** we saw earlier in the lecture (Deep Blue, Watson, AlphaGo)

Examples: ChatGPT (OpenAI), Claude (Anthropic), Gemini (Google), Llama (Meta).

Example LLMs:

Let's explore: https://bbycroft.net/llm



Model size and training cost

Models often are measured by **parameters**, which is the model's capacity to store patterns. More parameters generally means the model can represent more complex relationships in language (grammar, facts, reasoning).

Parameter counts have exploded:

- GPT-2 (2019) had 1.5 billion parameters
- GPT-3 (2020) scaled up to 175 billion parameters
- GPT-4 (2023) (reportedly) has ~1.8 trillion parameters

More parameters also mean more math the computer has to do during training, and since training cost is basically how much computing power you pay for, bigger models cost more to train.

Cost has scaled right alongside size:

- GPT-2 cost ~$43,000 to train
- GPT-3 cost ~$4-5 million
- GPT-4 cost ~$78-100+ million

More resources

Other resources for more reading:

- https://transformer-circuits.pub/2025/attribution-graphs/biology.html

You try: Transformer Visualization

- Work in groups of 2-3 students
- Visit: https://poloclub.github.io/transformer-explainer/

- At the top, choose one of the Examples or write your own prompt
  - Identify the next token the model predicts.
  - In your group, discuss the probability distribution of the other words.

- Now, change a single word in your prompt and re-generate
  - Did the predicted next token change? Can you think of why?

Diffusion Models

Why not use transformers for images, audio, video?

Text is **discrete and sequential**: a finite vocabulary of tokens, generated one at a time, where each token depends on the ones before it.

Images, audio, and video are **continuous and extremely high-dimensional.**

- A single image might have millions of pixel values, which need to be coherent *together* (a face needs two matching eyes, consistent lighting, etc.), not generated in a fixed left-to-right order.
- Generating images pixel-by-pixel (like text token-by-token) would be painfully slow and tends to lose global coherence; small local decisions don't add up to a realistic whole image.

Diffusion models take a fundamentally different generation strategy suited to this kind of data, which is why they (not standard text-transformers) power most image, and much audio/video, generation today.

Training a Diffusion Model

Core idea: start from pure random noise (like TV static) and gradually "denoise" it step by step until a coherent image emerges. This is closer to sculpting an image out of static than building it up piece by piece.

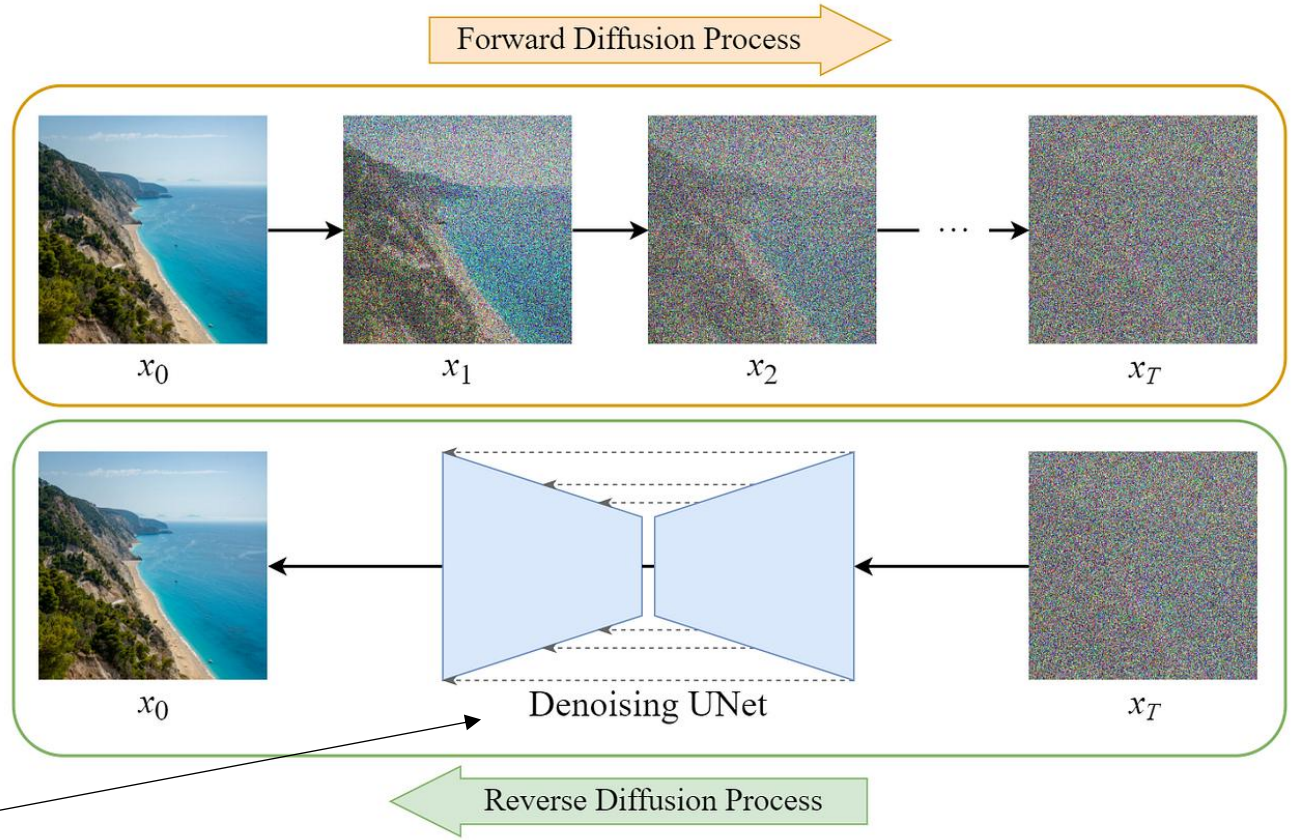
How can we train a model to do this? Start with lots of images, add noise to them to create labeled data, then train a model to **recognize noise in images**.

Once a model can recognize noise, it can **remove** that noise as well. That can be applied repeatedly to turn pure noise into an image.



How Diffusion Models Work

1. Adding noise to the image (no learning)



2. Removing noise from the image (learning)

An ML architecture (model)

https://medium.com/@steinsfu/diffusion-model-clearly-explained-cd331bd41166

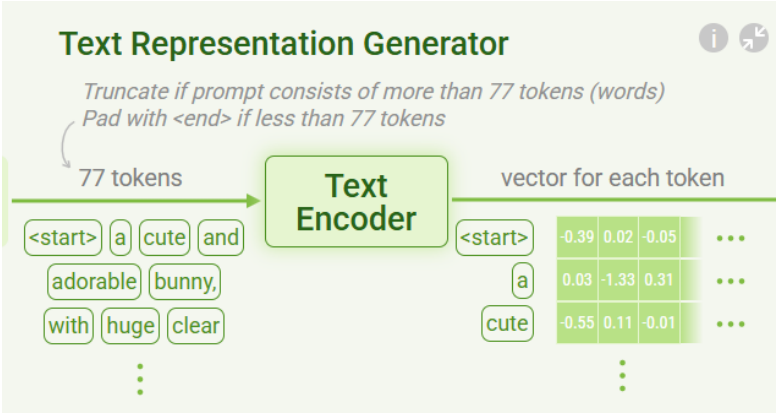
Matching Images to Prompts

How does the model match a text prompt to an image?

The initial dataset was formed of real images **paired with text descriptions**. Numerical encodings were created for both images & text in such a way that the encodings are related to each other.

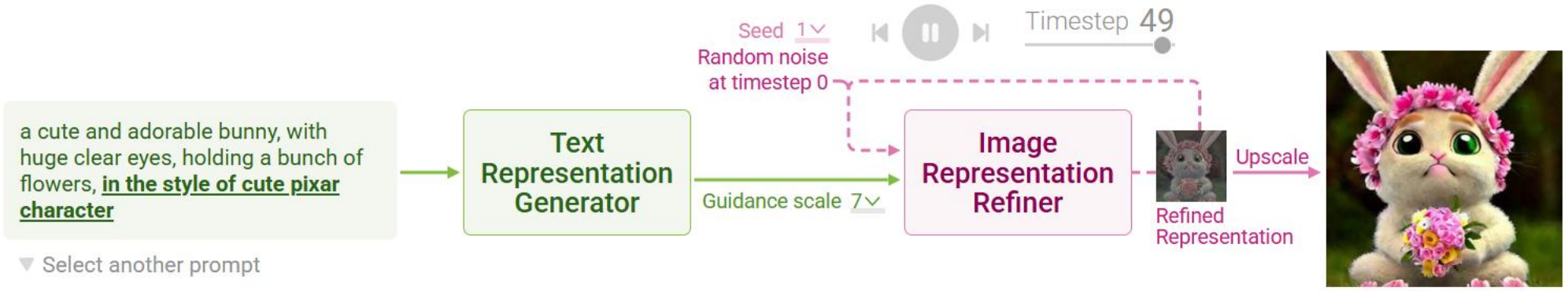
When an image is denoised, the direction is **guided by the description encoding**. That should lead towards an image similar to images with similar text descriptions in the training set.

Why it works so well: many small, gradual refinements (instead of generating everything in one shot) lets the model adjust its goal image over time while keeping it well-aligned with the prompt.



Examples:

- https://poloclub.github.io/diffusion-explainer/



- https://towardsdatascience.com/a-visual-guide-to-how-diffusion-models-work/

State-of-the-art image and video models (as of 2026)

- Image:
  - **GPT Image 2** (OpenAI) - https://developers.openai.com/api/docs/models/gpt-image-2
  - **Imagen 4 Ultra** (Google DeepMind) - https://deepmind.google/models/gemini-image/

- Video:
  - **Veo 3.1** (Google DeepMind) - https://deepmind.google/models/veo/
  - **Kling 3.0** (Kuaishou) - https://kling.ai/quickstart/klingai-video-3-model-user-guide

Sidebar: Reality of today's models

This transformer-vs-diffusion split isn't perfectly clean!

In reality, most state-of-the-art image/video models combine both ideas into a **diffusion transformer**.

Today's most advanced GenAI still comes from mixing and scaling the set of core ideas we have been discussing throughout the lecture.

Transformer vs Diffusion Models

	Transformer	Diffusion	Diffusion transformer (hybrid)
Best for	Text, code, sequential data	Images, audio, some video	State-of-the-art video and image generation
Generation Approach	Predicts next token step-by-step	Iteratively denoises random noise into output	Transformer attention used inside the diffusion denoising process
Data type	Discrete, sequential tokens	Continuous, high-dimensional pixels/waveforms	Continuous, long sequences
Example Models	GPT, Claude, Gemini, Llama	Stable Diffusion, Midjourney, DALL•E	Sora, Veo, HunyuanVideo

Activity: Visualize Diffusion Models

Go to the following blog: https://erdem.pl/2023/11/step-by-step-visual-introduction-to-diffusion-models

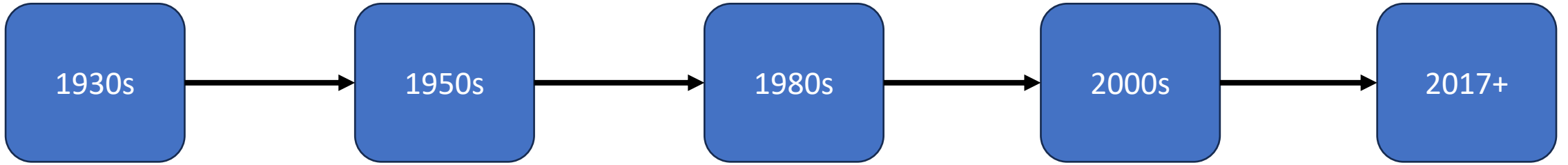
1. Go to the "Forward diffusion process" section
   - In Figure 2, walk step-by-step through the forward diffusion process. What do you notice?

2. Now, go to the "Reverse diffusion process" section
   - In Figure 9, walk step-by-step through the reverse diffusion process. What do you notice?

3. With a friend, compare how a diffusion model generates an image to how a transformer-based language model generates text, one token at a time.

Tomorrow might look different!

AI History: https://tingtingchung.github.io/aihistory/



Conceptual Foundation

Early AI Programs

Rise of Machine Learning

Big Data & Big Compute

GPT, LLMs, & Agentic AI

Learning Goals

- Differentiate between **Artificial Intelligence**, **Machine Learning**, and **Generative AI** as nested but distinct fields.

- Recognize how **transformers** and **diffusion models** are used by **generative artificial intelligence** to produce new text or media based on a prompt.