Types of Gen AI Models

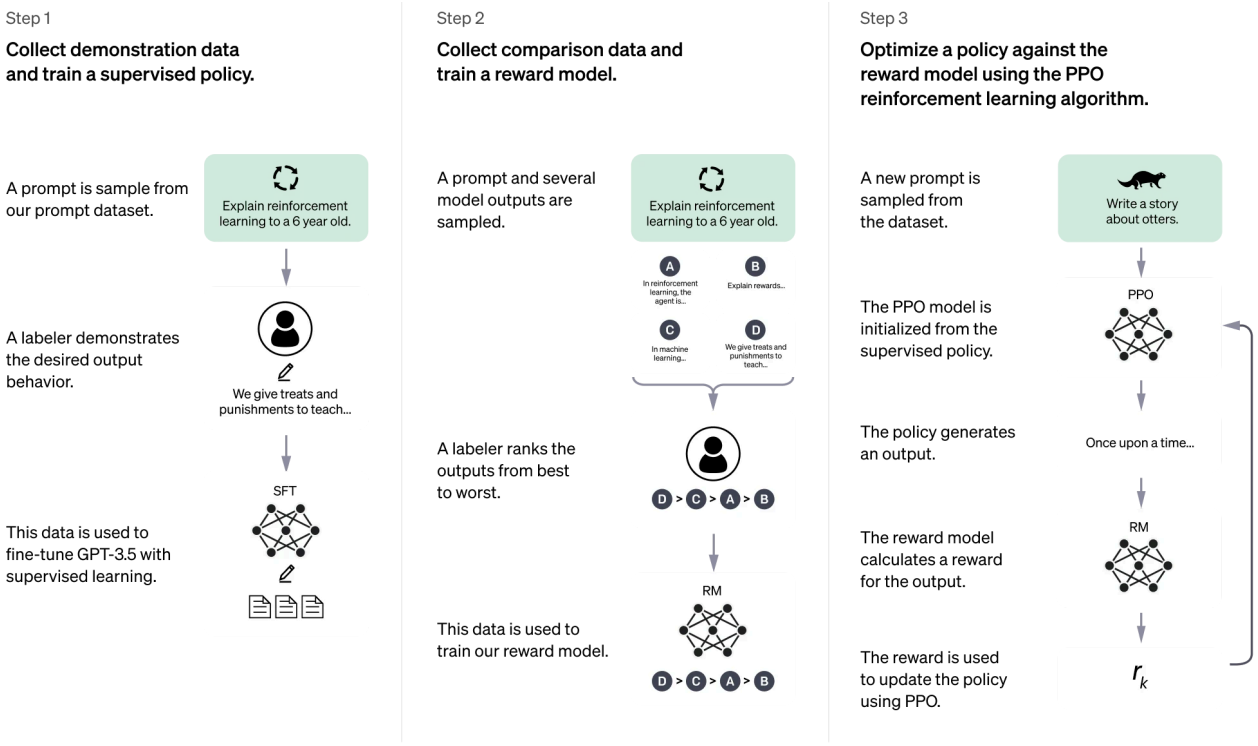
GPT (Generative Pre-trained Transformer)

GPT is a type of **Large Language Model (LLM)** primarily designed to **process and generate human-like text**. It's the foundational technology behind tools like ChatGPT

Key Details and Function:

Generative: It creates new content, rather than just classifying or summarizing existing data.
Pre-trained: It undergoes extensive initial training on a massive, diverse corpus of text and code from the internet (billions of data points). This allows it to learn grammar, facts, reasoning patterns, and various writing styles.
Transformer: This is the specific **neural network architecture** it uses. The Transformer is characterized by an **"attention mechanism"** which enables the model to weigh the importance of different words in the input text when generating the next word. This is crucial for maintaining long-range context and coherence.

How it Works



GPT generates text by predicting the most probable **next token** (word, sub-word, or character) in a sequence, based on the previous tokens it has seen. When you give it a prompt, it iteratively selects the next best token until it forms a complete, coherent response.

Use Cases:

Content creation (articles, emails, scripts), coding assistance, conversational chatbots, summarization, translation, and personalized learning.

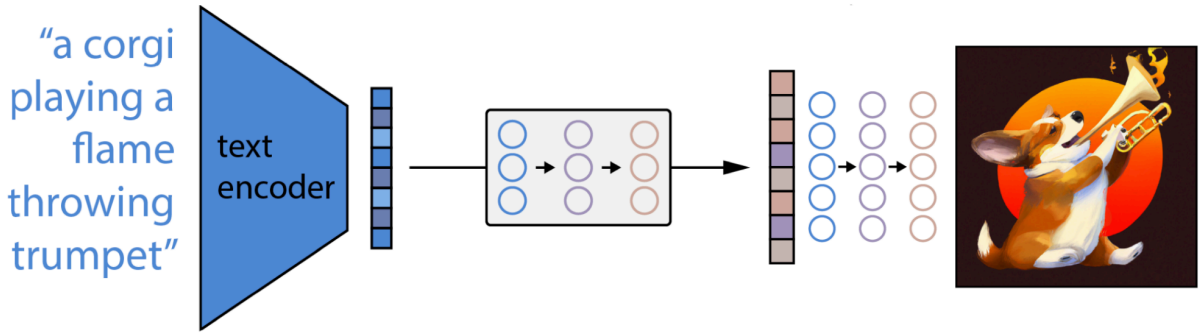
DALL-E

DALL-E is a powerful **text-to-image** generative AI model developed by OpenAI. Its purpose is to create novel digital images from natural language descriptions, or **prompts**.

Key Details & Function:

Text-to-Image Generation: The core function is to visualize and create an image that accurately and creatively represents the user's text prompt, even for surreal or non-existent concepts (e.g., "a sloth wearing a beret and painting a landscape").
Training Data: It's trained on a colossal dataset of **image-text pairs**, allowing it to learn the association between visual concepts and their linguistic descriptions

Architecture



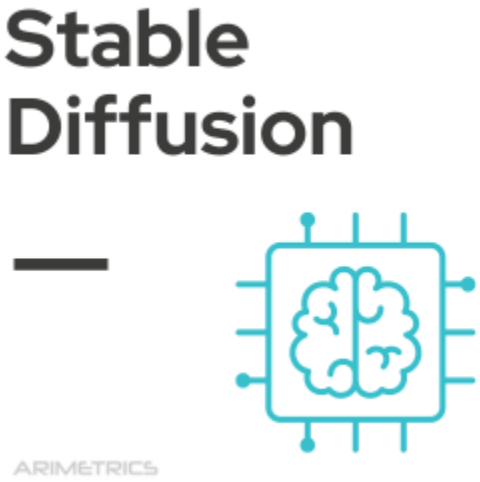
The model typically uses a multi-step process involving an encoder and a decoder.

1. **Text Encoding:** The text prompt is first encoded into a numerical representation (embedding) by a language model (often related to CLIP).
2. **Image Generation (Prior/Diffusion):** A model, often based on a **diffusion process** (like Stable Diffusion), uses this text embedding to create an initial, compressed image representation, often starting from noise.
3. **Decoding/Upscaling:** This compressed image is then decoded and upscaled to a high-resolution, final image.

Use Cases

Concept art, digital illustration, marketing visuals, unique visual aids for education, and general creative expression.

Stable Diffusion



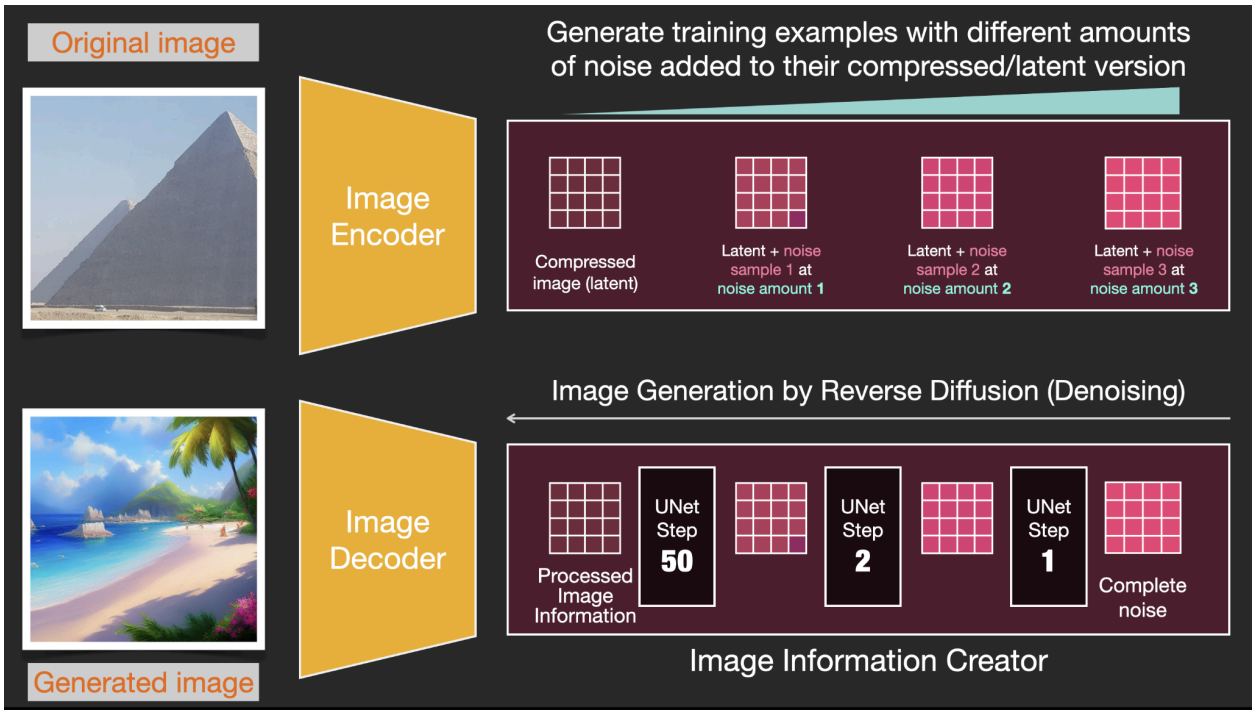
Stable Diffusion is another prominent **text-to-image** model, notable for being **open-source** and computationally efficient enough to run on consumer-grade hardware. It belongs to a class of models called **Latent Diffusion Models (LDMs)**.

Key Details & Function

Text-to-Image (and beyond): It generates images from text prompts but is also extensively used for **Image-to-Image** translation (modifying an existing image with a prompt) and inpainting/outpainting (editing specific parts or extending the canvas).

Latent Diffusion Model: This is its key architectural feature. Unlike earlier models that operated in the pixel space (millions of data points), Stable Diffusion operates in a much smaller, compressed dimensional space called the **latent space**. This significantly reduces the computational power and time required.

How it Works



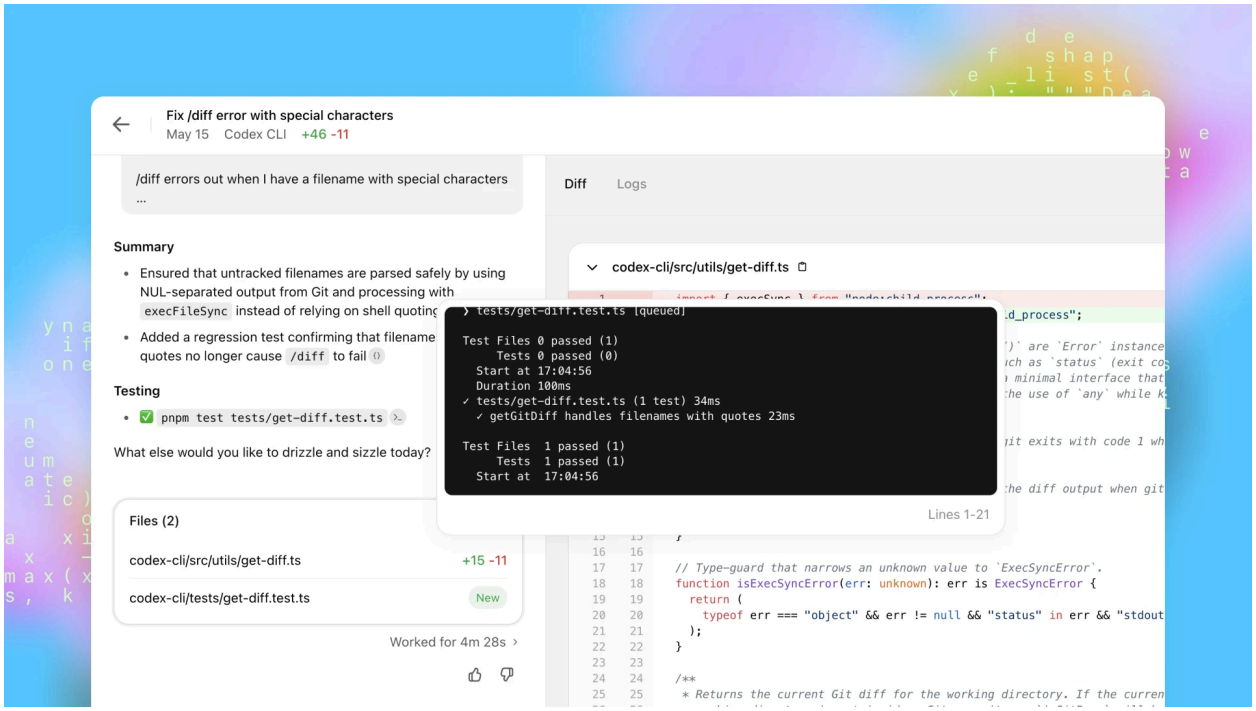
The generation process is an iterative **denoising** process:

1. **Start with Noise:** The process begins with an image of pure random noise in the latent space.
2. **Denoising:** A component called the **U-Net** iteratively predicts and subtracts the noise from the image over many steps. This denoising is guided by the text prompt's embedding (using a separate Text Encoder, like CLIP, to condition the process).
3. **Final Output:** Once the denoising steps are complete, a component called the **Decoder** converts the refined latent representation back into a full-resolution image.

Use Cases

Generative art, photo editing, creating custom assets for gaming and design, and rapid prototyping.

Codex



OpenAI Codex is a highly specialized family of AI models, developed by OpenAI, whose primary function is to **generate, understand, and interact with computer code.**

In its most recent iteration (powered by models like codex-1 from the o3 family), Codex has evolved beyond a simple code generator into a sophisticated software engineering agent.

Key Details & Functions:

Natural Language to Code Translation: Codex's key capability is translating instructions written in human language (e.g., "create a Python function that sorts a list and removes duplicates") into functional code in dozens of programming languages (most proficient in Python, JavaScript, etc.).

GPT-Based Foundation: The original Codex was a fine-tuned version of **GPT-3**—a Generative Pre-trained Transformer. This means its core mechanism is still sequential token prediction, but the tokens it predicts are heavily weighted toward programming syntax.

Specialized Training: It was trained on billions of lines of code, allowing it to learn the nuances, conventions, and common APIs of different languages far better than a general-purpose language model.

Use Cases:

Code Generation, Code Explanation, Debugging, Refactoring, Adding Tests