Evolution of Generative AI

From Origins to State of the Art

Dr. Muhammad Sajjad
RA: **Wajahat Ullah**
RA: **Kaleem Ullah**

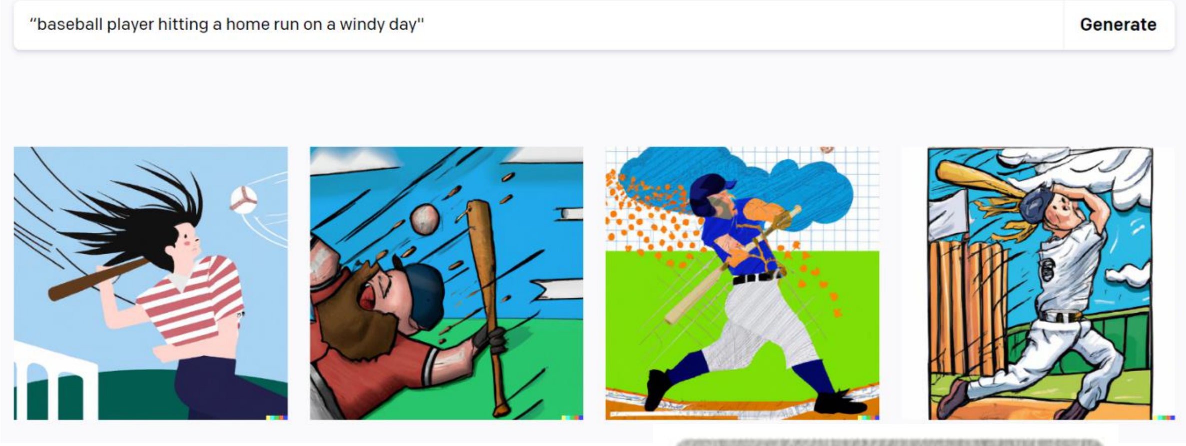
Agenda

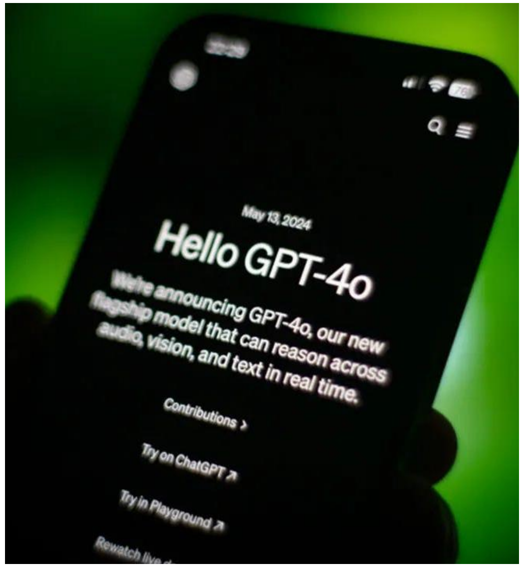
- Introduction to Generative AI
- History and evolution
- Core concepts and approaches
- Applications and use cases
- Prominent Generative AI models and tools
- Emerging trends and future outlook

What is Generative AI?

Generative AI is a branch of **A**rtificial **I**ntelligence that focuses on generating new data that resembles the structure and style of the training data.

- Unlike Discriminative AI which makes decision or predictions about new data based on existing data, Generative AI can generate entirely new data that it has not seen before.
- By learning the underlying patterns and distributions in the training data.
- The inputs and outputs can be in various forms like text, audio, video, 3D models, etc.





The image you uploaded shows a simple, hand-drawn character with a surprised expression, reaching towards a sparkling diamond.

The Origins of Generative AI

Generative AI has its roots dating back several decades, with early explorations in fields like statistics, probability, and information theory.

- **Early days of Generative AI, 1950s:** Statistical models, like **Markov Chains**, were capable of generating new sequences based on input sequence.
- **The Rise of Neural Networks, 1980s:** Neural Networks were used to generate new data. Tools like **Boltzmann Machines** were developed to generate new data

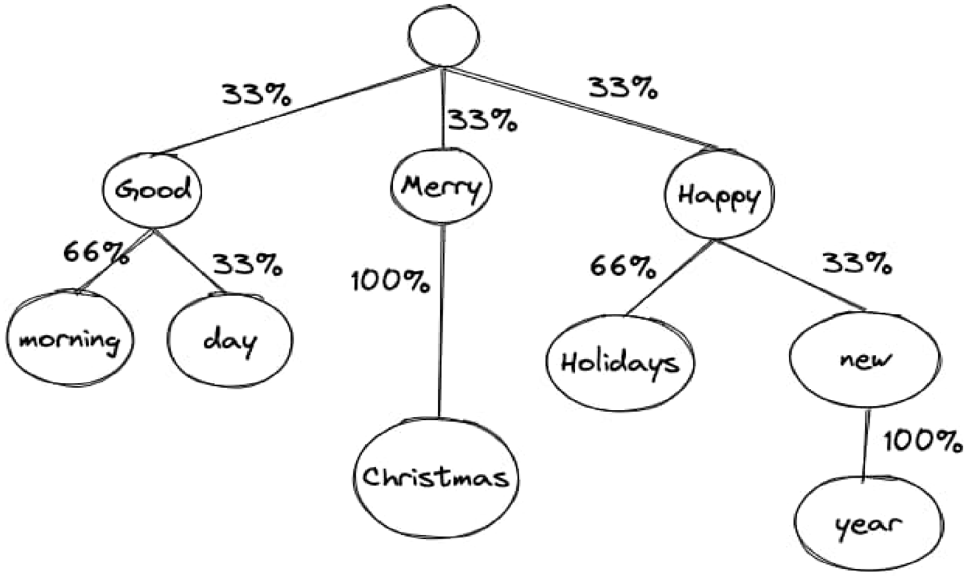


Figure: Markov Chains

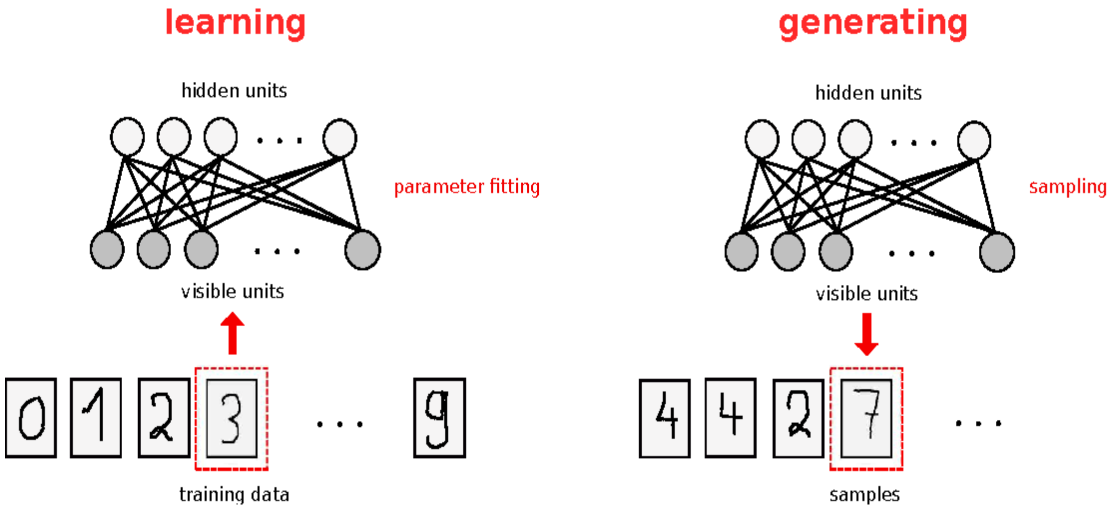


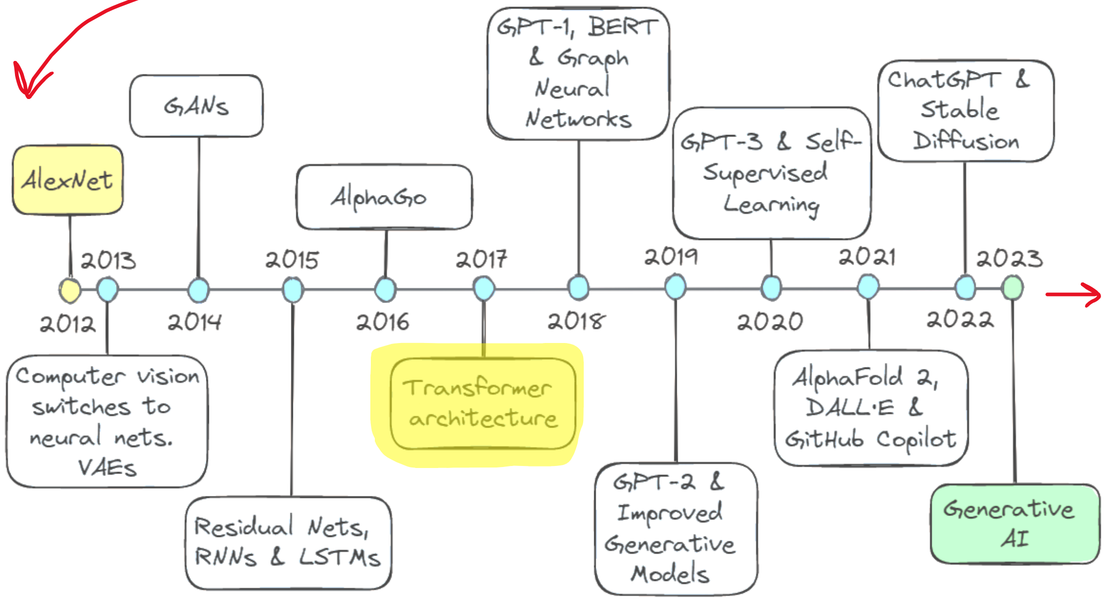
Figure: Boltzmann Machines

Evolution of Generative AI

In the early 2010s, Generative AI witnessed a significant leap with the breakthrough of deep learning and introduction of powerful techniques for generative modeling.

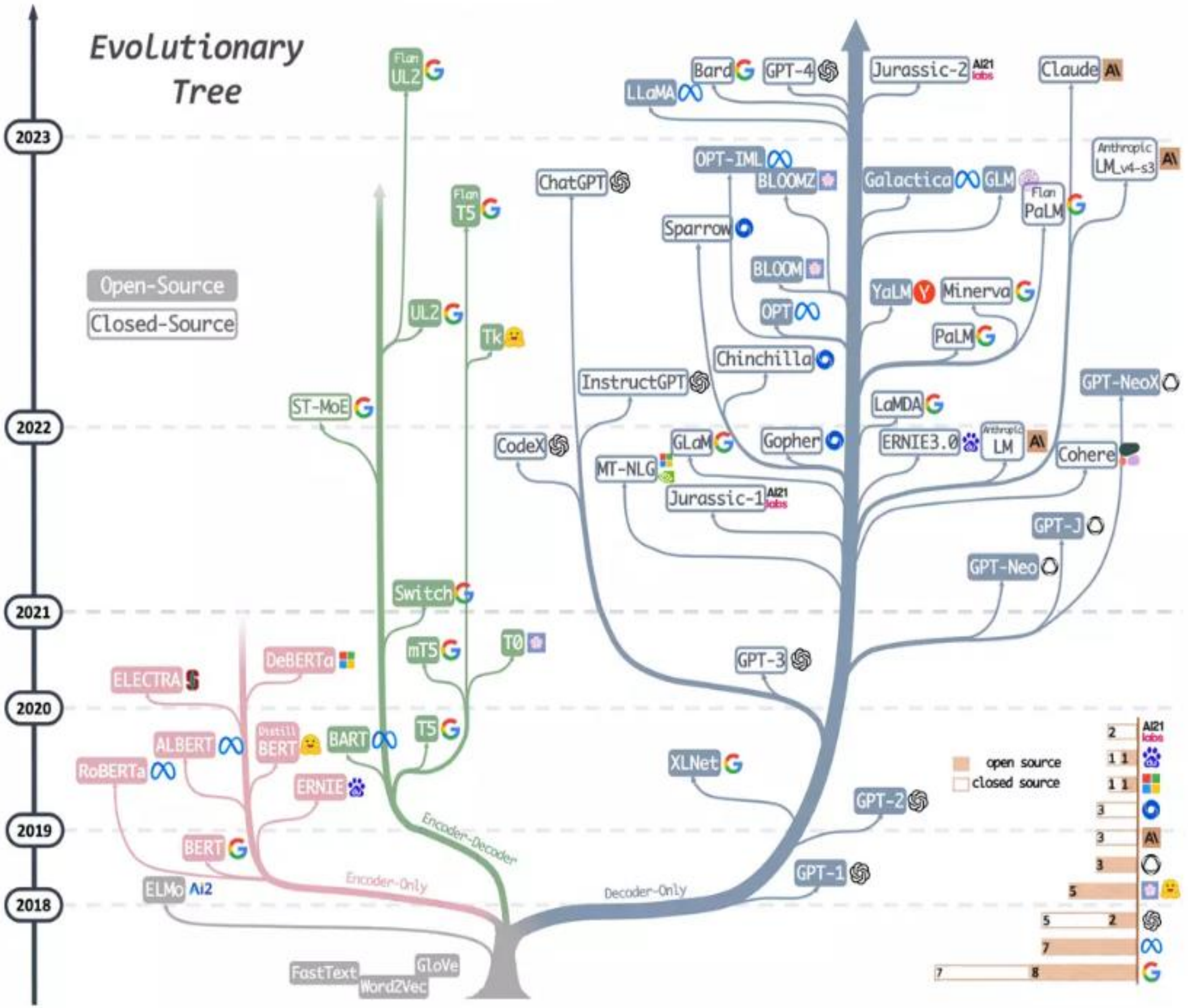
ILSVRC

- **VAEs** and **GANs** emerged as two prominent generative modeling approaches.
- VAEs learn compressed representations of data from which it generates new data.
- GANs use adversarial training to improve the quality of the generated data.
- Now, architectures based on **Transformer**, and **Diffusion models** are continuously pushing the boundaries of Generative AI.



Evolution of Generative AI
2024 Era:

- **OpenAI Sora:** A generative text-to-video model that combines diffusion models with transformers.
- **Gemini 1.5:** Developed by Google DeepMind, a large language model for various applications.
- **Llama 3:** Released by Meta AI, and is available in 8 billion and 70 billion parameter versions.
- **Gen2:** By RunwayML, a text-to-video generation tool
- **Code Llama:** Meta's model which is focused on code generation



Core Concepts and Approaches

Generative AI models are built upon several fundamental concepts and techniques

- **Unsupervised learning:** Generative models learn from unlabeled data in unsupervised or semi-supervised manner to capture the underlying distributions of the data.

- **Latent variable models:** Some models also use latent variables to learn compressed representation of input data which is then used to generate new data.

- **Adversarial training:** models like GANs use adversarial training, where a generator and discriminator model compete against each other.

- **Autoregressive models:** models like RNNs and Transformers generate data sequentially, predicting one element at a time based on previous outputs.

- **Diffusion Models:** These are new class of generative models that can generate high fidelity outputs particularly in image generation.

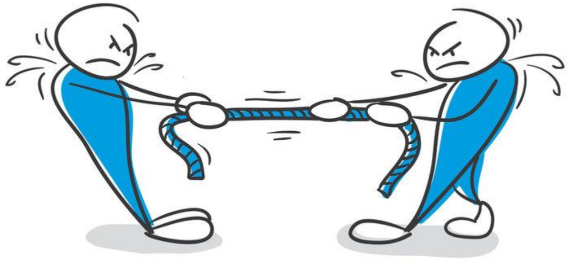


Figure: Adversarial training

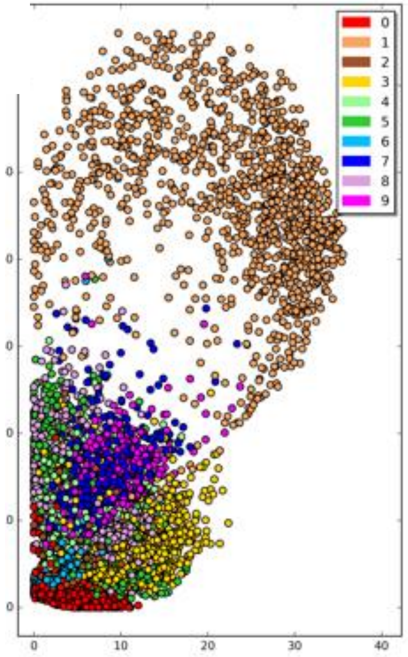


Figure: Latent variables

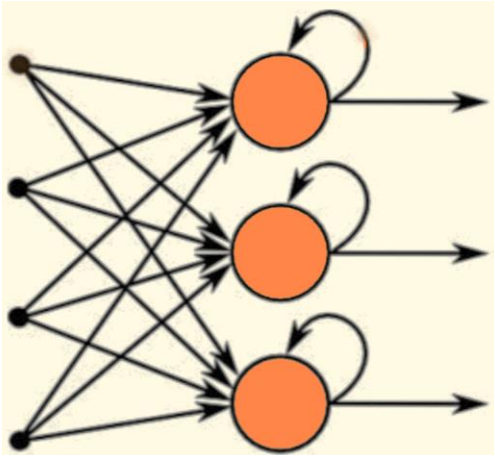
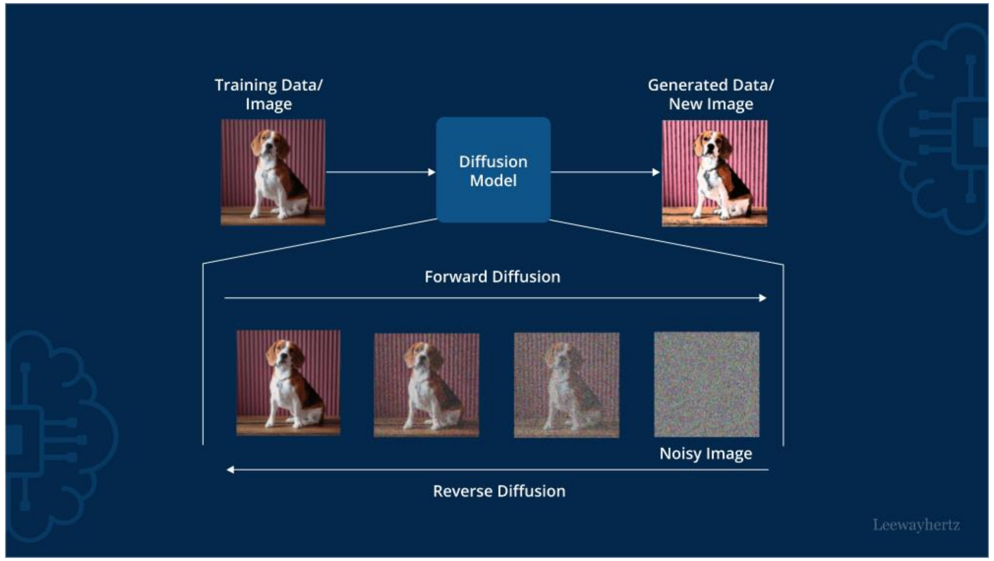


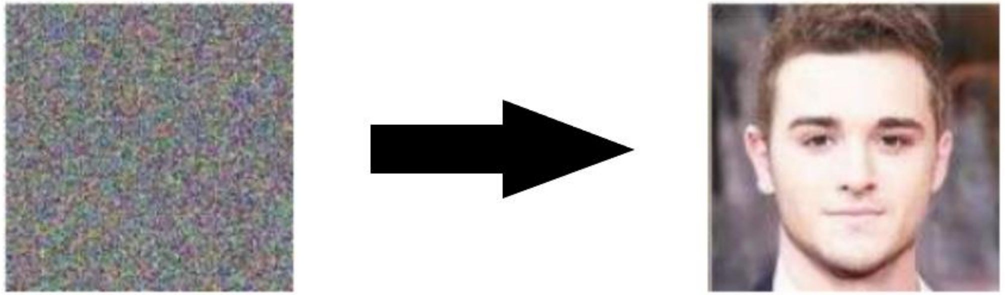
Figure: Autoregressive models

Diffusion Models

A class of generative models introduced in 2015, that learn to generate data by reversing a gradual noise process.

- **Input:** These models takes as input a random noise distribution, typically from a Gaussian distribution.
- **Output:** Generate images that resemble original data distribution by iteratively removing the noise.





Diffusion Models

There are two main processes in the diffusion models.

- **Forward Diffusion:** In this process, a small amount of noise from a predefined distribution is added to the sample image iteratively.

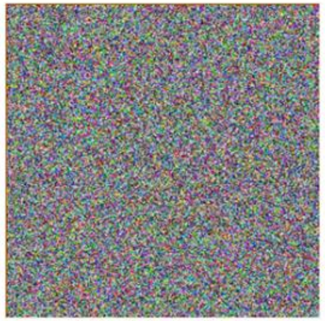
t=0

Diffusion Models

There are two main processes in the diffusion models.

- **Forward Diffusion:** In this process, a small amount of noise from a predefined distribution is added to the sample image iteratively.

t=0

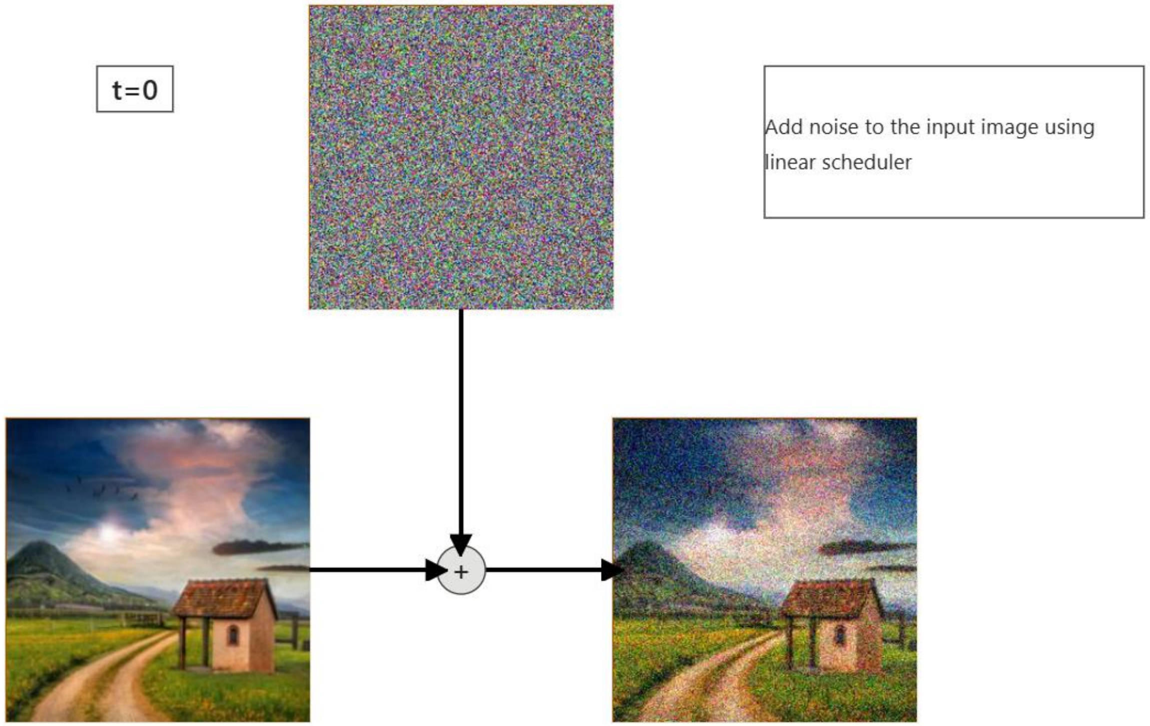


Generate noise for current step **t**

Diffusion Models

There are two main processes in the diffusion models.

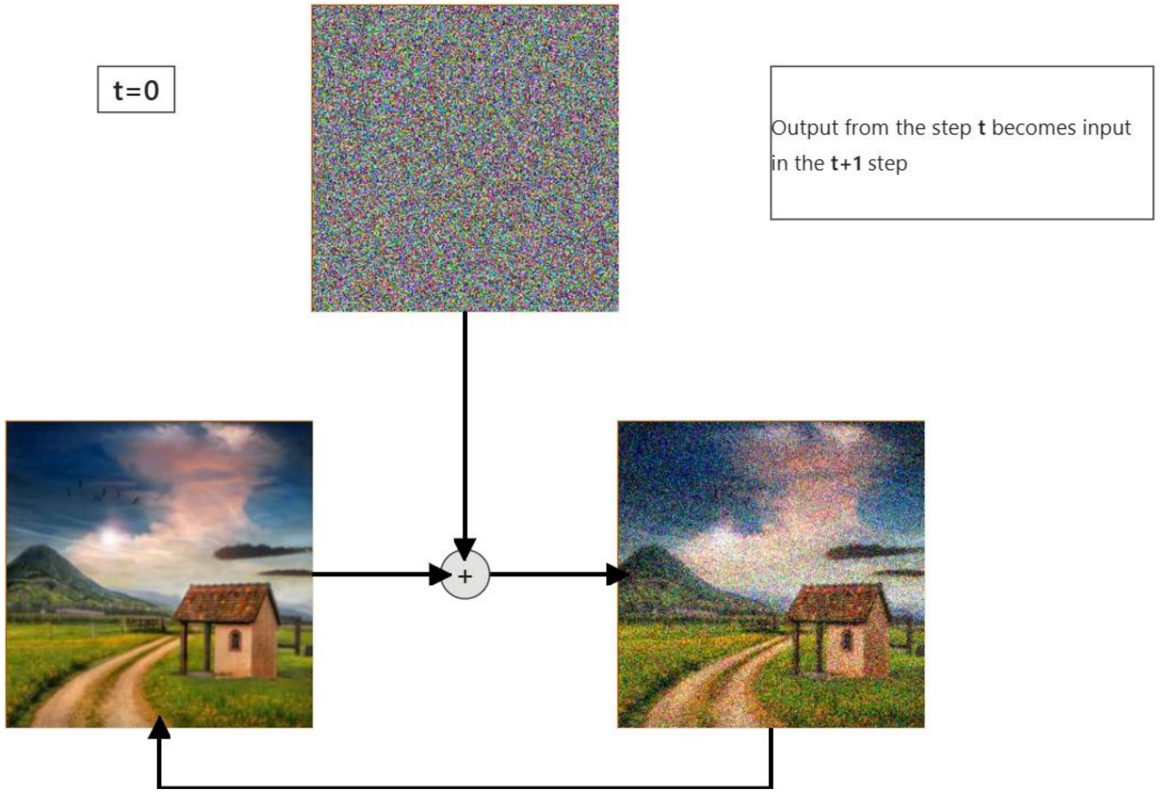
- **Forward Diffusion:** In this process, a small amount of noise from a predefined distribution is added to the sample image iteratively.



Diffusion Models

There are two main processes in the diffusion models.

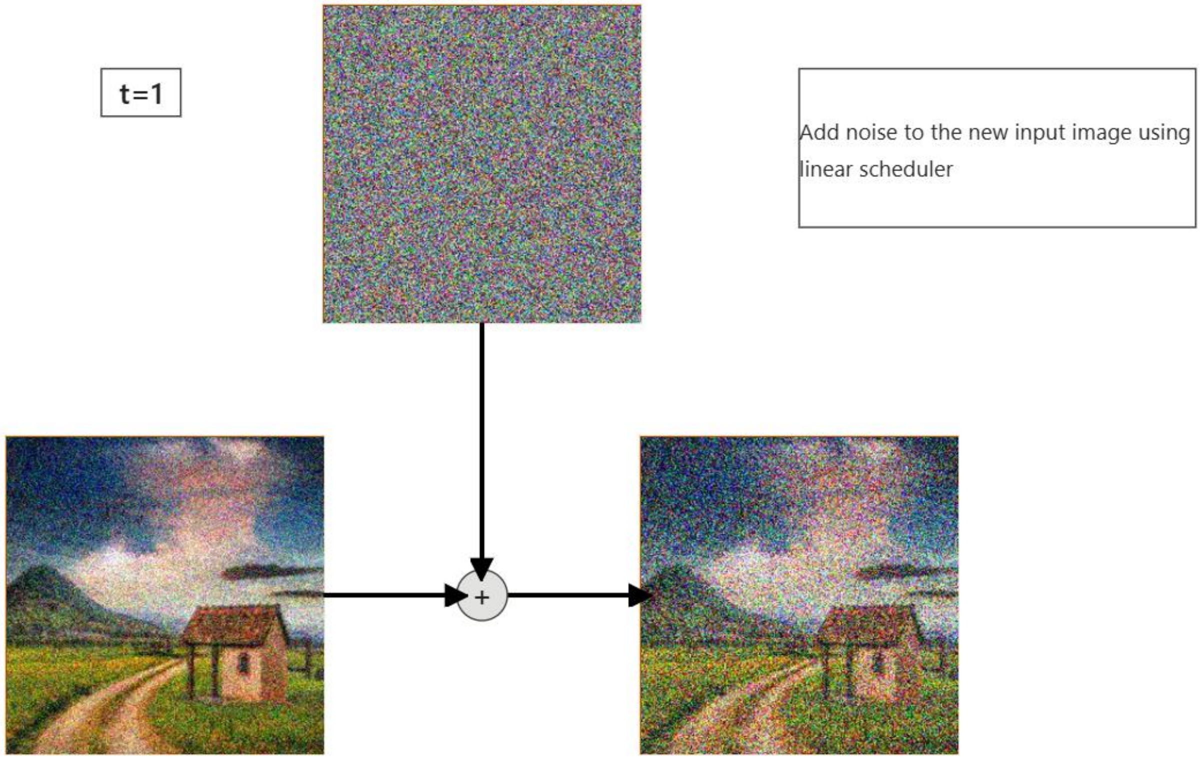
- **Forward Diffusion:** In this process, a small amount of noise from a predefined distribution is added to the sample image iteratively.



Diffusion Models

There are two main processes in the diffusion models.

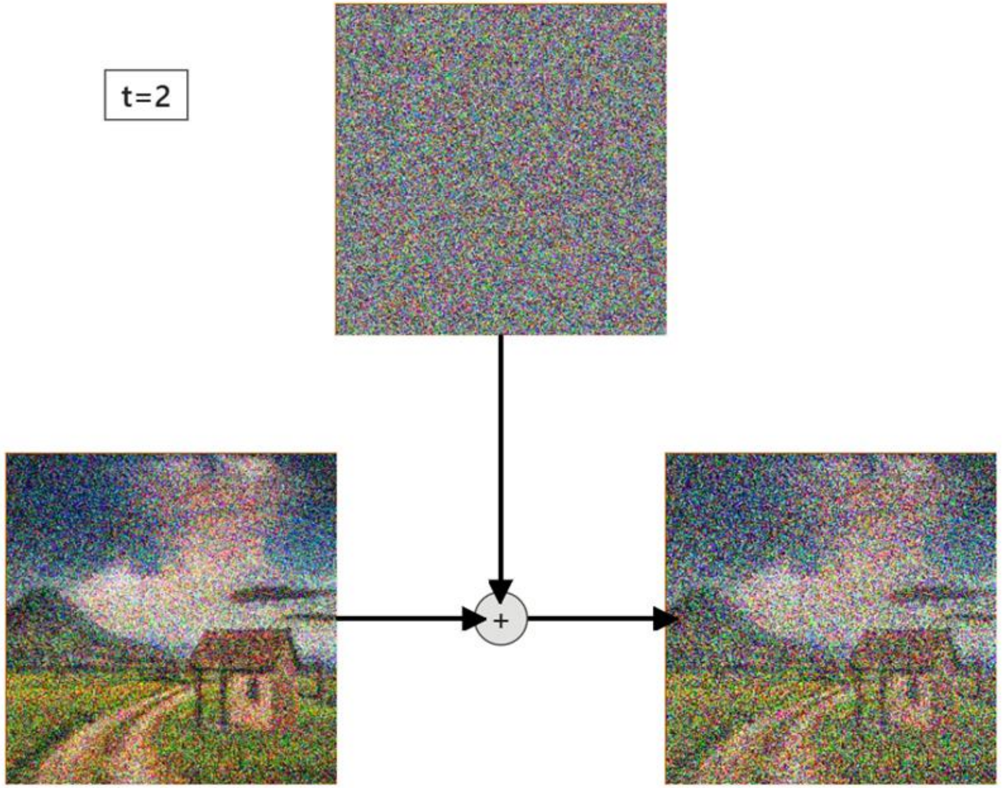
- **Forward Diffusion:** In this process, a small amount of noise from a predefined distribution is added to the sample image iteratively.



Diffusion Models

There are two main processes in the diffusion models.

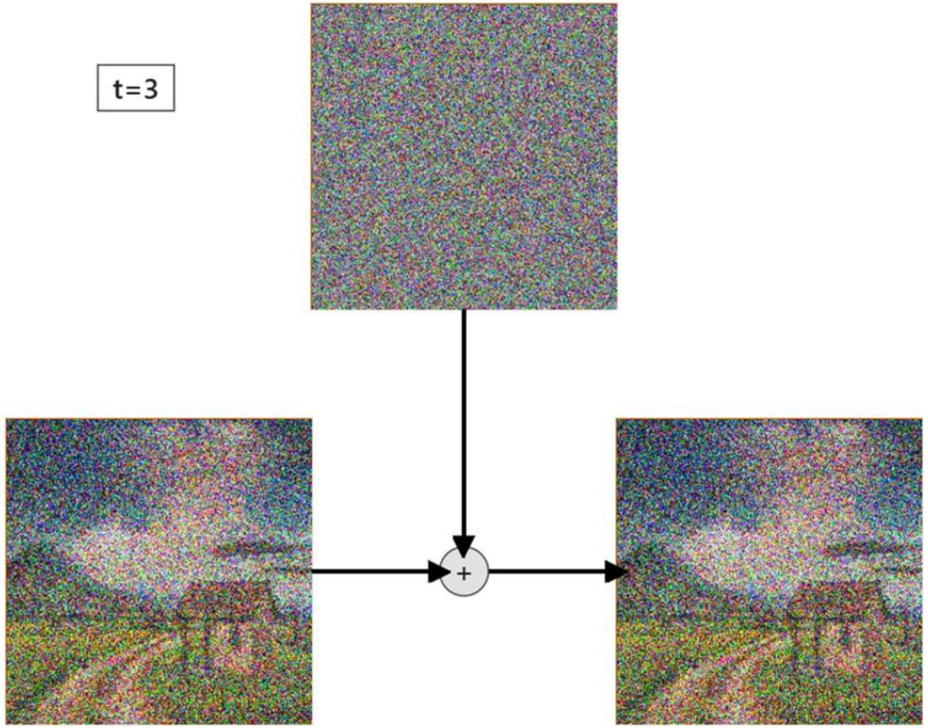
- **Forward Diffusion:** In this process, a small amount of noise from a predefined distribution is added to the sample image iteratively.



Diffusion Models

There are two main processes in the diffusion models.

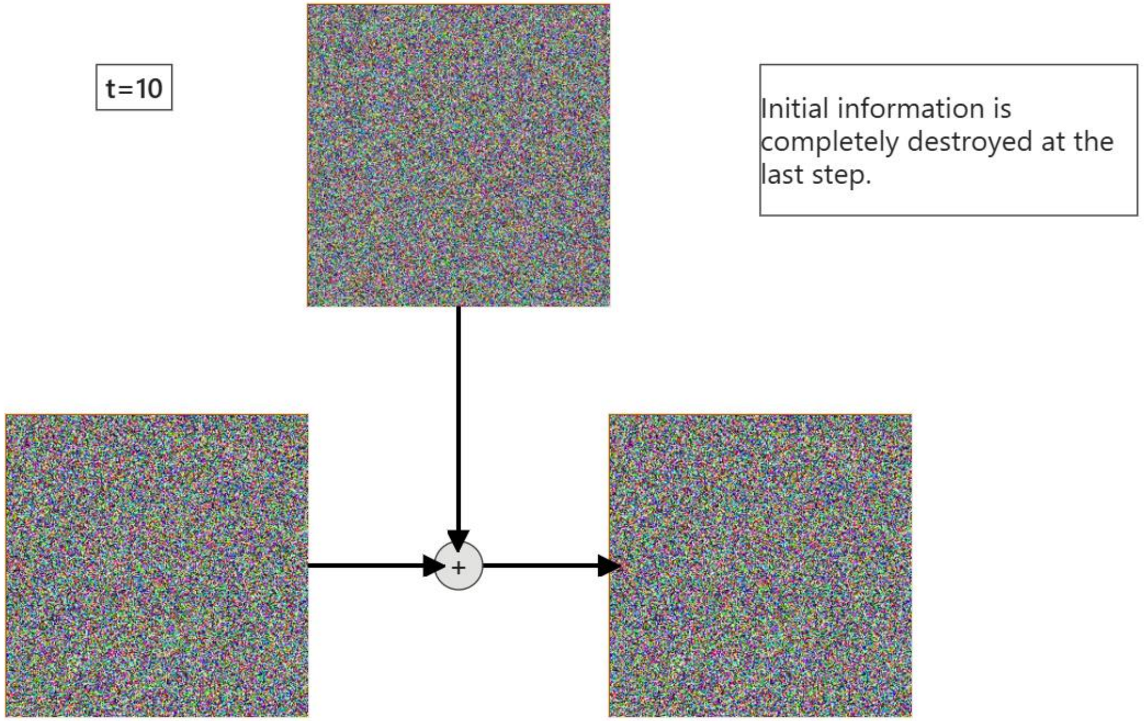
- **Forward Diffusion:** In this process, a small amount of noise from a predefined distribution is added to the sample image iteratively.



Diffusion Models

There are two main processes in the diffusion models.

- **Forward Diffusion:** In this process, a small amount of noise from a predefined distribution is added to the sample image iteratively.



Diffusion Models

There are two main processes in the diffusion models.

- **Forward Diffusion:** In this process, a small amount of noise from a predefined distribution is added to the sample image iteratively.

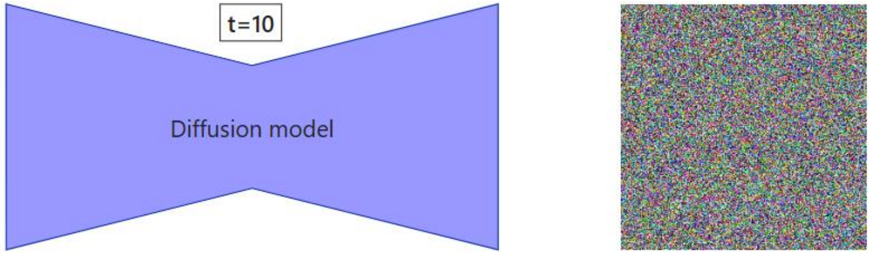
*Entire forward diffusion process using **linear schedule***

*Forward diffusion process using **cosine schedule**.*

Diffusion Models

There are two main processes in the diffusion models.

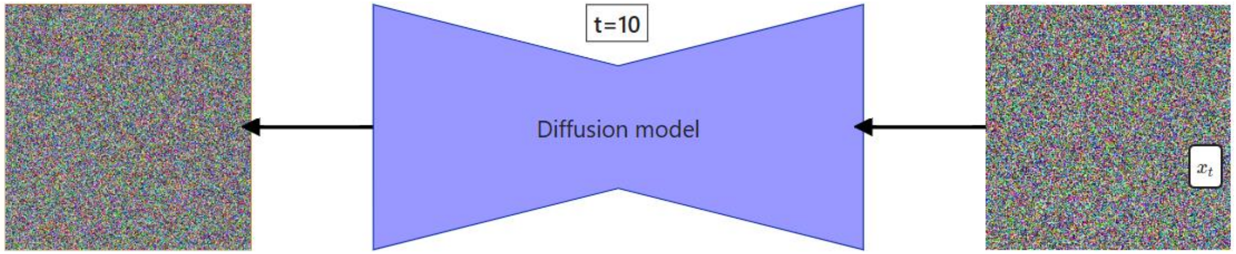
- **Reverse Diffusion:** In this process, the noise is iteratively removed from the diffused image to reconstruct the original image.



Diffusion Models

There are two main processes in the diffusion models.

- **Reverse Diffusion:** In this process, the noise is iteratively removed from the diffused image to reconstruct the original image.

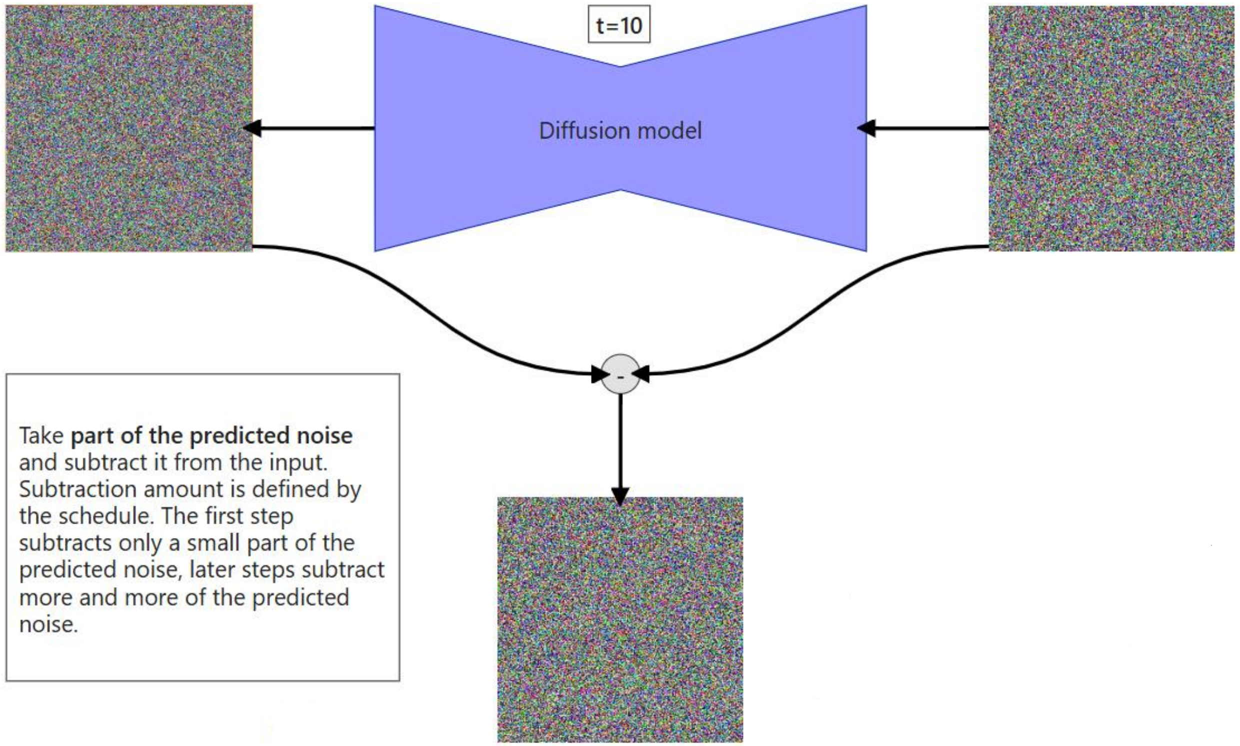


Notice!
Output of the model is an **entire noise** that the model predicted to be removed from the input. This noise is later scaled based on the schedule to be subtracted from the input.

Diffusion Models

There are two main processes in the diffusion models.

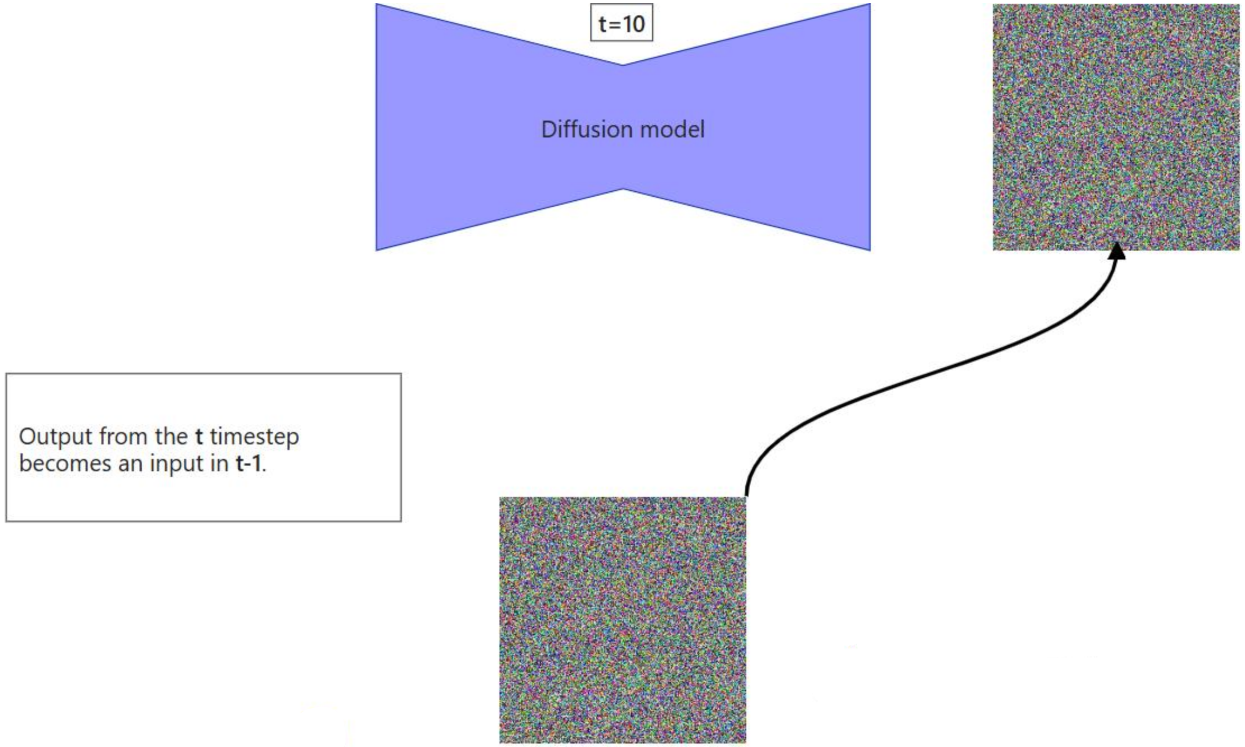
- **Reverse Diffusion:** In this process, the noise is iteratively removed from the diffused image to reconstruct the original image.



Diffusion Models

There are two main processes in the diffusion models.

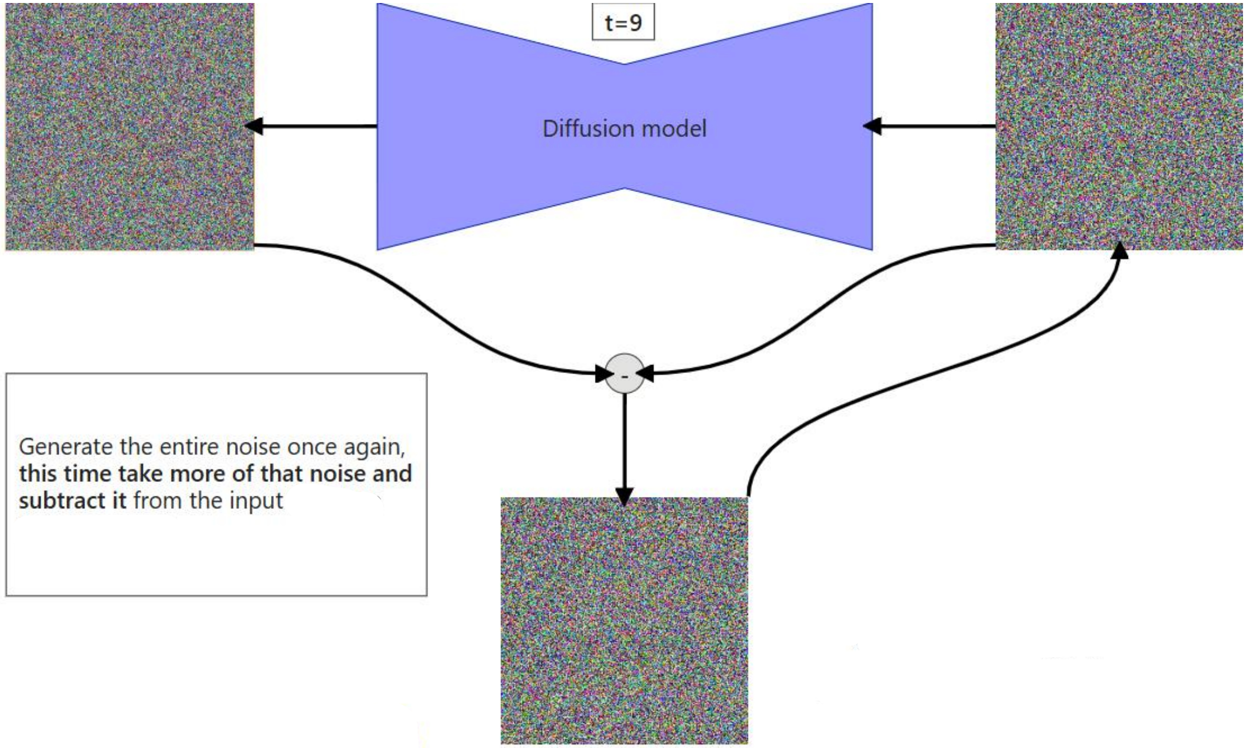
- **Reverse Diffusion:** In this process, the noise is iteratively removed from the diffused image to reconstruct the original image.



Diffusion Models

There are two main processes in the diffusion models.

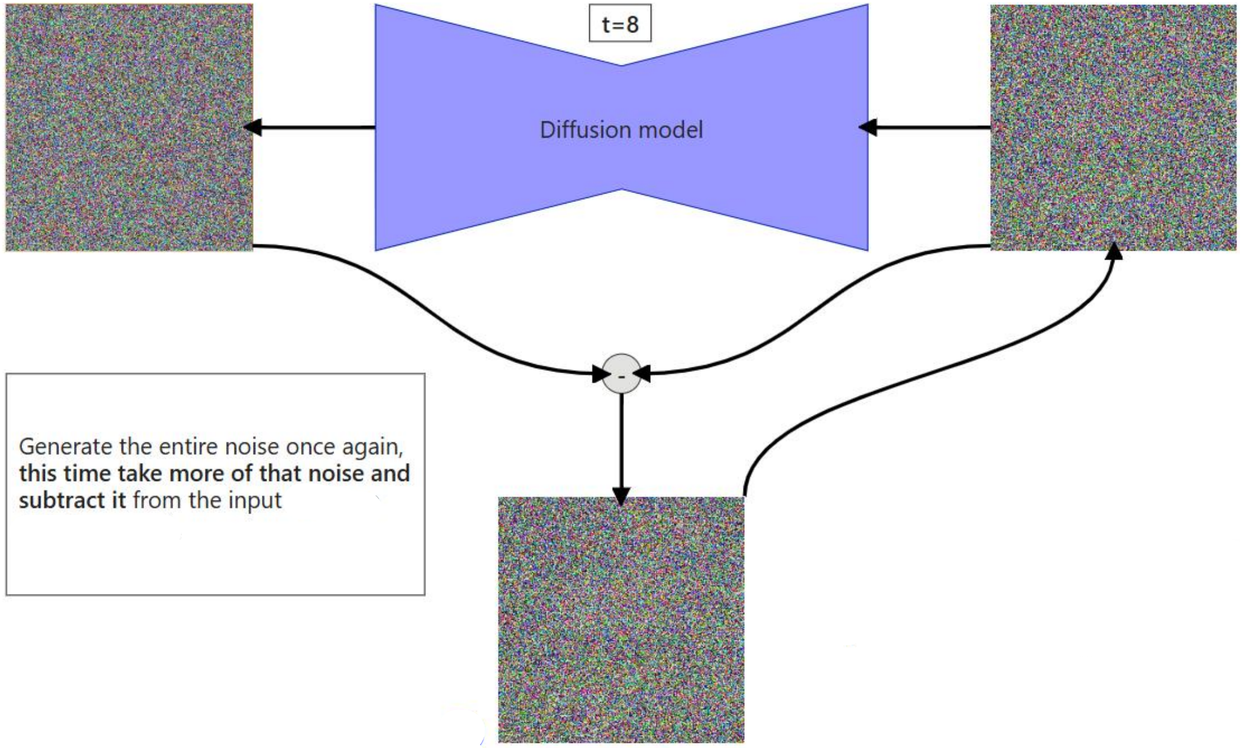
- **Reverse Diffusion:** In this process, the noise is iteratively removed from the diffused image to reconstruct the original image.



Diffusion Models

There are two main processes in the diffusion models.

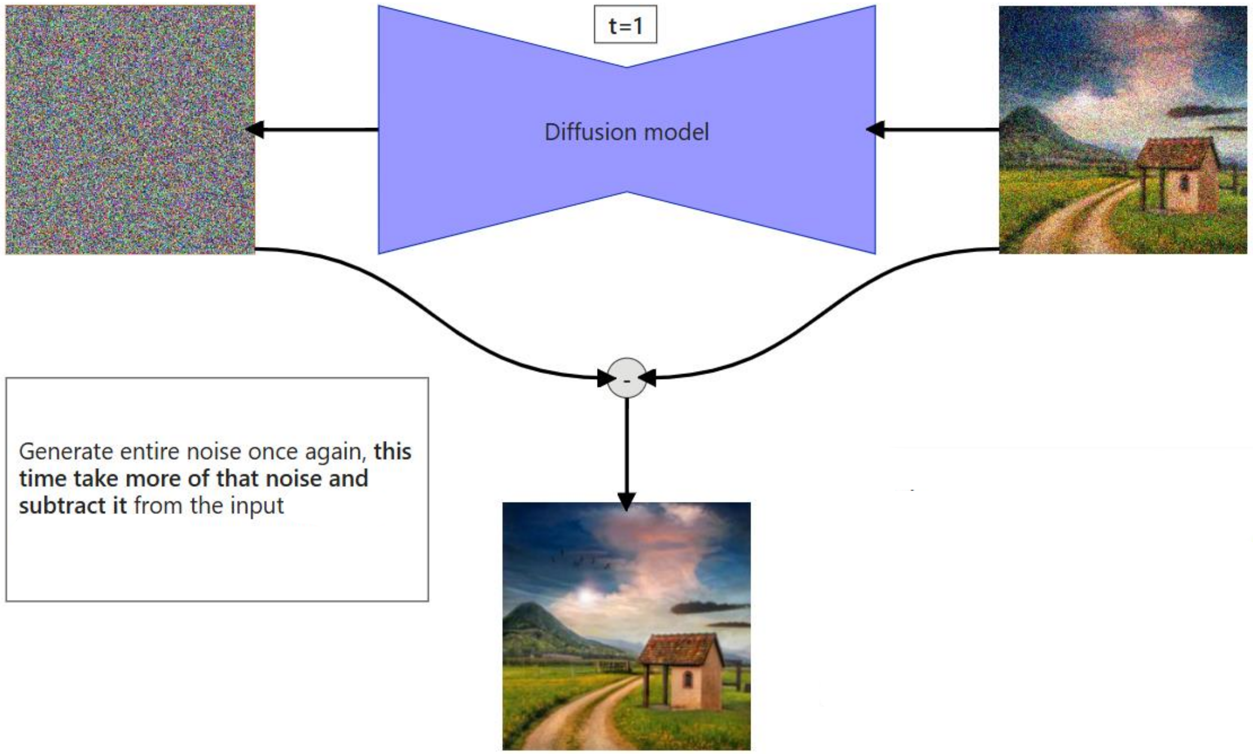
- **Reverse Diffusion:** In this process, the noise is iteratively removed from the diffused image to reconstruct the original image.



Diffusion Models

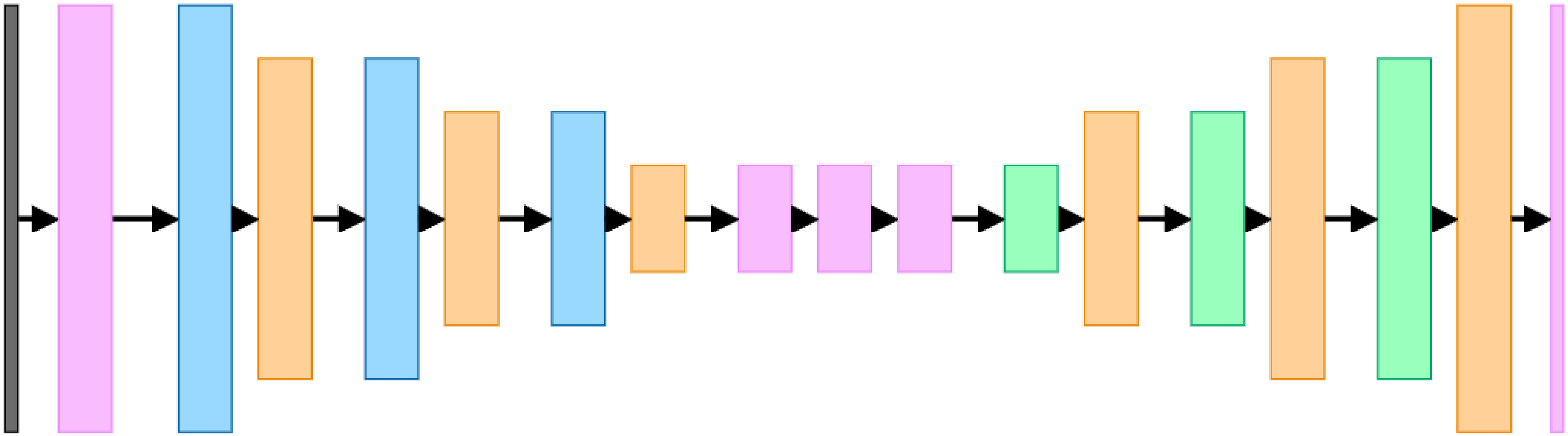
There are two main processes in the diffusion models.

- **Reverse Diffusion:** In this process, the noise is iteratively removed from the diffused image to reconstruct the original image.



Diffusion Models

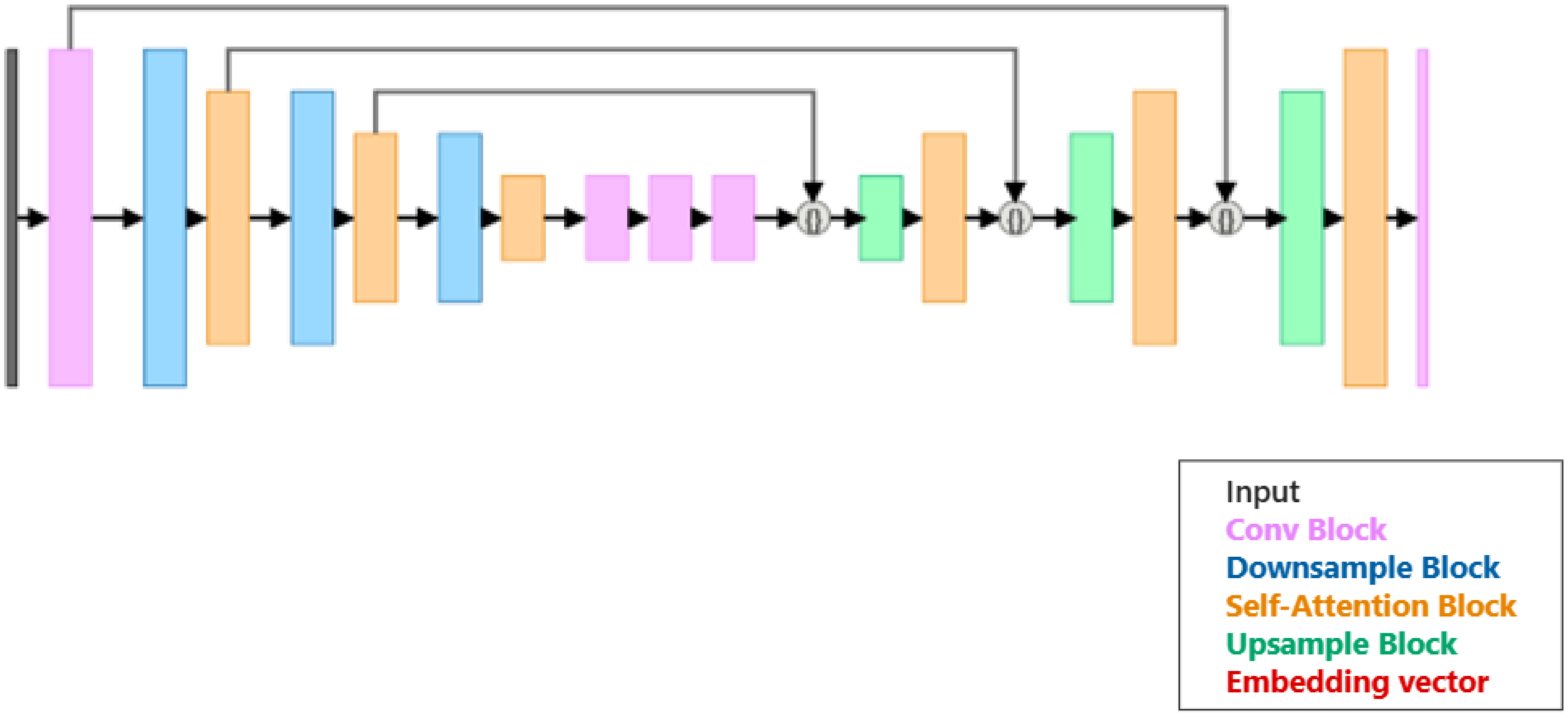
The original architecture that was used for predicting noise to be removed was a modified version of U-Net architecture. We can see that self-attention blocks have also been added.



Input
Conv Block
Downsample Block
Self-Attention Block
Upsample Block
Embedding vector

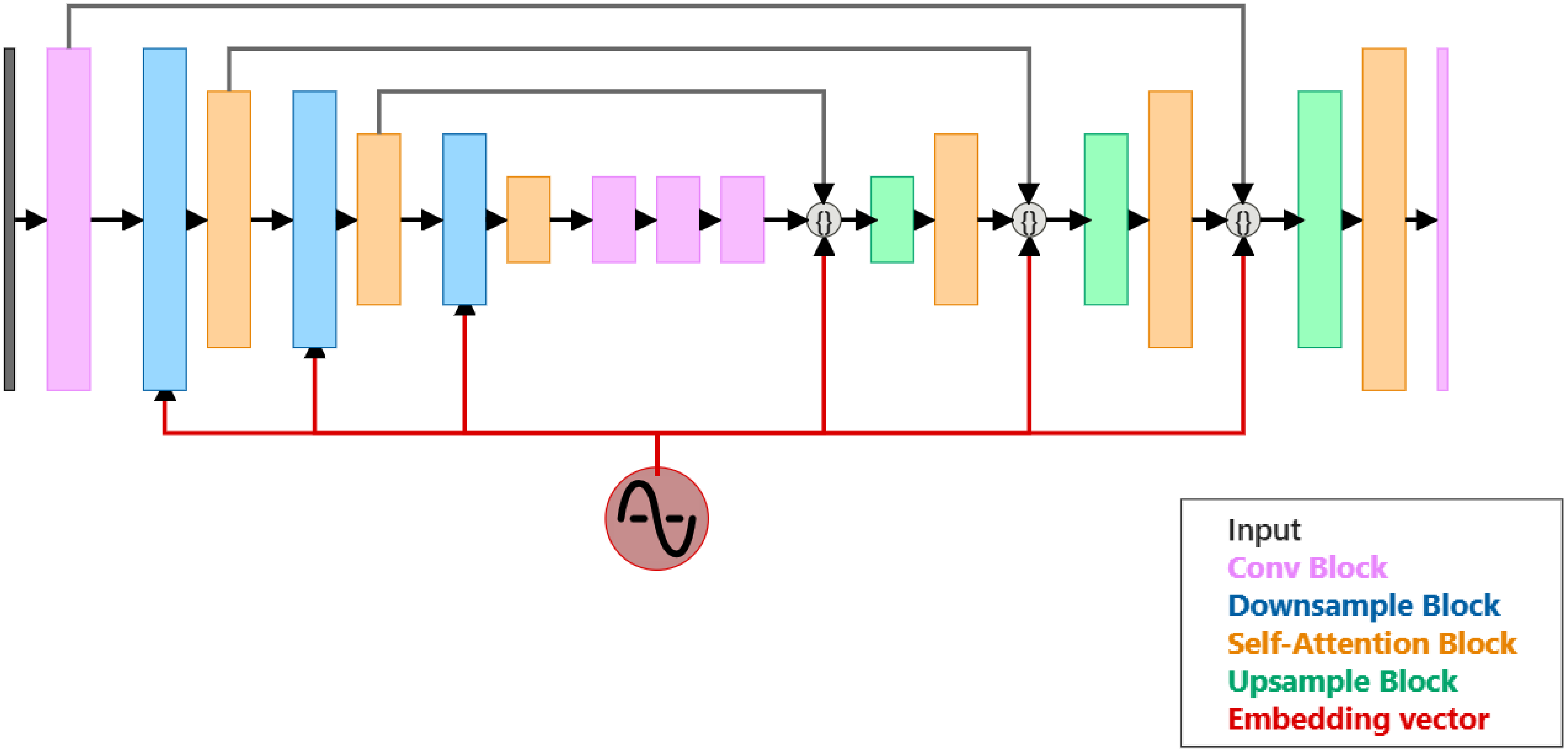
Diffusion Models

To prevent information loss, skip connections are added to the up sampling layers.



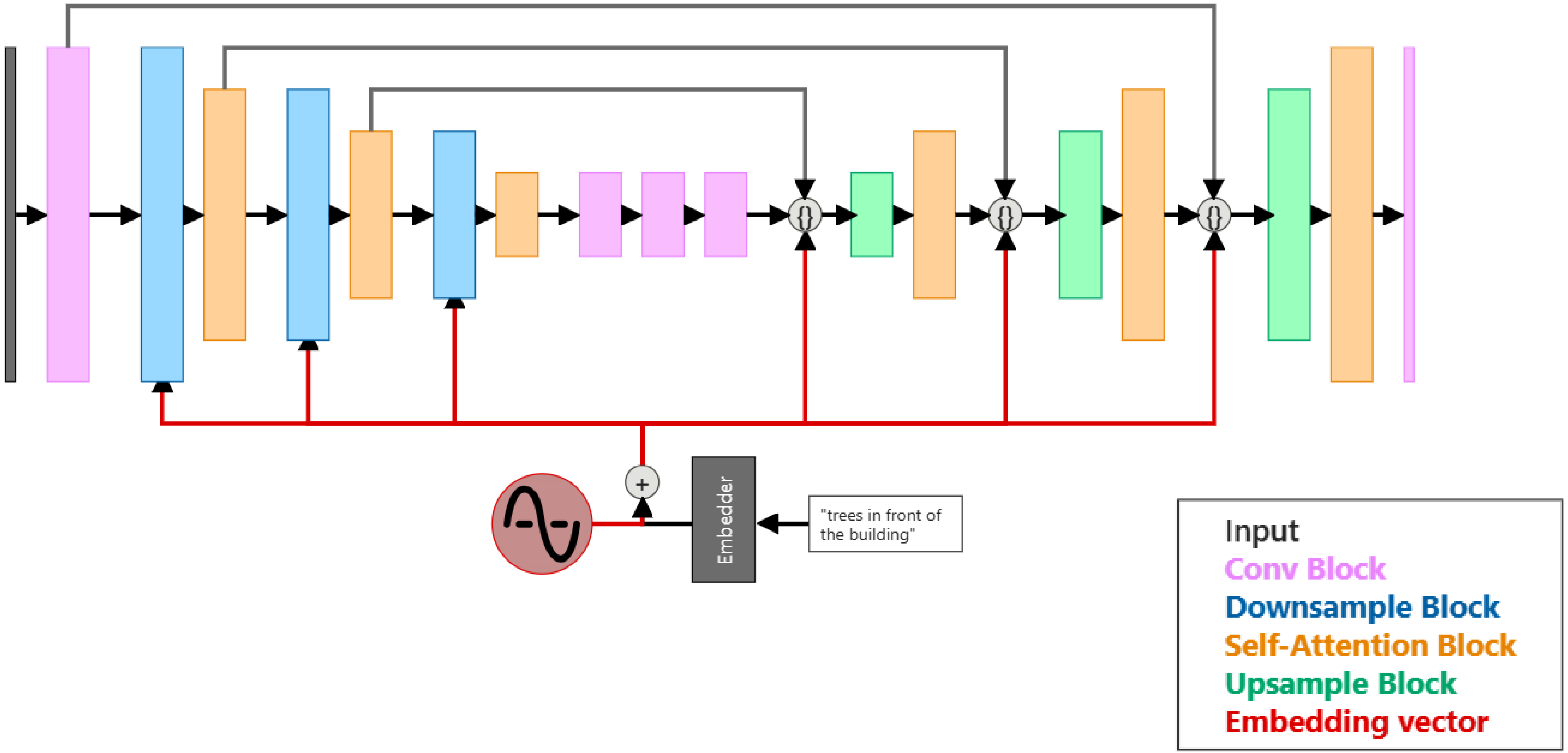
Diffusion Models

Positional encoding is used to carry information about the current time step. Hence, Sinusoidal embeddings are being added to all down sample and up sample blocks.



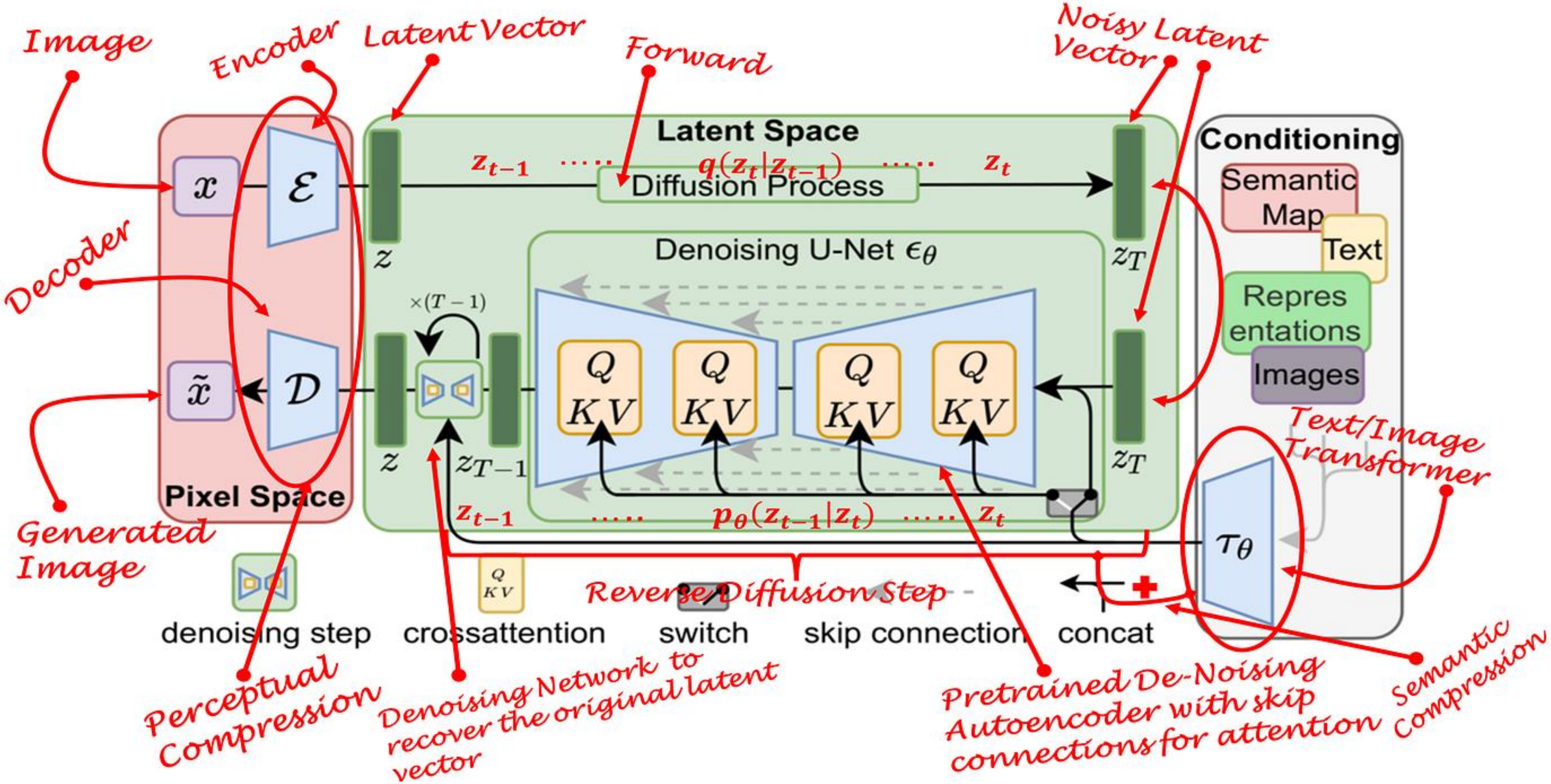
Diffusion Models

For conditional generation (depending on input prompt), a text embedder is also added to guide the diffusion process.



Diffusion Models

Here is the overview of the complete architecture



Applications of Generative AI

It has a wide range of applications across various domains due to its ability to create new content that is similar to its training data.

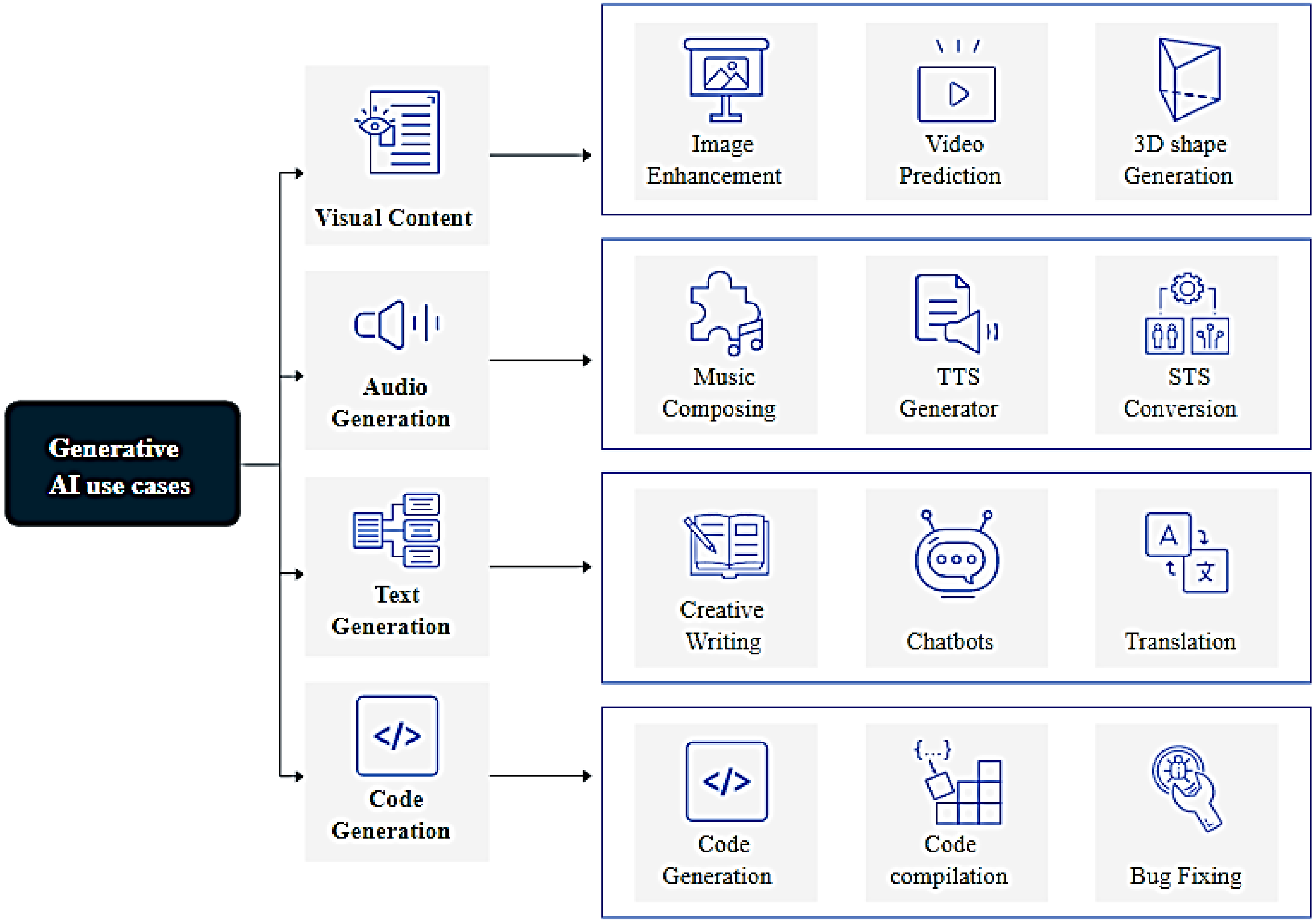


Image Generation

One of the most prominent and visible applications of Generative AI, creating highly realistic and imaginative visuals.

- **Early methods (GANs and variants):** GANs were the earliest successful models for generating new images.
- **Multimodal Models:** These models are capable of handling and generating multiple types of data. For image generation the combination of transformers with VAEs or Diffusion models works very well.
- **Diffusion models:** These models use a new approach to generate data using the diffusion process. A random noise is incrementally added to the input image and the model learns how to reverse this process to generate image that is similar to the input.

Figure: Images generated by GANs

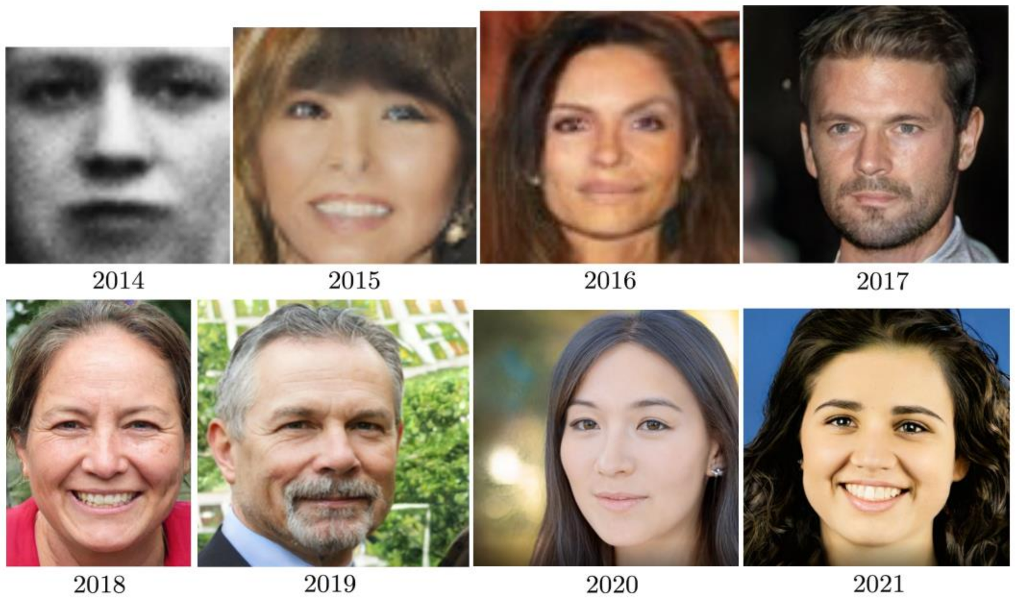
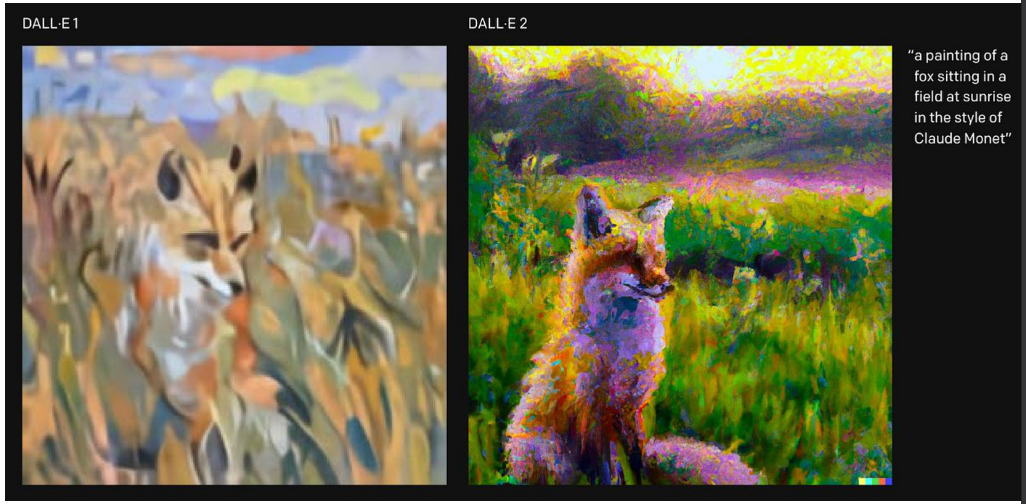


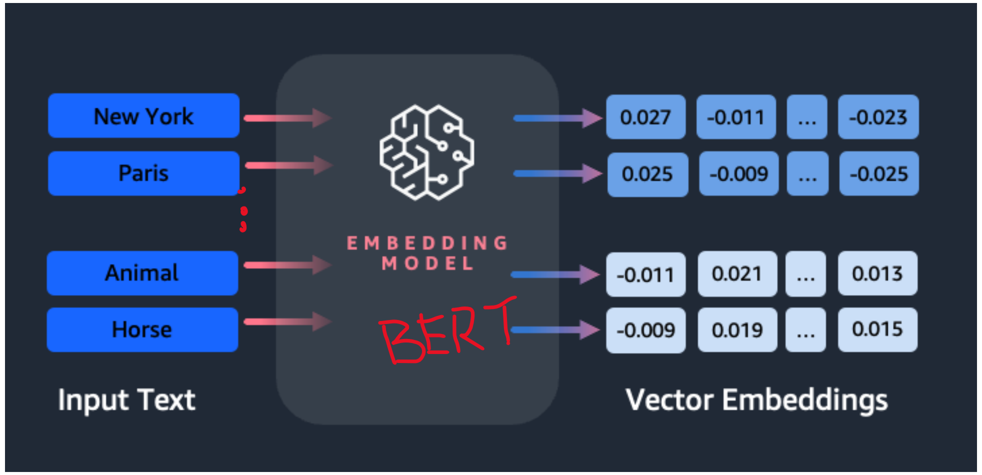
Figure: Comparison of DALL-E and DALL-E 2

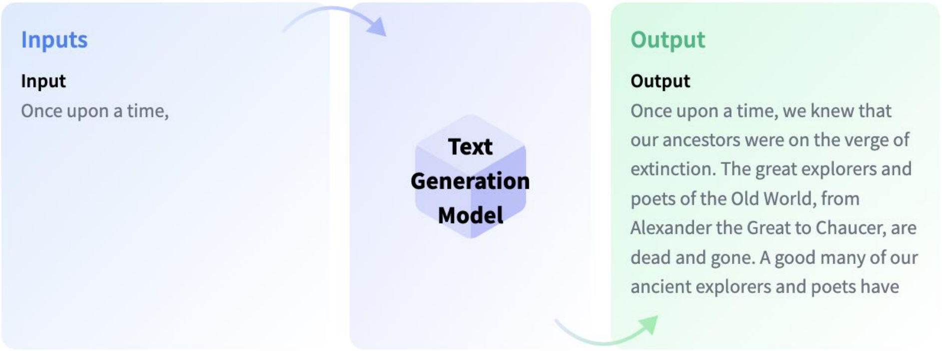


Text Generation

Generative models have made great advancements in NLP, enabling the generation of human-like text and powering a wide range of language-related task by learning from massive text data.

- **Early methods (RNNs):** Used to predict the next word in a sequence by modeling dependencies over time.
- **Transformers:** In 2017, the transformer architecture was introduced based on self-attention mechanism which enhanced sequence modeling by focusing on the important parts of the inputs.
- **Breakthrough models:** In 2018, **BERT** was developed to understand the context of words from both directions. The release of **GPT-3** in 2020 marked a new era in language modeling. Its massive parameters enabled it to be used for various NLP tasks.

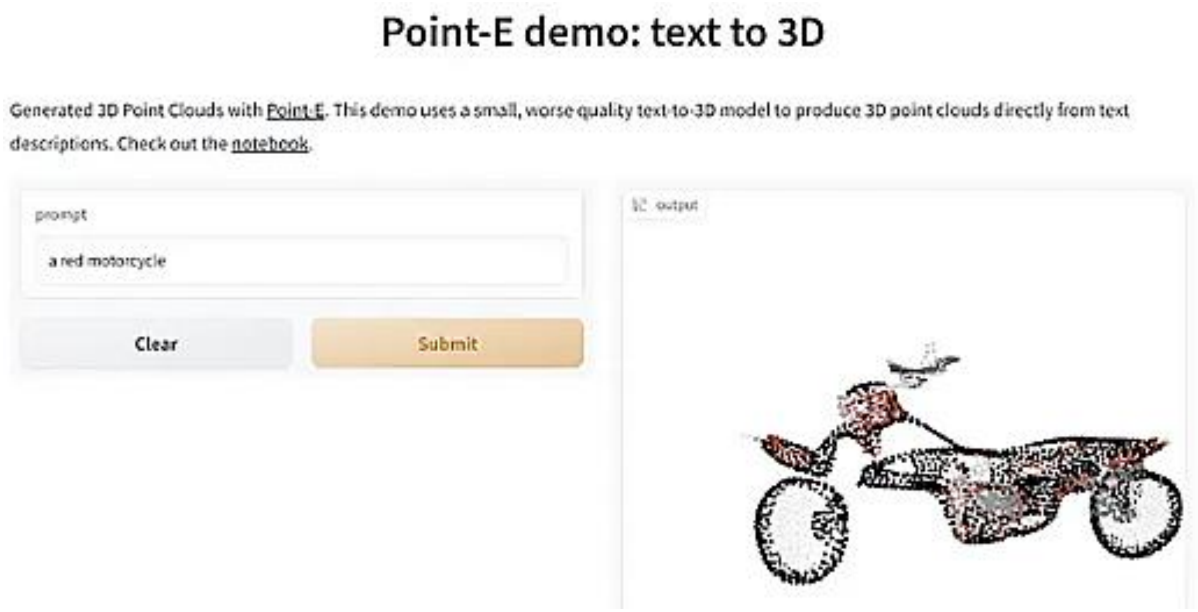




3D Object Generation

Generative AI can be extended in the 3D domain to generate 3D objects from various input modalities.

- **Text-to-3D:** By leveraging LLMs and Diffusion models these models can generate 3D objects from a text prompt.
- **Metaverse Integration:** Generative models can aid in rapid and automated creation of 3D environments in metaverse.



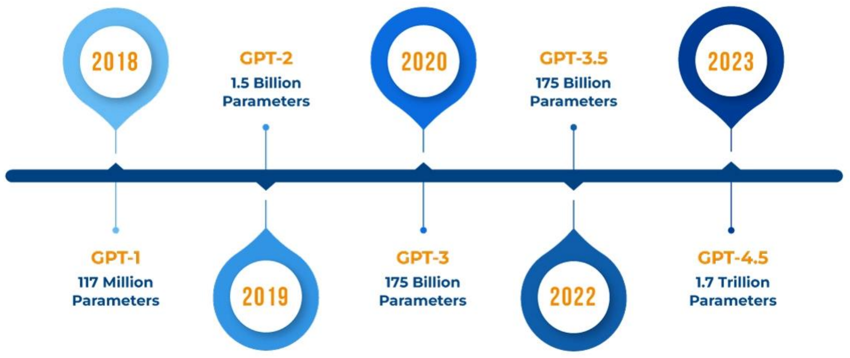
Video Generation

Nowadays, Generative models are also excelling in generating videos.

- **Text-to-Video Synthesis:** LLMs and Diffusion models are being combined for generating videos based on a prompt.
- **OpenAI SORA:** Sora is OpenAI's text-to-video model, announced in February 2024. It can create highly realistic and creative videos up to 60 seconds long based on text prompts.

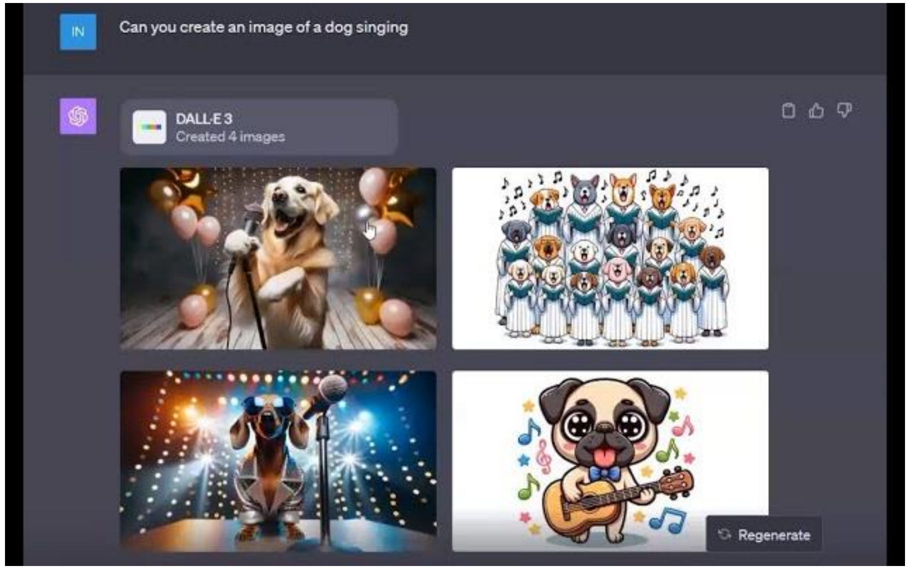
Leading Companies in AI

One of the leading companies is **OpenAI**, founded in 2015 by Elon Musk, Sam Altman, and others with their focus on **AGI**.



- **ChatGPT:** It is a conversational AI application, which is powered by GPT-3 and GPT-4 language models fine tuned specifically for conversation. These models have billions of parameters which are trained on huge text data.

- **DALL-E (1, 2, 3):** These are text-to-image models which utilizes CLIP to understand relationship between text and image and diffusion models to generate the image. These are integrated inside ChatGPT plus.

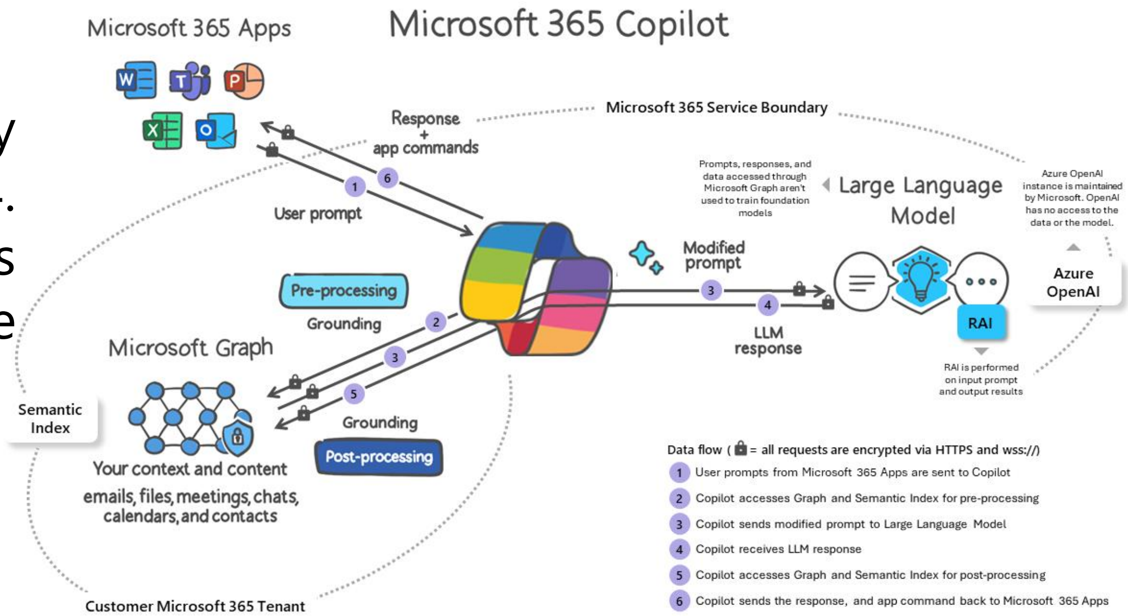


Leading Companies in AI

Microsoft has made significant strides in integrating generative AI into its products, leveraging advancements from its collaborations with OpenAI.

- **Microsoft Copilot:** It is an AI powered assistant that is integrated into the Windows OS. It can respond to prompts, generate code, and also enhance productivity in office applications like Word, Excel , PowerPoint, etc.

- **Bing:** A search engine, which is significantly enhanced with the integration of GPT-4. Beyond search engine, it is used as conversational assistant with image generation capabilities.



Leading Companies in AI:

Google has made significant progress in generative AI and has built various end-user products based on them.

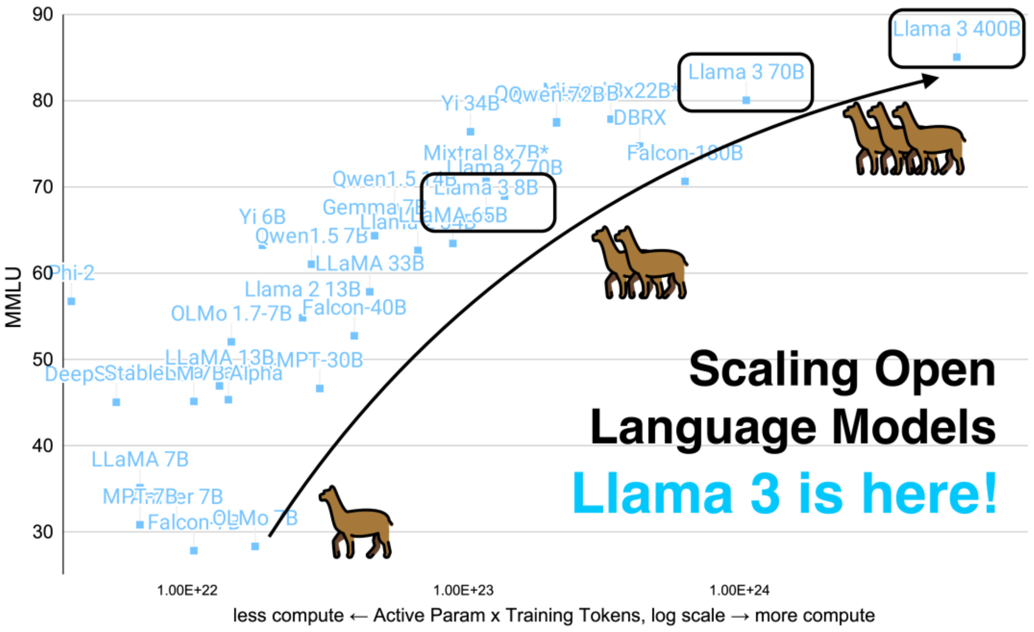
- **Imagen:** Imagen is text-to-image generation too that uses LLMs to understand the textual prompt and diffusion models to generate realistic photos.
- **Gemini Models:** The Gemini family of models was introduced in 2023, specifically focused on text but also has capabilities of understanding images and videos.
- **Google Generative AI Studio:** It is a platform that provides end users to experiment with Google's generative models. We can fine-tune the Gemini models for our task and integrate them in our applications through the use of APIs.

A dog looking curiously in the mirror, seeing a cat.

Leading Companies in AI

Meta AI is also developing generative AI technologies to integrate into their products.

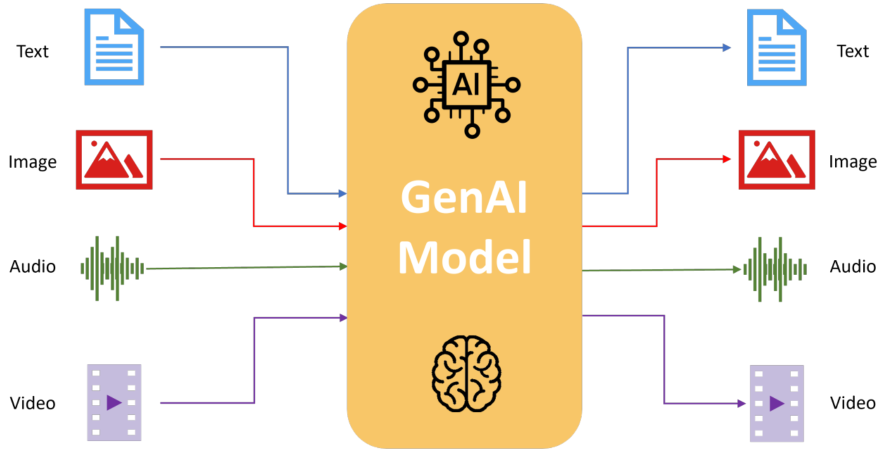
- **LLaMA models:** There contribution in LLMs is the LLaMA series of models. These models are capable of doing various kinds of tasks related to text. The LLaMA 3 model has been recently integrated into Facebook, Instagram, and WhatsApp. This specific model has 70 billion parameters.

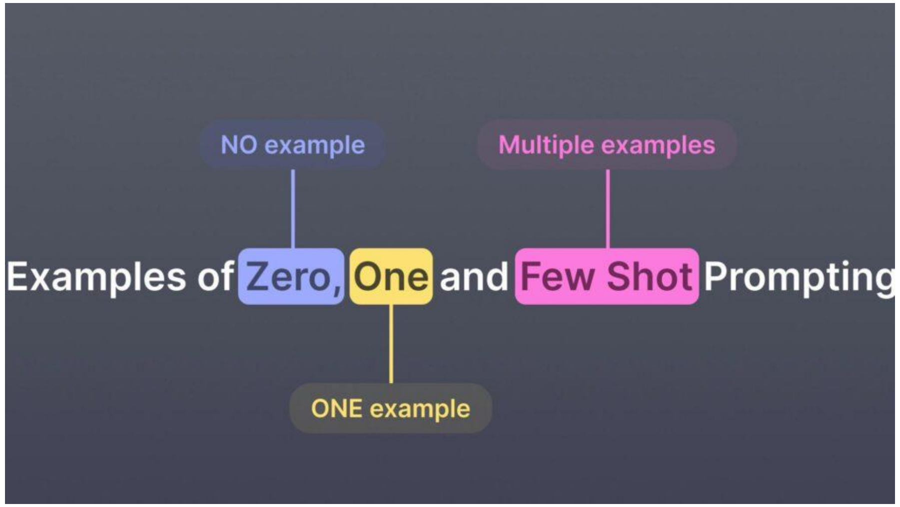


Emerging Trends

The field of generative AI is rapidly evolving and there are some exciting techniques that aim to push the boundaries of what these models can achieve.

- **Multimodal Generation:** It is an area of significant interest, where models can generate content across multiple modalities, such as text, images, audio, and even 3D data.
- **One-shot Learning:** Models can learn to generate new content by being exposed to a single example, leveraging their understanding of the underlying concepts and patterns.
- **Few-shot Learning:** With a handful of examples (typically less than 10), models can quickly adapt and generate content that is consistent with the provided examples.
- **Zero-shot Learning:** In this setting, models can perform tasks or generate content without being explicitly trained on examples from that particular domain. They rely on their broad knowledge and understanding to infer and generalize.





Challenges and Open Problems

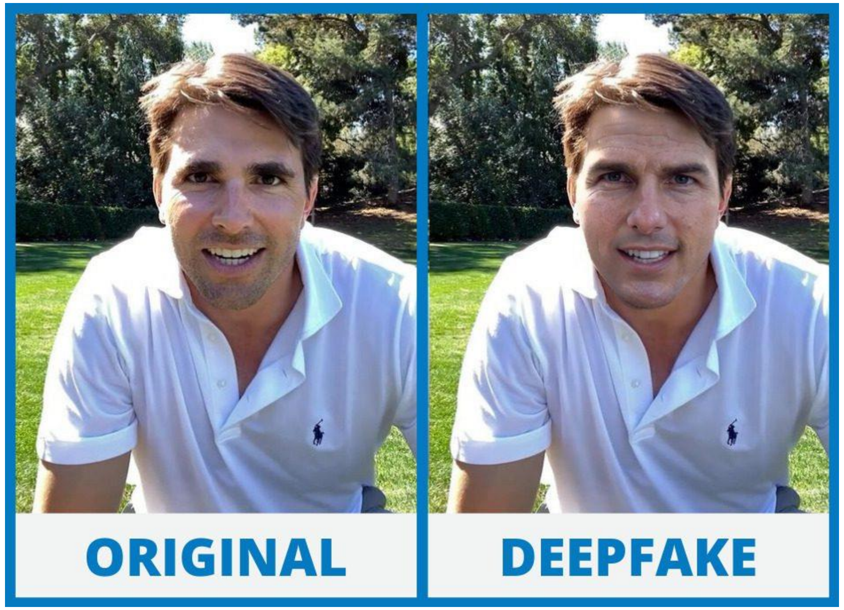
The field of generative AI is getting extremely popular and thus it is also posing some challenges.

- **Bias and Fairness:** AI models can amplify biases present in training data, leading to unfair and discriminatory outcomes.
- **Copyright and Ownership:** The use of existing content in training generative models raises legal issues regarding intellectual property.
- **Scalability and Efficiency:** Generative models, especially large ones, require significant computational resources and can be challenging to scale for real time applications.

Ethical Considerations

Ethical considerations in Generative AI are critical to ensuring that these technologies benefit society while minimizing potential harms.

- **Deepfakes and Synthetic media:** Deepfakes have potential for spreading false information while synthetic media can erode public trust in digital media.
- **AI Existential Risk Debates:** Discussions around the potential long-term risks that AI technologies pose to humanity, including the possibility of superintelligent AI.
- **Regulation and Governance:** Establishing legal frameworks and policies to guide the development and use of AI technologies.



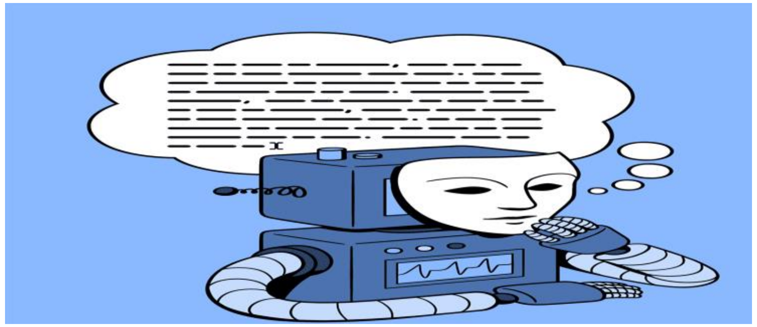
Unveiling the Power of Deep Learning

Large Language Models (LLMs)

Overview

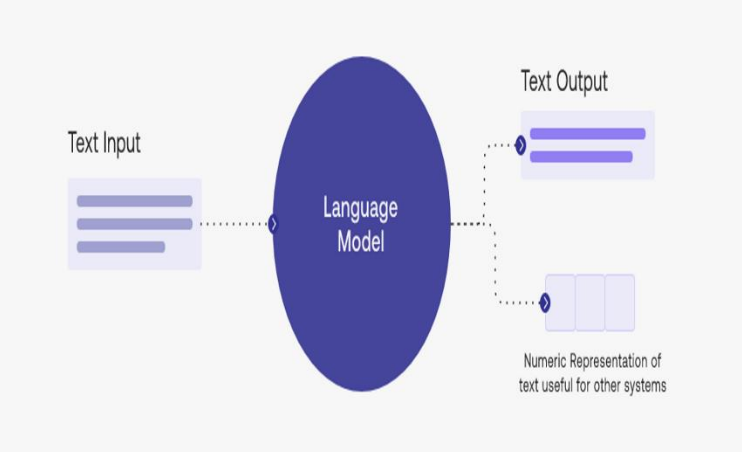
- **Understanding Large Language Models:**
  - **LLMs:**
    - What are Large Language Models (LLMs)?
    - What are LLMs used for?
    - Evolution of Large Language Models
    - How do LLMs work?
    - How are Large Language Models Trained?
    - Key Components of Large Language Models (LLMs)
    - Challenges of Large Language Models
    - Application of Large Language Models
    - Evolution of ChatGPT From GPT-1 to GPT-3.5
    - Understanding BERT and its Functionality
    - BERT's Architecture
    - Gemini: The Next Generation AI by DeepMind

What are Large Language Models (LLMs)?



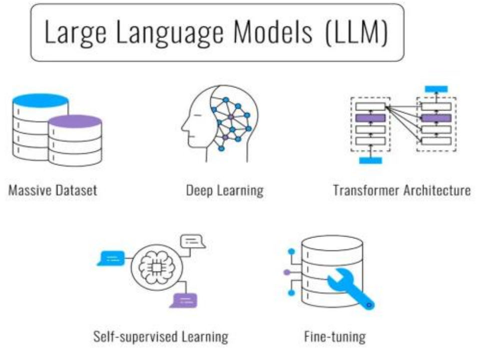
- A large language model is a computer program that learns and generates human-like language using a transformer architecture trained on vast training data.

- Large Language Models (LLMs) are foundational machine learning models that use deep learning algorithms to process and understand natural language.



- These models are trained on massive amounts of text data to learn patterns and entity relationships in the language.

- LLMs can perform many types of language tasks, such as translating languages, analyzing sentiments, chatbot conversations, and more.



- They can understand complex textual data, identify entities and relationships between them, and generate new text that is coherent and grammatically accurate, making them ideal for sentiment analysis.

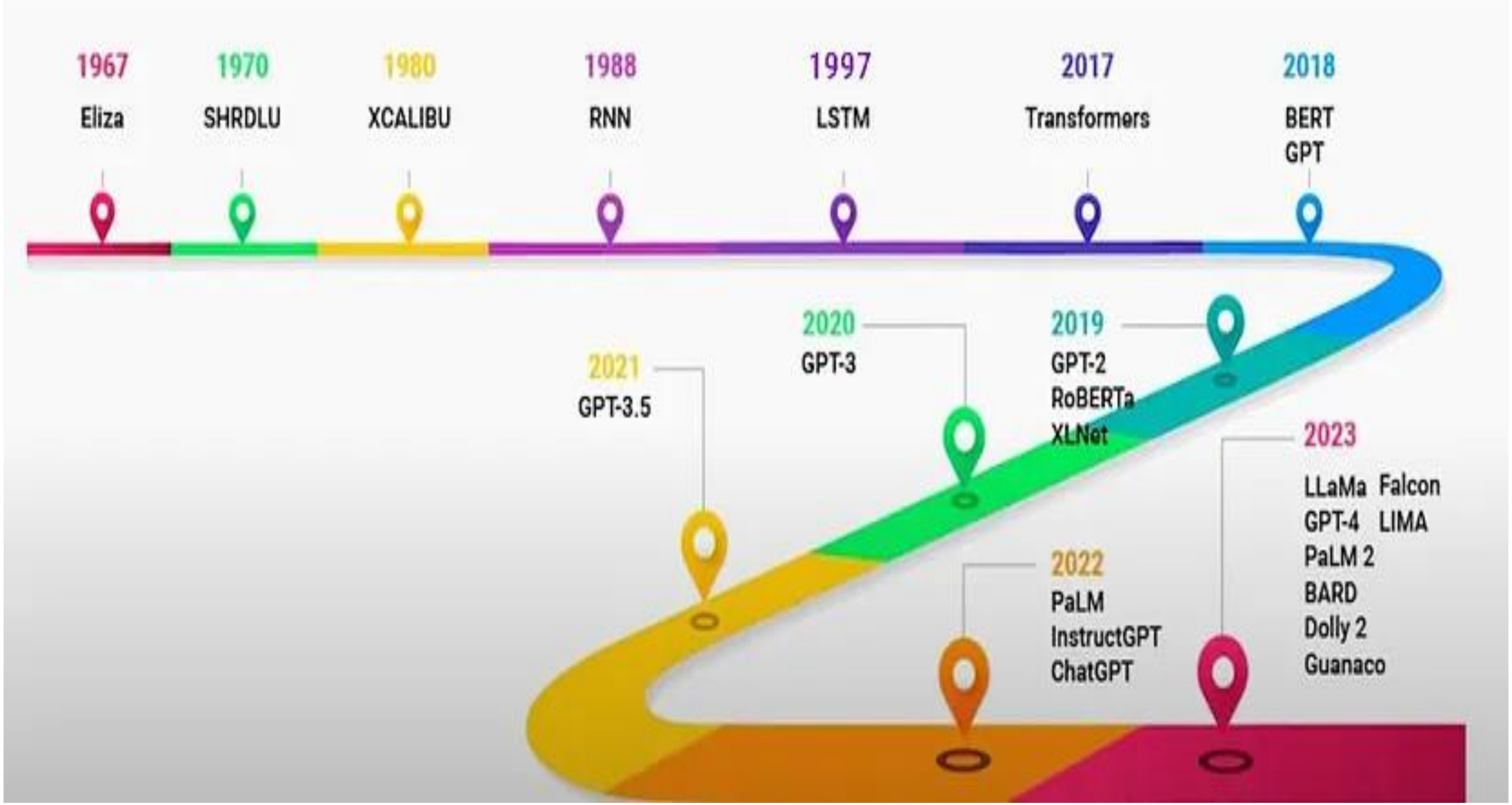
What are LLMs used for?

- **Content creation and communication:** LLMs can be used to generate different creative text formats, like poems, code, scripts, musical pieces, emails, and letters. They can also be used to summarize information, translate languages, and answer your questions in an informative way.

- **Analysis and insights:** LLMs are capable of analyzing massive amounts of text data to identify patterns and trends. This can be useful for tasks like market research, competitor analysis, and legal document review.

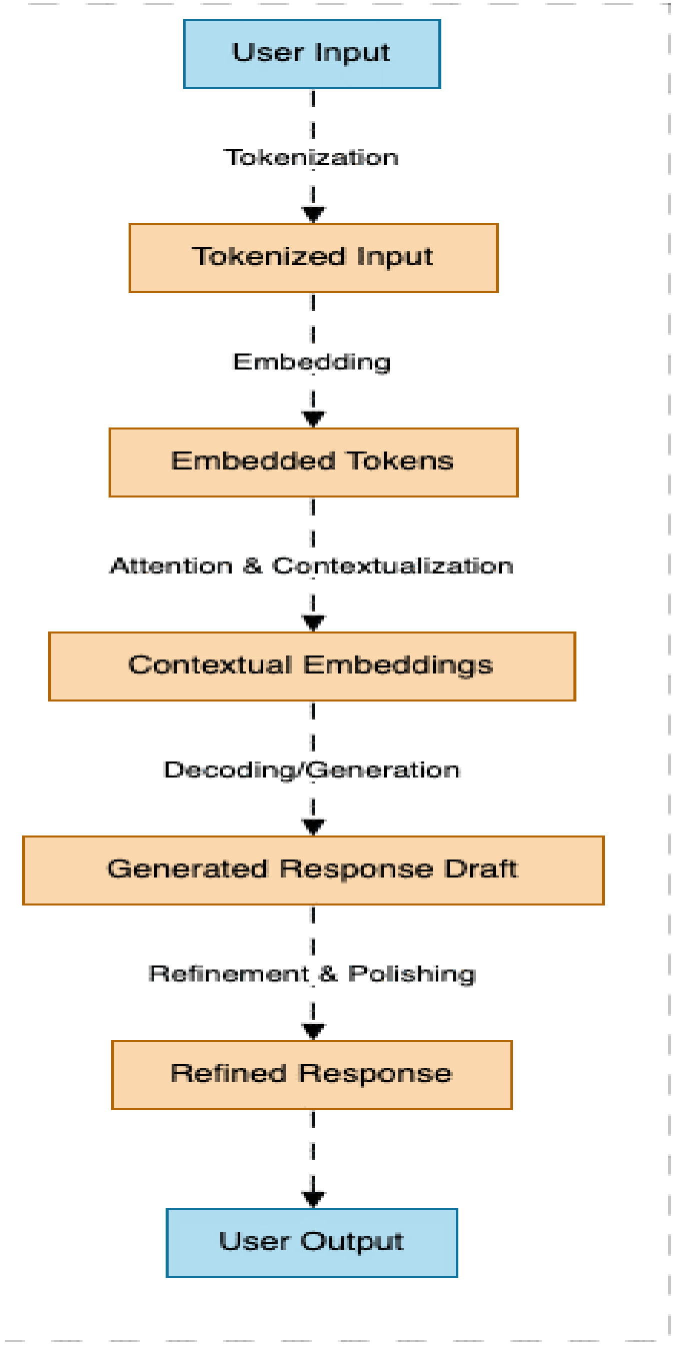
- **Education and training:** LLMs can be used to create personalized learning experiences and provide feedback to students. They can also be used to develop chatbots that can answer student questions.

Evolution of Large Language Models



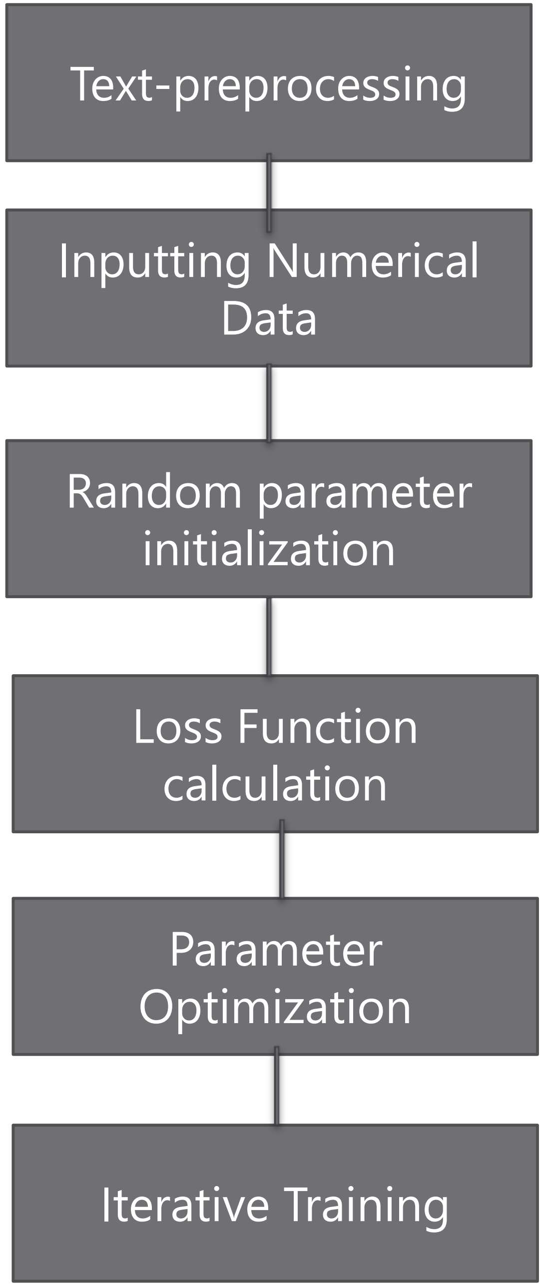
How do LLMs work?

- **Learning from Lots of Text:** These models start by reading a massive amount of text from the internet. It's like learning from a giant library of information.
- **Innovative Architecture:** They use a unique structure called a transformer, which helps them understand and remember lots of information.
- **Breaking Down Words:** They look at sentences in smaller parts, like breaking words into pieces. This helps them work with language more efficiently.
- **Understanding Words in Sentences:** Unlike simple programs, these models understand individual words and how words relate to each other in a sentence. They get the whole picture.
- **Getting Specialized:** After the general learning, they can be trained more on specific tasks to get good at certain things, like answering questions or writing about particular subjects.
- **Doing Tasks:** When you give them a prompt (a question or instruction), they use what they've learned to respond. It's like having an intelligent assistant that can understand and generate text.



How are Large Language Models Trained?

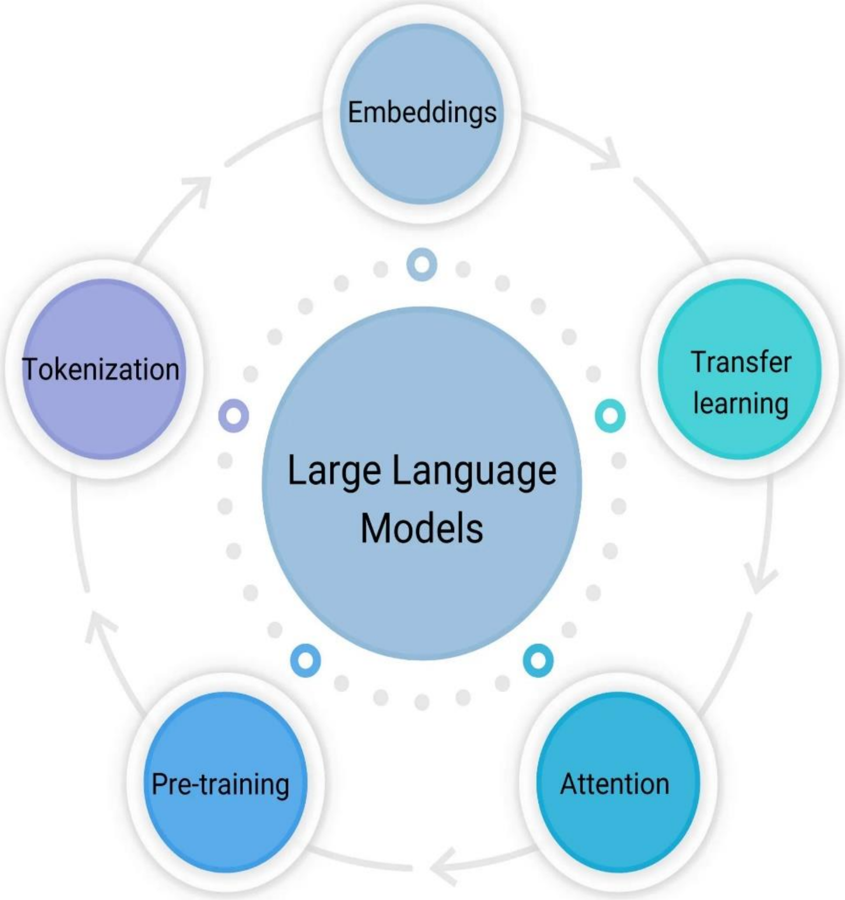
- Most LLMs are pre-trained on a large, general-purpose data set. The purpose of pre-training is for the model to learn high-level features that can be transferred to the fine-tuning stage for specific tasks.
- The training process of a large language model involves:
- Pre-processing the text data to convert it into a numerical representation that can be fed into the model.
- Randomly assigning the model's parameters.
- Feeding the numerical representation of the text data into the model.
- Using a loss function to measure the difference between the model's outputs and the actual next word in a sentence.
- Optimizing the model's parameters to minimize loss.
- Repeating the process until the model's outputs reach an acceptable level of accuracy.



Key Components of Large Language Models (LLMs)

- Large Language Models (LLMs) are complex neural network architectures that have revolutionized natural language processing (NLP) tasks.

- These models are composed of several key components that work together to enable them to understand, generate, and manipulate human language with remarkable fluency and accuracy, finding divers real-word application of LLMs.



1. Tokenization
Tokenization is the process of breaking down text into smaller pieces so that a model can understand and work with any text effectively.

example:
"She enjoys reading books."
["She", "enjoys", "reading", "books"]
"Unpredictable weather can be challenging."
["Unpredictable", "weather", "can", "be", "challenging"]

Technique of tokenization
Whitespace
Punctuation-Based Tokenization
Word Tokenization
Sub word Tokenization
Character Tokenization

2. Embeddings

Embeddings are a way to represent words or pieces of text as numbers. This helps a computer understand their meanings and relationships. Here's a simple breakdown
Representation as Numbers: Words are turned into sets of numbers (vectors) instead of using plain text. This numerical format is easier for a computer to work with.

3. Attention Mechanisms

Attention mechanisms help models focus on the most important parts of a sentence. Here's a simple explanation

Focus on Important Words: Attention allows the model to pay more attention to the important words or phrases in a sentence.

Assigning Weights: The model assigns different weights to each word, giving more weight to important words and less to less important ones.

Understanding Context: This helps the model understand the context and meaning better, especially when dealing with long sentences or complex language.

Example

Consider the sentence: "The cat, which was very fluffy, sat on the mat."
Without attention, the model might treat every word equally. But with attention, the model can focus more on "cat" and "sat on the mat" to understand that the main action is about the cat sitting on the mat.

4. Pre-training

- Pre-training is the first step in training a large language model (LLM). Here's a simple explanation:

- **Learning Basics:** The model is trained on a large amount of text to learn general language patterns and word relationships.

- **Unsupervised Learning:** This training doesn't need labeled data (specific answers). Instead, the model learns by predicting missing words or understanding the text by itself.

5. Transfer learning

- Transfer learning is the technique of leveraging the knowledge gained during pretraining and applying it to a new, related task. In the context of LLMs, transfer learning involves fine-tuning a pre-trained model on a smaller, task-specific dataset to achieve high performance on that task.

- The benefit of transfer learning is that it allows the model to benefit from the vast amount of general language knowledge learned during pretraining, reducing the need for large labeled datasets and extensive training for each new task.

Challenges of Large Language Models

- Scaling and maintaining large language models can be difficult and expensive.

- Building a foundational large language model often requires months of training time and millions of dollars.

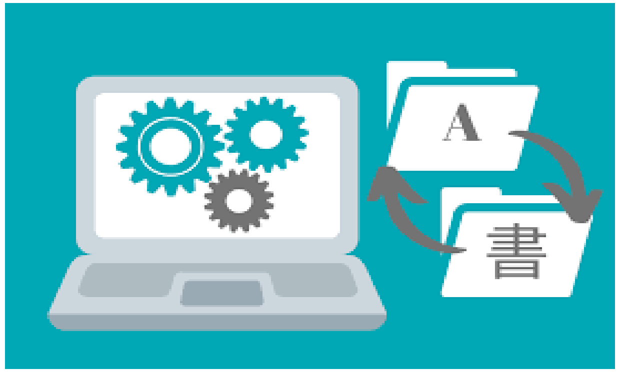
- LLMs require a significant amount of training data, developers and enterprises can find it a challenge to access large-enough datasets.

- Due to the scale of large language models, deploying them requires technical expertise, including a strong understanding of deep learning, transformer models and distributed software and hardware.

- Many leaders in tech are working to advance development and build resources that can expand access to large language models, allowing consumers and enterprises of all sizes to reap their benefits.

Application of Large Language Models

- Chatbots and Virtual Assistant
- Content Creation
- Machine Translation
- Natural Language Processing
- Sentiment Analysis



Evolution of ChatGPT: From GPT-1 to GPT-3.5

Introduction to ChatGPT

• ChatGPT is a series of conversational AI models developed by OpenAI, designed to generate human-like text based on input prompts.

• Importance: These models have transformed the field of natural language processing (NLP) by enabling advanced conversational agents, content generation,

- GPT-1

- **Overview**

- **Release:** 2018

- **Architecture:** Transformer architecture with 12 layers, 12 attention heads, and 117 million parameters.

- **Functionality**

- **Pre-training:** Trained on a large corpus of text data using unsupervised learning.

- **Fine-tuning:** Adapted to specific tasks using supervised learning on task-specific datasets.

- **Applications**

- **Text Generation:** Early demonstrations of generating coherent paragraphs of text.

- **Limitations:** Limited context understanding and coherence compared to later models.

GPT-2

•**Overview**

- Release: 2019
- Architecture: Transformer architecture with 1.5 billion parameters.

•**Improvements Over GPT-1**

- **Increased Size:** Significantly larger model with improved language understanding and generation capabilities.

- **Better Coherence:** Generates more coherent and contextually relevant text.

- **More Parameters:** GPT-2 boasts a whopping 1.5 billion parameters, a tenfold increase compared to GPT-1. This allows it to learn more complex patterns in language

•**Applications**

- **Enhanced Text Generation:** More sophisticated text generation for applications like chatbots and content creation.

- **Creative Writing:** Used for generating stories, poetry, and other creative content.

- # GPT-3
- **Overview**
- **Release:** 2020
- **Architecture:** Transformer architecture with 175 billion parameters.
- **Key Features**
- **Few-Shot Learning:** Demonstrates strong few-shot, one-shot, and zero-shot learning capabilities.
- **Wide Applications:** Capable of performing a wide range of NLP tasks without fine-tuning.
- **Applications**
- **Chatbots:** Highly effective in creating conversational agents that can handle diverse topics.
- **Content Creation:** Used for writing articles, summarizing text, and generating code.
- **Code Generation:** Applied in tools like GitHub Copilot to assist developers.

• GPT-3.5

• **Overview**

- **Release**: 2021
- **Architecture**: Enhanced version of GPT-3 with optimizations for efficiency and performance.

• **Improvements Over GPT-3**

- **Fine-Tuning**: Improved fine-tuning techniques for better adaptation to specific tasks.

- **Improved Efficiency:** Enhanced performance with optimizations for faster response times and reduced latency.

- **Refined Multimodal Capabilities:** Improved ability to process and generate responses from both text and image inputs, supporting richer interactions.

• **Applications**

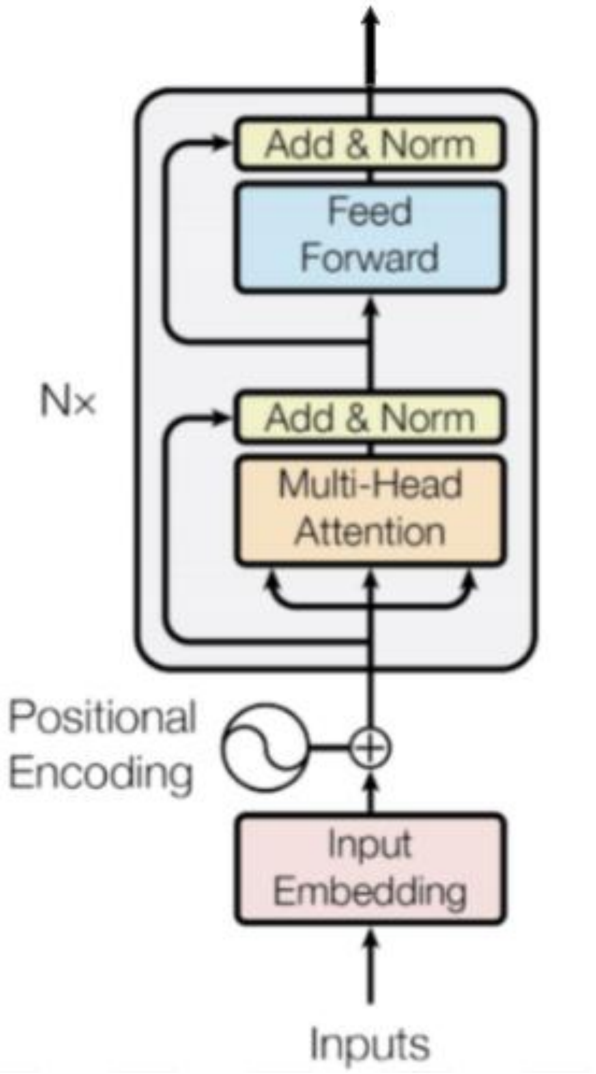
- **Advanced Conversational AI**: More responsive and context-aware chatbots.
- **Enterprise Applications**: Deployed in customer service, virtual assistants, and other enterprise solutions.

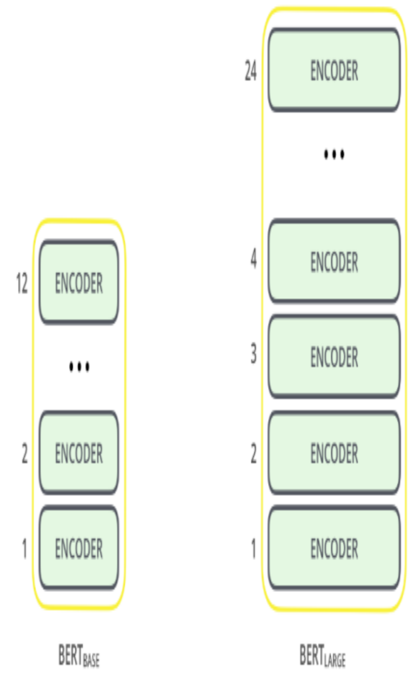
Understanding BERT and its Functionality

- **What is BERT?**
- BERT (Bidirectional Encoder Representations from Transformers) is a language representation model developed by Google.
- **Purpose:** Designed to understand the context of words in a sentence more accurately than previous models by processing text bidirectionally.
- **Bidirectional Training:** Unlike previous models that read text left-to-right or right-to-left, BERT reads in both directions simultaneously. This allows it to understand the context of a word based on all surrounding words in a sentence.
- **Transformer Architecture:** Utilizes the Transformer model, specifically the encoder part, which includes self-attention mechanisms to weigh the importance of different words in a sentence.
- **Pre-training Objectives:**
- **Masked Language Model (MLM):** Randomly masks some words in the input and trains the model to predict these masked words based on the context provided by the other words.
- **Next Sentence Prediction (NSP):** Trains the model to understand the relationship between two sentences by predicting if one sentence follows another in the text.
- **Applications of BERT**
- **Search Engines:** Improves understanding of search queries to return more relevant results.
- **Sentiment Analysis:** Analyzes sentiment in text data, such as customer reviews.
- **Named Entity Recognition (NER):** Identifies and classifies entities in text, such as names, dates, and organizations.

BERT’s Architecture

- The BERT architecture builds on top of Transformer. We currently have two variants available:
- BERT Base: 12 layers (transformer blocks), 12 attention heads, and 110 million parameters
- BERT Large: 24 layers (transformer blocks), 16 attention heads and, 340 million parameters





Gemini: The Next Generation AI by DeepMind

Key Features

Advanced NLP
- Combines state-of-the-art transformers
- Enhanced contextual understanding

Reinforcement Learning
- Learns from interactions
- Adaptive decision-making

Multimodal Capabilities
- Handles text, images, and audio

How Gemini Works
- User Input: Text, image, or audio
- Tokenization & Embedding: Converts input to vectors
- Contextual Embedding: Understands context with transformers
- Reinforcement Learning: Improves from feedback
- Multimodal Processing: Integrates various data types
- Response Generation: Produces refined outputs

Benefits and Applications

- **Enhanced NLP:** Better chatbots, virtual assistants, content creation
- **Adaptive Learning:** Continuous improvement
- **Comprehensive Interaction:** Suitable for complex task

Thank You