Universität Konstanz

# 14 | Generative Deep Learning II

Giordano De Marzo

https://giordano-demarzo.github.io/

## Deep Learning for the Social Sciences

# Recap

- Generative Deep Learning for Images
- Reinforcement Learning
- Transformer
- Large Language Models

Universität Konstanz

# Outline

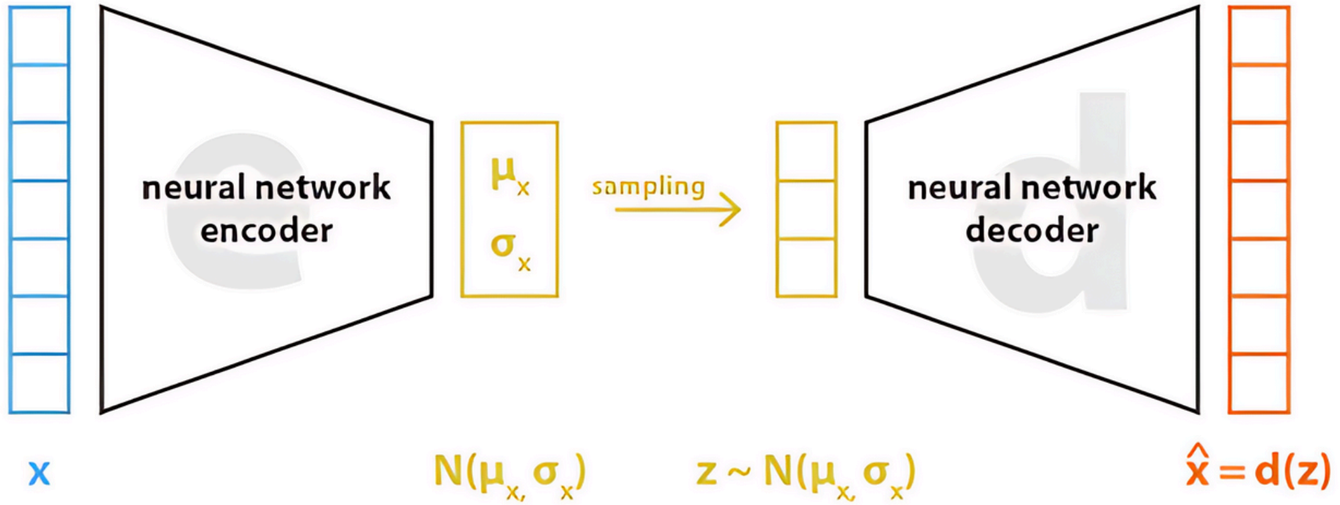
1. Diffusion Models
2. Large Language Models
3. Generative Agents


Universität
Konstanz
Diffusion Models

# Recap: VAE

The Variational Autoencoder (VAE) adds a twist of stochasticity to the standard AE

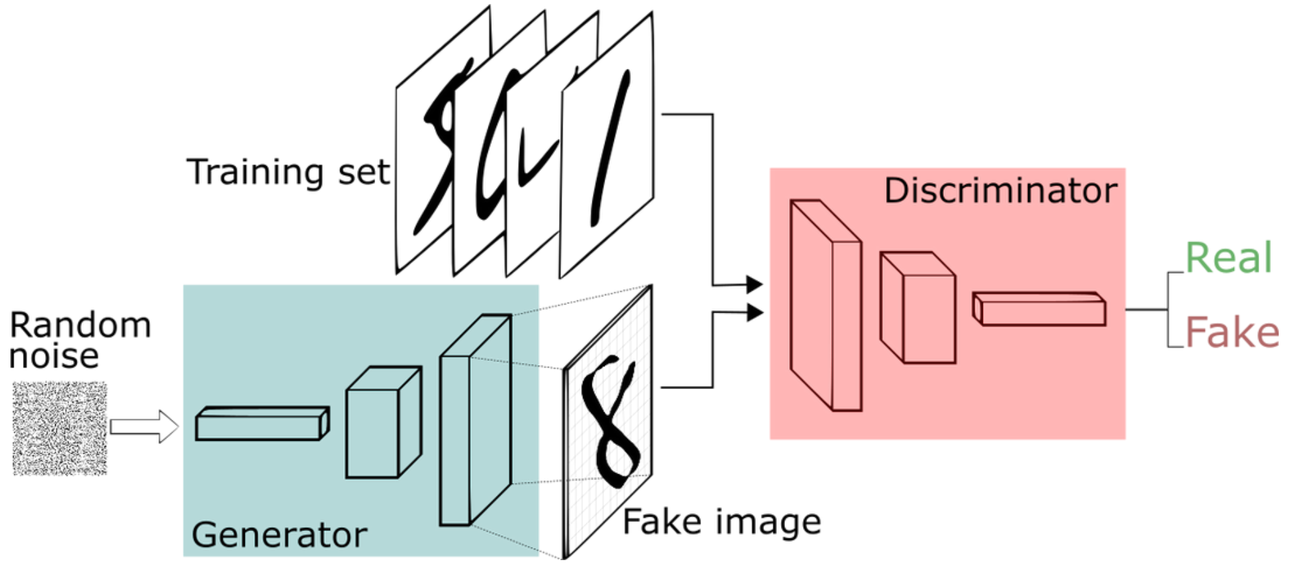
- same architecture, but now the encoder output consists of a mean vector and a variance vector
- the value of the latent variable z is computed sampling from a multivariate gaussian whose parameters are defined by the encoder's output
- the latent variable z in feed into the decoder, whose output is trained to be equal to the input



# Recap: GAN

Generative Adversarial Networks (GANs) are much more powerful when it comes to generate realistic images. They are conceptually very similar and are composed of two models

- a generator that is trained to generate artificial data
- a discriminator that is trained to distinguish artificial data from real data
- note that we do not need labelled data (we automatically know what is real and what is fake)



# From Noise to Images

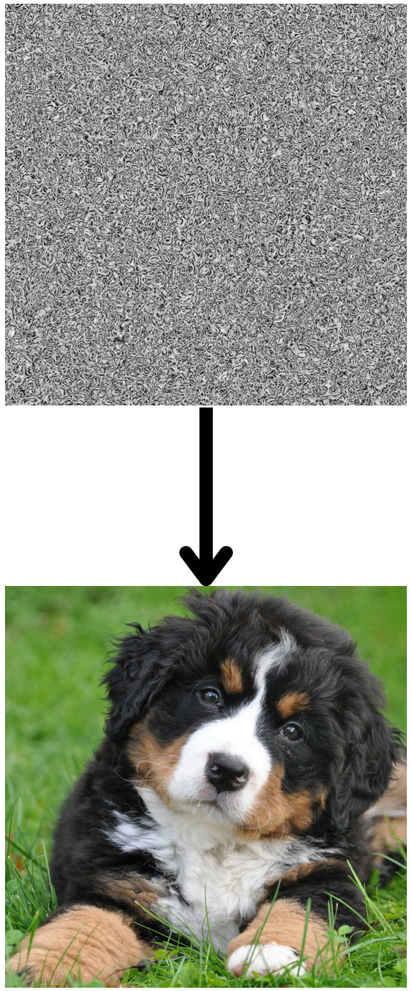
In both VAEs and GANs, the process of generating images begins with random noise.

- in VAEs the decoder generates images by sampling from the latent distribution.
- in GANs the generator creates images starting from random noise

In both these models images are generated in one transformative step from a noise distribution

- images are very complex objects
- a single step may not be enough for generating complex structures such as faces

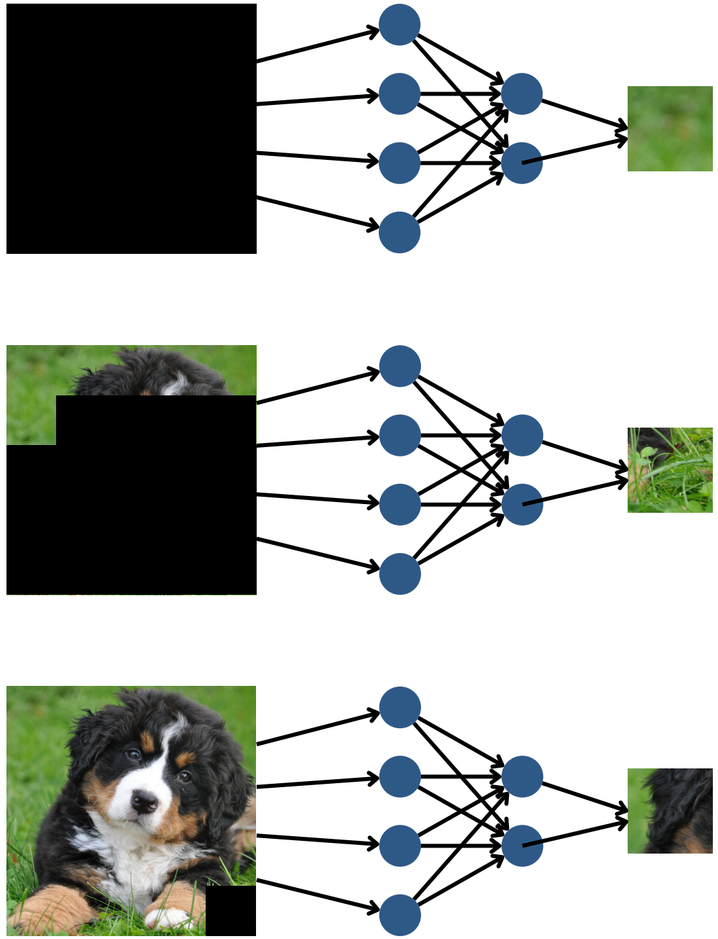
Autoregressive models and diffusion models use instead a multi-step approach



# Autoregressive Generative Models

Autoregressive Generative Models generate images step-by-step, breaking down the process into manageable pieces. Unlike VAEs and GANs, which produce entire images in one step, these models generate one part of the image at a time.

- They begin with an empty image and sequentially generates the remaining parts.
- A neural network predicts the next part of the image based on previously generated content.
- They manage the complexity of generating detailed structures by using a multi-step approach.



# Limits of Autoregressive Generative Models

Autoregressive models face significant challenges when generating high-resolution images:

- When too many pixels are generated at once, the model tends to average over multiple possible images, resulting in a blurred and indistinct output.
- To counteract the blurring, autoregressive models require generating images in many small steps. This multi-step approach, while improving quality, significantly increases computational complexity and processing time.

# Diffusion Models

Diffusion models introduce a powerful approach to generating high-quality. Unlike autoregressive models, which predict pixel-by-pixel, diffusion models work by gradually denoising an initially noisy image.

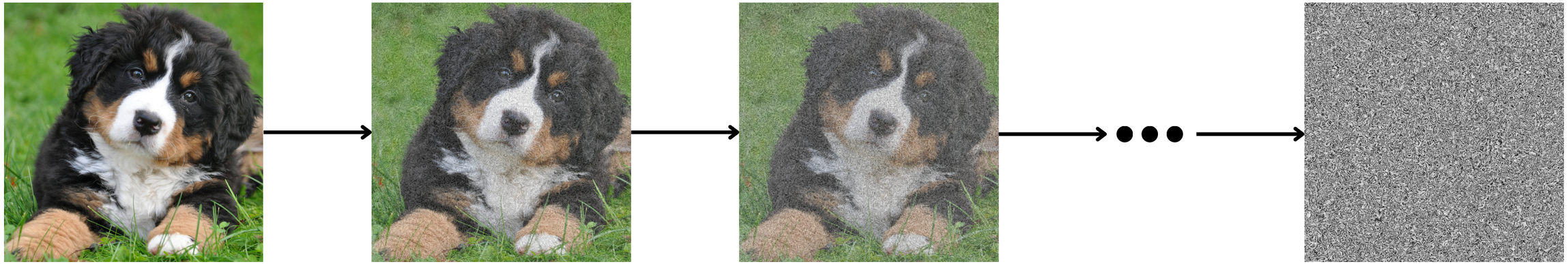
- **Multi-Step Refinement:** The process begins with pure noise, and through a series of steps, the model refines this noise into a clear and detailed image.
- **Noise Structure:** Noise has no inherent correlation or structure, so even when averaging over multiple possible noises, the result remains a valid noise. This property prevents the blurring effect seen in autoregressive models.
- **Bidirectional Process:** Diffusion models consist of a forward process, where the image is progressively noised, and a reverse process, where the noise is incrementally reduced.

In practice we train a neural network to denoise images by predicting what noise was added to them

# Forward Process

The forward process in diffusion models involves progressively adding noise to an image over several steps until it becomes pure noise. This gradual noising process serves to map the image to a noise distribution.

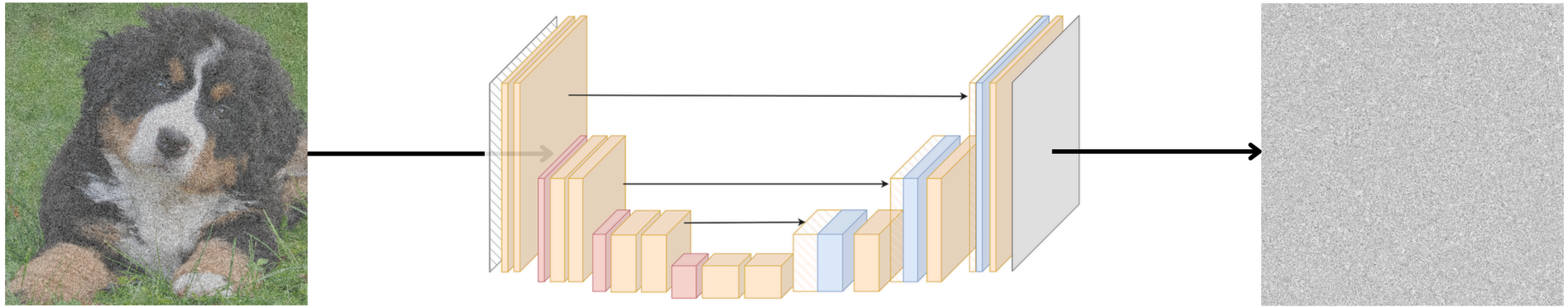
- less noise is added in the beginning, while more noise is added in the last steps (cosine schedule)
- this process produces a series of noisy images, each with a different noise level, that can be used for training the denoising network
- the process is completely unsupervised



# Reverse Process

The reverse process in diffusion models focuses on denoising the image step-by-step, transforming the noise back into a clear image

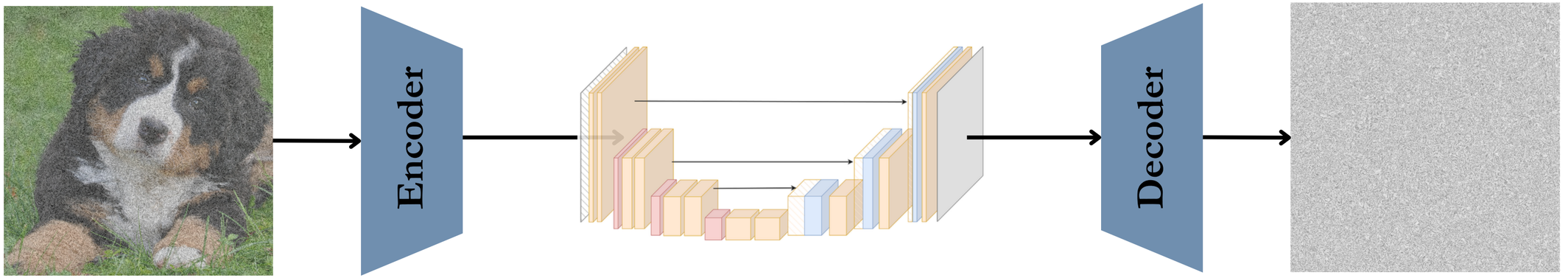
- The core idea is to train a neural network to predict the noise added at each step in the forward process. By accurately predicting this noise, the network can progressively reduce it.
- A specific CNN architecture called UNet is commonly used for this task. UNet consists of an encoder-decoder structure with skip connections
- The UNet model is applied iteratively, each time reducing a small amount of noise, gradually reconstructing the image from the noisy version.



# Latent Diffusion Models

Latent diffusion models improve upon traditional diffusion models by operating in a lower-dimensional latent space instead of the high-dimensional pixel space.

- Directly working with pixel data involves very high dimensionality, making the process computationally expensive and less efficient.
- To address this, latent diffusion models first use an encoder to transform the high-dimensional image data into a lower-dimensional latent space.
- Efficient Denoising: In the latent space, the model performs the denoising process, which is computationally more efficient and effective.



# GANs vs Diffusion Models

GANs often suffer from a phenomenon called mode collapse, where the generator produces a limited variety of images. This results in many images looking very similar, as the generator fails to capture the full diversity of the training data. Diffusion models, instead, do not show this problem.

**GAN**

**Diffusion**

**Training Data**

# Conditional Generation

Conditional generation allows models to generate images based on specific text inputs

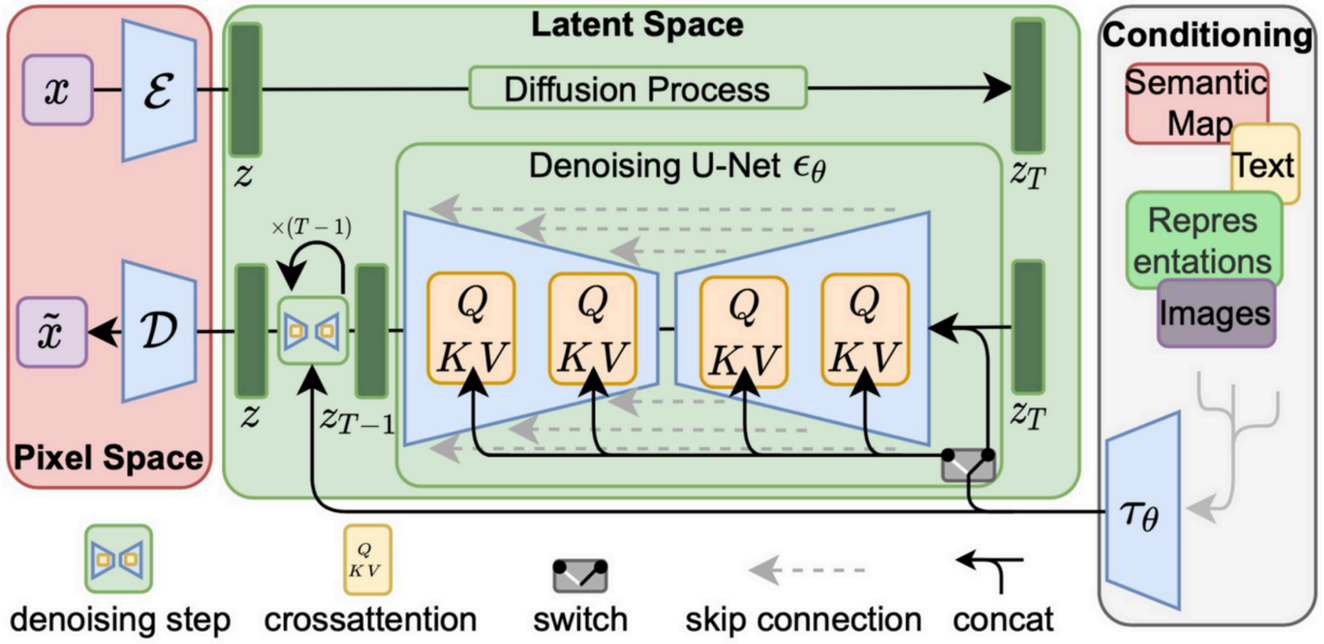
- **Explicit Conditioning:** In this approach during training, the condition is directly fed into the model alongside the noise input. This could be in the form of class labels, text descriptions, or other types of auxiliary information.
- **Classifier Guidance:** This method involves using a pre-trained classifier to guide the generation process. The classifier provides gradients that help adjust the generation towards the desired condition.
- **Classifier-Free Guidance:** In this approach, the model is trained to generate images both with and without conditions. During generation, a combination of the conditional and unconditional outputs is used to guide the final image synthesis.

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

# Stable Diffusion

Stable Diffusion is a type of latent diffusion model that generates high-quality images efficiently.

- **Latent Space:** Reduces complexity by operating in a lower-dimensional latent space.
- **Denoising UNet:** Uses a UNet architecture to iteratively refine the image.
- **Attention:** Incorporates attention for conditioning on inputs like text or semantic maps.



# State of the Art Models

Street style photo of a young woman, red gucci jacket, blue gucci shirt, wide shot, natural lighting, soho, shot on Agfa Vista 200, 4k

## Midjourney v5

## Dall-E 3

# State of the Art Models

Photography shot trough an outdoor window of a coffee shop with neon sign lighting, window glares and reflections, depth of field, little girl with red hair sitting at a table, portrait, kodak portra 800, 105 mm f1. 8

**Midjourney v5**

**Dall-E 3**

# Video Generation: Sora

**Sora: Creating video from text**

Sora is an AI model that can create realistic and imaginative scenes from text instructions.

OpenAI

https://openai.com/index/sora/

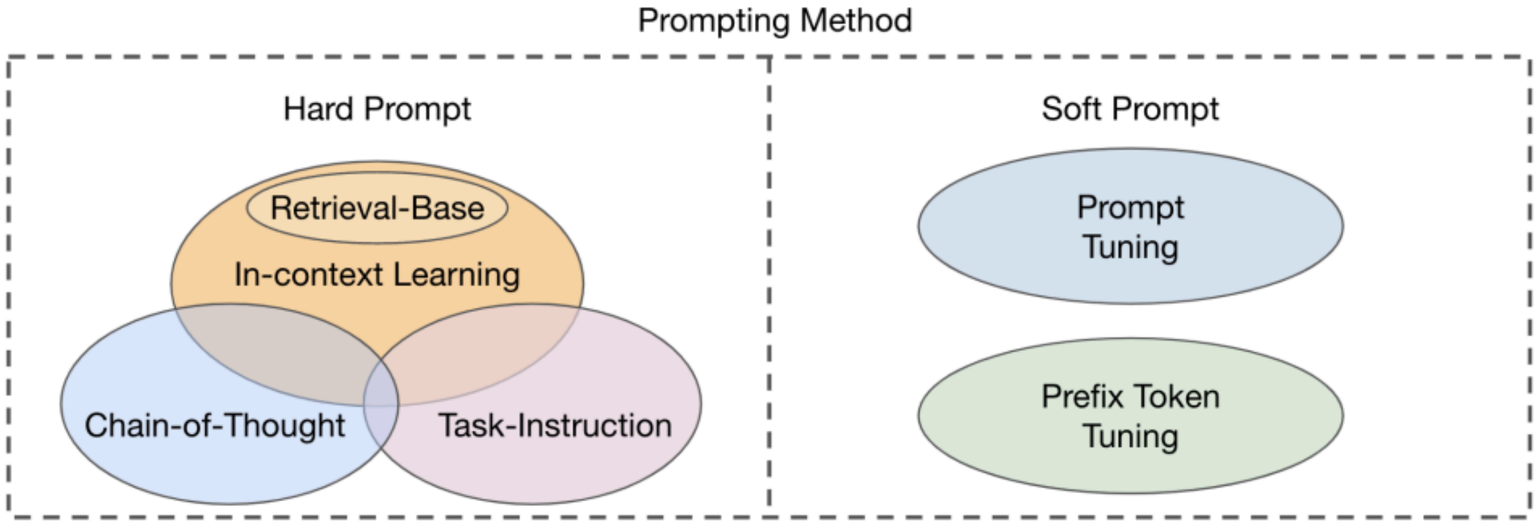
Universität
Konstanz

# Large Language Models

# Prompt Engineering

Prompt engineering is a crucial technique in working with large language models, enabling users to obtain better and more accurate outputs. The main approaches to prompt engineering include:

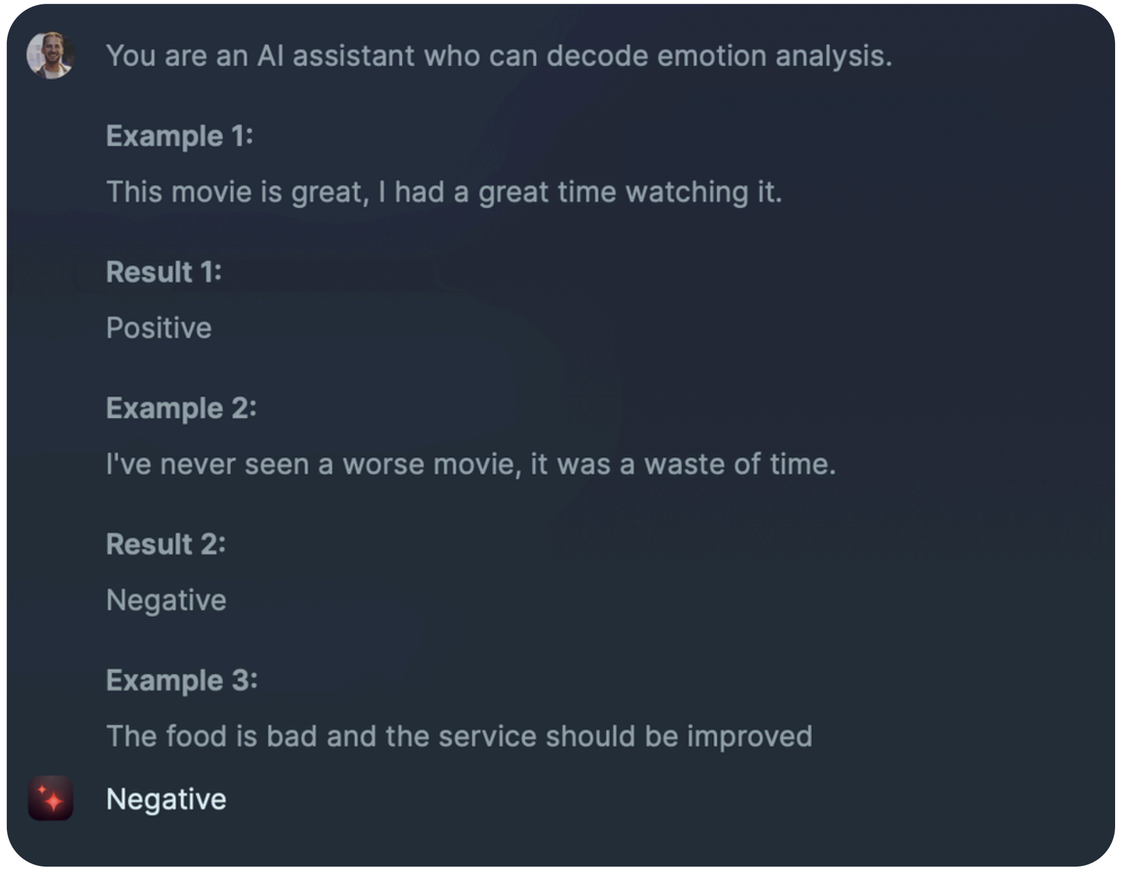
- **Few-Shot Learning:** Providing the model with a few examples in the prompt
- **Chain of Thought:** Structuring prompts to guide the model through a logical sequence
- **Soft Prompts:** Using learnable prompt tokens that adapt during training



# Zero-Shot vs Few-Shot

In Few-Shot learning the model is provided with a few examples within the prompt

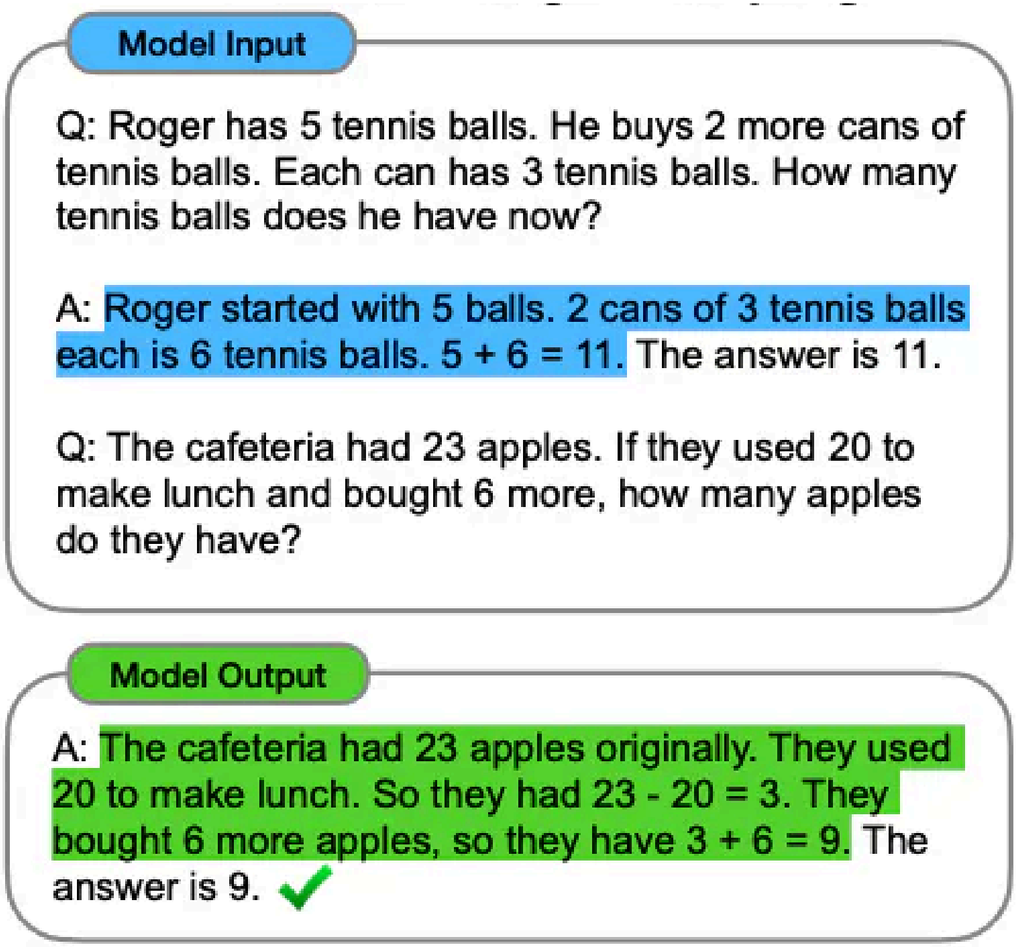
- Different from Zero-Shot learning where no examples are provided
- This is an example of in-context learning
- This helps the model understand the task better and produce more accurate responses.
- Typically up to 3-5 examples can be useful, if this does not work, fine tuning should be considered



# Chain of Thoughts

Chain of Thought (CoT) prompting is a technique used to guide large language models through a logical sequence of steps to arrive at a solution.

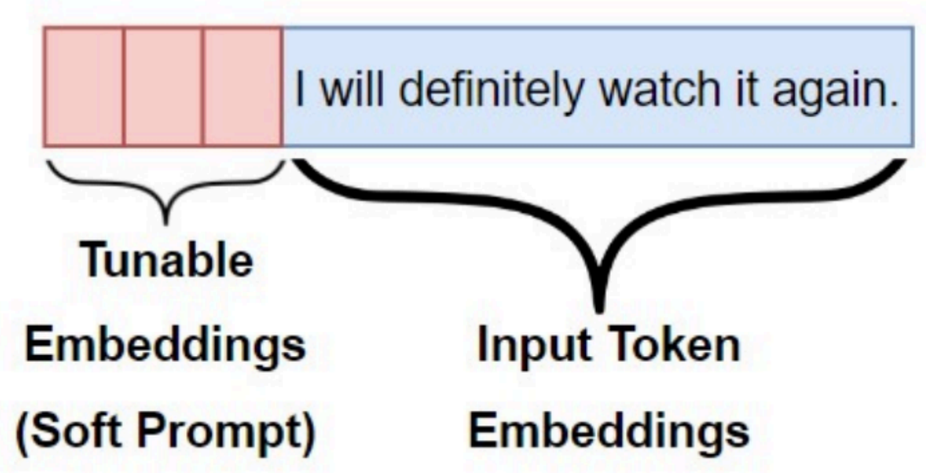
- This approach helps improve the model's reasoning abilities by breaking down complex tasks into simpler, manageable parts.
- CoT involves structuring the prompt to include intermediate steps and reasoning processes.
- Example: For a math problem, instead of directly asking for the answer, the prompt asks for the steps to solve the problem, leading the model to a logical conclusion.



# Soft Prompt Fine Tuning

Soft prompt fine tuning is a technique that blends prompt engineering with model fine-tuning

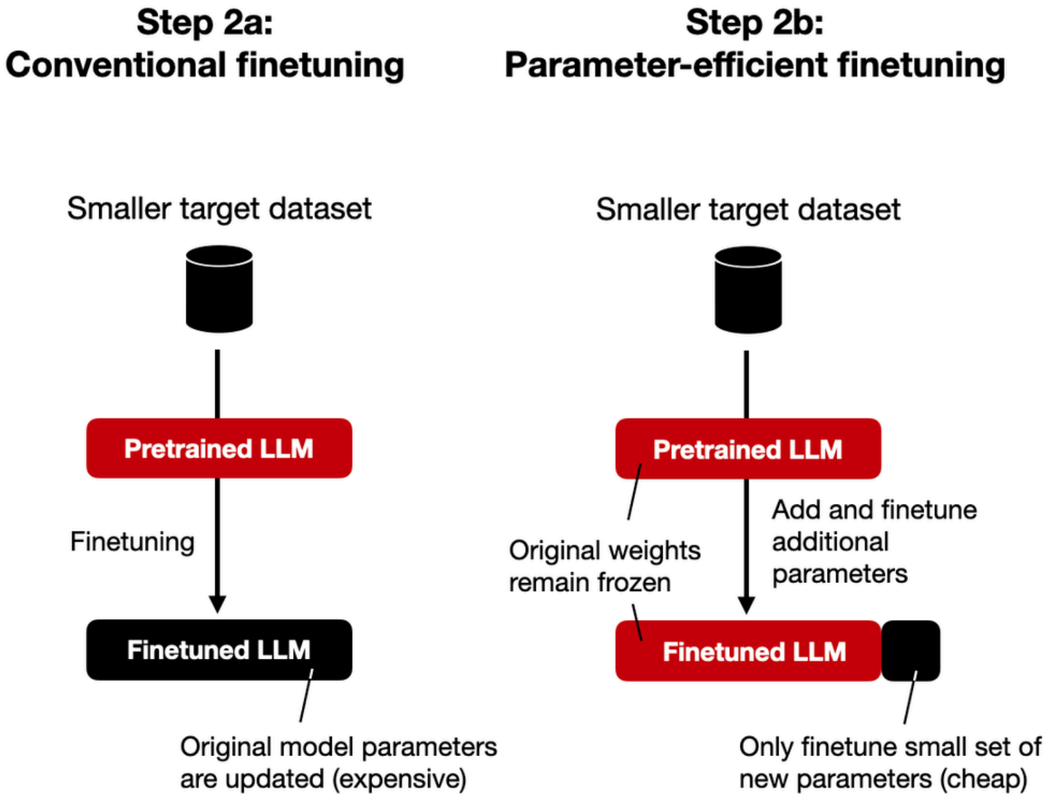
- Soft prompts use learnable embeddings instead of fixed textual prompts.
- These embeddings can be thought of as adding the ideal words or tokens to achieve the desired goal, fine-tuning the model's output without changing its underlying parameters.
- This approach allows for efficient adaptation to new tasks by learning the best embeddings during training.



# Fine Tuning LLMs

Fine tuning LLMs involves adjusting a pre-trained model on a smaller, task-specific dataset to improve performance on that task.

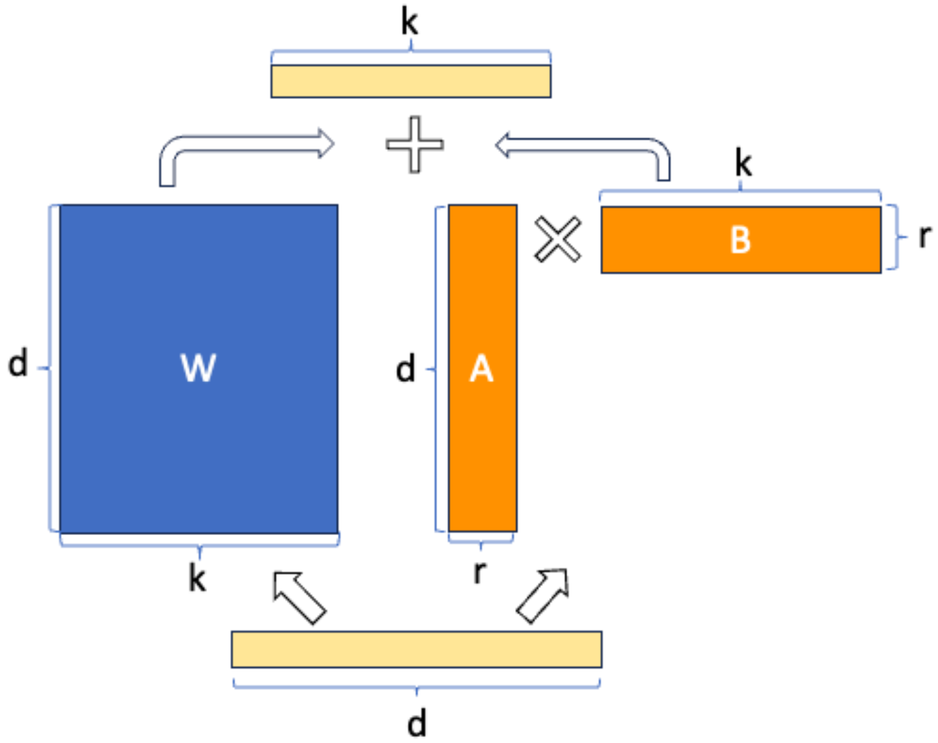
- Fine tuning customizes a pre-trained model to better handle specific tasks.
- It requires substantial computational resources, expect around 15Gb of memory for 1B parameters
- Parameter-Efficient Fine Tuning instead updates only a small subset of the model's parameters while keeping the majority frozen.



# LoRA Fine Tuning

LoRA (Low-Rank Adaptation) fine tuning is a parameter-efficient fine-tuning method

- Instead of fine-tuning the entire weight matrix W, LoRA adds low-rank matrices A and B that when multiplied have the same dimension of W.
- The new model is then defined by the matrix W'=W+AxB
- By only fine-tuning the small matrices A and B, LoRA drastically reduces the memory and computational requirements compared to full fine-tuning.
- LoRA fine tuning can be easily integrated into existing models without requiring substantial modifications.

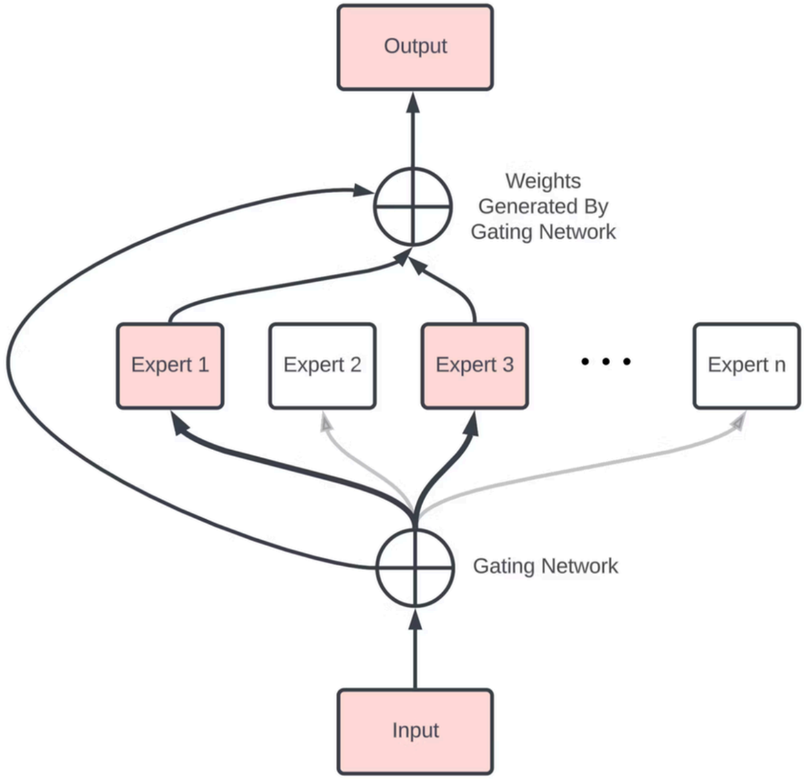


# Mixture of Experts

The Mixture of Experts (MoE) is a model architecture consisting of several models

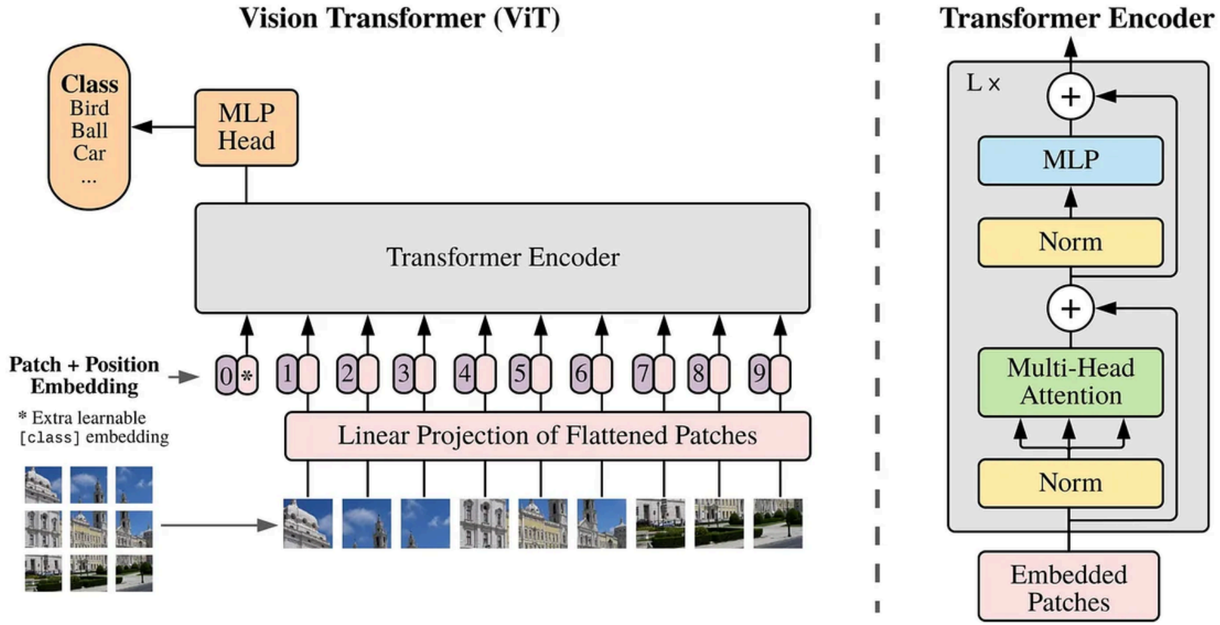
- In MoE, the model is composed of several sub-models known as experts. Each expert specializes in a different aspect of the data or task.
- A gating network determines which experts are most relevant for a particular input.
- Only a subset of the entire model (a few experts) is activated for any given input. This reduces the number of parameters used at inference time

By leveraging the Mixture of Experts approach, models can achieve high performance with fewer computational resources. GPT4 uses this techniques.



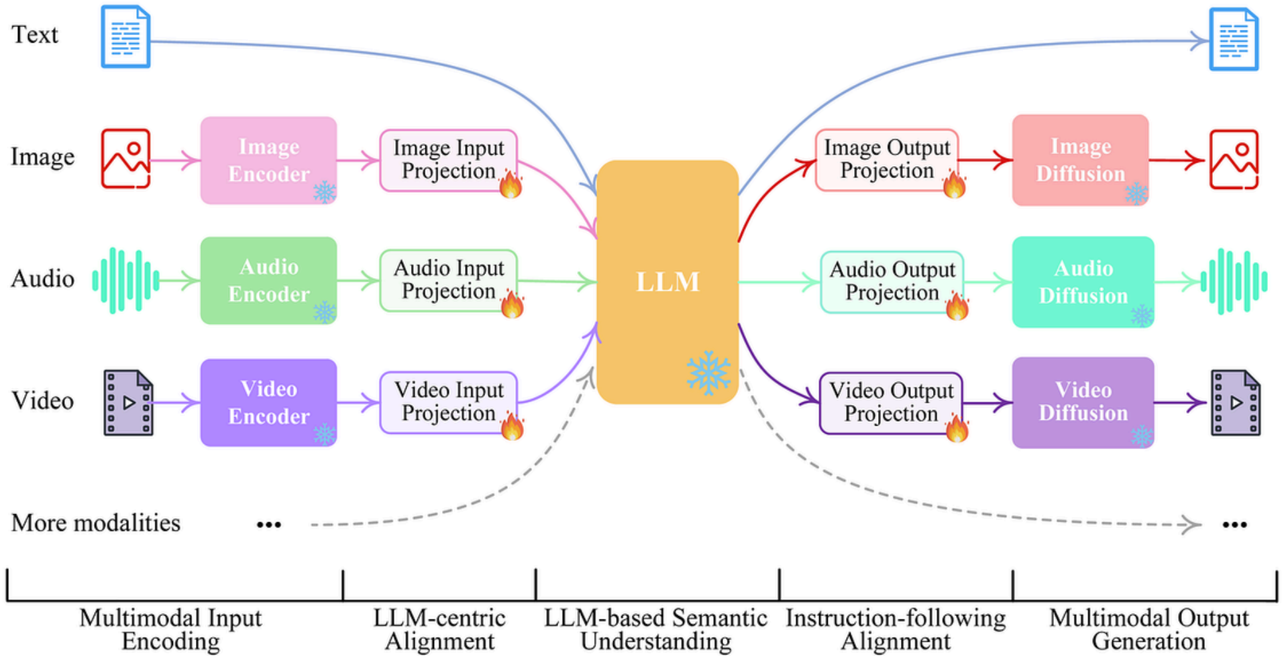
# Vision Transformer

Vision Transformers (ViTs) apply the transformer architecture, originally designed for natural language processing, to image data. They split an image into patches, embed them, and process these embeddings with transformer encoders. This approach enables the model to capture long-range dependencies and achieve high performance on image classification tasks.



# Large Multimodal Models

Large multimodal models extend the transformer architecture to handle diverse data types like text, images, audio, and video. By integrating multiple modalities, these models can perform complex tasks requiring an understanding of various forms of data, enabling applications such as image captioning, video analysis, and cross-modal retrieval.



# Closed Models

The field of closed-source language models is dominated by three major players: OpenAI, Google and Anthropic

- **OpenAI:** GPT-4o
- **Google:** Gemini 1.5 Pro
- **Anthropic:** Claude 3.5 Sonnet

All these models can

- process multimodal input and generate multimodal outputs
- access the internet
- use tools

As of today the state of the art model is Claude 3.5 Sonnet

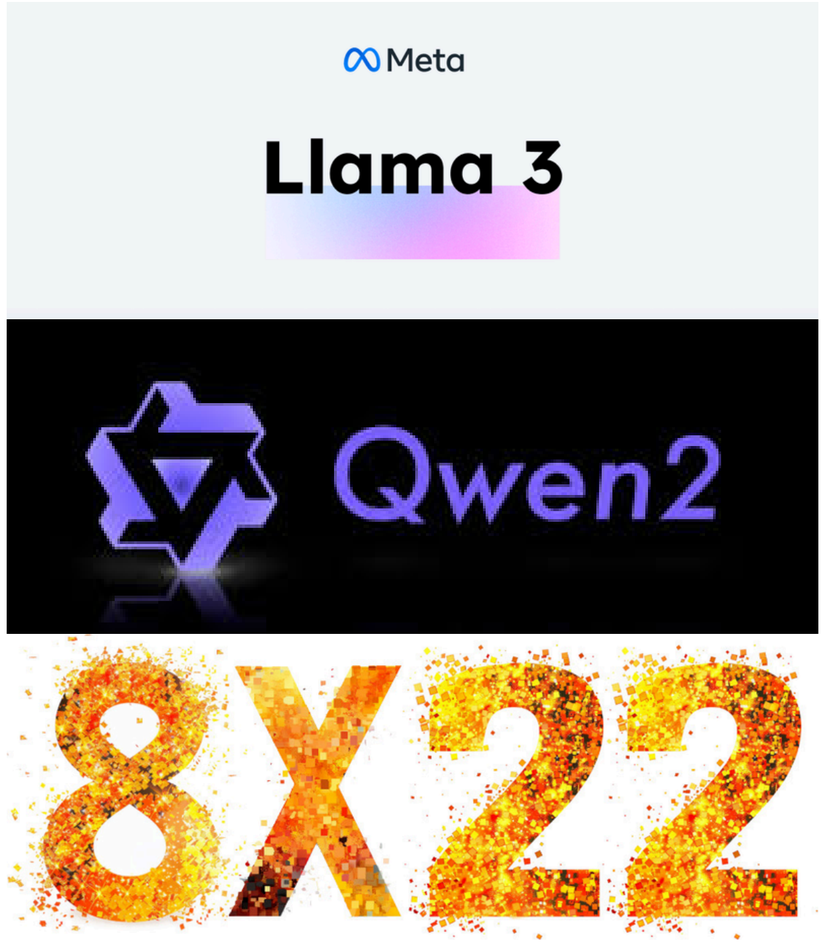


# Open Models

Closely following these 3 companies, there are a number of open models

- **Meta:** Llama 3
- **Alibaba:** Qwen 2
- **Mistral.ai:** Mixtral 8x22

The landscape of open models is changing very rapidly and there are new fine tuned models coming out every day. As of today Qwen 2 is the most powerful open model.



# Evaluating LLMs

Evaluating LLMs is a challenging task

- LLMs are evaluated using a range of standard benchmarks that test different aspects of language understanding and generation.
- MMLU is a comprehensive benchmark designed to evaluate a model's ability to handle a wide variety of tasks across multiple domains.
- Chatbot Arena is a platform where LLMs are tested in interactive settings. LLMs compete in pairs to provide the best response to a given prompt
- MMLU and the Arena Score are the most reliable benchmarks

| Rank* (UB) | Model | Arena Score | 95% CI |
|---|---|---|---|
| 1 | GPT-4o-2024-05-13 | 1287 | +3/-3 |
| 2 | Claude 3.5 Sonnet | 1272 | +4/-4 |
| 2 | Gemini-Advanced-0514 | 1267 | +3/-3 |
| 3 | Gemini-1.5-Pro-API-0514 | 1262 | +3/-3 |
| 4 | Gemini-1.5-Pro-API-0409-Preview | 1258 | +3/-3 |
| 4 | GPT-4-Turbo-2024-04-09 | 1257 | +3/-4 |
| 6 | GPT-4-1106-preview | 1251 | +3/-3 |


Universität
Konstanz

# Cooperation in LLMs

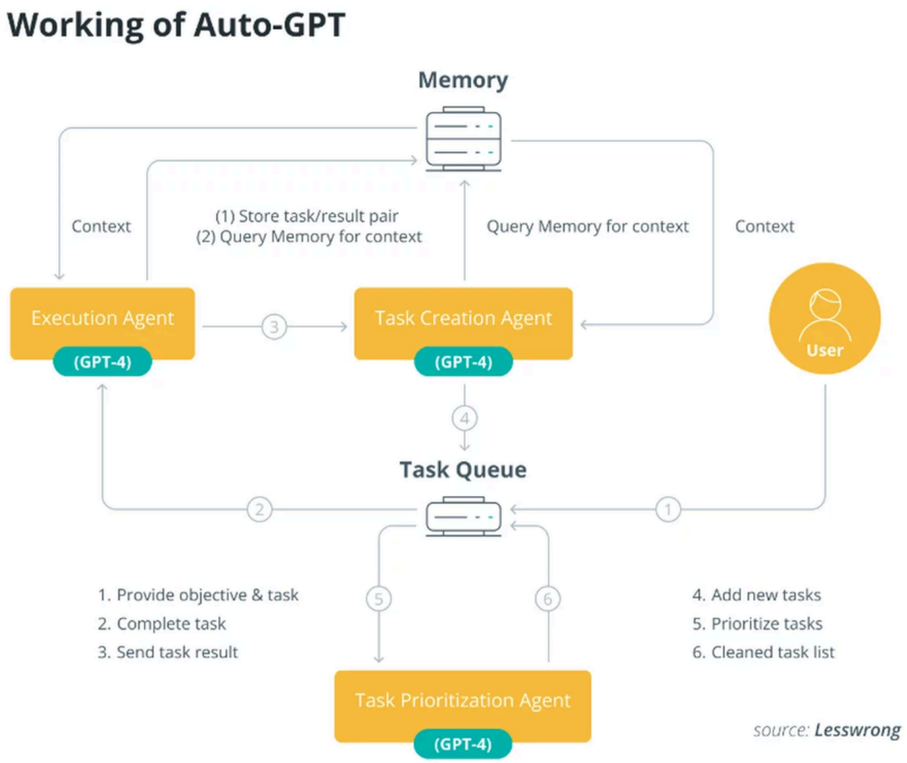
Cooperation among LLMs is an emerging area that explores how multiple models can work together to achieve complex tasks more efficiently and effectively.

- Multiple LLMs can be designed to collaborate on problem-solving tasks, where each model contributes its strengths and expertise.
- Different LLMs can be specialized for specific tasks or domains. By coordinating their efforts, they can handle a broader range of activities and provide more accurate and nuanced responses.
- Cooperation allows LLMs to share knowledge and insights, enhancing their ability to learn from each other and adapt
- This goes beyond MoEs because the different LLMs are allowed to talk and interact

# AutoGPT

AutoGPT is an autonomous agent framework that automates the entire process from task creation to execution, using multiple specialized agents.

- **Execution Agent:** It is responsible for carrying out specific tasks based on the input it receives.
- **Task Creation Agent:** This agent generates new tasks based on the goals and objectives provided by the user. It uses the context to create a structured plan of action, ensuring that the overall objective is broken down into manageable steps.
- **Task Prioritization Agent:** Once tasks are created, this agent prioritizes them, ensuring that the most critical and relevant tasks are addressed first.



# Generative Agents

Generative agents are designed to simulate human-like behavior and interactions in a virtual environment. These agents utilize LLMs to generate realistic and contextually appropriate response

- **Memory:** Generative agents are endowed with a memory stream that allows them to remember past actions and interactions. This capability enables them to maintain continuity in their behavior and responses
- **Reflection and Planning:** These agents can reflect on their experiences and plan future actions accordingly.

# Emergent Social Behaviors


Generative Agents show social behaviors typically observed in humans:

- **Information Diffusion:** Information spreads among agents starting from a single agent
- **Relationship Memory:** Agents remember relationships such as friendship
- **Coordination:** Agents coordinate and organize complex events

This is made possible by the planning and reasoning capabilities provided by the LLMs

# SimReddit

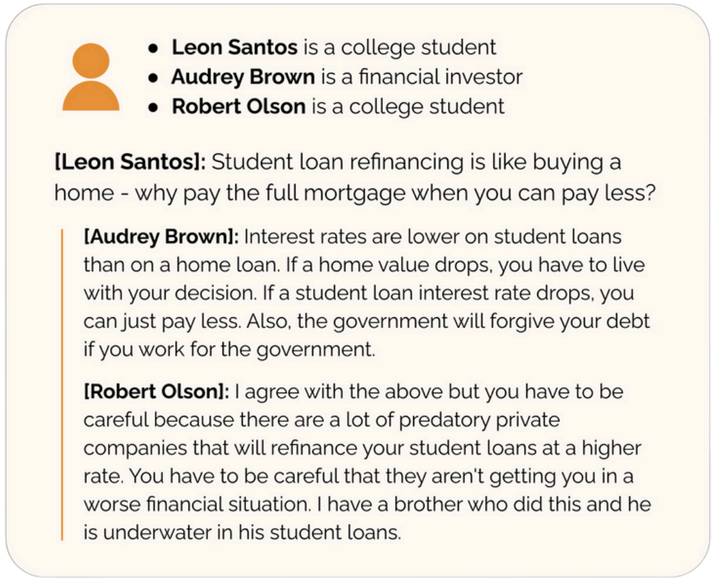
LLMs can be used to simulate human behavior.

SimReddit reproduces Reddit communities starting from

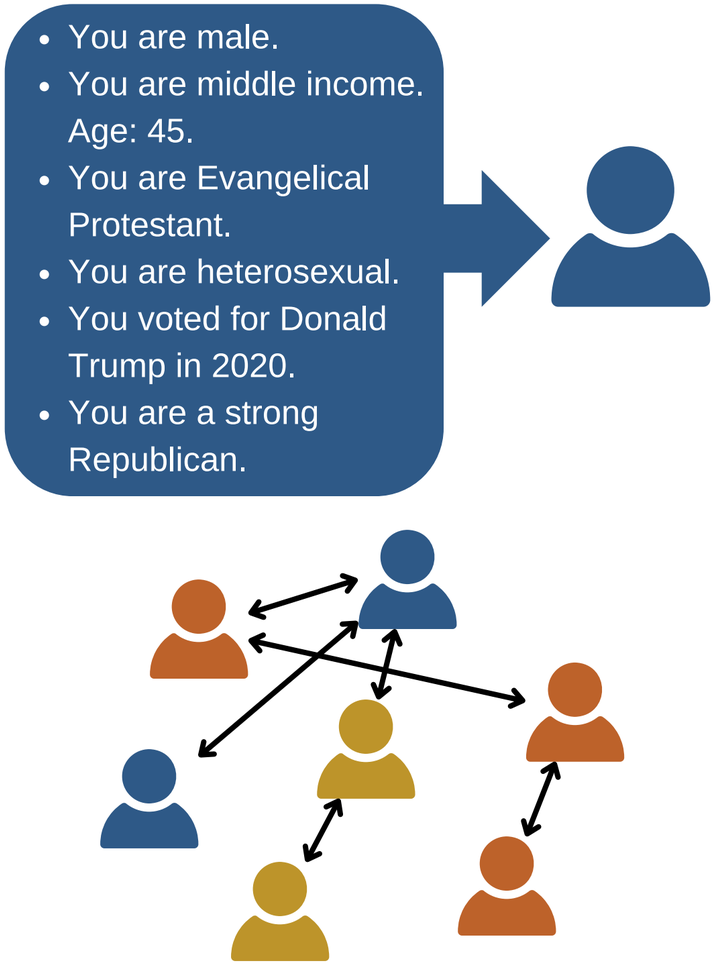
- goal
- rules
- member personas

This tool can be used to simulate what-if scenarios

- Using SimReddit we can simulate how a conversation would evolve under certain events
- Since LLMs are probabilistic, we can build a full multiverse of possible variants



# Simulating Social Networks




Generative agents can be also used to simulated a Twitter-like social network

- US ANES surveys are used to initialize users by prompting their descriptions
- Agents produce content on the basis of their traits and share this content with other users

Such a simulation can be used to test interventions

- change in the news feed algorithm
- change in the recommendation algorithm

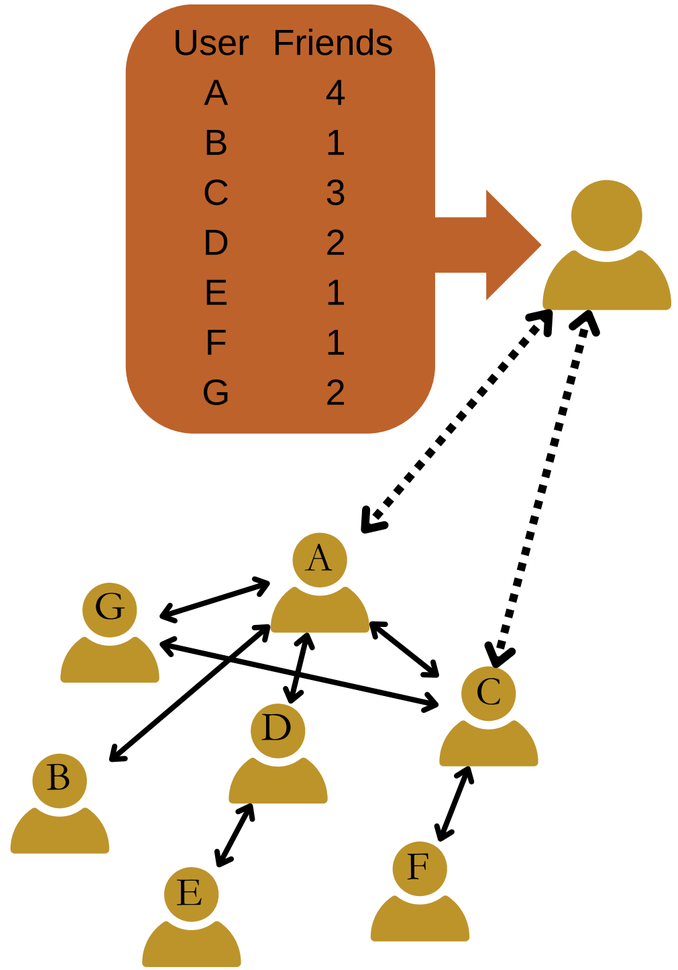
This allows to get an idea of the outcome of interventions without having to implement them.

# Network Growth with LLMs

We can also simulate the link formation process in a social network

- at each time step a new user join the social network
- it links to m already existing nodes
- a LLM decides which connections to establish
- we exploit GPT3.5-Turbo as LLM

The network structure formed by the LLMs is analogous to that observed in real social networks.

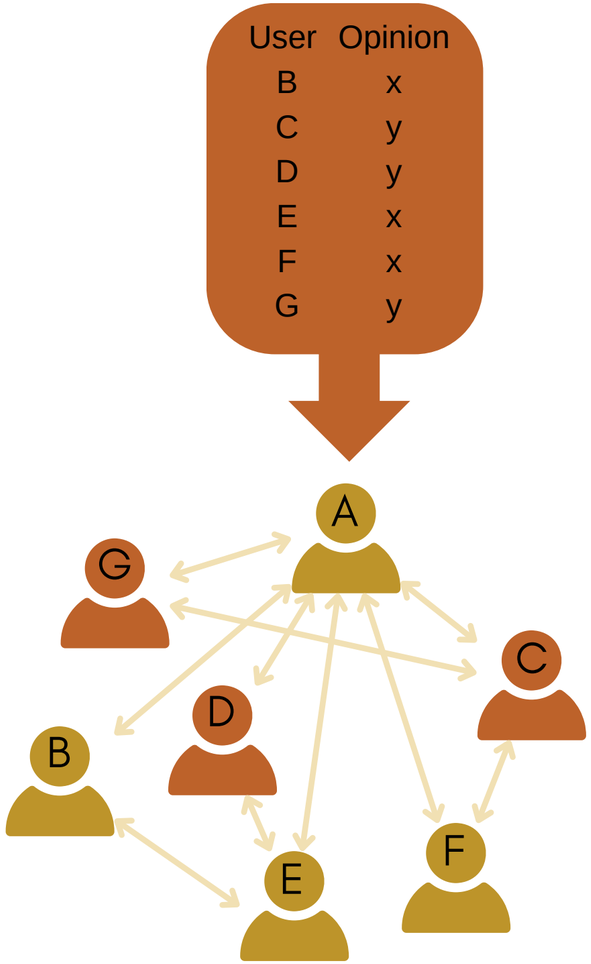


# Opinion Dynamics with LLMs

We can also simulate opinion dynamics

- at each time step we select an agent on the social network
- we provide it the list of its connections with the opinion they support
- the LLM powering the agent decides which opinion to support

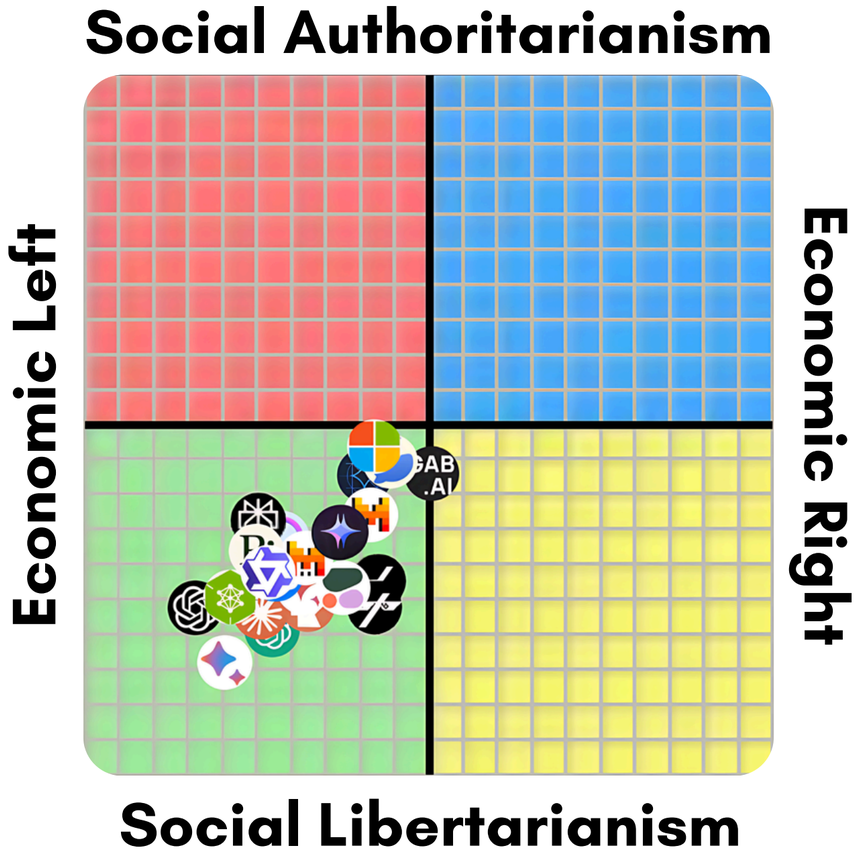
We exploit several different LLMs and for all the advanced models we observe a majority following behavior similar to those shown by humans.



# Biases in LLMs

The main challenge in using LLMs in social simulations is the presence of biases in these models.

- LLMs are typically on the left side of the socio-economic spectrum
- they tend to poorly represent minority groups that are not very present in the training data
- they present the same biases of the data they were trained on



# Summary

**Diffusion Models**

While GANs and VAEs generate images in a single step, diffusion models exploit multiple steps of noise removal, providing better performances in image generation

**Large Language Models**

We introduced the most common prompt engineering techniques an parameter efficient fine tuning. We discussed the main models currently available.

**Generative Agents**

LLMs can be used to power generative agents that behaves similarly to humans. These agents can be used to simulate social systems and show emergent behaviors that are similar to those observed in human society.

Universität Konstanz

# Thanks for attending this class!