

NLP with Deep Learning

LLMs and Agentic Workflows



Incheon Paik
University of Aizu

Contents

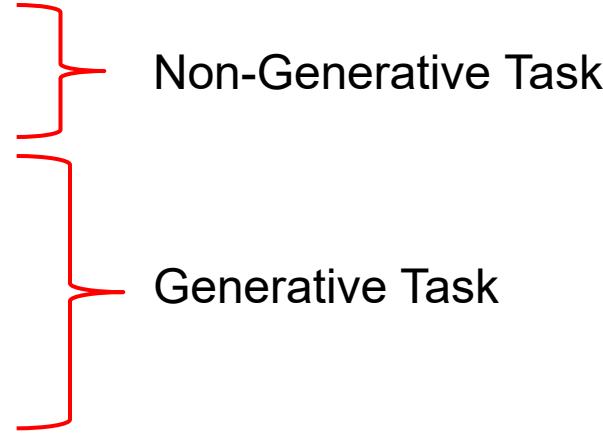
- ◆ Generative/Non-Generative AI
- ◆ Browse State-of-the-Art in DL and Case Studies
- ◆ Basics of AI
- ◆ LLM Models for AI (Size of Model, Data, Computing)
- ◆ GPT-4 and GPT-5, and Next
- ◆ LLMs and Its Rapid Evolution - Multi-Modal Learning
- ◆ Performance of Gemini,GPT-4o and GPT5
- ◆ Evolution of Interacting with LLMs
- ◆ AI Agent / Agentic Workflow
- ◆ Performance of Agent Workflow with LLM
- ◆ About AGI
- ◆ About Social Changes by AI

Two Kinds of AI & Research Topics

Generative AI and Non-Generative AI

◆ Several Tasks of Deep Learning Architectures

- Classification, Clustering, etc
- Auto Encoding
- Translation
- Summarization
- Chat
- Question and Answering



◆ DL Architectures

- MLP, CNN, RNN, LSTM, GRU, etc
- Sequence-to-Sequence and Attention
- Attention only - Transformer
- Generative Adversarial Networks (GAN)
- Bidirectional Encoder Representations from Transformers (BERT)
- Generative Pre-trained Transformer (GPT)

Browse State-of-the-Art in DL

- ◆ **Computer Vision** - Semantic Segmentation, Image Classification, Object Detection, Representation Learning, Contrastive Learning
- ◆ **NLP** - Language Modelling, Decoder, Translation, Question-Answering, Machine Translation
- ◆ **Medical** - Medical Image Segmentation, EEG, Drug Discovery, Property Prediction, Lesion Segmentation
- ◆ **Miscellaneous** - Retrieval, Reinforcement Learning, Representation Learning, General Classification, Classification
- ◆ **Time Series** - Time Series Analysis, Computational Efficiency, Time Series Forecasting, EEG, Imputation
- ◆ **Graphs** - Graph Neural Network, Link Prediction, Node Classification, Graph Learning, Graph Embedding
- ◆ **Speech** - Speech Recognition, Automatic Speech Recognition, Speech Synthesis, Dialogue Generation, Speech Enhancement
- ◆ **Audio** - Speech Synthesis, Audio Classification, Voice Conversion, Music Generation, Language Identification

Browse State-of-the-Art in DL

- ◆ **Reasoning** - Decision Making, Navigate, Common Sense Reasoning, Visual Reasoning, Logical Reasoning
- ◆ **Computer Code** - Semantic Segmentation, RL, NMT, Code Generation, Text-To-SQL
- ◆ **Playing Games** - Sentence, Continuous Control, Atari Games, Offline RL, OpenAI Gym
- ◆ **Adversarial** - Adversarial Robustness, Adversarial Attack, Open-Domain Q/A, Adversarial Defense, Backdoor Attack
- ◆ **Robots** - Benchmarking, Motion Planning, Robot Navigation, Visual Navigation, Visual Odometry
- ◆ **Knowledge Base** - Knowledge Graphs, Causal Inference, Explainable AI(XAI), Causal Discovery, Knowledge Graph Completion
- ◆ **Music** - Music Generation, Music Information Retrieval, Music Source Separation, Music Transcription, Music Modeling

Studies in Generative/Non-Generative Tasks

- ◆ HONG N.DAO, TUYEN NGUYEN, CHERUBIN MUGISHA, INCHEON PAIK, A Multimodal Transfer Learning Approach using PubMedCLIP for Medical Image Classification, IEEE Access, Accepted on May 2024.
- ◆ Kungan Zeng, Incheon Paik, Web service embedding: Representing the invocation association between services with practical-valued vectors, Expert Systems with Applications, Volume 238, Part F, DOI: <https://doi.org/10.1016/j.eswa.2023.122196>, 15 March 2024.
- ◆ Cherubin Mugisha, Incheon Paik, Medical BigBERTa: An Optimized Transformer for Long Medical Documents, Submitted to JBHI, Pre-publication at Techrxiv:
https://www.techrxiv.org/articles/preprint/Medical_BigBERTa_An_Optimized_Transformer_for_Long_Medical_Document/21673598/2 , June. 2023.
- ◆ Chérubin Mugisha,Incheon Paik, Bridging the Gap between Medical Tabular Data and NLP Predictive Models: A Fuzzy-Logic-Based Textualization Approach, MDPI Electronics 2023, 12(8), 1848;
<https://doi.org/10.3390/electronics12081848>, Apr. 2023.
- ◆ Cherubin Mugisha,Incheon Paik, Comparison of Neural Language Modeling Pipelines For Outcome Prediction From Unstructured Medical Text Notes, IEEE Access (Volume: 10), DOI: 10.1109/ACCESS.2022.3148279, Page: 16489-16498, Feb. 2022.

Studies in Generative/Non-Generative Tasks

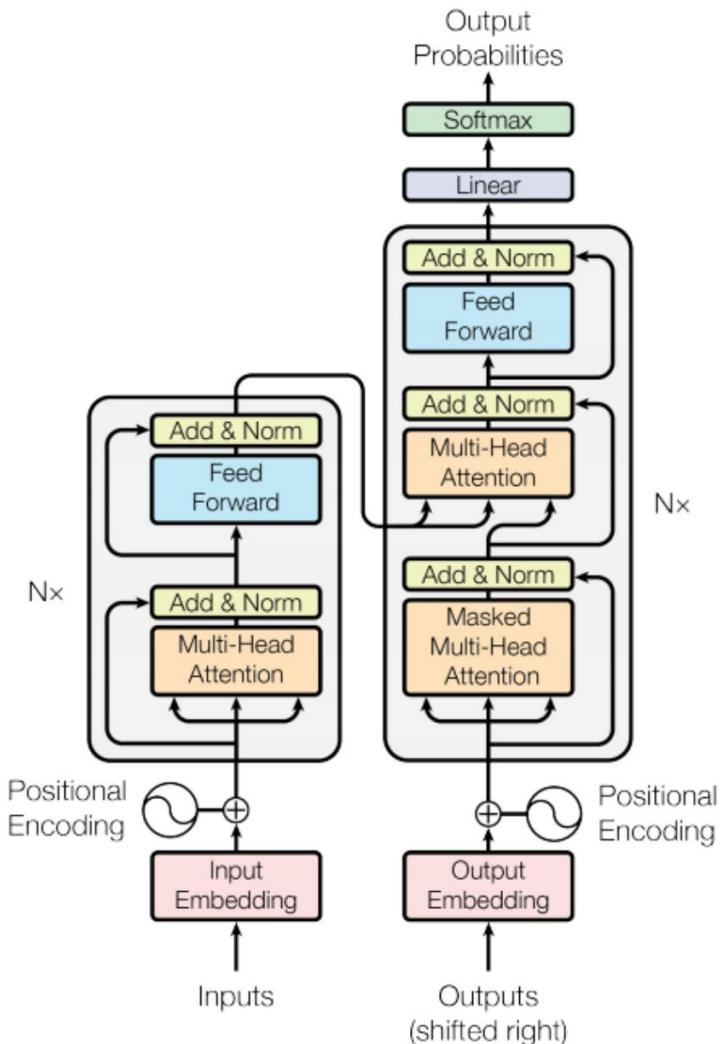
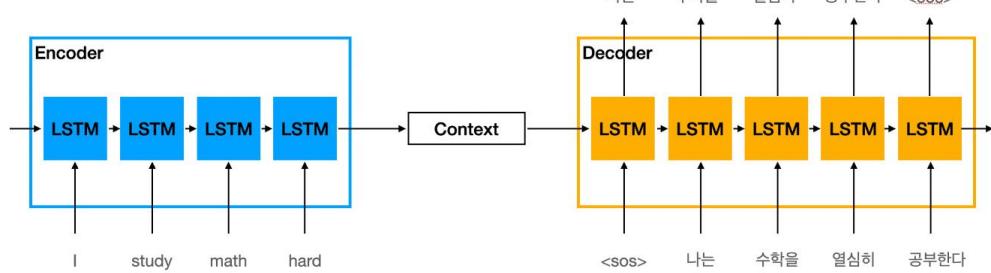
- ◆ Ranto Sawai, Incheon Paik, Ayato Kuwana, Sentence Augmentation for Language Translation using GPT-2, Journal of MDPI Electronics 2022, 10, 3082. <https://doi.org/10.3390/electronics10243082>, Dec. 10, 2021.
- ◆ Incheon Paik, Jun-Wei Wang, Improving Text-to-Code Generation with Features of Code Graph on GPT-2, Journal of MDPI Electronics, <https://doi.org/10.3390/electronics10212706>, Nov., 2021.
- ◆ Ayato Kuwana, Atsushi Oba, Ranto Sawai, Incheon Paik, Automatic Taxonomy Classification by Pretrained Language Model, Journal of MDPI Electronics, <https://doi.org/10.3390/electronics10212656>, Oct. 2021.
- ◆ Kungan Zeng, Incheon Paik, Semantic Service Clustering with Lightweight BERT-based Service Embedding Using Invocation Sequences, IEEE Access (Volume: 9), DOI: 10.1109/ACCESS.2021.3069509, pp. 54298 - 54309, March 2021.
- ◆ Quang-Minh Do, Kungan Zeng and Incheon Paik, Resolving Lexical Ambiguity in English-Japanese Neural Machine Translation, 3rd Artificial Intelligence and Cloud Computing Conference(AICCC 2020), Kyoto, Japan, Dec. 2020.

Basics of AI (Deep Learning) & Size of LLMs

Sequence to Sequence, Transformer

◆ Transformer

◆ Sequence to Sequence



GPT-2

◆ Evolution of Transformer Block



THE TRANSFORMER

DECODER BLOCK

Feed Forward Neural Network

Encoder-Decoder Self-Attention

Masked Self-Attention

Input

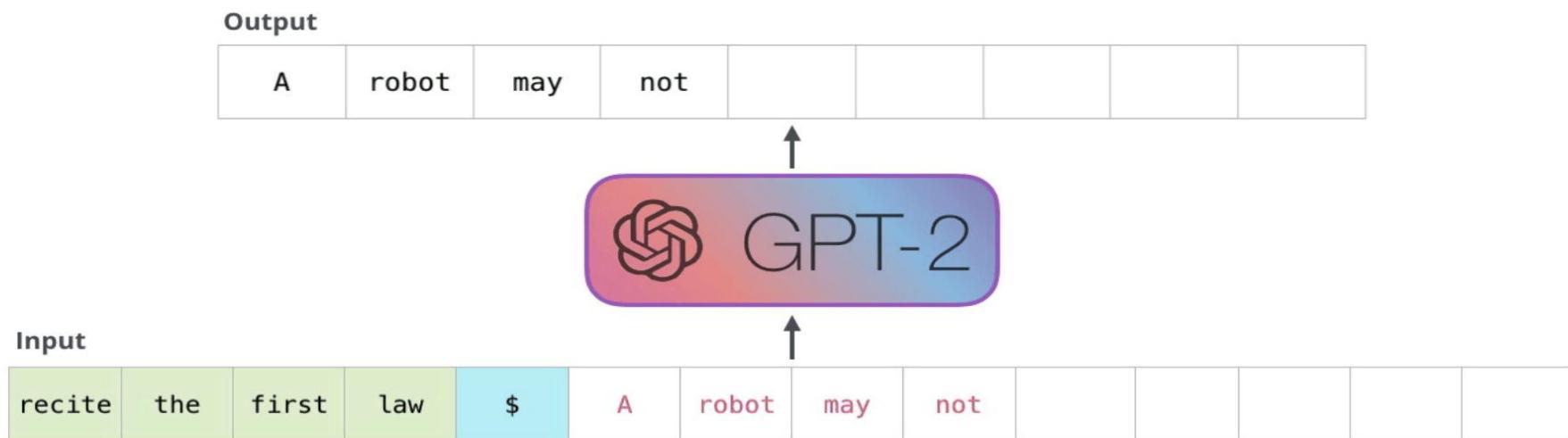
<s>	robot	must	obey					512
1	2	3	4	5	6			

Reference: <http://jalammar.github.io/illustrated-gpt2/>

GPT-2

◆ Work of GPT-2

The way these models actually work is that after each token is produced, that token is added to the sequence of inputs. And that new sequence becomes the input to the model in its next step. This is an idea called “auto-regression”. This is one of the ideas that made RNNs unreasonably effective.



The GPT2, and some later models like TransformerXL and XLNet are auto-regressive in nature. BERT is not. That is a trade off. In losing auto-regression, BERT gained the ability to incorporate the context on both sides of a word to gain better results. XLNet brings back autoregression while finding an alternative way to incorporate the context on both sides.

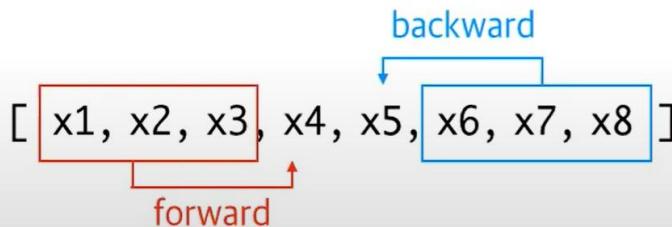
Reference: <http://jalammar.github.io/illustrated-gpt2/>

Autoregressive LLM

Autoregressive vs Autoencoding

- AR language model (GPT)

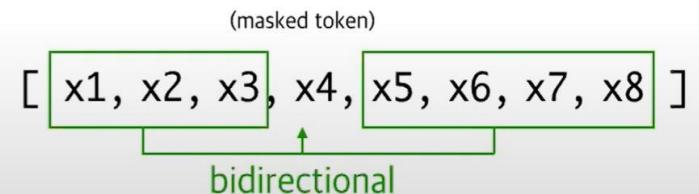
$$\max_{\theta} \log p_{\theta}(x) = \sum_{t=1}^T \log p_{\theta}(x_t | x_{<t})$$



Good at text generation

- AE language model (BERT)

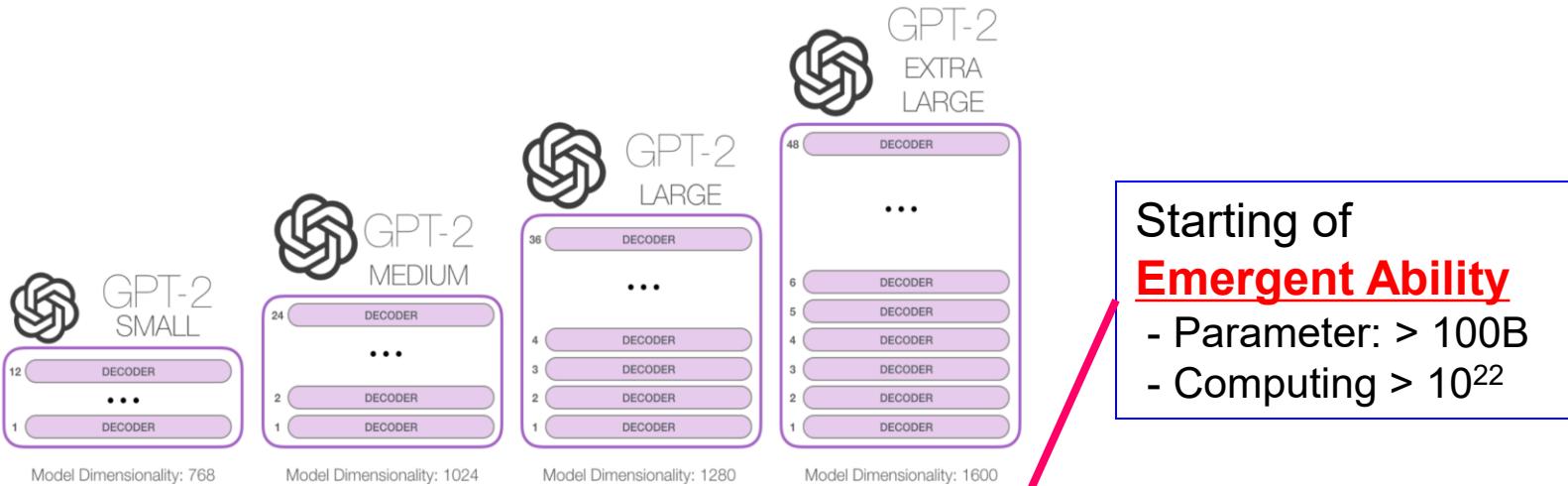
$$\max_{\theta} \log p_{\theta}(\bar{x} | \hat{x}) \approx \sum_{t=1}^T m_t \log p_{\theta}(x_t | \hat{x})$$



Good at language understanding

GPT-2

◆ Several Models of GPT-2



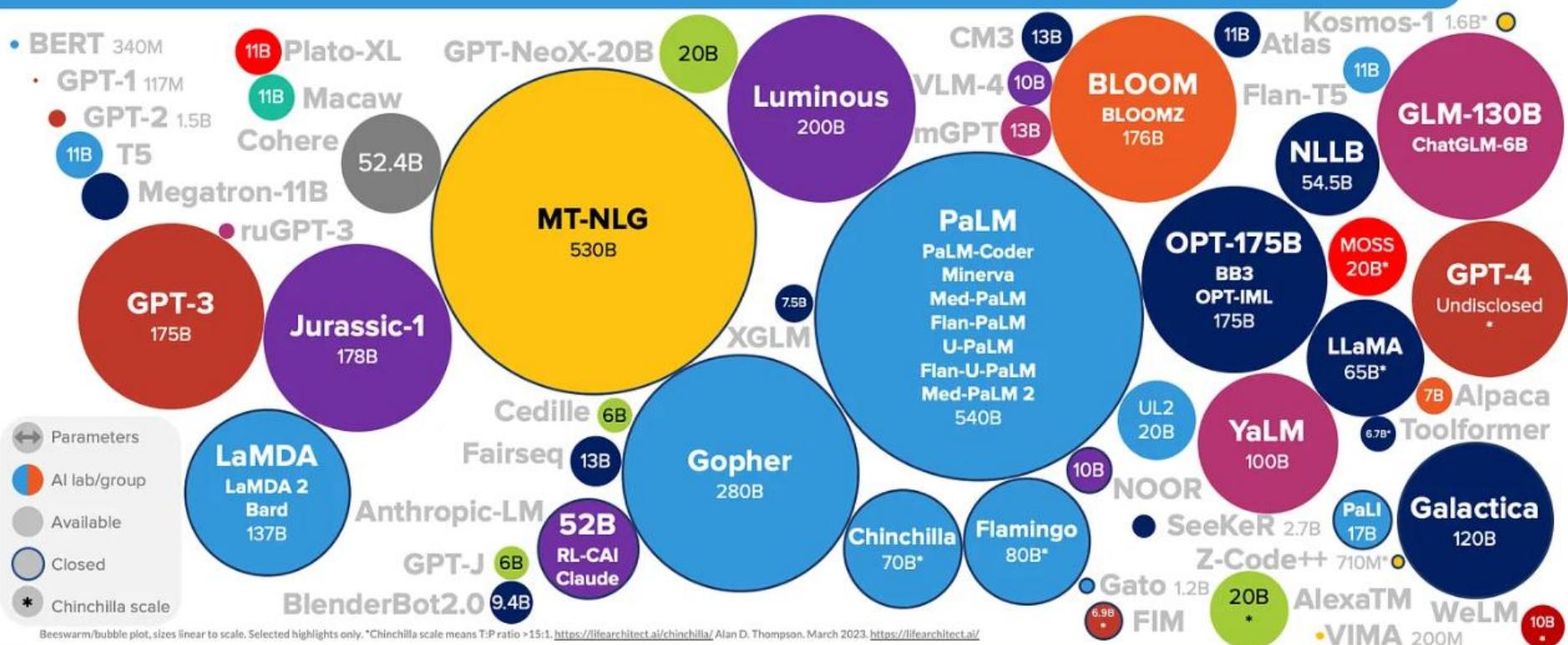
	GPT-1	GPT-2	GPT-3
Parameters	117 Million	1.5 Billion	175 Billion
Decoder Layers	12	48	96
Context Token Size	512	1024	2048
Hidden Layer	768	1600	12288
Batch Size	64	512	3.2M

Reference: <http://jalammar.github.io/illustrated-gpt2/>

LLM Models for AI

◆ Increase in Model Size

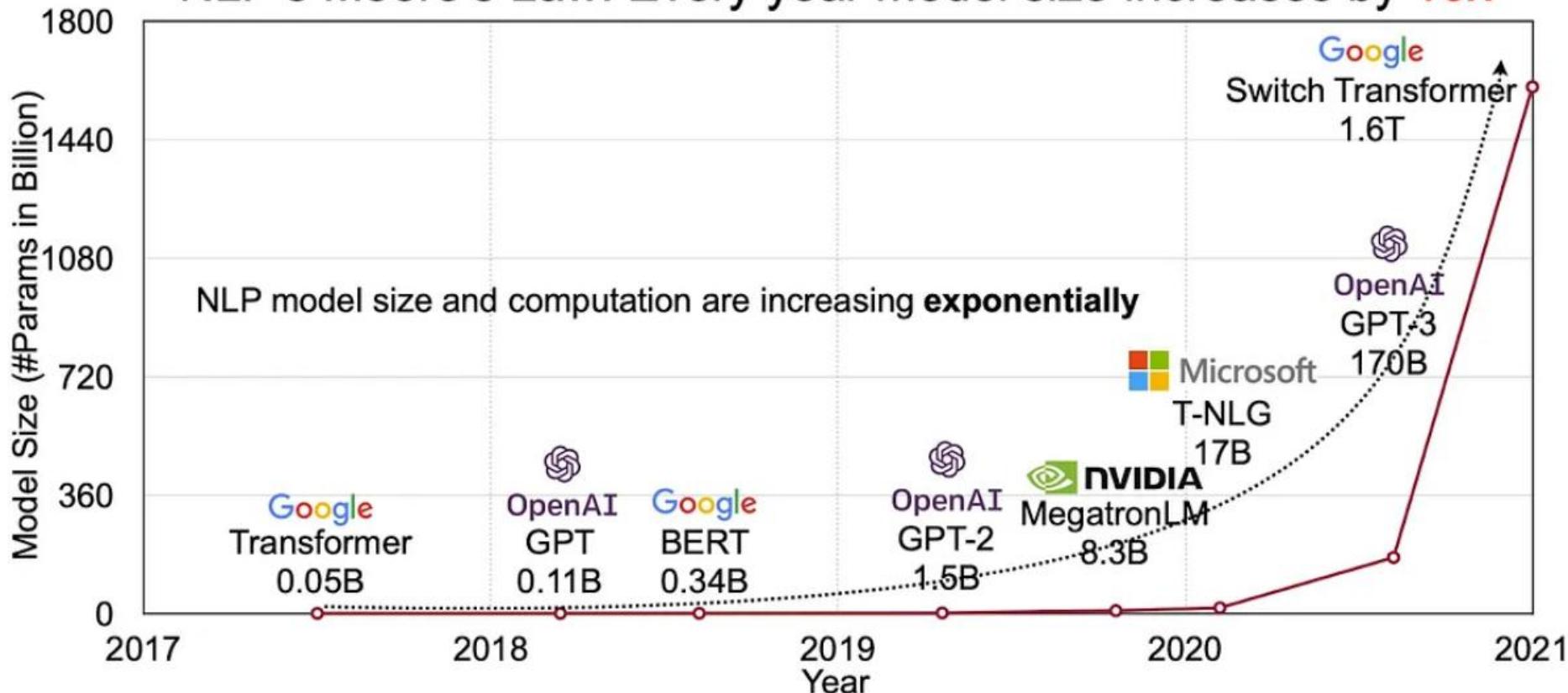
LANGUAGE MODEL SIZES TO MAR/2023



LLM Models for AI

◆ Increase in Model Size

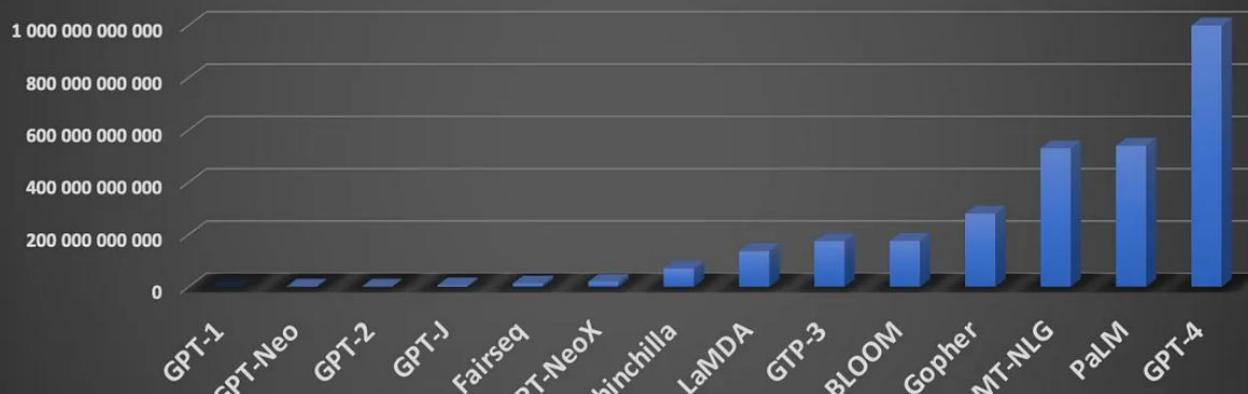
NLP's Moore's Law: Every year model size increases by **10x**



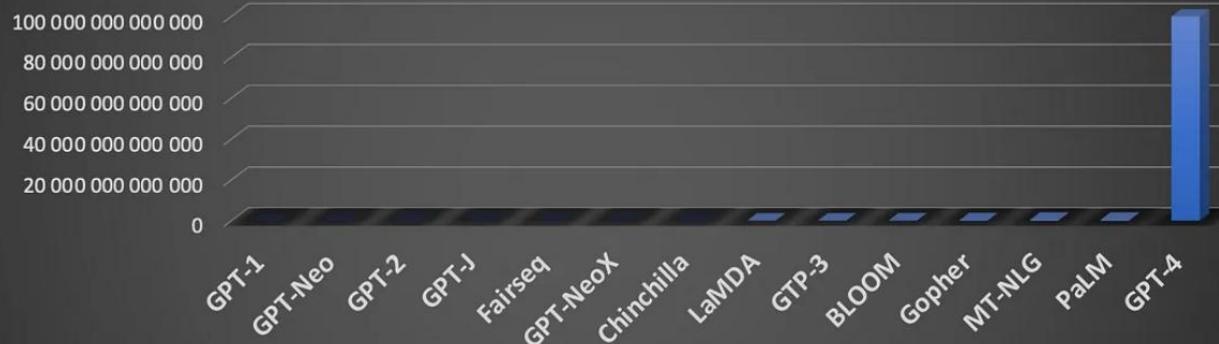
LLM Models for AI

◆ Increase in Model Size

GPT-4 @ 1 Trillion Parameters



GPT-4 @ 100 Trillion Parameters



LLM Models for AI

◆ Increase in Model Size

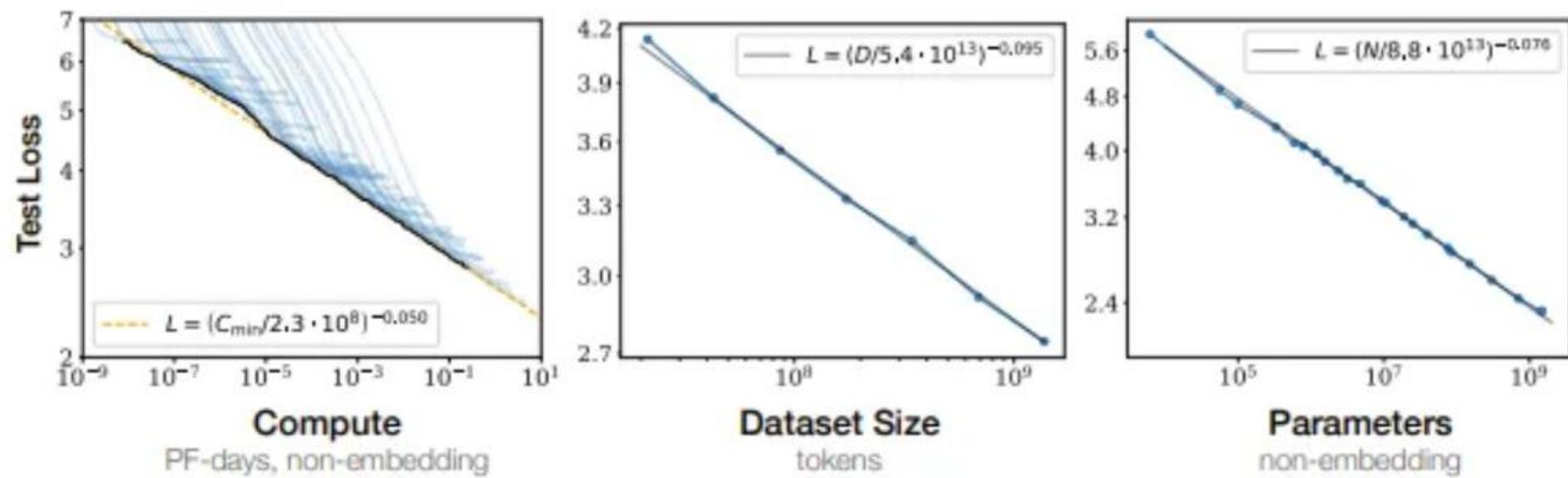


Figure 1 Language modeling performance improves smoothly as we increase the model size, dataset size, and amount of compute² used for training. For optimal performance all three factors must be scaled up in tandem. Empirical performance has a power-law relationship with each individual factor when not bottlenecked by the other two.

LLM Models for AI

◆ Model Training Compute

Model	Training end	Chip type	TFLOP/s (max)	Chip count	Wall clock (days)	Total time (years)	Retail (US\$)	MMLU
GPT-3 175B	Apr/2020	V100	130	10,000	15 days	405y	\$9M	43.9
Llama 1 65B	Jan/2023	A100	312	2,048	21 days	118y	\$4M	63.4
Llama 2 70B	Jun/2023	A100	312	2,048	35 days	196y	\$7M	68.0
Titan 200B	Apr/2023	A100	312	13,760	48 days	1,319y	\$45M	70.4
GPT-4 1.7T	Aug/2022	A100	312	25,000	95 days	6,507y	\$224M	86.4
Gemini	Nov/2023	TPUv4	275	57,000	100 days	15,616y	\$440M	90.0
Llama 3 405B	Apr/2024	H100	989	24,576	50 days	3,366y	\$125M	85+
GPT-5	Apr/2024	H100	989	50,000	120 days	16,438y	\$612M	
Grok 2	Jun/2024	H100	989	20,000	50 days	6,571y	\$245M	
Olympus	Aug/2024	H100	989					
Gemini 2	Nov/2024	TPUv6	1,847					
Grok 3	Dec/2024	H100	989	100,000	50 days	32,855y	\$1.2B	

Alan D. Thompson. May/2024. LifeArchitect.ai

LLM Models for AI

◆ The GPT-4 Model Family: In Plain English

THE GPT-4 MODEL FAMILY: IN PLAIN ENGLISH

Size estimate	Model	Release date ▼	Data cutoff	MMLU score	GPQA score
	GPT-4 Classic	gpt-4-0314	14/Mar/2023	Sep/2021	86.4
		gpt-4-0613	13/Jun/2023	Sep/2021	-
	GPT-4 Turbo	gpt-4-1106-preview	6/Nov/2023	Apr/2023	▼ 84.7
		gpt-4-0125-preview	25/Jan/2024	Dec/2023	▲ 84.8
		gpt-4-turbo-2024-04-09	9/Apr/2024	Dec/2023	▲ 86.7
	GPT-4o Omni	gpt-4o-2024-05-13	13/May/2024	Oct/2023	▲ <u>88.7</u>
		gpt-4o-mini-2024-07-18	18/Jul/2024	Oct/2023	▼ 82.0
					▲ <u>53.6</u>
					▼ 40.2

Showing highest of chatgpt/assistant pairs from official evals: <https://github.com/openai/simple-evals/blob/main/README.md>. Additional GPT-4 data from <https://www.alignmentforum.org/posts/JbE7KvnwshwkXPJAJ/anthropic-release-claude-3-claims-greater-than-gpt-4>. Alan D. Thompson. 2024.

LLM Models for AI

◆ The GPT-4 Model Family: In Plain English

DATASETS FOR LANGUAGE MODELS: SIMPLE VIEW

APR/
2023

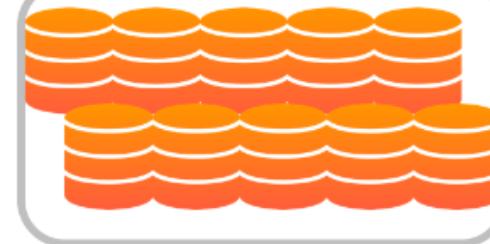
Close to scale. Selected highlights only. Using images from FlatIcon.com. Alan D. Thompson, April 2023. <https://lifearchitecture.com>



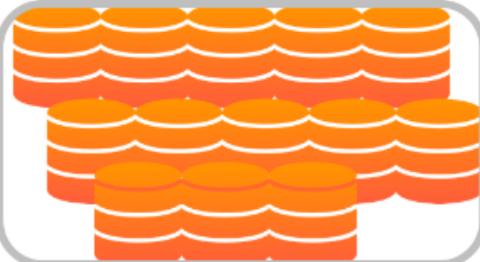
GPT-3 dataset
499B tokens / 0.75TB



The Pile dataset (GPT-Neo)
247B tokens / 0.8TB



MassiveText dataset (Chinchilla)
2.3T tokens / 10.5TB



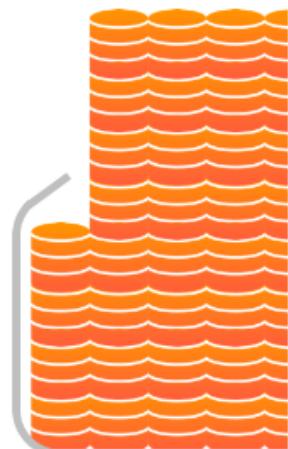
Infiniset dataset (LaMDA)
2.8T tokens / 12.6TB



Stability The Pile dataset
1.5T tokens / 5TB



RedPajama dataset
1.2T tokens / 4TB



GPT-4 (estimate)
20T tokens / 40TB

Development of LLMs

- ◆ Dramatic Fast Development of LLM (Multi-Modal) after ChatGPT 3.5 (Oct. 2022) During 1.5 years.
 - GPT 4
 - Gemini
 - Claude, etc
 - GPT4o
 - Agentic Workflow with LLM
 - GPT5
 - AGI?

LLMs and Its Rapid Evolution

GPT4

Sparks of Artificial General Intelligence: Early experiments with GPT-4

Sébastien Bubeck Varun Chandrasekaran Ronen Eldan Johannes Gehrke
Eric Horvitz Ece Kamar Peter Lee Yin Tat Lee Yuanzhi Li Scott Lundberg
Harsha Nori Hamid Palangi Marco Tulio Ribeiro Yi Zhang

Microsoft Research

Abstract

Artificial intelligence (AI) researchers have been developing and refining large language models (LLMs) that exhibit remarkable capabilities across a variety of domains and tasks, challenging our understanding of learning and cognition. The latest model developed by OpenAI, GPT-4 [Ope23], was trained using an unprecedented scale of compute and data. In this paper, we report on our investigation of an early version of GPT-4, when it was still in active development by OpenAI. We contend that (this early version of) GPT-4 is part of a new cohort of LLMs (along with ChatGPT and Google’s PaLM for example) that exhibit more general intelligence than previous AI models. We discuss the rising capabilities and implications of these models. We demonstrate that, beyond its mastery of language, GPT-4 can solve novel and difficult tasks that span mathematics, coding, vision, medicine, law, psychology and more, without needing any special prompting. Moreover, in all of these tasks, GPT-4’s performance is strikingly close to human-level performance, and often vastly surpasses prior models such as ChatGPT. Given the breadth and depth of GPT-4’s capabilities, we believe that it could reasonably be viewed as an early (yet still incomplete) version of an artificial general intelligence (AGI) system. In our exploration of GPT-4, we put special emphasis on discovering its limitations, and we discuss the challenges ahead for advancing towards deeper and more comprehensive versions of AGI, including the possible need for pursuing a new paradigm that moves beyond next-word prediction. We conclude with reflections on societal influences of the recent technological leap and future research directions.

<https://arxiv.org/pdf/2303.12712.pdf>

GPT4

GPT-4

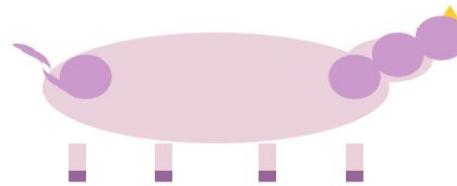
Prompt: Can you write a proof that there are infinitely many primes, with every line that rhymes?

GPT-4:

```
Yes, I think I can,
Though it might take a clever plan.
I'll start by noting Euclid's proof,
Which shows that primes aren't just aloof.
Assume we have a finite list,
Of primes, and that none have been missed.
Multiply them all together,
And add one, just to be clever.
The result will be a number,
That has no prime factors, oh wonder!
But every number has a prime,
That divides it, it's just a matter of time.
So we've found a contradiction,
And our finite list needs eviction.
There must be infinitely many primes,
And that's the end of my rhyming lines.s
```

Prompt: Draw a unicorn in TiKZ.

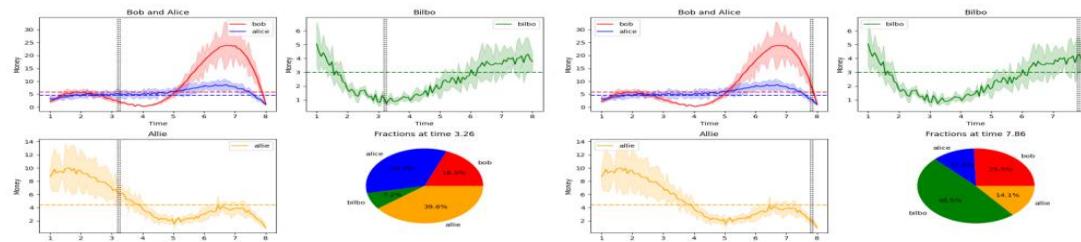
GPT-4: [Produces L^AT_EX compiling to following picture.]



Prompt: Can you generate a pyplot for the following data: $x = [1, 3, 5, 6, 8]$, $y_1 = [2, 3, 5, 18, 1]$, $y_2 = [3, 5, 6, 8, 1]$, $y_3 = [5, 1, 2, 3, 4]$, $y_4 = [9, 7, 2, 3, 1]$. I want y_1 , y_2 to be in the same plot, but y_3 is in another plot next to that plot, y_4 is in below. I want the legend of y_1 to be "bob", y_2 to be "alice", y_3 to be "bilbo", y_4 to be "allie". I want the x -axis to be labeled with "time" and y axis to be labeled with "money". I want to add a 10 %-40% random error bar to each curve, through all times (including non-integers). I want smoothed curves to show the plot, and smoothed error bar. Do not use linear interpolation, use smooth interpolation! I want to also add some small zig-zag to the smoothed curve to make it look more real. I want to put a baseline as the mean of each line. I want to put a pie chart below indicating the fraction of the four people in each time step. I also want an animation to show how the fractions are changing in the pie chart in **continuous time**. Interpolate the missing fractions! I also want vertical line animation in other three plots to match the pie chart. I want the fanciest plot. Please add as many fancy things as possible.

Here is a possible code to generate a pyplot for the following data, with some comments to explain the steps:

GPT-4: [We give two snapshots from the animation resulting from the produced code]



Prompt: Suppose $g(x) = f^{-1}(x)$, $g(0) = 5$, $g(4) = 7$, $g(3) = 2$, $g(7) = 9$, $g(9) = 6$ what is $f(f(f(6)))$?

GPT-4: There are different ways to approach this problem, but one possible method is to use the fact that $g(x) = f^{-1}(x)$ implies that $f(g(x)) = x$ for any x in the domain of g . That is, applying f to both sides of the equation $g(x) = f^{-1}(x)$ gives $f(g(x)) = f(f^{-1}(x)) = x$. This means that we can undo the effect of f by applying g , and vice versa. Using this idea, we can try to find $f(6)$ by looking for the value of x such that $g(x) = 6$. From the given information, we see that $g(9) = 6$, so $f(6) = f(g(9)) = 9$. Similarly, we can find $f(9)$ by looking for the value of x such that $g(x) = 9$. We see that $g(7) = 9$, so $f(9) = f(g(7)) = 7$. Finally, we can find $f(7)$ by looking for the value of x such that $g(x) = 7$. We see that $g(4) = 7$, so $f(7) = f(g(4)) = 4$. Therefore, $f(f(f(6))) = f(f(9)) = f(7) = 4$.

GPT-4 Training - Guess

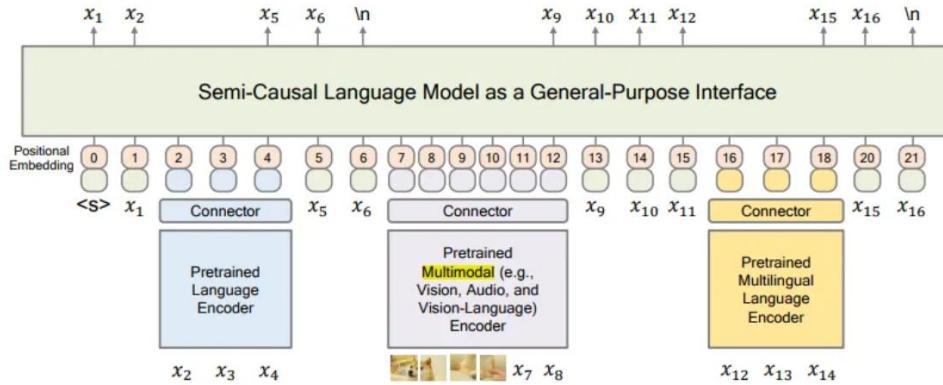


Figure 2: Overview of METALM. The semi-causal language model serves as a general-purpose interface and supports interactions with various foundation models.

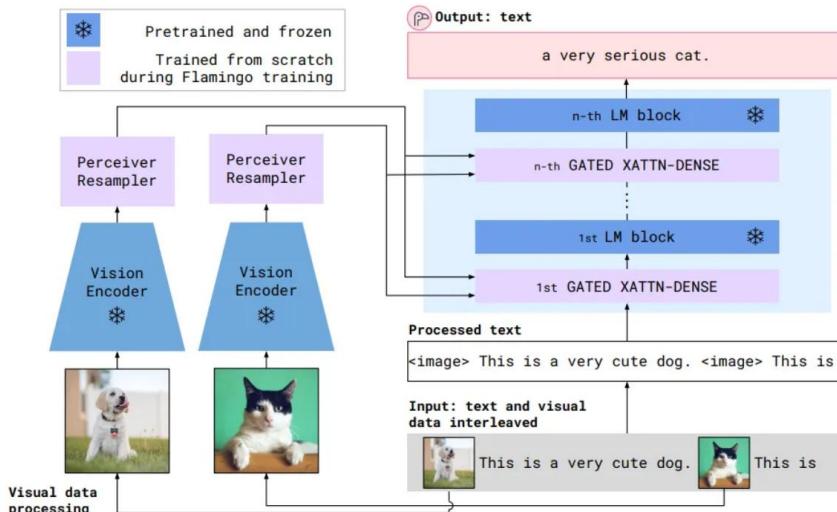


Figure 4: Overview of the Flamingo model. The Flamingo models are a family of visual language model (VLM) that can take as input visual data interleaved with text and can produce free-form text as output. Key to its performance are novel architectural components and pretraining strategies. [Source: DeepMind]

Google Gemini

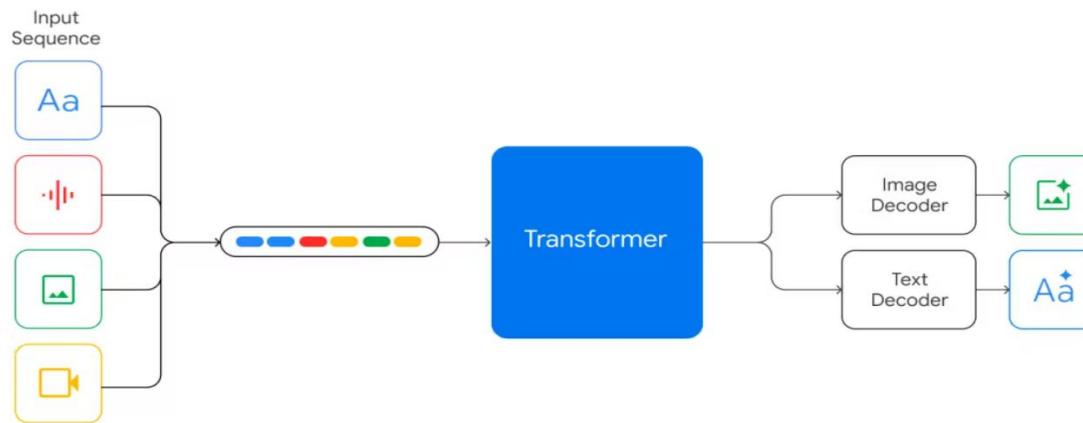


Figure 2 | Gemini supports interleaved sequences of text, image, audio, and video as inputs (illustrated by tokens of different colors in the input sequence). It can output responses with interleaved image and text.

Google Gemini: A Family of Highly Capable Multimodal Models

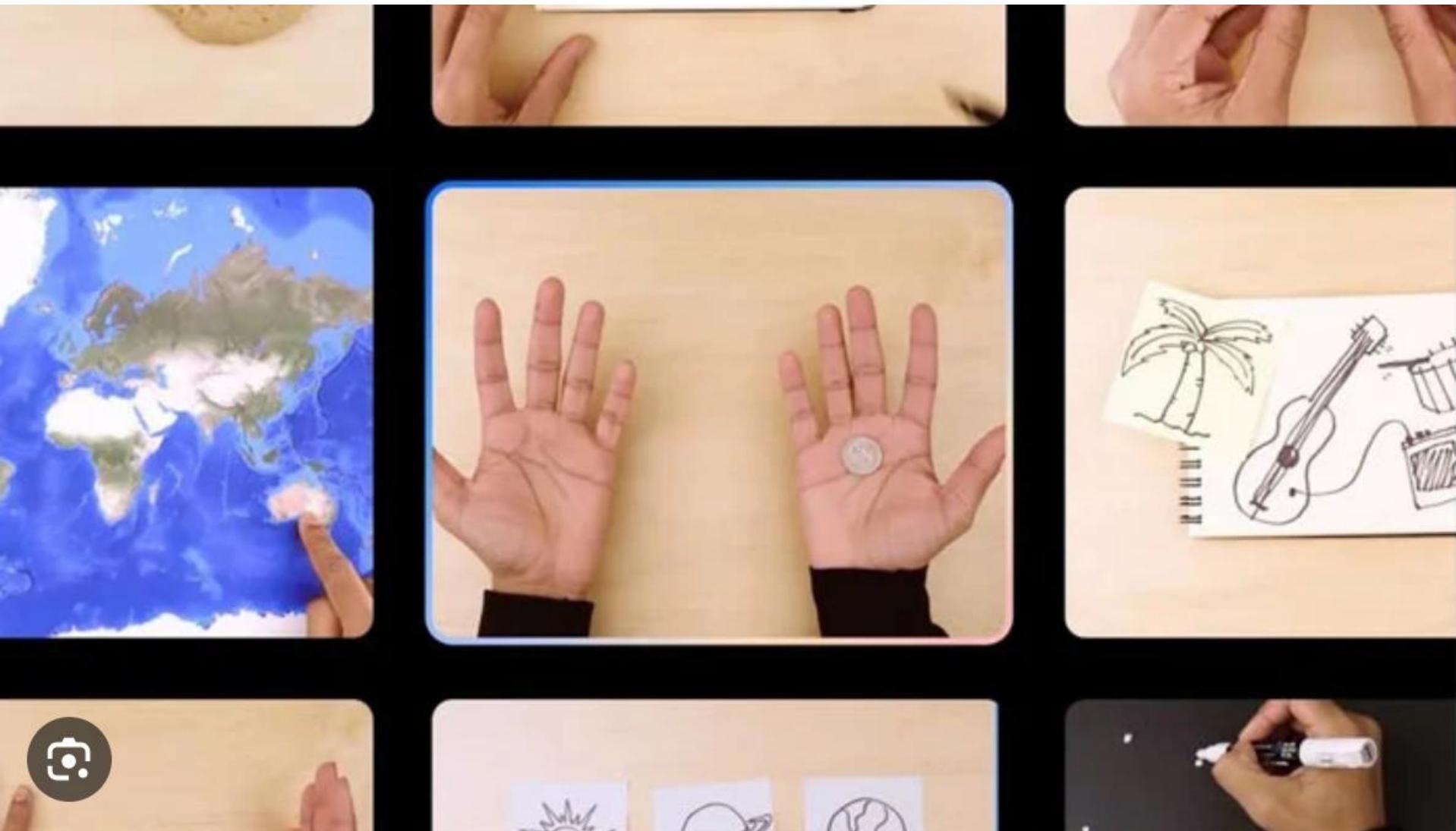
Input sequence: The user provides inputs in various formats—text, images, audio, video, 3D models, graphs, etc.

Encoder: The encoder takes these inputs and converts them into a common language that the decoder can understand. This is done by transforming the different data types into a unified representation.

Model: The encoded inputs are then fed into the model. The multi-modal model doesn't need to know the specifics of the task. It simply processes the inputs based on the task at hand.

Image and text decoder: The decoder takes the processed inputs from the model and generates the outputs. As of this time, Gemini can only generate text and image outputs.

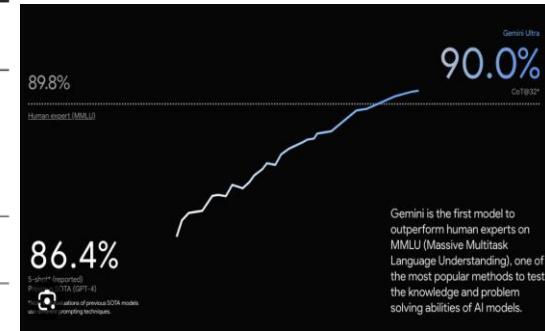
Google Gemini



Performance of Google Gemini

Gemini: A Family of Highly Capable Multimodal Models

	Gemini Ultra	Gemini Pro	GPT-4	GPT-3.5	PaLM 2-L	Claude 2	Inflection-2	Grok 1	LLAMA-2
MMLU Multiple-choice questions in 57 subjects (professional & academic) (Hendrycks et al., 2021a)	90.04% CoT@32*	79.13% CoT@8*	87.29% CoT@32 (via API**)	70% 5-shot	78.4% 5-shot	78.5% 5-shot CoT	79.6% 5-shot	73.0% 5-shot	68.0%***
	83.7% 5-shot	71.8% 5-shot	86.4% 5-shot (reported)						
GSM8K Grade-school math (Cobbe et al., 2021)	94.4% Maj1@32	86.5% Maj1@32	92.0% SFT & 5-shot CoT	57.1% 5-shot	80.0% 5-shot	88.0% 0-shot	81.4% 8-shot	62.9% 8-shot	56.8% 5-shot
MATH Math problems across 5 difficulty levels & 7 subdisciplines (Hendrycks et al., 2021b)	53.2% 4-shot	32.6% 4-shot	52.9% 4-shot (via API**)	34.1% 4-shot (via API**)	34.4% 4-shot	—	34.8%	23.9% 4-shot	13.5% 4-shot
			50.3% (Zheng et al., 2023)						
BIG-Bench-Hard Subset of hard BIG-bench tasks written as CoT problems (Srivastava et al., 2022)	83.6% 3-shot	75.0% 3-shot	83.1% 3-shot (via API**)	66.6% 3-shot (via API**)	77.7% 3-shot	—	—	—	51.2% 3-shot
HumanEval Python coding tasks (Chen et al., 2021)	74.4% 0-shot (IT)	67.7% 0-shot (IT)	67.0% 0-shot (reported)	48.1% 0-shot	—	70.0% 0-shot	44.5% 0-shot	63.2% 0-shot	29.9% 0-shot
Natural2Code Python code generation. (New held-out set with no leakage on web)	74.9% 0-shot	69.6% 0-shot	73.9% 0-shot (via API**)	62.3% 0-shot (via API**)	—	—	—	—	—
DROP Reading comprehension & arithmetic. (metric: F1-score) (Dua et al., 2019)	82.4 Variable shots	74.1 Variable shots	80.9 3-shot (reported)	64.1 3-shot	82.0 Variable shots	—	—	—	—
HellaSwag (validation set) Common-sense multiple choice questions (Zellers et al., 2019)	87.8% 10-shot	84.7% 10-shot	95.3% 10-shot (reported)	85.5% 10-shot	86.8% 10-shot	—	89.0% 10-shot	—	80.0%***
WMT23 Machine translation (metric: BLEURT) (Tom et al., 2023)	74.4 1-shot (IT)	71.7 1-shot	73.8 1-shot (via API**)	—	72.7 1-shot	—	—	—	—



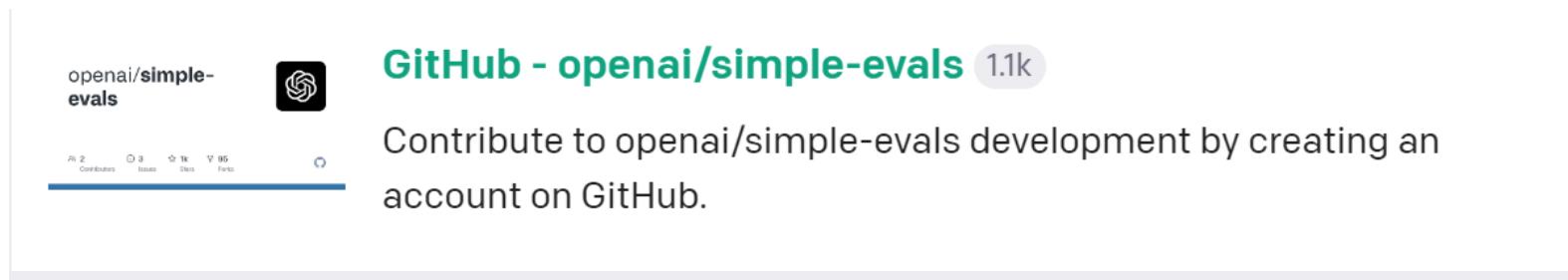
Performance Comparison (2024/4/24)

	Claude 3 Opus	GPT-4	GPT-3.5	Gemini 1.5 Pro	Gemini 1.0 Ultra	Gemini 1.0 Pro
Undergraduate level knowledge <i>MMLU</i>	86.8% 5 shot	86.4% 5 shot	70.0% 5-shot	81.9% 5-shot	83.7% 5-shot	71.8% 5-shot
Graduate level reasoning <i>GPQA, Diamond</i>	50.4% 0-shot CoT	35.7% 0-shot CoT	28.1% 0-shot CoT	—	—	—
Grade school math <i>GSM8K</i>	95.0% 0-shot CoT	92.0% 5-shot CoT	57.1% 5-shot	91.7% 11-shot	94.4% Maj1@32	86.5% Maj1@32
Math problem-solving <i>MATH</i>	60.1% 0-shot CoT	52.9% 4-shot	34.1% 4-shot	58.5% 4-shot	53.2% 4-shot	32.6% 4-shot
Multilingual math <i>MGSM</i>	90.7% 0-shot	74.5% 8-shot	—	88.7% 8-shot	79.0% 8-shot	63.5% 8-shot
Code <i>HumanEval</i>	84.9% 0-shot	67.0% 0-shot	48.1% 0-shot	71.9% 0-shot	74.4% 0-shot	67.7% 0-shot
Reasoning over text <i>DROP, F1 score</i>	83.1 3-shot	80.9 3-shot	64.1 3-shot	78.9 Variable shots	82.4 Variable shots	74.1 Variable shots
Mixed evaluations <i>BIG-Bench-Hard</i>	86.8% 3-shot CoT	83.1% 3-shot CoT	66.6% 3-shot CoT	84.0% 3-shot CoT	83.6% 3-shot CoT	75.0% 3-shot CoT
Knowledge Q&A <i>ARC-Challenge</i>	96.4% 25-shot	96.3% 25-shot	85.2% 25-shot	—	—	—
Common Knowledge <i>HellaSwag</i>	95.4% 10-shot	95.3% 10-shot	85.5% 10-shot	92.5% 10-shot	87.8% 10-shot	84.7% 10-shot

GPT-4o Demonstration



Performance Comparison (GPT-4o)



The screenshot shows a GitHub repository page for "openai/simple-evals". The repository has 1.1k stars. A prominent message encourages users to contribute by creating a GitHub account.

Model	Prompt	MMLU	GPQA	MATH	HumanEval	MGSM	DROP (F1,3-shot)
OPENAI GPT4s							
gpt-4o	chatgpt ^[1]	88.7	53.6	76.6	90.2	90.5	83.4
gpt-4o	assistant ^[2]	87.2	49.9	76.6	91.0	89.9	83.7
gpt-4-turbo-2024-04-09	chatgpt	86.5	49.1	72.2	87.6	88.6	85.4
gpt-4-turbo-2024-04-09	assistant	86.7	49.3	73.4	88.2	89.6	86.0
gpt-4-1106(-vision)-preview	chatgpt	84.6	42.1	64.1	82.2	86.5	81.3
gpt-4-1106(-vision)-preview	assistant	84.7	42.5	64.3	83.7	87.1	83.2
gpt-4-0125-preview	chatgpt	84.8	39.7	64.2	88.2	83.7	83.4
gpt-4-0125-preview	assistant	85.4	41.4	64.5	86.6	85.1	81.5

Performance Comparison (GPT-4o)

r/CustomAI • 2 mo. ago

GPT4o Benchmark

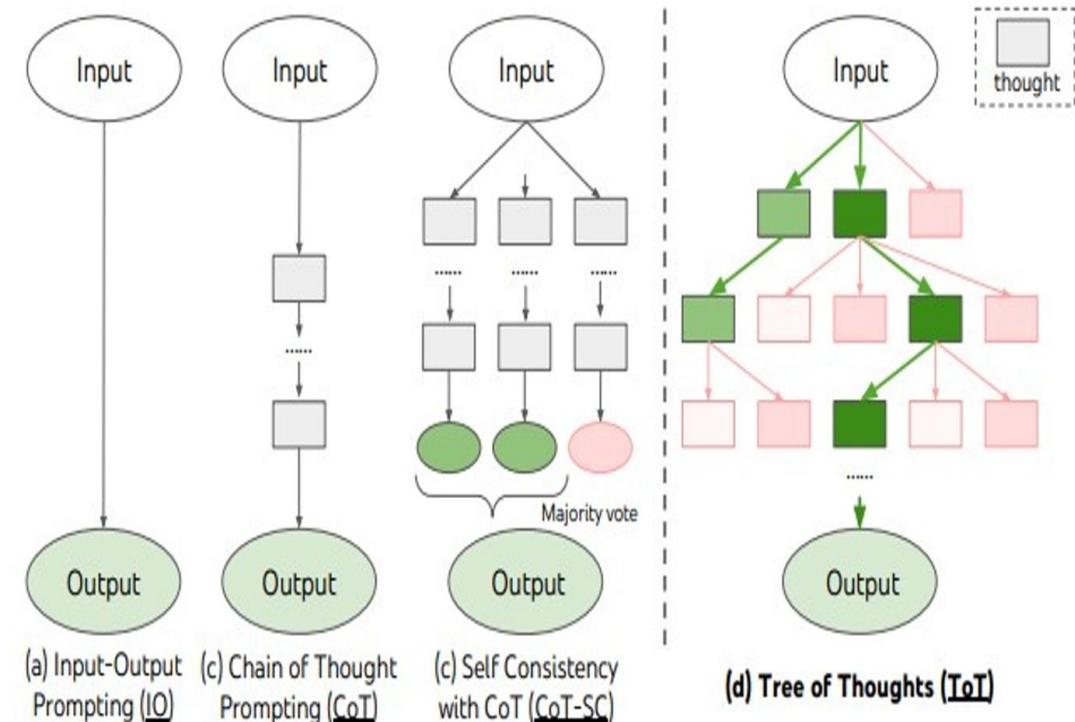
Eval Sets	GPT-4o	GPT-4T 2024-04-09	Gemini 1.0 Ultra	Gemini 1.5 Pro	Claude Opus
MMMU (%) (val)	69.1	63.1	59.4	58.5	59.4
MathVista (%) (testmini)	63.8	58.1	53.0	52.1	50.5
AI2D (%) (test)	94.2	89.4	79.5	80.3	88.1
ChartQA (%) (test)	85.7	78.1	80.8	81.3	80.8
DocVQA (%) (test)	92.8	87.2	90.9	86.5	89.3
ActivityNet (%) (test)	61.9	59.5	52.2	56.7	
EgoSchema (%) (test)	72.2	63.9	61.5	63.2	

Evolution of Interacting with LLMs

- ◆ Zero-shot Prompts / Few-shot Prompts
- ◆ Chain of Thought(CoT) / Self-Confidence CoT / Tree of Thoughts
- ◆ Retrieval Augmented Generation (RAG) Systems
- ◆ Autonomous / AI /LLM Agents
- ◆ Teams of Agents / Agentic(Multi-Agents) Systems

Single Prompt

- IO: Zero/Few-shot
- CoT: Enables complex reasoning capabilities through intermediate Reasoning steps
- CoT-SC: Multiple, diverse reasoning paths through few-shot CoT.
- ToT: Each thought is evaluated and best path found (BFS/DFS/Beam Search)



Few-Shot Prompts(In Context Learning)

Prompt

classify these conversations:

Text: My service was awesome

Label: Positive

Text: My order never arrived

Label: Negative

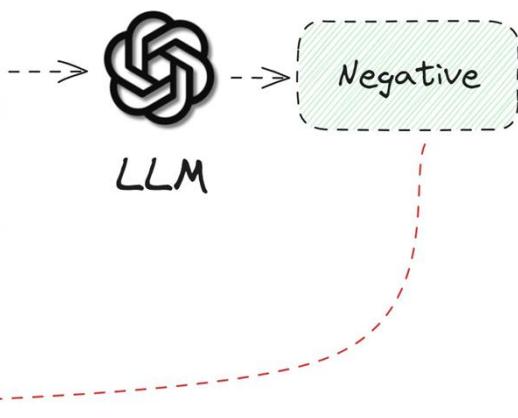
Text: Thanks for the great

service

Label: Positive

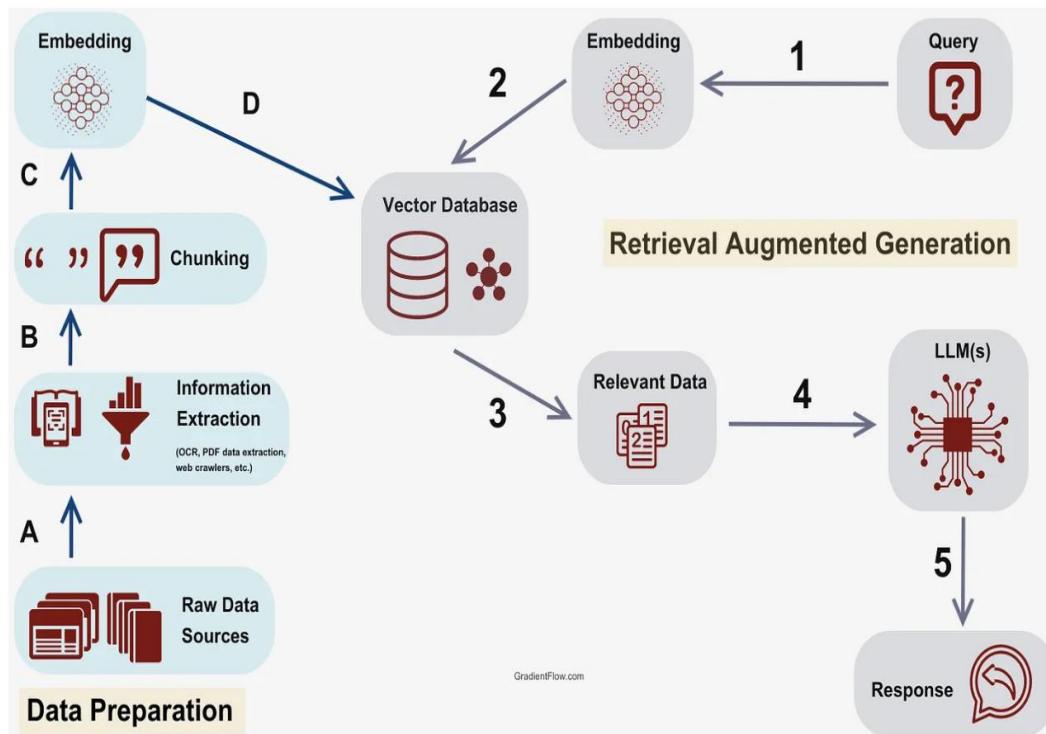
Text: Horrible customer service

Label:



- A technique to enable in-context learning by providing demonstrations in the prompt to steer the model
- The demonstrations serve as conditioning for subsequent examples to generate a response.

Retrieval Augmented Generation (RAG)



- RAG takes an input and retrieves a set of relevant/supporting documents given a source.
- The documents are concatenated as context with the original input prompt and fed to the text generator which produces the final output.
- RAG generates responses that are more actual, specific, and diverse.

Agentic Workflows

Agentic Workflow

LLM-based agents

Non-agentic workflow (zero-shot):

Please type out an essay on topic X from start to finish in one go, without using backspace.



Agentic workflow:

Write an essay outline on topic X

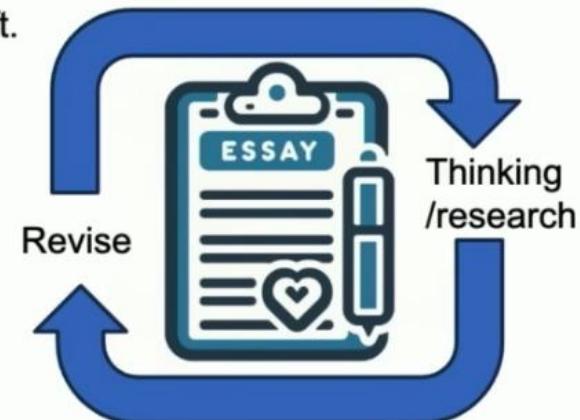
Do you need any web research?

Write a first draft.

Consider what parts need revision or more research.

Revise your draft.

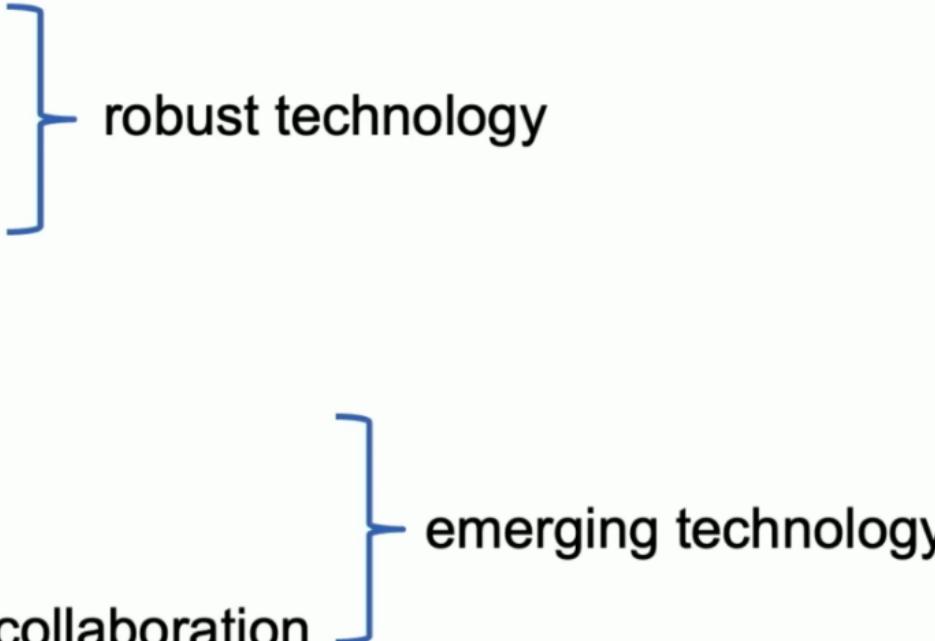
....



- <https://masterdai.blog/exploring-agentic-workflows-a-deep-dive-into-ai-enhanced-productivity/>

Agentic Workflow

Agentic Reasoning Design Patterns

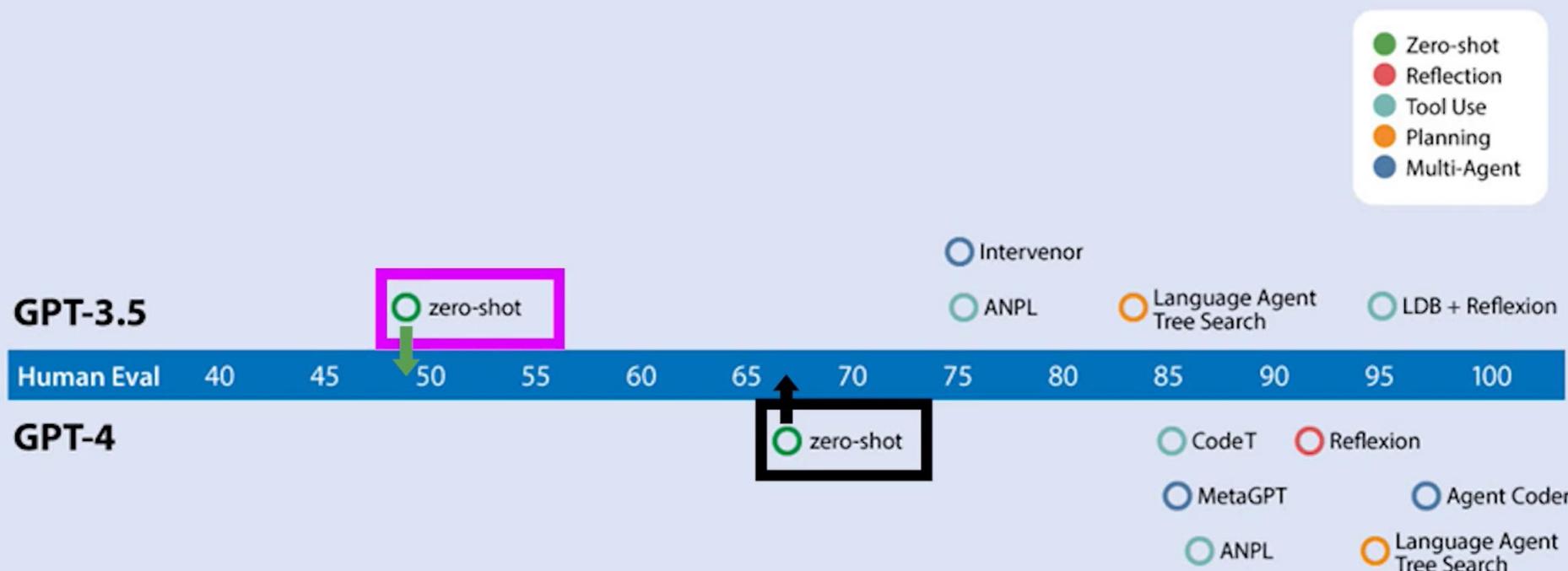
1. Reflection
 2. Tool use
 3. Planning
 4. Multi-agent collaboration
- A diagram illustrating the classification of agentic reasoning design patterns. It shows four numbered items: 1. Reflection, 2. Tool use, 3. Planning, and 4. Multi-agent collaboration. A blue brace on the left side groups items 1 and 2, which are both labeled "robust technology". Another blue brace on the right side groups items 3 and 4, which are both labeled "emerging technology".

Andrew Ng

- <https://masterdai.blog/exploring-agentic-workflows-a-deep-dive-into-ai-enhanced-productivity/>

Performance of Agentic Workflow with LLM

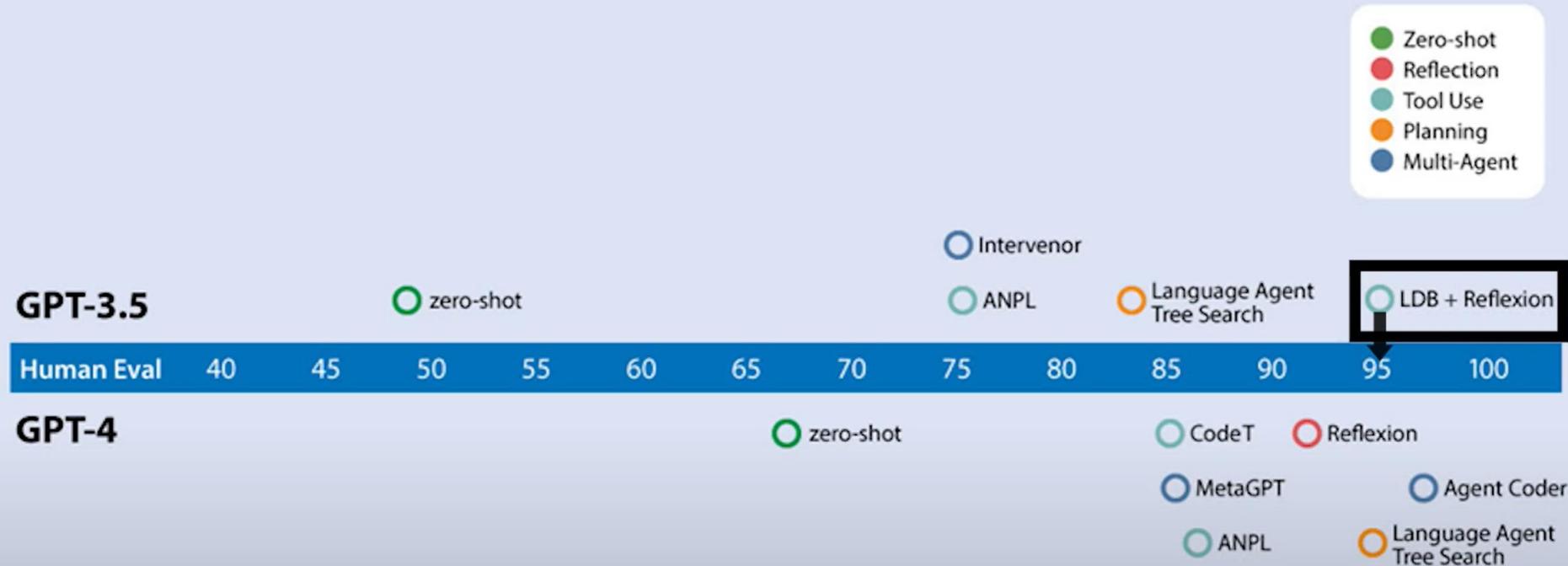
GPT-3.5 and GPT-4 performance using zero-shot and agent workflows



Performance of GPT-3.5 and GPT-4 (zero-shot) on HumanEval, along with algorithms that use agent workflows on top of GPT-3.5 or GPT-4. Thanks to Joaquin Dominguez and John Santerre for help with this analysis.

Performance of Agentic Workflow with LLM

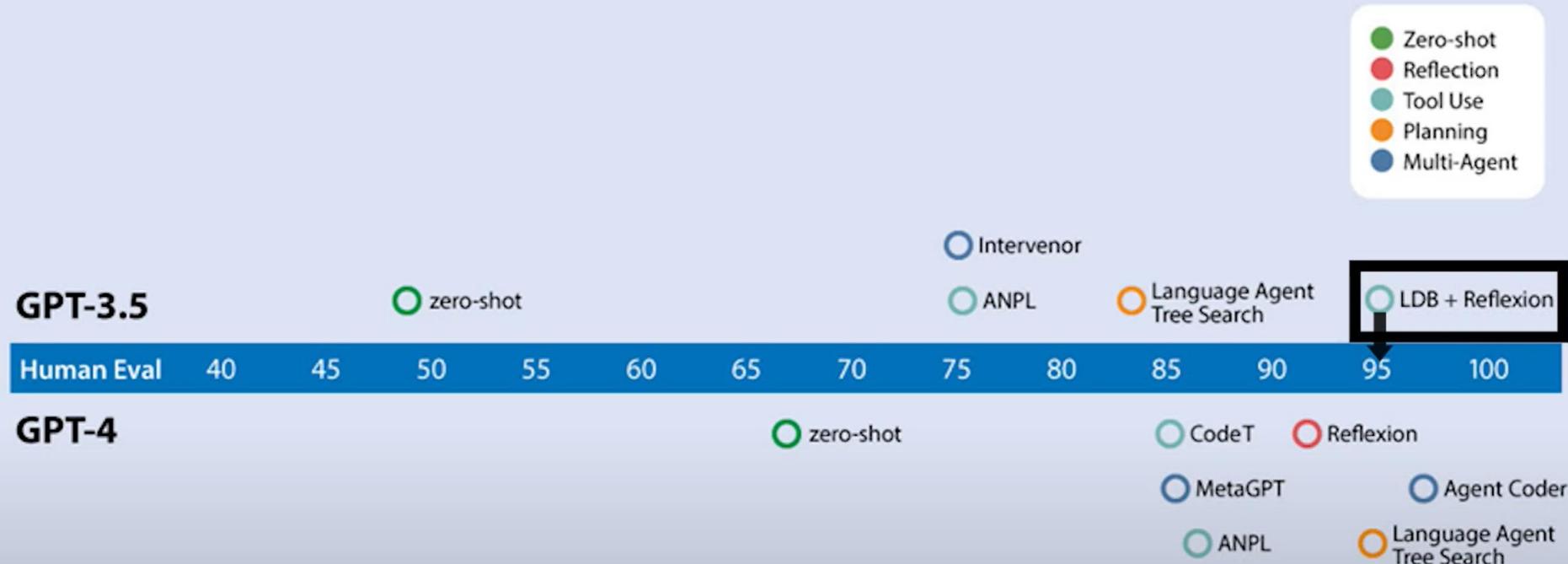
GPT-3.5 and GPT-4 performance using zero-shot and agent workflows



Performance of GPT-3.5 and GPT-4 (zero-shot) on HumanEval, along with algorithms that use agent workflows on top of GPT-3.5 or GPT-4. Thanks to Joaquin Dominguez and John Santerre for help with this analysis.

Performance of Agentic Workflow with LLM

GPT-3.5 and GPT-4 performance using zero-shot and agent workflows



Performance of GPT-3.5 and GPT-4 (zero-shot) on HumanEval, along with algorithms that use agent workflows on top of GPT-3.5 or GPT-4. Thanks to Joaquin Dominguez and John Santerre for help with this analysis.

Agentic Workflow

1. Reflection



Please write code for {task}

Here's code intended for {task}:

```
def do_task (x):  
    ...
```

Check the code carefully for correctness, style and efficiency, and give constructive criticism for how to improve it.

def do_task(x): ...

def do_task_v2(x):

def do_task_v3(x):



Coder Agent
(LLM)

There's a bug on line 5. Fix it by ...

It failed Unit Test 3. Try changing ...



Critic Agent
(LLM)

Recommended reading:

- Self-Refine: Iterative Refinement with Self-Feedback, Madaan et al. (2023)
- Reflexion: Language Agents with Verbal Reinforcement Learning, Shinn et al., (2023)

Andrew Ng

- <https://masterdai.blog/exploring-agentic-workflows-a-deep-dive-into-ai-enhanced-productivity/>

Agentic Workflow

• Reflection

(a) Dialogue: x, y_t

User: I am interested in playing Table tennis.

Response: I'm sure it's a great way to socialize, stay active

(b) FEEDBACK \mathbf{fb}

Engaging: Provides no information about table tennis or how to play it.

User understanding: Lacks understanding of user's needs and state of mind.

(c) REFINE y_{t+1}

Response (refined): That's great to hear (...) ! It's a fun sport requiring quick reflexes and good hand-eye coordination. Have you played before, or are you looking to learn?

(d) Code optimization: x, y_t

```
Generate sum of 1, ..., N
def sum(n):
    res = 0
    for i in range(n+1):
        res += i
    return res
```

(e) FEEDBACK \mathbf{fb}

This code is slow as it uses brute force. A better approach is to use the formula ... $(n(n+1))/2$.

(f) REFINE y_{t+1}

Code (refined)

```
def sum_faster(n):
    return (n*(n+1))//2
```

Figure 2: Examples of SELF-REFINE: an initial output  generated by the base LLM and then passed back to the *same* LLM to receive feedback  to the *same* LLM to refine the output . The top row illustrates this for dialog generation where an initial dialogue response can be transformed into a more engaging one that also understands the user by applying feedback. The bottom row illustrates this for code optimization where the code is made more efficient by applying feedback.

- “Self-Refine: Iterative Refinement with Self-Feedback” by Madaan et al. (2023). <https://arxiv.org/pdf/2303.17651>

Agentic Workflow

• Reflection

Algorithm 1 SELF-REFINE algorithm

Require: input x , model \mathcal{M} , prompts $\{p_{\text{gen}}, p_{\text{fb}}, p_{\text{refine}}\}$, stop condition $\text{stop}(\cdot)$

```
1:  $y_0 = \mathcal{M}(p_{\text{gen}} \| x)$                                      ▷ Initial generation (Eqn. 1)
2: for iteration  $t \in 0, 1, \dots$  do
3:    $fb_t = \mathcal{M}(p_{\text{fb}} \| x \| y_t)$                                ▷ Feedback (Eqn. 2)
4:   if  $\text{stop}(fb_t, t)$  then                                         ▷ Stop condition
5:     break
6:   else
7:      $y_{t+1} = \mathcal{M}(p_{\text{refine}} \| x \| y_0 \| fb_0 \| \dots \| y_t \| fb_t)$  ▷ Refine (Eqn. 4)
8:   end if
9: end for
10: return  $y_t$ 
```

Figure 3: The SELF-REFINE algorithm. See (§2) for a discussion of each component.

- “Self-Refine: Iterative Refinement with Self-Feedback” by Madaan et al. (2023). <https://arxiv.org/pdf/2303.17651>

Agentic Workflow

• Reflection

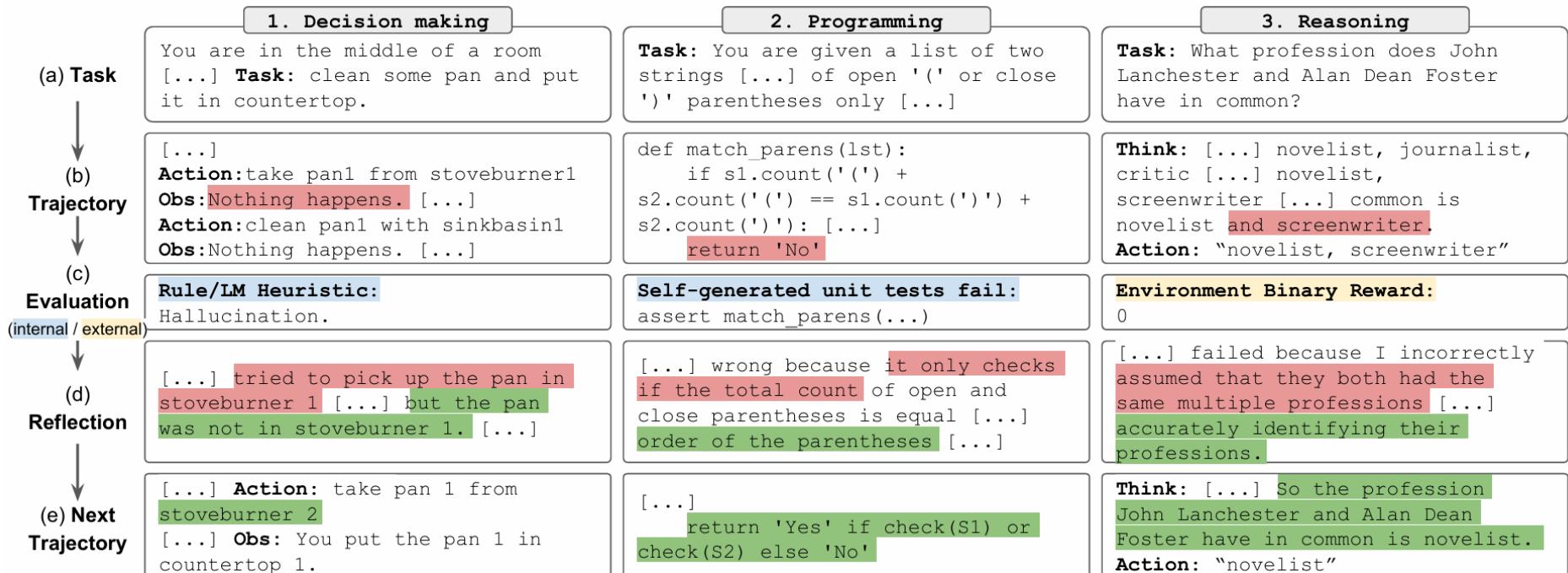
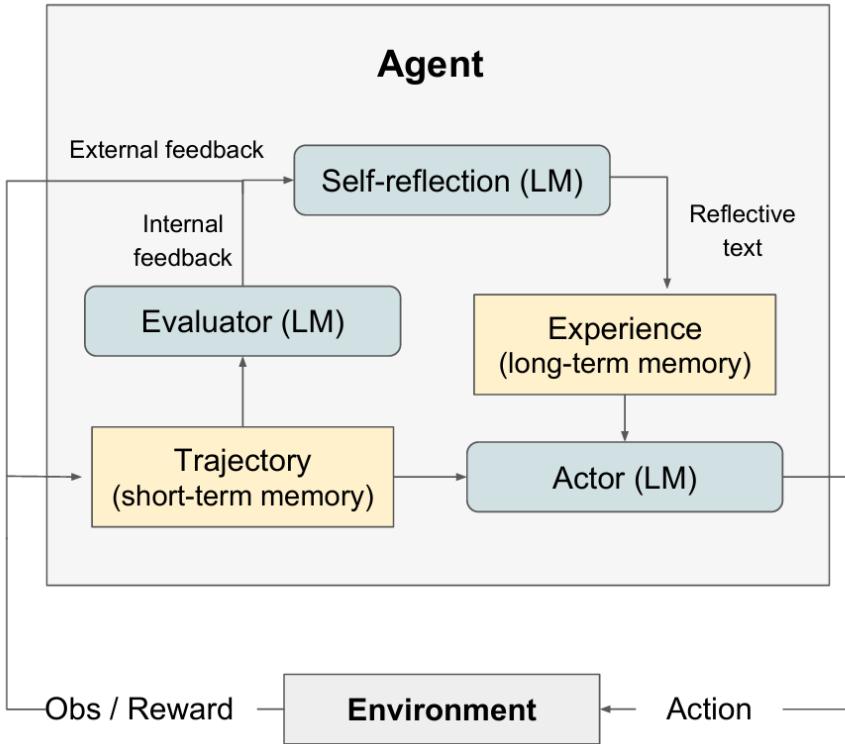


Figure 1: Reflexion works on decision-making [4.1], programming [4.3], and reasoning [4.2] tasks.

- “Reflexion: Language Agents with Verbal Reinforcement Learning” by Shinn et al. (2023). <https://arxiv.org/pdf/2303.11366>

Agentic Workflow

• Reflection



Algorithm 1 Reinforcement via self-reflection

```
Initialize Actor, Evaluator, Self-Reflection:  
     $M_a, M_e, M_{sr}$   
Initialize policy  $\pi_\theta(a_i|s_i), \theta = \{M_a, mem\}$   
Generate initial trajectory using  $\pi_\theta$   
Evaluate  $\tau_0$  using  $M_e$   
Generate initial self-reflection  $sr_0$  using  $M_{sr}$   
Set  $mem \leftarrow [sr_0]$   
Set  $t = 0$   
while  $M_e$  not pass or  $t < \text{max trials}$  do  
    Generate  $\tau_t = [a_0, o_0, \dots, a_i, o_i]$  using  $\pi_\theta$   
    Evaluate  $\tau_t$  using  $M_e$   
    Generate self-reflection  $sr_t$  using  $M_{sr}$   
    Append  $sr_t$  to  $mem$   
    Increment  $t$   
end while  
return
```

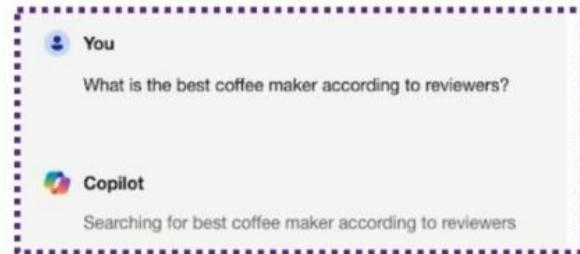
Figure 2: (a) Diagram of Reflexion. (b) Reflexion reinforcement algorithm

- “Reflexion: Language Agents with Verbal Reinforcement Learning” by Shinn et al. (2023). <https://arxiv.org/pdf/2303.11366>

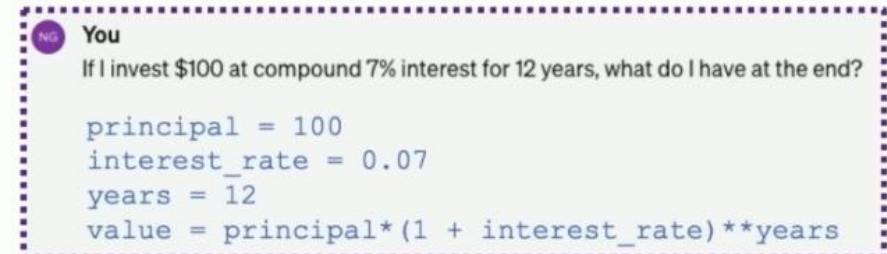
Agentic Workflow

• Tool Use

Web search tool



Code execution tool



Analysis

- Code Execution
- Wolfram Alpha
- Bearly Code Interpreter

Research

- Search engine
- Web browsing
- Wikipedia

Productivity

- Email
- Calendar
- Cloud Storage

Images

- Image generation (e.g., Dall-E)
- Image captioning
- Object detection

Recommended reading:

- Gorilla: Large Language Model Connected with Massive APIs, Patil et al. (2023)
- MM-REACT: Prompting ChatGPT for Multimodal Reasoning and Action, Yang et al. (2023)

Andrew Ng

- <https://masterdai.blog/exploring-agentic-workflows-a-deep-dive-into-ai-enhanced-productivity/>

Agentic Workflow

- Tool Use

GPT-4

```
<domain>: Speech-to-Text
<api_provider>: TorchHub
<code>:
asr_model =
    torch.hub.load(
        'snakers4/silero-models',
        'asr',
        source='local')
result =
    asr_model.transcribe(
        audio_path)
```

Claude

```
<domain>: Audio-Translation
<api_provider>: Pytorch
<code>:
import torchaudio
translation =
    Torchaudio.pipelines.
    WAV2VEC2_ASR_PIPELINE(
        "audio.wav")
```

Gorilla

```
<domain>: Speech-to-Text
<api_provider>: TorchHub
<code>:
asr_model =
    torch.hub.load(
        'snakers4/silero-models',
        'silero_sst')
result =
    asr_model.transcribe(
        audio_path)
```



Hallucinate!



Wrong library!



Good to go!

Prompt: Help me find an API to convert the spoken language in a recorded audio to text using Torch Hub.

Figure 1: **Examples of API calls.** Example API calls generated by GPT-4 [29], Claude [3], and Gorilla for the given prompt. In this example, GPT-4 presents a model that doesn't exist, and Claude picks an incorrect library. In contrast, our Gorilla model can identify the task correctly and suggest a fully-qualified API call.

- “Gorilla: Large Language Model Connected with Massive APIs” by Patil et al. (2023). <https://arxiv.org/pdf/2305.15334>

Agentic Workflow

- Tool Use

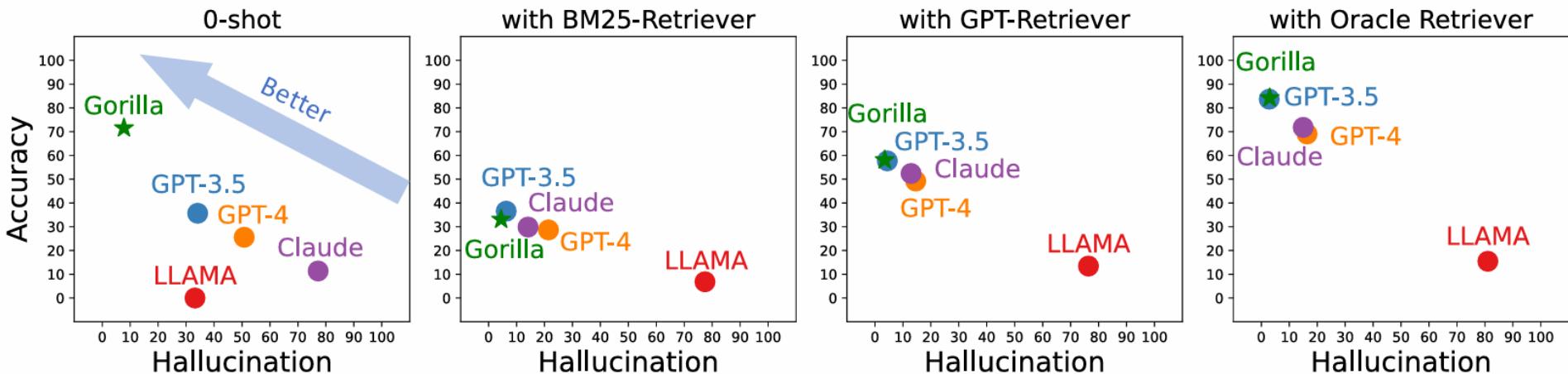


Figure 2: **Accuracy (vs) hallucination** in four settings, that is, *zero-shot* (i.e., without any retriever), and *with retrievers*. BM25 and GPT are commonly used retrievers and the oracle retriever returns relevant documents at 100%, indicating an upper bound. Higher in the graph (higher accuracy) and to the left is better (lower hallucination). Across the entire dataset, our model, Gorilla, improves accuracy while reducing hallucination.

- “Gorilla: Large Language Model Connected with Massive APIs” by Patil et al. (2023). <https://arxiv.org/pdf/2305.15334>

Agentic Workflow

• Tool Use

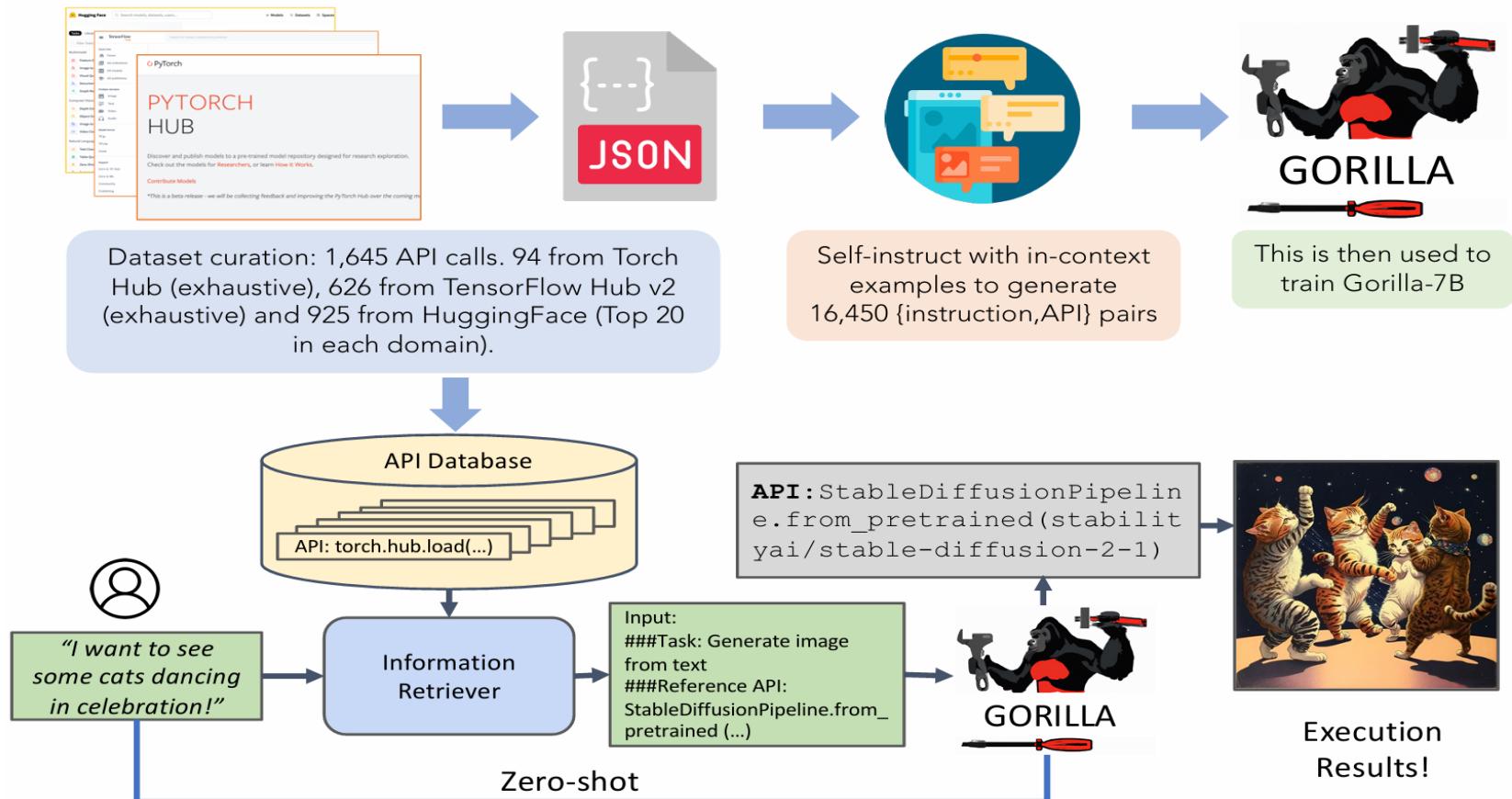


Figure 3: **Gorilla: A system for enabling LLMs to interact with APIs.** The upper half represents the training procedure as described in Sec 3. This is the most exhaustive API data-set for ML to the best of our knowledge. During inference (lower half), Gorilla supports two modes - with retrieval, and zero-shot. In this example, it is able to suggest the right API call for generating the image from the user's natural language query.

Agentic Workflow

• Tool Use

Visual Math and Text Reasoning

What is “a” in the first equation?
What is y in the last equation?

$5 + a = 7$	$40 + y = 45$
$10 + y = 15$	$39 + a = 41$
$20 + a = 22$	$24 + y = 29$
$48 + a = 50$	$15 + a = 17$
$10 + a = 12$	$34 + y = 39$

The value of “a” in the first equation “ $5 + a = 7$ ” is 2.

The value of “y” in the last equation “ $34 + y = 39$ ” is 5.

Visual-Conditioned Joke/Meme

What is funny about it?



The humor in the comic may come from the situation, the characters, or the dialogue. The person is feeling unwell and is asked if they are pregnant or have the coronavirus, which is a current and relevant topic. The humor may come from the absurdity of the situation or the unexpectedness of the question.

Spatial/Coordinate Understanding

Where is the frisbee?
What is the relative position of the frisbee and the man?



The frisbee is located at coordinates (63,119,133,177).

The frisbee is located above the man in the image. The man is falling on the ground while the frisbee is in the air. The frisbee is also closer to the top of the image than the man..

Visual Planning and Prediction

What should I do after kneading the dough, before dividing it into pieces? For how long?



After kneading the dough, you should put it in a warm place before dividing it into pieces, according to the recipe in the image.

You should put the dough in a warm place for 1 hour.

Multi-Image Reasoning

How much in total did I spend on travel?
How much in total did I pay for taxes?



You spent \$1299.90 on airfare and \$43.83 on an Uber ride for a total of \$1343.73 on travel.

You paid \$323.23 in total for taxes. This includes \$321.90 for taxes and fees for the flight purchase and \$1.33 for sales tax for the Windmill Family Restaurant.

Multi-Hop Document Understanding

Excluding the regular cab, what is the average total fueling cost?



The average total fueling cost excluding the Ford F150 (Regular Cab) is \$76.55. This is calculated by taking the sum of the total fueling cost of Toyota RAV4 and Honda Civic and dividing it by 2. (\$82.56 + \$70.55) / 2 = \$76.55.

Open-World Concept Understanding

What is the brand of these drink? Which one is unsweetened?



The brand of these drinks is “BETTER THAN MILK”. The unsweetened one is “BETTER THAN MILK organic OAT DRINK”.

Who are the celebrities in this image?



Keira Knightley and Natalie Portman are the celebrities in this image.

Is it safe to eat this mushroom?



Morel mushrooms are generally considered safe to eat when properly cooked.

Video Summarization/Event Localization

Can you breakdown this video tutorial to step-by-step instructions, with start and end time?



Here's a step-by-step breakdown of the tutorial with start and end time.



... Add Lettuce Add Tomato Add Bacon ...

- “MM-REACT: Prompting ChatGPT for Multimodal Reasoning and Action” by Yang et al. (2023). <https://arxiv.org/pdf/2303.11381>

Agentic Workflow

• Tool Use

- image caption
- image tagging
- dense caption
- celebrity
- receipt
- OCR
- Bing search
- :
- vision experts

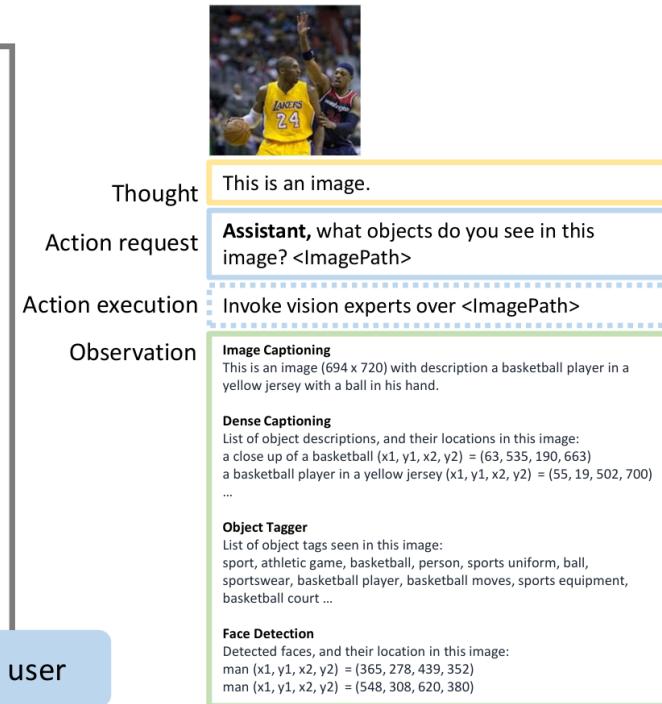
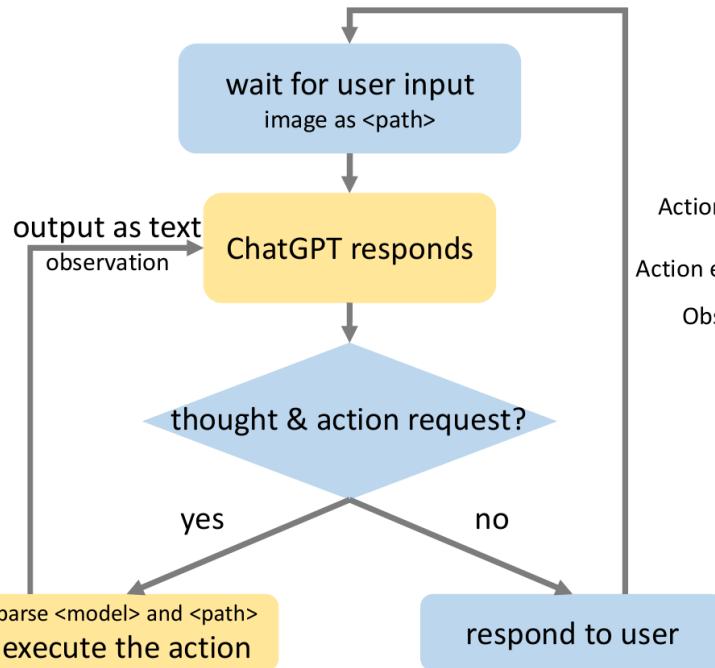
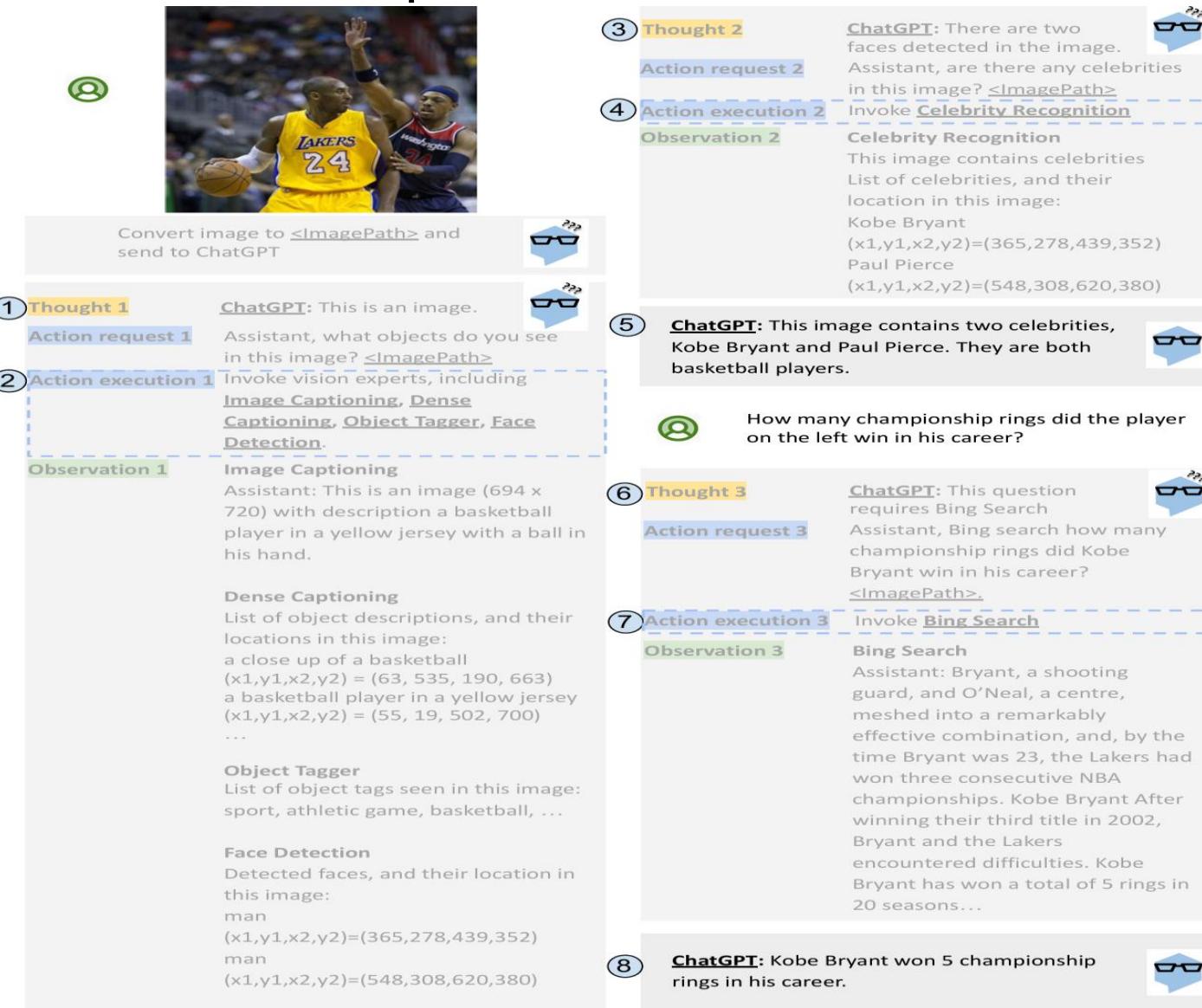


Figure 2. Flowchart of MM-REACT for enhanced visual understanding with ChatGPT. The user input can be in the form of text, images, or videos, with the latter two represented as file path strings. ChatGPT is instructed to say specific watchwords in *action request* if a vision expert is required to interpret the visual inputs. Regular expression matching is applied to parse the expert's name and the file path, which are then used to call the vision expert (*action execution*). The expert's output (*observation*) is serialized as text and combined with the history to further activate ChatGPT. If no extra experts are needed, MM-REACT would return the final response to the user. The right figure shows a single-round vision expert execution, which is the component that constructs the full execution flow illustrated in Figure 3.

- “MM-REACT: Prompting ChatGPT for Multimodal Reasoning and Action” by Yang et al. (2023). <https://arxiv.org/pdf/2303.11381>

Agentic Workflow

• Tool Use – An example of MM-REACT's full execution flow



Agentic Workflow

• Tool Use – Case study of MM-REACT's capabilities



$5 + a = 7$	$40 + y = 45$
$10 + y = 15$	$39 + a = 41$
$20 + a = 22$	$24 + y = 29$
$48 + a = 50$	$15 + a = 17$
$10 + a = 12$	$34 + y = 39$

This image contains 10 math equations.



What is "a" in the first equation?

The value of "a" in the first equation " $5 + a = 7$ " is 2.



What is y in the last equation?

The value of "y" in the last equation " $34 + y = 39$ " is 5.



What is the answer to the third equation on the left?

The answer to the third equation on the left " $20 + a = 22$ " is 2.



What is the answer to the second last equation on the right?

The answer to the second last equation on the right " $15 + a = 17$ " is 2.



This image is likely a classroom with children writing on a chalkboard. The texts in this image are mathematical equations.



What is the answer to the first equation from the left hand side?

The answer to the first equation " $3 \times 3 =$ " is 9.



What is the answer to the first equation from right?

The answer to the first equation from right " $11 - 2 =$ " is 9.



What is the answer to the equation in the middle?

The answer to the equation in the middle " $7 \times 2 =$ " is 14.



$5 + a = 7$	$40 + y = 45$
$10 + y = 15$	$39 + a = 41$
$20 + a = 22$	$24 + y = 29$
$48 + a = 50$	$15 + a = 17$
$10 + a = 12$	$34 + y = 39$

What is the answer to the second last equation on the right?

What is "a" in the first equation on the right?

To solve for a :

$$15 + a = 17$$

Subtract 15 from both sides:

$$a = 17 - 15$$

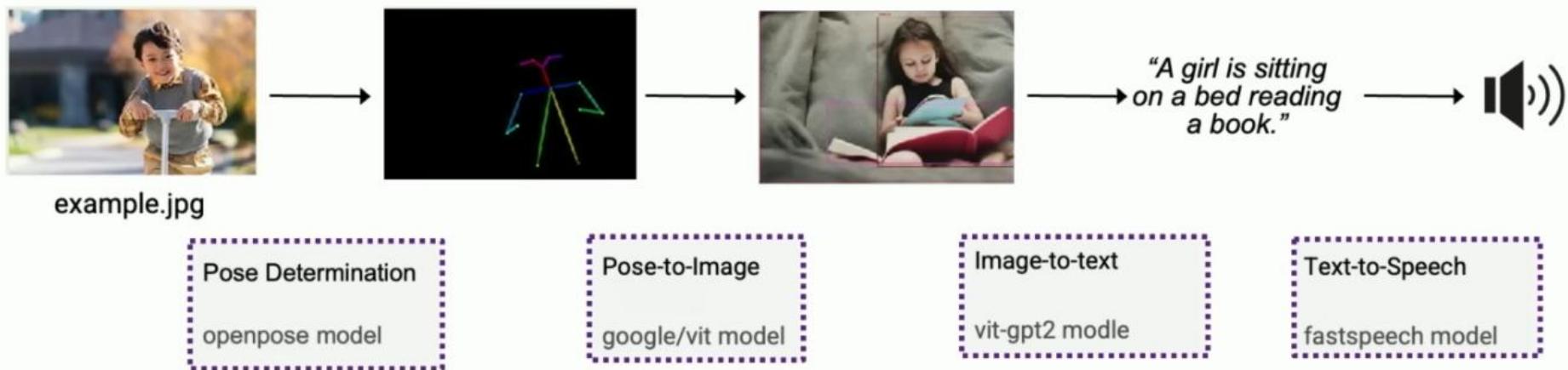
$$a = 2$$

So, the value of a in the equation $15 + a = 17$ is 2.

Agentic Workflow

• Planning

Request: Please generate an image where a girl is reading a book, and her pose is the same as the boy in the image example.jpg, then please describe the new image with your voice.



[Example adapted from HuggingGPT paper]

Recommended reading:

- Chain-of-Thought Prompting Elicits Reasoning in Large Language Models, Wei et al., (2022)
- HuggingGPT: Solving AI Tasks with ChatGPT and its Friends in Hugging Face, Shen et al. (2023)

Andrew Ng

- <https://masterdai.blog/exploring-agentic-workflows-a-deep-dive-into-ai-enhanced-productivity/>

Agentic Workflow

• Planning

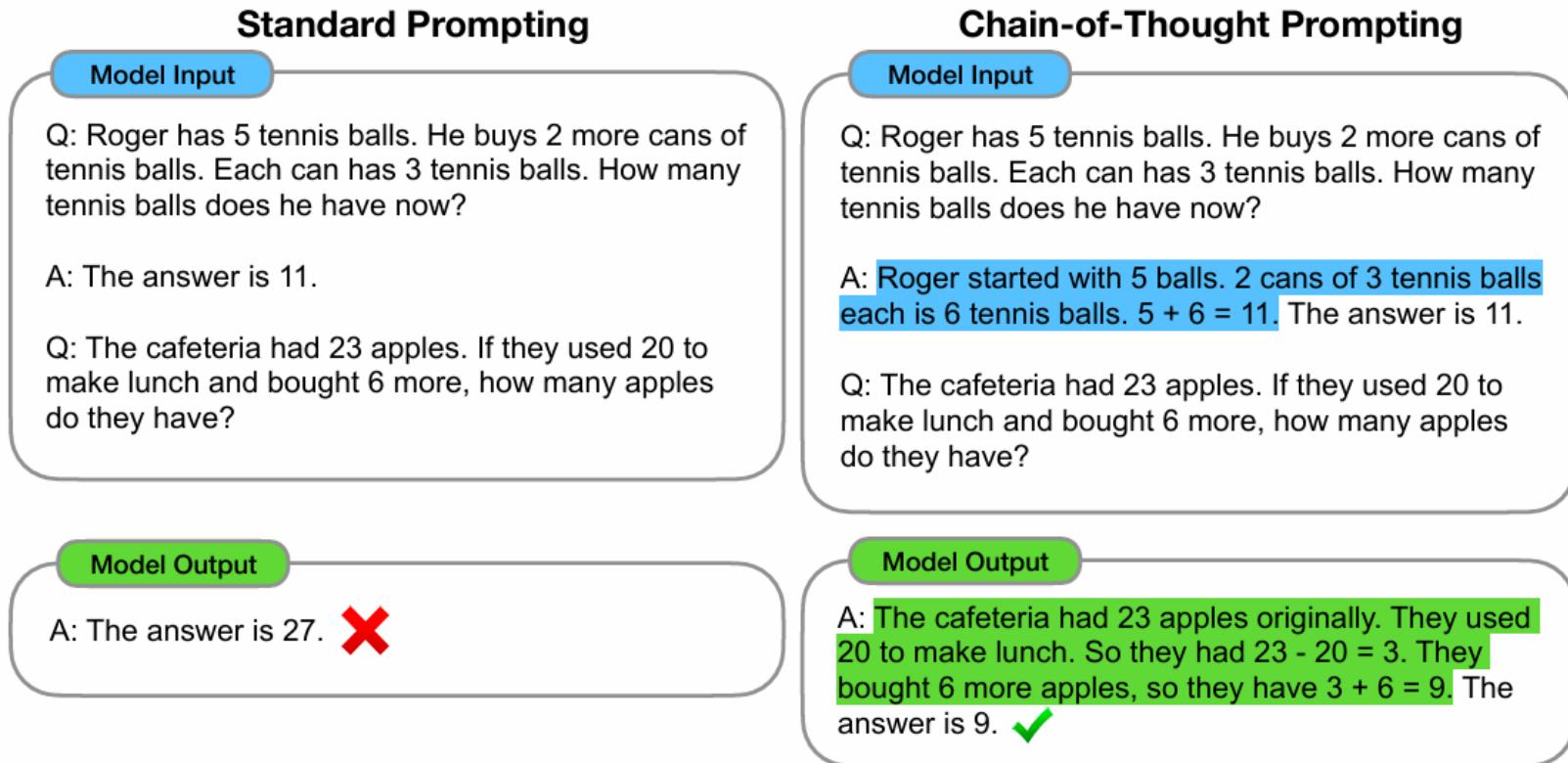


Figure 1: Chain-of-thought prompting enables large language models to tackle complex arithmetic, commonsense, and symbolic reasoning tasks. Chain-of-thought reasoning processes are highlighted.

- “Chain-of-Thought Prompting Elicits Reasoning in Large Language Models” by Wei et al. (2022). <https://arxiv.org/pdf/2201.11903>

Agentic Workflow

• Planning

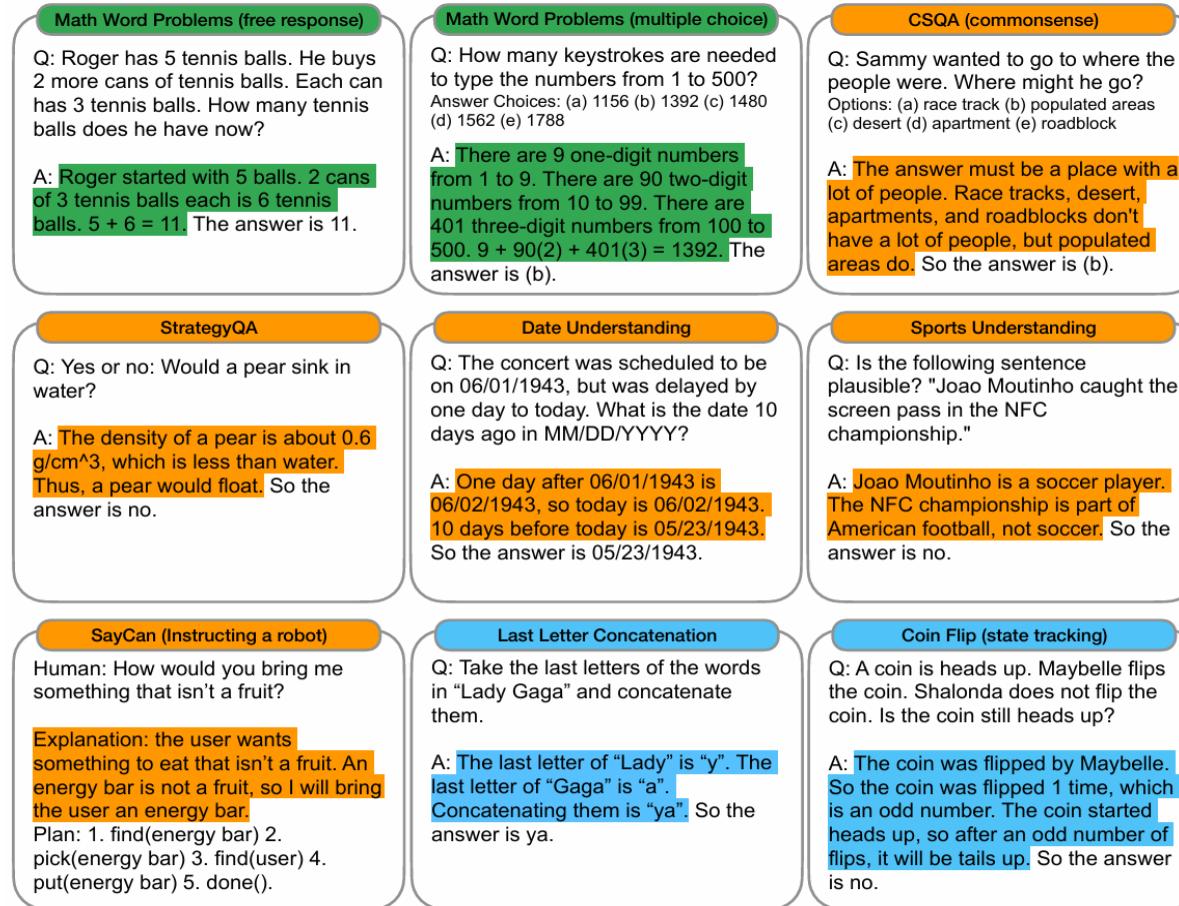


Figure 3: Examples of \langle input, chain of thought, output \rangle triples for arithmetic, commonsense, and symbolic reasoning benchmarks. Chains of thought are highlighted. Full prompts in Appendix G.

- Chain-of-Thought Prompting Elicits Reasoning in Large Language Models" by Wei et al. (2022). <https://arxiv.org/pdf/2201.11903>

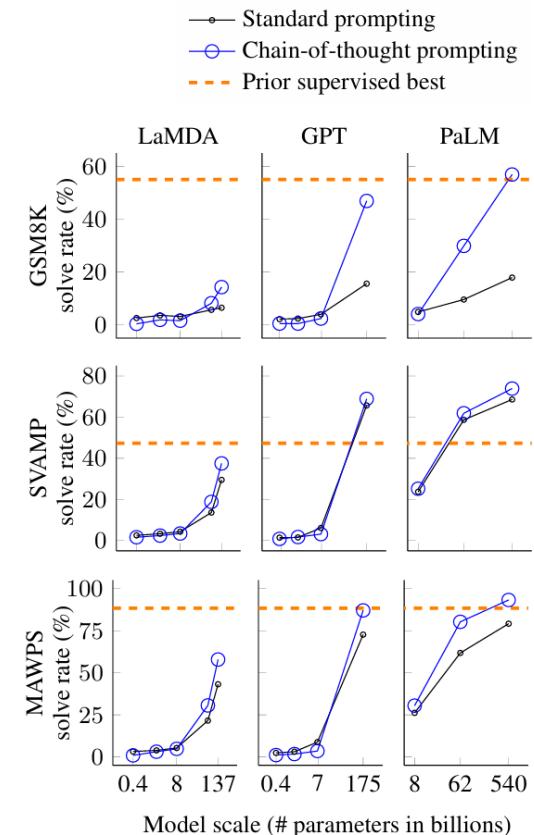


Figure 4: Chain-of-thought prompting enables large language models to solve challenging math problems. Notably, chain-of-thought reasoning is an emergent ability of increasing model scale. Prior best numbers are from Cobbe et al. (2021) for GSM8K, Jie et al. (2022) for SVAMP, and Lan et al. (2021) for MAWPS.

Agentic Workflow

• Planning

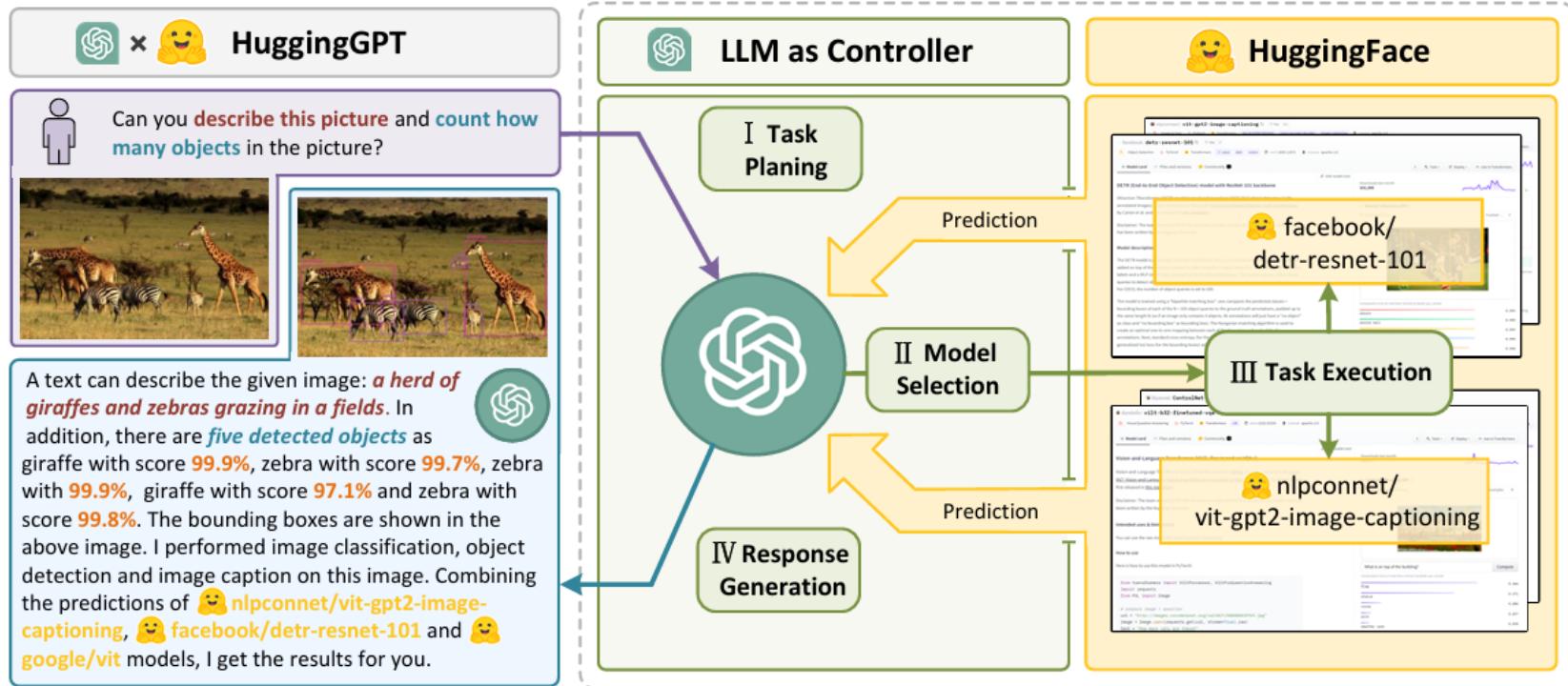


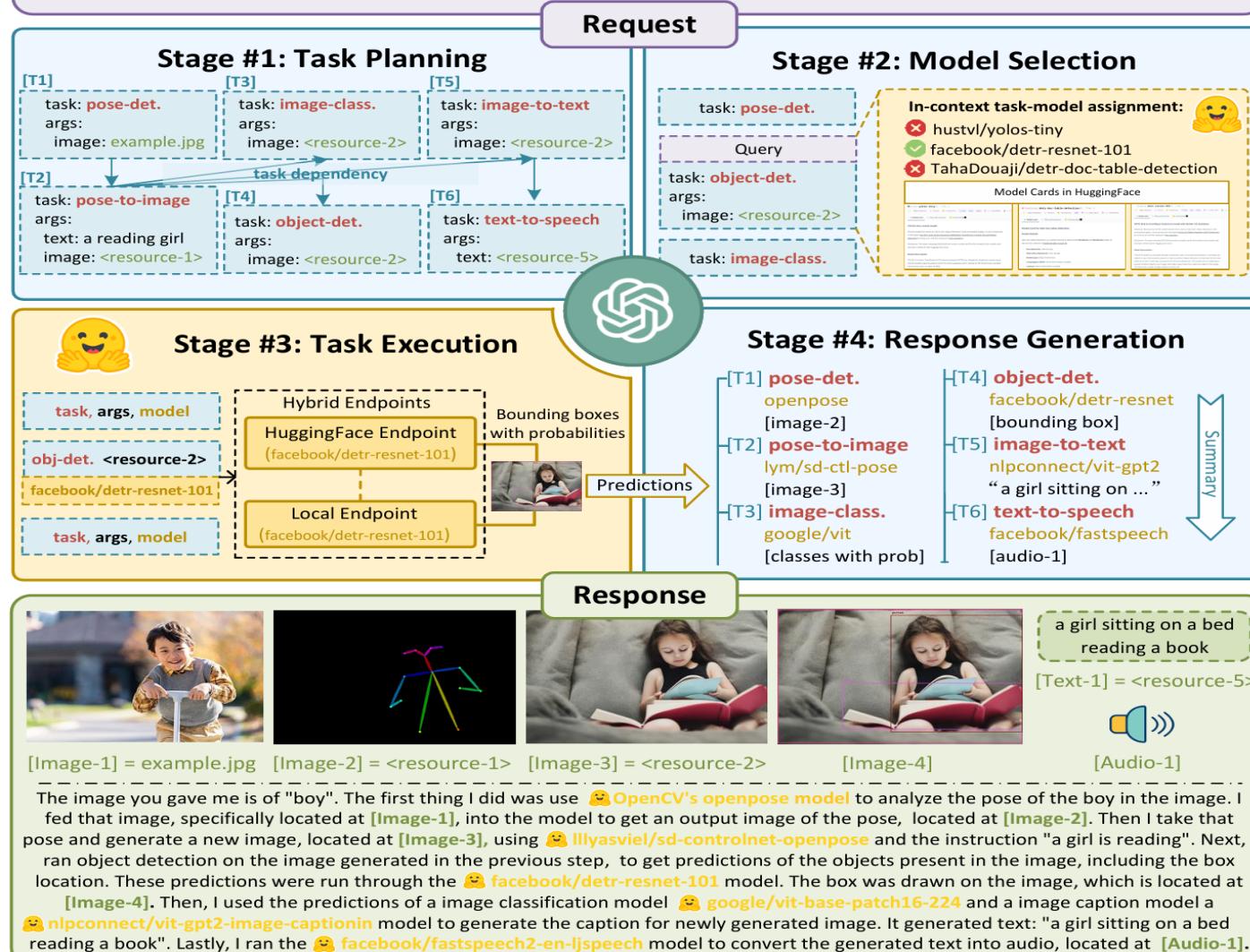
Figure 1: Language serves as an interface for LLMs (e.g., ChatGPT) to connect numerous AI models (e.g., those in Hugging Face) for solving complicated AI tasks. In this concept, an LLM acts as a controller, managing and organizing the cooperation of expert models. The LLM first plans a list of tasks based on the user request and then assigns expert models to each task. After the experts execute the tasks, the LLM collects the results and responds to the user.

- HuggingGPT: Solving AI Tasks with ChatGPT and its Friends in Hugging Face" by Shen et al. (2023). <https://arxiv.org/pdf/2303.17580>

Agentic Workflow

• Overview of HuggingGPT

Please generate an image where a girl is reading a book, and her pose is the same as the boy in the image example.jpg, then please describe the new image with your voice.



Agentic Workflow

- Details of the prompt design in HuggingGPT

Task Planning	<p>Prompt</p> <pre>#1 Task Planning Stage - The AI assistant performs task parsing on user input, generating a list of tasks with the following format: [{"task": task, "id": task_id, "dep": dependency_task_ids, "args": {"text": text, "image": URL, "audio": URL, "video": URL}}]. The "dep" field denotes the id of the previous task which generates a new resource upon which the current task relies. The tag "<resource>-task_id" represents the generated text, image, audio, or video from the dependency task with the corresponding task_id. The task must be selected from the following options: {{ Available Task List }}. Please note that there exists a logical connections and order between the tasks. In case the user input cannot be parsed, an empty JSON response should be provided. Here are several cases for your reference: {{ Demonstrations }}. To assist with task planning, the chat history is available as {{ Chat Logs }}, where you can trace the user-mentioned resources and incorporate them into the task planning stage.</pre> <p>Demonstrations</p> <table border="0"><tr><td data-bbox="368 593 706 659">Can you tell me how many objects in e1.jpg?</td><td data-bbox="706 593 1532 659">[{"task": "object-detection", "id": 0, "dep": [-1], "args": {"image": "e1.jpg"}]}</td></tr><tr><td data-bbox="368 696 706 748">In e2.jpg, what's the animal and what's it doing?</td><td data-bbox="706 659 1532 803">[{"task": "image-to-text", "id": 0, "dep": [-1], "args": {"image": "e2.jpg"}}, {"task": "image-cls", "id": 1, "dep": [-1], "args": {"image": "e2.jpg"}}, {"task": "object-detection", "id": 2, "dep": [-1], "args": {"image": "e2.jpg"}}, {"task": "visual-question-answering", "id": 3, "dep": [-1], "args": {"text": "what's the animal doing?", "image": "e2.jpg"}}]</td></tr><tr><td data-bbox="368 803 706 912">First generate a HED image of e3.jpg, then based on the HED image and a text "a girl reading a book", create a new image as a response.</td><td data-bbox="706 803 1532 912">[{"task": "pose-detection", "id": 0, "dep": [-1], "args": {"image": "e3.jpg"}}, {"task": "pose-text-to-image", "id": 1, "dep": [0], "args": {"text": "a girl reading a book", "image": "<resource>-0"}}]</td></tr></table>	Can you tell me how many objects in e1.jpg?	[{"task": "object-detection", "id": 0, "dep": [-1], "args": {"image": "e1.jpg"}]}	In e2.jpg, what's the animal and what's it doing?	[{"task": "image-to-text", "id": 0, "dep": [-1], "args": {"image": "e2.jpg"}}, {"task": "image-cls", "id": 1, "dep": [-1], "args": {"image": "e2.jpg"}}, {"task": "object-detection", "id": 2, "dep": [-1], "args": {"image": "e2.jpg"}}, {"task": "visual-question-answering", "id": 3, "dep": [-1], "args": {"text": "what's the animal doing?", "image": "e2.jpg"}}]	First generate a HED image of e3.jpg, then based on the HED image and a text "a girl reading a book", create a new image as a response.	[{"task": "pose-detection", "id": 0, "dep": [-1], "args": {"image": "e3.jpg"}}, {"task": "pose-text-to-image", "id": 1, "dep": [0], "args": {"text": "a girl reading a book", "image": "<resource>-0"}}]
Can you tell me how many objects in e1.jpg?	[{"task": "object-detection", "id": 0, "dep": [-1], "args": {"image": "e1.jpg"}]}						
In e2.jpg, what's the animal and what's it doing?	[{"task": "image-to-text", "id": 0, "dep": [-1], "args": {"image": "e2.jpg"}}, {"task": "image-cls", "id": 1, "dep": [-1], "args": {"image": "e2.jpg"}}, {"task": "object-detection", "id": 2, "dep": [-1], "args": {"image": "e2.jpg"}}, {"task": "visual-question-answering", "id": 3, "dep": [-1], "args": {"text": "what's the animal doing?", "image": "e2.jpg"}}]						
First generate a HED image of e3.jpg, then based on the HED image and a text "a girl reading a book", create a new image as a response.	[{"task": "pose-detection", "id": 0, "dep": [-1], "args": {"image": "e3.jpg"}}, {"task": "pose-text-to-image", "id": 1, "dep": [0], "args": {"text": "a girl reading a book", "image": "<resource>-0"}}]						
Model Selection	<p>Prompt</p> <pre>#2 Model Selection Stage - Given the user request and the call command, the AI assistant helps the user to select a suitable model from a list of models to process the user request. The AI assistant merely outputs the model id of the most appropriate model. The output must be in a strict JSON format: {"id": "id", "reason": "your detail reason for the choice"}. We have a list of models for you to choose from {{ Candidate Models }}. Please select one model from the list.</pre> <p>Candidate Models</p> <pre>{"model_id": model id #1, "metadata": meta-info #1, "description": description of model #1} {"model_id": model id #2, "metadata": meta-info #2, "description": description of model #2} ... {"model_id": model id #K, "metadata": meta-info #K, "description": description of model #K}</pre>						
Response Generation	<p>Prompt</p> <pre>#4 Response Generation Stage - With the input and the inference results, the AI assistant needs to describe the process and results. The previous stages can be formed as - User Input: {{ User Input }}, Task Planning: {{ Tasks }}, Model Selection: {{ Model Assignment }}, Task Execution: {{ Predictions }}. You must first answer the user's request in a straightforward manner. Then describe the task process and show your analysis and model inference results to the user in the first person. If inference results contain a file path, must tell the user the complete file path. If there is nothing in the results, please tell me you can't make it.</pre>						

Agentic Workflow

• Planning

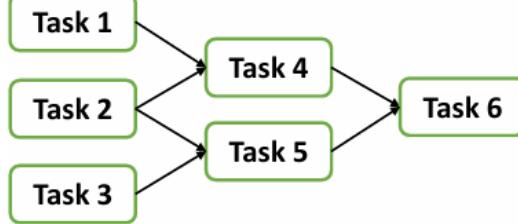
Task Type	Diagram	Example	Metrics
Single Task		Show me a funny image of a cat	Precision, Recall, F1, Accuracy
Sequential Task		Replace the cat with a dog in example.jpg	Precision, Recall, F1 Edit Distance
Graph Task		Given a collection of image A: a.jpg, B: b.jpg, C: c.jpg, please tell me which image is more like image B in terms of semantic, A or C?	Precision, Recall, F1 GPT-4 Score

Table 2: Evaluation for task planning in different task types.

- HuggingGPT: Solving AI Tasks with ChatGPT and its Friends in Hugging Face” by Shen et al. (2023). <https://arxiv.org/pdf/2303.17580>

Agentic Workflow

- Multiagent Collaboration



Multiagent Debate

Task	Single agent	Multi-agent
Biographies	66.0%	73.8%
MMLU	63.9%	71.1%
Chess move	29.3%	45.2%

(Du et al., 2023)

Recommended reading:

- Communicative Agents for Software Development, Qian et al., (2023)
- AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation, Wu et al. (2023)

- <https://masterdai.blog/exploring-agentic-workflows-a-deep-dive-into-ai-enhanced-productivity/>

Andrew Ng

Agentic Workflow

- Multiagent Collaboration - ChatDev



Figure 1: ChatDev, a chat-powered software development framework, integrates LLM agents with various social roles, working autonomously to develop comprehensive solutions via multi-agent collaboration.

- “Communicative Agents for Software Development” by Qian et al. (2023).
<https://arxiv.org/pdf/2307.07924>

Agentic Workflow

- Multiagent Collaboration - ChatDev

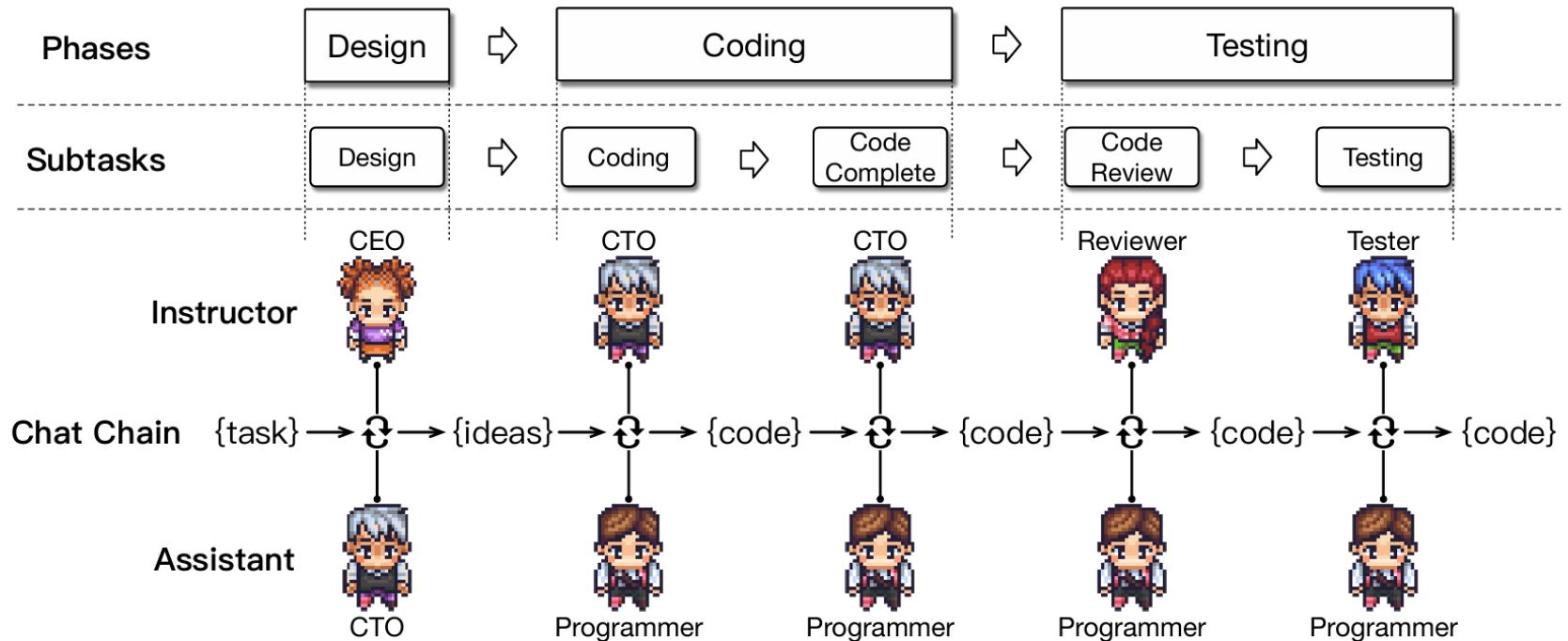


Figure 2: Upon receiving a preliminary task requirement (e.g., “*develop a Gomoku game*”), these software agents engage in multi-turn communication and perform instruction-following along a chain-structured workflow, collaborating to execute a series of subtasks autonomously to craft a comprehensive solution.

- Communicative Agents for Software Development” by Qian et al. (2023).
<https://arxiv.org/pdf/2307.07924>

Agentic Workflow

• Multiagent Collaboration - ChatDev

Method	Paradigm	Completeness	Executability	Consistency	Quality
GPT-Engineer	😊	<u>0.5022</u> [†]	0.3583 [†]	<u>0.7887</u> [†]	0.1419 [†]
MetaGPT	😊😊	0.4834 [†]	<u>0.4145</u> [†]	0.7601 [†]	<u>0.1523</u> [†]
ChatDev	😊😊	0.5600	0.8800	0.8021	0.3953

Table 1: Overall performance of the LLM-powered software development methods, encompassing both single-agent (😊) and multi-agent (😊😊) paradigms. Performance metrics are averaged for all tasks. The top scores are in bold, with second-highest underlined. † indicates significant statistical differences ($p \leq 0.05$) between a baseline and ours.

Method	Evaluator	Baseline Wins	ChatDev Wins	Draw
GPT-Engineer	GPT-4	22.50%	77.08%	00.42%
	Human	09.18%	90.16%	00.66%
MetaGPT	GPT-4	37.50%	57.08%	05.42%
	Human	07.92%	88.00%	04.08%

Table 2: Pairwise evaluation results.

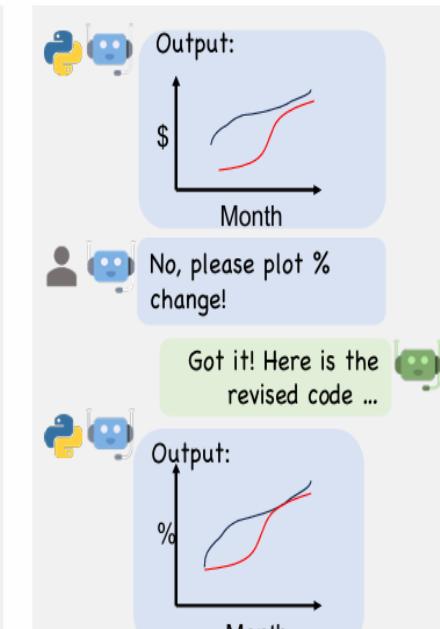
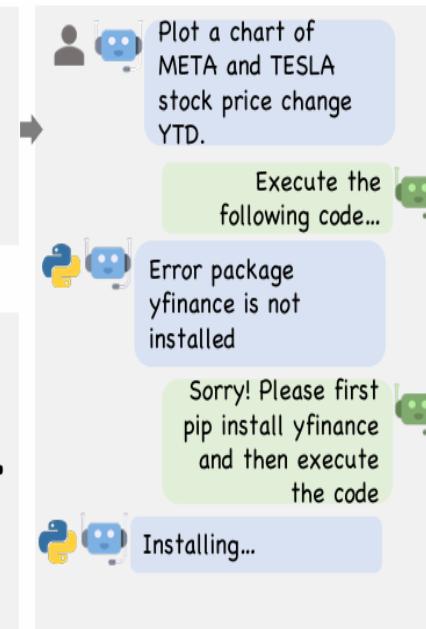
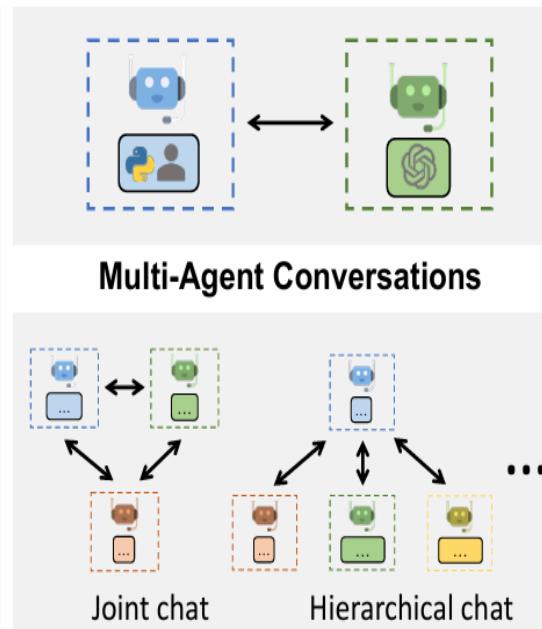
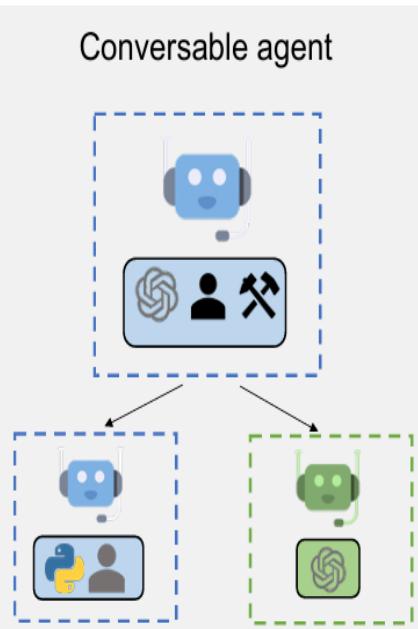
Method	Duration (s)	#Tokens	#Files	#Lines
GPT-Engineer	15.6000	7,182.5333	3.9475	70.2041
MetaGPT	154.0000	29,278.6510	4.4233	153.3000
ChatDev	148.2148	22,949.4450	4.3900	144.3450

Table 3: Software statistics include Duration (time consumed), #Tokens (number of tokens used), #Files (number of code files generated), and #Lines (total lines of code across all files) in the software generation process.

- Communicative Agents for Software Development” by Qian et al. (2023).
<https://arxiv.org/pdf/2307.07924>

Agentic Workflow

• Multiagent Collaboration – AutoGen



Agent Customization

Flexible Conversation Patterns

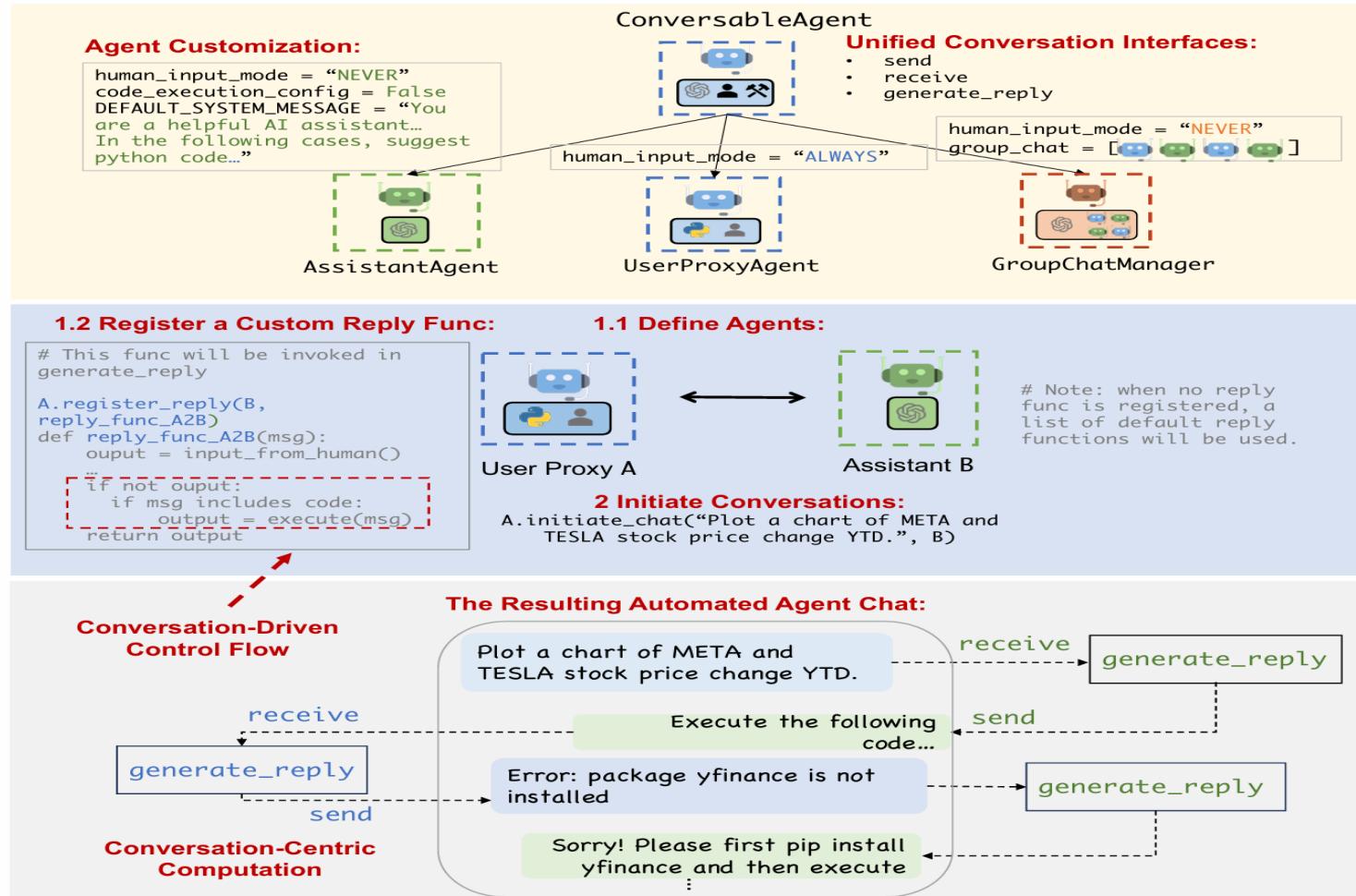
Example Agent Chat

- AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation" by Wu et al. (2023). <https://arxiv.org/pdf/2308.08155>

Agentic Workflow

• Multiagent Collaboration - How to use AutoGen

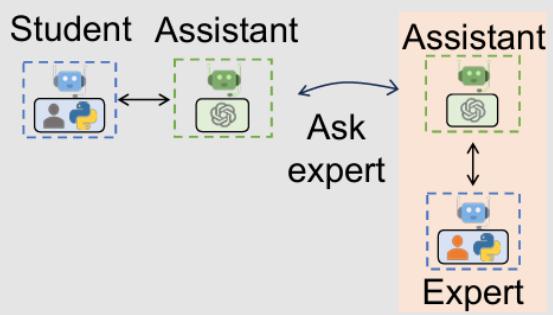
AutoGen Agents



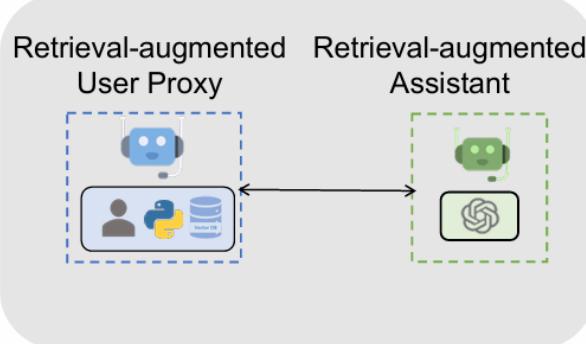
- AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation" by Wu et al. (2023). <https://arxiv.org/pdf/2308.08155>

Agentic Workflow

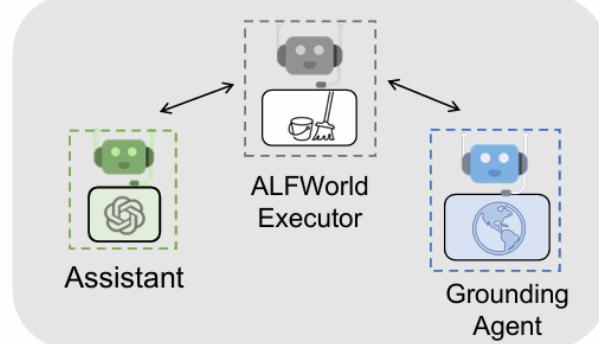
- Multiagent Collaboration – applications built using AutoGen



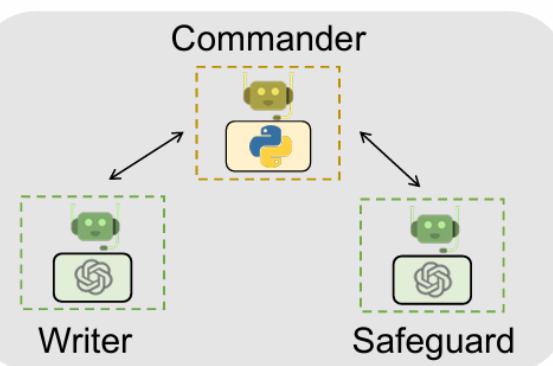
A1. Math Problem Solving



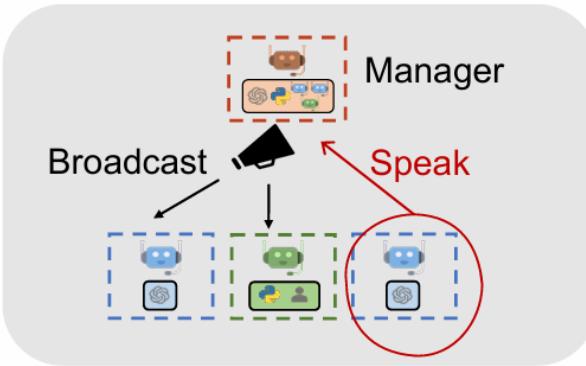
A2. Retrieval-augmented Chat



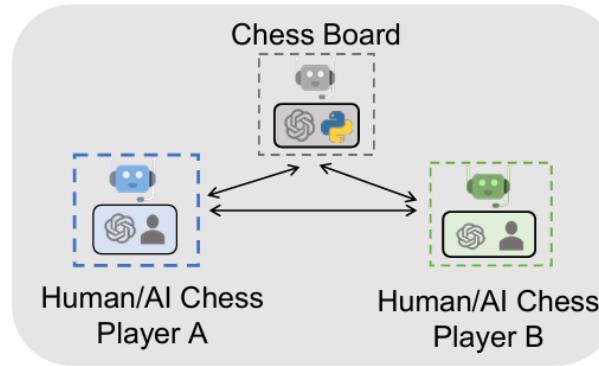
A3. ALF Chat



A4. Multi-agent Coding



A5. Dynamic Group Chat

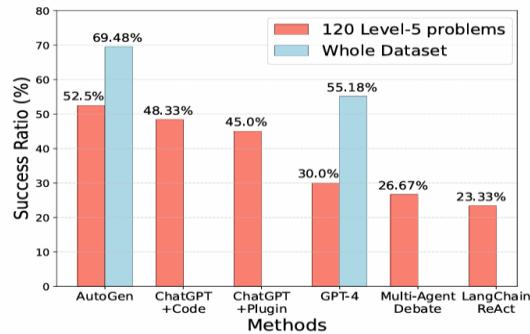


A6. Conversational Chess

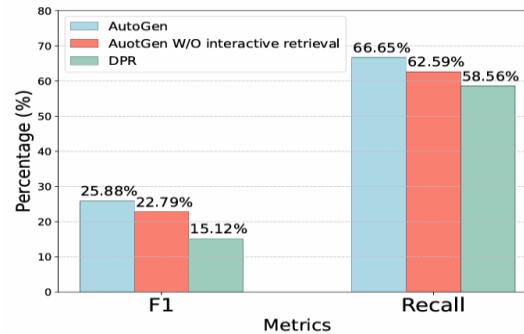
- AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation”
by Wu et al. (2023). <https://arxiv.org/pdf/2308.08155>

Agentic Workflow

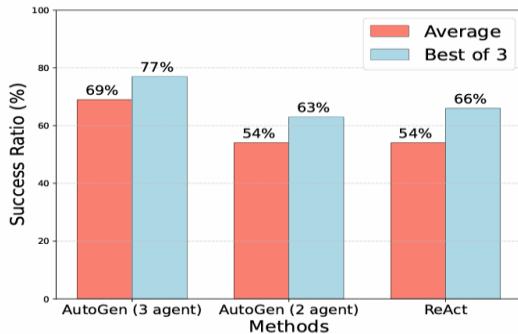
- Multiagent Collaboration – Performance Evaluation of AutoGen



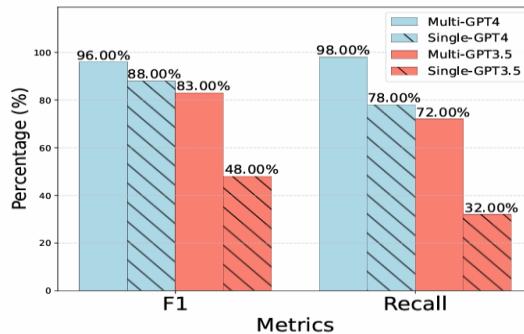
(a) A1: Performance on MATH (w/ GPT-4).



(b) A2: Q&A tasks (w/ GPT-3.5).



(c) A3: Performance on ALFWorld.



(d) A4: Performance on OptiGuide.

Figure 4: Performance on four applications A1-A4. (a) shows that AutoGen agents can be used out of the box to achieve the most competitive performance on math problem solving tasks; (b) shows that AutoGen can be used to realize effective retrieval augmentation and realize a novel interactive retrieval feature to boost performance on Q&A tasks; (c) shows that AutoGen can be used to introduce a three-agent system with a grounding agent to improve performance on ALFWorld; (d) shows that a multi-agent design is helpful in boosting performance in coding tasks that need safeguards.

- AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation” by Wu et al. (2023). <https://arxiv.org/pdf/2308.08155>

About Artificial General Intelligence (AGI)

Levels of AGI

Performance (rows) x Generality (columns)	Narrow <i>clearly scoped task or set of tasks</i>	General <i>wide range of non-physical tasks, including metacognitive abilities like learning new skills</i>
Level 0: No AI	Narrow Non-AI calculator software; compiler	General Non-AI human-in-the-loop computing, e.g., Amazon Mechanical Turk
Level 1: Emerging <i>equal to or somewhat better than an unskilled human</i>	Emerging Narrow AI GOFAI ⁴ ; simple rule-based systems, e.g., SHRDLU (Winograd, 1971)	Emerging AGI ChatGPT (OpenAI, 2023), Bard (Anil et al., 2023), Llama 2 (Touvron et al., 2023)
Level 2: Competent <i>at least 50th percentile of skilled adults</i>	Competent Narrow AI toxicity detectors such as Jigsaw (Das et al., 2022); Smart Speakers such as Siri (Apple), Alexa (Amazon), or Google Assistant (Google); VQA systems such as PaLi (Chen et al., 2023); Watson (IBM); SOTA LLMs for a subset of tasks (e.g., short essay writing, simple coding)	Competent AGI not yet achieved
Level 3: Expert <i>at least 90th percentile of skilled adults</i>	Expert Narrow AI spelling & grammar checkers such as Grammarly (Grammarly, 2023); generative image models such as Imagen (Shaharia et al., 2022) or Dall-E 2 (Ramesh et al., 2022)	Expert AGI not yet achieved
Level 4: Virtuoso <i>at least 99th percentile of skilled adults</i>	Virtuoso Narrow AI Deep Blue (Campbell et al., 2002), AlphaGo (Silver et al., 2016, 2017)	Virtuoso AGI not yet achieved
Level 5: Superhuman <i>outperforms 100% of humans</i>	Superhuman Narrow AI AlphaFold (Jumper et al., 2021 ; Varadi et al., 2021), AlphaZero (Silver et al., 2018), Stockfish (Stockfish, 2023)	Artificial Superintelligence (ASI) not yet achieved

<https://arxiv.org/pdf/2311.02462.pdf>

OpenAI's 5-level scale of AI progress

Level	Name	Description
1	Chatbots	AI with natural conversational language capabilities, like the current ChatG
2	Reasoners	AI capable of advanced problem solving, like a person with a PhD-level education
3	Agents	AI systems that can take actions independently or based on instructions
4	Innovators	AIs that can invent new ideas and contribute to human knowledge
5	Organizations	AIs that can independently perform tasks throughout an organization

Future of AGI

■ What is future of AGI?

- ◆ Benefit or doom?
- ◆ What is future job?

Sam Altman Says He Intends to Replace Normal People With AI

"Comparing AI to even the idea of median or average humans is a bit offensive."

/ ArtificialIntelligence / AI/Bots / OpenAI / SamAltman



'Godfather of A.I.' leaves Google after a decade to warn society of technology he's touted

PUBLISHED MON, MAY 1 2023 3:19 PM EDT | UPDATED MON, MAY 1 2023 5:58 PM EDT

Jennifer Elias
@JENN_ELIAS

SHARE f t in e

KEY POINTS

- Geoffrey Hinton, known as the "godfather of AI," is leaving his role at Google and plans to warn of the risks of the technology he's long promoted.
- Hinton told CNBC that he now thinks digital intelligence can "acquire hugely more knowledge than any individual biological agent."
- Google and Microsoft are among the mega-cap tech companies investing heavily in new AI models.

In this article

GOOGL +\$1.06 (+0.87%) 🔍 +

Follow your favorite stocks
[CREATE FREE ACCOUNT](#)



CNBC TV
Last Call
UP NEXT | [Shark Tank 08:00](#)



Explore 3M TwinAx high speed cable solutions

3M

[APPLY NOW](#)

Warning to Fast Development of AI



A Right to Warn about Advanced Artificial Intelligence

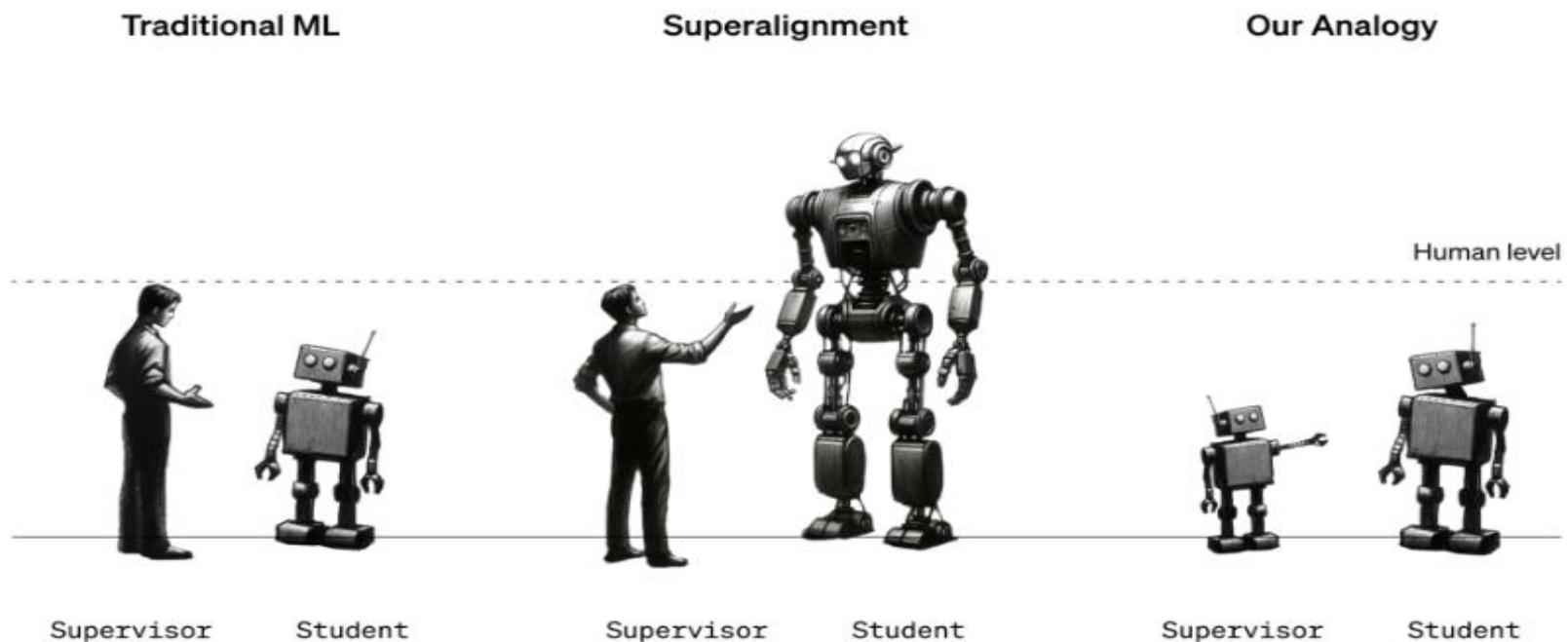
We are current and former employees at frontier AI companies, and we believe in the potential of AI technology to deliver unprecedented benefits to humanity.

We also understand the serious risks posed by these technologies. These risks range from the further entrenchment of existing inequalities, to manipulation and misinformation, to the loss of control of autonomous AI systems potentially resulting in human extinction. AI companies themselves have acknowledged these risks [[1](#), [2](#), [3](#)], as have governments across the world [[4](#), [5](#), [6](#)] and other AI experts [[7](#), [8](#), [9](#)].

We are hopeful that these risks can be adequately mitigated with sufficient guidance from the scientific community, policymakers, and the public. However, AI companies have strong financial incentives to avoid effective oversight, and we do not believe bespoke structures of corporate governance are sufficient to change this.

Countermeasure for Danger of AGI

WEAK-TO-STRONG GENERALIZATION: ELICITING STRONG CAPABILITIES WITH WEAK SUPERVISION by OpenAI



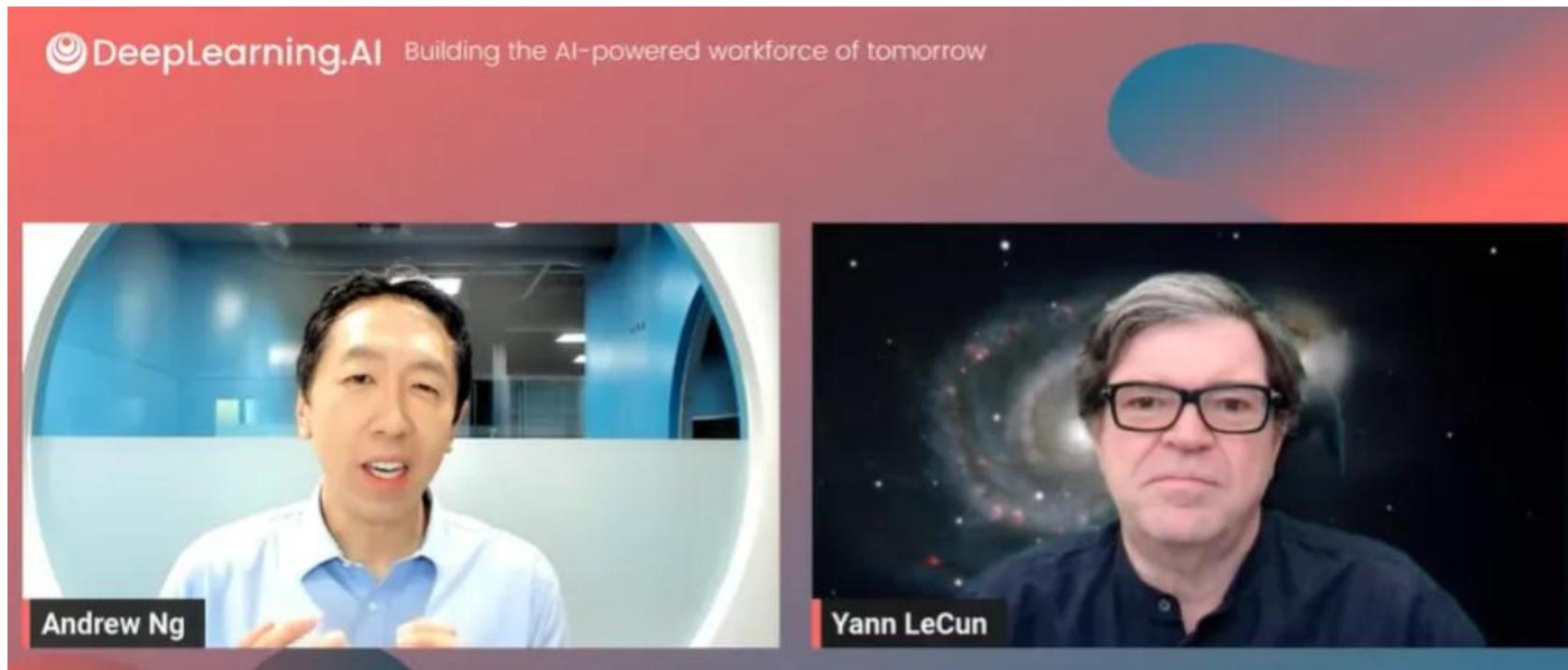
A simple analogy for superalignment: In traditional machine learning (ML), humans supervise AI systems weaker than themselves (left). To align superintelligence, humans will instead need to supervise AI systems smarter than them (center). We cannot directly study this problem today, but we can study a simple analogy: can small models supervise larger models (right)?

<https://cdn.openai.com/papers/weak-to-strong-generalization.pdf>

Future of AGI

■ Andrew Ng and Yann LeCun

- Don't be afraid. We're not there yet.
- The current Transformer structure won't create an AGI. We need a new structure, like human learning.



Problem of LLM (Yann LeCun)

◆ Intelligent Behavior and Learning in AI:

- For AI to exhibit human-like intelligence, it must understand world, remember, reason, and plan based on the physical world. Current LLMs rely on language-based learning and lack the sensory-based learning that humans use.

◆ Difference Between Language and Sensory Data:

- Humans deeply understand the world through sensory experiences, while LLMs learn solely through language. Sensory data offers much richer information than language, and human knowledge is largely derived from these experiences.

◆ Limitations of World Models:

- LLMs have a limited understanding of the real world and struggle to effectively process visual data, which hinders their ability to fully model the physical world.

Problem of LLM (Yann LeCun)

- ◆ Generative Models and Self-Supervised Learning:
 - While self-supervised learning has been crucial to the success of LLMs, generative models have shown limitations in effectively learning about the physical reality.
- ◆ Differences Between Human Intelligence and AI:
 - Humans learn and understand the world through observation and experience, whereas LLMs are restricted to language-based, limited learning, lacking the interaction with the physical world.
- ◆ Is it possible to make AGI with the current LLM structure?

Ability of Current LLM

- ◆ Emergent Ability
- ◆ We don't know mechanism of the ability in the black box.
- ◆ It learns better multi-modal data and more data.
- ◆ How does the auto-regressive attention learning work well in human like inference?
- ◆ We had not expect this situation several years ago and now expect dramatic evolution.
- ◆ Problem of electric power and adequacy of return on investment issues
- ◆ Large World Model for AGI
- ◆ Solution: Evolution of models, Future Computing



Thank you!

Discussion and Questions?