Enterprise Trends for Generative AI

Guest Speaker: Burak Gokturk

Burak Gokturk

VP, Cloud AI and AI/ML for Developers, Google Cloud

burakg@google.com

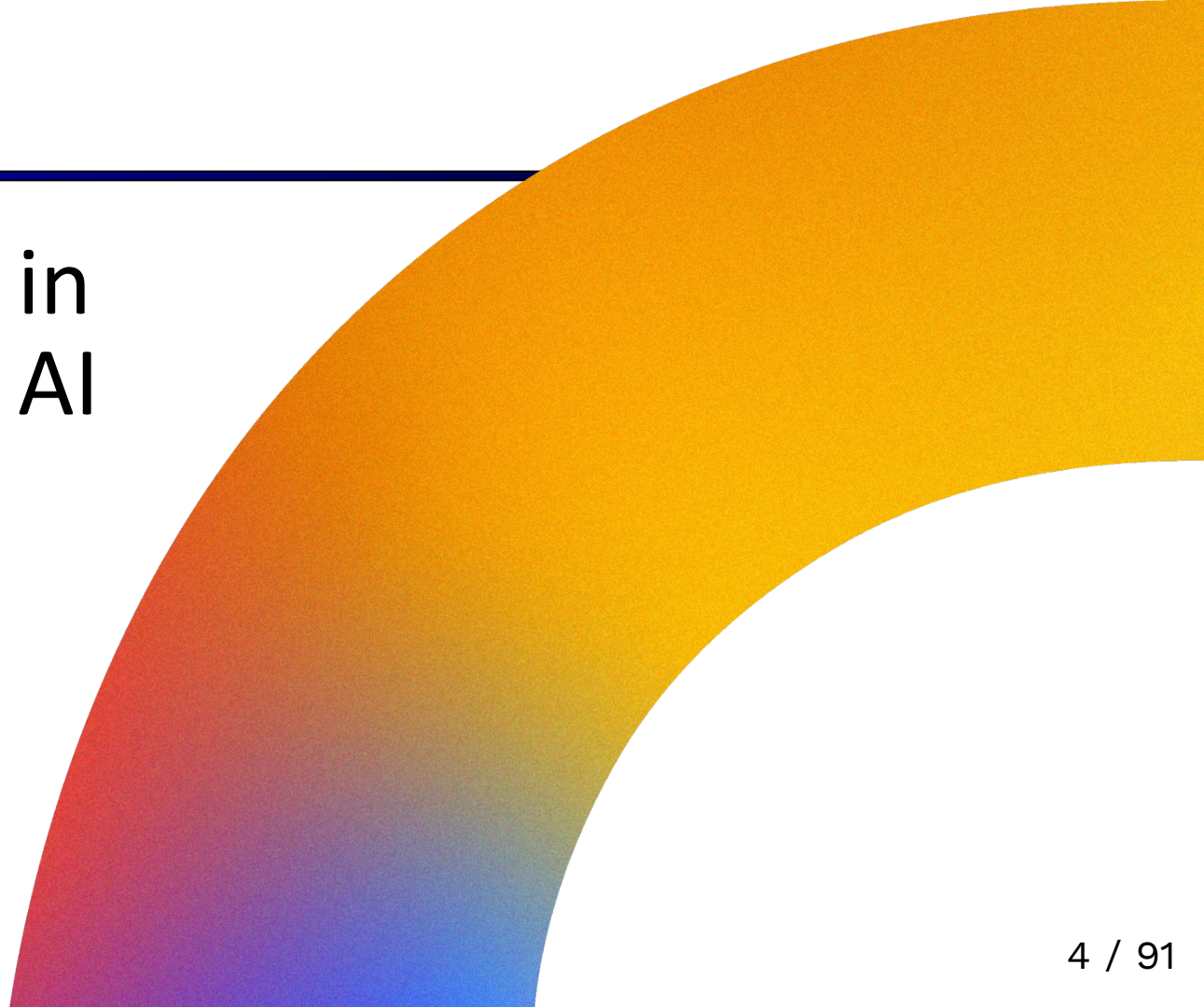
LLM Agents

Enterprise Trends for Generative AI

Key Blocks to build successful agents

What's next?

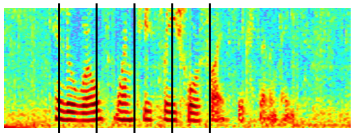
Key Trends in Generative AI



Some observations

- In recent years, ML has completely changed our expectations of what is possible with computers
- Increasing scale (compute, data, model size) delivers better results
- The kinds of computations we want to run and the hardware on which we run them is changing dramatically

A decade of amazing progress in what computers can do

Input		Output
Pixels:		“leopard”
Audio:		“How cold is it outside?”
“Hello, how are you?”		“Bonjour, comment allez-vous?”
Pixels:		“A cheetah lying on top of a car”

A decade of amazing progress in what computers can do

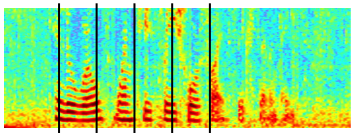
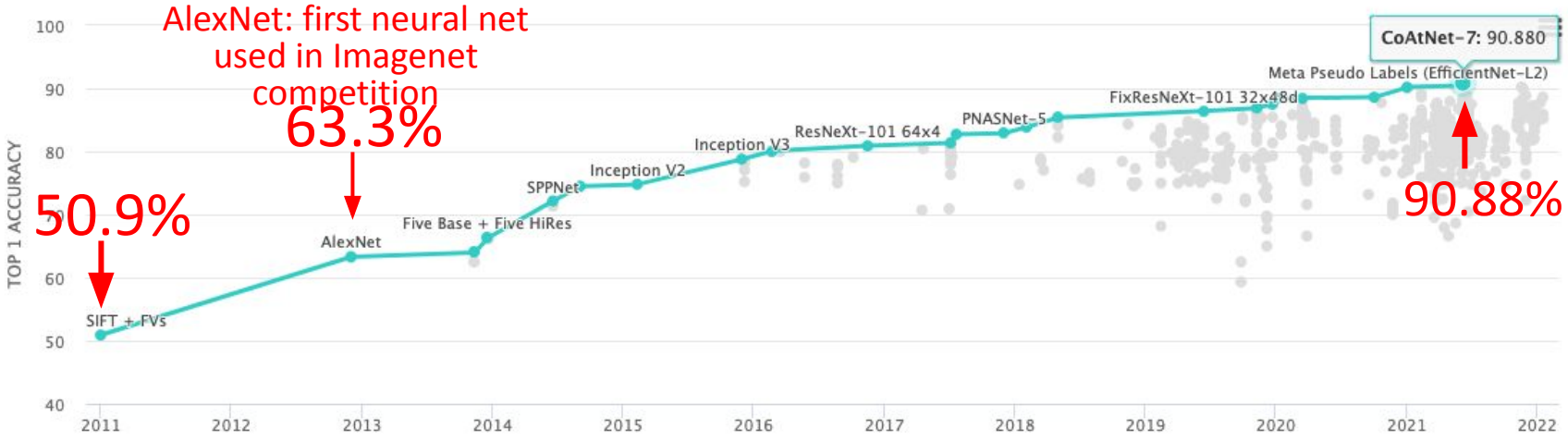
Output		Input
Pixels:		“leopard”
Audio:		“How cold is it outside?”
“Hello, how are you?”		“Bonjour, comment allez-vous?”
Pixels:		“A cheetah lying on top of a car”

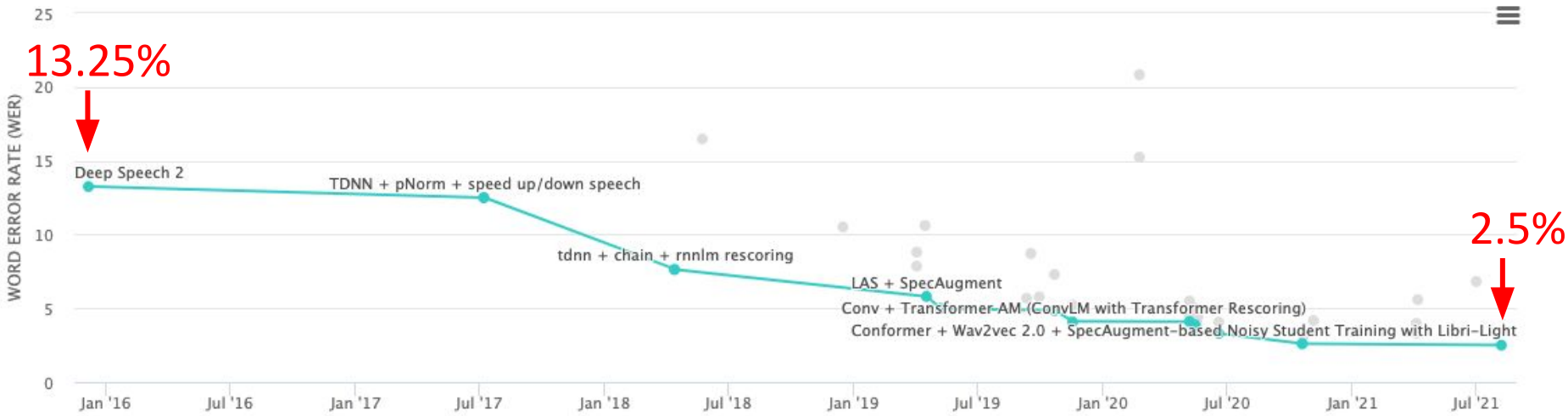
Image Classification on ImageNet

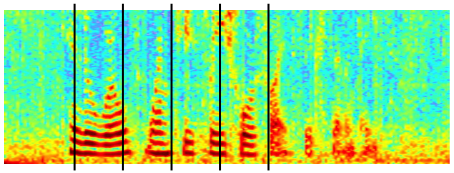


"Leopard"

Source: https://paperswithcode.com/sota/image-classification-on-imagenet

Speech Recognition on LibriSpeech test-other



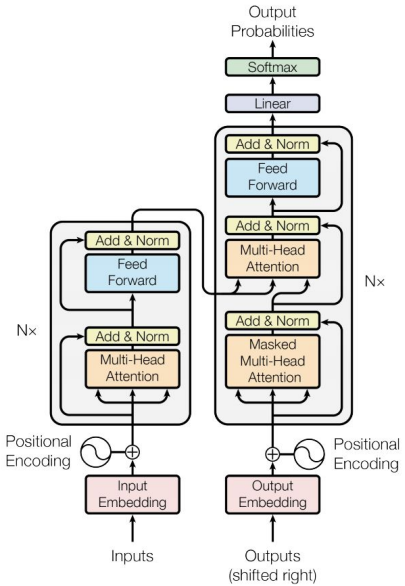


"How cold is it outside?"

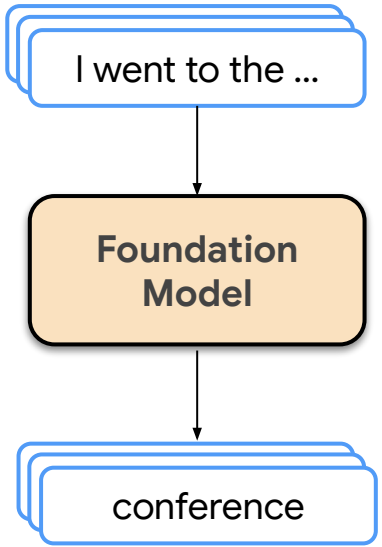
Source: https://paperswithcode.com/sota/speech-recognition-on-librispeech-test-other

Transformers + autoregressive training + massive data

Backbone architecture



Next token prediction



Pre-training on trillions of tokens

The cat sat on the mat

The teacher read a book

⋮

I love to dance

For tokenization: github.com/google-research/bert/blob/master/tokenization.py

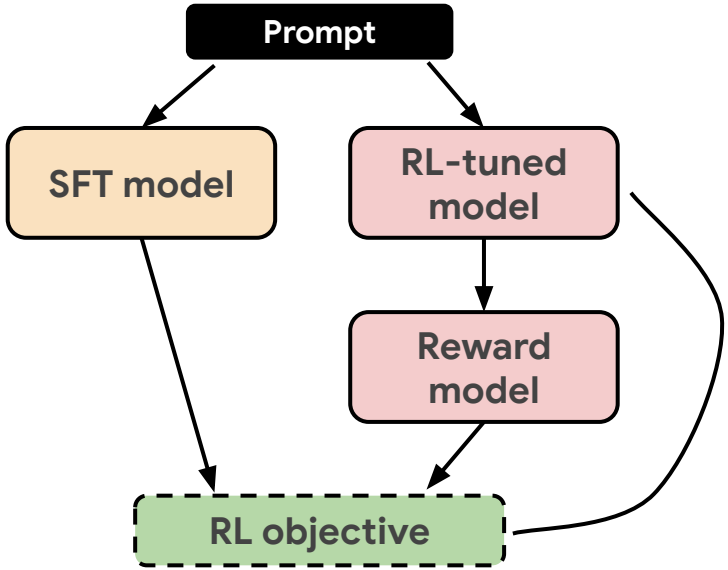
Towards useful AI agents

Supervised Fine-Tuning (SFT)

Prompt:
"I have pains in my lower back while sleeping. What could be causing this?"

Output (expert response):
"You might be experiencing a lower back strain, typically caused by lifting heavy objects or abrupt movements. This can lead to sharp pain in the lower back, especially when moving or lifting ..."

Reinforcement Learning from Human Feedback (RLHF)



Gemini

Training High Quality Multimodal Models at Scale

Project started in Feb 2023
Many collaborators from Google DeepMind, Google Research, and rest of Google

Goal: Train highly capable multimodal models and use them all across Google

Initial 1.0 public release in Dec 2023, 1.5 in Feb 2024

https://blog.google/technology/ai/google-gemini-ai

https://g.co/gemini

Gemini: A Family of Highly Capable Multimodal Models, by the Gemini Team, arxiv.org/abs/2312.11805

Gemini - Multimodal from the start

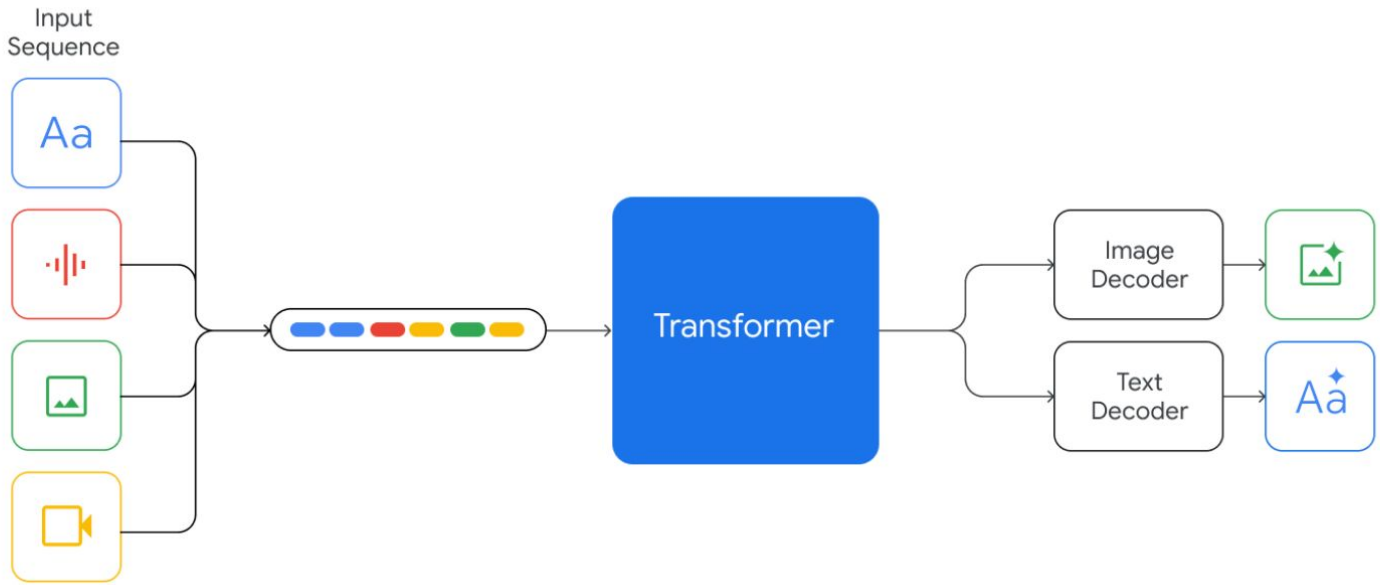


Figure 2 | Gemini supports interleaved sequences of text, image, audio, and video as inputs (illustrated by tokens of different colors in the input sequence). It can output responses with interleaved image and text.

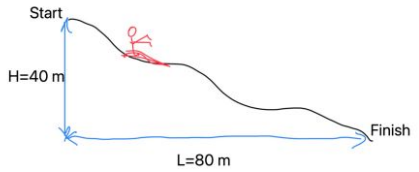
Gemini: A Family of Highly Capable Multimodal Models, by the Gemini Team, arxiv.org/abs/2312.11805

Gemini Multimodal Reasoning

Prompt

Here's a solution to a Physics problem by a student.

23. A skier slides down the frictionless slope as shown. What is the skier's speed at the bottom?



Solution:

The conservation of energy means that the energy of the skier is same at the beginning and at the end.

Start: $E = mgL$

End: $E = \frac{1}{2}mv^2$

$\Rightarrow mgL = \frac{1}{2}mv^2$

$\Rightarrow v = \sqrt{2gL} = \sqrt{2 \times 9.81 \times 80} = 39.6\ m.s^{-1}$

Try to reason about the question step by step.

Did the student get the correct answer? If the solution is wrong, please explain what is wrong and solve the problem.

Make sure to use LaTeX for math and round off the final answer to two decimal places.

Response

The student did not get the correct answer. The student made a mistake in the calculation of the potential energy at the start of the slope. The potential energy at the start of the slope is given by mgh, where m is the mass of the skier, g is the acceleration due to gravity, and h is the height of the slope. The student used the length of the slope, L, instead of the height, H, in the calculation of the potential energy.

The correct solution is:

The conservation of energy means that the total energy of the skier is the same at the beginning and at the end of the slope.

At the start of the slope, the skier has only potential energy, which is given by mgh.

At the end of the slope, the skier has only kinetic energy, which is given by $\frac{1}{2}mv^2$

Therefore, we can write:

$$mgh = \frac{1}{2}mv^2$$

Canceling out the mass m on both sides, we get:

$$gh = \frac{1}{2}v^2$$

Solving for v, we get:

$$v = \sqrt{2gh}$$

Substituting the given values, we get:

$$v = \sqrt{2 \times 9.81 \times 40} = 28.01\ m/s$$

Therefore, the skier's speed at the bottom of the slope is 28.01 m/s.

Gemini 1.5: Initial release in February 2024

Long context capability demonstrated up to 10M tokens, with public preview offering up to 1M tokens of text and/or video

information in the context window is "**clearer**" (it hasn't been mixed together/perturbed over many gradient descent steps with other training data)

⇒ less hallucination
⇒ enables in-context learning

Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context, arxiv.org/abs/2403.05530

Try it at https://ai.google.dev/

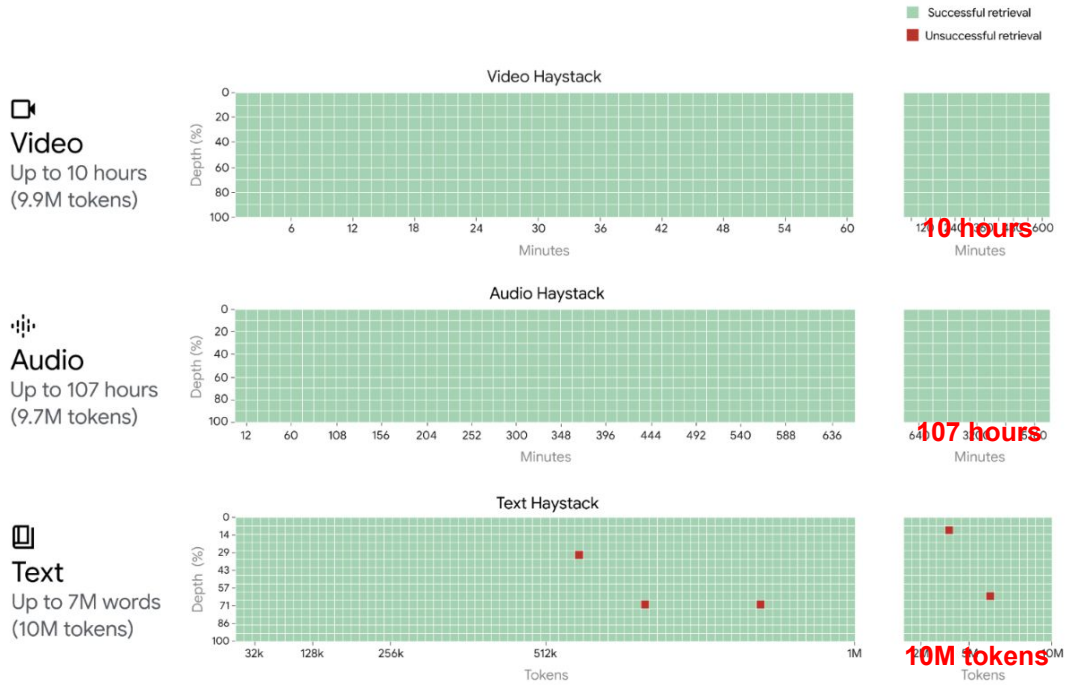
Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context, arxiv.org/abs/2403.05530

Gemini 1.5 Pro Long Context Evaluation

"Needle in haystack" test for text, audio, and video

Green is good, and red is not good, and these are almost entirely green (>99.7% recall), even out to 10M tokens.

Modeled after Greg Kamradt's github.com/gkamradt/LLMTest_NeedleInAHaystack test that probes a model's ability to retrieve specific information from its context.



Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context, arxiv.org/abs/2403.05530

Gemini 1.5 Pro-Exp-0801 evaluation via lmsys.org in August, 2024

lmsys.org
@lmsysorg

Exciting News from Chatbot Arena!

@GoogleDeepMind's new Gemini 1.5 Pro (Experimental 0801) has been tested in Arena for the past week, gathering over 12K community votes.

For the first time, Google Gemini has claimed the #1 spot, surpassing GPT-4o/Claude-3.5 with an impressive score of 1300 (!), and also achieving #1 on our Vision Leaderboard.

Category: Overall

Overall Questions

#models: 123 (100%) #votes: 1,575,901 (100%)

Google Gemini 1.5 Pro (0801) #1 in Arena!!

Rank* (UB)	Model	Arena Score	95% CI	Votes	Organization	License
1	Gemini-1.5-Pro-Exp-0801	1300	+6/-5	12672	Google	Proprietary
2	GPT-4o-2024-05-13	1286	+3/-2	69832	OpenAI	Proprietary
2	GPT-4o-mini-2024-07-18	1280	+6/-4	12047	OpenAI	Proprietary
4	Claude 3.5 Sonnet	1271	+3/-4	40174	Anthropic	Proprietary
4	Gemini-Advanced-0514	1266	+3/-4	50686	Google	Proprietary
4	Meta-Llama-3.1-405b-Instruct	1262	+6/-7	8454	Meta	Llama 3.1 Community
5	Gemini-1.5-Pro-API-0514	1261	+3/-3	62018	Google	Proprietary
6	Gemini-1.5-Pro-API-0409-Preview	1257	+4/-3	55673	Google	Proprietary
6	GPT-4-Turbo-2024-04-09	1257	+3/-2	79537	OpenAI	Proprietary
10	GPT-4-1106-preview	1251	+3/-2	90007	OpenAI	Proprietary
10	Claude 3 Opus	1248	+2/-3	151003	Anthropic	Proprietary
10	Athene-70b	1245	+6/-6	5511	NexusFlow	CC-BY-NC-4.0
10	Meta-Llama-3.1-70b-Instruct	1242	+10/-7	4450	Meta	Llama 3.1 Community

https://x.com/lmsysorg/status/1819048821294547441

Enterprise Trends

Trend 1- The obvious: AI is moving so much faster

AI is moving faster...Why?

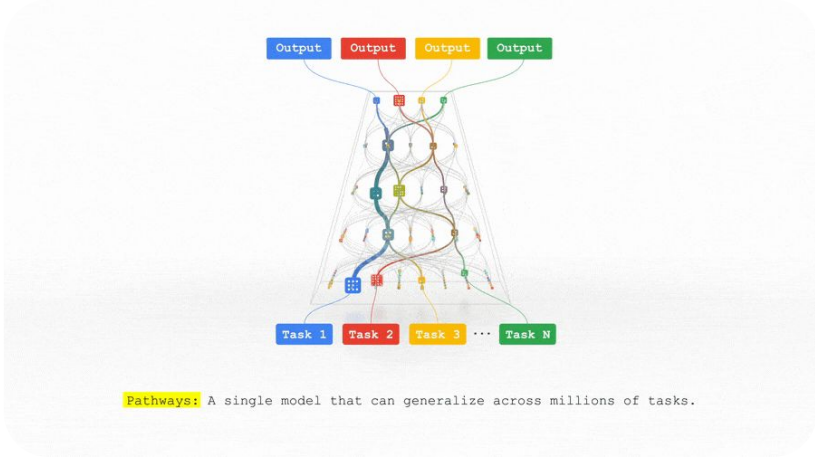
The amount of data needed has come down

Anyone can develop AI

Trend 2- Technical Trends

Where are we headed?

Separate models for different tasks



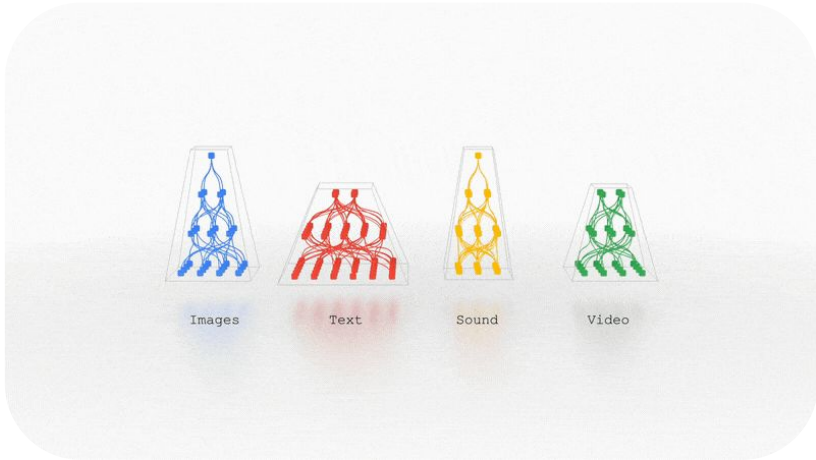
Single model that can generalize across millions of tasks

Where are we headed?

Dense models

Efficient sparse models

Where are we headed?



Single modality models

Models that deal with many modalities

Trend 3- It is the choice of the platform that matters

LMSys Chat Leaderboard Ranks

Rank* (UB)	Model	Arena Score	95% CI	Votes	Organization	License	Knowledge Cutoff
1	ChatGPT-4o-latest (2024-08-08)	1316	+4/-4	24023	OpenAI	Proprietary	2023/10
2	Gemini-1.5-Pro-Exp-0827	1301	+5/-5	19910	Google	Proprietary	2023/11
2	Gemini-1.5-Pro-Exp-0801	1298	+4/-4	25211	Google	Proprietary	2023/11
2	Grok-2-08-13	1295	+6/-6	10019	xAI	Proprietary	2024/3
5	GPT-4o-2024-05-13	1286	+3/-2	82934	OpenAI	Proprietary	2023/10
6	GPT-4o-mini-2024-07-18	1274	+4/-4	23147	OpenAI	Proprietary	2023/10
6	Gemini-1.5-Flash-Exp-0827	1271	+7/-6	6282	Google	Proprietary	2023/11
6	Claude 3.5 Sonnet	1270	+3/-3	53352	Anthropic	Proprietary	2024/4
6	Gemini Advanced App (2024-05-14)	1266	+3/-3	52225	Google	Proprietary	Online

Key success factors for generative AI

Access to a broad set of models

...so you can find the best model for their use case and budget

Ability to customize models with your data

...so you can improve the quality, latency, and performance of the model

A platform for managing models in production

...so you can deploy and manage models in a scalable way

Choice and flexibility at every level

....so you can avoid vendor lock-in and use the best tools for the job

Trend 4- Cost of API calls is approaching 0

Gemini 1.5 Flash

Input price (per 1M tokens)

$0.075 for <= 128K tokens
$0.15 for > 128K tokens

Context caching (per 1M tokens)

$0.01875 for <= 128K tokens
$0.0375 for > 128K tokens
$1.00 / 1 million tokens per hour (storage)
Learn more

Output price (per 1M tokens)

$0.30 for <= 128K tokens
$0.60 for > 128K tokens

Gemini 1.5 Pro

Input price (per 1M tokens)

$3.50 for <= 128K tokens
$7.00 for > 128K tokens

Context caching (per 1M tokens)

$0.875 for <= 128K tokens
$1.75 for > 128K tokens
$4.50 / 1 million tokens per hour (storage)
Learn more

Output price (per 1M tokens)

$10.50 for <= 128K tokens
$21.00 for > 128K tokens

Trend 5- Search

Another big realization

LLM and Search need to come together.

Trend 6- Enterprise Search/Assistant

Enterprise Learnings with their AI investments

Key success factors for generative AI

Access to a broad set of models

...so you can find the best model for their use case and budget

Ability to customize models with your data

...so you can improve the quality, latency, and performance of the model

A platform for managing models in production

...so you can deploy and manage models in a scalable way

Choice and flexibility at every level

....so you can avoid vendor lock-in and use the best tools for the job

Vertex AI

Google Cloud Generative AI platform

Agent Builder

OOTB and custom agents | Search
Orchestration | Extensions | Connectors | Document processors | Retrieval engines | Rankers | Grounding

Model Builder

Prompt | Serve | Tune | Distill | Eval | Notebooks I Training | Feature store | Pipelines | Monitoring

Model Garden

Google | Open | Partner

Google Cloud Infrastructure (GPU/TPU) | Google Data Cloud

Vertex AI

Google Cloud Generative AI platform

Agent Builder

OOTB and custom agents | Search
Orchestration | Extensions | Connectors | Document processors | Retrieval engines | Rankers | Grounding

Model Builder

Prompt | Serve | Tune | Distill | Eval | Notebooks I Training | Feature store | Pipelines | Monitoring

Model Garden

Google | Open | Partner

Google Cloud Infrastructure (GPU/TPU) | Google Data Cloud

Vertex AI Model Garden

130+

curated models


Gemini

New Preview

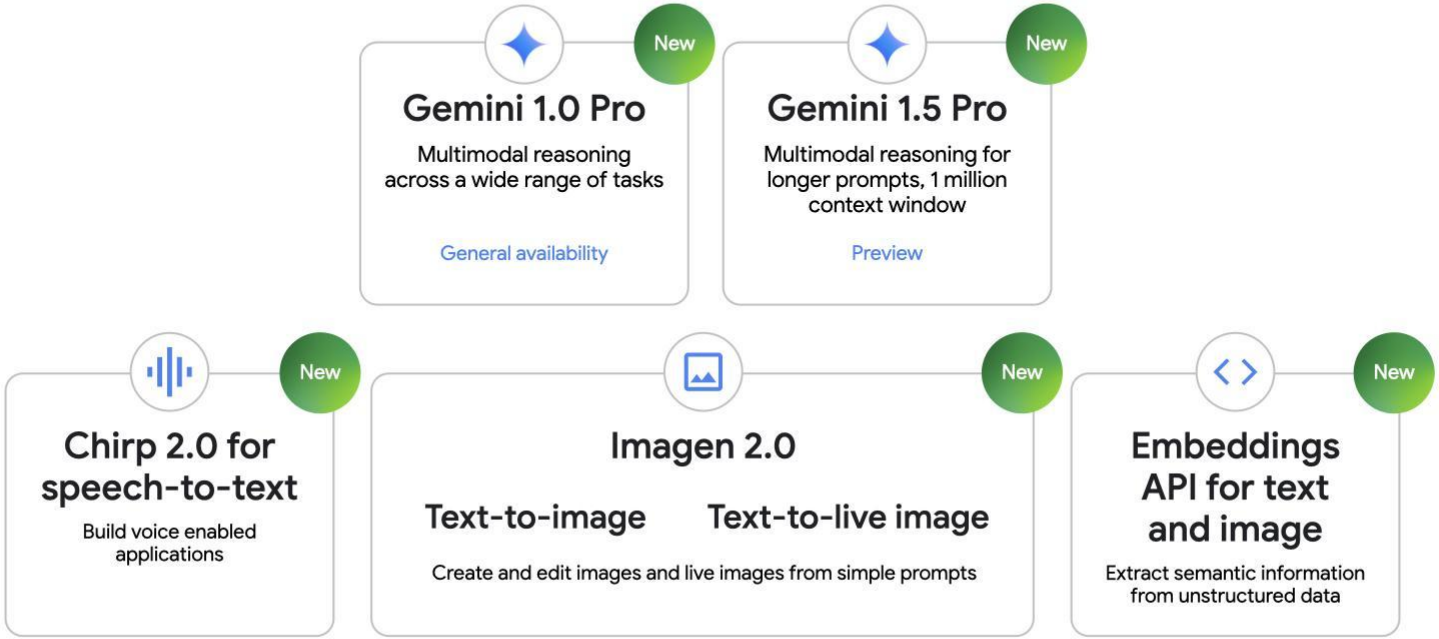
Gemini 1.5 Pro

1 M context window

Audio modality and video inclusive of audio

Foundation models on Vertex AI

Across a variety of model sizes to address use cases



Open model ecosystem

New availability

Claude 3 on Vertex AI

Haiku | Sonnet | Opus*

*Coming soon

Foundation models are powerful tools, but they are only as valuable as your ability to use them in context of your business use case and successfully deploy them to production.

Vertex AI

Google Cloud Generative AI platform

Agent Builder

OOTB and custom agents | Search
Orchestration | Extensions | Connectors | Document processors | Retrieval engines | Rankers | Grounding

Model Builder

Prompt | Serve | Tune | Distill | Eval | Notebooks | Training | Feature store | Pipelines | Monitoring

Model Garden

Google | Open | Partner

Google Cloud Infrastructure (GPU/TPU) | Google Data Cloud

Vertex AI

Google Cloud Generative AI platform

Agent Builder

OOTB and custom agents | Search
Orchestration | Extensions | Connectors | Document processors | Retrieval engines | Rankers | Grounding

Model Builder

Prompt | Serve | Tune | Distill | Eval | Notebooks | Training | Feature store | Pipelines | Monitoring

Model Garden

Google | Open | Partner

Google Cloud Infrastructure (GPU/TPU) | Google Data Cloud

Augmentation tools to enhance capabilities of foundation models

Tuning/Distillation

- Customize based on specific data and use case
- Create a smaller model for cost/latency purposes

Grounding

- Combine with search to make it factual

Extensions/Function Calling

- Function calling to be able to make LLMs on areas where they perform poorly

Key Components of Customizations and Agent Builder

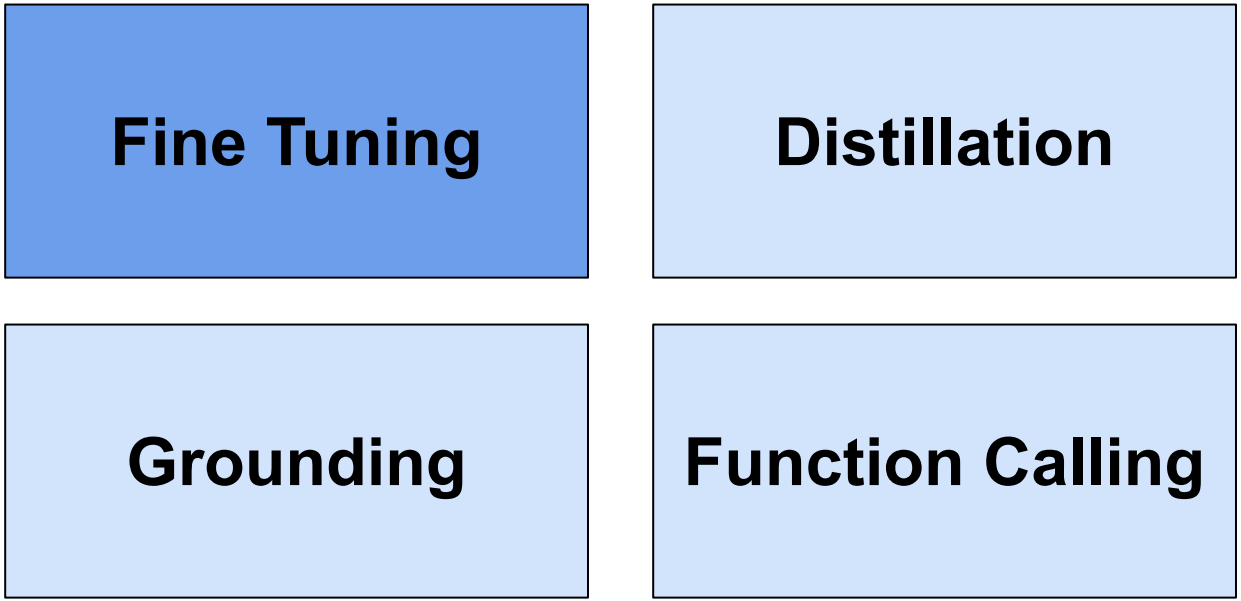
Fine Tuning

Distillation

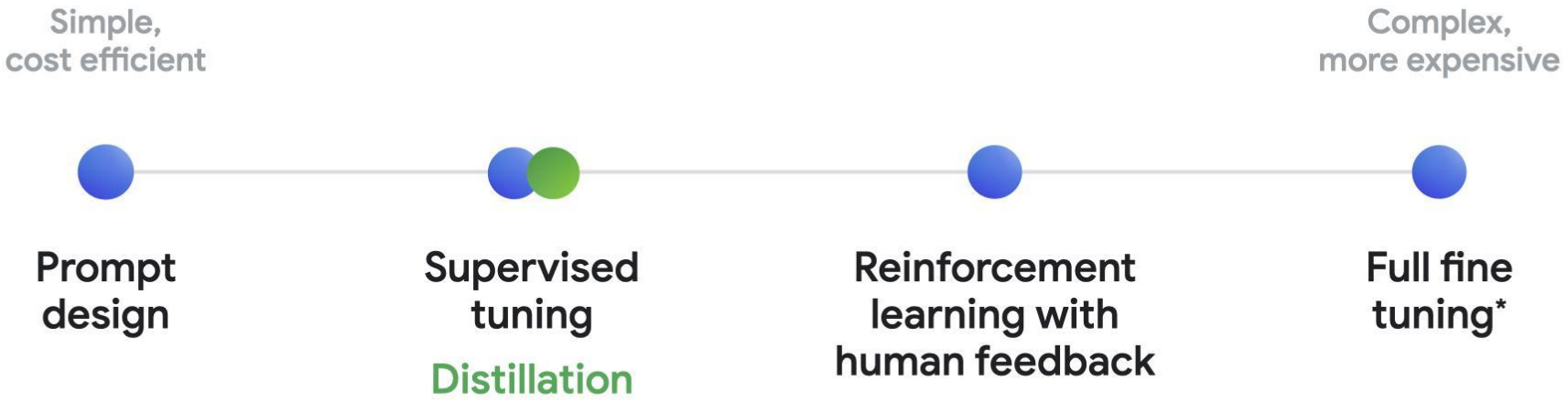
Grounding

Function Calling

Key Components of Customizations and Agent Builder

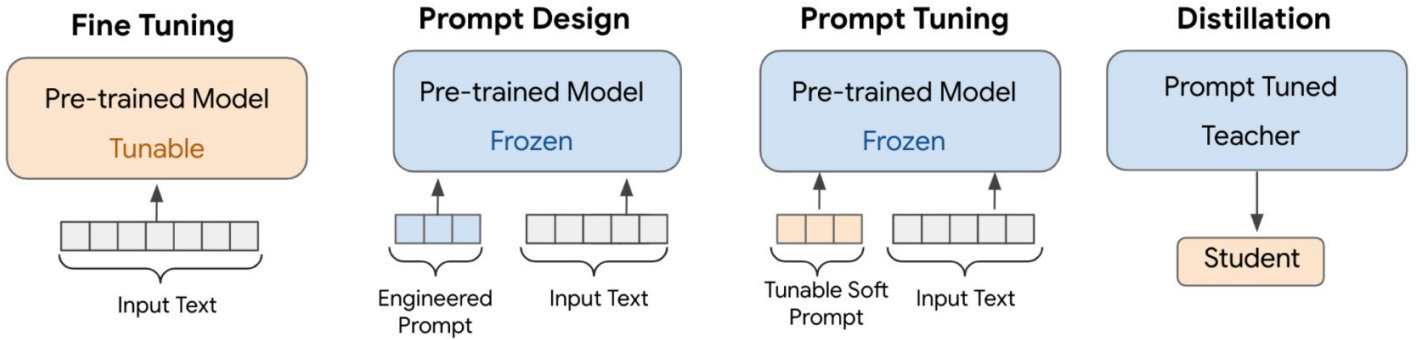


Tune and customize with your data



*For open models on Vertex AI

Popular model adaptation approaches and research goals



How does it work?	Regular full model fine-tuning	No training, just craft (design) the input to the model with optional examples	Train only 1-100 soft tokens (token size = 8192 floats)	Distill 64B teacher to a 1B ULM or xxxM T5
Training data	10k - 100k	1-10 examples (2048 tokens)	100 - 1k	100 - 1k labelled 100k - 100M unlabelled
Training cost	Prohibitively Expensive Meaningful only for 1B ULM perhaps	Zero	Average cost	Expensive
Training time	Weeks / Months	Zero	Tens of minutes for 64B ULM	Days
Quality	Very High	High	Very High (as good as fine-tuning for >10B)	Very High (slightly lower than teacher)
Inference Cost	High	Zero-shot: Higher (2x) Few-shot: Even Higher (2x - 5x higher)	High - only slightly higher vs fine-tuned model	Very Low
LLM size	All sizes; FLAN-540B is very expensive.	Large LLM >62B. 540B	moderate-size (e.g. T5-3B) 1 ~ 64B	64B → 1B or T5

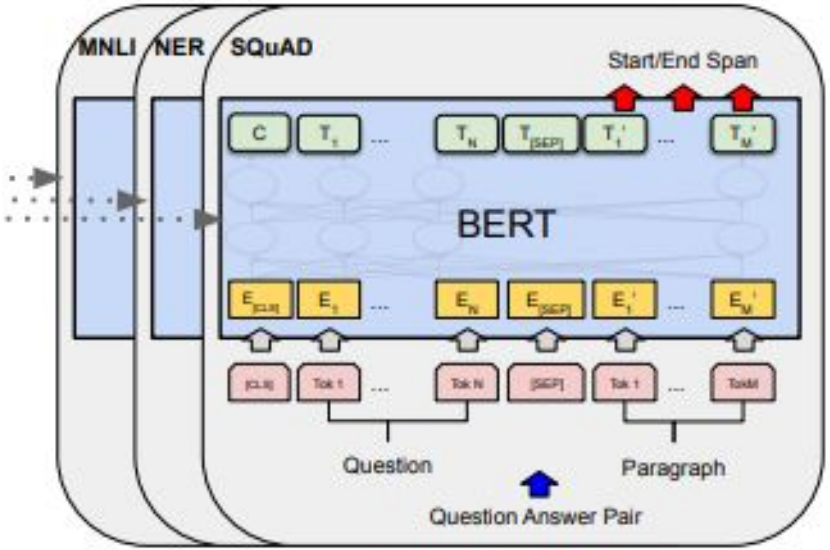
Conventional Fine-tuning

Basic steps:

- Get a pretrained model checkpoint (e.g. BERT)
- Have a new dataset/task
- Do supervised learning on new dataset and update the weights of the new model

Requires:

- Modest amount of compute (e.g. xxx chips for a few days for a 340B model)
- In-depth knowledge of the model architecture

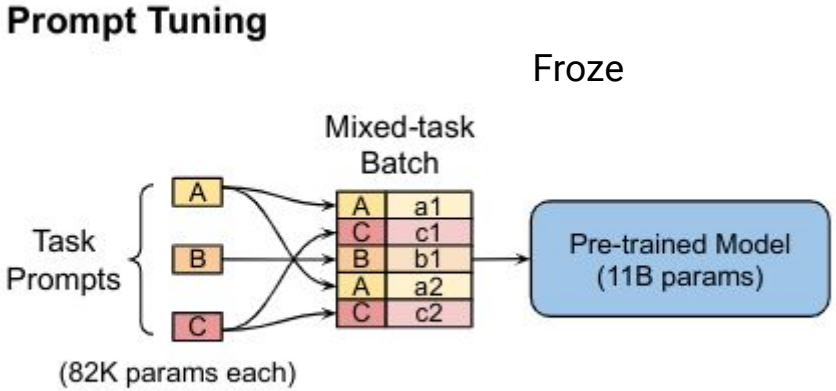


Fine-Tuning

Conventional Prompt Tuning

Basic steps:

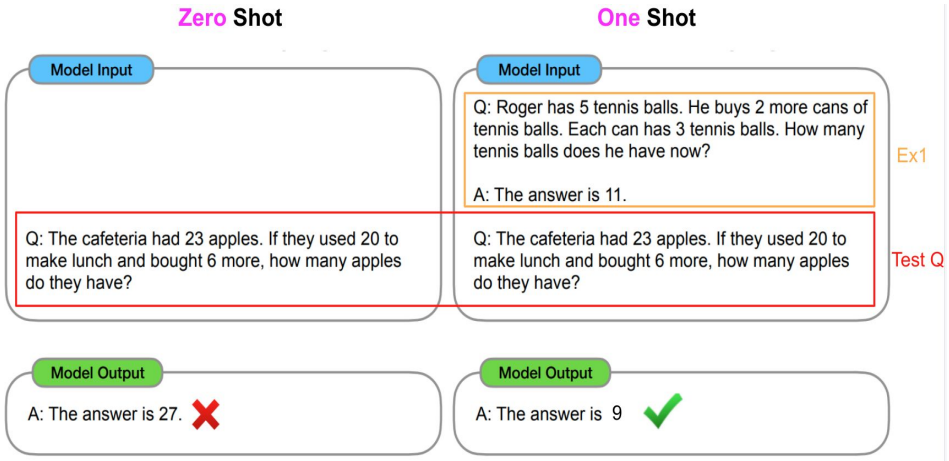
- Freeze the backbone model.
- Prepend a soft prompt (e.g. learnable) to the input
- Only optimize the prompt to adapt to downstream tasks.



Lester et al https://arxiv.org/pdf/2104.08691.pdf

Overview of Prompting

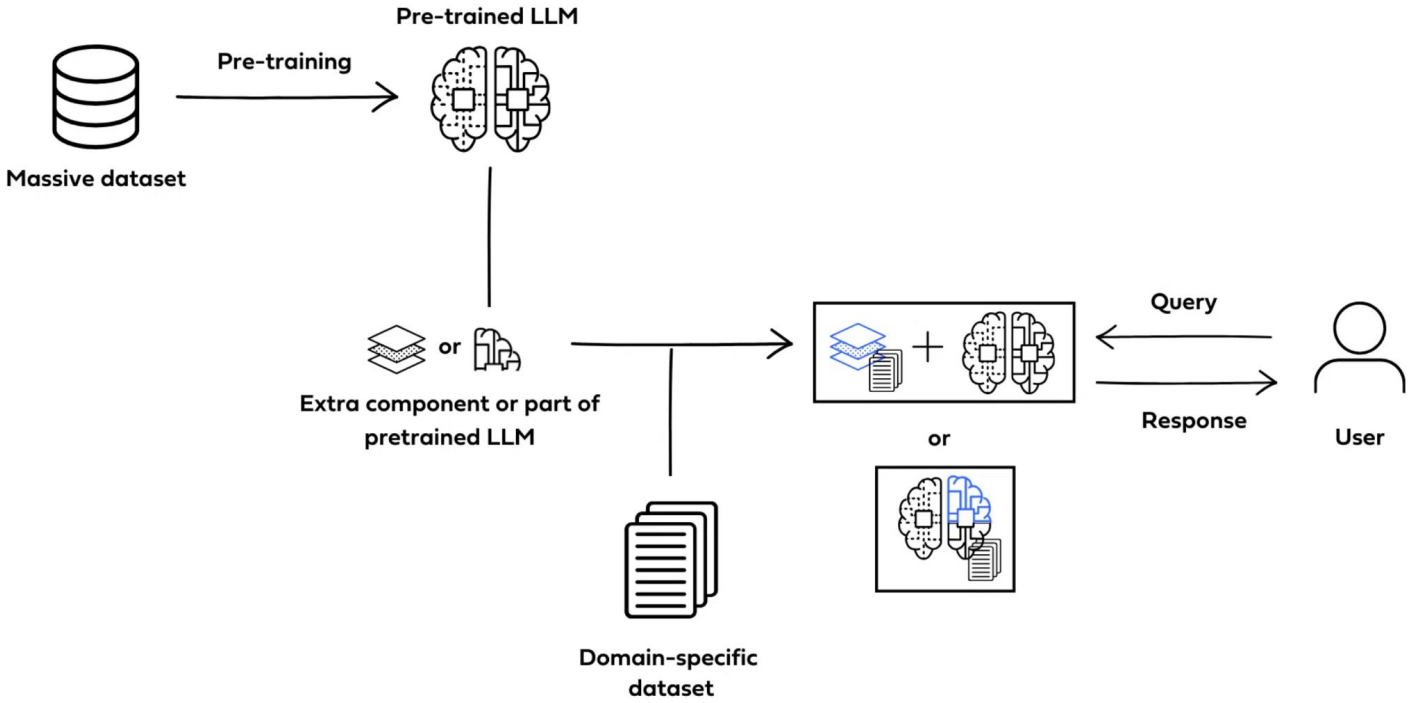
- **In-context learning**: an LLM is given a prompt including a few training examples and a test instance as input.
- By leveraging its "**autoregressive power**", LLMs understand and learn from the in-context examples to generate the output for the test instance directly, without any update to its parameters.
- Perform in a **zero-shot/few-shot** manner.
- Context learning is one of the most common ways to use giant LLMs



Advantages:

- No training, only inference
- Enables real time interaction (e.g. ChatGPT)
- Lowered requirements on train dataset size or label cost: only a few examples (<10) are enough
- Better generalization and less likely to overfit on the training data

What is parameter-efficient fine tuning



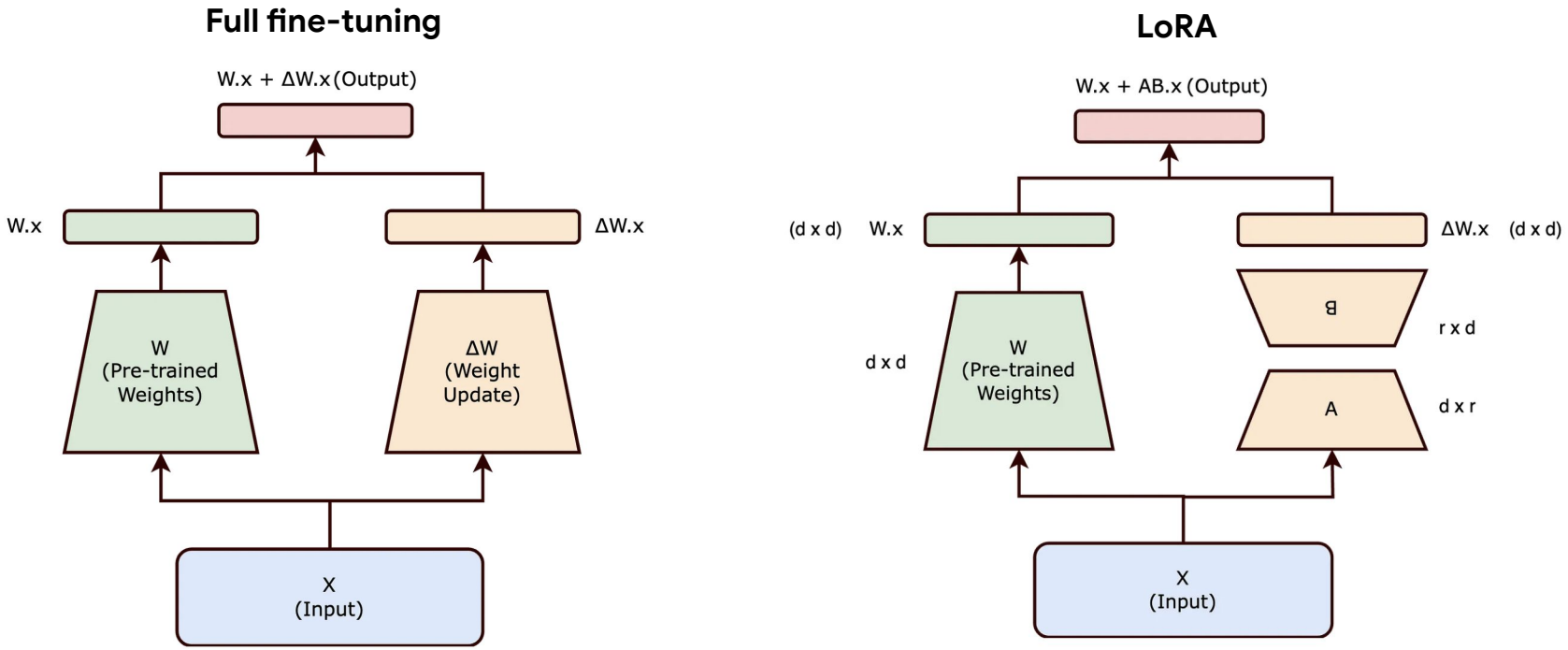
Resource link

Why do we need PEFT? Pros and Cons

Criteria	Parameter-efficient Fine Tuning (PEFT)	Conventional Fine-tuning
Objective	Enhance pre-trained models in data-scarce, low-compute areas	Boost pre-trained models with ample data & compute
Data Necessity	Limited examples in dataset	Abundant examples in dataset
Time Required for Training	Quicker relative to conventional method	Slower relative to PEFT
Resource Consumption	Economizes computational means	Utilizes extensive computational means
Model Parameter Adjustment	Alters select model components	Overhauls entire model structure
Overfitting Susceptibility	Lower risk due to restricted modification	Higher risk due to thorough modification
Outcome of Training	Adequate but generally below conventional method	Usually outperforms PEFT
Typical Applications	Best in resource-constrained or data-limited situations	Optimal for resource-rich environments

Resource link

Key Idea



- Decomposing updated parameters into a product of lower rank matrices

Resource link

Details of LoRA

Pre-trained model weight: $W_0 \in \mathbb{R}^{d \times k}$

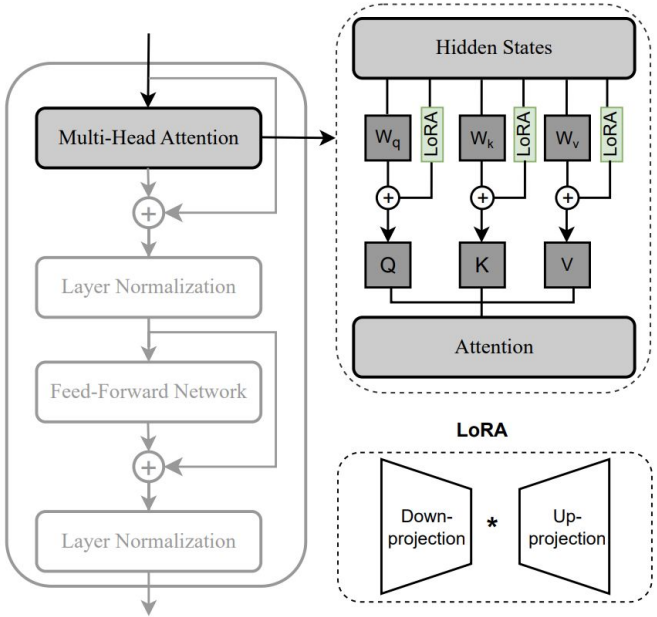
Weights with LoRA tuning: $W_0 + \Delta W = W_0 + BA$

Trainable parameters: $B \in \mathbb{R}^{d \times r}, A \in \mathbb{R}^{r \times k}$

Rank of trainable parameters: $\text{rank } r \ll \min(d, k)$

Original forward pass: $h = W_0 x$

Modified forward pass: $h = W_0 x + \Delta W x = W_0 x + BAx$



- This can apply to any dense layers in the foundation models.
  - But applying to all dense layers may not be the optimal.
  - Only applying adaptation on attention weights (not on MLPs)

Resource link

Advantages of LoRA

- No additional inference latency

Batch Size Sequence Length $\|\Theta\|$	32 512 0.5M	16 256 11M	1 128 11M
Fine-Tune/LoRA	1449.4±0.8	338.0±0.6	19.8±2.7
$\text{Adapter}^{\text{L}}$	1482.0±1.0 (+2.2%)	354.8±0.5 (+5.0%)	23.9±2.1 (+20.7%)
$\text{Adapter}^{\text{H}}$	1492.2±1.0 (+3.0%)	366.3±0.5 (+8.4%)	25.8±2.2 (+30.3%)

- In some cases, it shows better performance even in comparison to full fine tuning

Model&Method	# Trainable Parameters	WikiSQL Acc. (%)	MNLI-m Acc. (%)	SAMSum R1/R2/RL
GPT-3 (FT)	175,255.8M	**73.8**	89.5	52.0/28.0/44.5
GPT-3 (LoRA)	4.7M	73.4	**91.7**	**53.8/29.8/45.9**
GPT-3 (LoRA)	37.7M	**74.0**	**91.6**	53.4/29.2/45.1

Key Components of Customizations and Agent Builder

Fine Tuning

Distillation

Grounding

Function Calling

Deploying LLMs continues to be challenging

- **Customers do not need Unicorn/Gemini-XL/GPT4 for every task**
- **Can we reduce the serving cost & latency while maintaining high performance**

Report: GitHub Copilot Loses an Average of $20 Per User Per Month

sentiments.ts | write_sql.go | parse_expenses.py | addresses.rb

```ts
#!/usr/bin/env ts-node

import { fetch } from "fetch-h2";

// Determine whether the sentiment of text is positive
// Use a web service
async function isPositive(text: string): Promise<boolean> {
  const response = await fetch(`http://text-processing.com/api/sentiment/`, {
    method: "POST",
    body: `text=${text}`,
    headers: {
      "Content-Type": "application/x-www-form-urlencoded",
    },
  });
  const json = await response.json();
  return json.label === "pos";
}
```

Copilot

Clem Delangue 🤗 • Following
Co-founder & CEO at Hugging Face
2w •

My prediction: in 2024, most companies will realize that smaller, cheaper, more specialized models make more sense for 99% of AI use-cases.

Memory & Compute Efficient LLM

Quantization

Speedup 6B LLM inference by 2x

Retrieval Enhancement

Model size reduction by 50x in text generation

Side Tuning

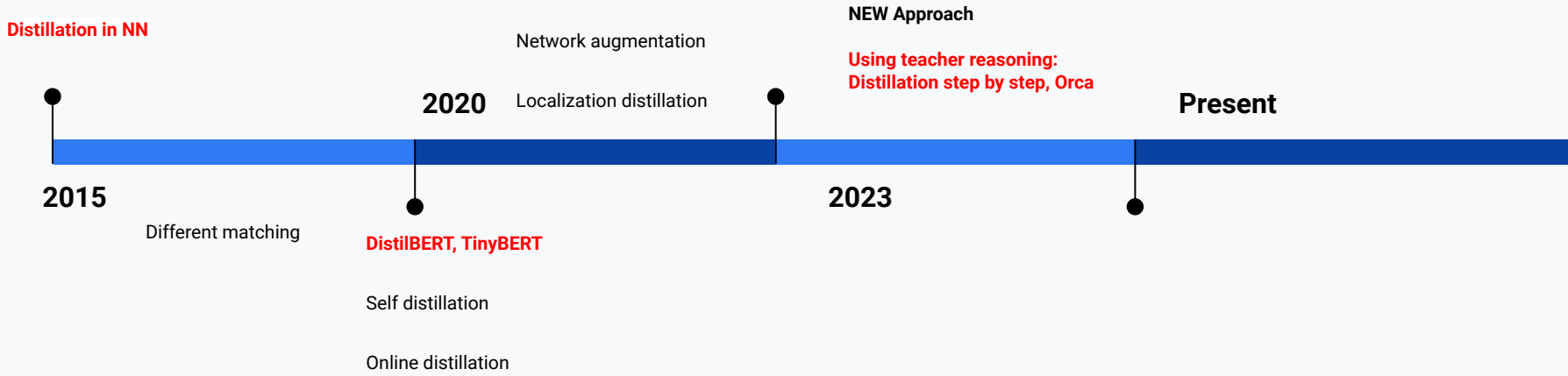
Avoid back-prop the backbone model

Distillation

Model size reduction & perf improvement

go/cair_efficient_llm

History

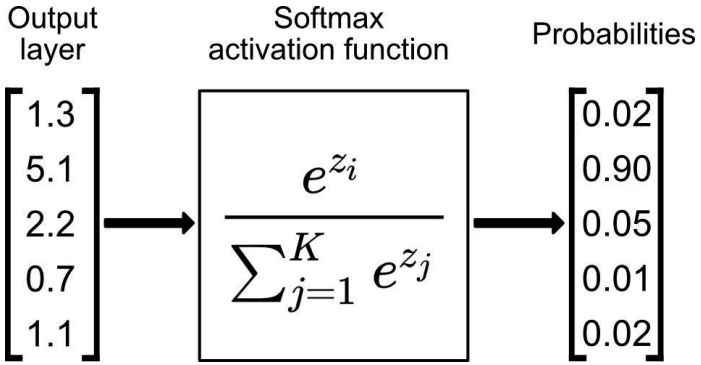

Distillation in NN
2015
Different matching
2020
DistilBERT, TinyBERT
Self distillation
Online distillation
Network augmentation
Localization distillation
2023
NEW Approach
Using teacher reasoning:
Distillation step by step, Orca
Present

Key Terms

- Teacher model
  - The original (large model) that we want to extract the knowledge from it
- Student model
  - The new model with smaller size
- Soft labels
  - The output of **teacher model** when the Temperature is larger than 1 **(T>1)**
- Soft predictions
  - The output of **student model** when the Temperature is larger than 1 **(T>1)**
- Hard predictions
  - When **regular softmax (T=1)** is used in the student model
- Hard labels
  - Ground truth label in one-hot encoding

The key insights

- Softmax function



- Key insight
  - Relative similarity between other classes and this information is useful
  - Image of 2 may be closer or similar to an example image of 3 than it is to that of 7
  - How close or similar the examples are is of much importance to understand what the network has actually learned
- Limitation
  - Giving importance to the most likely class, push **the rest to very small values**

The key insights (2): Temperature T

- Softmax function with temperature to smooth the output

$$\frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \longrightarrow q_i = \frac{exp(z_i/T)}{\sum_j exp(z_j/T)}$$

Teacher Network

$$\frac{\exp(5/1)}{\exp(5/1)+\exp(1/1)}$$

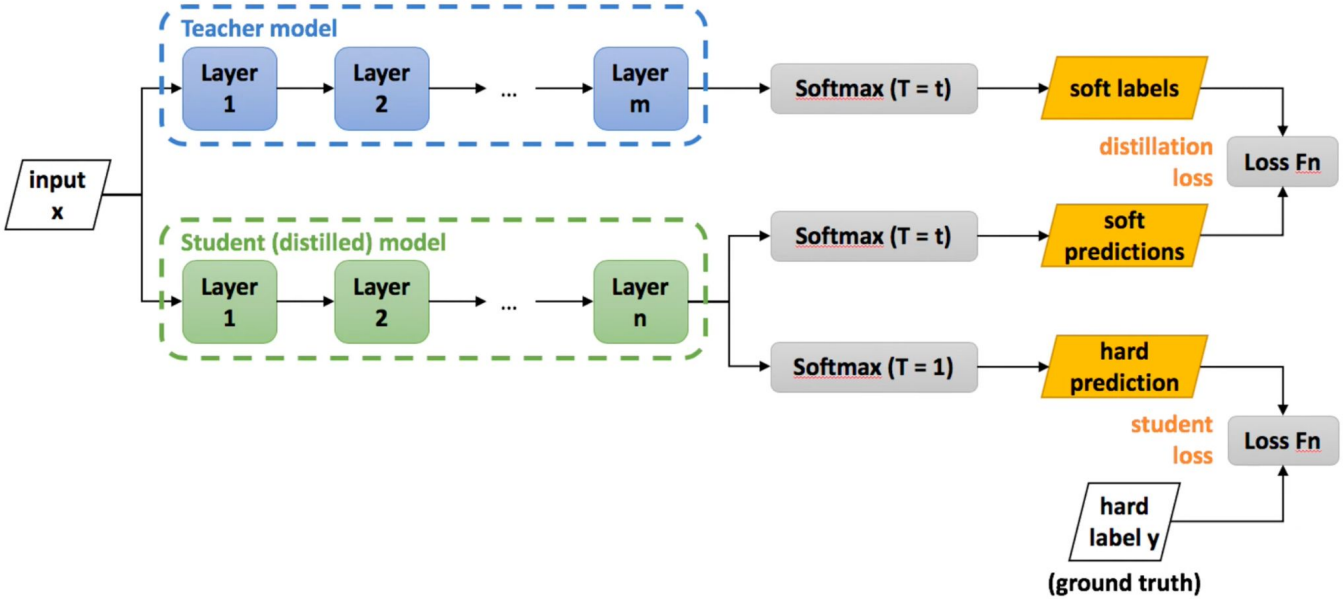
	Logits	Probabilities (T=1)	Probabilities (T=10)
Cat	5	0.982	0.599
Dog	1	0.017	0.401

$$\frac{\exp(5/10)}{\exp(5/10)+\exp(1/10)}$$

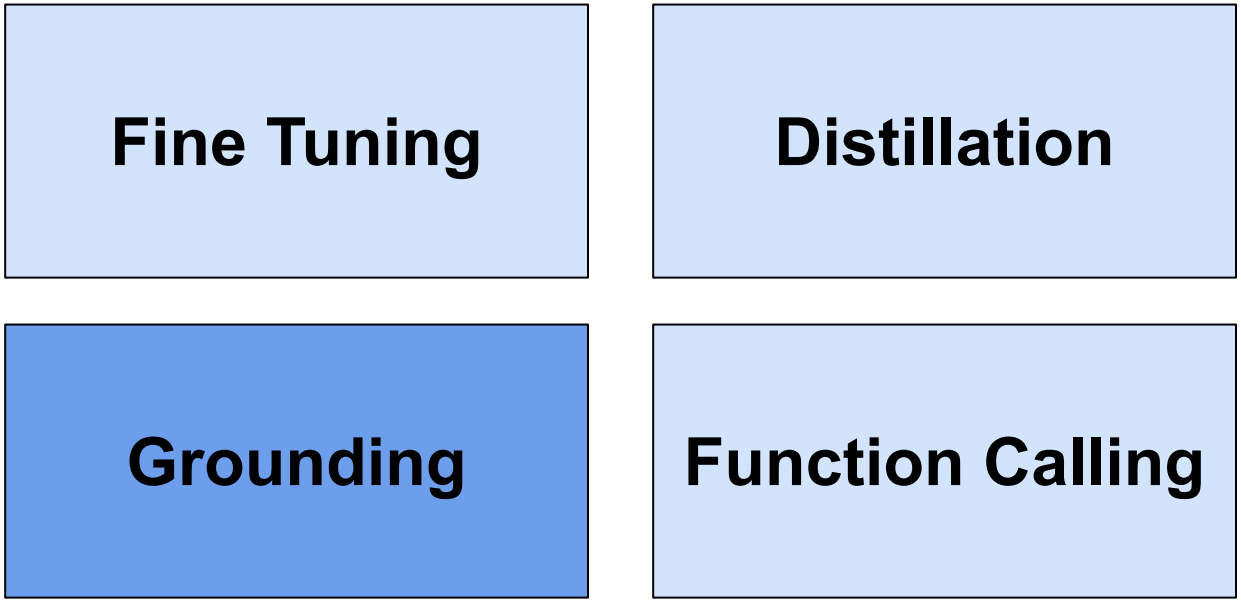
A larger temperature smooths the output probability distribution.

Architecture of distillation in NN

- Train smaller model based on the **hard labels** and the **teacher soft labels**`



Key Components of Customizations and Agent Builder



Some shortcomings of LLMs when it comes to factuality

LLMs produce output that we can check that it is factually wrong

LLMs are trained in the past

LLMs data is frozen in past. So it will not know about recent developments or facts.

LLMs hallucinate

LLMs are creative by nature. This might result in hallucinations at a significant rate.

LLMs can't cite sources

LLMs are good in reasoning, but less so on quoting which sources they used to come to a conclusion..

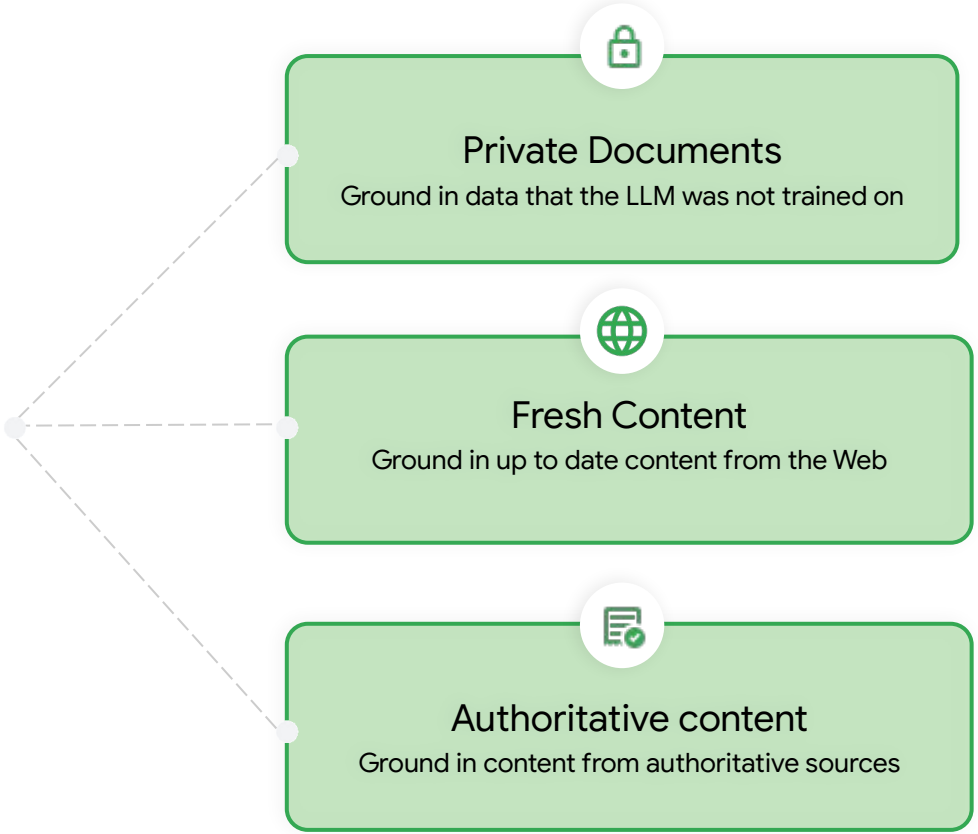
Minimizing hallucinations boils down to solving 3 problems

Right context

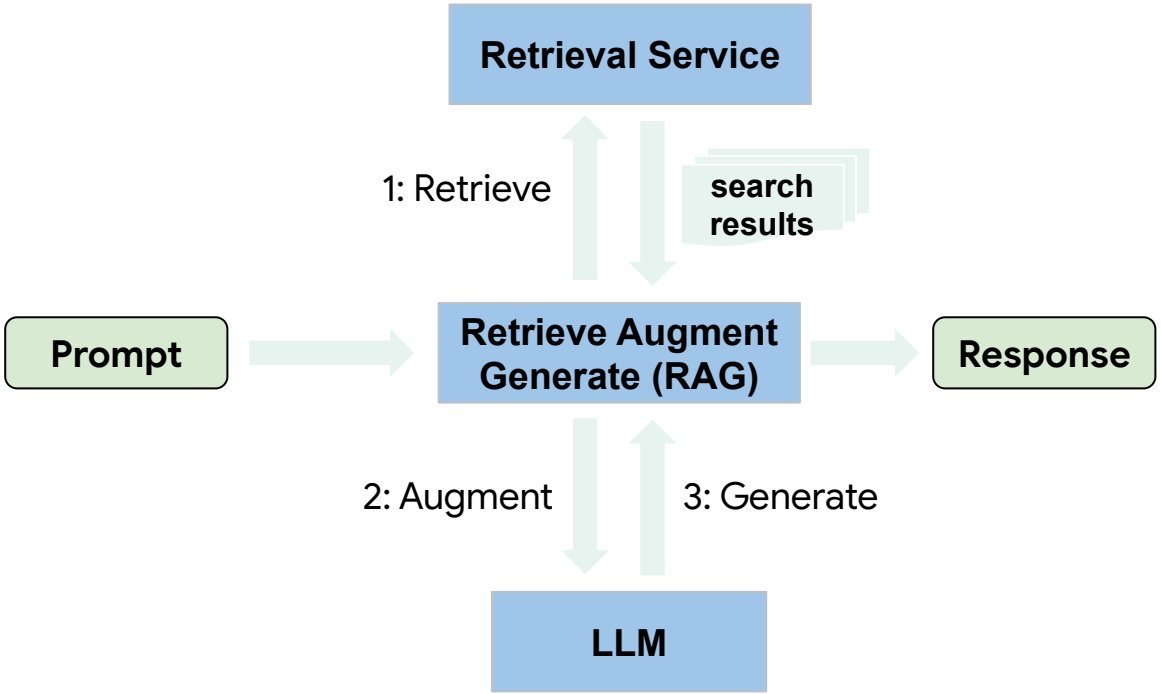
Better models

User experience

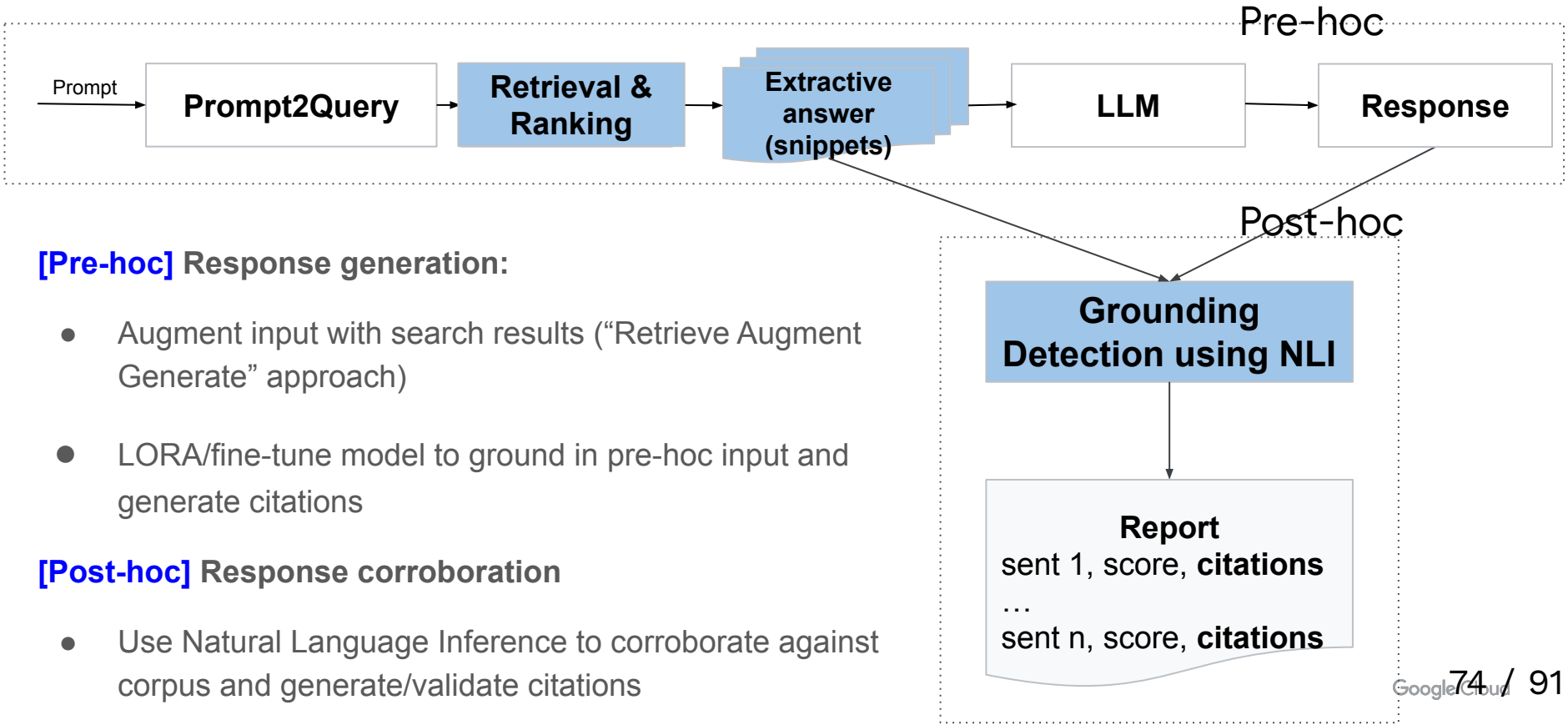
Right context



Retrieve and Generate (RAG)



Typical RAG/NLI based grounding architecture in a nutshell



[Pre-hoc] Response generation:

- Augment input with search results (“Retrieve Augment Generate” approach)
- LORA/fine-tune model to ground in pre-hoc input and generate citations

[Post-hoc] Response corroboration

- Use Natural Language Inference to corroborate against corpus and generate/validate citations

User Experience

Metric: Attribution

Félicette was a stray Parisian cat that became the first cat launched into space on 16 October 1963 as part of the French space program.

How do you know this is true?

Supporting sources: 4/7
Contradicting sources: 2/7
Sources: stuartatkinson.wordpress.com, en.wikipedia.org, ...

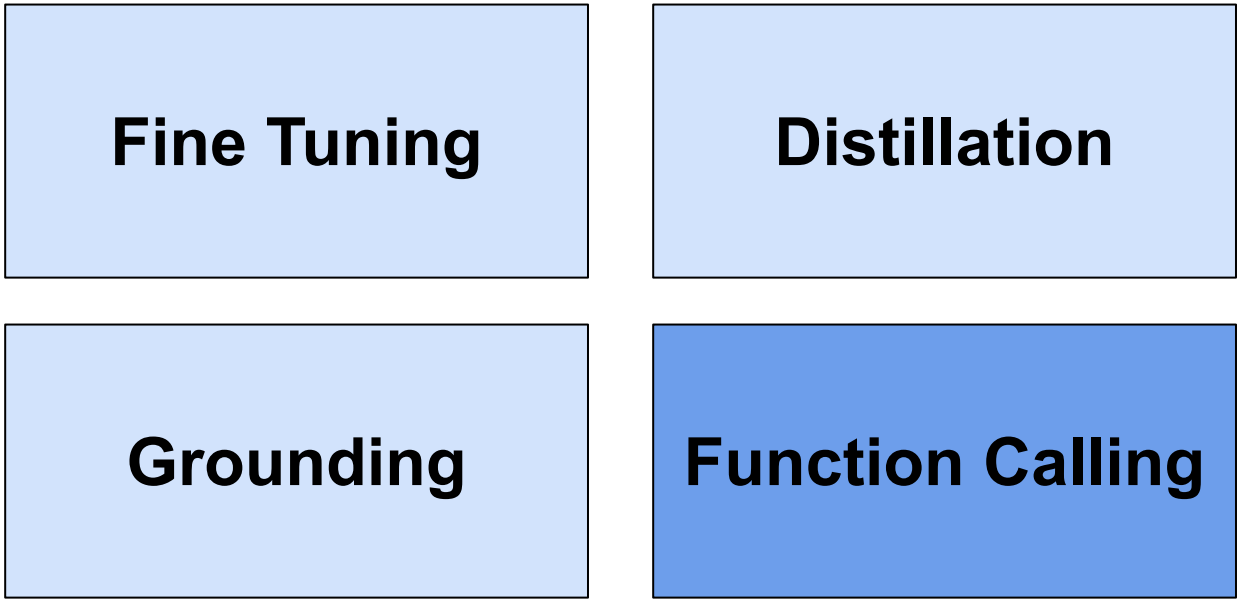
Supported
Supporting sources: 6/8
Contradicting sources: 0/8
Sources: en.wikipedia.org, english.elpais.com

Félicette was a stray Parisian cat [1,2,3] that became the first cat launched into space [4,5] on 16 October 1963 [6,7] as part of the French space program [8].

Supported
Supporting sources: 5/11
Contradicting sources: 0/11
Sources: www.reddit.com, en.wikipedia.org, ...

Contradicted
Supporting sources: 0/8
Contradicting sources: 2/8
Sources: starwalk.space, en.wikipedia.org, ...

Key Components of Customizations and Agent Builder



Have you ever received a non-answer about outdated information?

```
As of my limited knowledge up to April 2023 [...] it's
always best to check the latest rates from a reliable
source, such as a currency converter or a bank.
```

Have you ever tried to get consistent outputs?

```
{
"name": "alice",
"occupation": ["pets", "music"]
"address": false,
"email": "alice@example"
}
```

Have you ever tried to reference an external website, video, or API?

I am sorry, but as a language model, I do not have the ability to access external websites or videos. Therefore, I cannot summarize the article or the YouTube video you have provided.

Limitations of generative models

Inconsistent Output

Even with good prompting, it's **difficult to get LLMs to produce consistently structured output** for downstream implementation.

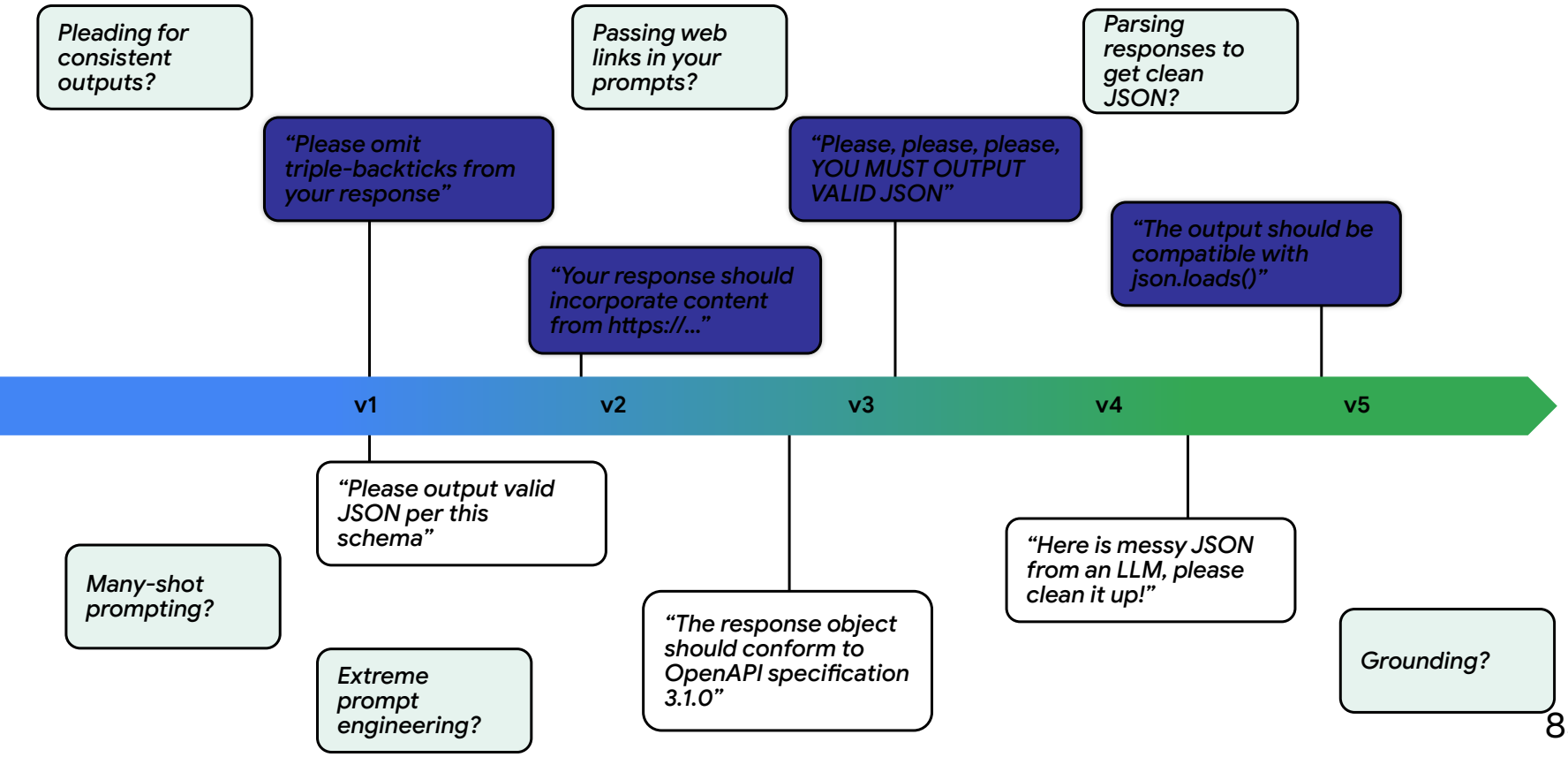
Frozen in time

Large language models (LLMs) **lack access to information after their training date**, leading to stale, inaccurate responses.

Disconnected from the world

LLMs have no inherent way of interacting with the world. This **limits their ability to take action** on behalf of the user through interfaces like APIs.

When you run into limitations with generative models, what workarounds do you try?


Pleading for consistent outputs?
Passing web links in your prompts?
Parsing responses to get clean JSON?
“Please omit triple-backticks from your response”
“Please, please, please, YOU MUST OUTPUT VALID JSON”
“The output should be compatible with json.loads()”
“Your response should incorporate content from https://...”
v1
v2
v3
v4
v5
“Please output valid JSON per this schema”
“Here is messy JSON from an LLM, please clean it up!”
Many-shot prompting?
“The response object should conform to OpenAPI specification 3.1.0”
Grounding?
Extreme prompt engineering?

But they probably didn't work as expected.

How can we solve this problem?

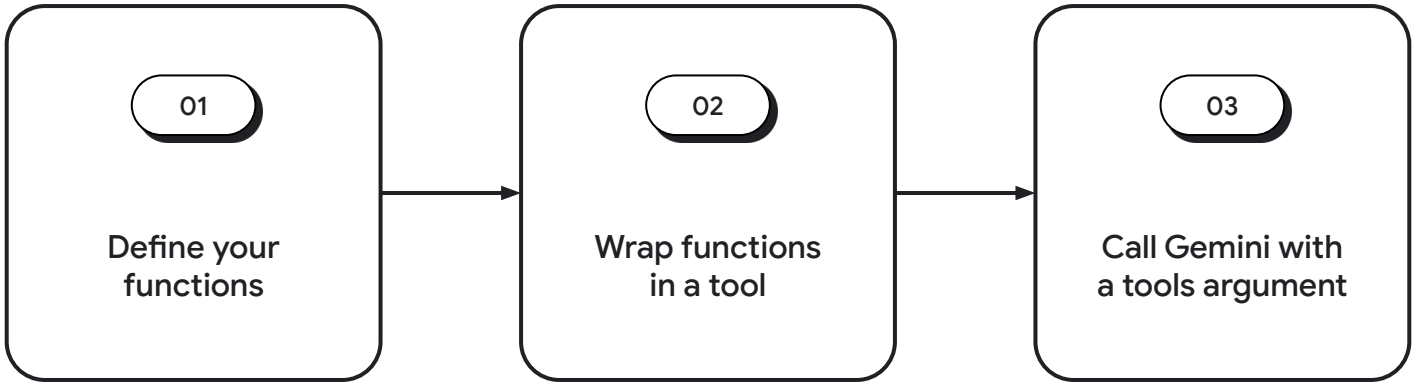
Wait, what is Function Calling?

Does it call functions?

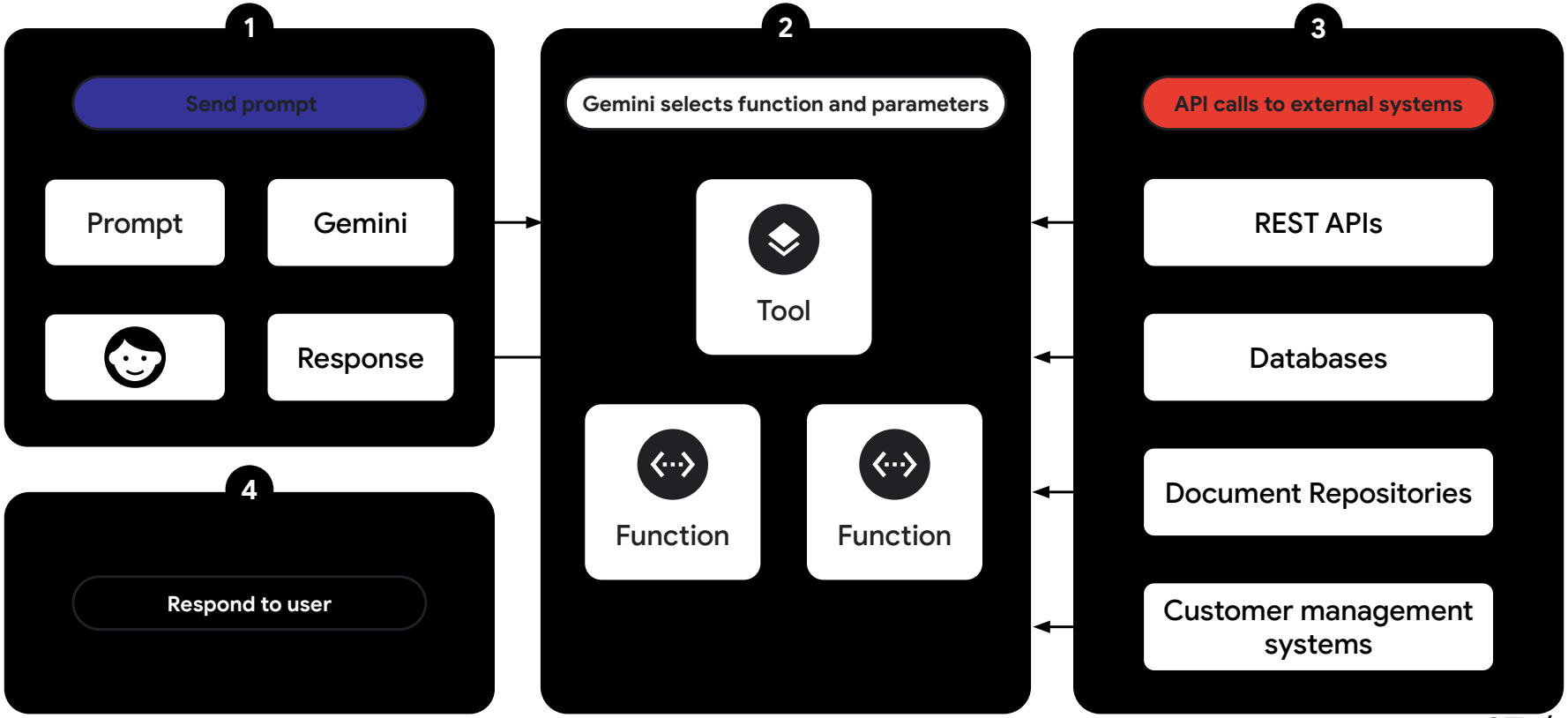
Sort of!

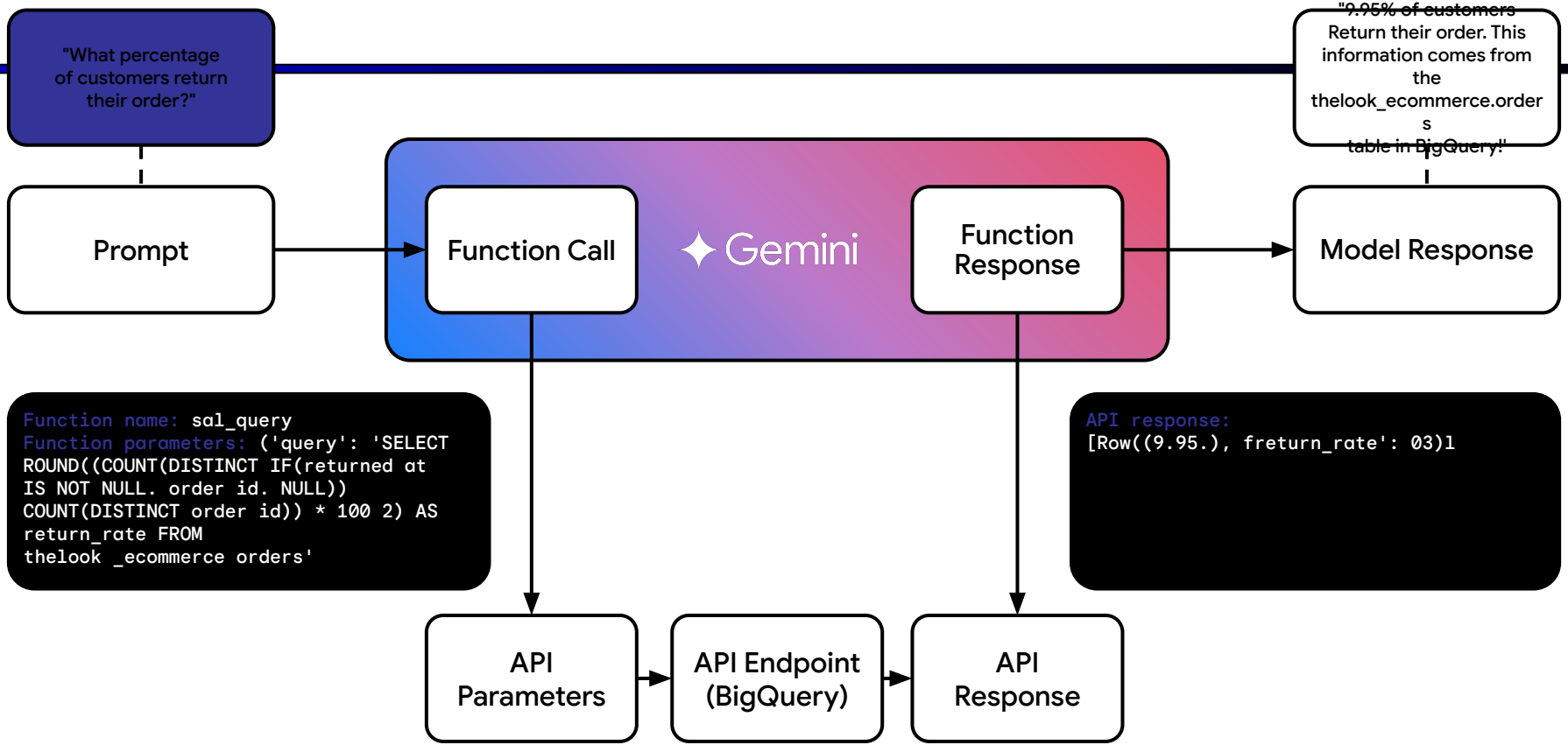
Let’s dig in...

How Function Calling works



A day in the life of a Function Call

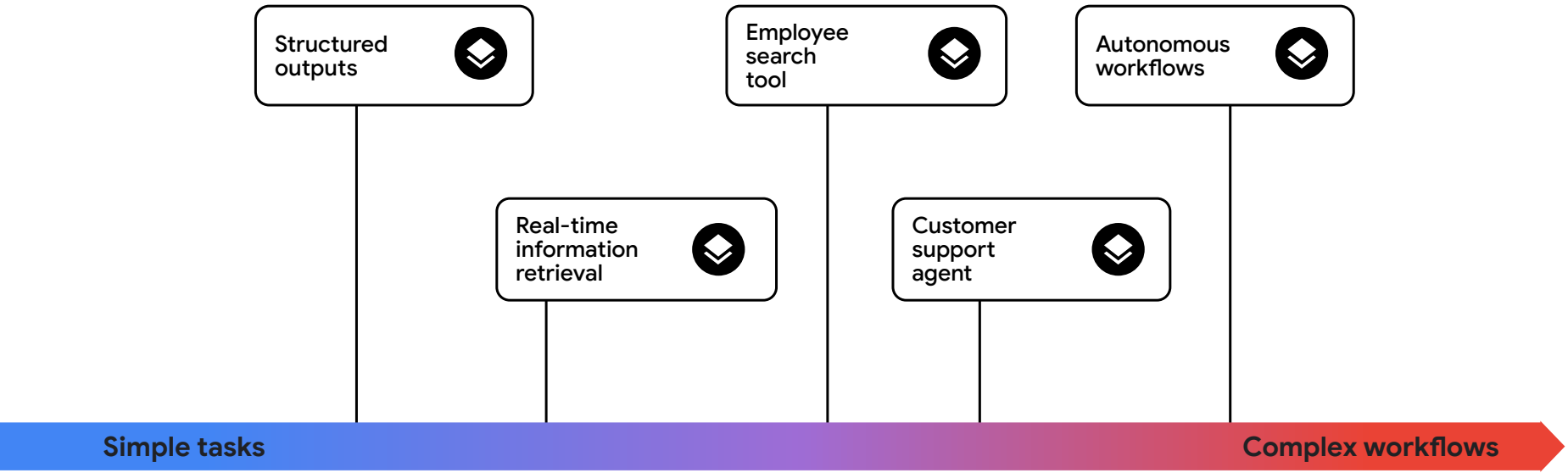



"What percentage of customers return their order?"
Prompt
Function Call
Gemini
Function Response
"9.95% of customers Return their order. This information comes from the thelook_ecommerce.orders table in BigQuery."
Model Response
Function name: sal_query
Function parameters: ('query': 'SELECT ROUND((COUNT(DISTINCT IF(returned at IS NOT NULL. order id. NULL)) COUNT(DISTINCT order id)) * 100 2) AS return_rate FROM thelook _ecommerce orders'
API response:
[Row((9.95.), freturn_rate': 03)1
API Parameters
API Endpoint (BigQuery)
API Response

What can you build with Function Calling?

Lots of things!

Function calling is all about developer control and flexibility



Finance

Fetch real-time financial and currency exchange information

Business

Read and write to documents and spreadsheets in Google Drive

Databases

Perform real-time queries on datasets in BigQuery

Documents

Search and summarize across thousands of documents

Travel

Fetch live flight and hotel information from travel systems

Support

Retrieve messages from customer support ticketing systems

Customers

Search records in customer management systems

Inventory

Make live queries to product inventory systems to check stock