Superwise tensorops

To train or not to train your LLM

How to strike the right balance

Oren Razon, CO-Founder & CEO @ Superwise | oren.razon@superwise.ai | linkedin/oren-razon

Gad Benram, Founder & CTO @ TensorOps | gad@tensorops.ai | linkedin/gad-benram

About us

Gad Benram

Founder & CTO @ **TensorOps**

Oren Razon

CO-Founder & CEO @ **Superwise**

Jose Bastos

AI Engineer @ **TensorOps**

Superwise

Model observability

built for scale

We empower data science, ML engineering, and operational teams with visibility and control to **scale AI activities**

Klarna. King monday.com

riskified Fundbox

tensorops

The AI Experts

We simply help machines learn

We build end-to-end AI solutions for businesses; Specializing in time-series forecasting, search, OpenAI and Google Cloud.

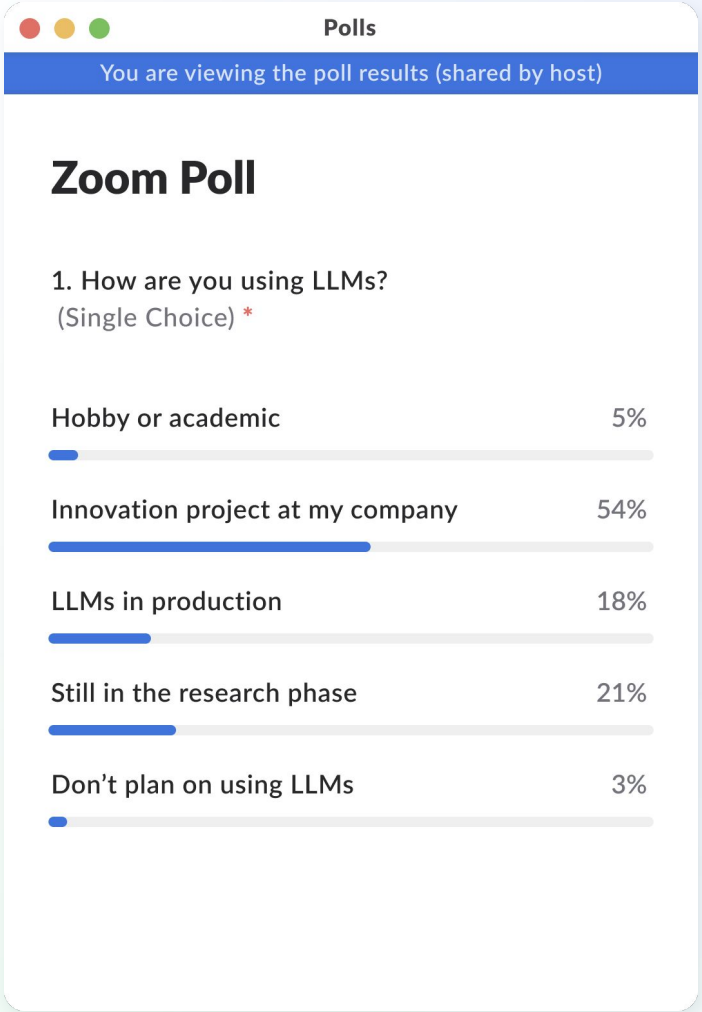
Panaya VERSATILE N

onebeat Qwak Zesty

Agenda

- Typical use cases
- LLM types
- To train or not to train?
- Pros & cons
- Types of training/tuning
- Monitoring LLMs
- Takeaways & going beyond tuning

How are you using LLMs?



Typical LLM use cases

Tasks

- Chatbots
- Embedding
- Organizational knowledge
- Productivity
- Code completion
- Marketing
- Raising funding in 2023

Real-world

- Jasper
- Semantic search
- Google gen app builder
- Notion
- Co-Pilot
- SlidesAI.io
- Mistral AI

Types of LLMs

API

No or limited visibility into code and data, but easier to deploy out of the box

Bard

OpenAI

Open source

Offer the most transparency and flexibility but require time and expertise to set up

falcon

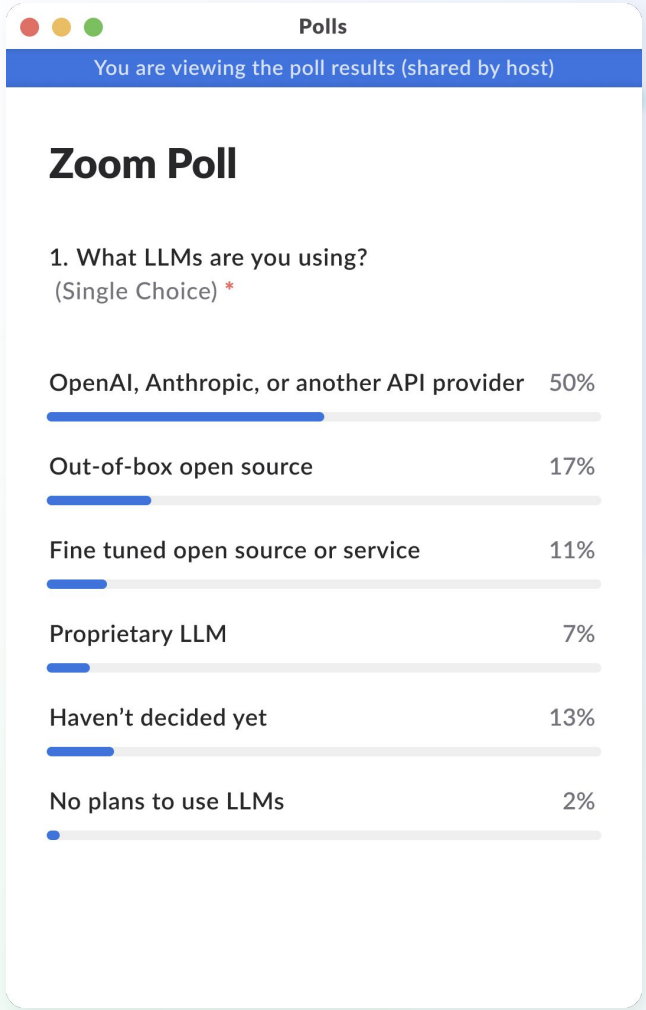
LLaMA

Proprietary

Built in-house from scratch for the organization

Bloomberg

What LLMs are you using?



Out of the box

vs

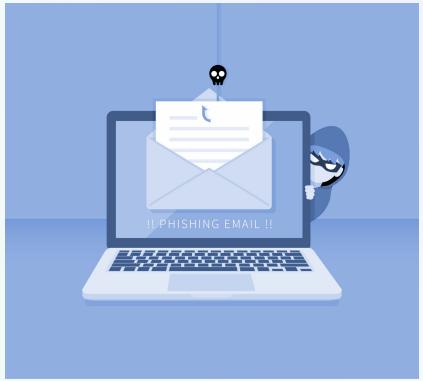
Tuned

LLMs in the wild

Real world use-case

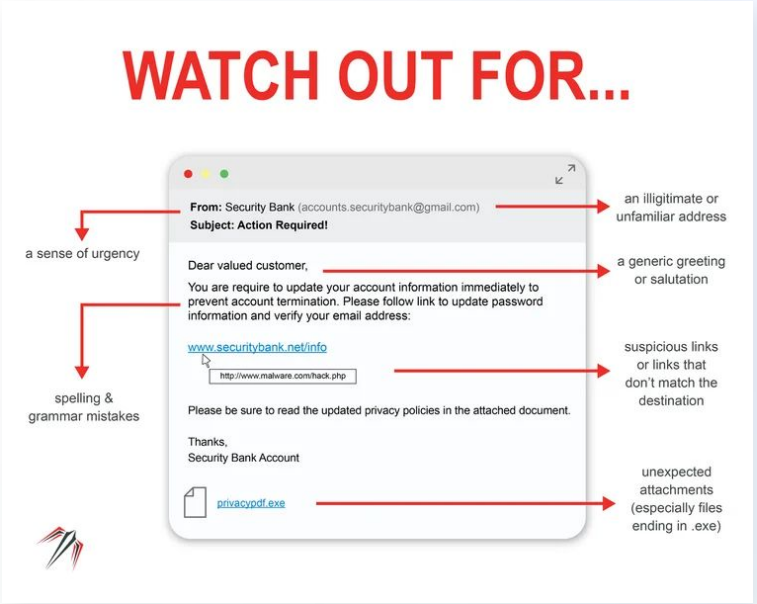
Phishing email detector

- MailProtect is a (made up) company that offers email protection service for large organizations
- They have a solution for detecting and blocking phishing emails



LLMs in the wild

Cyber LLM use-cases



Classical ML

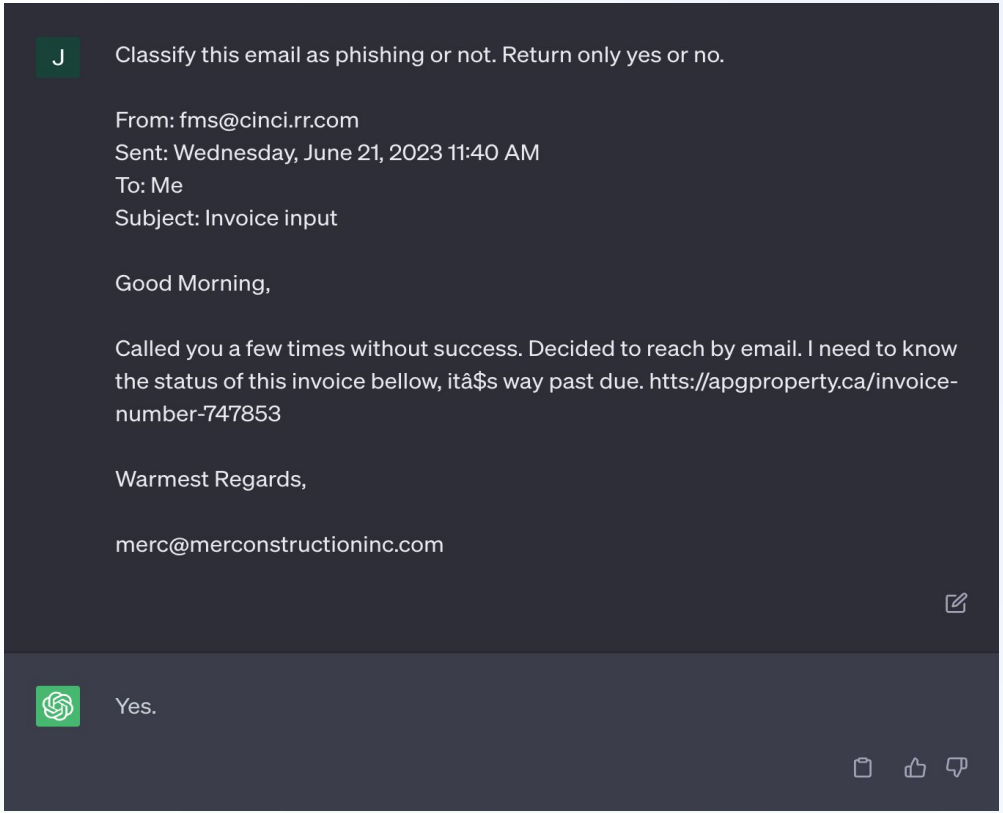
Deals with anomaly detection and tabular data.

LLMs improve textual pipelines

Improve textual pipelines that rely on classification, segmentation, generation, interpretation.

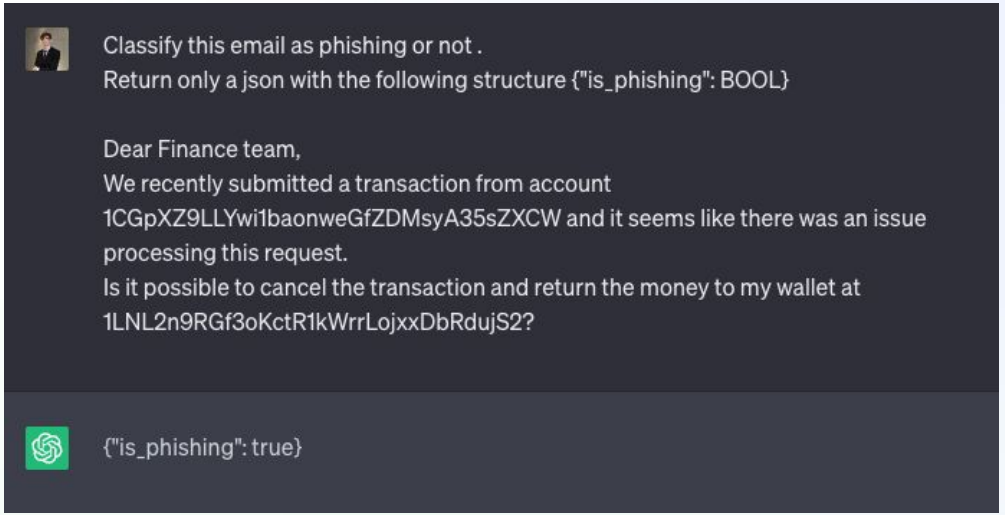
LLMs in the wild

ChatGPT works!



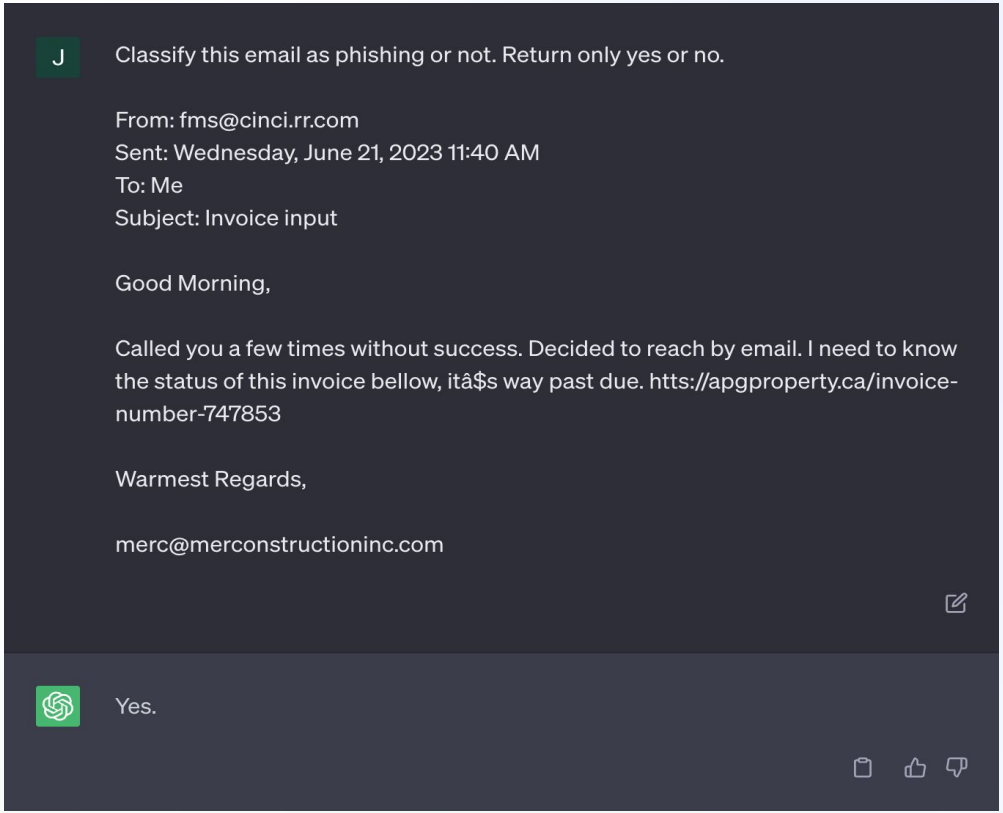
LLMs in the wild

ChatGPT works?



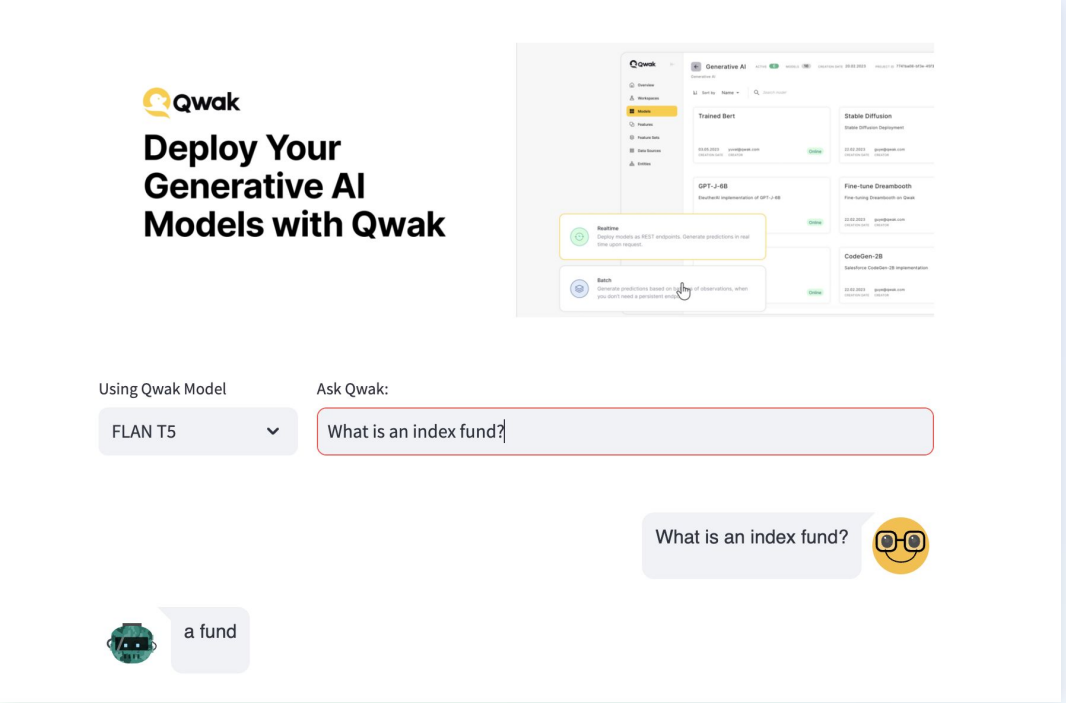
Tuning in the wild

Let's look at an example from a financial use case



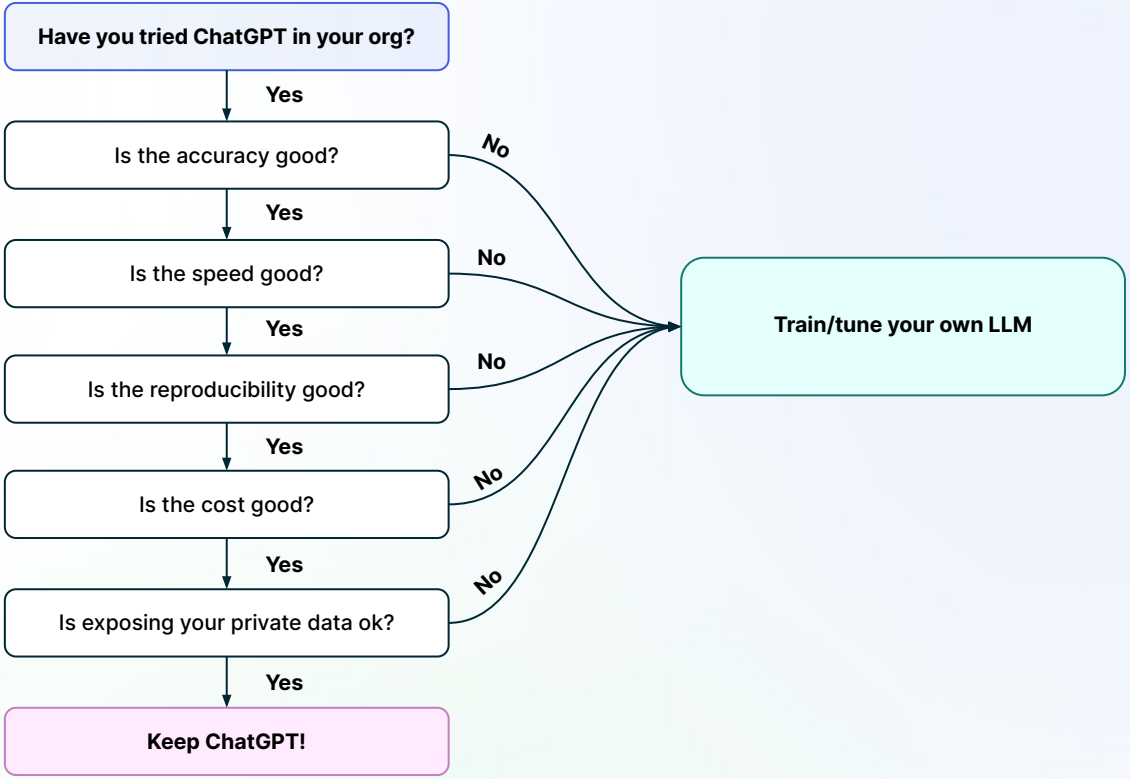
Demo:

Does tuning really work?



Do you even need tuning?

Have you taken ChatGPT to production?



Pros and cons of training LLMs

Pros

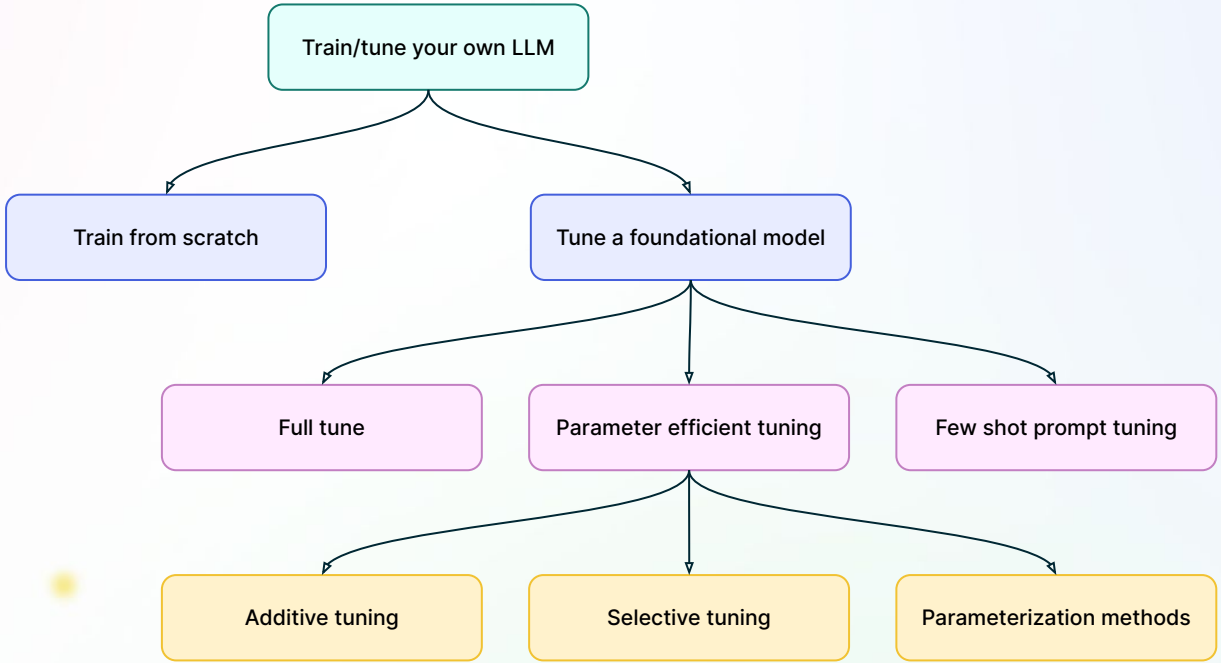
- Better accuracy
- Potential to reduce inference cost
- Control data
- Control infrastructure

Cons

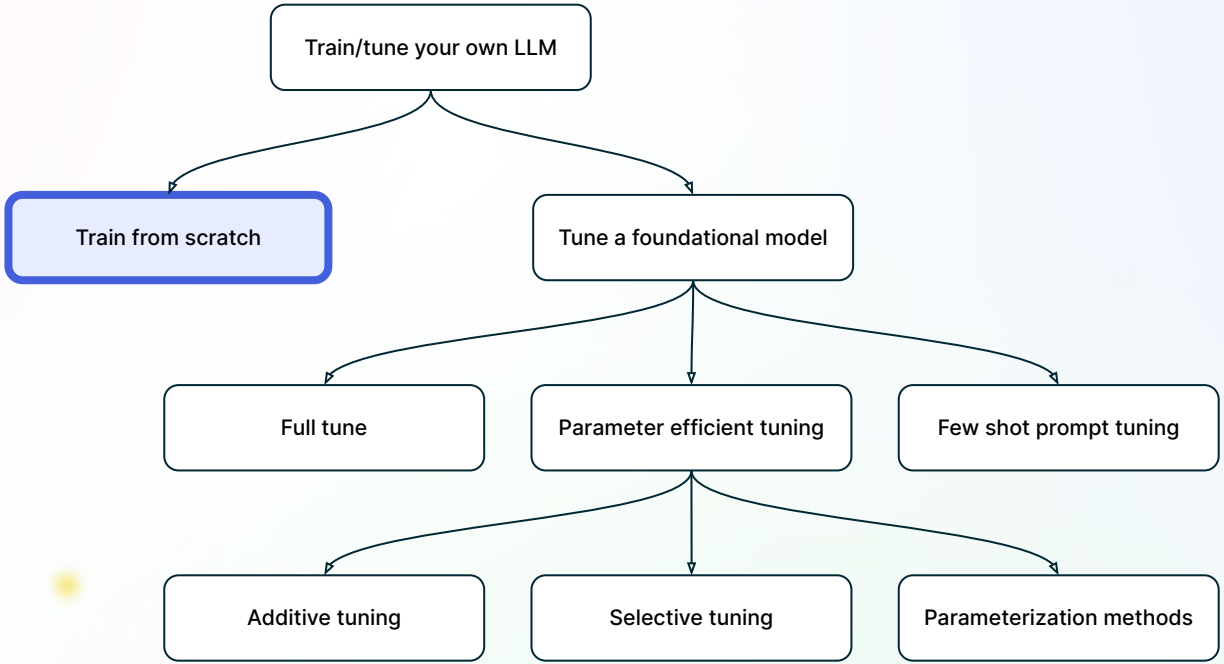
- Data leakage
- Engineering complexity
- Reduced performance
- Cost

What does “training an LLM” even mean?

To train or not to train your LLM?



Train from scratch



Train from scratch

Need huge amounts of data and compute power and takes some months to do it

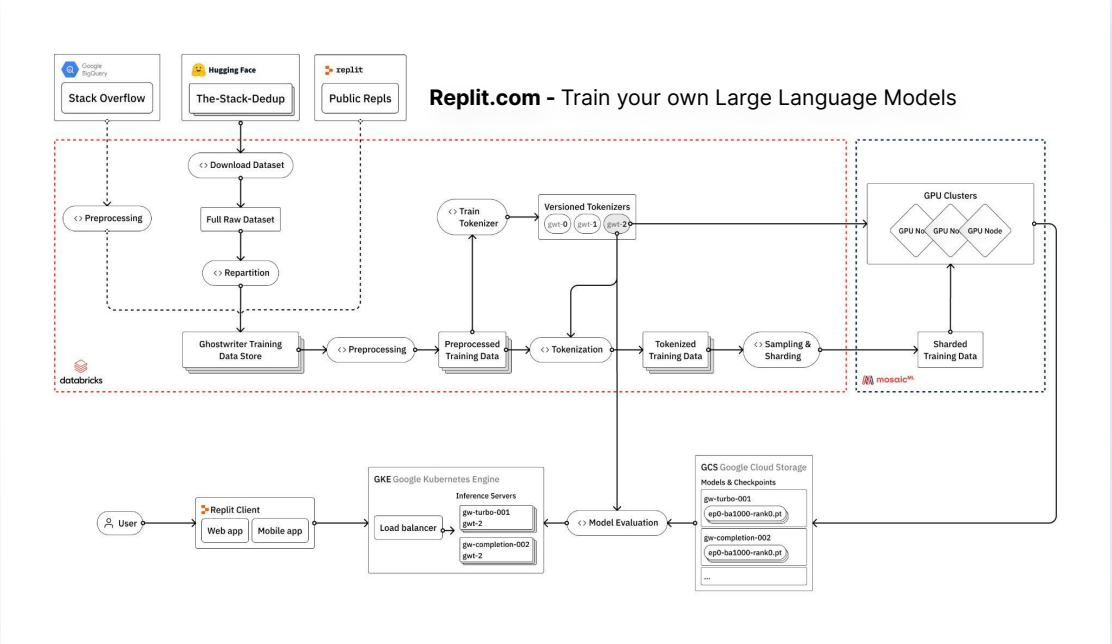
Pros:

→ Best fit (?)
→ Full ownership (model + data)

Cons:

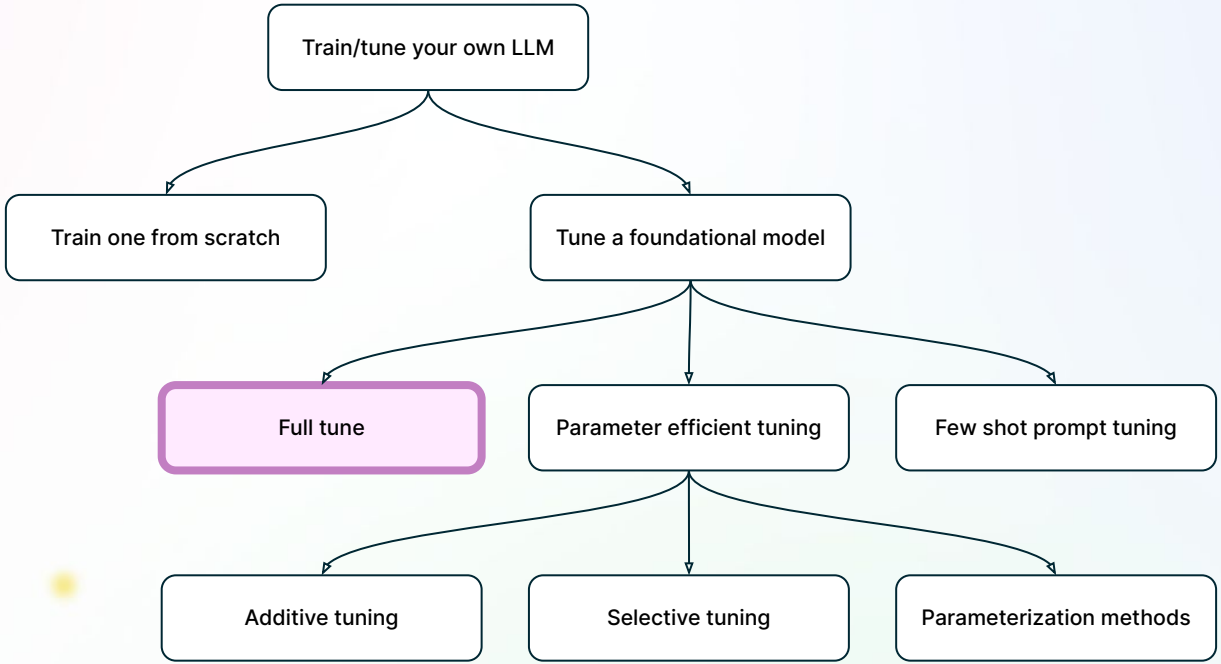
→ **Huge** amounts of data
→ **Huge** amounts of compute power
→ Takes **months**

Huge costs!



Superwise tensorops

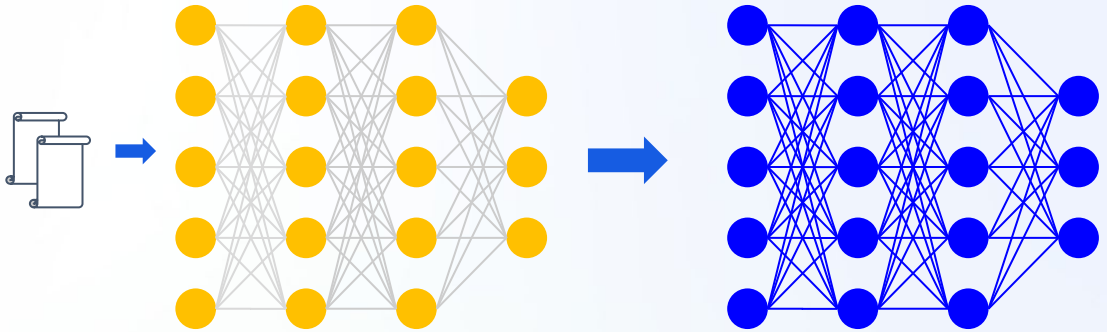
Full tune


Train/tune your own LLM
Train one from scratch
Tune a foundational model
Full tune
Parameter efficient tuning
Few shot prompt tuning
Additive tuning
Selective tuning
Parameterization methods

Full tune

Fully tune an existing LLM

Customize an existing model with your specific use cases. Gradually change the model by feeding it examples of the expected behavior



Pros:

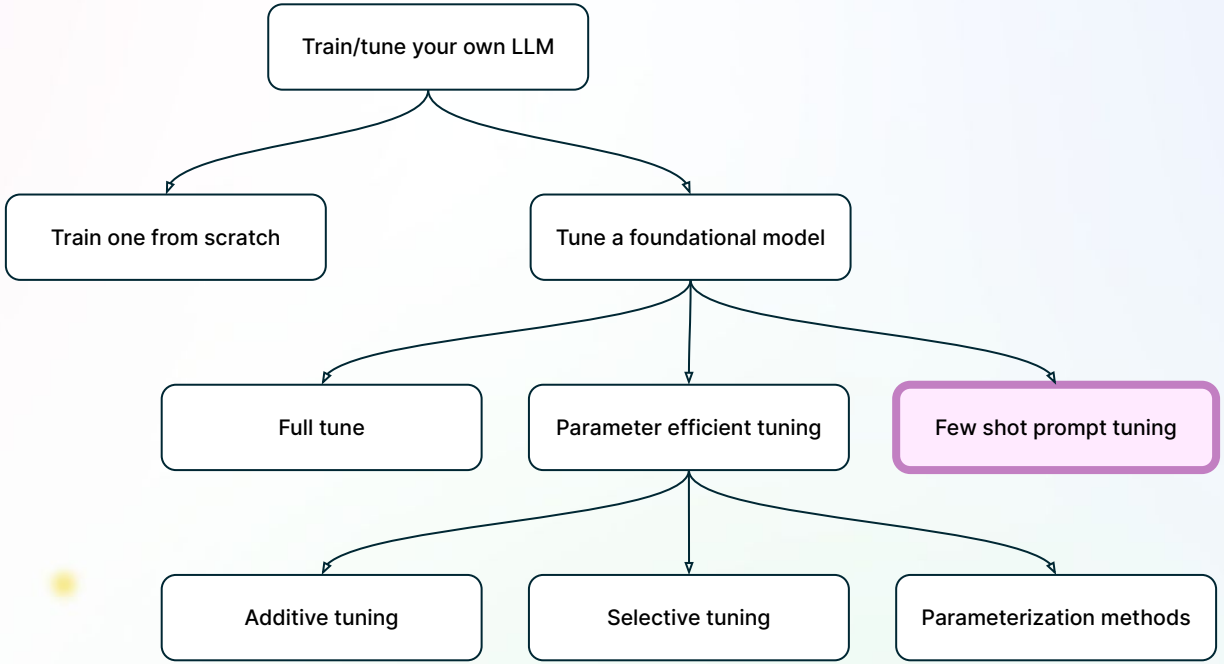
→ Small downgrade in accuracy*
→ Almost full ownership (model + tuning data)

Cons:

→ **Moderate** amounts of data
→ **Moderate** amounts of compute power
→ Takes **days/weeks**

* For a considerable about of tuning data

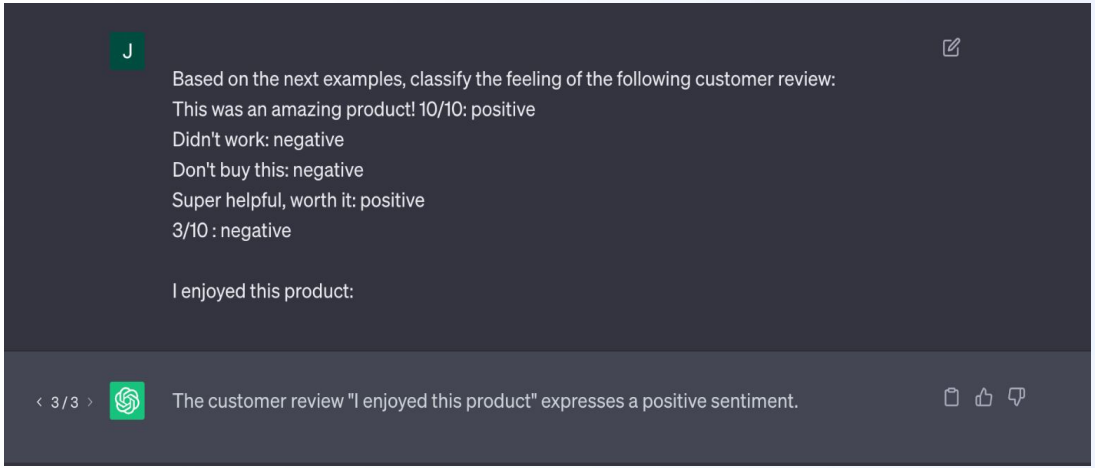
Few shot prompt tuning



Few shot prompt tuning

Naive prompt tuning or few shot tuning

Providing the models with examples within the prompt context



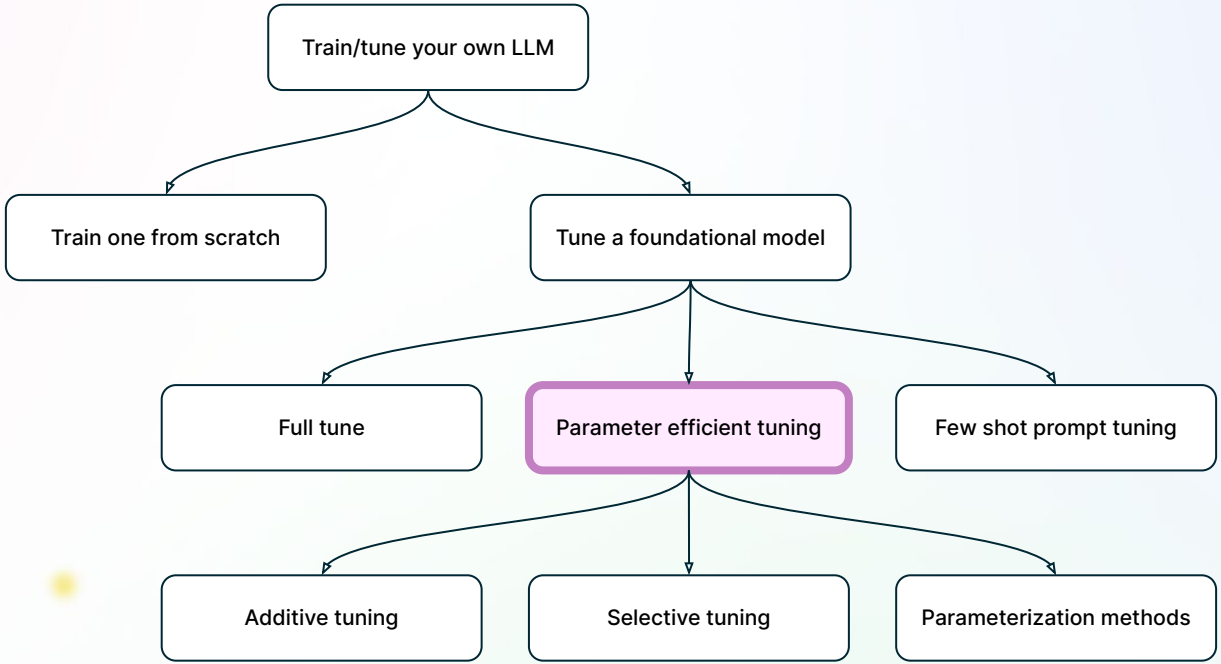
Pros:

→ **No data needed**
→ **No extra compute** (just inference)
→ **Instant**

Cons:

→ Bad performance (non robust + non scalable)
→ Extensive prompt engineering

Parameter efficient tuning

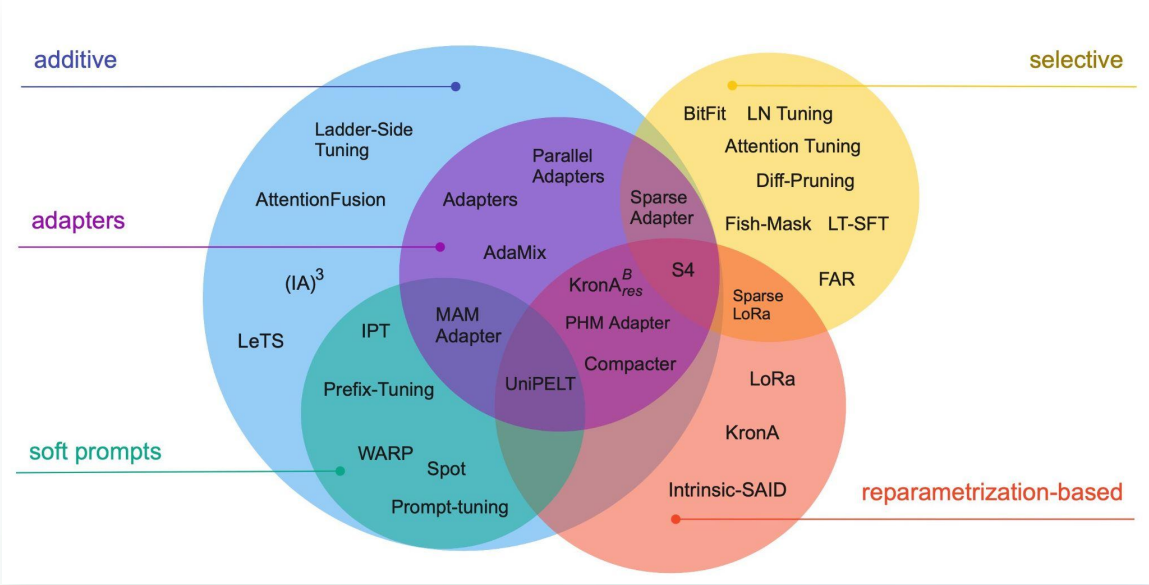


Parameter Efficient Fine Tuning (PEFT)

Make more with less!
Use less data/compute power achieve comparable performance

⬇

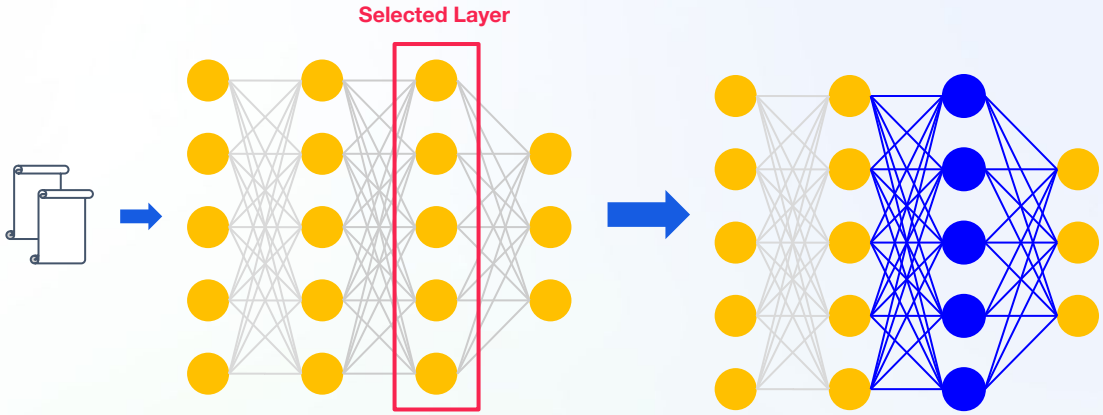
Reduce costs!



Selective tuning

Selective methods

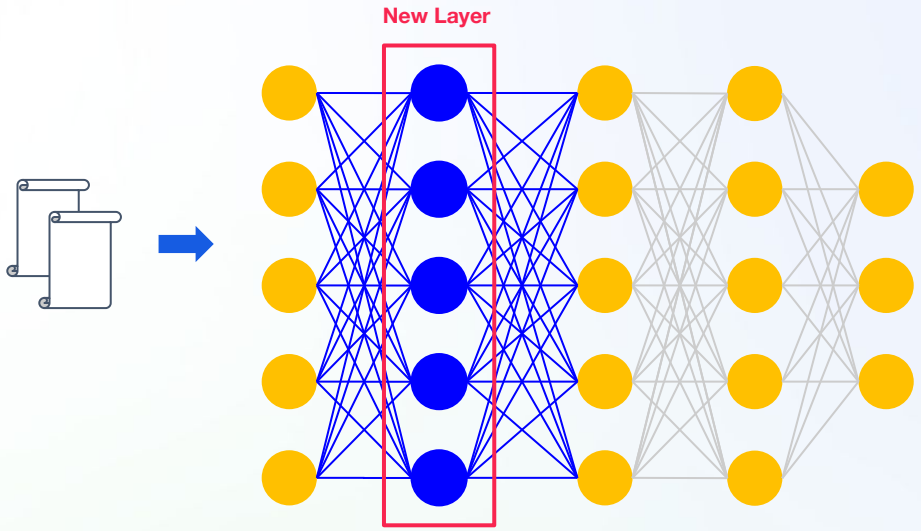
Re-train only a few layers/parameters, keeping all other parameters fixed



Additive tuning

Additive methods

Add a few new parameters and train them from scratch, keeping all other parameters fixed



(original model only had 4 layers)

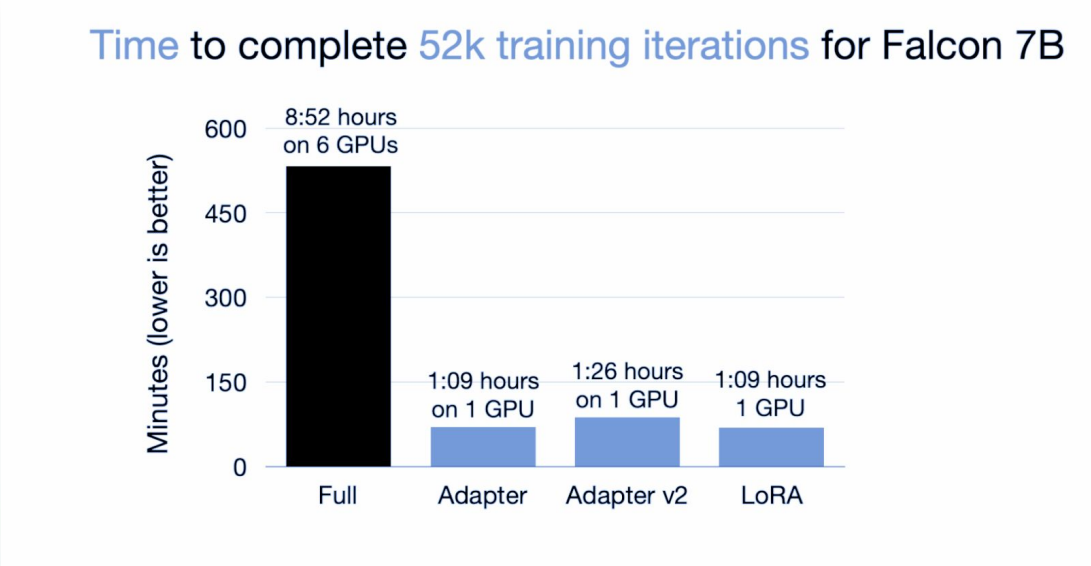
Parameter-ization tuning

Parameterization methods
Decrease foundational model size before tuning

→ Low rank approximation/adaptation
→ Lowering precision

Parameter efficient tuning

Parameter efficient fine tuning (PEFT)



Lightning.ai - Fine tuning Falcon LLMs more Efficiently with LoRA and adapters

Parameter efficient tuning

PEFT Pros:

→ Significantly reduces compute power

→ Significantly reduces training time **(hours)**

→ **Reduce costs!**

→ **Comparable accuracy**

PEFT Cons:

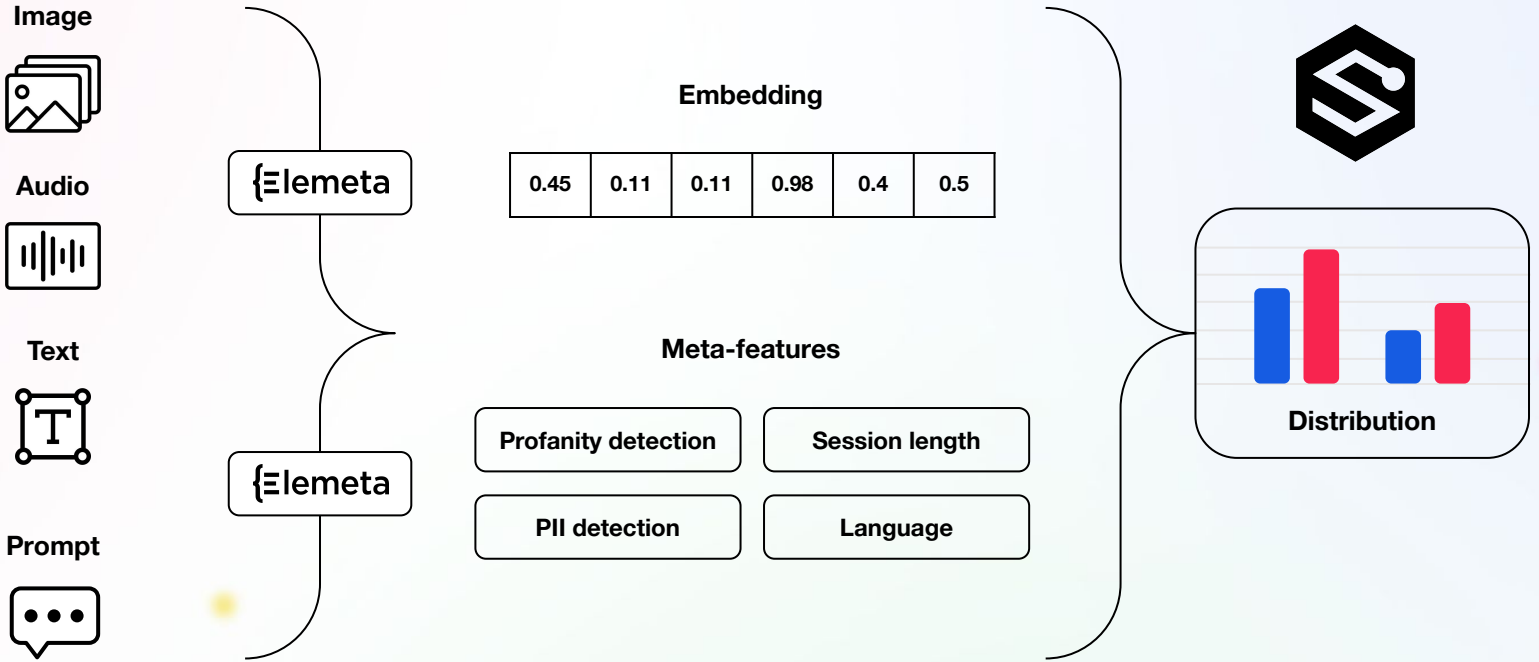
→ Lacks extensive accuracy evaluation for every method

→ Small downgrade in accuracy

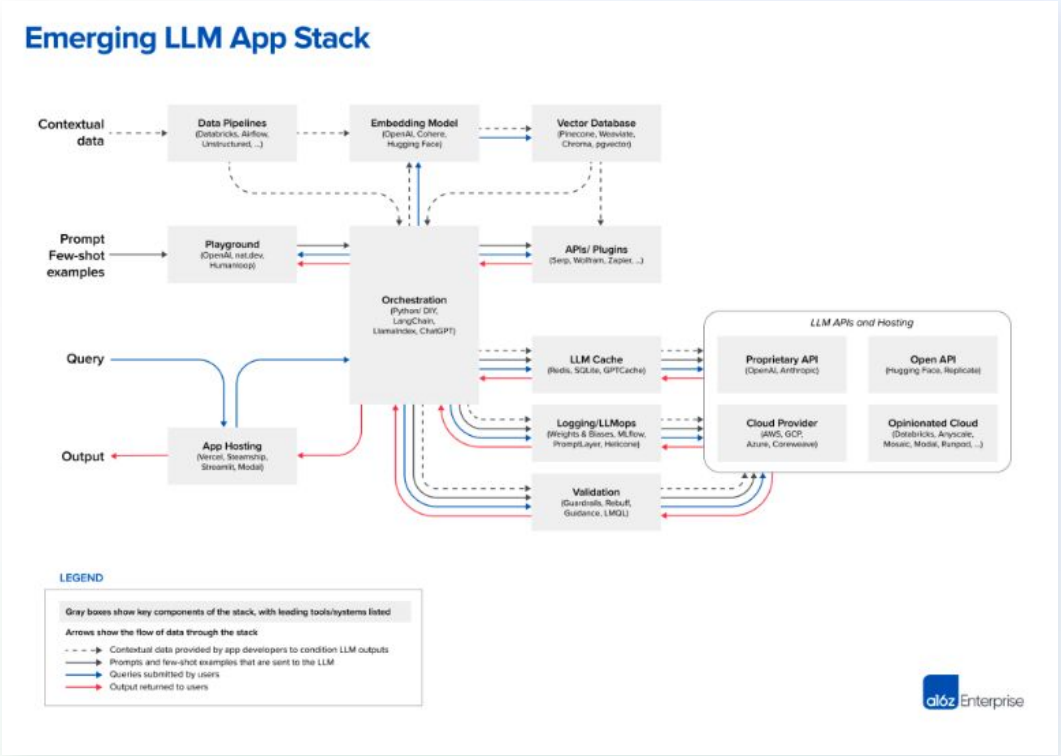
Monitoring LLMs

Integrity	• Prompts not understood • Readability match	• Sentiment match • Language match	• User feedback • Cutoff responses
Drift	• Changes in language • Changes in vocabulary	• Topic drift • Prompts out of training data	
Governance	• Bias and profanity indicators • Governance & compliance rules	• Personal information • Privacy preservation	
Hallucinations	• Response outliers • Similarity scoring	• Reference (URL) validation	
Attacks	• Adversarial attempts to extract data • Bypass safety controls	• Prompt injection • Prompt leakage	

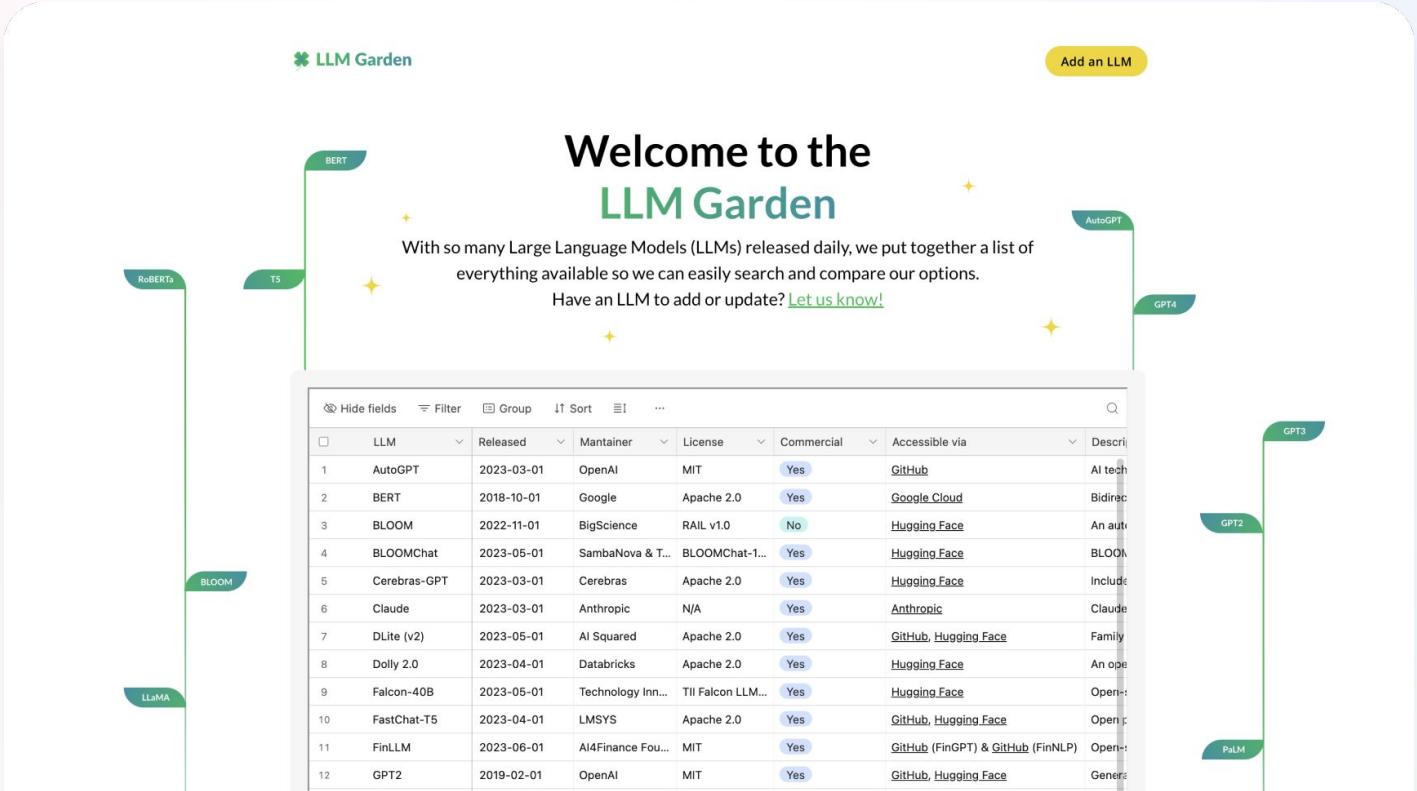
Monitoring LLMs with Elemeta



Beyond tuning



LLM Garden (https://llm.garden/)



Q&A

Oren Razon, CO-Founder & CEO @ Superwise | oren.razon@superwise.ai | linkedin/oren-razon

Gad Benram, Founder & CTO @ TensorOps | gad@tensorops.ai | linkedin/gad-benram