# RAG and the Fine Tuners

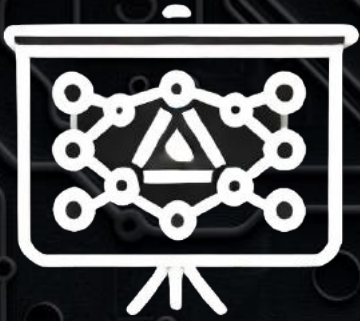## Getting the Band Together

**UNPARSED**

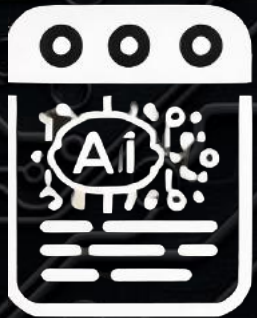LONDON, UK / ONLINE

AI Assisted

# This Presentation and Code

Presentation:
bit.ly/unparsed-finetuning

Code:
bit.ly/unparsed-finetuning-code

# Who am I

**Roger Kibbe**

- Head of Conversational AI Developer Relations, *Samsung Research America*

- Startup Advisor: *Ollang, LiftLab, AgreeWe*

- Entrepreneur
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- Dad - two teen daughters
- San Francisco Native
- UC Berkeley Graduate – Go Bears!

www.linkedin.com/in/rkibbe/

# Roger's Presentation and views
# not my employer's

# State of GenAI



**AI Demo Excitement**          **Using ChatGPT**          **Trying to go Live**

# The Questions

*I use prompt engineering and RAG. What about fine-tuning? When should I use it? What is it? How do I use it?*

A Battle?

RAG & Fine Tuning

Complimentary

# Cooking Analogy

- *Base LLM Training:* Basic culinary school. Broad foundational learning
- *RAG:* Using cookbooks. Integrating external resources
- *Fine-tuning:* Specialized cooking techniques. Mastering advanced techniques
- *Prompt Engineering:* Creating menus/adapting recipes to events or dietary needs

# RAG ≠ Easy

## Agentive Corrective RAG Agent – LangChain & AI Jason
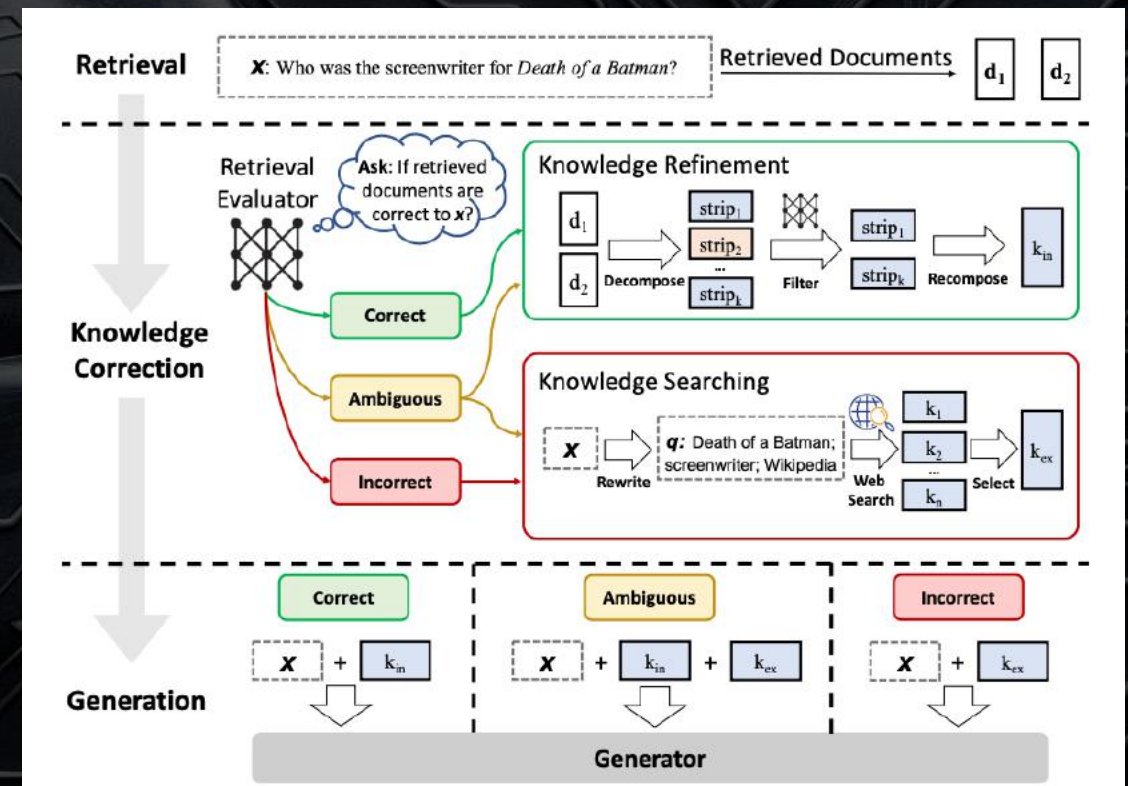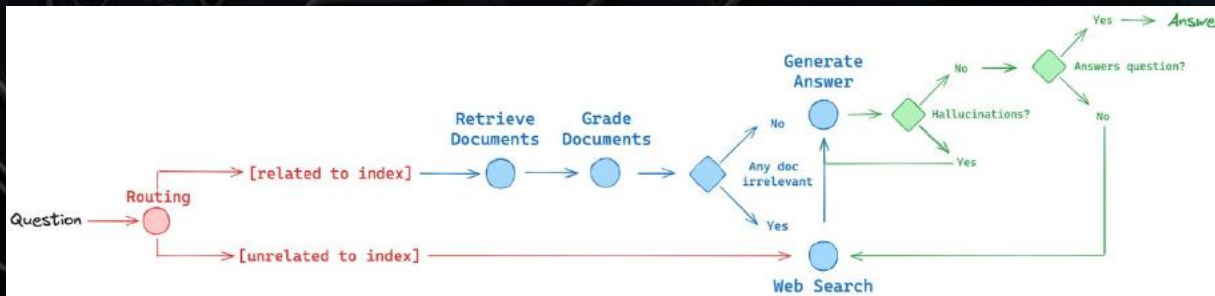
# When to Fine Tune

# Remember

## Fine-Tuning
### *[fahyn too-ning] noun*

a process in machine learning where a pre-trained model is further trained on a specific dataset to enhance its capabilities in a particular domain or task, often resulting in improved accuracy and relevance of outputs.

Data
Knowledge*

* except a full fine-tuning, which is data and capabilities. A PEFT fine-tune is capabilities

# Common Use Cases

*Text Classification:* topic classification, sentiment analysis

*Document Parsing:* Extract info from complex document formats

*Coding Style/Languages:* code style guidelines, proprietary languages

*Sentiment Analysis:* Analyze sentiment in text

*Style Copying:* Mimic style/brand voice from documents

*Named Entity Recognition:* Extract entities, e.g., names, locations, dates, etc.

# Industries/Verticals

**Law**
- Contract analysis
- Compliance
- Classification

**Finance/Investing**
- Sentiment Analysis
- Document parsing
- Compliance

**Medicine**
- Classification
- Research summarization
- Patient communication

**Retail**
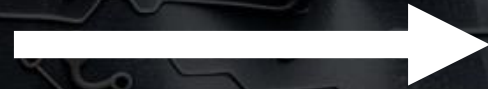- Customer Service
- Personalization
- Marketing

My Prompt doesn't work
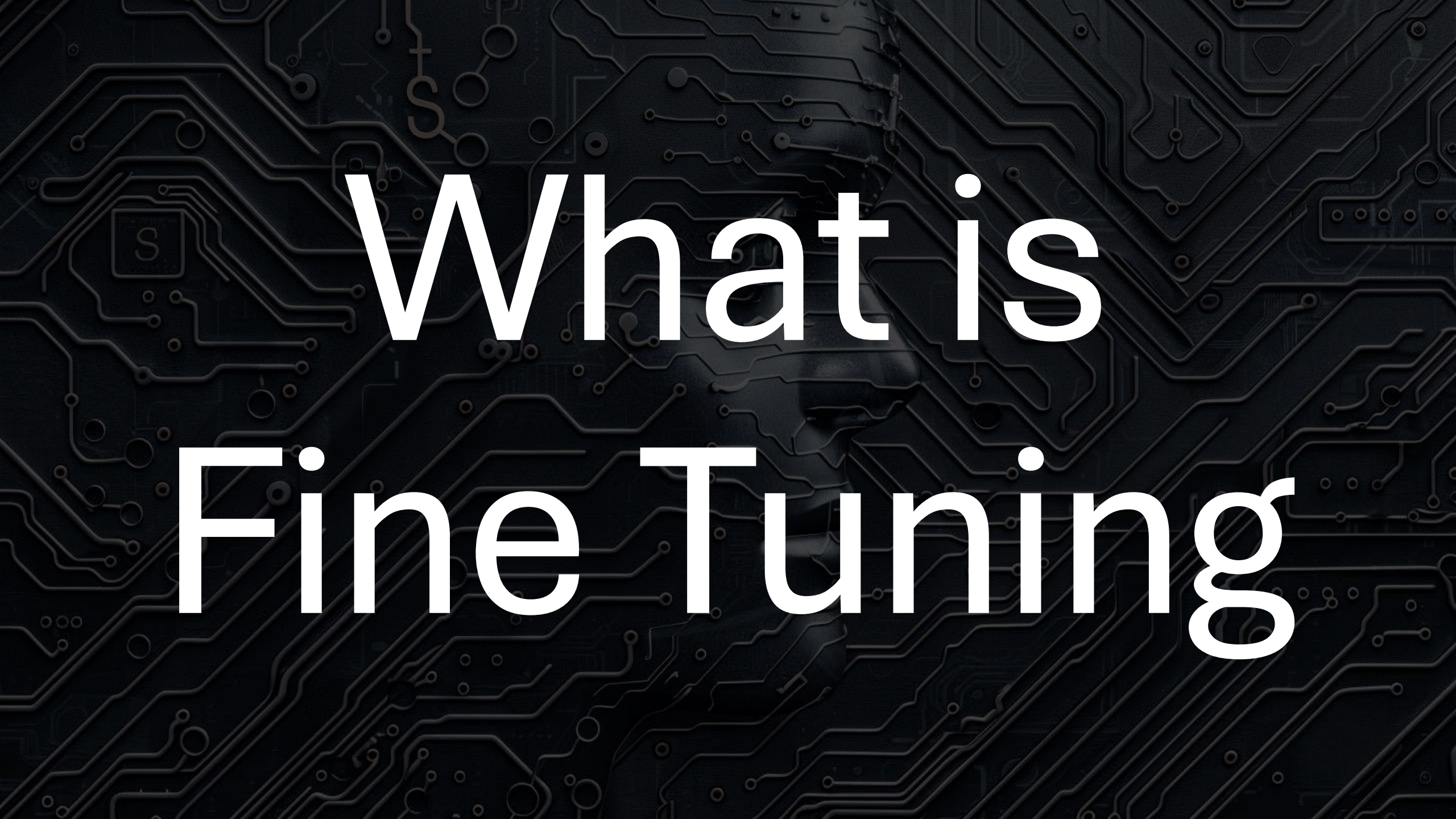
# Fine-Tuned Pricing Comparison

| Fine Tuned Model | Input – 1M Tokens | Output – 1M Tokens | Reference Model |
|---|---|---|---|
| GPT 3.5 | $3.00<br>40% Cheaper | $6.00<br>60% Cheaper | GPT-4o<br>Input: $5.00<br>Output: $15.00 |
| Mistral 7B | $0.75<br>80% Cheaper | $0.75<br>95% Cheaper | Mistral Large<br>Input: $4.00<br>Output: $12.00 |
| Gemini Pro 1.0 | $0.50<br>85% Cheaper | $1.50<br>85% Cheaper | Gemini 1.5 Pro<br>Input: $3.50<br>Output: $10.50 |

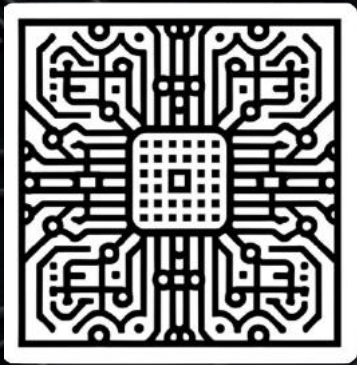But: Most fine-tuned models likely require a dedicated hosted instance. Pricing will vary.

What is Fine Tuning

# You have already (kinda) Fine-Tuned

Few shot/many shot prompting is "fine-tuning light"
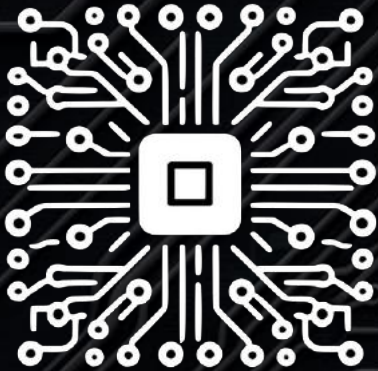
- Shot examples are good fine-tuning data
- Fine-tuning is like enormous shot prompting
- Fine-tuning is done once, prompting needs to be repeated

# Types of Fine Tuning

Full Fine-Tuning
- Updates all model parameters
- Substantial training data
- Expensive

PEFT: Parameter Efficient Fine-Tuning
- Updates small subset of parameters
- Small training data
- Cheaper and faster

# PEFT: LoRA, QLoRA

dk

**LoRA: Low-Rank Adaptation**

- Adds a small set of parameters that are fine-tuned
- During inference, the input parameters are multiplied by the new parameters and the result is added to the original parameter output
- Base model parameters are frozen
- Very efficient

**QloRa**

- Add quantization to make the process faster and more memory efficient
- Even more efficient

GIGO: Garbage In, Garbage Out

# Data Formats

- SFT: Supervised Fine Tuning
- DPO: Direct Preference Optimization
- Others include: PPO, KTO, CPO, etc. SFT and DPO are the most common

# SFT: Supervised Fine Tuning Data

- Typically, in pairs
- Examples
  - Q&A pairs
  - Prompts and completions
  - Translations
  - Summary
  - Entity Recognition
  - Sentiment

| △ Sentence | △ Sentiment |
|---|---|
| Index | Financial sentence |
| **5322**<br>unique values | neutral 54%<br>positive 32%<br>Other (860) 15% |
| The GeoSolutions technology will leverage Benefon 's GPS solutions by providing Location Based Searc... | positive |
| $ESI on lows, down $1.50 to $2.50 BK a real possibility | negative |
| For the last quarter of 2010 , Componenta 's net sales doubled to EUR131m from EUR76m for the same p... | positive |

# DPO: Direct Preference Optimization

- Tuple:
  - Question
  - Good response
  - Bad response
- Examples
  - CSR responses
  - Content moderation
  - Safe/unsafe code
  - Product recommendations



Source: huggingface.co/datasets/CyberNative/Code_Vulnerability_Security_DPO

# How Much Data?

- Depends upon the task and the model
- The simpler the task, the less data
  - Simple: Text classification, sentiment analysis, named entity recognition and style match
  - More complex: translation, long text generation and code generation
- The closer the task to the base model training, the less data required
- Start smaller (50-100 examples), test and add data as needed

That's the Hard Part
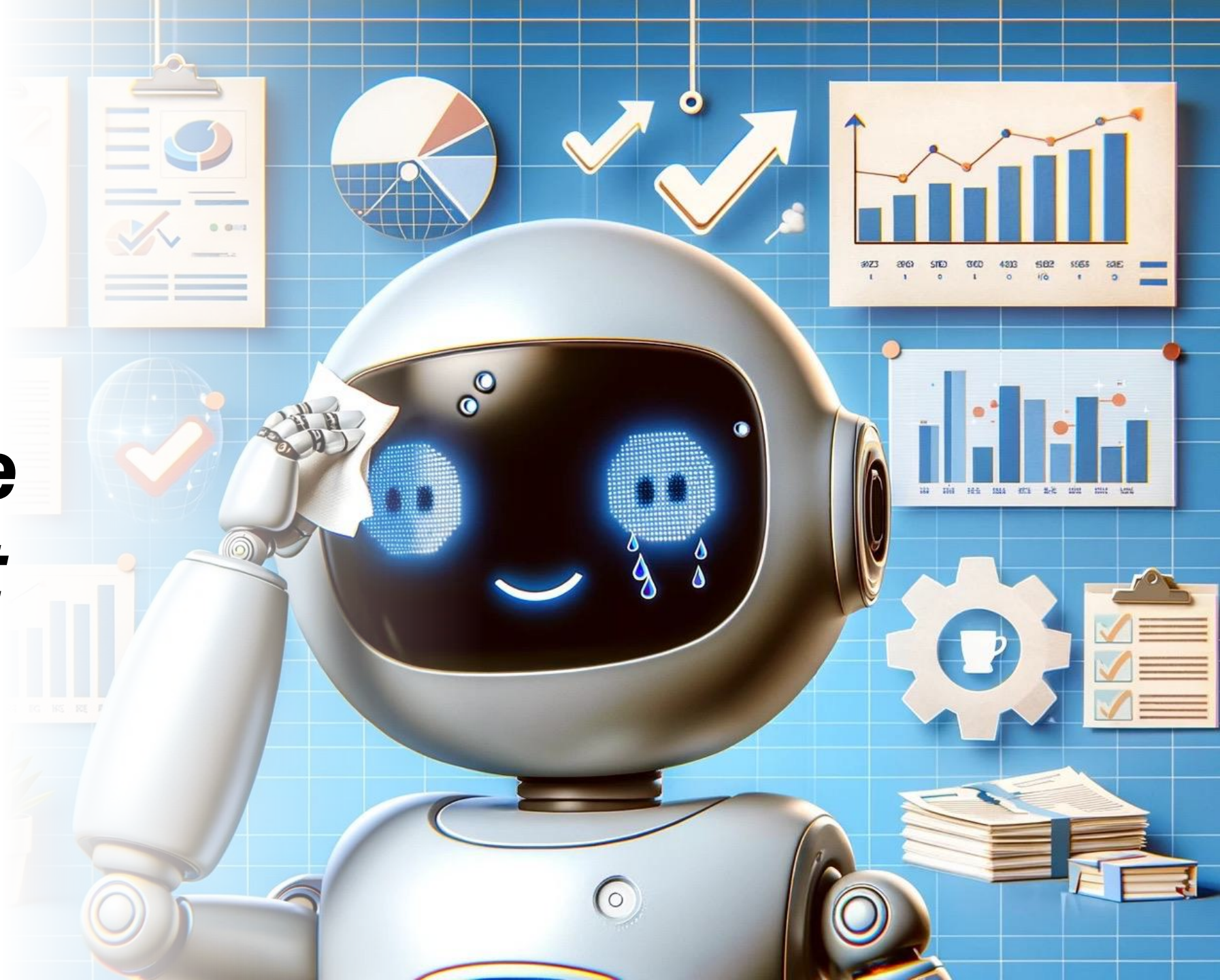
# Fine Tune Tools

- Axolotl
- OpenAI
- Mistral
- Google Gemini
- Others

# Axolotl



- Popular open-source library for fine-tuning LLMs
- Broad support for LLM architectures and fine-tuning methods
- Uses YAML file for configuration
- Wrapper for low-level Hugging Face libraries
- Excellent integration with Modal (modal.com), a serverless LLM platform

github.com/OpenAccess-AI-Collective/axolotl

# OpenAI Fine Tuning

- Fine-Tune GPT-3.5 Turbo (GPT-4 in beta)
- Uses OpenAI JSONL format training files
- Simple API or UI for fine-tuning
- Trained model available as endpoint
- Cost:

| | Train 1M Tokens (training only) | Input 1M Tokens | Output 1M Tokens |
| --- | --- | --- | --- |
| Base 3.5 | N/A | $0.50 | $1.50 |
| Fine-Tuned 3.5 | $8.00 | $3.00 | $6.00 |
| GPT-4o | N/A | $5.00 | $15.00 |

platform.openai.com/docs/guides/fine-tuning

# Mistral Fine Tuning

**MISTRAL AI_**

- New – announced June 5th
- Fine tune 7B and Small (8x7B, 8x22B – not hosted)
- Simple Mistral API and YAML configuration file
- Trained model available as endpoint
- Cost:

|  | Train 1M Tokens (training only) | Input 1M Tokens | Output 1M Tokens |
|---|---|---|---|
| Mistral 7B | N/A | $0.25 | $0.25 |
| Fine-Tune | $2.00 + $2.00 month | $0.75 | $0.75 |
| GPT-4o | N/A | $5.00 | $15.00 |

docs.mistral.ai/guides/finetuning/

# Mistral Fine Tuning

# Let's check on our fine-tuning

# Gemini Fine-Tuning

- Fine tune Gemini 1.0 Pro (more coming)
- Uses Gemini JSONL format training files
- Simple tuning API with JSON configuration
- Cost: **Same!!!**

| | Train 1M Tokens (training only) | Input 1M Tokens | Output 1M Tokens |
|---|---|---|---|
| Gemini 1.0 Pro | N/A | $0.50 | $1.50 |
| Fine-Tune | Free during preview | $0.50 **(Same!!)** | $1.50 **(Same!!)** |
| GPT-4o | N/A | $5.00 | $15.00 |

cloud.google.com/vertex-ai/generative-ai/docs/models/tune-gemini-overview

# Other Tools

HuggingFace AutoTrain
huggingface.co/autotrain

UnSloth
unsloth.ai
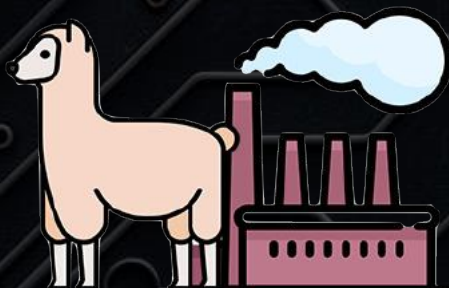
LLaMA Factory
github.com/hiyouga/LLaMA-Factory

W&B

Weights and Biases (wandb)
wandb.ai

# LoRA Adapters

- Serve multiple fine-tuned models from a single base model
- Inference time selection and loading of fine-tuned model
- What Apple (and Google?) use for their LLMs embedded in phones
- Cost effective, works well in constrained environment e.g. edge SLMs as well as data center

Tweaking fine-tuning hyperparameters

Improving fine-tuning data

# RAFT: Best of Both

Retrieval Augmented Fine Tuning

- RAG provides context based on semantic similarity but doesn't improve a model's understanding.

- Fine-tuning equips the model with new capabilities but doesn't introduce your data.

- Fine-tuning complements RAG: FT adapts to the domain, enabling RAG to identify the most relevant information.

gorilla.cs.berkeley.edu/blogs/9_raft.html

# RAFT: Best of Both



"Closed book" — Bake in Knowledge at Train Time — query → answer

"Open book" — Model can use External Docs at Test — query → answer

RAFT (Proposed) — Teach Model to use External Docs at Test — query → answer
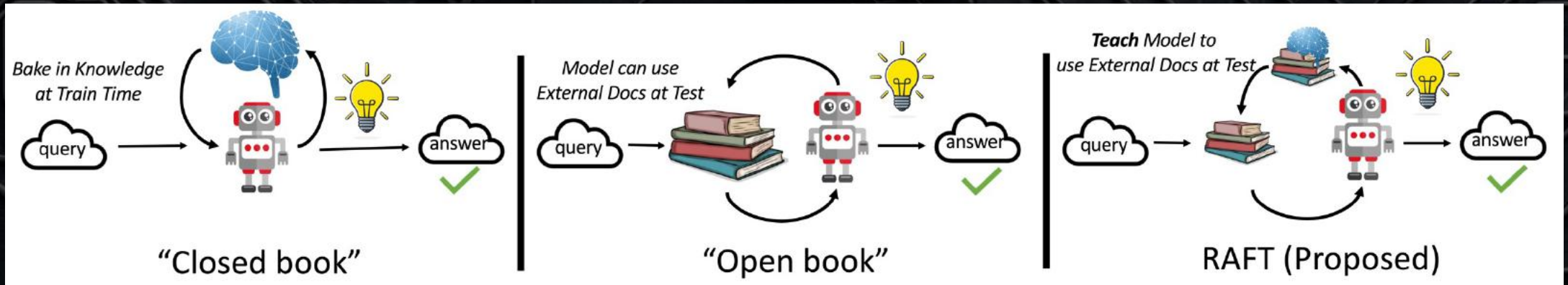
Textbook Analogy
- Fine-tuning is like studying textbook before class
- RAG is like an open book test
- RAFT combines the best of both

gorilla.cs.berkeley.edu/blogs/9_raft.html

# Closing Thoughts

- Prompt engineering, RAG and fine-tuning are all complimentary tools
- Easiest to hardest: Prompt engineering -> RAG -> Fine-Tuning
- RAG is about data, fine-tuning is about behavior
- Choose the right tool(s) for the job
- Don't be afraid of fine-tuning

# Thank You & Stay in Touch

## www.linkedin.com/in/rkibbe/

Presentation:
bit.ly/unparsed-finetuning
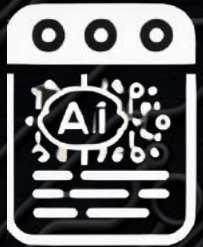
Code:
bit.ly/unparsed-finetuning-code

# Appendix

# Resources

Tools
- Axolotl: https://openaccess-ai-collective.github.io/axolotl/
- AutoTrain: https://huggingface.co/autotrain
- Gemini Fine Tune: https://cloud.google.com/vertex-ai/generative-ai/docs/models/tune-gemini-overview
- Mistral Fine Tune: https://docs.mistral.ai/guides/finetuning/
- OpenAI Fine Tune: https://platform.openai.com/docs/guides/fine-tuning
- RAFT: https://gorilla.cs.berkeley.edu/blogs/9_raft.html

# Big Models are Expensive



**Mistral 7B and LLaMA 3 8B are 25 X cheaper input and 75 X cheaper output than GPT-4o**

Project ⌄

⌄ 🗀 unparsed-2024-talk ~/Documents/Development/

⊕  ✂  ⧉  ⧉  ↑  ↓  ▷  🐞  ◻  ⟳  ⤇  🧹  🗑      Code ⌄      Managed: http://localhost:8889 ⌄      ◯ Python 3 (ipykernel) ⌄   ☐ Trusted   ⌃ ⌄ 🌐

> 🗀 .venv

> 🗀 images                                                                                                      ⚠1  ✓8  ⌃ ⌄

⌄ 🗀 mistral-fine-tune

   ☰ .env

   {} bofh_chunk_eval_mistral.jsonl

   {} bofh_chunk_train_mistral.jsonl

   {} bofh_training_data_mistral.jsonl

   🗎 mistral_finetune.ipynb

   🐍 simple_mistral_cli_client.py

   🐍 simple_mistral_gui_client.py

   🗀 openai-fine-tune

   🗀 presentation

   ⊘ .gitignore

   ☰ LICENSE

   M↓ README.md

> 📚 External Libraries

```
In _    1  # List all jobs
        2  jobs = client.jobs.list()
        3  pprint(jobs)


In _    1  # Retrieve latest job
        2  retrieved_jobs = client.jobs.retrieve(created_jobs.id)
        3  pprint(retrieved_jobs)


In 61   1  # The question we will ask the BOFH
        2  the_question = "My mouse is broken"
           Executed at 2024.06.13 12:21:20 in 1ms


In _    1  # Try the fined tuned model
        2  from mistralai.models.chat_completion import ChatMessage
        3
        4  chat_response = client.chat(
        5      model=retrieved_jobs.fine_tuned_model,
```

(.venv) rogerkibbe - mistral-fine-tune %