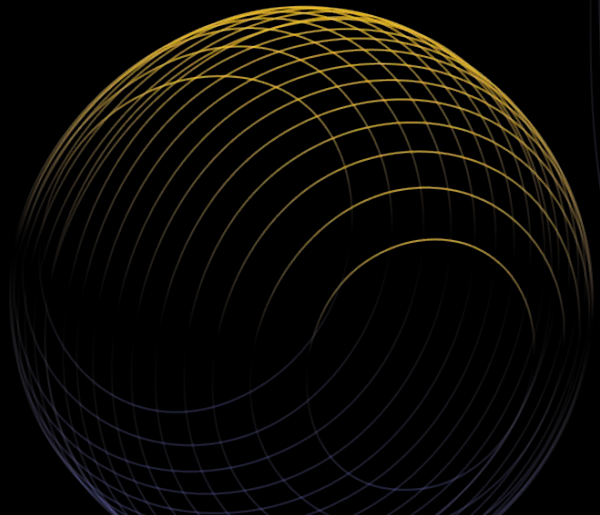
Fine-tuning in the Era of Large Models

Saahil Ognawala, *Sr. Product Manager @ Jina AI*

MLOps Community Munich Meetup, 19th Sept. 2023



Our Vision

Jina AI envisions paving the way towards the future of AI as a multimodal reality. We recognize that the existing machine learning and software ecosystems face challenges in handling multimodal AI.

Our vision is to play a crucial role in helping the world harness the vast potential of multimodal AI and truly revolutionize the way we interpret and interact with information.

About Me

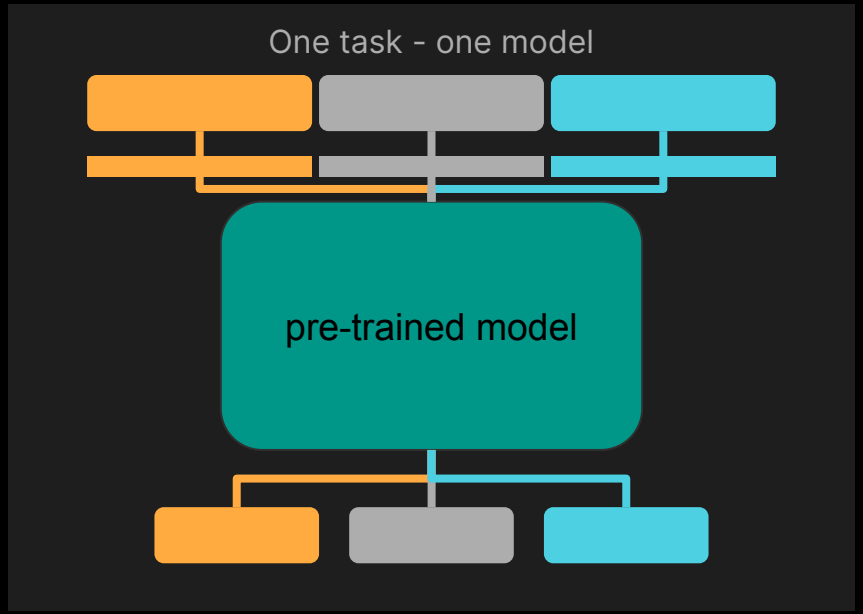
- 2022-Now: Senior Product Manager, Jina AI
- 2019-2022: Product Manager, Munich Re
- 2015-2019: Ph.D. in CS, TU Munich
- 2012-2014: M.Sc. in Informatics, TU Munich
- 2011-2012: Software Engg., Hewlett-Packard Enterprise
- 2007-2011: B.E. in CS, Manipal University, India

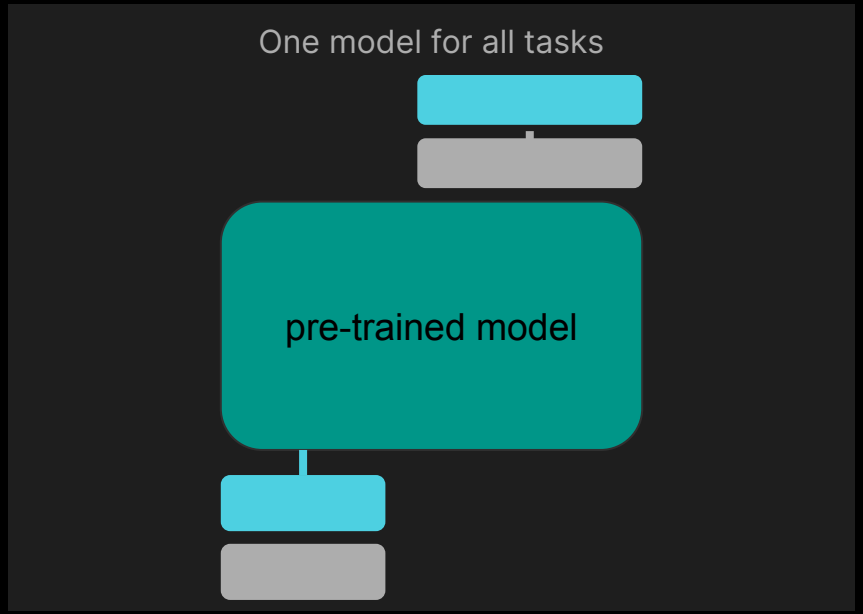

“We need a customized model”

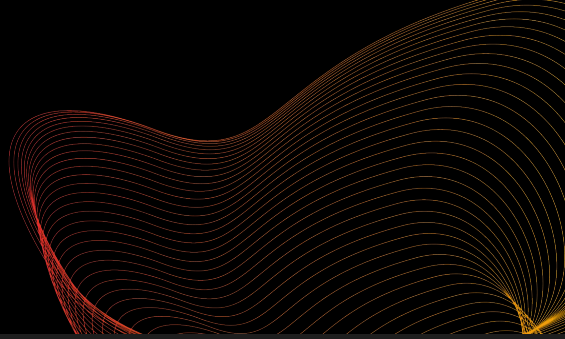
~~“We need a customized model”~~

“We need to customize an LLM to handle our tasks.”

A Paradigm Shift in ML Models



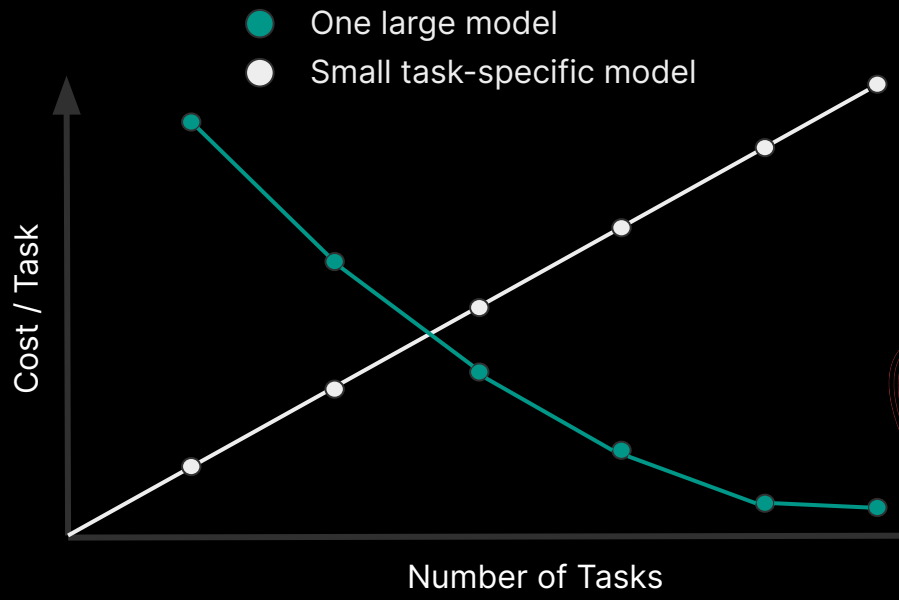




“The emergence of large models in AI has caused the cost of obtaining information to become fixed *(from marginal)*, leading to an inflection point similar to that of Google in the PC era.”

Dr. Lu Qi, *MiraclePlus*, YCombinator, Baidu ...

Implications on TCO of ML Models




How to enable large models to handle customized tasks?
Or
How to communicate effectively with a large model?

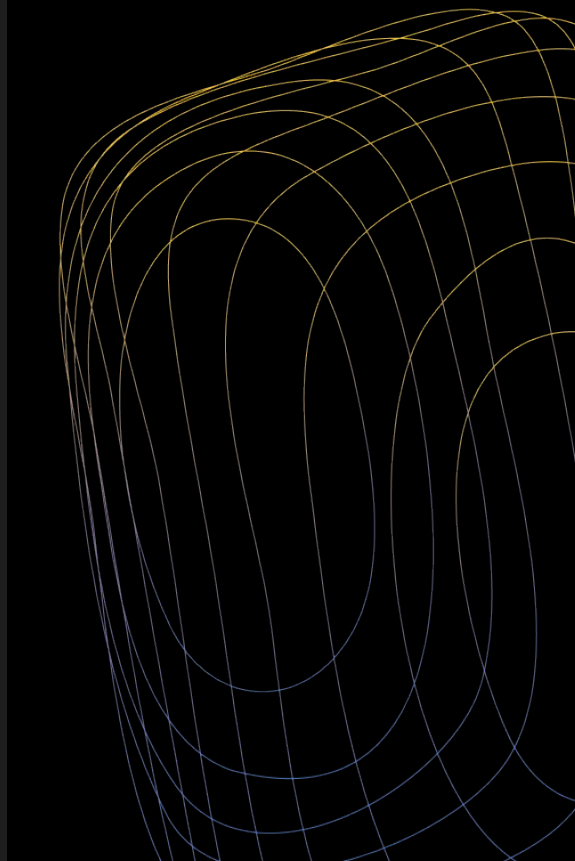
Using Johari Window to Understand Communication with Large Models

- Framework to understand our conscious and unconscious bias
- Decrease blind spots, avoid misunderstandings
- *Increase consensus*

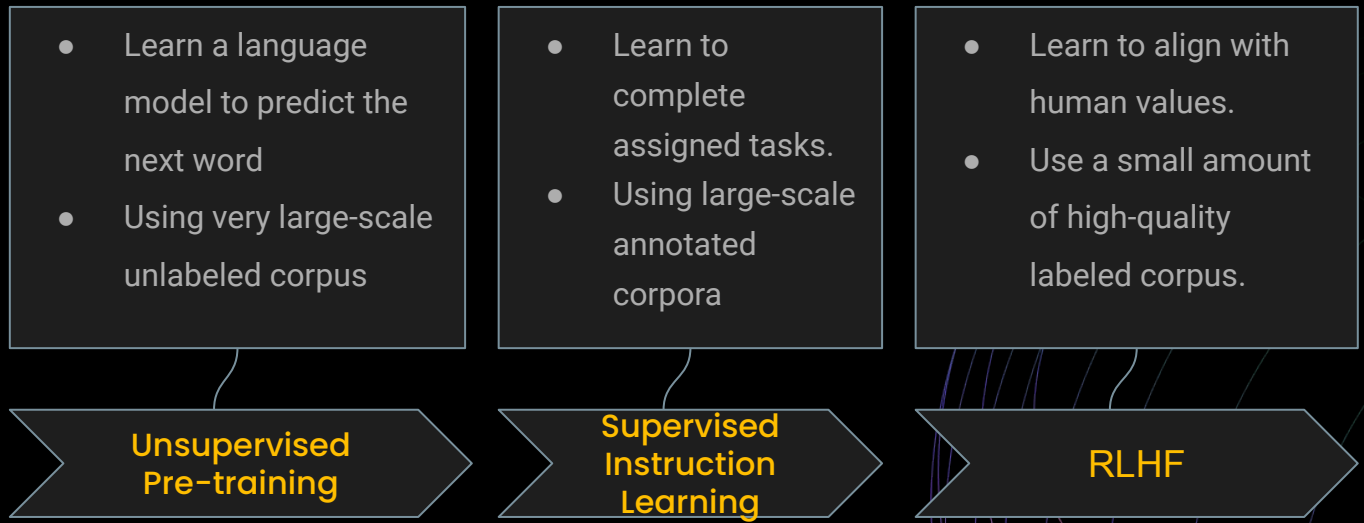
	Humans know	**Humans don't know**
Large model knows	**Consensus** Classification, query expansion, translation	**Blind spot** Encyclopedia knowledge
Large model doesn't know	**Misunderstanding** Private knowledge base	**????**

Two Paths to Increase Consensus With Large Models

1. Fine-tuning the Model: Reduce blind-spots
2. Fine-tuning the Prompt: Avoid misunderstandings

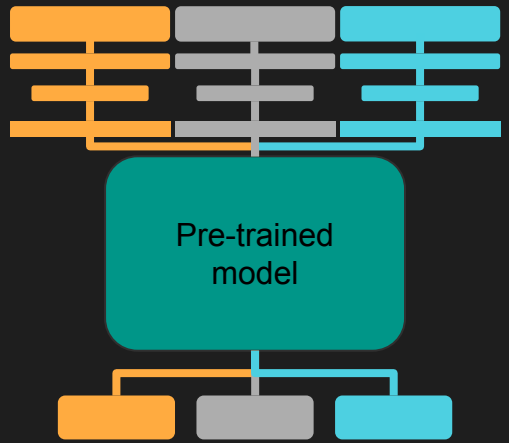


How a Large Language Model (LLM) is Trained

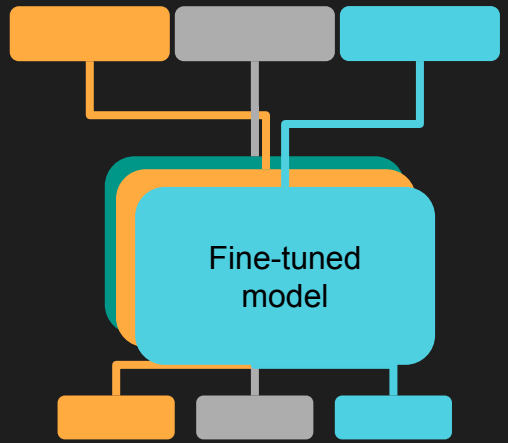


Path 1: Fine-tuning a Large Model

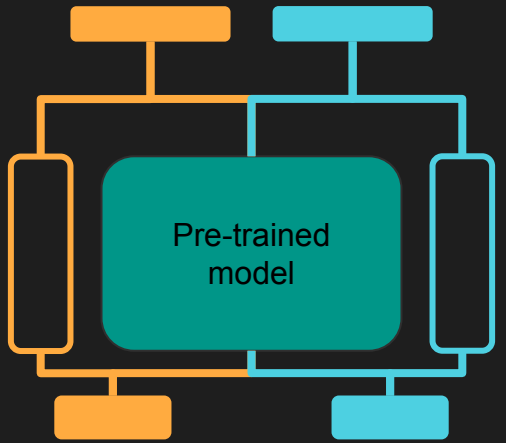
Partial Fine-tuning



Complete Fine-tuning



Parameter Efficient LoRa



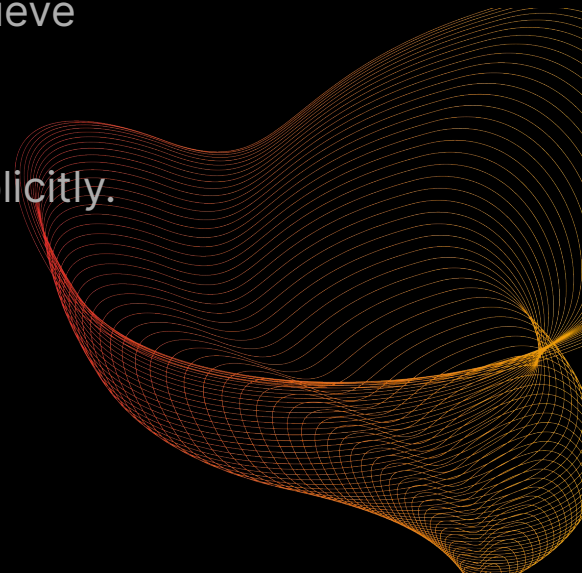
Finetuner by Jina

- Open-source Python library for fine-tuning multimodal models
- Jina Embedding models
  - 14M params, fastest inference time on HuggingFace MTEB board
  - Apache 2.0 licence
- *Finetuner+*: Enterprise product for finetuning large models with corporate data
  - Including state-of-the-art cost-efficient BYOC hosting

```
import finetuner
from finetuner import DocumentArray, Document
from finetuner.callback import EvaluationCallback

finetuner.login()

train_data =
DocumentArray.pull('finetuner/quora-train-da',
show_progress=True)
query_data =
DocumentArray.pull('finetuner/quora-test-query-da',
show_progress=True)
index_data =
DocumentArray.pull('finetuner/quora-test-index-da',
show_progress=True)

run = finetuner.fit(
    model='bert-base-en',
    train_data='finetuner/quora-train-da',
    loss='TripletMarginLoss',
    optimizer='Adam',
    learning_rate = 1e-5,
    epochs=3,
    batch_size=128,
    device='cuda',
    callbacks=[
        EvaluationCallback(

query_data='finetuner/quora-test-query-da',

index_data='finetuner/quora-test-index-da',
            batch_size=32
        )
    ]
)
artifact = run.save_artifact('bert-model')

model = finetuner.get_model(artifact=artifact,
device='cuda')

query = DocumentArray([Document(text='How can I be
an engineer?')])

finetuner.encode(model=model, data=query)
finetuner.encode(model=model, data=index_data)
assert query.embeddings.shape == (1, 768)

query.match(index_data, limit=10, metric='cosine')
```

Path 2: Prompt Tuning

- Prompt tuning is the process of optimizing the input queries (prompts) to a pre-trained machine learning model to achieve desired outputs without altering the model itself.
- Goal for increasing consensus with humans: Reduce misunderstandings by specifying the context of tasks explicitly.



Prompt Tuning vs Model Tuning

Prompt Tuning

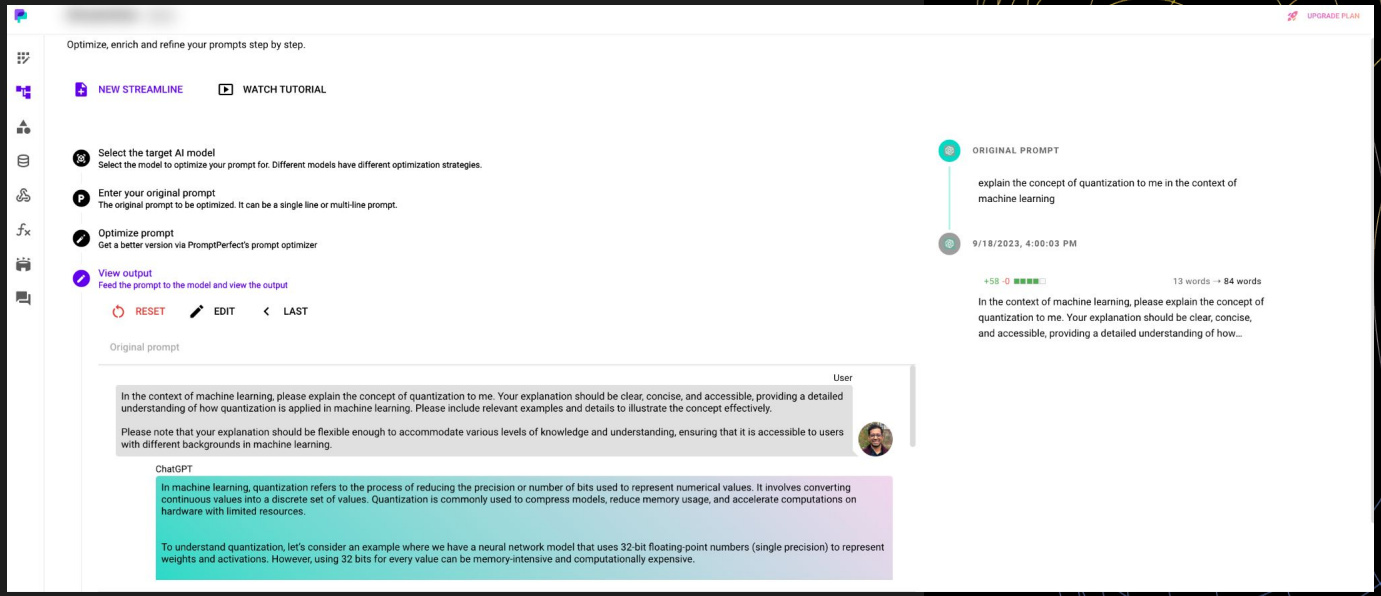
- Quick and cost-effective way to provide private context for tasks unknown to the model.
- Under limited budget: Comparable performance to a parametric fine-tuned model.
- Capabilities of the underlying model are difficult to adapt with only limited input.

Model Tuning

- For higher accountability, longer term projects with significant amount of complex and interlinked context information.
- Can be a costly process of training *and hosting*!

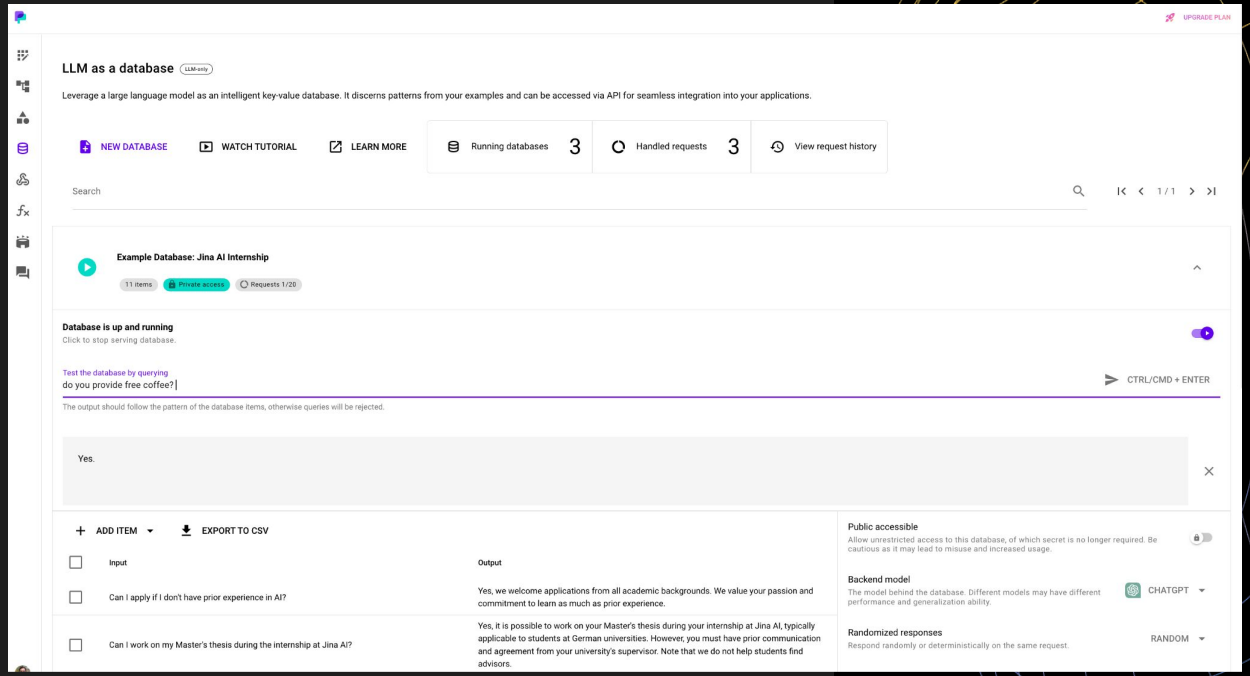
PromptPerfect by Jina

Iterative optimization based on user's prompt and selected model



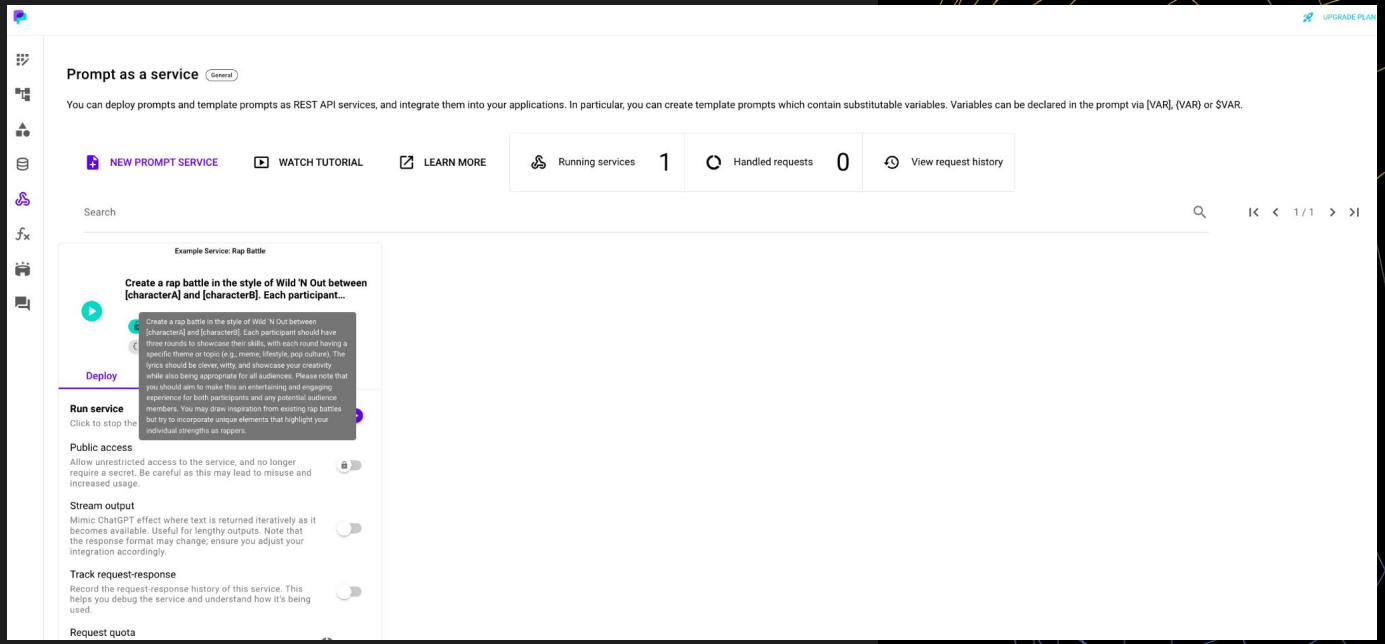
PromptPerfect by Jina

Store Key-value pairs, to create a quick lightweight version of RAG



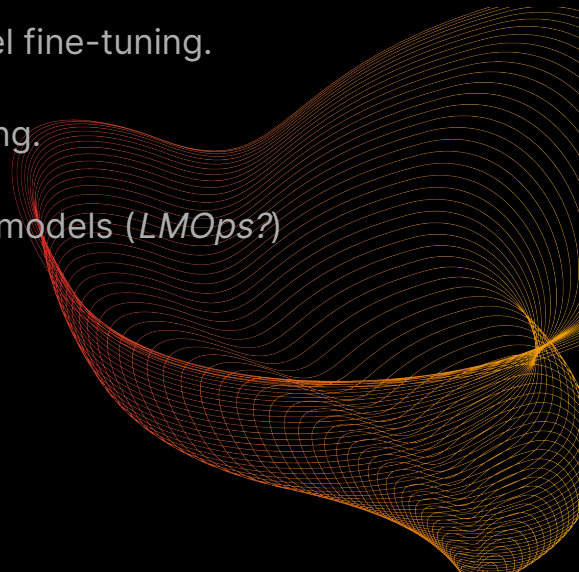
PromptPerfect by Jina

Quickly deploy prompts as APIs

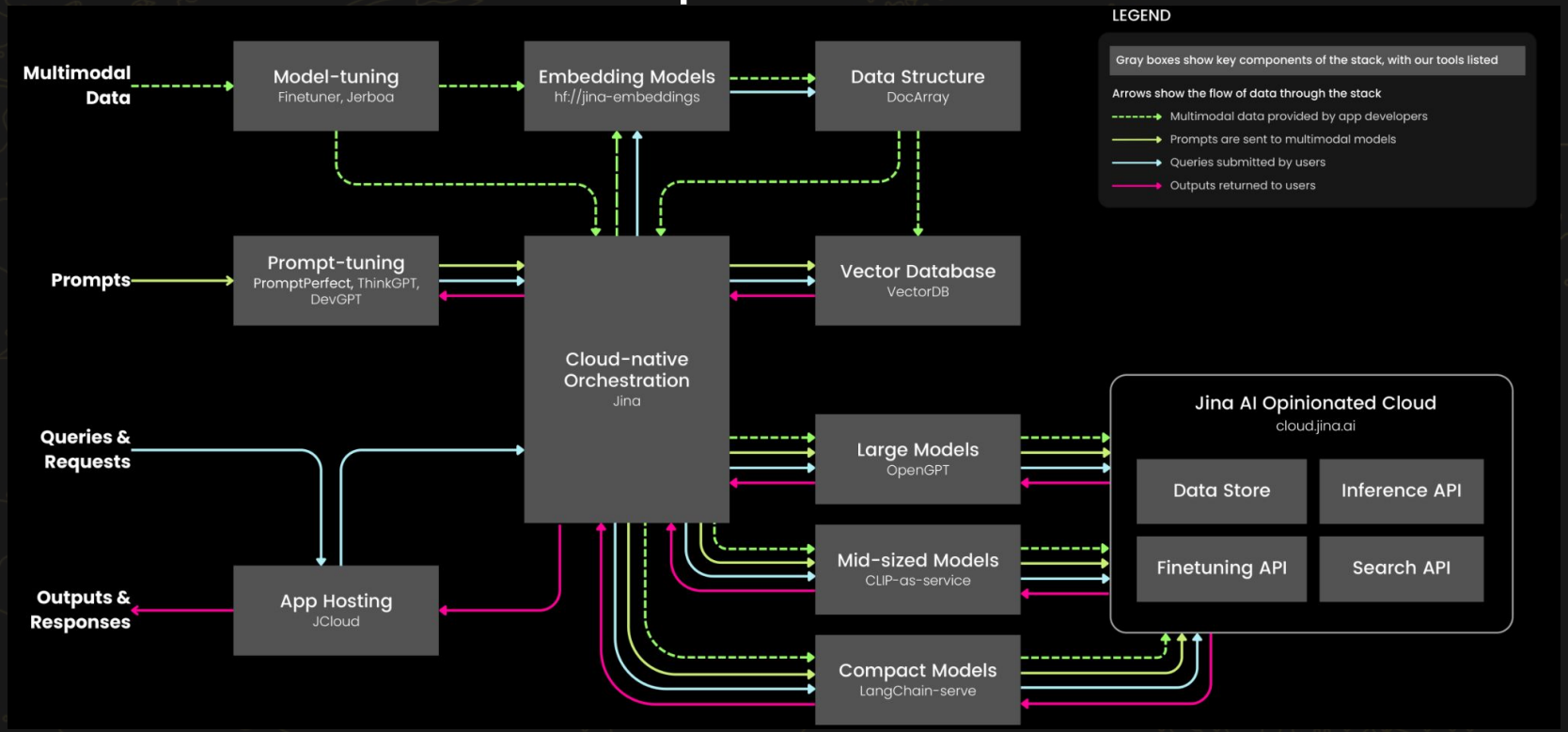


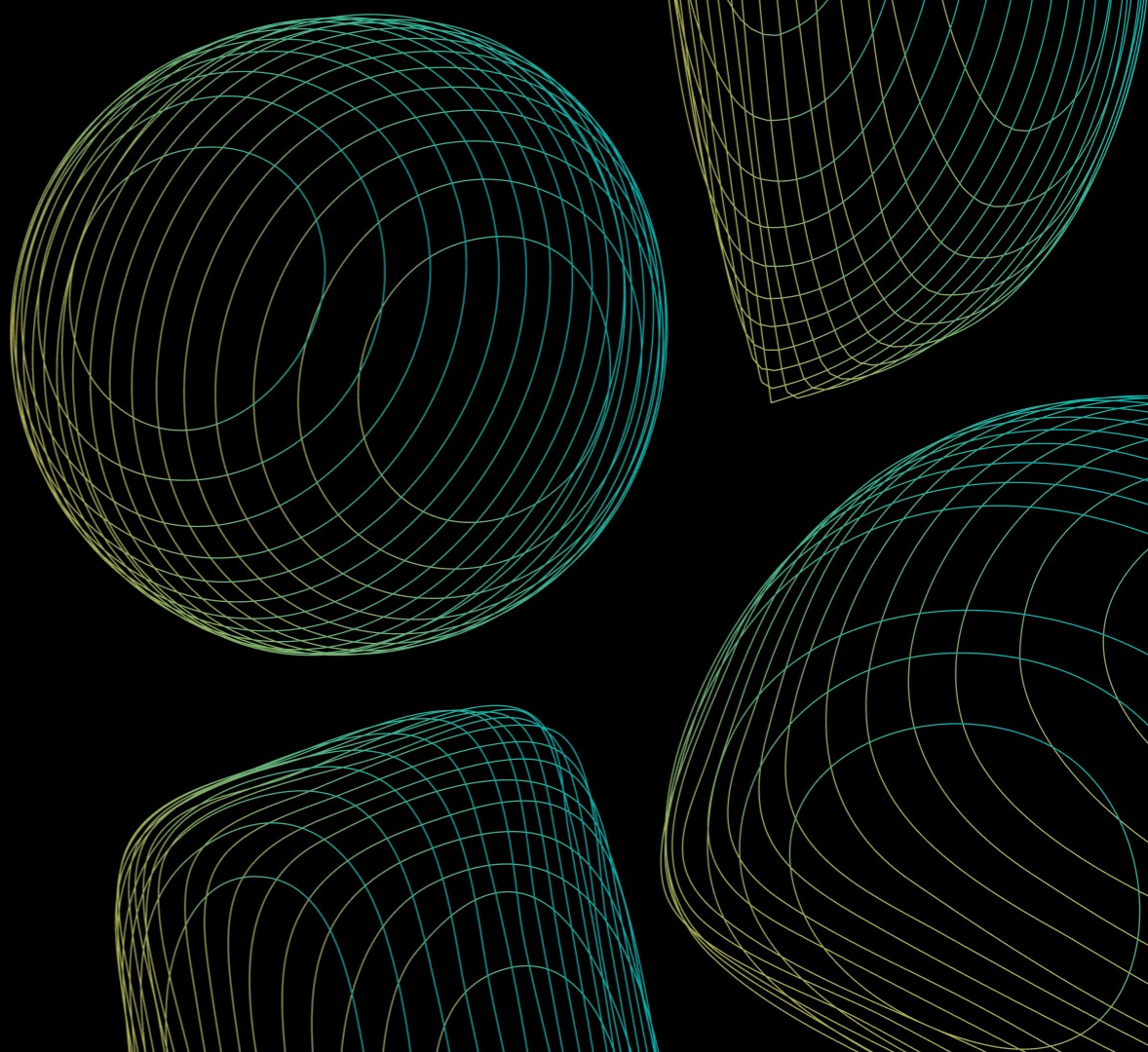
Summary

- Advent of large models will reduce the TCO for deploying ML models for handling different tasks in enterprises.
- (Ab)using Johari Window to understand fine-tuning of large models: *The goal* is to increase consensus between human and model knowledge.
- Reducing blind spots from human knowledge: Using parametric model fine-tuning.
  - Jina Finetuner and Finetuner+
- Reducing misunderstandings in model knowledge: Using prompt tuning.
  - Jina PromptPerfect
- Not discussed so far: High computation demands in MLOps for large models (*LMOps?*)



Jina AI's Product Landscape





Thank you

saahil.ognawala@jina.ai

oss.jina.ai

jina.ai

discord.jina.ai

@JinaAI_