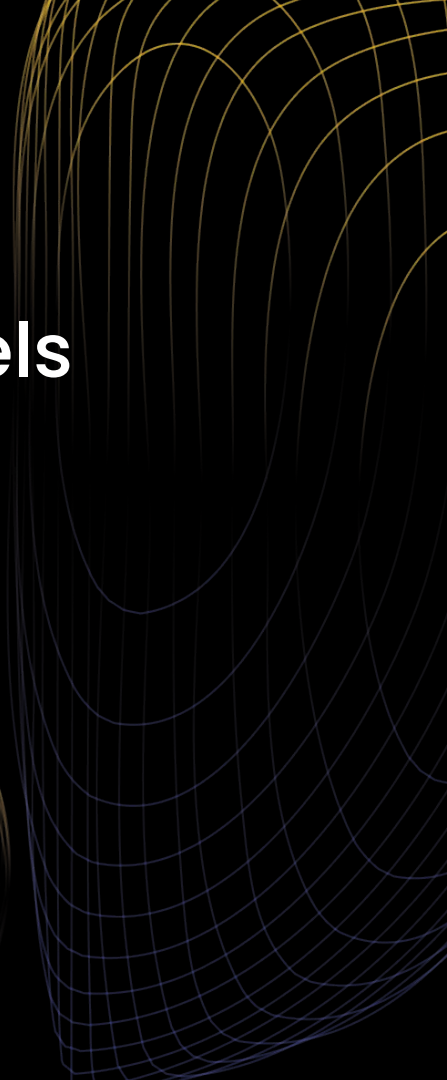
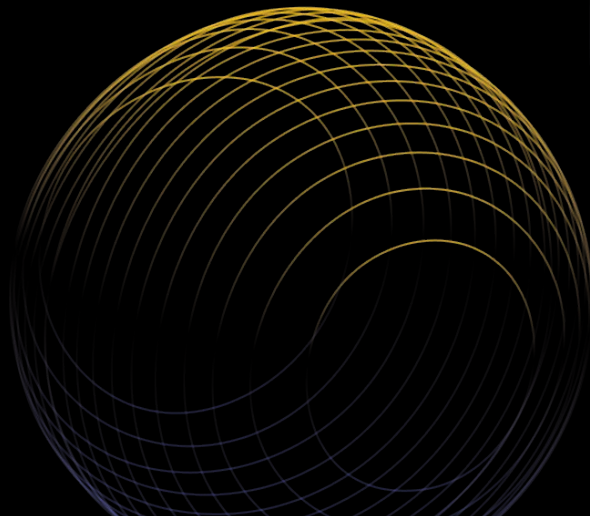




Fine-tuning in the Era of Large Models

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"~~

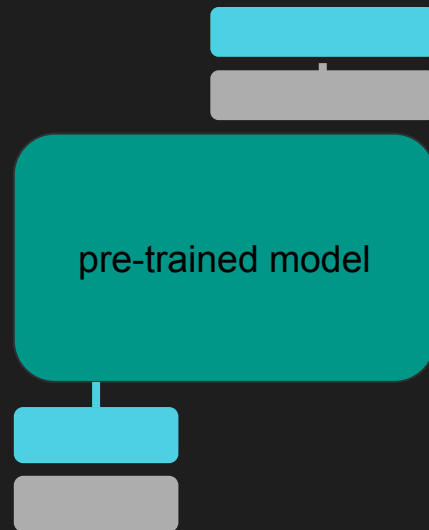
"We need to customize an LLM to
handle our tasks."

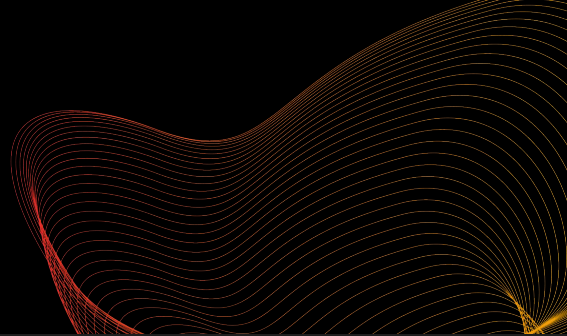
A Paradigm Shift in ML Models

One task - one model



One model for all tasks

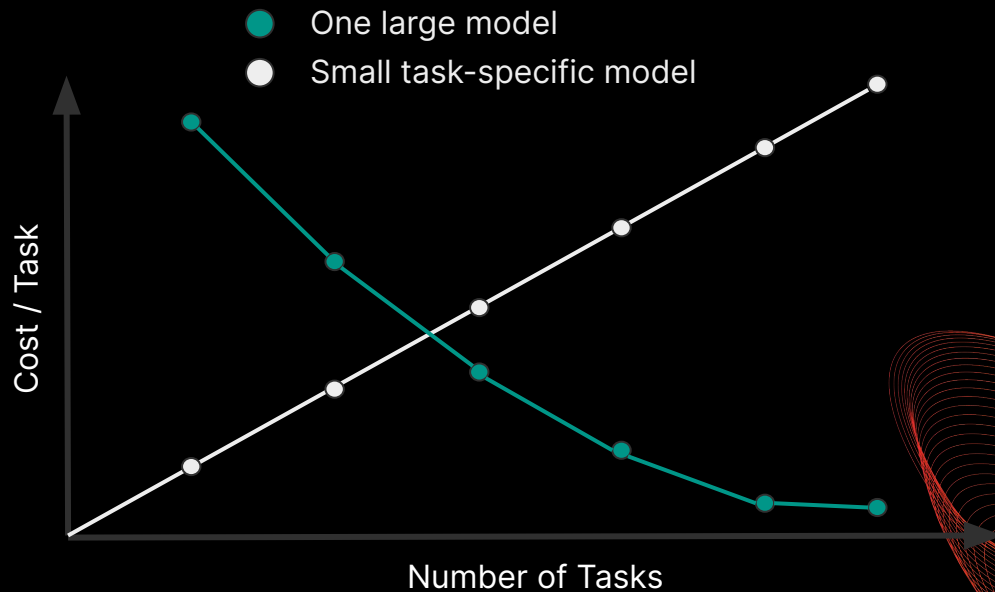




“The emergence of large models in AI has caused the cost of obtaining information to become fixed (*from marginal*), leading to an inflection point similar to that of Google in the PC era.”

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

Implications on TCO of ML Models





How to enable large models to handle customized tasks?

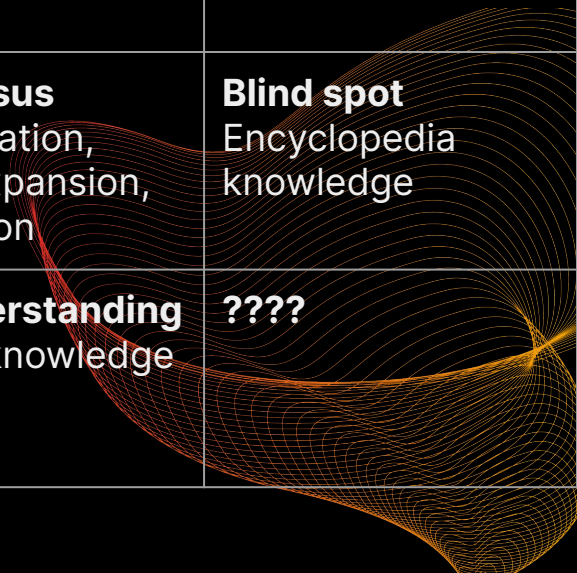
Or

How to communicate effectively with
a large model?

Using Johari Window to Understand Communication with Large Models

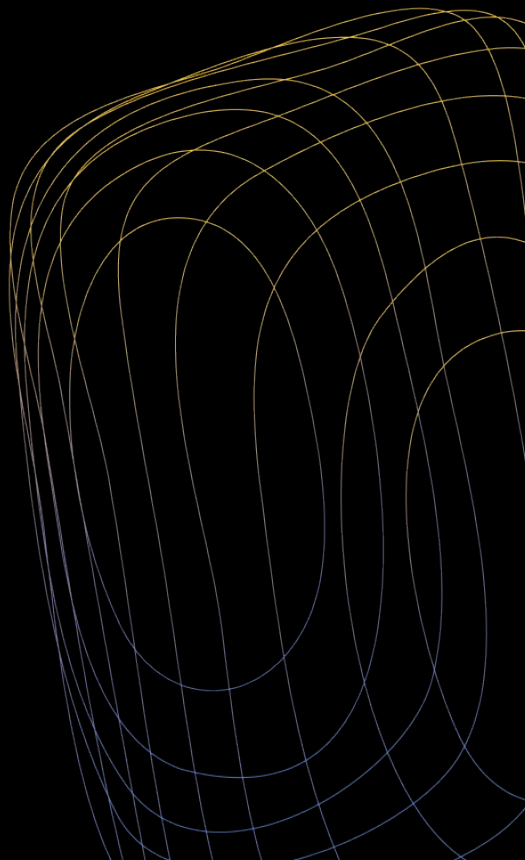
- Framework to understand our conscious and unconscious bias
- Decrease blind spots, avoid misunderstandings
- *Increase consensus*

	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	????

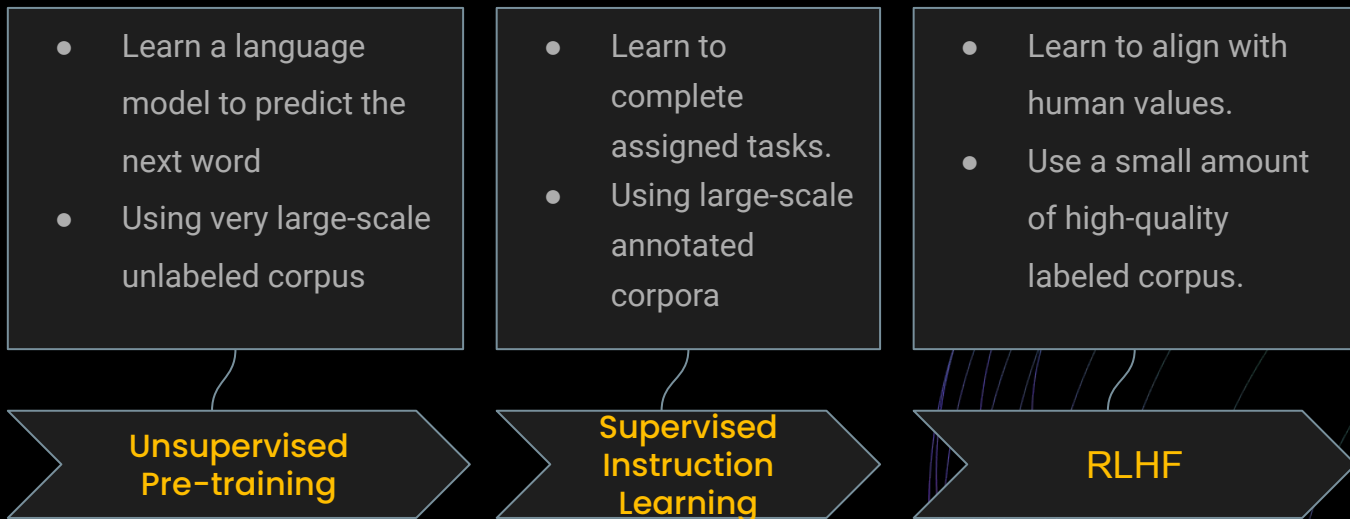


Two Paths to Increase Consensus With Large Models

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

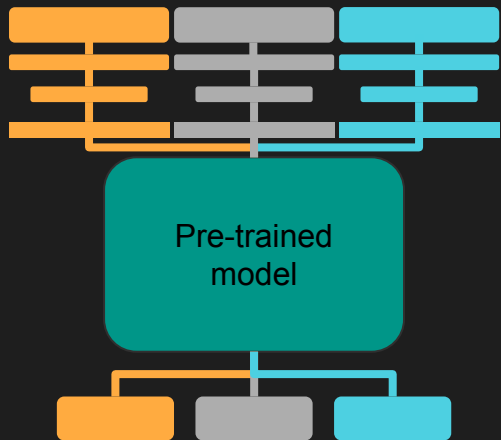


How a Large Language Model (LLM) is Trained

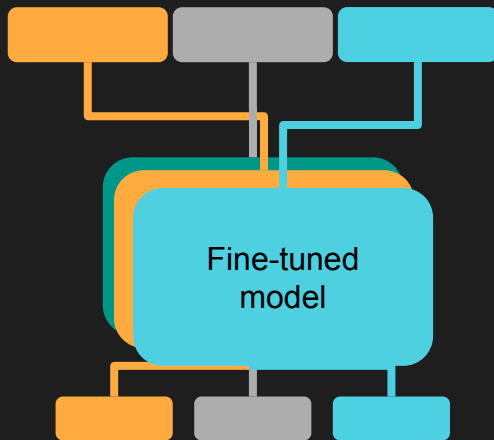


Path 1: Fine-tuning a Large Model

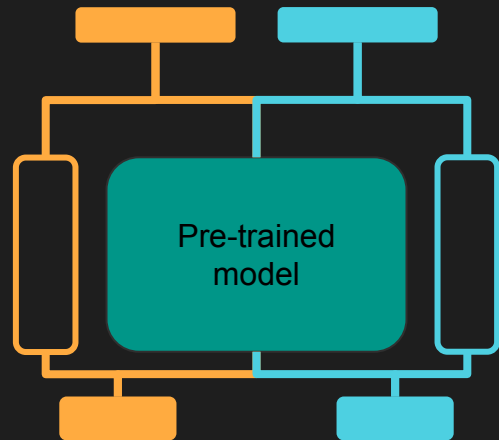
Partial Fine-tuning



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

The screenshot displays the PromptPerfect web interface, which is designed for iterative prompt optimization. The interface includes a sidebar with navigation icons, a main workspace, and a right-hand panel for prompt history.

Optimize, enrich and refine your prompts step by step.

NEW STREAMLINE **WATCH TUTORIAL**

- Select the target AI model**
Select the model to optimize your prompt for. Different models have different optimization strategies.
- Enter your original prompt**
The original prompt to be optimized. It can be a single line or multi-line prompt.
- Optimize prompt**
Get a better version via PromptPerfect's prompt optimizer.
- View output**
Feed the prompt to the model and view the output.

RESET **EDIT** **< LAST**

Original prompt

User

In the context of machine learning, please explain the concept of quantization to me. Your explanation should be clear, concise, and accessible, providing a detailed understanding of how quantization is applied in machine learning. Please include relevant examples and details to illustrate the concept effectively.

Please note that your explanation should be flexible enough to accommodate various levels of knowledge and understanding, ensuring that it is accessible to users with different backgrounds in machine learning.

ChatGPT

In machine learning, quantization refers to the process of reducing the precision or number of bits used to represent numerical values. It involves converting continuous values into a discrete set of values. Quantization is commonly used to compress models, reduce memory usage, and accelerate computations on hardware with limited resources.

To understand quantization, let's consider an example where we have a neural network model that uses 32-bit floating-point numbers (single precision) to represent weights and activations. However, using 32 bits for every value can be memory-intensive and computationally expensive.

ORIGINAL PROMPT

explain the concept of quantization to me in the context of machine learning

9/18/2023, 4:00:03 PM

+58 -0 ■■■■■ 13 words → 84 words

In the context of machine learning, please explain the concept of quantization to me. Your explanation should be clear, concise, and accessible, providing a detailed understanding of how...



PromptPerfect by Jina

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

The screenshot displays the PromptPerfect web interface. At the top, the title "LLM as a database" is followed by a description: "Leverage a large language model as an intelligent key-value database. It discerns patterns from your examples and can be accessed via API for seamless integration into your applications." Below this, there are buttons for "NEW DATABASE", "WATCH TUTORIAL", and "LEARN MORE". A summary bar shows "Running databases: 3", "Handled requests: 3", and a link to "View request history".

The main section is titled "Example Database: Jina AI Internship" and shows "11 items", "Public access", and "Requests 1/20". A status message says "Database is up and running" with a "Click to stop serving database." link. Below this, a query input field contains the text "do you provide free coffee?". A button with a right arrow and "CTRL/CMD + ENTER" is next to the input. The output area shows the response "Yes."

At the bottom, there are controls for "ADD ITEM" and "EXPORT TO CSV". A table with three rows is shown, each with an "Input" column, an "Output" column, and a "Public accessible" checkbox. The first row has an empty input field. The second row has the input "Can I apply if I don't have prior experience in AI?" and the output "Yes, we welcome applications from all academic backgrounds. We value your passion and commitment to learn as much as prior experience." The third row has the input "Can I work on my Master's thesis during the internship at Jina AI?" and the output "Yes, it is possible to work on your Master's thesis during your internship at Jina AI, typically applicable to students at German universities. However, you must have prior communication and agreement from your university's supervisor. Note that we do not help students find advisors."

On the right side of the interface, there are settings for "Public accessible" (checked), "Backend model" (set to "CHATGPT"), and "Randomized responses" (set to "RANDOM").



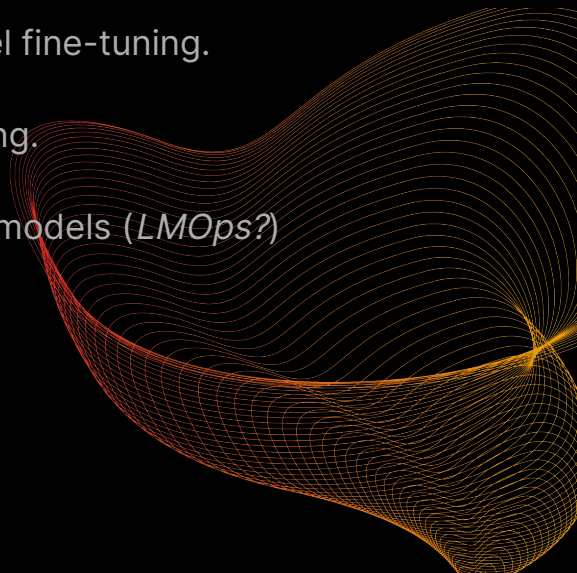
PromptPerfect by Jina

Quickly deploy prompts as APIs

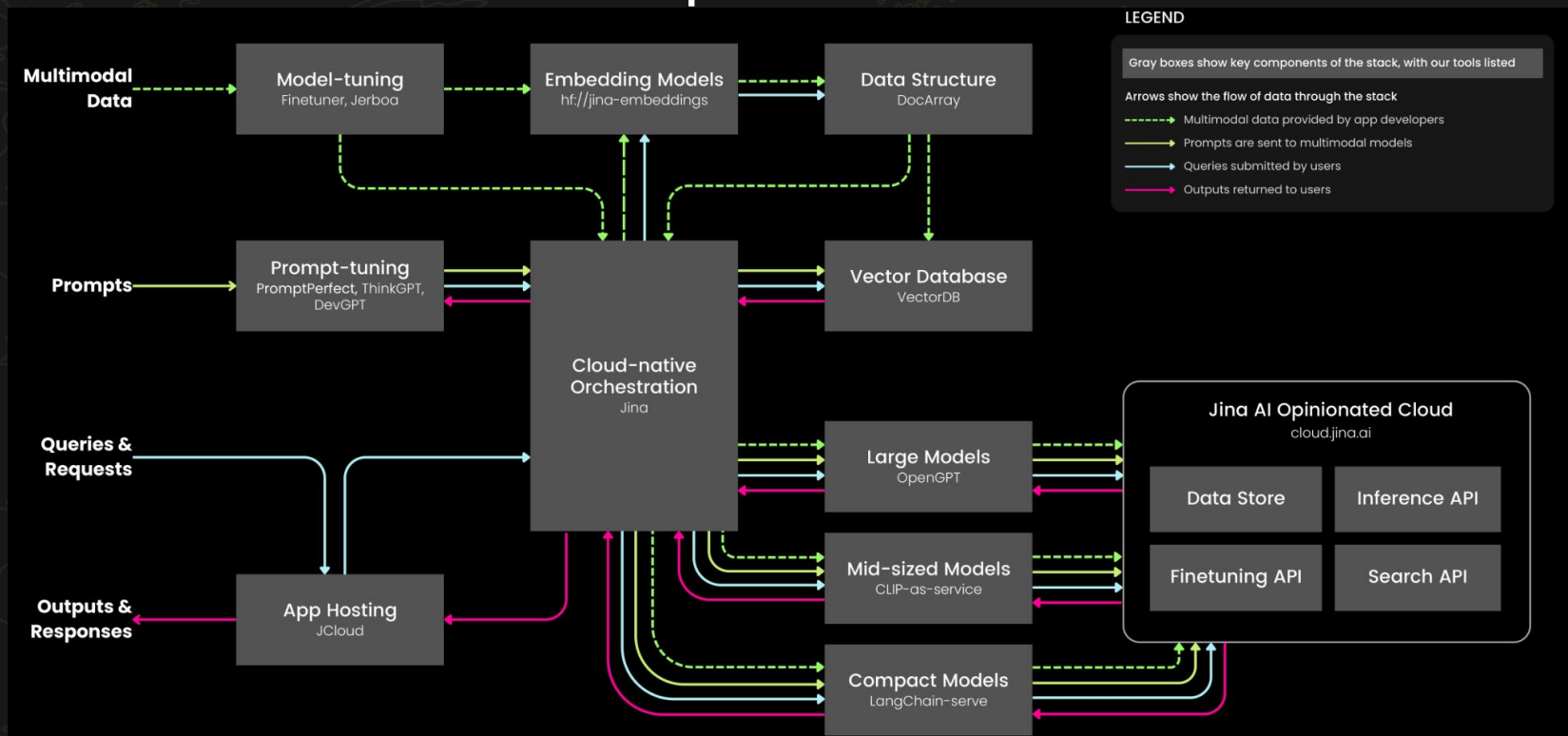
The screenshot displays the PromptPerfect by Jina web interface. At the top, the title "Prompt as a service" is followed by a "General" tab. Below this, a descriptive paragraph states: "You can deploy prompts and template prompts as REST API services, and integrate them into your applications. In particular, you can create template prompts which contain substitutable variables. Variables can be declared in the prompt via [VAR], {VAR} or \$VAR." A navigation bar includes links for "NEW PROMPT SERVICE", "WATCH TUTORIAL", and "LEARN MORE". A status bar shows "Running services 1", "Handled requests 0", and a "View request history" link. A search bar is located below the navigation bar. The main content area features an "Example Service: Rap Battle" section. It includes a "Deploy" button, a "Run service" button with a "Click to stop this service" link, and a "Public access" toggle switch. Below these are sections for "Stream output" and "Track request-response", each with a toggle switch. A "Request quota" section is also visible at the bottom. A modal window is open, displaying the prompt text: "Create a rap battle in the style of Wild 'N Out between [characterA] and [characterB]. Each participant...". The modal also contains a detailed description of the prompt's purpose and usage instructions.

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_](https://twitter.com/JinaAI_)