ECS 2025

Personalize your AI: model Fine-Tune with your data in Azure OpenAI

Massimo Bonanni

Senior Technical Trainer @ Microsoft

Premium Partner

Microsoft

Premium Sponsor

EXPERTS INSIDE

Technology Partner

run events

Diamond Sponsor

AURUM

docusign

dox42

LightningTools

ShareGate

Platinum Sponsor

AvePoint

Capgemini

CoreView

empowerID

glueck kanja

involv

ITENOS

Jabra GN

nintex

Nordcloud

NUTANIX

q.beyond

Rencore

Syskit

Xylos

Gold Sponsor

365TRIBE

BCC

BlueCallom

ClickShare

DEKOM | neat.

dstny

d.velop

Gamma

loopup

protiviti

Red Hat

skybow

SquaredUp

U2U

UDS

VISIONET

Silver Sponsor

Apvee

brandplane

CROSSWARE

CYQUEO

daenet

datacamp

EPOS

ichicraft

Lucid

OnTime

opium.

panagenda

rocketta

Pearson VUE

pointfire

PowerSyncPro

Yealink

Bronze Sponsor

Alpha Variance Solutions | GmbH

Analytics365

Callroute

epiq

HeiReS

Intelligent Decisioning

iThink365

kurmi

leapwork

sapio365

SOLUTIONS 2SHARE

WristPlanner

Startup Silver

Graia

ShArc

Startup Bronze

AXL hub

MEGALADATA

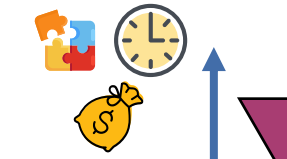
TeamsFox

vshosting

Startup Sponsor

emgc

The way to use your data in LLM



Cost/Complexity

Prompt Engineering

- Uses prompts to improve the accuracy and relevancy of responses from natural language processing models.
- By optimizing prompts, the model's performance is enhanced

Retrieval Augmented Generation

- Enhances performance by fetching data from external sources and incorporating it into a prompt.
- This method allows to create customized solutions, maintain data relevance, and control costs.

Fine Tuning

- Retrains a pre-existing LLM using example data to create a custom version of the model.
- This customization optimizes model based on the provided examples.

Train a model

- You need to have talent, data and GPU's to train a new model...
- ....and time and money also!!!!

Prompt Engineering and RAG

Will my sleeping bag work for my trip to Patagonia next month?

+

Basic Prompt Engineering

- Tone and Style
- Example Responses
- Intent Mapping

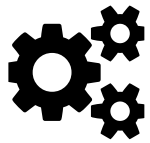
Retrieval / RAG

- Weather lookup
- Personalization
- ..and more!

=

Yes, your Elite Eco sleeping bag is rated to 21.6F, which is below the average low temperature in Patagonia in September

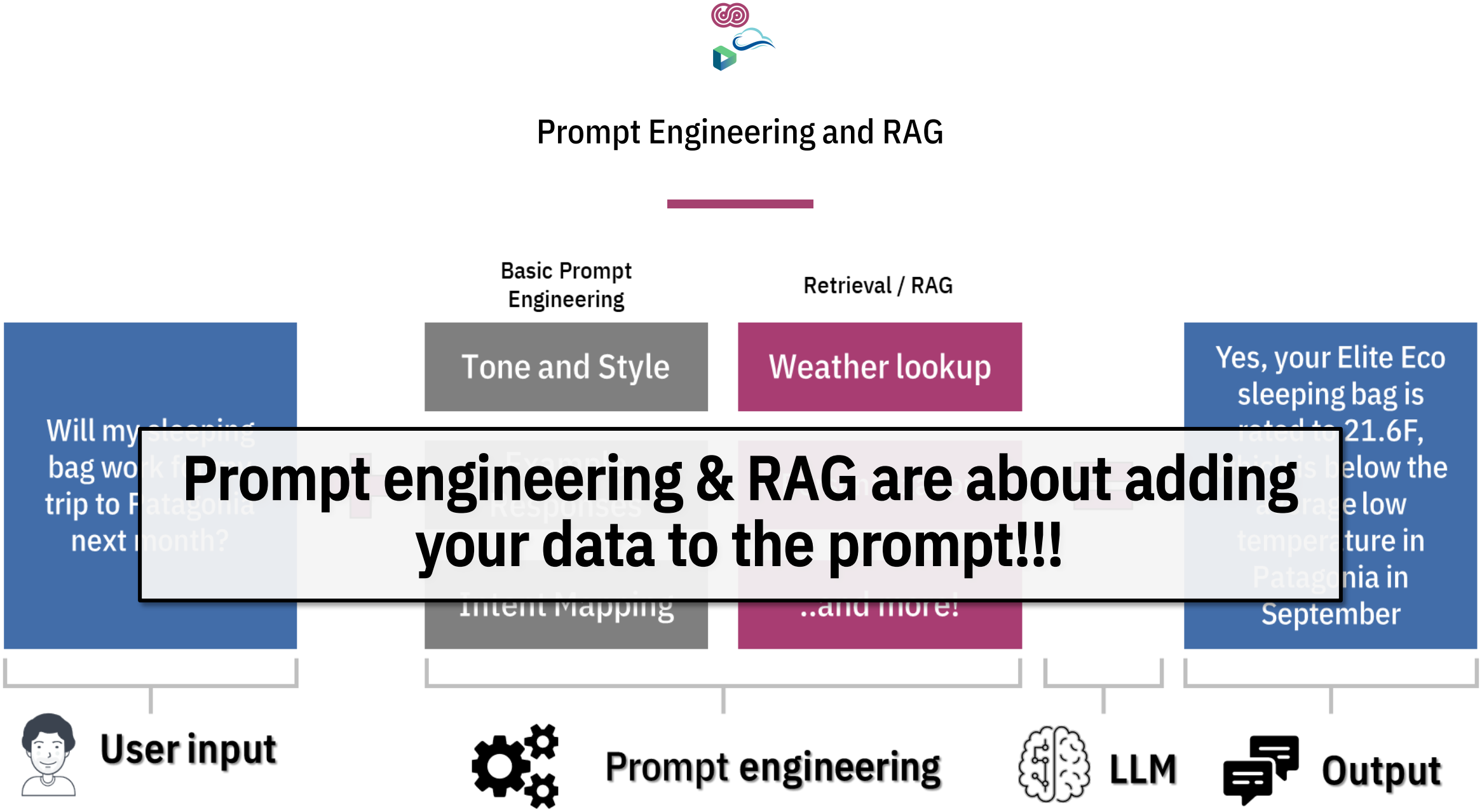
User input



Prompt engineering

LLM

Output


Prompt Engineering and RAG
Basic Prompt Engineering
Retrieval / RAG
Tone and Style
Weather lookup
Yes, your Elite Eco sleeping bag is
21.6F,
elow the
e low
ture in
nia in
September
Prompt engineering & RAG are about adding your data to the prompt!!!
User input
Prompt engineering
LLM
Output

Fine-Tuning

Basic Prompt Engineering

Retrieval / RAG

Tone and Style

Weather lookup

Will my sleeping bag work for my trip to Patagonia next month?

Yes, your Elite Eco sleeping bag is rated to 21.6F, which is below the average low temperature in Patagonia in September

Fine tuning is about adding your data to the LLM itself

Intent Mapping

..and more!

User input

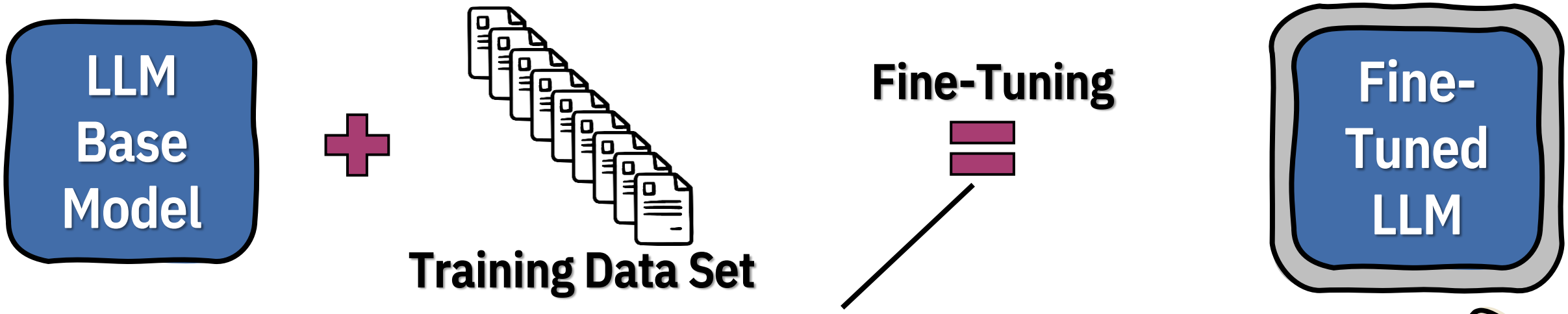
Prompt engineering

Fine-Tuned LLM

Output

What is Fine-Tuning

Fine-tuning refers to customizing a pre-trained LLM with additional training on a specific task or new dataset for enhanced performance and accuracy



Supervised Fine Tuning (SFT): supported by all non-reasoning models.
Direct Preference Optimization (Preview) (DPO): supported by GPT-4o.
Reinforcement Fine Tuning (Preview) (RFT): supported by reasoning models, like o4-mini.

The fine-tuning workflow

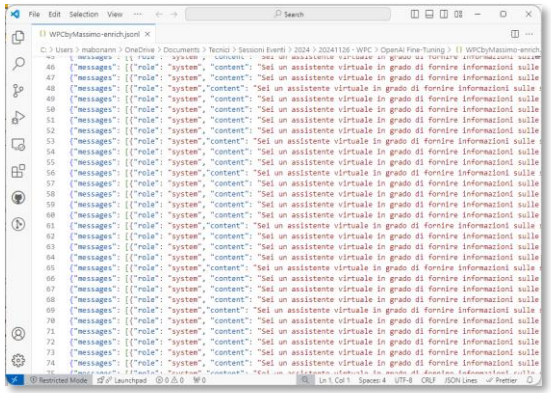
Data

Prepare 100s+ examples

Format as chat completion

Define training & validation sets

Upload data to the service



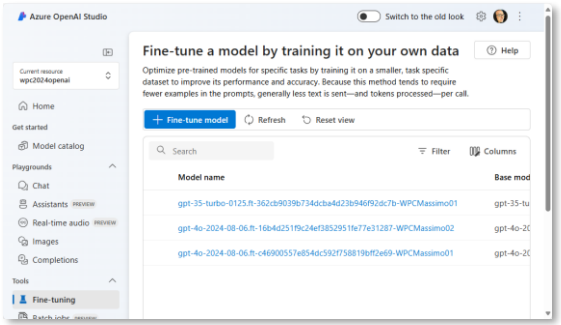
Training

Select base model

Set hyperparameters

Specify data

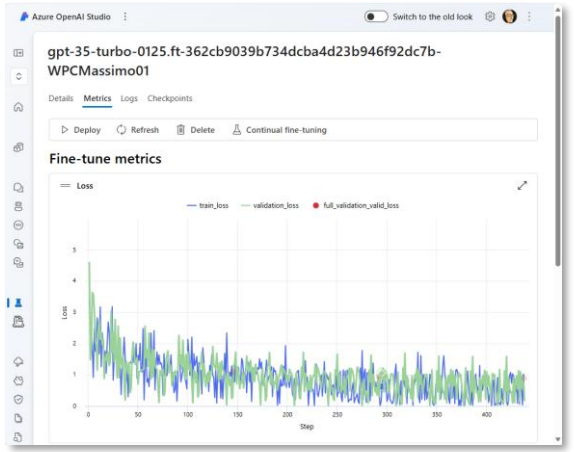
Executes on Shared Compute



Evaluation

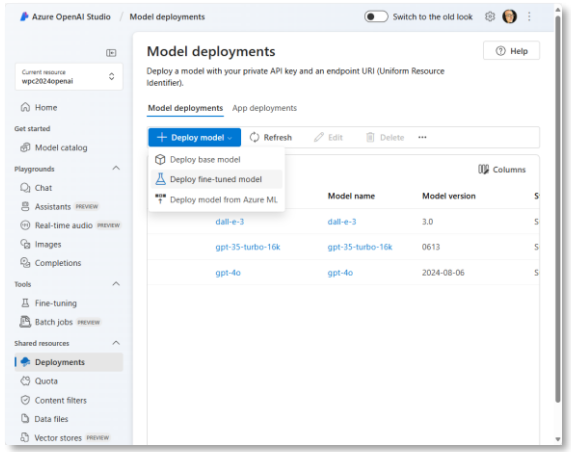
Training & validation loss

Training & validation accuracy



Deployment

Model endpoint for inferencing with AOAI



Prepare training and validation data

- Your training and validation datasets should contain input and output samples that reflect the model's expected performance.
- Ensure the data is formatted as a JSON Lines (JSONL) document.
- Providing a larger number of training examples is beneficial; aim for hundreds or thousands for best results.
- Generally, expanding the dataset size can improve model quality, but be cautious as low-quality examples can negatively impact performance.

```
{
    "messages": [
        {
            "role": "system",
            "content": "You are an helpful assistant that help people to kn
        },
        {
            "role": "user",
            "content": "What does the Apparecchio tableware set include?"
        },
        {
            "role": "assistant",
            "content": "The Apparecchio tableware set includes stylish plat
    ]
}
{
    "messages": [
        {
            "role": "system",
            "content": "You are an helpful assistant that help people to kn
        }
        ,
        {
            "role": "user",
            "content": "What materials are the Apparecchio plates and bowls
        },
        {
            "role": "assistant",
            "content": "The Apparecchio plates and bowls are made of durabl
        }
    ]
}
```

Fine-tune parameters

Batch Size: Number of training examples processed at once. Large batches = more stable updates.

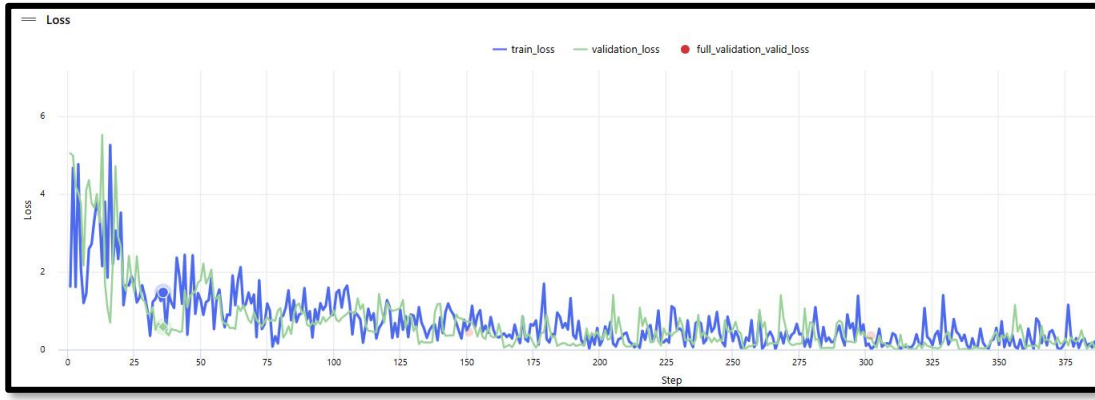
Learning Rate: Speed of learning. Higher values = faster, but riskier.

Epochs: How many times the data is used during training.

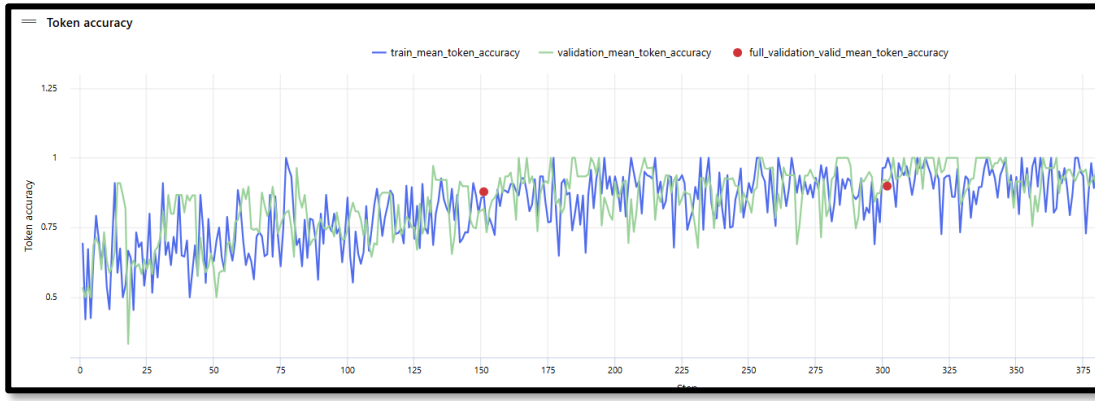
Seed: Makes the training repeatable (same input = same result).

Model for performance and fit

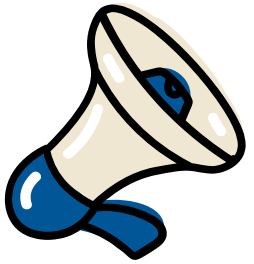
- **Loss**: Loss measures how wrong the model's predictions are. A lower loss indicates better performance. For example, a loss of 0 means perfect predictions, while a higher loss indicates more errors.



- **Token Accuracy**: Token accuracy measures the percentage of that the model predicts correctly. Higher token accuracy means the model is better at predicting the correct tokens in the output.



Auto Deployment in Developer mode



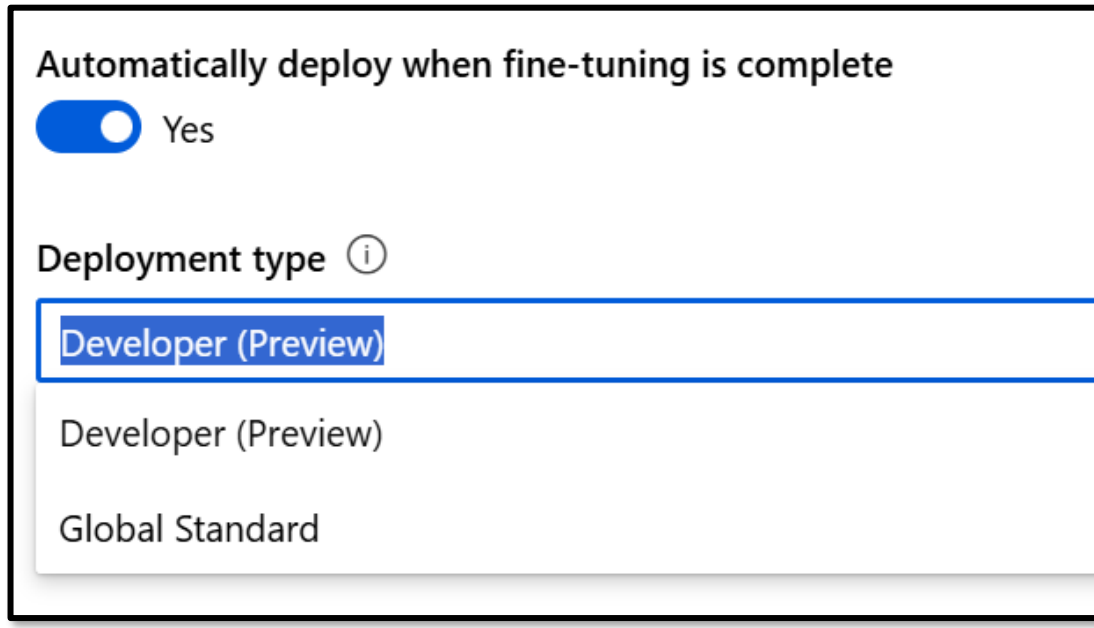
@ Build 2025

To save time, you can optionally enable auto-deployment for your resulting model.

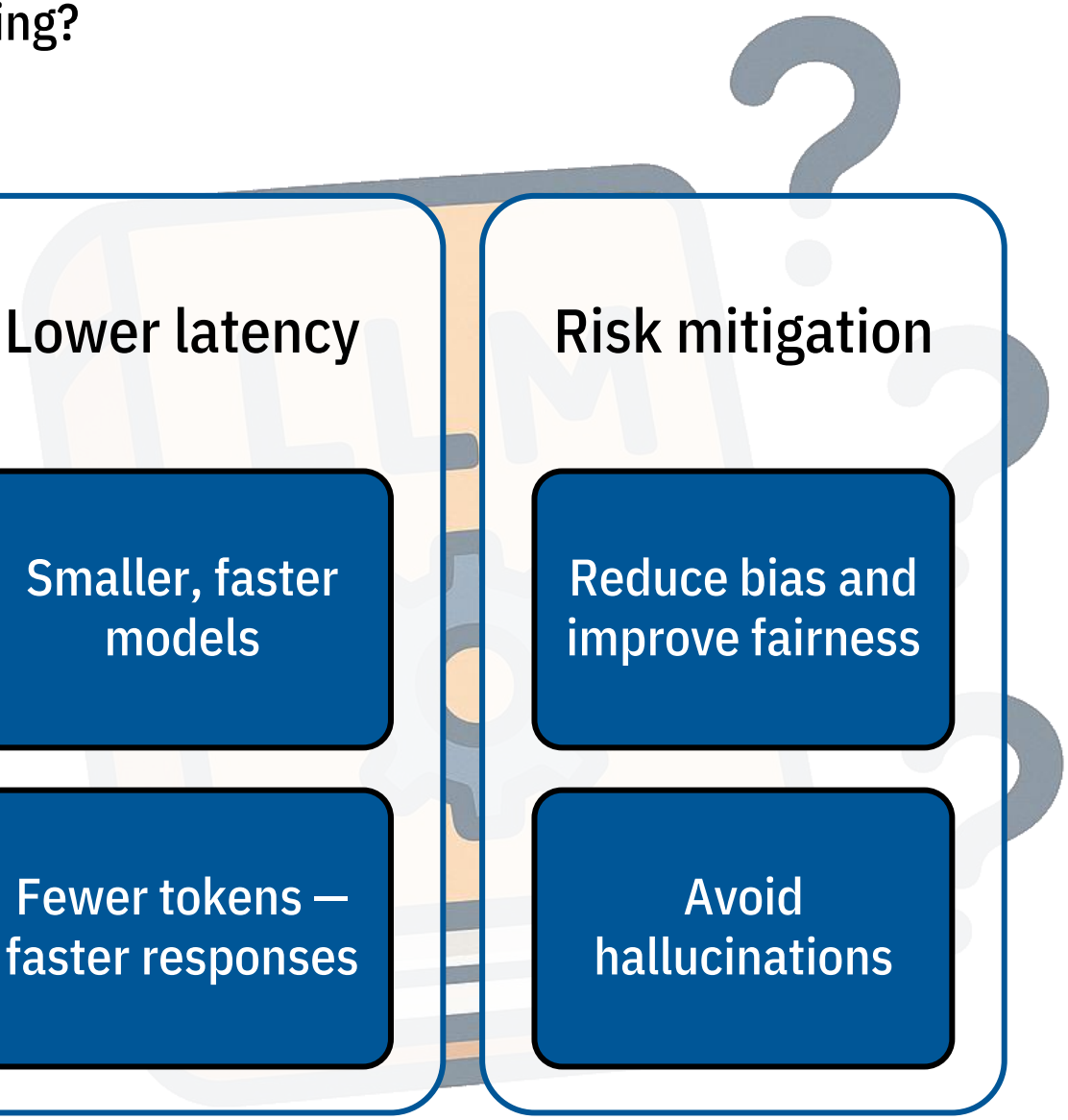
If training completes successfully, the model will be deployed using the selected deployment type.

Fine-tuned models support a **Developer** deployment that offers an affordable way to evaluate new models for a finite time paying only per-token.

It offers **no data residency guarantees nor does it offer an SLA** but it is **cheaper**!!!



Why Fine-Tuning?



Enhance performance and accuracy

- Domain-specific customization
- Task-specific optimization

Cost reduction and efficiency

- Reduced token consumption
- Efficient resource utilization

Lower latency

- Smaller, faster models
- Fewer tokens — faster responses

Risk mitigation

- Reduce bias and improve fairness
- Avoid hallucinations

What approach.....

PE
Steer model with a few examples

PE RAG
Simple & quick implementation

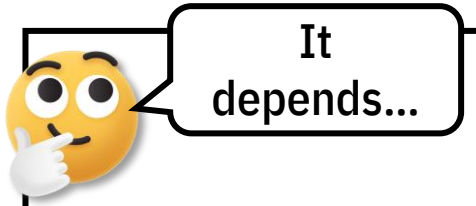
RAG
Improve model relevancy

RAG
Up to date information

RAG
Factual grounding

FT
Optimize for specific tasks

FT
Instructions won't fit in a prompt



Lower costs

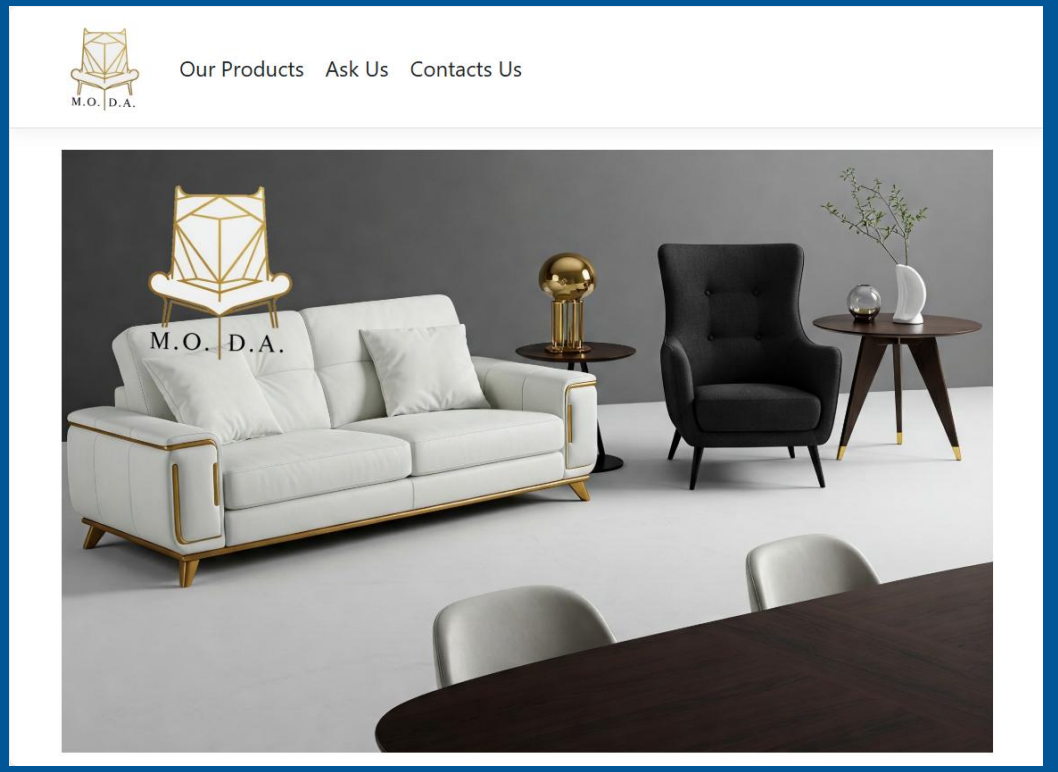
PE RAG FT
Complex, novel data or domains

Prompt Engineering PE

RAG RAG

Fine-Tuning FT

DEMO



M.O.D.A.
Modern Outstanding Design Assembled

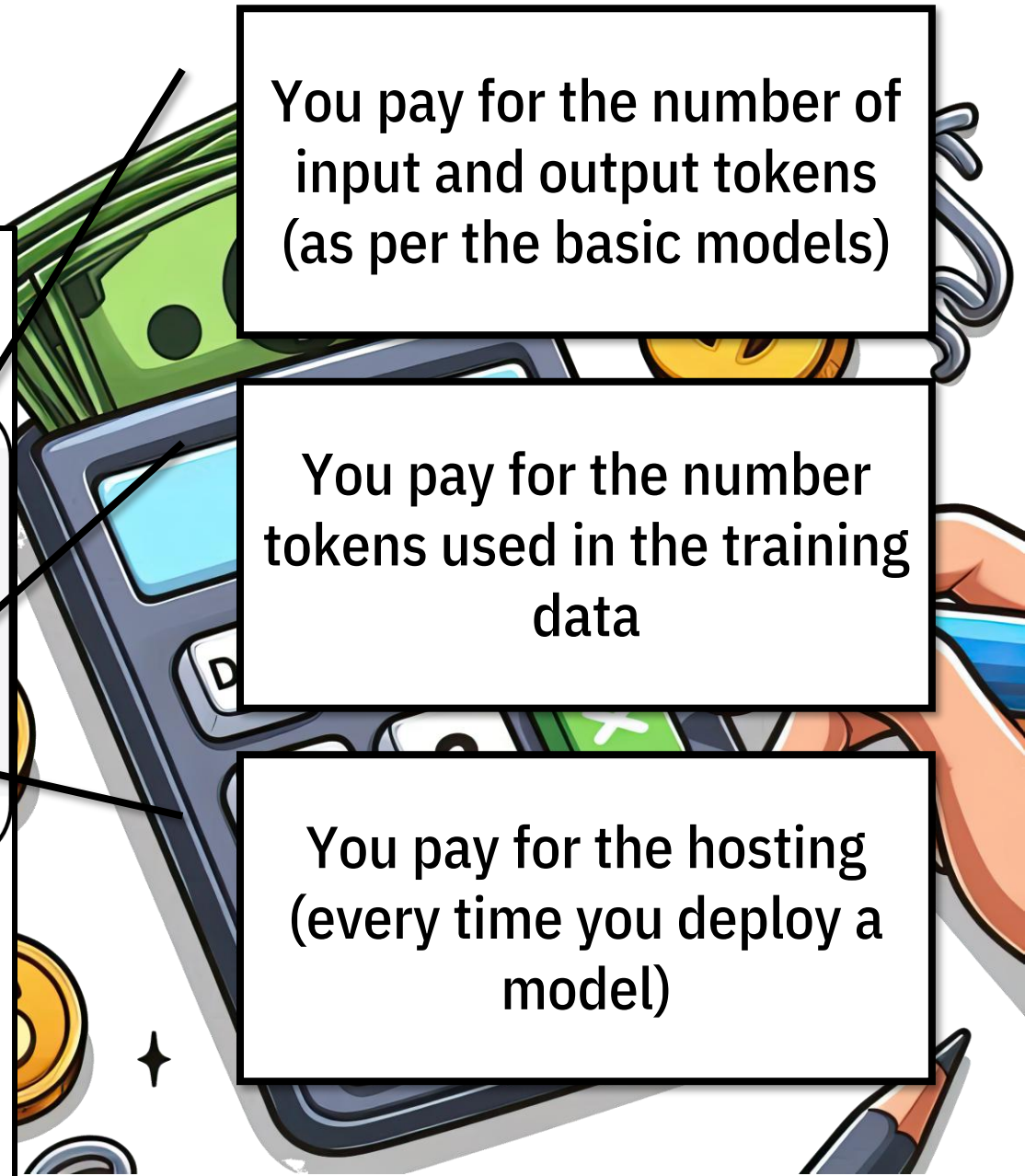
Pricing

Model		Pricing
GPT-4o-2024-08-06	Regional	Input: €2.640/1M tokens Cached Input: €1.453/1M tokens Output: €10.56/1M tokens Training: €26.45/1M tokens
GPT-4o-mini	Regional	Input: €0.159/1M tokens Cached Input: €0.088/1M tokens Output: €0.64/1M tokens Training: €3.2/1M tokens Hosting: €1.7/hour
GPT-4-0613 (8K)	Regional	Output: €57.59/1M tokens Training: €76.8/1M tokens Hosting: €4.8/hour
GPT-3.5-Turbo (16K)	Regional	Input: €0.5/1M tokens Output: €1.5/1M tokens Training: €7.68/1M tokens Hosting: €1.7/hour
GPT-3.5-Turbo (4K)	Regional	Input: €0.5/1M tokens Output: €1.5/1M tokens Training: €7.7/1M tokens

You pay for the number of input and output tokens (as per the basic models)

You pay for the number tokens used in the training data

You pay for the hosting (every time you deploy a model)



Pricing

Model		Pricing
GPT-4o-2024-08-06	Regional	Input: €2.640/1M tokens Cached Input: €1.453/1M tokens Output: €10.56/1M tokens Training: €26.45
GPT-4o-mini	Regional	Input: €0.159/1M tokens Cached Input: €0.088/1M tokens Output: €0.64/1M tokens Training: €3.2/1M tokens Hosting: €1.7/hour
GPT-4-0613 (8K)	Regional	7.59/1M tokens .8/1M tokens 8/hour
GPT-3.5-Turbo (16K)	Regional	68/1M tokens Hosting: €1.7/hour
GPT-3.5-Turbo (4K)	Regional	Input: €0.5/1M tokens Output: €1.5/1M tokens Training: €7.7/1M tokens

2.333 €

Model attributes

ID
ftjob-36cb824ebbc64ced8e20a123b1dbf3c0

Status
Completed

Base model
gpt-4o-mini-2024-07-18

Created on
Mar 22, 2025 5:22 PM

Training file
Training_QA.jsonl

Validation file
Training_QA.jsonl

Azure OpenAI Service resource
ECS2025OpenAI

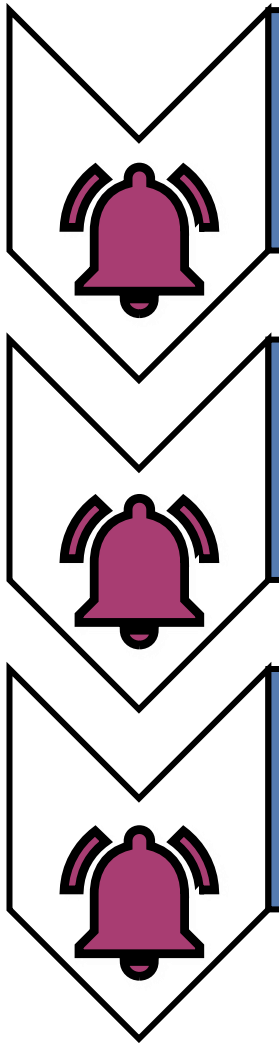
Weights & Biases integration enabled?
No

Method of Customization
Supervised

Duration
1h 26m 2s

Training tokens billed
729,000

You are **not be ready** for fine-tuning if....

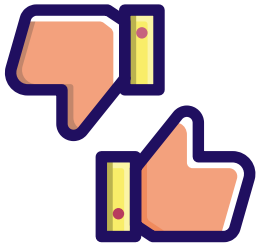


No clear use case for fine tuning, or an inability to articulate much more than "*I want to make a model better*".

If you identify cost as your primary motivator, proceed with caution

If you want to add out of domain knowledge to the model.

Pros vs Cons



	Pro(s)	Cons(s)
RAG	Cost effective Dynamic, Up-to-date Domain Flexibility	Vector Db Dependency Relies on Data Quality Introduces Latency
Fine-Tuning	Domain specialization Improved Accuracy	Higher Costs Static Knowledge

References

- Customize a model with Azure OpenAI in Azure AI Foundry Models - Microsoft Learn
- Fine-tuning and distillation with Azure AI Foundry - Build 2025 (session)
- Microsoft Build 2025 Book of News

- massimobonanni/MODA-FineTuning

THANK YOU, YOU ARE AWESOME ❤

PLEASE RATE THIS SESSION IN THE MOBILE APP.

Massimo Bonanni

Sr Technical Trainer @ Microsoft

massimo.bonanni@microsoft.com
@massimobonanni

aka.ms/maxlinkedin