



# LLM Quality Optimization Techniques

August 8, 2024



Follow along

Thierry Moreau  
Co-founder & Head of Developer Relations  
<https://www.linkedin.com/in/dr-thierry-moreau>

Alyss NoLand  
Head of Product Marketing  
<https://www.linkedin.com/in/alyssnoland>

# If this sounds familiar **then this talk is for you!**



*"My GenAI spend has gone through the roof"*



*"Quality is just not where it needs to be for us to use LLMs in production"*

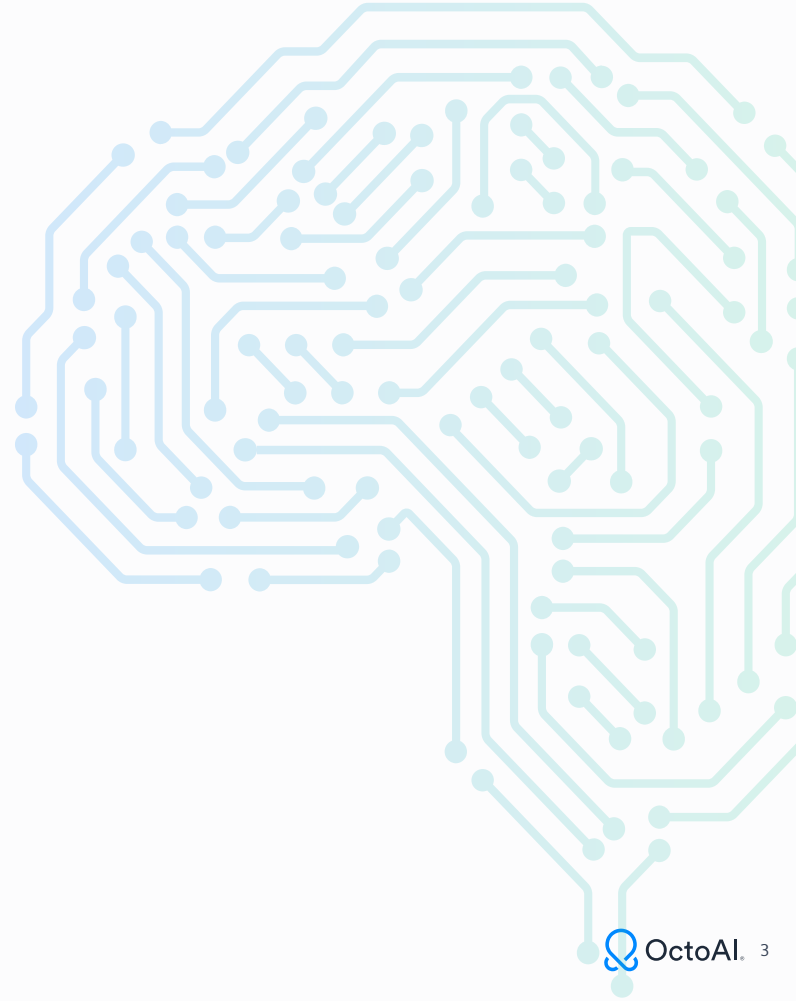
# Agenda

✓ What is LLM quality optimization?

✓ Three ways to optimize LLMs

✓ **Demo:** fine-tuning for PII redaction

✓ How to get started on your own

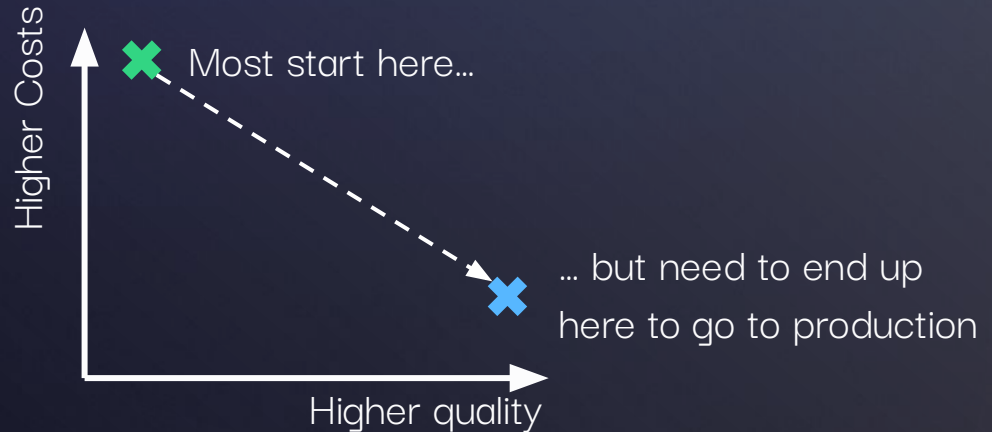


# Motivation: better quality, lower costs

## What limits GenAI adoption in most businesses today?

Limited availability of GPUs - drives \$\$\$ up

Initial PoCs don't reach the expected quality bar



# Optimizing LLM quality



**Mega Model APIs**  
The "all around" champs,  
excellent at many things

*GPT-4o, Claude 3.5,  
Llama 3.1 405b*



**Prompt Engineering**  
Increase accuracy with a clear  
target and aim

*Prompt specificity  
Few-shot prompting  
Chain of thought prompting*



**Multi-Agent Systems**  
A team of experts completes  
a complex action

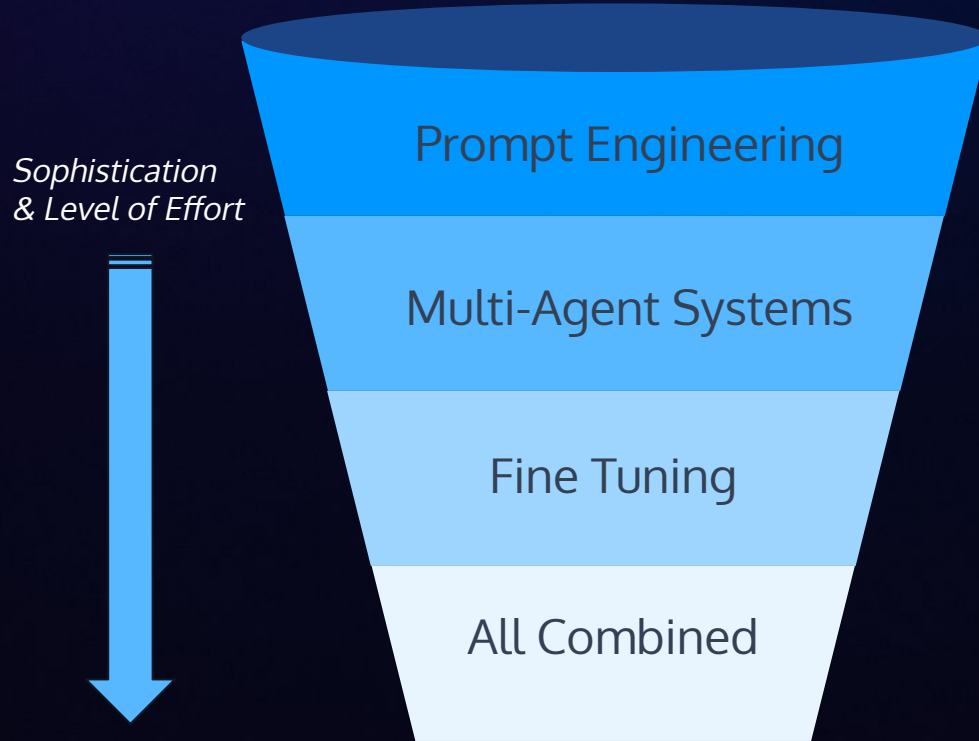
*Tool-calling, JSON, Orchestration*



**Fine-Tuning**  
Highly trained for excellence  
at one specialized task

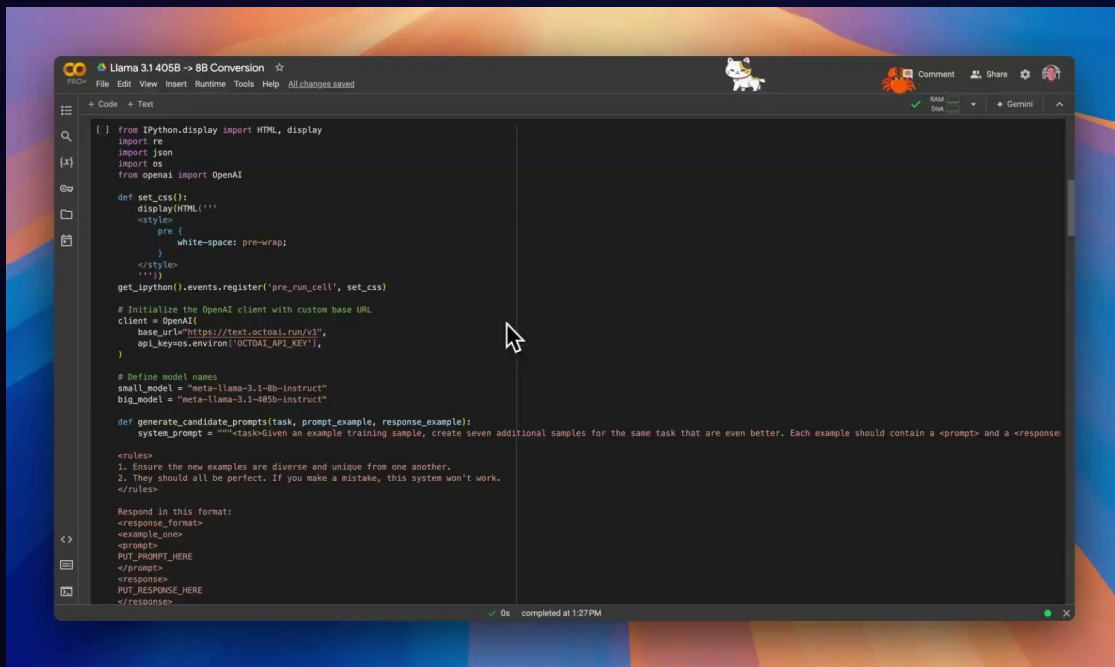
*Training dataset curation  
Model fine tuning  
Quality evaluation*

# A crawl, walk, run progression for **model optimization**



# Prompt engineering delivers big bang for your buck

## Using Llama 3.1 405b to prompt-tune 8b



```
[ ] from IPython.display import HTML, display
import re
import json
import os
from openai import OpenAI

def set_css():
    display(HTML("""
<style>
pre {
white-space: pre-wrap;
}
</style>
""))

get_ipython().events.register('pre_run_cell', set_css)

# Initialize the OpenAI client with custom base URL
client = OpenAI(
    base_url="https://text.octoai.run/v3",
    api_key=os.environ["OCTOAI_API_KEY"],
)

# Define model names
small_model = "meta-llama-3.1-8b-instruct"
big_model = "meta-llama-3.1-405b-instruct"

def generate_candidate_prompts(task, prompt_example, response_example):
    system_prompt = """<task>Given an example training sample, create seven additional samples for the same task that are even better. Each example should contain a <prompt> and a <response>

<rules>
1. Ensure the new examples are diverse and unique from one another.
2. They should all be perfect. If you make a mistake, this system won't work.
</rules>

Respond in this format:
<response_format>
<example_one>
<prompt>
PUT_PROMPT_HERE
</prompt>
<response>
PUT_RESPONSE_HERE
</response>
"""
```

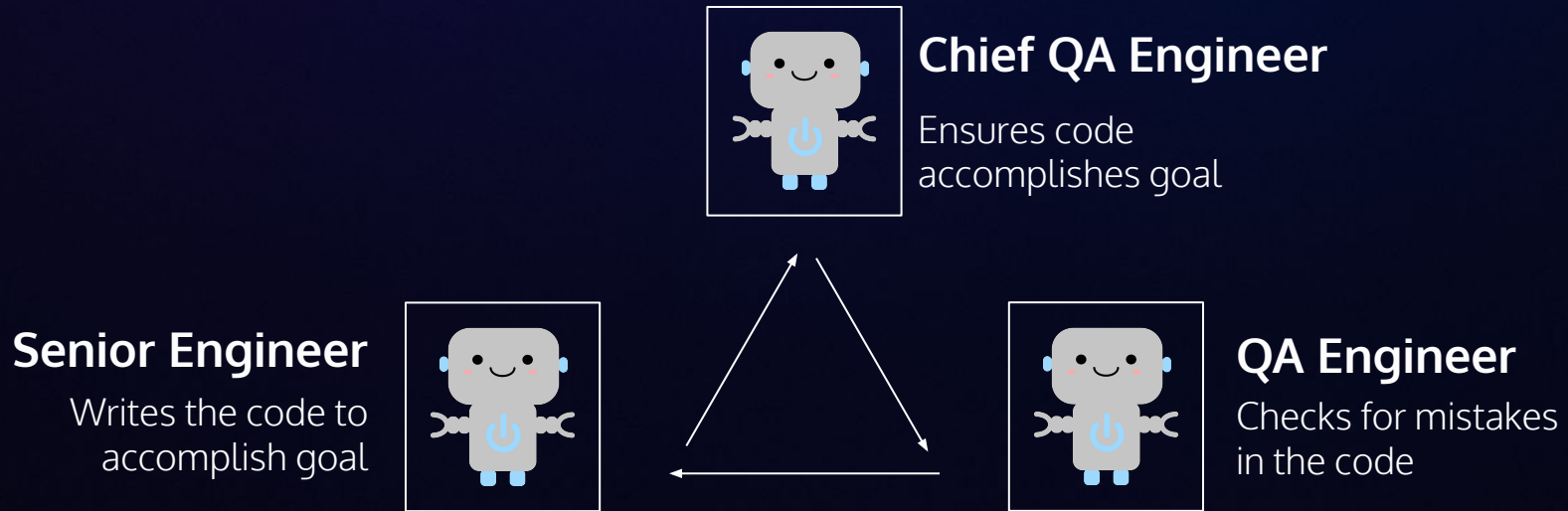
Get the quality of Llama 3.1 405B, at a fraction of the cost and latency.

Give one example of your task, and 405B will teach 8B (~30x cheaper!!) how to do the task perfectly

<https://github.com/mshumer/gpt-prompt-engineer>

[Video permalink](#)

# Multi-agent Scenario: Code Generation





# Fine-tuned Llama 3.1 8B for PII redaction

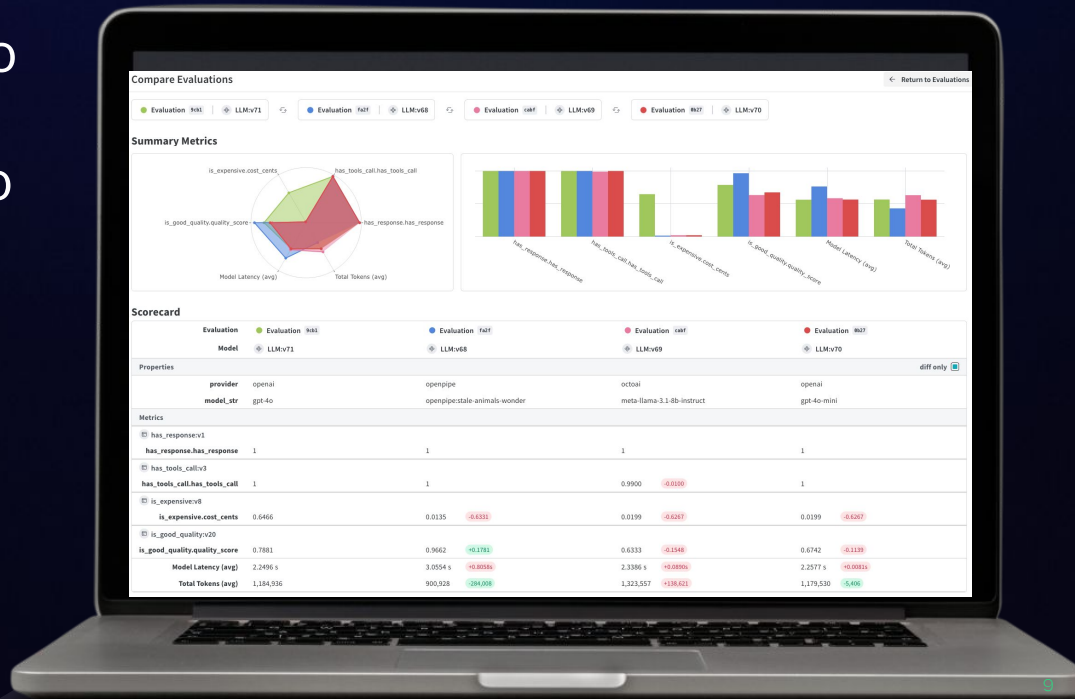
1.23x better accuracy vs. GPT-4o

25x less expensive than GPT-4o

[Dashboard link](#)

[Colab notebook link](#)

[Dataset download link](#)



# Use Case Study: LLMs for PII redaction

You are an expert model trained to redact potentially sensitive information from documents. Your goal is to accurately redact the sensitive information from the document. Sensitive information can be in one of the following categories: ...

I've noticed some unusual activities related to the credit card number **6381973478101820**. The transactions are coming from the IP address **215.114.180.213**. As our **Investor Program Supervisor**, could you please investigate?



# 1. Build the dataset

- ✓ We're using a synthetic dataset in this case
- ✓ We format each input/output pair from the dataset as logged LLM messages
- ✓ This constitutes a training dataset that we store as JSON file and upload to OpenPipe
- ✓ We produce 10k training samples and split into training - validation at 90%-10% split

## 2. Fine-tune the model

- ✓ We used OpenPipe for this step, which uses PEFT to fine-tune models
- ✓ We chose the Llama 3.1 8B base model as it's open source and small

| Overview             |  |
|----------------------|--|
| Provider             | openpipe   |
| Base Model           | Llama 3.1 8B   |
| Dataset              | <a href="#">pii-masking-10000</a>                                |
| Training Set Size    | 9,000  |
| Test Set Size        | 1,000  |
| Test Set Performance | <a href="#">View Evaluation</a>                                  |
| Training Config      | learning_rate_multiplier: 1<br>num_epochs: 1<br>batch_size: auto |
| Created At           | July 24 2:10 PM  |
| Status               | <b>DEPLOYED</b>  |
| Notes                | <a href="#">✎</a>  |

### 3. **Deploy** the fine-tune to OctoAI

- ✓ We used OctoAI to deploy our LoRA fine tune
- ✓ Export the model weights from OpenPipe to then upload to OctoAI
- ✓ Once the asset is uploaded, we can invoke it on OctoAI's Llama 3.1 8B SaaS endpoint

## 4. Evaluate the model

- ✓ We built a custom evaluation metric that uses the `privacy_mask` labels from the PII masking 200k dataset as ground truth.
- ✓ Our LLM is evaluated against that ground truth information using a scoring system
- ✓ We penalize the LLM when a PII was missed (false negative) or mistakenly added (false positive)
- ✓ We use a similarity distance metric to get a match score for each PII string-class pair (see next slide)

# 4.1 Evaluate quality

## Ground truth

```
{'fields_to_redact':  
  [  
    {  
      'string': 'Ms.',  
      'pii_type': 'PREFIX'  
    },  
    {  
      'string': 'Billie',  
      'pii_type': 'MIDDLENAME'  
    },  
    {  
      'string': '44:77:2c:cc:25:4c',  
      'pii_type': 'MAC'  
    }  
  ]  
}
```

Score: 1.0

## Fine-tune

```
{'fields_to_redact':  
  [  
    {  
      'string': 'Ms.',  
      'pii_type': 'PREFIX'  
    },  
    {  
      'string': 'Billie',  
      'pii_type': 'FIRSTNAME'  
    },  
    {  
      'string': '44:77:2c:cc:25:4c',  
      'pii_type': 'MAC'  
    }  
  ]  
}
```

Score: 0.91

## GPT-4o

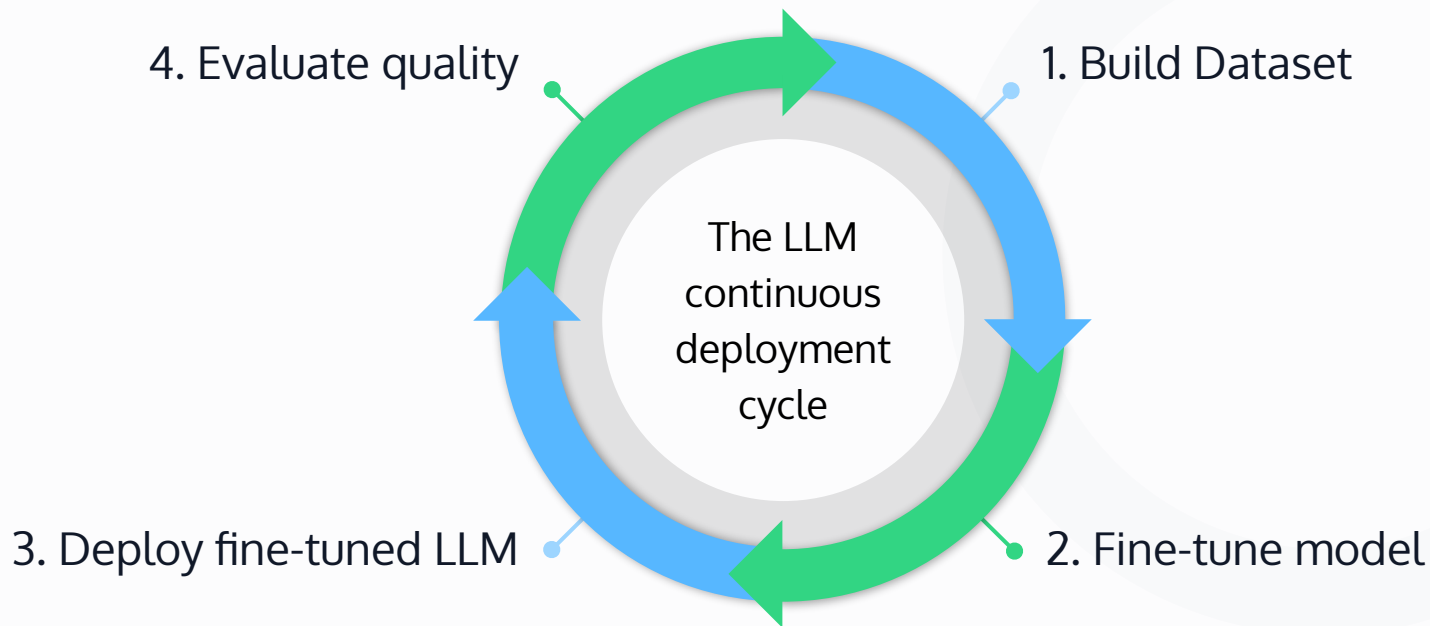
```
{'fields_to_redact':  
  [  
    {  
      'string': '44:77:2c:cc:25:4c',  
      'pii_type': 'MAC'  
    }  
  ]  
}
```

Score: 0.33

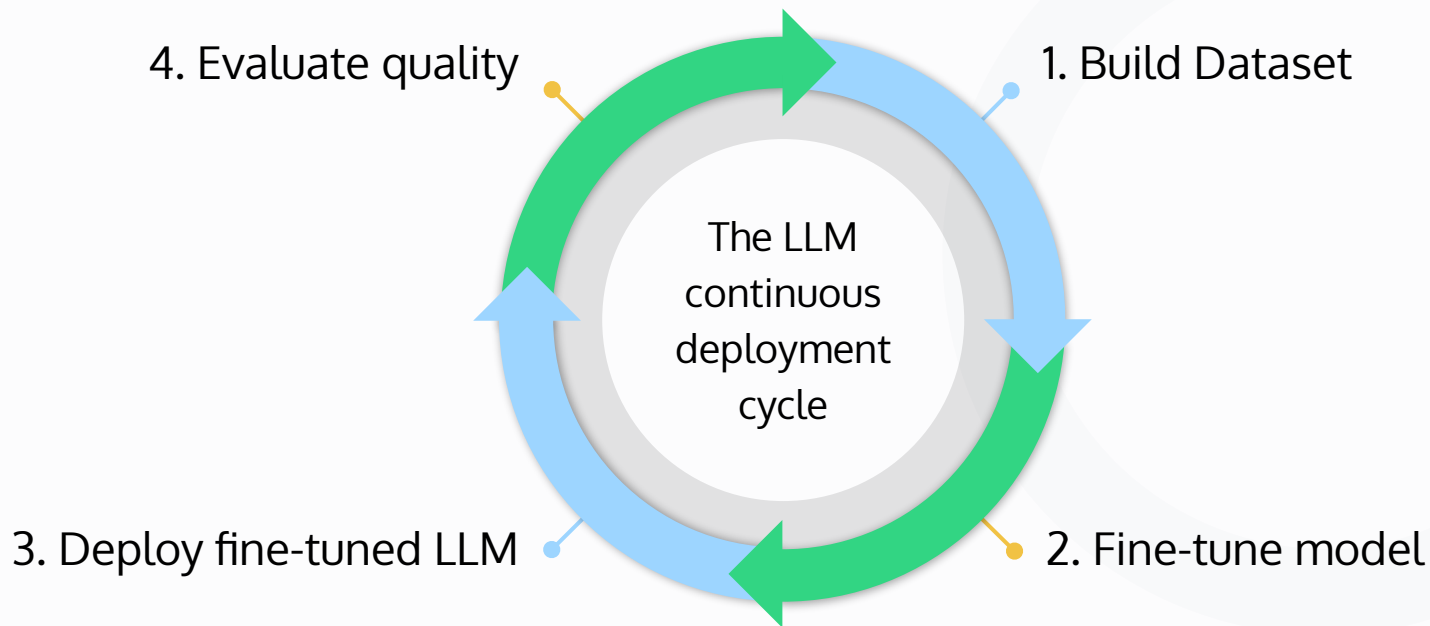
# Dashboard walkthrough



# LLM fine-tuning **continuous deployment cycle**



# There is truly **no finish line**...



# Start Optimizing LLMs with OctoAI



## Guided No-Cost Proof of Concept (>1B tok/d)

- Tune a Llama 3.1 (or any OSS LLM) on us
- Complimentary inference credits on OctoAI's serverless endpoints
- Hands-on consultation to achieve quality, cost, and performance goals
- Optionally deploy in your environment/on-prem
- Let's talk [octo.ai/contact-us](https://octo.ai/contact-us)



## Self-Starter Optimization (<1B tok/d)

- Step-by-step optimization tutorial
- Complimentary tuning & inference credits
- Live technical (Intercom) & community support (Discord)
- Check your email for details



# Q&A Time!

Drop your questions in the designated Q&A section