Building Next-Generation Machine Learning Applications

Shreya Shankar and Aditya G. Parameswaran
{shreyashankar,adityagp}@berkeley.edu

EPIC DATA

Introduction

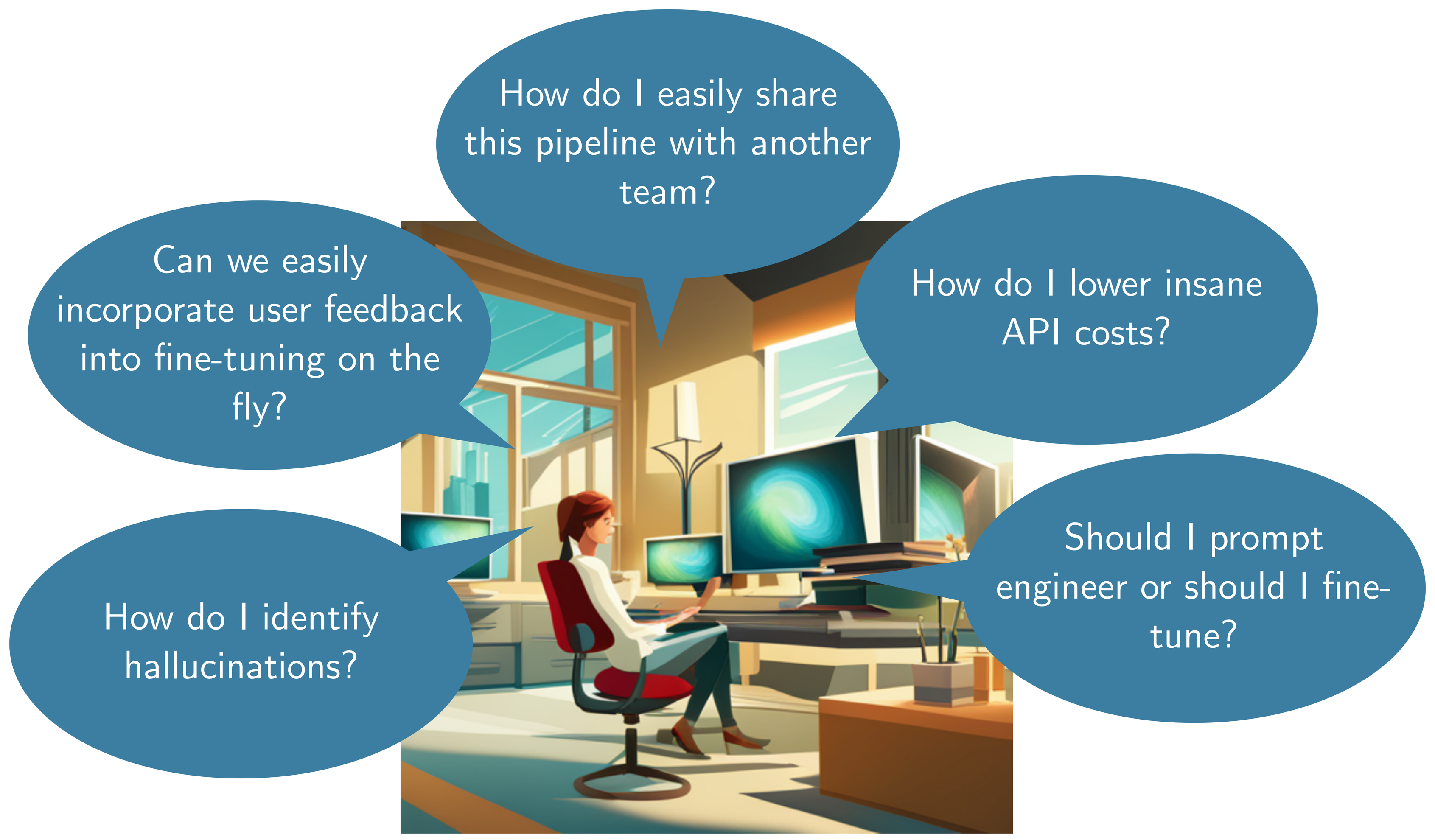
We are in a new era of AI. People can use big ML models without significant data or ML expertise!

While it's easier than ever to create a prototype or demo, it's difficult to translate this into a production application.

Our goal is to enable citizen software developers to build ***production-ready*** *ML applications with* ***minimal post-deployment hassle.***

Pain Points

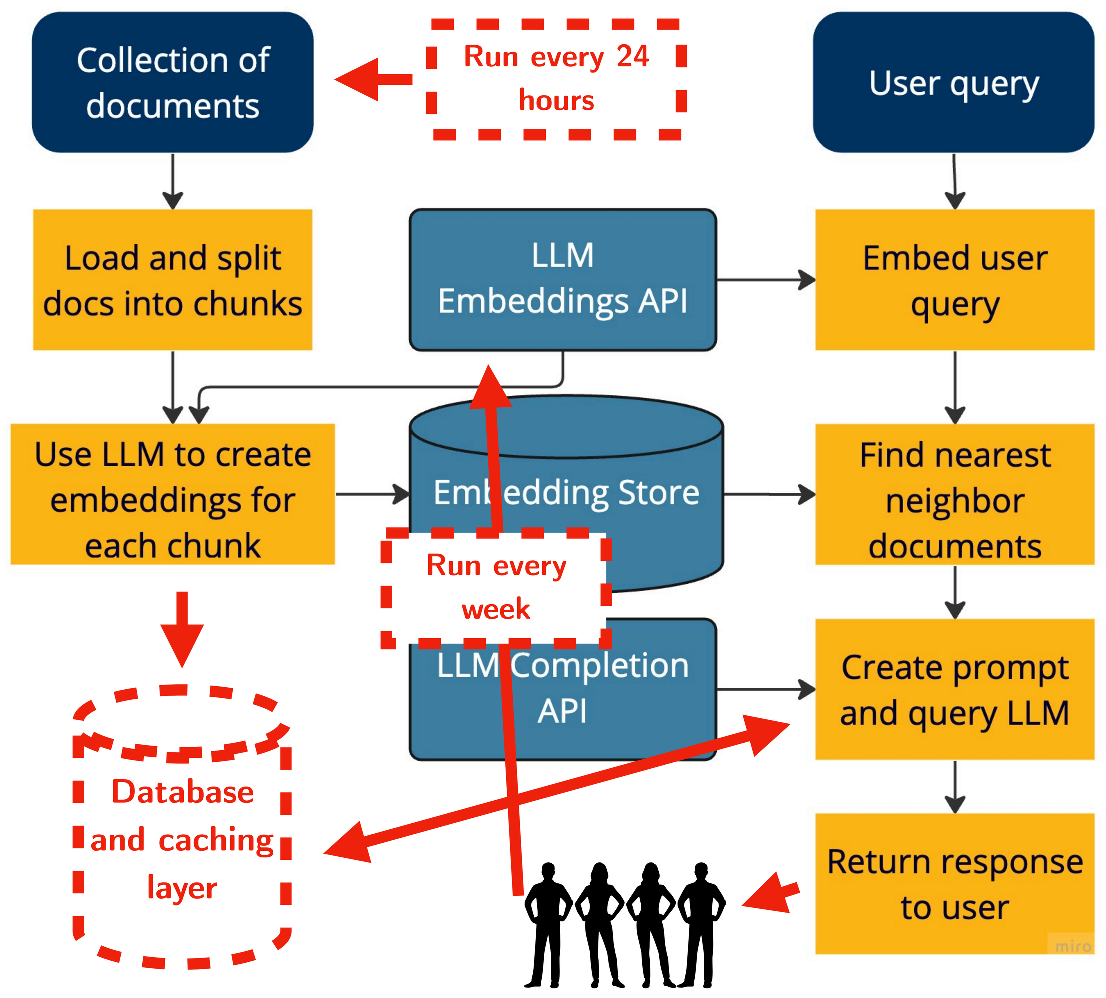
Existing frameworks to develop production-grade ML pipelines are full of tricky issues.



Current Tools Produce Messy, Ad-Hoc Pipelines

Existing tools are built for developing ML applications with *static data* assumptions. But production is dynamic!

This leads to redundant computation, high costs, and many MLOps headaches.

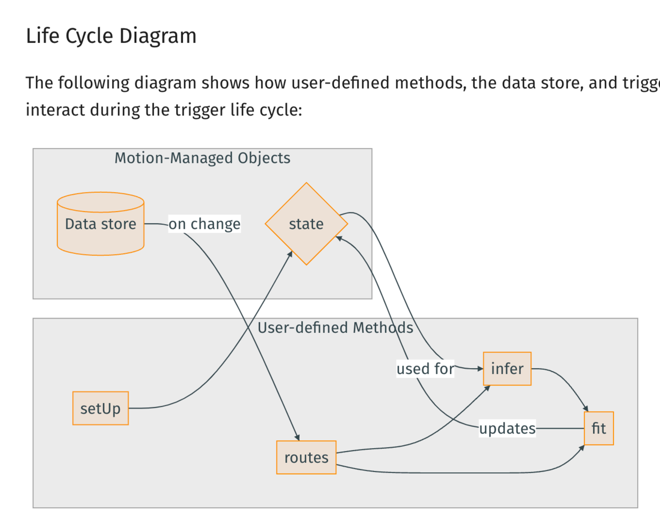


Motion: A New Framework

Our Python framework, Motion, allows developers to build ML applications with **continually-updating state**. Motion listens for changes in data and runs ML-specific logic as triggers. Building an application in Motion consists of these steps:

1. Define data *relations*, with their corresponding schemas
2. Define *triggers* to run when relations are updated. Triggers have setUp (initialize state), infer (state read-only), and fit (state write-allowed, runs in background) operations.
3. Deploy!

Motion coordinates trigger state and schedules operations.



What You Get With Motion

	Traditional Workflow	Motion Workflow
Pre-Deployment	**Low upfront effort** 🚀 ● Flexibility to look at and operate on full batches of data ● No need to specify data and dependencies ● No need to think about fine-tuning	**Higher upfront effort** 👷 ● Must define schema ● Must separate logic into state read-only and write-allowed (infer vs fit)
Post-Deployment	**High ops effort** 😩 ● Need to rewrite existing pipelines when adding new functionality (e.g., ingesting new documents, fine-tuning) ● Need to validate data and monitor for shift ● Need to coordinate different jobs	**Low ops effort** 🤑 ● Can add new functionality without modifying existing pipeline code ● Data is type-checked, validated, and monitored ● All jobs done on one machine (unless explicitly outsourced in infer or fit methods)

Work In Progress

Improving Experimentation Support 🧪

- Allowing users to easily answer, "Should I prompt engineer or should I fine-tune?"

Auto-Refit on Data Drift 🔄

- Profiling summaries of data within relations to check for data drift
- Trigger a state update when summaries have changed

Check Out Our Work!

Motion Docs: https://dm4ml.github.io/motion/
Motion Github: https://github.com/dm4ml/motion

Shankar, S., Fawaz, L., Gyllstrom, K., & Parameswaran, A. G. (2023). Moving Fast With Broken Data. *arXiv preprint arXiv:2303.06094.*
Shankar, S., Garcia, R., Hellerstein, J. M., & Parameswaran, A. G. (2022). Operationalizing machine learning: An interview study. arXiv preprint arXiv:2209.09125.
Shankar, S., & Parameswaran, A. (2021). Towards Observability for Production Machine Learning Pipelines. *arXiv preprint arXiv:2108.13557.*