

Can We Make Model Alignment a Software Engineering Process?

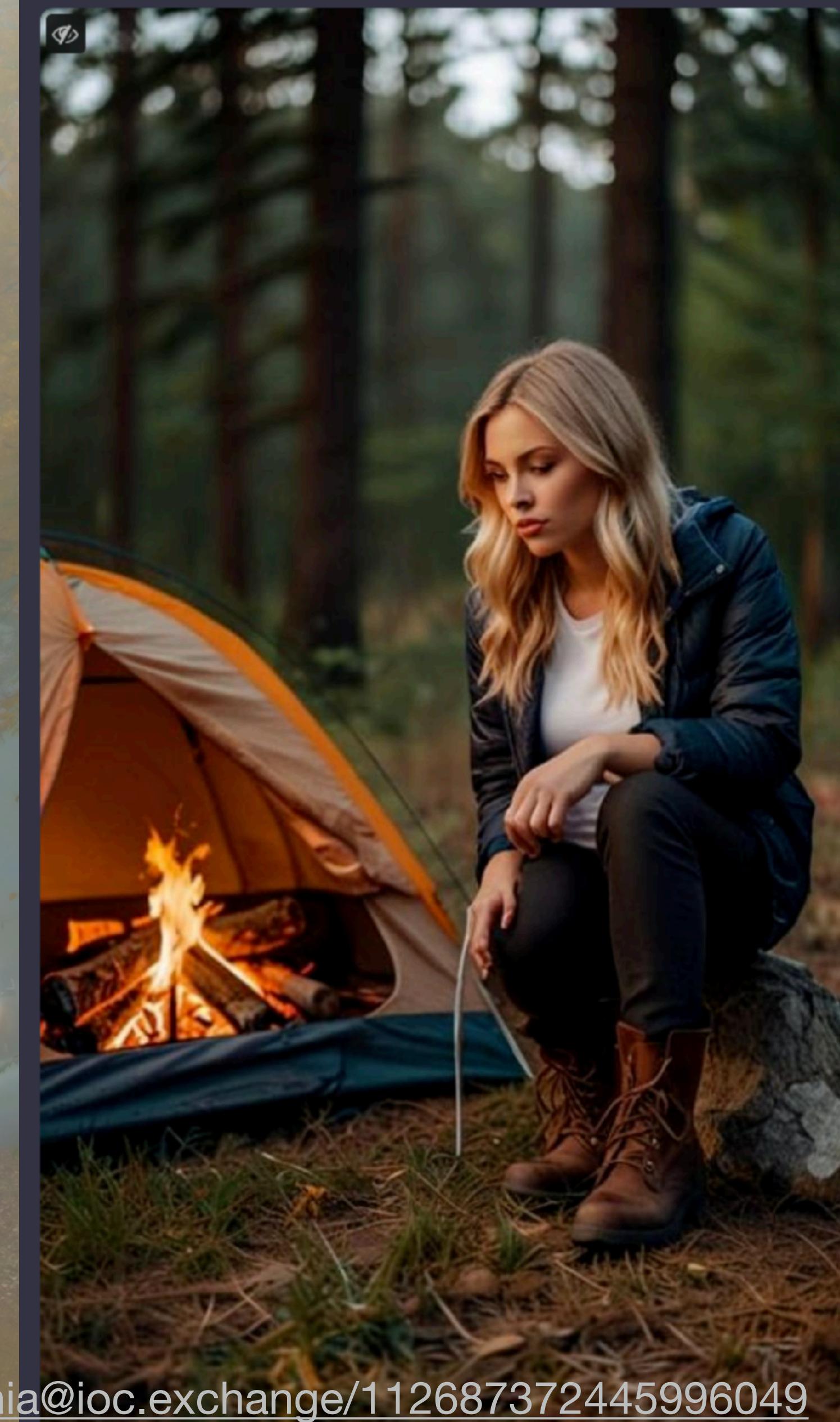
The AI Conference, San Francisco
September 2024
Dean Wampler, Ph.D.
The AI Alliance and IBM Research
thealliance.ai

deanwampler.com/talks



About the Images.

I used Adobe Firefly to “enhance” my real photographs.



AI and Software Engineering?

- Two topics:
 1. Can we make model alignment (e.g., tuning) more iterative and incremental?
 2. Automated testing of probabilistic systems is dang hard!



AI Alliance

thealliance.ai

Our core beliefs in AI that is open is the tie that binds us, despite our differences.

Member organizations from academia, commercial, research and non-profits and span the globe.

+120 organizations in +20 countries, and growing



thealliance.ai

U.S. - Indiana
• University of

U.S. - Utah
• University

U.S. - Ohio
• Cleveland

U.S. - Connecticut
• Yale University

U.S. - New Hampshire
• Dartmouth

France
• Institut Polytechnique de Paris

Germany
• TU Munich

Six Focus Areas:

1. Education and research
2. Trust and safety
3. Tools for building models and apps
4. Hardware portability
5. Open models and datasets
6. Policy and advocacy

Spreading knowledge, research

Technical initiatives

Effective, measured regulations

• National Science Foundation*

U.S. - North Carolina
• Red Hat

Ontocord.AI
• Simons Foundation & Flatiron Institute

for Theoretical Physics
• International School for Advanced Study

Zayed University of Artificial Intelligence
• Core42

Australia
• Fast.ai

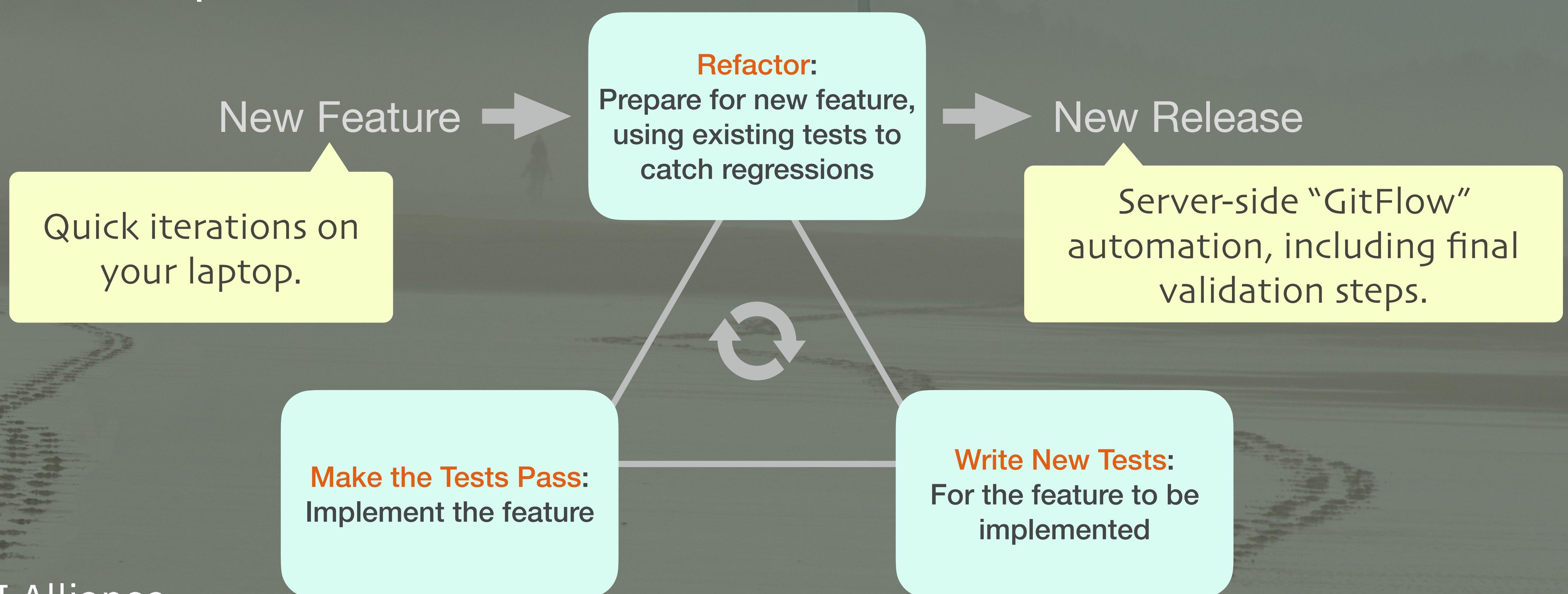
Taiwan
• MediaTek Research

Iterative and Incremental Model Tuning

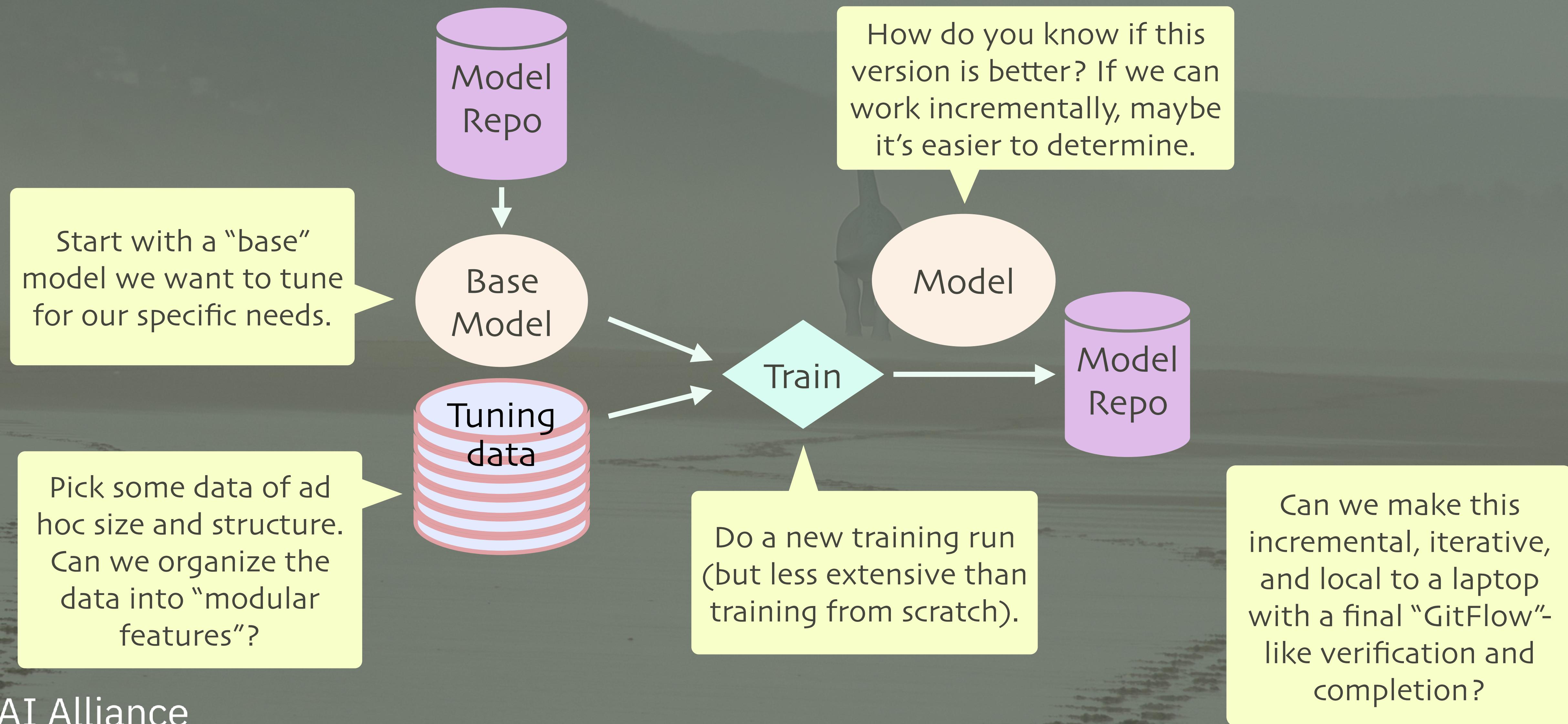


What Software Developers Like

- Features are implemented and released incrementally.
- The process is iterative.



What Model Tuning Is Often Like



One Approach: InstructLab

Open sourced by
IBM and Red Hat

<https://github.com/instructlab>

The screenshot shows the GitHub repository page for InstructLab. The repository has a blue icon featuring a white dog wearing glasses. The name 'InstructLab' is displayed in bold black text. Below it, there are 1.7k followers and a link to the project's website at <https://instructlab.ai/>. The main navigation tabs are Overview (selected), Repositories (16), Discussions, Projects (1), Packages, and People. The 'Overview' tab displays the README.md file, which contains the text 'Welcome to the 🐕 InstructLab Project'. It features a cartoon dog icon and a large blue graphic. The description below states: 'InstructLab is a model-agnostic open source AI project that facilitates contributions to Large Language Models (LLMs).'

Open sourced by
IBM and Red Hat

<https://github.com/instructlab>

InstructLab

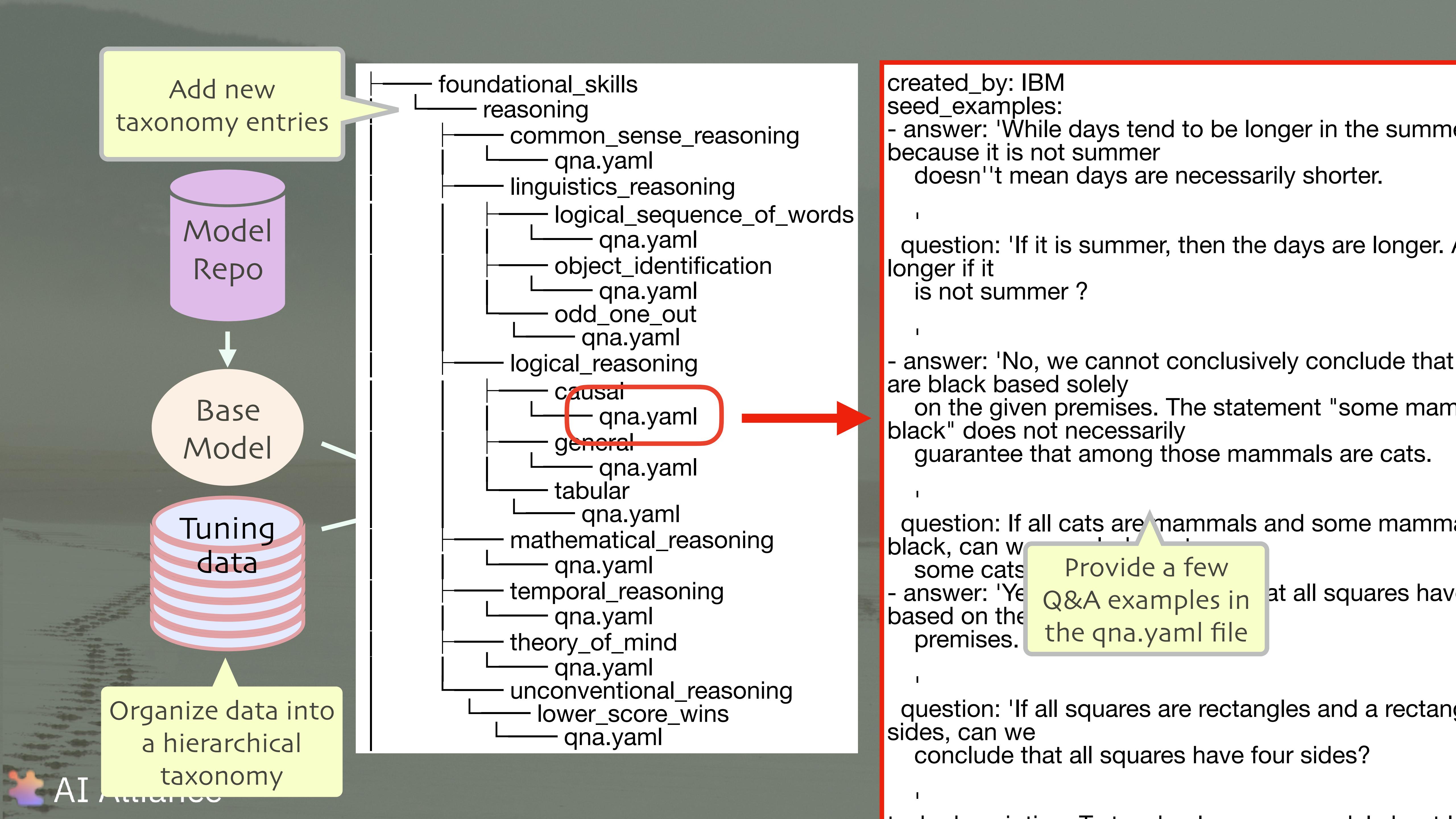
1.7k followers <https://instructlab.ai/>

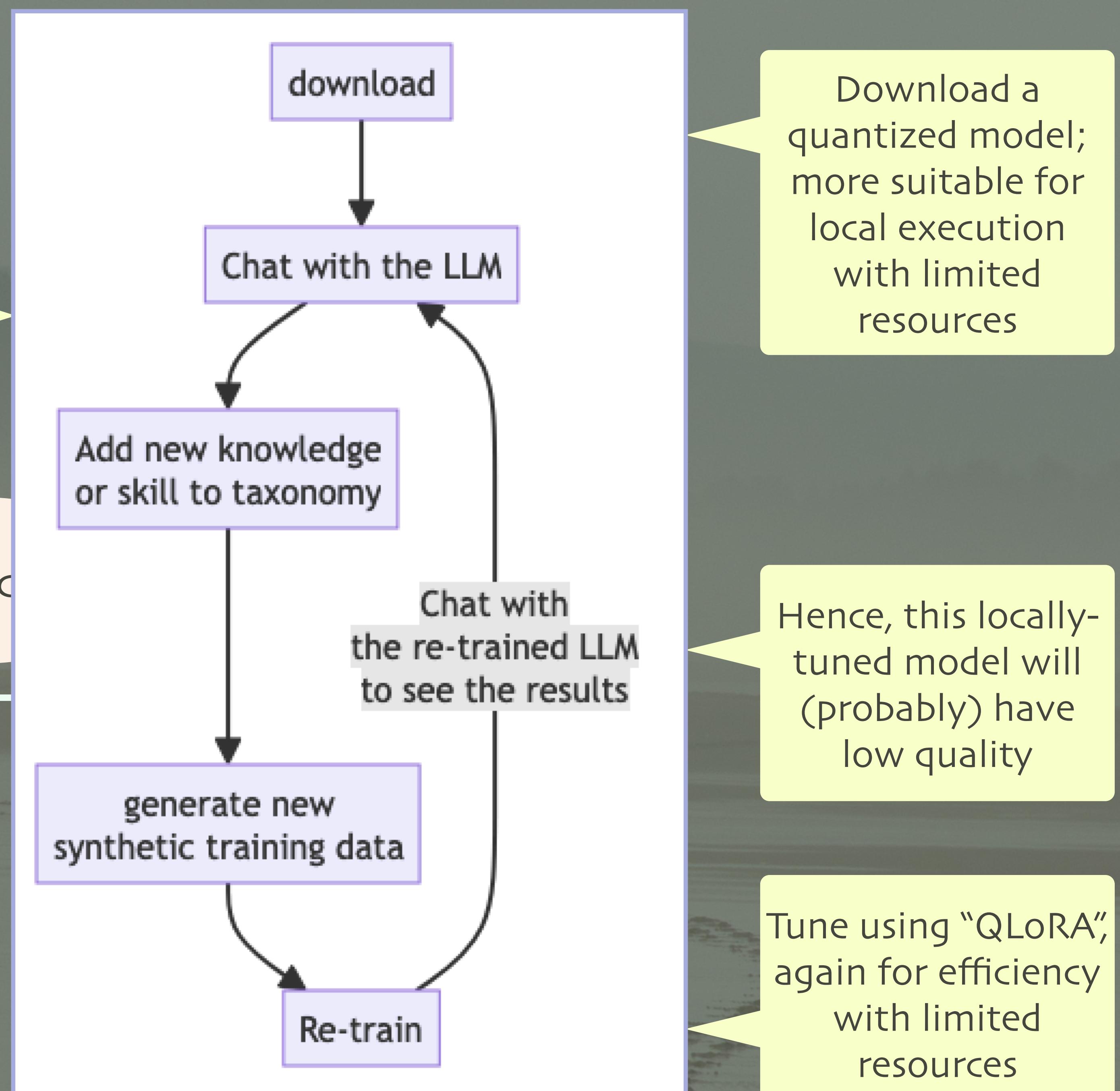
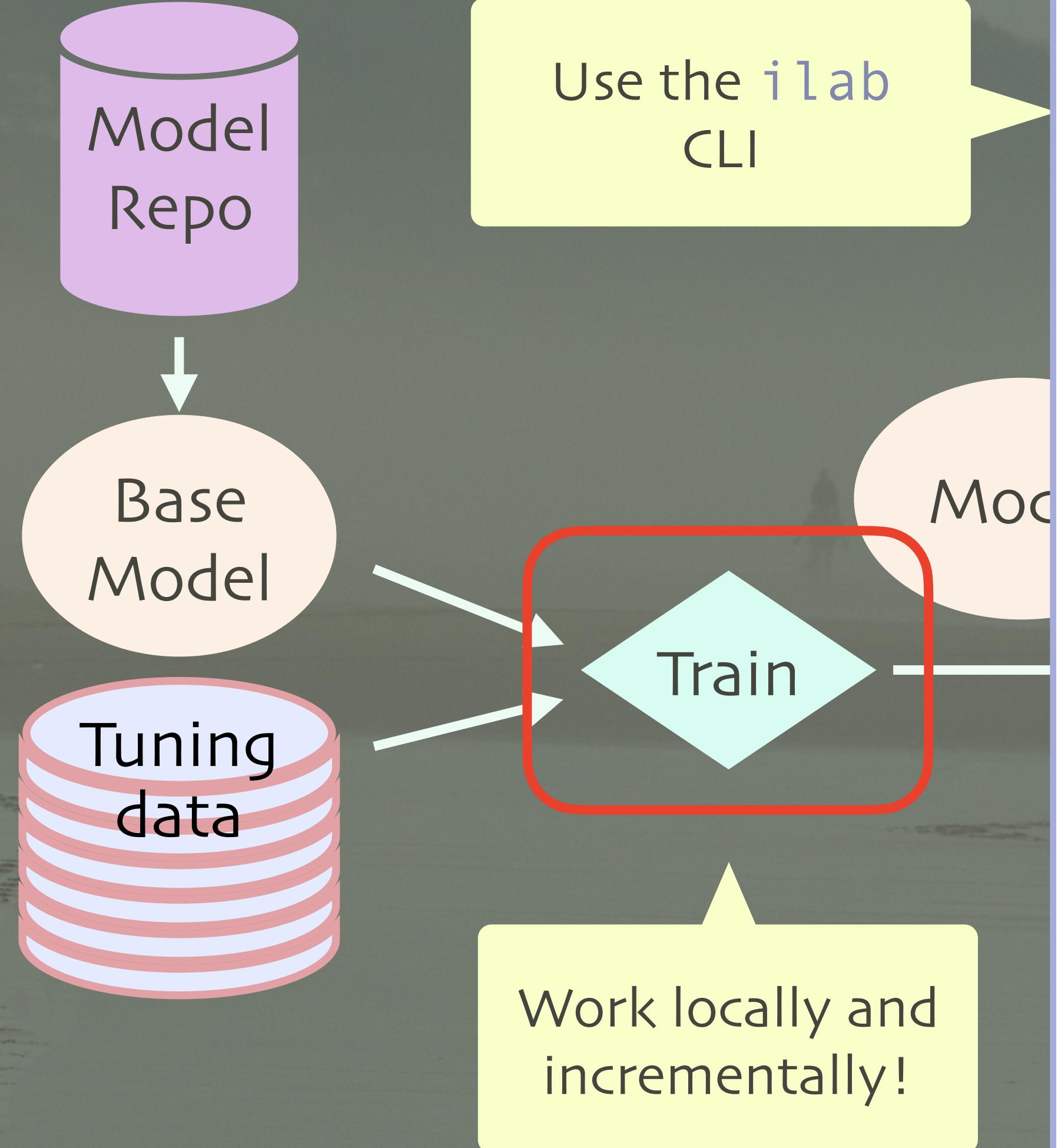
Overview Repositories 16 Discussions Projects 1 Packages People

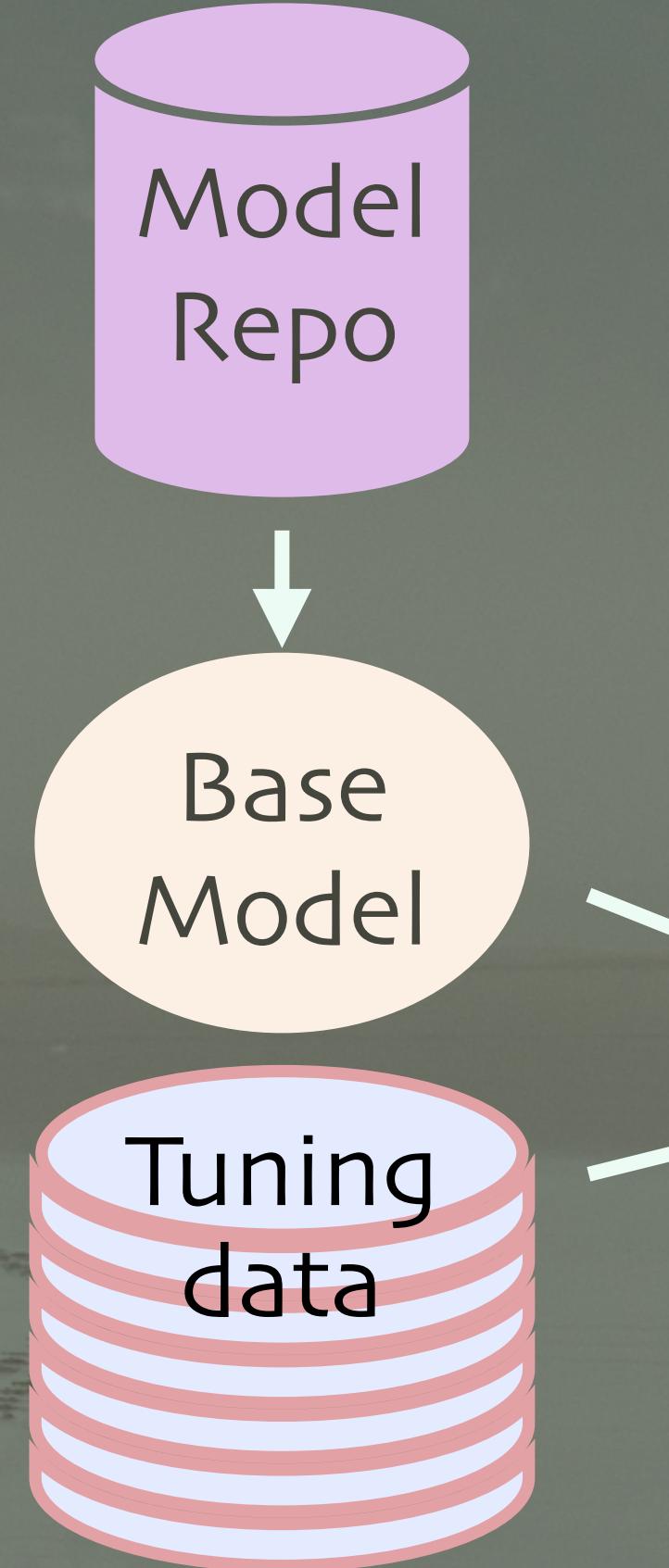
README.md

Welcome to the 🐕 InstructLab Project

InstructLab is a model-agnostic open source AI project that facilitates contributions to Large Language Models (LLMs).







```
foundational_skills
  reasoning
    common_sense_reasoning
      qna.yaml
    linguistics_reasoning
```

created_by: IBM
seed_examples:
- answer: 'While days tend to be longer in the summer because it is not summer doesn't mean days are necessarily shorter.'

Once you are satisfied, issue a pull request for the taxonomy changes **only**.

A GitFlow process repeats the data synthesis and tuning steps with a larger, more powerful teacher model and unquantized base model, etc., etc.

```
qna.yaml
theory_of_mind
  qna.yaml
  unconventional_reasoning
    lower_score_wins
      qna.yaml
```

based on the given premises.

question: 'If all squares are rectangles and a rectangle has four sides, can we conclude that all squares have four sides?'

InstructLab

Cons (1/2)

- Testing!
 - Supports a combination of standard benchmarks and chat (“try it out”), but...
 - Still need “real” test-driven development.
 - It’s still easy to miss regressions, like older, unchanged taxonomy areas get worse!
 - We’ll return to this topic...

InstructLab

Cons (2/2)

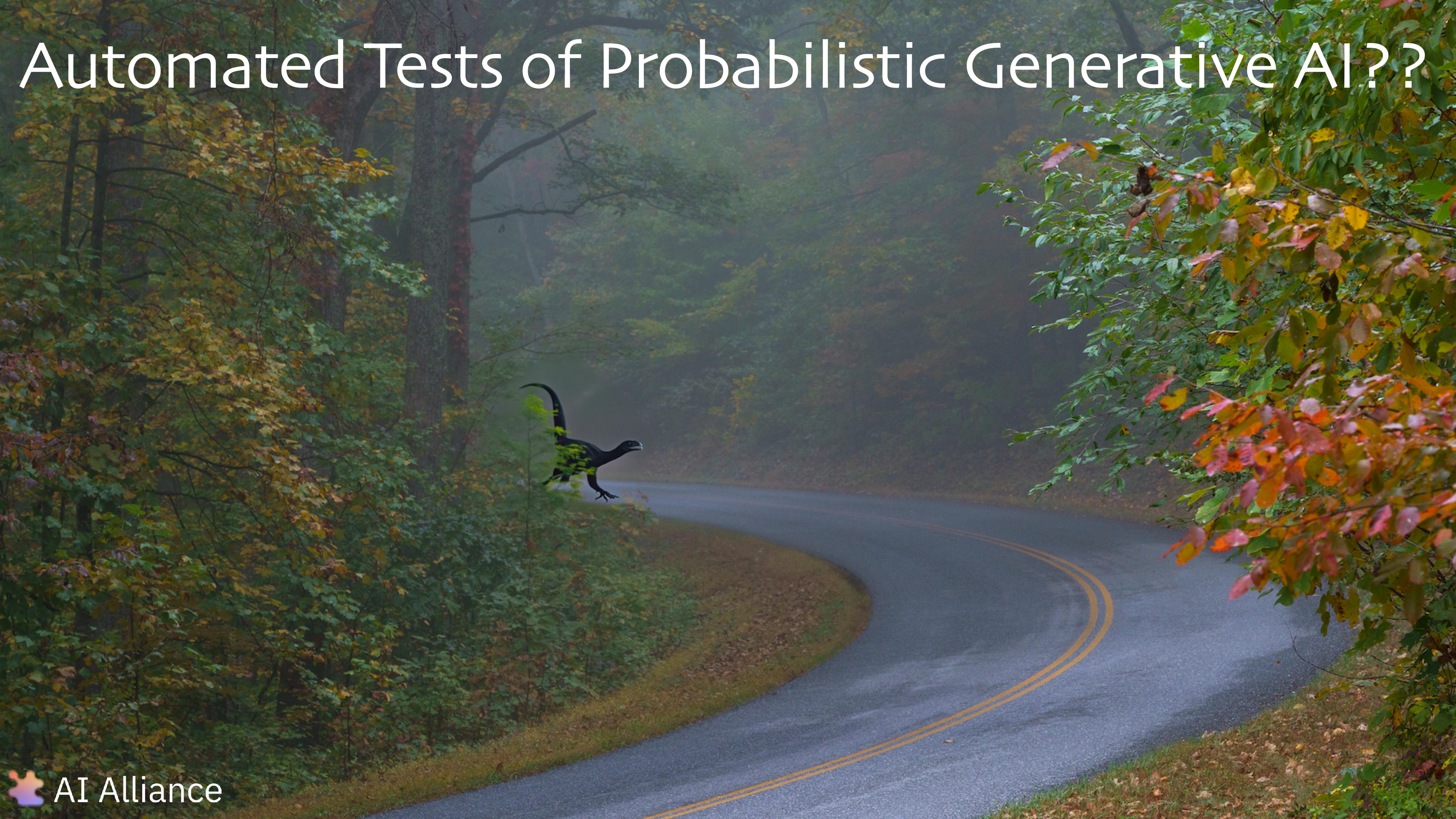
- You still need a server-side final tuning stage.
- The InstructLab project is setting up a community collaboration on public models where they provide the “GitFlow” process for PRs.
- But for your private needs, you still need the ability to do this server-side tuning yourself.
- It might be too expensive to tune on **each** PR.

InstructLab

Pros

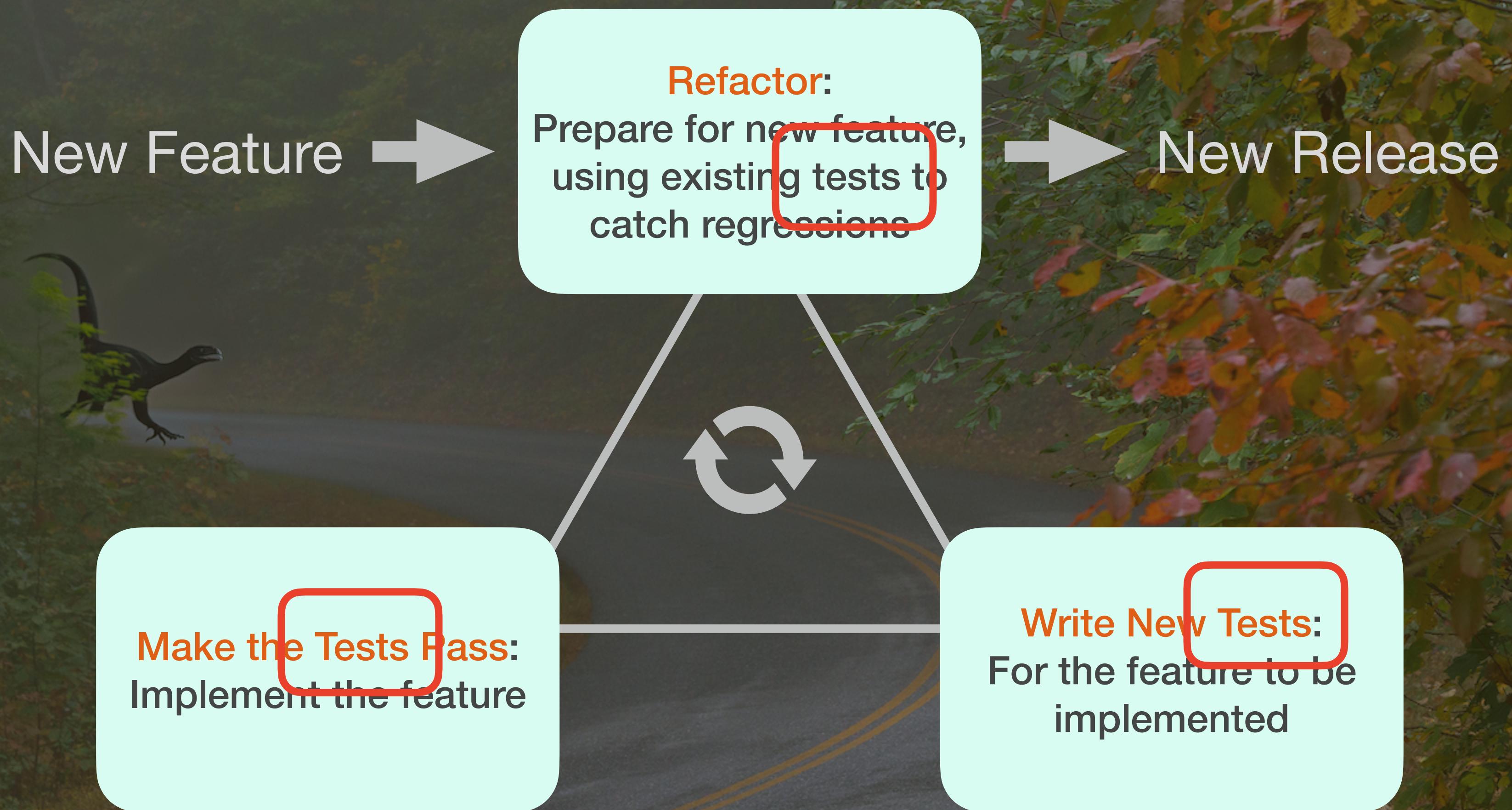
- Defines useful conventions for the taxonomy structure and Q&A examples for each taxonomy topic.
- i lab hides and automates much of the complexity for all the steps.
- You can work locally, incrementally, and iteratively!

Automated Tests of Probabilistic Generative AI??



Automated Tests of Probabilistic Generative AI??

Remember this?



Testing is integral to this process.

What Do Developers Expect?

Developers expect software to be deterministic[‡]:

- The same input → the same output.
- e.g., $\sin(\pi) = -1$
- The output is different? Something is broken!
- Developers rely on determinism to help ensure correctness and reproducibility, to catch regressions!

What Do Developers Expect?

Developers expect software to be deterministic[‡]:

- The system always produces the same output given the same input.
- e.g. $2 + 2 = 4$
- The code is predictable.
- Developers can reason about the correctness of their code.

Put another way, the determinism makes it easier to specify the *system* invariants. What should remain true before and after some step?

oken!
ensure

What's new with Gen. AI?

Generative models are probabilistic[‡]:

- The same prompt → **different** output.
- `chatgpt("Write a poem")` → **insanity**
- Without determinism, how do you write repeatable, reliable tests? Specifically,
- Is that new model actually **better** or worse than the old model?
- Did any **regressions** in other behavior occur?

“Insanity is doing the same thing over and over again and expecting different results.”
— not Einstein

What's new with Gen. AI?

Generative models are probabilistic[‡]:

- The model can change its mind.
- With enough data, it can repeat itself.
- Is there a bug in the model?

Put another way, the **invariants** are much less **clear** and therefore much less **enforceable**.

- Did any regressions in other behavior occur?

[‡] A tunable “temperature” controls how probabilistic.

Automated Tests of Probabilistic Generative AI??

- What about benchmarks?
 - Most existing benchmarks aren't specific enough, like good tests should be.
 - They aren't written like typical tests.
 - However, more specific, use case focused benchmarks should help.

Automated Tests of Probabilistic Generative AI??

- What about using a “critic” model or a “voting panel” of models as experts?
- The largest LLMs are being used this way.
 - For example, InstructLab uses a powerful teacher model to check the generated synthetic data and reject bad Q&A examples.

Automated Tests of Probabilistic Generative AI??

- Can Data Science help?
- We developers could use help from you data scientists to build statistically-appropriate testing techniques.

What other ideas to you?
What techniques have you tried?

Thank you!

thealliance.ai

dwampler@thealliance.ai

Mastodon and Bluesky: @deanwampler

deanwampler.com/talks



Notes

© Text 2023-2024, Dean Wampler, © Images 2004-2024, Dean Wampler, except where noted. Most of the images are based on my photographs (flickr.com/photos/deanwampler/), but they are manipulated by AI to add “new content”:

1. The title image is adapted from [from this image](#) taken on a foggy day on the Blue Ridge Parkway.
2. The “Automated testing” image is from the same foggy day on the Blue Ridge Parkway (not on Flickr).
3. The “Iterative and Incremental Model Tuning” image is based on [this image](#) from the Oregon coast of a real horse and rider in the fog.
4. The “Thank you” slide uses [this Chicago Park image](#).