Practical Guidance on Building Reasoning Capabilities using Distillation and Reinforcement Learning

Monday, Aug 4 & 1:00 PM – 4:00 PM


TORONTO
KDD2025

Tutorial Abstract

With reasoning models like DeepSeek-R1 and OpenAI's o1 demonstrating breakthrough capabilities in complex problem-solving, there's growing interest in the AI community about how to unlock similar capabilities in other large language models (LLMs).

This hands-on tutorial dives into practical methods for building reasoning capabilities in LLMs through two primary approaches:

Knowledge Distillation: Transferring capabilities from advanced reasoning models
Reinforcement Learning: Further enhancing capabilities through post-training techniques

Participants will learn how to transfer reasoning capabilities from cutting-edge models like DeepSeek-R1 into smaller LLMs such as Qwen and Llama, and then explore how reinforcement learning can take these capabilities even further.

Our Team

Tutorial Authors

Alex Qiu
Solution Architect,
NVIDIA

Lark Zhang
Solution Architect,
NVIDIA

Shuang Yu
Solution Architect,
NVIDIA

Lab Instructors

Gerald Shen
Research Scientist

Adil Asif
Deep Learning Algorithm Engineer,
NVIDIA

Lab Assistants and Contributors

Ronay Ak
Senior Data Scientist
NVIDIA

Chris Deotte
Senior Data Scientist
NVIDIA

Wenwen Gao
Lead Product Manager
NVIDIA

Tutorials Track

Hands-on Lab Content

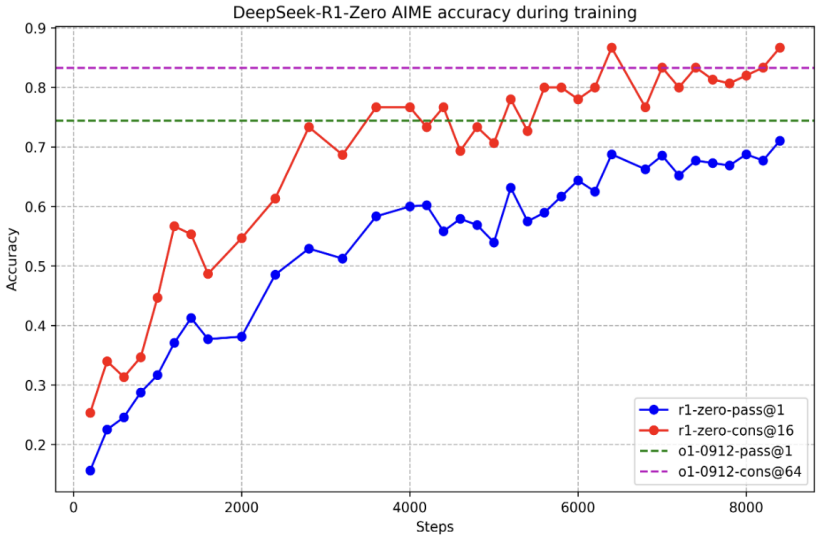
Part	Time Allocation
Background introduction	20 mins
Lab overview and environment setup	30 mins
Lab A: Knowledge distillation	120 mins
Lab B: Reinforcement learning	120 mins
Conclusion and resources	10 mins

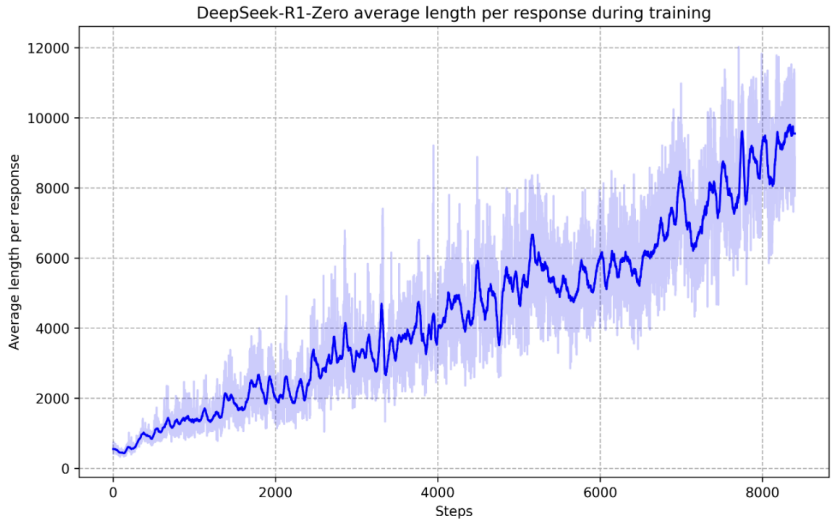
Background Introduction

Reasoning Model

Reasoning language models (RLMs) have emerged as a transformative advancement in the evolution of large language models (LLMs).

Reasoning language models are evolving after o1 and R1, including Qwen3、Llama 4、Seed-thinking、Llama-Nemotron, and all of them use reinforcement learning in post-training phase.





Figures from Deepseek-R1 Paper: https://arxiv.org/abs/2501.12948

Background Introduction

Reasoning Model

Reasoning language models (RLMs) think before they answer, producing a long internal chain of thought before responding to user.

```
Question: Convert the point $(0,3)$ in rectangular
coordinates to polar coordinates. Enter your answer in the
form $(r,\theta),$ where $r > 0$ and $0 \le \theta < 2 \pi.$
```

Alright, let's see. I need to convert the rectangular coordinates (0, 3) to polar coordinates. Hmm, polar coordinates are represented as (r, θ), where r is the distance from the origin to the point, and θ is the angle made with the positive x-axis.
...
The point (0, 3) is on the y-axis, right? Because the x-coordinate is 0, so it's 3 units up along the y-axis... Let me verify that.
In the rectangular coordinate system, when x is 0 and y is positive, the point lies along the positive y-axis. The angle from the positive x-axis to the positive y-axis is a quarter of a full circle, which is 2π/4 = π/2 radians. Yeah, that makes sense. So θ should be π/2.
But wait, just to be thorough, ...
Let me double-check everything. The formulas for converting rectangular to polar ...
Alternatively, if we use the coordinate system quadrants to determine θ, since the point is on the axis ...
I think that's all. The key was recognizing that when x is 0, the point is on the y-axis, so θ is either π/2 or 3π/2 depending on the sign of y. Since y is positive here, it's π/2.
Final Answer
The polar coordinates of the point (0, 3) are \boxed{(3, \frac{\pi}{2})}.

Thinking Process

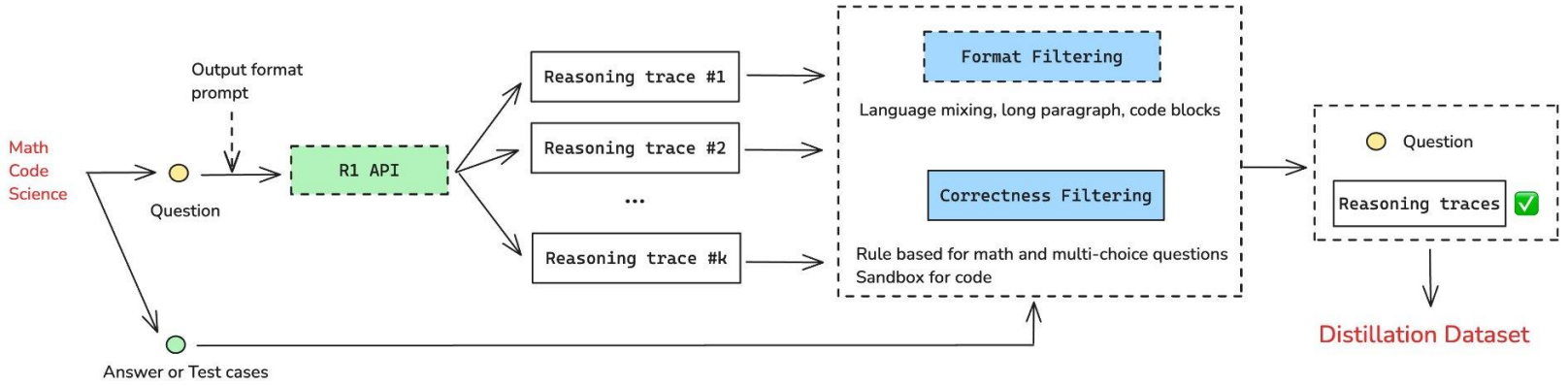
Background Introduction

Knowledge Distillation

Knowledge Distillation is a simple and effective technique to significantly enhance the reasoning abilities of smaller models.

In this process:

- A large **teacher model** (e.g., DeepSeek-R1) generates high-quality long chain-of-thought data that contains detailed reasoning steps.
- Smaller **student models** (e.g., Qwen-1.5B) learn from reasoning data, rapidly improving their reasoning capabilities.



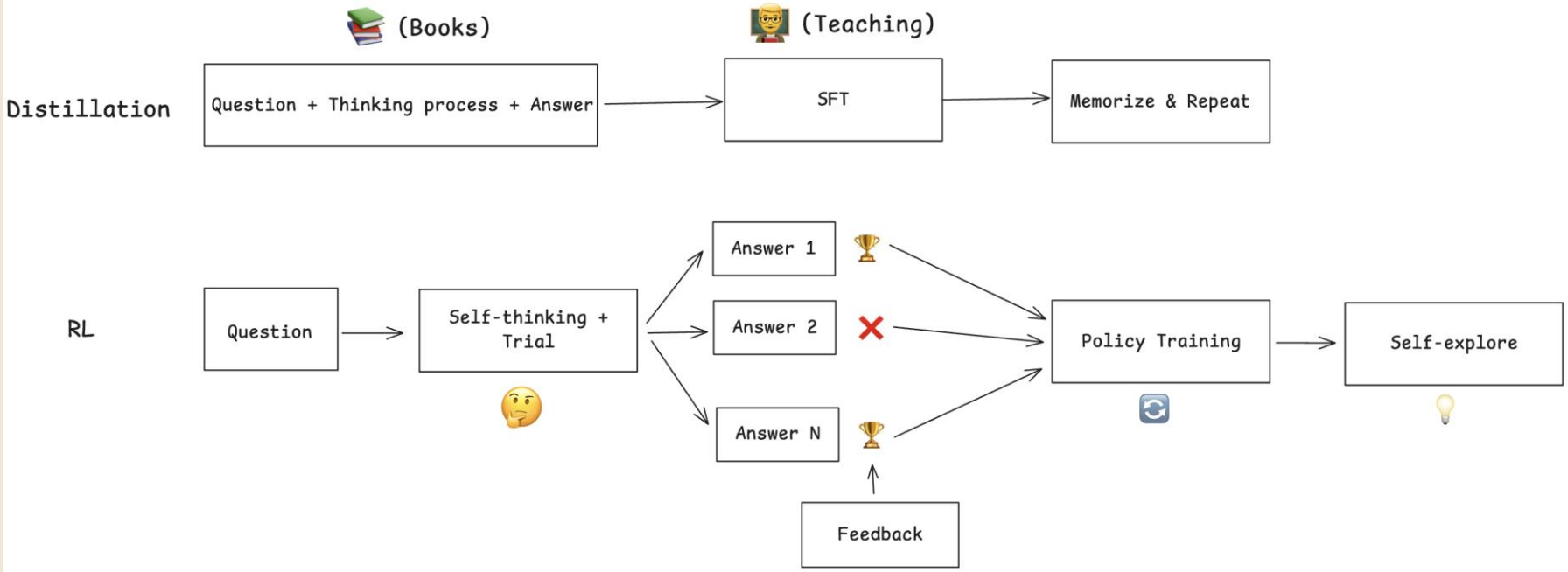
Background Introduction

RL in Reasoning Models

The most advanced reasoning models, like R1, Qwen3, OpenAI's o series and Grok all use **reinforcement learning** in the post-training stage.

Distillation (SFT) is like "Imitation learning": Student gets to learn from a strong teacher, but never gets to attempt things on their own. Like reading a textbook without doing any of the practice questions.

RL is like "Solving Problems": Students improve by practicing and experimenting with different solutions. Though slower at first, this helps them find better strategies and build real reasoning skills with strong **generalization**.



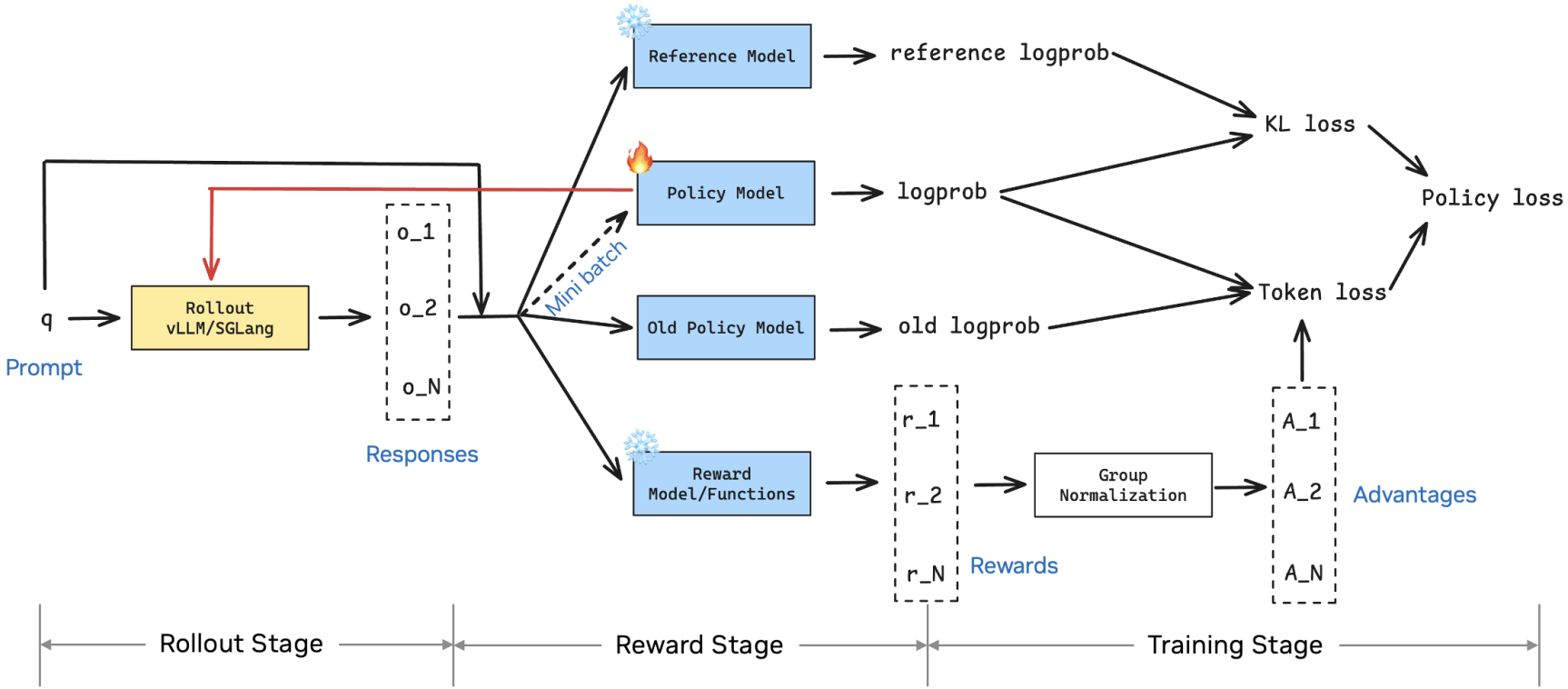
Background Introduction

RL in Reasoning Models

Optimization objective
Optimize LLM policy to maximize the expected reward of the generated responses.

Three models
Policy model, reward model/function, reference model

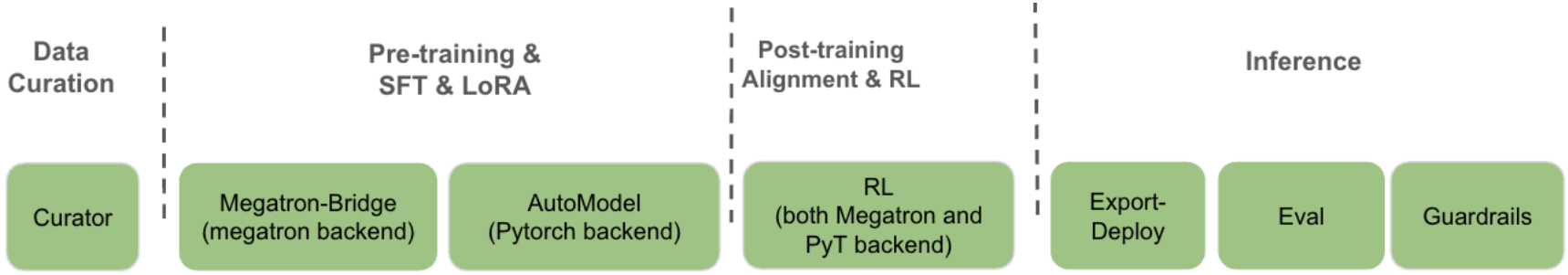
Training flow



Background Introduction

NeMo Framework - https://github.com/NVIDIA-NeMo

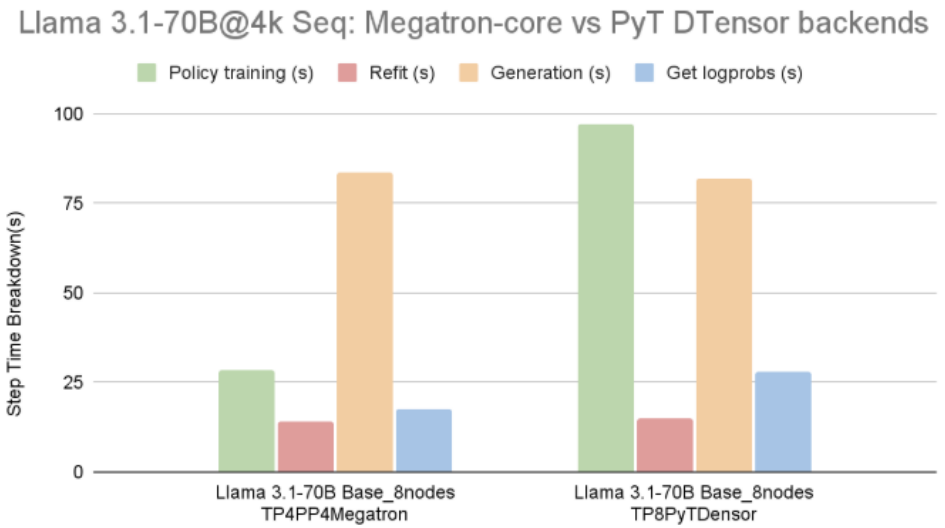
- GPU accelerated, end-to-end training framework for LLMs, multi-modal models and speech models. It has a collection of repos - a few highlights
  - RL (https://github.com/NVIDIA-NeMo/RL) [Focus of today's lab]
    - Dual training backend: (1) PyT Dtensor for seamless integration with 🤗 HuggingFace (2) Megatron for speed and scalability. VLLM for generation
    - Accuracy curves, speed benchmarks and charts are https://nvidia-nemo.github.io/blog/2025/07/21/nemo-rl-v0.3/#getting-started-with-megatron-training
  - AutoModel (https://github.com/NVIDIA-NeMo/Automodel)
    - Accelerated PyT backend to provide Day 0 pretraining and finetuning support for 🤗 HuggingFace models
    - Will add distillation support in coming release
  - Megatron-bridge (https://github.com/NVIDIA-NeMo/Megatron-Bridge)
    - Accelerated Megatron backend for high throughput pretraining and finetuning for top community models
    - To be released in Aug 2025
  - NeMo (https://github.com/NVIDIA/NeMo) - To be refactored to focus only on Speech
    - Original NeMo Repo, monolithic!
    - Will be refactored to focus on Speech models



Background Introduction

NeMo RL - https://github.com/NVIDIA-NeMo/RL

- Dual Training backend: PyT DTensor for 🤗 HF integration, and Megatron backend for SOL



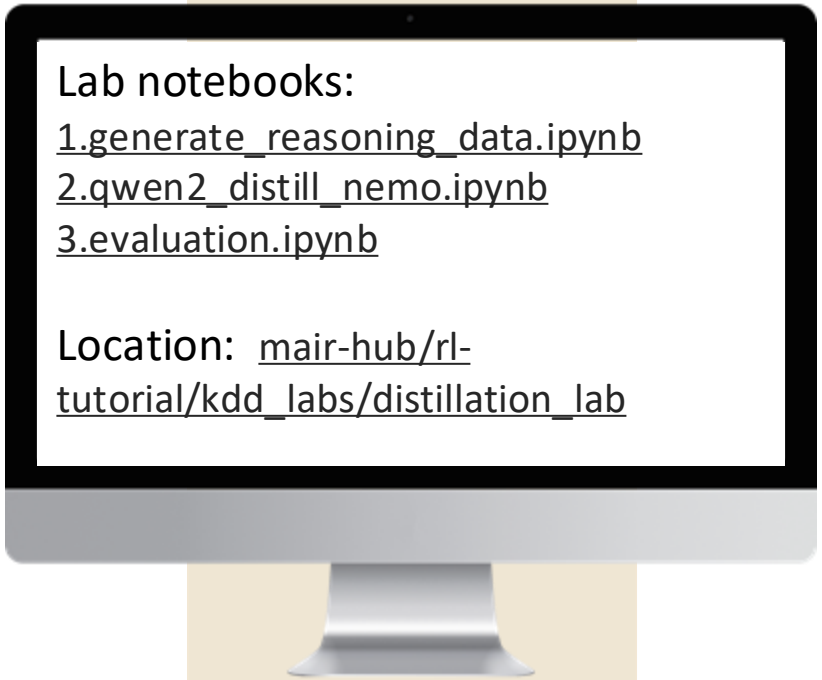
- Highlights from upcoming roadmap
  - Large MOE (DSV3, Qwen-235B) optimizations (already supported, will optimize further for longer sequence)
  - FP8 training and FP8 roll out
  - Asynchronous RL with split resource placement
  - Robust multi-tool use with sandbox env
  - Gym

Lab Overview and Environment Setup

Lab Overview and Logistics

- Tutorials of the lab can be located here: github repo
- You are encouraged to choose one of the labs as both distillation and RL take time to get to meaningful results.
- Each Lab will take about 2 hours, with an additional 30 mins for environment setup no matter which lab you pick.
- Detailed instructions are enclosed in the Jupyter notebooks of the labs.
- Reach out to Lab instructors whenever you have questions or run into problems.

Lab A: Knowledge Distillation



Before Training

- Configure NVIDIA NIM API for DeepSeek-R1 access
- Prepare distillation dataset from DeepSeek-R1
- Set up NeMo 2.0 framework and container environment

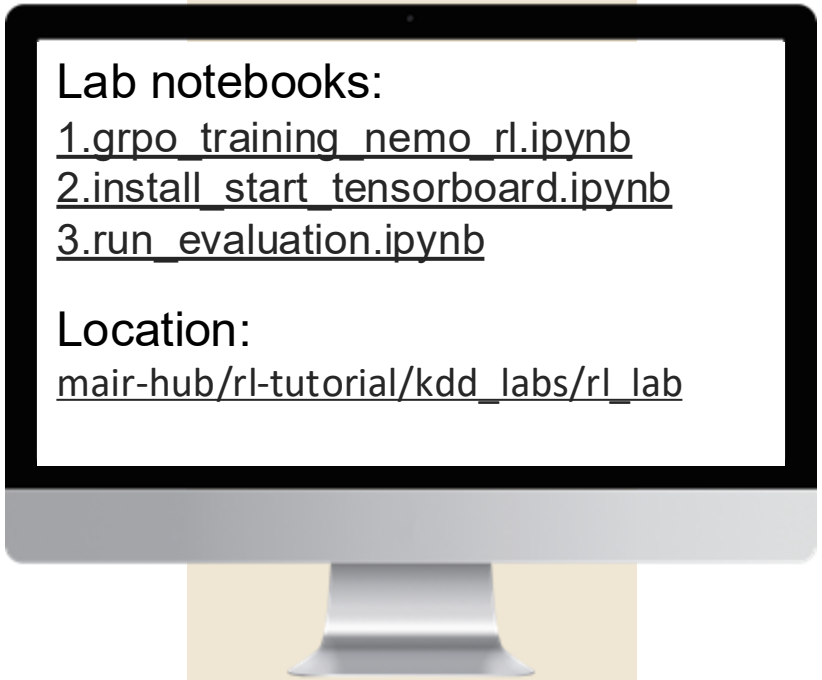
Training

- Configure supervised fine-tuning components
- Execute fine-tuning with NeMo-Run
- Generate and validate model outputs

After Training

- Deploy model for evaluation
- Run evaluation on Math500 benchmarks
- Document distillation effectiveness and best practices

Lab B: Fine-tune with RL



Setup

- Setup the development environment and NeMo-RL framework

Training

- Recap the GRPO algorithm
- Launch a GRPO training experiment on the math reasoning task
- Monitor the RL process using TensorBoard

Understand the Training Process *(while training is in progress)*

- Learn about the key hyperparameters of RL training
- Analyze the 4 stages of the RL process (sample prompts, rollout, reward, and training) using the output logs

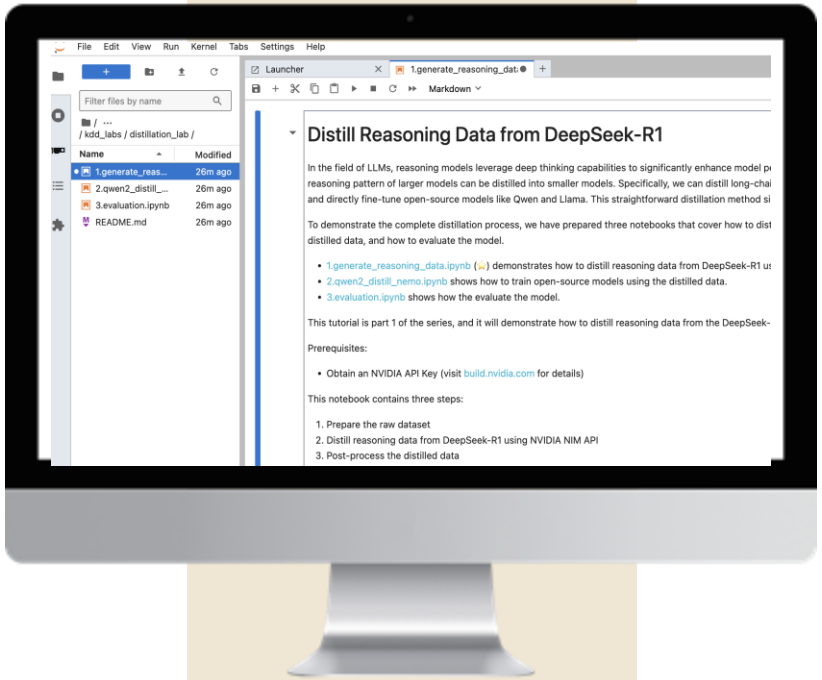
Evaluation

- Evaluate the trained model and analyze the results

Next Steps & Summary

- Customize the RL experiment and derive some takeaways

Lab Environment Setup

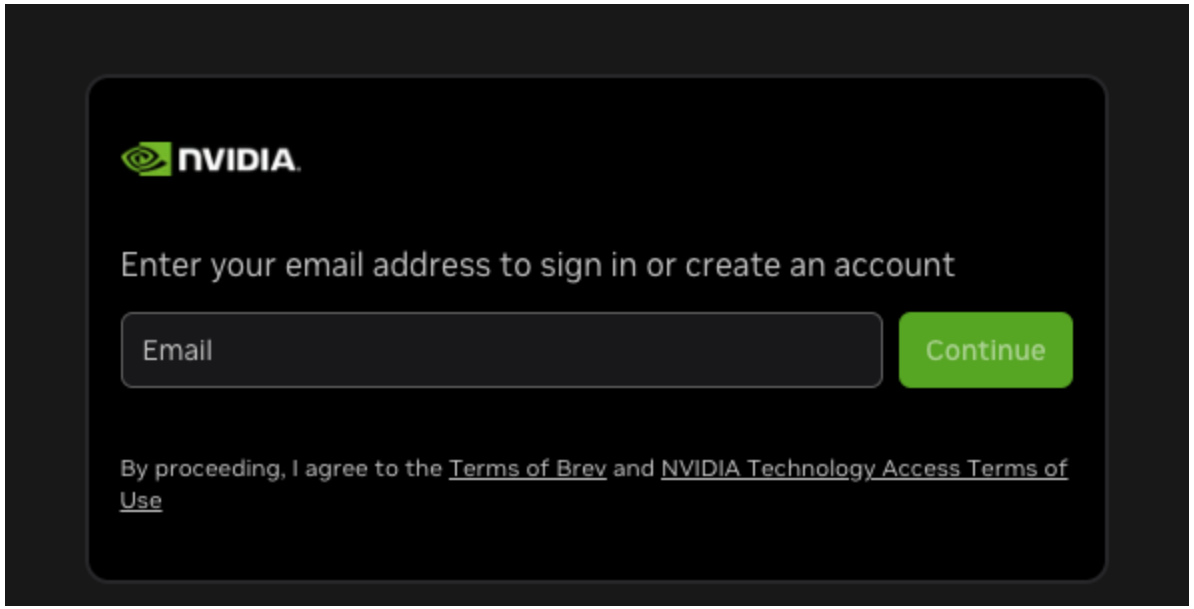


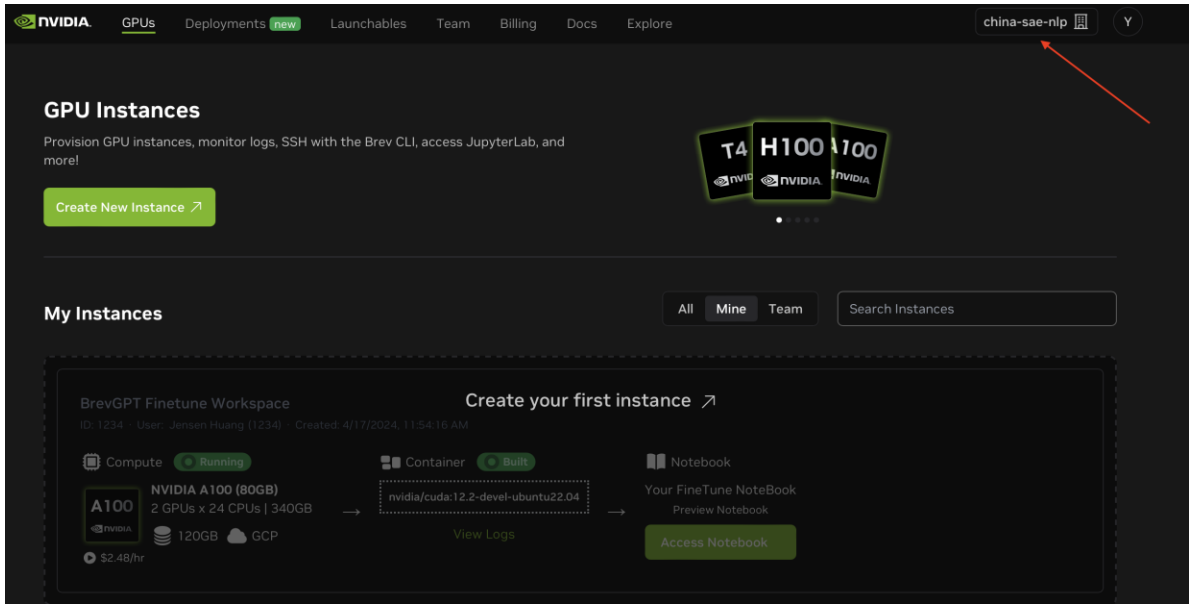
- Register to NVIDIA Brev: https://brev.nvidia.com/
- Join the lab organization on Brev
- Deploy the lab launchable on Brev
- For Lab A only:
  - Get the NIM API key https://build.nvidia.com/

Brev Environment Setup

Registration:
Create new account using your personal email on https://brev.nvidia.com

Join the Lab organization
Use this invitation link to join the organization
Verify you are in the "**china-sae-nlp**" organization.



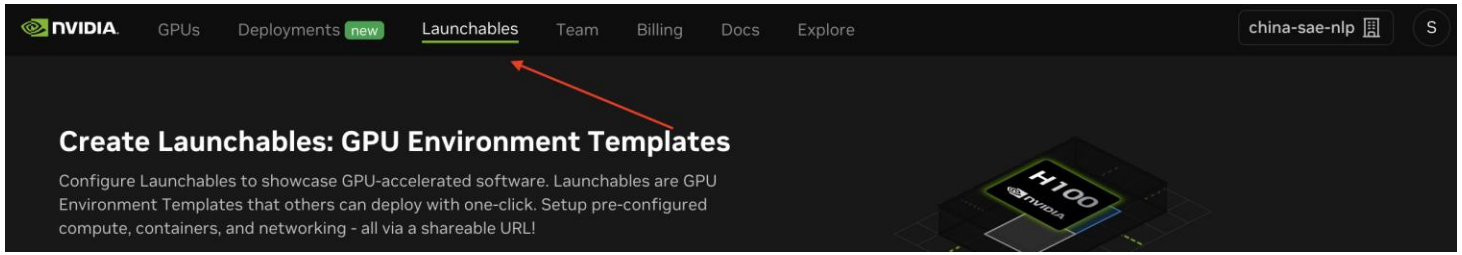


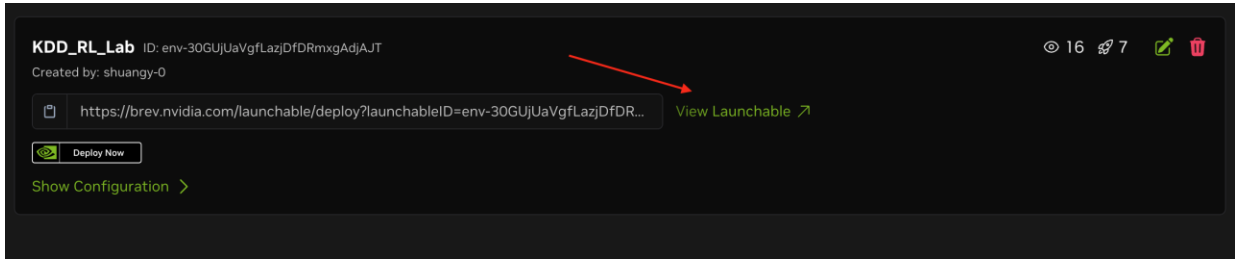
August 3–7, 2025
KDD2025

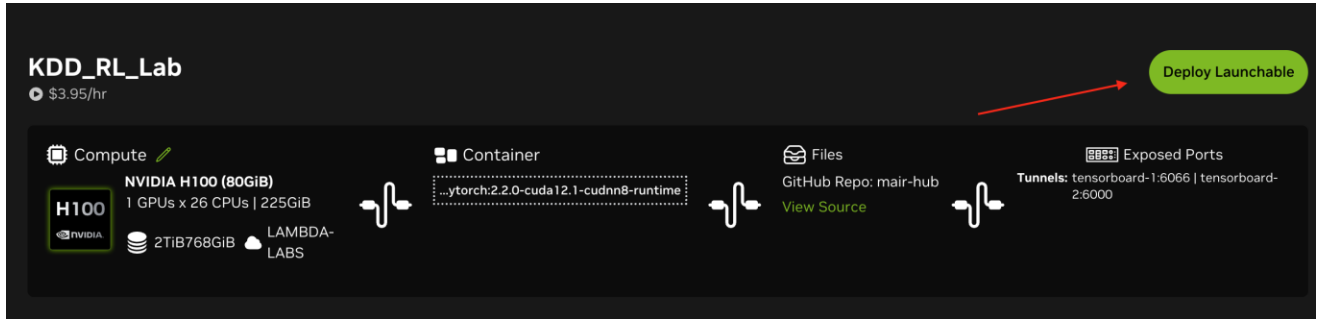
Brev Environment Setup

Deploy the Lab launchable
Go to the "Launchables" page, locate the launchable of the lab and click "View Launchable", then hit the "Deploy Launchable" button.

Launchable for Lab A: *KDD_Distillation_Lab*
Launchable for Lab B: *KDD_RL_Lab*

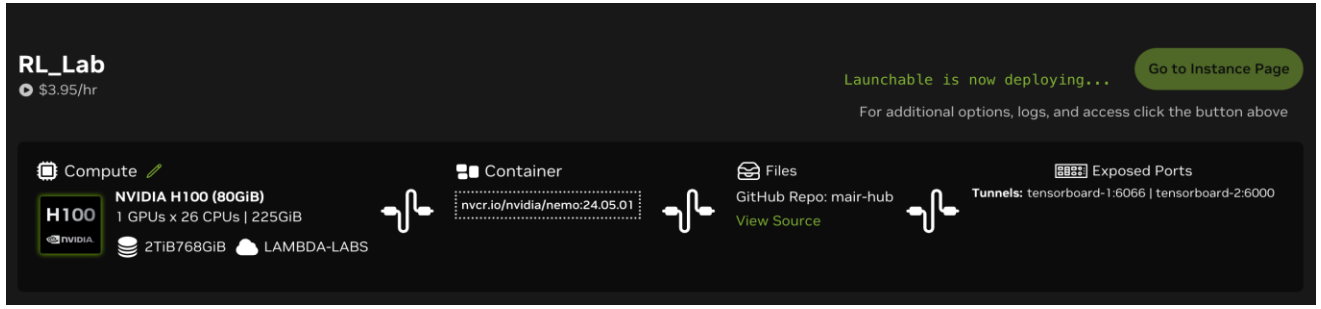




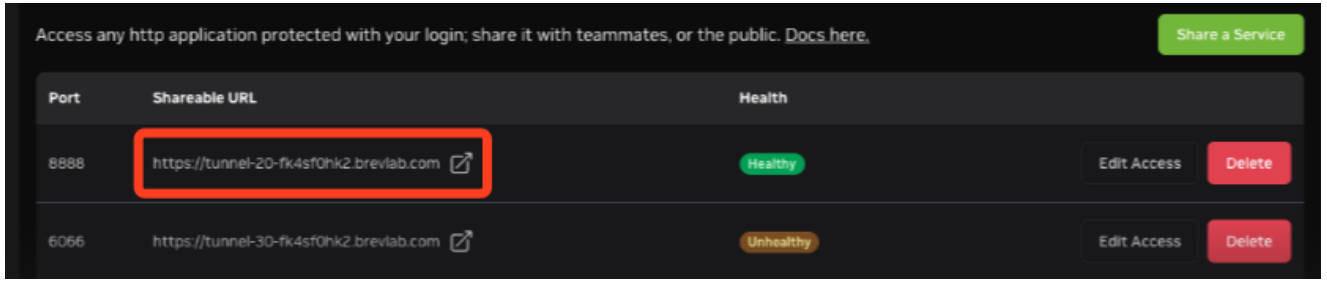


Access the Jupyter Lab environment

Click the "Go to Instance Page" to monitor the deployment progress. The deploy process takes about 15 mins.

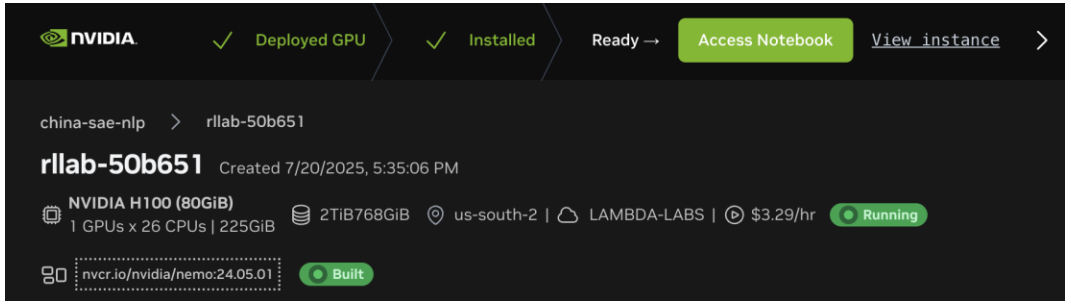


Once finished, get to the Jupyter Notebook environment through:



Lab A – Distillation Lab:

Click the link for port 8888.



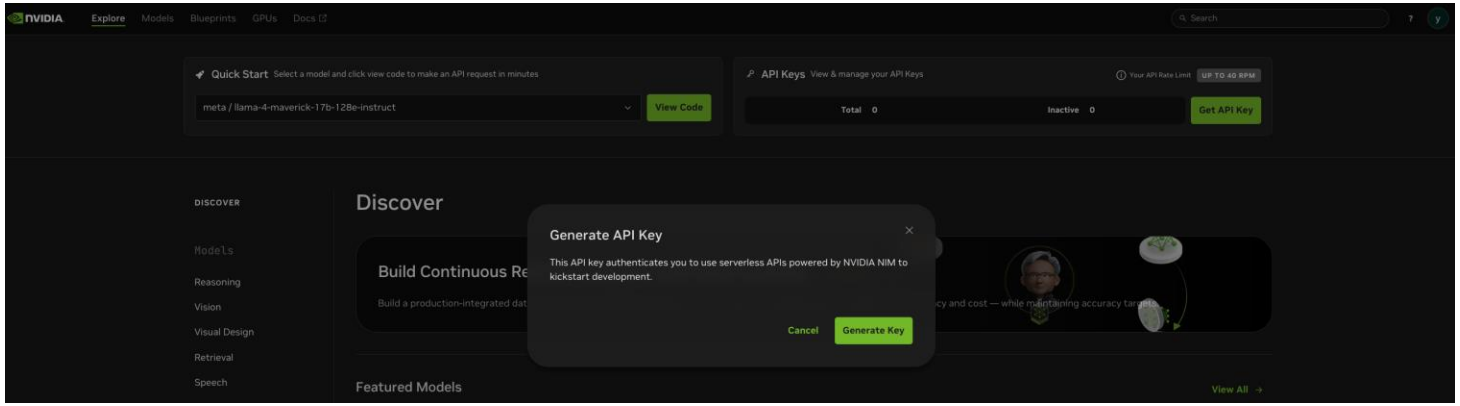
Lab B – RL Lab:

Click the "Access Notebook" button.

NIM API Key Generation

- Required only if you choose Lab A: Knowledge distillation.
- Register and login to https://build.nvidia.com/.
- Click the "Get API Key" and then "Generate Key" button to create your key.

Copy and remember the key which will be used in the lab later.

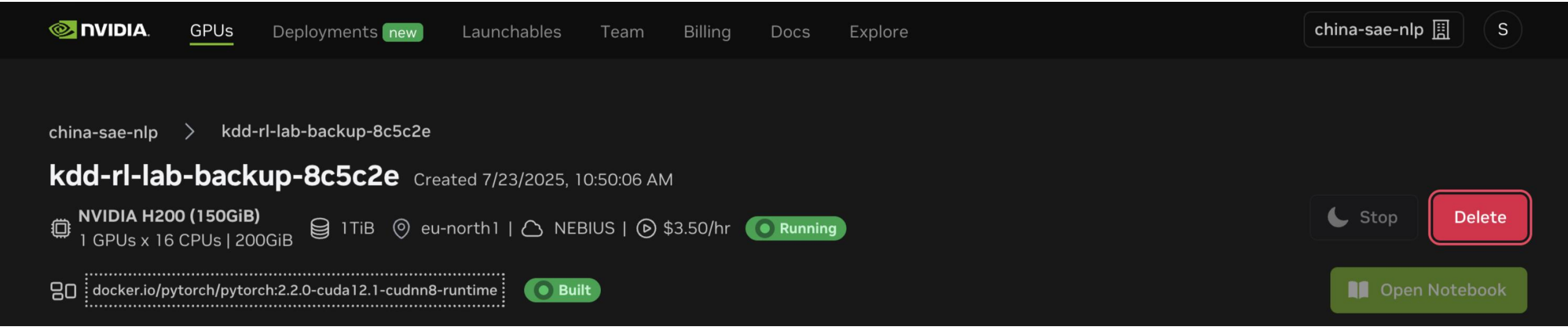


August 3–7, 2025

KDD2025

Delete the Running Instance

- Delete the running instance from the GPUs page before the end of the lab



Conclusion and Resources

What We Achieved in 3 Hours:

- Incentivized reasoning capabilities in Qwen2.5-1.5B model through two methods:
- Knowledge distillation – Transferred reasoning skills from DeepSeek R1 to Qwen2.5-1.5B, using NeMo's end-to-end pipeline.
- RL Post-Training – Applied RLVR (GRPO algorithm) to Qwen2.5-1.5B model using NeMo RL framework.
- Access lab notebooks for rerun and extension on your own environment.

🙏 If you found today's workshop useful, a quick star ⭐ or watch on the GitHub repos for new recipes, updates and community discussions:

- NVIDIA/NeMo — Unified framework for pre-training, fine-tuning, and deploying LLMs & multimodal models at any scale.
- NVIDIA-NeMo/RL — A scalable and efficient post-training library designed for models ranging from 1 GPU to thousands, and from tiny to over 100 billion parameters.
- MAIR-Hub – A central repository hosting RL and fine-tuning tutorials covering LLM, VLM and Speech models and agentic tasks.

THANK YOU!