

Update OpenEnv extra 24 hours deadline

Early OpenEnv Deadline + 24 hours to Monday, October 26th @ 7 PM PT

**WELCOME
BACK!**

Schedule

Submission Deadlines:

- **Early OpenEnv Deadline: Monday, October 26th @ 7 PM PT See Slide 4**
- **Final Deadline: Wednesday, 29th October @ 7 PM PT**

Friday 24th October

- 7 PM PT Kickoff

Saturday 25th October:

- 10 AM PT: Opening talk [Discord Event Link](#)
- 10:30 AM PT: Daniel - **Reinforcement Learning** with OpenEnv & Unsloth. [Discord Event Link](#)
- 4 PM PT: Office hours. [Discord Event Link](#)

Sunday 26th October:

- 10 AM PT: OpenEnv and Synthetic Data Best Practises: Sanyam. [Discord Event Link](#)
- 4 PM PT: Office hours. [Discord Event Link](#)

Prizes

Synthetic-Data Track



\$3000 + 1200 hrs of GPU credits
+ trophy



\$1500 + 600 hrs of GPU credits
+ trophy



\$900 + 300 hrs of GPU credits
+ trophy

New OpenEnv RL Track!

2 EARLY Submissions (7PM Monday PT) will win EXTRA!

2 x Each (\$500 GPU Credits + 1 Ray-Ban Meta)

PLUS ALSO THE REGULAR TIMELINE:

3 extra Ray-Bans Meta (still deciding how to split)

Early submissions qualify for the Regular Prizel!



\$2500 GPU credits



\$1000 GPU credits



\$500 GPU credits

OpenEnv Reinforcement Learning

OpenEnv: Agentic Execution Environments

We're using the new [OpenEnv](#) library which has over 2000+ environments for RL!

To run this, press "Runtime" and press "Run all" on a [free](#) Tesla T4 Google Colab instance!



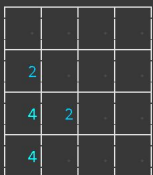
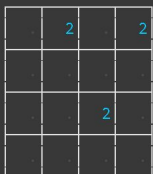
Join our Discord

Documentation

Join Discord if you need help + ★ Star us on [Github](#) ★

To install Unsloth your local device, follow [our guide](#).

```
▶ action = OpenSpielAction(action_id = 0, game_name = "2048")
result = openenv_process.step(action)
current_state = result.observation
print(render_board(current_state))
```



```
def strategy(board):
    size = len(board)
    # helper to detect if move merges
    def can_move_dir(dx,dy):
        for i in range(size):
            for j in range(size):
                x,y=i+dx,j+dy
                if 0<=x<size and 0<=y<size:
                    if board[i][j]!=0 and board[i][j]==board[x][y]:
                        return True
                    return False
    # priority order: 3=left,2=right,1=up,0=down
    dirs = [(0,-1),"3", (0,1),"2", (-1,0),"1", (1,0),"0"]
    for (dx,dy),move in zip(dirs[1::2], dirs[1::2]):
        if can_move_dir(dx,dy):
            return move
    return "0"
```

 OpenEnv



[OpenEnv](#) is a new PyTorch package offering 2000+ environments designed for reinforcement learning.

This special track will give **2 early submissions (Sunday 7PM PT)**
\$500 + 1 Ray Ban each!

Ideas:

1. Customize [Unsloth + OpenEnv 2048 Game notebook](#) to work on other RL environments
2. Create universal reward functions for all 2000+ envs.
3. Design new environments using the OpenEnv spec.



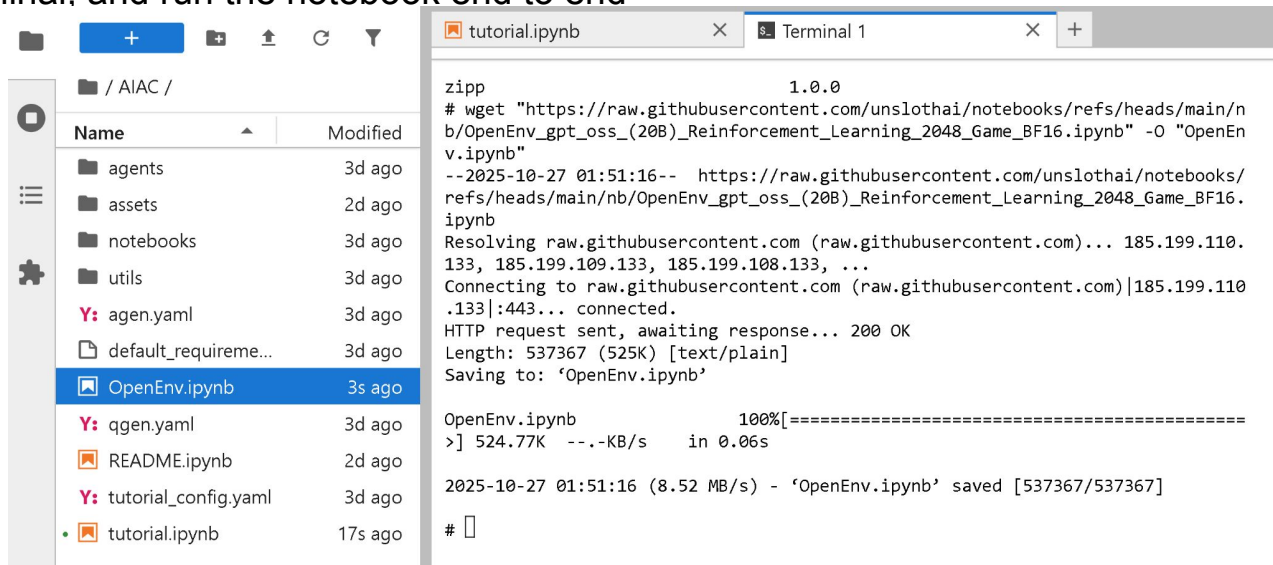
together we advance_

OpenEnv Setup

!!! FIRST FOLLOW THE STEPS FROM SLIDE 8 ONWARDS

Then, please execute in a terminal, and run the notebook end to end

```
wget
"https://raw.githubusercontent.com/unslothai/notebooks/refs/heads/main/nb/OpenEnv_gpt_oss_(20B)_Reinforcement_Learning_2048_Game_BF16.ipynb"
-O "OpenEnv.ipynb"
```



The screenshot shows a file explorer window on the left and a terminal window on the right. The file explorer displays the contents of the /AIAC/ directory, with 'OpenEnv.ipynb' highlighted. The terminal window shows the execution of the 'wget' command to download the notebook and the subsequent saving of the file.

Name	Modified
agents	3d ago
assets	2d ago
notebooks	3d ago
utils	3d ago
Y: agen.yaml	3d ago
default_requireme...	3d ago
OpenEnv.ipynb	3s ago
Y: qgen.yaml	3d ago
README.ipynb	2d ago
Y: tutorial_config.yaml	3d ago
tutorial.ipynb	17s ago

```
zip 1.0.0
# wget "https://raw.githubusercontent.com/unslothai/notebooks/refs/heads/main/nb/OpenEnv_gpt_oss_(20B)_Reinforcement_Learning_2048_Game_BF16.ipynb" -O "OpenEnv.ipynb"
--2025-10-27 01:51:16-- https://raw.githubusercontent.com/unslothai/notebooks/refs/heads/main/nb/OpenEnv_gpt_oss_(20B)_Reinforcement_Learning_2048_Game_BF16.ipynb
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.110.133, 185.199.109.133, 185.199.108.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.110.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 537367 (525K) [text/plain]
Saving to: 'OpenEnv.ipynb'

OpenEnv.ipynb      100%[=====]
>] 524.77K  --.-KB/s   in 0.06s

2025-10-27 01:51:16 (8.52 MB/s) - 'OpenEnv.ipynb' saved [537367/537367]

#
```

[https://github.com/unslothai/notebooks/blob/main/nb/OpenEnv_gpt_oss_\(20B\)_Reinforcement_Learning_2048_Game_BF16.ipynb](https://github.com/unslothai/notebooks/blob/main/nb/OpenEnv_gpt_oss_(20B)_Reinforcement_Learning_2048_Game_BF16.ipynb)

Weather environment / other environments RL FAQ

If you are making a custom environment, I would:

1. Ask an Open Source LLM to generate a weather environment - ie it should say today's weather (sunny, not sunny), with temperature, humidity, wind speed etc etc. It should be governed by some physical laws or mathematical transition process (Markov Chains)
2. Create a reward function to predict the weather and do RL on this!

Get faster inference with vLLM

Unsloth supports faster inference for some models with `fast_inference = True`. For eg Llama, Qwen models. GPT-OSS will use Unsloth internal inference which is a little bit slower. See

[https://github.com/unslothai/notebooks/blob/main/nb/Advanced_Llama3_2_\(3B\)_GRPO_LoRA.ipynb](https://github.com/unslothai/notebooks/blob/main/nb/Advanced_Llama3_2_(3B)_GRPO_LoRA.ipynb) Do:

```
import os
os.environ["UNSLOTH_VLLM_STANDBY"] = "1" # [NEW] Extra 30% context lengths!
from unsloth import FastLanguageModel
max_seq_length = 2048 # Can increase for longer reasoning traces
lora_rank = 64 # Larger rank = smarter, but slower
model, tokenizer = FastLanguageModel.from_pretrained(
    model_name = "meta-llama/Llama-3.2-3B-Instruct",
    max_seq_length = max_seq_length,
    load_in_4bit = False, # False for LoRA 16bit
    fast_inference = True, # Enable vLLM fast inference
    max_lora_rank = lora_rank,
```

ReadTimeoutError for OpenEnv

If you see the error below:

```
else: # right
    newb = [list(reversed(slide(list(reversed(r)))) for r in board]
    if newb != board:
        return str(m)
return "0"
```

.	.	.	.
.	.	.	2
.	.	.	2
.	.	.	.

```
[2025-10-27 02:56:24] WARNING connectionpool.py:868: Retrying (Retry(total=2, connect=None, read=None, redirect=None, status=None)) after connection broken by 'ReadTimeoutError("HTTPConnectionPool(host='localhost', port=8081): Read timed out. (read timeout=15.0)")': /step
[2025-10-27 02:56:39] WARNING connectionpool.py:868: Retrying (Retry(total=1, connect=None, read=None, redirect=None, status=None)) after connection broken by 'ReadTimeoutError("HTTPConnectionPool(host='localhost', port=8081): Read timed out. (read timeout=15.0)")': /step
[2025-10-27 02:56:55] WARNING connectionpool.py:868: Retrying (Retry(total=0, connect=None, read=None, redirect=None, status=None)) after connection broken by 'ReadTimeoutError("HTTPConnectionPool(host='localhost', port=8081): Read timed out. (read timeout=15.0)")': /step
```

I added a fix since 1AM 27th October PT time. Please do:

1. Update Unsloth pip install --upgrade --force-reinstall --no-deps --no-cache-dir unsloth unsloth_zoo
2. Re-get notebook as "OpenEnv_NEW.ipynb":

wget

```
"https://raw.githubusercontent.com/unslothai/notebooks/refs/heads/main/nb
/OpenEnv_gpt_oss_(20B)_Reinforcement_Learning_2048_Game_BF16.ipynb"
"OpenEnv_NEW.ipynb"
```

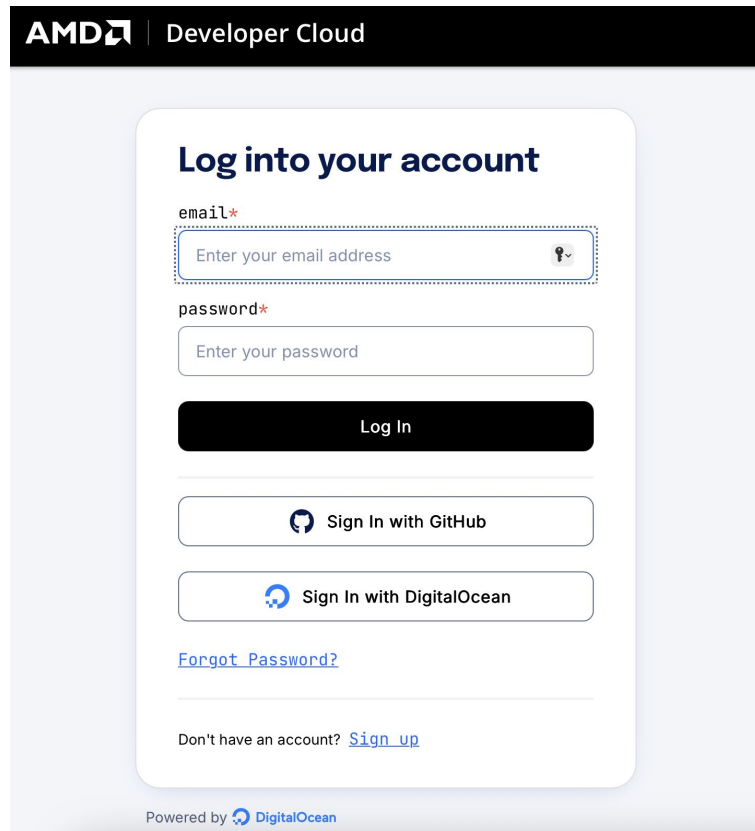
How to form teams

- Create an account on <https://devcloud.amd.com/>
- Fill out the Google form (one per team!) <https://forms.gle/RPV7fURLNHDjz2yr9>

Please see

<https://docs.unsloth.ai/new/unsloth-amd-pytorch-synthetic-data-hackathon>

for debugging and more



The screenshot shows the AMD Developer Cloud login interface. At the top, there is a black header with the AMD logo and the text "Developer Cloud". Below this, the main content area has a light blue background. A white rounded rectangle contains the login form. The form is titled "Log into your account" in bold. It has two input fields: "email*" with a red asterisk and a placeholder "Enter your email address", and "password*" with a red asterisk and a placeholder "Enter your password". Below the password field is a black "Log In" button. Underneath the button are two social login options: "Sign In with GitHub" and "Sign In with DigitalOcean", each with its respective logo. A link "Forgot Password?" is located below the social login options. At the bottom of the form, it says "Don't have an account? Sign up" with a link. At the very bottom of the page, it says "Powered by DigitalOcean" with the DigitalOcean logo.

Create a GPU Instance

The screenshot displays the AMD Developer Cloud interface. On the left, a sidebar lists 'PROJECTS' and 'CORE SERVICES'. Under 'CORE SERVICES', 'GPU Droplets' is highlighted with a 'NEW' badge. The main area shows a project named 'first-project' with a 'DEFAULT' tag and the description 'Machine learning / AI / Data processi'. Below this, there are tabs for 'Resources', 'Activity', and 'Settings'. A red arrow points from the 'DROPLETS (56)' link to a dropdown menu that appears when the 'Create' button is clicked. The dropdown menu lists four options: 'Droplets' (Create cloud servers), 'GPU Droplets' (Create cloud servers with GPUs), 'Kubernetes' (Create Kubernetes clusters), and 'App Platform' (Deploy your code). The 'GPU Droplets' option is highlighted with a red border.

AMD
DEVELOPER CLOUD

Search by resource name or public IP (Cmd+B)

Create ▾

PROJECTS ▾

CORE SERVICES

GPU Droplets **NEW**

Agent Platform

CPU Droplets

Backups & Snapshots

first-project **DEFAULT**
Machine learning / AI / Data processi

Resources Activity Settings

DROPLETS (56)

Create ^

Droplets
Create cloud servers

GPU Droplets
Create cloud servers with GPUs

Kubernetes
Create Kubernetes clusters

App Platform
Deploy your code

Configure your instance

MI300X

Credits applicable

These GPU plans use your AMD GPU credits. You have AMD credits available for use.

This information is updated daily.



i Note: After the credits have been used, it will charge your payment method. AMD credit only covers GPU access. All other services would be charged to your payment method.

GPU Plans

YOU MUST USE ROCM 6.4.0 NOT 7



MI300X x8

8 GPU - 1.5 TB VRAM - 160 vCPU - 1920 GB RAM

Boot disk: 2 TB NVMe- Scratch disk: 40 TB NVMe

\$1.99/GPU/hr



MI300X

1 GPU - 192 GB VRAM - 20 vCPU - 240 GB RAM

Boot disk: 720 GB NVMe- Scratch disk: 5 TB NVMe

\$1.99/GPU/hr

Configure your instance

YOU MUST USE ROCM 6.4.0 NOT 7

Choose an image

Quick Start

Bare OS

Backups (1)

Custom Images



ROCm™ Software

ROCm. Install any AI tools you need.

[Details](#)

6.4.0 ^

7

6.4.0



Quick Start Packages

The following environments provide users with AMI to-use Docker image, and example JupyterLab notebooks.



Configure your instance

If you're on a team, make sure to add your teammates ssh keys as well!



Add an SSH Key for authentication

An SSH key pair is a more secure way to connect to your Droplet

Add an SSH Key

Configure your instance

 6.4.0-gpu-mi300x1-192gb-devcloud-atl1

Web Console

Actions ▾

● Active •  eda-test •  ATL1 •  ROCm™ Software 6.4.0 on Ubuntu 24.04

Getting to know your GPU Droplet



Even if your GPU Droplet is powered off, you will still be billed as GPU, CPU, and associated resources are still reserved. To avoid charges, be sure to **destroy** the instance if it's no longer in use. Before destruction, you can take a **snapshot** to return back to.

**Only start one instance or you will
not have enough credits for
submission!**

Access your instance

6.4.0-gpu-mi300x1-192gb-devcloud-atl1

Web Console

Actions ▾

Active • eda-test • ATL1 • ROCm™ Software 6.4.0 on Ubuntu 24.04

Getting to know your GPU Droplet

Even if your GPU Droplet is powered off, you will still be billed as GPU, CPU, and associated resources are still reserved. To avoid charges, be sure to **destroy** the instance if it's no longer in use. Before destruction, you can take a **snapshot** to return back to.

Overview

Insights

Networking

Volumes

Backups & Snapshots

Activity

Settings

```
ssh root@<public_ip>
```

CONNECTION DETAILS

Public IPv4

<public_ip>



For eg: Type `ssh root@134.XXX.XXX` into terminal

Start your workstation (Copy paste docker run)

In your terminal:

```
$ ssh root@<public_ip>
```

```
root@public_ip:~$
```

COPY PASTE BELOW:

```
docker run -it --rm \  
  --network=host \  
  --device=/dev/kfd \  
  --device=/dev/dri \  
  --group-add=video \  
  --ipc=host \  
  --cap-add=SYS_PTRACE \  
  --security-opt seccomp=unconfined \  
  --shm-size 8G \  
  -w /workspace \  
  --name rocm-jupyter \  
  edaamd/aiac:latest
```

**Make sure you manually backup
any work done in Jupyter! It will not
persist if the Jupyter server or
docker get kills**

The Dockerfile is at
<https://github.com/edamamez/Unsloth-AMD-Fine-Tuning-Synthetic-Data/blob/main/AIAC/Dockerfile>

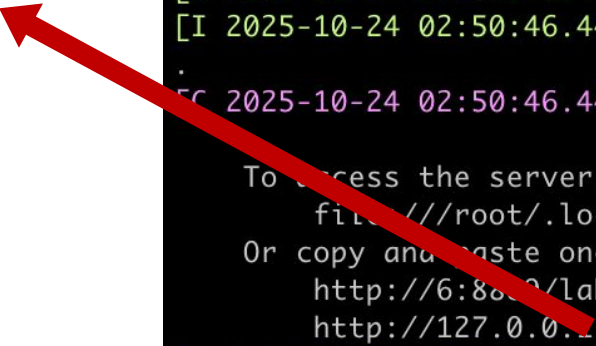
Accessing your workstation

In a separate terminal, start port forwarding:

```
$ ssh -N -v -L  
localhost:8889:localhost:8889  
root@<public_ip>
```

For eg in the new separate terminal:

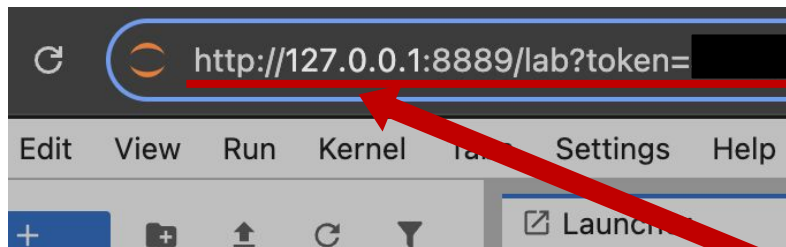
```
ssh -N -v -L  
localhost:8889:localhost:8  
889 root@134.XXX.XXX.XXX
```



```
[I 2025-10-24 02:50:46.440 ServerApp] Serving notebooks from  
[I 2025-10-24 02:50:46.440 ServerApp] Jupyter Server 2.17.0  
[I 2025-10-24 02:50:46.440 ServerApp] http://6:8889/lab?token=  
[I 2025-10-24 02:50:46.440 ServerApp] http://127.0.0.1:  
[I 2025-10-24 02:50:46.440 ServerApp] Use Control-C to stop  
.  
[I 2025-10-24 02:50:46.442 ServerApp]  
To access the server, open this file in a browser:  
file:///root/.local/share/jupyter/runtime/jpserver-  
Or copy and paste one of these URLs:  
http://6:8889/lab?token=  
http://127.0.0.1:8889/lab?token=  
[I 2025-10-24 02:50:46.454 ServerApp] Skipped non-installed  
-server-nodejs, javascript-typescript-langserver, jedi-lang  
ge-server, python-lsp-server, r-languageserver, sql-languag  
er, vscode-css-languageserver-bin, vscode-html-languageserv
```

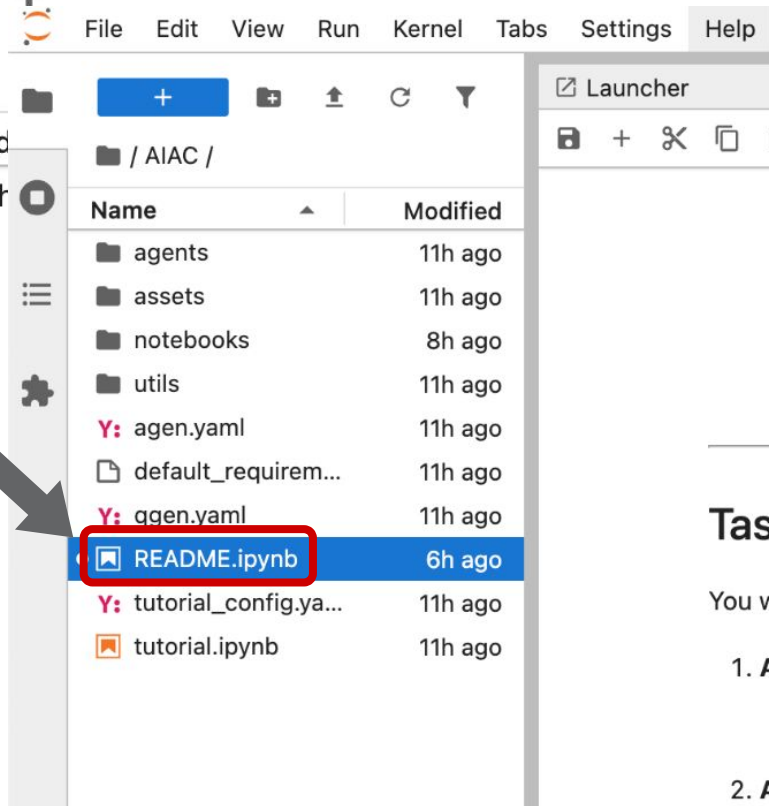
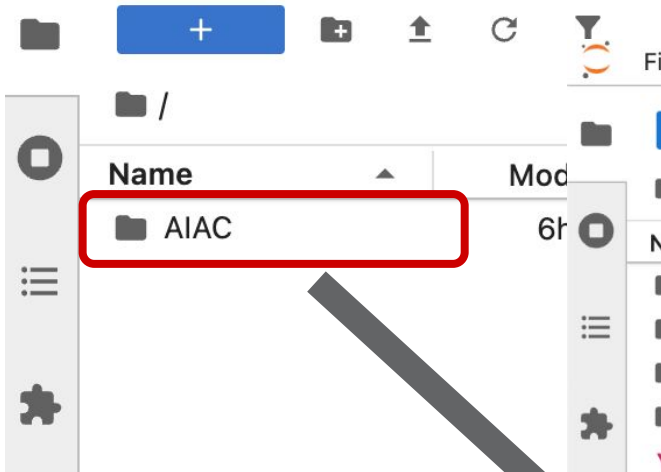

Accessing your workstation

Open jupyter server in your **local web browser**!



```
[I 2025-10-24 02:50:46.440 ServerApp] Serving notebooks from
[I 2025-10-24 02:50:46.440 ServerApp] Jupyter Server 2.17.0
[I 2025-10-24 02:50:46.440 ServerApp] http://6:8889/lab?token=
[I 2025-10-24 02:50:46.440 ServerApp] http://127.0.0.1:8889/lab?token=
[I 2025-10-24 02:50:46.440 ServerApp] Use Control-C to stop this process and
quit this notebook.
[C 2025-10-24 02:50:46.442 ServerApp]

To access the server, open this file in a browser:
    file:///root/.local/share/jupyter/runtime/jpserver-
Or copy and paste one of these URLs:
    http://6:8889/lab?token=
    http://127.0.0.1:8889/lab?token=
[I 2025-10-24 02:50:46.454 ServerApp] Skipped non-installed
-server-nodejs, javascript-typescript-langserver, jedi-lang
ge-server, python-lsp-server, r-languageserver, sql-languag
er, vscode-css-languageserver-bin, vscode-html-languageserv
```



Launcher x README.ipynb

Markdown

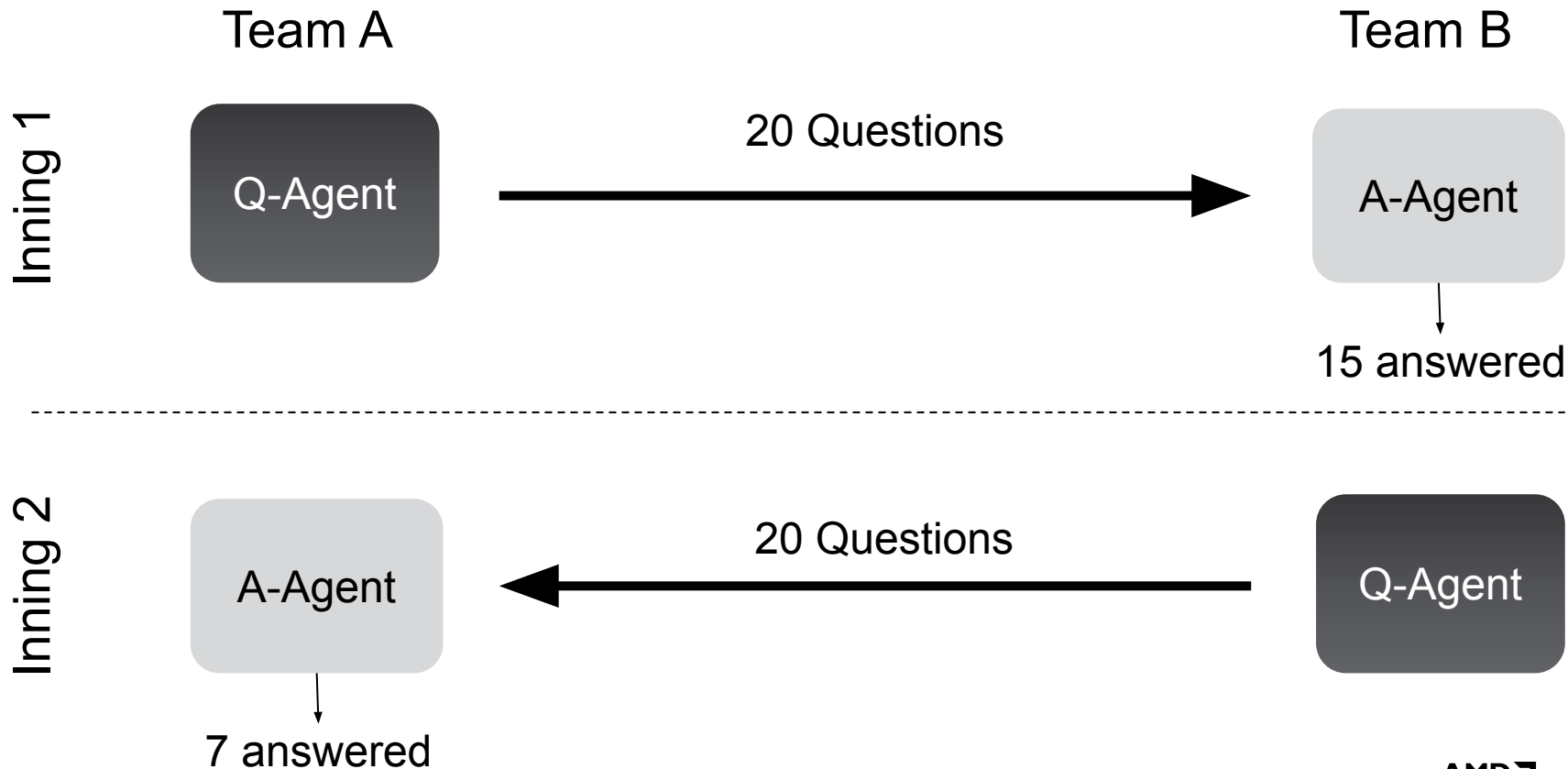
Welcome

Task

You will be building:

1. **A question agent** that will ask N puzzle-based
 - Create your model in [question_model.py](#)
 - Your question agent must output question
2. **An answer agent** that answers questions a

🏆 AIAC Overview 🏆



🚩 AAIPL Score 🚩

Team A

Team B

Inning 1
Inning 2

For **Q**-Agent: 5/20

For A-Agent: 15/20

For A-Agent: 7/20

For **Q**-Agent: 13/20

Team A Score: **12**

Team B Score: **28**



Team B Wins!



Guidelines

Format Overview

- Q-Agent: Given a topic, the Q-agent should generate questions in the specified JSON format:

```
{
  "topic": "<Topic of the Question>",
  "question": "<full question text>",
  "choices": [
    "A) <choice A text>",
    "B) <choice B text>",
    "C) <choice C text>",
    "D) <choice D text>"
  ],
  "answer": "<correct choice letter only>",
  "explanation": "brief explanation within 100 words for why the answer is correct"
}
```

from which we will extract **ONLY** the **"Question"** and **"Choices"** keys and feed it to the answer agent. The **"Topic"**, **"Question"**, **"Choices"**, and **"Answer"** will be verified for correctness from an Oracle.

- A-agent: Given a Question and Choices, A-agent should produce answer in the format of:

```
{
  "answer": "<correct choice letter only>",
  "reasoning": "brief reasoning within 100 words for why the answer is correct"
}
```

where we will extract **ONLY** the **"Answer"** key and compare it with **"Answer"** from the opponent's question.

- Remarks: Having explanation and reasoning is a plus. Not having them doesn't disqualify the question or answer being correct.

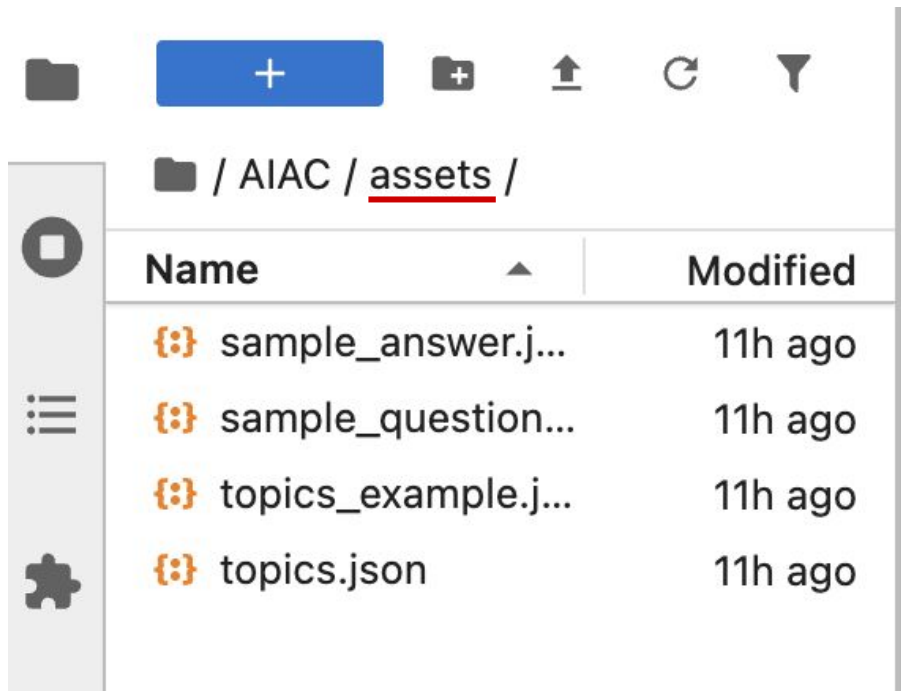
Note: We will only consider those responses from the Q-agent and the A-agent which follow the above format.







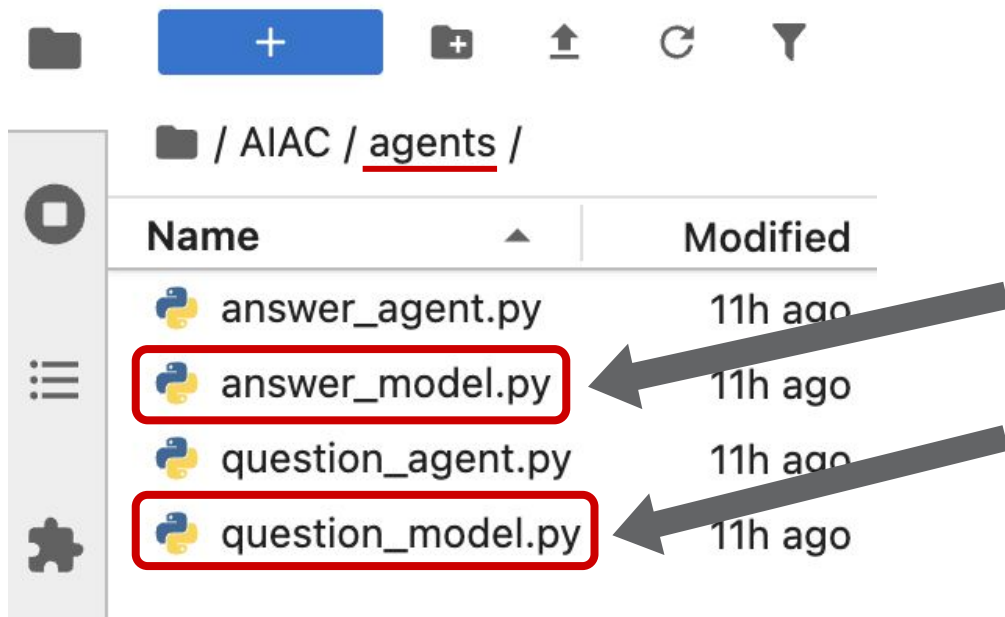
Examples



1. `sample_answer.json` details how answers should look like.
2. `sample_questions.json` details how questions should look like.
3. `topics.json` consists of two topics.
 - a. **Seating Arrangements (Circular and Linear)**. *Don't include any numeric style seating arrangements questions, e.g., how many permutations such arrangements possible, etc.*
 - b. **Blood Relations and Family Tree**
4. `topics_example.json` consists of **3** examples from each topic.



Name	Modified
 sample_answer.j...	11h ago
 sample_question...	11h ago
 topics_example.j...	11h ago
 topics.json	11h ago



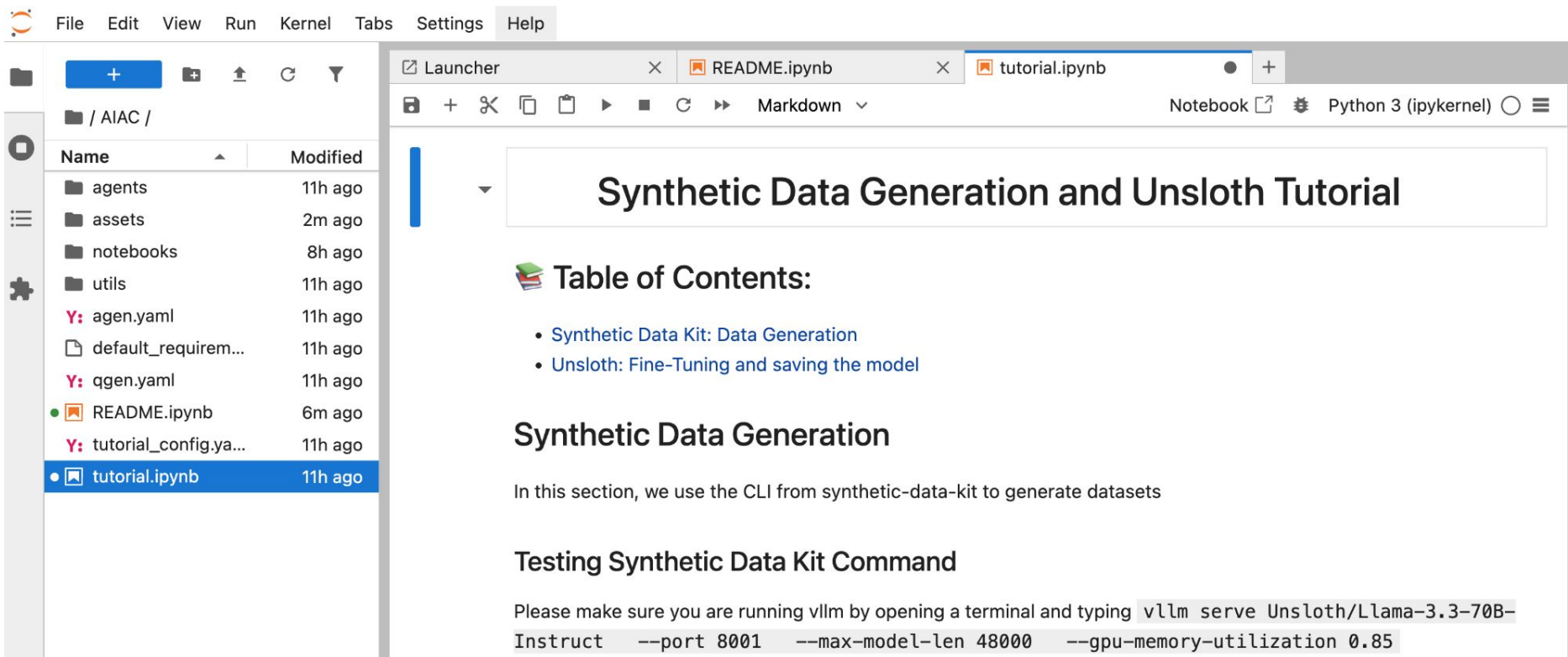
Put your code here!

Submission Check

```
# Run the following code to generate questions.  
# For demo purpose, we have used the base Qwen3-4B model for Q-Agent.  
!python -m agents.question_agent \  
    --output_file "outputs/questions.json" \  
    --num_questions 20 \  
    --verbose
```

```
# Same instructions apply for the answer agent.  
# For demo purpose, we have used the base Qwen3-4B model for A-agent.  
!python -m agents.answer_agent \  
    --input_file "outputs/filtered_questions.json" \  
    --output_file "outputs/answers.json" \  
    --verbose
```


Synthetic Data and Fine Tuning



The screenshot shows a Jupyter Notebook environment. On the left is a file explorer for the directory `/ AIAC /`. It lists several folders and files with their modification times:

Name	Modified
agents	11h ago
assets	2m ago
notebooks	8h ago
utils	11h ago
agen.yaml	11h ago
default_requirem...	11h ago
qgen.yaml	11h ago
README.ipynb	6m ago
tutorial_config.ya...	11h ago
tutorial.ipynb	11h ago

The main notebook area displays the title **Synthetic Data Generation and Unsloth Tutorial**. Below the title is a **Table of Contents:**

- [Synthetic Data Kit: Data Generation](#)
- [Unsloth: Fine-Tuning and saving the model](#)

Synthetic Data Generation

In this section, we use the CLI from synthetic-data-kit to generate datasets

Testing Synthetic Data Kit Command

Please make sure you are running vllm by opening a terminal and typing `vllm serve Unsloth/Llama-3.3-70B-Instruct --port 8001 --max-model-len 48000 --gpu-memory-utilization 0.85`



What will you submit?

Submission

You need to submit your code which should contain these main files:

1. All work must be within the `AIAC` folder. Do NOT change the folder name.
2. No need to upload anything anywhere, we'll collect your agent code at from your Jupyter Server.
 - A. The agents will be called by `python -m agents.question_agent` and `python -m agents.answer_agent` , respectively.
3. ENSURE model checkpoint(s) (e.g., `model.safetensors` or `.pt` or `.pth`) is(are) loading and expected files are getting generated from Q-agent and A-agent, when inference is done.
 - A. Outputs must be saved to `outputs/questions.json` and `outputs/answers.json` , respectively.

You can test your submission by running the commands in the [Getting Started](#) section.



Restrictions

Deadline: 7 pm PT, Wednesday Oct 29

1. **NO LAST Minute Submission:** The submission deadline is strict. Any changes to your code after the deadline may disqualify your submission.
2. RAG (Retrieval Augmented Generation) techniques are not allowed.
3. Adversarial approaches will lead to disqualification, e.g. making A-agents hallucinate.
4. Only English language is allowed for both Q-agent and A-agent.
5. Strictly stay within the `max_tokens` limits specified in `agen.yaml` & `qgen.yaml` . Other parameters can be changed.
6. Questions must pertain to the topics listed in `topics.json` .
7. Each question should be generated under `10 secs` . Questions exceeding this limit will not be considered.
8. Each answer should be generated under `6 secs` . Answers exceeding this limit will not be considered.

Feel free to reach out in the Discord channel for any clarifications or questions!

Takeaways

- ❑ We encourage everyone to **follow the rules** and the **format** strictly, otherwise your questions and answers won't be considered.
- ❑ Also very importantly, not just the format, but also the quality. **Question, Choices,** and **Answer** correctness of Q-agent.
- ❑ Further, give **equal** importance to your answer agent, i.e., A-agent.
- ❑ Ensure **four** .py files in agents/ folder for both Q-agent and A-agent.
- ❑ Do NOT cross the RESTRICTIONS as all of them will be enforced.
- ❑ **Feel free to ask any questions in the Discord!**



Ground Rules

- Be Respectful of Participants
- Use AMD GPUs (Each team gets MI300X Access)
- Use Synthetic-Data (Synthetic-Data-Kit from Meta usage is encouraged)
- Think about maximising GPU Memory Usage: 192GB is a lot!
- Remember to have fun and Drink Chai!

Resources

- Everything will be shared in #challenge-news
- AMD Blog:
<https://www.amd.com/en/developer/resources/technical-articles/2025/10x-model-fine-tuning-using-synthetic-data-with-unsloth.html>
- Unsloth Fine-Tuning Notebook: <https://github.com/unslothai/notebooks>
- Synthetic-Data-Kit: <https://github.com/meta-llama/synthetic-data-kit/>
- OpenEnv Reinforcement Learning 2048 Example:
[https://github.com/unslothai/notebooks/blob/main/nb/OpenEnv_gpt_oss_\(20B\)_Reinforcement_Learning_2048_Game_BF16.ipynb](https://github.com/unslothai/notebooks/blob/main/nb/OpenEnv_gpt_oss_(20B)_Reinforcement_Learning_2048_Game_BF16.ipynb)
- See <https://docs.unsloth.ai/new/unsloth-amd-pytorch-synthetic-data-hackathon> for debugging

Weather environment / other environments RL FAQ

If you are making a custom environment, I would:

1. Ask an Open Source LLM to generate a weather environment - ie it should say today's weather (sunny, not sunny), with temperature, humidity, wind speed etc etc. It should be governed by some physical laws or mathematical transition process (Markov Chains)
2. Create a reward function to predict the weather and do RL on this!

Get faster inference with vLLM

Unsloth supports faster inference for some models with `fast_inference = True`. For eg Llama, Qwen models. GPT-OSS will use Unsloth internal inference which is a little bit slower. See

[https://github.com/unslothai/notebooks/blob/main/nb/Advanced_Llama3_2_\(3B\)_GRPO_LoRA.ipynb](https://github.com/unslothai/notebooks/blob/main/nb/Advanced_Llama3_2_(3B)_GRPO_LoRA.ipynb) Do:

```
import os
os.environ["UNSLOTH_VLLM_STANDBY"] = "1" # [NEW] Extra 30% context lengths!
from unsloth import FastLanguageModel
max_seq_length = 2048 # Can increase for longer reasoning traces
lora_rank = 64 # Larger rank = smarter, but slower
model, tokenizer = FastLanguageModel.from_pretrained(
    model_name = "meta-llama/Llama-3.2-3B-Instruct",
    max_seq_length = max_seq_length,
    load_in_4bit = False, # False for LoRA 16bit
    fast_inference = True, # Enable vLLM fast inference
    max_lora_rank = lora_rank,
```

ReadTimeoutError for OpenEnv

If you see the error below:

```
else: # right
    newb = [list(reversed(slide(list(reversed(r)))) for r in board]
    if newb != board:
        return str(m)
return "0"
```

.	.	.	.
.	.	.	2
.	.	.	2
.	.	.	.

```
[2025-10-27 02:56:24] WARNING connectionpool.py:868: Retrying (Retry(total=2, connect=None, read=None, redirect=None, status=None)) after connection broken by 'ReadTimeoutError("HTTPConnectionPool(host='localhost', port=8081): Read timed out. (read timeout=15.0)")': /step
[2025-10-27 02:56:39] WARNING connectionpool.py:868: Retrying (Retry(total=1, connect=None, read=None, redirect=None, status=None)) after connection broken by 'ReadTimeoutError("HTTPConnectionPool(host='localhost', port=8081): Read timed out. (read timeout=15.0)")': /step
[2025-10-27 02:56:55] WARNING connectionpool.py:868: Retrying (Retry(total=0, connect=None, read=None, redirect=None, status=None)) after connection broken by 'ReadTimeoutError("HTTPConnectionPool(host='localhost', port=8081): Read timed out. (read timeout=15.0)")': /step
```

I added a fix since 1AM 27th October PT time. Please do:

1. Update Unsloth pip install --upgrade --force-reinstall --no-deps --no-cache-dir unsloth unsloth_zoo
2. Re-get notebook as "OpenEnv_NEW.ipynb":

wget

```
"https://raw.githubusercontent.com/unslothai/notebooks/refs/heads/main/nb
/OpenEnv_gpt_oss_(20B)_Reinforcement_Learning_2048_Game_BF16.ipynb"
"OpenEnv_NEW.ipynb"
```


vLLM GPT-OSS 20B / 120B

DO NOT UPDATE vLLM in the Docker!!

Do: pip install --upgrade numba numpy

Then serve it:

```
vllm serve unsloth/gpt-oss-20b \  
  --no-enable-prefix-caching \  
  --compilation-config '{"full_cuda_graph": true}' \  
  --port 8001 \  
  --max-model-len 48000 \  
  --gpu-memory-utilization 0.85
```

For 120b:

```
vllm serve unsloth/gpt-oss-120b \  
  --no-enable-prefix-caching \  
  --compilation-config '{"full_cuda_graph": true}' \  
  --port 8001 \  
  --max-model-len 24000 \  
  --gpu-memory-utilization 0.9
```

```
(APIServer pid=8919) INFO 10-27 08:49:23 [launcher.py:46] Route: /v1/responses, Methods: POST  
(APIServer pid=8919) INFO 10-27 08:49:23 [launcher.py:46] Route: /v1/responses/{response_id}, Methods: GET  
(APIServer pid=8919) INFO 10-27 08:49:23 [launcher.py:46] Route: /v1/responses/{response_id}/cancel, Methods: POST  
(APIServer pid=8919) INFO 10-27 08:49:23 [launcher.py:46] Route: /v1/chat/completions, Methods: POST  
(APIServer pid=8919) INFO 10-27 08:49:23 [launcher.py:46] Route: /v1/completions, Methods: POST  
(APIServer pid=8919) INFO 10-27 08:49:23 [launcher.py:46] Route: /v1/embeddings, Methods: POST  
(APIServer pid=8919) INFO 10-27 08:49:23 [launcher.py:46] Route: /pooling, Methods: POST  
(APIServer pid=8919) INFO 10-27 08:49:23 [launcher.py:46] Route: /classify, Methods: POST  
(APIServer pid=8919) INFO 10-27 08:49:23 [launcher.py:46] Route: /score, Methods: POST  
(APIServer pid=8919) INFO 10-27 08:49:23 [launcher.py:46] Route: /v1/score, Methods: POST  
(APIServer pid=8919) INFO 10-27 08:49:23 [launcher.py:46] Route: /v1/audio/transcriptions, Methods: POST  
(APIServer pid=8919) INFO 10-27 08:49:23 [launcher.py:46] Route: /v1/audio/translations, Methods: POST  
(APIServer pid=8919) INFO 10-27 08:49:23 [launcher.py:46] Route: /rerank, Methods: POST  
(APIServer pid=8919) INFO 10-27 08:49:23 [launcher.py:46] Route: /v1/rerank, Methods: POST  
(APIServer pid=8919) INFO 10-27 08:49:23 [launcher.py:46] Route: /v2/rerank, Methods: POST  
(APIServer pid=8919) INFO 10-27 08:49:23 [launcher.py:46] Route: /scale_elastic_ep, Methods: POST  
(APIServer pid=8919) INFO 10-27 08:49:23 [launcher.py:46] Route: /is_scaling_elastic_ep, Methods: POST  
(APIServer pid=8919) INFO 10-27 08:49:23 [launcher.py:46] Route: /invocations, Methods: POST  
(APIServer pid=8919) INFO 10-27 08:49:23 [launcher.py:46] Route: /metrics, Methods: GET  
(APIServer pid=8919) INFO: Started server process [8919]
```



AMD 

Good Luck!