

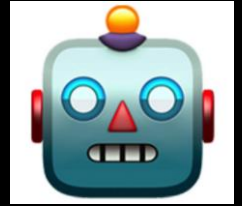


Using the AMD AI Assistant

Presenter: Giacomo Capodaglio
Date: October 13th, 2025
AMD @ CASTIEL

AMD 
together we advance_

Interact with the AMD AI Assistant: USERS BEWARE



Before we even start keep this in mind:

You will be using a large language model to fetch information.

Pros:

1. Easier than a Google search or deep dive into documentation
2. Has the latest information on AMD software as knowledge base
3. Can iterate to get further details through a conversation
4. Reports the sources it used to provide its answers so you can cross examine

Cons:

1. Can (and most likely will) provide incorrect information at times
2. The interaction is probabilistic: you will not get the same answer twice with the same prompt (most likely)
3. The model is not fine-tuned to be perfect: some answers will be better than others
4. You cannot consider the answers you receive as gold and will still have to validate them independently

Prompt Engineering – UPDATE THE EXAMPLES IN THIS SLIDE

- Prompt engineering means crafting and refining natural language instructions to guide AI models in generating accurate, relevant, and useful outputs.
- Several approaches:
 1. Be specific and detailed: “Write a HIP kernel that computes a reduction on an array of integers. Make sure the code works”
 2. Provide context and background Information: “I am building a shallow water model with periodic boundary conditions...”
 3. Include example code or templates (Few-Shot Learning): “Here is an example HIP kernel computing convolutions...”
 4. Specify the output format and programming language: “The code should be written in C++...”
 5. Assign a coding persona or role: “You are a C++ expert. Write a code that implements...”
 6. Break down complex tasks (Chain-of-Thought): “Step 1: recognize easily parallelizable loops. 2) Provide a parallel implementation”
 7. Use Multi-Step prompts and iterative refinement: “Parallelize the code. Next, refined the code adding error handling”
 8. Highlight constraints and best practices: “The code must show great performance, try to avoid atomic operations as much as possible.”
 9. Experiment with prompt variations: Zero/One/Few shots learning.

Example: *You are an expert [C++|C|Fortran] programmer. I want to convert a CPU code to run on an AMD GPU using OpenMP® standard 6.0. Use target directives and make sure the code works. Recognize cases when parallelism creates race conditions and take appropriate measures to avoid that.*

Key Elements of a Prompt

Definition

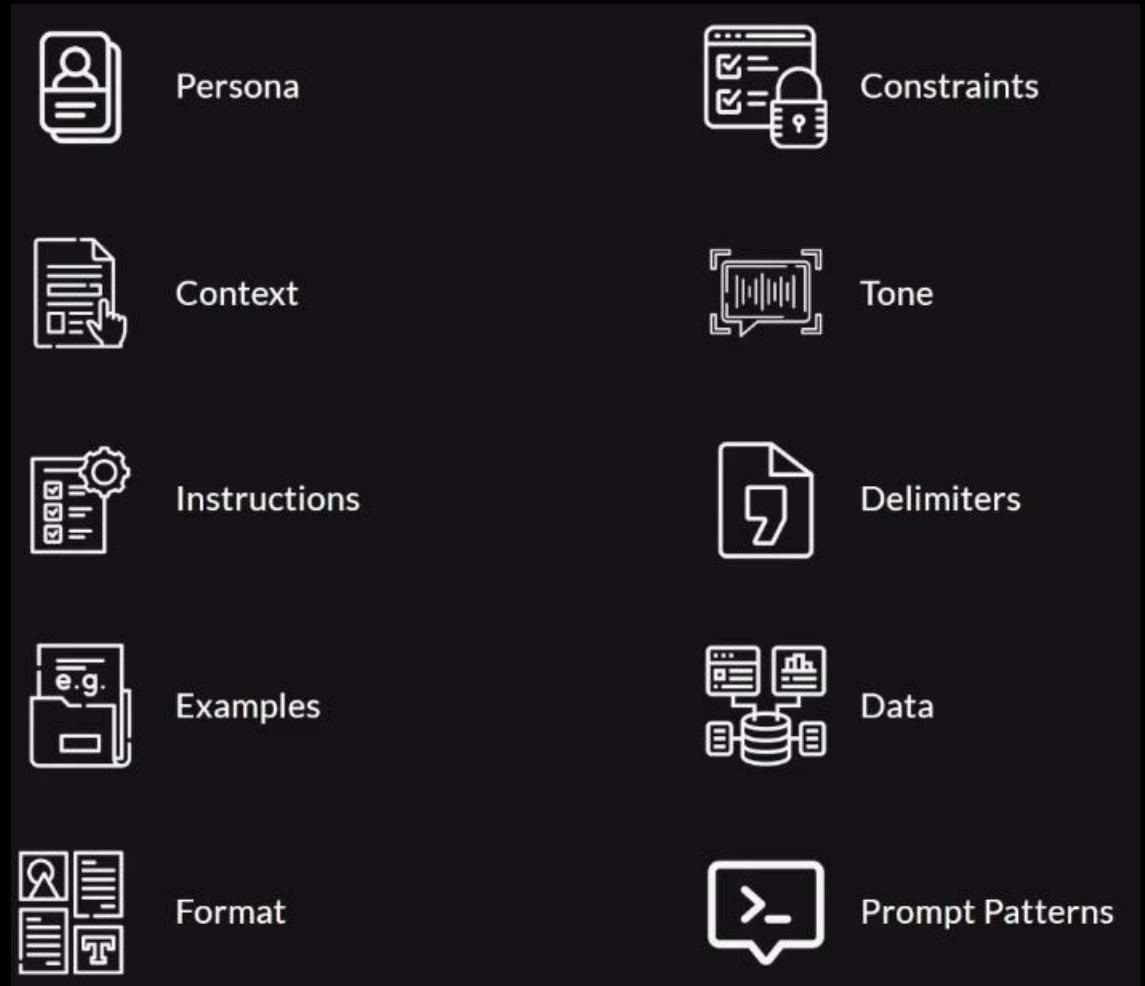
Prompts are initial text inputs given to a model that guide its responses

Purpose

They serve as instructions for the AI model to execute specific tasks

Goal

Prompt engineering refines these inputs to enhance accuracy and relevance



More on Prompt Engineering: best practice

Aspect	How to write good prompts	How not to write it
Clarity	Be explicit about the task and the desired outcome	Use vague or generic instructions
Specificity	Include detailed context or constraints (e.g., format, tone)	Provide insufficient or ambiguous details
Structure	Use step-by-step instruction if needed	Overload the prompt with multiple unrelated tasks
Audience Awareness	Specify the target audience or purpose of the response	Assume the AI understands implicit or broad context
Tone & Style	Indicate the desired tone or style (e.g., formal, casual, humorous, etc)	Leave tone and style undefined, leading to mismatched output
Examples	Provide examples to guide the AI's understanding	Avoid examples, event for complex tasks
Constraints	Define word limits, format or specific requirements	Ignore constraints, results in unpredictable outputs

More on Prompt Engineering: prompt styles

Name	Description	Advantages	Typical Use Cases
Role / Persona	Assign a specific role or identity to the model to influence the tone, structure, or type of response	- Controls style and voice - Enhances creativity or empathy - Simulates expert knowledge	- Tutoring (e.g., “Act as a math teacher”) - Customer support simulation - Creative writing or storytelling
Chain of Thought (CoT)	Encourages the model to reason step-by-step instead of jumping directly to an answer	- Improves reasoning accuracy, - Helps with complex, - multi-step problems	Math and logic problems, scientific reasoning, legal or ethical analysis
Self-Consistency	Ask the model to generate multiple reasoning paths and then choose the most frequent or sensible answer	Increases accuracy through majority consensus	Multiple-choice question answering, generative search/ranking
ReACT (Reasoning + Acting)	Combines reasoning (explaining thought process) with actions (e.g., tool use, search, function calls)	- Powerful for agents that need to use tools or APIs - Bridges reasoning with real-world execution	- Tool-using agents (e.g., calling APIs or databases) - Interactive tutoring systems - AI assistants with planning capabilities

Interact with the AMD AI Assistant: SSH tunnel

We are prototyping the use of an **AMD AI assistant** that has knowledge of the latest documentation and material on AMD GPU software and programming

You will be able to interact with this AI assistant during this training, to quickly fetch information that you will likely need as you navigate your way through the material

First thing to do: create an SSH tunnel to forward a local port (3000) to a remote server's port (3001) via an intermediate server (gpu-60)

From the terminal window you are using to log in to gpu-60 do:

```
ssh -L 3000:gpu-60:3001 193.143.78.200
```

(or alternatively `ssh -L 3000:gpu-60:3001 <username>@193.143.78.200`)

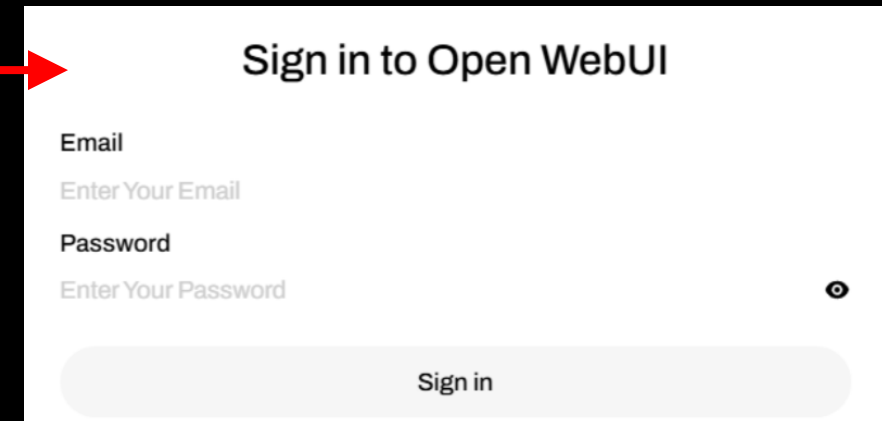
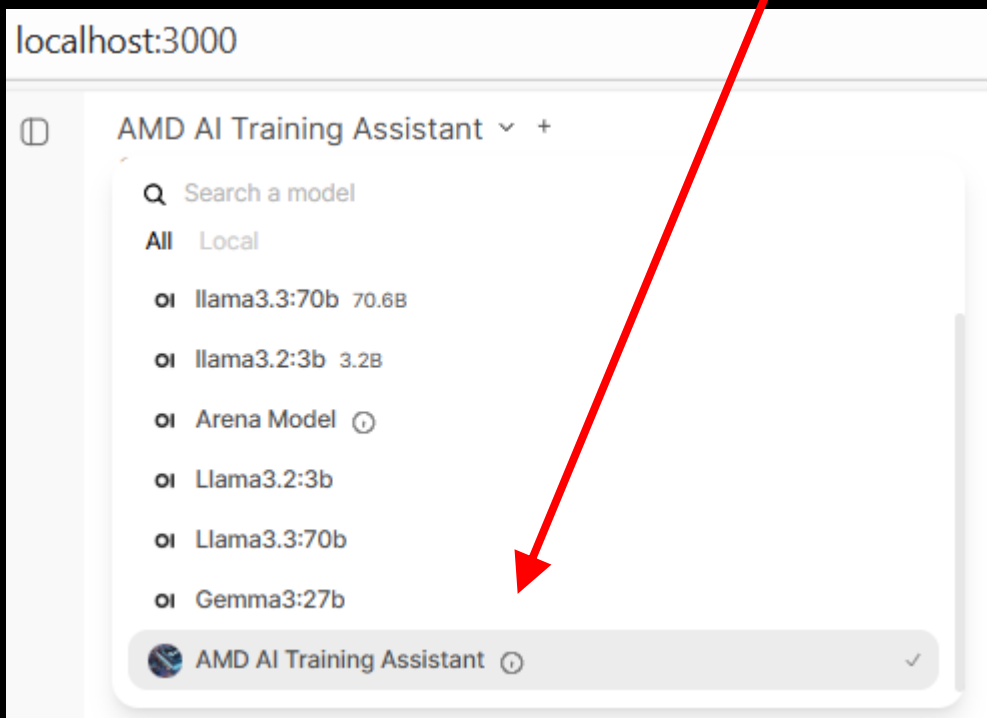
Then open a browser window, and on the address bar type:

```
http://localhost:3000
```

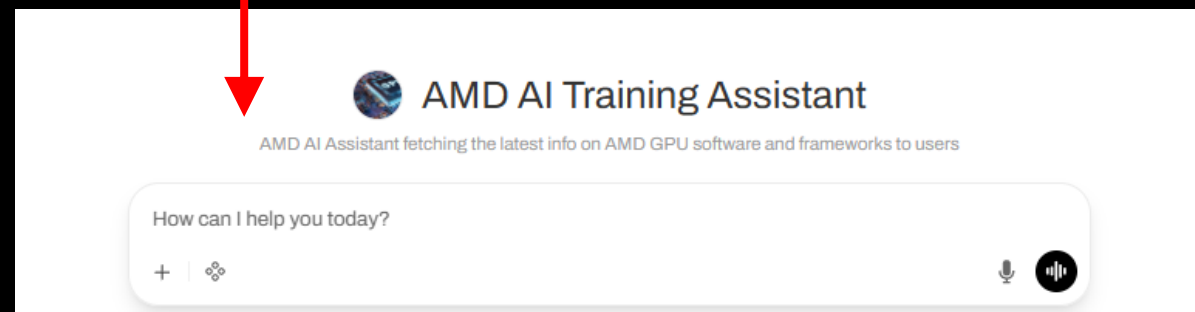
Interact with the AMD AI Assistant: logging in

1) Type the email and password that have been provided to you

2) Select the model from the drop-down list



3) Ask a question to the model



Model based on gemma3:27b



what is this

OI Gemma3:27b

Based on the image, this is a **cat**.

Specifically, it appears to be a **tabby cat**, as you can see the distinctive striped markings on its fur. It has beautiful green eyes and looks like a sweet and curious feline!

✎ 🗑 🔄 🔍 📌 🗑 🔄

Follow up

What are some common characteristics of tabby cats?

Can you tell me more about why cats have green eyes?

What breed of cat do you think this is?

How can I tell the difference between different cat breeds?

OI llama3.3:70b

It seems like you're trying to share an image, but as a text-based AI model, I don't have the capability to view or directly access images. If you can describe the image or provide more context about what it depicts or represents, I'd be happy to try and help you understand or discuss it further!

✎ 🗑 🔄 🔍 📌 🗑 🔄

Follow up

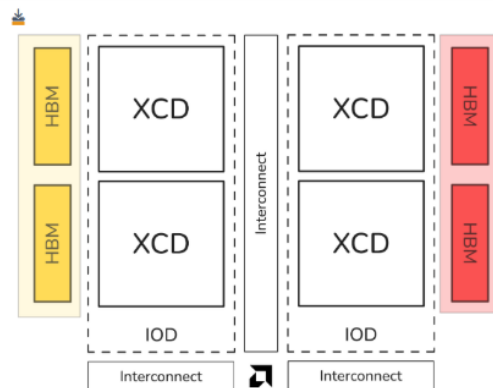
How can I describe an image effectively to get your help?

Are there any specific details I should focus on when...

Can you give me examples of how others have successful...

gemma3 is capable of recognizing images as inputs, llama3.3 is not

See how RAG knowledge refines the answer



o1 **Gemma3:27b**

Here's a breakdown of the diagram, describing the components and their connections:

Overall Architecture:

The diagram illustrates a multi-chip module (MCM) or chiplet-based design, likely targeting high-performance computing (HPC) or AI applications. It's composed of multiple smaller processing units connected together.

Components:

- **XCD (Compute Die):** These are the core processing units. Likely standing for Compute Die. They contain the processing logic. There are four of these in the diagram.
- **IOD (I/O Die):** These are Input/Output Dies. They act as interfaces for the Compute Dies, providing connections to external systems, memory, and other devices. There are two IODs here, each connecting to two XCDs.



AMD AI Training Assistant

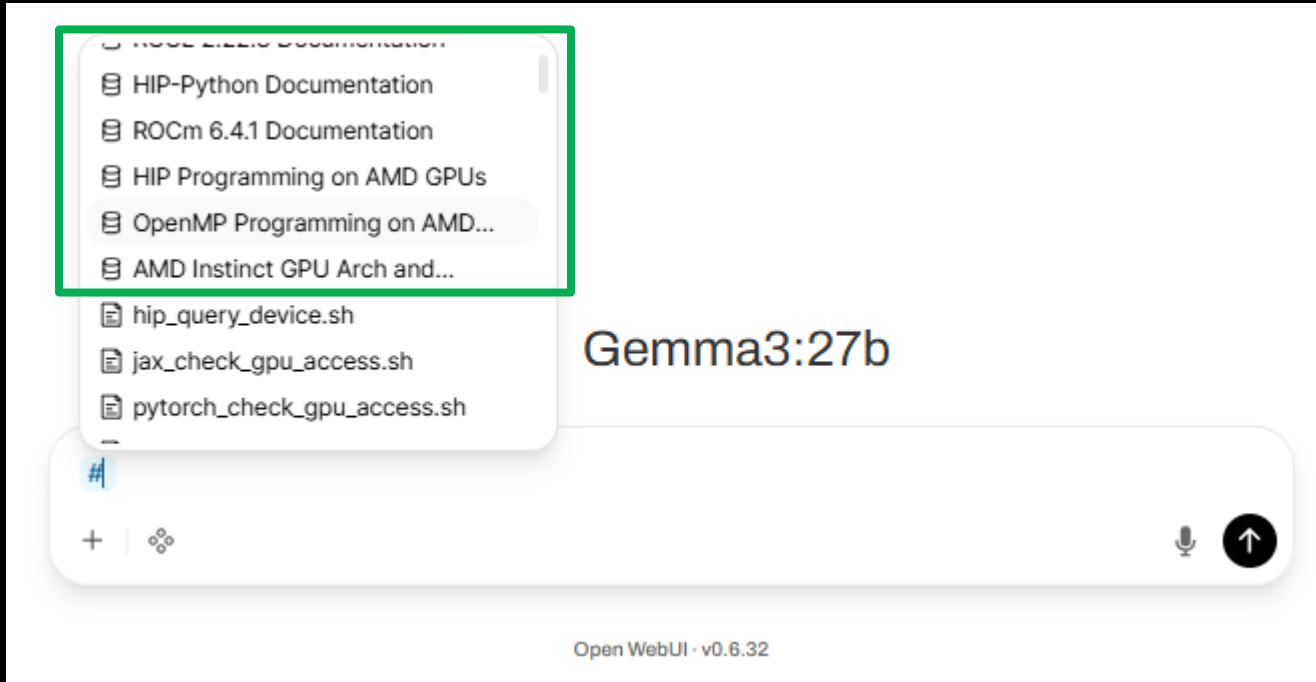
Okay, I've reviewed the provided text corpus. It's a substantial collection of materials detailing AMD's Instinct GPUs, the CDNA architecture, ROCm software stack, and related technologies. Here's a breakdown of the key themes and information, organized for clarity. I'll also highlight what seems to be the *most* important takeaways.

I. Core Technologies & Architectures

- **CDNA Architecture:** This is AMD's GPU architecture specifically designed for HPC and AI workloads. Key features include:
 - **Cache Coherency:** A major advantage, enabling simpler application development and potentially better performance, especially when working with both CPU and GPU memory.
 - **High Bandwidth:** Emphasis on fast memory access (HBM) and interconnects (Infinity Fabric).

the knowledge base for the AMD AI Training Assistant made the answer more accurate than the base model

Use any model and attach a knowledge base with



You can query a knowledge base or a single document in the knowledge base

Note that these will not have names that are as descriptive so we advise against doing that

Compare models on same knowledge base

Today at 2:38 AM

CASTIEL/AMD AI... Collection

What are two easy steps to start profiling pytorch workloads?

OI

Llama3.2:3b

Retrieved 8 sources

To start profiling PyTorch workloads, you can follow these two easy steps:

1. Run your PyTorch application with the `--torch-profile` flag enabled: `python micro_benchmarking_pytorch.py --network resnet50 --batch-size 32 --iterations 20 --torch-profile`

2. Use the `run_deepspeed_flops.sh` script to profile and benchmark your DeepSpeed FLOPS Profiler.

These steps will help you get started with profiling your PyTorch workloads and identifying potential bottlenecks in your application.

8 Sources

Follow up

Can you provide more information on how to use the `--`...

What are some common issues when using DeepSpeed...

How does one interpret the profiling results?

OI

Gemma3:27b

Retrieved 8 sources

Two easy steps to start profiling PyTorch workloads are to use the PyTorch Profiler and ROCm profiler integration. Specifically, you can start by running your code with `torch.profiler.profile()` as a context manager, and then integrate the ROCm profiler to further analyze performance

16_Training Ver...xample.pdf . You can also start with simple, high-level profiling tools and progress to more detailed ones as needed

13. AI Optimiza...erview.pdf .

8 Sources

Follow up

Can you show me a simple code example using...

What kind of performance insights does the ROCm profile...

What are some examples of 'simple, high-level profiling...

How do I interpret the results from `torch.profiler.profile()` `?

Model output comparison on PyTorch related question

what is the expected value of `torch.cuda.is_available()` if I am running PyTorch on AMD GPUs?

INCORRECT

Q1 Gemma3:27b

If you are running PyTorch on AMD GPUs, the expected value of `torch.cuda.is_available()` is **False**.

Here's why:

- `torch.cuda.is_available()` specifically checks for **NVIDIA CUDA compatibility**. The name itself implies this. It's designed to detect the presence of NVIDIA CUDA-enabled GPUs and the necessary drivers.
- **AMD GPUs do *not* use CUDA**. They rely on ROCm (Radeon Open Compute platform) which is a different GPU programming framework.
- **PyTorch's support for AMD GPUs**: PyTorch has ROCm support, but `torch.cuda.is_available()` *does not* check for ROCm. You'll need to use `torch.backends.rocm.is_available()` to check if ROCm is available and PyTorch is configured to use it.

Therefore, running `torch.cuda.is_available()` on a system with only AMD GPUs will consistently return **False**.

How to check for AMD GPU availability in PyTorch:

Use `torch.backends.rocm.is_available()` and `torch.backends.rocm.device_count()`.

CORRECT



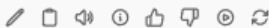
AMD AI Training Assistant

Based on the provided test suite and documentation, the expected value of `torch.cuda.is_available()` when running PyTorch on AMD GPUs is **True**.

The test suite explicitly checks for this condition (`test_pytorch_cuda_available.sh`) and relies on it being true to confirm a successful ROCm/PyTorch setup. The script aims to verify that PyTorch can see and utilize the AMD GPU.

Several other tests also depend on the assumption that `torch.cuda.is_available()` returns **True** when running on AMD GPUs with ROCm enabled.

61 Sources



Follow up

What specific ROCm versions are supported by this...

Does `torch.cuda.is_available()` provide any information...

If I'm having issues with PyTorch not recognizing my AMD...

Are there any known compatibility issues between PyTorch...

How does this compare to how `torch.cuda.is_available()`...

Compare the output of the same questions asked to the base model and to the AMD AI Assistant

One of the confusing things for PyTorch users interested in running on AMD GPUs is that the syntax is still `pytorch.cuda` even if you are querying AMD GPUs

Ask questions about ROCm enabled installations



AMD AI Training Assistant

Okay, here's a breakdown of how to install PyTorch to run on AMD GPUs, based on the provided documents. It's a bit involved, but here's the gist, along with key points and links to more detailed instructions.

Core Concept: ROCm

The key is **ROCm** (Radeon Open Compute platform). ROCm is AMD's software stack that enables GPU computing on AMD hardware. PyTorch needs to be built *with* ROCm support to utilize AMD GPUs.

Steps (Based on the Documents):

1. **Ensure ROCm is Installed:** This is the foundation. You need to have the ROCm platform installed and configured on your system *before* you attempt to install PyTorch. The documents don't provide detailed ROCm installation instructions, but you'll need to follow AMD's official ROCm installation guide for your specific Linux distribution. (This is a crucial first step, and the documents assume you've already done it).
2. **Load ROCm Modules:** After ROCm is installed, you'll need to load the ROCm modules in your shell. This makes the ROCm libraries and tools available to your environment. The test scripts consistently use:

```
bash
1 module load rocm
2 module load rocm pytorch
```

3. **Install PyTorch with ROCm Support:** The documents don't provide a single command for installation. However, they point to a setup script within a repository. You'll need to follow the instructions in the following repository:

https://github.com/amd/HPCTrainingDock/blob/main/extras/scripts/pytorch_setup.sh

This script likely handles the necessary steps to build or install a PyTorch version that's linked against ROCm.

Verification (Important):

After installation, it's *critical* to verify that PyTorch is correctly using your AMD GPU. The provided test scripts demonstrate how to do this:

- **Check GPU Visibility:**

```
bash
1 python3 -c "import torch; print(torch.cuda.get_device_name(0))"
```

This should print the name of your AMD GPU.

You can ask how to install AI software to run on AMD GPUs

The answer is suggesting to:

1. Install ROCm first
2. Load ROCm module
3. Install PyTorch with ROCm support
 - Note that it is referencing the script from our model installation repo
4. Verify that PyTorch can see the AMD GPU

Check out sources for more details

By following these steps, you should be able to install PyTorch and run it on your AMD GPU.

References from 1 [pytorch_check_gpu_name.sh](#) 2 [pytorch_check_gpu_access.sh](#) and 23 more ^

3 README91.md 4 Creating a PyTorch_TensorFlow code environment on AMD ... 5 README92.md 6 rocm_setup.sh
7 tensorflow_check_gpu_name.sh 8 pytorch_mnist_2gpus.batch 9 README86.md 10 pytorch_check_import.sh
11 Programming AMD GPUs with Julia — ROCm Blogs.pdf 12 Accelerating Large Language Models with Flash Attention o... 13 pytorch_setup.sh
14 pytorch_mnist_venv.batch 15 pytorch_mnist.batch 16 Deep dive into the MI300 compute and memory partition m...
17 Getting to Know Your GPU_ A Deep Dive into AMD SMI — R... 18 Developing Triton Kernels on AMD GPUs — ROCm Blogs.pdf
19 TorchTune on AMD GPUs How-To Guide_ Fine-tuning and S... 20 pytorch_mnist_venv_2gpus.batch 21 CMakeLists.txt 22 amd-cdna-3-white-paper.pdf
23 torchvision_check_import.sh 24 Day 0 Developer Guide_ Running the Latest Open Models fr...
25 GEMM Kernel Optimization For AMD GPU — ROCm Blogs.p...



At the end of every response, you can examine the sources that have been used to provide the answer

Citation

Relevance

84.88% (0.8488)

Content

```
#!/bin/bash
```

```
# This test checks that the GPU  
# seen by PyTorch is an AMD GPU
```

```
# NOTE: this test assumes PyTorch has been installed according  
# to the instructions available in the model installation repo:  
# https://github.com/amd/HPCTrainingDock/blob/main/extras/scripts/pytorch_setup.sh
```

```
module load rocm  
module load rocm pytorch
```

```
python3 -c "import torch; print(f'device name [0]:', torch.cuda.get_device_name(0))"
```

You can click on each source and examine it

Each source has a percentage of relevance that the model has attributed to it

A note on the legal implications of using AI-generated code

1. Copyright and Ownership

- No Human, No Copyright: In the U.S. and EU, AI-generated code without human authorship is not protected by copyright – it defaults to public domain
- Partial Protection: If a human modifies AI output significantly, they may claim copyright over those modifications but not the raw AI output

2. Attribution & Licensing Risks

- Open-Source Compliance: AI may generate GPL, MIT, or Apache-licensed code, triggering obligations (e.g., attribution, license adherence)

3. Infringement & Legal Risks

- Direct Copying: AI may reproduce verbatim or near-verbatim segments from training data, risking copyright infringement
- Derivative Works: Modifying or translating protected code via AI may be an infringement, if the AI output is too similar to the original



Takeaway: AI-generated code **may not be copyrightable** but **can still infringe copyrights**. Treat AI as an assistant, **not an autonomous coder**

Disclaimer

The information presented in this document is for informational purposes only and may contain technical inaccuracies, omissions, and typographical errors. The information contained herein is subject to change and may be rendered inaccurate for many reasons, including but not limited to product and roadmap changes, component and motherboard version changes, new model and/or product releases, product differences between differing manufacturers, software changes, BIOS flashes, firmware upgrades, or the like. Any computer system has risks of security vulnerabilities that cannot be completely prevented or mitigated. AMD assumes no obligation to update or otherwise correct or revise this information. However, AMD reserves the right to revise this information and to make changes from time to time to the content hereof without obligation of AMD to notify any person of such revisions or changes.

THIS INFORMATION IS PROVIDED 'AS IS.' AMD MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND ASSUMES NO RESPONSIBILITY FOR ANY INACCURACIES, ERRORS, OR OMISSIONS THAT MAY APPEAR IN THIS INFORMATION. AMD SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL AMD BE LIABLE TO ANY PERSON FOR ANY RELIANCE, DIRECT, INDIRECT, SPECIAL, OR OTHER CONSEQUENTIAL DAMAGES ARISING FROM THE USE OF ANY INFORMATION CONTAINED HEREIN, EVEN IF AMD IS EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Third-party content is licensed to you directly by the third party that owns the content and is not licensed to you by AMD. ALL LINKED THIRD-PARTY CONTENT IS PROVIDED "AS IS" WITHOUT A WARRANTY OF ANY KIND. USE OF SUCH THIRD-PARTY CONTENT IS DONE AT YOUR SOLE DISCRETION AND UNDER NO CIRCUMSTANCES WILL AMD BE LIABLE TO YOU FOR ANY THIRD-PARTY CONTENT. YOU ASSUME ALL RISK AND ARE SOLELY RESPONSIBLE FOR ANY DAMAGES THAT MAY ARISE FROM YOUR USE OF THIRD-PARTY CONTENT.

© 2025 Advanced Micro Devices, Inc. All rights reserved. AMD, the AMD Arrow logo, AMD CDNA, AMD ROCm, AMD Instinct, and combinations thereof are trademarks of Advanced Micro Devices, Inc. in the United States and/or other jurisdictions. Other names are for informational purposes only and may be trademarks of their respective owners.

qwq and qwen2.5-coder are open-source models from Alibaba Cloud

The OpenMP® name and the OpenMP® logo are registered trademarks of the OpenMP Architecture Review Board

