
Use ROCm™ on Radeon™ GPUs Documentation

Advanced Micro Devices, Inc.

Jan 05, 2026

CONTENTS

1 Prerequisites to use ROCm™ on Radeon™ desktop GPUs for machine learning development	3
1.1 Supported hardware	3
1.2 Supported operating systems	3
1.3 Recommended system configuration	4
2 How to guide - Use ROCm™ on Radeon™ GPUs	5
2.1 Install Radeon software for Linux with ROCm	5
2.2 Install PyTorch for ROCm	8
3 Compatibility matrices	11
3.1 ROCm 5.7	11
4 Limitations	13
4.1 Multi-GPU configuration	13
5 AI community	15
6 Report a bug	17

Turn your desktop into a Machine Learning platform with the latest AMD Radeon™ 7900 series GPUs

AMD has expanded support for PyTorch Machine Learning Development on RDNA™ 3 GPUs with Radeon Software for Linux 23.20.00.48 with ROCm 5.7!

Researchers and developers working with Machine Learning (ML) models and algorithms using PyTorch can now also use ROCm 5.7 on Linux® to tap into the parallel computing power of the latest AMD Radeon 7900 series desktop GPUs which are based on the AMD RDNA 3 GPU architecture.

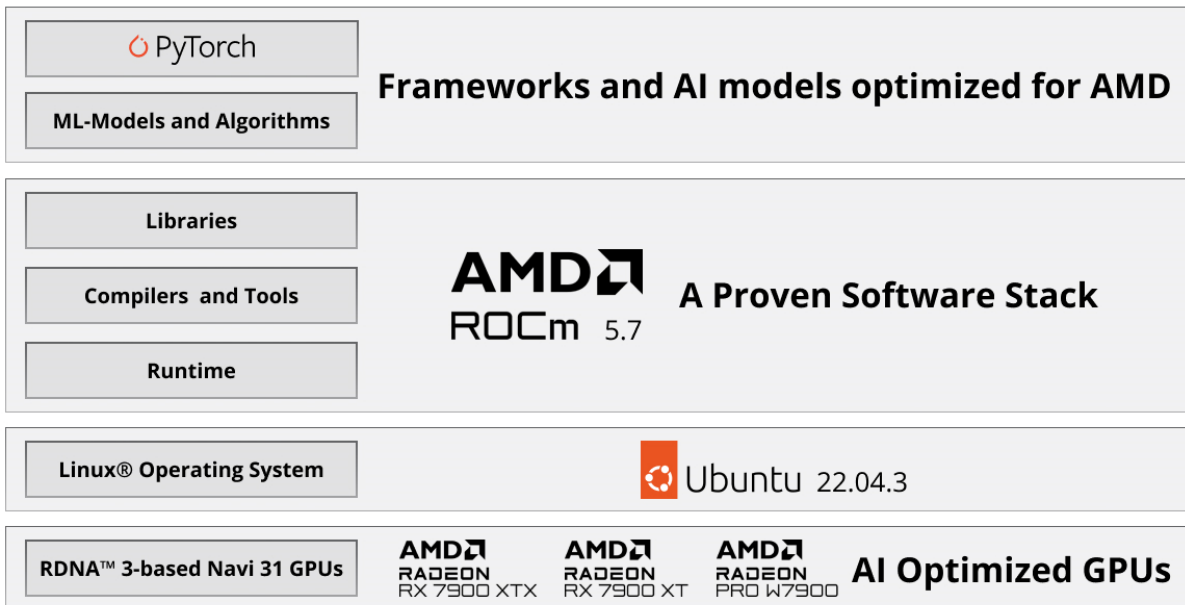
A client solution built on powerful high-end AMD GPUs enables a local, private and often more cost-effective workflow to develop ROCm and train ML (PyTorch) for the users who previously relied solely on cloud-based solutions.

More ML performance for your desktop

- With today’s models easily exceeding the capabilities of standard hardware and software not designed for AI, ML engineers are looking for cost-effective solutions to develop and train their ML-powered applications. Due to the availability of significantly large GPU memory sizes of 24GB or 48GB, utilization of a local PC or workstation equipped with Radeon 7900 series GPU offers a robust/potent yet economical option to meet these expanding ML workflow challenges.
- Radeon 7900 series GPUs are built on the RDNA 3 GPU architecture,
 - featuring more than 2x higher AI performance per Compute Unit (CU) compared to the previous generation
 - now comes with up to 192 AI accelerators
 - offers up to 24GB or 48GB of GPU memory to handle large ML models.



ML Development on the Desktop



Note

Based on AMD internal measurements, November 2022, comparing the Radeon RX 7900 XTX at 2.505 GHz boost clock with 96 CUs issuing 2X the Bfloat16 math operations per clock vs. the Radeon RX 6900 XT GPU at 2.25 GHz boost clock and 80 CUs issue 1X the Bfloat16 math operations per clock. Results may vary. RX-821.

Migrate your application from the desktop to the datacenter

- ROCm is the open-source software stack for Graphics Processing Unit (GPU) programming. ROCm spans several domains: General-Purpose computing on GPUs (GPGPU), High Performance Computing (HPC) and heterogeneous computing.
- The latest AMD ROCm 5.7 software stack for GPU programming unlocks the massively parallel compute power of these RDNA 3 GPUs for use with PyTorch 2.0.1, one of the leading ML frameworks. The same software stack also supports the AMD CDNA™ GPU architecture so developers can migrate their PyTorch-based applications into the datacenter.

Freedom to customize

ROCm is primarily Open-Source Software (OSS) that allows developers the freedom to customize and tailor their GPU software for their own needs while collaborating with a community of other developers, and helping each other find solutions in an agile, flexible, rapid and secure manner. AMD ROCm allows users to maximize their GPU hardware investment. ROCm is designed to help develop, test and deploy GPU accelerated HPC, AI, scientific computing, CAD, and other applications in a free, open-source, integrated and secure software ecosystem.

Note

Visit [AMD ROCm Documentation](#) for the latest on ROCm.

PREREQUISITES TO USE ROCm™ ON RADEON™ DESKTOP GPUS FOR MACHINE LEARNING DEVELOPMENT

Before you start with the installation, it's essential to ensure that your system meets the necessary requirements such as supported hardware, a compatible operating system, and the recommended system configuration to ensure optimal performance and functionality.

See *Compatibility matrices* for more information.

1.1 Supported hardware

1.1.1 Supported graphics processing units

To successfully install ROCm for machine learning development, ensure that your system is operating on one of the following Radeon Desktop GPUs:

- Radeon RX 7900 XTX
- Radeon RX 7900 XT
- Radeon PRO W7900

1.1.2 Recommended memory

These specifications are required for complex AI/ML workloads:

- 64GB Main Memory
- 24GB GPU Video Memory

Minimum recommendations

Note that low system memory may cause issues running inference models on CPU.

- 16GB Main Memory
- 8GB GPU Video Memory

1.2 Supported operating systems

To successfully install ROCm™ for machine learning development, ensure that your operating system is up-to-date.

Supported operating systems include:

- Ubuntu® 22.04.3 Desktop Version with hardware enablement kernel (HWE)

1.2.1 Update Ubuntu® operating system

Use the following commands to bring your OS up-to-date:

```
sudo apt-get update
sudo apt-get dist-upgrade
```

1.3 Recommended system configuration

This section guides users on how to optimize system configuration for ROCm™ usage, ensuring smooth and performant ROCm operation.

1.3.1 Disable iGPU

The iGPU is non-essential for AI and ML workloads and not officially supported. Disable iGPU in SBIOS before proceeding to avoid unknown issues.

Alternatively, use environment variables to select the target GPU.

Here are examples to disable iGPU on some AMD motherboards:

Gigabyte™ X670 AORUS ELITE AX

1. Enter BIOS

Path: *Advanced* → *AMD CBS* → *NBIO Common Options* → *GFX Configuration* → *iGPU Configuration*

2. Set iGPU to `Disabled`

ASUS Prime X670-P WIFI

1. Enter BIOS

Path: *Advanced* → *NB Configuration* → *Integrated Graphics*

2. Set to `Disabled`

NOTE: This step only applies to AMD motherboards, no action is required for non-AMD motherboards.

There are no minimum motherboard hardware requirements.

Alternative option: Use environment variables to select target GPU

An alternative option to disabling the iGPU is to use environment variable to select the GPU.

See [GPU Isolation Techniques](#) to specify the device indices you would like to expose to your application.

HOW TO GUIDE - USE ROCM™ ON RADEON™ GPUS

In this guide we will walk you through the ROCm and PyTorch installation processes to pair with the latest AMD Radeon 7900 series desktop GPUs, and get started on a fully-functional environment for AI and ML development.

2.1 Install Radeon software for Linux with ROCm

The ROCm Software Stack and other Radeon software for Linux components are installed using the `amdgpu-install` script to assist you in the installation of a coherent set of stack components.

- Simplifies the installation of the AMDGPU stack by encapsulating the distribution specific package installation logic and by using command line options that allows you to specify the:
 - Usecase of the AMDGPU stack to be installed (Graphics or Workstation)
 - Combination of components (Pro stack, or user selection)
- Performs post-install checks to verify whether the installation was performed successfully.
- Installs the uninstallation script to allow you to remove the whole AMDGPU stack from the system by using a single command.

The script is provided by the installer package. See *Compatibility matrices* for support information.

2.1.1 Install AMD unified driver package repositories and installer script

Download and install the `amdgpu-install` script on the system.

Enter the following commands to install the installer script for Ubuntu® version 22.04.3:

```
sudo apt update
wget https://repo.radeon.com/amdgpu-install/23.20.00.48/ubuntu/jammy/amdgpu-install_5.
↳7.00.48.50700-1_all.deb
sudo apt install ./amdgpu-install_5.7.00.48.50700-1_all.deb
```

2.1.2 Install AMD unified kernel-mode GPU driver, ROCm, and graphics

After the Unified Driver Deb Package repositories are installed, run the installer script with appropriate `--usecase` parameters to install the driver components.

AMD recommends installing the Graphics usecase by default. Only consider the alternative install option if you have an applicable Workstation usecase scenario.

Enter the following command to display a list of available usecases:

```
sudo amdgpu-install --list-usecase
```

Install the Graphics usecase

1. Run the following command to install open source graphics and ROCm.

```
amdgpu-install -y --usecase=graphics,rocm
```

Watch for output warning or errors indicating an unsuccessful driver installation.

NOTE: The `-y` option installs non-interactively. This step may take several minutes, depending on internet connection and system speed.

2. Reboot the system.

```
sudo reboot
```

See [Using the amdgpu-install script](#) for more information.

Alternative install option: Workstation usecase

1. Run the following command to install workstation graphics and ROCm.

```
amdgpu-install -y --usecase=workstation,rocm
```

NOTE: The `-y` option installs non-interactively. This step may take several minutes, depending on internet connection and system speed.

2. Reboot the system.

```
sudo reboot
```

See [Using the amdgpu-install script](#) for more information.

Set permissions for Groups to allow access to GPU hardware resources

Once the driver is installed, add any current user to the render and video groups to access GPU resources.

Reboot in order for group changes to take effect.

Add user to render and video groups

1. Enter the following command to check groups in the system:

```
groups
```

2. Add user to the render and video group using the command:

```
sudo usermod -a -G render,video $LOGNAME
```

3. Reboot the system.

```
sudo reboot
```

See [Setting Permissions for Groups](#) for more information.

Post-install verification checks

Run these post-installation checks to verify that the installation is complete:

1. Verify that the current user is added to the render and video groups.

```
groups
```

Expected result:

```
<username> adm cdrom sudo dip video plugdev render lpadmin lxd sambashare
```

<username> indicates the current user, and this result will vary in your environment.

2. Check if *amdgpu* kernel driver is installed.

```
dkms status
```

Expected result:

```
amdgpu/x.x.x-xxxxxxx.xx.xx, x.x.x-xx-generic, x86_64: installed
```

3. Check if the GPU is listed as an agent.

```
rocminfo
```

Expected result:

```
[...]
*****
Agent 2
*****
  Name:                gfx1100
  Uuid:                GPU-5ecee39292e80c37
  Marketing Name:     Radeon RX 7900 XTX
  Vendor Name:        AMD
  [...]
[...]
```

4. Check if the GPU is listed.

```
clinfo
```

Expected result:

```
[...]
  Platform Name:        AMD Accelerated Parallel Processing
  Number of devices:    1
  Device Type:         CL_DEVICE_TYPE_GPU
  Vendor ID:           1002h
  Board name:          Radeon RX 7900 XTX
  [...]
[...]
```

See [Installing the all open usecase](#) for additional troubleshooting tips.

2.1.3 Advanced install methods

For advanced install methods, such as Multi-Version and Package Manager, refer to [AMD GPU Install Script](#).

2.1.4 Uninstall ROCm

Run the following command to uninstall the ROCm software stack and other Radeon software for Linux components:

```
sudo amdgpu-uninstall
```

2.1.5 Upgrade to newer versions of Radeon software for Linux

The recommended method to upgrade is to uninstall, followed by an install.

Radeon Software for Linux does not support in-place upgrades.

2.2 Install PyTorch for ROCm

Refer to this section for the recommended PyTorch via PIP installation method, as well as Docker based installation.

2.2.1 Option A: PyTorch via PIP installation method

AMD recommends the PIP install method to create a PyTorch environment when working with ROCm for machine learning development.

Check [Pytorch.org](https://pytorch.org) for latest PIP install instructions and availability. See *Compatibility matrices* for support information.

To install PyTorch,

1. Enter the following command to unpack and begin set up.

```
sudo apt install python3-pip -y
```

2. Enter this command to update the pip wheel.

```
pip3 install --upgrade pip wheel
```

3. Enter this command to install Torch and Torchvision for ROCm AMD GPU support.

```
wget https://repo.radeon.com/rocm/manylinux/rocm-rel-5.7/torch-2.0.1%2Brocm5.7-  
→cp310-cp310-linux_x86_64.whl  
wget https://repo.radeon.com/rocm/manylinux/rocm-rel-5.7/torchvision-0.15.2  
→%2Brocm5.7-cp310-cp310-linux_x86_64.whl  
pip3 install --force-reinstall torch-2.0.1+rocm5.7-cp310-cp310-linux_x86_64.whl_  
→torchvision-0.15.2+rocm5.7-cp310-cp310-linux_x86_64.whl
```

This may take several minutes.

Important! AMD recommends proceeding with ROCm WHLs available at repo.radeon.com. The ROCm WHLs available at [PyTorch.org](https://pytorch.org) are not tested extensively by AMD as the WHLs change regularly when the nightly builds are updated.

Next, *verify your PyTorch installation*.

2.2.2 Option B: Docker installation method

Using Docker provides portability, and access to a prebuilt Docker container that has been rigorously tested within AMD. Docker also cuts down compilation time, and should perform as expected without installation issues.

Prerequisites

Docker for Ubuntu® must be installed.

To install Docker for Ubuntu, enter the following command:

```
sudo apt install docker.io
```

Use Docker image with pre-installed PyTorch

Follow these steps for installing using a Docker image.

1. Enter the following command to pull the public PyTorch Docker image.

```
sudo docker pull rocm/pytorch:rocm5.7_ubuntu22.04_py3.10_pytorch_2.0.1
```

Optional: You can also download a specific and supported configuration with different user-space ROCm versions, PyTorch versions, and supported operating systems.

Refer to hub.docker.com/r/rocm/pytorch to download the PyTorch Docker image.

2. Start a Docker container using the downloaded image.

```
sudo docker run -it --cap-add=SYS_PTRACE --security-opt seccomp=unconfined --
↪device=/dev/kfd --device=/dev/dri --group-add video --ipc=host --shm-size 8G
↪rocm/pytorch:rocm5.7_ubuntu22.04_py3.10_pytorch_2.0.1
```

This will automatically download the image if it does not exist on the host. You can also pass the `-v` argument to mount any data directories from the host onto the container.

Next, *verify your PyTorch installation*.

See [PyTorch Installation for ROCm](#) for more information.

2.2.3 Verify PyTorch installation

Confirm if PyTorch is correctly installed.

1. Verify if Pytorch is installed and detecting the GPU compute device.

```
python3 -c 'import torch' 2> /dev/null && echo 'Success' || echo 'Failure'
```

Expected result:

```
Success
```

2. Enter command to test if the GPU is available.

```
python3 -c 'import torch; print(torch.cuda.is_available())'
```

Expected result:

```
True
```

3. Enter command to display installed GPU device name.

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

Expected result: Example: *device name [0]: Radeon RX 7900 XTX*

```
device name [0]: <Supported AMD GPU>
```

4. Enter command to display component information within the current PyTorch environment.

```
python3 -m torch.utils.collect_env
```

Expected result:

```
PyTorch version
ROCM used to build PyTorch
OS
Is CUDA available
GPU model and configuration
HIP runtime version
MIOpen runtime version
```

Environment set-up is complete, and the system is ready for use with PyTorch to work with machine learning models, and algorithms.

COMPATIBILITY MATRICES

This section provides information on the compatibility of ROCm™ components, Radeon™ GPUs, and the Radeon Software for Linux® version (Kernel Fusion Driver).

3.1 ROCm 5.7

These are the applicable matrices for ROCm 5.7.

3.1.1 GPU support matrix

ROCm Version	Radeon Software for Linux® Version	Supported AMD Radeon Hardware
5.7.0	23.20.00.48	AMD Radeon RX 7900 XTXAMD Radeon RX 7900 XTAMD Radeon PRO W7900

3.1.2 PyTorch + ROCm support matrix

PyTorch Version	ROCm Version	Comments
2.0.1	5.7	Official production support. Available from AMD.com.
2.1+/Nightly	5.7	Available from PyTorch.org nightly builds, not tested extensively by AMD.
2.1/Stable	5.6	Not supported for Radeon 7000 series.

3.1.3 Docker support matrix

See [Docker Image Support Matrix](#) for the latest version of the software support matrices for ROCm container releases.

LIMITATIONS

This section provides information on software and configuration limitations.

4.1 Multi-GPU configuration

Due to limited validation of ROCm on Radeon multi-GPU configuration at this time, we have identified common errors, and applicable recommendations.

Important! ROCm 5.7 release is limited to preview support for multi-GPU configuration.

At this time, only a limited amount of validation has been performed. AMD recommends only proceeding with advanced know-how and at user discretion.

Visit the [AI community](#) to share feedback, and [Report a bug](#) if you find any issues.

4.1.1 Recommended system configuration for multi-GPU

PCIe® slots connected to the GPU must have identical PCIe lane width or bifurcation settings, and support PCIe 3.0 Atomics.

Refer to [How ROCm uses PCIe Atomics](#) for more information.

Example:

- ✓ - GPU0 PCIe x16 connection + GPU1 PCIe x16 connection
- ✓ - GPU0 PCIe x8 connection + GPU1 PCIe x8 connection
- X - GPU0 PCIe x16 connection + GPU1 PCIe x8 connection

Important!

- Only use PCIe slots connected by the CPU and to avoid PCIe slots connected via chipset. Refer to product-specific motherboard documentation for PCIe electrical configuration.
- Ensure the PSU has sufficient wattage to support multiple GPUs.

4.1.2 Errors due to GPU and PCIe configuration

When using two AMD Radeon 7900XTX GPUs, the following HIP error is observed when running PyTorch micro-benchmarking if any one of the two GPUs are connected to a non-CPU PCIe slot (PCIe on chipset):

```
RuntimeError: HIP error: the operation cannot be performed in the present state
HIP kernel errors might be asynchronously reported at some other API call, so the
↳stacktrace below might be incorrect.
For debugging consider passing HIP_LAUNCH_BLOCKING=1.
Compile with [ ]TORCH_USE_HIP_DSA[ ] to enable device-side assertions.
```


AI COMMUNITY

Want to share your experiences, find answers, or contribute to resolving issues?

Explore the [AMD AI Community Forum](#), where you will find a like-minded community, passionate about all things AI!

REPORT A BUG

Found a defect? Report issues through [ROCm GitLab](#), and contribute to improving our user experience.