



# prep me a full setup guide for gpu pc post first time pipeline run,

PC: windows 11, gpu- 7700XT 12gb vram, higg end cpu(should give no bottleneck issues), 16gb ram.

should include steps like git cloning, cuda setup or changes, etc

This is critical information to address first before the full guide.

## ⚠ Important: RX 7700 XT is NOT Supported on Windows ROCm

AMD's official ROCm Windows support matrix as of ROCm 7.2 only lists **RX 9070 (gfx1201/RDNA4)** for Windows PyTorch. The RX 7700 XT is RDNA3 (gfx1101) — officially unlisted for Windows. <sup>[1]</sup> <sup>[2]</sup>

You have three realistic options:

Option	Effort	Reliability
<b>WSL2 on Windows</b> (Ubuntu inside Windows, ROCm on Linux)	Medium	✓ Best — Linux ROCm fully supports RDNA3
<b>Dual boot Ubuntu</b>	Low	✓ Best performance
<b>DirectML</b> (Windows-native, no ROCm)	Low	⚠ Works but slower than ROCm

**WSL2 is the recommended path** — you keep Windows 11, but PyTorch runs inside a Ubuntu 22.04 subsystem where RDNA3 ROCm support is mature and stable.

## Full Setup Guide — CivicPulse on GPU PC (WSL2 Path)

### Phase 1 — Windows Prerequisites

#### Step 1 — Enable WSL2:

```
# Run in PowerShell as Administrator
wsl --install
wsl --set-default-version 2
# Reboot when prompted
```

#### Step 2 — Install Ubuntu 22.04:

```
wsl --install -d Ubuntu-22.04
# Launch Ubuntu from Start Menu, set username + password
```

### Step 3 — Install Git on Windows (for repo management from Windows side):

```
winget install Git.Git
```

## Phase 2 — ROCm + Python inside WSL2 (Ubuntu)

All commands below run inside the **Ubuntu WSL2 terminal**:

### Step 4 — System update:

```
sudo apt update && sudo apt upgrade -y
sudo apt install -y python3.11 python3.11-venv python3-pip git wget curl
```

### Step 5 — Install ROCm 6.x for RDNA3:

```
# Add AMD ROCm repo
wget https://repo.radeon.com/amdgpu-install/6.1.3/ubuntu/jammy/amdgpu-install_6.1.60103-1
sudo apt install ./amdgpu-install_6.1.60103-1_all.deb
sudo amdgpu-install --usecase=rocm --no-dkms
sudo usermod -aG render,video $USER
# Reboot WSL
wsl --shutdown # run this in Windows PowerShell, then reopen Ubuntu
```

### Step 6 — Verify ROCm sees your GPU:

```
rocm-smi
# Should show: Radeon RX 7700 XT
rocminfo | grep gfx
# Should show: gfx1101
```

## Phase 3 — Clone Repo + Python Environment

### Step 7 — Clone your CivicPulse repo:

```
cd ~
git clone https://github.com/YOUR_USERNAME/civicpulse.git
cd civicpulse
```

### Step 8 — Create venv and install PyTorch ROCm:

```
python3.11 -m venv .venv
source .venv/bin/activate

# PyTorch with ROCm 6.1 for gfx1101 (RDNA3)
pip install --upgrade pip
pip install torch torchvision torchaudio \
    --index-url https://download.pytorch.org/whl/rocm6.1

# Project dependencies
pip install numpy h5py rasterio geopandas pandas \
    tqdm jupyter python-dotenv matplotlib
```

## Step 9 — Verify GPU is available to PyTorch:

```
python -c "
import torch
print('ROCM available:', torch.cuda.is_available())
print('GPU:', torch.cuda.get_device_name(0))
print('VRAM:', round(torch.cuda.get_device_properties(0).total_memory/1e9, 1), 'GB')
"

# Expected:
# ROCm available: True
# GPU: Radeon RX 7700 XT
# VRAM: 12.0 GB
```

## Phase 4 — Transfer Data + Config

### Step 10 — Copy data from Windows into WSL2:

```
# Your Windows C: drive is mounted at /mnt/c in WSL2
cp /mnt/c/Users/YOUR_NAME/civicpulse/data/processed/india_sample.h5 \
    ~/civicpulse/data/processed/
```

### Step 11 — Create .env for GPU PC:

```
cat > ~/civicpulse/.env << EOF
CIVICPULSE_DEVICE=cuda
CIVICPULSE_BATCH_SIZE=128
CIVICPULSE_DATA_MODE=normal
CIVICPULSE_PATCH_SIZE=256
CIVICPULSE_HIDDEN_CHANNELS=64
CIVICPULSE_NUM_LAYERS=2
CIVICPULSE_LEARNING_RATE=0.001
EOF
```

**Note:** config.py uses torch.cuda which ROCm maps to — CIVICPULSE\_DEVICE=cuda works unchanged. No code edits needed.

**Step 12 — Run the benchmark notebook first** to confirm optimal batch size for your 12GB VRAM:

```
jupyter notebook --no-browser --port=8888
# Open in Windows browser: http://localhost:8888
# Run 03-PERF_Performance_Benchmark.ipynb
# Update .env with the benchmarked BATCH_SIZE
```

## Phase 5 — Training

**Step 13 — Set full epochs in NB05 Cell 5:**

```
training_stages = {
    "stage1_coarse": dict(..., epochs=20, ...),
    "stage2_medium": dict(..., epochs=50, ...),
    "stage3_fine": dict(..., epochs=100, ...),
}
```

**Step 14 — Enable DataLoader workers** (Linux WSL2 has no Windows multiprocessing issue):

```
# In Cell 7 – safe to use num_workers on Linux
train_loader = DataLoader(train_set, batch_size=cfg["batch_size"],
                          shuffle=True, num_workers=4,
                          pin_memory=True, persistent_workers=True)
```

## Phase 6 — Push Checkpoint Back to GitHub

After training completes:

```
cd ~/civicpulse
git add models/checkpoints/best_model.pt logs/training_summary.json
git commit -m "full GPU training run - RX7700XT - 20/50/100 epochs"
git push origin main
```

Then on your local PC:

```
git pull # best_model.pt now available locally for NB06/07
```

## Expected GPU Training Time

With 12GB VRAM on RX 7700 XT vs your current CPU:

Stage	CPU (847s/ep)	GPU estimate
S1 — 20 ep	~4.7h	~8 min
S2 — 50 ep	~47h	~45 min
S3 — 100 ep	~188h	~2.5h
<b>Total</b>	<b>~240h</b>	<b>~3.5h</b>



1. [https://rocm.docs.amd.com/projects/radeon-ryzen/en/latest/docs/compatibility/compatibilityrad/windows/windows\\_compatibility.html](https://rocm.docs.amd.com/projects/radeon-ryzen/en/latest/docs/compatibility/compatibilityrad/windows/windows_compatibility.html)
2. <https://rocm.docs.amd.com/projects/radeon-ryzen/en/latest/docs/install/installrad/windows/install-pytorch.html>
3. [https://www.reddit.com/r/ROCm/comments/1p7knvx/amd\\_released\\_rocm\\_711\\_for\\_windows\\_with\\_pytorch/](https://www.reddit.com/r/ROCm/comments/1p7knvx/amd_released_rocm_711_for_windows_with_pytorch/)
4. [https://www.reddit.com/r/Amd/comments/179dncu/amd\\_rocm\\_pytorch\\_now\\_supported\\_with\\_the\\_radeon\\_rx/](https://www.reddit.com/r/Amd/comments/179dncu/amd_rocm_pytorch_now_supported_with_the_radeon_rx/)
5. [https://www.reddit.com/r/Amd/comments/180musj/amd\\_extends\\_pytorch\\_rocm\\_support\\_to\\_the\\_radeon\\_rx/](https://www.reddit.com/r/Amd/comments/180musj/amd_extends_pytorch_rocm_support_to_the_radeon_rx/)
6. [https://www.reddit.com/r/ROCm/comments/1nffbzt/install\\_rocm\\_pytorch\\_on\\_windows\\_with\\_amd\\_radeon/](https://www.reddit.com/r/ROCm/comments/1nffbzt/install_rocm_pytorch_on_windows_with_amd_radeon/)
7. [https://www.reddit.com/r/ROCm/comments/1nfn4as/amd\\_rocm\\_644\\_brings\\_pytorch\\_support\\_on\\_windows/](https://www.reddit.com/r/ROCm/comments/1nfn4as/amd_rocm_644_brings_pytorch_support_on_windows/)
8. [https://www.reddit.com/r/learnmachinelearning/comments/1je91m0/how\\_to\\_guide\\_pytorchtensorflow\\_on\\_amd\\_rocm\\_in/](https://www.reddit.com/r/learnmachinelearning/comments/1je91m0/how_to_guide_pytorchtensorflow_on_amd_rocm_in/)
9. [https://www.reddit.com/r/radeon/comments/1j31wvv/amd\\_for\\_pytorch\\_in\\_2025/](https://www.reddit.com/r/radeon/comments/1j31wvv/amd_for_pytorch_in_2025/)
10. [https://www.reddit.com/r/Amd/comments/1nqw8ne/amd\\_enables\\_windows\\_pytorch\\_support\\_for\\_radeon\\_rx/](https://www.reddit.com/r/Amd/comments/1nqw8ne/amd_enables_windows_pytorch_support_for_radeon_rx/)
11. <https://www.youtube.com/watch?v=ViQ4MKKLzfg>
12. [https://www.reddit.com/r/ROCm/comments/1enh0z3/rocm\\_pytorch\\_native\\_on\\_windows/](https://www.reddit.com/r/ROCm/comments/1enh0z3/rocm_pytorch_native_on_windows/)
13. <https://rocm.docs.amd.com/projects/install-on-linux/en/develop/install/3rd-party/pytorch-install.html>
14. <https://discuss.pytorch.org/t/about-rocm-5-6-0-on-windows/177490>
15. Notebooks\_04-07\_BenchmarkOptimized.md
16. 00\_Setup\_India\_Boundaries.ipynb
17. 01\_Preprocess\_sample\_States.ipynb
18. 03-PERF\_Performance\_Benchmark.ipynb
19. 02\_Create\_HDF5\_Dataset.ipynb
20. 03\_Clip\_Full\_India.ipynb
21. <https://www.amd.com/en/resources/support-articles/release-notes/RN-AMDGPU-WINDOWS-PYTORCH-7-1-1.html>

