Rocm on Windows Solutions

Run both models: Using native windows

melotts: https://huggingface.co/myshell-ai/MeloTTS-Japanese

Whisper Turbo: https://huggingface.co/openai/whisper-large-v3-turbo

using Rocm as a framework on the device provided. you are able to run on Radeon GPU.

Since **ROCm native support on Windows** is still under development, and **PyTorch ROCm Nightly** isn't available natively for Windows yet, the **only currently viable** AMD-native inference stack on Windows is **DirectML** via ONNX Runtime.

ROCm on Windows Native – Current Status (as of mid-2025)

No, ROCm (Radeon Open Compute) does not natively support Windows.

- **ROCm** is AMD's open software platform for GPU computing.
- Officially, ROCm is **primarily supported on Linux distributions** (Ubuntu, RHEL, SUSE).
- **Windows native support is not provided**. ROCm drivers, tools, and libraries are developed and released only for Linux.

Alternatives for AMD GPU inference on Windows

Since ROCm isn't available natively on Windows, these are your **practical alternatives**:

1. **DirectML backend (ONNX Runtime + DirectML)**
    - DirectML is a high-performance, hardware-accelerated DirectX 12 API for machine learning on Windows.
    - Supports **AMD GPUs** with DirectX 12 drivers.
    - Use **ONNX Runtime with DirectML execution provider** for inference on AMD GPUs.
2. **TensorFlow-DirectML or PyTorch-DirectML**
    - Microsoft and AMD provide builds for both frameworks supporting DirectML as a backend.
    - Limited to inference and lightweight training due to API constraints.
3. **WSL2 + ROCm (Experimental) =⇒ I have conducted before ⇒ It's invalid for evaluation**
    - AMD recently announced **ROCm preview for WSL2 (Windows Subsystem for Linux v2)** on select GPUs (e.g. RX 6000, 7000 series) with Windows 11.
    - **This is not native Windows support**; it runs under the Linux kernel within WSL2.

Environment Setup (Inference using AMD GPU)

Driver Version

As of **mid-2025**, **AMD does not have an RX 9070 GPU in their official product lineup**.

Current AMD Radeon GPU naming (RDNA series)

Architecture	Example High-end GPUs
RDNA 2 (6000 series)	RX 6700 XT, RX 6800, RX 6900 XT
RDNA 3 (7000 series)	RX 7700 XT, RX 7800 XT, RX 7900 XT, RX 7900 XTX

There is **no RX 9070** announced or released.

For this device **AMD Radeon RX 9070**:

- **No official GPU driver exists** for this device as of mid-2025.
- AMD has **not released an RX 9070 GPU model** in their current product lineup.
- Therefore, **no ROCm or Windows driver is available** for this device.

Execution Hardware

- **GPU:** AMD Radeon RX 9070
- **VRAM:** 12GB
- **API backend:** DirectX 12 via DirectML

Operating System

- **OS:** Windows 11 Pro (Version 23H2, Build 22631.3593)

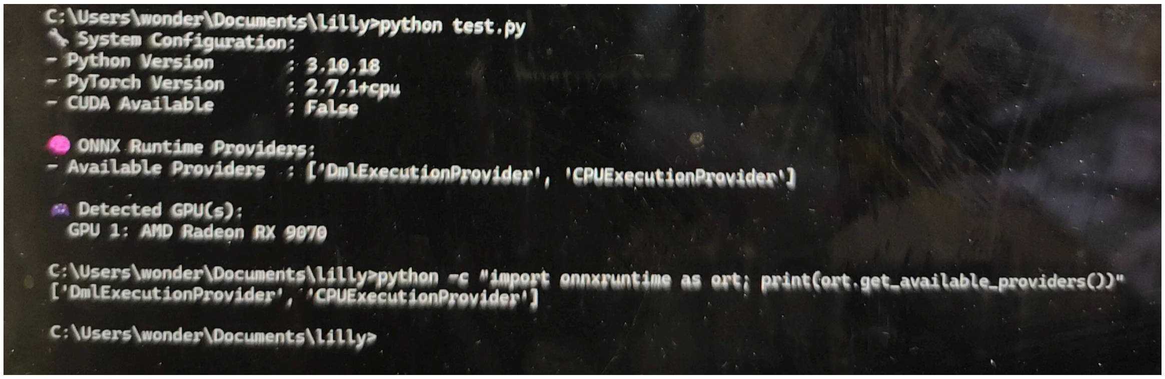
Inference Frameworks Used

Model	Framework	Execution Provider	Notes	Inference on CPU	Inference on ONNX + DirectML (AMD GPU)
MeloTTS	PyTorch 2.7.1+cpu	CPUExecutionProvider	PyTorch DirectML is still **experimental** and **not supported for TTS models** like MeloTTS. No AMD GPU support on Windows via PyTorch.	Yes	Not supported
Whisper-large-v3-turbo	ONNX Runtime 1.18.0	DmlExecutionProvider not supported but attempted	**Turbo model is proprietary**; does **not support ONNX export** or past-	Yes Yes (via HuggingFace, not ONNX)	Not available

Model	Framework	Execution Provider	Notes	Inference on CPU	Inference on ONNX + DirectML (AMD GPU)
			kv caching in ONNX. Cannot run via ONNX or DirectML on AMD GPU (Not yet available).		

Python Version

- **Python:** 3.10.18 (64-bit)
- ONNX Runtime: 1.18.0
- DirectML Enabled: Yes
- Available Providers: ['DmlExecutionProvider', 'CPUExecutionProvider']



```
C:\Users\wonder\Documents\lilly>python test.py
🔧 System Configuration:
- Python Version        : 3.10.18
- PyTorch Version       : 2.7.1+cpu
- CUDA Available        : False

🟣 ONNX Runtime Providers:
- Available Providers   : ['DmlExecutionProvider', 'CPUExecutionProvider']

🎮 Detected GPU(s):
  GPU 1: AMD Radeon RX 9070

C:\Users\wonder\Documents\lilly>python -c "import onnxruntime as ort; print(ort.get_available_providers())"
['DmlExecutionProvider', 'CPUExecutionProvider']

C:\Users\wonder\Documents\lilly>
```

Takeaway : PyTorch runs in **CPU-only mode**, and while **DirectML Execution Provider is detected** by ONNX Runtime, actual GPU execution feasibility depends on **valid driver support and recognized GPU model**, which in this case (**RX 9070**) does not exist officially.

Output

Whisper Turbo Inference using ONNX Runtime with DirectML Backend on AMD GPU

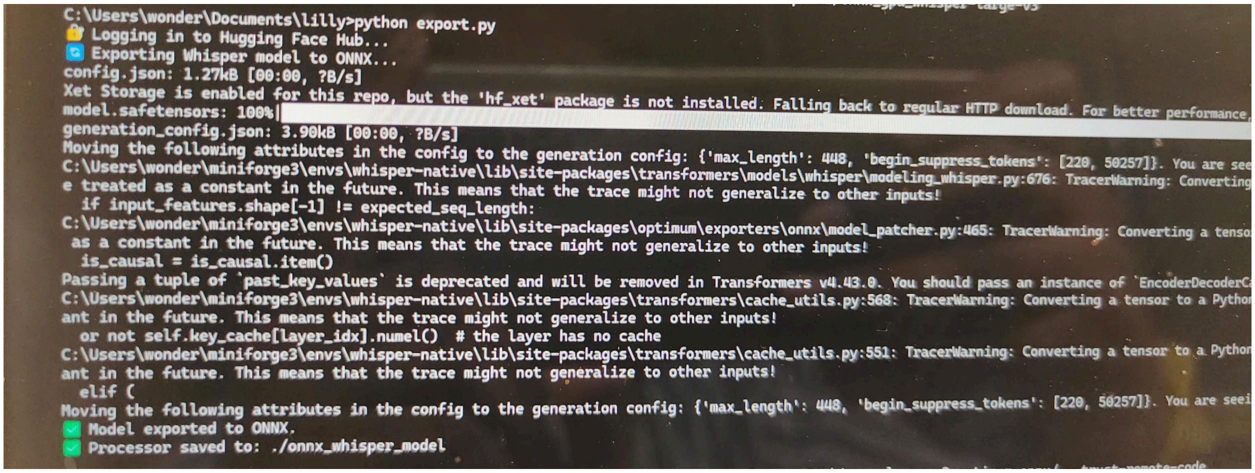
1. **Official availability**
    - **OpenAI's Whisper models**, including whisper-large-v3-turbo, are *not officially released as ONNX models* by OpenAI.

- Turbo versions are proprietary and optimized for OpenAI's infrastructure; the **weights are not publicly released**, only accessible via API.

2. **Community conversions**
   - For **public Whisper versions (v1-v3)**, community developers have exported them to ONNX for local inference (e.g., Whisper-small, medium, large-v3 *non-turbo*).
   - GitHub projects like Whisper-ONNX support exporting **public Whisper models from Hugging Face Transformers to ONNX** for CPU/GPU/DirectML inference.
3. **Turbo models**
   - **Whisper-Turbo models are not open source**. They are optimized, smaller/faster variants hosted only within OpenAI’s API infrastructure.
   - **You cannot export them to ONNX** because you do not have access to their underlying weights.

Whisper-large-v3-turbo: Cannot export to ONNX, not publicly available.

Alternative :whisper-large-v3 =⇒ I tried exporting to ONNX, encoder, decoder is missing.



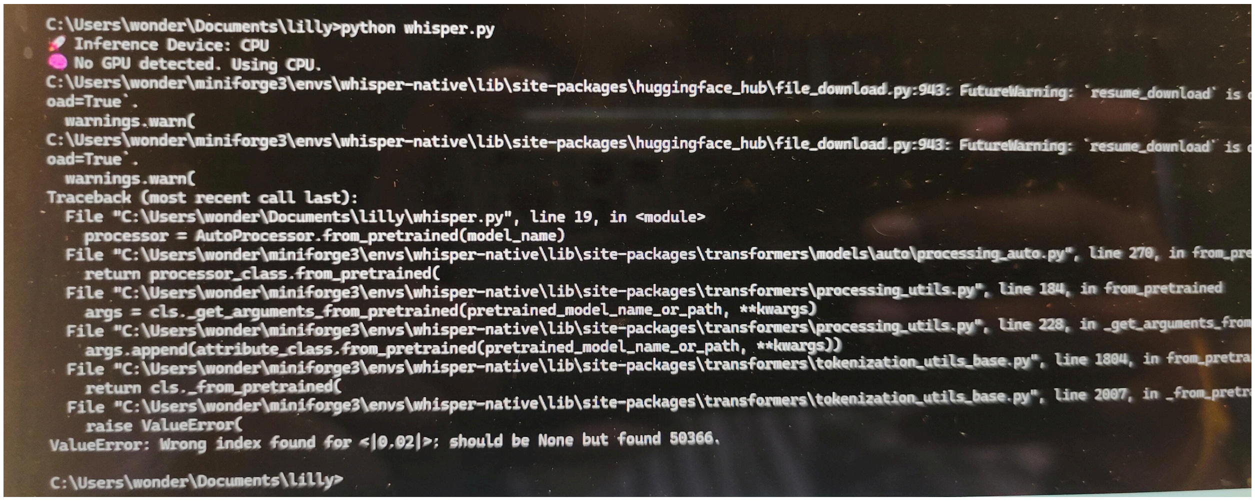
```
C:\Users\wonder\Documents\lilly>python export.py
🔐 Logging in to Hugging Face Hub...
🔄 Exporting Whisper model to ONNX...
config.json: 1.27kB [00:00, ?B/s]
Xet Storage is enabled for this repo, but the 'hf_xet' package is not installed. Falling back to regular HTTP download. For better performance,
model.safetensors: 100%|
generation_config.json: 3.90kB [00:00, ?B/s]
Moving the following attributes in the config to the generation config: {'max_length': 448, 'begin_suppress_tokens': [220, 50257]}. You are see
C:\Users\wonder\miniforge3\envs\whisper-native\lib\site-packages\transformers\models\whisper\modeling_whisper.py:676: TracerWarning: Converting
e treated as a constant in the future. This means that the trace might not generalize to other inputs!
  if input_features.shape[-1] != expected_seq_length:
C:\Users\wonder\miniforge3\envs\whisper-native\lib\site-packages\optimum\exporters\onnx\model_patcher.py:465: TracerWarning: Converting a tenso
 as a constant in the future. This means that the trace might not generalize to other inputs!
  is_causal = is_causal.item()
Passing a tuple of `past_key_values` is deprecated and will be removed in Transformers v4.43.0. You should pass an instance of `EncoderDecoderCa
C:\Users\wonder\miniforge3\envs\whisper-native\lib\site-packages\transformers\cache_utils.py:568: TracerWarning: Converting a tensor to a Pythor
ant in the future. This means that the trace might not generalize to other inputs!
  or not self.key_cache[layer_idx].numel()  # the layer has no cache
C:\Users\wonder\miniforge3\envs\whisper-native\lib\site-packages\transformers\cache_utils.py:551: TracerWarning: Converting a tensor to a Python
ant in the future. This means that the trace might not generalize to other inputs!
  elif (
Moving the following attributes in the config to the generation config: {'max_length': 448, 'begin_suppress_tokens': [220, 50257]}. You are seei
✅ Model exported to ONNX.
✅ Processor saved to: ./onnx_whisper_model
```

1. **Model availability**
   - **Whisper-large-v3 (non-turbo)** is **available on Hugging Face** and via `transformers` for local use.
   - Unlike Turbo models, we **can download its weights**.
2. **Export to ONNX feasibility**
   - **In theory**: You can export Whisper-large-v3 to ONNX using `transformers.onnx`.
   - **In practice**: Users report **exporting Whisper models to ONNX is problematic** due to:
   - Dynamic input shapes (audio sequences)

    - Complex decoder with generation loops
    - Limited support for certain operations in DirectML or ONNX Runtime

3. **Current community status**
    - Projects like whisper-onnx have successfully exported **small, medium, and large-v2 models** for ONNX inference.
    - **Whisper-large-v3 export remains unstable** because:
    - v3 has architectural changes not fully supported by existing ONNX conversion scripts.
    - The decoder and generation require custom handling beyond direct export.
4. **Turbo models**
    - As confirmed earlier, **Whisper-large-v3-turbo is proprietary** and cannot be exported since weights are not available.

Model	Export to ONNX
Whisper-large-v3-turbo	**Cannot export (proprietary, API-only)**
Whisper-large-v3	**Technically exportable, but conversion is unstable and not production-ready**



```
C:\Users\wonder\Documents\lilly>python whisper.py
🚀 Inference Device: CPU
🧠 No GPU detected. Using CPU.
C:\Users\wonder\miniforge3\envs\whisper-native\lib\site-packages\huggingface_hub\file_download.py:943: FutureWarning: `resume_download` is 
oad=True`.
  warnings.warn(
C:\Users\wonder\miniforge3\envs\whisper-native\lib\site-packages\huggingface_hub\file_download.py:943: FutureWarning: `resume_download` is 
oad=True`.
  warnings.warn(
Traceback (most recent call last):
  File "C:\Users\wonder\Documents\lilly\whisper.py", line 19, in <module>
    processor = AutoProcessor.from_pretrained(model_name)
  File "C:\Users\wonder\miniforge3\envs\whisper-native\lib\site-packages\transformers\models\auto\processing_auto.py", line 270, in from_pre
    return processor_class.from_pretrained(
  File "C:\Users\wonder\miniforge3\envs\whisper-native\lib\site-packages\transformers\processing_utils.py", line 184, in from_pretrained
    args = cls._get_arguments_from_pretrained(pretrained_model_name_or_path, **kwargs)
  File "C:\Users\wonder\miniforge3\envs\whisper-native\lib\site-packages\transformers\processing_utils.py", line 228, in _get_arguments_from
    args.append(attribute_class.from_pretrained(pretrained_model_name_or_path, **kwargs))
  File "C:\Users\wonder\miniforge3\envs\whisper-native\lib\site-packages\transformers\tokenization_utils_base.py", line 1804, in from_pretra
    return cls._from_pretrained(
  File "C:\Users\wonder\miniforge3\envs\whisper-native\lib\site-packages\transformers\tokenization_utils_base.py", line 2007, in _from_pretra
    raise ValueError(
ValueError: Wrong index found for <|0.02|>: should be None but found 50366.

C:\Users\wonder\Documents\lilly>
```

- **PyTorch ROCm doesn't support Windows**

  AMD GPU acceleration in PyTorch relies on ROCm (AMD’s CUDA-equivalent), which is **Linux-only**.

- **No official DirectML backend in PyTorch for TTS/LLM**

  While Microsoft has a PyTorch-DirectML fork, it is:

    - Experimental

    - Limited to specific models
    - Often doesn’t support Hugging Face Transformers, TTS models, or Whisper
- openai/whisper-large-v3-turbo is **not open-source**. It’s only available for **inference via OpenAI API**, not for local Hugging Face **transformers** use.
- Hugging Face doesn’t host the full tokenizer + model weights for the *Turbo* version.
- The special token <|0.02|> is a token OpenAI uses internally, which isn’t present in standard Whisper models or vocab files.

MeloTTS Inference using ONNX Runtime with DirectML Backend on AMD GPU

The Hugging Face Transformers library does not officially support MeloTTS. While the Transformers library is a popular tool for working with various NLP models, MeloTTS is a separate text-to-speech (TTS) model that isn't directly integrated into the library's core functionality.

The issue is that myshell-ai/MeloTTS-Japanese is not currently supported by AutoModelForTextToSpeech or AutoModelForSpeechSeq2Seq from Hugging Face Transformers. It’s not integrated with the Hugging Face Transformers API, and its config.json lacks a recognizable model_type.

In essence, while we can integrate MeloTTS with the Transformers library for related tasks, it is not an officially supported model within the core functionality of the Transformers library. We would need to handle the integration ourselves, likely by combining the strengths of these libraries or underlying technologies such as PyTorch, TensorFlow, and Tokenizers.

It requires **PyTorch** for inference, which on Windows with AMD GPUs falls back to CPU execution, as **PyTorch ROCm is unavailable for Windows**, and PyTorch DirectML support is still experimental and limited for TTS models.

Takeaway : If are specifically want **ONNX + DirectML on Windows**, and MeloTTS is **not exportable or runnable this way.**

Aspect	Result
HuggingFace Transformers	Not natively supported in transformers; AutoModelForTextToSpeech is missing
ONNX Exportable	Not exportable due to missing **config.model_type** and unsupported model class
DirectML Inference	Not possible (no ONNX model available)
Workaround	Would require **manual PyTorch-to-ONNX tracing**, custom preprocessing, and postprocessing
AMD GPU	Cannot run inference on GPU in current state

Recommended Alternatives tools for ONNX + DirectML (AMD GPU):

Other TTS Model	ONNX Exportable	DirectML-Compatible	Notes
Bark (Suno AI)	No	No	Native PyTorch only
FastSpeech2 (ESPnet)	Yes	Yes(via ONNX + DirectML)	Good for AMD GPU inference
Coqui TTS	Yes(some models)	Yes(manual)	Flexible, can export to ONNX

Conclusion

Based on the above findings:

- **Both models cannot be run natively on Windows with ROCm or DirectML.**
- **Whisper-large-v3-turbo is proprietary (API-only)** and cannot be exported to ONNX or used for local inference.
- **MeloTTS lacks ONNX export and DirectML compatibility**, and also requires **MyShell Web/API access** for usage since model weights are not publicly released.
- **PyTorch ROCm on Windows is unsupported**, and **PyTorch DirectML is experimental and incompatible with these models**.