

# | WinWSL2-GPU环境安装与配置实验

## | 1. 实验目的

本实验旨在让学生掌握在Windows系统上安装和配置WSL2环境以支持GPU加速的关键技能。通过本实验，学生将能够：

1. 理解WSL1和WSL2的区别，并掌握切换和配置方法。
2. 成功安装WinWSL2-GPU增量环境，支持CUDA、PyTorch等GPU加速库。
3. 验证GPU加速功能，并测试大矩阵运算的性能提升。
4. 掌握外挂环境和覆盖安装两种安装方式的使用场景和操作步骤。
5. 熟悉WSL2环境下的开发工具链，包括Docker、Anaconda等。

## | 2. 实验要求

- 软件要求：Windows 10及以上版本；RflySim工具链<sup>[1]</sup>。
- 硬件要求：Windows 11 22H2以上 PC，支持虚拟化（需启用Virtual Machine Platform），具备NVIDIA GPU（支持CUDA）<sup>[2]</sup>。

## | 3. 实验地址

例程目录：[\[安装目录\]\RflySimAPIs\1.RflySimIntro\2.AdvExps\e13.WinWSL2-GPU](#)

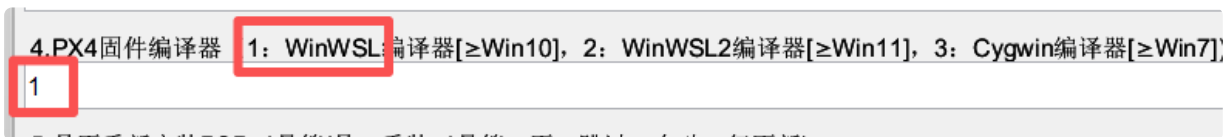
- [BackupWinWSL2.bat](#)：备份WSL2环境脚本。
- [Clear-ubuntu.sh](#)：清理Ubuntu环境的脚本。
- [EnableWSL2.bat](#)：启用WSL2功能的批处理脚本。
- [gpu\\_test.py](#)：GPU加速测试Python脚本。
- [Install.bat](#)：外挂方式安装脚本。
- [RosSwitch2.bat](#)：ROS环境切换脚本。
- [Uninstall2.bat](#)：卸载脚本。
- [WinWSL2.bat](#)：启动WinWSL2-GPU环境的批处理脚本。

- [WslGUI2.bat](#)：启动图形界面的批处理脚本。

## 4. 实验内容或步骤

### 4.1 步骤1：启用WSL2功能并安装RflySim基础环境

1. 首次使用WSL2：运行 [EnableWSL2.bat](#)，若提示重启输入 Y 并重启，目的是开启 VirtualMachinePlatform 以启用 WSL2。GPU 加速与 AI 训练必须在 WSL2 下进行，WSL1 功能受限。（具体可参见 [\[RflySim安装目录\]/HowToInstall.pdf](#) 中的**启用WSL子系统功能**）
2. PX4 SITL 建议放在 WSL1：其编译后相当于 Windows 可执行程序，占用资源极小，与 CopterSim 联机更稳定。WSL2 与视觉算法共享同一 Ubuntu 资源，易出现性能争用。建议让默认 RflySim-20.04 保持 WSL1，WinWSL2-GPU 为 WSL2，用于 Docker/GPU/AI。若需更新 RflySim 工具链，可参见 [\[RflySim安装目录\]/HowToInstall.pdf](#) 中的**一键全自动安装**进行。重新运行一键脚本时，将第 "4" 项设为 1 后确认，即可把 PX4 SITL 编译与运行切到 WSL1，保证仿真稳定。



注意事项：

- 使用安装脚本的切换方法，会删除掉原WinWSL环境，并直接以WSL1方式导入新的WinWSL，可能丢失数据，请提前备份好其中的源码或重要信息。也可使用【桌面】\RflyTools\WslSwit2 并输入1来切换，这种方式不会丢失数据，但是切换时间可能达到半个小时到1个小时，请耐心等待。
- 如果使用"4.3 步骤3：直接覆盖方式安装WinWSL2-GPU"则不需要执行上述WSL2切换步骤，因为它会直接以WSL2方式导入新的WinWSL2-GPU.tar环境，并替换为WinWSL/RflySim-20.04。此时，PX4 SITL和视觉算法将运行在同一个WSL环境中。
- 上述覆盖安装方法，在更新RflySim工具链新版时，可能会被还原为安装包内自带的精简WSL2环境。因此，使用"4.2 步骤2：外挂方式安装WinWSL2-GPU（推荐）"的另一个好处是，更新RflySim工具链不会影响到WSL2-GPU开发环境，可以持续在上面开发。若需要升级新的环境，可以先按下文方法，备份当前环境，再下载新的环境包并安装即可。

## 4.2 步骤2：外挂方式安装WinWSL2-GPU（推荐）

首先下载WSL2-GPU镜像：

链接: <https://pan.baidu.com/s/11L6nCJWCjHg7Db73h366DQ?pwd=nvaq> 提取码: nvaq

### RflySim 推荐外挂方式加载 GPU 加速环境：

1. 运行"[Install.bat](#)"就能开始自动安装，中间会提示输入安装目录，若WinWSL2-GPU发行版已存在，还会弹出确认选项，经过确认后才会覆盖，请根据需求放心运行本脚本。
2. 首先是安装目录的选择，由于本环境比较大，需要50G的空间，本增量环境不必和RflySim安装目录相同，用户可自行根据硬盘剩余情况进行选择。若直接回车，将选择默认路径：D:\RflyGPU。例如，下图输入的是D:\RflyGPU，并回车来配置自定义安装目录。

```
管理员: C:\Windows\system32\cmd.exe
Checking if Virtual Machine Platform is enabled...
Virtual Machine Platform is enabled. Proceeding...
Prompt user for installation directory
About 30G of free space is required.
Press Enter will use default D:\RflyGPU
You can also choose another folder.
WSL installation folder (e.g., D:\RflyGPU):
```

其次，如果之前安装过外挂环境，会弹出是否卸载原WinWSL2-GPU镜像，并开始装新的，直接输入Y即可（不需要回车）。

```
Checking for existing RflySim 20.04 distro...
Existing RflySim-20.04 distro found. It will be
Confirm uninstall? (Y/N) [Y,N]?
```

3. 测试GPU加速环境。双击"[WinWSL2.bat](#)"，并输入python3 [gpu\\_test.py](#) 可进行测试。输出结果如下：说明环境部署成功，启用了GPU加速。从下图可以看到，在进行大矩阵运算时，GPU加速能提升接近50倍的速度提升，对于机载系统算法部署而言非常重要。

```

root@DaiPC:/mnt/c/Users/dream/Desktop/Test# python3 gpu_test.py
系统信息：
操作系统：Linux
Python版本：3.10.12 (main, Aug 15 2025, 14:32:43) [GCC 11.4.0]
PyTorch版本：2.6.0+cu124
CUDA是否可用：True
CUDA版本：12.4
GPU设备数量：1
GPU 0: NVIDIA GeForce RTX 4090
GPU 0 显存总量：23.99 GB
GPU 0 当前可用显存：0.00 MB

开始矩阵乘法测试...

使用矩阵大小：20000x20000
创建CPU矩阵...
执行CPU计算...
CPU计算耗时：15.2606 秒

转移数据到GPU...
预热GPU...
执行GPU计算...
GPU计算耗时：0.3434 秒
GPU加速比：44.44x
root@DaiPC:/mnt/c/Users/dream/Desktop/Test# |

```

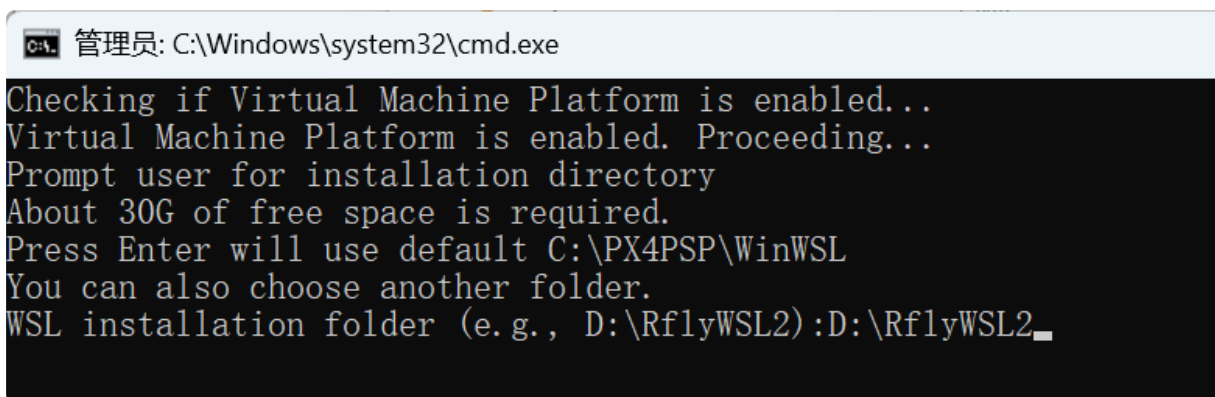
4. 若要在平台中，使用增量环境运行Python、Shell等脚本，可以直接将本文件夹的"[WinWSL2.bat](#)"拷贝到代码目录，双击就能进入WinWSL2-GPU发行版环境，再运行程序即可。
5. 若要修改之前的一些一键bat脚本，使其以WinWSL2-GPU发行版运行，而非默认的RflySim-20.04，可参考 `8.RflySimVision\1.BasicExps\2-BaseDemoAuto\Ubuntu` 例程目录下WinWSLRunALL.bat和WinWSLRunALL2.bat的修改方法，核心思想是将bat脚本复制重命名，然后右键编辑新的bat脚本，并搜索"RflySim-20.04"字符，将其替换为"WinWSL2-GPU"即可。注：先运行 `2-BaseDemoAuto\Windows\SITLPosStrOnekey.bat` ，再运行WinWSLRunCV2.bat可测试外挂环境整体联机效果，飞机能正常起飞，且起飞后可以看到前视和下视摄像头数据，说明环境配置正确。若无法实现上述效果，请确认Windows版本，自行调整程序。
6. 若需要打开图形界面，请使用 [WslGUI2.bat](#) 脚本
7. 需要更新环境，下载新的环境增量包，直接再次运行Install2.bat即可。
8. 若需要切换ROS环境，请使用 [RosSwitch2.bat](#) 脚本
9. 如果需要卸载本外挂环境，请使用 [Uninstall2.bat](#) 脚本。
10. 备份镜像与还原，如果用户在镜像基础上，又定制了自身的环境，可以双击 [BackupWinWSL2.bat](#) ，将当前WSL环境备份到指定目录。后续，只需要将备份的文件

WinWSL2-GPU.tar拷贝回当前目录，替换掉原文件，再次双击 [Install.bat](#) 就能切换到自己备份的环境。

## 4.3 步骤3：直接覆盖方式安装WinWSL2-GPU

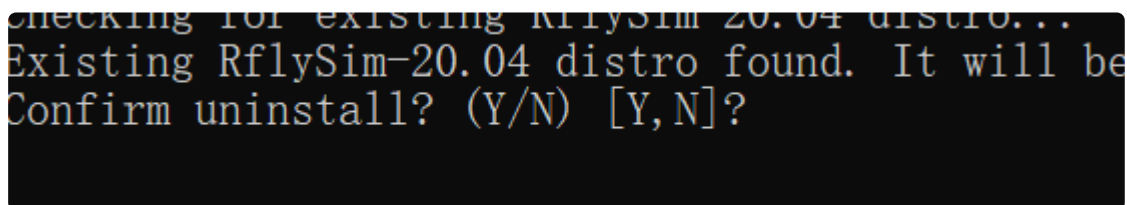
覆盖安装的源文件在 [Override](#) 目录，先进入该目录：

1. 运行 `InstallOverride.bat` 开始自动安装，会提示输入安装目录并询问是否覆盖 WinWSL/RflySim-20.04，确认后继续。
2. 两个确认步骤：先选安装目录（约需 50G，回车则用默认【安装目录】\WinWSL，如示例输入 D:\RflyWSL2）。



```
C:\> 管理员: C:\Windows\system32\cmd.exe
Checking if Virtual Machine Platform is enabled...
Virtual Machine Platform is enabled. Proceeding...
Prompt user for installation directory
About 30G of free space is required.
Press Enter will use default C:\PX4PSP\WinWSL
You can also choose another folder.
WSL installation folder (e.g., D:\RflyWSL2):D:\RflyWSL2_
```

随后提示是否卸载原 RflySim-20.04 镜像并安装新的，输入 Y（无需回车）。



```
Checking for existing RflySim 20.04 distro...
Existing RflySim-20.04 distro found. It will be
Confirm uninstall? (Y/N) [Y,N]?
```

3. 测试 GPU：双击 `WinWSL.bat`，执行 `python3 gpu_test.py`，输出正常即表示 GPU 已启用。

```

root@DaiPC:/mnt/c/Users/dream/Desktop/Test# python3 gpu_test.py
系统信息：
操作系统：Linux
Python版本：3.10.12 (main, Aug 15 2025, 14:32:43) [GCC 11.4.0]
PyTorch版本：2.6.0+cu124
CUDA是否可用：True
CUDA版本：12.4
GPU设备数量：1
GPU 0: NVIDIA GeForce RTX 4090
GPU 0 显存总量：23.99 GB
GPU 0 当前可用显存：0.00 MB

开始矩阵乘法测试...

使用矩阵大小：20000x20000
创建CPU矩阵...
执行CPU计算...
CPU计算耗时：15.2606 秒

转移数据到GPU...
预热GPU...
执行GPU计算...
GPU计算耗时：0.3434 秒
GPU加速比：44.44x
root@DaiPC:/mnt/c/Users/dream/Desktop/Test# |

```

4. 更新：下载新的增量包后，直接再次运行 `InstallOverride.bat`。
5. 卸载：可使用 RflySim 自带的一键卸载脚本。
6. 备份/还原：如已定制环境，可双击 `BackupWinWSL.bat` 备份到指定目录。恢复时将 `WinWSL2-GPU.tar` 复制回当前目录覆盖原文件，再双击 `Install.bat` 即可切换到备份环境。

## 5. 关键知识点

### 关键知识点1：增量WinWSL2-GPU环境包含功能

本增量WinWSL2-GPU发行版环境相对于默认的RflySim-20.04发行版，主要增加了如下开发环境，重点在于GPU加速和AI训练的支持，同时也更贴近机载板卡的开发环境，后续会针对性开发更多例程：

1. Cuda开发所需环境，暂定使用cuda-toolkit-12-4
2. Pytorch-GPU版本，暂定使用 `torch==2.6.0 torchvision==0.21.0 torchaudio==2.6.0`
3. cuDNN环境，默认9.10版本

4. OpenCV-GPU环境，OpenCV-4.10 (4.10.0.84)，支持arch8.6，8.9版本，支持Cuda加速
5. numpy==1.26.4，注意ROS1的原因，暂时无法升级numpy 2.0。
6. ollama环境，自带qwen3:0.6b（还可以加载别的本地大模型，这里提供了启元大模型增量包：[https://pan.baidu.com/s/15DXwSOOLMnKKhsaKh\\_1M5A?pwd=2dzq](https://pan.baidu.com/s/15DXwSOOLMnKKhsaKh_1M5A?pwd=2dzq)）
7. Anaconda3，暂定2024.10-1版本
8. tensorflow，暂定版本为2.20.0+tf-keras==2.20.1
9. docker环境，已打通GPU驱动
10. Stable-Baselines3==2.7.0 + Gymnasium==1.2.2 用于强化学习训练
11. Genesis==0.3.6 强化学习框架+ rsl-rl-lib==2.2.4
12. Gazebo (gz-garden版本) 支持PX4 1.14以上版本固件使用make px4\_sitl gz\_\*\*来运行SITL仿真。注：Gazebo11通过PPA方式并行保留，命令行使用gz11来引用。

## 关键知识点2：安装版本说明

目前，支持两种安装方式：

1. **方案一：外挂环境。**和原有WinWSL环境并存，将本环境外挂导入为WinWSL2-GPU的编译环境，这种方式支持Windows10 和Windows11的所有版本。注意：Windows11 22H2之后的系统支持镜像网络，可以和WSL1一样和Windows共享IP地址，便于互联互通。在安装外挂环境时，会默认配置镜像网络（见 `C:\Users\<用户名>\.wslconfig` 文件内的 `networkingMode=mirrored` 参数），因此**请用户升级系统到Windows11 22H2及之后版本再使用本功能**。若使用Windows11 22H2及之前的版本，**请手动删除 `C:\Users\<用户名>\.wslconfig` 文件**，使用NAT网络模式（可以看做一台独立电脑，和主机有着不同IP，无法直接通过内网互联）来运行程序。需要注意的是，若部分例程无法正常运行，可能需要指定IP地址，请自行按照例程 `8.RflySimVision\1.BasicExps\3-BaseDemoManu` 的方法，或参考例程 `8.RflySimVision\1.BasicExps\2-BaseDemoAuto` 使用自动请求IP的方式。
2. **方案二：直接覆盖。**直接覆盖原来的WinWSL编译器（镜像名：RflySim-20.04）并切换为WSL2，这样就能增强原有的GPU/AI编译环境。但是，GPU加速需要WSL2的支持，且需要支持镜像网络才能进行SITL仿真，**此功能在Windows11 22H2之后才支持**。因此，若Windows11 22H2之前的版本直接覆盖原有WinWSL编译器，会导致SITLRun后飞机无法起飞，也无法完成实验。

**总结起来：推荐用户升级到Windows11 22H2，然后使用方案一外挂环境方式安装增量环境包，若系统版本低则需要手动修改IP或使用自动获取方式。在需要运行GPU加速的算法时，可将 `WinWSL2.bat` 文件拷贝到例程目录，再双击加载外挂环境WinWSL2-GPU，并执**

行Ubuntu命令。注意：核心语句如下，通过 `wsl -d WinWSL2-GPU` 在启动WSL时，自动进入WinWSL2-GPU发行版，而非默认的RflySim-20.04。更多调用方法如下：

在bat脚本中，调用WSL执行命令的方法，命令使用分号";"间隔

```
wsl -d WinWSL2-GPU -e bash -lic "你的命令"
```

在bat脚本中，新开一个窗口来运行WSL命令，不会产生阻塞，可以并行运行多条命令。

```
start wsl -d WinWSL2-GPU -e bash -lic "你的命令"
```

## 关键知识点3：WSL1和WSL2的区别总结

WSL (Windows Subsystem for Linux) 是 Windows 上运行 Linux 环境的工具。WSL1 和 WSL2 的核心差异在于架构、性能和兼容性。下面是简要对比表格：

方面	WSL1	WSL2
架构	使用系统调用翻译层（兼容层），将 Linux 调用转换为 Windows 调用，无真实 Linux 内核。	使用轻量级虚拟机（基于 Hyper-V），运行真实的 Linux 内核。
性能	文件系统 I/O 较慢，尤其在跨 Windows/Linux 文件访问时（例如 <code>npm install</code> 可能慢 10-50 倍）。	文件系统性能显著提升，更接近原生 Linux；网络和计算任务更快（如 SSH 传输速度可达 WSL1 的 3 倍）。
兼容性	系统调用兼容性有限，不支持某些 Linux 功能（如 Docker 原生支持、GPU 计算）。	完整系统调用支持，支持 Docker、USB 设备、GPU 等更多 Linux 特性。
资源使用	资源消耗较低，与 Windows 更紧密集成。	资源消耗较高（如内存、CPU），因为是虚拟机，但可动态调整。
网络与集成	网络与 Windows 共享 IP，文件系统集成更好，但可能有权限问题。	独立网络栈（虚拟 IP），文件访问需通过 <code>/mnt</code> 挂载，可能需额外配置。

方面	WSL1	WSL2
适用场景	适合简单命令行任务、开发测试，对性能要求不高。	适合复杂应用、容器化（如 Docker）、高性能需求。

总体而言，WSL2 是推荐的版本（自 2019 年推出），除非特定场景下文件系统集成更重要。如果需要切换，可用 `wsl --set-version` 命令。

## 关键知识点4：测试代码解析：`gpu_test.py`

该脚本位于根目录，用于测试GPU加速功能。整体框架如下：

```
1 import torch
2 import time
3
4 # 检查CUDA可用性
5 print(torch.cuda.is_available())
6
7 # 创建大矩阵进行运算测试
8 device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
9 a = torch.randn(10000, 10000).to(device)
10 b = torch.randn(10000, 10000).to(device)
11
12 # 计时运算
13 start = time.time()
14 c = torch.mm(a, b)
15 end = time.time()
16
17 print(f"运算时间: {end - start:.2f}秒")
```

运行结果显示GPU加速可提升约50倍性能，输出类似：

```
1 True
2 运算时间: 0.12秒 # GPU模式
3 # CPU模式可能需要6秒以上
```

## 6. 参考资料

1. [RflySim官方文档](#)
2. [WSL官方文档](#)
3. [CUDA安装指南](#)

4. PyTorch GPU安装

5. RflySim工具链用户手册

## 7. 常见问题

### Q1: 安装后无法启动WSL2环境

A1: 确认Windows版本为11 22H2+, 或删除 `.wslconfig` 文件使用NAT模式

### Q2: GPU测试显示CUDA不可用

A2: 检查NVIDIA驱动版本, 确保支持CUDA 12.4

### Q3: SITL仿真不稳定

A3: 使用WSL1运行PX4 SITL, WSL2用于视觉算法

---

1. <https://rflysim.com/> ↩

2. 推荐配置请见: <https://rflysim.com/doc/zh/HowToInstall.pdf> ↩