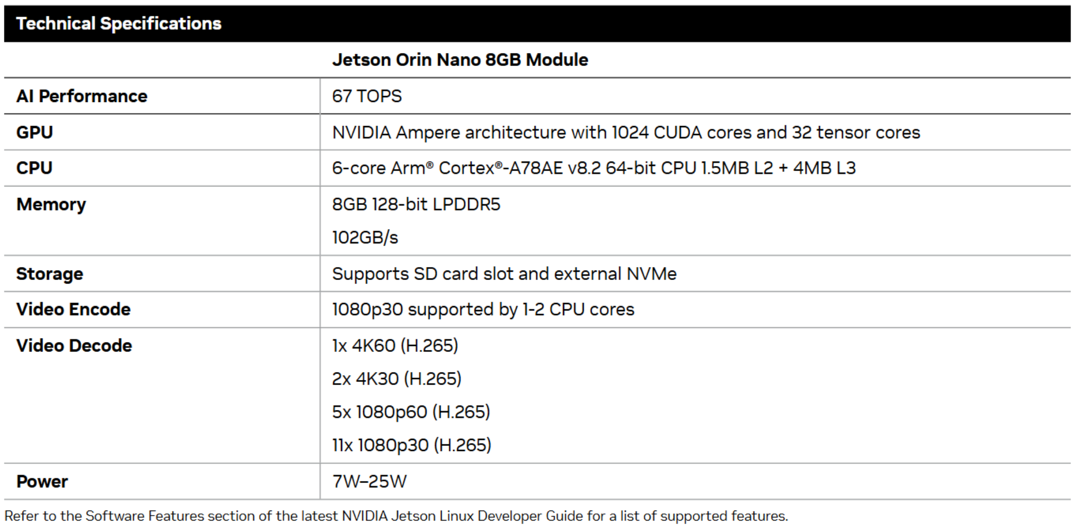
Jetson Orin Nano 与 JetPack 认知及环境搭建纪要

一、Jetson Orin Nano 的调研信息

从下图中可以看到，Jetson Orin Nano 的 GPU 采用了 2020 年发布的 Ampere 架构，包含 1024 个 CUDA 核心和 32 个 Tensor 核心。这个 GPU 可以支持 DLSS、RTX 光追，适合实时 AI 推理，如 YOLO、Transformer 模型。

CPU 为 6 核 Arm Cortex-A78AE（v8.2 64 位），AE 后缀表示面向汽车/工业场景，支持功能安全（锁步模式）。

内存为 8GB LPDDR5（128 位总线，102GB/s）。其中，带宽是 Jetson Xavier Nano（51.2GB/s）的 2 倍，可缓解 GPU 的瓶颈。但是 8GB 对大型模型（如 10B+参数的 LLM）可能不足，需量化或裁剪模型。



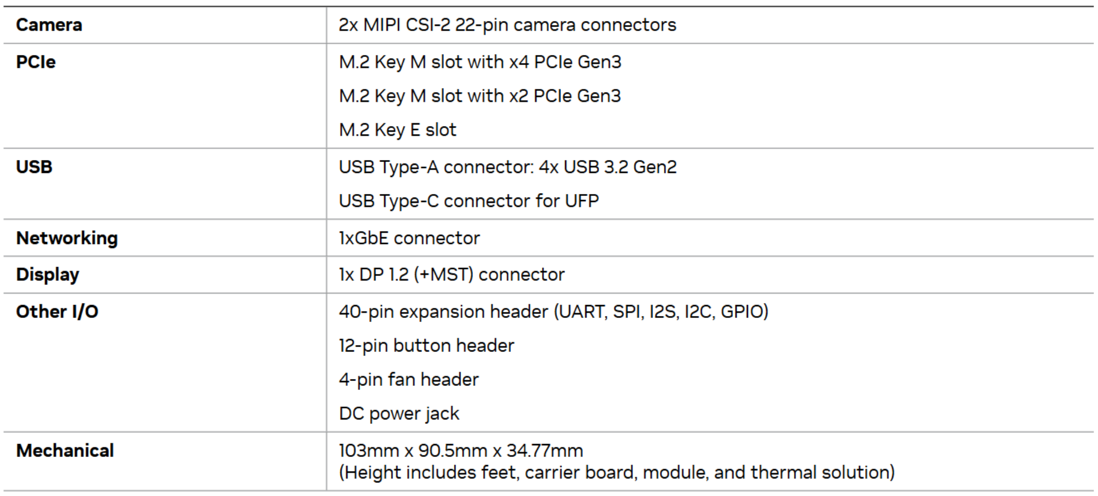
Technical Specifications

	Jetson Orin Nano 8GB Module
AI Performance	67 TOPS
GPU	NVIDIA Ampere architecture with 1024 CUDA cores and 32 tensor cores
CPU	6-core Arm® Cortex®-A78AE v8.2 64-bit CPU 1.5MB L2 + 4MB L3
Memory	8GB 128-bit LPDDR5 102GB/s
Storage	Supports SD card slot and external NVMe
Video Encode	1080p30 supported by 1-2 CPU cores
Video Decode	1x 4K60 (H.265) 2x 4K30 (H.265) 5x 1080p60 (H.265) 11x 1080p30 (H.265)
Power	7W–25W

Refer to the Software Features section of the latest NVIDIA Jetson Linux Developer Guide for a list of supported features.

图 1.1：Jetson Orin Nano 的硬件规格

(from https://nvdam.widen.net/s/zkfqjmtds2/jetson-orin-datasheet-nano-developer-kit-3575392-r2)



Camera	2x MIPI CSI-2 22-pin camera connectors
PCIe	M.2 Key M slot with x4 PCIe Gen3 M.2 Key M slot with x2 PCIe Gen3 M.2 Key E slot
USB	USB Type-A connector: 4x USB 3.2 Gen2 USB Type-C connector for UFP
Networking	1xGbE connector
Display	1x DP 1.2 (+MST) connector
Other I/O	40-pin expansion header (UART, SPI, I2S, I2C, GPIO) 12-pin button header 4-pin fan header DC power jack
Mechanical	103mm x 90.5mm x 34.77mm (Height includes feet, carrier board, module, and thermal solution)

图 1.2：Jetson Orin Nano 的关键接口

(from https://nvdam.widen.net/s/zkfqjmtds2/jetson-orin-datasheet-nano-developer-kit-3575392-r2)

根据上图，Jetson Orin Nano 包含 2 个 MIPI CSI-2（22-pin）摄像头接口，可以连接双目摄像头和深度相机，但需兼容 MIPI 协议的摄像头模块。
有 3 个 PCIe 扩展，其中，M.2 Key M（x4 PCIe Gen3 可以安装 NVMe SSD，扩展高速存储；M.2 Key M（x2 PCIe Gen3）可以接驳低速 SSD 或 AI 加速卡；M.2 Key E 可以扩展 Wi-Fi 或蓝牙模块。
除此之外，还有 4 个 USB Type-A 接口和 1 个 USB Type-C 接口；40-pin 扩展头，12-pin 按钮头，4-pin 风扇头，DC 电源口等。
Jetson Orin Nano 搭配上完善的软件生态 JetPack SDK， Ubuntu + CUDA/cuDNN/TensorRT，开箱即用。在入门级边缘 AI 中提供了 性能、功耗与易用性的最佳平衡，尤其适合需要低延迟 AI 推理的视觉项目。

二、JetPack SDK 的调研信息

NVIDIA JetPack 包括 3 个组件：Jetson Linux, Jetson AI Stack, and Jetson Platform Services。

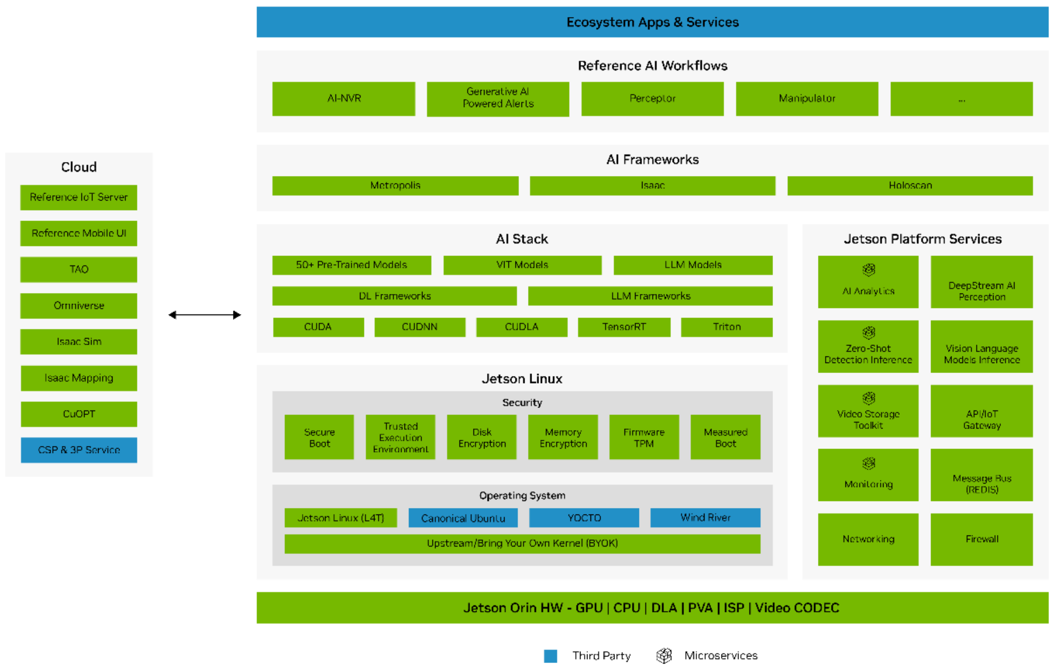
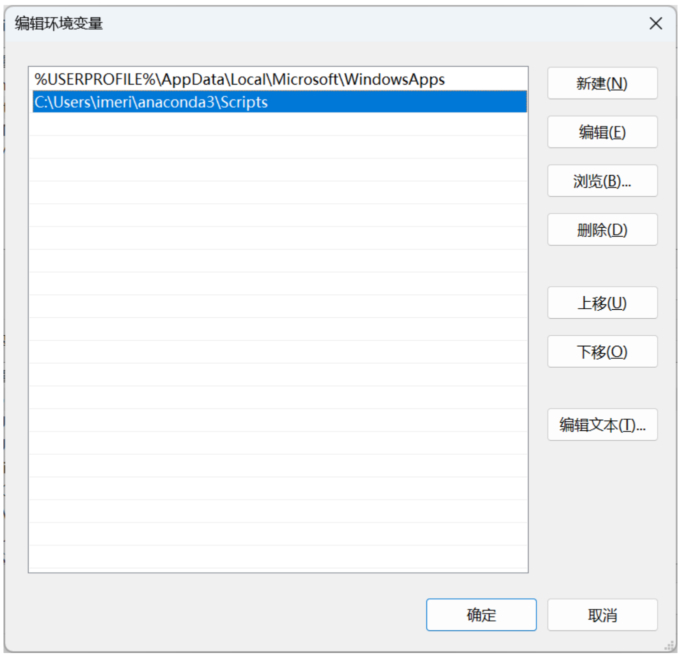


图 2.1：JetPack 架构
(from https://developer.nvidia.com/embedded/develop/software)

在 Jetson Linux 中，L4T 是 Jetson Linux 的核心底层，提供硬件抽象层（HAL）和内核优化。L4T 是基于定制化 Ubuntu 操作系统 + Tegra 硬件驱动，专为 Tegra SoC 优化。
CUDA Toolkit，cuDNN，TensorRT，DeepStream 都是 Jetson AI Stack 的核心组成。在 AI 开发中，L4T 作为定制化操作系统，为 Jetson 设备提供硬件驱动和基础运行环境；CUDA Toolkit 是 GPU 加速的基础，通过并行计算支持 AI 任务； cuDNN 针对深度学习算子进行优化，加速模型训练和推理； TensorRT 作为高性能推理引擎，通过层融合和量化技术显著提升部署效率；而 DeepStream SDK 则专注于视频分析，集成硬件加速和多路流处理能力。L4T 提供硬件支持，CUDA 和 cuDNN 构建计算基础，TensorRT 优化模型部署，DeepStream 实现视频 AI 应用，形成从数据输入到推理输出的完整闭环，使 Jetson 成为边缘 AI 和机器人开发的理想平台。

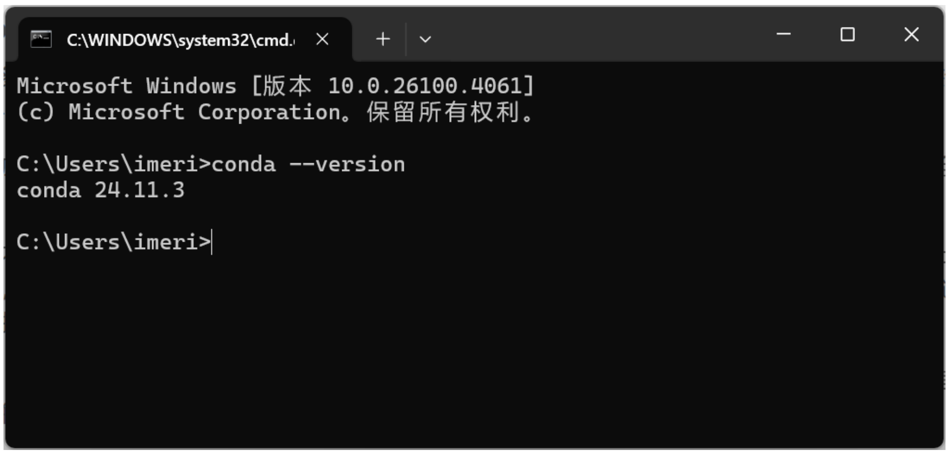
三、PC 端环境开发（Windows）

配置 conda 环境：安装 Anaconda（包含 Python 3.12）



```
%USERPROFILE%\AppData\Local\Microsoft\WindowsApps
C:\Users\imeri\anaconda3\Scripts
```

图 3.1：在环境变量中添加 Anaconda 的路径

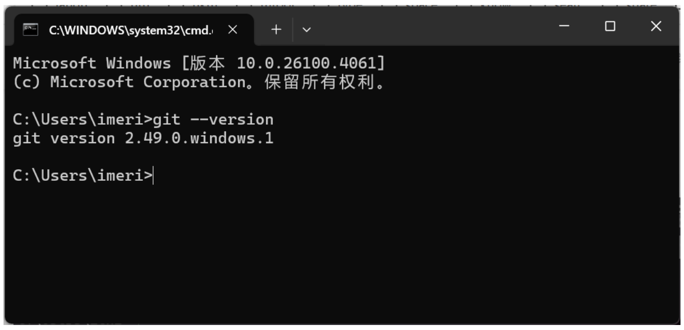


```
Microsoft Windows [版本 10.0.26100.4061]
(c) Microsoft Corporation。保留所有权利。

C:\Users\imeri>conda --version
conda 24.11.3

C:\Users\imeri>
```

图 3.2：Conda 已安装成功

表 3.1：常用包和环境管理命令

操作	命令
安装包	Conda install package_name
卸载包	Conda remove package_name
更新包	Conda update package_name
列出所有包	Conda list
创建环境	Conda create -n env_name package_name
激活环境	Activate env_name
离开环境	Deactivate
列出环境	Conda env list
删除环境	Conda env remove -n env_name

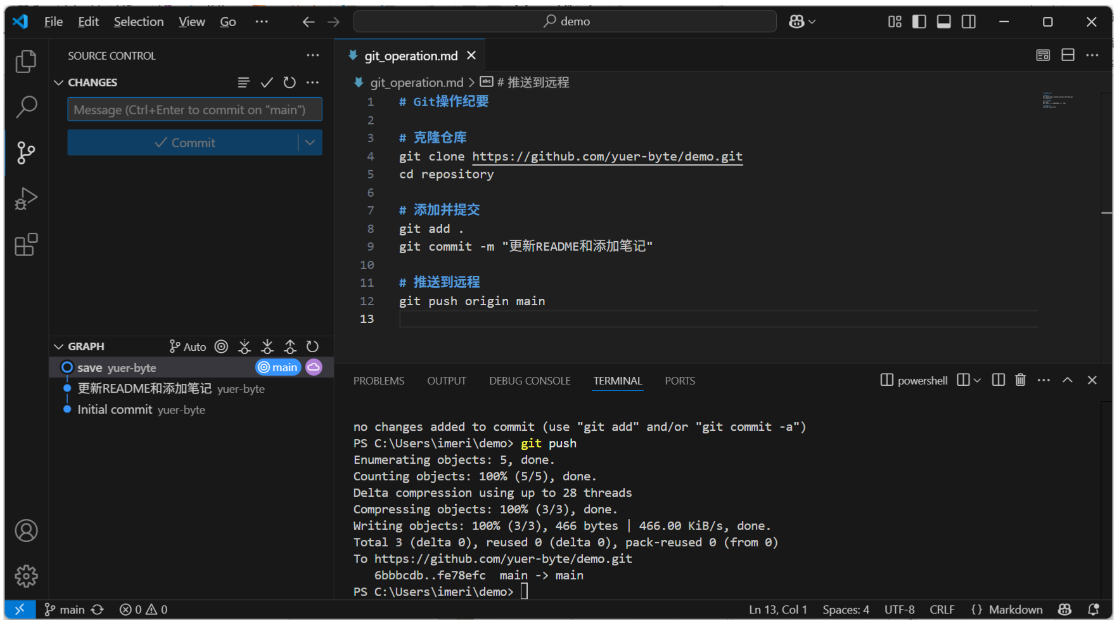


```
Microsoft Windows [版本 10.0.26100.4061]
(c) Microsoft Corporation。保留所有权利。

C:\Users\imeri>git --version
git version 2.49.0.windows.1

C:\Users\imeri>
```

图 3.3：Git 已安装成功

GitHub 仓库链接：https://github.com/yuer-byte/demo.git
设置用户名和邮箱地址：

```
$ git config --global user.name “yuer-byte”
$ git config --global user.email “935778457@qq.com”
```

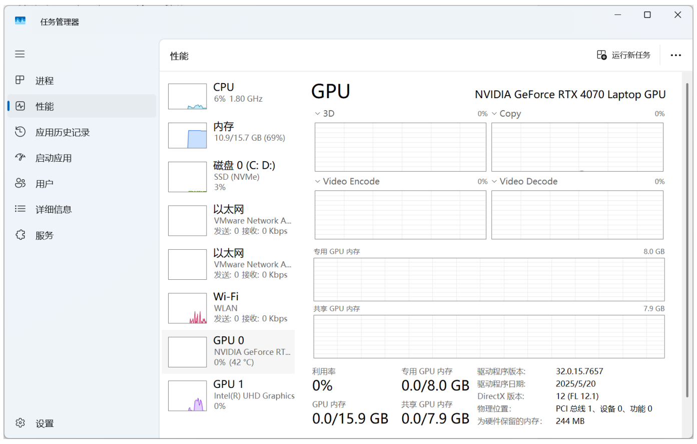


File Edit Selection View Go
demo
SOURCE CONTROL
CHANGES
Message (Ctrl+Enter to commit on "main")
✓ Commit
GRAPH
Auto
save yuer-byte
main
更新README和添加笔记 yuer-byte
Initial commit yuer-byte
git_operation.md
git_operation.md > # 推送到远程
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell
main ⊗ 0 ⚠ 0
Ln 13, Col 1 Spaces: 4 UTF-8 CRLF {} Markdown

```
# Git操作纪要

# 克隆仓库
git clone https://github.com/yuer-byte/demo.git
cd repository

# 添加并提交
git add .
git commit -m "更新README和添加笔记"

# 推送到远程
git push origin main
```

```
no changes added to commit (use "git add" and/or "git commit -a")
PS C:\Users\imeri\demo> git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 28 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 466 bytes | 466.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/yuer-byte/demo.git
   6bbbcdb..fe78efc  main -> main
PS C:\Users\imeri\demo> 
```

图 3.4：使用 VS Code 进行 Git clone, add, commit, push 练习



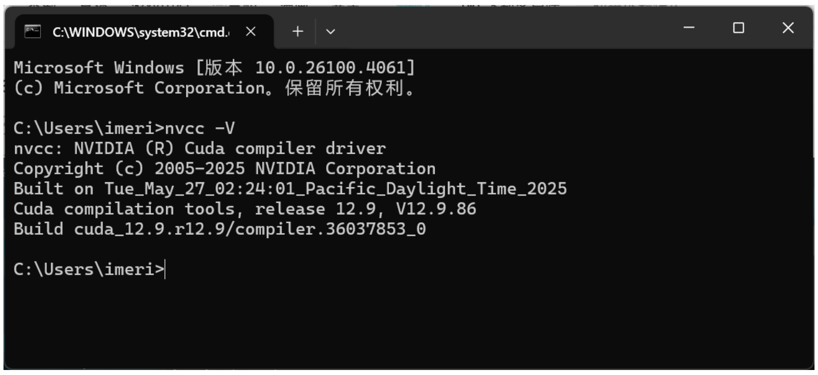
性能

CPU 6% 1.80 GHz
内存 10.9/15.7 GB (69%)
磁盘 0 (C: D:) SSD (NVMe) 3%
以太网 VMware Network A... 发送: 0 接收: 0 Kbps
以太网 VMware Network A... 发送: 0 接收: 0 Kbps
Wi-Fi WLAN 发送: 0 接收: 0 Kbps
GPU 0 NVIDIA GeForce RT... 0% (42 °C)
GPU 1 Intel(R) UHD Graphics 0%

GPU NVIDIA GeForce RTX 4070 Laptop GPU
3D 0% Copy 0%
Video Encode 0% Video Decode 0%
专用 GPU 内存 8.0 GB
共享 GPU 内存 7.9 GB

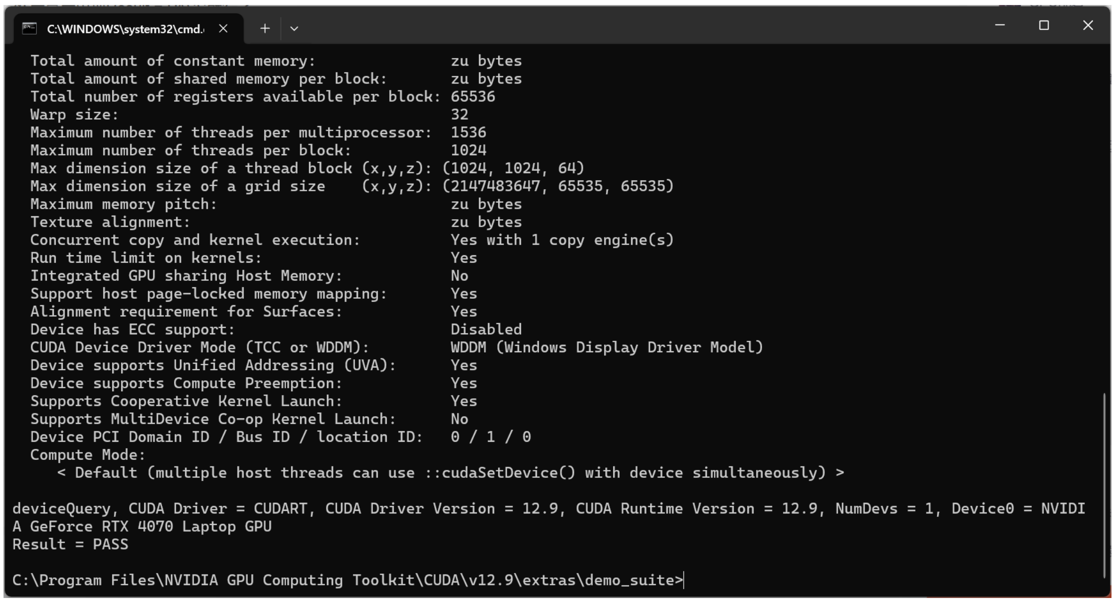
利用率	专用 GPU 内存	驱动程序版本:	32.0.15.7657
0%	0.0/8.0 GB	驱动程序日期:	2025/5/20
		DirectX 版本:	12 (FL 12.1)
GPU 内存	共享 GPU 内存	物理位置:	PCI 总线 1、设备 0、功能 0
0.0/15.9 GB	0.0/7.9 GB	为硬件保留的内存:	244 MB

图 3.5：显卡为 NVIDIA GeForce RTX 4070



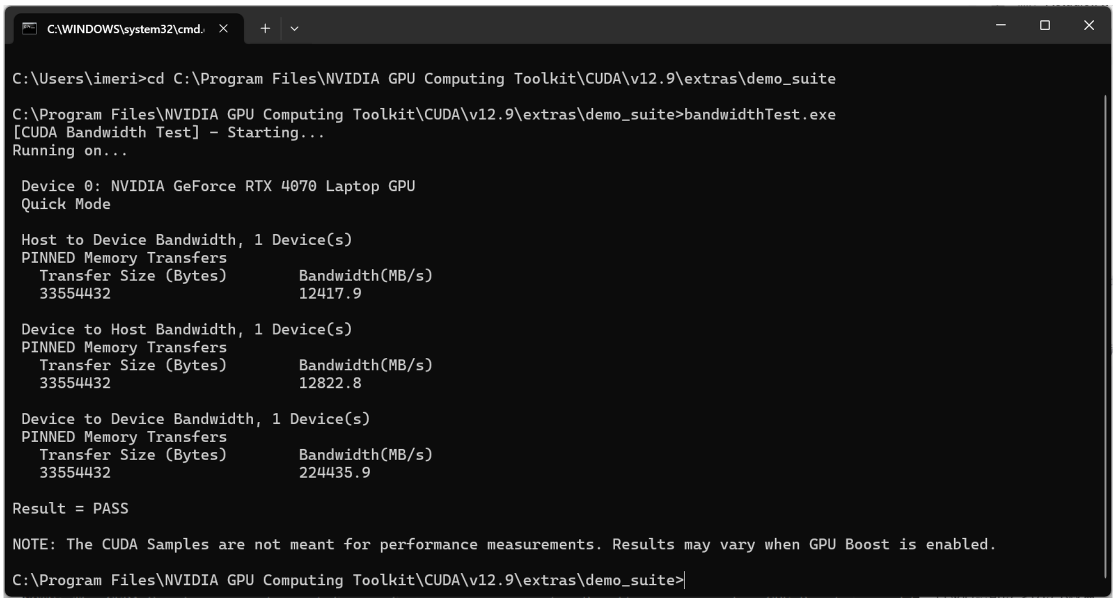
```
Microsoft Windows [版本 10.0.26100.4061]
(c) Microsoft Corporation。保留所有权利。

C:\Users\imeri>nvcc -V
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2025 NVIDIA Corporation
Built on Tue_May_27_02:24:01_Pacific_Daylight_Time_2025
Cuda compilation tools, release 12.9, V12.9.86
Build cuda_12.9.r12.9/compiler.36037853_0

C:\Users\imeri>
```

图 3.5：CUDA 已安装成功



```
  Total amount of constant memory:               zu bytes
  Total amount of shared memory per block:       zu bytes
  Total number of registers available per block: 65536
  Warp size:                                     32
  Maximum number of threads per multiprocessor:  1536
  Maximum number of threads per block:           1024
  Max dimension size of a thread block (x,y,z): (1024, 1024, 64)
  Max dimension size of a grid size    (x,y,z): (2147483647, 65535, 65535)
  Maximum memory pitch:                          zu bytes
  Texture alignment:                             zu bytes
  Concurrent copy and kernel execution:          Yes with 1 copy engine(s)
  Run time limit on kernels:                     Yes
  Integrated GPU sharing Host Memory:            No
  Support host page-locked memory mapping:       Yes
  Alignment requirement for Surfaces:            Yes
  Device has ECC support:                        Disabled
  CUDA Device Driver Mode (TCC or WDDM):         WDDM (Windows Display Driver Model)
  Device supports Unified Addressing (UVA):      Yes
  Device supports Compute Preemption:            Yes
  Supports Cooperative Kernel Launch:            Yes
  Supports MultiDevice Co-op Kernel Launch:      No
  Device PCI Domain ID / Bus ID / location ID:   0 / 1 / 0
  Compute Mode:
     < Default (multiple host threads can use ::cudaSetDevice() with device simultaneously) >

deviceQuery, CUDA Driver = CUDART, CUDA Driver Version = 12.9, CUDA Runtime Version = 12.9, NumDevs = 1, Device0 = NVIDI
A GeForce RTX 4070 Laptop GPU
Result = PASS

C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v12.9\extras\demo_suite>
```

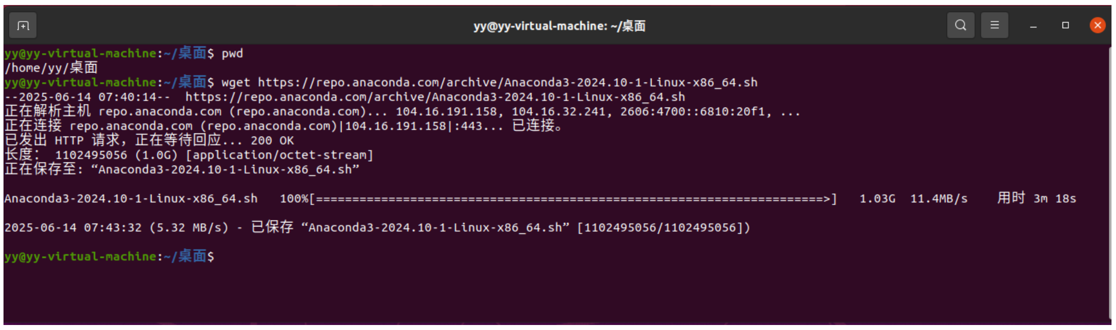
图 3.6：deviceQuery 运行截图



```
C:\Users\imeri>cd C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v12.9\extras\demo_suite

C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v12.9\extras\demo_suite>bandwidthTest.exe
[CUDA Bandwidth Test] - Starting...
Running on...

 Device 0: NVIDIA GeForce RTX 4070 Laptop GPU
 Quick Mode

 Host to Device Bandwidth, 1 Device(s)
 PINNED Memory Transfers
   Transfer Size (Bytes)        Bandwidth(MB/s)
   33554432                     12417.9

 Device to Host Bandwidth, 1 Device(s)
 PINNED Memory Transfers
   Transfer Size (Bytes)        Bandwidth(MB/s)
   33554432                     12822.8

 Device to Device Bandwidth, 1 Device(s)
 PINNED Memory Transfers
   Transfer Size (Bytes)        Bandwidth(MB/s)
   33554432                     224435.9

Result = PASS

NOTE: The CUDA Samples are not meant for performance measurements. Results may vary when GPU Boost is enabled.

C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v12.9\extras\demo_suite>
```

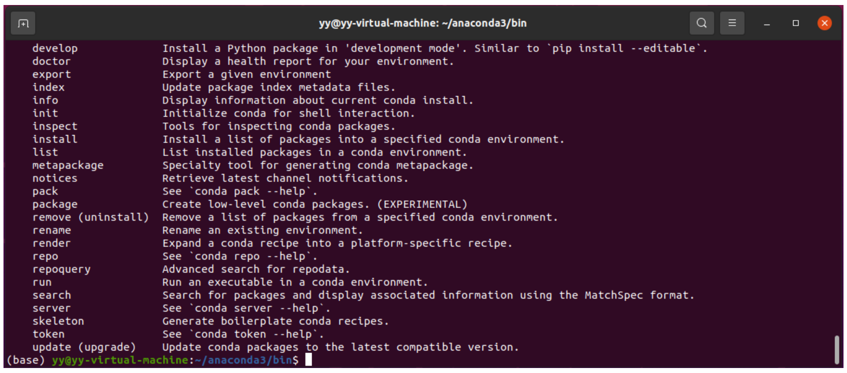
图 3.7：bandwidthTest 运行截图

四、PC 端环境开发（Linux）



```
yy@yy-virtual-machine:~/桌面$ pwd
/home/yy/桌面
yy@yy-virtual-machine:~/桌面$ wget https://repo.anaconda.com/archive/Anaconda3-2024.10-1-Linux-x86_64.sh
--2025-06-14 07:40:14--  https://repo.anaconda.com/archive/Anaconda3-2024.10-1-Linux-x86_64.sh
正在解析主机 repo.anaconda.com (repo.anaconda.com)... 104.16.191.158, 104.16.32.241, 2606:4700::6810:20f1, ...
正在连接 repo.anaconda.com (repo.anaconda.com)|104.16.191.158|:443... 已连接。
已发出 HTTP 请求，正在等待回应... 200 OK
长度：  1102495056 (1.0G) [application/octet-stream]
正在保存至: “Anaconda3-2024.10-1-Linux-x86_64.sh”

Anaconda3-2024.10-1-Linux-x86_64.sh   100%[===================================================================>]   1.03G  11.4MB/s    用时 3m 18s

2025-06-14 07:43:32 (5.32 MB/s) - 已保存 “Anaconda3-2024.10-1-Linux-x86_64.sh” [1102495056/1102495056])

yy@yy-virtual-machine:~/桌面$
```

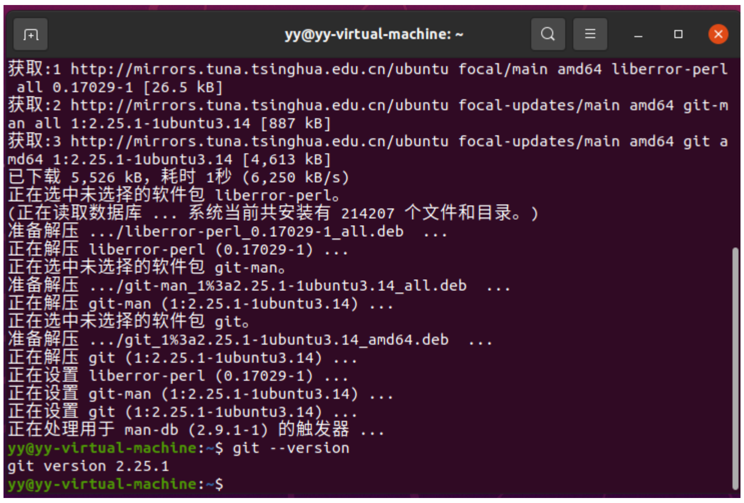
图 4.1：通过 wget 命令下载 Anaconda 安装包



```
    develop               Install a Python package in 'development mode'. Similar to `pip install --editable`.
    doctor                Display a health report for your environment.
    export                Export a given environment
    index                 Update package index metadata files.
    info                  Display information about current conda install.
    init                  Initialize conda for shell interaction.
    inspect               Tools for inspecting conda packages.
    install               Install a list of packages into a specified conda environment.
    list                  List installed packages in a conda environment.
    metapackage           Specialty tool for generating conda metapackage.
    notices               Retrieve latest channel notifications.
    pack                  See `conda pack --help`.
    package               Create low-level conda packages. (EXPERIMENTAL)
    remove (uninstall)    Remove a list of packages from a specified conda environment.
    rename                Rename an existing environment.
    render                Expand a conda recipe into a platform-specific recipe.
    repo                  See `conda repo --help`.
    repoquery             Advanced search for repodata.
    run                   Run an executable in a conda environment.
    search                Search for packages and display associated information using the MatchSpec format.
    server                See `conda server --help`.
    skeleton              Generate boilerplate conda recipes.
    token                 See `conda token --help`.
    update (upgrade)      Update conda packages to the latest compatible version.
(base) yy@yy-virtual-machine:~/anaconda3/bin$
```

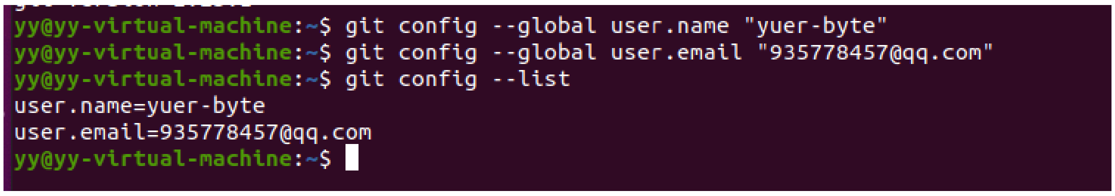
图 4.2：使用 bash 命令安装，并使用 nano 命令添加环境变量

操作	命令
重新加载 shell 配置	source ~/.bashrc
激活 base	conda activate base
退出 base	conda deactivate
自动激活 base	conda config --set auto_activate_base true
取消自动激活 base	conda config --set auto_activate_base false



```
获取:1 http://mirrors.tuna.tsinghua.edu.cn/ubuntu focal/main amd64 liberror-perl
 all 0.17029-1 [26.5 kB]
获取:2 http://mirrors.tuna.tsinghua.edu.cn/ubuntu focal-updates/main amd64 git-m
an all 1:2.25.1-1ubuntu3.14 [887 kB]
获取:3 http://mirrors.tuna.tsinghua.edu.cn/ubuntu focal-updates/main amd64 git a
md64 1:2.25.1-1ubuntu3.14 [4,613 kB]
已下载 5,526 kB，耗时 1秒 (6,250 kB/s)
正在选中未选择的软件包 liberror-perl。
(正在读取数据库 ... 系统当前共安装有 214207 个文件和目录。)
准备解压 .../liberror-perl_0.17029-1_all.deb  ...
正在解压 liberror-perl (0.17029-1) ...
正在选中未选择的软件包 git-man。
准备解压 .../git-man_1%3a2.25.1-1ubuntu3.14_all.deb  ...
正在解压 git-man (1:2.25.1-1ubuntu3.14) ...
正在选中未选择的软件包 git。
准备解压 .../git_1%3a2.25.1-1ubuntu3.14_amd64.deb  ...
正在解压 git (1:2.25.1-1ubuntu3.14) ...
正在设置 liberror-perl (0.17029-1) ...
正在设置 git-man (1:2.25.1-1ubuntu3.14) ...
正在设置 git (1:2.25.1-1ubuntu3.14) ...
正在处理用于 man-db (2.9.1-1) 的触发器 ...
yy@yy-virtual-machine:~$ git --version
git version 2.25.1
yy@yy-virtual-machine:~$
```

图 4.3：使用 sudo apt install git 命令安装 git



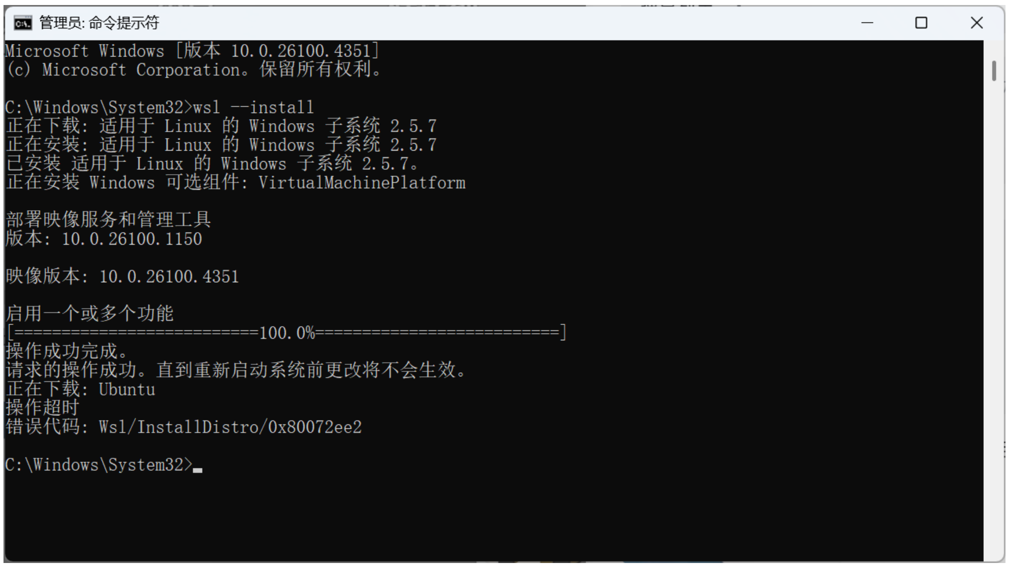
```
yy@yy-virtual-machine:~$ git config --global user.name "yuer-byte"
yy@yy-virtual-machine:~$ git config --global user.email "935778457@qq.com"
yy@yy-virtual-machine:~$ git config --list
user.name=yuer-byte
user.email=935778457@qq.com
yy@yy-virtual-machine:~$ 
```

图 4.4：配置 git

表 4.1：常用 git 命令

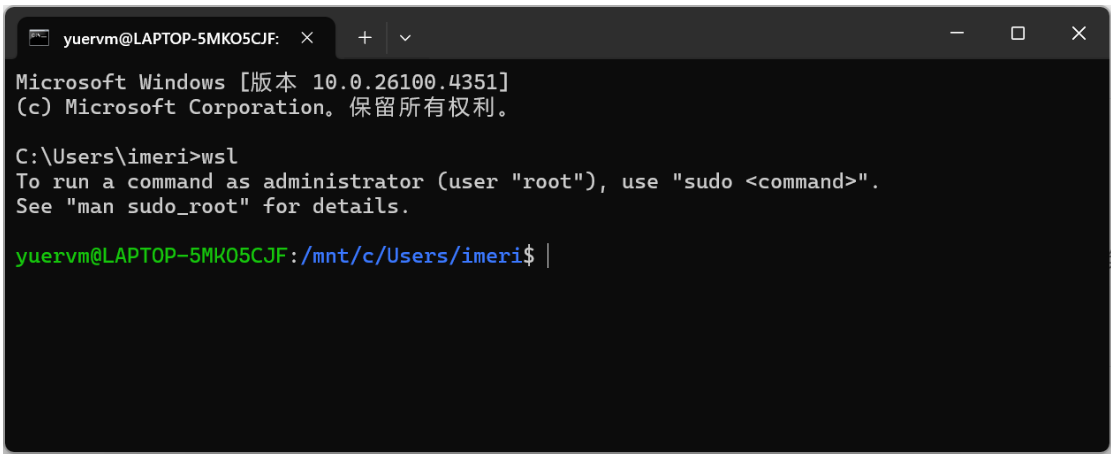
操作	**命令**
初始化仓库	Git init
查看状态	Git status
克隆远程仓库	Git clone https://github.com/yuer-byte/demo.git
添加文件到暂存区	Git add . #所有文件 Git add filename #单个文件
提交更改	Git commit -m “commit message”
推送更改到远程仓库	Git push origin main
拉取远程更新	Git pull origin main
查看提交历史	Git log
创建分支	Git branch new-branch Git checkout new-branch #切换分支
合并分支	Git checkout main Git merge new-branch
撤销更改	Git restore filename #未暂存的更改 Git reset HEAD filename #取消暂存 Git checkout – filename #丢弃更改

五、WSL 环境配置



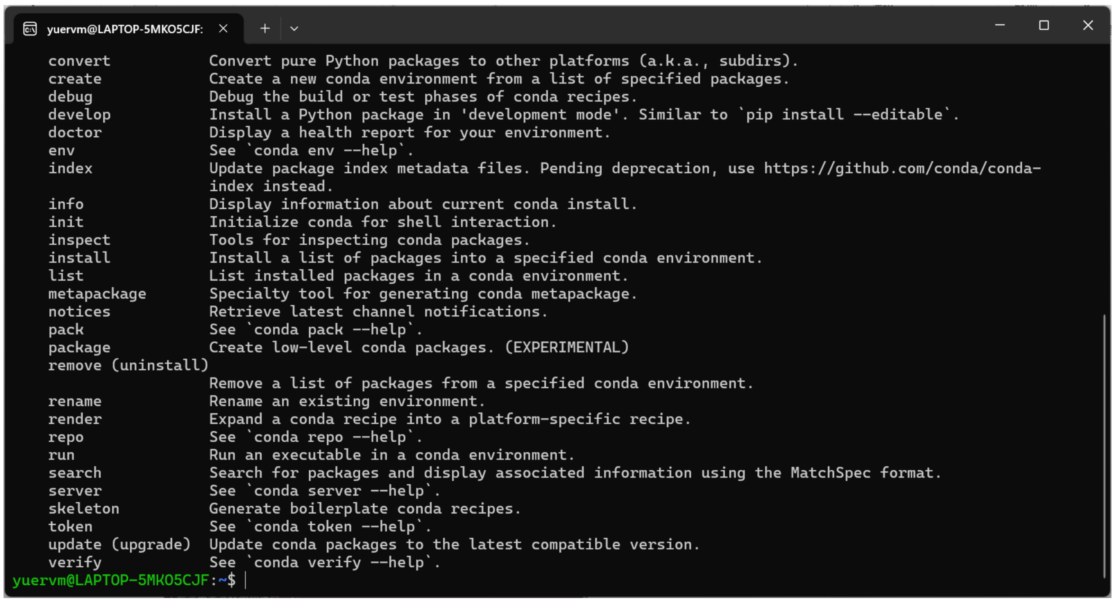
```
Microsoft Windows [版本 10.0.26100.4351]
(c) Microsoft Corporation。保留所有权利。

C:\Windows\System32>wsl --install
正在下载: 适用于 Linux 的 Windows 子系统 2.5.7
正在安装: 适用于 Linux 的 Windows 子系统 2.5.7
已安装 适用于 Linux 的 Windows 子系统 2.5.7。
正在安装 Windows 可选组件: VirtualMachinePlatform

部署映像服务和管理工具
版本: 10.0.26100.1150

映像版本: 10.0.26100.4351

启用一个或多个功能
[==========================100.0%==========================]
操作成功完成。
请求的操作成功。直到重新启动系统前更改将不会生效。
正在下载: Ubuntu
操作超时
错误代码: Wsl/InstallDistro/0x80072ee2

C:\Windows\System32>
```

图 5.1：安装 WSL



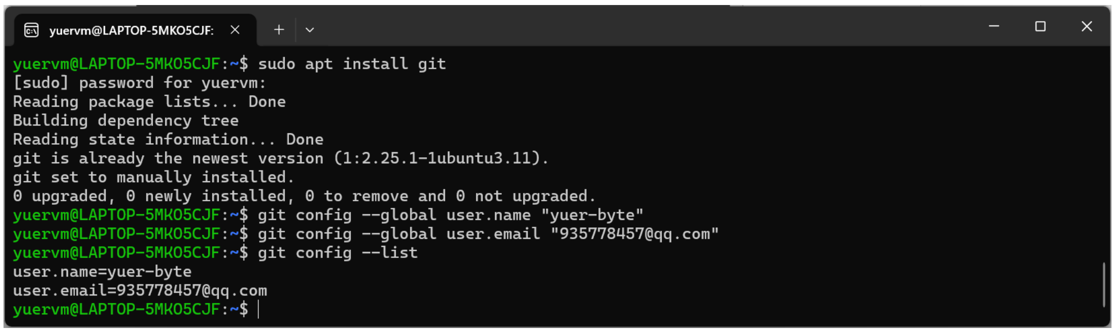
```
Microsoft Windows [版本 10.0.26100.4351]
(c) Microsoft Corporation。保留所有权利。

C:\Users\imeri>wsl
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

yuervm@LAPTOP-5MKO5CJF:/mnt/c/Users/imeri$
```

图 5.2：WSL (ubuntu 20.04) 安装成功



```
    convert            Convert pure Python packages to other platforms (a.k.a., subdirs).
    create             Create a new conda environment from a list of specified packages.
    debug              Debug the build or test phases of conda recipes.
    develop            Install a Python package in 'development mode'. Similar to `pip install --editable`.
    doctor             Display a health report for your environment.
    env                See `conda env --help`.
    index              Update package index metadata files. Pending deprecation, use https://github.com/conda/conda-
                       index instead.
    info               Display information about current conda install.
    init               Initialize conda for shell interaction.
    inspect            Tools for inspecting conda packages.
    install            Install a list of packages into a specified conda environment.
    list               List installed packages in a conda environment.
    metapackage        Specialty tool for generating conda metapackage.
    notices            Retrieve latest channel notifications.
    pack               See `conda pack --help`.
    package            Create low-level conda packages. (EXPERIMENTAL)
    remove (uninstall)
                       Remove a list of packages from a specified conda environment.
    rename             Rename an existing environment.
    render             Expand a conda recipe into a platform-specific recipe.
    repo               See `conda repo --help`.
    run                Run an executable in a conda environment.
    search             Search for packages and display associated information using the MatchSpec format.
    server             See `conda server --help`.
    skeleton           Generate boilerplate conda recipes.
    token              See `conda token --help`.
    update (upgrade)   Update conda packages to the latest compatible version.
    verify             See `conda verify --help`.
yuervm@LAPTOP-5MKO5CJF:~$
```

图 5.3：安装并配置 Anaconda

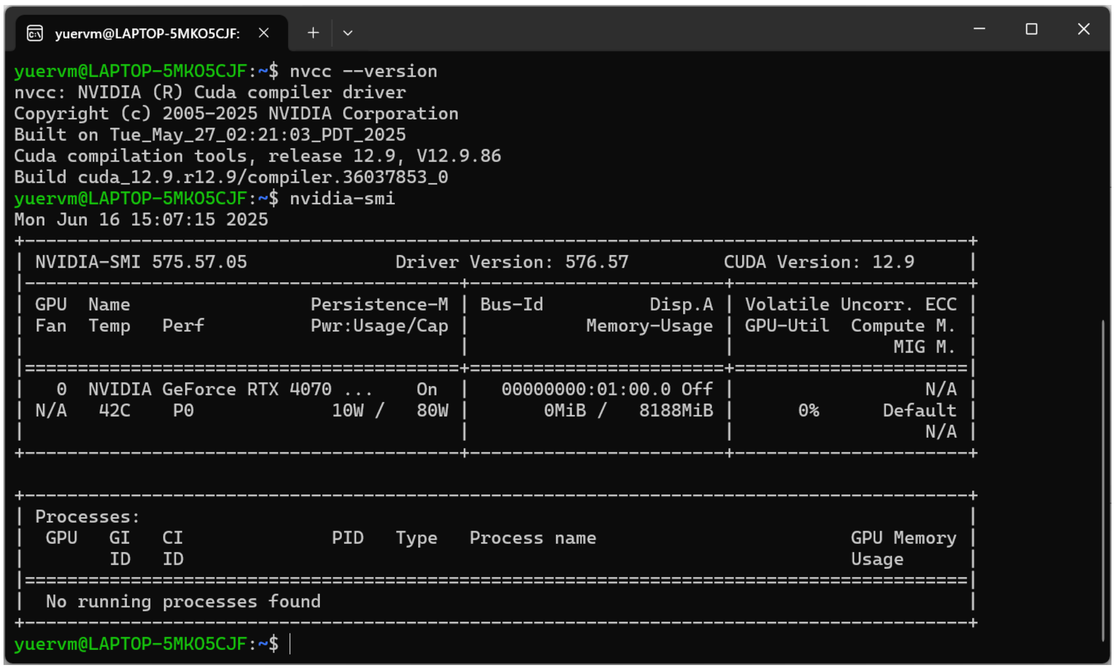


```
yuervm@LAPTOP-5MKO5CJF:~$ sudo apt install git
[sudo] password for yuervm:
Reading package lists... Done
Building dependency tree
Reading state information... Done
git is already the newest version (1:2.25.1-1ubuntu3.11).
git set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
yuervm@LAPTOP-5MKO5CJF:~$ git config --global user.name "yuer-byte"
yuervm@LAPTOP-5MKO5CJF:~$ git config --global user.email "935778457@qq.com"
yuervm@LAPTOP-5MKO5CJF:~$ git config --list
user.name=yuer-byte
user.email=935778457@qq.com
yuervm@LAPTOP-5MKO5CJF:~$
```

图 5.4：安装并配置 Git

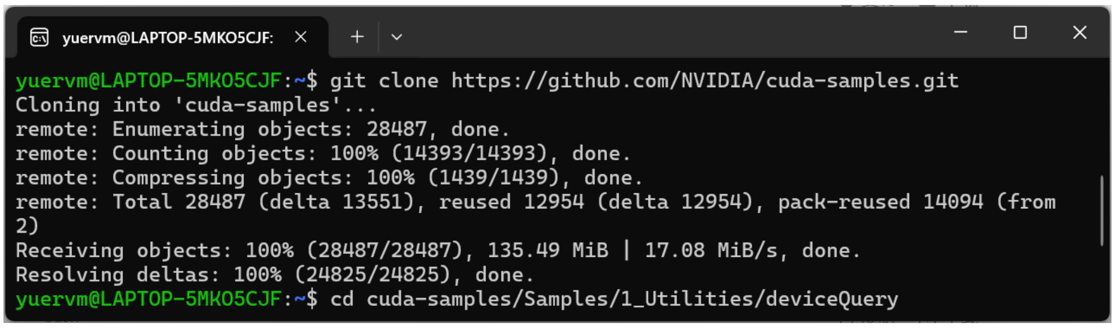
遇到无法连接 ubuntu 的问题，更换为国内镜像源：

```
sudo sed -i 's|http://.*ubuntu.com|http://mirrors.aliyun.com|g' /etc/apt/sources.list
sudo apt update
```



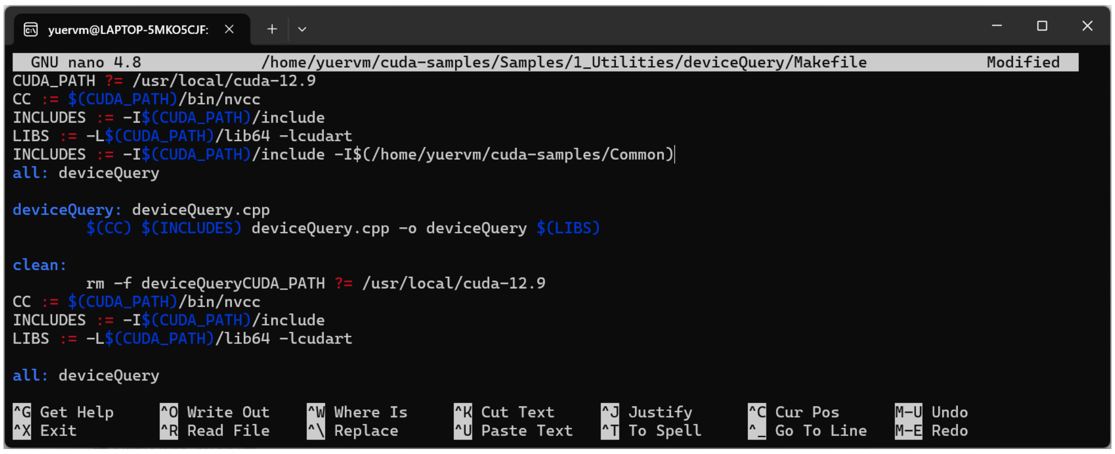
```
yuervm@LAPTOP-5MKO5CJF:~$ nvcc --version
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2025 NVIDIA Corporation
Built on Tue_May_27_02:21:03_PDT_2025
Cuda compilation tools, release 12.9, V12.9.86
Build cuda_12.9.r12.9/compiler.36037853_0
yuervm@LAPTOP-5MKO5CJF:~$ nvidia-smi
Mon Jun 16 15:07:15 2025
+-----------------------------------------------------------------------------------------+
| NVIDIA-SMI 575.57.05              Driver Version: 576.57         CUDA Version: 12.9     |
|-----------------------------------------+------------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id          Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |           Memory-Usage | GPU-Util  Compute M. |
|                                         |                        |               MIG M. |
|=========================================+========================+======================|
|   0  NVIDIA GeForce RTX 4070 ...    On  |   00000000:01:00.0 Off |                  N/A |
| N/A   42C    P0             10W /   80W |       0MiB /   8188MiB |      0%      Default |
|                                         |                        |                  N/A |
+-----------------------------------------+------------------------+----------------------+

+-----------------------------------------------------------------------------------------+
| Processes:                                                                              |
|  GPU   GI   CI              PID   Type   Process name                        GPU Memory |
|        ID   ID                                                               Usage      |
|=========================================================================================|
|  No running processes found                                                             |
+-----------------------------------------------------------------------------------------+
yuervm@LAPTOP-5MKO5CJF:~$
```

图 5.5：CUDA 安装成功



```
yuervm@LAPTOP-5MKO5CJF:~$ git clone https://github.com/NVIDIA/cuda-samples.git
Cloning into 'cuda-samples'...
remote: Enumerating objects: 28487, done.
remote: Counting objects: 100% (14393/14393), done.
remote: Compressing objects: 100% (1439/1439), done.
remote: Total 28487 (delta 13551), reused 12954 (delta 12954), pack-reused 14094 (from
2)
Receiving objects: 100% (28487/28487), 135.49 MiB | 17.08 MiB/s, done.
Resolving deltas: 100% (24825/24825), done.
yuervm@LAPTOP-5MKO5CJF:~$ cd cuda-samples/Samples/1_Utilities/deviceQuery
```

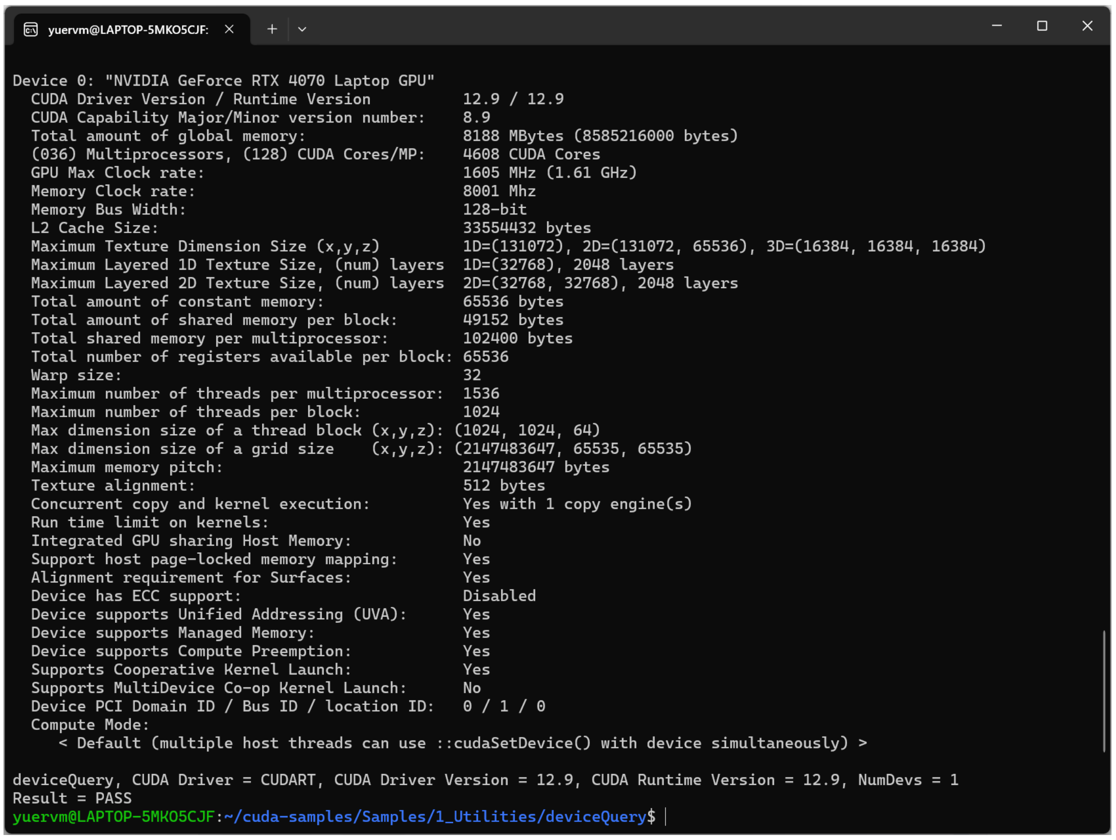
图 5.6：对缺少的 Samples 文件夹进行手动添加



```
  GNU nano 4.8            /home/yuervm/cuda-samples/Samples/1_Utilities/deviceQuery/Makefile            Modified
CUDA_PATH ?= /usr/local/cuda-12.9
CC := $(CUDA_PATH)/bin/nvcc
INCLUDES := -I$(CUDA_PATH)/include
LIBS := -L$(CUDA_PATH)/lib64 -lcudart
INCLUDES := -I$(CUDA_PATH)/include -I$(/home/yuervm/cuda-samples/Common)
all: deviceQuery

deviceQuery: deviceQuery.cpp
        $(CC) $(INCLUDES) deviceQuery.cpp -o deviceQuery $(LIBS)

clean:
        rm -f deviceQueryCUDA_PATH ?= /usr/local/cuda-12.9
CC := $(CUDA_PATH)/bin/nvcc
INCLUDES := -I$(CUDA_PATH)/include
LIBS := -L$(CUDA_PATH)/lib64 -lcudart

all: deviceQuery

^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur Pos     M-U Undo
^X Exit        ^R Read File   ^\ Replace     ^U Paste Text  ^T To Spell    ^_ Go To Line  M-E Redo
```

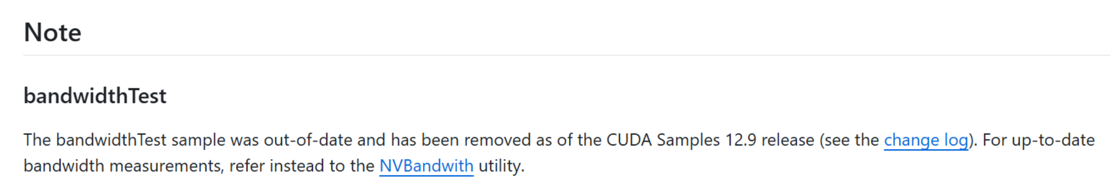
图 5.7：在 Make 文件中添加 helper_cuda.h 的路径

```
COMMON_DIR := /home/yuervm/cuda-samples/Common
INCLUDES := -I$(CUDA_PATH)/include -I$(COMMON_DIR)
```



```
Device 0: "NVIDIA GeForce RTX 4070 Laptop GPU"
  CUDA Driver Version / Runtime Version          12.9 / 12.9
  CUDA Capability Major/Minor version number:    8.9
  Total amount of global memory:                 8188 MBytes (8585216000 bytes)
  (036) Multiprocessors, (128) CUDA Cores/MP:    4608 CUDA Cores
  GPU Max Clock rate:                            1605 MHz (1.61 GHz)
  Memory Clock rate:                             8001 Mhz
  Memory Bus Width:                              128-bit
  L2 Cache Size:                                 33554432 bytes
  Maximum Texture Dimension Size (x,y,z)         1D=(131072), 2D=(131072, 65536), 3D=(16384, 16384, 16384)
  Maximum Layered 1D Texture Size, (num) layers  1D=(32768), 2048 layers
  Maximum Layered 2D Texture Size, (num) layers  2D=(32768, 32768), 2048 layers
  Total amount of constant memory:               65536 bytes
  Total amount of shared memory per block:       49152 bytes
  Total shared memory per multiprocessor:        102400 bytes
  Total number of registers available per block: 65536
  Warp size:                                     32
  Maximum number of threads per multiprocessor:  1536
  Maximum number of threads per block:           1024
  Max dimension size of a thread block (x,y,z): (1024, 1024, 64)
  Max dimension size of a grid size    (x,y,z): (2147483647, 65535, 65535)
  Maximum memory pitch:                          2147483647 bytes
  Texture alignment:                             512 bytes
  Concurrent copy and kernel execution:          Yes with 1 copy engine(s)
  Run time limit on kernels:                     Yes
  Integrated GPU sharing Host Memory:            No
  Support host page-locked memory mapping:       Yes
  Alignment requirement for Surfaces:            Yes
  Device has ECC support:                        Disabled
  Device supports Unified Addressing (UVA):      Yes
  Device supports Managed Memory:                Yes
  Device supports Compute Preemption:            Yes
  Supports Cooperative Kernel Launch:            Yes
  Supports MultiDevice Co-op Kernel Launch:      No
  Device PCI Domain ID / Bus ID / location ID:   0 / 1 / 0
  Compute Mode:
     < Default (multiple host threads can use ::cudaSetDevice() with device simultaneously) >

deviceQuery, CUDA Driver = CUDART, CUDA Driver Version = 12.9, CUDA Runtime Version = 12.9, NumDevs = 1
Result = PASS
yuervm@LAPTOP-5MKO5CJF:~/cuda-samples/Samples/1_Utilities/deviceQuery$
```

图 5.8：deviceQuery 运行截图

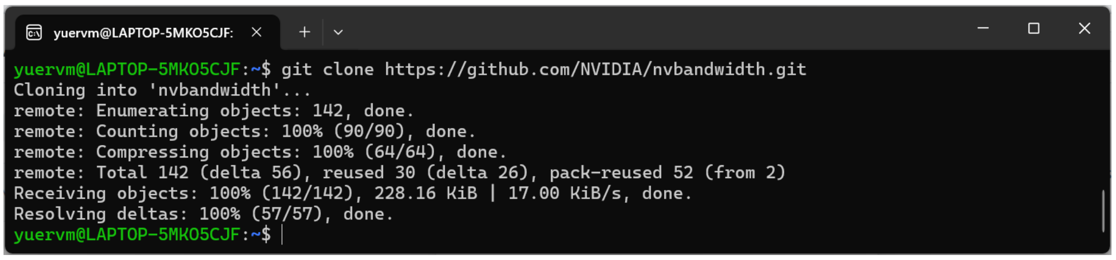


Note

bandwidthTest

The bandwidthTest sample was out-of-date and has been removed as of the CUDA Samples 12.9 release (see the change log). For up-to-date bandwidth measurements, refer instead to the NVBandwith utility.

图 5.9：clone 的 Samples 仓库中没有 bandwidthTest 文件

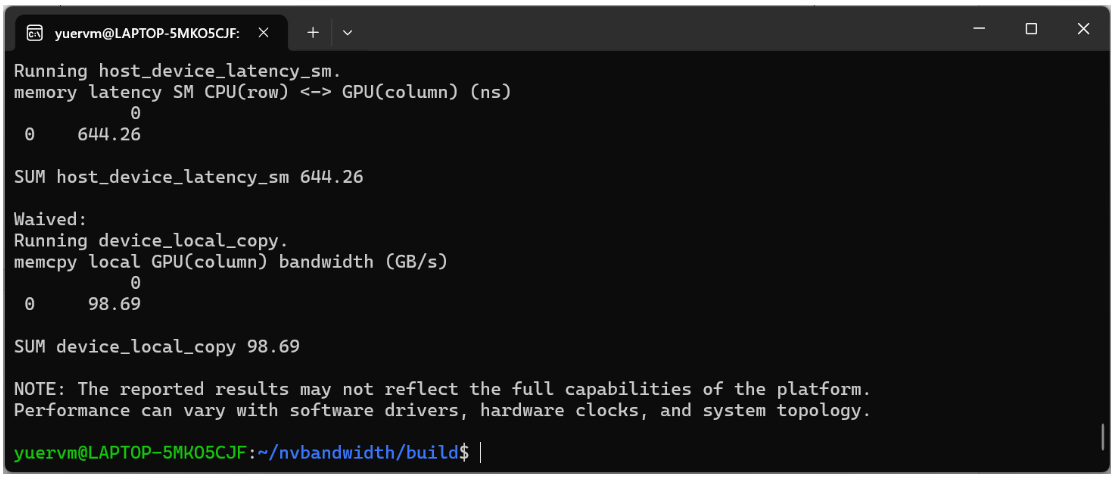


```
yuervm@LAPTOP-5MKO5CJF:~$ git clone https://github.com/NVIDIA/nvbandwidth.git
Cloning into 'nvbandwidth'...
remote: Enumerating objects: 142, done.
remote: Counting objects: 100% (90/90), done.
remote: Compressing objects: 100% (64/64), done.
remote: Total 142 (delta 56), reused 30 (delta 26), pack-reused 52 (from 2)
Receiving objects: 100% (142/142), 228.16 KiB | 17.00 KiB/s, done.
Resolving deltas: 100% (57/57), done.
yuervm@LAPTOP-5MKO5CJF:~$
```

图 5.10：下载 nvbandwidth

由于 nvbandwidth 仓库使用的是 CMake 构建系统，安装 CMake：

```
sudo apt update
sudo apt install -y cmake build-essential
sudo apt remove --purge cmake
wget -O - https://apt.kitware.com/keys/kitware-archive-latest.asc 2>/dev/null | sudo apt-key add
sudo apt-add-repository 'deb https://apt.kitware.com/ubuntu/ focal main'
sudo apt install cmake
```



```
Running host_device_latency_sm.
memory latency SM CPU(row) <-> GPU(column) (ns)
           0
 0    644.26

SUM host_device_latency_sm 644.26

Waived:
Running device_local_copy.
memcpy local GPU(column) bandwidth (GB/s)
           0
 0     98.69

SUM device_local_copy 98.69

NOTE: The reported results may not reflect the full capabilities of the platform.
Performance can vary with software drivers, hardware clocks, and system topology.

yuervm@LAPTOP-5MKO5CJF:~/nvbandwidth/build$
```

图 5.11：nvbandwidth 运行截图