

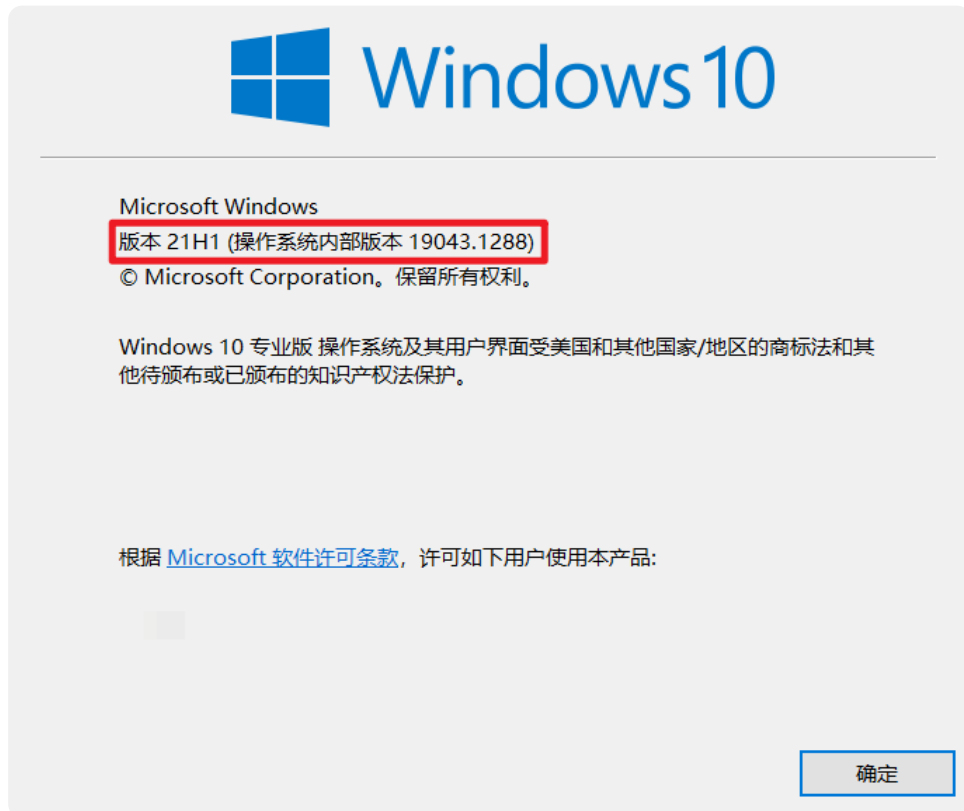
# SAM2环境配置指南

## 1. 准备工作

### 1.1 系统要求检查

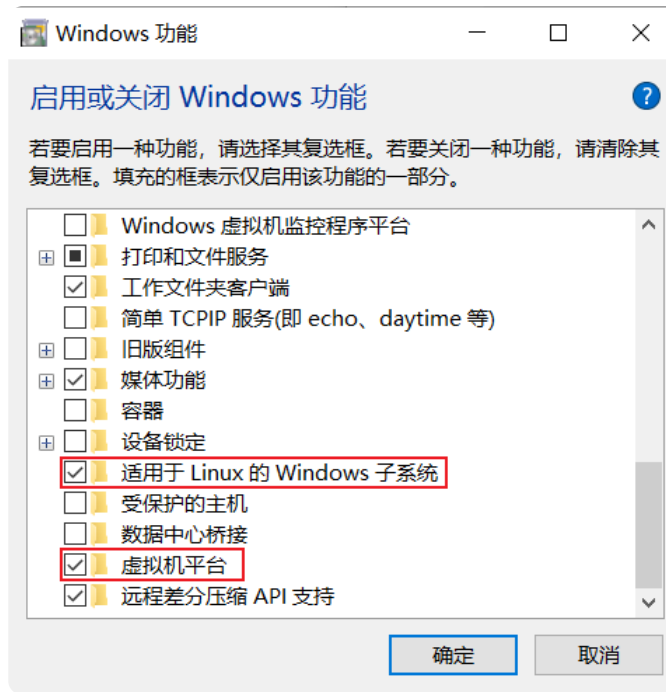
必须 Windows 10 版本 2004 及更高版本（内部版本 19041 及更高版本）或 Windows 11 才能继续使用如下命令。

查看方法：在键盘上按下 **win+R** 调出运行窗口，随后输入 **winver** 命令并回车：



### 1.2 启用适用于Linux的Windows子系统

打开控制面板-->程序-->启用或关闭Windows功能，依次勾选 **Hyper-V**、**适用于Linux的Windows子系统**、**虚拟机平台**，然后确认：

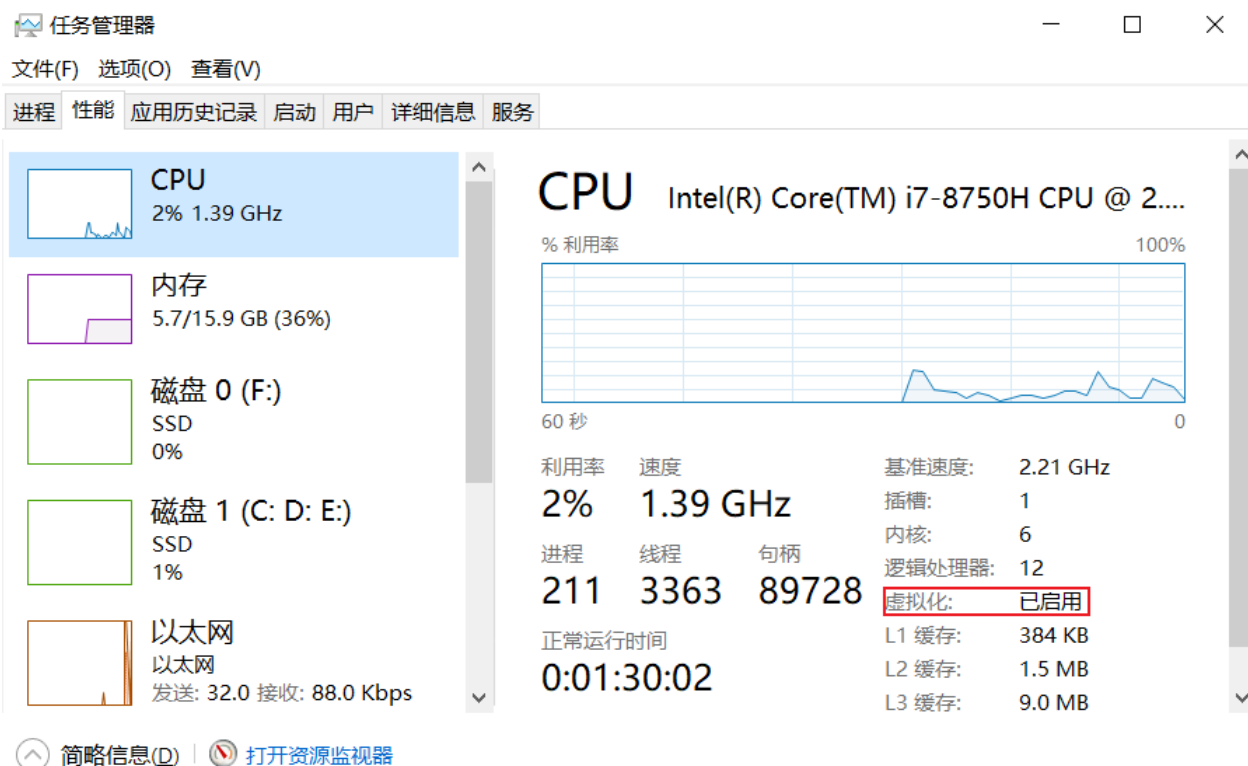


等待操作系统应用所做更改，完后后选择 **立即重新启动**：



### 1.3 确认已经启用虚拟机功能

打开任务管理器-->性能-->CPU，查看是否已经启动虚拟化：



## 2. 下载WSL Ubuntu分发

### 2.1 检查适用于本机的Ubuntu分发版本

按住 **Windows + X** 键找到 **Windows PowerShell(管理员)**，依次执行下面的命令更新wsl命令

```
wsl --update
wsl --shutdown
```

然后检查适用于本机WSL的Ubuntu分发版本:

```
wsl --list --online
```

```
PS C:\Windows\system32> wsl --list --online
以下是可安装的有效分发的列表。
请使用“wsl --install -d <分发>”安装。

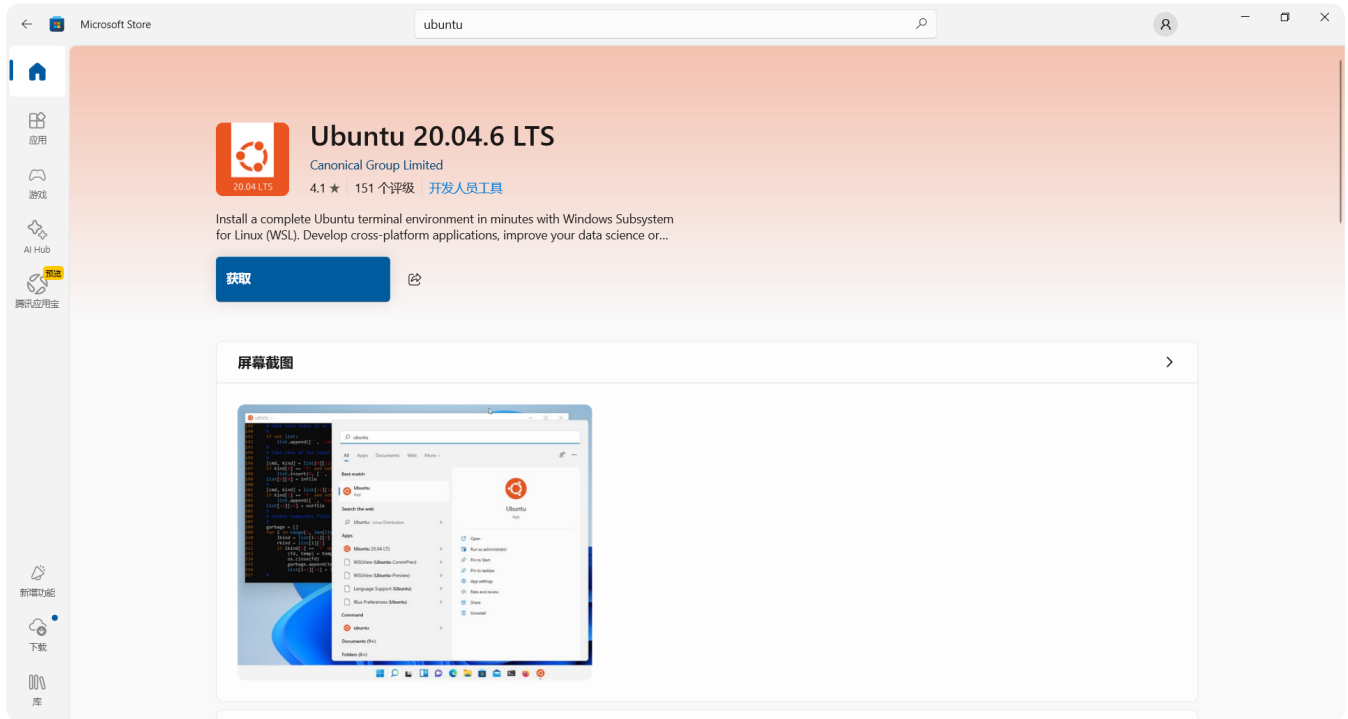
NAME                FRIENDLY NAME
Ubuntu              Ubuntu
Debian              Debian GNU/Linux
kali-linux           Kali Linux Rolling
openSUSE-42          openSUSE Leap 42
SLES-12              SUSE Linux Enterprise Server v12
Ubuntu-16.04         Ubuntu 16.04 LTS
Ubuntu-18.04         Ubuntu 18.04 LTS
Ubuntu-20.04         Ubuntu 20.04 LTS
PS C:\Windows\system32>
```

### 2.2 安装适合版本的Ubuntu分发

可以使用 `wsl --install -d Ubuntu-20.04` 命令行的方式安装Ubuntu分发，但是可能会报错：

```
PS C:\Windows\system32> wsl --install -d Ubuntu-20.04
正在下载: Ubuntu 20.04 LTS
安装过程中出现错误。分发名称: 'Ubuntu 20.04 LTS' 错误代码: 0x80072ee7
PS C:\Windows\system32>
```

这种情况一般是网络问题，可以选择到微软商店手动下载WSL Ubuntu20.04版本：



**注意：** 这个过程必须退出个人VPN，否则微软商店无法正常运行

## 2.3 配置Ubuntu系统

下载完成后，选择立即启动Ubuntu，此时系统会花几分钟时间配置Ubuntu，然后便可以进行相关设置：

1. 输入UNIX username:

```
Ubuntu 20.04.6 LTS
Installing, this may take a few minutes...
Please create a default UNIX user account. The username doe
For more information visit: https://aka.ms/wslusers
Enter new UNIX username:
```

**注意：** 用户名不能以大写字母开头

2. 输入password: (我的是12345)

jtl@DESKTOP-JTL: ~

```
adduser: Please enter a username matching the regular expression configured
via the NAME_REGEX[_SYSTEM] configuration variable. Use the '--force-badname'
option to relax this check or reconfigure NAME_REGEX.
Enter new UNIX username: 
New password: 
Retype new password: 
passwd: password updated successfully
Installation successful!
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 4.4.0-19041-Microsoft x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage
```

**注意：** 这里输入密码时，后边不会显示任何内容，输入完成后回车即可

- 在 Windows Powershell(管理员) 中使用命令设置WSL为版本2:

```
ws1.exe --set-version Ubuntu-20.04 2
```

最后一个数字就是版本号，需要将这里的分发版替换为实际下载的分发版

- 输入密码后就完成了基本的Ubuntu系统设置，可以再次使用wsl命令 `ws1 -l -v` 检查WSL Ubuntu版本号:

```
PS C:\Windows\system32> wsl -l -v
NAME                STATE              VERSION
* Ubuntu-20.04      Stopped            2
PS C:\Windows\system32>
```

## 2.4 (可选)更改Ubuntu系统位置

由于微软商店默认是将Ubuntu系统下载到C盘，但是进行大模型训练通常需要占用较大内存，因此可以将Ubuntu转移到其他非系统盘。

方法1 (不推荐) : 按住 **Windows + I** 打开设置->应用->Ubuntu->移动:

1747618788007

选择合适的盘符后移动，移动后会在目标盘符下创建两个目录:

WindowsApps	2025/5/19 9:40	文件夹
WpSystem	2025/5/19 9:39	文件夹

方法2: 在PowerShell(管理员)下执行如下命令:

- 确认WSI 已经关闭:

```
wsl --shutdown
```

- 在其他非系统盘新建一个目录，用于暂存WSL文件 (这个目录在完成系统钱以后可以删除) :

```
wsl --export Ubuntu-22.04 E:\Ubuntu\Ubuntu.tar
```

**注意：** 必须现在D盘下创建了Ubuntu后才能执行，因为这条命令不会自动创建目录，命令中的 **Ubuntu-22.04** 修改为自己的Ubuntu系统版本，可以通过 `ws1-l-v` 命令查看自己的版本号

- 确定在此目录下可以看见备份Ubuntu.tar文件之后，注销原有的wsl:

```
ws1 --unregister Ubuntu-22.04
```

4. 将第2步的备份文件迁移到自己想要放置的非系统盘位置中去：

```
ws1 --import Ubuntu-22.04 F:\Ubuntu E:\Ubuntu\ubuntu.tar
```

**注意：**命令中的最后一部分 **E:\Ubuntu\ubuntu.tar** 对应第2步自己的文件备份位置，命令中的 **F:\Ubuntu** 对应自定义的安装位置，最后会出现这么一个结果：

F:\Ubuntu				
名称	修改日期	类型	大小	
ext4.vhdx	2025/6/15 23:11	硬盘映像文件	1,306,624...	

说明系统迁移成功，这个文件以后就不要动了。

5. （如果已经配置好了conda环境，需要进行这一步）设置默认用户，这个用户名也就是之前ws1里的用户名，如果不设置这一步，会找不到conda位置，导致以前配置的环境消失，需要更复杂的操作才能找到conda。

```
This message is shown once a day.
/home/jtl/.hushlogin file.
(base) jtl@DESKTOP-JTL:~$
```

@符号前的就是你之前的用户名，然后执行以下命令：

```
ubuntu2204.exe config --default-user jtl # jtl就是自己的用户名
```

在导入任意盘linux系统时，我是用的默认名字，所以这里是ubuntu2204.exe；如果你起的名字是Ubuntu-20.04，那这里就是ubuntu2004.exe；如果你起的名字是ubuntu-18.04，那这里就是ubuntu1804.exe，名字可以在windows powershell通过 **ws1 -l -v** 命令查看：

```
PS C:\Users\JTL> ws1 -l -v
NAME      STATE      VERSION
* Ubuntu-22.04 Running    2
PS C:\Users\JTL>
```

### 3. 在Ubuntu中安装Anaconda 3

#### 3.1 确认anaconda版本

在 <https://repo.anaconda.com/archive/> 网站中找到想要安装的anaconda 3的版本(后缀名为.sh)，并复制其文件名：

Anaconda3-2023.07-0-Linux-aarch64.sh	711.9M	2023-07-11 13:17:00	bf486b448c111
Anaconda3-2023.03-1-Windows-x86_64.exe	786.6M	2023-04-24 12:41:07	f13a2ae812d20
Anaconda3-2023.03-1-MacOSX-x86_64.sh	601.6M	2023-04-24 12:41:07	3593921c8a551
Anaconda3-2023.03-1-MacOSX-x86_64.pkg	600.1M	2023-04-24 12:41:06	561ea77b7172e
Anaconda3-2023.03-1-MacOSX-arm64.sh	566.0M	2023-04-24 12:41:06	85152324c423f
Anaconda3-2023.03-1-MacOSX-arm64.pkg	564.4M	2023-04-24 12:41:06	d22ab7a22ab4b
Anaconda3-2023.03-1-Linux-x86_64.sh	860.6M	2023-04-24 12:41:05	95102d7c73241
Anaconda3-2023.03-1-Linux-s390x.sh	361.2M	2023-04-24 12:41:05	5af1406c6350b
Anaconda3-2023.03-1-Linux-ppc64le.sh	435.1M	2023-04-24 12:41:05	a31f2d6da8353
Anaconda3-2023.03-1-Linux-aarch64.sh	618.7M	2023-04-24 12:41:04	54e600faa2af6
Anaconda3-2023.03-0-Windows-x86_64.exe	786.0M	2023-03-20 10:41:36	849daee6c1926
Anaconda3-2023.03-0-MacOSX-x86_64.sh	601.0M	2023-03-20 10:41:36	cc37b1eb85bdc
Anaconda3-2023.03-0-MacOSX-x86_64.pkg	599.7M	2023-03-20 10:41:36	52d1109e37185
Anaconda3-2023.03-0-MacOSX-arm64.sh	565.4M	2023-03-20 10:41:35	d27ee54324389

#### 3.2 下载Anaconda安装脚本

打开Ubuntu终端，使用wget命令下载Anaconda的安装脚本：

```
wget https://repo.anaconda.com/archive/Anaconda3-2023.03-1-Linux-x86_64.sh
```

**注意：**将最后一部分替换为刚刚选择的版本名称。

这是使用anaconda官网的，下载速度比较慢，可以使用清华大学的镜像源：

```
wget https://mirrors.tuna.tsinghua.edu.cn/anaconda/archive/Anaconda3-2023.03-1-Linux-x86_64.sh
```

### 3.3 运行Anaconda安装脚本

下载完成后，使用 **bash** 命令运行脚本：

```
bash Anaconda3-2023.03-1-Linux-x86_64.sh
```

**注意：**将最后一部分替换为刚刚选择的版本名称。

```
jtl@DESKTOP-JTL: ~$ bash Anaconda3-2023.03-1-Linux-x86_64.sh

Welcome to Anaconda3 2023.03-1

In order to continue the installation process, please review the license
agreement.
Please, press ENTER to continue
>>> ENTER
```

### 3.4 遵循安装向导

回车键按下后，会弹出相关协议，可以一直回车翻阅协议，直到最后输入yes接受许可协议，然后可以按下回车键以接受默认安装位置或者自定义新位置：

```
>>> yes

Anaconda3 will now be installed into this location:
/home/jtl/anaconda3

- Press ENTER to confirm the location
- Press CTRL-C to abort the installation
- Or specify a different location below

[/home/jtl/anaconda3] >>> 
PREFIX=/home/
Unpacking payload ...
Extracting : platformdirs-2.5.2-py310h06a4308_0.conda: 61% | 266/439 [00:21<00:16, 10.29it/s]
```

### 3.5 初始化Anaconda

安装结束后，安装程序将询问你是否希望运行conda init，这将允许你在任何新的终端会话中使用conda。输入yes以继续：

```

done
installation finished.
Do you wish the installer to initialize Anaconda3
by running conda init? [yes|no]
[no] >>> yes
no change      /home/jtl/.../condabin/conda
no change      /home/jtl/.../bin/conda
no change      /home/jtl/.../bin/conda-env
no change      /home/jtl/.../bin/activate
no change      /home/jtl/.../bin/deactivate
no change      /home/jtl/.../etc/profile.d/conda.sh
no change      /home/jtl/.../etc/fish/conf.d/conda.fish
no change      /home/jtl/.../shell/condabin/Conda.ps1
no change      /home/jtl/.../shell/condabin/conda-hook.ps1
no change      /home/jtl/.../lib/python3.10/site-packages/xontrib/conda.xsh
no change      /home/jtl/.../etc/profile.d/conda.csh
modified       /home/jtl/.bashrc

==> For changes to take effect, close and re-open your current shell. <==

If you'd prefer that conda's base environment not be activated on startup,
set the auto_activate_base parameter to false:

conda config --set auto_activate_base false

Thank you for installing Anaconda3!
jtl@DESKTOP-JTL:~$

```

### 3.6 激活配置

可以选择重新启动Ubuntu系统或者键入以下命令来激活配置：

```

source ~/.bashrc

```

### 3.7 确认Anaconda安装成功

激活配置后，命令提示符前会显示(base)，这是conda的默认环境，还可以使用 `conda list` 命令检查安装的包：

```

jtl@DESKTOP-JTL:~$ source ~/.bashrc
(base) jtl@DESKTOP-JTL:~$ conda list
# packages in environment at /home/jtl/tanlin:
#
# Name                                Version           Build    Channel
#-----
libgcc_mutex                         0.1               main
openmp_mutex                         5.1               1_gnu
alabaster                            0.7.12            pyhd3eb1b0_0
anaconda-client                      1.11.2            py310h06a4308_0
anaconda-navigator                   2.4.0             py310h06a4308_0
anaconda-project                     0.11.1            py310h06a4308_0
anyio                                 3.5.0             py310h06a4308_0
appdirs                              1.4.4             pyhd3eb1b0_0
argon2-cffi                          21.3.0            pyhd3eb1b0_0
argon2-cffi-bindings                 21.2.0            py310h7f8727e_0

```

### 3.8 创建新的环境

为了避免以后出错而影响到原来的默认环境，建议新建一个环境：



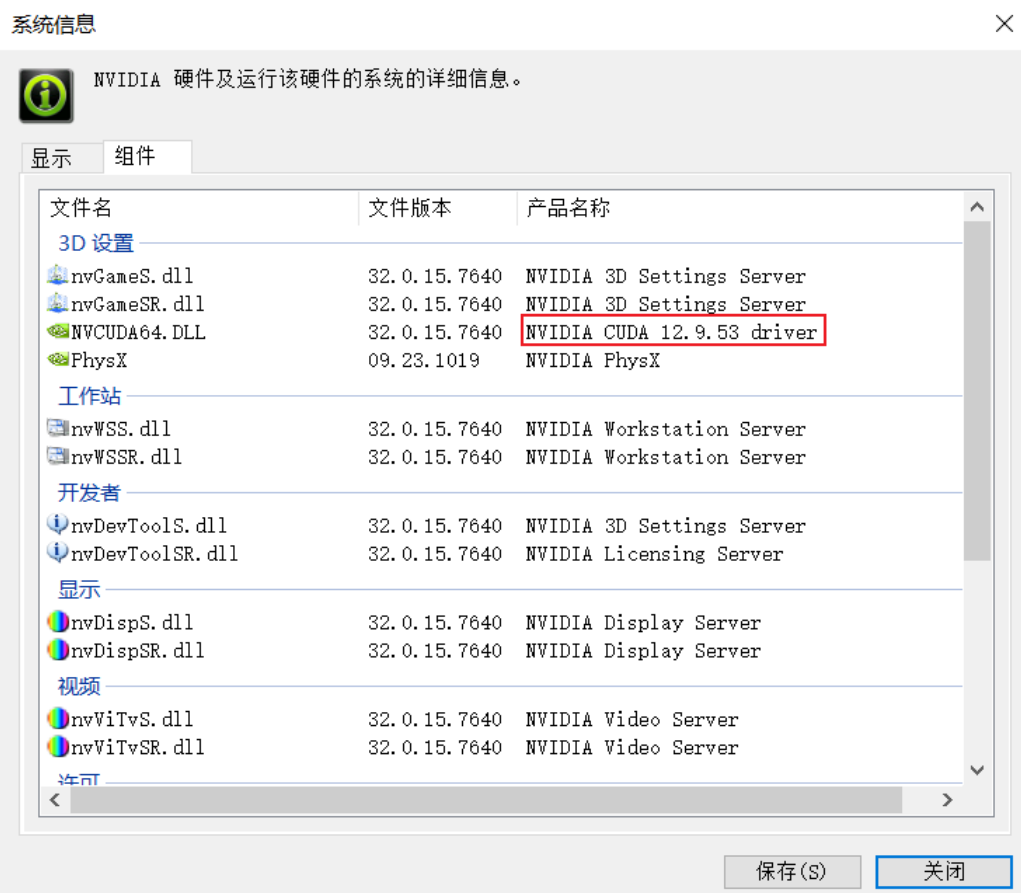


```
> conda create -n DL python=3.11 # -n是name的缩写 roomName可以自己取名字, python=3.11是自己指定的python版本
>
> conda activate DL # 进入新的环境
```

## 4. 在Anaconda中安装CUDA

### 4.1 确认驱动信息

在桌面空白处, 右键-->NVIDIA控制面板-->左下角系统信息-->组件-->3D设置, 查看Cuda版本:



或者在 **cmd命令行** 中使用命令: **nvidia-smi** :

```
C:\Users\JTL>nvidia-smi
Mon May 19 20:36:15 2025

+-----+
| NVIDIA-SMI 576.40                  Driver Version: 576.40      CUDA Version: 12.9 |
+-----+-----+
| GPU   Name                Driver-Model  Bus-Id        Disp. A   Volatile Uncorr. ECC |
| Fan  Temp  Perf           Pwr:Usage/Cap     Memory-Usage GPU-Util  Compute M. |
|                               |                      |              MIG M. |
+-----+-----+
|  0  NVIDIA GeForce GTX 1050 Ti  WDDM        00000000:01:00.0 Off          N/A |
| N/A   42C    P8              N/A / 18W           0MiB / 4096MiB    0%      Default |
|                               |                      |              N/A |
+-----+-----+

Processes:
+-----+
| GPU  GI  CI           PID  Type  Process name                      GPU Memory |
|   ID  ID  ID              |    |               |                      Usage |
+-----+-----+
|  0  N/A N/A           6724  C+G  C:\Windows\explorer.exe           N/A |
|  0  N/A N/A           7176  C+G  ...26wp6bftszej\TranslucentTB.exe N/A |
|  0  N/A N/A           8800  C+G  ...h_cw5nlh2txyewy\SearchApp.exe  N/A |
|  0  N/A N/A          10668  C+G  ...wy\InputApp\TextInputHost.exe   N/A |
|  0  N/A N/A          11912  C+G  ...b3d8bbwe\Microsoft.Photos.exe  N/A |
|  0  N/A N/A          13672  C+G  ...xyewy\ShellExperienceHost.exe   N/A |
|  0  N/A N/A          16284  C+G  ...ntrolPanel\SystemSettings.exe   N/A |
+-----+-----+
```

确保已经安装nvidia驱动，只需在Windows上安装好GPU驱动程序，WSL2上便可以获得驱动的支持。

4.2 安装CUDA Toolkit

前往<https://developer.nvidia.com/cuda-toolkit-archive>找到对应的CUDA Toolkit，并选择适合自己电脑的cuda：

Operating System

Linux

Windows

Architecture

x86\_64

arm64-sbsa

aarch64-jetson

Distribution

Amazon-Linux

Azure-Linux

Debian

Fedora

KylinOS

OpenSUSE

Oracle-Linux

RHEL

Version

2.0

Installer Type

deb (local)

deb (network)

runfile (local)

选择好相关信息后，网页下边会出现相关的命令，把这些命令依次复制自己新建的anaconda环境中执行：

## Download Installer for Linux WSL-Ubuntu 2.0 x86\_64

The base installer is available for download below.

### > CUDA Toolkit Installer

#### Installation Instructions:

```
$ wget https://developer.download.nvidia.com/compute/cuda/repos/wsl-ubuntu/x86_64/cuda-keyring_1.1-1_all.deb
$ sudo dpkg -i cuda-keyring_1.1-1_all.deb
$ sudo apt-get update
$ sudo apt-get -y install cuda-toolkit-12-9
```

Additional installation options are detailed [here](#).

The CUDA Toolkit contains Open-Source Software. The source code can be found [here](#).

The checksums for the installer and patches can be found in [Installer Checksums](#).

For further information, see the [Installation Guide for Linux](#) and the [CUDA Quick Start Guide](#).

#### Installation Instructions:



```
wget https://developer.download.nvidia.com/compute/cuda/repos/wsl-ubuntu/x86_64/cuda-keyring_1.1-1_all.deb
sudo dpkg -i cuda-keyring_1.1-1_all.deb
sudo apt-get update
sudo apt-get -y install cuda-toolkit-12-9
```

只需遵循安装向导一步步执行即可。

如果在第一步 **wget** 遇到问题：

```
(DL) jt1@DESKTOP-JTL:~$ wget https://developer.download.nvidia.com/compute/cuda/repos/wsl-ubuntu/x86_64/cuda-keyring_1.1-1_all.deb
--2025-06-07 00:44:37-- https://developer.download.nvidia.com/compute/cuda/repos/wsl-ubuntu/x86_64/cuda-keyring_1.1-1_all.deb
Resolving developer.download.nvidia.com (developer.download.nvidia.com)... 23.55.44.101, 23.55.44.103, 88.221.81.192, ...
Connecting to developer.download.nvidia.com (developer.download.nvidia.com)|23.55.44.101|:443... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://developer.download.nvidia.cn/compute/cuda/repos/wsl-ubuntu/x86_64/cuda-keyring_1.1-1_all.deb [following]
--2025-06-07 00:44:42-- https://developer.download.nvidia.cn/compute/cuda/repos/wsl-ubuntu/x86_64/cuda-keyring_1.1-1_all.deb
Resolving developer.download.nvidia.cn (developer.download.nvidia.cn)... failed: Temporary failure in name resolution.
wget: unable to resolve host address 'developer.download.nvidia.cn'
```

仅需将DNS修改为：Google DNS：8.8.8.8，8.8.4.4

第三步和最后一步需要花费一定的时间，最后一步花费的时间最长（几分钟到十几分钟）

## 5. 在Anaconda中安装Pytorch

在官网<https://pytorch.org/get-started/locally/>中找到对应自己系统的torch版本：

PyTorch Build	Stable (2.7.0)			Preview (Nightly)	
Your OS	Linux		Mac		Windows
Package	Pip		LibTorch		Source
Language	Python			C++ / Java	
Compute Platform	CUDA 11.8	CUDA 12.6	CUDA 12.8	ROCm 6.3	CPU
Run this Command:	pip3 install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu128				

我的CUDA版本是12.9，但这里最高是12.8，经验证，12.8依然可以正确安装并运行。

● ● ●

```
pip3 install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu128
```

将命令复制到Anaconda自己的环境中执行，执行完成后可通过 `conda list` 查看torch是否安装完成

这一步也需要花费较长时间

## 6. 安装SAM2大模型

前往Sam2官网<https://github.com/facebookresearch/sam2>找到Installation部分，这里是关于sam2的安装教程，依次执行下面的命令：

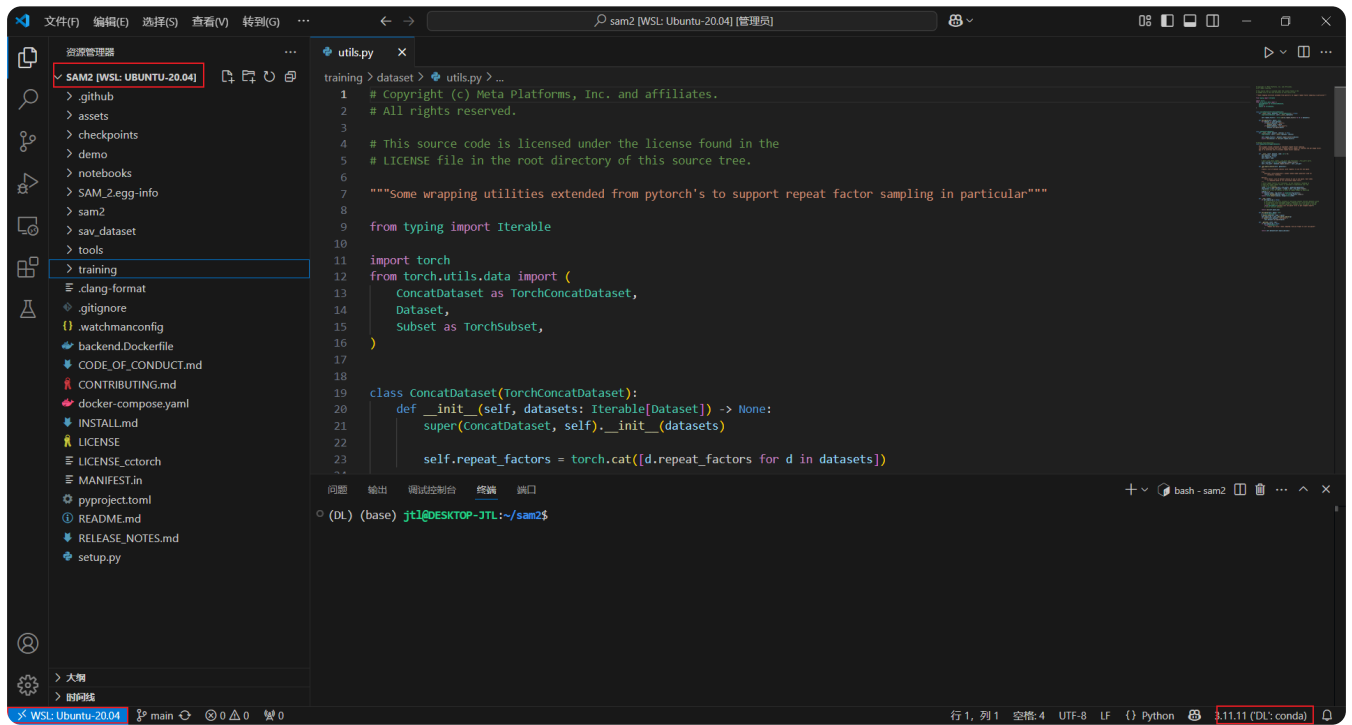
● ● ●

```
git clone https://github.com/facebookresearch/sam2.git && cd sam2

pip install -e .
```

## 7. 链接WSL2和VSCode

在wsl ubuntu中，先进入想要打开的文件目录，比如进入sam2，然后通过 `code .` 命令即可启动VS Code:



VSCode中需要安装的插件:

