# Experiment 3 : Pytorch implementation of graph convolutional neural network classification

## Environment preparation

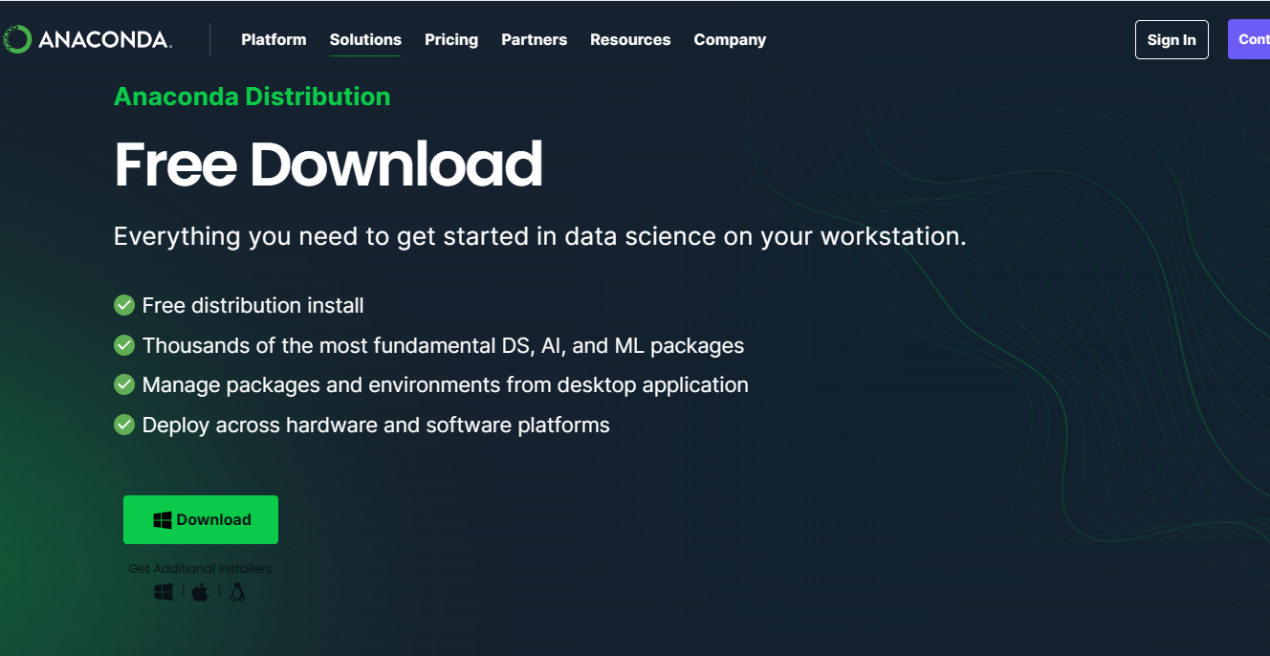
If you already have installed anaconda and a python IDE like pycharm or vscode just skip the step of install Anaconda and pycharm

### Anaconda

#### Download

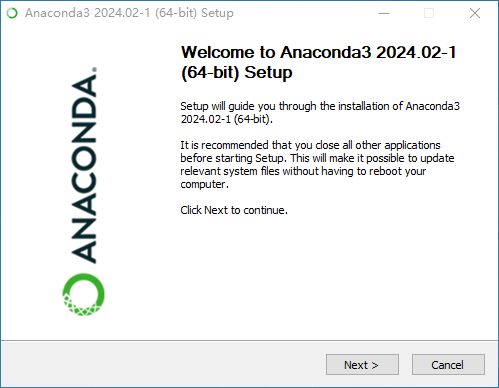
You can download the anaconda ( an open-source package and environment management system ) by clicking here or visiting the official website https://www.anaconda.com/download/success.

Click the Download as the picture to download the anaconda.

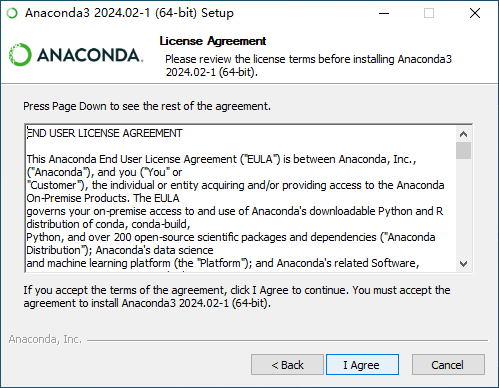


#### Install anaconda:

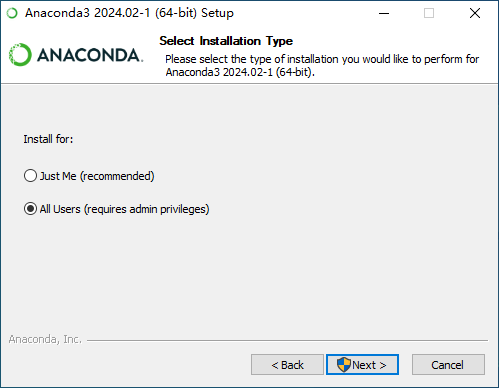
Find the folder where you downloaded the installation package and click Installer. You will see this page



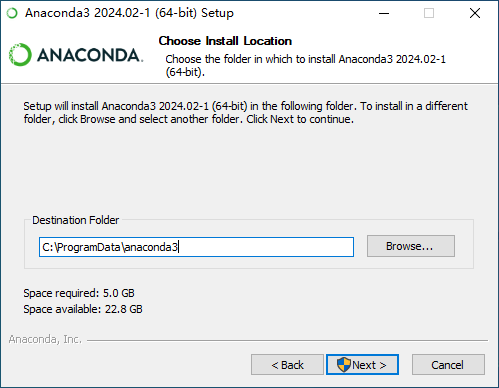
Click the Next and will jump into this page.



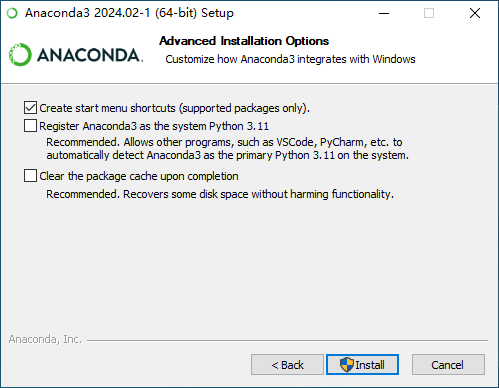
Click the I Agree.



Recommend choose All Users and click Next.



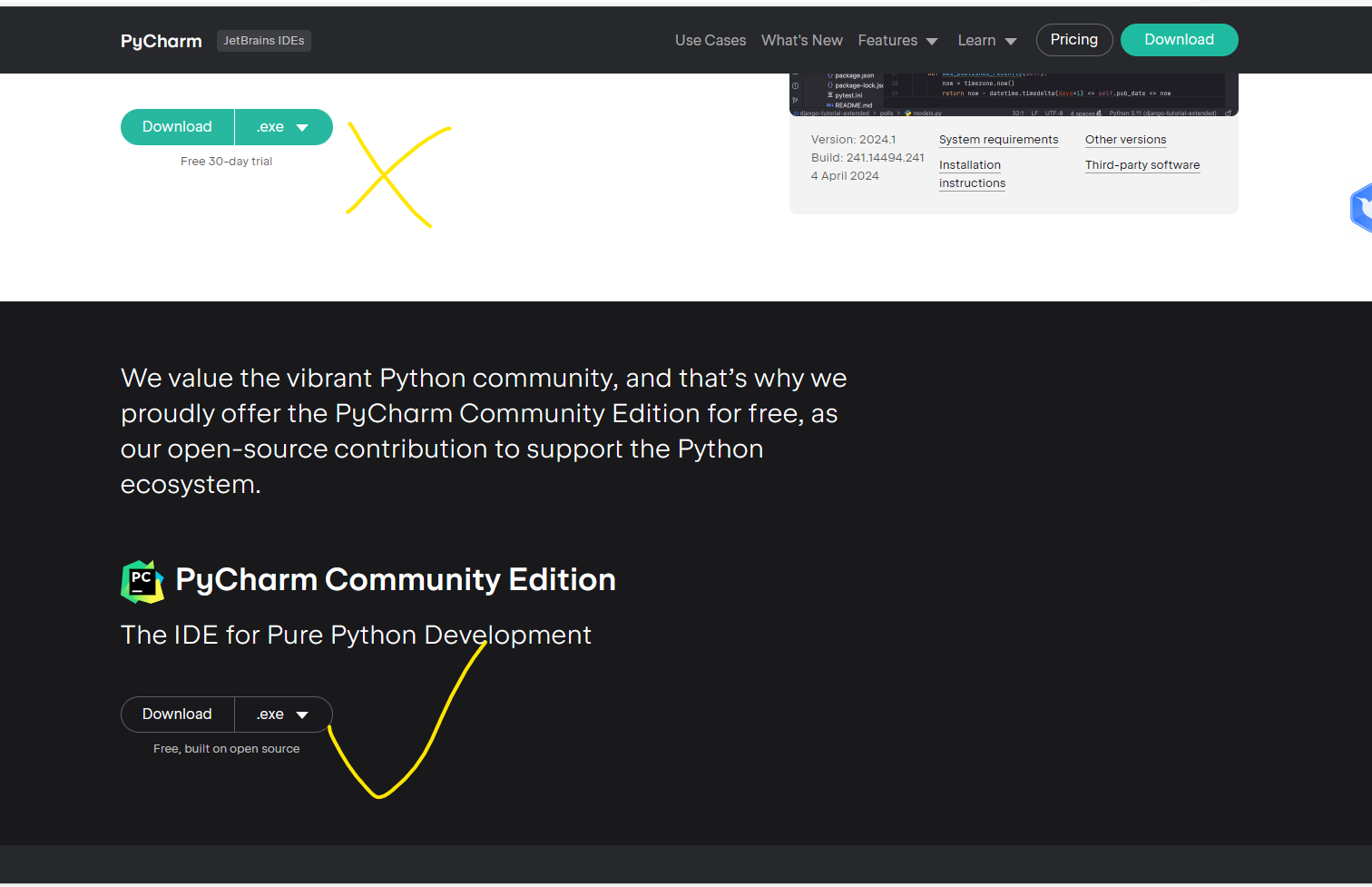
Choose one folder (In order to avoid possible errors, try not to have Chinese characters in the file path.)to install the anaconda. Then click the Next.



Click the Install, Wait until the anaconda is installed completely.

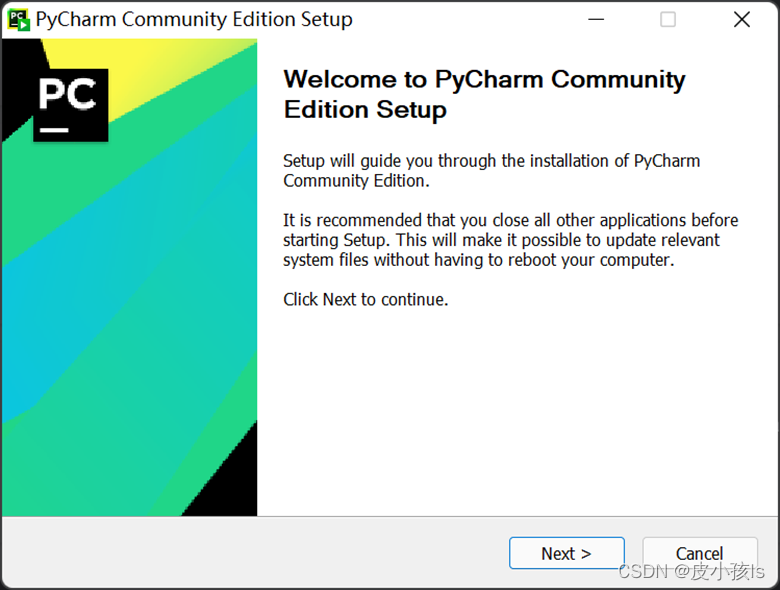
## Download and insatll Pycharm

Visit Jetbrains' official website: <https://www.jetbrains.com/pycharm/download/?section=windows> **Notice !** Find the Pycharm Community Edition not the Professional and download it. Follow the instruction in the picture .Or you can just click [here](https://download-cdn.jetbrains.com/python/pycharm-community-2024.1.exe).

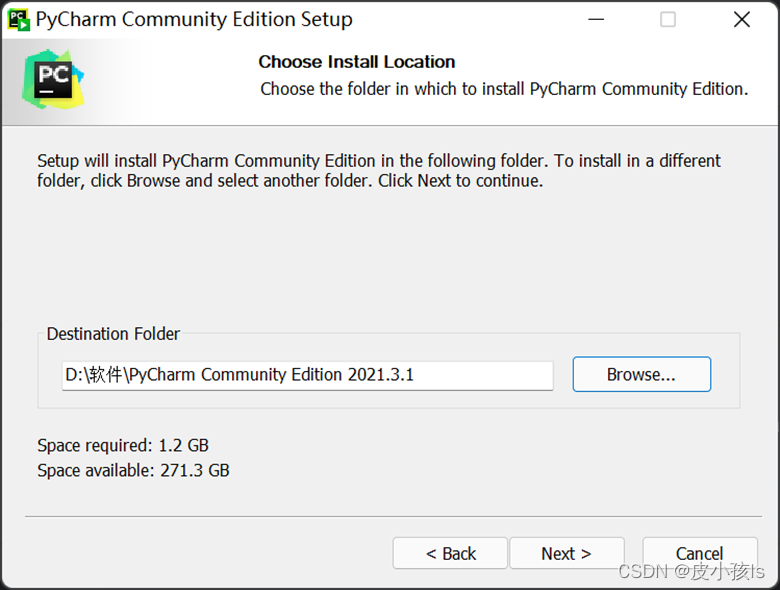


## Install the Pycharm (reference from [CSDN](https://blog.csdn.net/qq_44809707/article/details/122501118) ):

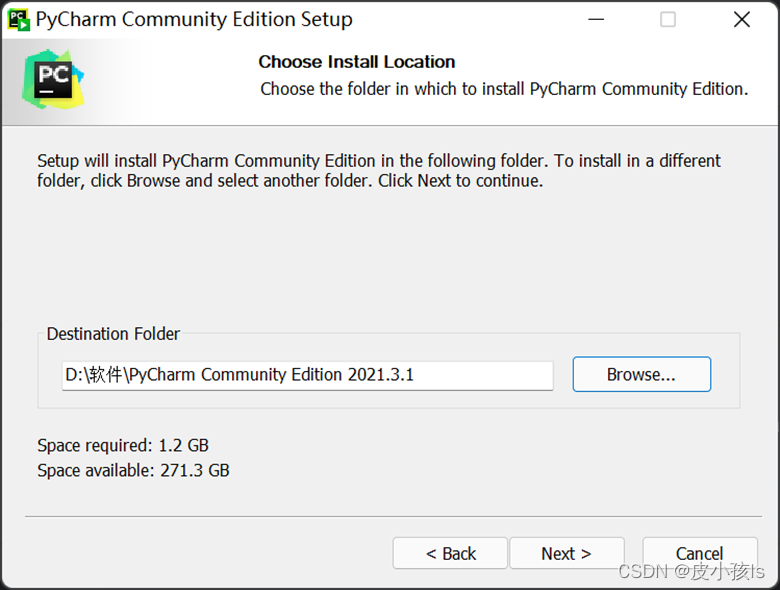
Find the folder where you downloaded the installation package and click Installer. You will see this page



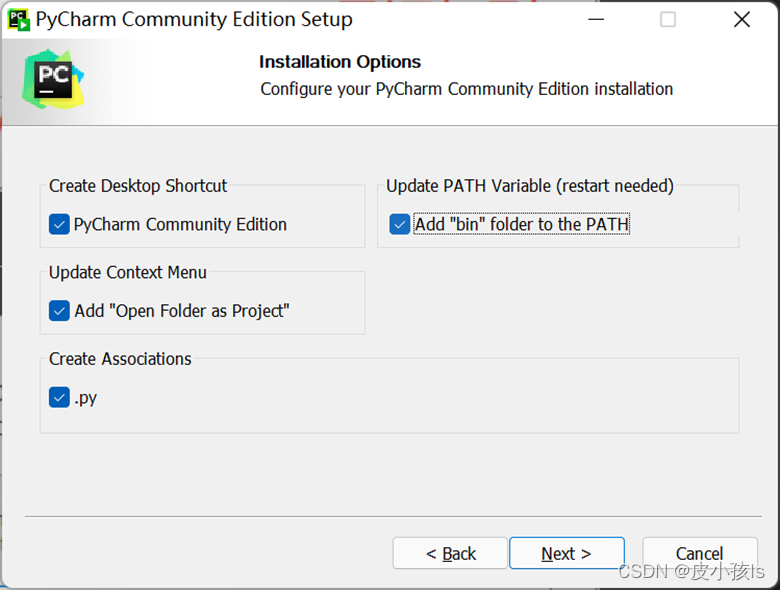
Click the Next.



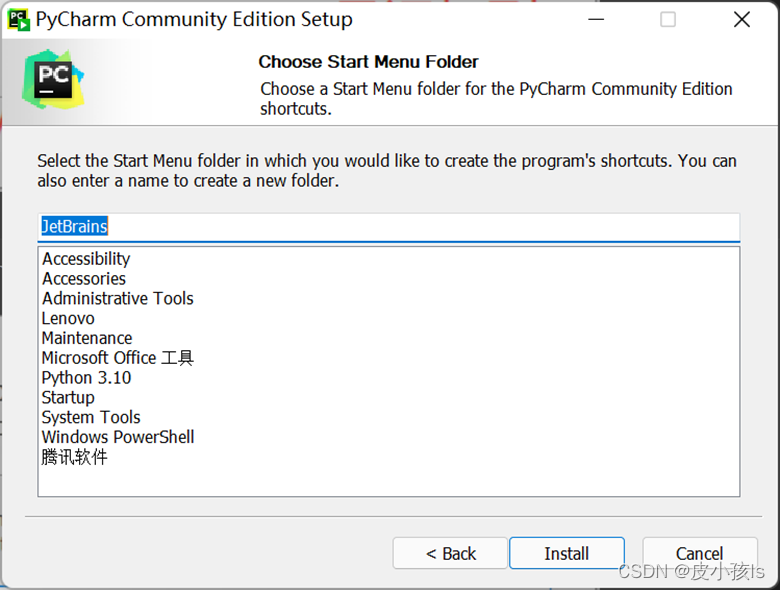
Select one folder to install the Pycharm.

[](https://github.com/WhynotChen0105/Experiment-1/blob/main/src/image-9.png)

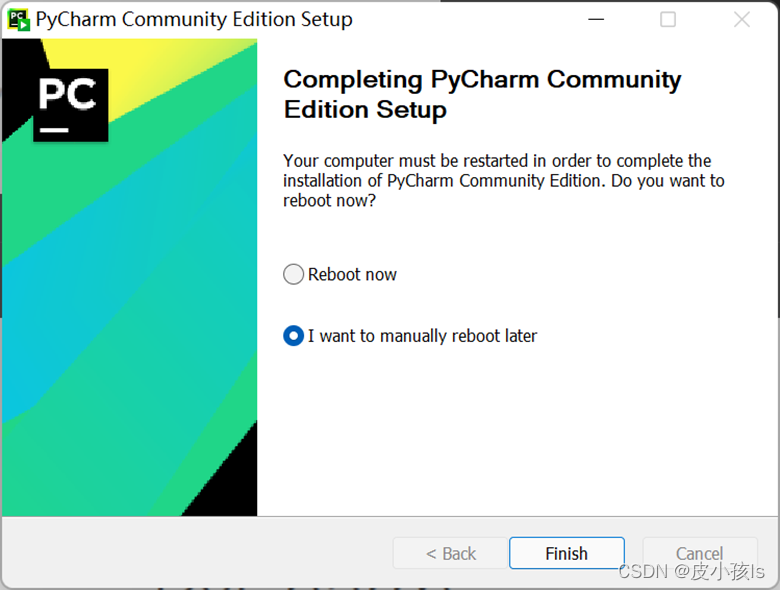
Chosse all and click the Next.

[](https://github.com/WhynotChen0105/Experiment-1/blob/main/src/image-10.png)

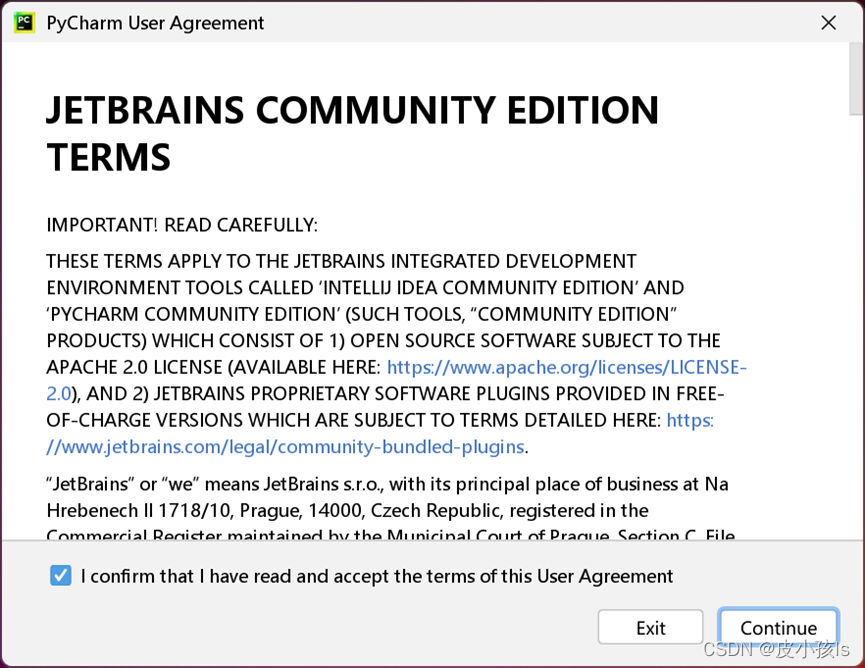
Click the Install.

[](https://github.com/WhynotChen0105/Experiment-1/blob/main/src/image-11.png)

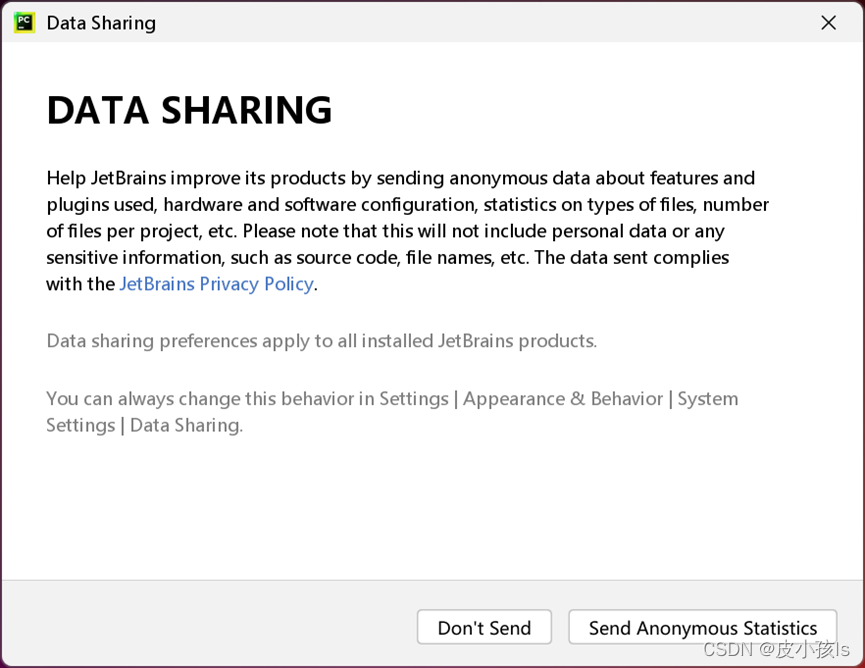
Choose "I want to manually reboot later" and click the Finish.

[](https://github.com/WhynotChen0105/Experiment-1/blob/main/src/image-12.png)

After installing, open the Pycharm. Click "I confirm that I have read and accept the terms of this User Agreement" and Continue.

[](https://github.com/WhynotChen0105/Experiment-1/blob/main/src/image-13.png)

Select the "Don't Send".

[](https://github.com/WhynotChen0105/Experiment-1/blob/main/src/image-14.png)

Download this github then click the "Open" and select the folder, or you can create a new project and code.

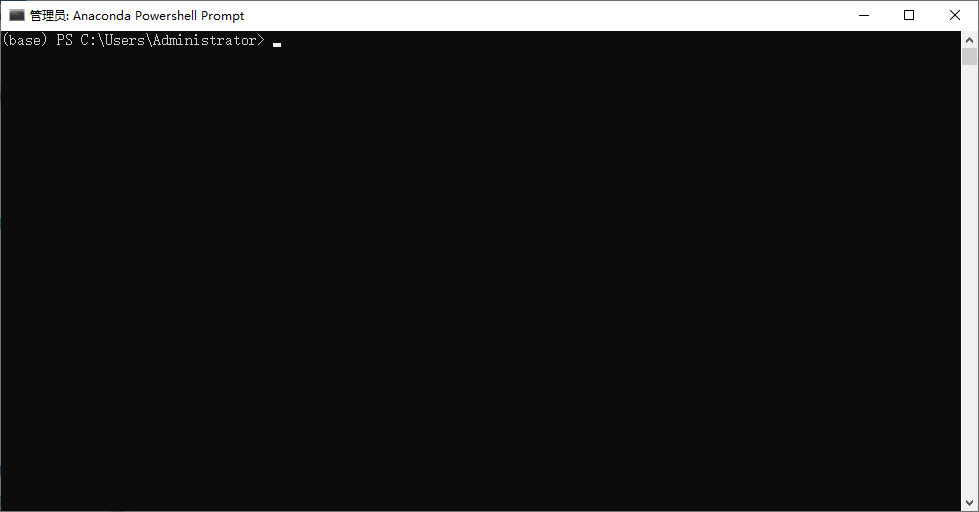
[](https://github.com/WhynotChen0105/Experiment-1/blob/main/src/image-15.png)

### Create a virtual environment

Find the Anaconda Powershell Prompt from the "Win" which on the left-botttm of desktop or you can directly search the Anaconda Powershell Prompt by pressing "Win+S":

[](https://github.com/WhynotChen0105/Experiment-1/blob/main/src/image16.png)

Open the Anaconda Powershell Prompt:

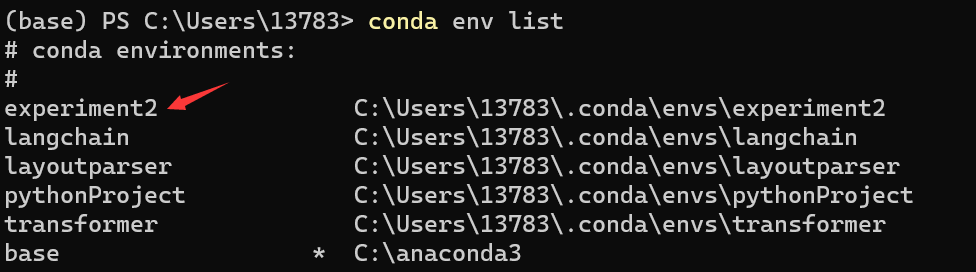
[](https://github.com/WhynotChen0105/Experiment-1/blob/main/src/image-17.png)

Input the command following to create a virtual environment named "experiment2" with python 3.10 (don't use the VPN):

**conda create -n experiment2 python=3.10 -y**

Use this command to check whether your environment is successfully installed.

**conda env list**

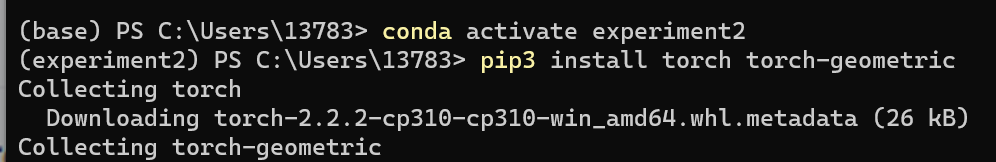


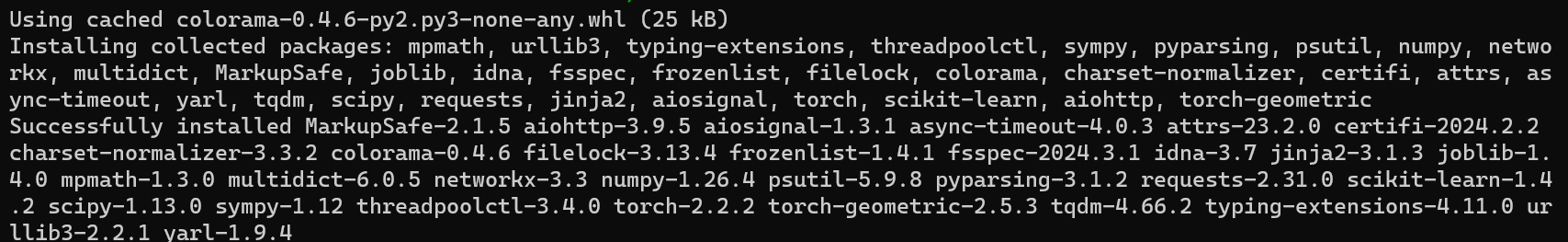
Activate the virtual environment and install the torch, torch-geometric:

**conda activate experiment2**

**pip3 install torch**

**pip3 install** **torch-geometric**



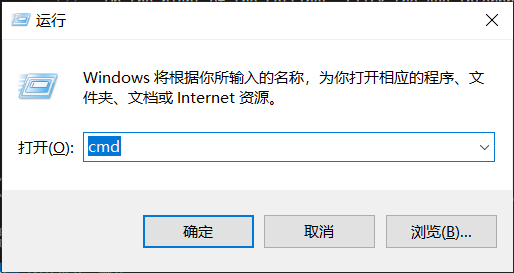
When you see the prompt "Installing collected packages:" in the following picture, it means that your python package has been installed successfully. ****

Notice !!!In this experiment we use the CPU-Version torch, but if your computer have the Nvidia GPU you can install the GPU-Version torch by replacing the pip3 insatll torch torchvison with the GPU-Version command. Next we will teach you how to find the GPU-Version install command for your computer.

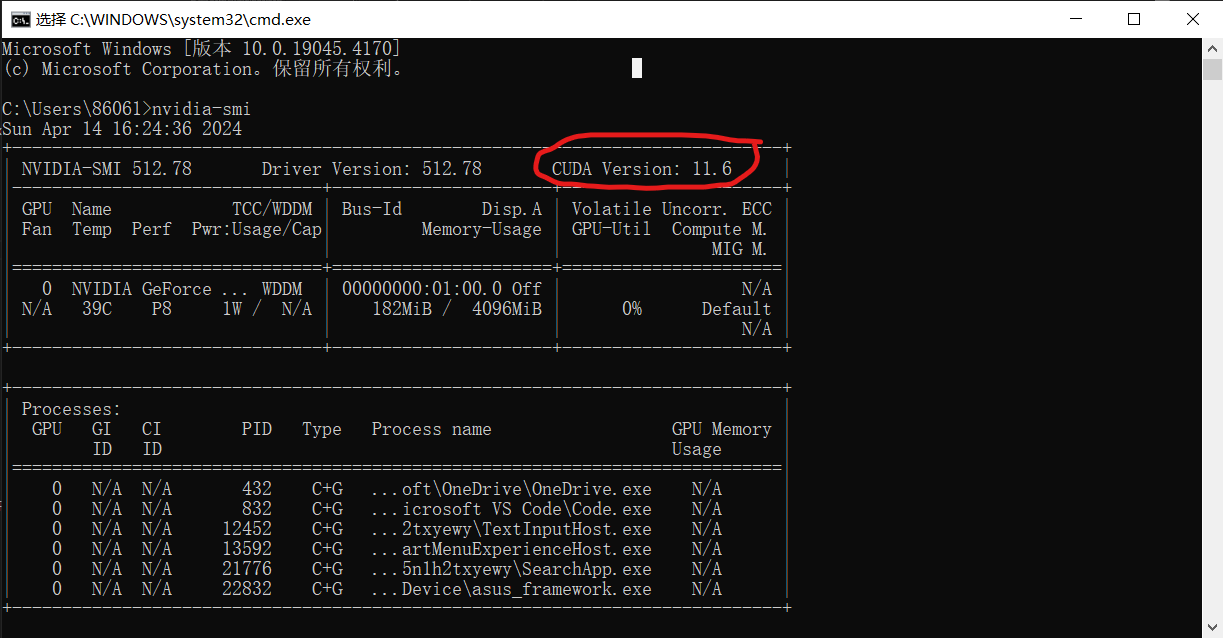
**Find the GPU-Version torch install command for your computer**

Find your CUDA version:

Press the 'Win+R', then input 'cmd' and press the 'Enter' to open the cmd:

[](https://github.com/WhynotChen0105/Experiment-1/blob/main/src/image-30.png)

Input 'nvidia-smi' in cmd, you can see you cuda version as following:

[](https://github.com/WhynotChen0105/Experiment-1/blob/main/src/image-31.png)

Please remember the cuda version. We will use it to install the torch-GPU now.

1 Open the [pytorch](https://pytorch.org/get-started/previous-versions/)'s official website : <https://pytorch.org/get-started/previous-versions/> (maybe you need the VPN).

2 Find the first install command whose cuda version is same to your computer. For example, if your cuda version is 11.6, you can find the command: (Notice: It must be the 'Wheel' not the 'Conda', and 'Linux and Windows' not the 'OSX' ! )

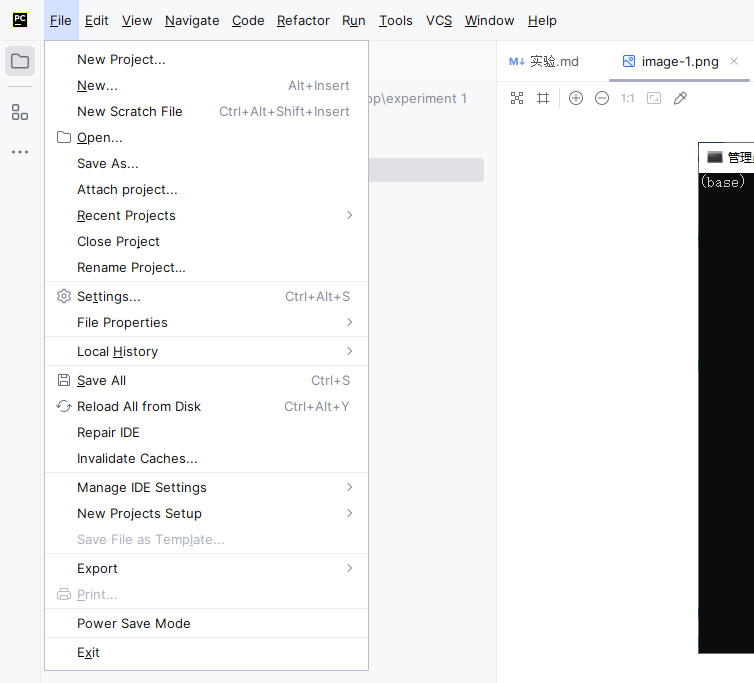
[](https://github.com/WhynotChen0105/Experiment-1/blob/main/src/image-32.png)

3 Copy the Command (in our example it is 'conda install pytorch==1.13.1 torchvision==0.14.1 torchaudio==0.13.1 pytorch-cuda=11.6 -c pytorch -c nvidia')

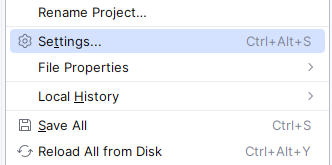
4 Now you have find the command you need.

## Select the virtual environment as the main environment of Pycharm

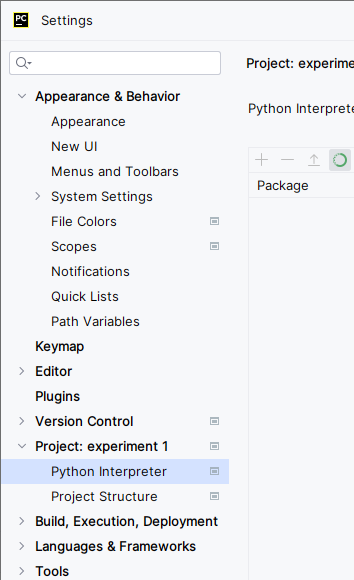
Click the left-top "Main-Menu" or use 'Alt+\'

[](https://github.com/WhynotChen0105/Experiment-1/blob/main/src/image-20.png)

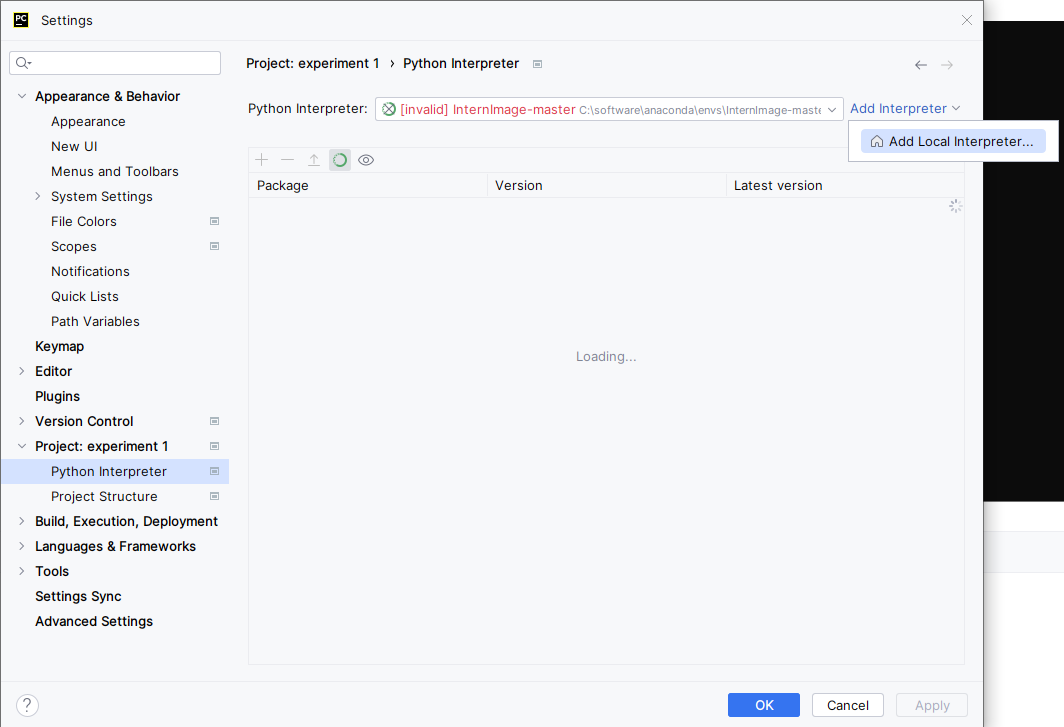
Click the "Settings".

[](https://github.com/WhynotChen0105/Experiment-1/blob/main/src/image-21.png)

In the Settings, we click the "Project: experiment 1" then click the "Python Interpreter".

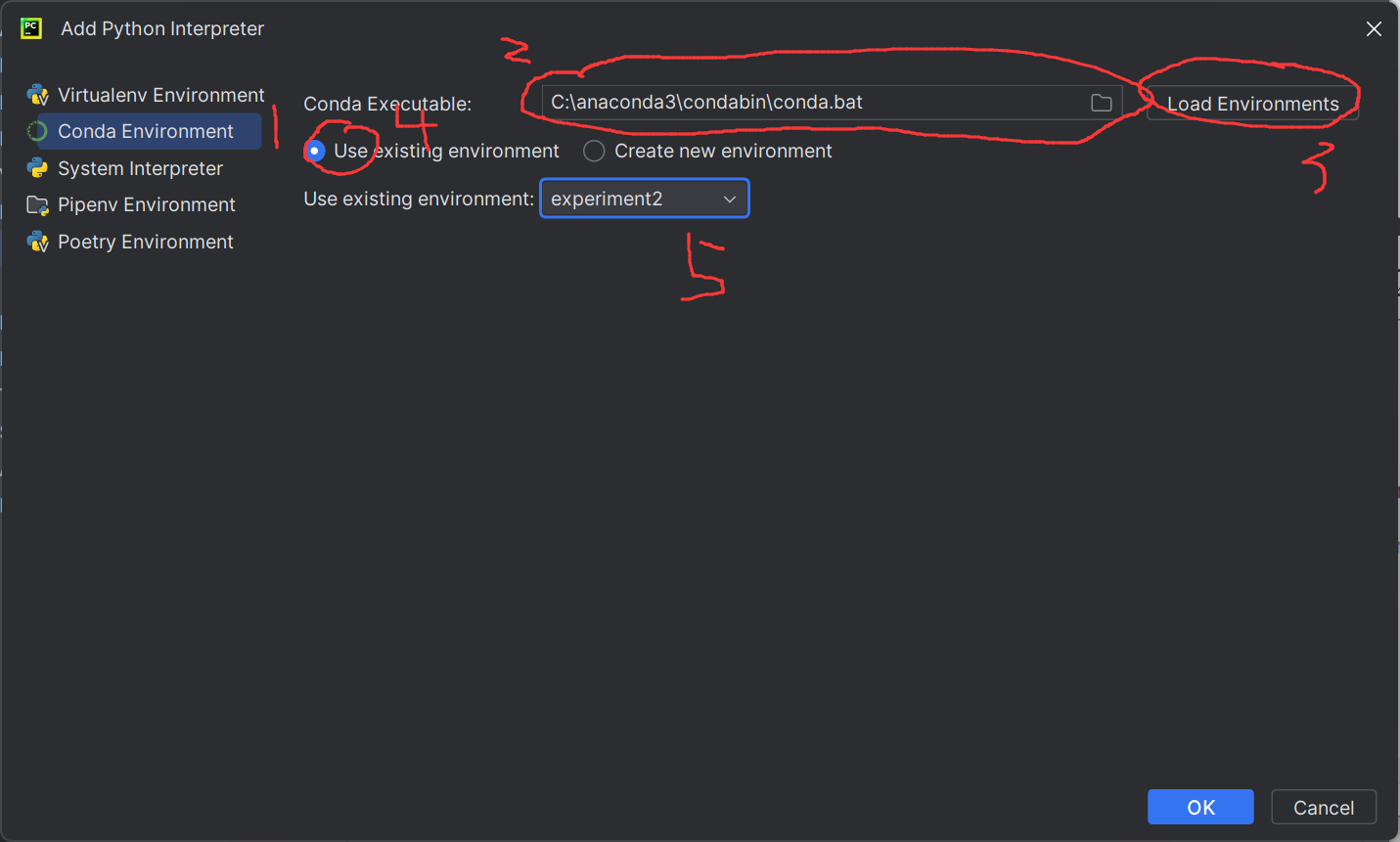
[](https://github.com/WhynotChen0105/Experiment-1/blob/main/src/image-22.png)

On the right of the settings, click the Add Interpreter then click the "Add Local Interpreter".

[](https://github.com/WhynotChen0105/Experiment-1/blob/main/src/image-23.png)

Choose the Conda Environment on the left, find your Conda Executable :Find your anaconda [installation folder](#安装位置) and select the condabin subfolder in that file and then select the file conda.bat)

Click Load Environments; choose the "Use existing environment", then choose the virtual environment "experiment2".

 Now you have completed the environment configuration.

### Dataset

This Time we choose Core as our train dataset. Cora is a graph-structured data set about citation relationships between scientific documents. The data set contains a graph, which includes 2708 documents (nodes) and 10556 citation relationships (edges). Each node has a 1433-dimensional feature vector, which is the embedding vector of the document content. The literature is divided into seven categories: Computer Science, Physics, etc.You can find more details in this page [The Cora dataset - Graph Data Science Consulting (graphsandnetworks.com)](https://graphsandnetworks.com/the-cora-dataset/)

Notice!!!! all code list here you don’t need to copy,you can find them in the file attached

We use Planetoid to access this dataset,it will wait a little time to download the data when you firstly run this code

1.  from torch\_geometric.datasets import Planetoid

2.  cora = Planetoid(root='./data', name='Cora')[0]

3.  print(cora)

## Build the Graph Convolution Nerual Network

We use the GCNConv package in torch to build our GCN network.

import torch

from torch import nn

from torch\_geometric.nn import GCNConv

import torch.nn.functional as F

from torch.optim import Adam

class GCN(nn.Module):

  def \_\_init\_\_(self, in\_channels, hidden\_channels, class\_n):

    super(GCN, self).\_\_init\_\_()

    self.conv1 = GCNConv(in\_channels, hidden\_channels)

    self.conv2 = GCNConv(hidden\_channels, class\_n)

  def forward(self, x, edge\_index):

    x = torch.relu(self.conv1(x, edge\_index))

    x = torch.dropout(x, p=0.5, train=self.training)

    x = self.conv2(x, edge\_index)

    return torch.log\_softmax(x, dim=1)

### Build the model and optimizer

model = GCN(cora.num\_features, 16, cora.y.unique().shape[0])

opt = Adam(model.parameters(), 0.01, weight\_decay=5e-4)

**Notice if you use gpu to run the code you need to change the first line code to this**

model = GCN(cora.num\_features, 16, cora.y.unique().shape[0]).to('cuda')

### Code the train function

def train(its):

  model.train()

  for i in range(its):

    y = model(cora.x, cora.edge\_index)

    loss = F.nll\_loss(y[cora.train\_mask], cora.y[cora.train\_mask])

    loss.backward()

    opt.step()

    opt.zero\_grad()

### Code the test function

def test():

  model.eval()

  y = model(cora.x, cora.edge\_index)

  right\_n = torch.argmax(y[cora.test\_mask], 1) == cora.y[cora.test\_mask]

  acc = right\_n.sum()/cora.test\_mask.sum()

  print("Acc: ", acc)

### Code the main function

def main():

  for i in range(10):

    train(1)

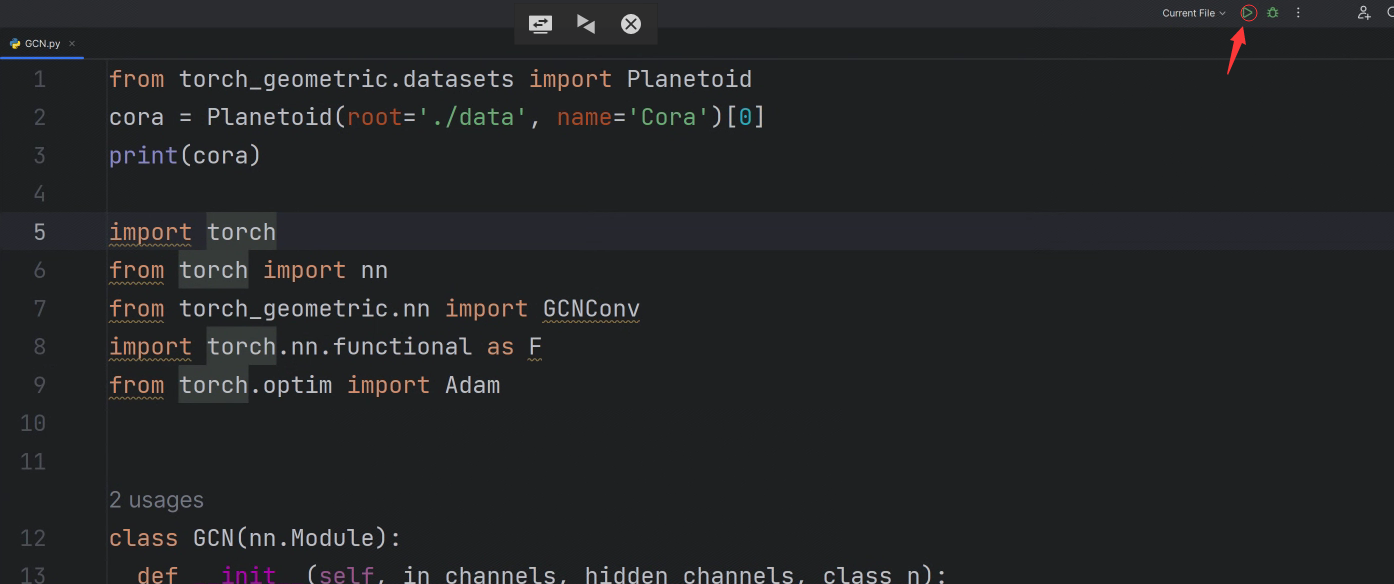
    test()

if \_\_name\_\_ == '\_\_main\_\_':

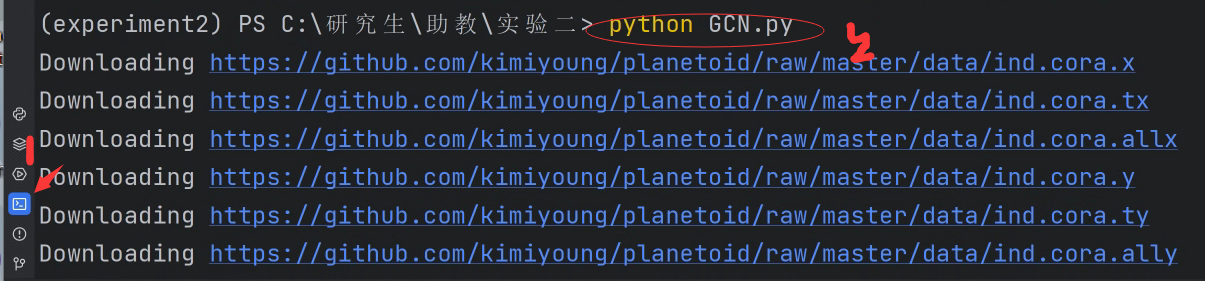
    main()

Now we have completed the all codes, now we need to run the code.

**Run the code**

Open the GCN.py in Pycharm, then you can run the code by clicking the "Run" button on the top , or you can run the code in the terminal by inputing following command and press the "Enter" :

python GCN.py



## Homework Grading

* ‌**Complete 60% of baseline operations;**
* **Extra 40% bonus** 
  + **Other algorithm improvements for accuracy enhancement.**
  + **New datasets and applications.**