如何从零复现论文代码

1 找文献

2 找代码

3 配环境（难）

4 调试代码（难）

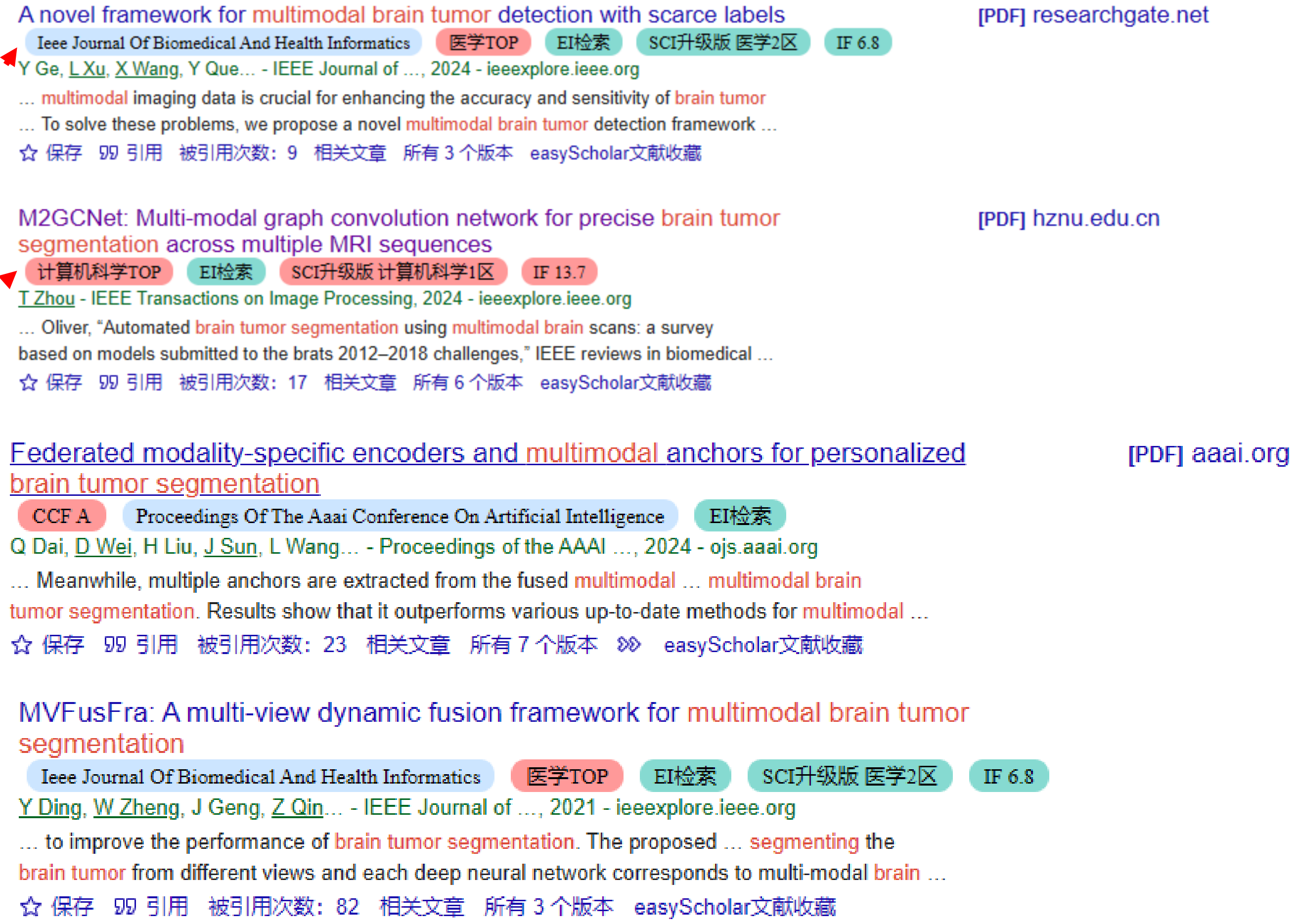
5 服务器训练

找文献

- 找SCI 2区或CCF B以上的论文（顶会、顶刊）
- 找公布源代码的论文（会议较多，期刊较少）

easyScholar
一款助力科研的浏览器扩展！

目前已经有10W+用户，
正在使用easyScholar来提高阅读文献的效率！



A novel framework for multimodal brain tumor detection with scarce labels — [PDF] researchgate.net
Ieee Journal Of Biomedical And Health Informatics | 医学TOP | EI检索 | SCI升级版 医学2区 | IF 6.8
Y Ge, L Xu, X Wang, Y Que... - IEEE Journal of ..., 2024 - ieeexplore.ieee.org
... multimodal imaging data is crucial for enhancing the accuracy and sensitivity of brain tumor ... To solve these problems, we propose a novel multimodal brain tumor detection framework ...
☆ 保存 99 引用 被引用次数：9 相关文章 所有 3 个版本 easyScholar文献收藏

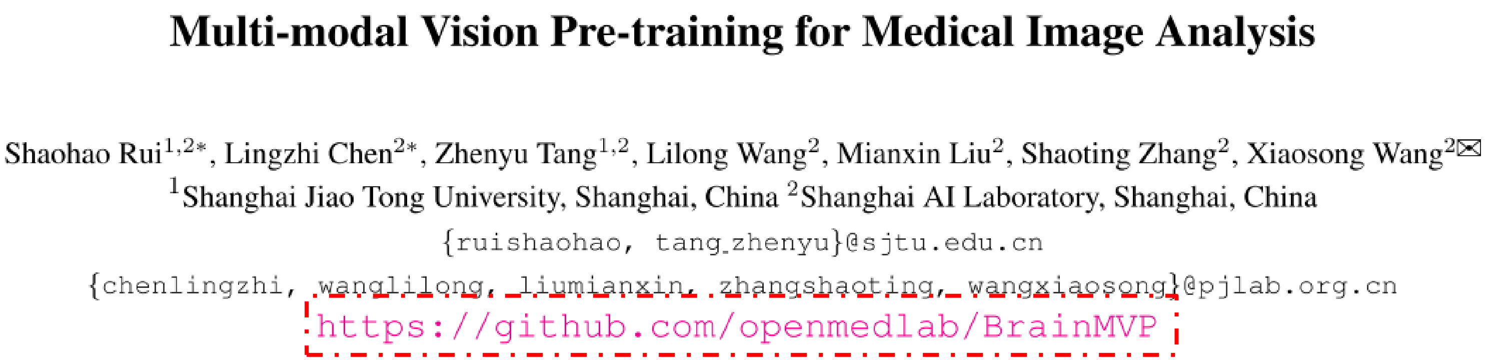
M2GCNet: Multi-modal graph convolution network for precise brain tumor segmentation across multiple MRI sequences — [PDF] hznu.edu.cn
计算机科学TOP | EI检索 | SCI升级版 计算机科学1区 | IF 13.7
T Zhou - IEEE Transactions on Image Processing, 2024 - ieeexplore.ieee.org
... Oliver, “Automated brain tumor segmentation using multimodal brain scans: a survey based on models submitted to the brats 2012–2018 challenges,” IEEE reviews in biomedical ...
☆ 保存 99 引用 被引用次数：17 相关文章 所有 6 个版本 easyScholar文献收藏

Federated modality-specific encoders and multimodal anchors for personalized brain tumor segmentation — [PDF] aaai.org
CCF A | Proceedings Of The Aaai Conference On Artificial Intelligence | EI检索
Q Dai, D Wei, H Liu, J Sun, L Wang... - Proceedings of the AAAI ..., 2024 - ojs.aaai.org
... Meanwhile, multiple anchors are extracted from the fused multimodal ... multimodal brain tumor segmentation. Results show that it outperforms various up-to-date methods for multimodal ...
☆ 保存 99 引用 被引用次数：23 相关文章 所有 7 个版本 ≫ easyScholar文献收藏

MVFusFra: A multi-view dynamic fusion framework for multimodal brain tumor segmentation
Ieee Journal Of Biomedical And Health Informatics | 医学TOP | EI检索 | SCI升级版 医学2区 | IF 6.8
Y Ding, W Zheng, J Geng, Z Qin... - IEEE Journal of ..., 2021 - ieeexplore.ieee.org
... to improve the performance of brain tumor segmentation. The proposed ... segmenting the brain tumor from different views and each deep neural network corresponds to multi-modal brain ...
☆ 保存 99 引用 被引用次数：82 相关文章 所有 3 个版本 easyScholar文献收藏

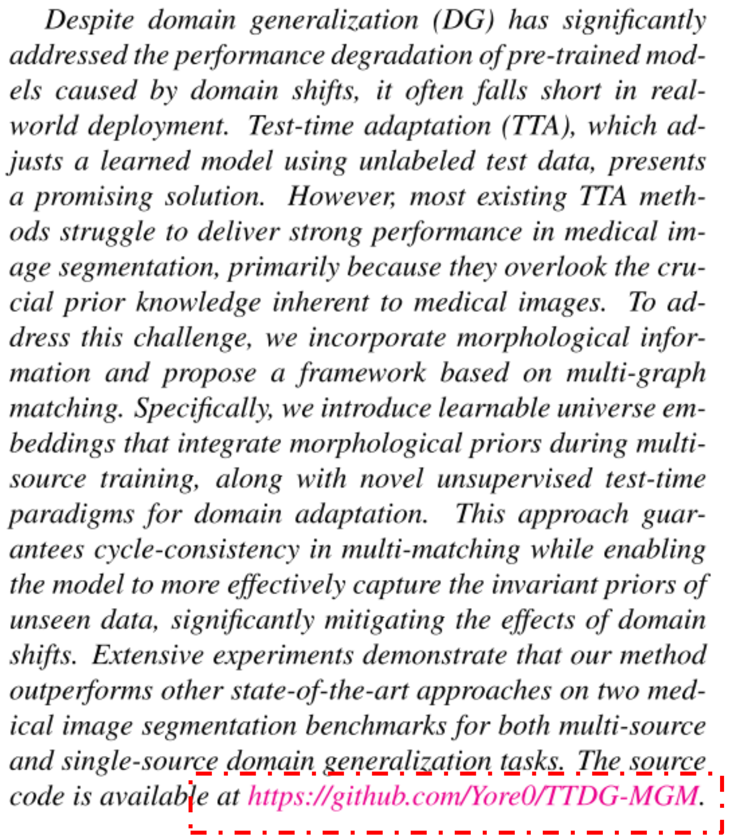
找代码

➢ **文献中直接给出（摘要或实验部分）**

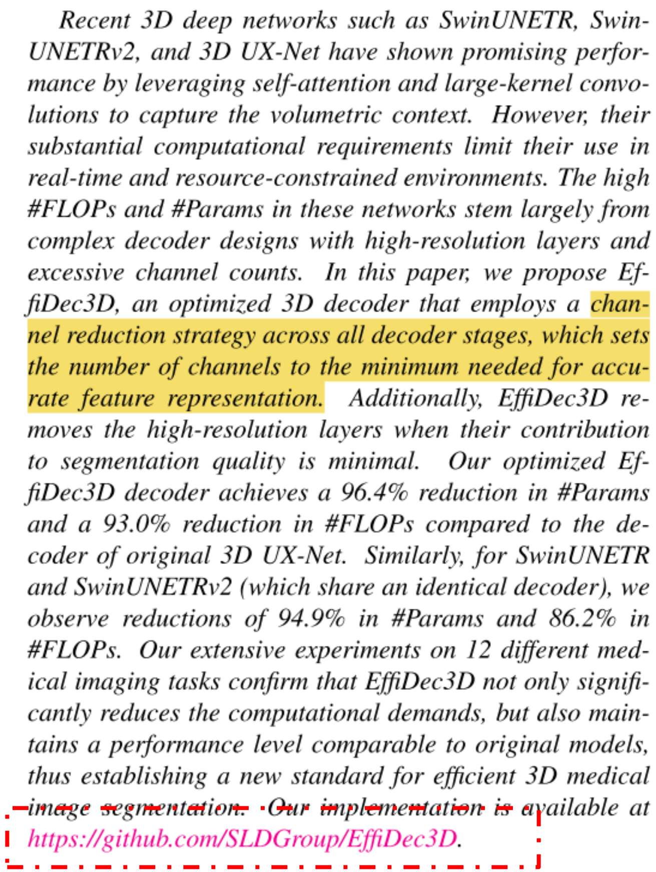


Multi-modal Vision Pre-training for Medical Image Analysis

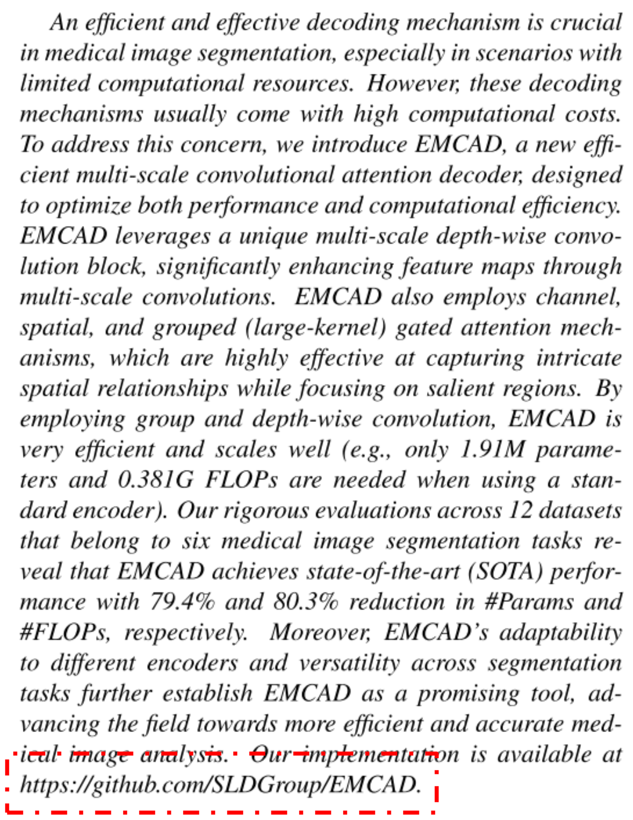
Shaohao Rui[1,2*], Lingzhi Chen[2*], Zhenyu Tang[1,2], Lilong Wang[2], Mianxin Liu[2], Shaoting Zhang[2], Xiaosong Wang[2]✉
[1]Shanghai Jiao Tong University, Shanghai, China [2]Shanghai AI Laboratory, Shanghai, China
{ruishaohao, tang_zhenyu}@sjtu.edu.cn
{chenlingzhi, wanglilong, liumianxin, zhangshaoting, wangxiaosong}@pjlab.org.cn
https://github.com/openmedlab/BrainMVP



Despite domain generalization (DG) has significantly addressed the performance degradation of pre-trained models caused by domain shifts, it often falls short in real-world deployment. Test-time adaptation (TTA), which adjusts a learned model using unlabeled test data, presents a promising solution. However, most existing TTA methods struggle to deliver strong performance in medical image segmentation, primarily because they overlook the crucial prior knowledge inherent to medical images. To address this challenge, we incorporate morphological information and propose a framework based on multi-graph matching. Specifically, we introduce learnable universe embeddings that integrate morphological priors during multi-source training, along with novel unsupervised test-time paradigms for domain adaptation. This approach guarantees cycle-consistency in multi-matching while enabling the model to more effectively capture the invariant priors of unseen data, significantly mitigating the effects of domain shifts. Extensive experiments demonstrate that our method outperforms other state-of-the-art approaches on two medical image segmentation benchmarks for both multi-source and single-source domain generalization tasks. The source code is available at https://github.com/Yore0/TTDG-MGM.



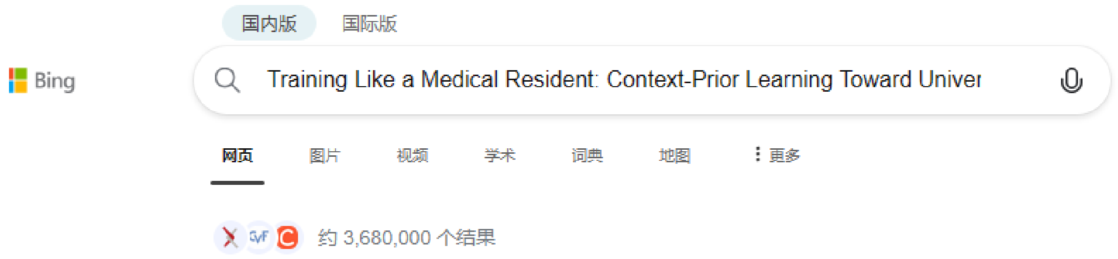
Recent 3D deep networks such as SwinUNETR, SwinUNETRv2, and 3D UX-Net have shown promising performance by leveraging self-attention and large-kernel convolutions to capture the volumetric context. However, their substantial computational requirements limit their use in real-time and resource-constrained environments. The high #FLOPs and #Params in these networks stem largely from complex decoder designs with high-resolution layers and excessive channel counts. In this paper, we propose EffiDec3D, an optimized 3D decoder that employs a channel reduction strategy across all decoder stages, which sets the number of channels to the minimum needed for accurate feature representation. Additionally, EffiDec3D removes the high-resolution layers when their contribution to segmentation quality is minimal. Our optimized EffiDec3D decoder achieves a 96.4% reduction in #Params and a 93.0% reduction in #FLOPs compared to the decoder of original 3D UX-Net. Similarly, for SwinUNETR and SwinUNETRv2 (which share an identical decoder), we observe reductions of 94.9% in #Params and 86.2% in #FLOPs. Our extensive experiments on 12 different medical imaging tasks confirm that EffiDec3D not only significantly reduces the computational demands, but also maintains a performance level comparable to original models, thus establishing a new standard for efficient 3D medical image segmentation. Our implementation is available at https://github.com/SLDGroup/EffiDec3D.



An efficient and effective decoding mechanism is crucial in medical image segmentation, especially in scenarios with limited computational resources. However, these decoding mechanisms usually come with high computational costs. To address this concern, we introduce EMCAD, a new efficient multi-scale convolutional attention decoder, designed to optimize both performance and computational efficiency. EMCAD leverages a unique multi-scale depth-wise convolution block, significantly enhancing feature maps through multi-scale convolutions. EMCAD also employs channel, spatial, and grouped (large-kernel) gated attention mechanisms, which are highly effective at capturing intricate spatial relationships while focusing on salient regions. By employing group and depth-wise convolution, EMCAD is very efficient and scales well (e.g., only 1.91M parameters and 0.381G FLOPs are needed when using a standard encoder). Our rigorous evaluations across 12 datasets that belong to six medical image segmentation tasks reveal that EMCAD achieves state-of-the-art (SOTA) performance with 79.4% and 80.3% reduction in #Params and #FLOPs, respectively. Moreover, EMCAD's adaptability to different encoders and versatility across segmentation tasks further establish EMCAD as a promising tool, advancing the field towards more efficient and accurate medical image analysis. Our implementation is available at https://github.com/SLDGroup/EMCAD.

找代码

- ➢ **文献中没有直接给出，搜索引擎检索（google、Bing等）**

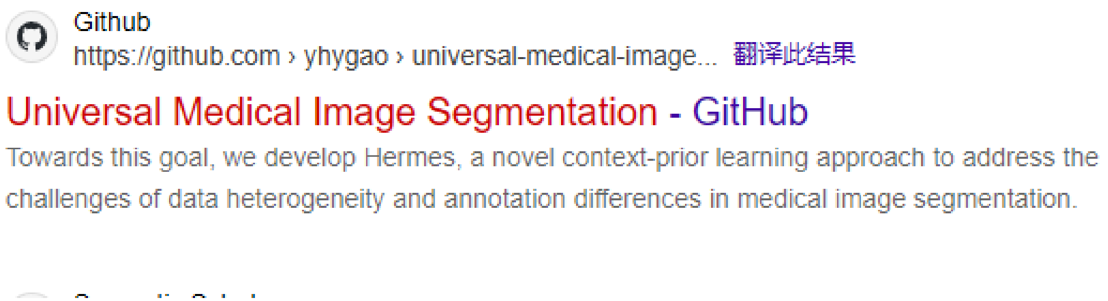


Bing

国内版 国际版

Training Like a Medical Resident: Context-Prior Learning Toward Univer

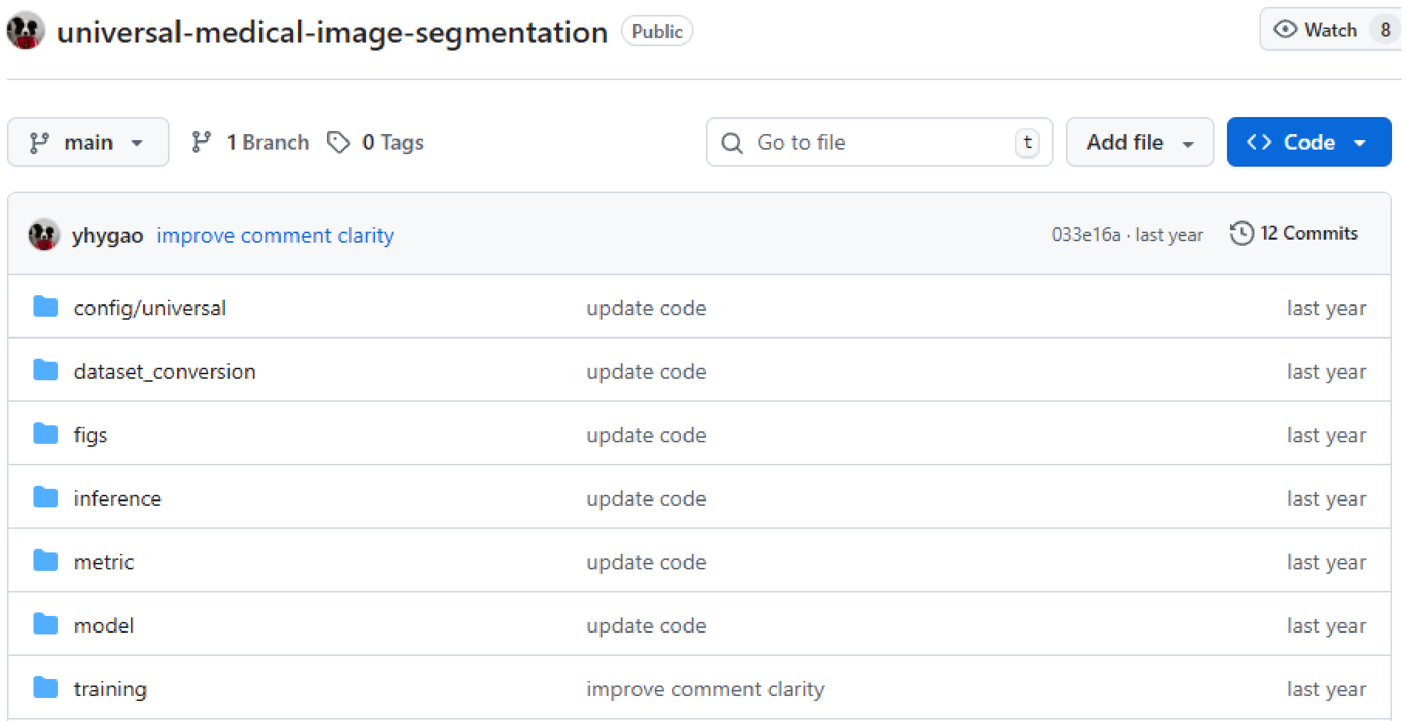
约 3,680,000 个结果



Github
https://github.com › yhygao › universal-medical-image... 翻译此结果

Universal Medical Image Segmentation - GitHub

Towards this goal, we develop Hermes, a novel context-prior learning approach to address the challenges of data heterogeneity and annotation differences in medical image segmentation.



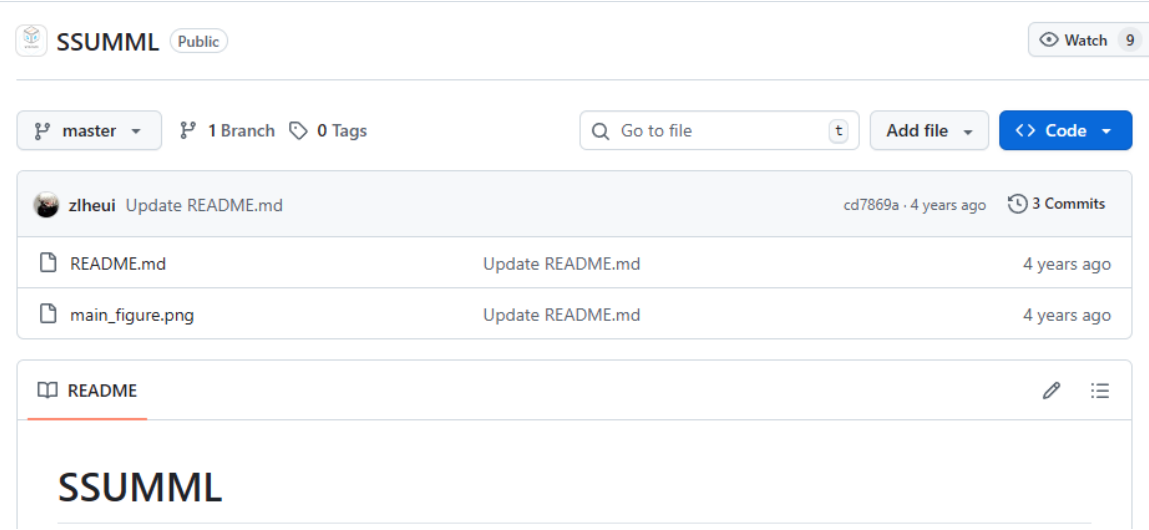
universal-medical-image-segmentation Public

yhygao improve comment clarity 033e16a · last year 12 Commits

config/universal	update code	last year
dataset_conversion	update code	last year
figs	update code	last year
inference	update code	last year
metric	update code	last year
model	update code	last year
training	improve comment clarity	last year

找代码

例外：空仓库，只有README文件

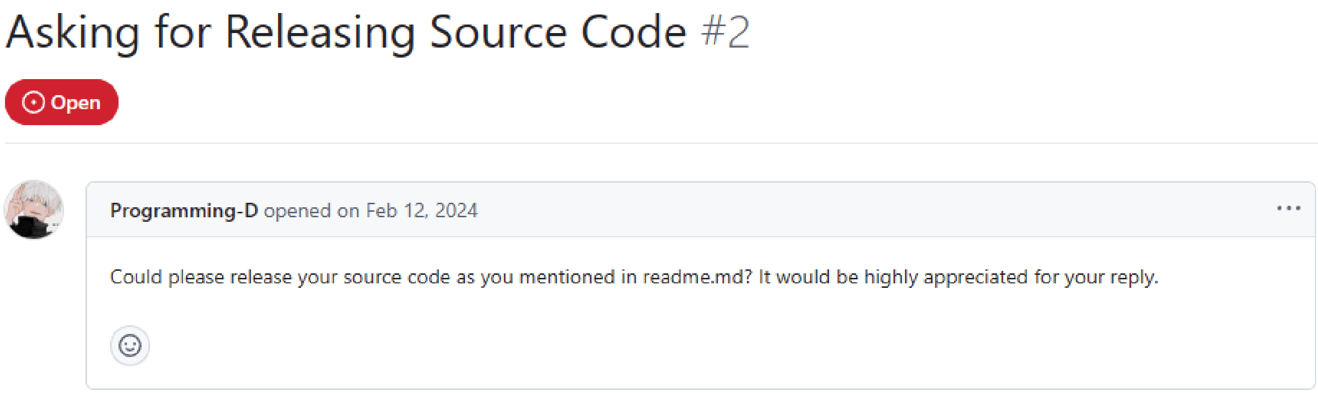


SSUMML Public

zlheui Update README.md cd7869a · 4 years ago 3 Commits

README.md	Update README.md	4 years ago
main_figure.png	Update README.md	4 years ago

SSUMML



Asking for Releasing Source Code #2

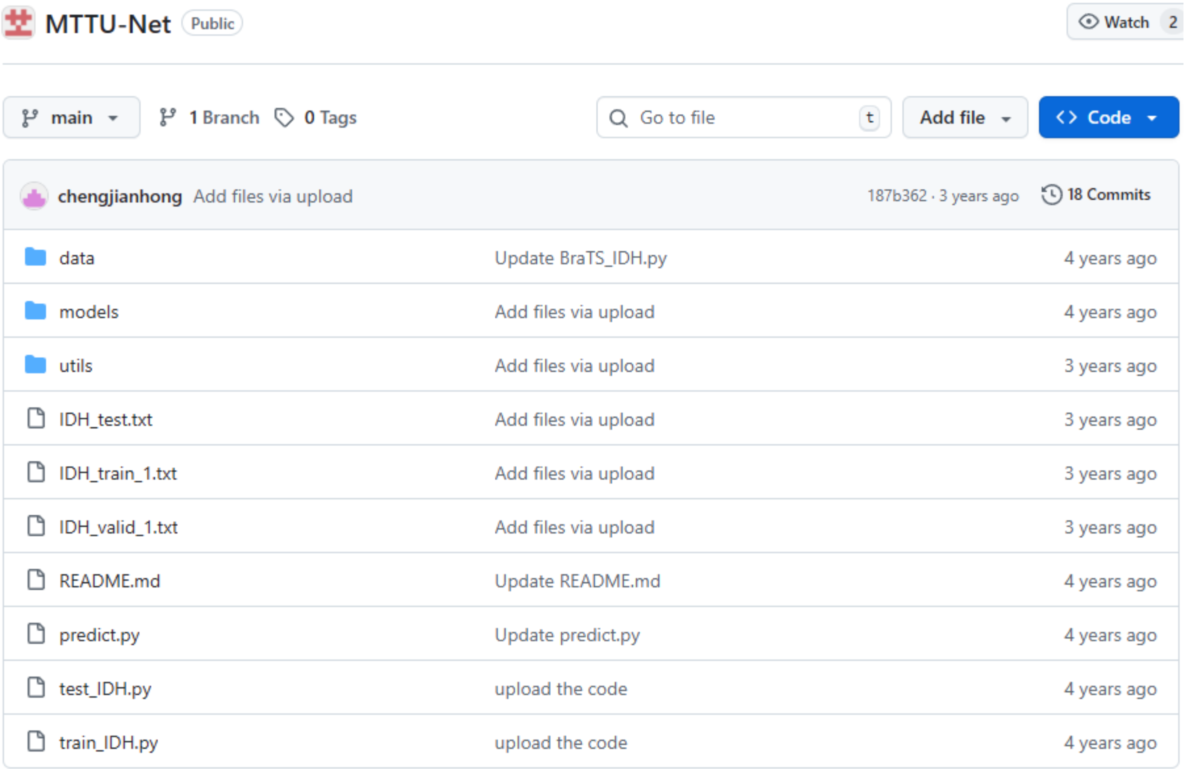
Programming-D opened on Feb 12, 2024

Could please release your source code as you mentioned in readme.md? It would be highly appreciated for your reply.

✗

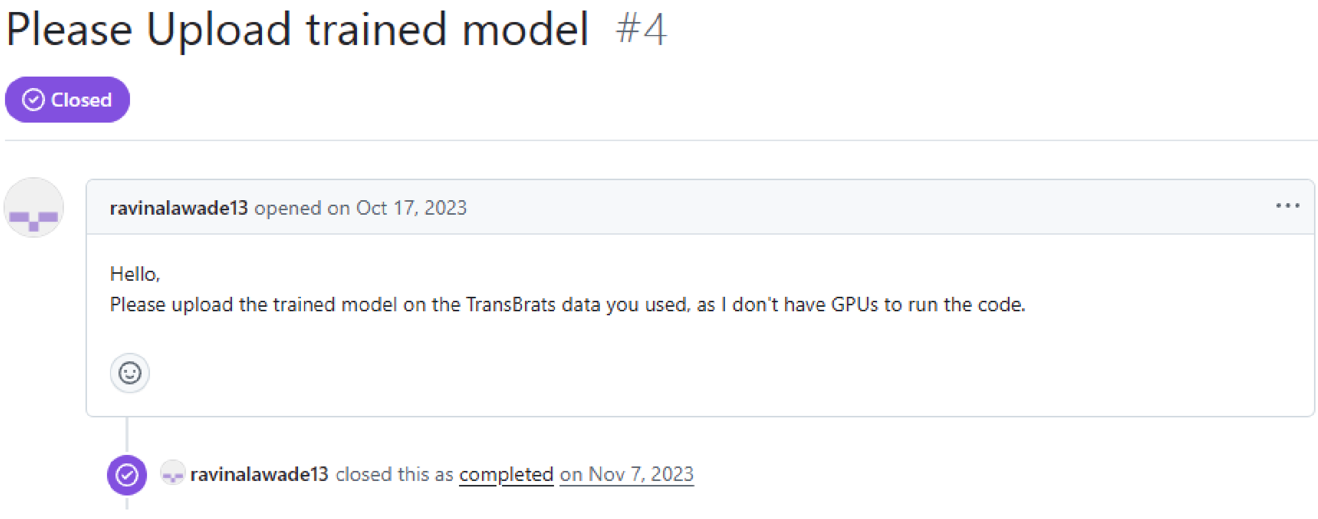
找代码

例外：提供的代码不完整



chengjianhong Add files via upload 187b362 · 3 years ago 18 Commits

data	Update BraTS_IDH.py	4 years ago
models	Add files via upload	4 years ago
utils	Add files via upload	3 years ago
IDH_test.txt	Add files via upload	3 years ago
IDH_train_1.txt	Add files via upload	3 years ago
IDH_valid_1.txt	Add files via upload	3 years ago
README.md	Update README.md	4 years ago
predict.py	Update predict.py	4 years ago
test_IDH.py	upload the code	4 years ago
train_IDH.py	upload the code	4 years ago



Please Upload trained model #4

ravinalawade13 opened on Oct 17, 2023

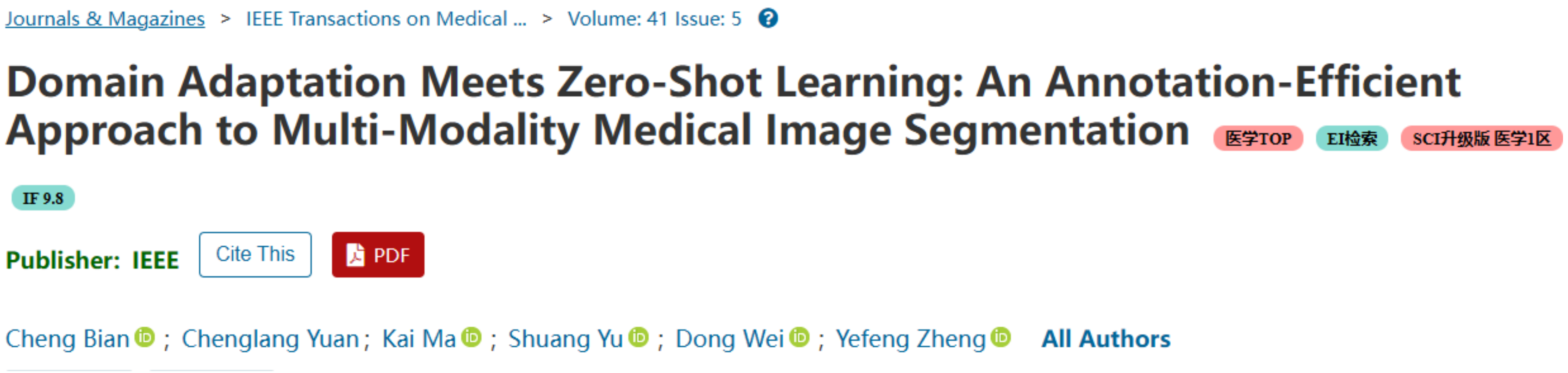
Hello,
Please upload the trained model on the TransBrats data you used, as I don't have GPUs to run the code.

ravinalawade13 closed this as completed on Nov 7, 2023

✗

找代码

➢ 发邮件询问作者，成功率比较低



Journals & Magazines > IEEE Transactions on Medical ... > Volume: 41 Issue: 5

Domain Adaptation Meets Zero-Shot Learning: An Annotation-Efficient Approach to Multi-Modality Medical Image Segmentation

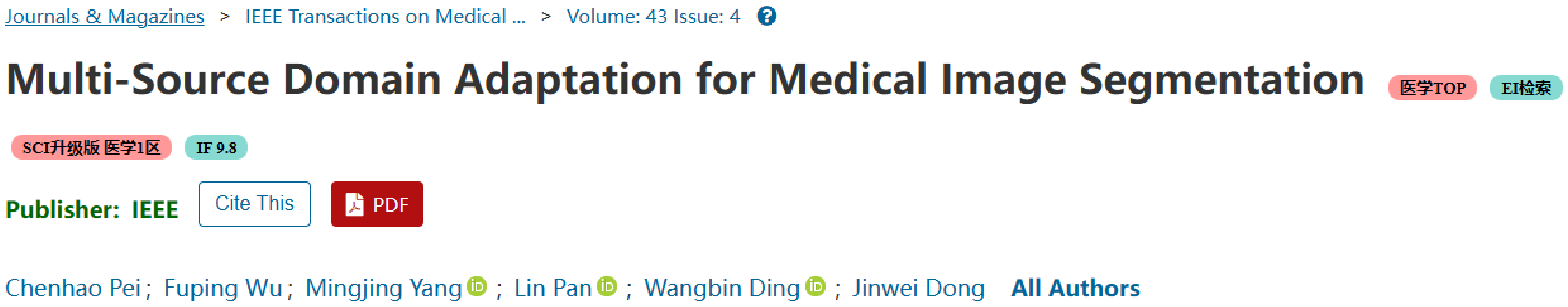
医学TOP EI检索 SCI升级版 医学1区

IF 9.8

Publisher: IEEE Cite This PDF

Cheng Bian ; Chenglang Yuan ; Kai Ma ; Shuang Yu ; Dong Wei ; Yefeng Zheng All Authors

期刊：TMI



Journals & Magazines > IEEE Transactions on Medical ... > Volume: 43 Issue: 4

Multi-Source Domain Adaptation for Medical Image Segmentation

医学TOP EI检索

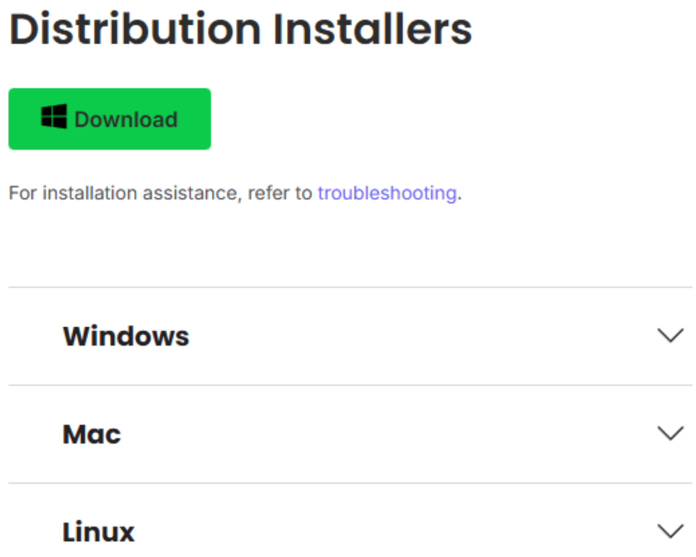
SCI升级版 医学1区 IF 9.8

Publisher: IEEE Cite This PDF

Chenhao Pei ; Fuping Wu ; Mingjing Yang ; Lin Pan ; Wangbin Ding ; Jinwei Dong All Authors

配环境（难）

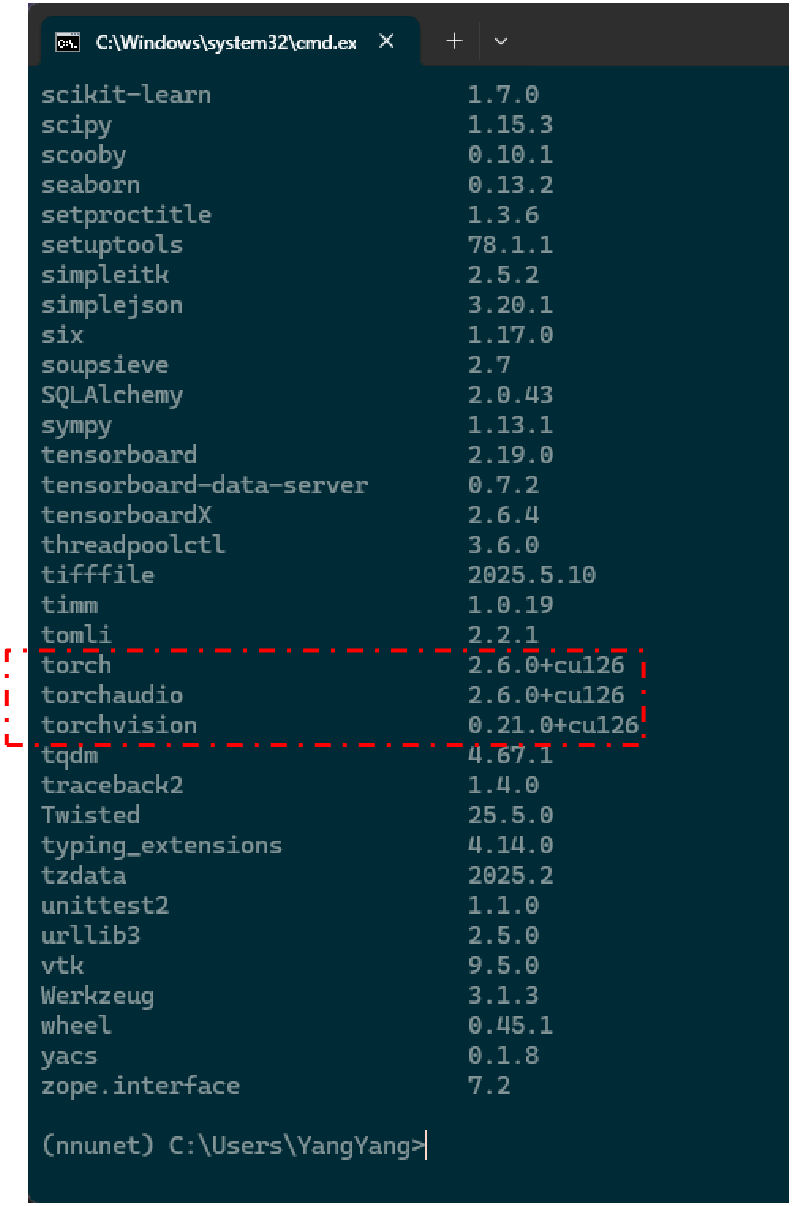
- **虚拟环境：anaconda**



Distribution Installers

For installation assistance, refer to troubleshooting.

- **基本的环境：Pytorch, opencv, numpy, cuda, scipy, matplotlib等**

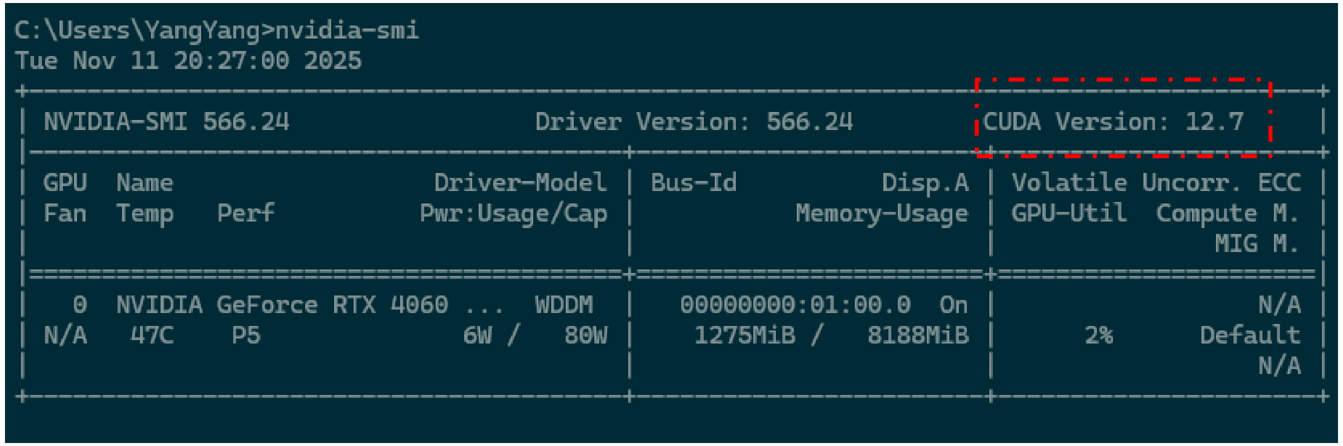


```
scikit-learn                    1.7.0
scipy                           1.15.3
scooby                          0.10.1
seaborn                         0.13.2
setproctitle                    1.3.6
setuptools                      78.1.1
simpleitk                       2.5.2
simplejson                      3.20.1
six                             1.17.0
soupsieve                       2.7
SQLAlchemy                      2.0.43
sympy                           1.13.1
tensorboard                     2.19.0
tensorboard-data-server         0.7.2
tensorboardX                    2.6.4
threadpoolctl                   3.6.0
tifffile                        2025.5.10
timm                            1.0.19
tomli                           2.2.1
torch                           2.6.0+cu126
torchaudio                      2.6.0+cu126
torchvision                     0.21.0+cu126
tqdm                            4.67.1
traceback2                      1.4.0
Twisted                         25.5.0
typing_extensions               4.14.0
tzdata                          2025.2
unittest2                       1.1.0
urllib3                         2.5.0
vtk                             9.5.0
Werkzeug                        3.1.3
wheel                           0.45.1
yacs                            0.1.8
zope.interface                  7.2

(nnunet) C:\Users\YangYang>
```

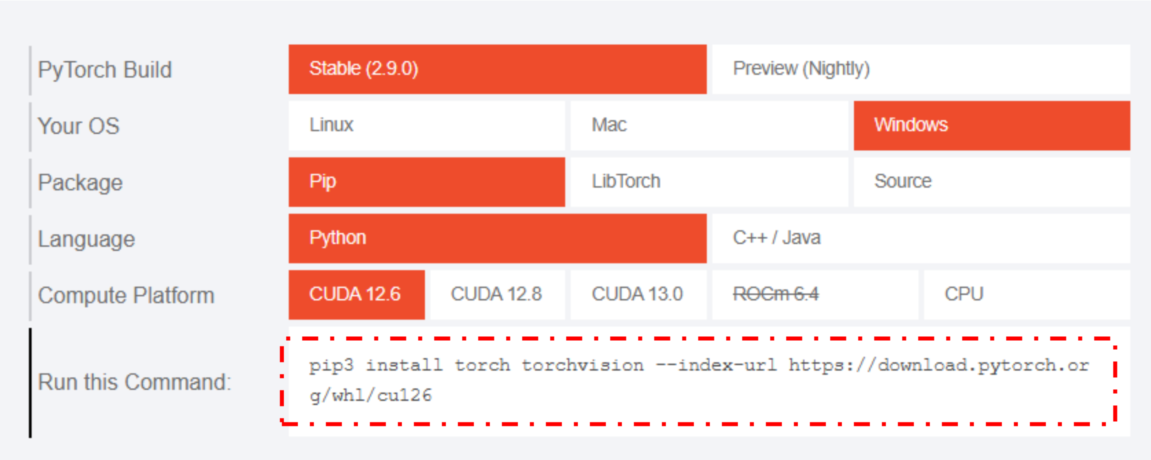
配环境（难）

➢ 配置Pytorch环境

□ 查看本机最高支持的cuda版本，例如我的显卡是4060，最高支持12.7的cuda



```
C:\Users\YangYang>nvidia-smi
Tue Nov 11 20:27:00 2025
+-----------------------------------------------------------------------------------------+
| NVIDIA-SMI 566.24                 Driver Version: 566.24         CUDA Version: 12.7     |
|-----------------------------------------+------------------------+----------------------+
| 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 RTX 4060 ...  WDDM  |   00000000:01:00.0  On |                  N/A |
| N/A   47C    P5              6W /   80W |    1275MiB /   8188MiB |      2%      Default |
|                                         |                        |                  N/A |
+-----------------------------------------+------------------------+----------------------+
```



PyTorch Build: [✓ Stable (2.9.0)] Preview (Nightly)

Your OS: Linux Mac [✓ Windows]

Package: [✓ Pip] LibTorch Source

Language: [✓ Python] C++ / Java

Compute Platform: [✓ CUDA 12.6] CUDA 12.8 CUDA 13.0 ~~ROCm 6.4~~ CPU

Run this Command:

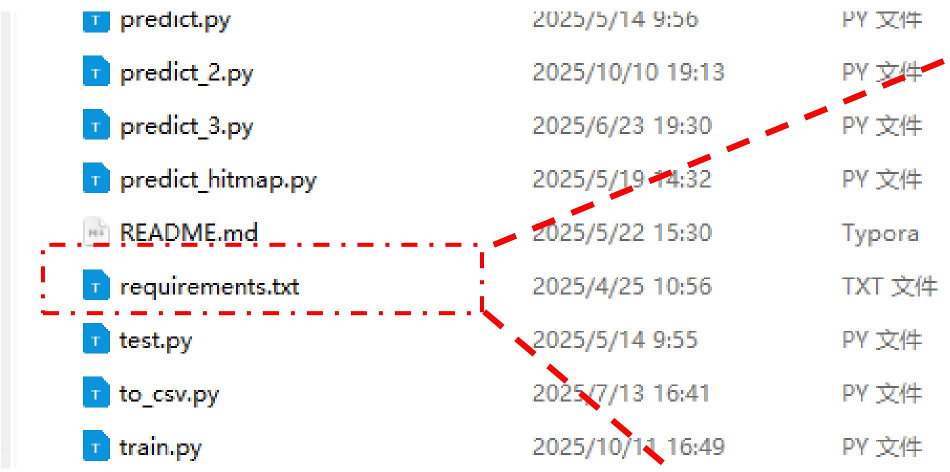
```
pip3 install torch torchvision --index-url https://download.pytorch.org/whl/cu126
```

□ 打开Pytorch官网，选择系统和cuda版本，复制命令到终端安装Pytorch

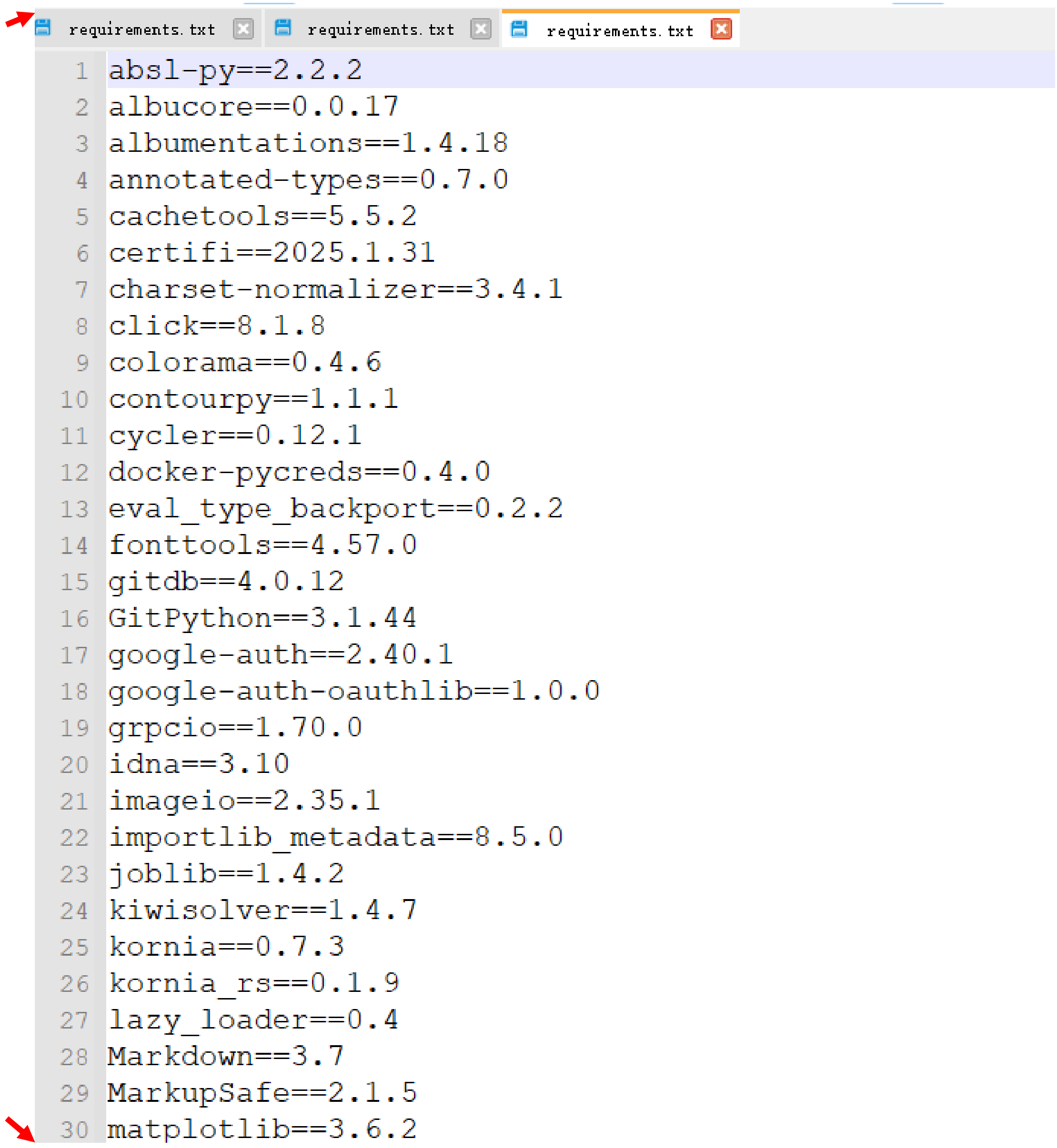
□ 直连官方的仓库，下载速度可能比较慢，可以使用阿里云或清华的conda镜像源

配环境（难）

- 配置项目环境：pip install –r requirements.txt



predict.py	2025/5/14 9:56	PY 文件
predict_2.py	2025/10/10 19:13	PY 文件
predict_3.py	2025/6/23 19:30	PY 文件
predict_hitmap.py	2025/5/19 14:32	PY 文件
README.md	2025/5/22 15:30	Typora
requirements.txt	2025/4/25 10:56	TXT 文件
test.py	2025/5/14 9:55	PY 文件
to_csv.py	2025/7/13 16:41	PY 文件
train.py	2025/10/11 16:49	PY 文件



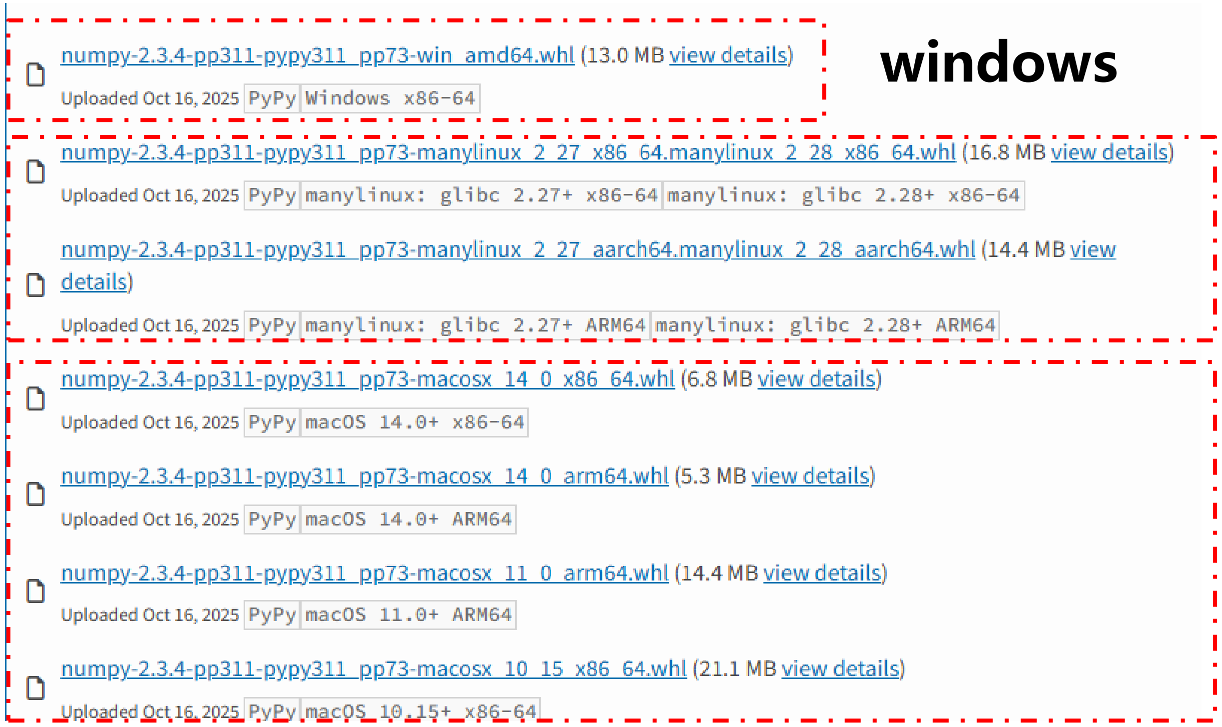
```
absl-py==2.2.2
albucore==0.0.17
albumentations==1.4.18
annotated-types==0.7.0
cachetools==5.5.2
certifi==2025.1.31
charset-normalizer==3.4.1
click==8.1.8
colorama==0.4.6
contourpy==1.1.1
cycler==0.12.1
docker-pycreds==0.4.0
eval_type_backport==0.2.2
fonttools==4.57.0
gitdb==4.0.12
GitPython==3.1.44
google-auth==2.40.1
google-auth-oauthlib==1.0.0
grpcio==1.70.0
idna==3.10
imageio==2.35.1
importlib_metadata==8.5.0
joblib==1.4.2
kiwisolver==1.4.7
kornia==0.7.3
kornia_rs==0.1.9
lazy_loader==0.4
Markdown==3.7
MarkupSafe==2.1.5
matplotlib==3.6.2
```

□ 遇到某一个包安装报错，可以先单独安装这个包，再执行上述命令

□ 如果单独安装还是报错，在https://pypi.org上下载离线的包

配环境（难）

- 离线安装



numpy-2.3.4-pp311-pypy311_pp73-win_amd64.whl (13.0 MB view details)
Uploaded Oct 16, 2025 PyPy Windows x86-64

windows

numpy-2.3.4-pp311-pypy311_pp73-manylinux_2_27_x86_64.manylinux_2_28_x86_64.whl (16.8 MB view details)
Uploaded Oct 16, 2025 PyPy manylinux: glibc 2.27+ x86-64 manylinux: glibc 2.28+ x86-64

numpy-2.3.4-pp311-pypy311_pp73-manylinux_2_27_aarch64.manylinux_2_28_aarch64.whl (14.4 MB view details)
Uploaded Oct 16, 2025 PyPy manylinux: glibc 2.27+ ARM64 manylinux: glibc 2.28+ ARM64

Linux

numpy-2.3.4-pp311-pypy311_pp73-macosx_14_0_x86_64.whl (6.8 MB view details)
Uploaded Oct 16, 2025 PyPy macOS 14.0+ x86-64

numpy-2.3.4-pp311-pypy311_pp73-macosx_14_0_arm64.whl (5.3 MB view details)
Uploaded Oct 16, 2025 PyPy macOS 14.0+ ARM64

numpy-2.3.4-pp311-pypy311_pp73-macosx_11_0_arm64.whl (14.4 MB view details)
Uploaded Oct 16, 2025 PyPy macOS 11.0+ ARM64

numpy-2.3.4-pp311-pypy311_pp73-macosx_10_15_x86_64.whl (21.1 MB view details)
Uploaded Oct 16, 2025 PyPy macOS 10.15+ x86-64

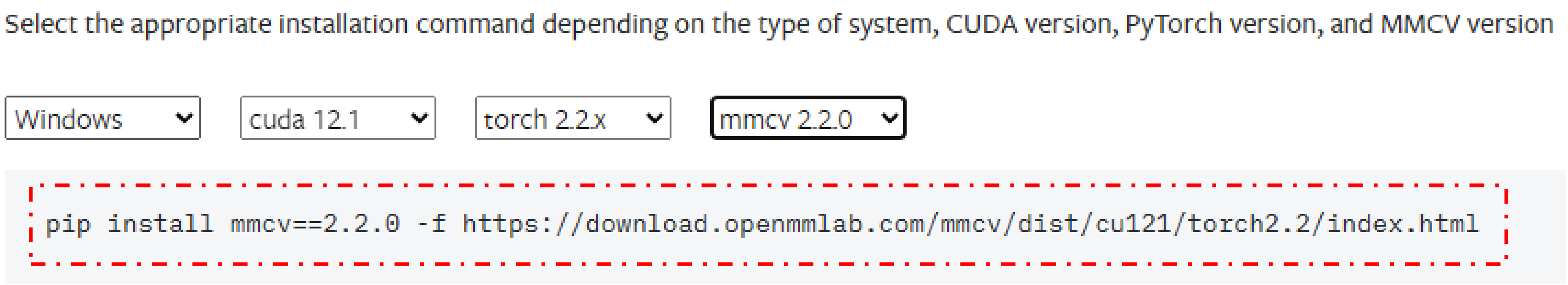
macos

```
pip install numpy-2.3.4-pp311-pypy311_pp73-win_amd64.whl
```

配环境（难）

➢ 其他的疑难杂症

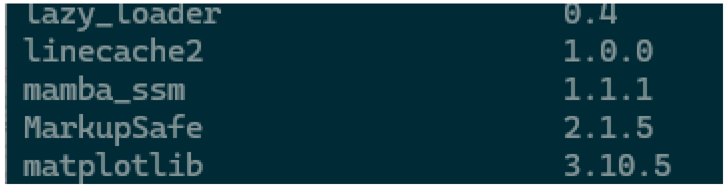
- □ mmcv直接使用pip install mmcv下载会报错，需要到官方复制下载命令



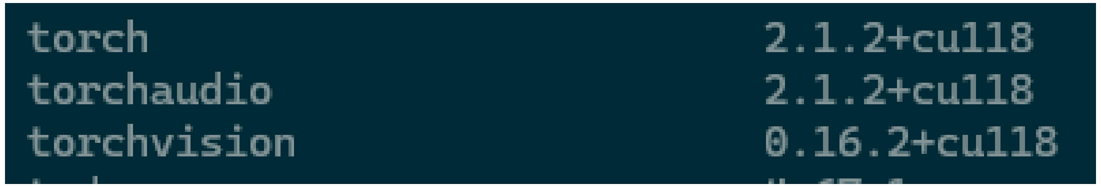
Select the appropriate installation command depending on the type of system, CUDA version, PyTorch version, and MMCV version

```
pip install mmcv==2.2.0 -f https://download.openmmlab.com/mmcv/dist/cu121/torch2.2/index.html
```

- □ Mamba块的安装，需要Pytorch与cuda版本的严格对齐：
  torch 2.1.2，cuda 11.8，mamba_ssm 1.1.1



```
lazy_loader                  0.4
linecache2                   1.0.0
mamba_ssm                    1.1.1
MarkupSafe                   2.1.5
matplotlib                   3.10.5
```

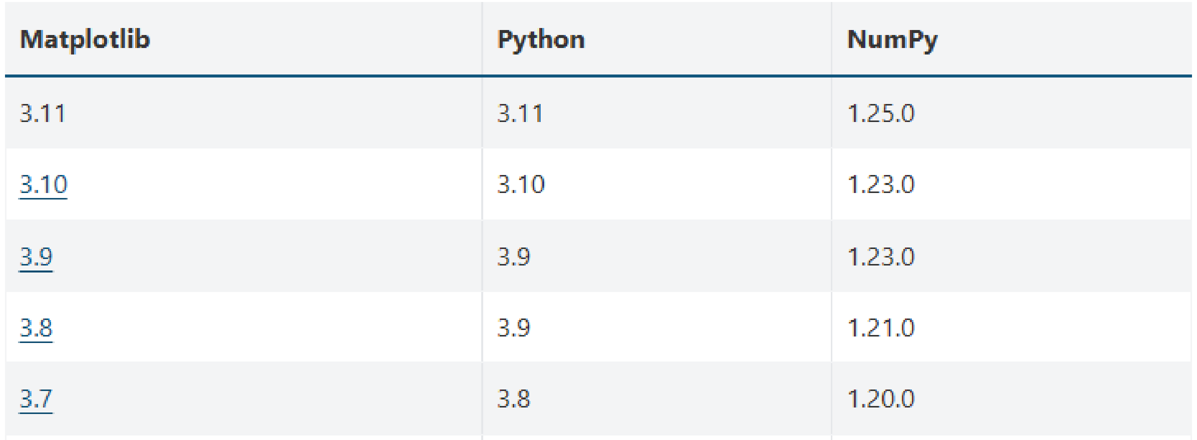


```
torch                              2.1.2+cu118
torchaudio                         2.1.2+cu118
torchvision                        0.16.2+cu118
```

配环境（难）

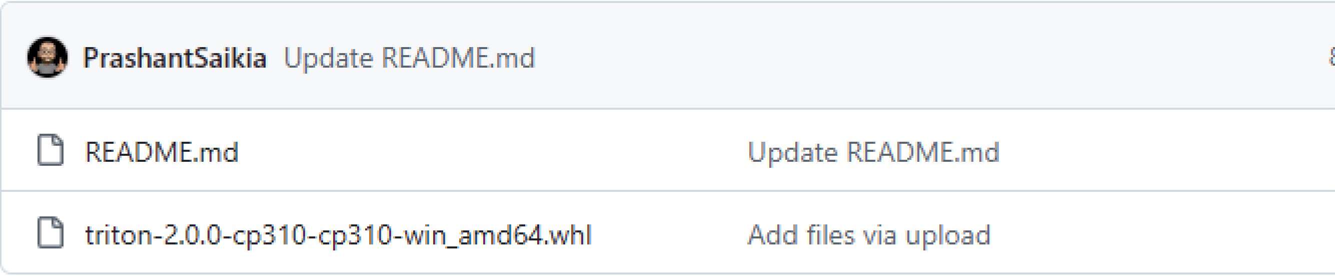
- ➢ **其他的疑难杂症**

  - □ **Pytho包之间存在的依赖关系**



Matplotlib	Python	NumPy
3.11	3.11	1.25.0
3.10	3.10	1.23.0
3.9	3.9	1.23.0
3.8	3.9	1.21.0
3.7	3.8	1.20.0

  - □ **triton官方只有linux版本，在windows安装会报错**



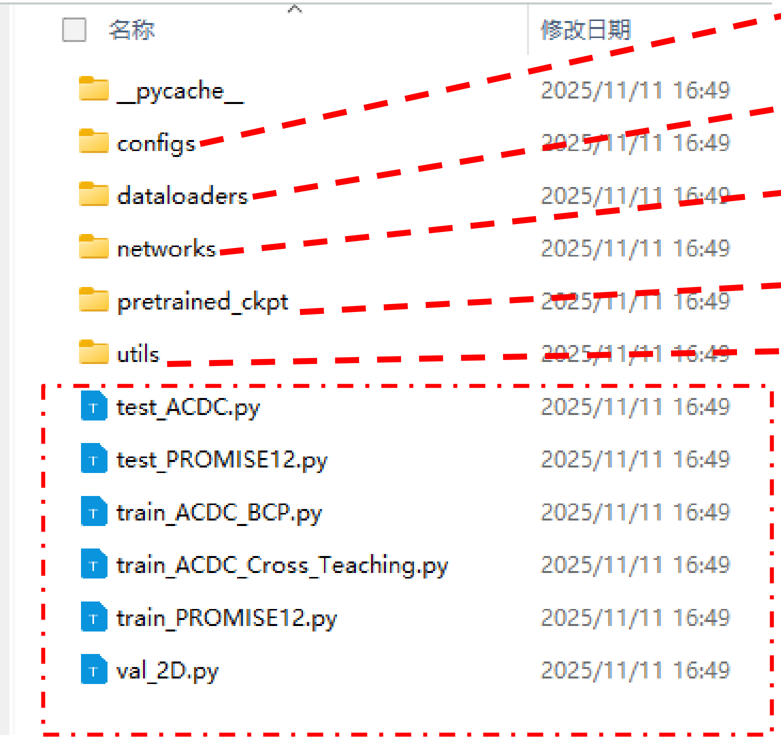
PrashantSaikia Update README.md

README.md	Update README.md
triton-2.0.0-cp310-cp310-win_amd64.whl	Add files via upload

https://github.com/PrashantSaikia/Triton-for-Windows

调试代码（难）

- 分析项目结构



名称	修改日期
__pycache__	2025/11/11 16:49
configs	2025/11/11 16:49
dataloaders	2025/11/11 16:49
networks	2025/11/11 16:49
pretrained_ckpt	2025/11/11 16:49
utils	2025/11/11 16:49
test_ACDC.py	2025/11/11 16:49
test_PROMISE12.py	2025/11/11 16:49
train_ACDC_BCP.py	2025/11/11 16:49
train_ACDC_Cross_Teaching.py	2025/11/11 16:49
train_PROMISE12.py	2025/11/11 16:49
val_2D.py	2025/11/11 16:49

配置文件

数据加载

网络模型文件

预训练权重

工具函数：数据预处理，loss计算等

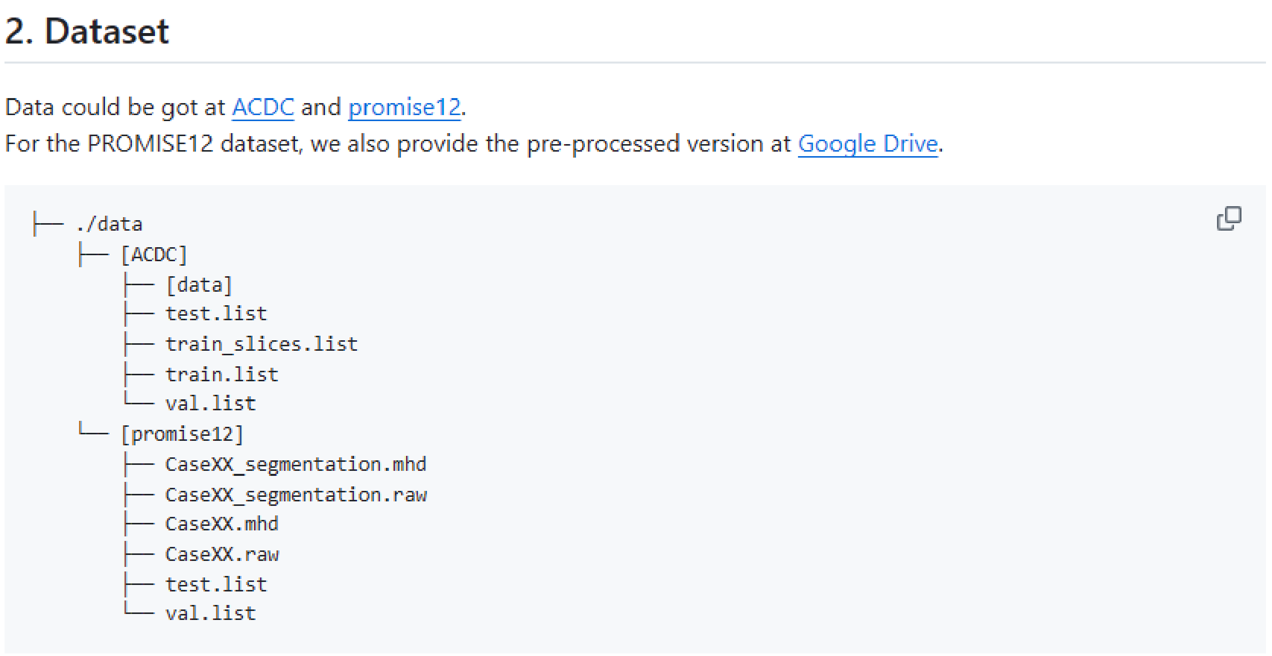
训练脚本，测试脚本和验证脚本

大部分的项目保持类似的结构，符合软件工程中的低耦合标准

调试代码（难）

➢ **下载数据集**

- □ **下载README中的数据集，并按要求组织数据集的文件结构**
- □ **公共数据集直接下载，比如COCO，ACDC，CHASE_DB1等**
- □ **私有数据集不公布，需要申请访问权限，成功率较低**



2. Dataset

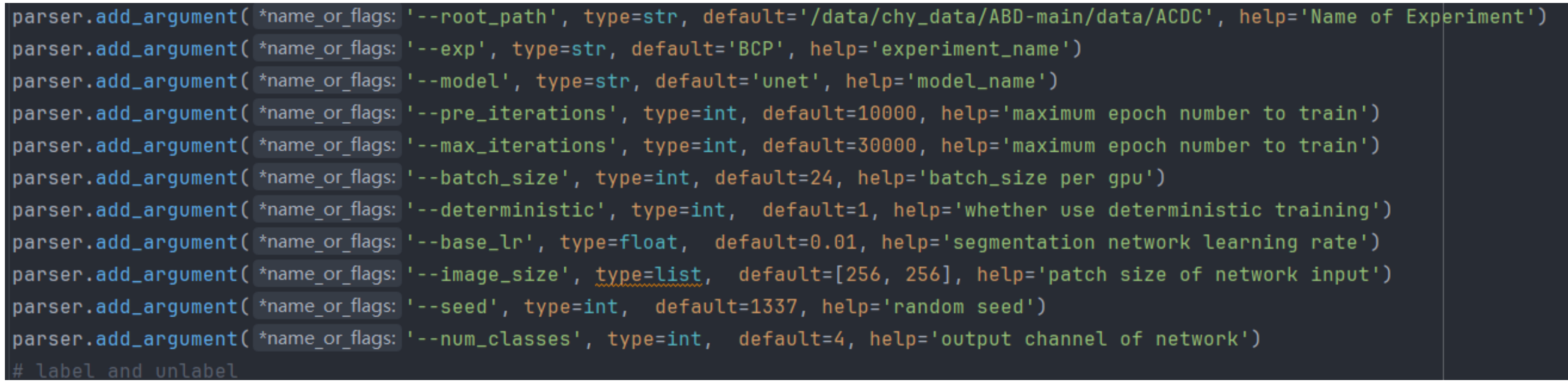
Data could be got at ACDC and promise12.
For the PROMISE12 dataset, we also provide the pre-processed version at Google Drive.

```
├── ./data
    ├── [ACDC]
        ├── [data]
        ├── test.list
        ├── train_slices.list
        ├── train.list
        └── val.list
    └── [promise12]
        ├── CaseXX_segmentation.mhd
        ├── CaseXX_segmentation.raw
        ├── CaseXX.mhd
        ├── CaseXX.raw
        ├── test.list
        └── val.list
```

调试代码（难）

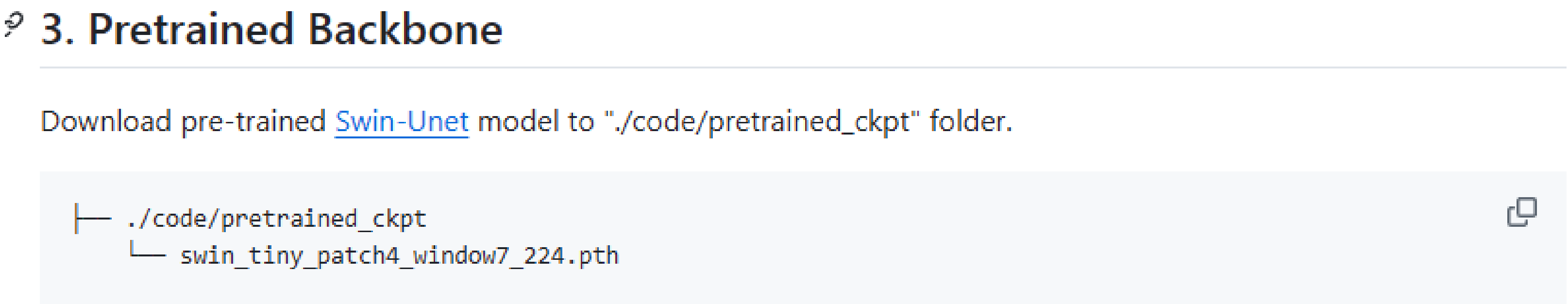
➢ 修改代码

□ 主要修改数据集路径，epoch，学习率，batch_size和GPU设备等



```
parser.add_argument( *name_or_flags: '--root_path', type=str, default='/data/chy_data/ABD-main/data/ACDC', help='Name of Experiment')
parser.add_argument( *name_or_flags: '--exp', type=str, default='BCP', help='experiment_name')
parser.add_argument( *name_or_flags: '--model', type=str, default='unet', help='model_name')
parser.add_argument( *name_or_flags: '--pre_iterations', type=int, default=10000, help='maximum epoch number to train')
parser.add_argument( *name_or_flags: '--max_iterations', type=int, default=30000, help='maximum epoch number to train')
parser.add_argument( *name_or_flags: '--batch_size', type=int, default=24, help='batch_size per gpu')
parser.add_argument( *name_or_flags: '--deterministic', type=int,  default=1, help='whether use deterministic training')
parser.add_argument( *name_or_flags: '--base_lr', type=float,  default=0.01, help='segmentation network learning rate')
parser.add_argument( *name_or_flags: '--image_size', type=list,  default=[256, 256], help='patch size of network input')
parser.add_argument( *name_or_flags: '--seed', type=int,  default=1337, help='random seed')
parser.add_argument( *name_or_flags: '--num_classes', type=int,  default=4, help='output channel of network')
# label and unlabel
```

□ 下载README中提到的预训练模型，放置在正确的位置



3. Pretrained Backbone

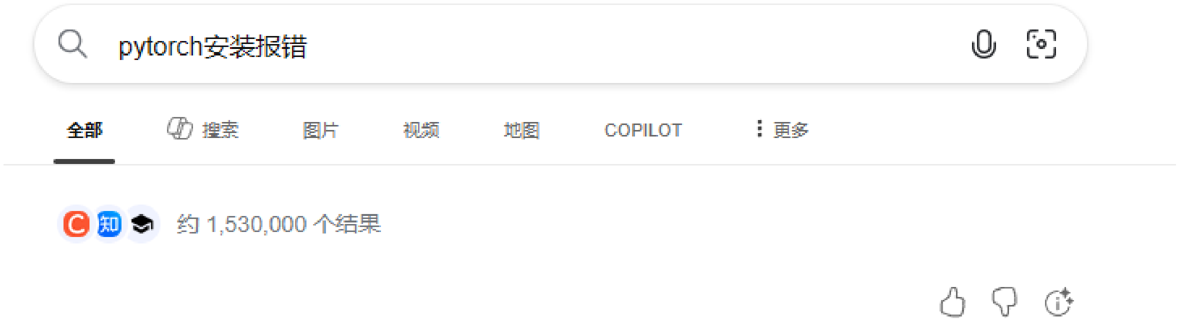
Download pre-trained Swin-Unet model to "./code/pretrained_ckpt" folder.

```
├── ./code/pretrained_ckpt
    └── swin_tiny_patch4_window7_224.pth
```

□ 这里99%会出现各种报错，除了硬件的报错，其他的大部分可以在各种网站或论坛（github、CSDN）上找到答案，最好还是在github的issue里找一找

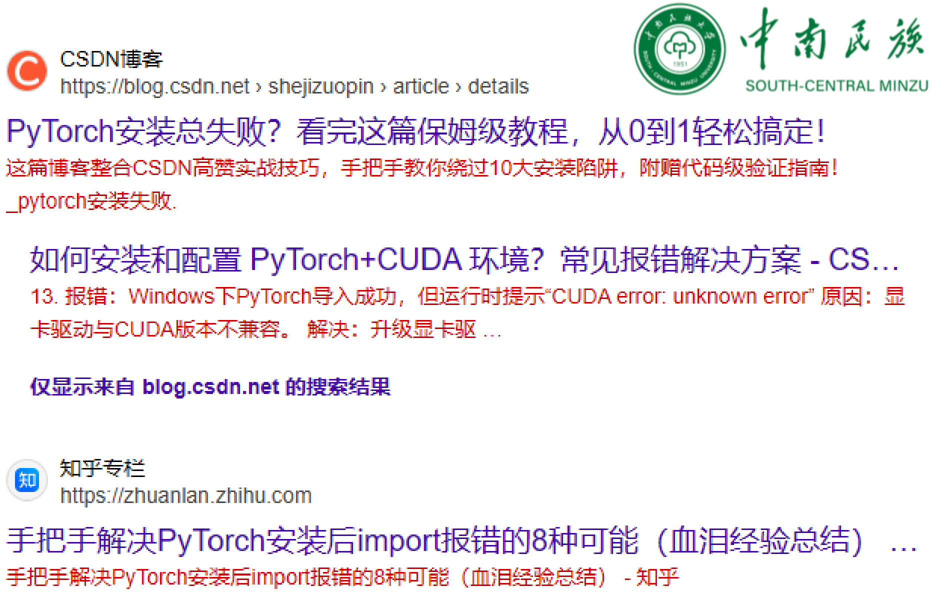
调试代码（难）

➢ 修改代码



pytorch安装报错

约 1,530,000 个结果



CSDN博客
https://blog.csdn.net › shejizuopin › article › details

PyTorch安装总失败？看完这篇保姆级教程，从0到1轻松搞定！

这篇博客整合CSDN高赞实战技巧，手把手教你绕过10大安装陷阱，附赠代码级验证指南！_pytorch安装失败.

如何安装和配置 PyTorch+CUDA 环境？常见报错解决方案 - CS…

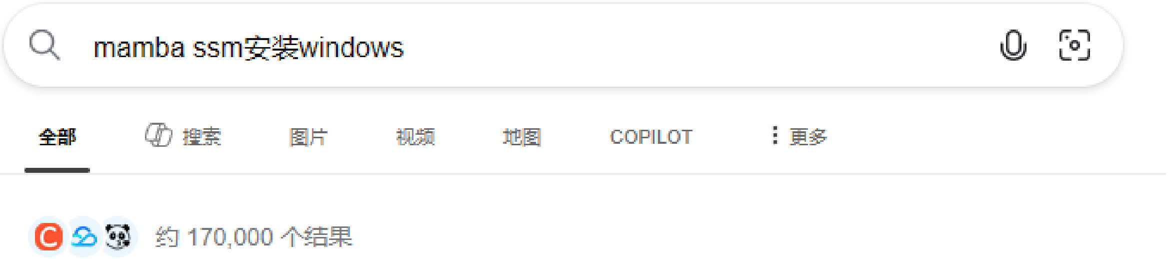
13. 报错：Windows下PyTorch导入成功，但运行时提示“CUDA error: unknown error” 原因：显卡驱动与CUDA版本不兼容。 解决：升级显卡驱 …

仅显示来自 blog.csdn.net 的搜索结果

知乎专栏
https://zhuanlan.zhihu.com

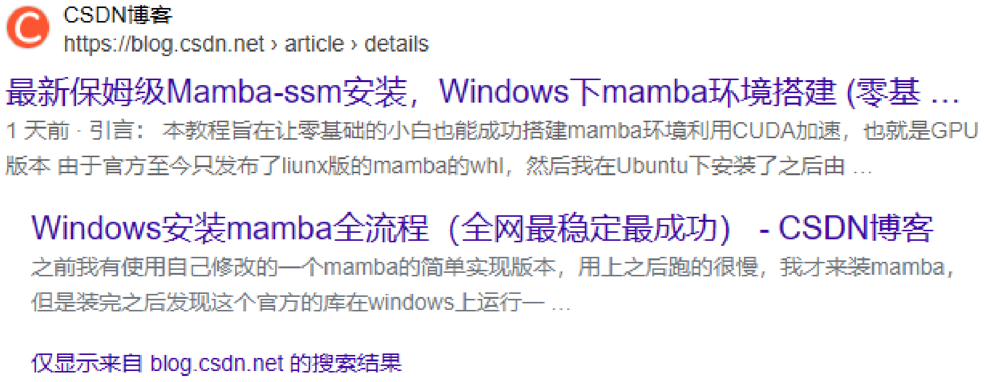
手把手解决PyTorch安装后import报错的8种可能（血泪经验总结） …

手把手解决PyTorch安装后import报错的8种可能（血泪经验总结） - 知乎



mamba ssm安装windows

约 170,000 个结果



CSDN博客
https://blog.csdn.net › article › details

最新保姆级Mamba-ssm安装，Windows下mamba环境搭建 (零基 …

1 天前 · 引言： 本教程旨在让零基础的小白也能成功搭建mamba环境利用CUDA加速，也就是GPU版本 由于官方至今只发布了liunx版的mamba的whl，然后我在Ubuntu下安装了之后由 …

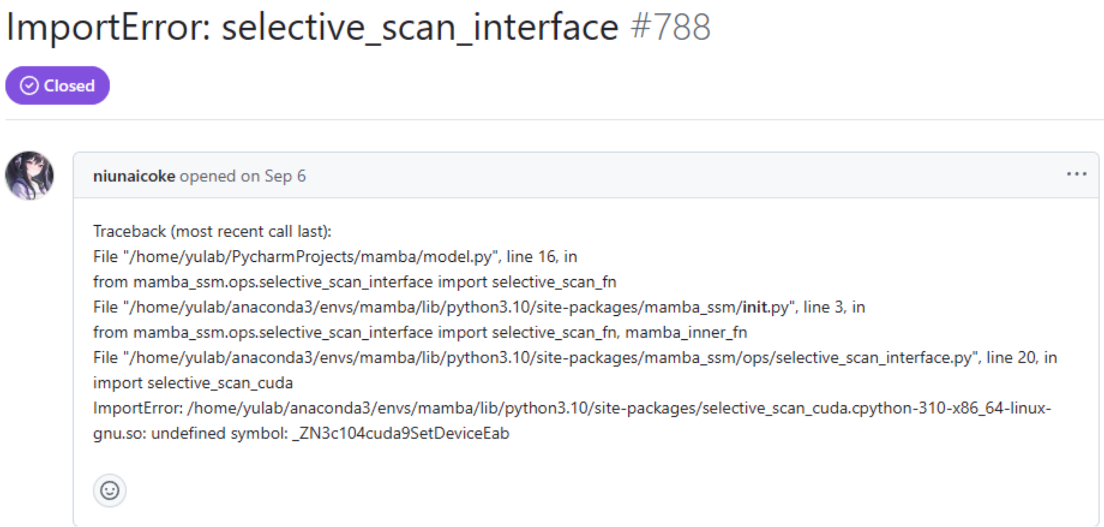
Windows安装mamba全流程（全网最稳定最成功） - CSDN博客

之前我有使用自己修改的一个mamba的简单实现版本，用上之后跑的很慢，我才来装mamba，但是装完之后发现这个官方的库在windows上运行一 …

仅显示来自 blog.csdn.net 的搜索结果

调试代码（难）

➢ 修改代码

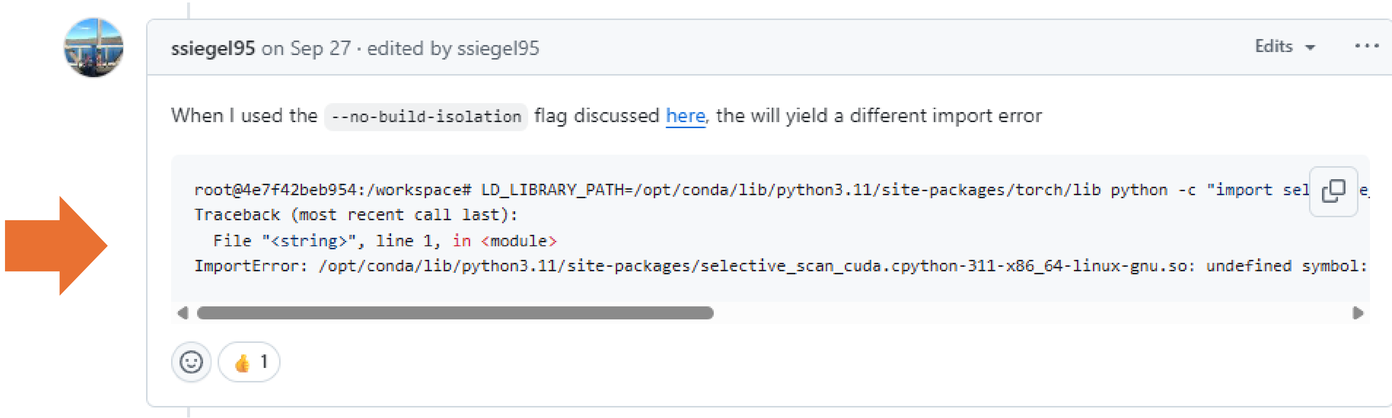


ImportError: selective_scan_interface #788

Closed

niunaicoke opened on Sep 6

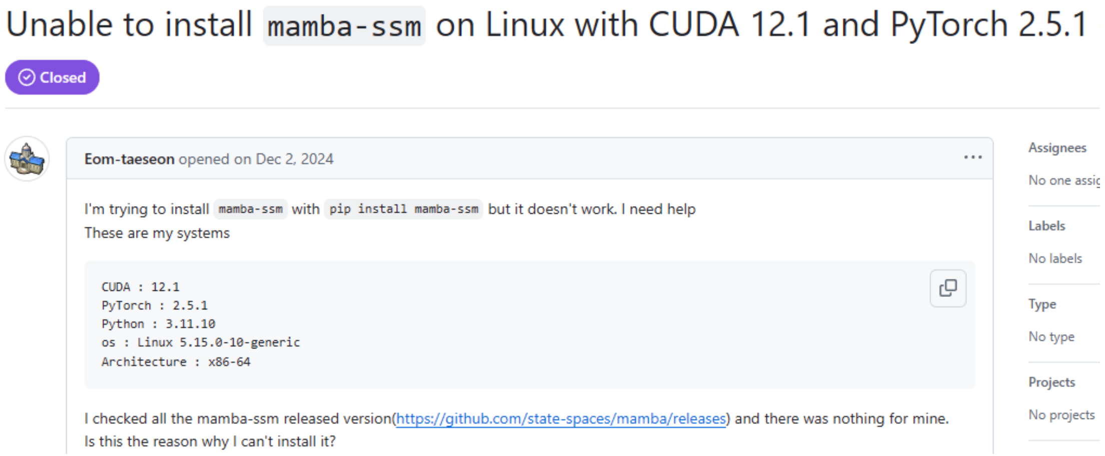
```
Traceback (most recent call last):
File "/home/yulab/PycharmProjects/mamba/model.py", line 16, in
from mamba_ssm.ops.selective_scan_interface import selective_scan_fn
File "/home/yulab/anaconda3/envs/mamba/lib/python3.10/site-packages/mamba_ssm/init.py", line 3, in
from mamba_ssm.ops.selective_scan_interface import selective_scan_fn, mamba_inner_fn
File "/home/yulab/anaconda3/envs/mamba/lib/python3.10/site-packages/mamba_ssm/ops/selective_scan_interface.py", line 20, in
import selective_scan_cuda
ImportError: /home/yulab/anaconda3/envs/mamba/lib/python3.10/site-packages/selective_scan_cuda.cpython-310-x86_64-linux-gnu.so: undefined symbol: _ZN3c104cuda9SetDeviceEab
```



ssiegel95 on Sep 27 · edited by ssiegel95

When I used the `--no-build-isolation` flag discussed here, the will yield a different import error

```
root@4e7f42beb954:/workspace# LD_LIBRARY_PATH=/opt/conda/lib/python3.11/site-packages/torch/lib python -c "import sel
Traceback (most recent call last):
  File "<string>", line 1, in <module>
ImportError: /opt/conda/lib/python3.11/site-packages/selective_scan_cuda.cpython-311-x86_64-linux-gnu.so: undefined symbol:
```



Unable to install `mamba-ssm` on Linux with CUDA 12.1 and PyTorch 2.5.1

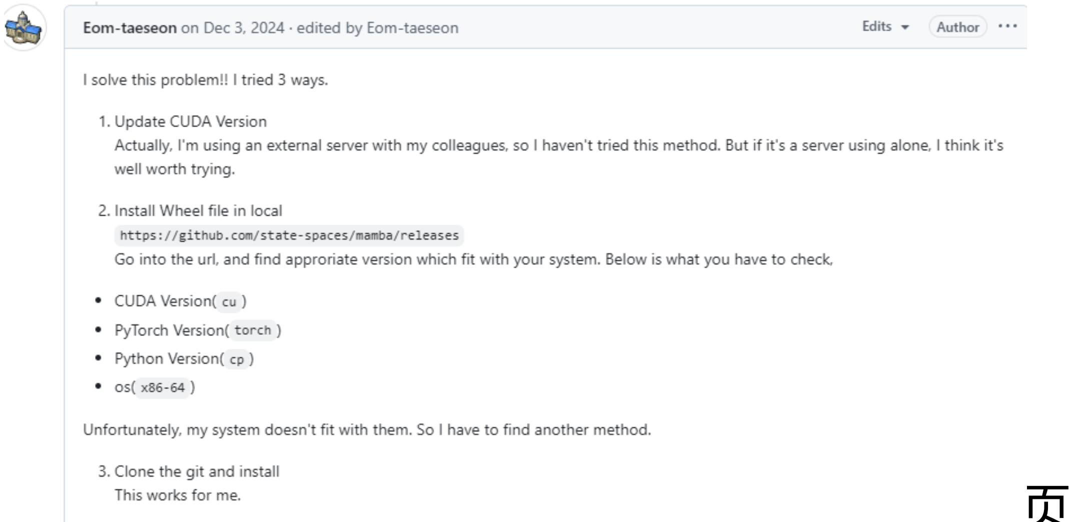
Closed

Eom-taeseon opened on Dec 2, 2024

I'm trying to install `mamba-ssm` with `pip install mamba-ssm` but it doesn't work. I need help
These are my systems

```
CUDA : 12.1
PyTorch : 2.5.1
Python : 3.11.10
os : Linux 5.15.0-10-generic
Architecture : x86-64
```

I checked all the mamba-ssm released version(https://github.com/state-spaces/mamba/releases) and there was nothing for mine.
Is this the reason why I can't install it?



Eom-taeseon on Dec 3, 2024 · edited by Eom-taeseon

I solve this problem!! I tried 3 ways.

1. Update CUDA Version
   Actually, I'm using an external server with my colleagues, so I haven't tried this method. But if it's a server using alone, I think it's well worth trying.

2. Install Wheel file in local
   `https://github.com/state-spaces/mamba/releases`
   Go into the url, and find approriate version which fit with your system. Below is what you have to check,

- CUDA Version(`cu`)
- PyTorch Version(`torch`)
- Python Version(`cp`)
- os(`x86-64`)

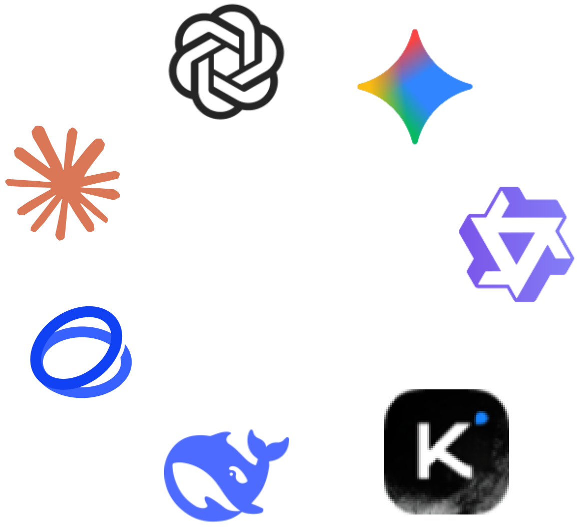
Unfortunately, my system doesn't fit with them. So I have to find another method.

3. Clone the git and install
   This works for me.

调试代码（难）

➢ 修改代码

□ 如果实在修改不了，可以借助大语言模型，比如claude，chatGPT，gemini，GLM，Kimi, DeepSeek，Qwen等

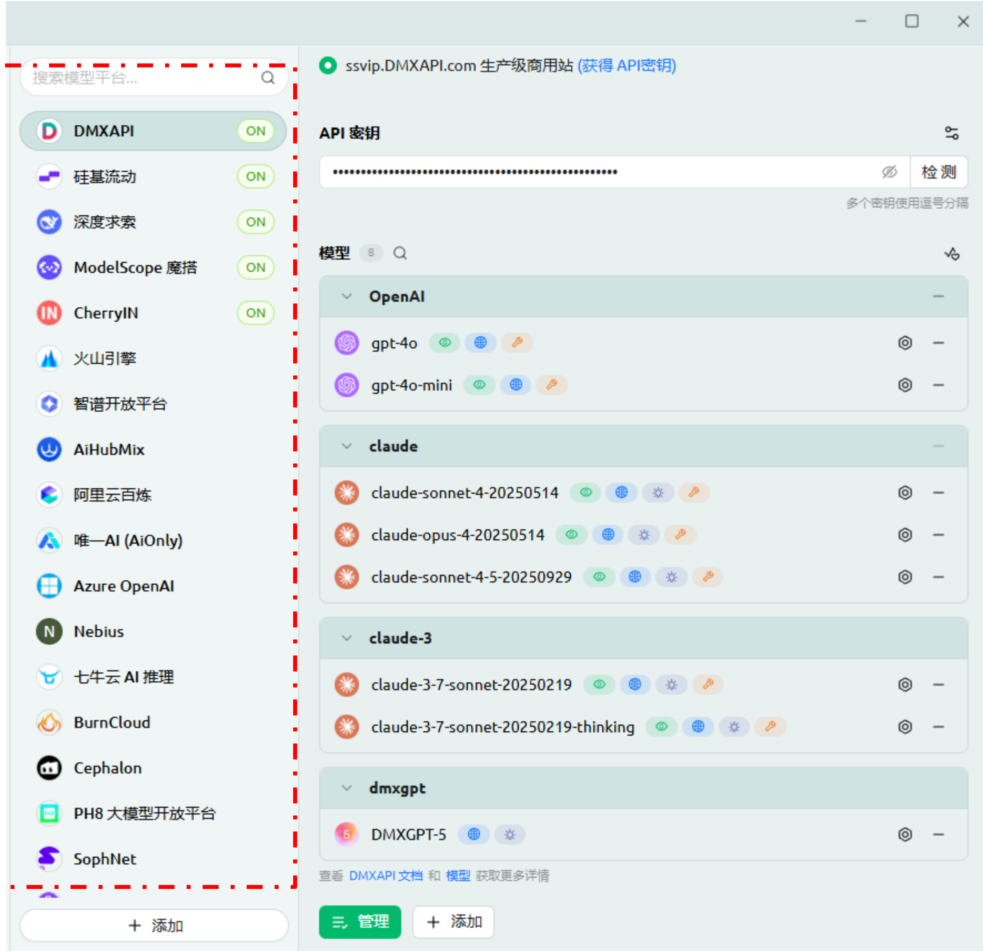


□ 不能过于相信大语言模型，部分大语言模型的幻觉较高，需要检查代码

调试代码（难）

➢ 修改代码

□ 使用大语言模型的API接口，按量付费，以Cherry Studio为例



ssvip.DMXAPI.com 生产级商用站 (获得 API密钥)

API 密钥

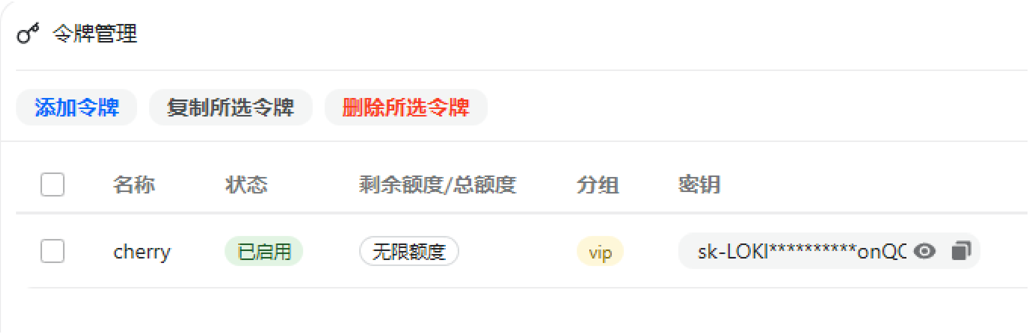
•• 检 测

多个密钥使用逗号分隔

模型 8

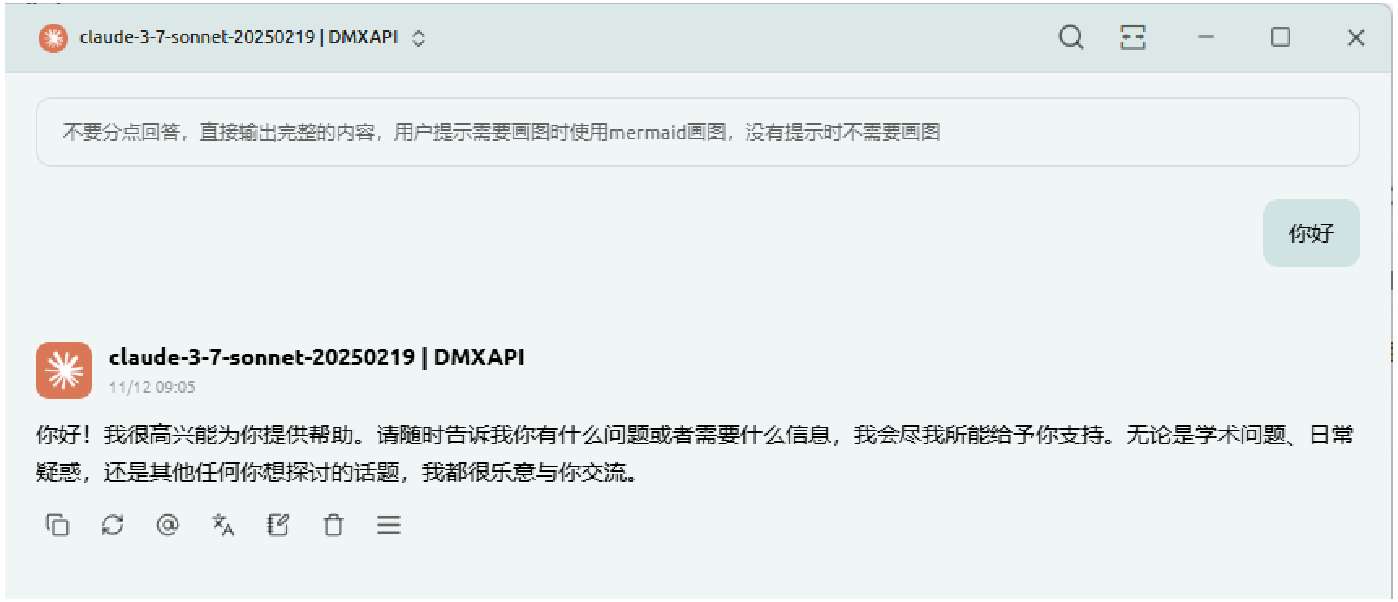
- OpenAI
  - gpt-4o
  - gpt-4o-mini
- claude
  - claude-sonnet-4-20250514
  - claude-opus-4-20250514
  - claude-sonnet-4-5-20250929
- claude-3
  - claude-3-7-sonnet-20250219
  - claude-3-7-sonnet-20250219-thinking
- dmxgpt
  - DMXGPT-5

查看 DMXAPI 文档 和 模型 获取更多详情



令牌管理

□	名称	状态	剩余额度/总额度	分组	密钥
□	cherry	已启用	无限额度	vip	sk-LOKI**********onQC



claude-3-7-sonnet-20250219 | DMXAPI

不要分点回答，直接输出完整的内容，用户提示需要画图时使用mermaid画图，没有提示时不需要画图

你好

claude-3-7-sonnet-20250219 | DMXAPI

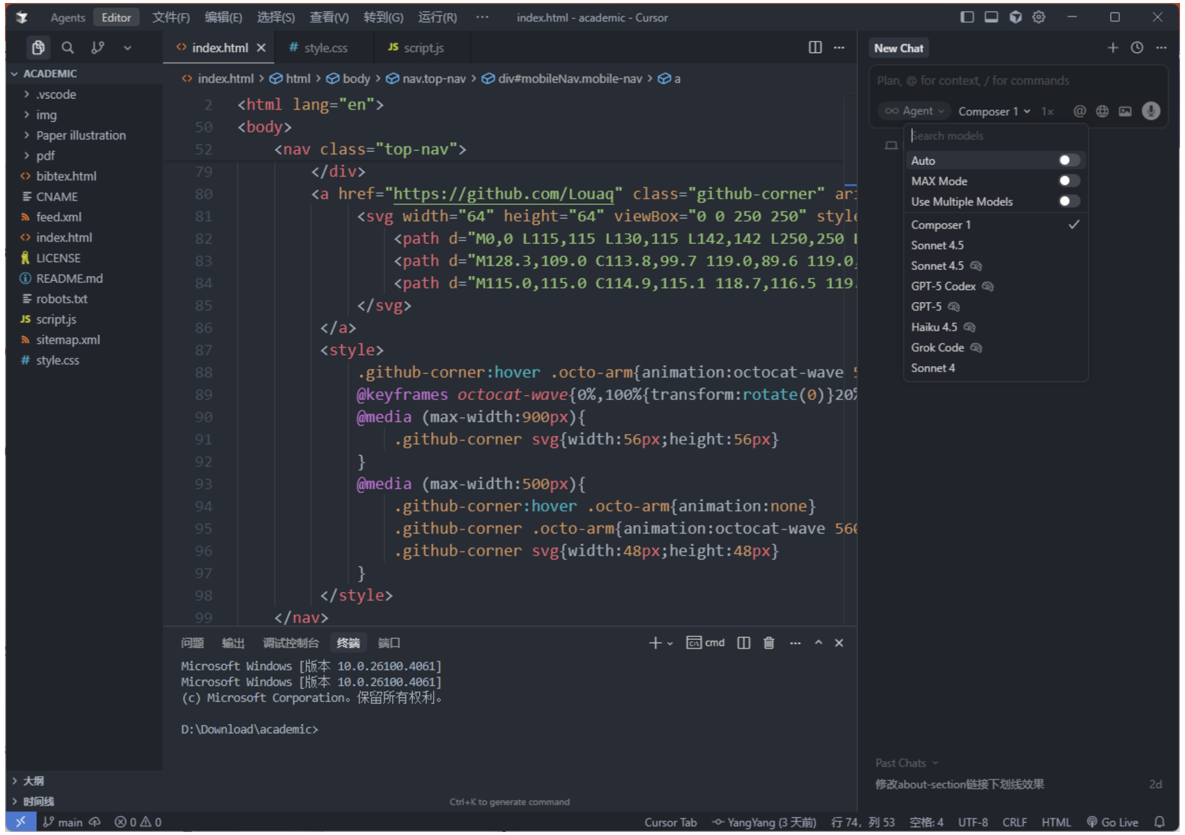
11/12 09:05

你好！我很高兴能为你提供帮助。请随时告诉我你有什么问题或者需要什么信息，我会尽我所能给予你支持。无论是学术问题、日常疑惑，还是其他任何你想探讨的话题，我都很乐意与你交流。

调试代码（难）

➢ 修改代码

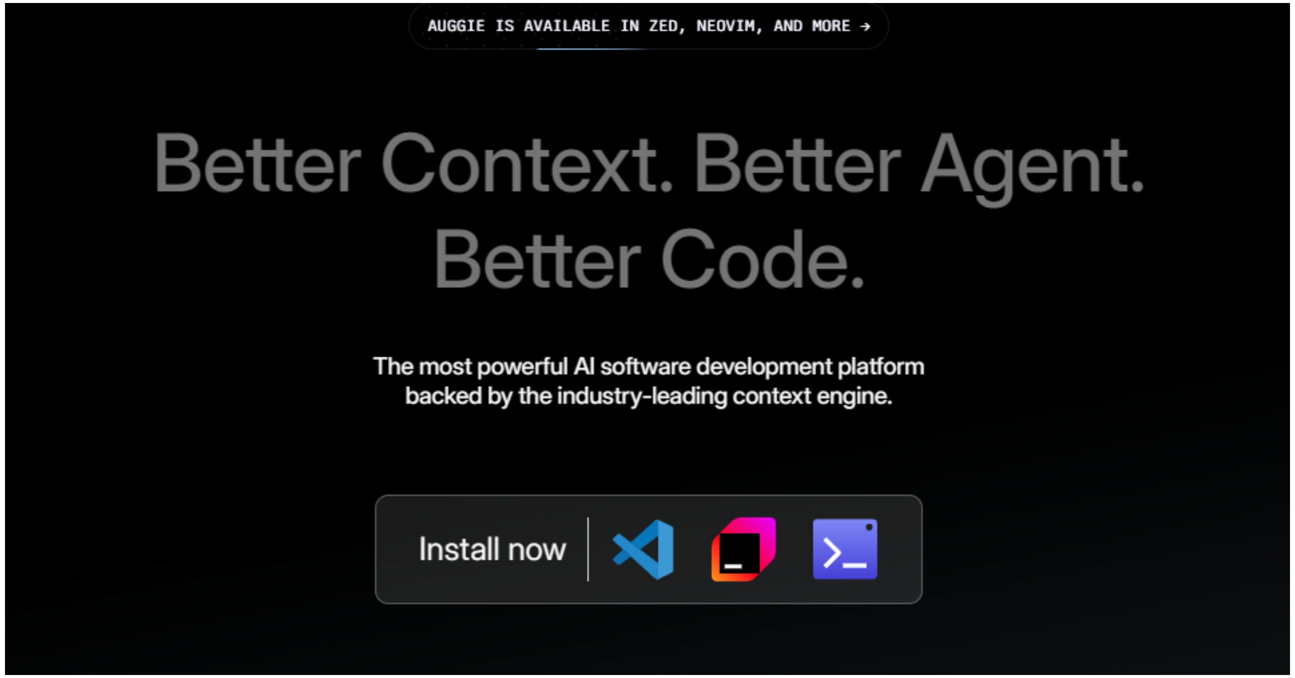
- □ 还可以使用AI 编码工具，比如cursor，augment coding（插件），codex等



```
<html lang="en">
<body>
    <nav class="top-nav">
        </div>
        <a href="https://github.com/Louaq" class="github-corner" ar
            <svg width="64" height="64" viewBox="0 0 250 250" style
                <path d="M0,0 L115,115 L130,115 L142,142 L250,250 L
                <path d="M128.3,109.0 C113.8,99.7 119.0,89.6 119.0,
                <path d="M115.0,115.0 C114.9,115.1 118.7,116.5 119.
            </svg>
         </a>
         <style>
            .github-corner:hover .octo-arm{animation:octocat-wave 5
            @keyframes octocat-wave{0%,100%{transform:rotate(0)}20%
            @media (max-width:900px){
                .github-corner svg{width:56px;height:56px}
            }
            @media (max-width:500px){
                .github-corner:hover .octo-arm{animation:none}
                .github-corner .octo-arm{animation:octocat-wave 560
                .github-corner svg{width:48px;height:48px}
            }
         </style>
    </nav>
```

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

D:\Download\academic>
```

修改about-section链接下划线效果



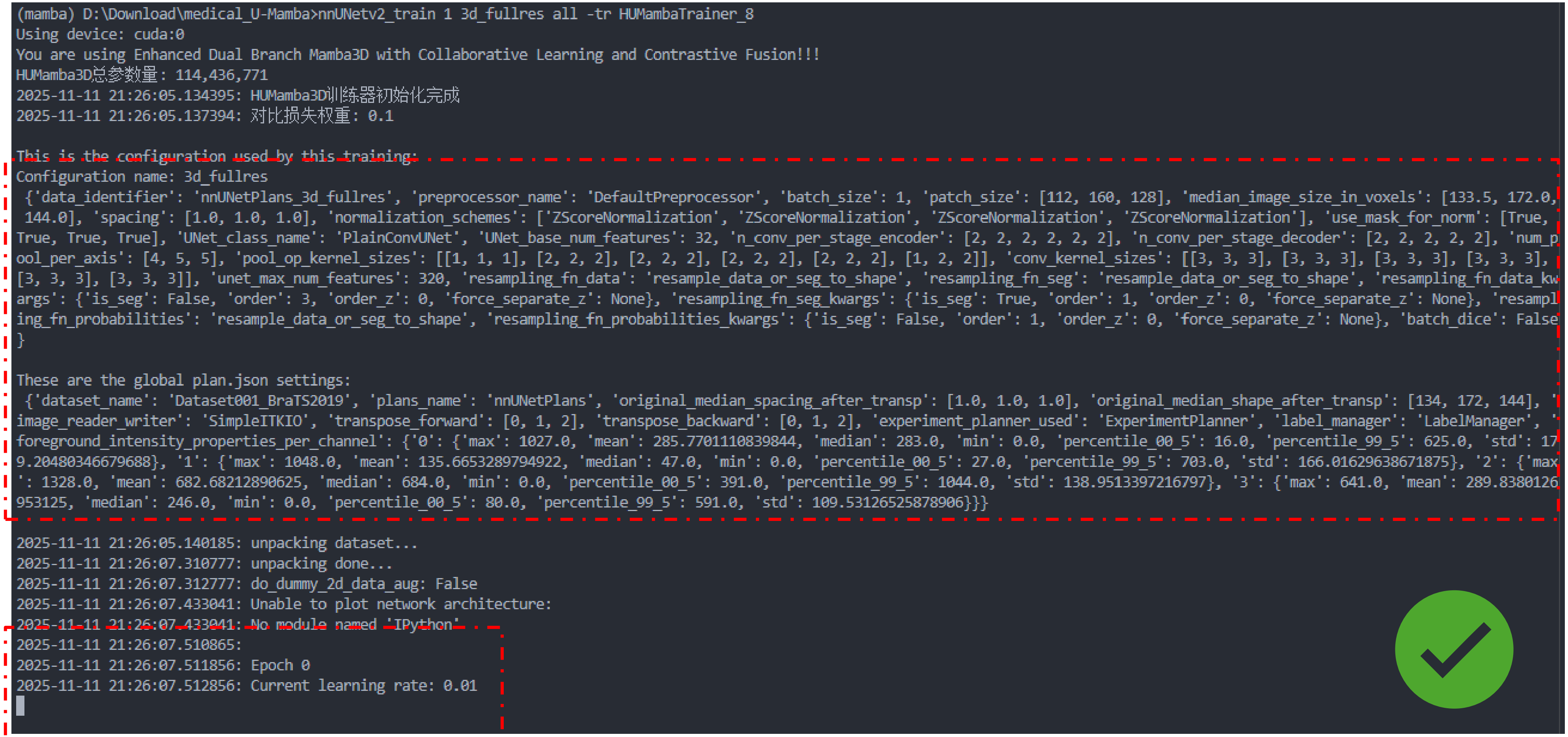
Better Context. Better Agent. Better Code.

The most powerful AI software development platform backed by the industry-leading context engine.

调试代码（难）

➢ 运行代码

□ 如果以上步骤没有出现较大的问题，那么可以执行训练代码，训练时间依据不同任务而不同



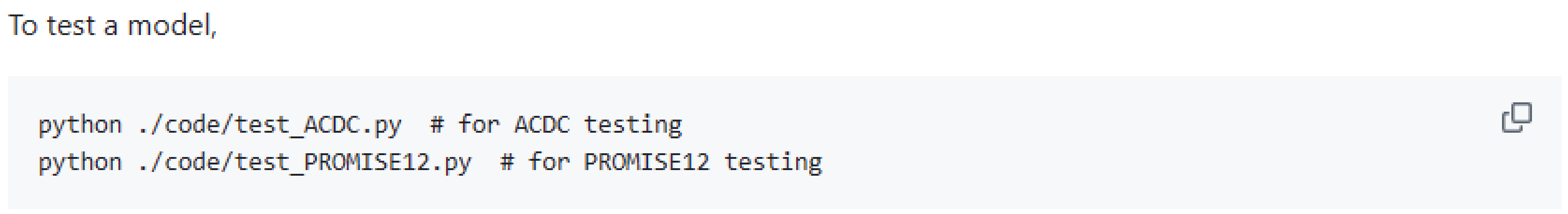
```
(mamba) D:\Download\medical_U-Mamba>nnUNetv2_train 1 3d_fullres all -tr HUMambaTrainer_8
Using device: cuda:0
You are using Enhanced Dual Branch Mamba3D with Collaborative Learning and Contrastive Fusion!!!
HUMamba3D总参数量: 114,436,771
2025-11-11 21:26:05.134395: HUMamba3D训练器初始化完成
2025-11-11 21:26:05.137394: 对比损失权重: 0.1

This is the configuration used by this training:
Configuration name: 3d_fullres
 {'data_identifier': 'nnUNetPlans_3d_fullres', 'preprocessor_name': 'DefaultPreprocessor', 'batch_size': 1, 'patch_size': [112, 160, 128], 'median_image_size_in_voxels': [133.5, 172.0,
 144.0], 'spacing': [1.0, 1.0, 1.0], 'normalization_schemes': ['ZScoreNormalization', 'ZScoreNormalization', 'ZScoreNormalization', 'ZScoreNormalization'], 'use_mask_for_norm': [True,
True, True, True], 'UNet_class_name': 'PlainConvUNet', 'UNet_base_num_features': 32, 'n_conv_per_stage_encoder': [2, 2, 2, 2, 2, 2], 'n_conv_per_stage_decoder': [2, 2, 2, 2, 2], 'num_p
ool_per_axis': [4, 5, 5], 'pool_op_kernel_sizes': [[1, 1, 1], [2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [1, 2, 2]], 'conv_kernel_sizes': [[3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3],
[3, 3, 3], [3, 3, 3]], 'unet_max_num_features': 320, 'resampling_fn_data': 'resample_data_or_seg_to_shape', 'resampling_fn_seg': 'resample_data_or_seg_to_shape', 'resampling_fn_data_kw
args': {'is_seg': False, 'order': 3, 'order_z': 0, 'force_separate_z': None}, 'resampling_fn_seg_kwargs': {'is_seg': True, 'order': 1, 'order_z': 0, 'force_separate_z': None}, 'resampl
ing_fn_probabilities': 'resample_data_or_seg_to_shape', 'resampling_fn_probabilities_kwargs': {'is_seg': False, 'order': 1, 'order_z': 0, 'force_separate_z': None}, 'batch_dice': False
}

These are the global plan.json settings:
 {'dataset_name': 'Dataset001_BraTS2019', 'plans_name': 'nnUNetPlans', 'original_median_spacing_after_transp': [1.0, 1.0, 1.0], 'original_median_shape_after_transp': [134, 172, 144], '
image_reader_writer': 'SimpleITKIO', 'transpose_forward': [0, 1, 2], 'transpose_backward': [0, 1, 2], 'experiment_planner_used': 'ExperimentPlanner', 'label_manager': 'LabelManager', '
foreground_intensity_properties_per_channel': {'0': {'max': 1027.0, 'mean': 285.7701110839844, 'median': 283.0, 'min': 0.0, 'percentile_00_5': 16.0, 'percentile_99_5': 625.0, 'std': 17
9.20480346679688}, '1': {'max': 1048.0, 'mean': 135.6653289794922, 'median': 47.0, 'min': 0.0, 'percentile_00_5': 27.0, 'percentile_99_5': 703.0, 'std': 166.01629638671875}, '2': {'max
': 1328.0, 'mean': 682.68212890625, 'median': 684.0, 'min': 0.0, 'percentile_00_5': 391.0, 'percentile_99_5': 1044.0, 'std': 138.9513397216797}, '3': {'max': 641.0, 'mean': 289.8380126
953125, 'median': 246.0, 'min': 0.0, 'percentile_00_5': 80.0, 'percentile_99_5': 591.0, 'std': 109.53126525878906}}}

2025-11-11 21:26:05.140185: unpacking dataset...
2025-11-11 21:26:07.310777: unpacking done...
2025-11-11 21:26:07.312777: do_dummy_2d_data_aug: False
2025-11-11 21:26:07.433041: Unable to plot network architecture:
2025-11-11 21:26:07.433041: No module named 'IPython'
2025-11-11 21:26:07.510865:
2025-11-11 21:26:07.511856: Epoch 0
2025-11-11 21:26:07.512856: Current learning rate: 0.01
```

网络配置信息

每一个epoch的损失和学习率

调试代码（难）

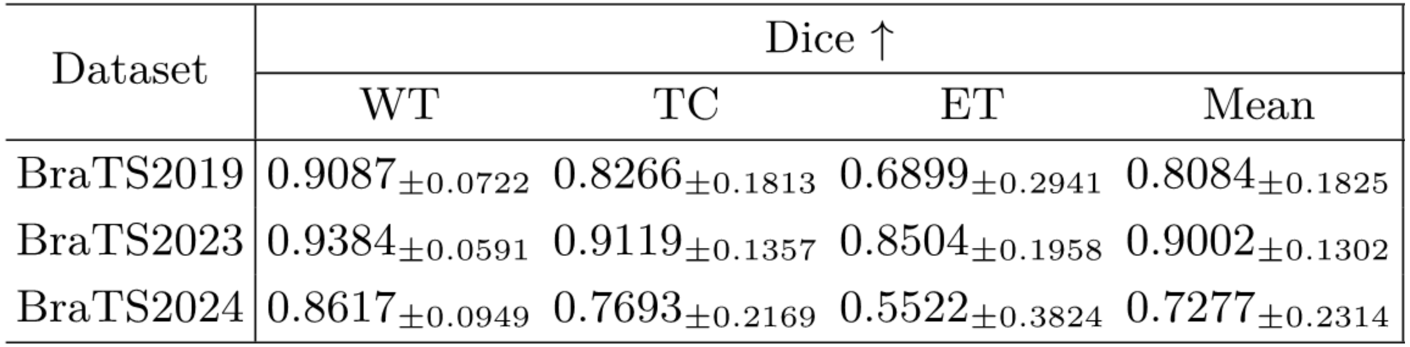
➢ 测试模型



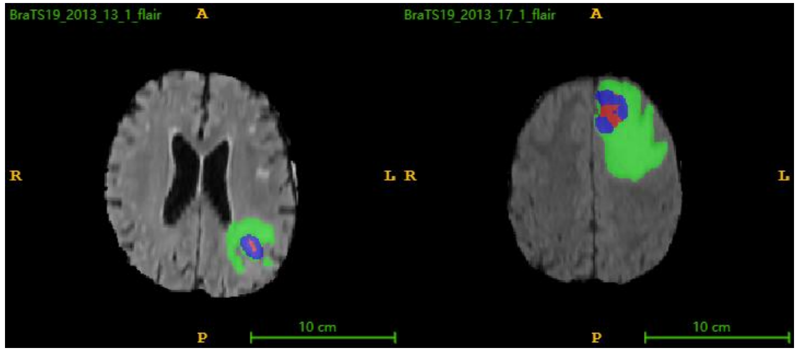
To test a model,

```
python ./code/test_ACDC.py  # for ACDC testing
python ./code/test_PROMISE12.py  # for PROMISE12 testing
```

□ 设置模型路径，运行测试脚本，得到预测的结果



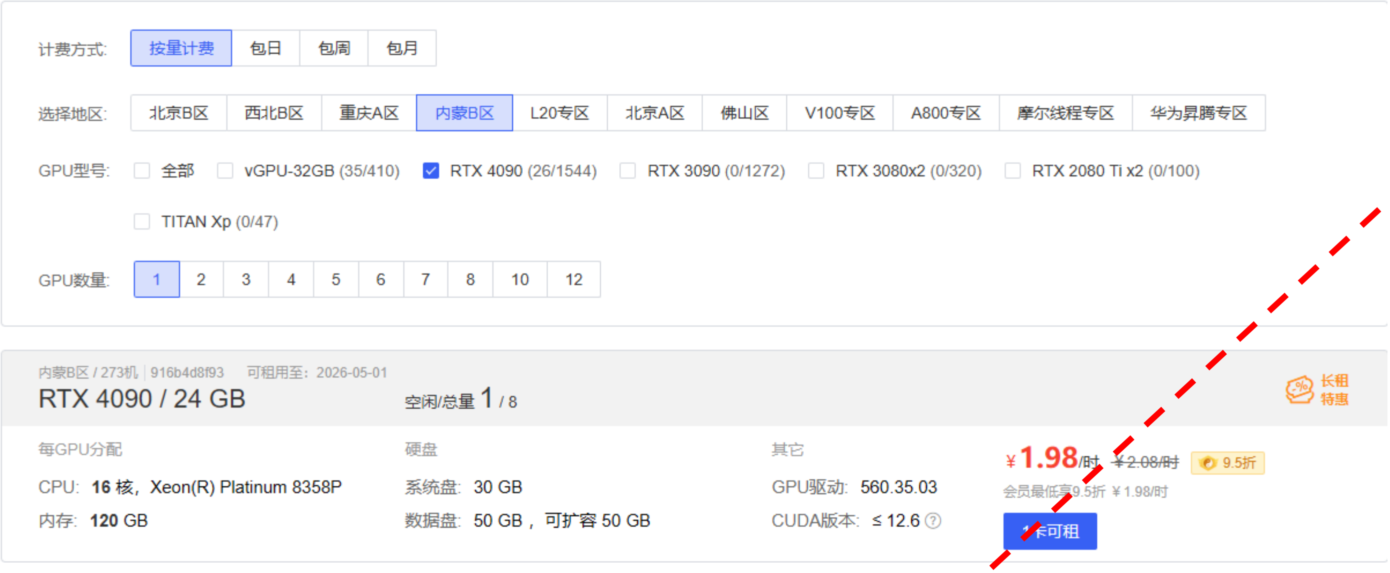
Dataset	Dice ↑			
	WT	TC	ET	Mean
BraTS2019	$0.9087_{\pm 0.0722}$	$0.8266_{\pm 0.1813}$	$0.6899_{\pm 0.2941}$	$0.8084_{\pm 0.1825}$
BraTS2023	$0.9384_{\pm 0.0591}$	$0.9119_{\pm 0.1357}$	$0.8504_{\pm 0.1958}$	$0.9002_{\pm 0.1302}$
BraTS2024	$0.8617_{\pm 0.0949}$	$0.7693_{\pm 0.2169}$	$0.5522_{\pm 0.3824}$	$0.7277_{\pm 0.2314}$



记录实验数据

服务器训练

- 由于本地的算力有限，所以利用服务器加速训练过程，以AutoDL为例



计费方式: 按量计费 | 包日 | 包周 | 包月

选择地区: 北京B区 | 西北B区 | 重庆A区 | 内蒙B区 | L20专区 | 北京A区 | 佛山区 | V100专区 | A800专区 | 摩尔线程专区 | 华为昇腾专区

GPU型号: ☐ 全部 ☐ vGPU-32GB (35/410) ☑ RTX 4090 (26/1544) ☐ RTX 3090 (0/1272) ☐ RTX 3080x2 (0/320) ☐ RTX 2080 Ti x2 (0/100) ☐ TITAN Xp (0/47)

GPU数量: 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 10 | 12

内蒙B区 / 273机 | 916b4d8f93 可租用至: 2026-05-01

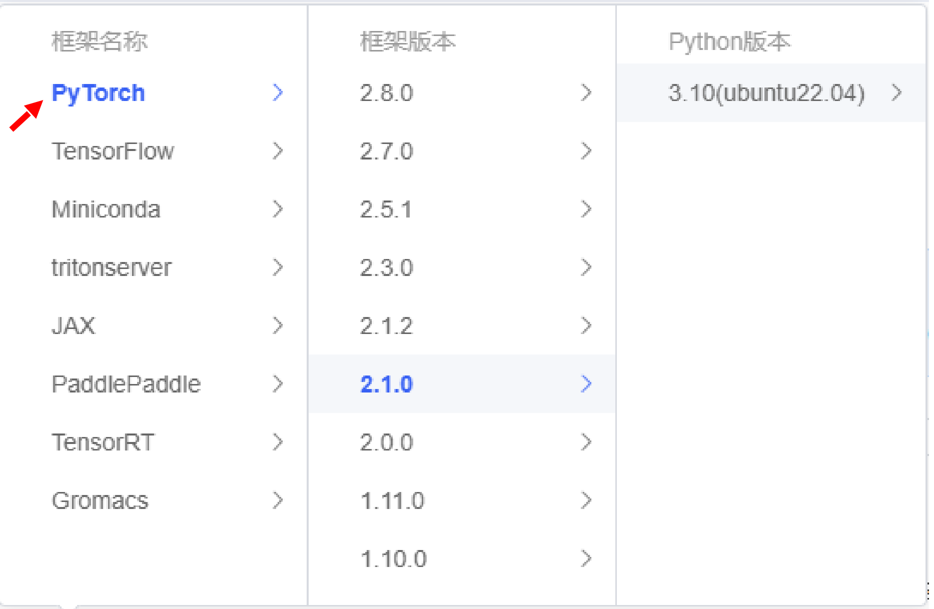
RTX 4090 / 24 GB 空闲/总量 1 / 8 长租特惠

每GPU分配	硬盘	其它
CPU: 16 核，Xeon(R) Platinum 8358P	系统盘: 30 GB	GPU驱动: 560.35.03
内存: 120 GB	数据盘: 50 GB，可扩容 50 GB	CUDA版本: ≤ 12.6

¥1.98/时 ~~¥2.08/时~~ 9.5折

会员最低享9.5折 ¥1.98/时

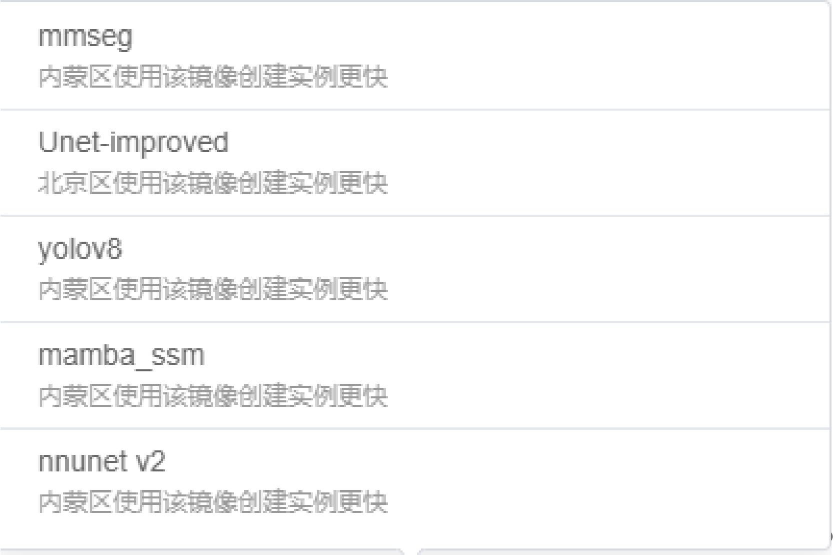
1卡可租



框架名称
- PyTorch
- TensorFlow
- Miniconda
- tritonserver
- JAX
- PaddlePaddle
- TensorRT
- Gromacs

框架版本
- 2.8.0
- 2.7.0
- 2.5.1
- 2.3.0
- 2.1.2
- 2.1.0
- 2.0.0
- 1.11.0
- 1.10.0

Python版本
- 3.10(ubuntu22.04)



mmseg
内蒙区使用该镜像创建实例更快

Unet-improved
北京区使用该镜像创建实例更快

yolov8
内蒙区使用该镜像创建实例更快

mamba_ssm
内蒙区使用该镜像创建实例更快

nnunet v2
内蒙区使用该镜像创建实例更快

- ☐ 选择租用的显卡，设置Pytorch版本和Python版本即可，也可以选择之前报存的镜像文件

- ☐ 如果训练任务复杂，可以使用多卡训练或分布式训练

服务器训练

➢ 熟悉基本的inux操作命令

- □ cd dataset，cd ../，vim train.py，cat train.py
- □ mkdir data，touch test.py
- □ mv Dataloader.py /root/data，cp Dataloader.py /root/data
- □ rm –rf val.py
- □ zip results.zip –r /root/autodl-tmp，unzip results.zip
- □ sudo apt-get update

总结

上述只是对如何复现论文代码的简单概述，具体的实践过程需要针对具体的代码，比如有的项目没有给出评估脚本和测试脚本，此时需要根据训练代码，自己写出评估和测试脚本

Thank you