部署QwenVL与Swift微调流程

QwenVL github官网：[QwenLM/Qwen3-VL: Qwen3-VL is the multimodal large language model series developed by Qwen team, Alibaba Cloud.]

Swift github官网：[modelscope/ms-swift: Use PEFT or Full-parameter to CPT/SFT/DPO/GRPO 500+ LLMs (Qwen3, Qwen3-MoE, Llama4, GLM4.5, InternLM3, DeepSeek-R1, ...) and 200+ MLLMs (Qwen3-VL, Qwen3-Omni, InternVL3.5, Ovis2.5, Llava, GLM4v, Phi4, ...) (AAAI 2025).]

□ 总览

1、参考AutoDL官方文档及Qwen等模型说明，尝试将Qwen、Deepseek等MLLM在AutoDL平台部署。
2、尝试将采集的证照图像送入MLLM，分析图像上的文本信息、污损信息、可能的伪造信息（自行对图像进行伪造）
3、对以上内容，撰写报告（包括实验过程、经验总结），11月20日提交

■ MLLM部署、使用与微调方法（以Qwen为例）

①Python环境部署，准备数据集（图文对，.json文件，格式参照文档）

②克隆MLLM的GitHub存储库：git clone https://github.com/QwenLM/Qwen2.5-VL

③下载模型权重：HF-Mirror: https://hf-mirror.com/

魔搭：https://www.modelscope.cn/my/overview

④ 参照模型官网的文档实现推理

⑤克隆ms-swift框架并安装所需环境：git clone https://github.com/modelscope/ms-swift.git

⑥参照swift官方文档，进行训练与推理

◘ AutoDL服务器使用说明（会用则跳过）

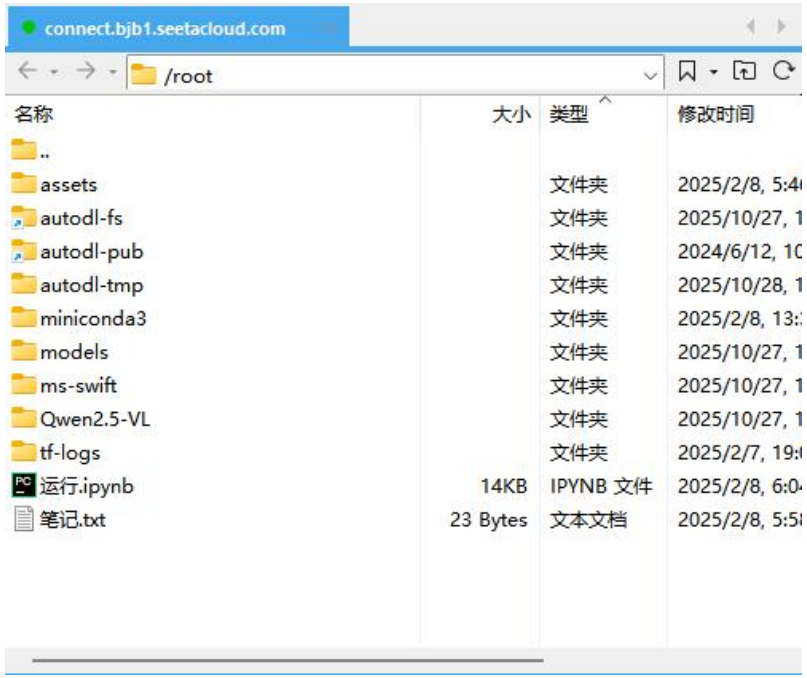
■ 请自行下载VsCode、Pycharm等能连接服务器的编译软件，或下载MobaXterm等远程终端工具，安装插件等自行百度搜索教程，以下给出两个教程网站。建议下载个Xftp以传输文件。

AutoDL服务器示例：

端口号Port　　用户名User/Name　　地址IP(Host)

ssh -p **30241** **root**@**connect.bjb1.seetacloud.com**

密码：**********



■ 请在 autodl-tmp中进行你所有的操作（共50G）

包括但不限于数据、代码等.....

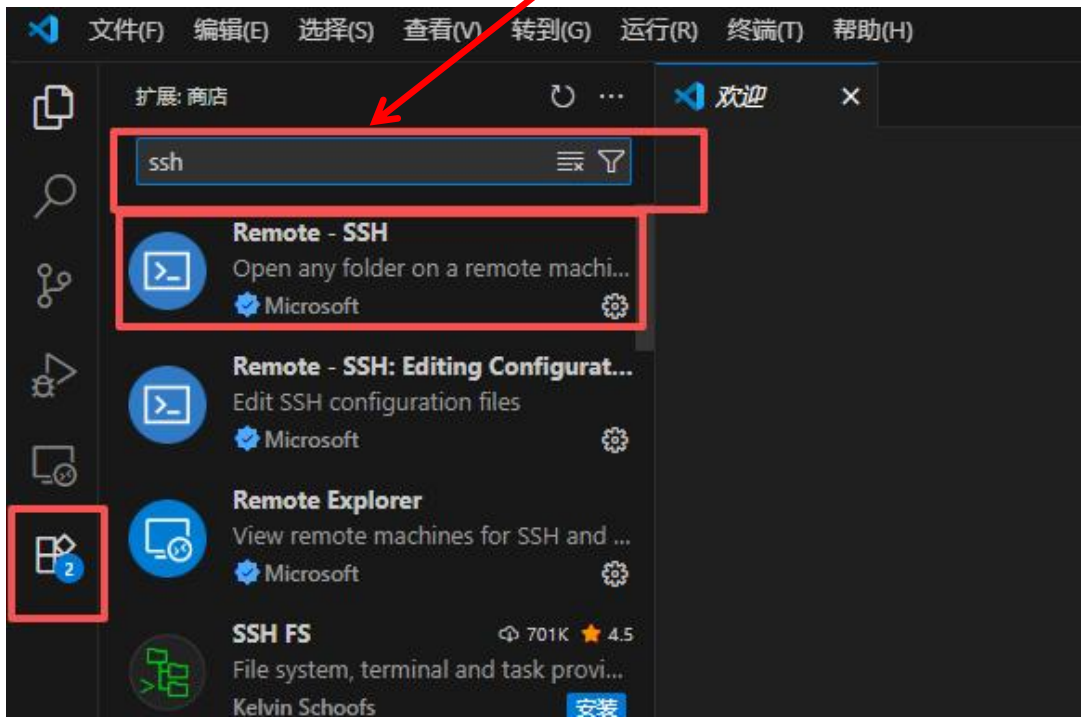
- **models——里面存储了Qwen2.5VL-3B和7B的模型权重**
- **Qwen2.5-VL——Qwen的github库，复制到自己的文件夹**
- **ms-swift——Swift框架的github库**

VScode连接远程服务器教程：　vscode连接远程服务器（傻瓜式教学）-CSDN博客

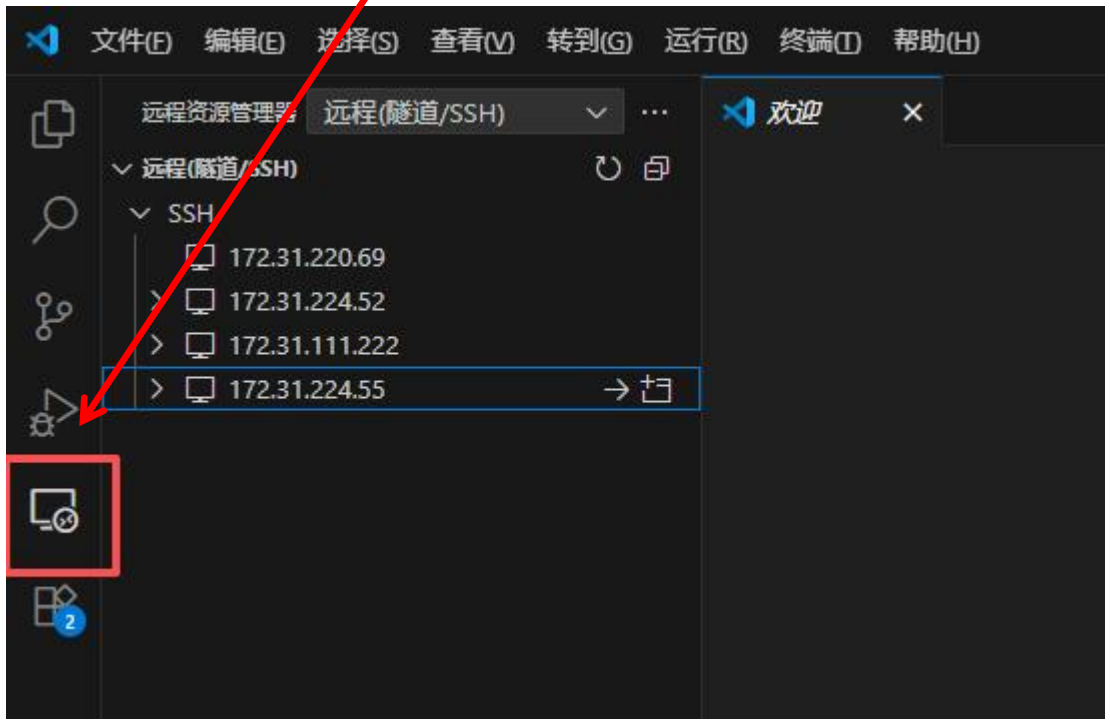
Pycharm连接远程服务器教程：从 PyCharm 连接到远程服务器 | PyCharm 文档

AutoDL服务器使用说明（会用则跳过）

Vscode安装插件Remote-SSH

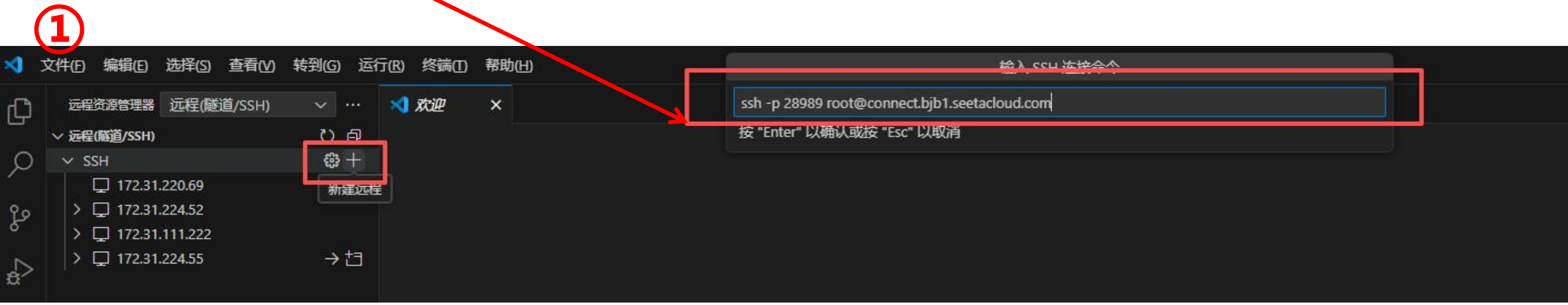


安装成功后会出现这个远程资源管理器

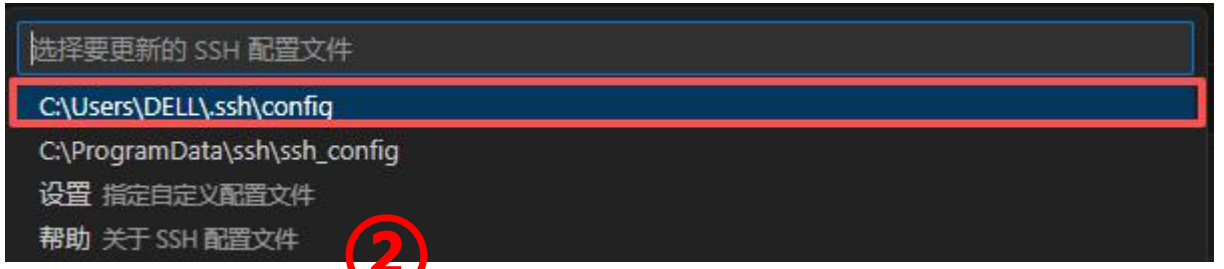


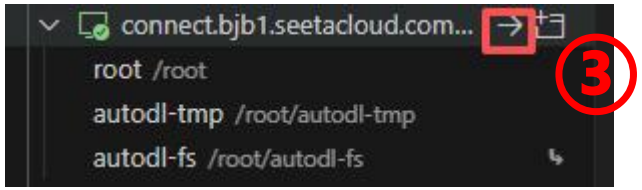
□ AutoDL服务器使用说明（会用则跳过）

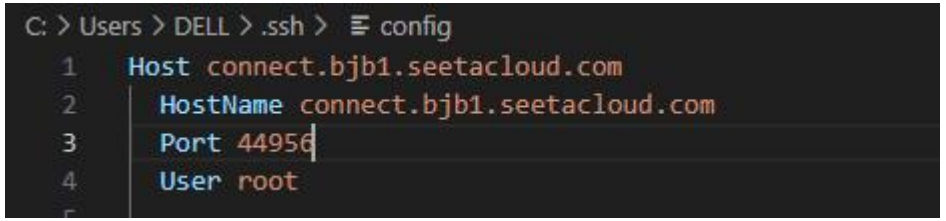
新建远程SSH，输入分发的服务器，



选择保存到ssh的config文件，左边会显示你的服务器，点击右箭头，连接服务器，然后输入密码，一直继续确定，第一次可能要选系统：linux







直接打开文件夹到根目录/root/，然后设置环境

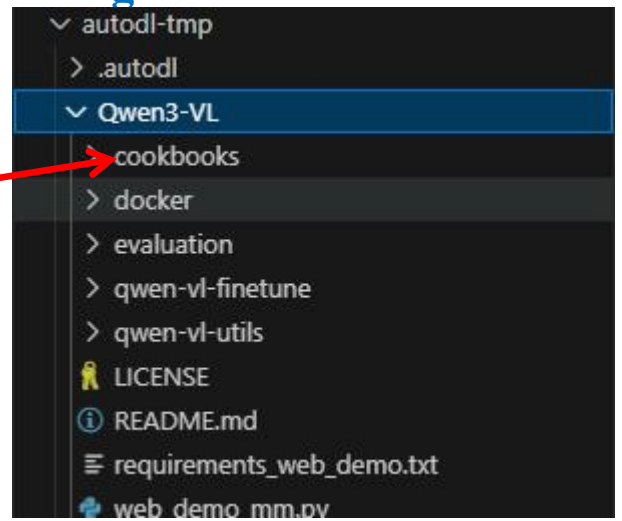
□ 模型环境部署（Qwen为例）

进入数据盘进行环境部署：**cd /root/autodl-tmp**

或者在Github网站本地下载后上传服务器

首先用Git克隆QwenVL的库到服务器：**git clone https://github.com/QwenLM/Qwen3-VL.git**

```
root@autodl-container-74204da04f-37c6c3f9:~/autodl-tmp# git clone https://github.com/QwenLM/Qwen3-VL.git
Cloning into 'Qwen3-VL'...
remote: Enumerating objects: 1086, done.
remote: Counting objects: 100% (139/139), done.
remote: Compressing objects: 100% (56/56), done.
remote: Total 1086 (delta 106), reused 83 (delta 83), pack-reused 947 (from 3)
Receiving objects: 100% (1086/1086), 145.90 MiB | 3.25 MiB/s, done.
Resolving deltas: 100% (477/477), done.
Updating files: 100% (116/116), done.
root@autodl-container-74204da04f-37c6c3f9:~/autodl-tmp#
```



安装依赖库：

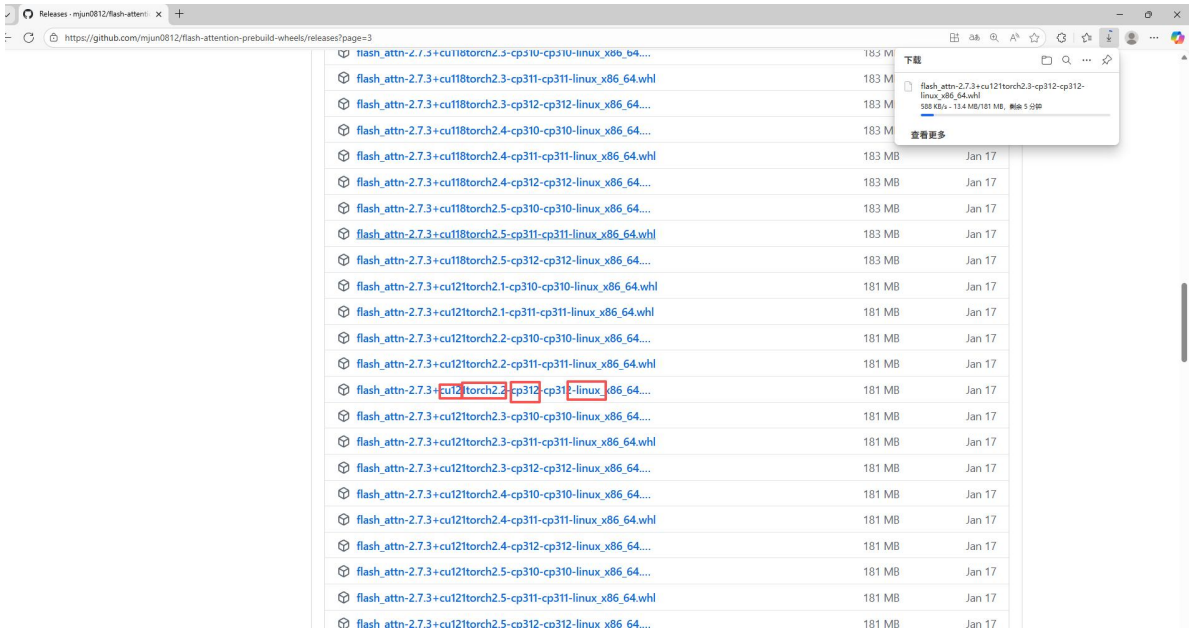
pip install "transformers>=4.57.0"
pip install qwen-vl-utils==0.0.14
pip install -U flash-attn --no-build-isolation
pip install accelerate
pip install ms-swift
git clone https://github.com/modelscope/ms-swift.git
cd ms-swift
pip install -e .

```
root@autodl-container-74204da04f-37c6c3f9:~/autodl-tmp/Qwen3-VL# pip install "transformers>=4.57.0"
Requirement already satisfied: urllib3<3,>=1.21.1 in /root/miniconda3/lib/python3.12/site-packages (from requests->transformers>=4.57.0) (2.1.0)
Requirement already satisfied: certifi>=2017.4.17 in /root/miniconda3/lib/python3.12/site-packages (from requests->transformers>=4.57.0) (2024.2.2)
Installing collected packages: safetensors, regex, hf-xet, huggingface-hub, tokenizers, transformers
Successfully installed hf-xet-1.2.0 huggingface-hub-0.36.0 regex-2025.10.23 safetensors-0.6.2 tokenizers-0.22.1 transformers-4.57.1
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use a virtual environme
nt instead: https://pip.pypa.io/warnings/venv
root@autodl-container-74204da04f-37c6c3f9:~/autodl-tmp/Qwen3-VL# pip install qwen-vl-utils==0.0.14
Looking in indexes: http://mirrors.aliyun.com/pypi/simple
Collecting qwen-vl-utils==0.0.14
  Downloading http://mirrors.aliyun.com/pypi/packages/c4/43/80f67e0336cb2fc725f8e06f7fe35c1d0fe946f4d2b8b2175e797e07349e/qwen_vl_utils-0.0.14-py3-none-any.whl (8.1 kB)
Collecting av (from qwen-vl-utils==0.0.14)
  Downloading http://mirrors.aliyun.com/pypi/packages/b2/7a/1305243ab47f724fdd99ddef7309a594e669af7f0e655e11bdd2c325dfae/av-16.0.1-cp312-cp312-manylinux_2_28_x86_64.whl (40.5 M
B)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 31.8/40.5 MB 233.6 kB/s eta 0:00:38
```

模型环境部署（Qwen为例）

flash_attn如果pip安装失败，可以在本地先下载wheel，再上传到服务器中进行安装（其它环境都可以这样）

Releases · mjun0812/flash-attention-prebuild-wheels



注意：根据自己服务器的系统、cuda、torch和python版本，选择对应的whl文件进行下载

上传后用命令进行安装：

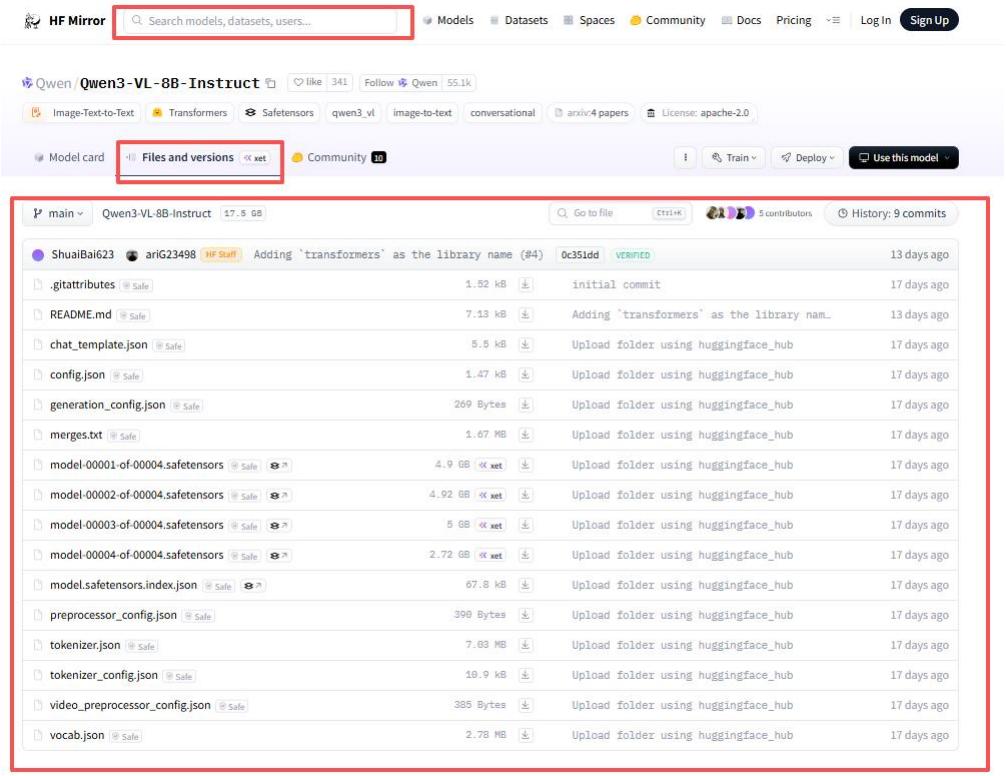
pip install /root/autodl-tmp/flash_attn-2.7.3+cu121torch2.3-cp312-cp312-linux_x86_64.whl --no-build-isolation

```
root@autodl-container-74204da04f-37c6c3f9:~/autodl-tmp/Qwen3-VL# pip install  /root/autodl-tmp/flash_attn-2.7.3+cu121torch2.3-cp312-cp312-linux_x86_64.whl --no-build-isolation
Requirement already satisfied: nvidia-curand-cu12==10.3.2.106 in /root/miniconda3/lib/python3.12/site-packages (from torch->flash-attn==2.7.3+cu121torch2.3) (10.3.2.106)
Requirement already satisfied: nvidia-cusolver-cu12==11.4.5.107 in /root/miniconda3/lib/python3.12/site-packages (from torch->flash-attn==2.7.3+cu121torch2.3) (11.4.5.107)
Requirement already satisfied: nvidia-cusparse-cu12==12.1.0.106 in /root/miniconda3/lib/python3.12/site-packages (from torch->flash-attn==2.7.3+cu121torch2.3) (12.1.0.106)
Requirement already satisfied: nvidia-nccl-cu12==2.20.5 in /root/miniconda3/lib/python3.12/site-packages (from torch->flash-attn==2.7.3+cu121torch2.3) (2.20.5)
Requirement already satisfied: nvidia-nvtx-cu12==12.1.105 in /root/miniconda3/lib/python3.12/site-packages (from torch->flash-attn==2.7.3+cu121torch2.3) (12.1.105)
Requirement already satisfied: nvidia-nvjitlink-cu12 in /root/miniconda3/lib/python3.12/site-packages (from nvidia-cusolver-cu12==11.4.5.107->torch->flash-attn==2.7.3+cu121torc
h2.3) (12.5.40)
Requirement already satisfied: MarkupSafe>=2.0 in /root/miniconda3/lib/python3.12/site-packages (from jinja2->torch->flash-attn==2.7.3+cu121torch2.3) (2.1.5)
Requirement already satisfied: mpmath<1.4.0,>=1.1.0 in /root/miniconda3/lib/python3.12/site-packages (from sympy->torch->flash-attn==2.7.3+cu121torch2.3) (1.3.0)
Installing collected packages: einops, flash-attn
Successfully installed einops-0.8.1 flash-attn-2.7.3
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use a virtual environme
nt instead: https://pip.pypa.io/warnings/venv
```

◘ 下载模型权重（以HF-Mirror)为例

■ 方式一：网页下载

搜索对应模型，选中模型文件页面下载，然后自己上传到服务器中



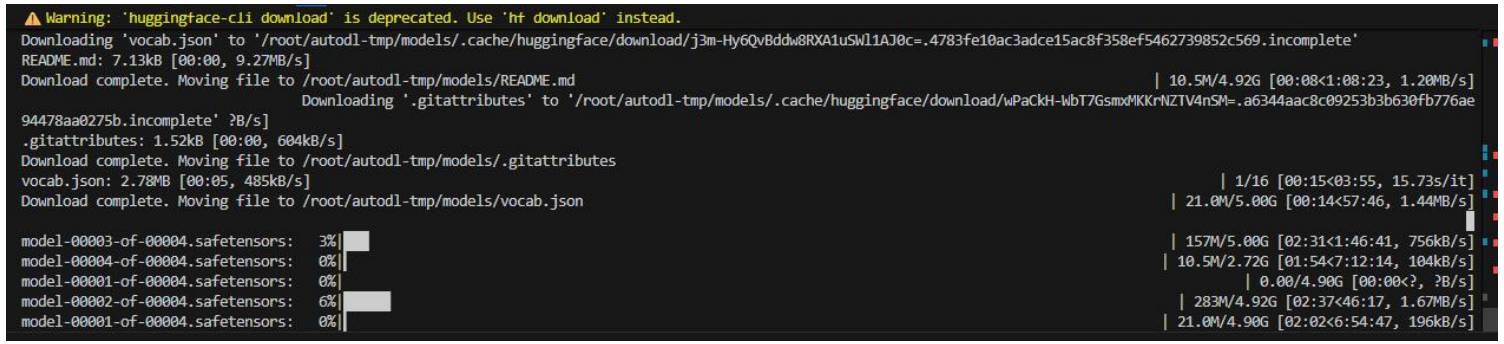
■ 方式二：CLI下载

设置环境：

```
pip install -U huggingface_hub
pip install hf-cli
export HF_ENDPOINT=https://hf-mirror.com
```

下载模型：

```
hf download Qwen/Qwen3-VL-8B-Instruct  --local-dir /root/autodl-tmp/models/
```



参考方法： [(27 封私信 / 6 条消息) 如何快速下载huggingface模型——全方法总结 - 知乎]

□ 单图Qwen3_VL_4B直接测试（zero-shot）

```
messages = [
    {
        "role": "user",
        "content": [
            {
                "type": "image",
                "image": "/root/autodl-tmp/data/real/real_ALB_0003_parallel_3.jpg",
            },
            {"type": "text", "text": query},
        ],
    }
]
```

```
query='''
我会提供你一张证照图像，这张图像上可能存在一些污损，请你帮我分析这张图像上有什么污损？
我对污损的定义是：图像上存在的任何影响证照信息识别的缺陷或损坏,它可能是物理层次的，也有可能是数字层次的。
'''

query='''
我会提供你一张证照图像，这张图像上可能存在一些污损，请你帮我分析这张图像上有什么污损？
我对污损的定义是：图像上存在的任何影响证照信息识别的缺陷或损坏,它可能是物理层次的，也有可能是数字层次的。
请你用json格式回答，回答中包含以下字段：
{
"污损类型": "描述污损的类型，例如划痕、污渍、折痕等",
"污损位置": "描述污损在图像上的具体位置，例如左上角、右下角等",
"污损严重程度": "描述污损的严重程度，例如轻微、中等、严重等",
"判断原因": "描述为什么将该位置判断为污损}
'''
```

[QwenLM/Qwen3-VL: Qwen3-VL is the multimodal large language model series developed by Qwen team, Alibaba Cloud.]

```
export greedy='false'
export seed=3407
export top_p=0.8
export top_k=20
export temperature=0.7
export repetition_penalty=1.0
export presence_penalty=1.5
export out_seq_length=32768
```

```
root@autodl-container-74204da04f-37c6c3f9:~/autodl-tmp/Qwen3-VL# python /root/autodl-tmp/Qwen3-
Loading checkpoint shards: 100%
/root/miniconda3/lib/python3.12/site-packages/torch/nn/modules/conv.py:605: UserWarning: Plan
or Failed cudnn_status: CUDNN_STATUS_NOT_SUPPORTED (Triggered internally at ../aten/src/ATen/na
  return F.conv3d(
['根据对这张阿尔巴尼亚护照图像的分析，存在以下污损：\n\n1.  **数字水印/标记覆盖**：图像上覆盖着大量“数据溯源码”和“SPECIMEN”字样。这些是数字水印，用于标识该护照为样本或复印件，而非真实证件。它们虽然不是物理损坏，但严重干扰了信息的阅读和识别，属于数字层次的污损。\n\n2.  **信息遮挡**：红色的“SPECIMEN”字样以大号字体斜向覆盖在护照的多个关键信息区域，包括姓名、出生日期、签发']
root@autodl-container-74204da04f-37c6c3f9:~/autodl-tmp/Qwen3-VL# python /root/autodl-tmp/Qwen3-VL/demo.py
Loading checkpoint shards: 100%| 2/2 [00:01<00:00,  1.00it/s]
/root/miniconda3/lib/python3.12/site-packages/torch/nn/modules/conv.py:605: UserWarning: Plan failed with a cudnnException: CUDNN_BACKEND_EXECUTION_PLAN_DESCRIPTOR: cudnnFinalize Descriptor
or Failed cudnn_status: CUDNN_STATUS_NOT_SUPPORTED (Triggered internally at ../aten/src/ATen/native/cudnn/Conv_v8.cpp:919.)
  return F.conv3d(
['{\n  "污损类型": "水印",\n  "污损位置": "整个图像",\n  "污损严重程度": "中等",\n  "判断原因": "图像上覆盖有大量重复的“数据溯源码”水印，这些水印干扰了证照信息的视觉识别，属于数字层次的污损，影响了信息的清晰度和可读性。"\n}']
```

□ 单图Qwen3_VL_4B直接测试（zero-shot）


```
query='''
我会提供你一张证照图像，这张图像上可能存在一些污损，请你帮我分析这张图像上有什么污损？
我对污损的定义是：图像上存在的任何影响证照信息识别的缺陷或损坏,它可能是物理层次的，也有可能是数字层次的。
这张图像上可能不止一种污损。
'''
```

```
root@autodl-container-74204da04f-37c6c3f9:~/autodl-tmp/Qwen3-VL# python /root/autodl-tmp/Qwen3-VL/demo.py
Loading checkpoint shards: 100%|| 2/2 [00:02<00:00,  1.04s/it]
/root/miniconda3/lib/python3.12/site-packages/torch/nn/modules/conv.py:605: UserWarning: Plan failed with a cudnnException: CUDNN_BACKEND_EXECUTION_PLAN_DESCRIPTOR: cudnnFinalize Descript
or Failed cudnn_status: CUDNN_STATUS_NOT_SUPPORTED (Triggered internally at ../aten/src/ATen/native/cudnn/Conv_v8.cpp:919.)
  return F.conv3d(
['```json\n[\n    {\n        "污损类型": "污渍",\n        "污损位置": "中央偏右",\n        "污损严重程度": "严重",\n        "判断原因": "图像中央偏右区域有一大块黄色液体状污渍，覆盖了部分
姓名、民族、出生日期、住址和公民身份号码等关键信息，严重影响了信息的可读性和识别度。"\n    }\n]\n```']
```

◘ 根据任务准备数据集（参考Swift技术文档）

- 数据：使用手上已有数据，或自行将图像进行篡改，并上传到服务器
- 标注文本：
  - 自行人工标注：“这张证件图像的姓名遭到篡改”“这张证件上覆盖了一些水滴”
  - 自动化标注：“真实图像”/“篡改图像”“污损图像”/“干净图像”
  - 大模型自动标注：“。。。。”

Swift中文技术文档：[自定义数据集 — swift 3.10.0.dev0 文档]

- 标注微调json格式（VQA任务）：

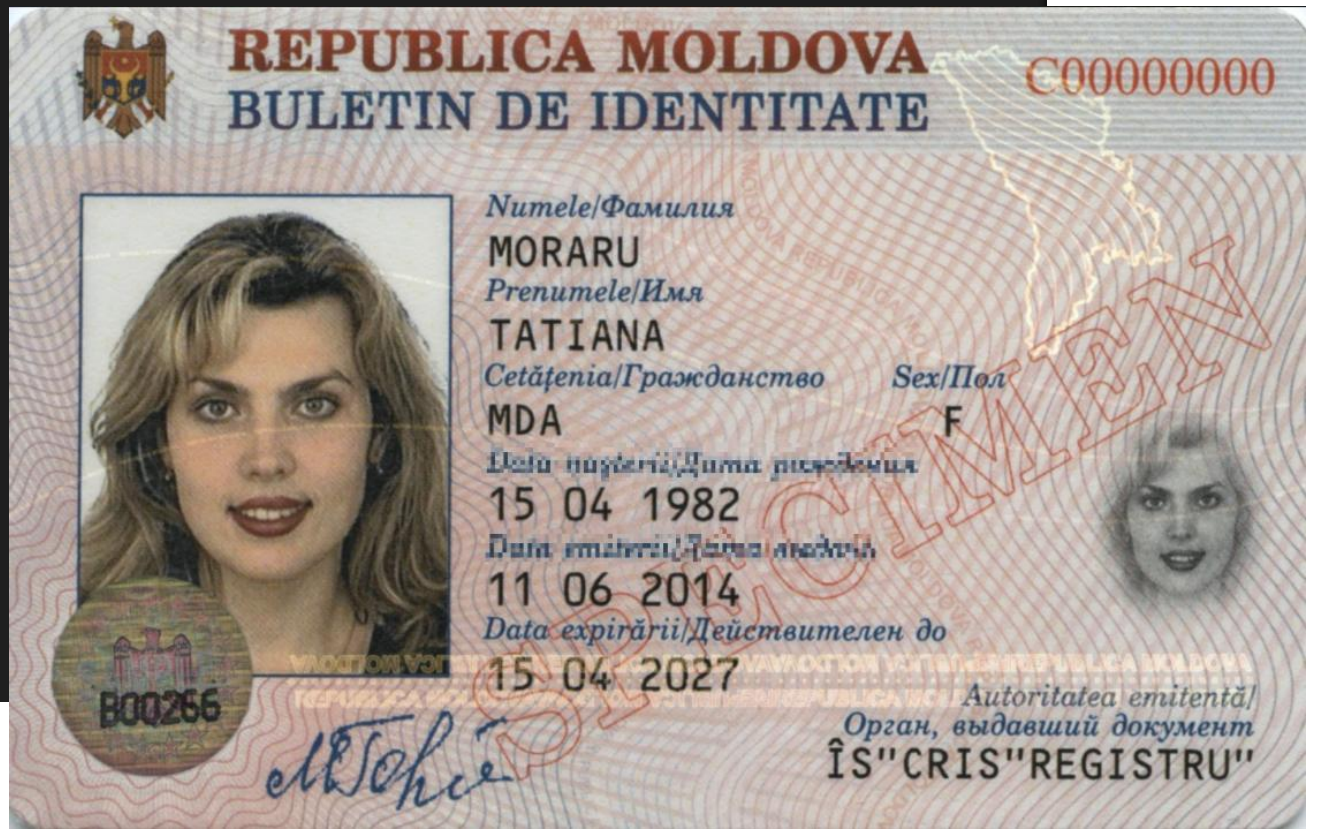
{"messages": [{"role": "user", "content": "浙江的省会在哪？"}, {"role": "assistant", "content": "浙江的省会在杭州。"}]}
{"messages": [{"role": "user", "content": "<image><image>两张图片有什么区别"}, {"role": "assistant", "content": "前一张是小猫，后一张是小狗"}], "images": ["/xxx/x.jpg", "/xxx/x.png"]}
{"messages": [{"role": "user", "content": "<audio>语音说了什么"}, {"role": "assistant", "content": "今天天气真好呀"}], "audios": ["/xxx/x.mp3"]}
{"messages": [{"role": "system", "content": "你是个有用无害的助手"}, {"role": "user", "content": "<image>图片中是什么，<video>视频中是什么"}, {"role": "assistant", "content": "图片中是一个大象，视频中是一只小狗在草地上奔跑"}], "images": ["/xxx/x.jpg"], "videos": ["/xxx/x.mp4"]}

- 标注微调json格式（物体检测任务，需要对应bbox）：

{"messages": [{"role": "system", "content": "You are a helpful assistant."}, {"role": "user", "content": "<image>描述图像"}, {"role": "assistant", "content": "<|object_ref_start|>一只狗<|object_ref_end|><|box_start|>(221,423),(569,886)<|box_end|>和<|object_ref_start|>一个女人<|object_ref_end|><|box_start|>(451,381),(733,793)<|box_end|>正在沙滩上玩耍"}], "images": ["/xxx/x.jpg"]}
{"messages": [{"role": "system", "content": "You are a helpful assistant."}, {"role": "user", "content": "<image>找到图像中的<|object_ref_start|>羊<|object_ref_end|>"}, {"role": "assistant", "content": "<|box_start|>(101,201),(150,266)<|box_end|><|box_start|>(401,601),(550,666)<|box_end|>"}], "images": ["/xxx/x.jpg"]}
{"messages": [{"role": "system", "content": "You are a helpful assistant."}, {"role": "user", "content": "<image>帮我打开谷歌浏览器"}, {"role": "assistant", "content": "Action: click(start_box='<|box_start|>(246,113)<|box_end|>')"}], "images": ["/xxx/x.jpg"]}

微调数据集准备（参考Swift技术文档）

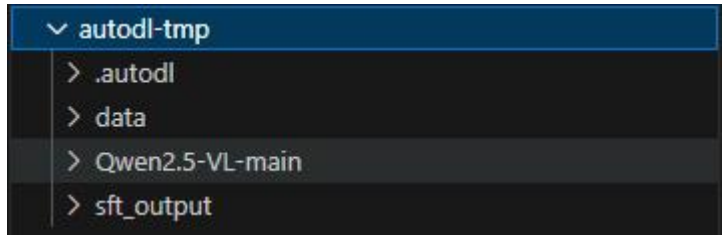

```
{
  "messages": [
    {
      "role": "user",
      "content": "<image>Is there any defect or damage in this document image that would affect the recognition of the document information? Highlight where it is.<ref-object>
    },
    {
      "role": "assistant",
      "content": "mosaic <bbox> <bbox>"
    }
  ],
  "images": [
    "/root/autodl-tmp/data/real/real_MDA_0008_single_2.jpg"
  ],
  "objects": {
    "ref": "mosaic",
    "bbox": [
      [
        1122,
        1233,
        2055,
        1312
      ],
      [
        1125,
        1030,
        2157,
        1122
      ]
    ]
  },
  "label": 1
},
```

微调模型CLI方法（参考Swift技术文档）

所有的环境，包括Swift都已经设置好了，只需要大家新建一个.sh文件，几行代码运行即可

- 在autodl-tmp中操作，新建data、sft_output、script文件夹，新建一个.sh文件




```
CUDA_VISIBLE_DEVICES=0 \
swift sft \
    --model Qwen/Qwen2.5-7B-Instruct \
    --train_type lora \
    --dataset 'AI-ModelScope/alpaca-gpt4-data-zh#500' \
              'AI-ModelScope/alpaca-gpt4-data-en#500' \
              'swift/self-cognition#500' \
    --torch_dtype bfloat16 \
    --num_train_epochs 1 \
    --per_device_train_batch_size 1 \
    --per_device_eval_batch_size 1 \
    --learning_rate 1e-4 \
    --lora_rank 8 \
    --lora_alpha 32 \
    --target_modules all-linear \
    --gradient_accumulation_steps 16 \
    --eval_steps 50 \
    --save_steps 50 \
    --save_total_limit 2 \
    --logging_steps 5 \
    --max_length 2048 \
    --output_dir output \
    --system 'You are a helpful assistant.' \
    --warmup_ratio 0.05 \
    --dataloader_num_workers 4 \
```

- .sh文件中敲入右边命令，终端命令行输入： bash <你sh文件的路径>
  - 如果要用其它模型训练，只需要修改 --model 对应模型权重路径即可
  - output_dir 输入sft_output对应绝对路径
  - dataset 输入你的json文件路径
  - num_train_epochs 设置你需要训练的epoch
  - 其余参数可以在技术文档中搜索
  - 添加--split_dataset_ratio 0.01 ，可以自动帮你采样1%的数据，用于后续测试验证

Swift中文技术文档： 快速开始 — swift 3.10.0.dev0 文档　　Qwen3-VL最佳实践 — swift 3.10.0.dev0 文档

□ 训练示例

```
export PYTORCH_CUDA_ALLOC_CONF='expandable_segments:True'
export IMAGE_MAX_TOKEN_NUM=1024
swift sft \
    --max_pixels 3010560 \
    --model /root/autodl-tmp/models/Qwen3-VL-4B-Instruct \
    --dataset /root/autodl-tmp/data/train_defacement_bbox.json \
    --load_from_cache_file true \
    --split_dataset_ratio 0.01 \
    --train_type lora \
    --torch_dtype bfloat16 \
    --num_train_epochs 3 \
    --per_device_train_batch_size 4 \
    --per_device_eval_batch_size 1 \
    --attn_impl flash_attn \
    --padding_free true \
    --learning_rate 1e-4 \
```

```
root@autodl-container-74204da04f-37c6c3f9:~/autodl-tmp# bash /root/autodl-tmp/train/sft.sh
The tokenizer has new PAD/BOS/EOS tokens that differ from the model config and generation config. The model config and generation config were aligned accordingly, being updated with the tokenizer's values. Update
 tokens: {'eos_token_id': 151645, 'bos_token_id': None}.
Train:   0%|                                                                                          | 0/267 [00:00<?, ?it/s
[INFO:swift] use_logits_to_keep: False
/root/miniconda3/lib/python3.12/site-packages/torch/nn/modules/conv.py:605: UserWarning: Plan failed with a cudnnException: CUDNN_BACKEND_EXECUTION_PLAN_DESCRIPTOR: cudnnFinalize Descriptor Failed cudnn_status: C
DNN_STATUS_NOT_SUPPORTED (Triggered internally at ../aten/src/ATen/native/cudnn/Conv_v8.cpp:919.)
  return F.conv3d(
{'loss': 3.21068788, 'grad_norm': 4.78407001, 'learning_rate': 7.14e-06, 'token_acc': 0.52847682, 'epoch': 0.01, 'global_step/max_steps': '1/267', 'percentage': '0.37%', 'elapsed_time': '24s', 'remaining_time': '
h 46m 56s', 'memory(GiB)': 39.43, 'train_speed(iter/s)': 0.041455}
{'loss': 2.35159707, 'grad_norm': 2.79926133, 'learning_rate': 3.571e-05, 'token_acc': 0.54493054, 'epoch': 0.06, 'global_step/max_steps': '5/267', 'percentage': '1.87%', 'elapsed_time': '1m 11s', 'remaining_time
: '1h 2m 39s', 'memory(GiB)': 39.44, 'train_speed(iter/s)': 0.069682}
{'loss': 2.33112888, 'grad_norm': 3.38580966, 'learning_rate': 7.143e-05, 'token_acc': 0.57258192, 'epoch': 0.11, 'global_step/max_steps': '10/267', 'percentage': '3.75%', 'elapsed_time': '2m 9s', 'remaining_time
: '55m 28s', 'memory(GiB)': 39.44, 'train_speed(iter/s)': 0.077206}
{'loss': 1.61835117, 'grad_norm': 4.07914925, 'learning_rate': 0.0001, 'token_acc': 0.61937331, 'epoch': 0.17, 'global_step/max_steps': '15/267', 'percentage': '5.62%', 'elapsed_time': '3m 7s', 'remaining_time':
52m 29s', 'memory(GiB)': 39.44, 'train_speed(iter/s)': 0.08001}
{'loss': 0.95367327, 'grad_norm': 2.13847256, 'learning_rate': 9.986e-05, 'token_acc': 0.70635401, 'epoch': 0.23, 'global_step/max_steps': '20/267', 'percentage': '7.49%', 'elapsed_time': '4m 5s', 'remaining_time
: '50m 34s', 'memory(GiB)': 39.44, 'train_speed(iter/s)': 0.081394}
Train:   7%|██████                                                                            | 20/267 [04:05<47:55, 11.64s/it
```

```
    --dataset_num_proc 4 \
    --dataloader_num_workers 4
```

◘ 测试微调后的模型（参考Swift技术文档）

在训练完成后，log中会输出你训练的最后模型存储位置以及最佳模型存储位置（lora）
然后同样是新建一个sh文件，按照命令行进行推理

```
root@autodl-container-74204da04f-37c6c3f9:~/autodl-tmp# bash /root/autodl-tmp/train/sft.sh
e': '1m 22s', 'memory(GiB)': 40.92, 'train_speed(iter/s)': 0.085001}
{'loss': 0.33454185, 'grad_norm': 3.38164449, 'learning_rate': 2e-08, 'token_acc': 0.86678733, 'epoch': 2.98, 'global_step/max_steps': '265/267', 'percentage': '99.25%', 'elapsed_time': '51m 57s', 'remaining_time'
: '23s', 'memory(GiB)': 40.92, 'train_speed(iter/s)': 0.085006}
Train: 100%| | 267/267 [52:12<00:00,  9.13s/it]
{'eval_loss': 0.31385016, 'eval_runtime': 4.8868, 'eval_samples_per_second': 1.432, 'eval_steps_per_second': 1.432, 'eval_token_acc': 0.85251492, 'epoch': 3.0, 'global_step/max_steps': '267/267', 'percentage': '10
0.00%', 'elapsed_time': '52m 16s', 'remaining_time': '0s', 'memory(GiB)': 40.92, 'train_speed(iter/s)': 0.085115}
Val: 100%| | 7/7 [00:03<00:00,  2.27it/s]
[INFO:swift] Saving model checkpoint to /root/autodl-tmp/models/sft_4B_qwenvl/v1-20251029-155338/checkpoint-267
{'train_runtime': 3138.151, 'train_samples_per_second': 0.675, 'train_steps_per_second': 0.085, 'train_loss': 0.50032763, 'epoch': 3.0, 'global_step/max_steps': '267/267', 'percentage': '100.00%', 'elapsed_time':
'52m 18s', 'remaining_time': '0s', 'memory(GiB)': 40.92, 'train_speed(iter/s)': 0.085082}
Train: 100%| | 267/267 [52:18<00:00, 11.75s/it]
```

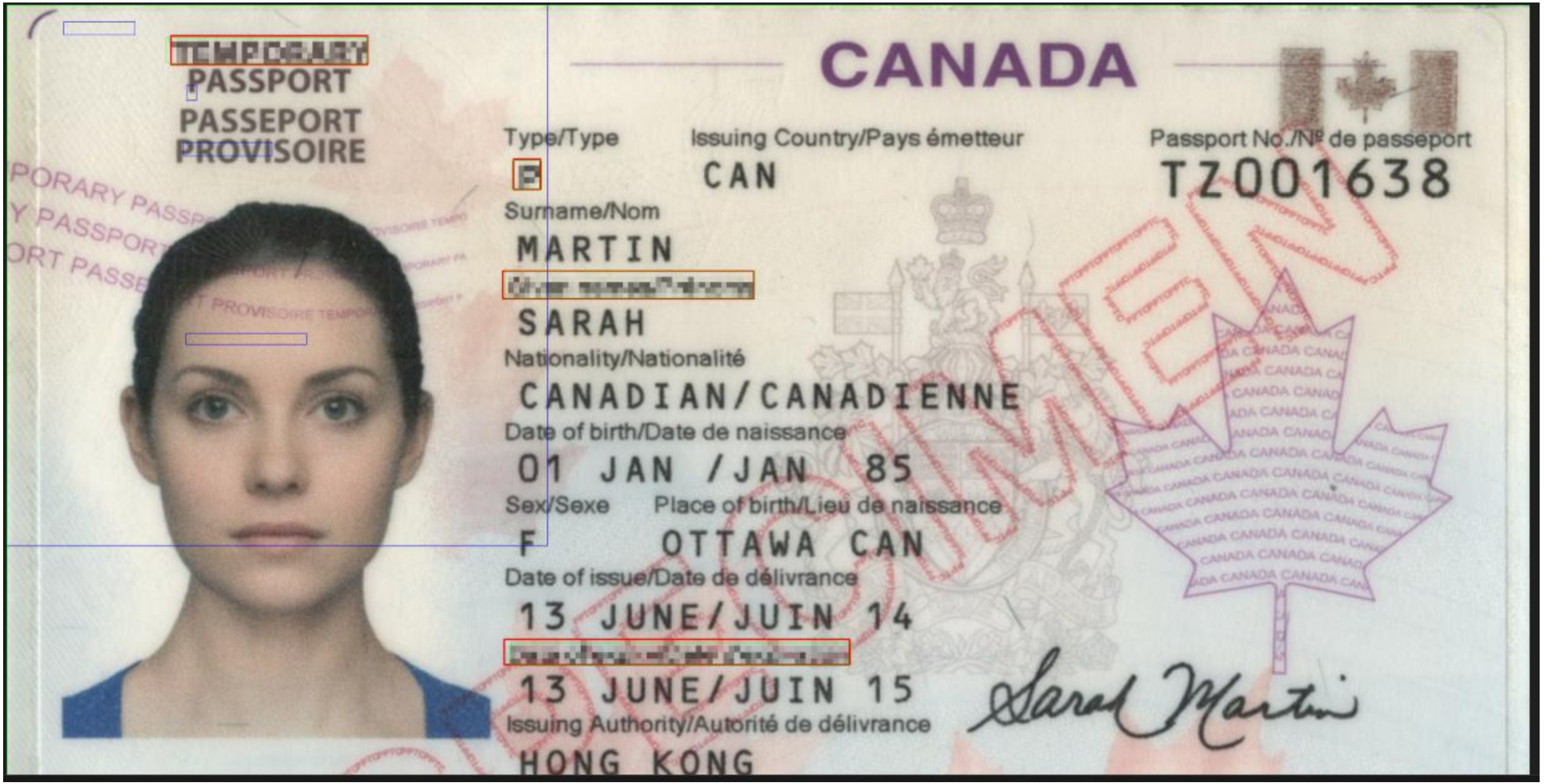
trans.py | sft.sh | demo.py U | test.sh

train > test.sh

```
export PYTORCH_CUDA_ALLOC_CONF='expandable_segments:True'
export CUDA_VISIBLE_DEVICES=0
export IMAGE_MAX_TOKEN_NUM=1024
export FPS_MAX_FRAMES=16
swift infer \
    --max_pixels 3010560 \
    --adapters /root/autodl-tmp/models/sft_4B_qwenvl/v1-20251029-155338/checkpoint-267 \
    --stream true \
    --max_new_tokens 2048 \
    --load_data_args true \
    --val_dataset /root/autodl-tmp/models/sft_4B_qwenvl/v1-20251029-155338/val_dataset.jsonl \
    --result_path /root/autodl-tmp/models/sft_4B_qwenvl/v1-20251029-155338/test_output_1029_1.jsonl
```

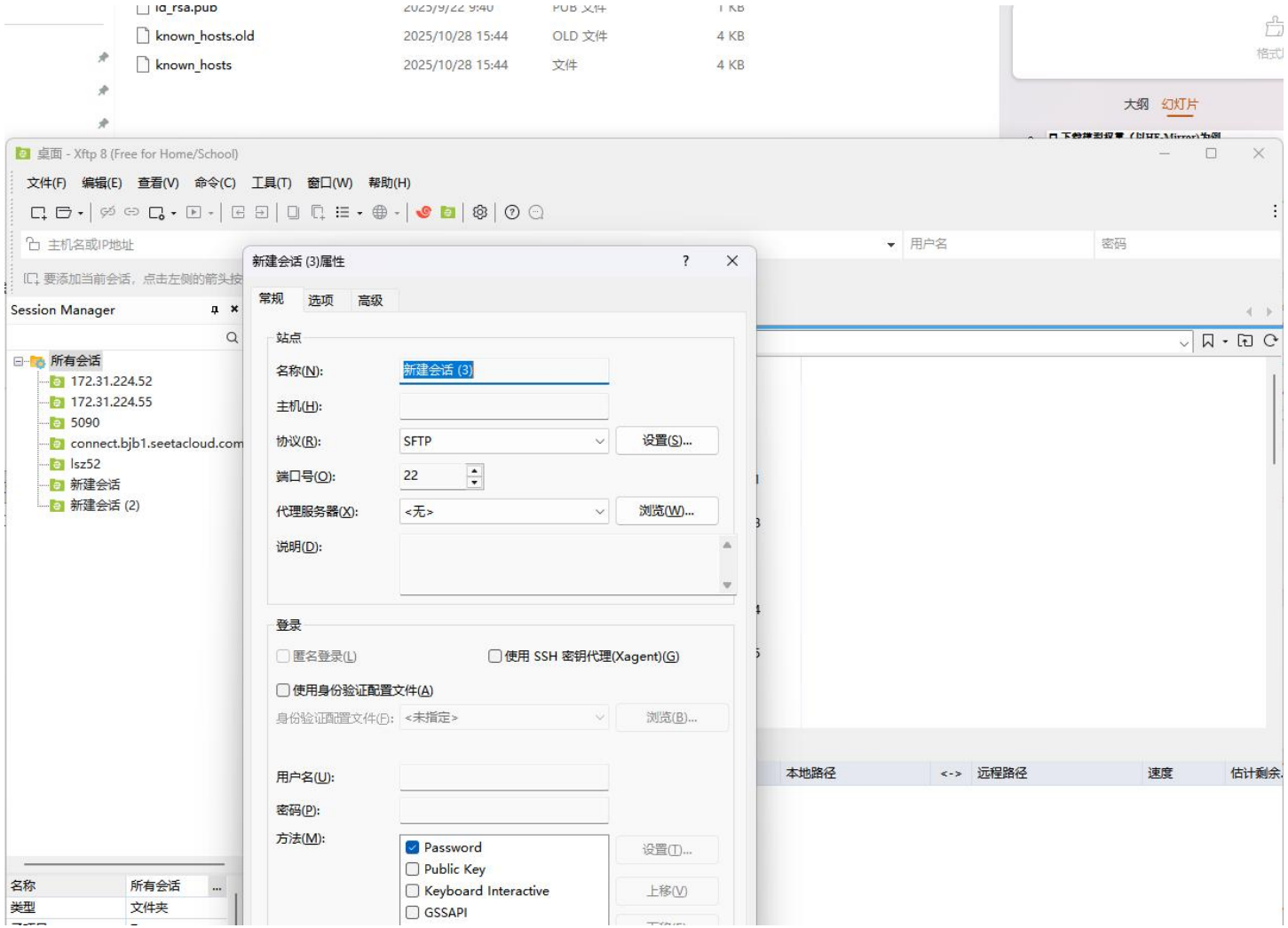
```
the recognition of the document information? Highlight where it is.<ref-object>

ng: Plan failed with a cudnnException: CUDNN_BACKEND_EXECUTION_PLAN_DESCRIPTOR: cudnnFinalize Descriptor Failed cudnn_status: CU
v8.cpp:919.)

x_start|>(283,336),(357,356)<|box_end|> <|box_start|>(286,500),(304,525)<|box_end|> <|box_start|>(0,751),(211,951)<|box_end|>

the recognition of the document information? Highlight where it is.<ref-object>

|> <|box_start|>(827,805),(987,867)<|box_end|>

the recognition of the document information? Highlight where it is.<ref-object>

[RESPONSE] Clean. <bbox>
--------------------------------------------------
[QUERY] <image>Is there any defect or damage in this document image that would affect the recognition of the document information? Highlight where it is.<ref-object>
[LABELS] Clean. <bbox>
[RESPONSE] Clean. <bbox>
--------------------------------------------------
[QUERY] <image>Is there any defect or damage in this document image that would affect the recognition of the document information? Highlight where it is.<ref-object>
[LABELS] dense_watermark <bbox>
[RESPONSE] dense_watermark <|box_start|>(0,0),(1000,1000)<|box_end|>
```

Swift中文技术文档： 快速开始 — swift 3.10.0.dev0 文档



蓝色：未进行坐标缩放的预测框　　红色：GT的bbox　　绿色：进行缩放后的预测结果

Xftp本地-服务器传输文件



点击文件，新建，输入信息，连接
主机：connect.bjb1.seetacloud.com
端口号：Port对应端口
用户名：root
密码：对应密码