

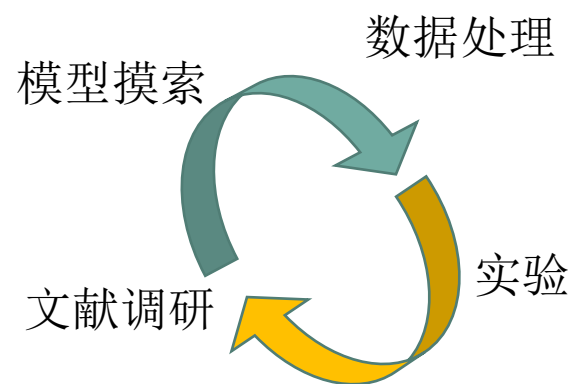
如何提高 炼丹效率？

深度学习工程向

A structural view of improving
DL experiments



系统非常稳定，
所有代码不要随便动



我他妈直接肯定

有条理的计划+事半功倍的标准化工具与框架

```
DomainPromptTuning.py
Project
├── dataprocessing
├── Hybrid
├── MIL
├── PromptModels
├── pytorch_grad_cam
├── utils
├── 5fold_test.py
├── DomainPromptTuning.py
├── MIL_test.py
├── MIL_train.py
├── PreTrain.py
├── Test.py
├── Train.py
├── External Libraries
├── Scratches and Consoles
├── Scratches
│   ├── scratch.gitignore
│   ├── scratch.py
│   ├── scratch_1.py
│   └── scratch_2.py
└── Run: DomainPromptTuning x

90 parser.add_argument('--log_dir', default='/root/tf-logs/runs',
91                     help='path where to tensorboard log')
92 parser.add_argument('--device', default='cuda',
93                     help='device to use for training / testing')
94 parser.add_argument('--seed', default=42, type=int) # ori 0 不过应该开
95
96 # dataloader setting
97 parser.add_argument('--num_workers', default=32, type=int) # Ori 10,
98 parser.add_argument('--pin_mem', action='store_true',
99                     help='Pin CPU memory in DataLoader for more effici
100 parser.add_argument('--no_pin_mem', action='store_false', dest='pin_me
101 parser.set_defaults(pin_mem=True)
102
103 # distributed training parameters
104 parser.add_argument('--world_size', default=1, type=int,
105                     help='number of distributed processes')
106 parser.add_argument('--local_rank', default=-1, type=int)
107 parser.add_argument('--dist_on_itp', action='store_true')
108 parser.add_argument('--dist_url', default='env://',
109                     help='url used to set up distributed training')
110
111 return parser
112
113 get_args_parser()
```

```
Run: DomainPromptTuning x
Epoch: [0] [1120/1251] eta: 0:02:55 lr: 0.000269 loss: 2.7949 (4.6344) time: 1.2726 data: 0.0005 max mem: 70026
Epoch: [0] [1140/1251] eta: 0:02:28 lr: 0.000273 loss: 2.7297 (4.6014) time: 1.2140 data: 0.0005 max mem: 70026
Epoch: [0] [1160/1251] eta: 0:02:01 lr: 0.000278 loss: 2.7281 (4.5690) time: 1.2220 data: 0.0005 max mem: 70026
Epoch: [0] [1180/1251] eta: 0:01:34 lr: 0.000283 loss: 2.7227 (4.5380) time: 1.2769 data: 0.0005 max mem: 70026
Epoch: [0] [1200/1251] eta: 0:01:08 lr: 0.000288 loss: 2.7254 (4.5079) time: 1.2114 data: 0.0009 max mem: 70026
Epoch: [0] [1220/1251] eta: 0:00:41 lr: 0.000293 loss: 2.6696 (4.4780) time: 1.0407 data: 0.0001 max mem: 70026
Epoch: [0] [1240/1251] eta: 0:00:14 lr: 0.000297 loss: 2.7021 (4.4497) time: 1.0460 data: 0.0001 max mem: 70026
Epoch: [0] [1250/1251] eta: 0:00:01 lr: 0.000300 loss: 2.6996 (4.4355) time: 1.0550 data: 0.0001 max mem: 70026
Epoch: [0] Total time: 0:27:36 (1.3238 s / it)
Averaged stats: lr: 0.000300 loss: 2.6996 (4.4355)
log_dir: /root/tf-logs/runs
Epoch: [1] [0/1251] eta: 8:25:43 lr: 0.000300 loss: 2.7199 (2.7199) time: 24.2552 data: 23.1448 max mem: 70026
```

CPU : 210 核心
内存 : 1680 GB
GPU : NVIDIA A100-SXM4-80GB, 8
存储:
系统盘 / : 14% 2.7G/20G
数据盘 /root/autodl-tmp: 73% 146G/200G
网盘 /root/autodl-nas: 0% 0/20G

*注意:
1. 系统盘较小请将大的数据存放于数据盘或网盘中, 重置系统时数据盘和网盘中的数据不

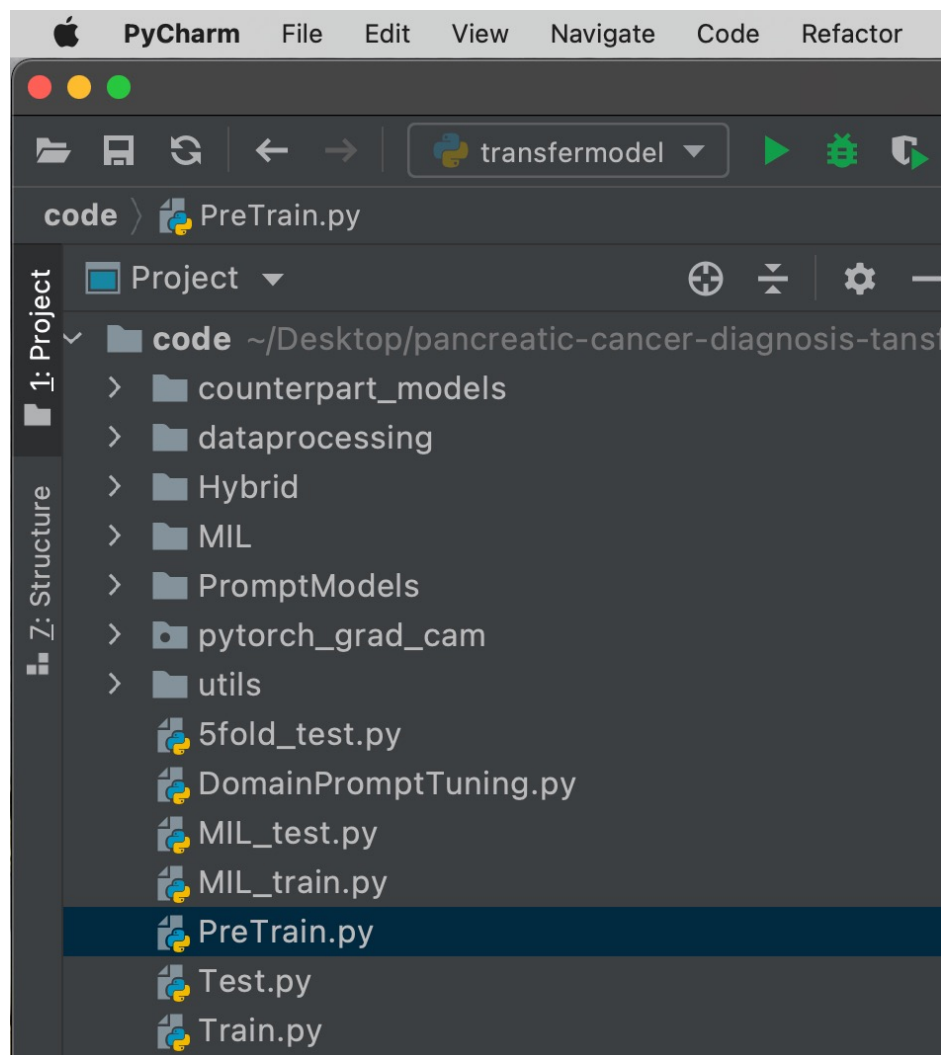
NVIDIA-SMI 510.47.03		Driver Version: 510.47.03		CUDA Version: 11.6	
GPU	Name	Persistence-M	Bus-Id	Disp.A	Volatile Uncorr. ECC
Fan	Temp	Perf	Pwr:Usage/Cap	Memory-Usage	GPU-Util Compute M. MIG M.
0	NVIDIA A100-SXM...	On	00000000:07:00.0	Off	0
N/A	55C	P0	337W / 400W	77525MiB / 81920MiB	100% Default Disabled
1	NVIDIA A100-SXM...	On	00000000:0A:00.0	Off	0
N/A	45C	P0	290W / 400W	74479MiB / 81920MiB	100% Default Disabled
2	NVIDIA A100-SXM...	On	00000000:45:00.0	Off	0
N/A	47C	P0	324W / 400W	74479MiB / 81920MiB	95% Default Disabled
3	NVIDIA A100-SXM...	On	00000000:4B:00.0	Off	0
N/A	56C	P0	307W / 400W	74537MiB / 81920MiB	100% Default Disabled
4	NVIDIA A100-SXM...	On	00000000:83:00.0	Off	0
N/A	53C	P0	334W / 400W	74537MiB / 81920MiB	100% Default Disabled
5	NVIDIA A100-SXM...	On	00000000:89:00.0	Off	0
N/A	45C	P0	316W / 400W	74537MiB / 81920MiB	100% Default Disabled
6	NVIDIA A100-SXM...	On	00000000:C0:00.0	Off	0
N/A	44C	P0	321W / 400W	74479MiB / 81920MiB	94% Default Disabled
7	NVIDIA A100-SXM...	On	00000000:C3:00.0	Off	0
N/A	50C	P0	102W / 400W	74411MiB / 81920MiB	83% Default Disabled

Processes:						
GPU	GI ID	CI ID	PID	Type	Process name	GPU Mem Usage
0	N/A	N/A	654429	C		
1	N/A	N/A	654429	C		
2	N/A	N/A	654429	C		
3	N/A	N/A	654429	C		
4	N/A	N/A	654429	C		
5	N/A	N/A	654429	C		
6	N/A	N/A	654429	C		
7	N/A	N/A	654429	C		

POINTS TO DECLEAR

1. Codings 代码工作效率
2. Experiment design and structure 实验系统的高效设计
3. Management tools 辅助系统的灵活使用

Codings 代码工作效率



体系化的设计代码，让不同的项目可以共用一系列框架

正确设计参数变量，灵活进行不同的设置的改变

与服务器资源相匹配的控制变量要单独注意，
这些都是方便在不同环境部署的关键

自己整理一个遇到的bug的手册，这个过程是最好的学习
(在崩溃中成长)

Experiment design and structure 实验系统的高效设计

体系化的实验脚本，让不同的实验一目了然，条件清晰准确

体系化服务器系统，让你的工作能够多机器并行开展

整理一系列实验模版，方便和其他人合作

自动化log整理

Experiment design and structure 实验系统的高效设计

体系化的设计代码，让不同的项目可以共用一系列框架

与服务器资源相匹配的控制变量要单独注意，这些都是方便在不同环境部署的关键

自己整理一个遇到的bug的手册，这个过程是最好的学习（在崩溃中成长）

Colab实验平台

整洁高校的实验记录，输出备份

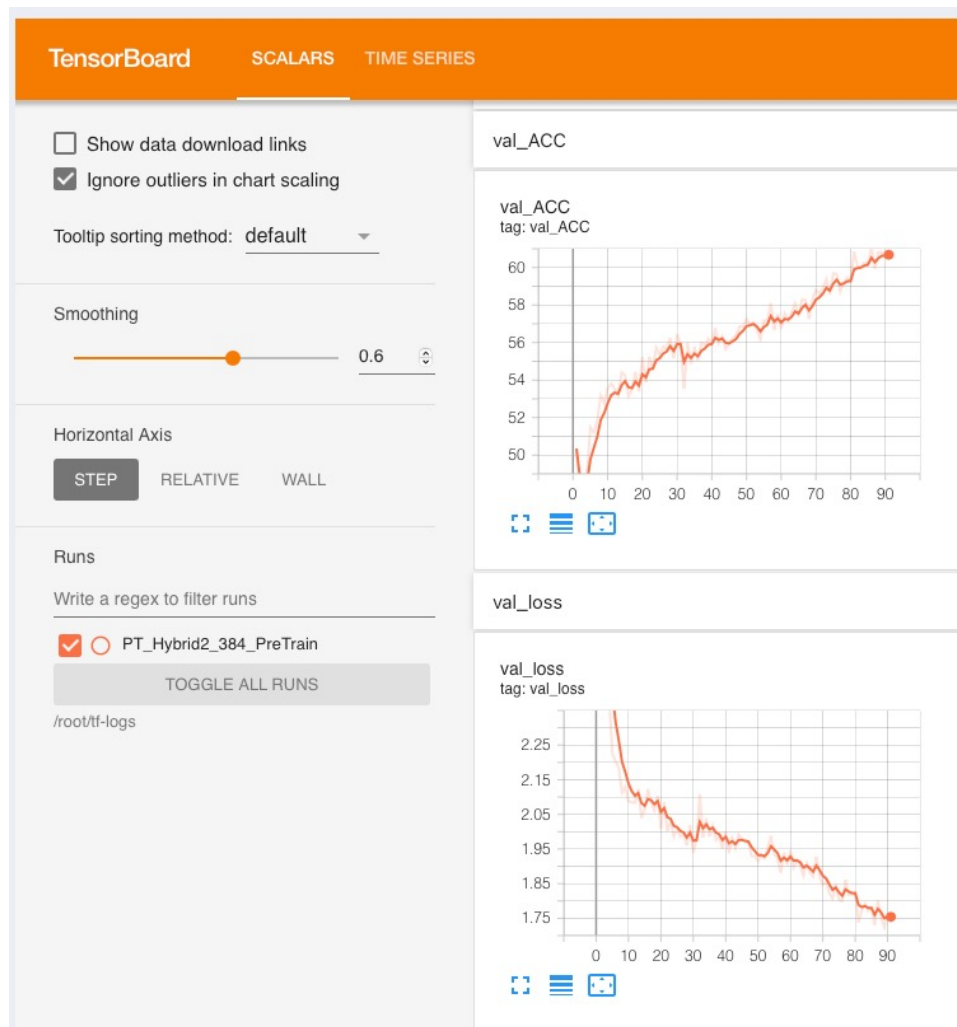
复现性与严谨的记录性质很适合论文

适合多人合作

老dorcker了，适合带新同学



Management tools 辅助系统的灵活使用



Tensorboard系统

自动记录日志的工具

自动状态监控与文件记录

```
notify.Reboost(mail_host='smtp.163.com', mail_user='xxxx@163.com', mail_pass='xxxx',  
               default_reciving_list=['xxxxxx@163.com'], # to use notify  
               log_root_path='log', max_log_cnt=5)
```

```
# notify.add_file(draw_path) # 自动将draw_path打包为
```

```
notify.add_text(' ')
```

```
notify.add_text('models idx ' + str(model_idx))  
notify.add_text(' ')
```

```
notify.add_text('GPU idx: ' + str(gpu_idx))  
notify.add_text(' ')
```

```
notify.add_text('classes number ' + str(num_classes))  
notify.add_text('edge size ' + str(edge_size))  
notify.add_text('batch_size on each GPU:' + str(batch_size))  
notify.add_text('number of GPUs:' + str(torch.cuda.device_count()))  
notify.add_text('num_epochs ' + str(num_epochs))  
notify.add_text('lr ' + str(lr))  
notify.add_text('opt_name ' + str(opt_name))  
notify.add_text('enable_sam ' + str(enable_sam))  
notify.send_log()
```

NOTIFYEMAIL的使用

15:15



[7876ab523a59 LOG]
Train__2022_04_14-11_21_55_log



t. ■■■■ 1/4
Sent to t. ■ ▾



```
start time: 2022_04_14 11:21:55
end time: 2022_04_14 12:37:00
source: 7876ab523a59
=====
```

update to the tensorboard

```
model_idx
ViT_224_401_Prompt_Deep_ROSE_lf25_b8
```

GPU idx: -1

```
cls number 2
class_names=['Negative', 'Positive']
edge size 224
```

```
batch_size 8
num_epochs 50
lr 1e-05
opt_name Adam
```

```
enable_sam False
augmentation_name None
data_augmentation_mode 0
```

Attachment(2)

Train__2022_04_14-11_2...5_log.rtf



115K ▾

server_status.rtf



1.6K ▾

↩ Reply

➡ Forward



15:16



Done Train__2022_04_14-11_21_55...

Positive TP: 1071.0
Positive TN: 2195.0
Positive FP: 125.0
Positive FN: 169.0

Epoch: 50 val index of 50 minibatch: 1 time used:
10.50404167175293
minibatch AVG loss: 0.2115384164080024

```
Epoch: 50 val
Loss: 0.2478 Acc: 88.9980
Negative precision: 93.4985 recall: 90.9639
Negative sensitivity: 90.9639 specificity: 87.7907
Negative FPR: 12.2093 NPV: 83.4254
Negative TP: 302.0
Negative TN: 151.0
Negative FP: 21.0
Negative FN: 30.0
Positive precision: 83.4254 recall: 87.7907
Positive sensitivity: 87.7907 specificity: 90.9639
Positive FPR: 9.0361 NPV: 93.4985
Positive TP: 151.0
Positive TN: 302.0
Positive FP: 30.0
Positive FN: 21.0
```

Training complete in 74m 47s
Best epoch idx: 44
Best epoch train Acc: 92.085321
Best epoch val Acc: 90.766208
Negative precision: 94.7531 recall: 92.4699
Negative sensitivity: 92.4699 specificity: 90.1163
Negative FPR: 9.8837 NPV: 86.1111
Positive precision: 86.1111 recall: 90.1163
Positive sensitivity: 90.1163 specificity: 92.4699
Positive FPR: 7.5301 NPV: 94.7531
model trained by GPU (idx:0) has been saved at /home/
Pathology_Experiment/saved_models/
PC_ViT_224_401_Prompt_Deep_ROSE_lf25_b8.pth
finished

```
=====  
Processing finished !  
start time: 2022_04_14 11:21:55  
end time: 2022_04_14 12:37:00  
source: 7876ab523a59
```

Preparing the email with auto log file :
Train__2022_04_14-11_21_55_log
as .rtf

15:16



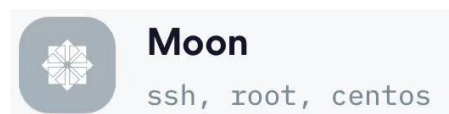
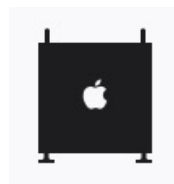
Done server_status.rtf

```
=====  
监控开始时间: 2022-04-14 11:21:55  
采样间隔(s): 5 | 计算均值写入日志间隔(s): 300  
=====  
时间: 2022-04-14 11:26:56 | CPU平均占用率: 30.14 | 内存占用率: 7.26  
时间: 2022-04-14 11:31:56 | CPU平均占用率: 31.39 | 内存占用率: 7.29  
时间: 2022-04-14 11:36:56 | CPU平均占用率: 31.46 | 内存占用率: 7.29  
时间: 2022-04-14 11:41:57 | CPU平均占用率: 30.74 | 内存占用率: 7.29  
时间: 2022-04-14 11:46:57 | CPU平均占用率: 31.45 | 内存占用率: 7.29  
时间: 2022-04-14 11:51:57 | CPU平均占用率: 31.56 | 内存占用率: 7.29  
时间: 2022-04-14 11:56:57 | CPU平均占用率: 30.9 | 内存占用率: 7.28  
时间: 2022-04-14 12:01:58 | CPU平均占用率: 31.69 | 内存占用率: 7.29  
时间: 2022-04-14 12:06:58 | CPU平均占用率: 31.69 | 内存占用率: 7.29  
时间: 2022-04-14 12:11:58 | CPU平均占用率: 30.93 | 内存占用率: 7.28  
时间: 2022-04-14 12:16:59 | CPU平均占用率: 31.69 | 内存占用率: 7.29  
时间: 2022-04-14 12:21:59 | CPU平均占用率: 31.62 | 内存占用率: 7.29  
时间: 2022-04-14 12:26:59 | CPU平均占用率: 31.07 | 内存占用率: 7.28  
时间: 2022-04-14 12:32:00 | CPU平均占用率: 31.74 | 内存占用率: 7.29  
时间: 2022-04-14 12:37:00 | CPU平均占用率: 30.85 | 内存占用率: 7.28  
=====  
监控结束时间: 2022-04-14 12:37:00  
平均CPU占用率: 31.26 | 平均内存占用率: 7.29  
最大CPU占用率: 34.39 | 最大内存占用率: 7.31
```

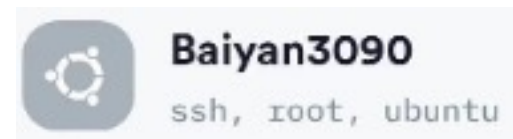
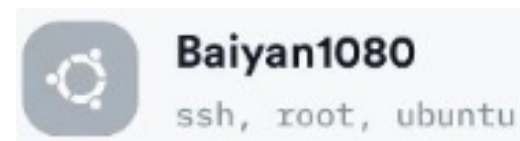
内网穿透技术



自动建立+动态分发ssh隧道



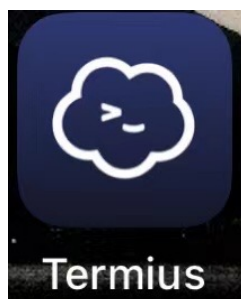
会自动根据物理ip优化路径
提高连接速度+稳定性



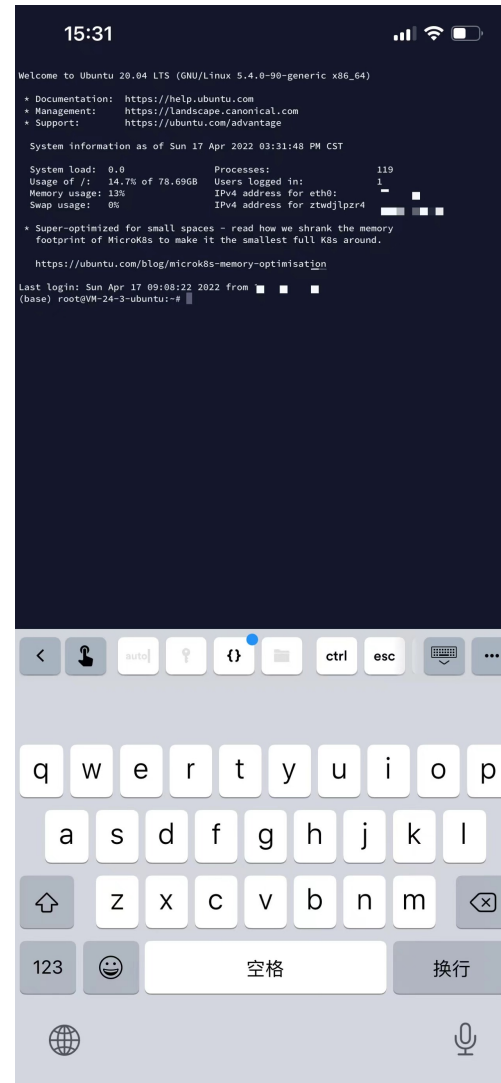
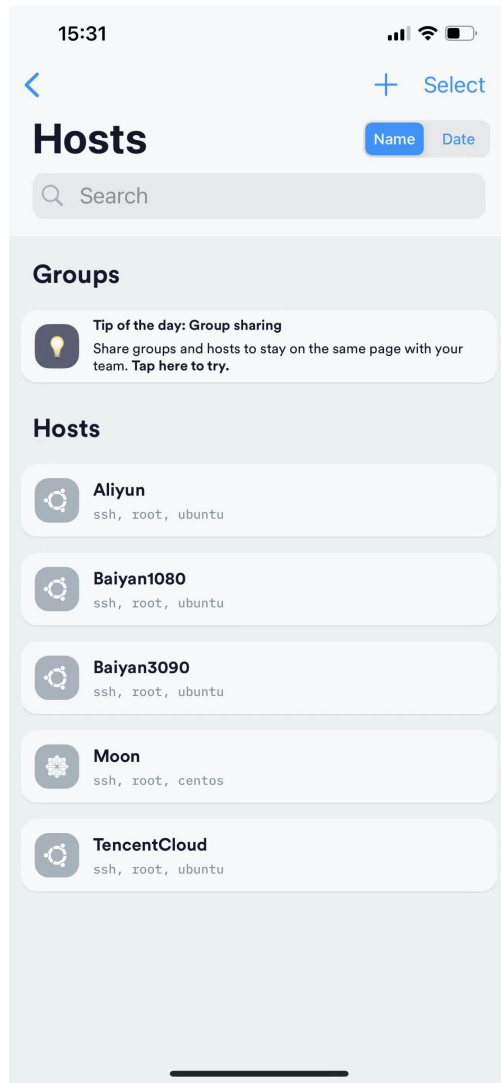
使用命令行，
或者提供命令行的app



or



获得权限的设备可以使用
伪内网ip进行链接访问





不过如此

Thanks

PPT: <https://github.com/sagizty/Insight>

NotifyEmail: <https://github.com/lsgqqq/notifyemail>

Code: [https://github.com/sagizty/Insight/blob/main/Colab%20Sample%20Demo%20\(MAE\).ipynb](https://github.com/sagizty/Insight/blob/main/Colab%20Sample%20Demo%20(MAE).ipynb)

特别安利: <https://mp.weixin.qq.com/s/KHnhSHNzdkFFTCXludBY9w>