aiDAPTIVCache User Guide (MW: NXUN204.A1)

Version

Version	Description	Editor	Reviewer	Date
draft	New Create	wd_lee/uhong_lin/kh_huang/henry_lee/ candy_du/eddie_chang/yuhsiang_chang	Kled/hao zhi_lee	2025/ 05/08
v0.1	preliminary version	wd_lee/uhong_lin/kh_huang/henry_lee/ candy_du/eddie_chang/yuhsiang_chang	Heine	2025/ 09/12

Purpose

This manual is intended solely for aiDAPTIV partners.

Middleware Version: vNXUN_2_04_A1
aiDAPTIV Toolkit Version: 2.3.1

aiDAPTIVCache Family：

Model
AI100E SSD

1 Check Device

In this section, we will check machine device GPU/CPU/RAM/aiDAPTIVCache before start to use aiDAPTIV middleware.
After completing this section, you will know which GPU/CPU is in Phison AVL, and how many RAM size for different LLM model.

1.1 Check GPU is in Approved Vendor List? (AVL)

- Check GPU is in AVL list (Appendix GPU AVL).

- Needs to do validation if GPU is not in AVL. Please refer Appendix E.
- It is recommended that the GPU must be connected to the PCIe Slot of X16. Connecting other Slots (ex: X8, X4) will decrease GPU performance.

1.2 Check the number of GPUs

- Number of GPUs = 2^n (n=0,1,2,3,4, GPUs = 1,2,4,8,16)

1.3 Check DRAM and aiDAPTIVCache from different LLM model size

- DRAM and aiDAPTIVCache from different LLM model size

LLM model size	**<13B**	**34B**	**70B**	**180B**
DRAM	64GB	64GB	128GB	128GB
aiDAPTIVCache capacity	1TB	1TB	2TB	2TB
aiDAPTIVCache count	1	2	2	4
aiDAPTIVCache slot count	1	2	2	4

  - Workstation or PC : Use m.2 slot or m.2 PCIe expansion card.
    - https://www.asus.com/tw/motherboards-components/motherboards/accessories/hyper-m-2-x16-gen5-card/
  - Rack server: Use U.2 slot.
  - Recommend Gen4 or above
  - Recommend Dram 2933MHz or above
  - Recommended DRAM size is 128GB or more
  - Recommended DRAM channel number is 8 or more, ex: 16GB x8

1.4 Check CPU configuration

- Check CPU is in AVL list (Appendix CPU AVL).
- Recommend CPU Cores minimum 8 or above

CPU lanes calculate by GPU count and aiDAPTIVCache count.
Total PCIe lanes = GPU count * 16 + aiDAPTIVCache count * 4

**Example : For GPU count =4 and 70B LLM model size (aiDAPTIVCache 2)*
Total 4 * 16 + 2 * 4 = 72 lanes, It means at least 72 lanes for optimal performance.

2 Installation and Preparation

In this section, we will start installing the aiDAPTIV middleware and explain how to use the middleware. After completing this section, you will know how to use aiDAPTIV for Domain training or finetuning.

2.1 Environment Preparation

2.1.1 Requirements

- Recommend enviroment

Category	**Detail**
OS	Ubuntu 24.04.3 Desktop (kernel version:6.14.0-28-generic)
40 series GPU driver	NVIDIA driver version 550 installation
50 series GPU driver	NVIDIA driver version 570 installation
Python	3.12

2.1.2 Install GPU Driver

2.1.2.1 For 40 series GPU

- Install Driver (Estimated time: 5 min)
    - Install NVIDIA Driver

```
sudo apt install nvidia-utils-550
sudo apt install nvidia-driver-550
```

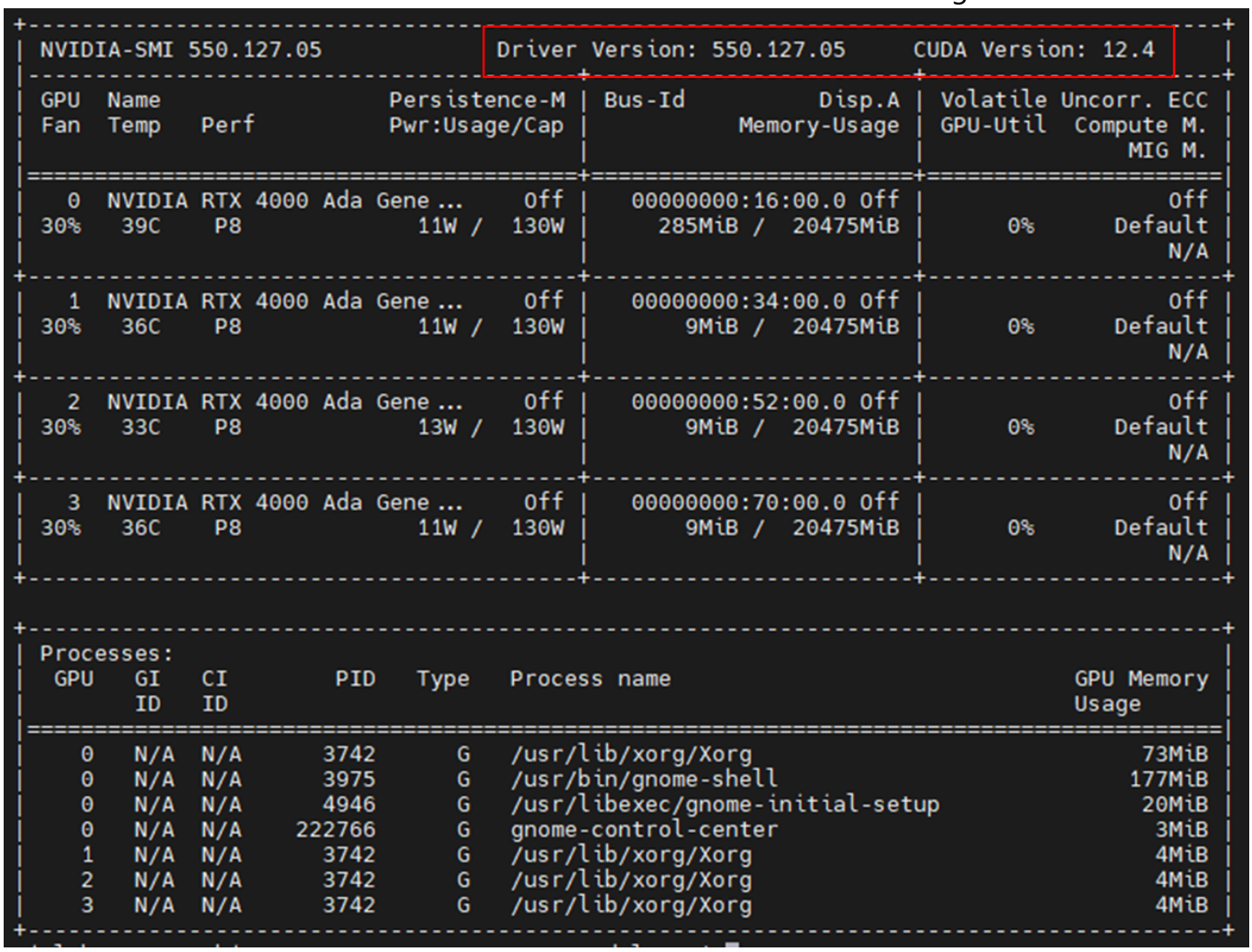
- Reboot system

```
sudo reboot
```

- Successful example

```
nvidia-smi
```

There would be a GPU driver version and CUDA version show in the log.



2.1.2.2 For 50 series GPU

- Install Driver (Estimated time: 5 min)
  - Install NVIDIA Driver

    ```
    sudo apt install nvidia-utils-570
    sudo apt install nvidia-driver-570-open
    ```

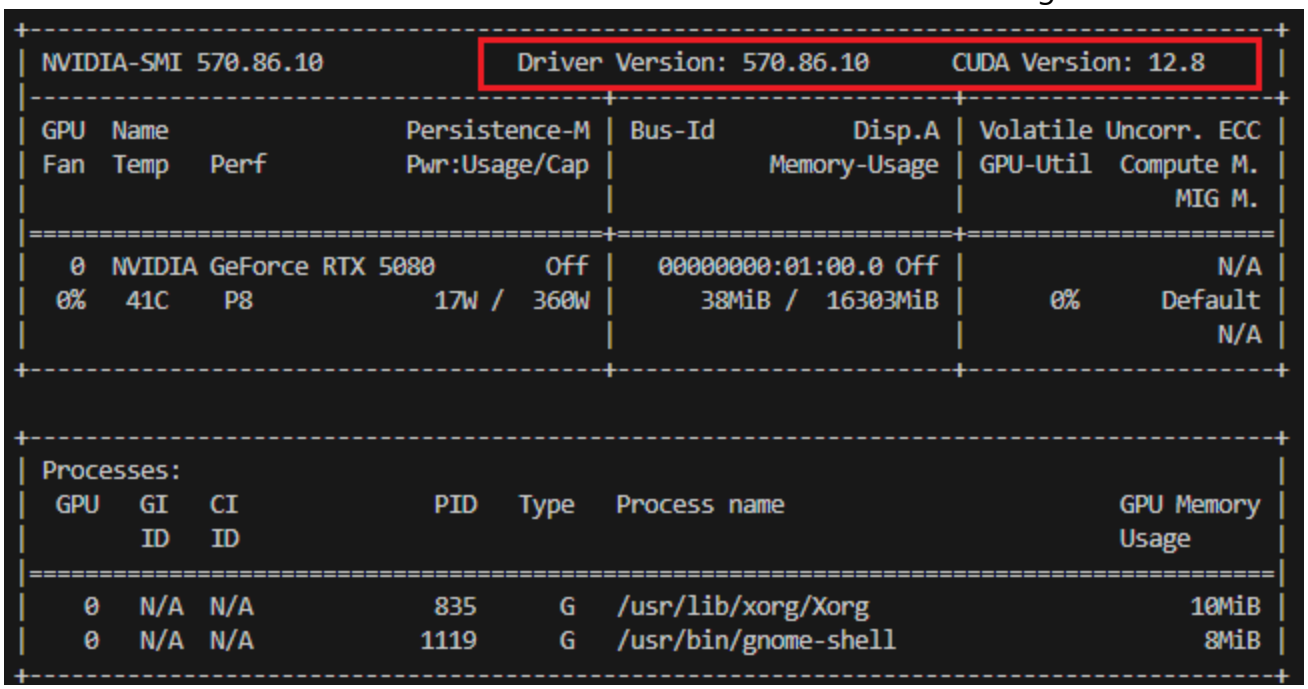
- Reboot system

  ```
  sudo reboot
  ```

- Successful example

```
nvidia-smi
```

There would be a GPU driver version and CUDA version show in the log.



2.1.3 Install GPU Toolkit : CUDA

2.1.3.1 For 40 series GPU

```
```bash
wget https://developer.download.nvidia.com/compute/cuda/12.4.1/local_installers/cuda_1
2.
4.1_550.54.15_linux.run
sudo sh cuda_12.4.1_550.54.15_linux.run
# Continue > accept > Look at the images below and do not check the box
# Turn [X] Driver --> [ ] Driver
# Then, move the cursor to Install and press Enter.
```
```

2.1.3.2 For 50 series GPU

```bash
wget wget https://developer.download.nvidia.com/compute/cuda/12.8.0/local_installers/cuda_12.8.0_570.86.10_linux.run
sudo sh cuda_12.8.0_570.86.10_linux.run
# Continue > accept > Look at the images below and do not check the box
# Turn [X] Driver --> [ ] Driver
# Then, move the cursor to Install and press Enter.
```

2.1.4 Install GPU Toolkit : cuDNN

2.1.4.1 For 40 series GPU (base on ubuntu version to install cuDNN)

- https://developer.nvidia.com/cudnn-9-4-0-download-archive

2.1.4.2 For 50 series GPU (base on ubuntu version to install cuDNN)

- https://developer.nvidia.com/cudnn-9-8-0-download-archive

2.2 aiDAPTIV Middleware Installation

2.2.1 Depoly aiDAPTIV

Please use a fresh Ubuntu system to avoid environment conflicts. If not using a fresh system, please deploy the environment using Docker (Appendix A).

- Native Installation option (The deployed user can be a custom user or root.)
  - Activate python3 virtual environment

    ```
    sudo apt install python3-venv
    python3 -m venv <your virtual environment name>
    source <your virtual environment name>/bin/activate
    ```

  - setup tool (middleware)

    ```
    wget https://phisonbucket.s3.ap-northeast-1.amazonaws.com/setup_vNXUN_2_04_A1.sh
    ```

  - Deploy aiDAPTIV (Install middleware)

```
bash setup_vNXUN_2_04_A1.sh
```

- Select “1. Deploy aiDAPTIV+”

```
phison@linux-BD220683:~$ bash setup_vNXUN_2_04_A1.sh
Hi user phison
Select an action:
1. Deploy aiDAPTIV+
2. Exit
Enter your choice (1, 2):
```

- If you can see the option ”FW Update” in the former step. You can select it and see the picture below. Select ‘Y’, the firmware update process will be initiated. And please **must** refer to **“2.2.3 FW Update”** to continue the update steps. After finish FW update, please go back here and continue the install process.

```
If the FW version does not match, do you need to update the SSD FW? (Y/N)
Please enter Y or N:
```

- Select ‘Y’ to download the installation file from the cloud, or select ‘N’ to provide the installation file path from the local end.

```
Hi user phison, would you like to download vNXUN_2_04_A1.tar from cloud? (Y/N)
Please enter Y or N:
```

- If you select ‘N’, please enter the path to the vNXUN_2_04_A1.tar file

```
Hi user phison, would you like to download vNXUN_2_04_A1.tar from cloud? (Y/N)
Please enter Y or N: n
Please enter the path to the vNXUN_2_04_A1.tar file:
```

- If you can’t get vNXUN_2_04_A1.tar from cloud, please enter the path to the vNXUN_2_04_A1.tar file

```
Can't get vNXUN_2_04_A1.tar from cloud, Please enter the path to the vNXUN_2_04_A1.tar file:
```

- Successful example
  - It will show successful message in the log.

```
Deploy Phison aiDAPTIV+ successfully, You MUST restart the session for the changes to take effect.
Hi user phison
Select an action:
1. Deploy aiDAPTIV+
2. Test aiDAPTIV+
3. Exit
4. FW update
Enter your choice (1, 2, 3, or 4):
```

  - There would be a “aiDAPTIV2” folder in ~/aiDAPTIV2

```
phison@phison-Z690-AERO-D:~/aiDAPTIV2$ ls
automatic-speech-recognition  customize_model_support  image-text-to-text  requirements_amd_mi.txt      requirements_nv.txt  vllm
commands                      fill-mask                phison_comm         requirements_amd_radeon.txt  text-generation
```

- Check Middleware verison:

```
phisonai2 -v
```

- example

```
root@3f58157b5ea9:/# phisonai2 -v
phisonai2: Phison aiDAPTIV aiDAPTIVLink
COPYRIGHT (C) 2024 Phison Electronics Corp.
version: vNXUN204.A1
```

"If this is your first time using this aiDAPTIVLink version, please must read and follow the instructions in 2.2.2 and 2.2.3"

2.2.2 Disk Setup (LVM Setting)

"If this is your first time using this aiDAPTIVLink version, please must read and follow the instructions to correctly mount SSD."

- Install LVM

```
sudo apt update;sudo apt install lvm2 xfsprogs
```

- Check disk locations:

```
lshw -class disk -class storage | grep -E 'ai100|logical name|version: EIFZ'
lsblk | grep nvme
# Confirm ai100 device names are, for example, nvme6n1 and nvme8n1. If not, make t
he necessary changes below to adapt as appropriate.
```

- Clear disks just in case

```
sudo wipefs -a /dev/nvme1n1 /dev/nvme2n1
```

```
phison@phison-X299-UD4-Pro:~$ sudo wipefs -a /dev/nvme1n1 /dev/nvme2n1
/dev/nvme1n1: 2 bytes were erased at offset 0x00000438 (ext4): 53 ef
```

- Create LVM

```
sudo pvcreate /dev/nvme1n1 /dev/nvme2n1
sudo vgcreate ai /dev/nvme1n1 /dev/nvme2n1
sudo lvcreate --type striped -i 2 -I 128k -l 100%FREE -n ai ai
```

```
phison@phison-X299-UD4-Pro:~$ sudo pvcreate /dev/nvme1n1 /dev/nvme2n1
  Physical volume "/dev/nvme1n1" successfully created.
  Physical volume "/dev/nvme2n1" successfully created.
phison@phison-X299-UD4-Pro:~$ sudo vgcreate ai /dev/nvme1n1 /dev/nvme2n1
  Volume group "ai" successfully created
phison@phison-X299-UD4-Pro:~$ sudo lvcreate --type striped -i 2 -I 128k -l 100%FREE -n ai ai
  Logical volume "ai" created.
```

- Mount LVM

```
#Format the disk.
sudo mkfs.xfs -f -s size=4k -m crc=0 /dev/ai/ai -f
#Mount the disk.
sudo mkdir -p /mnt/nvme0
sudo mount /dev/ai/ai /mnt/nvme0
sudo chown -R $USER:$USER /mnt/nvme0
```

- (optional) Make mount persistent

```
# Make mount persistent
sudo echo '/dev/ai/ai /mnt/nvme0 xfs defaults,nofail 0 0' | sudo tee -a /etc/fstab

# Remove permanent mount setting
sudo sed -i '/\/dev\/ai\/ai/d' /etc/fstab
```

- Successful example

```
lsblk
```

If LVM setting is successful, you will see the following successful configuration when input "lsblk".

```
phison@phison-X299-UD4-Pro:~$ lsblk | grep -E 'nvme1n1|ai-ai|nvme2n1'
nvme1n1     259:0    0   1.9T  0 disk
└─ai-ai     252:0    0   3.6T  0 lvm  /mnt/nvme0
nvme2n1     259:1    0   1.8T  0 disk
└─ai-ai     252:0    0   3.6T  0 lvm  /mnt/nvme0
```

- (optional) If you need to dissolve LVM Setting

```
sudo umount /mnt/nvme0;sudo lvremove -y ai;sudo pvremove -y /dev/nvme1n1 /dev/nvme
2n1 --force --force
```

- **If you only have one SSD, please follow the steps below to mount**:

```
sudo mkfs -t ext4 /dev/nvme1n1
sudo mkdir -p /mnt/nvme0
sudo mount /dev/nvme1n1 /mnt/nvme0
sudo chown -R $USER:$USER /mnt/nvme0
```

2.2.3 FW update

"If this is your first time using this aiDAPTIVLink version, please make sure to follow this procedure for Firmware Update."

If fw update is not performed during installation, you can use setup_vNXUN_2_04_A1.sh to do fw check/update.

If aiDaptiv+ is successfully installed is correct, a third option "FW update" will appear.

- Select "3. FW update"

```
Deploy Phison aiDAPTIV+ successfully, You MUST restart the session for the changes to take effect.
Hi user phison
Select an action:
1. Deploy aiDAPTIV+
2. Exit
3. FW update
Enter your choice (1, 2 or 3):
```

- You can choose to FW check/update

```
Choose an action:
1) Check FW version
2) Update FW version
3) Back
4) Exit
Enter your choice (1, 2, or 3):
```

- After selecting check/update, select 1 to add the device that needs to perform fw check/update

```
Choose an action:
1) Add nvme_path
2) Terminate adding nvme_path
Please select an action: 1
Enter nvme_path to add: /dev/nvme0n1
```

- The added device will be displayed at the top. If you make a mistake in adding, select 2 to remove it. If you are done adding, select 3 to continue.

```
Current nvme_path list:
1) /dev/nvme0n1
Choose an action:
1) Add nvme_path
2) Remove nvme_path
3) Terminate adding nvme_path
Please select an action:
```

- Confirm again whether the added device is correct

```
Double-check the nvme_path list:
Current nvme_path list:
1) /dev/nvme0n1
Choose an action:
1) Confirm
2) Back
Please select an action:
```

- Successful example
  - FW check

```
Double-check the nvme_path list:
Current nvme_path list:
1) /dev/nvme0n1
Choose an action:
1) Confirm
2) Back
Please select an action: 1
Confirming and checking...
Checking firmware for: /dev/nvme0n1
check_fw_version
[PHISON AIDAPTIV][INFO] Firmware of selected device needs to be updated
```

  - FW update

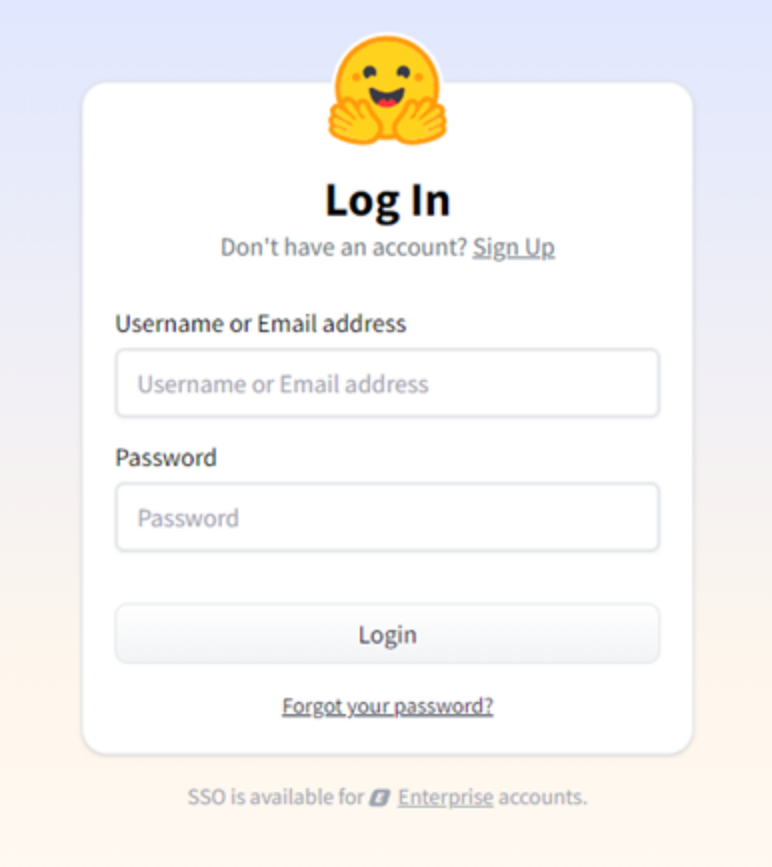
```
Double-check the nvme_path list:
Current nvme_path list:
1) /dev/nvme0n1
Choose an action:
1) Confirm
2) Back
Please select an action: 1
Confirming and Updating...
Updating firmware for: /dev/nvme0n1
update_fw_version
[PHISON AIDAPTIV][INFO] Firmware of selected device is up to date
```

2.3 Login huggingface

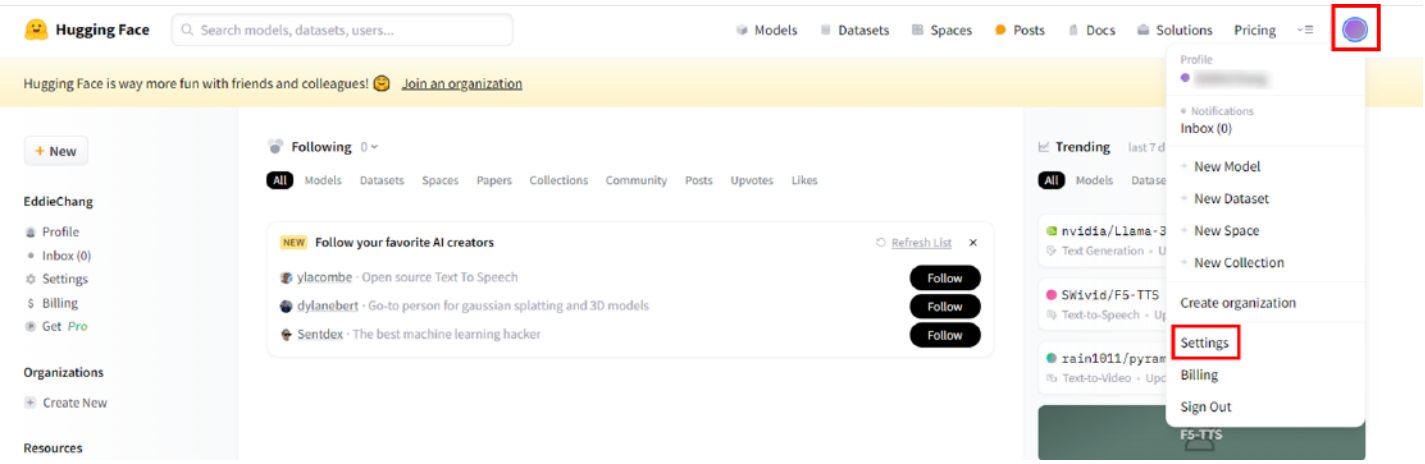
2.3.1 How to get Hugging Face token

Setting your huggingface token in the environment. You can obtain your personal token at: https://huggingface.co/settings/tokens

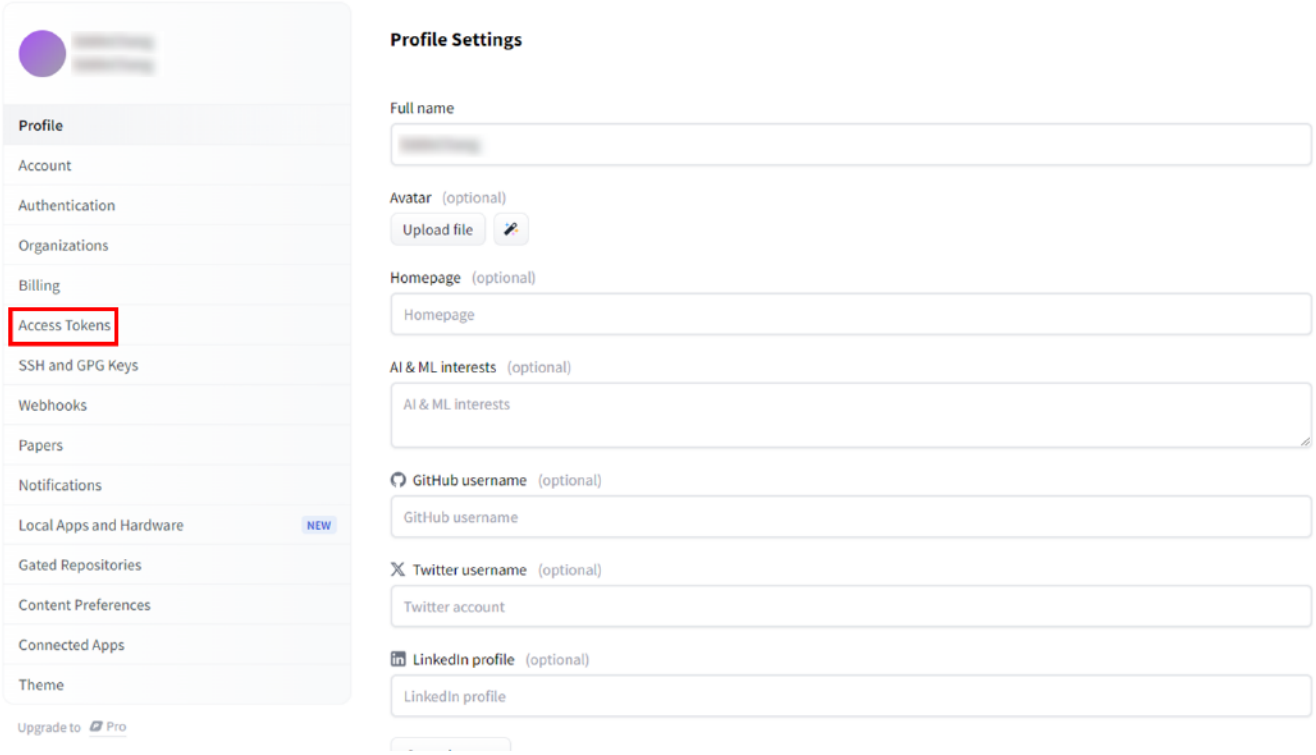
First, please register for a Hugging Face account. If you have already registered, please log in to Hugging Face directly.



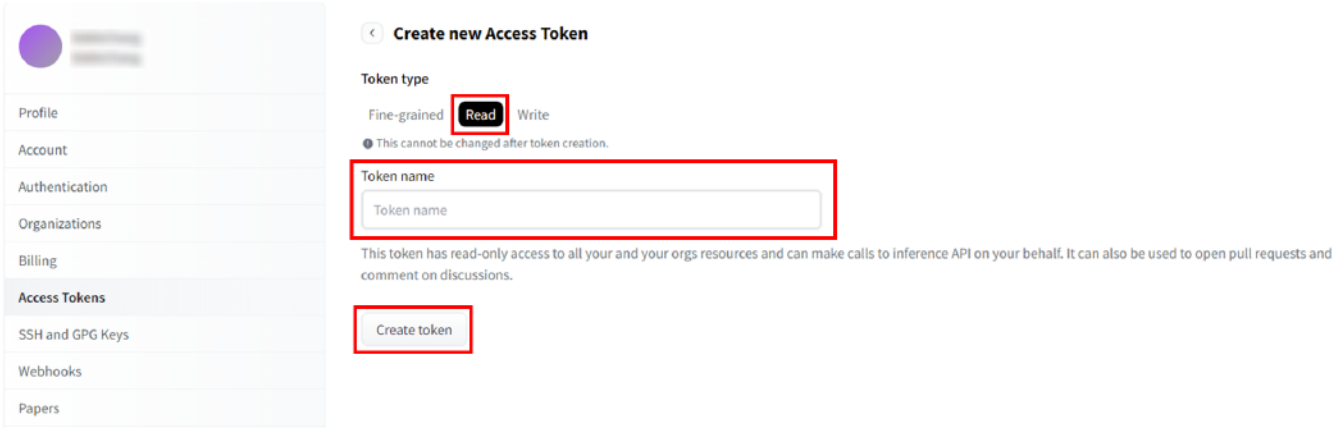
After logging in, click on the account in the top right corner, open the menu, and select 'Settings'.



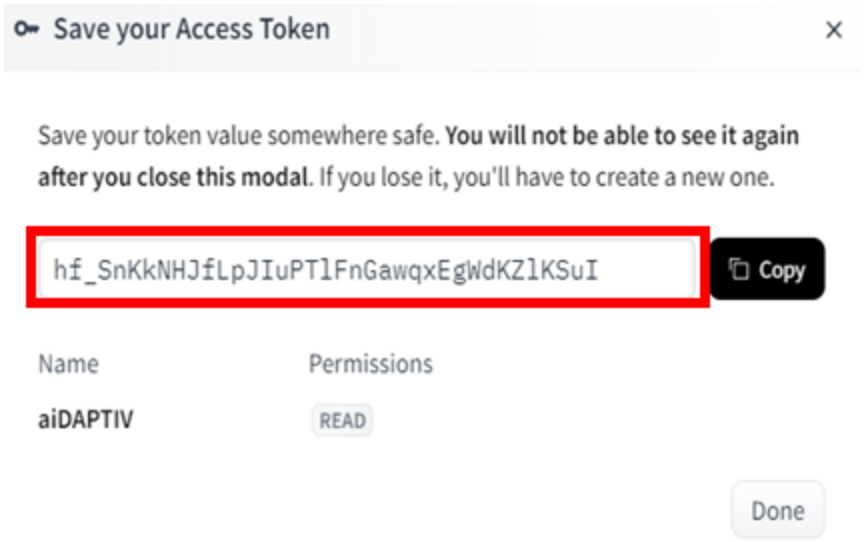
After entering ‘Settings’, click on ‘Access Tokens’ in the left column.



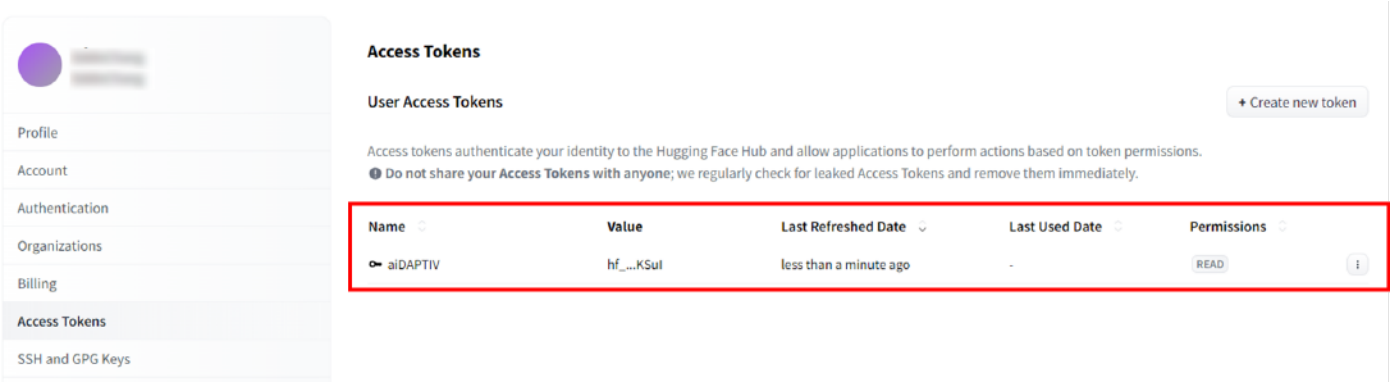
Choose the ‘Token type’ and enter the ‘Token name’, then you can create the required token. You can make a selection based on the content and description of the ‘Token type’. If you only need to download, choose ‘Read’.



After choosing to create, a Hugging Face token will appear. This token is used for logging in later.



The created token will be stored in the 'Access Token' in your personal account.



2.3.2 How to register the Hugging Face token

```
git config --global credential.helper store
huggingface-cli login
# login with <your_hf_token>
```

- Successful example
  There would be a "Login successful" message

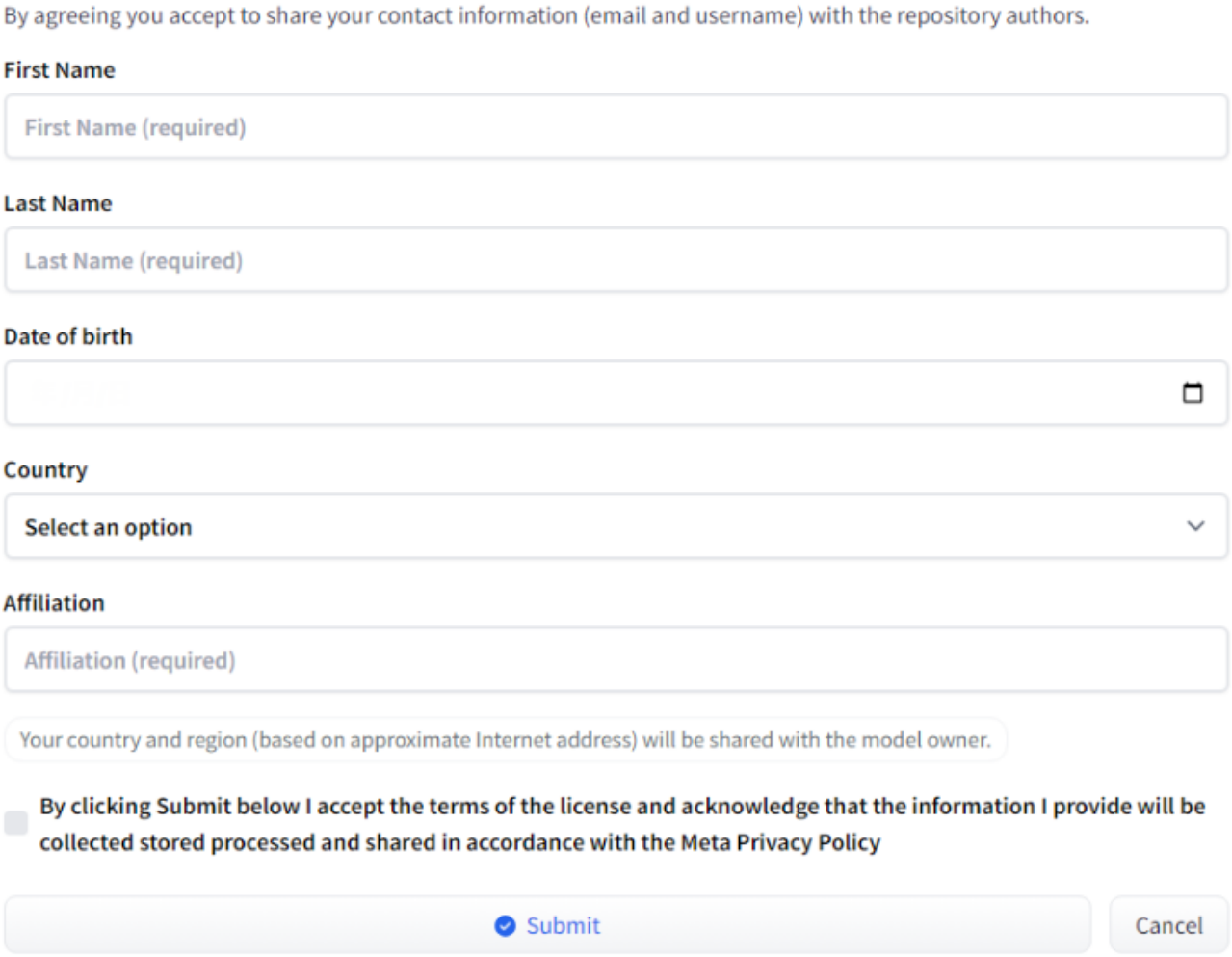
```
phison@phison-Pro-ET700I-W7:~/Desktop/aiDAPTIV2$ huggingface-cli login

    _|    _|  _|    _|    _|_|_|    _|_|_|  _|_|_|  _|      _|    _|_|_|      _|_|_|_|    _|_|      _|_|_|  _|_|_|_|
    _|    _|  _|    _|  _|        _|            _|    _|_|    _|  _|            _|        _|    _|  _|        _|
    _|_|_|_|  _|    _|  _|  _|_|  _|  _|_|      _|    _|  _|  _|  _|  _|_|      _|_|_|    _|_|_|_|  _|        _|_|_|
    _|    _|  _|    _|  _|    _|  _|    _|      _|    _|    _|_|  _|    _|      _|        _|    _|  _|        _|
    _|    _|    _|_|      _|_|_|    _|_|_|  _|_|_|  _|      _|    _|_|_|      _|        _|    _|    _|_|_|  _|_|_|_|

    To login, `huggingface_hub` requires a token generated from https://huggingface.co/settings/tokens .
Enter your token (input will not be visible):
Add token as git credential? (Y/n) y
Token is valid (permission: read).
Your token has been saved in your configured git credential helpers (store).
Your token has been saved to /home/phison/.cache/huggingface/token
Login successful
```

2.4 Download Llama-3.1-8B-Instruct

- Before downloading the Llama-3.1-8B-Instruct model, you need to request permission from huggingface.
    - https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct
    - It needs to wait for the license approval from huggingface



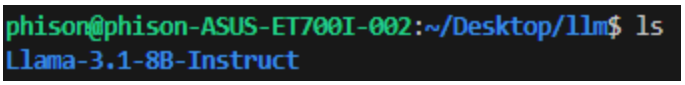
- Download Llama-3.1-8B-Instruct model

```
mkdir -p /home/$USER/Desktop/llm
cd /home/$USER/Desktop/llm
mkdir Llama-3.1-8B-Instruct
huggingface-cli download --token HF_TOKEN --resume-download meta-llama/Llama-3.1-8
B-Instruct \
--local-dir-use-symlinks False --local-dir Llama-3.1-8B-Instruct
# It would be take some time to download model.
```

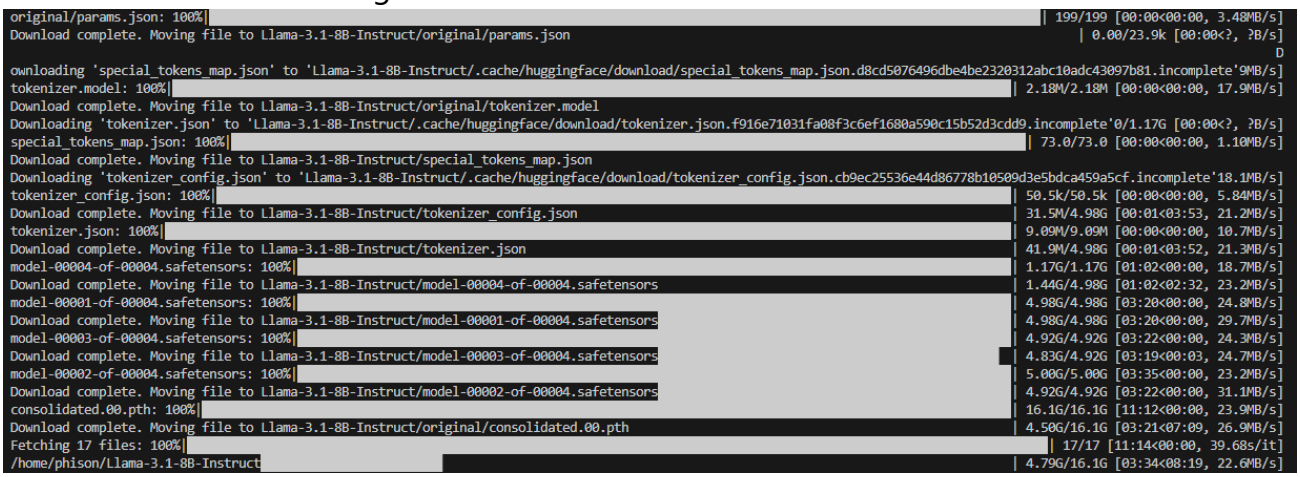
- remark
  HF_TOKEN: Please replace with <your_hf_token>
  meta-llama/Llama-3.1-8B-Instruct: You can replace it with the model you want to download.

Llama-3.1-8B-Instruct: Please replace it with the folder you created

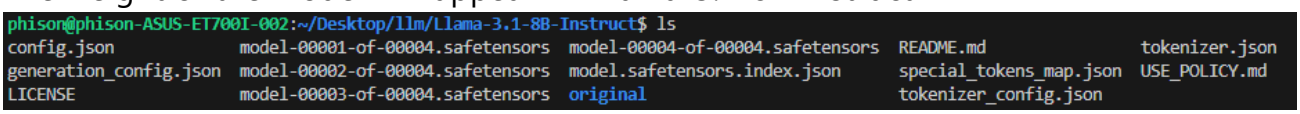
- Successful example
  - mkdir -p Llama-3.1-8B-Instruct in /home/$USER/Desktop/llm/



  - Download success message



  - The weight of the model will appear in “Llama-3.1-8B-Instruct”



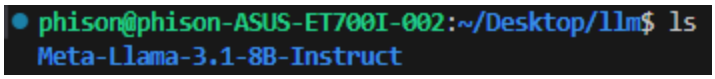
2.5 Run aiDAPTIV

“phisonai2” command is model training command.
Remind! It might take hours to finish the job. (base on your dataset size and device performance.)

- Start training your model.
  Training model “Llama-3-8B-Instruct” with dataset “Dahoas/rm-static”

  ```
  phisonai2 --env_config <env_config.yaml path> --exp_config <exp_config.yaml path>
  ```

  First, find the location where the model is stored.

  Use the ‘lsblk’ command to confirm the location of the mounted SSD.

  Write the required parameters into

  ‘/home/$USER/aiDAPTIV2/commands/env_config/env_config.yaml’

  ‘/home/$USER/aiDAPTIV2/commands/exp_config/exp_config.yaml’

  or replace it with the path where you store ‘env_config.yaml’ and ‘exp_config.yaml’.

  Remark:

  1. Remember to replace $USER with the current user.
  2. For more efficient training, it is recommended to set ‘triton’ to true.

- text-generation settings
  Example of env_config.yaml

```
# Save_path, nvme_path, log_name settings
path_settings:
  lora:
    lora_weight: ""       # whether to load lora_weight (only activated when lor
a: true)
    lora_output_dir: ""   # whether to save lora adapter weight (only activated wh
en lora: true)
  model_name_or_path: "/home/$USER/Desktop/llm/Llama-3.1-8B-Instruct"
  data_path:
    # - ./dataset_config/image-text-to-text/VQA_dataset_config.yaml
    # - ./dataset_config/text-generation/Pretrain_dataset_config.yaml
    # - ./dataset_config/text-generation/RAG_dataset_config.yaml
    - ./dataset_config/text-generation/QA_dataset_config.yaml
  nvme_path: "/mnt/nvme0"
  output_dir: "/home/$USER/output"
  log_name: "Llama-3.1-8B-Instruct.log"
```

Example of exp_config.yaml

```
process_settings:
  master_port: 8299
  num_gpus: 1
  specify_gpus: null
run_settings:
  task_type: "text-generation"
  task_mode: "train" # or "/home/$USER/aiDAPTIV2/text-generation/train.py"
  per_device_train_batch_size: 1
  per_update_total_batch_size: 4
  num_train_epochs: 1
  max_iter: -1
  max_seq_len: 2048
  triton: True
  weight_file_format: null
  from_config: false
  precision_mode: 1

 model_saver:
    max_num_of_saved_model_on_epoch_end: -1 # If the value is -1, it is equal to n
um_train_epochs.
    enable_save_model_on_iteration: false
    max_num_of_saved_model_on_iteration: 2
    num_of_iteration_to_save_model: 2

  lr_scheduler:
    mode: -1
    learning_rate: 0.000007

  optimizer:
    beta1: 0.9
    beta2: 0.95
    eps: 0.00000001
    weight_decay: 0.01

  lora:
    enable_lora: false
    lora_rank: 8
    lora_alpha: 16
    lora_task_type: "CAUSAL_LM"
```

- Monitor training status
    - Open another window, the log will be generated inside the folder where the command is executed.

```
#Open another session
tail -f <your_log>
# example: tail -f Phison_2024_05-15_1200.log
```

- If the training logs are checked and the loss after updates shows a decreasing trend, it indicates a successful training.

```
[2024-09-20 10:38:35.375315] [PHISON START] Epoch: 0, Iteration: 6
[2024-09-20 10:38:35.375436] [Forward][start]
[2024-09-20 10:38:37.103003] [Forward][time spent]:1.7275550365447998
[2024-09-20 10:38:37.198474] [Loss]:2.2202885150909424
[2024-09-20 10:38:37.198491] [Backward][start]
[2024-09-20 10:38:58.795814] [Backward][time spent]:21.597309350967407
[2024-09-20 10:38:58.796005] [Update][Start]
[2024-09-20 10:38:59.218915] [Update][time spent]:0.42287421226501465
[2024-09-20 10:38:59.219000] [PHISON END] Iteration: 6

[2024-09-20 10:38:59.222052] [PHISON START] Epoch: 0, Iteration: 7
[2024-09-20 10:38:59.222171] [Forward][start]
[2024-09-20 10:39:00.944030] [Forward][time spent]:1.721846580505371
[2024-09-20 10:39:01.040680] [Loss]:2.3363866806030273
[2024-09-20 10:39:01.040695] [Backward][start]
[2024-09-20 10:39:10.869449] [Backward][time spent]:9.828744173049927
[2024-09-20 10:39:10.869523] [Update][Start]
[2024-09-20 10:39:35.695246] [Update][time spent]:24.825713872909546
[2024-09-20 10:39:35.695333] [PHISON END] Iteration: 7

Training efficiency: 198.45754094494418 (tokens/s)

[2024-09-20 10:39:35.696001] [Save Model_Checkpoint][Start]
[2024-09-20 10:39:53.636669] [Save Model_Checkpoint][time spent]:17.94063425064087
Beginning of Epoch 2/4, Total Micro Batches 38128
[2024-09-20 10:39:53.642974] [PHISON START] Epoch: 1, Iteration: 0
[2024-09-20 10:39:53.654870] [Forward][start]
[2024-09-20 10:39:55.376546] [Forward][time spent]:1.7216582298278809
[2024-09-20 10:39:55.472200] [Loss]:1.9732047319412231
[2024-09-20 10:39:55.472217] [Backward][start]
[2024-09-20 10:40:00.176516] [Backward][time spent]:4.704289197921753
[2024-09-20 10:40:00.176582] [Update][Start]
[2024-09-20 10:40:00.394558] [Update][time spent]:0.21796607971191406
[2024-09-20 10:40:00.394635] [PHISON END] Iteration: 0

[2024-09-20 10:40:00.397485] [PHISON START] Epoch: 1, Iteration: 1
[2024-09-20 10:40:00.397604] [Forward][start]
[2024-09-20 10:40:02.151702] [Forward][time spent]:1.7540862560272217
[2024-09-20 10:40:02.248785] [Loss]:1.5672987699508667
[2024-09-20 10:40:02.248797] [Backward][start]
[2024-09-20 10:40:10.331515] [Backward][time spent]:8.082708597183228
[2024-09-20 10:40:10.331587] [Update][Start]
[2024-09-20 10:40:10.711014] [Update][time spent]:0.37941741943359375
[2024-09-20 10:40:10.711099] [PHISON END] Iteration: 1
```

)

- Successful example
  You get your first finetuned model in <output_dir> /home/$USER/output !!
  Finetuned model file: model-index.safetensors

```
phison@phison-ASUS-ET700I-002:~/output/finetuned_model_2024-09-20-10-36-37/epoch_3_step_7_Meta-Llama-3.1-8B-Instruct$ ls -l
total 15693120
-rw-rw-rw- 1 phison phison       951 九  20 10:46 config.json
-rw-rw-rw- 1 phison phison 5295466472 九  20 10:46 model-00001.safetensors
-rw-rw-rw- 1 phison phison 5352157840 九  20 10:46 model-00002.safetensors
-rw-rw-rw- 1 phison phison 4362258776 九  20 10:46 model-00003.safetensors
-rw-rw-rw- 1 phison phison 1050673280 九  20 10:46 model-00004.safetensors
-rw-rw-rw- 1 phison phison      22506 九  20 10:46 model.safetensors.index.json
-rw-rw-rw- 1 phison phison        345 九  20 10:46 special_tokens_map.json
-rw-rw-rw- 1 phison phison      50919 九  20 10:46 tokenizer_config.json
-rw-rw-rw- 1 phison phison    9084449 九  20 10:46 tokenizer.json
```

- image-text-to-text settings

  Example of env_config.yaml

```
# Save_path, nvme_path, log_name settings
path_settings:
  lora:
    lora_weight: ""       # whether to load lora_weight (only activated when lor
a: true)
    lora_optimizer: ""       # whether to load lora_optimizer (only activated whe
n lora: true)
    lora_output_dir: ""   # whether to save lora adapter weight (only activated wh
en lora: true)
  model_name_or_path: "/home/$USER/Desktop/llm/Llama-3.2-11B-Vision-Instruct"
  multi_node_env_path: null
  optimizer_path: "" # whether to load optimizer
  train_data_path:
    # - ./dataset_config/automatic-speech-recognition/dataset_config.yaml
     - ./dataset_config/image-text-to-text/VQA_dataset_config.yaml
    # - ./dataset_config/text-generation/Pretrain_dataset_config.yaml
    # - ./dataset_config/text-generation/RAG_dataset_config.yaml
    # - ./dataset_config/text-generation/QA_dataset_config.yaml
  val_data_path: null
    # - ./dataset_config/automatic-speech-recognition/dataset_config.yaml
    # - ./dataset_config/image-text-to-text/VQA_dataset_config.yaml
    # - ./dataset_config/text-generation/Pretrain_dataset_config.yaml
    # - ./dataset_config/text-generation/RAG_dataset_config.yaml
    # - ./dataset_config/text-generation/QA_dataset_config.yaml
  nvme_path: "/mnt/nvme0"
  output_dir: "/home/$USER/output"
  log_name: "Llama-3.2-11B-Vision-Instruct.log"
```

  Example of exp_config.yaml

```
process_settings:
 master_port: 8299
 num_gpus: 1
 specify_gpus: null
 master_port: 8299
 multi_node_settings:
     enable: False
     master_addr: "127.0.0.1"

run_settings:
 task_type: "image-text-to-text"
 task_mode: "train" # or "/home/$USER/aiDAPTIV2/text-generation/train.py"
 per_device_train_batch_size: 1
 per_update_total_batch_size: 4
 num_train_epochs: 1
 max_iter: -1
 max_seq_len: 2048
 triton: True
 weight_file_format: null
 from_config: false
 precision_mode: 1
 enable_save_optimizer_state: false

model_saver:
   max_num_of_saved_model_on_epoch_end: -1 # If the value is -1, it is equal to nu
m_train_epochs.
   enable_save_model_on_iteration: false
   max_num_of_saved_model_on_iteration: 2
   num_of_iteration_to_save_model: 2

 lr_scheduler:
   mode: -1
   learning_rate: 0.000007

 optimizer:
   beta1: 0.9
   beta2: 0.95
   eps: 0.00000001
   weight_decay: 0.01

 early_stop:
   enable: false
   min_delta: 0.01
   patience: 2
   verbose: false

 lora:
   enable_lora: false
```

```
    lora_rank: 8
    lora_alpha: 16
    lora_task_type: "CAUSAL_LM"
    lora_target_modules: null
```

Using the following argument, you can complete your training task.

env_config.yaml

```
path_settings:
  lora:
    lora_weight: "str"       # whether to load lora_weight (only activated when lo
ra: true)
    lora_optimizer: "str"       # whether to load lora_optimizer (only activated w
hen lora: true)
    lora_output_dir: "str"   # whether to save lora adapter weight (only activate
d when lora: true)
  model_name_or_path: "str"
  multi_node_env_path: null
  optimizer_path: "str" # whether to load optimizer
  train_data_path:
    # - ./dataset_config/image-text-to-text/VQA_dataset_config.yaml
    # - ./dataset_config/text-generation/Pretrain_dataset_config.yaml
    # - ./dataset_config/text-generation/RAG_dataset_config.yaml
    - ./dataset_config/text-generation/QA_dataset_config.yaml
  val_data_path: null
    # - ./dataset_config/image-text-to-text/VQA_dataset_config.yaml
    # - ./dataset_config/text-generation/Pretrain_dataset_config.yaml
    # - ./dataset_config/text-generation/RAG_dataset_config.yaml
    # - ./dataset_config/text-generation/QA_dataset_config.yaml
  nvme_path: "str"
  output_dir: "str"
  log_name: "str"
```

exp_config.yaml

```
process_settings:                    # <examples>
  num_gpus: 'int'                    # 1, 2, 4, 8
  specify_gpus: [null|'str']         # null, '0,1,2,3', '4,5'
  master_port: 'int'                 # 8299
  multi_node_settings:
    enable: 'bool'                  # true, false
    master_addr: 'str'              # "127.0.0.1"

run_settings:
  task_type 'str'                    # 'text-generation', 'automatic-speech-recogn
ition', 'fill-mask', 'image-text-to-text'
  task_mode 'str'                    # 'train', or '/home/$USER/aiDAPTIV2/text-gen
eration/train.py'
  per_device_train_batch_size 'int'  # 10
  per_update_total_batch_size 'int'  # 100 It must be set as a multiple of (num_gp
us * per_device_train_batch_size).
  num_train_epochs 'int'             # 5
  max_iter 'int'                     # -1, 100
  max_seq_len 'int'                  # 2048
  triton 'bool'                      # true, false
  weight_file_format [null|'str']    # null, 'bin', 'pt', 'safetensors'
  from_config 'bool'                 # true, false
  precision_mode 'int'               # 0, 1
  enable_save_optimizer_state: 'bool' # true, false

  model_saver:
    max_num_of_saved_model_on_epoch_end: -1 # If the value is -1, it is equal to n
um_train_epochs.
    enable_save_model_on_iteration: false
    max_num_of_saved_model_on_iteration: 2
    num_of_iteration_to_save_model: 2

  lr_scheduler
    mode 'int'                       # -1, 0, 1, 2, 3, 4, 5
    learning_rate 'float'            # 0.000007

  optimizer
    beta1 'float'                    # 0.9
    beta2 'float'                    # 0.95
    eps 'float'                      # 0.00000001
    weight_decay 'float'             # 0.01

  early_stop:
    enable: 'bool'                   # true, false
    min_delta: 'float'               # 0.01
    patience: 'int'                  # 2
    verbose: 'bool'                  # true, false
```

```
lora
  enable_lora 'bool'                # true, false
  lora_rank 'int'                   # 8
  lora_alpha 'int'                  # 16
  lora_task_type 'str'              # 'CAUSAL_LM'
  lora_target_modules: null,'str',list[str]      # null
```

Arguments Descriptions

- **path_setting**:
  - **lora**:
    - **lora_weight**: absolute path to the pretrained lora weight folder (**default: None**).
    - **lora_optimizer**: absolute path to the lora optimizer (**default: None**).
    - **lora_output_dir**: absolute path for saving lora model (default: None, lora model will only be saved if you provide lora_output_dir). (**default: None**).
  - **model_name_or_path**: **[Necessary!]** absolute path to the pretrained weight folder (downloaded from huggingface) (**default: None**).
  - **multi_node_env_path**: Multi node env config (**default: Null**).
  - **optimizer_path**: Input the path of the optimizer state saved from the previous training session. The current training session will continue based on this optimizer state. (**default: None**).
  - **train_data_path**:When adding dataset, the yaml file under **/commands/env_configs** need to be configured.
    - Currently only support:

```
- VQA datasets
- RAG datasets
- QA datasets
- Pretrain datasets
```

for RAG, QA dataset, **single** QA pair or **multi-turn** chat can both be accept

After configured, the yaml file path should be added to **commands/env_config/env_config.yaml**

```
train_data_path:
   - ./dataset_config/image-text-to-text/VQA_dataset_config.yaml
   - ./dataset_config/text-generation/Pretrain_dataset_config.yaml
   - ./dataset_config/text-generation/RAG_dataset_config.yaml
   - ./dataset_config/text-generation/QA_dataset_config.yaml
```

> **To run the default env_config.yaml, the execute path should locate at aiDAPTIV2/commands.**

- **val_data_path**: absolute path to the dataset used to validate the training (**default: None**).
- **nvme_path**: **[Necessary!]** mounting point to aiDAPTIVCache (ex:/mnt/nvme0) (**default: None**).
- **output_dir**: absolute path for saving finetuned model (**default: None, finetuned model will only be saved if you provide output_dir**).
- **log_name**: absolute path for saving training log (**default: Phison_"currenttime".log**).

- **process_settings**:
  - **num_gpus**: number of gpus to be utilized (**default: 1**).
  - **specify_gpus**: (optional) specify GPUs to use. If you want to use No3 and No2 GPU, set specify_gpus="3,2" (**default: null**).
  - **master_port**: port used by PyTorch distributed for communication during training (**default: 8299**).
  - **multi_node_settings**:
    - **enable**: trigger multi node training (**default: False**).
    - **master_addr**: Master node (rank 0)'s address, should be either the IP address or the hostname of node 0, for single node multi-proc training, the --master_addr can simply be 127.0.0.1 (**default: 127.0.0.1**).

- **run_settings**:
  - **task_type**: Choose a task type, the folder in ~/aiDAPTIV2. We only support "text-generation", "automatic-speech-recognition" and "fill-mask" in this version (**default: text-generation**). Check task type and model are in AVL (Appendix E Support Model list).
  - **task_mode**: Choose a task mode. We support "train", "inference", "eval", and absolute path to the execution python file, ex: "/home/$USER/aiDAPTIV2/text-generation/train.py" (**default: train**).

- **per_device_train_batch_size**: batch size in each GPU (**default: 1**).
- **per_update_total_batch_size**: batch size for one update (**default: 128**).
  remark : It must be set as a multiple of (num_gpus * per_device_train_batch_size). Ex: if you have 4 GPUs, each of them have 4 batches. you want to update the model every 80 batches. Then set the per_device_train_batch_size = 4, per_update_total_batch_size = 80. Your machine will run 80/4/4=5 iterations and update once.
- **num_train_epochs**: how many epoch you want to train your model (**default: 1**).
- **max_iter**: (optional) early stop iteration before running entire epoch. Set -1: execute all iterations for each epoch, set 10: only execute 10 iterations for each epoch. (**default: -1**).
- **max_seq_len**: sequence length you want to train your language model (**default: 2048**).
  - Model limitations: if you run bert model, you must set max_seq_len=512. For other models, please refer to the info on the huggingface's model page.
- **triton**: (optional) trigger triton training procedure. We include some of techniques from Triton to improve training efficiency. Llama, Mistral, Mixtral are currently supported (**default: True**).
- **weight_file_format**: file format for loading pretrained model. We only support "safetensors", "bin", "pt", "pth" and "None" in this version. (**default: None, we will automaticlly search file format.**).
- **from_config**: whether randomly init model weight. (**default: False**)
- **precision_mode**: determines what precision to train your model. 0 for "BF16" and 1 for "BF16, FP32 mixed". (**default: 1**)
- **enable_save_optimizer_state**: Default is false, Indicates whether to save the current optimizer state to the system disk. If set to true, it means you want to save the optimizer state from this training session for use in the next training session. (**default: False**).
- **model_saver**
  - **max_num_of_saved_model_on_epoch_end**: Specifies the maximum number of models to be saved during epoch cycles. If the number of saved models exceeds this value during the training process, the first model saved in the

epoch will be removed. Setting this value to -1 is equivalent to num_train_epochs, meaning that every model saved per epoch will be retained (default: -1).

- **enable_save_model_on_iteration**: Specifies whether to save the model during iteration cycles (default: false).
- **max_num_of_saved_model_on_iteration**: Specifies the maximum number of models to be saved during iteration cycles. The concept is similar to max_num_of_saved_model_on_epoch_end. The value must be greater than 0, and it is recommended to use at least 2 to avoid issues where models may be lost due to device problems during the saving process (default: 2).
- **num_of_iteration_to_save_model**: Specifies how often a model should be saved per number of steps, and it must be a multiple of args.gradient_accumulation_steps. The value must be greater than 0 (default: 2).

- **lr_scheduler**

  - **learning_rate**: learning rate, be aware of this hyper-parameter. It can affect training result (**default: 7e-6**).
  - **mode**: choose a learning rate mode (default: 1).
    - mode == -1: LinearLR(optimizer, start_factor=1, total_iters=1)
    - mode == 0: LinearLR(optimizer, start_factor=0.5, total_iters=20)
    - mode == 1: CosineAnnealingLR(optimizer, T_max=150)
    - mode == 2: ExponentialLR(optimizer, gamma=0.99)
    - mode == 3: MultiplicativeLR(optimizer, lr_lambda=lambda epoch: 0.95)
    - mode == 4: StepLR(optimizer, step_size=30, gamma=0.1)
    - mode == 5: MultiStepLR(optimizer, milestones=[30,80], gamma=0.1)

- **optimizer**

  - **beta1**: hyper-parameter for adam optimizer (**default: 0.9**).
  - **beta2**: hyper-parameter for adam optimizer (**default: 0.95**).
  - **eps**: hyper-parameter for adam optimizer (**default:1e-8**).
  - **weight_decay**: weight decay coefficient.(**default:1e-2**).

- **optimizer**

  - **beta1**: hyper-parameter for adam optimizer (**default: 0.9**).
  - **beta2**: hyper-parameter for adam optimizer (**default: 0.95**).
  - **eps**: hyper-parameter for adam optimizer (**default:1e-8**).

- **weight_decay**: weight decay coefficient.(**default:1e-2**).

- **early_stop**
  - **enable**: trigger early stop (**default: False**).
  - **min_delta**: The current val_loss must be more than min_delta away from the best val_loss to be considered improved (**default: 0.01**).
  - **patience**: If val_loss does not improve and exceeds the patience times, early stopping will be triggered (**default:2**).
  - **verbose**: trigger to print loss.(**default:False**).

- **lora**
  - **enable_lora**: trigger lora training procedure (**default: False**).
  - **lora_rank**: dimension of the low-rank matrices for lora (**default: 8**).
  - **lora_alpha**: scaling factor of the weight matrices for lora (**default: 16**).
  - **lora_task_type**: training task type for lora. We only support "CAUSAL_LM" in this version (**default: "CAUSAL_LM"**).
  - **lora_target_modules**: The names of the modules to apply the adapter to. If this is specified, only the modules with the specified names will be replaced (**default: "null"**).

2.6 Quick Start for pytorch

Without aiDAPTIV+ Middleware

*** Regular training procedure ***

Import optimizer

```
# Without aiDAPTIV+ Middleware
from torch.optim import Adam
```

Create model instance and add model.parameters to optimizer

```
# Without aiDAPTIV+ Middleware
model = prepare_bf16_hf_model(model_name_or_path=MODEL_NAME_OR_PATH, tokenizer=tokenizer).to(device)
optimizer = Adam(model.parameters(), LEARNING_RATE)
```

With aiDAPTIV+ Middleware

*** Only Few steps to adapt with aiDAPTIV+***

Import API from site-packages

```
# With aiDAPTIV+ Middleware
from phisonlib.moirai import initialize, save_model, MoiraiConfig
```

Create model instance and call initialize. And we're done!

```
# With aiDAPTIV+ Middleware
model = prepare_bf16_hf_model_init_stream(model_name_or_path=MODEL_NAME_OR_PATH, tokenizer=tokenizer)
moirai_config = prepare_config()
model, optimizer  = initialize(module=model, config=moirai_config)
```

3 Benchmark by aiDAPTIV Toolkit (Optional)

In this section, you will learn how to use aiDAPTIV Toolkit to check you machine performance.

Suggest to do benchmark by aiDAPTIV Toolkit, if your HW configuration is different to AVL or you want to test max batch size in your machine.
You can refer 3.3 to setup your batch size.

After completing this section, you will know that aiDAPTIV Toolkit is a specific evaluation tool which to help user to findout the best betch size setting for finetuning.

3.1 aiDAPTIV Toolkit Installtion flow

- aiDAPTIV Toolkit

```
wget https://phisonbucket.s3.ap-northeast-1.amazonaws.com/toolkit_v2_3_1.zip
```

- Unzip download file

```
unzip toolkit_v2_3_1.zip
```

- change to folder

```
cd toolkit_v2_3_1
```

- Deploy aiDAPTIV Toolkit and related library

```
pip install -r requirements.txt
sudo apt install nvme-cli
```

- Example

3.2 Start aiDAPTIV Toolkit

- Modify training setting in toolkit_v2_3_1/project.ini
  You can follow default project.ini setting.

```
[ENV_setting]
# Specify the training GPU index
specify_gpu_index=0,1
# GPU number of model training
num_gpus=2
# Training model path in local
model_name_or_path=/home/$USER/Desktop/llm/Llama-3.1-8B-Instruct
# Path of aiDAPTIVCache
nvme_path=/mnt/nvme0
# Password of root
pwd=test

[Performance_test]
# Start batch size of performance test
start_bs=8
# End batch size of performance test
end_bs=100
# Sequence length while training LLM Model
seq_len=2048
# Expect training time in hour
training_hour=0.5
# Enable Triton
triton=True
```

- Enter aiDAPTIV Toolkit folder

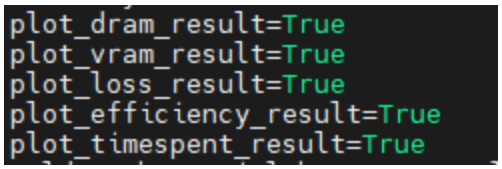
```
cd toolkit_v2_3_1/Script/Model_Test
```

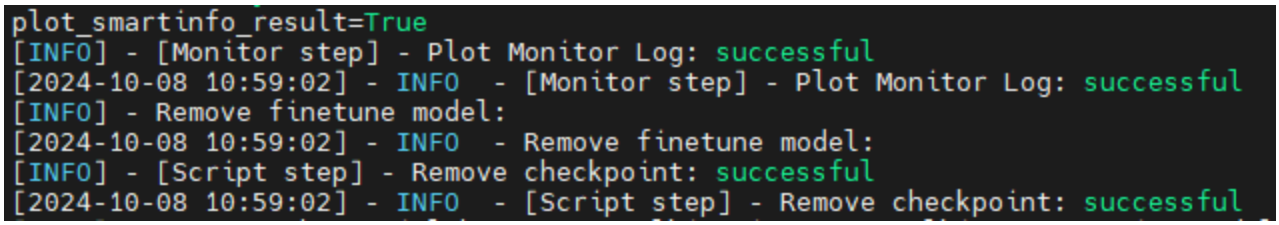
- Run aiDAPTIV Toolkit Performance Test

```
python3 aidaptest_run.py --t 1
```

```
  * Type == 1 (--t 1): Test performance
  * Type == 2 (--t 2): Find max batchsize
  * Type == 3 (--t 3): Find max batchsize + Test performance
```

- Successful example
  aiDAPTIVTest would be stop while finish.



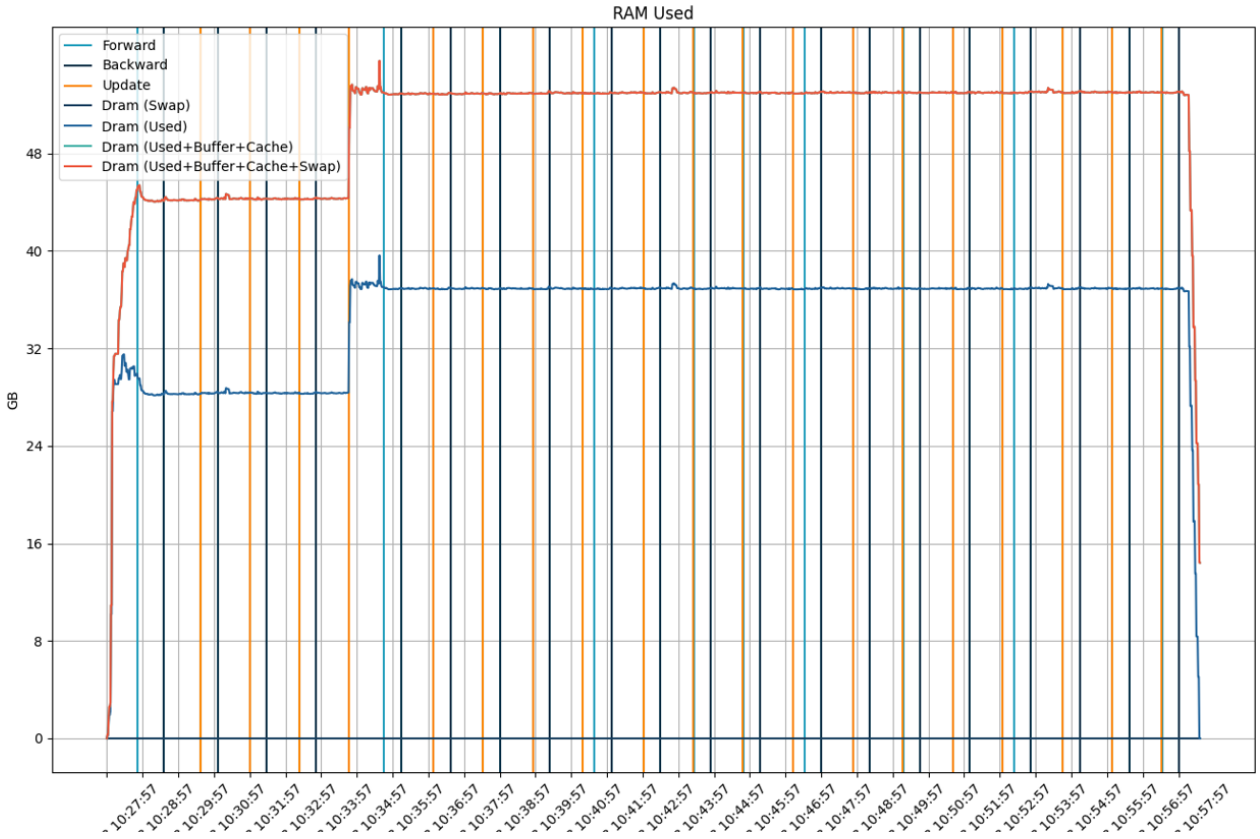


- Result
  After using the aiDAPTIV Toolkit, the results will be stored in toolkit_v2_3_1/Log directory.

3.3 Performance Result

3.3.1 Log directory Decription

1. Figure_4GPU_41bs (Model: Llama-3.1-8B-Instruct)
   This folder will store figure of DRAM usage, GPU usage, Forward time, Backward time, Update time, training Loss and training speed.



2. Training log 4GPU_41bs (Model: Llama-3.1-8B-Instruct)

   This log will store the output of terminal during aiDAPTIV training

```
41  Beginning of Epoch 1/1000, Total Micro Batches 61
42  [2024-10-08 10:28:49.499854] [PHISON START] Epoch: 0, Iteration: 0
43  [2024-10-08 10:28:49.510884] [Forward][start]
44  [2024-10-08 10:29:28.401107] [Forward][time spent]:38.89015984535217
45  [2024-10-08 10:29:32.831248] [Loss]:2.870795249938965
46  [2024-10-08 10:29:32.831346] [Backward][start]
47  [2024-10-08 10:30:35.113132] [Backward][time spent]:62.281774282455444
48  [2024-10-08 10:30:35.113201] [Update][Start]
49  [2024-10-08 10:30:35.446244] [Update][time spent]:0.33303284645080566
50  [2024-10-08 10:30:35.446299] [PHISON END] Iteration: 0
```

3. performance_result.xlsx (Model: Llama-3.1-8B-Instruct)

   This Excel will organize the training information for various parameters in the aiDAPTIV Toolkit Performance test.

A	B	C	D	E	F
Test Model	GPU Used Num	Dataset	Seq_len	Gradient_accumulation_steps	
Meta-Llama-3.1-8B-Instruct	4	['../../products/aiDAPTIVLi	2048	4	
Batch size	1st Efficiency(token/s)	2nd Efficiency(token/s)	3rd Efficiency(token/s)	Avg Efficiency(2~3)(token/s)	10M Dataset training time
40	3129.31	3601.95	3780.97	3691.46	2708.96 seconds
41	3391.7	3611.48	3789.91	3700.695	2702.2 seconds

3.3.2 Performance

- The maximum batch size's average efficiency and the time required to train a 10M dataset can be found in the Performance Info sheet of performance_result.xlsx (see the red box in the figure below).

- Model: Llama-3.1-8B-Instruct (with triton)

A	B	C	D	E	F
Test Model	GPU Used Num	Dataset	Seq_len	Gradient_accumulation_steps	
Meta-Llama-3.1-8B-Instruct	4	['../../products/aiDAPTIVLin	2048	4	
Batch size	1st Efficiency(token/s)	2nd Efficiency(token/s)	3rd Efficiency(token/s)	Avg Efficiency(2~3)(token/s)	10M Dataset training time
40	3129.31	3601.95	3780.97	3691.46	2708.96 seconds
41	3391.7	3611.48	3789.91	3700.695	2702.2 seconds

- You can optimize hardware configuration to enhance training efficiency and reduce training dataset time.

4 Dataset config

4.1 QA datasets

All QA datasets (local or from Huggingface) should be configured in QA_dataset_config.yaml\

Rules

- Use **- - -** to seperate multiple dataset
- Follow below format, where **DATASET NAME** must be different

```
<DATASET NAME 1>:
  data_path:
  strategy: "qa"
  system_prompt:
  user_prompt:
  question_key:
  answer_key:
  exp_type:
  label_key:
---
<DATASET NAME 2>:
  .
  .
  .
```

Description for each value

key	description
data_path	local path or huggingface dataset repository name
strategy	**qa**
system_prompt	system_prompt
user_prompt	user prompt should include **{question}** for question insertion (ex. user_prompt : solve the question {question})
question_key	key for question **(NOT SUPPORT NESTED FORMAT)**
answer_key	key for answer **(NOT SUPPORT NESTED FORMAT)**
exp_type	**train** or **inference** or **eval**
label_key	same as answer key

> **NOTE**
> if dataset is multiturn chat, the dataset should be converted to **question_key : [question1, question2, question3, ...], answer_key : [answer1, answer2, answer3, ...]**

4.2 Pretrain datasets

All Pretrain datasets (local or from Huggingface) should be configured in Pretrain_dataset_config.yaml

Rules

- Use **- - -** to seperate multiple dataset
- Follow following format, where **DATASET NAME** must be different

```
<DATASET NAME>:
  data_path:
  strategy: "pretrain"
  system_prompt:
  user_prompt:
  text:
  exp_type:  ## train or inference or eval
---
<DATASET NAME>:
  .
  .
  .
```

Description for each value

key	description
data_path	local path or huggingface dataset repository name
strategy	**pretrain**
system_prompt	system_prompt
user_prompt	user prompt should include **{question}** for question insertion (ex. user_prompt : solve the question {question})
text	key name for pretrain text **(NOT SUPPORT NESTED FORMAT)**
exp_type	**train** or **inference** or **eval**

4.3 RAG datasets

All RAG datasets (local or from Huggingface) should be configured in RAG_dataset_config.yaml

Rules

- use **—** to seperate multiple dataset
- follow the format, where **DATASET NAME** must be different

```
<DATASET NAME>:
  data_path:
  strategy: "rag"
  system_prompt:
  user_prompt:
  question_key:
  answer_key:
  rag_key:
  exp_type:
  label_key:
---
<DATASET NAME>:
  .
  .
  .
```

4.4 Description for each value

key	description
data_path	local path or huggingface dataset repository name
strategy	**rag**
system_prompt	[optional] system_prompt
user_prompt	user prompt should include **{question}** for question insertion and **{rag}** for rag data insertion (ex. Context: {rag}, based on the context answer the question : {question})
question_key	key for question
answer_key	key for answer
rag_key	key for rag data, **(SUPPORT NESTED FORMAT)** **
exp_type	**train** or **inference** or **eval**
label_key	same as answer key

```
** if RAG data has format
{context : {sentences : [RAG DATA]}}
the rag_key can be given

rag_key:
 - context
 - sentences
```

NOTE
if dataset is multiturn chat, the dataset should be convert to **question_key : [question1, question2, question3, ...], answer_key : [answer1, answer2, answer3, ...]**

4.4.1 example 1 (QA dataset)

```
{
    "question": "......",
    "cot_answer": "........"
},
```

```
<DATASET NAME>:
  data_path:
  strategy: "qa"
  system_prompt:
  user_prompt: answer the following question {question}
  question_key: question
  answer_key: cot_answer
  exp_type:
  label_key:
```

4.4.2 example 2 (RAG dataset)

```
{
    "question": [
        "....",
        "....",
        "....."
    ],
    "answer": [
        "....",
        "....",
        "....."
    ],
    "hybrid_chunks": [
        "....",
        "....",
        "....",
    ]
},
```

```
<DATASET NAME>:
  data_path:
  strategy: "rag"
  system_prompt:
  user_prompt: you are a doctor with this background knowledge {rag}, now solve the fo
llowing question {question}
  question_key: question
  answer_key: answer
  rag_key:
   - hybrid_chunks
  exp_type:
  label_key: answer
```

4.4.3 example 3 (nested RAG dataset)

```
"question": "...",
"cot_answer": "...",
"context": {
    "sentences": [
        "....."
    ]
}
```

```
<DATASET NAME>:
  data_path:
  strategy: "rag"
  system_prompt:
  user_prompt: you are a doctor with this background knowledge {rag}, now solve the fo
llowing question {question}
  question_key: question
  answer_key: answer
  rag_key:
    - context
    - sentence
    - 0
  exp_type:
  label_key: answer
```

4.5 Compare Change

> origin method seen rm-static as pretrain data, hence no system and user prompt and template

4.5.1 previous dataloader on Dahoas/rm-static after apply template

```
Human: How can I cook delicata squash?

Assistant: Try cutting it in half, cutting away the seeds and stringy parts, and layin
g it flat in a 400°F oven for 1 hour.  You can drizzle it with olive oil and salt befo
re you roast it if you want.

Human: Can you fry it?

Assistant: Try taking the squash halves you’ve roasted, and frying them in oil at a me
dium heat until it begins to brown and crisp on all sides.  Taste it to see if it need
s salt or spices.<|eot_id|>
```

4.5.2 Current dataloader on Dahoas/rm-static after apply template

```
<|begin_of_text|><|start_header_id|>system<|end_header_id|>

This is a chat between a user and an artificial intelligence assistant. The assistant
gives helpful, detailed, and polite answers to the user's questions based on the conte
xt. The assistant should also indicate when the answer cannot be found in the context.
<|eot_id|><|start_header_id|>user<|end_header_id|>

solve the question below :

Human: How can I cook delicata squash?

Assistant: Try cutting it in half, cutting away the seeds and stringy parts, and layin
g it flat in a 400°F oven for 1 hour.  You can drizzle it with olive oil and salt befo
re you roast it if you want.

Human: Can you fry it?

Assistant:<|eot_id|><|start_header_id|>assistant<|end_header_id|>

Try taking the squash halves you’ve roasted, and frying them in oil at a medium heat u
ntil it begins to brown and crisp on all sides.  Taste it to see if it needs salt or s
pices.<|eot_id|><|start_header_id|>assistant<|end_header_id|>
```

5 Multimodality Dataset config

1. When adding dataset, the yaml file under /commands/image-text-to-text/env_configs need to be configured
2. Currently only support
    - VQA datasets : QA dataset with vision
3. After configured, the yaml file path should be added to commands/env_config/env_config.yaml ``` yaml = data_path:
    - ./dataset_config/image-text-to-text/VQA_dataset_config.yaml```

- to run the default env_config.yaml, the execute path should locate at aiDAPTIV2/commands

5.1 VQA datasets

All VQA datasets (local or from Huggingface) should be configured in image-text-to-text/VQA_dataset_config.yaml

- Rules - use `"---"` to seperate multiple dataset - follow below format, where DATASET NAME must be different

```
<DATASET NAME>:
  data_path:
  image_folder : ""
  strategy: "qa"
  system_prompt: ""
  user_prompt: "{question}"
  question_key: ""
  answer_key: ""
  image_key : ""
  label_key: ""
  ----
  <DATASET NAME>:
```

5.2 description for each value

key	description
data_path	local path / local folder or huggingface dataset repository
image_folder	The image folder for the custom dataset will be concatenated with the path within the dataset.
strategy	**qa**
system_prompt	system_prompt
user_prompt	user prompt should include **{question}** for question insertion (ex. user_prompt : solve the question {question})
question_key	key for question
rag_key	key for rag data, **(SUPPORT NESTED FORMAT)** **
answer_key	key for answer
image_key	The key for the image content can be a URL, local file path, or a PIL Image. Leave this field blank for text-only datasets.
label_key	same as answer key

5.3 Supported dataset :

1. Repository from huggingface : With given repository name we support automatically download from huggingface. (ex. Multimodal-Fatima/OK-VQA_train)
2. Local huggingface dataset : If the dataset is cloned or downloaded from HuggingFace and comprises *.parquet files, designate the root folder as the data_path for subsequent steps.
    - Ex. for following Directory structure, the data_path should be Desktop/OK-VQA_train

```
Desktop/OK-VQA_train
    |- data
        |- train-0000-of-0004.parquet
        |- train-0001-of-0004.parquet
        |-            .
        |-            .
    |-README.md
```

5.3.1 Custom dataset : The data path should be in the format of **./.json or ./.csv** to properly parse the data.

- Dataset_config example (QA dataset)
  For example, utilize AI4Math/MathVista in HuggingFace for demonstration. The header include [pid, question, image, decode_image, choice, unit, precision, answer, ...]

The Dataset configuration can be set up as follows without the need to download the entire dataset locally.

5.3.2 config example (MathVista QA dataset)

```
<DATASET NAME>:
  data_path: "AI4Math/MathVista"
  image_folder : ""
  strategy: "qa"
  system_prompt: "you are a helpful asssistant"
  user_prompt: "Please address the following question using the accompanying image.
  {question}"
  question_key: "question"
  answer_key: "answer"
  image_key : "decode_image"
  label_key: ""
```

5.3.3 dataset example

- The header in train.json should include [question, decode_image, answer] at least.

```
// train.json
[
    {
        "id": data_id,
        "question": question_to_ask,
        "answer": ["answer1_to_reponse", "answer2_to_reponse", ...],
        "image": image_path,
        .
        .
        .
    },
    .
    .
    .
]
```

- `data_id` is the id of the data
- `question_to_ask` is the question
- `answer_to_reponse` is the answer of the data. If there are multiple answers, it should be in a list.
- `image_path` is the path of the image

5.3.4 dataset structure example

```
Desktop/custom_dataset
    |- train.json
    |- image_folder
        |- image1
        |- image2
        |-            .
        |-            .
    |-README.md
```

- Only support .json or .csv for Custom dataset

5.3.5 Custom dataset config example

```
<DATASET NAME>:
  data_path: "Desktop/custom_dataset/train.json"
  image_folder : "Desktop/custom_dataset/image_folder"
  strategy: "qa"
  system_prompt: "The prompt you want before the training data."
  user_prompt: "The prompt with the question {question}"
  question_key: "question"
  answer_key: "answer"
  image_key : "image"
  label_key: "label"
---
<DATASET NAME>:
```

The data with system_prompt and user_prompt used for training would become :

```
"The prompt you want before the training data."
image in dataset
"The prompt with the question {`question in dataset`}"
```

Appendix A. How to train model in Docker.

In this section, you will learn how to use docker to do your training. You can use GPU resource and aiDAPTIVCache in docker.

A.1 Docker Installation option

- Docker offical website
  https://docs.docker.com/engine/install/ubuntu/
- Download NVIDIA Container Toolkit
  https://docs.nvidia.com/datacenter/cloud-native/container-toolkit/latest/install-guide.html

  ```
  # Restart docker service to adopt the change
  sudo systemctl restart docker
  ```

- Docker image

```
wget https://phisonbucket.s3.ap-northeast-1.amazonaws.com/aiDAPTIV_vNXUN_2_04_A1.tar.gz
```

- Load docker image

```
docker load < aiDAPTIV_vNXUN_2_04_A1.tar.gz
```

```
phison@linux-AH240023-741GE-TNRT:~$ docker load < aiDAPTIV_vNXUN_2_04_A1.tar.gz
2f9257129af9: Loading layer [==================================================>]  59.61MB/59.61MB
bd673114a197: Loading layer [==================================================>]  23.55MB/23.55MB
a336d27150b5: Loading layer [==================================================>]  22.53kB/22.53kB
6cde63a39768: Loading layer [==================================================>]  245.5MB/245.5MB
e082bc1a8261: Loading layer [==================================================>]  103.6MB/103.6MB
a39bd610731b: Loading layer [==================================================>]   12.6MB/12.6MB
2e662c545a4f: Loading layer [==================================================>]  23.58MB/23.58MB
225d429ae71b: Loading layer [==================================================>]   9.36GB/9.36GB
c2dc0e6b991f: Loading layer [==================================================>]  4.096kB/4.096kB
Loaded image: aidaptiv:vNXUN_2_04_A1
```

- Check docker image list

```
docker image list
```

```
phison@linux-AH240023-741GE-TNRT:~$ docker images
REPOSITORY   TAG              IMAGE ID       CREATED         SIZE
aidaptiv     vNXUN_2_04_A1    a19b7faf2e1c   2 hours ago     9.84GB
```

- The commands folder within docker will be deployed at:

```
/home/root/aiDAPTIV2/commands
```

There will be two folders env_config, exp_config and example.sh. There will be env_config.yaml and exp_config.yaml in env_config and exp_config respectively, which can be modified directly.

```
--commands
   | --env_config
       |--env_config.yaml
   | --exp_config
       |--env_config.yaml
   | --example.sh
```

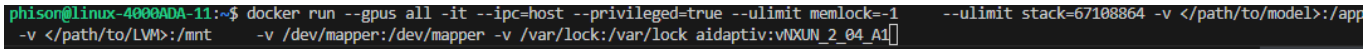
A.2 Run aiDAPTIV Image

If you are deploying the environment using Docker, please execute the following. If you are using a native environment, please ignore this item.

- docker and nvidia gpu runtime Installation
  - https://docs.docker.com/engine/install/
  - https://docs.nvidia.com/datacenter/cloud-native/container-toolkit/latest/install-guide.html

```
docker run --gpus all -it --ipc=host --privileged=true --ulimit memlock=-1 \
--ulimit stack=67108864 -v </path/to/model>:/app -v </path/to/LVM>:/mnt \
-v /dev/mapper:/dev/mapper -v /var/lock:/var/lock aidaptiv:vNXUN_2_04_A1
```

- Successful example



Appendix B. How to set Swap File

- Enable swapping provides extra memory for DRAM. This can extend the range of batch size that you can use if you still have enough memory on the GPU.

```
# Create swap file
sudo dd if=/dev/zero of=/mnt/nvme0/swapfile bs=1M count=256k
# Modify permission
sudo chmod 0600 /mnt/nvme0/swapfile
# Initialize swap file
sudo mkswap /mnt/nvme0/swapfile
# Enable the swap
sudo swapon /mnt/nvme0/swapfile
# Make the swap permanent
sudo echo '/mnt/nvme0/swapfile none swap sw 0 0' | sudo tee -a /etc/fstab
```

- If you would like to remove the swap or unplug aiDAPTIVCash, please make sure to follow the steps below to prevent unexpected system issues.

```
# Disable the swap
sudo swapoff /mnt/nvme0/swapfile
# Remove permanent swap setting
sudo sed -i '/\/mnt\/nvme0\/swapfile/d' /etc/fstab
# Remove swapfile (optional)
sudo rm /mnt/nvme0/swapfile
```

Appendix C. Approved Vendor List (AVL)

- GPU AVL list

Vendor	Product Name	Bus	Memory
NVIDIA	H100	PCIe 4.0 x16	80 GB, HBM2e, 5120 bit
NVIDIA	RTX50 series		
NVIDIA	RTX A6000	PCIe 4.0 x16	48 GB, GDDR6, 384 bit
NVIDIA	RTX A5000	PCIe 4.0 x16	24 GB, GDDR6, 384 bit
NVIDIA	GeForce RTX 4090	PCIe 4.0 x16	24 GB, GDDR6X, 384 bit
NVIDIA	L40	PCIe 4.0 x16	48 GB, GDDR6, 384 bit
NVIDIA	L40S	PCIe 4.0 x16	48 GB, GDDR6, 384 bit
NVIDIA	RTX 6000 Ada Generation	PCIe 4.0 x16	48 GB, GDDR6, 384 bit
NVIDIA	GeForce RTX 4090 D	PCIe 4.0 x16	24 GB, GDDR6X, 384 bit
NVIDIA	RTX 4000 Ada Generation	PCIe 4.0 x16	20 GB, GDDR6, 160 bit
NVIDIA	RTX 4000 SFF Ada Generation	PCIe 4.0 x16	20 GB, GDDR6, 160 bit
NVIDIA	RTX 5000 Ada Generation	PCIe 4.0 x16	32 GB, GDDR6, 256 bit

- CPU AVL

Brand	Naming	Count	Cores	Clk	Lanes
Intel	Xeon Gold 5320	2	26	2.2	64
Intel	Xeon Gold 6330	2	28	2	64
Intel	Xeon w5-3425	1	12	3.2	112
Intel	Xeon Gold 6538Y+	2	32	2.2	80
Intel	Xeon Silver 4410T	2	10	2.7	80
Intel	i9-13900	1	24	1.5	20
Intel	i9-12900E	1	16	1.7	20
Intel	Xeon Silver 4410Y	2	8	2	80
Intel	Xeon Silver 5315Y	2	8	3.2	64
AMD	Ryzen Threadripper 7980X 64-Cores	1	64	5.1	92
AMD	EPYC 7713P	1	64	2	128
AMD	EPYC 9174F	1	16	4.1	128

- Support Model list
  RTX 4000 Ada :

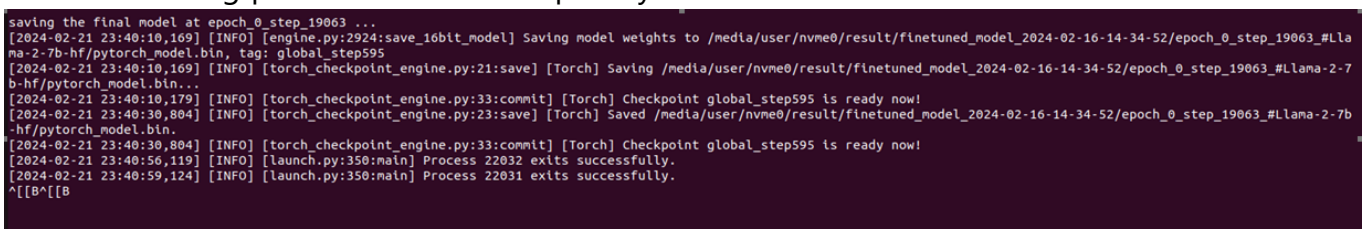
Task Type	Model Name	參數規模(M/B)	模型大小(MB/GB)
fill-mask	Bert	110M	104MB
text-generation	Llama II	7B	13 GB
text-generation	Mistral	7B	15 GB
text-generation	Mixtral 8x7B	8X7B	87 GB
text-generation	Baichuan2-7B-Chat	7B	14 GB
text-generation	chatglm3-6b	6B	11 GB
text-generation	Gemma2_9b	9B	18 GB
text-generation	Llama3.1_8b	8B	15 GB
text-generation	Meta-Llama-3-8B	8B	15 GB
text-generation	Qwen2-7B	7B	15 GB
text-generation	Yi-1.5-6B	6B	12 GB
text-generation	Yuan2-M32-hf	40B	75 GB
text-generation	ko-llm-llama-2-7b-LoRA-IA3	627M	1.25 GB
image-text-to-text	InternVL2-8B	8B	16.4 GB
image-text-to-text	Llama-3.2-11B-Vision-Instruct	10.7B	21.5 GB
image-text-to-text	llava-1.5-7b-hf	7B	14.2 GB
image-text-to-text	Qwen2-VL-7B-Instruct	7B	17 GB

RTX 5080 :

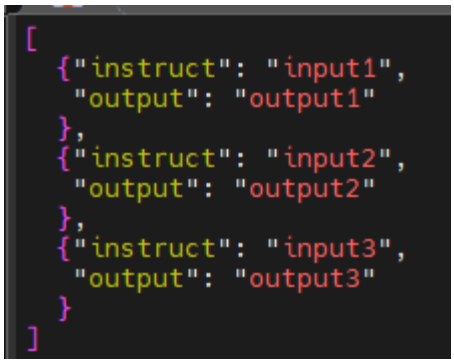
Task Type	Model Name	參數規模(M/B)	模型大小(MB/GB)
text-generation	Llama3.1_8b	8B	15 GB
text-generation	Qwen2.5-7B	7B	15 GB

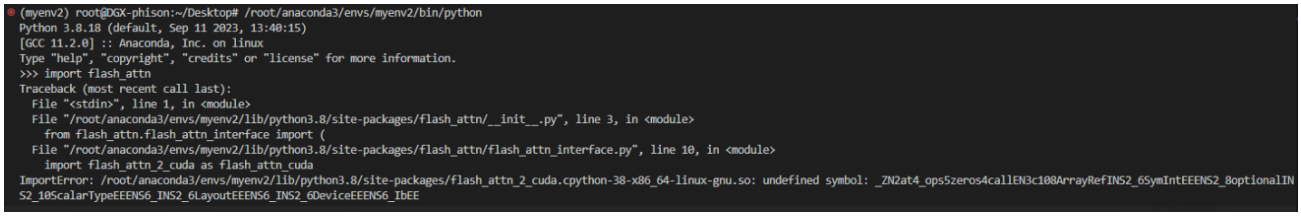
Q&A EN

- Q1. If I want to change the directory for downloading the LLAMA 2 Model, do I just need to modify the download directory path in setup.sh?
    - You can directly modify the path in setup.sh, but it is recommended to download first according to the original setup.sh and then move it. Remember to also change the model path in the phisonai2 parameter Model path after modification.
- Q2. How can I find out the number of tokens in datasets on HuggingFace? Is there a way to calculate the number of tokens?
    - The number of tokens in one training epoch = number of data rows * token length (default is 2048).
    - Taking the Hugging Face open dataset rm-static as an example, the number of tokens in one epoch is 76300 * 2048 = 156M (tokens).
- Q3. Can all datasets on HuggingFace be used?
    - In the source code's utils/data/data_utils.py, the get_raw_dataset function lists datasets. We have only tested Dahoas/rm-static, others can be tried but we cannot guarantee successful training.
- Q4. How can I view the test data and results?
    - During training, the efficiency of training will be logged, showing tokens/s, which indicates how many tokens are trained per second on the given device conditions. The trained model will output to the specified output directory.
- Q5. The training process has not completely terminated. Is this normal?



    - You can use tail -f filename to view a log file. Press Ctrl+C to exit. (Detailed explanation of the tail command)
- Q6. I want to use my custom Dataset.
    - Create a test.json file with the following format and input it into the --data_path parameter.

- 

- Q7. Supported 33B Model
  - https://huggingface.co/lmsys/vicuna-33b-v1.3
- Q8. Is Training Efficiency normal?
  - Provide the following information:
    1. GPU model, quantity
    2. Number of SSDs in Raid0
    3. DRAM Size
    4. CPU model, quantity
    5. Executed Command parameters
- Q9. Will the speed of the disk where the model is stored affect the training speed?
  - no
- Q10. Flash_attn undefined symbol
  - 

  - Please refer to https://phisonbucket.s3.ap-northeast-1.amazonaws.com/patch_0220.txt
- Q11 Phison aiDAPTIV+ not support!
  1. Confirm the disk status and check if the AI100 is mounted.
     - lsblk
     - df -h
  2. phisonai2 command
     - Provide the phisonai2 command to confirm if nvme_path is running on AI100.
  - The example from the customer

        - The output of lsblk shows a line with mdxxx, indicating a RAID array, typically associated with AI100. You can see that it is mounted to the Ubuntu system at /media/test/820f95...
        - phisonai2 --nvme_path沒有跑在系統掛載的位置(/media/test/820f95...)

- Q12 How to observe Phison SSD Usage
    - Use the nmon tool to monitor.
- Q13 Error Code: Attribute busid of node nic not found
    - use --specify_gpus to specify GPUs. If you want to use No3 and No2 GPU, set specify_gpus="3,2".
- Q14 Unable to JIT load the async_io op due to it not being compatible due to hardware/software issue.
    - sudo apt install libaio-dev
- Q15 Cmath error
    - sudo apt install libstdc++-12-dev
- Q16 mdadm: cannot open /dev/nvme0n1p1: Device or resource busy
    - It may be due to unresolved RAID. Run sudo mdadm stop

```
sudo mdadm  --stop /dev/md*
```

- Q17 Permission denied: '/media/user/md4
    - sudo chmod 777 -R /media/user/md4 (https://github.com/microsoft/DeepSpeed/pull/4538)
- Q18 You need to have sentencepiece installed to convert a slow tokenizer to a fast one.

```
pip install --user sentencepiece
```

- Q19 vicuna 33b about protobuf

```
pip install --upgrade protobuf
```

- Q20 GPU Fan Error
    - https://askubuntu.com/questions/16371/how-do-i-disable-x-at-boot-time-so-that-the-system-boots-in-text-mode
- Q21 /dev/ubuntu-vg/ubuntu-lv out of space

```
sudo lvextend -l +100%FREE /dev/ubuntu-vg/ubuntu-lv
sudo resize2fs /dev/ubuntu-vg/ubuntu-lv
```

- Q22 ImportError: libcudnn.so.8: cannot open shared object file: No such file or directory
    - uninstall the existing version of torch and reinstall torch with CUDA support.

```
pip install torch==2.1.2 torchvision==0.16.2 torchaudio==2.1.2 --index-url \
https://download.pytorch.org/whl/cu121
```

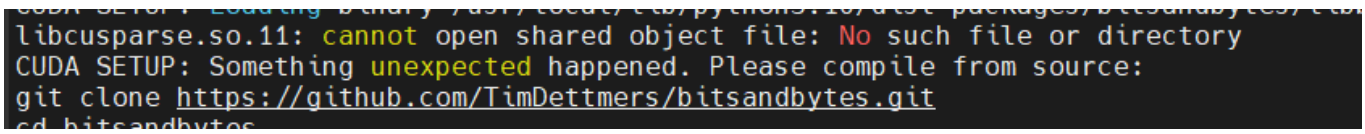
- Q23 ImportError: libcudnn.so.11: cannot open shared object file: No such file or directory
    - Reinstall transformers and flash-attn.

```
pip uninstall transformers flash-attn
pip install transformers==4.35.2 flash_attn-2.4.2+cu122torch2.1cxx11abiFALSE-cp310-cp310-linux_x86_64.whl
```

- Q24 RuntimeError: Ninja is required to load C++ extensions
    - If in a virtual environment, ensure to use the outermost ninja module by running conda deactivate to the outer layer, then:

```
conda deactivate
pip install ninja
```

- Q25



    - First run pip list | grep nv to confirm if the CUDA version matches the version installed with torch. If they are different, uninstall and reinstall torch

    - https://github.com/TimDettmers/bitsandbytes/issues/63： `sudo ldconfig /usr/local/cuda/lib64`
    - There may be an issue in the root environment.
- Q26 Encountering ninja build issue, missing python.h
    - Refer to https://stackoverflow.com/questions/21530577/fatal-error-python-h-no-such-file-or-directory
- Q27 How to install aiDAPTIV if I use RockyLinux 9.3
    - Please update your Linux kernel version to 6.8.9 first, and follow the installation steps above.

Contact Window and Resource

- Contact Window
  aiDaptivCache_support@phison.com
- Resource
  https://aidaptiv.phison.ai/