aiDAPTIV+
Pro Suite 2.0 50 GPU
Support User Guide

Version 1.0

Phison Electronics Corporation
Tel: +886-37-586-896 Fax: +886-37-587-868
E-mail: sales@phison.com / suppport@phison.com

Phison may make changes to specifications and product description at any time without notice. PHISON and the Phison logo are trademarks of Phison Electronics Corporation, registered in the United States and other countries. Products and specifications discussed herein are for reference purposes only. Copies of documents which include information of part number or ordering number, or other materials may be obtained by emailing us at sales@phison.com or support@phison.com.

REVISION HISTORY

Revision	Draft Date	History	Pro Suite Version	Author
0.1	2025/09/10	Draft version	NXUN_2.0.7	Sean Liou
1.0	2025/09/16	First release	NXUN_2.0.7	Sean Liou

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. ENVIRONMENT PREPARATION

1.1. Supported OS and Nvidia driver version

Category	Detail
OS	Ubuntu 24.04 LTS Desktop
GPU driver	Nvidia driver version 570 or later version
Kernel	6.8.0-41-generic

Note: Due to the current Pro Suite version only matching this kernel version, when installing the OS, **DO NOT** connect to the internet to avoid automatic updates to the kernel version.

1.2. Browser suggestion and precaution

- **Google Chrome** (The recommended default browser for use with the Pro Suite service.)
- **Mozilla Firefox**

Note : When logging in for the first time, you may log in with the following account

Default system administrator account password
Account: admin@aidaptiv.com
Password: Admin8299

2. DESCRIPTION

The aiDAPTIV+ Pro Suite is a web-based GUI program that enables a **No Code** approach to model training. It streamlines the entire process from **Dataset generation**, **Fine-Tuning**, **Validation** to **Inference**. This allows users to quickly convert documents into files that can be used for training their own fine-tuned models, and build their own AI models.

3. FUNCTION INTRODUCTION

Users can access Pro Suite main functions through the tabs at the top of the webpage. Below are detailed instructions for each function.

1. Dataset
   - aiDAPTIVGuru
2. Fine-tune
3. Monitor
4. Validation
5. Benchmark (Option)
6. Inference
7. Models
8. Management

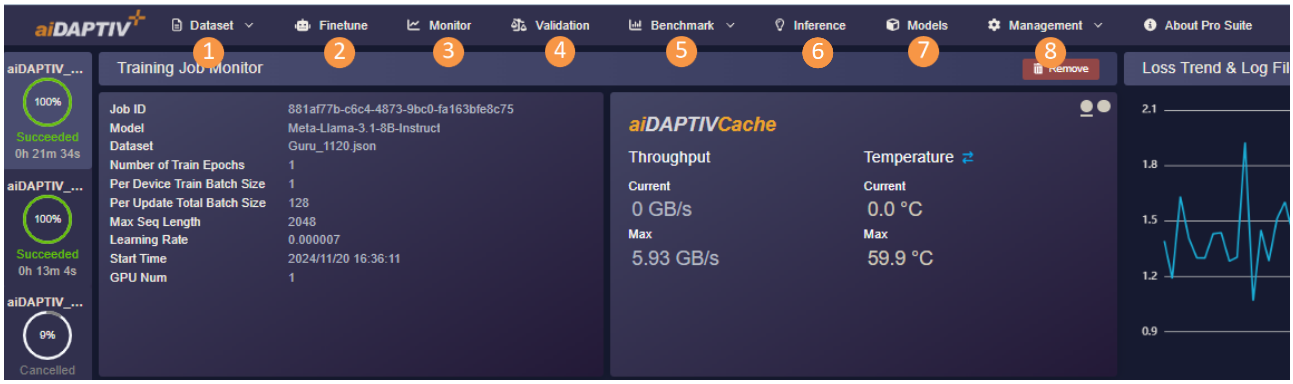


Figure 3-1 Pro Suite main function

3.1. Dataset

There're 2 main functions in the Dataset tab: Upload and aiDAPTIVGuru. The Upload function allows users to upload an existing dataset to Pro Suite and manage the uploaded datasets. After clicking the Upload tab, users will see the page below. This page is divided into the upload area on the left and list area on the right.

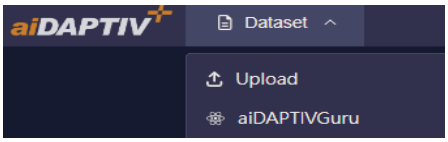


Figure 3-2 Dataset

3.1.1. Upload

3.1.1.1. Dataset upload

- Field description:
  - File upload location : Support **JSON, JSONL and Parquet** file format. (Upload **one** file at a time)
- Function description:
  - Upload
  - : Remove temporary files from the storage area

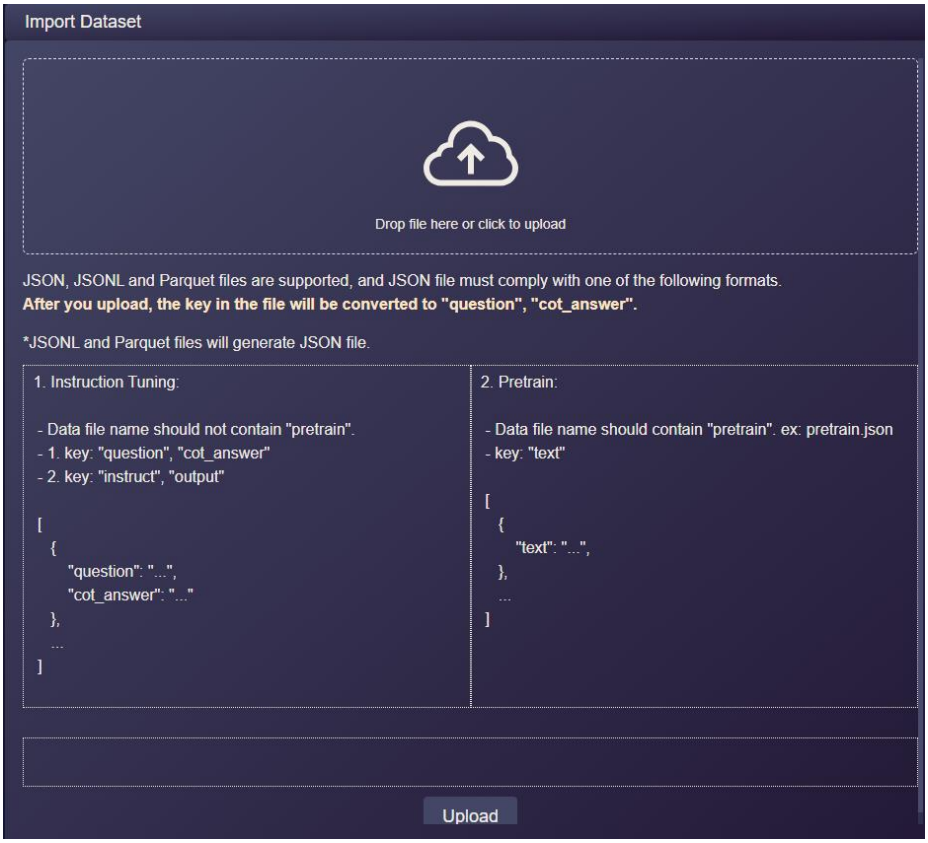


Figure 3-3 Dataset upload

- JSONL and Parquet dataset upload
  - Step1: Upload JSONL or Parquet dataset.
  - Step2: Select the corresponding Key values in the Question and Answer fields.

3.1.1.2. Dataset Management

- Function description:

1. : View dataset content
2. : Download dataset
3. : Delete dataset

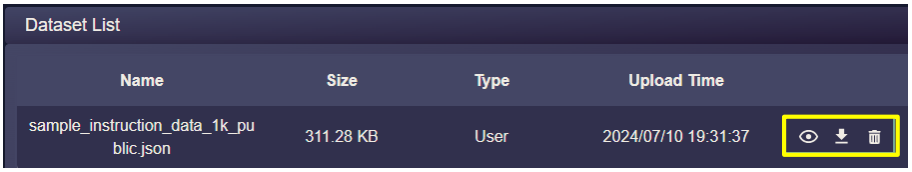


Figure 3-4 Dataset management

3.1.1.3. Example of dataset file format

- Instruction Tuning:
    - Data file name **Should NOT** contain "pretrain".
    - Key: “instruct”, “output”

```
[
    {
        "instruct": "What is the more nutrient food in the convenience store?",
        "output": "I think that it might be a big ol chocolate bar."
    },
    {
        "instruct": "Where could I get the best Italian food in town?",
        "output": "In my neighborhood, the food truck right next to the cross street."
    }
        ]
```

- Pretrain:
    - The data file name **Should** contain "pretrain". (For example: pretrain.json)
    - Key: “text”

```
[
    {
        "text": "Regular exercise and a balanced diet are important for maintaining
good health."
    },
    {
        "text": "Drinking enough water and getting an adequate amount of sleep can
contribute to overall well-being."
    }
]
```

3.1.2. aiDAPTIVGuru

Dataset preparation is a very labor-intensive process. aiDAPTIVGuru is a feature of Pro Suite that enhances the dataset generation. It transforms user-provided documents or files with domain-specific knowledge (such as product manuals, technical documents, specifications...etc) into Q&A sets and will automatically create a training dataset.

- *For the best usage of aiDAPTIVGuru, please refer to Section 3.6.2.2*

Table 3-1 Specification of aiDAPTIVGuru

Item	aiDAPTIVGuru_**Entry**	aiDAPTIVGuru_**Pro (Option)**
File Format	pdf、docx、txt	pdf、docx、txt
Supported Model	Llama-3.1-8B-Instruct (-AWQ) DeepSeek-R1-Distill-Qwen-14B-AWQ-INT4	Llama-3.1-8B-Instruct (-AWQ) Llama-3.1-70B-Instruct (-AWQ) DeepSeek-R1-Distill-Qwen-14B-AWQ-INT4
Upload multiple files of same/different formats at once	Y	Y

You can download the model from huggingface and then quantize it using Pro Suite.

Note: aiDAPTIVGuru_Pro is an additional value-added service. For more information on enabling this service, please contact Phison's Sales account.

3.1.2.1. Parameter setting & file upload

aiDAPTIVGuru parameter settings and document upload.

- Field description:
  1. **Dataset Name**
  2. **Model:** The model needs to be pinned first in order to appear in the model list.
  3. **Embedding model**: Retrieval model.
  4. **QA Pairs Count**: Number of Instruction Dataset data.
  5. **Training QA ratio(%)**: The proportion of data in the data set that is used as training data (the remaining proportion is used as scoring data).
  6. **Chunk Size**: Split size of the data file.
  7. **Overlap**: The amount of data overlap when the data file is divided into chunks.
  8. **Number of reference chunk per question**: Number of reference Chunk for each data instruction.
  9. **Chunk Shuffle**: Mix the data chunks or distribute the data evenly.
  10. **Language Evaluation**: After activation, the content generated by the Dataset through Guru will refer to the original document's language. Only supports zh-TW, zh-CN, en-US.
      Note: Please upgrade to the aiDAPTIVGuru_pro version to support this feature.
  11. **Upload File Area**: Domain file upload area when generating a dataset.
- Function description:
  1. **Remove all**: Remove temporary files.
  2. **Upload**: Upload the file for pre-processing.

3.1.2.2. Confirm data pre-processing results

Confirm Inspect the data pre-processing results. The user can edit and adjust data online.

- Function description:
-  : View the pre-processed txt file content and **edit online**.
-  : Download the temporary txt document.
-  : Delete the temporary txt document.
- Remove all: Delete all temporary txt files.

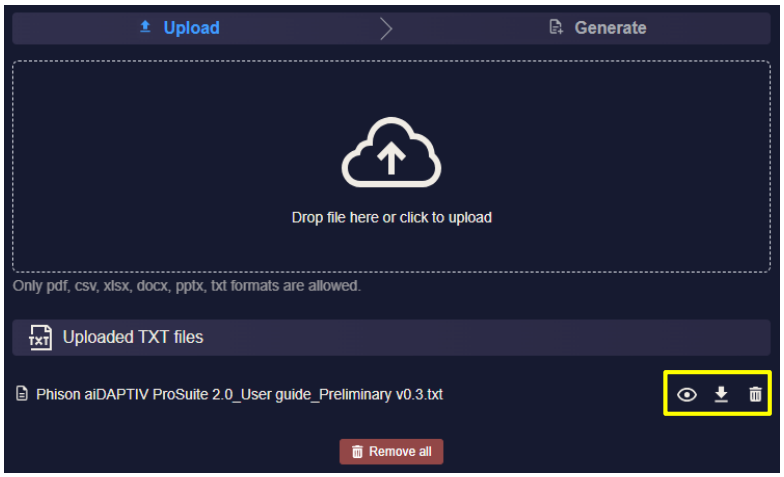


Figure 3-5 aiDAPTIVGuru pre-processed file management

3.1.2.3. Generate Dataset

Execute aiDAPTIVGuru.

- Function description:
  - **Generate :** After confirming the pre-processing results, click “Generate” .

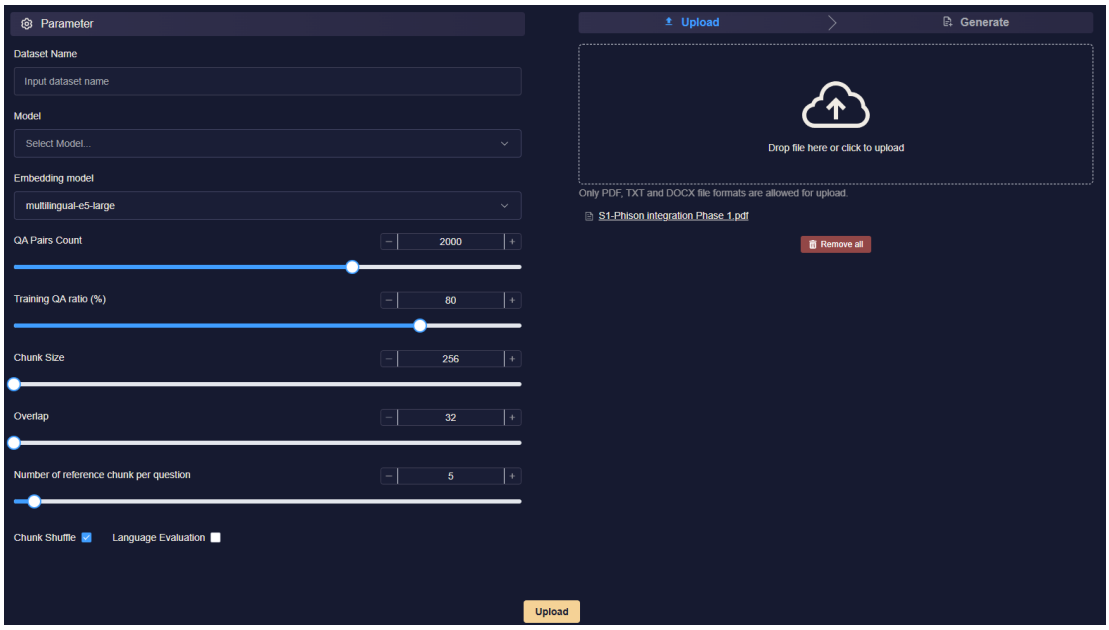


Figure 3-6 Generate dataset

  - **Cancel**

Monitor the dataset generation progress. Pressing “Cancel” will terminate the operation.

Figure 3-7 Dataset generation progress

3.2. Fine-tune

Until aiDAPTIV+, small and medium-sized businesses have been limited to small, imprecise training models with the ability to scale beyond Llama-2 7b.

Phison's aiDAPTIV+ solution enables the training of significantly larger models, giving you the opportunity to run workloads previously reserved for data-centers.

Pro Suite's fine-tune feature is integrated with Phison's aiDAPTIV+ technology, reducing hardware resources.
The function will be divided into three stages, **hardware specification preview**, **parameter setting** and **final confirmation**.

- Number of GPUs = 2^n (n=0,1,2,3,4, GPUs = 1,2,4,8)
- When selecting the number of GPUs, make sure there are enough GPU resources to perform the fine-tuning.
- Please refer to Appendix A for model support list.
- Please refer to Appendix B for recommended hardware configuration.

3.2.1. Hardware specification preview

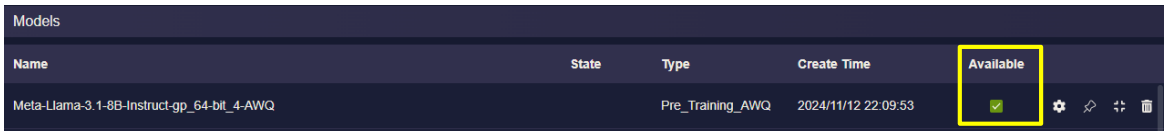
System hardware configuration (GPU, VRAM, system memory, aiDAPTIVLink, aiDAPTIVCache, OS...) .

Item	Information
GPU	1-NVIDIA RTX 4000 Ada Generation 2-NVIDIA RTX 4000 Ada Generation 3-NVIDIA RTX 4000 Ada Generation 4-NVIDIA RTX 4000 Ada Generation
GPU Count	4
VRAM	80 GB
System Memory	503 GB
aiDAPTIVLink	aidaptiv:vNXUN_2_01_00
aiDAPTIVCache : life remaining	/dev/nvme0n1 (1907.73GB) : 100.00% /dev/nvme1n1 (1907.73GB) : 100.00%
OS	Ubuntu 22.04.4 LTS
OS Disk	439 GB

Figure 3-8 Hardware specification preview

3.2.2. Parameter setting

- Field description:
  1. **Model** : Select the model to fine-tune. (Only Pre-training and Fine-tune models will be displayed in the list. AWQ quantified models will not be included in this list.)



Note: The model needs to be available first in order to appear in the model list.

  2. **Dataset**: Select the dataset for fine-tuning.
  3. **Available GPU**: GPU model and number to be used for training
  4. **Epoch**: The number of epochs to train the model. (Range 1 ~ 5, default=1)
  5. **Per Device Train Batch Size**: Batch size for each GPU.
  6. **Per Update Total Batch Size**: Set the total batch size for one update. For example, if you are running on 4 GPUs with per_device_train_batch_size=4 and want to update the model every 80 batches, then you should set the per_update_total_batch_size to 80. The machine will run 80/4/4 = 5 iterations and update the model once. If not divisible, round up to the next whole number.
  7. **Max Seq Length**: Define the maximum sequence length.
     Note: Click the ***Advice*** button to automatically calculate the appropriate Max_Seq_Length value.
  8. **Learning Rate**: Set the learning rate.
  9. **Triton**: Trigger triton training procedure. It can shorten the model training time. (Please refer to Appendix A for the applicable model list.)
     Note: If the user selects a model that does not support Triton for training, the following error message will appear after the training begins: "Phison Accelerator does not support," and the training process will be terminated.
  10. **Job Name:** Allow users to identify different training tasks.

- Function description:
  1. Previous: Return to hardware specifications preview
  2. Next

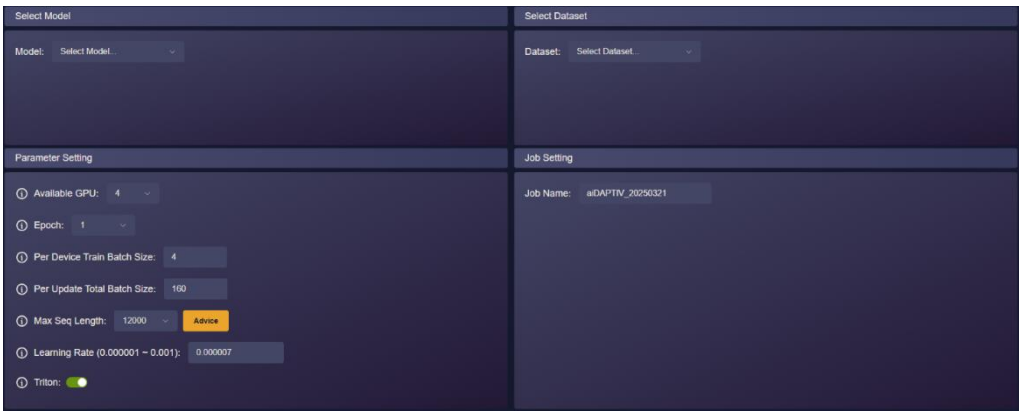


Figure 3-9 Parameters setting for fine-tuning

3.2.3. Confirm hardware configuration and parameter for fine-tuning

- Function description:
  1. Previous: Return to parameter settings.
  2. Run: Execute fine-tune.

Item	Information
GPU	1-NVIDIA GeForce RTX 4090 2-NVIDIA GeForce RTX 4090 3-NVIDIA GeForce RTX 4090 4-NVIDIA GeForce RTX 4090
GPU Count	4
VRAM	96 GB
System Memory	504 GB
aiDAPTIVLink	licensesp/aidaptiv:vNXUN_2_01_00
aiDAPTIVCache : life remaining	/dev/nvme0n1 (1907.73GB) : 100.00% /dev/nvme1n1 (1907.73GB) : 100.00%
OS	Ubuntu 22.04.4 LTS
OS Disk	3519 GB
Model	Meta-Llama-3.1-8B-Instruct
Dataset	sample_instruction_data_1k_public.json
Selected GPUs	4
Batch Size	1
Epoch	1
Per Update Total Batch Size	128
Max Seq Length	2048
Learning Rate	0.000007
Task Name	aiDAPTIV_20241113

Previous Run

Figure 3-10 Final confirmation

3.3. Monitor

Monitor the fine-tuning status, including basic information, progress, hardware resource usage (aiDAPTIVCache, GPU, system memory...) of each fine-tune job, training loss trend chart and complete log of aiDAPTIVLink are available.

- Field description:
  1. List of all finetune jobs (yellow block in the Figure 3-11)
  2. Basic information and hardware usage of a single finetune job (red block in the Figure 3-11)
  3. Trend chart of training loss in a single finetune job (purple block in the Figure 3-11)
  4. Complete Log information of aiDAPTIVCache in a single fine-tune job (orange block in the Figure 3-11)
  5. CPU and Memory usage (blue block in the Figure 3-11)

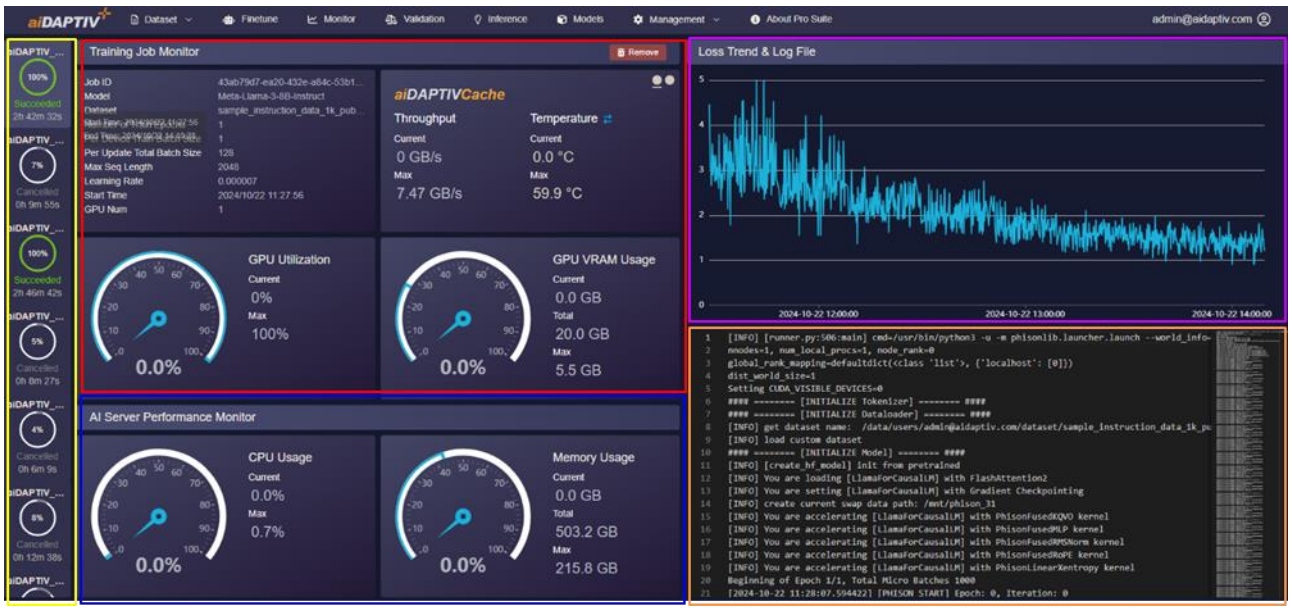


Figure 3-11 Monitor

3.3.1. Cancel job

Only jobs whose status is “Running” can be cancelled. It can take several seconds for the GPU resources to be released when a job has been canceled.

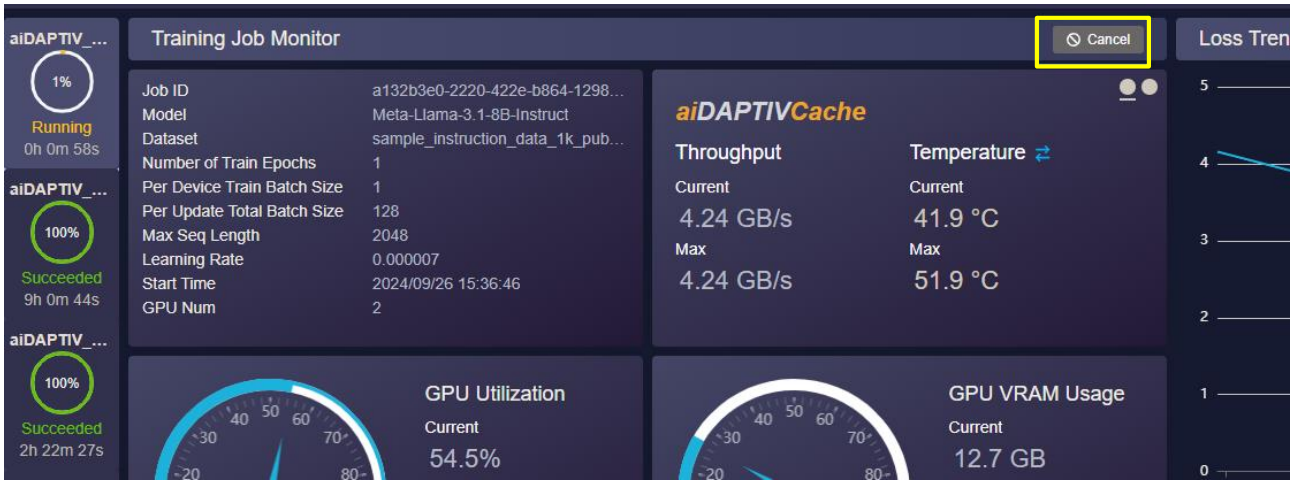


Figure 3-12 Cancel job

3.3.2. Remove job

Only jobs whose status is “Succeeded / Fail” can be removed.

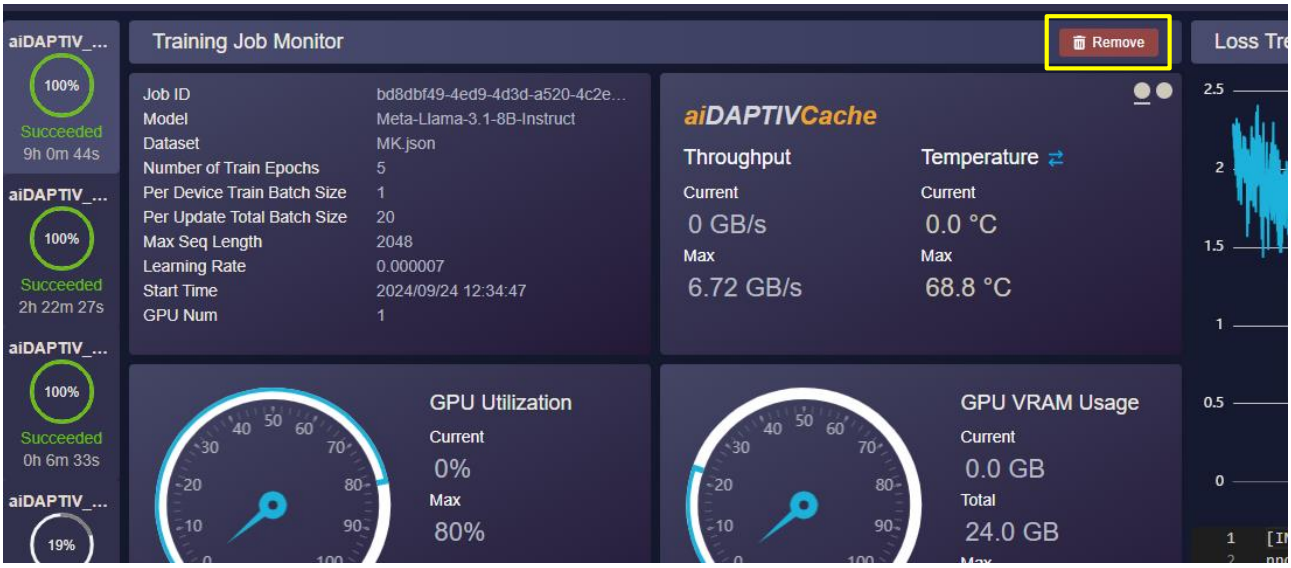


Figure 3-13 Remove job

3.4. Validation

Used to compare the results of the fine-tuned model against the original model or any other models. Users can ask questions to confirm whether the fine-tuning results meet expectations.

Note: The user can validate up to **4** models simultaneously.

3.4.1. Put questions

- Field description:
  1. **Model** : the model to be verified.
  2. **System Prompt** : A predefined instruction or message given to a software system to guide its behavior or output. It typically helps set the context, tone, or specific parameters for the interaction
  3. **Max tokens** : This parameter specifies the maximum number of tokens that the model can use to generate responses, where a token generally represents a part of a word or a whole word. (Range: 1000 ~ 12000)
  4. **Temperature** : This parameter controls the randomness of the generation process. A higher temperature results in more diverse responses, while a lower temperature makes responses more likely to follow common patterns seen in the training data. (Range: 0 ~ 1)
  5. **Top-p** : Similarly, a sampling technique with temperature called nucleus sampling, you can control how deterministic the model is at generating a response. If you are looking for exact and factual answers keep this low. (Range: 0 ~ 1)
  6. **Include chat history** : Determines whether to include previous dialogue interactions in the context for generating current responses.
  7. **Input Question area**: Click submit after entering questions.
  8. **RAG** : Please refer to section 3.6.2

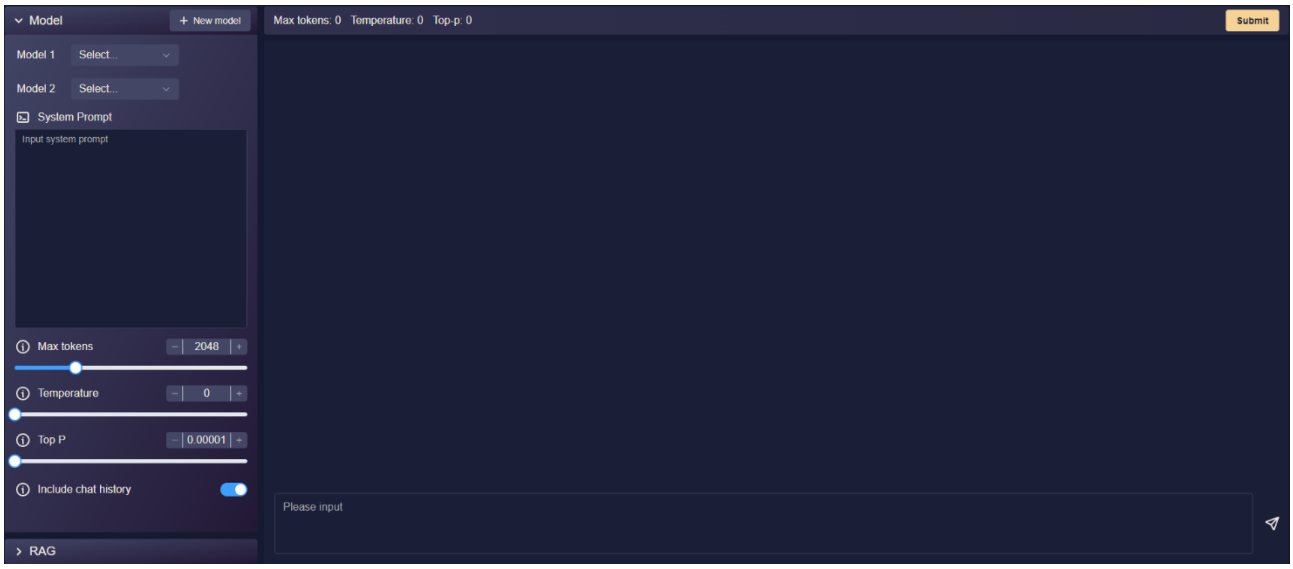


Figure 3-14 Setting of question

- Function description:
  1. **Submit** : Click submit after entering questions.
  2. **Cancel** : Can be used to cancel a half-executed task.

Figure 3-15 View result of question

3.4.2. Compare result

- Function description:
  1. **Reset**: Reset models, questions and parameters

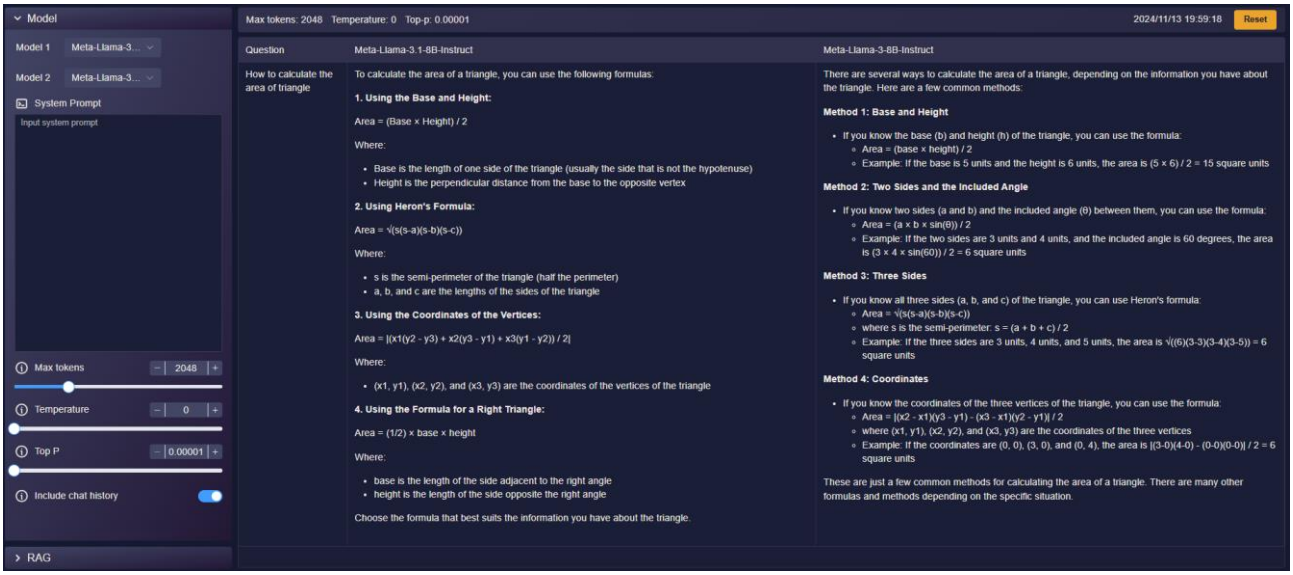


Figure 3-16 Compare results from different models

3.5. Benchmark (Option)

Score model performance.

Note : This is an additional value-added service. For more information on enabling this service, please contact Phison's Sales account.

3.5.1. Score

Set the parameters of the model.

- Field description:
  1. **Model** : the model to be scored
  2. **Benchmark model** : model as a reference
  3. **Embedding model** : retrieval model
  4. **Dataset**: Dataset used to test the model's answering ability
  5. **Temperature** : This parameter controls the randomness of the generation process. A higher temperature results in more diverse responses, while a lower temperature makes responses more likely to follow common patterns seen in the training data.
  6. **Max token** : This parameter specifies the maximum number of tokens that the model can use to generate responses, where a token generally represents a part of a word or a whole word. (Range: 1 ~ 12000)
  7. **Top-p** : Similarly, a sampling technique with temperature called nucleus sampling, you can control how deterministic the model is at generating a response. If you are looking for exact and factual answers keep this low. (Range: 0 ~ 1)
  8. **Recall Size** : Refers to the number of documents retrieved from a database before generating a response. (Range: 1 ~ 40)
- Function description:
  1. **Add** : Add a new model to be scored
  2. **Start benchmarking**

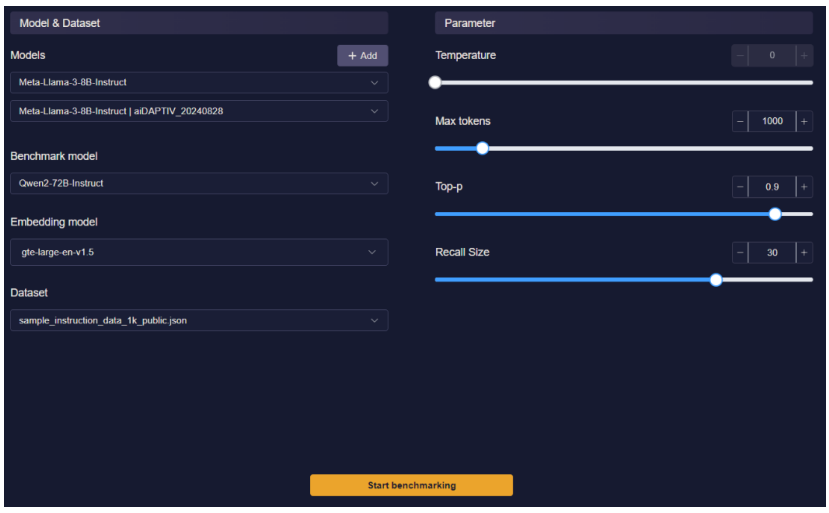


Figure 3-17 Parameter setting

3.5.2. Scoring Progress

- Field description:
  1. **Benchmark model**
  2. **Embedding model**
  3. **Dataset**
  4. **Temperature**
  5. **Top-p**
  6. **Benchmark Grid** :
     1. **Index** : Serial number
     2. **Model** : the model to be scored
     3. **Status**: Scoring status. Pending, Running, Finish, Fail
     4. **Progress** : Scoring progress
- Function description:
  1. Cancel all unfinished tasks: Cancel unfinished scoring tasks
  2. Return to the settings page
  3. View Result: View the graphical results of the rating

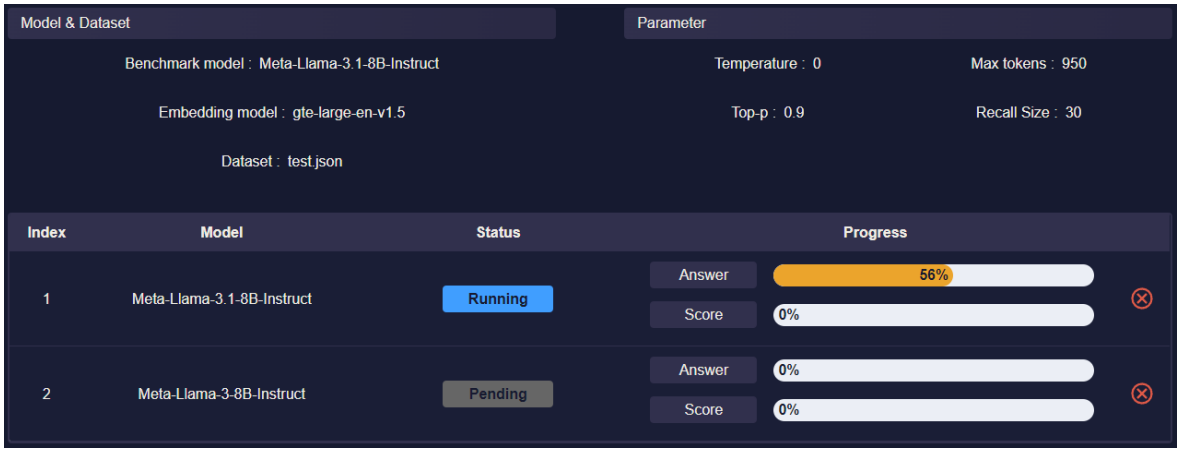


Figure 3-18 Scoring progress

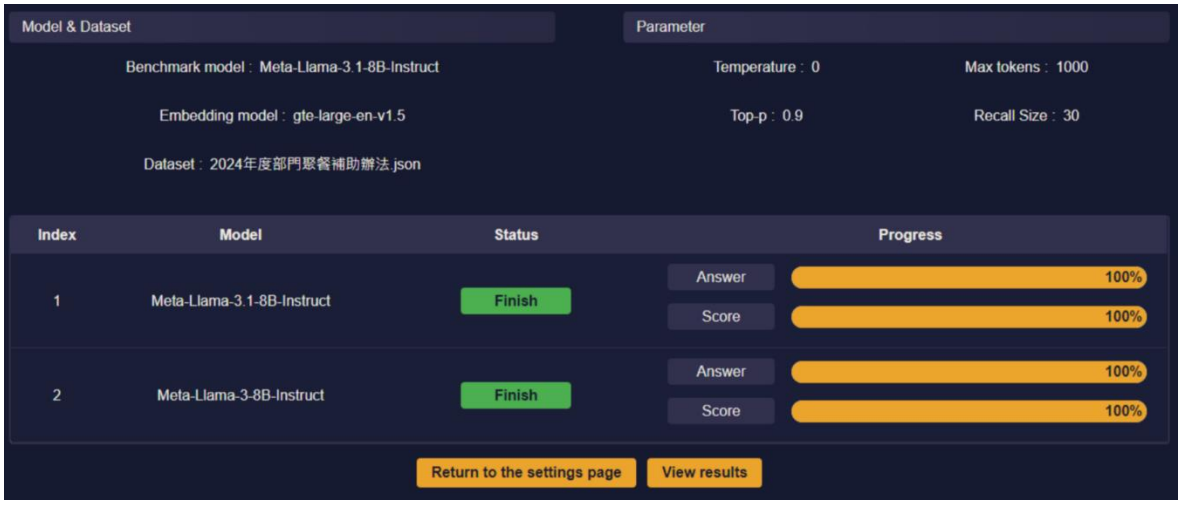


Figure 3-19 Scoring completed

3.5.3. Data

View all records containing past rating data.

- Field description:
  1. **Filter**
     - **Model** : the model to be scored
     - **Benchmark model** : model as a reference
     - **Embedding model** : retrieval model
     - **Dataset**: Dataset to test the model's answering ability
     - **Temperature** : This parameter controls the randomness of the generation process. A higher temperature results in more diverse responses, while a lower temperature makes responses more likely to follow common patterns seen in the training data.
     - **Max token** : This parameter specifies the maximum number of tokens that the model can use to generate responses, where a token generally represents a part of a word or a whole word. (Range: 1 ~ 12000)
     - **Top-p** : Similarly, a sampling technique with temperature called nucleus sampling, you can control how deterministic the model is at generating a response. If you are looking for exact and factual answers keep this low. (Range: 0 ~ 1)
     - **Status** : Scoring status
     - **Execution time** : The date and time the scoring was performed.

  2. **Benchmark Grid**
     - **Model** : the model to be scored
     - **Benchmark model** : model as a reference
     - **Embedding model** : retrieval model
     - **Dataset**: Dataset to test the model's answering ability
     - **Parameter** : Parameter settings when scoring (Temperature, Max tokens, Top-p)
     - **Status** : Scoring status
     - **Execution time** : The date and time the scoring was performed.

- Function description:

1. : Clear filter condition
2. : Select the records you want to view. Multiple items can be selected
3. : Delete scoring record
4. Click to render chart: Turn the scoring data in to a chart

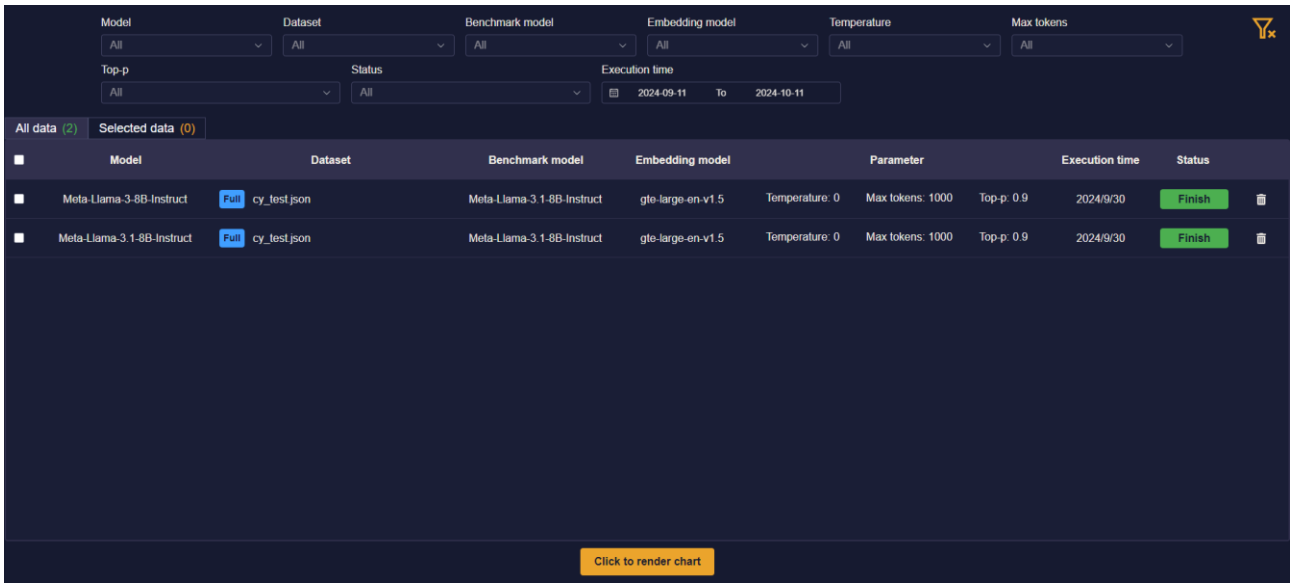


Figure 3-20 Scoring data

3.5.4. Chart

Turn the scoring data into a chart.

- Field description:
  1. **QA Pairs** : Number of scoring questions.
  2. **Max tokens**: Max tokens of the model being scored when scoring
  3. **Temperature**: Temperature of the model being scored when scoring
  4. **Top-p**: Top-p of the rated model when scoring
  5. **Y axis**: number of questions
  6. **X axis** : score

- Function description:
  Bar chart: Click on the bar chart to view the rating content in detail.

Figure 3-21 Bar chart

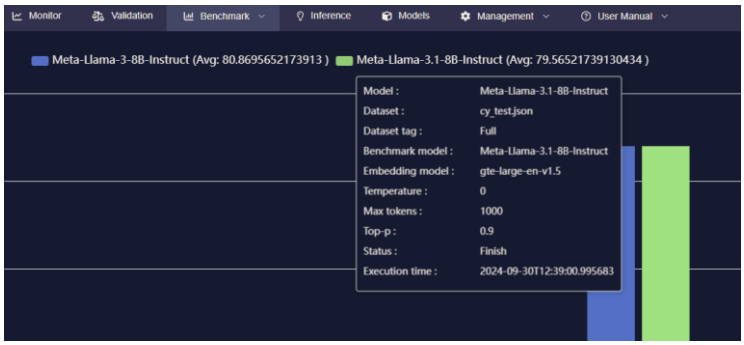


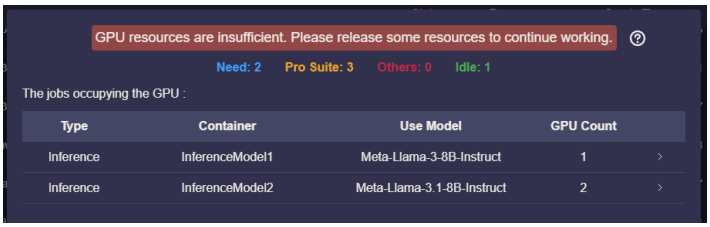
Figure 3-22 Model and parameter information

Figure 3-23 Detail of scoring content

3.6. Inference

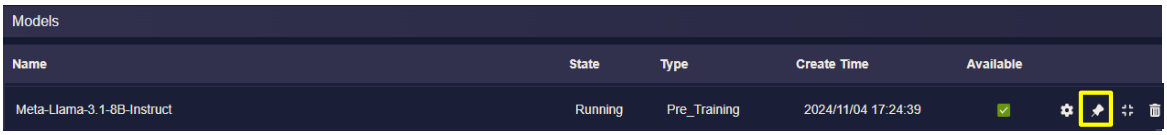
If the fine-tuned model verification results are satisfactory, you can create a chat room through the Inference function to provide a complete question and answer service.

- Up to 20 chat rooms can be created.
- Number of GPUs = 2^n (n=0,1,2,3,4, GPUs = 1,2,4,8)
- When selecting the number of GPUs, make sure there are enough GPU resources to perform the inference.



3.6.1. Chat

- Field description:
  1. **Model :** Select the model to inference.



Note : The model needs to be pinned first in order to appear in the model list.

  2. **System Prompt :** A predefined instruction or message given to an AI or software system to guide its behavior or output. It typically helps set the context, tone, or specific parameters for the interaction.
  3. **Max tokens** : This parameter specifies the maximum number of tokens that the model can use to generate responses, where a token generally represents a part of a word or a whole word. (Range: 1000 ~ 12000)
  4. **Temperature** : This parameter controls the randomness of the generation process. A higher temperature results in more diverse responses, while a lower temperature makes responses more likely to follow common patterns seen in the training data. (Range: 0 ~ 1)
  5. **Top-p** : Similarly, a sampling technique with temperature called nucleus sampling, you can control how deterministic the model is at generating a response. If you are looking for exact and factual answers keep this low. (Range: 0 ~ 1)
  6. **Include chat history** : Determines whether to include previous dialogue interactions in the context for generating current responses.
  7. **Input Question area**

- Function description:
  1. New Chat
  2.  : Edit chat room name.
  3.  : Delete chat room.

Figure 3-24 Chat room

3.6.2. RAG

Based on the chat room function, files can be uploaded for RAG (Retrieval-augmented generation). RAG search can be used to improve the accuracy of model answers or conversations.

Table 3-2 Specification of RAG

Item	RAG
File Format	pdf、log、json、docx、txt
Upload multiple files of same/different formats at once	Y

- Field description:
  1. **Enable RAG**: Whether to enable RAG function
  2. **Recall Size** : Refers to the number of documents retrieved from a database before generating a response. (Range: 1 ~ 40 counts)
  3. **Collection list** : The collection uploaded by the user. Only one collection can be selected.

- Function description:
  1. Upload new collection: Upload/create new Collection

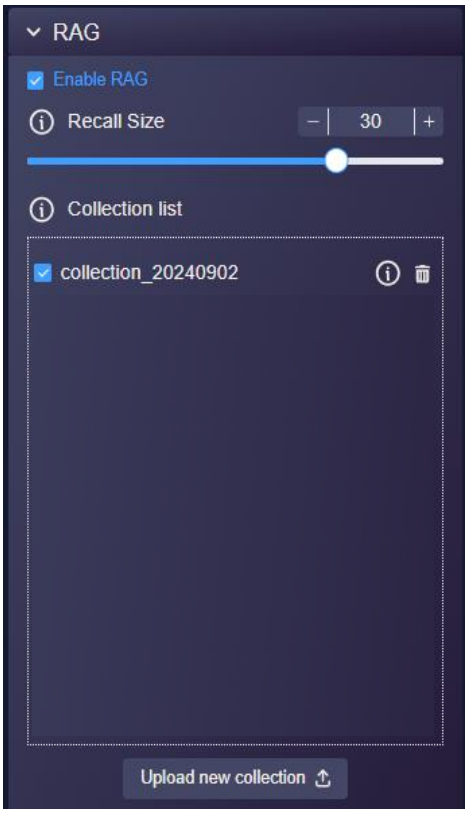


Figure 3-25 RAG

3.6.2.1. Upload new collection

Upload files to create a Collection

- Field description:
  1. **Collection Name**: Collection name
  2. **Chunk Size** : The amount of data contained in each chunk when processing retrieved documents. (Range: 256 ~ 2048 tokens)
  3. **Chunk Overlap** : The amount of data that overlaps between consecutive chunks when processing. (Range: 0 ~ Chunk Size * 0.5 tokens)
  4. **Upload File Area**: File upload area for output Collection.

- Function description:
  1. : Delete uploaded file
  2. : Delete all uploaded files
  3. : Upload/create collection

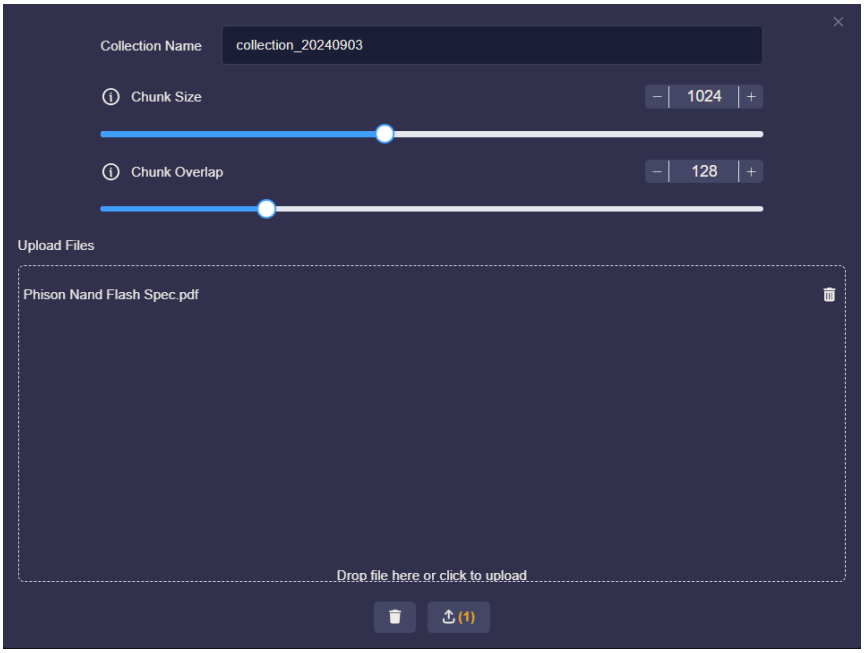


Figure 3-26 Collection management

3.6.2.2. Recommended usage – using with aiDAPTIVGuru

When you generate a dataset using aiDAPTIVGuru, a corresponding collection file is also created.

After training a model using an aiDAPTIVGuru generated dataset, it is recommended that you use this collection file and enable the RAG (Retrieval-Augmented Generation) feature when performing inference with the model to achieve the best results.

3.7. Models

Management of all models.

3.7.1. Model upload

- Function description:

1. Method 1: Drag the Model folder directly to the model storage location **/usr/local/models/**

   Command : sudo cp -r {source Model folder} {destination folder} (ex: sudo cp -r Meta-Llama-3.1-8B-Instruct/ /usr/local/models/)

2. Method 2: Compress the files in the model folder using either **zip** or **tar**, then click or drag them into the window to upload.

   Note: The fine-tuned model will automatically appear in the model list and will also be stored in the following path: /opt/phisonai/data/users/{user account}/jobs/{finetune job id}

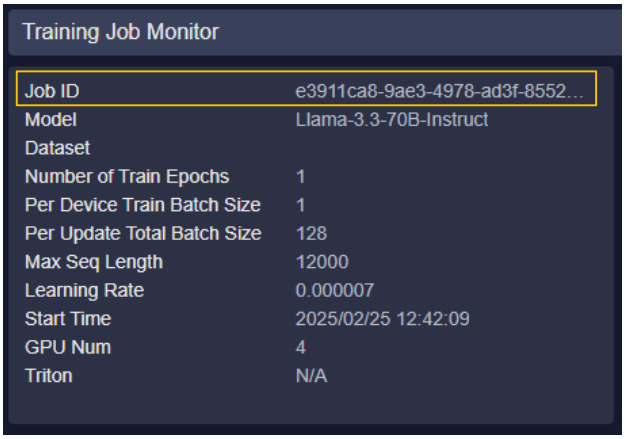


Figure 3-27 The storage path of trained models

3.7.2. Model list

- Field description:
  1. **Name**: Model name
  2. **State**: Model state. If Running is displayed, it means that the model is being Inferenced.
  3. **Model type**: If the name ends with AWQ, it indicates a quantized model.
     a. Pre_Training
     b. Finetune
     c. Pre_Training_AWQ
     d. Finetune_AWQ
  4. **Create Time**: Model upload/output time

- Function description:
  1. Available: The model must be checked and activated by the user before it can be seen in the model menus in Pro Suite.
  2.  : Set model Inference parameters
  3.  : Pin Select Model Button
  4.  : Button to quantize model
  5.  : Button to delete model (Model folder will also be deleted)

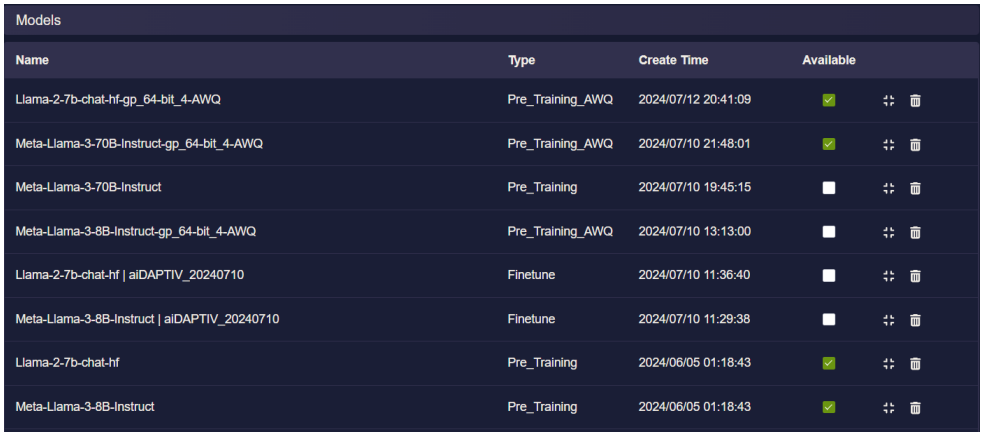


Figure 3-28 Model list description

Note : If a model already been pinned, then it cannot be deleted.

3.7.2.1. Enable model

- It will be automatically enabled(checked) after uploading through **Method 2**.
- If you have uploaded the model using **Method 1**, **after fine-tune** or **manual quantification**, you need to manually check the box to enable it.
- Only enabled models will be displayed in the model menus for Fine-tune, Validation and Inference.

3.7.2.2. Set model Inference parameters

- **GPU**: Number of GPUs used for Inference. Must be in power of 2.
- **Max token length**: Display the maximum token length according to the configuration of different models. If it is a combination from the table below, the system will automatically set the Max Token Length value. For other combinations, the user will need to set it manually.
  If you touch a value by mistake, you can restore it to the initial preset value by pressing the "Reset to Default" button.

Table 3-3 Recommended inference parameter settings

	Total Remain VRAM Size	GPU utilization	Max Token Length
Llama-3.1-8B-Instruct	48	0.95	131072
	20	0.95	14000
Llama-3.1-8B-Instruct-AWQ-INT4	48	0.95	131072
	16	0.95	67000
Llama-3.1-70B-Instruct	192	0.95	131072
Llama-3.1-70B-Instruct-AWQ-INT4	96	0.95	131072
	48	0.95	10000

- **GPU memory utilization**: The utilization rate of a single GPU. Default value is 0.9 .

Figure 3-29 Inference parameters setting

3.7.2.3. Pin the resident inference model

- Only enabled models can be pinned
- After pinning a model the “Running” prompt will be displayed indicating that the pinning process was successful and the model has been loaded into memory.
- Only the pinned model will be displayed in Inference's model menu.
- Model pinning will fail if the GPU resources are insufficient, an error message will appear. You may select “Yes” to view the error log.
- If a “network error” occurs after pinning the model, please refresh the page.

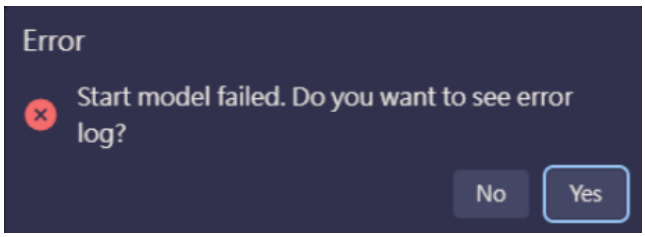


Figure 3-30 Pin model failed

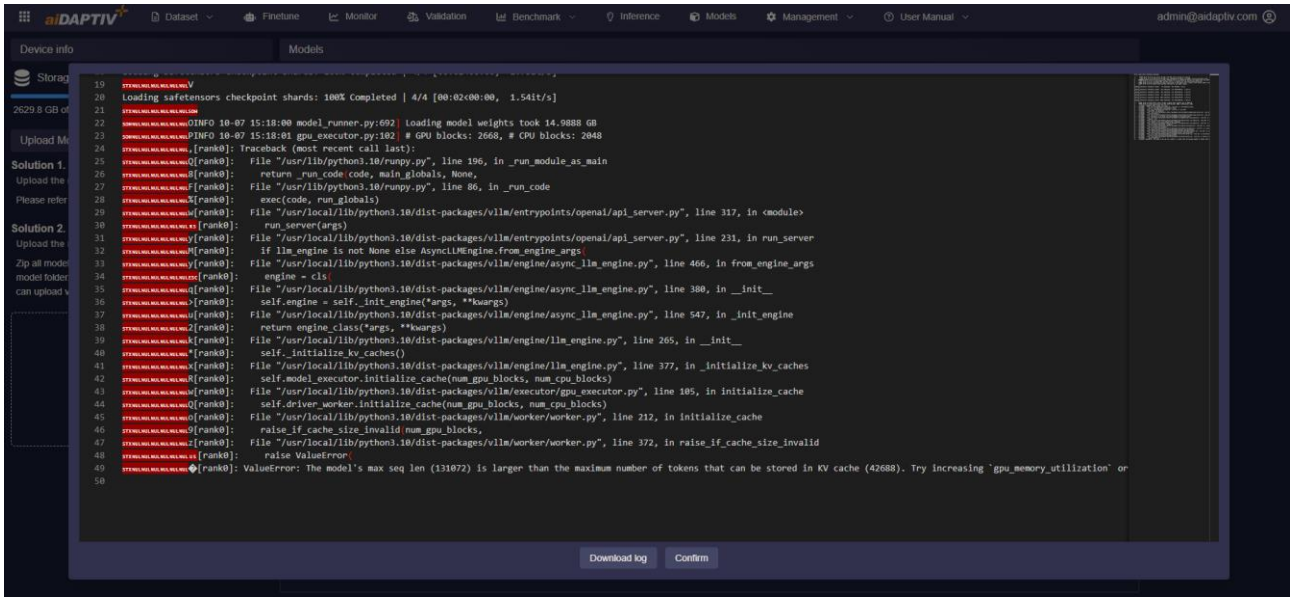


Figure 3-31 Pin model failed error log

3.7.2.4. Quantized model

Quantize the model by converting the model weights into fixed points or integers to reduce the model size, the computing cost, and accelerate the inference of the model. After quantization, the model will be displayed in the model list with a type ending in "_AWQ".

- Field description:
  1. **Available GPU**: Sets the GPUs number to be used for quantization. Must be in powers of 2.
  2. **Group Size**: Model parameter group size. Larger values will reduce the accuracy of the model, but can improve the quantization efficiency and reduce the model size. Must be in powers of 2.
  3. **Bit**: Sets the bit-width for the quantized model parameters. Lower values reduce model size and increase calculation speed but may affect model accuracy. Must be in powers of 2.

Figure 3-32 Setting of model quantization

- Function description:
  1. Cancel: Cancel quantization

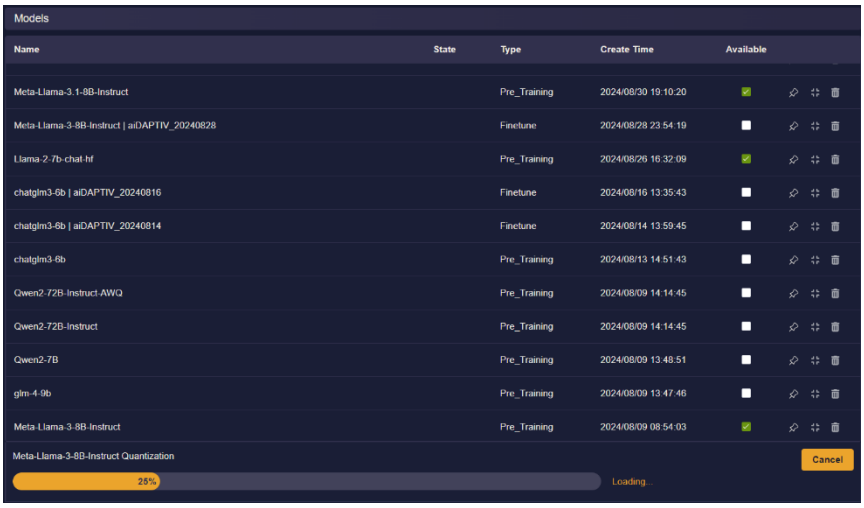


Figure 3-33 Cancel model quantization

3.8. Management

Note: Only admin accounts will be allowed to use the following features.

3.8.1. Authorization

User account role management.

Default system administrator account password
Account: admin@aidaptiv.com
Password: Admin8299

3.8.1.1. Features

Function settings. Set the permission for Read and Write of each function.

- Field description:
  1. **Features**: Pro Suite feature list, click to set the function permissions of each Role.
  2. **Role Grid** :
     - Role Name
     - Read : Only has permission to read this function.
     - Write : Have permission to write and edit this function.
- Function description:
  1. Search : Search Role
  2. Add: Add role to a specific feature to set permissions

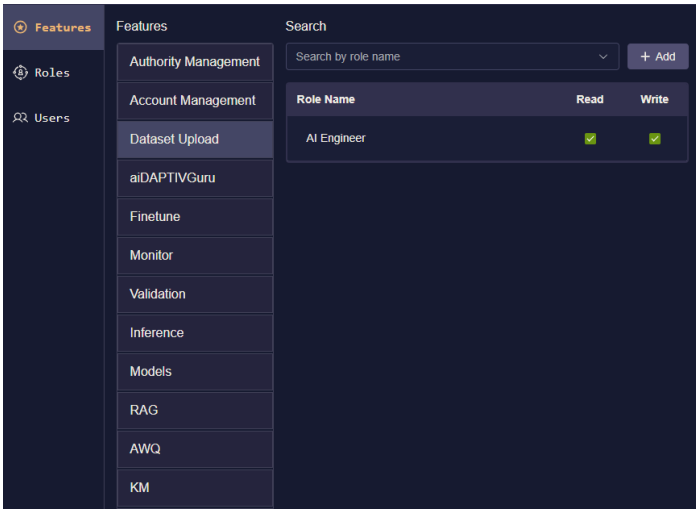


Figure 3-34 Feature setting of role

- Example:

Table 3-4 Recommend setting of authorization

Features	Admin		AI Engineer		General User	
	Read	Write	Read	Write	Read	Write
Authority Management	-	Y	-	N	-	N
Account Management	-	Y	-	N	-	N
Dataset Upload	Y	Y	Y	Y	N	N
Guru	-	Y	-	Y	-	N
Finetune	-	Y	-	Y	-	N
Monitor	-	Y	-	Y	-	N
Validation	-	Y	-	Y	-	N
Inference	-	Y	-	Y	-	Y
Models	Y	Y	Y	Y	Y	N
RAG	-	Y	-	Y	-	Y
AWQ	-	Y	-	Y	-	N
KM		Y		Y		Y

Note: The settings for these three roles will be preset in the system.

3.8.1.2. Roles

Character setting. Create a character and set character features.

- Field description:

1. **Role Grid** :
   - Role Name
   - Enable

- Function description:

1. Search : Search Role
2. Role Grid :
   - : Check the users under a specific Role.
   - : Rename Role name
   - : Delete Role

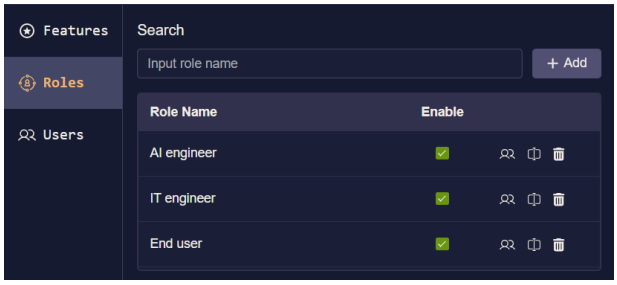
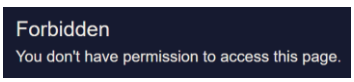


Figure 3-35 Role management

- A role cannot be deleted if there are accounts that are using it.
- When a role is disabled, the accounts associated with it will not be able to log into Pro Suite.



- When the role name is changed, the associated accounts will also be updated accordingly.

3.8.1.3. Users

User account settings. Create a user account and set the corresponding role.

- Field description:
  1. **User Grid** :
     - **Name**: user name
     - **Email**: User Email. Sign in as a user.
     - **Role**: User role. Settings can be switched directly.
     - **Enable**: enabled state.
     - **Last Login**: Last login time.
     - **Disable Time**: Disable time.
     - **Action**: Reset user password
- Function description:
  - Create account

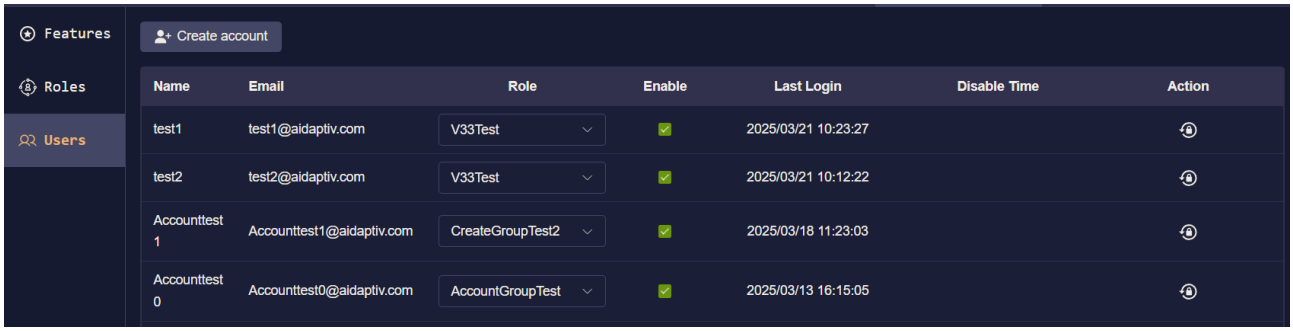


Figure 3-36 User management

3.8.1.3.1. Create Account

- Field description:
  1. **Name**: User's name. Only English and underscores are allowed. Maximum length is 20 characters.
  2. **Email**: User's log-in email account. Must be in a valid email format.
  3. **Password**: User's password. Maximum length is 20 characters.
  4. **Repeat Password**: Confirmation of the user's password. Maximum length is 20 characters.
  5. **Role**: Assign a predefined role to the user account.
- Function description:
  1. Cancel
  2. Submit

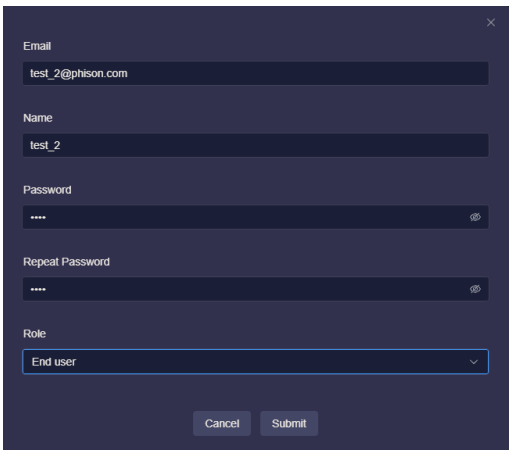


Figure 3-37 Create account

4. APPLICATION

4.1. aiDAPTIVInbox (Option)

This function is an additional value-added service. For more information on enabling this service, please contact Phison's Sales account.

For the introduction to aiDAPTIVInbox, please refer to the following document : *aiDAPTIVInbox User Manual_092024_v1.1 .pdf*

aiDAPTIVInbox is an AI Email Assistant created by Phison Electronics Corp. Powered by its AI technology invention solution called aiDAPTIV+, aiDAPTIVInbox is aimed at improving daily work processes, employee efficiency, and enhanced corporate productivity. aiDAPTIVInbox is designed to be deployed as an on-premise solution to ensure the confidentiality of corporate data by keeping all sensitive information securely stored within the organization's own infrastructure, reducing exposure to external threats and maintaining full control over access and data handling. Through aiDAPTIVInbox, employees can significantly reduce working hours, eliminate time-consuming and tedious tasks, and redirect their focus toward innovation and research & development, thereby creating greater opportunities for the enterprise

Note: For pre-installation confirmation and post-installation checks, please refer to Appendix C.
Inbox support server system: Microsoft Exchange Server 2019, Mail2000, Hgiga(1132)

- Field description:
  1. **Model**: model used by aiDAPTIVInbox inference
  2. **System Prompt (Constraint)**: Define the name and role of AI, function description, etc.
  3. **Service Status**: Inbox service status
  4. **Language** : Select language. (zh-TW, zh-CN, en-US, ms-MY)
  5. **User Mail Account**: the mail account used by the mail assistant
  6. **User Mail Address**: The mail address used by the mail assistant
  7. **User Mail Password**: The mail password used by the mail assistant
  8. **Domain** : Mail domain. (Please fill in the email format. Should contain "@" and ".")
  9. **Answer Presfix** : Letter opening. (ex: This is the reply from the email assistant:)
  10. **Answer Suffix** : Ending of letter. (ex: Thank you)
  11. **Open for All Users**: No restrictions on sender domain, anyone can use the AI function to send and receive messages
  12. **Web Search** : Internet search function
  13. **White List** : Open to senders on this list and outside of the configured Domain.

- Function description:
  1. **Add** : Added a whitelist acceptable to Mail Assistant
  2. **Save and restart**: Save Mail Assistant settings and restart the service
  3. **Stop**: Stop service

4.1.1. EWS (Exchange Web Services)

1. User Mail Address: Please fill in the email format. (Should contain "@" and "." , ex: test @phison.com)
2. Mail Server: Only domain name can be filled in, not the IP. (String length: 2~63. Should contain "." , ex: mail.phison.com)
3. Office 365: Verification of Office 365 cloud authentication and authorization service usage.
   - Client ID
   - Client Secret
   - Tenant ID

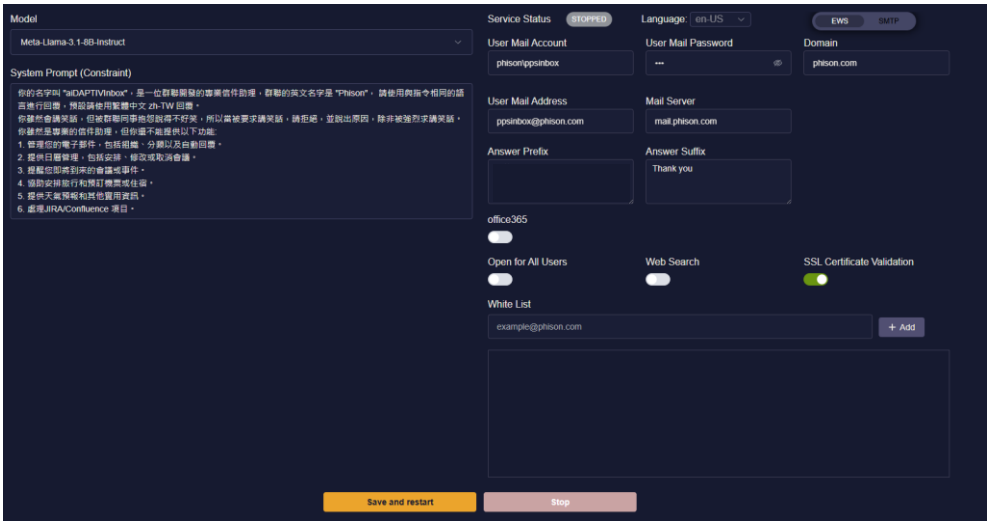


Figure 4-1 EWS Setting

4.1.2. SMTP (Simple Mail Transfer Protocol)

1. SMTP Server IP : Server domain for sending emails (Should contain "." , ex: mail.phison.com)
2. SMTP Port : Port for sending emails. (Support: 25, 465, 587)
3. IMAP Server IP: Server domain for receiving emails. (Should contain "." , ex: mail.phison.com)
4. IMAP Port : Port for receiving emails. (Support: 993)

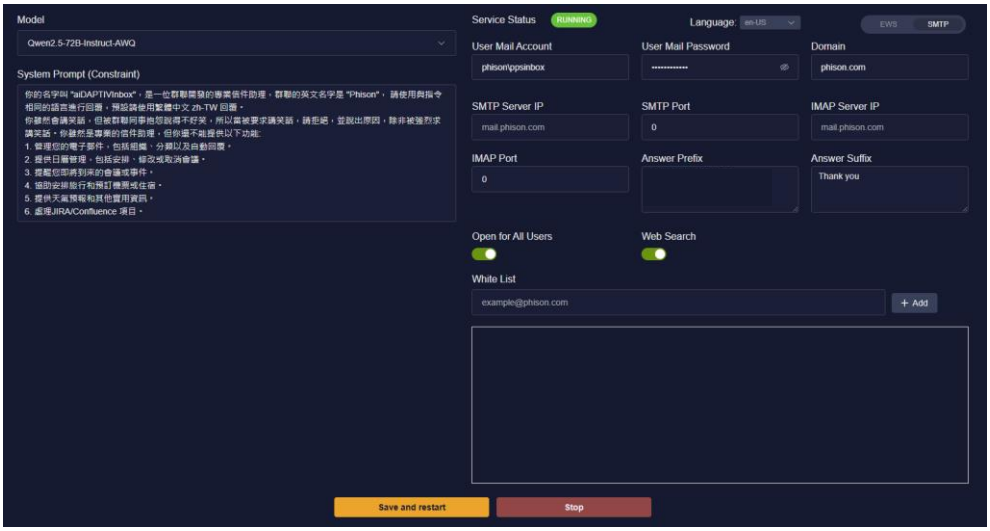


Figure 4-2 SMTP Setting

APPENDIX A – MODEL SUPPORT LIST

Table A-1 Support model list

No	Model Name	Model Size (MB/GB)	Fine-tune	Inference	Support Triton
1	Llama-3.1-8B-Instruct	29.9GB	Y	Y	Y
2	Llama-3.1-70B-Instruct	262.9GB	Y	Y	Y
3	Qwen2.5-72B-Instruct	135.4GB	Y	Y	Y
4	Llama-3.3-70B-Instruct	525.7GB	Y	Y	Y
5	QwQ-32B	61GB	Y	Y	Y
6	DeepSeek-R1-Distill-Qwen-32B	61GB	Y	Y	Y
7	DeepSeek-R1-Distill-Llama-70B	131.4GB	Y	Y	Y
8	Mistral-7B-Instruct-v0.1	27.5GB	Y	Y	Y
9	deepseek-moe-16b-chat	30.5GB	Y	Y	N
10	chatglm3-6b	46.5GB	Y	Y	N
11	glm-4-9b-chat	17.5GB	Y	Y	N

APPENDIX B – RECOMMENDED CONFIGURATION

- DRAM and aiDAPTIVCache with different LLM model size

Table B-1 Recommend Configuration

	AITPC	Work Station	Server
GPU Configuration	NVIDIA 4060Ti (16GB)*1	NVIDIA RTX 4000 Ada *4 NVIDIA RTX A6000 *4	NVIDIA RTX A6000 *8
LLM model size	**≤13B**	**<100B**	**<200B**
DRAM	DDR5 4800 64GB	DDR5 4800 512GB DDR5 4800 1024GB	DDR5 4800 1024GB
aiDAPTIVCache capacity	320GB	2TB	2TB
aiDAPTIVCache count	1	2	4

  - Recommend Gen4 or above.
  - Recommend DRAM 2933MHz or above.
  - Recommend DRAM channel number is 8 or more, ex: 16GB x8

APPENDIX C – INBOX MAIL SERVER TEST

The main purpose of this section is to help users perform basic environment checks before installing aiDAPTIVInbox, and to test whether the installation is correct after aiDAPTIVInbox has been installed.

- Test script : smtp_imap_connection_test.py

mail_account :	Account to log in to the mail server
mail_address :	Complete email address
mail_password :	Password to log in to the mail server
smtp_server_ip :	Server domain for sending emails
smtp_port :	Support: 25, 465, 587
imap_server_ip :	Server domain for receiving emails.
imap_port :	Support: 993
test_mail :	Email address for testing. After the script is tested, a test email will be sent to this email address.

- Test script parameter configuration file : smtp_imap_connection_test.json

C.1 Precautions before testing

1. Place ***Test script*** and ***Test script parameter configuration file*** in the same folder.
2. Enable :
    - SMTP : SMTP_Server_IP, SMTP port
    - IMAP : IMAP_Server_IP, IMAP port
3. Confirm that the SMTP and IMAP functions of the mail server are enabled, and the corresponding ports also need to be enabled (Not blocked by the firewall).
4. Confirm that the IMAP of the mail server can perform the following operations on the mailbox :
    - Can check mailbox
    - Have permission to download emails
    - Can change email status (eg: read, unread)
    - Have permission to move emails to different email folders

C.2 Execute test script

Enter the following command in the terminal to execute the test script.

```
python3 smtp_imap_connection_test.py
```

C.3 Test result

- If the test result is ***Pass*** : User will recive a e-mail in the test mail. (The subject of the email is: Test subject - "Time of program execution")

- If the test result is ***Fail*** : Users can refer to the errorcode below to troubleshoot the problem.

Table C-1 Error message definition

Error message	Definition
Account or password incorrect, check account and password	Mail Account
	Mail Password
SMTP error: Check the smtp_server_ip and smtp_port	SMTP Server IP
	SMTP Port
IMAP error: Check the imap_server_ip and imap_port	IMAP Server IP
	IMAP Port