GIGABYTE™

AI TOP Utility Lite

User Manual

1. Installation and Preparation..... 3

1-1 System Requirements..... 3

1-1-1 Issue: Linux Kernel not compared..... 4

1-1-2 Issue: Uninstall NVIDIA Driver..... 5

1-2 Hardware Condition..... 5

1-2-1. Motherboard..... 5

1-2-2. Graphics card..... 7

1-2-3. Power supply..... 7

1-3 Graphics Card Driver Installation..... 7

1-3-1. Nvidia graphic cards..... 7

1-3-2. AMD graphic cards..... 7

1-4 AI TOP Utility Software Installation..... 8

1-4-1 AI TOP Utility Installation on Linux..... 8

1-4-2 AI TOP Utility Installation on Windows (WSL)..... 9

1-4-3 Initial Setup and Environment Configuration..... 14

1-4-4 Uninstall AI TOP Utility..... 16

2. Feature Descriptions..... 17

2-1 Dashboard..... 17

● Dashboard - Dataset..... 18

● Dashboard - Fine-tune..... 19

2-2 Model Hub..... 20

2-2-1 Model List..... 20

2-2-2 Model Convert..... 20

● Model Hub - Status..... 20

● Model Type..... 21

● Modality..... 21

● Format..... 21

● Model Hub - Action..... 21

2-3 Datasets..... 22

● Dataset- Action..... 23

2-4 Fine-tune..... 23

● General Setting & Finetuning Method..... 24

● Fine-tune - Status..... 27

● Fine-tune - Action..... 28

2-4-1 Information..... 28

2-4-2 Benchmark..... 29

2-4-3 Validation..... 29

2-5 Inference....29
2-6 Settings....30
3. User Workflow Guide....31
3-1 System Dashboard Overview....31
● Activity....32
3-2. How to download models from the Model Hub?....33
3-2-1 Convert....34
3-3. How to Generate Structured Datasets from Raw Data?....36
3-4. Model Fine-tuning Workflow....40
3-4-1 Information....45
3-4-2 Benchmark....46
3-4-3 Validation....46
3-3-4 Action....49
3-5. Inference....50
3-6 Settings....53
3-6-1 Software Settings....53
3-6-2 Repair & Reinstall....53
3-7 Finetune Expand config syntax....55
4. Supported Models....56

1. Installation and Preparation

1-1 System Requirements

- **(Important) Secure Boot Configuration** :

  For systems using NVIDIA or AMD dedicated graphics cards, Secure Boot must be set to **[Disabled]** in the motherboard BIOS. Secure Boot prevents unsigned kernel modules from loading, which may cause graphics drivers to fail initialization and prevent proper graphics card detection.

- Linux installation:

  The validated and officially supported environment is: **Ubuntu 24.04 LTS with Linux Kernel 6.17.0-40-generic**. Other operating system or kernel combinations have not been fully validated and may not guarantee full functionality.

  Please make sure your system modules are up to date by:

  ```
  sudo apt update
  sudo apt upgrade
  ```

  You can check your linux kernel version by command:

  ```
  uname -r
  ```

  ```
  ❯ uname -r
  6.17.0-40-generic
  ~
  ```

  If your kernel version is not compared. Fix it by > **Section 1-1-1**

  For NVIDIA GPU Users. NVIDIA driver version must be 580.95.05 with CUDA Driver 13.0. You can check your driver version by command:

  ```
  nvidia-smi
  ```

```
❯ nvidia-smi
Wed Aug 19 14:19:10 2026
+-----------------------------------------------------------------------------------------+
| NVIDIA-SMI 580.95.05              Driver Version: 580.95.05      CUDA Version: 13.0     |
+-----------------------------------------+------------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id          Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |           Memory-Usage | GPU-Util  Compute M. |
|                                         |                        |                  MIG M. |
|=========================================+========================+======================|
|   0  NVIDIA GeForce RTX 5090        On  |   00000000:41:00.0  On |                  N/A |
|  0%   45C    P3             34W /  600W |     759MiB /  32607MiB |      0%      Default |
|                                         |                        |                  N/A |
+-----------------------------------------+------------------------+----------------------+
```

If your system already has not compared drivers. Please Uninstall it by > **Section 1-1-2**

- Windows WSL installation:

  Please directly refer to **Section 1-4-2**.

1-1-1 Issue: Linux Kernel not compared

1. Install Kernel Package:

   Modules:

   ```
   sudo apt-get install linux-image-6.17.0-40-generic
   sudo apt-get install linux-headers-6.17.0-40-generic
   ```

   Driver :

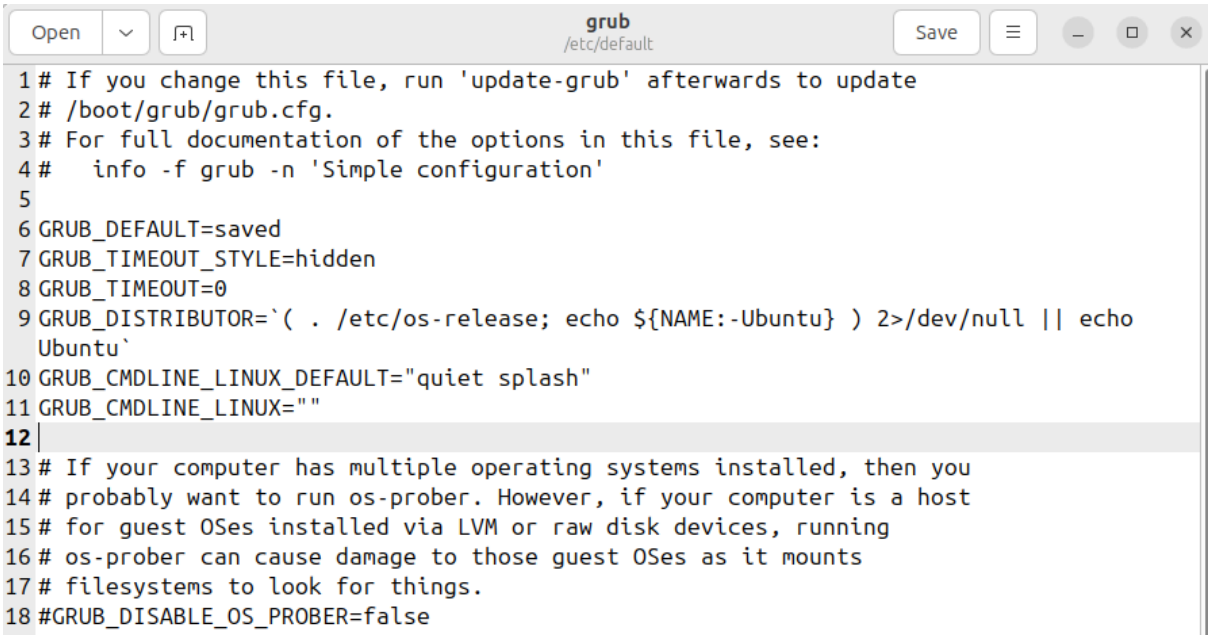
   ```
   sudo apt-get install linux-modules-extra-6.17.0-40-generic
   ```

2. Update the initramfs file for kernel version

   ```
   sudo update-initramfs -u -k 6.17.0-40-generic
   ```

3. Change the grub boot choose:

   ```
   sudo apt install gedit -y
   sudo gedit /etc/default/grub
   ```

   Set GRUB_DEFAULT=saved to make GRUB remember and boot the last selected entry by default.

4. Update GRUB:

   ```
   sudo update-grub
   ```

5. Change default kernel:

```
sudo grub-set-default 'Advanced options for Ubuntu>Ubuntu, with Linux 6.17.0-40-generic'
```

6. Reboot PC:

```
sudo reboot
```

Reboot finish. You can use `uname -r` to check whether the kernel is compared or not.

```
trx50@trx50-TRX50-AI-TOP:~$ uname -r
6.17.0-40-generic
```

7. Remove old kernels:

Make sure your kernel is 6.17.0-40-generic. Remove your old kernel package by:

(X.XX.X-XX is your old kernel version)

```
sudo apt purge linux-image-X.XX.X-XX-generic
sudo apt purge linux-headers-X.XX.X-XX-generic
sudo apt autoremove
```

Then reboot PC:

```
sudo reboot
```

For NVIDIA users. If you see your `nvidia-smi` become failed. Please follow **Section 1-1-2** to reinstall.

```
trx50@trx50-TRX50-AI-TOP:~$ nvidia-smi
NVIDIA-SMI has failed because it couldn't communicate with the NVIDIA driver. Ma
ke sure that the latest NVIDIA driver is installed and running.
```

1-1-2 Issue: Uninstall NVIDIA Driver

Use command to uninstall an old NVIDIA Driver:

```
sudo apt --purge remove "*cuda*" "*cublas*" "*cufft*" "*cufile*" "*curand*" "*cusolver*" "*cusparse*" "*gds-tools*" "*npp*" "*nvjpeg*" "nsight*" "*nvvm*" "*nvidia*"
```

Then, Follow **Section 1-3-1** to install a verified NVIDIA driver version.

1-2 Hardware Condition

AI TOP Utility only supports Gigabyte hardware.

1-2-1. Motherboard

- Other old GIGABYTE motherboards with the newest BIOS, such as
  - Intel :
    - Z890 Aorus Pro ICE

        - H810M S2H, Z790 Aorus Master X
        - B760 Gaming X AX
        - B660 Aorus Master
        - H610M D3W WIFI6
    - AMD :
        - A620 A620M GAMING X 1.0
        - B650 B650 AORUS ELITE AX 1.2
        - X670 X670E AORUS MASTER 1.03
        - B840 B840M DS3H 1.0
        - B850 B850M EAGLE WIFI6E 1.1
        - X870E X870E AORUS XTREME X3D AI TOP 0.5
        - X870 X870 AORUS ELITE WIFI7 1.11
        - A620A A620I AX 2.01
- TRX50 AI TOP

  Product Page: https://www.gigabyte.com/za/Motherboard/TRX50-AI-TOP
- TRX50 AERO D

  Product Page: https://www.gigabyte.com/za/Motherboard/TRX50-AERO-D-rev-12
- W790 AI TOP

  Product Page: https://www.gigabyte.com/za/Motherboard/W790-AI-TOP
- Z890 AORUS XTREME AI TOP

  Product Page:

  https://www.gigabyte.com/za/Motherboard/Z890-AORUS-XTREME-AI-TOP
- Z890 AORUS MASTER AI TOP

  Product Page:

  https://www.gigabyte.com/za/Motherboard/Z890-AORUS-MASTER-AI-TOP
- Z890 AI TOP

  Product Page: https://www.gigabyte.com/za/Motherboard/Z890-AI-TOP
- X870E AORUS XTREME AI TOP

  Product Page:

  https://www.gigabyte.com/za/Motherboard/X870E-AORUS-XTREME-AI-TOP-rev-1x
- B850 AI TOP

  Product Page: https://www.gigabyte.com/za/Motherboard/B850-AI-TOP

***Note: please first update bios for motherboard**

Step 1: Download the newest BIOSversion of your AI TOP Motherboards via product page

Example: https://www.gigabyte.com/Motherboard/Z890-AORUS-XTREME-AI-TOP/support

Step 2: unzip the corresponding bios file of your motherboard

Step 3: copy bios file to a usb

Step 4: plug that usb into your motherboard

Step 5: restart PC and press Delete button to access BIOS settings

Step 6: in BIOS settings screen, press F8 and select corresponding bios file, press Enter

Step 7: wait until bios updating is completed.

Step 8: restart PC and start installing AI TOP Utility software

***Note: If using an AMD motherboard, disable the integrated graphics in the BIOS.**

1-2-2. Graphics card

Reference link: https://www.gigabyte.com/Graphics-Card

- Gigabyte GeForce RTX 50 series: RTX 5090, RTX 5090 D, RTX 5080 SUPER, RTX 5080, RTX 5070 Ti SUPER, RTX 5070 Ti, RTX 5070 SUPER, RTX 5070, RTX5060 Ti.
- Gigabyte GeForce RTX 40 series: RTX 4090, RTX 4090 D, RTX 4080 SUPER, RTX 4080, RTX 4070 Ti SUPER, RTX 4070 Ti, RTX 4070 SUPER, RTX 4070, RTX 4060 Ti, RTX 4060.
- GIGABYTE Radeon™ AI PRO R9700 AI TOP 32G; Gigabyte Radeon Pro W7000 and Radeon RX 7000 series: Radeon Pro W7900, Radeon Pro W7800; Radeon RX 7900 XTX, Radeon RX 7900 XT, Radeon RX 7900 GRE.

1-2-3. Power supply

Power monitoring is only supported on AORUS P1600W 80+ Titanium Modular PCIe 5.1 AI TOP. Other PSU models are not supported.

Product link: GP-AP1600TM PG5 AI TOP

1-3 Graphics Card Driver Installation

1-3-1. Nvidia graphic cards

- For **Ubuntu 24.04.4** :

  NVIDIA driver version 580.95.05 installation + CUDA 13.0 installation.

  For quick installation, please run the following command in the terminal:

  ```
  bash nvidia_driver_cuda_2404.sh
  ```

1-3-2. AMD graphic cards

- For **Ubuntu 24.04.4** :

  AMD driver version 7.1.70100-1 installation

  For quick installation, please run the following command in the terminal:

  ```
  bash amd_driver_rocm_2404.sh
  ```

***Note: Please reboot your PC after the driver installation above is finished.**

1-4 AI TOP Utility Software Installation

This section describes how to install AI TOP Utility on Linux and Windows (WSL) environments.

For **Linux** systems, please follow **Section 1-4-1**.

For **Windows** systems, first complete WSL installation and configuration by following **Section 1-4-2**, then install AI TOP Utility.

After the installation is complete, proceed to **Section 1-4-3** to complete the initial setup and environment configuration.

To download the installation package, please visit the following link.

1-4-1 AI TOP Utility Installation on Linux

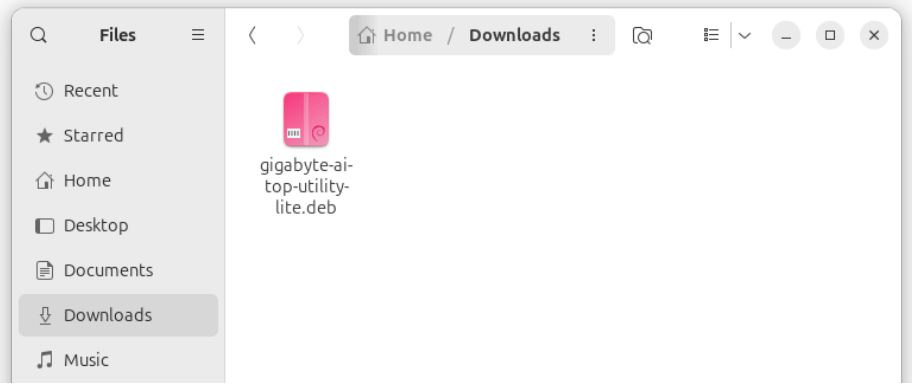
Before proceeding with the installation, verify that the graphics driver installation has been completed successfully:

- NVIDIA users: Refer to **Section 1-3-1 Nvidia Graphic Cards**
- AMD users: Refer to **Section 1-3-2 AMD Graphic Cards**

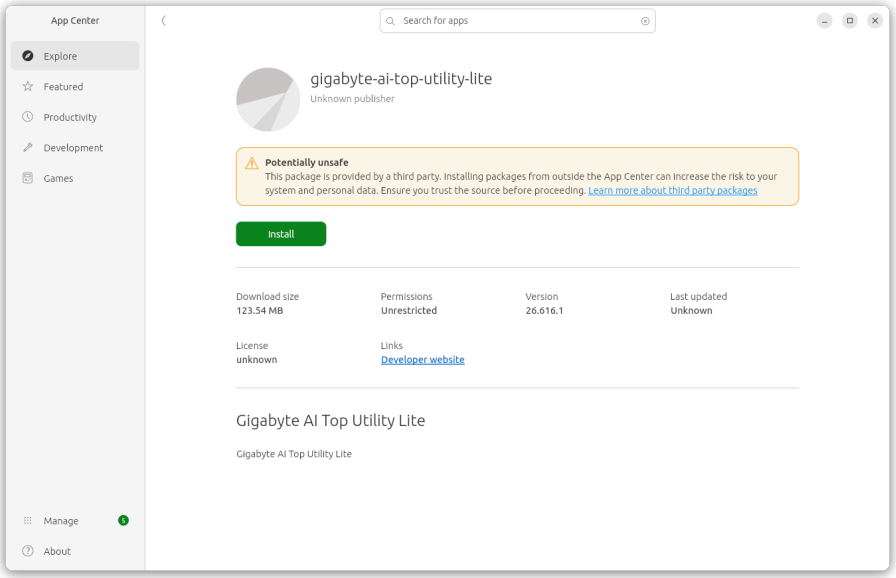
AI TOP Utility requires a properly installed GPU driver and runtime environment (CUDA or ROCm) to enable GPU acceleration.

After the installation package download is complete, locate the **.zip** file in the folder.

Extract the **.zip** file to access the installation package.



Then, right-click the .deb file and select **"Open with App Center"**. In the App Center window, click **Install** to begin the installation.



1-4-2 AI TOP Utility Installation on Windows (WSL)

Windows users must install Ubuntu through Windows Subsystem for Linux (WSL) before installing AI TOP Utility. This section describes how to set up WSL and Ubuntu on a Windows system.

Requirements

Before installation, please ensure the following:

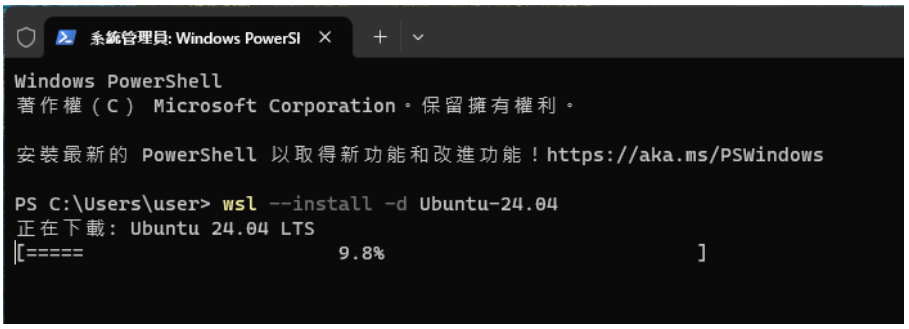
- Uninstall any existing Ubuntu installation from **Applications** (if applicable).
- It is recommended to use **Windows 11 with WSL2** for optimal compatibility.

- AMD GPUs are currently not supported in WSL.

(1) Install WSL and Ubuntu

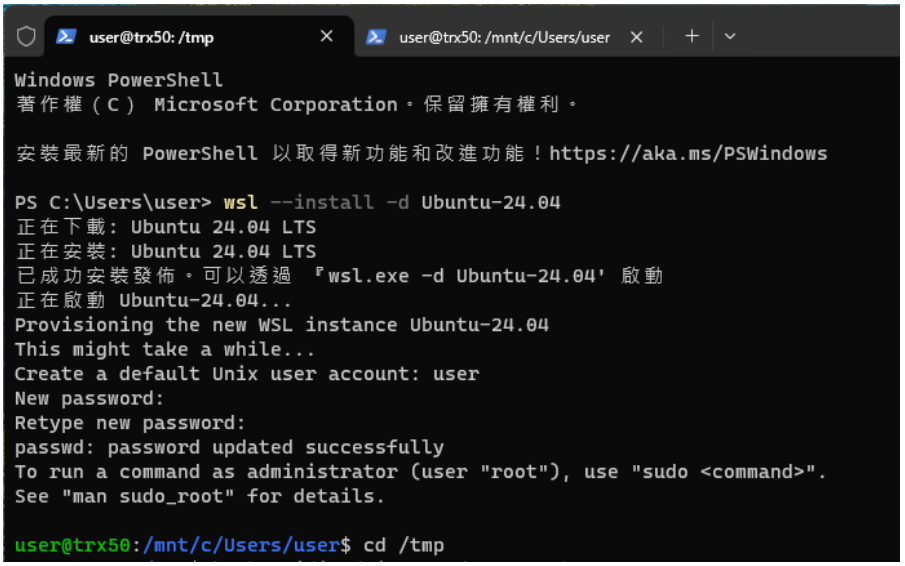
On the Desktop, right-click on the “Start” menu > select “Windows Powershell (System Administrator)”, then execute:

```
wsl --install -d Ubuntu-24.04
```



When Ubuntu launches for the first time:

1. Enter a username when prompted with **"Enter new UNIX username:"**.
2. Enter and confirm a **password** when prompted.
3. After the account setup is completed, Ubuntu will finish the WSL initialization process and open the terminal.



***Alternative Ways to Launch Ubuntu (Optional)**

After completing the WSL installation, Ubuntu can be launched using one of the following methods:

Method 1: Start Menu / Search

In the Windows Start menu or Search bar, type "Ubuntu 24.04", then click the corresponding result to launch the application.

Method 2: Using WSL Command

You can also launch Ubuntu directly from the terminal by running:

```
wsl -d Ubuntu-24.04
```

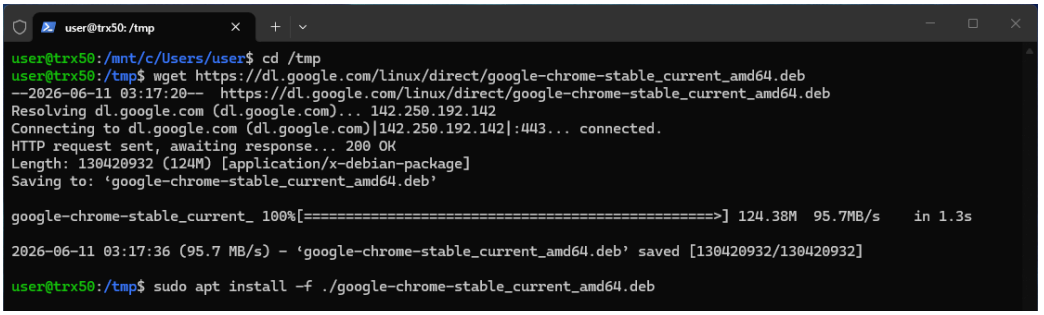
This will launch the specified WSL distribution.

(2) Install Google Chrome for Linux (WSL GUI Support)

To ensure proper functionality of AI TOP Utility features such as opening external links or accessing folders, WSL must support GUI applications.

Please install the Linux version of Google Chrome inside WSL:

Google Chrome inside WSL



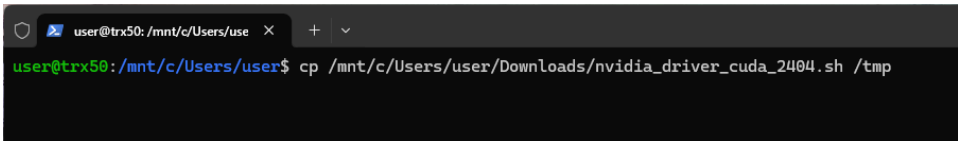
(3) Driver Installation:

Copy the driver installation script to the `/tmp` directory using one of the following methods:

Method 1 : In the WSL terminal, run:

```
cp
/mnt/c/Users/<your_username>/Downloads/nvidia_driver_cuda
_2404.sh /tmp
```

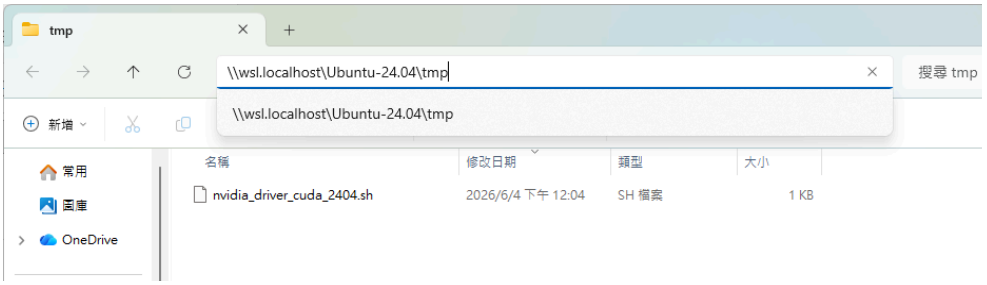
**Note: Replace <your_username> with your actual Windows user name.*



Method 2 (Windows File Explorer):

Copy the file directly to:

\\wsl.localhost\Ubuntu-24.04\tmp



Then, navigate to the directory:

```
cd /tmp
```

Then follow the instructions **Section 1-3 Graphics Card Driver Installation** to install the appropriate GPU driver.
Ensure that the correct driver is installed before proceeding with the WSL and AI TOP setup.

For example:

```
bash nvidia_driver_cuda_2404.sh
```



(4) Setting WSL system memory

By default, WSL2 automatically allocates up to 50% of your total system RAM to the virtual machine (VM) where your Linux distribution runs.

However, you can manually configure how much memory WSL should use by editing the .wslconfig file.

On the Desktop, right-click on the **"Start"** menu > select **"Windows Powershell (System Administrator)"**
Create the .wslconfig file and edit it using the following commands:

```
cd ~
notepad .\.wslconfig
```



memory=192GB → WSL will use a maximum of 192GB of RAM instead of the default 50%. Adjust this based on your needs.
swap=8GB → Creates an 8GB swap file in case RAM usage exceeds the limit (optional).

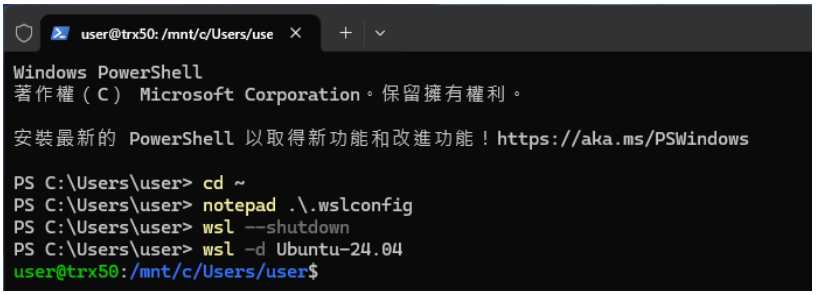


After saving the .wslconfig file, restart WSL for the settings to take effect:

```
wsl --shutdown
```

Then, restart WSL by launching it again:

```
wsl -d Ubuntu-24.04
```



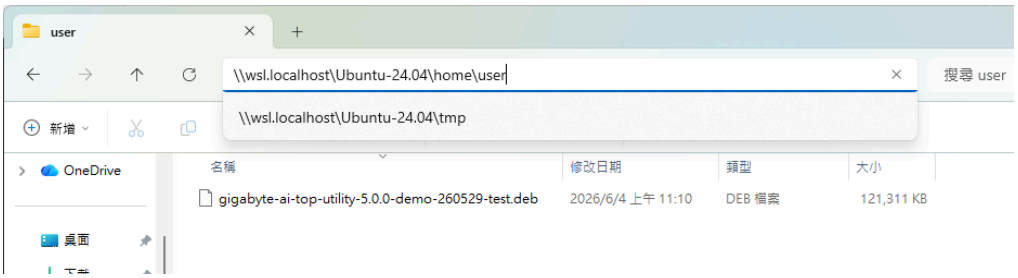
(5) AI TOP Utility Installation

First , download the Installation Package (Required).

Then copied the **.deb** file to this path:

\\wsl.localhost\Ubuntu-24.04\home\<*your_username*>

**Note: Replace <your_username> with your actual user name.*



Then follow the steps below to complete the installation:

Step 1: Update system and install required dependencies

```
cd ~
sudo apt update
sudo apt install -y libasound2t64 libnss3 libxss1
libatk-bridge2.0-0 libgtk-3-0 libgbm1 libxshmfence1
libglu1-mesa libnotify4 xdg-utils libsecret-1-0
```

Step2 : Install AI TOP Utility

```
sudo dpkg -i gigabyte-ai-top-utility-lite.deb
sudo apt install -f
sudo dpkg -i gigabyte-ai-top-utility-lite.deb
```

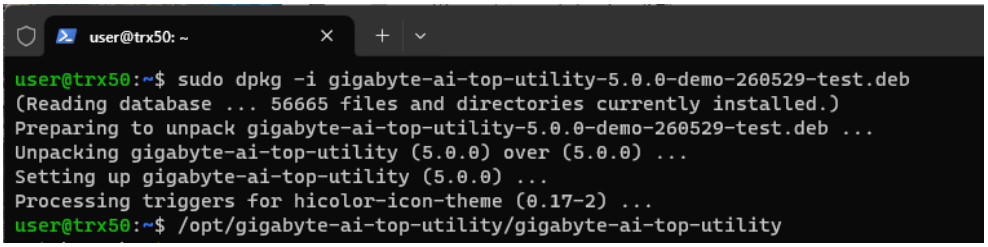
***Note:**

If any dependency issues occur, run:

```
sudo apt --fix-broken install
```

Step 3: Launch AI TOP Utility

```
/opt/gigabyte-ai-top-utility/gigabyte-ai-top-utility
```



Alternatively, the AI TOP Utility can be launched from Windows by searching for **AI TOP Utility** in the Start menu or search bar and clicking the application to open it.



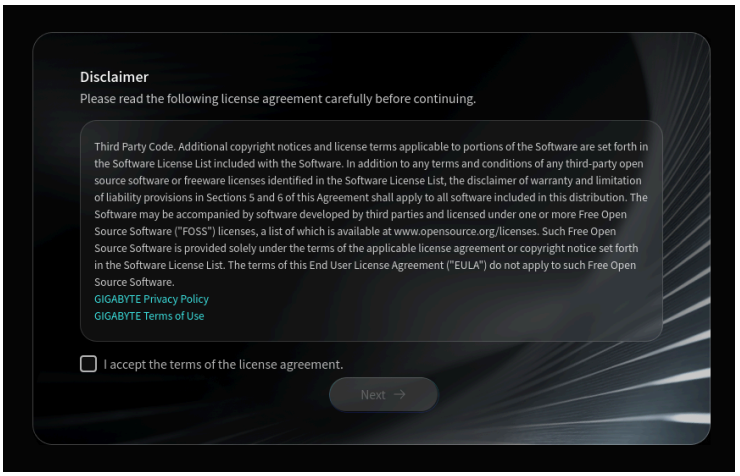
The AI TOP Utility application will open after execution.

1-4-3 Initial Setup and Environment Configuration

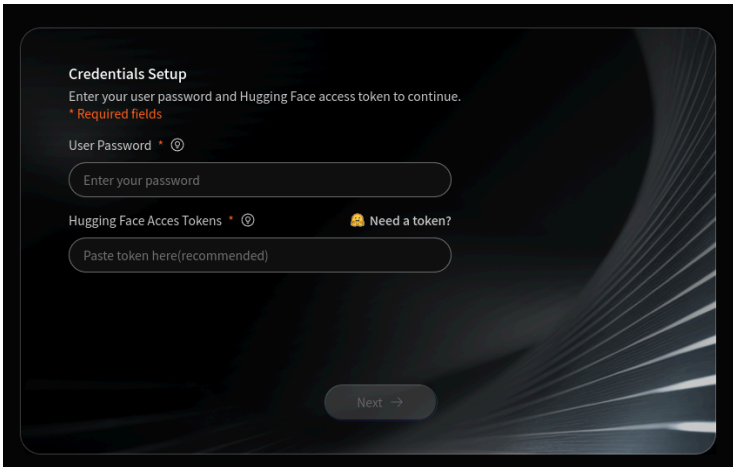
Once all installation and preparation steps have been completed, proceed with the initial setup and environment configuration of AI TOP Utility.



Start by reviewing and confirming the privacy settings.



Then enter your system password and token key. If you do not have a token key, you can click the **Hugging Face** link in the top-right corner to obtain one.



After completing these steps, the system will automatically proceed with downloading the required components and setting up the environment.

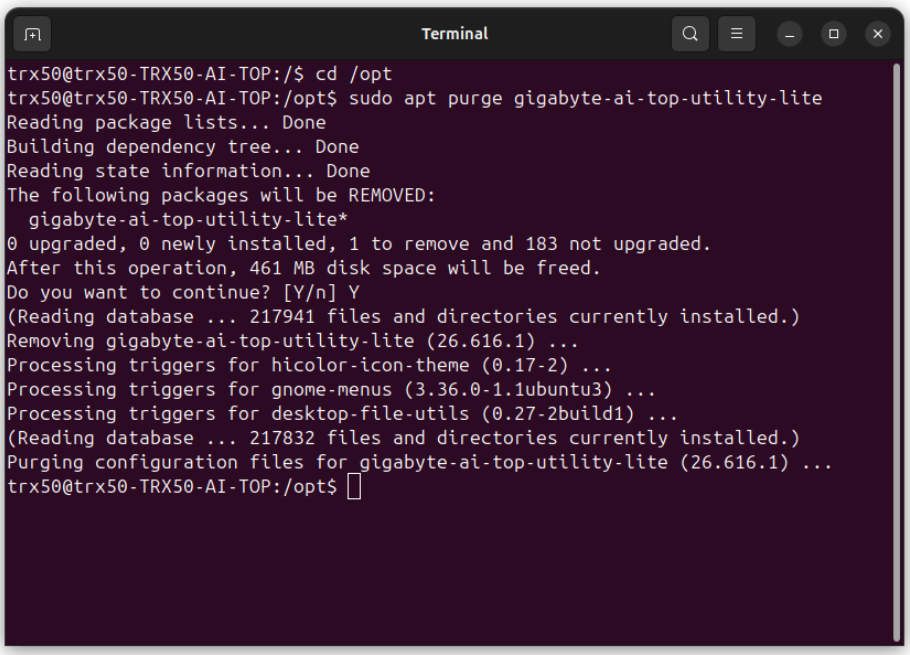


Once the installation is finished, the **AI-Top-Utility-Data** folder will be created in your home directory.

1-4-4 Uninstall AI TOP Utility

To uninstall AI TOP Utility, open a terminal.Windows users should first launch WSL before executing the command:

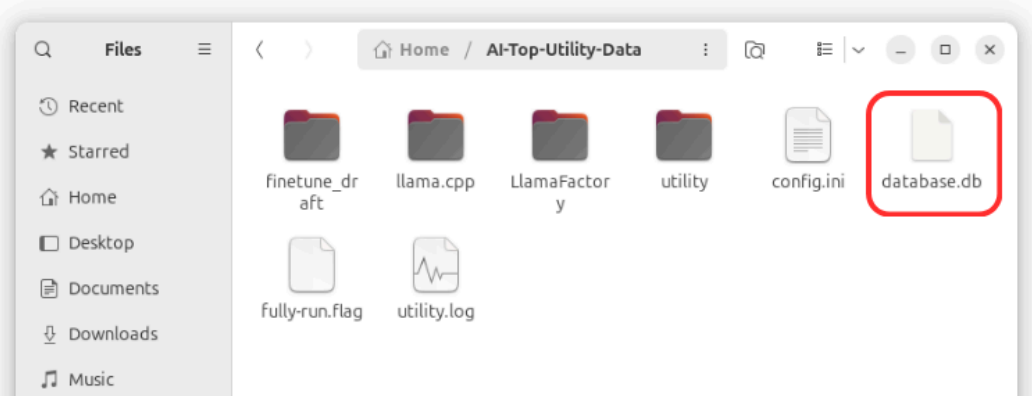
```
sudo apt purge gigabyte-ai-top-utility-lite
```



If users want to completely remove all AI TOP Utility components, they may also delete the AI-Top-Utility-Data directory.

WSL path:
\\wsl.localhost\Ubuntu-24.04\home\<*your_username*>\AI-Top-Utility-Data
**Note: Replace <your_username> with your actual user name.*

If users wish to retain the data stored in the database, **keep the database.db** directory inside the AI-Top-Utility-Data folder. This ensures existing user data is preserved while other components can be safely removed.



2. Feature Descriptions

2-1 Dashboard

Current GPU / CPU load	The usage (%) of GPU/CPU at the moment. Indicates the current percentage of GPU and CPU processing power utilization. This is often monitored to ensure that resources are being efficiently utilized and to diagnose performance bottlenecks.
VRAM state (%)	Displays the percentage of GPU VRAM currently in use on each device, along with the current GPU temperature. This indicates how much of the total GPU memory is being utilized. VRAM is essential for storing the model, its weights, and training data batches during GPU computation.
DRAM state (%)	Displays system DRAM usage, including total memory capacity, used memory, and utilization percentage.

	This metric reflects the overall consumption of dynamic random-access memory (DRAM) in the system. When available memory becomes insufficient, it may limit data processing throughput and impact overall performance. The DRAM state helps users monitor memory pressure and determine whether the system is approaching its memory resource limits.
CPU state (%)	Displays the average utilization percentage across all CPU cores. This indicates how much of the total CPU processing capacity is currently being used. Monitoring CPU usage helps understand its workload and role in the overall system performance, especially for tasks that are not offloaded to the GPU. It also shows the current CPU temperature for hardware monitoring and thermal management.
Power Consumption	Displays the power usage information of the system and PSU during operation. This section includes the accumulated energy consumption, real-time power consumption, and current power usage. The accumulated power consumption (kWh) is measured from system boot and represents the total energy consumed over time. The current power consumption (W) reflects the real-time power draw during workload execution. The power usage indicator provides an overall view of the system's power consumption status. *For now, this feature only supports the GP-AP1600TM PG5 AI TOP.
Activity	A notification center located in the top-right corner of the interface. Users can view ongoing tasks, task progress, and system notifications.
Task	Lists currently running tasks and their execution progress.
Notification	Provides notifications when tasks are completed or failed.

- Dashboard - Dataset

Dataset name	Displays the user-defined name of the dataset.
Generated Data	Displays the total number of data records successfully generated so far. This value updates in real time as the task progresses.

Progress	Visually represents the overall completion percentage (%) of the current dataset generation task.
ETA	Estimated time of accomplishment of the dataset job.
LLM Model	The name of the Large Language Model (LLM) selected for the current dataset generation task.
Embedding Model	The name of the text embedding model selected for vectorization or retrieval evaluation in the current task.

- Dashboard - Fine-tune

Experiment name	Name of your fine-tuning experiment (job). Type a name of your fine-tuning experiment.
Tokens/s	Displays the real-time throughput of the model training at the current step, measured by the number of tokens processed per second.
Progress	Visually represents the overall completion percentage (%) of the current dataset generation task.
ETA	Estimated time of accomplishment of the experiment job.
Training loss	Monitor the training loss at every step. A metric that quantifies the difference between the predicted values by the model and the actual values in the training data. The goal of training is typically to minimize the loss, which is indicative of the model's performance and accuracy.
Training Configs	This section encompasses the core settings configured in the experiment setup tab, dynamically adapting based on the selected Finetune_Type. This helps users to double check the configs with current hardware states in order to get better change in next training settings.
Log	Monitor whatever happens in the training process and check error messages. These are records of events that occur during the training process. Logs include information about the training progress, performance metrics, error messages, and other diagnostic information. They are crucial for debugging and optimizing the training process.

2-2 Model Hub

2-2-1 Model List

Explore Models	Download GIGABYTE-verified and recommended models.
Name	Displays the name of the model shown in the Model Hub.
Status	Displays the current availability and processing state of the model. The status indicates whether the model is ready for use, currently processing, stopped, failed, or being used by another task or service.
Model Type	Defines the category and primary purpose of the model.
Modality	Defines the type of input and output data supported by the model.
Format	Indicates the file format used to store and distribute the model.
Action	The Actions section provides a set of operations for managing models and related tasks. For detailed information about each action, please refer to **Model Hub - Action**.

2-2-2 Model Convert

Convert Format	Select the file format to convert the model into. Supported formats: ● FP16: 16-bit floating point format, commonly used for balanced performance and memory usage. ● INT8: 8-bit integer quantization format, optimized for reduced memory usage with minimal accuracy loss. ● INT4: 4-bit integer quantization format, highly compressed format for maximum memory efficiency.
Quantization Type	Select the quantization precision type to reduce model size and improve inference efficiency. Only GGUF format is currently supported.

- Model Hub - Status

In Progress	The model is currently processing or downloading.

Failed	The operation failed due to an error.
Missing	Required model files are missing.
Completed	The model is ready to use.
In Use	The model is currently being used by another task or service and cannot be used for other tasks.

- Model Type

LLM	The language model used for dataset generation.
Embedding Model	The embedding model used in the dataset generation pipeline to convert data into vector representations.
LoRA Adapter	A lightweight fine-tuning adapter based on LoRA (Low-Rank Adaptation) technology, used together with a base model to efficiently adapt the model for specific tasks without modifying the original model weights.

- Modality

Text	Specifies that the model is a text-only large language model (LLM).

- Format

safetensors	A standard model format commonly used for model training and GPU inference. It provides fast and secure tensor loading with broad framework compatibility.
gguf	A lightweight model format optimized for local inference and quantized model deployment. It is suitable for resource-limited or edge-device environments.

- Model Hub - Action

Cancel	Cancels the current operation or task in progress.
Locate Model	Allows users to reassign the local file path of a missing model so it can be restored and used again.
Delete	Removes the selected model, task, or configuration from the system.

Retry	Allows a previously failed or stopped task to be executed again, with the ability to adjust the original settings before retrying.
Show in Folder	Opens the local directory where the selected model or generated files are stored on the user's device.
Generate Dataset	Navigates to the dataset generation page using the selected model as the base model.
Inference	Navigates to the inference page and runs the selected model for prediction or output generation.
Fine-tune	Navigates to the fine-tuning page and uses the selected model as the base model for fine tuning.
Convert	Creates a copy of the selected model and converts it into a different format or quantization type without modifying the original model. For detailed information, please refer to 2-2-2 Model Convert.
View Exepriment	Displays the fine-tuning experiment from which this model was generated, including its configuration, logs, and training results.

2-3 Datasets

Generate Datasets	Generate your Dataset.
Name	Displays the name of the model shown in the Datasets list.
Status	Displays the current availability and processing state of the model. The status indicates whether the model is ready for use, currently processing, stopped, failed, or being used by another task or service.
Data Quantity	Defines the category and primary purpose of the model.
Create time	Defines the type of input and output data supported by the model.
Action	The Actions section provides a set of operations for managing models and related tasks. For detailed information about each action, please refer to the descriptions below.
Dataset Name	Users can enter and define a custom name for the dataset to help identify and manage datasets easily.

LLM Model	The language model used for dataset generation.
Embedding Model	The embedding model used in the dataset generation pipeline to convert data into vector representations.
Verbatim	The Verbatim setting controls how the model generates answers based on the provided dataset. ● True: The model generates answers strictly based on the source text only. ● False: The model refines the answer using its internal knowledge.
Upload File	Upload reference files for dataset generation.
Add Data	Allows users to manually add a single dataset entry with Instruction, Input, and Output content.
Action	The Actions section provides a set of operations for managing dataset and related tasks. For detailed information about each action, please refer to **Dataset - Action**.

- Dataset- Action

Fine-tune	Navigates to the fine-tuning page and uses the selected dataset for finetuning.
View Detail	Display key information and metadata of the dataset, such as dataset status, data quantity, and related model information.
Rename	Rename the dataset.
Retry	Re-runs dataset creation using existing settings, with optional modifications before execution.
Save Current Data	Saves the dataset generated up to the current point when generation is manually paused or stopped.
Delete	Deletes the selected dataset. The data will no longer be available for use.

2-4 Fine-tune

New Experiment	Create a New Fine-tuning Experiment.

Name	Displays the name of the model shown in the Model Hub. Each model entry may include additional sub-information displayed under the name, such as training runtime, model type and ETA.
Status	Displays the current availability and processing state of the model. The status indicates whether the model is ready for use, currently processing, stopped, failed, or being used by another task or service.
Finetune Type	Describes the fine-tuning method, including: Full, Freeze, LoRA, QLoRA.
Duration	Indicates the total time consumed by a task or process, reflecting the overall duration from start to completion.
Create time	Displays the timestamp when the task or item was created in the system.
Action	The Actions section provides a set of operations for managing fine-tune and related tasks. For detailed information about each action, please refer to **Finetune - Action**.

- General Setting & Finetuning Method

Experiment Name	Name of your fine-tuning experiment (job). Type a name of your fine-tuning experiment.
LLM Model	Refers to the pre-trained model architecture used as the starting point for fine-tuning. Select a pre-trained large language model (LLM) model that you want to process fine-tuning.
Dataset	Specifies the dataset used for training the model during fine-tuning. Select a dataset for fine-tuning process (txt, csv, json, json).
Finetuning Strategy	Defines the level of control and configuration complexity for the fine-tuning process. This feature is designed by GIGABYTE to provide simplified and user-friendly fine-tuning workflows, especially for users without prior AI or machine learning experience.
Quick Start	A GIGABYTE-designed simplified fine-tuning mode that automatically configures essential training parameters based on optimized internal presets.

	This mode allows users to start fine-tuning without requiring detailed knowledge of model configurations or training settings.
Customization	Provides full manual control over fine-tuning parameters for advanced users who require detailed configuration flexibility.
Save	Saves the current fine-tuning configuration and all modified parameters for later use or further editing.
Finetuning Type	Describes the fine-tuning method, including: full, freeze, (q)lora.
Full	Description: All parameters of the model are updated during the training process. This approach involves adjusting the weights of all layers of the model based on new training data. Advantages: High performance can be achieved on specific tasks or data sets because the model fully adapts to the nuances of new data. Disadvantages: Comprehensive fine-tuning requires significant computing resources and may lead to overfitting if the fine-tuning data set is small compared to the original training data set. It can also lead to “catastrophic forgetting,” where the model loses its ability to perform on the task it was originally trained on.
Freeze	Description: Some layers of the model remain frozen during fine-tuning (i.e. their weights are not updated), while other layers are allowed to update. Advantages: Freezing layers reduces computational cost and the number of parameters that need to be updated. It also helps retain the general knowledge learned by the model and mitigates the risk of catastrophic forgetting. Disadvantages: Since only a part of the parameters of the model are updated, the model's ability to adapt to new tasks may be limited, which may be insufficient compared with comprehensive fine-tuning.
Freeze Trainable Modules	Name of trainable module when using freeze.
Freeze Trainable Layers	Specifies how many layers of the model are open to adjustments during the fine-tuning process.
LoRA	LoRA (Low-Rank Adaptation) is a lightweight fine-tuning method that modifies pre-trained models by adding small trainable low-rank matrices to the existing model weights instead of replacing the original parameters.

QLoRA	Quantized version of LoRA, likely involving fewer bits to represent the LoRA parameters, which could save memory and computation.
Lora Target / QLora Target	Specifies which parts of the model the LoRA adaptation targets. It refers to the modules (query, key, value, output, others.. projections) to apply the adapter to. The target modules will vary depending on each LLM model.
Lora Rank / QLora Rank	The dimension of the matrix decomposition used in LoRA. Its range follows the rule 2^n to optimize the learning principle of neural networks. Range: [0, 2, 4, 8,16, 32, 64, 128, 256,...]
Lora Alpha / QLora Alpha	The scaling factor for the lora weights. Its range follows the rule 2^n to optimize the learning principle of neural networks. Range: [0, 2, 4, 8,16, 32, 64, 128, 256,...]
Lora dropout / QLora dropout	The probability of applying dropout to the LoRA weights during training. Range: [0 ~ 1].
Mixed Precision	The datatype of the weights in the LLM backbone decides the precision of the model after fine-tuning. Options: fp32, fp16, bf16, pure bf16.
Learning Rate	The learning rate is the speed at which the model updates its weights after processing each mini-batch of data. Range: [0 ~ 1].
LR Scheduler Type	The strategy for adjusting the learning rate over between epochs or iterations as the training progresses. Options: cosine, linear(Only for LLM finetune)
Flash Attention	An efficient attention mechanism that reduces VRAM usage and accelerates training speed (Tokens/s).
Batch size	The number of training examples a mini-batch uses per single GPU during an iteration of the training model. Its range follows the rule 2^n to optimize the learning principle of neural networks. Range: [0, 2, 4, 8,16, 32, 64, 128...].
Epochs	The number of times the learning algorithm goes through the entire training dataset.
Save Checkpoint Strategy	The checkpoint saving strategy to adopt during training. Options: None, steps, epochs “no”: save no checkpoint until finishing training process “steps”: periodically save checkpoint at every n training steps “epoch”: periodically save checkpoint at every training epoch.

Save Steps	Number of training steps between checkpoint saves. Example: save_steps = 100 means periodically save checkpoint at every 100 training steps (100, 200, 300, …) finishing training process.
Max Length	The maximum length of the input sequences LLM used during model training. It decides the maximum length of the "input query" and "output answer" to inference the LLM. Normally 1000 tokens ~ 750 words by Open AI standard. Range: [512, 1024, 2048,...,128k] depends on the capability of pre-trained LLM.
Number of GPUs Used	Number of GPUs used for training process. It automatically marks all the available GPUs in the current workstation PC.
Offloading Memory Strategy	The unique technique by Gigabyte to provide users multiple choices to offload model optimizer states, gradients, parameters and optionally activations to CPU to optimize hardware capability and avoid out-of-memory issues.
Expand Config	The unique technique by Gigabyte to provide expert mode for professional LLM trainers who want to edit or add more training configs to fine-tune LLM in their way. Please refer to **Section 3-7 Finetune Expand config syntax.** Example: --key1 value1 --key2 value2
Enable Validation	Determines whether validation is automatically executed after fine-tuning is completed. When enabled, the system will run a validation process immediately after training finishes to evaluate model performance.

- Fine-tune - Status

In Progress	A fine-tuning experiment that is currently running.
Stopped	A fine-tuning experiment that was manually stopped before completion.
Failed	A fine-tuning experiment that failed due to an error during execution.
Completed	A fine-tuning experiment that has finished successfully.
Draft	A fine-tuning experiment that has been created and saved, but has not started execution.

- Fine-tune - Action

Inference	Use the selected finetuned model for inference. This action will redirect the user to the Inference page, where the selected model is preloaded for testing or evaluation.
Copy	Copy all configuration parameters of the selected finetune task. This allows users to quickly duplicate the same training setup for a new experiment.
View Model	View the finetuned model in the Model Hub. This action will redirect to the corresponding Model Hub page, where model metadata and files are available.
Download Report	Download the full finetune report for the selected experiment. This provides a complete snapshot of the run for offline review or sharing, including configuration and results where applicable.
Delete	Permanently delete the selected finetune task. This action will remove: ● the finetune task record ● all related local data ● the finetuned model artifacts (if the model is available in the Model Hub) Warning: This action is irreversible.
Stop	Stop the fine tuning progress.
Rerun	Rerun the stop or failed finetune task.

2-4-1 Information

Training Loss	Monitor the training loss at every step. A metric that quantifies the difference between the predicted values by the model and the actual values in the training data. The goal of training is typically to minimize the loss, which is indicative of the model's performance and accuracy.
Training Metrics	Shows token usage during training. Average Tokens(Tokens/s): Average number of tokens processed per training step. Total Tokens: Total number of tokens processed during training.
Training Configs	Provides the configuration settings used for fine-tuning, including key training parameters.
Log	Monitor whatever happens in the training process and check error messages. These are records of events that occur during the training

	process. Logs include information about the training progress, performance metrics, error messages, and other diagnostic information. They are crucial for debugging and optimizing the training process.

2-4-2 Benchmark

GPUs Number	Total number of GPUs available or in use.
GPUs Average Usage	Average GPU utilization during training or inference.
GPUs Average Temperature	Average GPU temperature during execution.
CPUs Number	Total number of CPU cores available or in use.
CPUs Average Usage	Average CPU utilization during training or inference.
CPUs Average Temperature	Average CPU temperature during execution.
VRAM usage	GPU memory usage statistics during training, including peak and total usage.
DRAM usage	CPU memory usage statistics during training, including peak and total usage.

2-4-3 Validation

Create Validation	Creates a validation task to evaluate the fine-tuned model against its backbone model using ROUGE scores. (ROUGE scores are multiplied by 100 for display purposes.)
Before score	ROUGE score of the backbone (base) model, used as the baseline for comparison.
After score	ROUGE score of the fine-tuned model after training, used to measure performance improvement over the backbone model.

2-5 Inference

Inference Type	Select the model inference mode.
LLM / LMM Model	Specifies the model type used for inference. Currently supports LLM (Large Language Model) models.
Maximum Tokens	Maximum response length for the model. Reduce this value if you want shorter replies, or increase it for more detailed

	responses.
Offload GPU	Sets how many model layers are loaded into GPU VRAM for processing. Higher values generally provide better performance.
CPU Threads	Number of CPU threads used to process the remaining model layers that are not offloaded to the GPU (handled in DRAM).
Content Window	The length of context the model can retain and use. Higher values allow the model to process more input, but require more memory and computing resources.
Top-p	Controls how wide the model samples possible outputs. Higher values increase diversity, while lower values make responses more focused and stable.
Temperature	Controls output randomness. Lower values produce more consistent responses, while higher values increase creativity and variation.
System Prompt	Defines the model's role, behavior, and response style, guiding how it should respond to user inputs.
Chat	Provides an interface for interacting with the selected LLM model through natural language conversation. Users can ask questions and receive real-time responses generated by the model. The chat interface also includes the following features: 1. Real-time system resource monitoring, including VRAM usage (per GPU), DRAM usage (current and total), and number of GPUs in use. 2. An Erase function that clears the current conversation history, allowing users to start a new session.

2-6 Settings

Model Path	Defines the default directory where downloaded models are stored.
User Password	Your computer's root (administrator) password. This is used during environment setup.
Hugging Face Access Token	Provide your Hugging Face access token. This is used to download models.
Environment Settings (Repair & Reinstall)	Use this option to repair and reinstall the software environment in case of installation failures or corrupted files. This feature is

	recommended only when troubleshooting serious setup issues.
Version Info	This section displays the AI TOP Utility version, company information, copyright notice, license information, and versions of the software components used by the system.
Policy	End user license agreement (EULA) in English, Simplified Chinese and Traditional Chinese version.

3. User Workflow Guide

This section provides a step-by-step guide for using the AI TOP Utility Lite, including model management, dataset generation, model training, and inference.

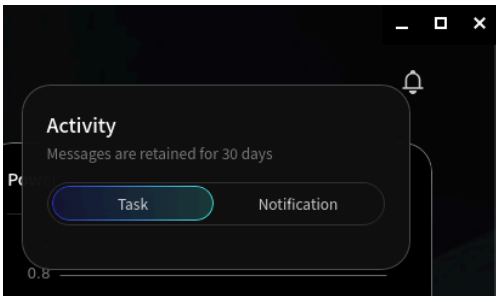
3-1 System Dashboard Overview

This page provides real-time monitoring of system hardware performance. It helps users understand overall system status by displaying GPU and CPU usage, memory consumption, temperature, and power usage in real time.

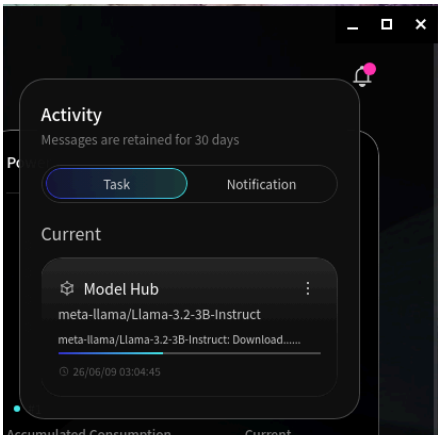


- Activity

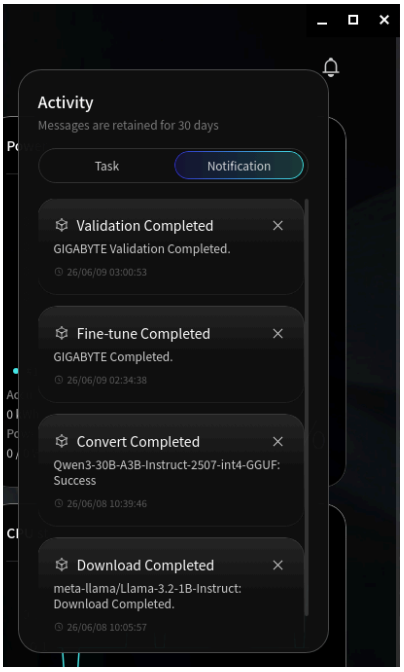
The Activity panel is accessed via the bell icon on the page. It contains two sections: Tasks and Notifications.



The Tasks section displays all ongoing processes, along with their progress status.

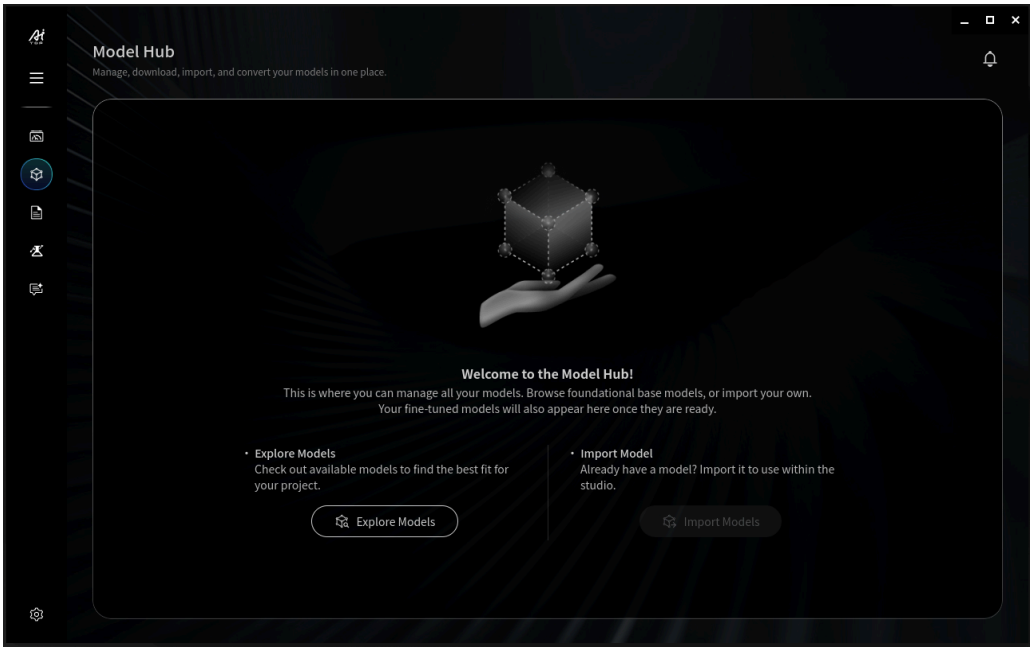


The Notifications section shows status updates such as completion or failure. Users can click on a notification to navigate directly to the corresponding page for more details.



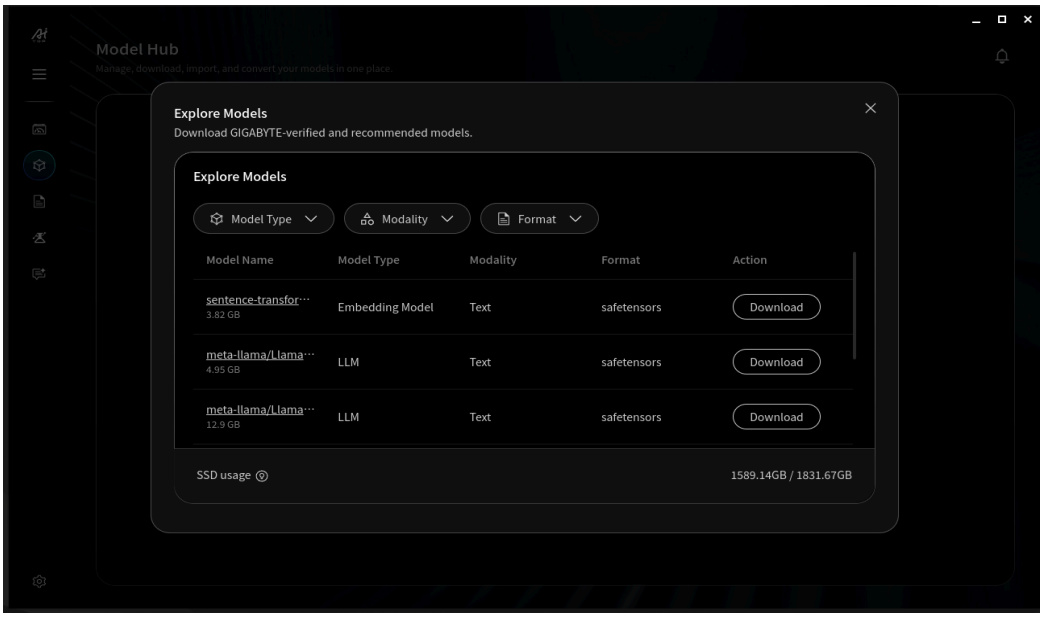
3-2. How to download models from the Model Hub?

Users can browse and download recommended models from the Model Hub for dataset generation, fine-tuning, and inference. Downloaded models can be managed in the Model List for further operations.



(1) Explore Models

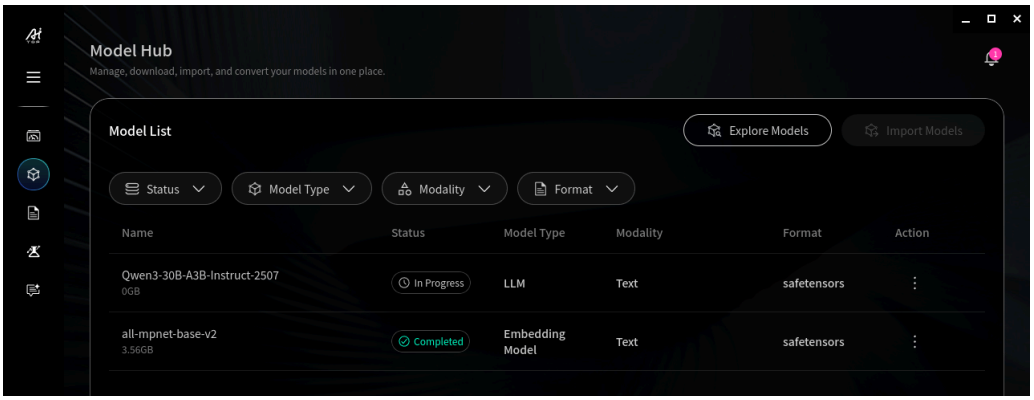
Click the **"Explore Models"** button to view a list of recommended models available in the Model Hub. The model list displays GIGABYTE-verified recommended models.



You can filter the available models by Model Type, Modality, and Format.

Clicking a model name will open its corresponding page on **Hugging Face**, where you can view detailed information about the model.

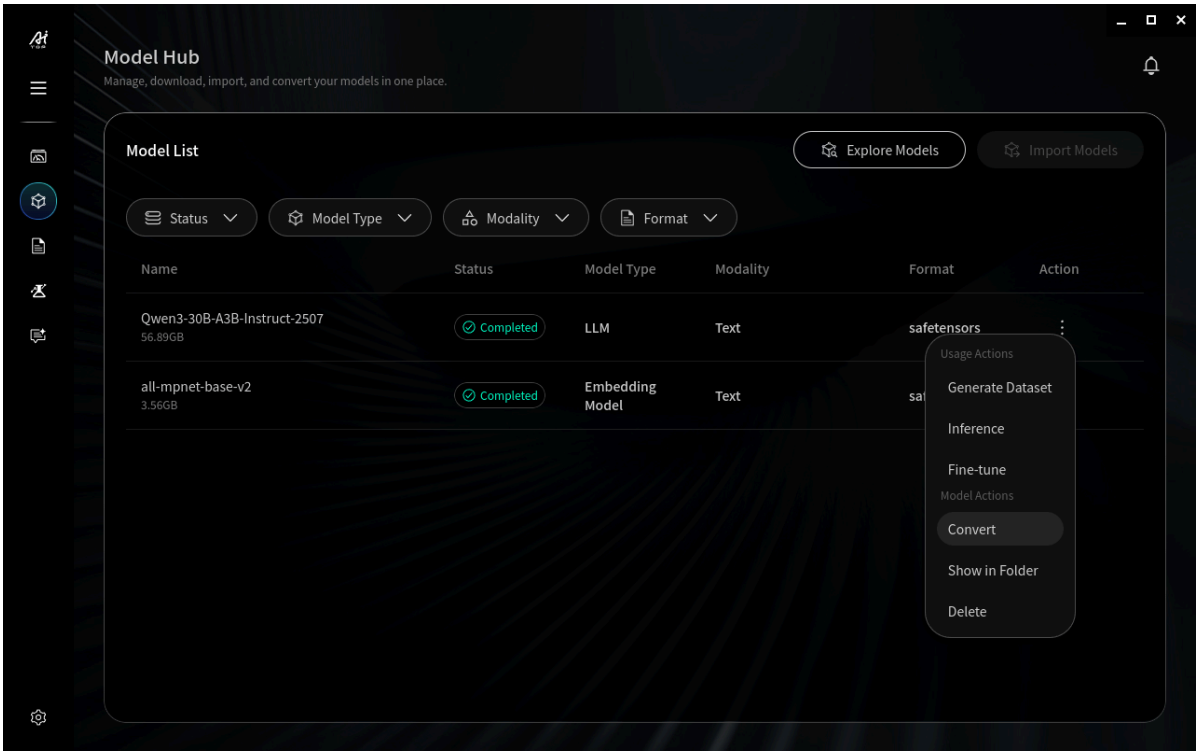
(2) Click **“Download”** to download the selected model.
During the download process, the model status will be displayed as ***In Progress*** in the Model List. After the download is completed, the status will be updated to ***Completed***. Users can also monitor the progress in the Activity panel.



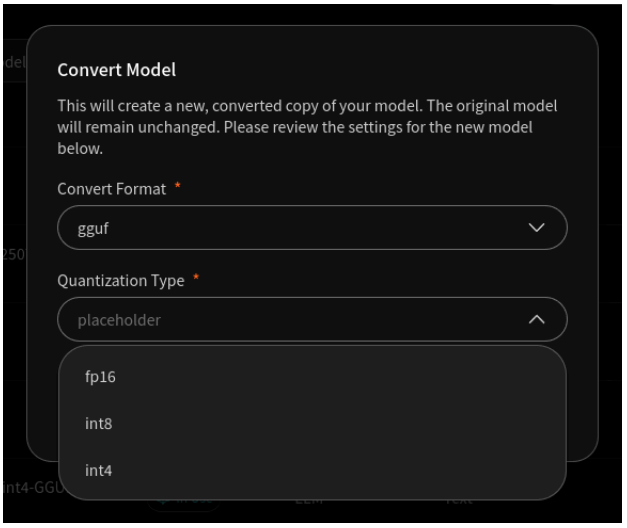
3-2-1 Convert

The Model Convert feature enables users to convert models into optimized formats with different quantization levels, balancing performance, memory usage, and accuracy for efficient inference.

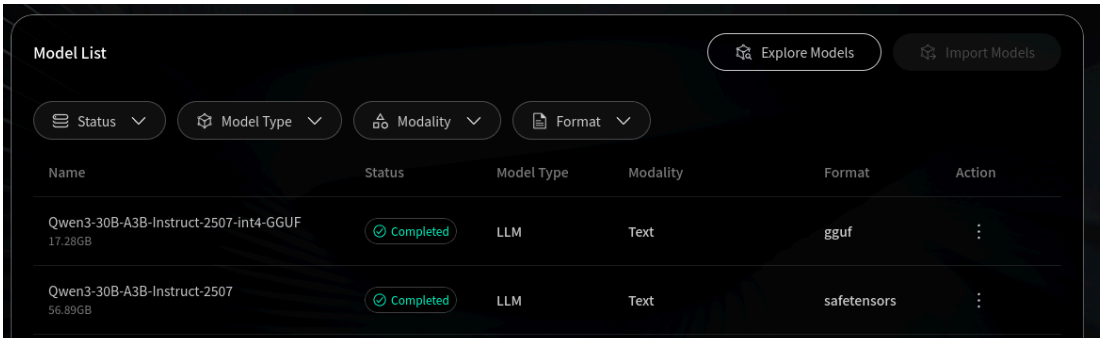
1. To convert a model, select a model from the Model List and click **“Action” → “Convert”**.



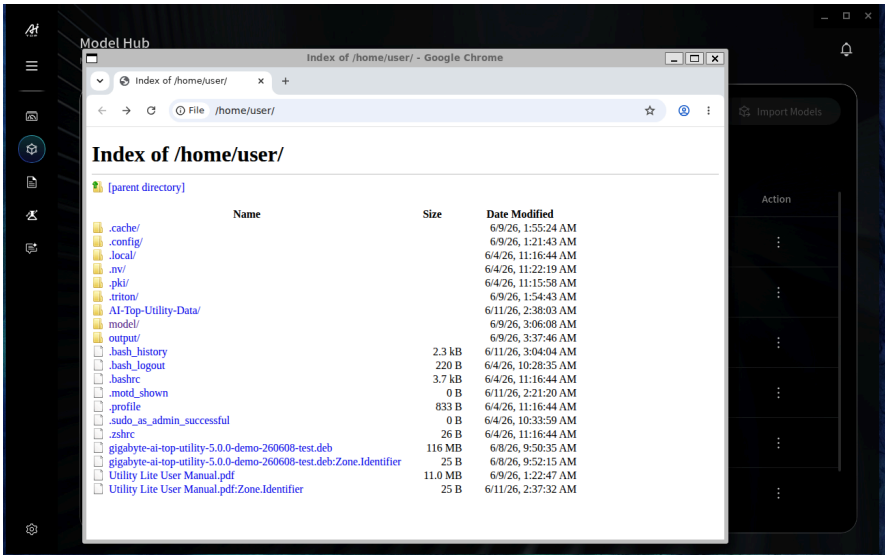
2. We support multiple quantization types, including FP16, INT8, and INT4, with the convert format set to GGUF.



3. During the conversion process, the selected source model will be marked as In Use, and the newly generated model will be marked as In Progress. Models marked as ***In Use*** cannot perform other tasks simultaneously.
4. After the conversion is completed, the converted model will be available in the Model List.

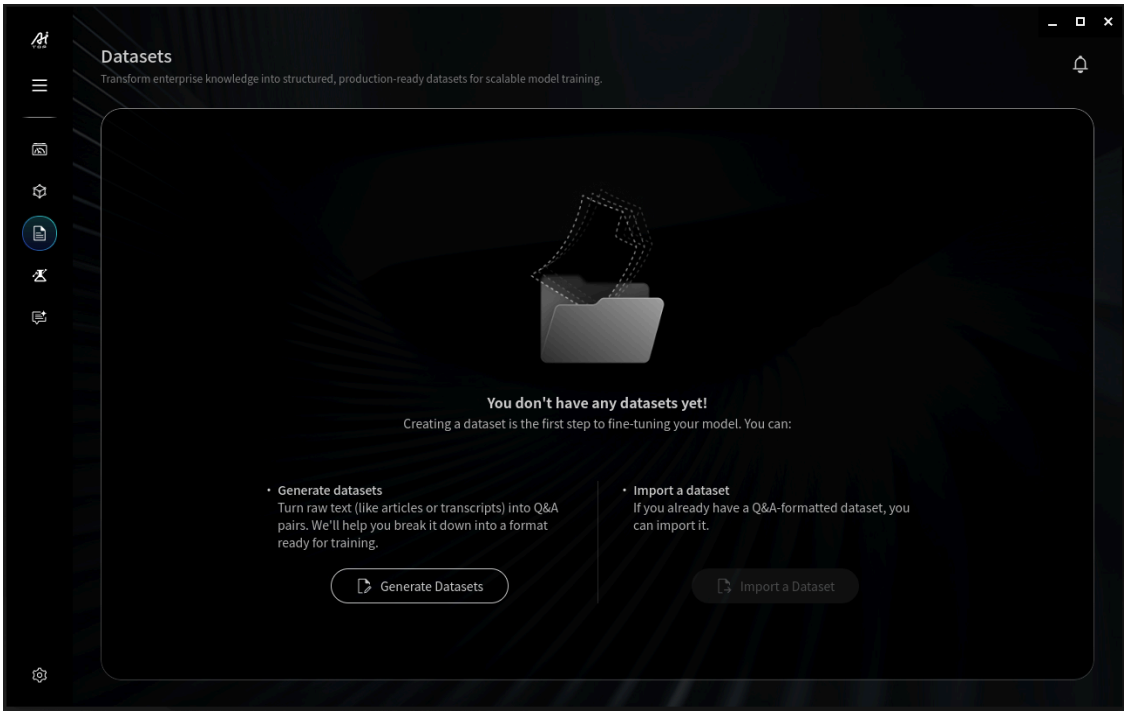


5. For WSL users, the **“Show in Folder”** function will open the **/home/user** directory. To view the model files, please navigate into the **model** folder.



3-3. How to Generate Structured Datasets from Raw Data?

The Dataset feature converts raw documents into structured Q&A datasets for LLM fine-tuning. Users can upload reference documents to generate training-ready datasets that can be directly used in the model training workflow.



(1) Click the **"Generate Datasets"** button.
Enter the dataset name, select the LLM model and embedding model, and choose whether to enable the verbatim feature for dataset generation.

Recommended Models

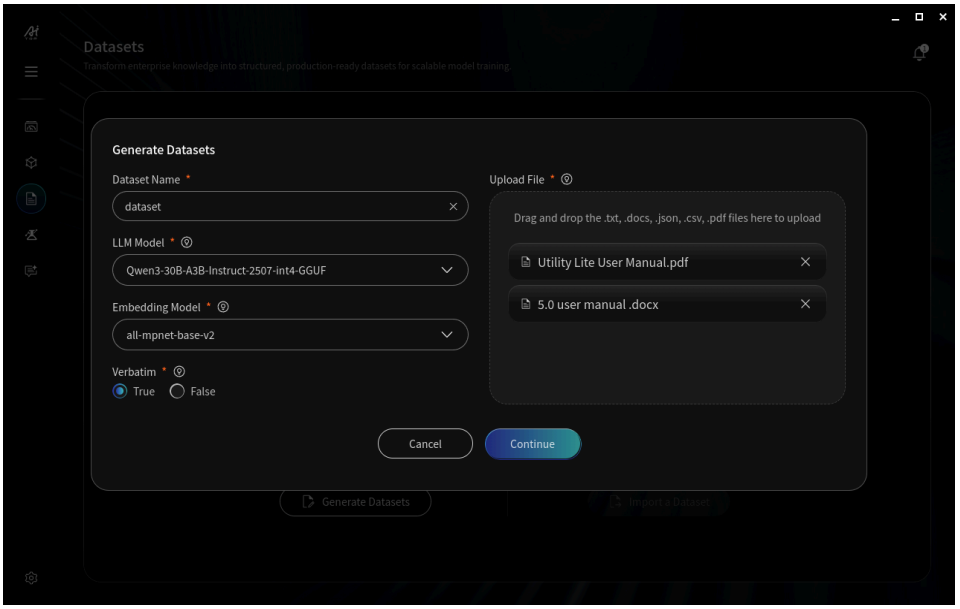
- **LLM Model:** Qwen3-30B-A3B-Instruct-2507 int4 GGUF (Converted)
- **Embedding Model:** all-mpnet-base-v2

If no model is available, please refer to the model download steps in **Section 3-2.**

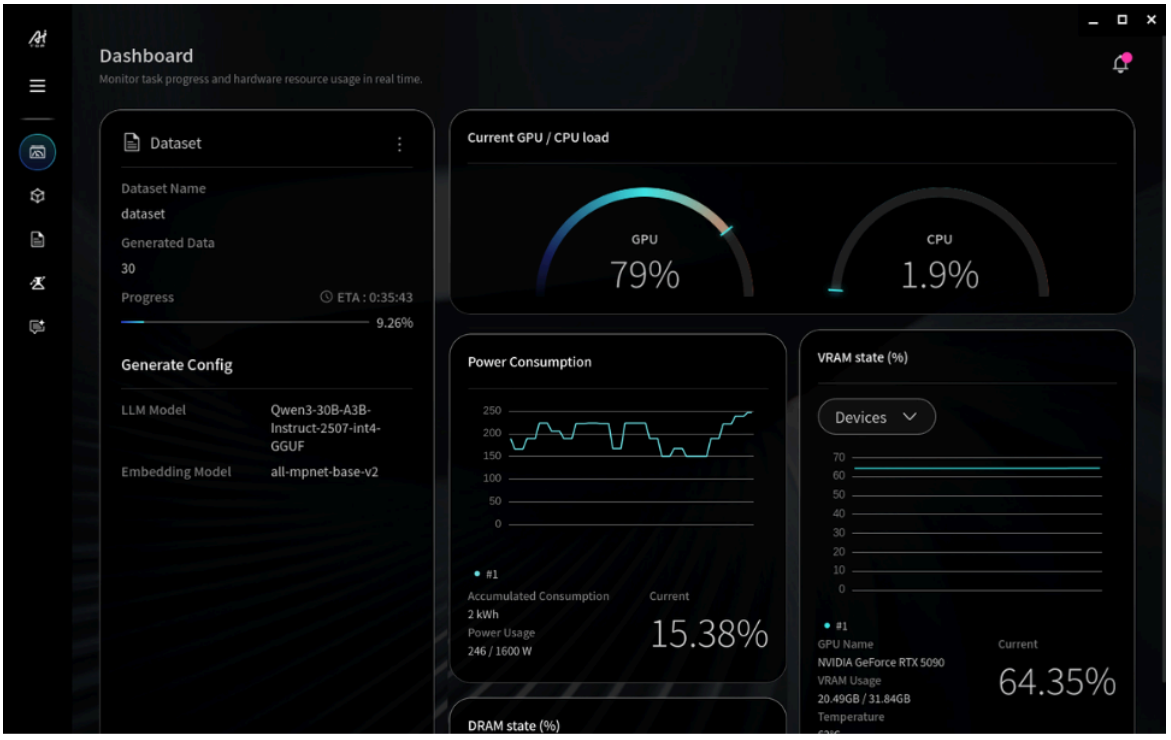
(2) Upload one or more document files used to generate the dataset. These files serve as the source data for dataset generation. We support the following formats: **TXT, DOCX, JSON, CSV, PDF**.

WSL users must copy dataset files into the WSL directory before uploading to ensure proper access. (e.g. **\\wsl.localhost\Ubuntu-24.04\home\<*your_username*>**)
**Note: Replace <your_username> with your actual user name.)*

*Additionally, WSL does not support dragging and dropping files directly from Windows into the application. Files must be accessed through the WSL file system.



(3) When dataset generation is in progress, a new panel will appear on the left side of the Dashboard to display the current status and details of the dataset.



This panel includes the dataset name, the number of generated data entries, a progress bar showing the generation progress, and the estimated time of arrival (ETA). It also displays the generation settings, including the selected LLM model and embedding model.

***Note: It is recommended to use at least 30 dataset entries to ensure that fine-tuning and validation can be executed properly. This serves as a suggested baseline for system operation, but does not guarantee the stability or accuracy of training and evaluation results. Using a smaller dataset may still lead to unstable behavior or less reliable evaluation outcomes.**

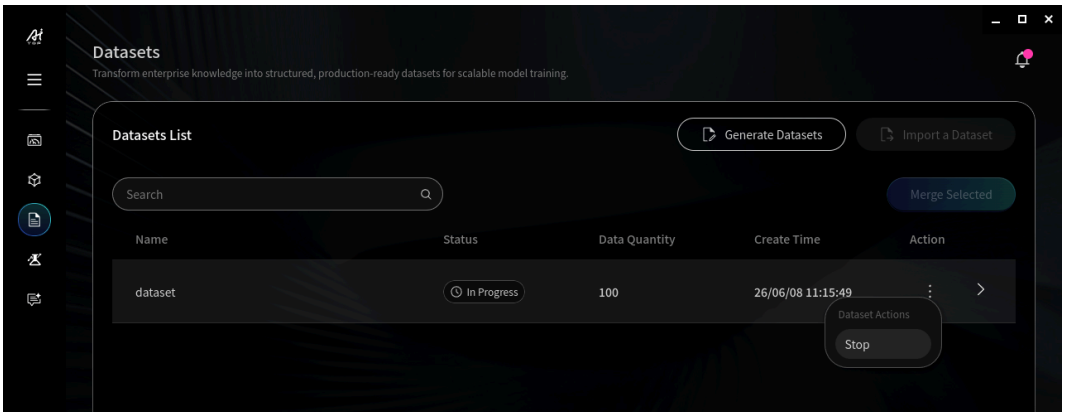
(4) Dataset Generation Completion

There are two possible ways for dataset generation to be completed.

1. The model automatically generates the dataset until it determines that no additional data can be produced. Once the process is finished, the dataset will be marked as Completed. In this case, the total number of generated entries is determined by the model itself, and is not predefined by the user.

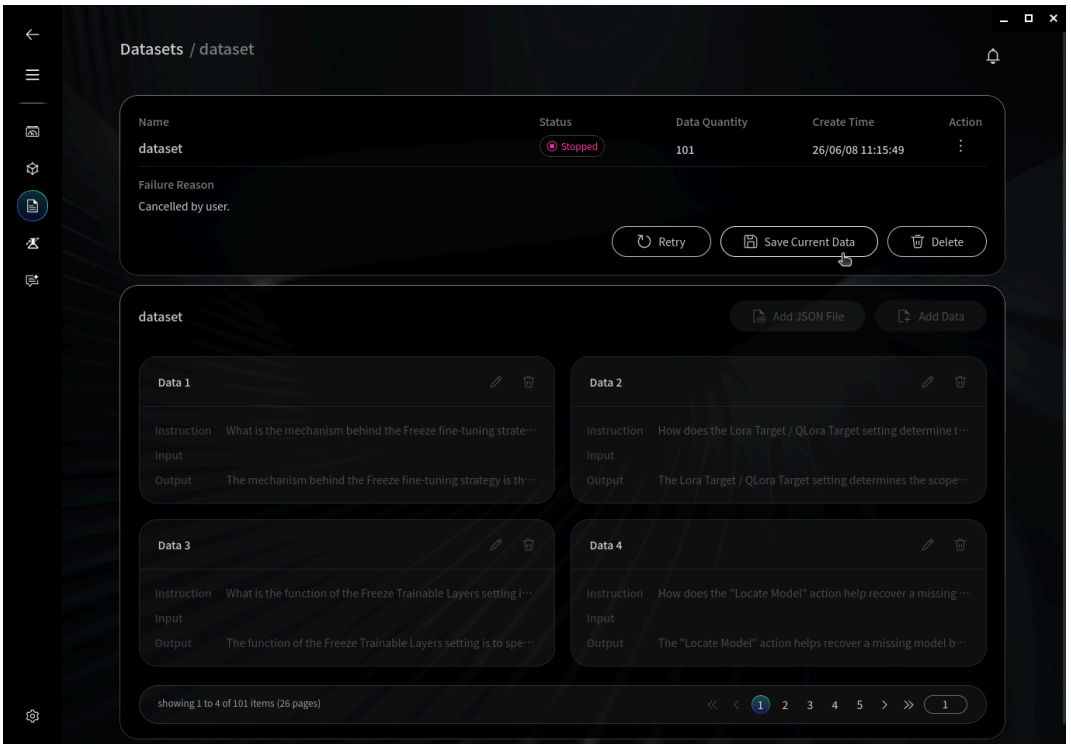
2. The generation process is manually stopped by the user.

   This typically occurs when the dataset reaches the desired number of entries or meets the user's requirements (for example, generating 100 entries). The user can click the **"Stop"** button to pause the generation process.

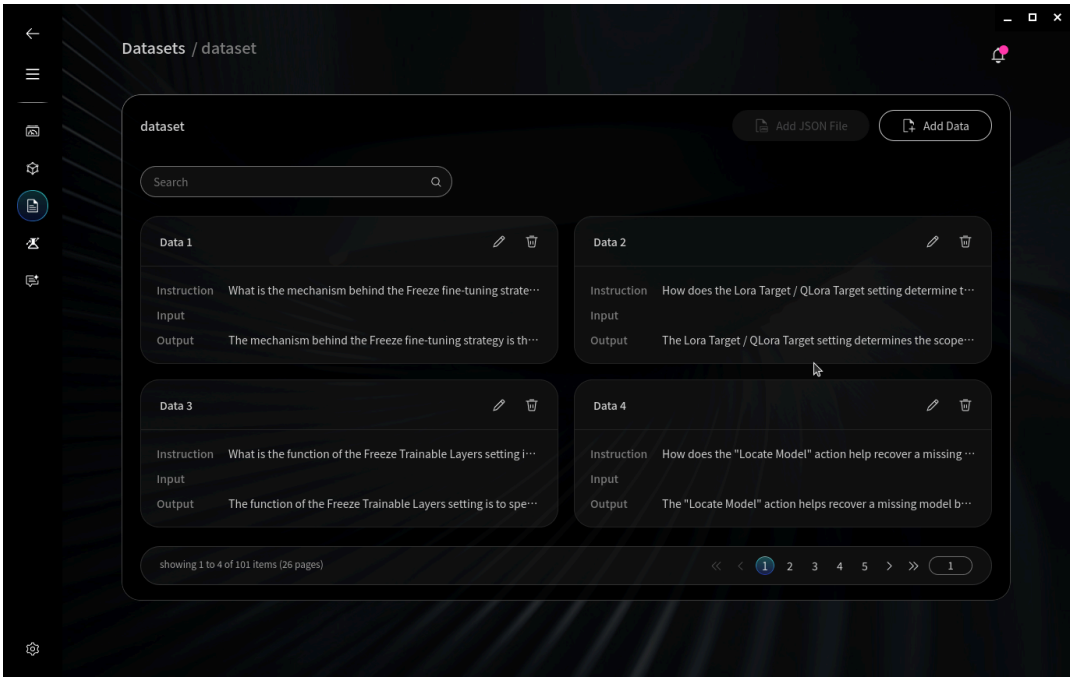


After stopping the process, the user may click **"Save Current Data"** in the dataset panel to store all currently generated data. The saved dataset will be

marked as completed.



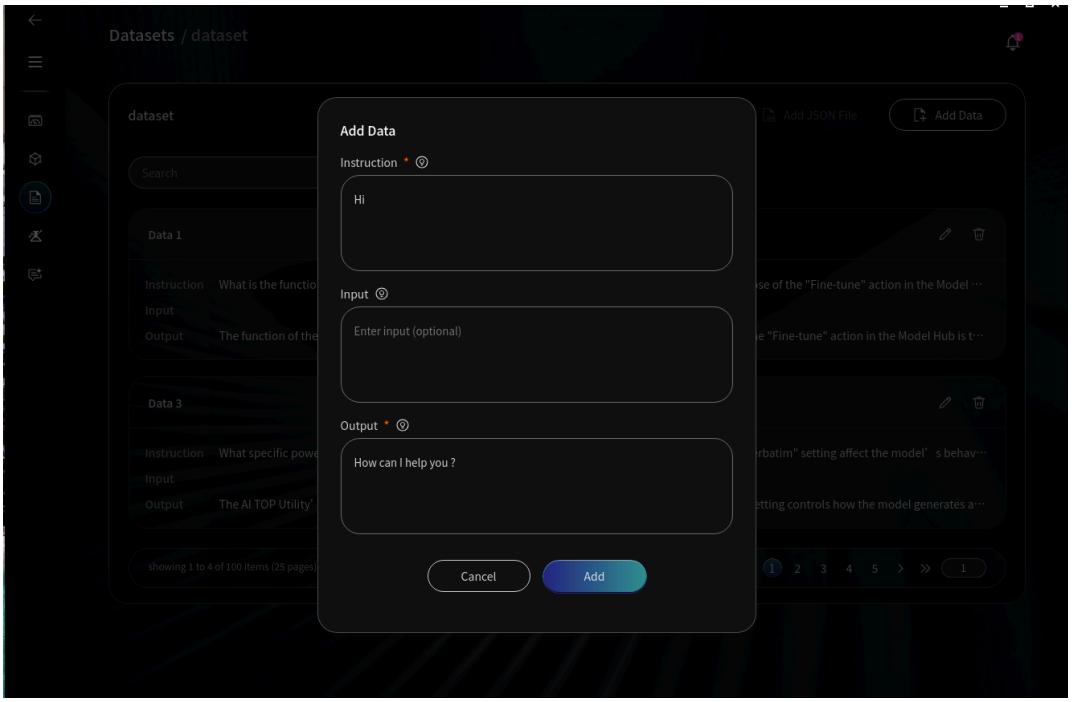
Once either of these scenarios is finished, the dataset generation process is considered complete.Then you can see your dataset



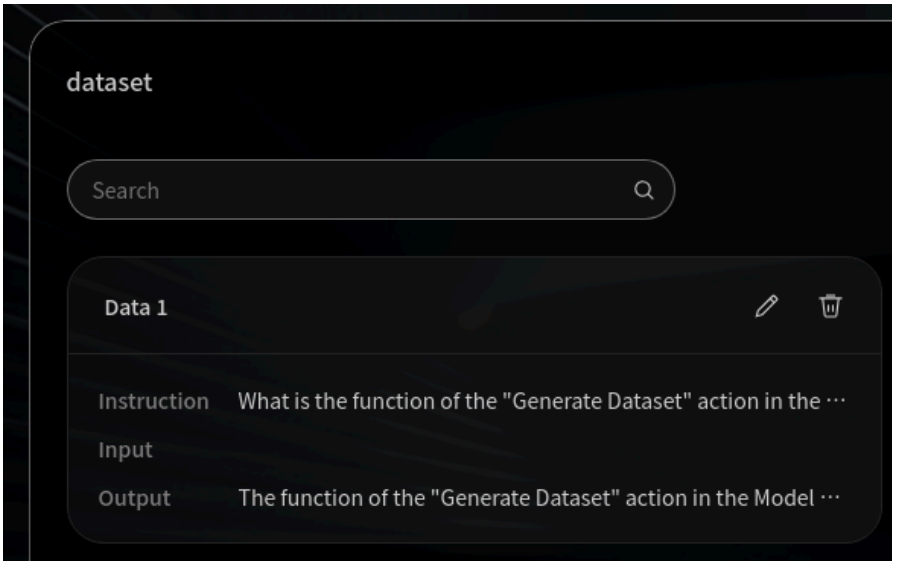
(5) Add and Edit Data

You can manually manage individual data entries after dataset generation using the **Add Data** function.

By clicking **“Add Data”**, users can create a new entry by filling in the Instruction, Input (optional), and Output fields, then clicking **“Add”** to submit.



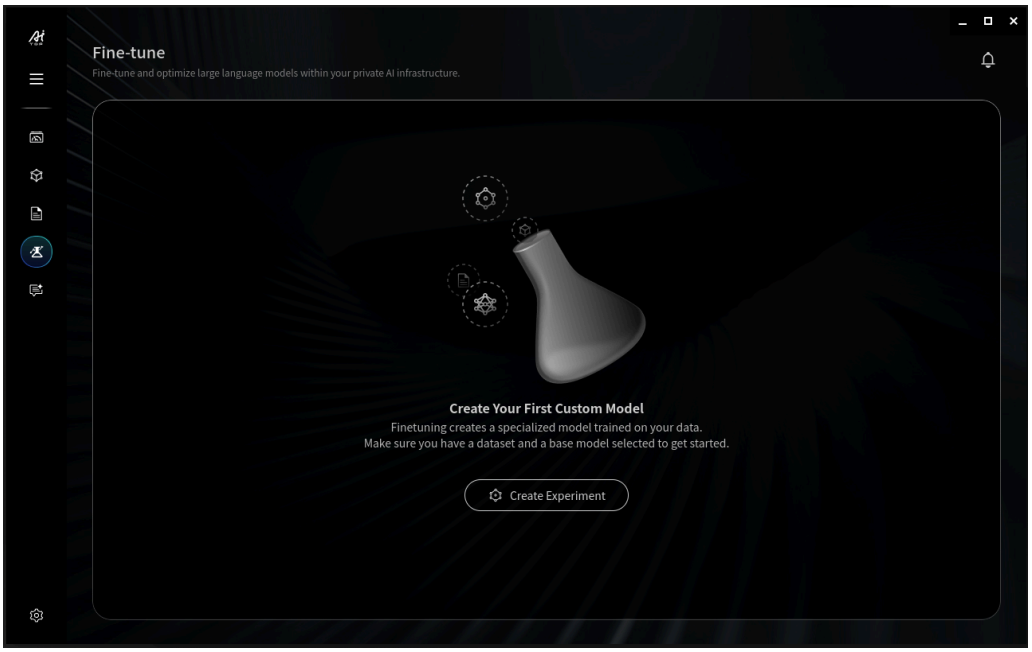
Existing data entries can also be edited or deleted individually. Clicking the pencil icon allows users to edit a single entry, while the trash icon removes the selected entry.



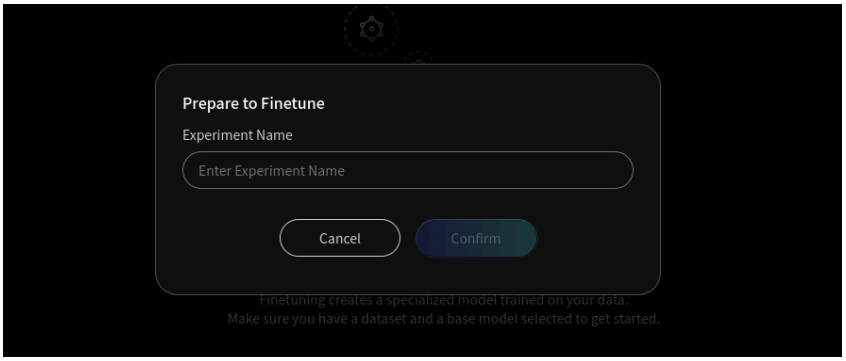
3-4. Model Fine-tuning Workflow

The Fine-tune feature enables users to train a pretrained model on a selected dataset to create a specialized model for specific tasks. It provides a quick-start option that allows users to begin training easily without prior experience in model training. It is designed to be beginner-friendly, enabling users to start fine-tuning with minimal setup. For advanced users,

it also offers a customization mode where training parameters can be adjusted for more control over the fine-tuning process.



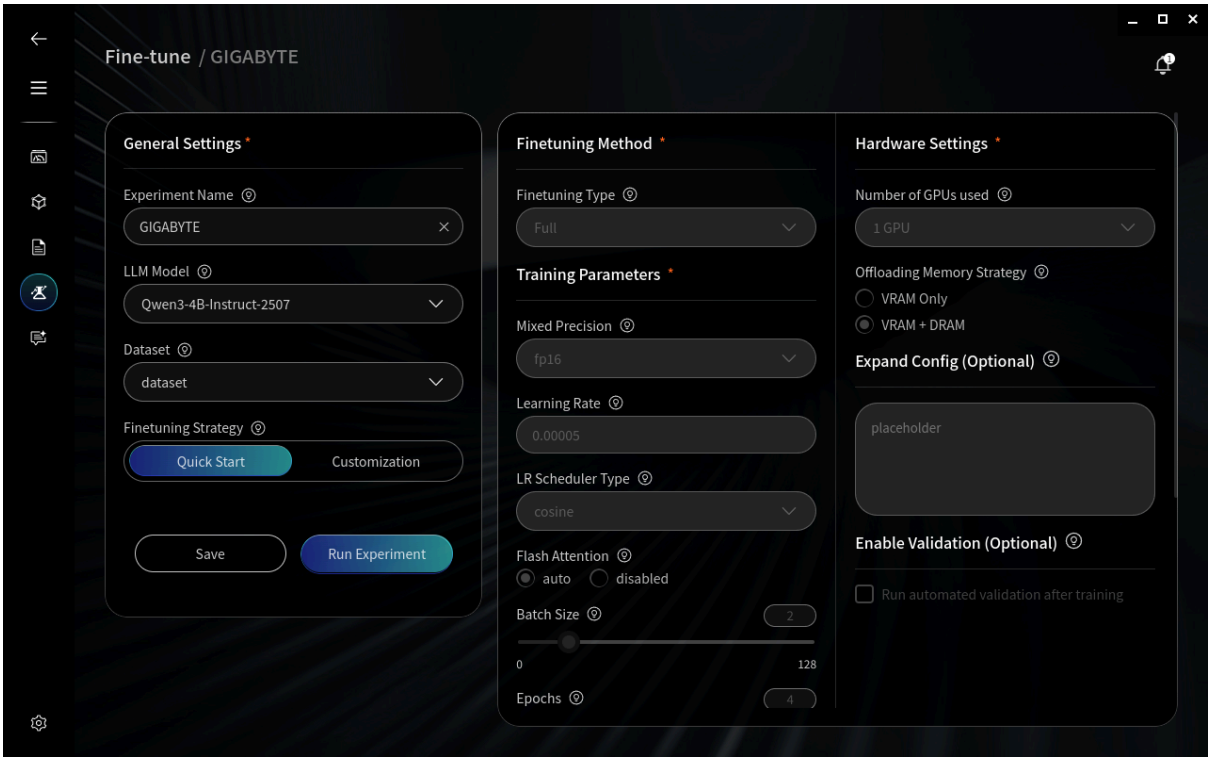
(1) Click **“Create Experiment”** (or **“New Experiment”**) and enter a name to create a new experiment.



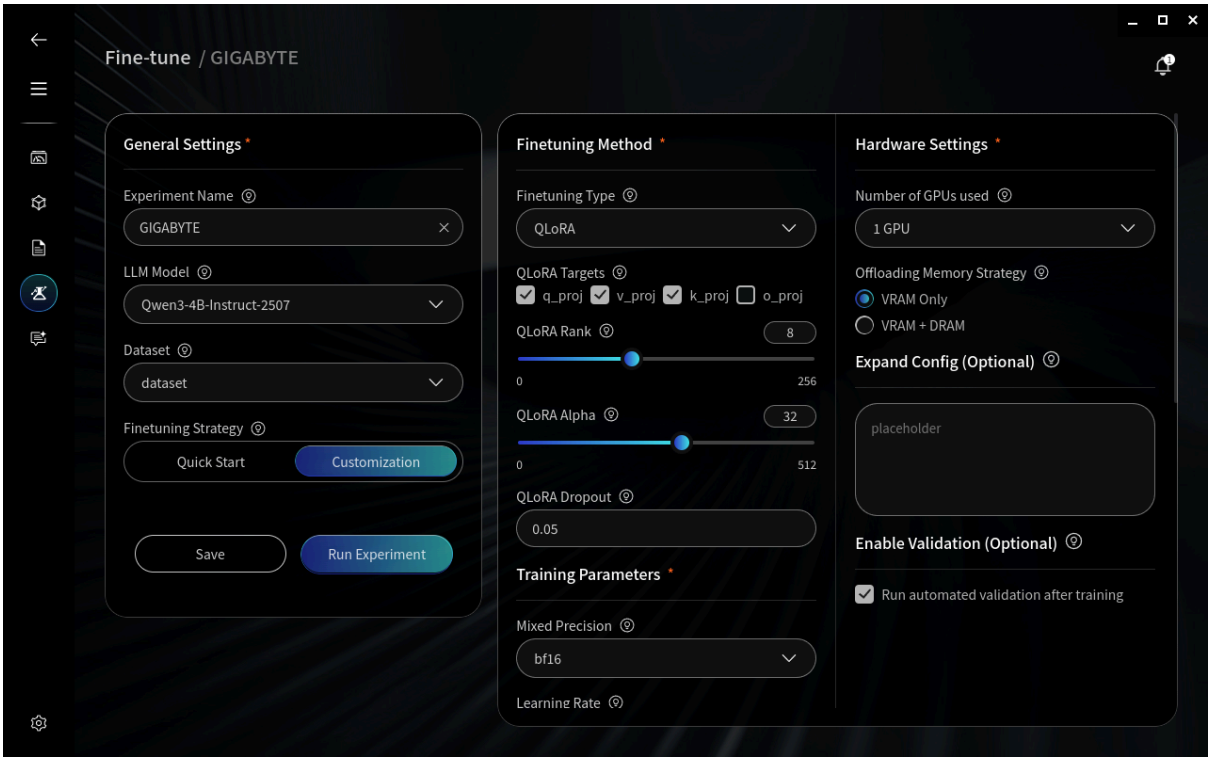
(2) The left panel contains the basic settings for fine-tuning, where users can select the LLM model, Dataset, and Finetuning strategy.

The fine-tuning strategy includes **Quick Start** and **Customization** modes.

- **Quick Start** uses predefined default parameters to help users begin training quickly with minimal setup.In this mode, the configuration fields in the right panel are locked and cannot be modified.
- **Customization** mode allows users to adjust training parameters for more control over the fine-tuning process.

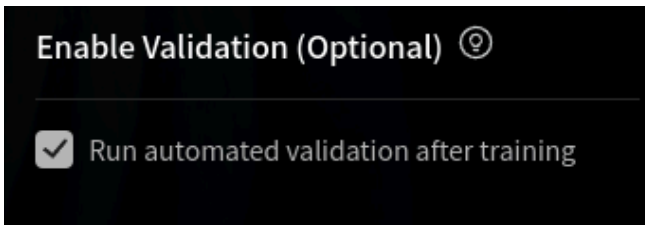


(3) The right panel displays the fine-tuning parameter settings. When **Customization** mode is selected, users can adjust the training parameters based on their requirements. Detailed parameter descriptions are provided in **Section 2-4 Fine-tune**.



- Expand Config
  Expand Config provides advanced customization options for fine-tuning. It is designed to allow users to fine-tune model behavior with additional parameters beyond the standard configuration, supporting more flexible and task-specific training setups.
  For detailed configuration syntax and supported parameters, please refer to **Section 3-7 Fine-tune Expand Config Syntax**.

- Enable Validation
  The **Validation** option allows users to automatically run validation after fine-tuning is completed. To enable automatic validation after the training process, select the checkbox.



***Note: Validation tasks do not support pause or resume functionality. If automatic validation is enabled, the validation process will begin immediately after fine-tuning completes and must run until completion.**

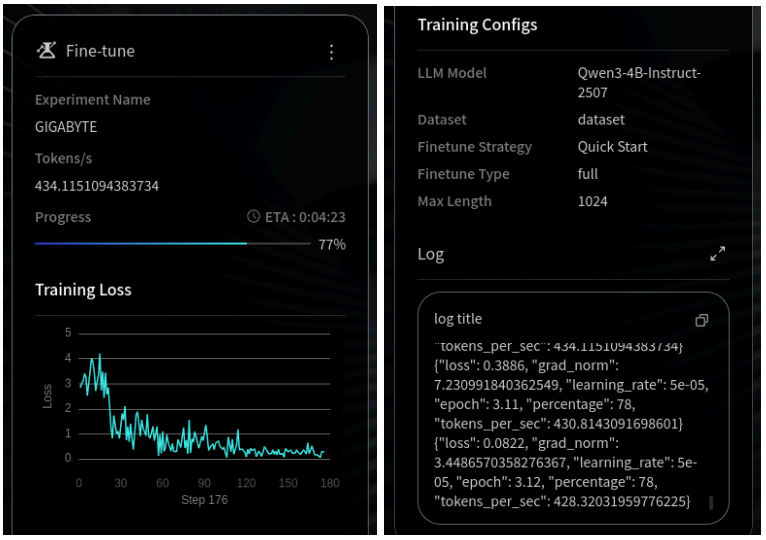
(4) After completing the parameter configuration, click "**Run Experiment"** to begin the training process.

(5) When fine-tuning is in progress, a new panel will appear on the left side of the Dashboard to display the current status and details of the training task.

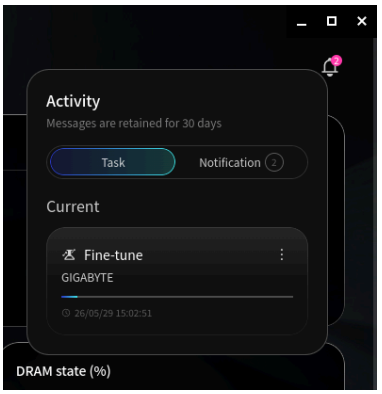


The right panel is dedicated to real-time hardware monitoring, providing system-level metrics during the fine-tuning process.

The left panel shows key fine-tuning information, including the fine-tuning name, real-time tokens per second (tokens/sec), estimated time of arrival (ETA), and a training progress bar. It also includes a training loss chart that is updated and plotted in real time, a section for important training configuration parameters, and a log view that updates continuously to reflect the current training status.

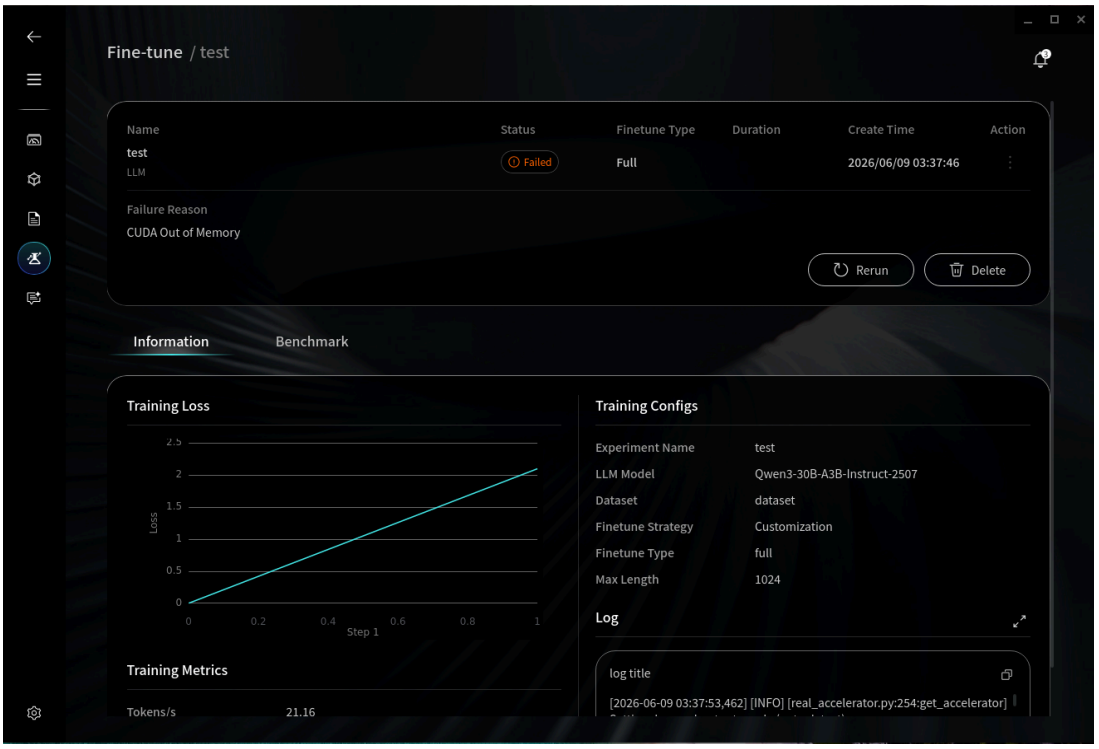


(6) Users can also monitor the progress in the Activity panel.



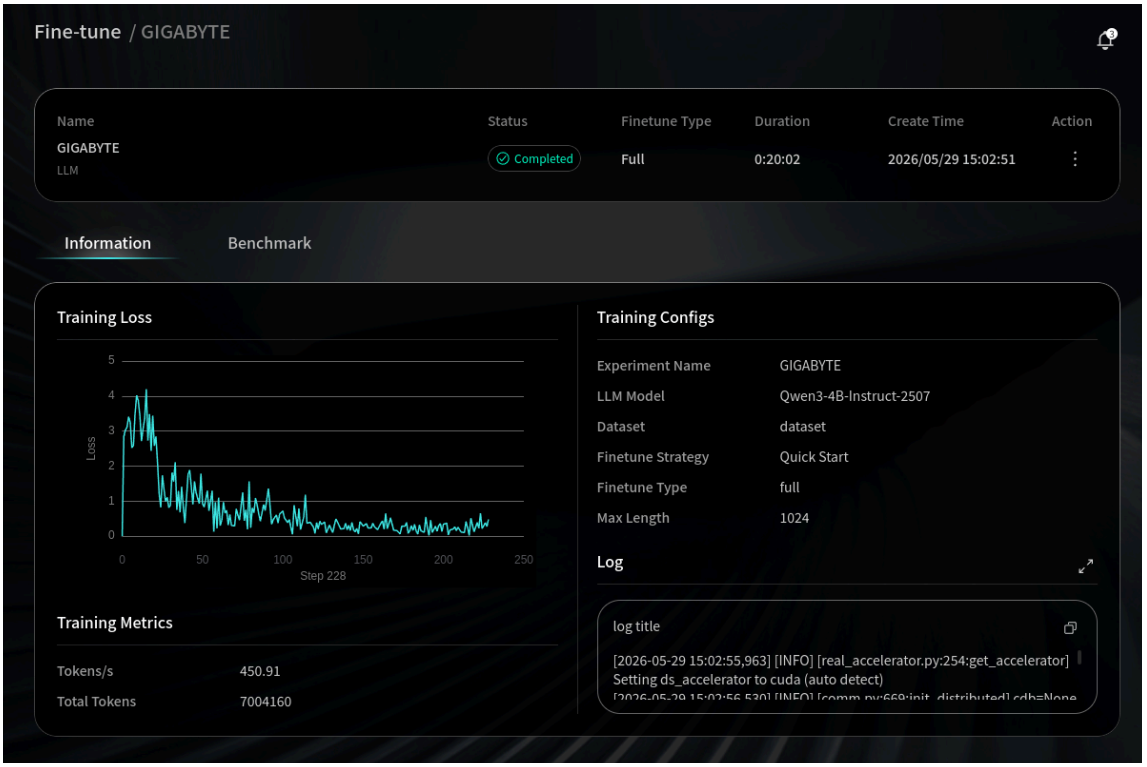
(7) If the fine-tuning fails, the failure reason will be displayed in this section. Users can review the error, click **“Rerun”** to return to the experiment settings page for

parameter adjustments, or click **“Delete”** to remove the experiment if necessary.



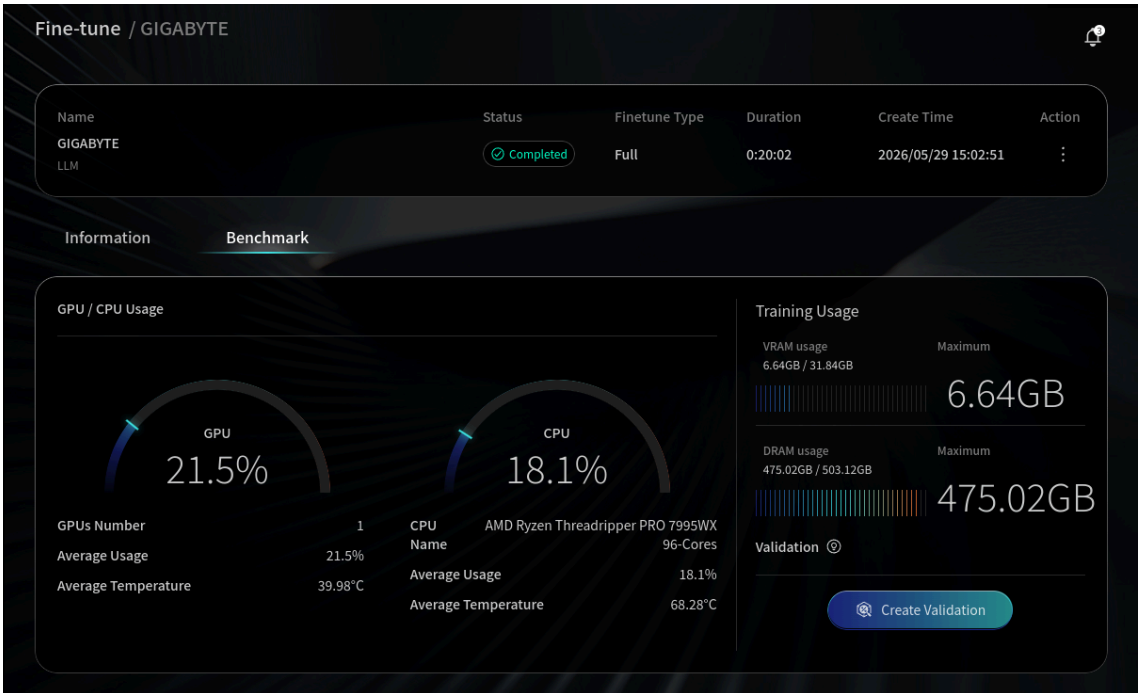
3-4-1 Information

The Information tab provides comprehensive details about the training process, including loss curves for monitoring model convergence, token statistics (average and total), key training configuration parameters, and complete training logs for troubleshooting and analysis.



3-4-2 Benchmark

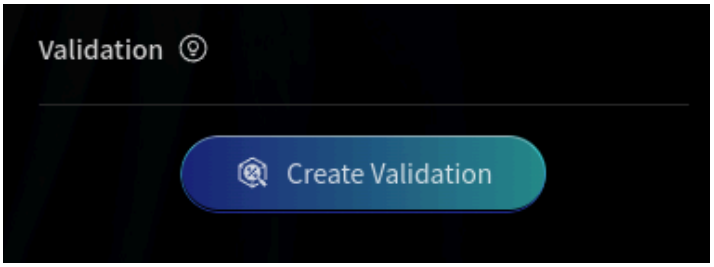
The Benchmark tab displays hardware performance metrics and validation results collected during the training process, helping users analyze system resource usage and model training performance.



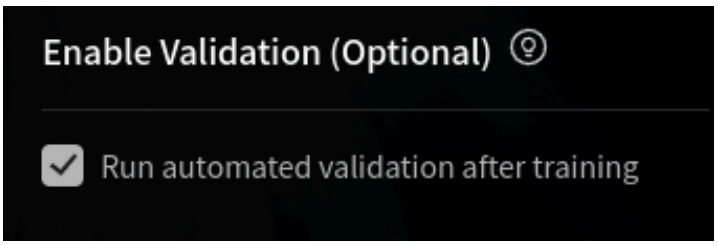
3-4-3 Validation

Validation runs an evaluation using the same dataset as training to measure model performance, providing metrics such as loss and accuracy for analysis and comparison.

(1) Click "**Create Validation**" to create and start a model validation task.

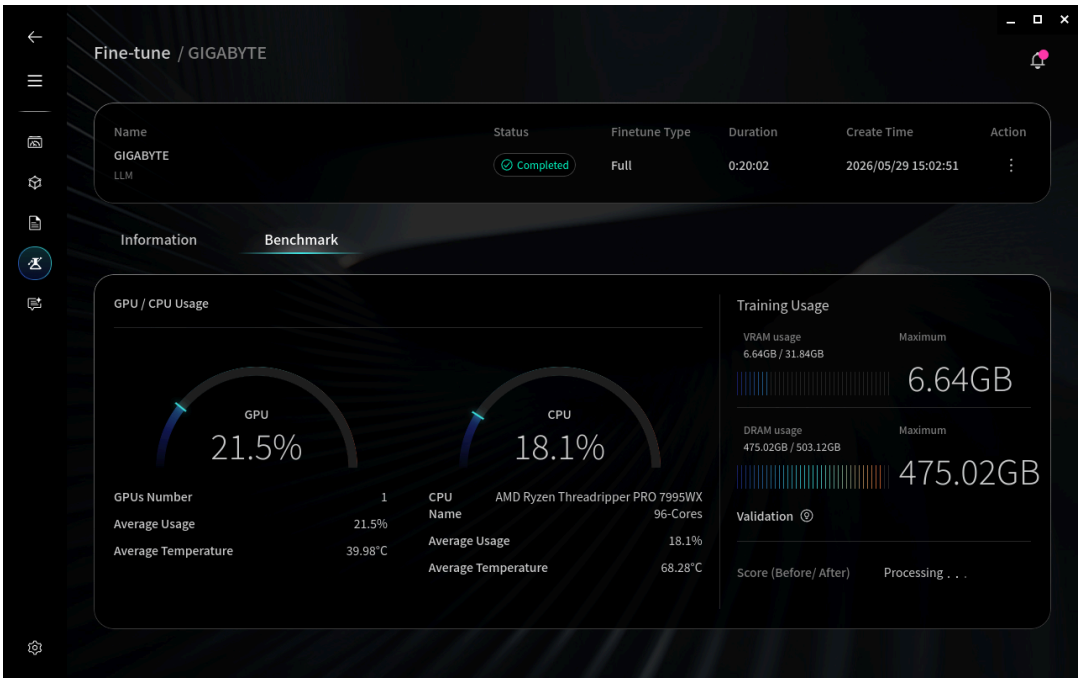


Alternatively, you can enable "**Run automated validation after training**" in the fine-tuning settings, which will automatically trigger validation after fine-tuning completes.



***Note: Validation tasks do not support pause or resume once started.**

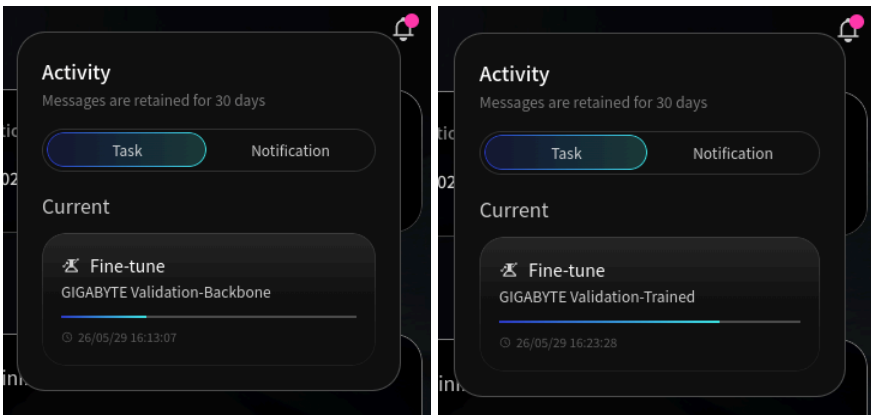
(2) The **"Processing…"** status indicates that validation is in progress.



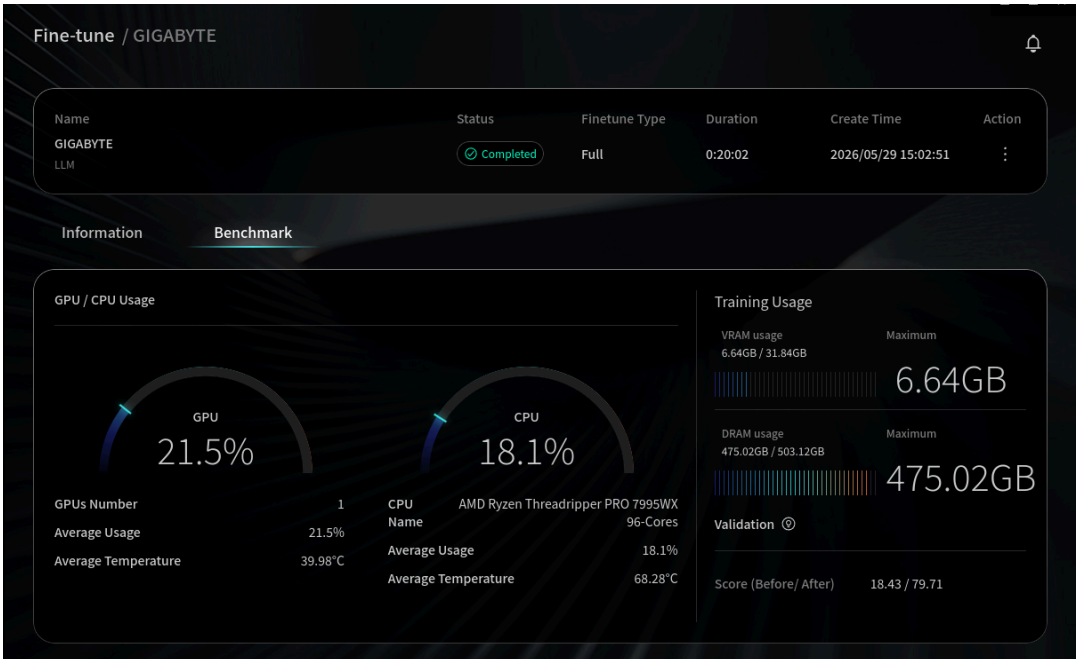
(3) The task progress can be monitored in the **Activity** panel.

Validation is divided into two stages.

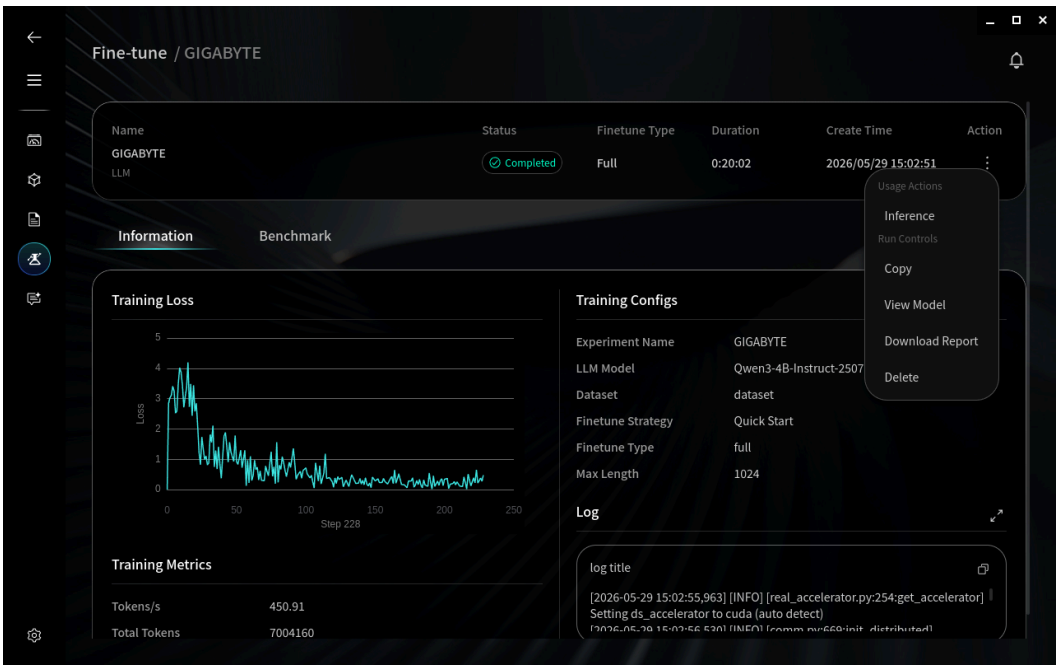
The first stage evaluates the backbone model before fine-tuning to establish a baseline performance **(Validation-Backbone)**. The second stage evaluates the fine-tuned model to measure improvements based on the training results **(Validation-Trained)**. This two-stage design allows users to compare model performance before and after fine-tuning.



(4) After validation, scores for **Before (Backbone Model)** and **After (Fine-tuned Model)** will be displayed for comparison.



3-3-4 Action



(1) Download report

This feature allows users to download a complete report of the current fine-tuning experiment. The report includes key data generated during the fine-tuning run, such as **Information (training details)**, **Benchmark (hardware and performance evaluation results)**, and **Validation (evaluation scores)**.


GIGABYTE AITOP UTILITY Finetuning Information & Benchmark Experiment Report

Training Report: GIGABYTE

Training Configuration

LLM Model	Qwen3-4B-Instruct-2507
Dataset	dataset
Finetune Strategy	Quick Start
Max Length	1024
Finetune Type	full

Performance Metrics

Tokens / Second (Average)	450.91 tokens/sec
Total Tokens (Estimated)	7,004,160 tokens

Training Loss Analysis

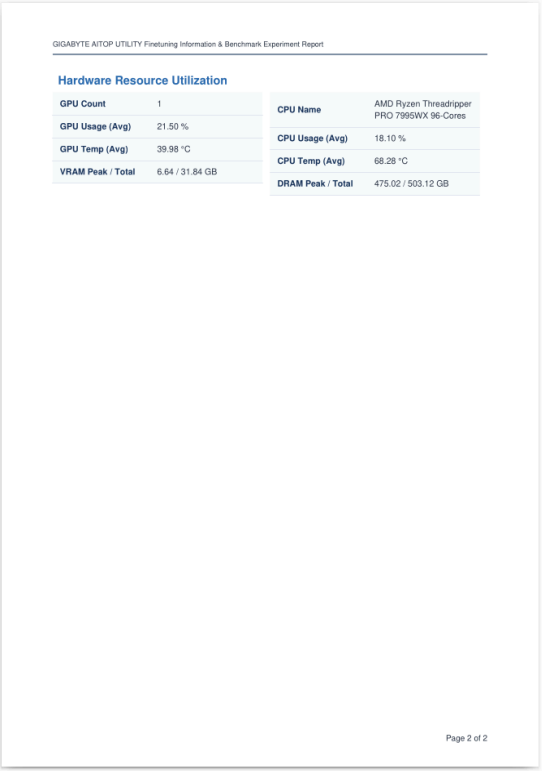
Training Loss Convergence

Original
Smoothed Trend
Loss
Training Step

Model Validation Results

Baseline Evaluation (Before)	18.43
Finetuned Evaluation (After)	79.71

Page 1 of 2


GIGABYTE AITOP UTILITY Finetuning Information & Benchmark Experiment Report

Hardware Resource Utilization

GPU Count	1
GPU Usage (Avg)	21.50 %
GPU Temp (Avg)	39.98 °C
VRAM Peak / Total	6.64 / 31.84 GB

CPU Name	AMD Ryzen Threadripper PRO 7995WX 96-Cores
CPU Usage (Avg)	18.10 %
CPU Temp (Avg)	68.28 °C
DRAM Peak / Total	475.02 / 503.12 GB

Page 2 of 2

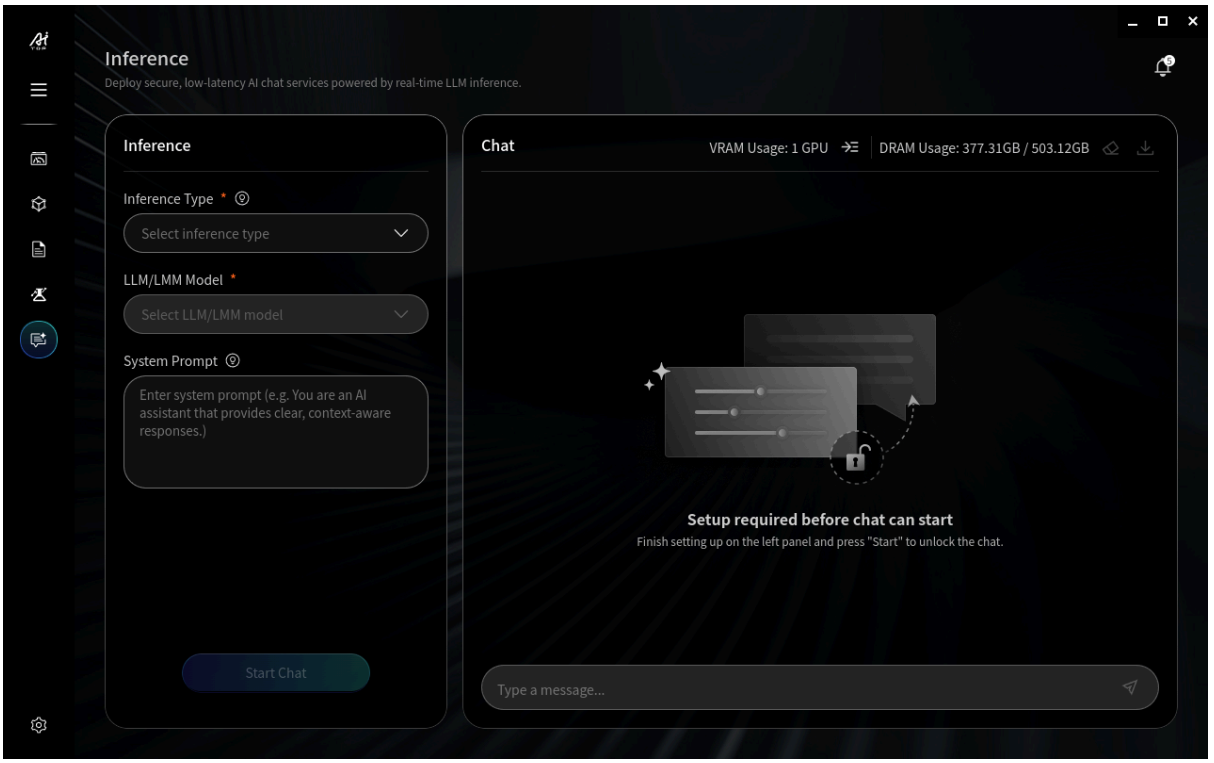
By downloading the report, users can easily review the full configuration and results of a single fine-tuning run, quickly understand the key changes during the training process, and compare performance differences across multiple fine-tuning runs. It also serves as a record of experiments for future analysis.

3-5. Inference

The built-in Inference tool supports Text-to-Text generation only. It allows users to run models in Safetensors or GGUF format to generate text outputs based on input prompts.

Users can use models downloaded from the Model Download tab, fine-tuned in the Experiment tab, or converted via the Model Convert.

***Note: LoRA / QLoRA models are not supported for inference in GGUF format.**



(1) Select Inference type and LLM model

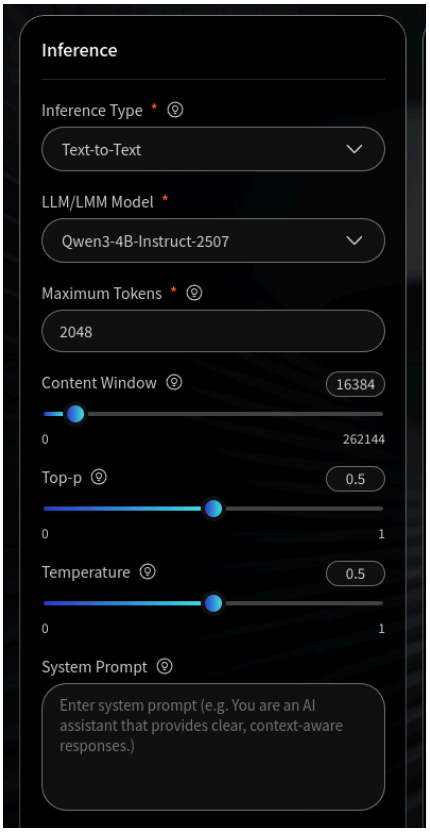
1.1 **Safetensors Format Model Settings** :

Set the following parameters according to your use case:

- **Context Window**: Controls how much input the model can reference during generation.

- **Top-p**:Controls how many candidate tokens are considered before choosing the next token.
- **Temperature**: Controls randomness of responses.
- **System Prompt**: Defines the model's role, behavior, and response style.

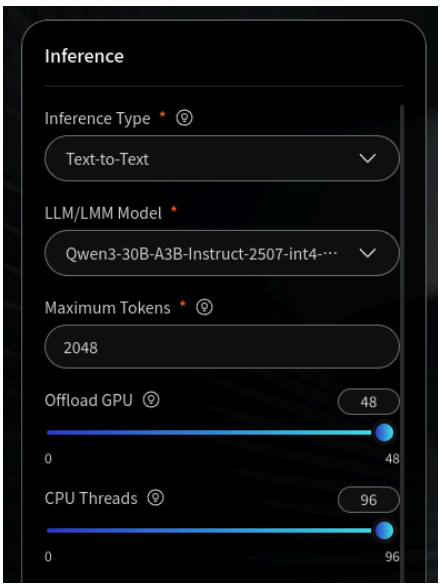
Adjust values based on desired output quality and behavior, then proceed to run the model.



1.2 **GGUF Format Model Settings**

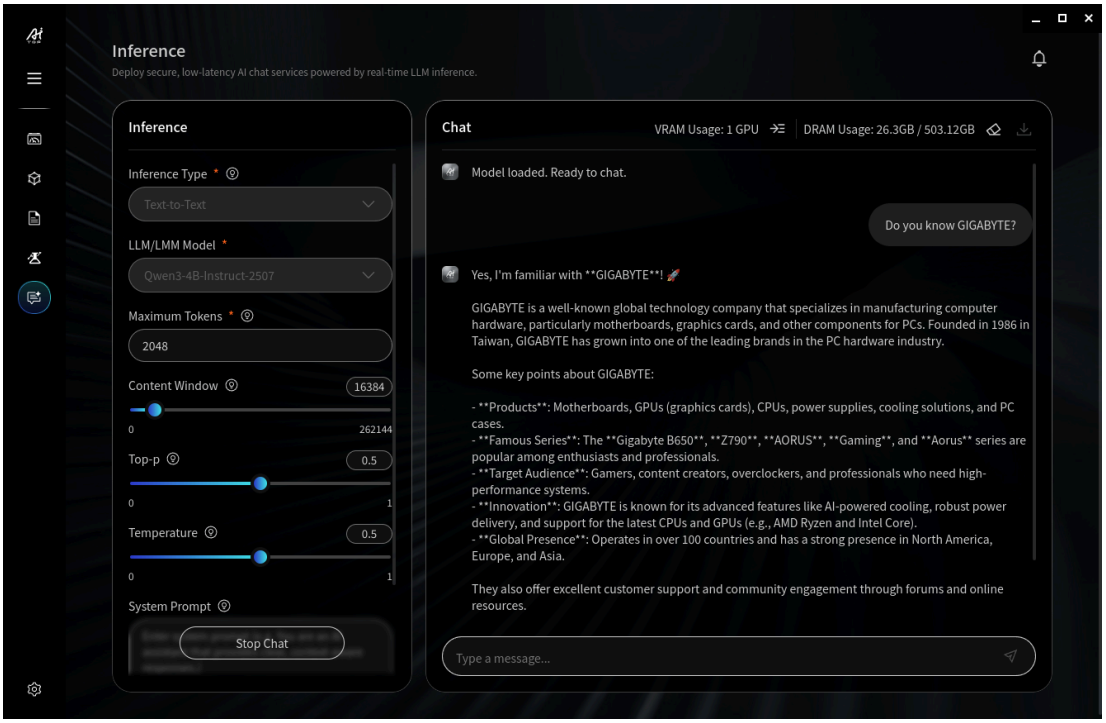
Configure the same settings as for a Safetensors format model, then adjust the additional GGUF-specific settings below.

- **GPU Offload Layers**: Specifies how many layers are loaded into VRAM (GPU) for acceleration.
- **CPU Threads**: Defines the number of CPU threads used to process the remaining layers in system memory (DRAM).



(2) Once the configuration is completed, click **"Start Chat"** to start a conversation with the selected model.

(3) Chat with the model by entering questions or prompts in the input box.

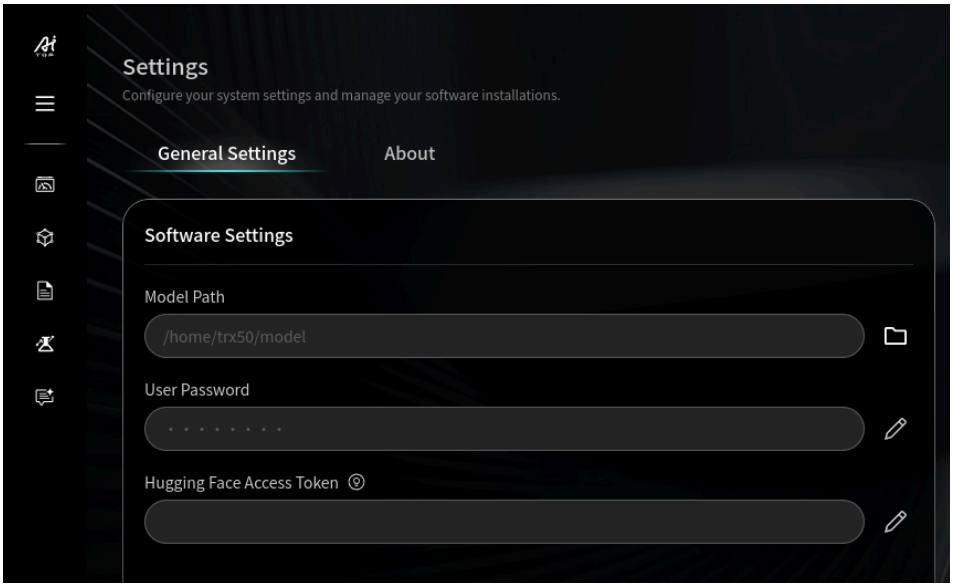


3-6 Settings

This page provides essential configuration for model storage, authentication, and external service access, as well as environment maintenance tools and license information to support system setup and troubleshooting.

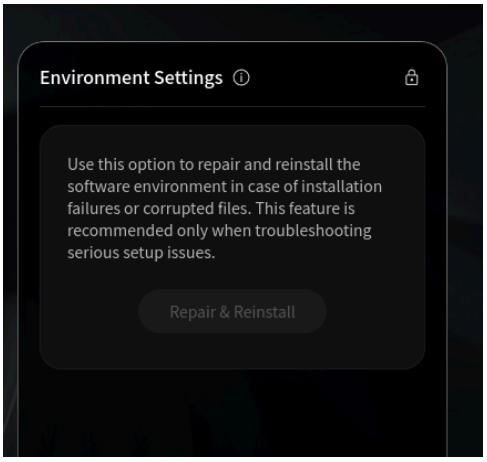
3-6-1 Software Settings

This section shows the model storage location, which can be clicked to open the corresponding folder on the local system. Users can also update their user password and Hugging Face access token here.

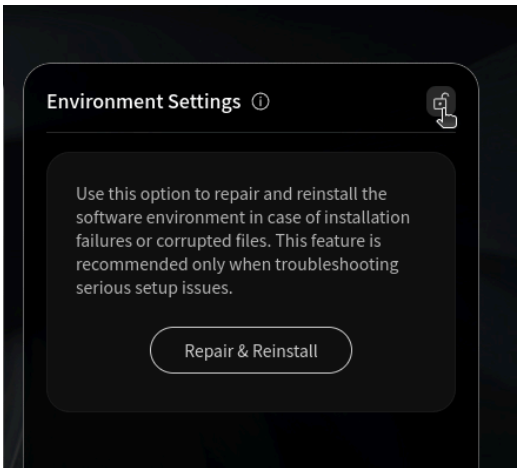


3-6-2 Repair & Reinstall

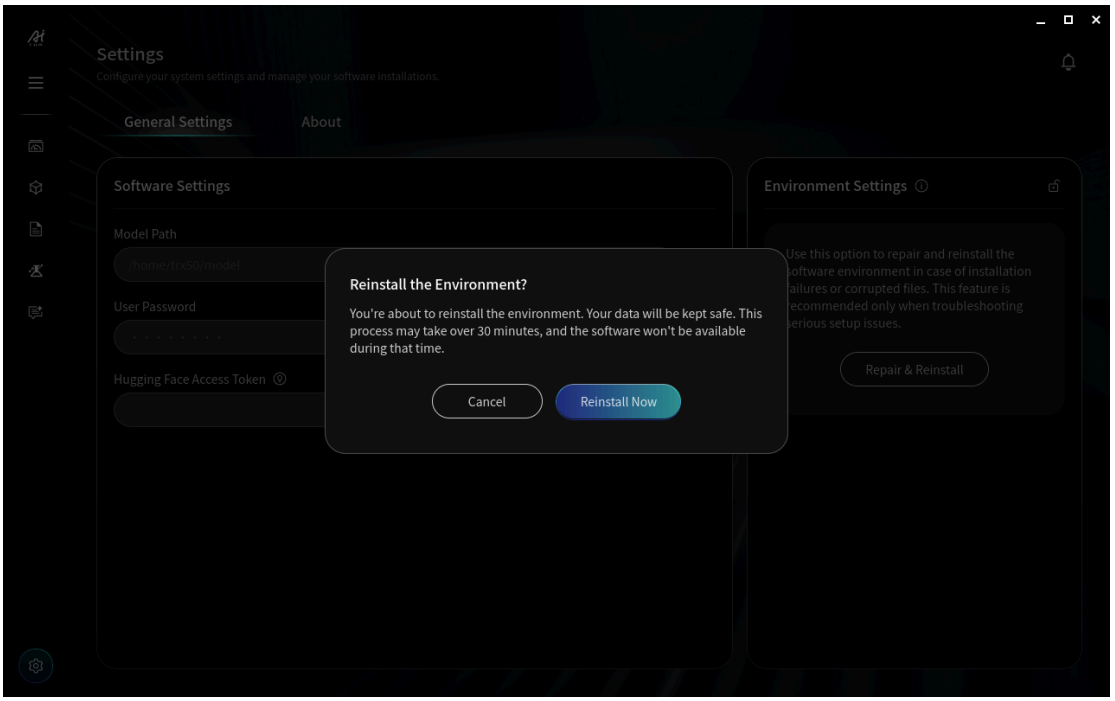
This option is used to repair and reinstall the software environment in case of installation failures or corrupted files. It is recommended to use this feature only when troubleshooting serious setup issues.



To proceed, click the **lock icon** in the top-right corner to unlock the settings. Once unlocked, the **Repair & Reinstall** button becomes available.



After clicking it, a confirmation prompt will appear. Select **“Reinstall Now”** to start the reinstallation process.



The software will then return to the initial download stage and reinstall the environment.

3-7 Finetune Expand config syntax

This section defines the officially supported configuration scope of the system. Using unlisted or additional parameters via **extra_config**, or setting unsupported or unreasonable values for existing parameters, may lead to training instability, errors, or unexpected behavior. Such usage may affect training performance and system stability and is not covered by the system's support or guarantee scope.

--max_length
    Maximum sequence length for training inputs (default range: 2048).
--num_train_epochs
    Total number of training epochs to perform. (default: 3)
--per_device_train_batch_size
    Batch size per GPU/TPU/MPS/NPU core/CPU for training. (default: 8)
--gradient_accumulation_steps
    Number of update steps to accumulate before performing a backward/update pass. (default: 1)
--preprocessing_num_workers
    The number of processes to use for the pre-processing. (default: 96)
--learning_rate
    The initial learning rate for AdamW. (default: 0.001)
--lr_scheduler_type {"cosine", "linear"}
    The scheduler type to use. (default: cosine)
--logging_steps
    Log every X update steps. Should be an integer or a float in range `[0,1)`.
    If smaller than 1,it will be interpreted as a ratio of total training steps. (default: 1)
--save_strategy {no , steps , epoch}
    The checkpoint save strategy to use. (default: steps)
    if you choose save strategy as steps, you also need to enter save_steps
--save_steps
    Save checkpoint every X update steps. Should be an integer or a float in range `[0,1)`.
    If smaller than 1, it will be interpreted as a ratio of total training steps. (default: 100)
--plot_loss {true , false}
    Whether or not to save the training loss curves. (default: True)
--do_train {true , false}
    Whether to execute the training process. (default: True)
--overwrite_cache {true , false}
    Whether to overwrite cached preprocessing data. (default: True)

4. Supported Models

Users need to download LLM backbone models from the Hugging Face website or another open-source platform before fine-tuning. We recommend that users conduct thorough research on how to fine-tune each LLM backbone model by setting training configurations and then try to fine-tune using AI TOP Utility.

Model	Model detail	Model Type	Remarks
all-mpnet-base-v2	all-mpnet-base-v2	Embedding model	Only supported for dataset generation.
Llama-3.2-1B-Instruct	Llama-3.2-1B-Instruct	LLM model	-
Llama-3.2-3B-Instruct	Llama-3.2-3B-Instruct	LLM model	-
Llama-3.1-8B-Instruct	Llama-3.1-8B-Instruct	LLM model	-
Qwen3-4B-Instruct-2507	Qwen3-4B-Instruct-2507	LLM model	-
Qwen3-30B-A3B-Instruct-2507	Qwen3-30B-A3B-Instruct-2507	LLM model	Recommended for dataset generation after model conversion.