

Pre-Workshop Setup Guide for AI For Science Workshop

Hanwen Wang, Shyam Sankaran

March 17, 2025

Welcome!

This workshop will provide an introduction to core concepts in Artificial Intelligence and Machine Learning, tailored for engineers from diverse fields who are interested in applying AI to their work. To make the most of our hands-on sessions, please follow this guide to set up the necessary software and accounts on your computer *before* the workshop.

This guide will walk you through installing the following tools:

- **Visual Studio Code (VS Code):** A powerful and user-friendly code editor that we will use for writing and running Python code.
- **Anaconda:** A platform for managing Python environments and installing necessary libraries for data science and machine learning.
- **Windows Subsystem for Linux (WSL):** Allows Windows users to run a Linux environment alongside Windows, which is often beneficial for AI development.
- **GitHub Copilot:** An AI-powered coding assistant that can help you write code more efficiently.
- **Weights & Biases (W&B):** A tool for tracking and visualizing machine learning experiments.
- **Hugging Face:** A platform for accessing and using pre-trained AI models and datasets.
- **(Alternative) Google Colab:** A cloud-based environment for running Python notebooks in your browser, offering an alternative if you prefer not to install software locally.

Please take the time to complete these steps before the workshop to ensure a smooth and productive learning experience. If you encounter any issues, please refer to the troubleshooting tips within each section or reach out to Hanwen Wang (wangh@sas.upenn.edu) or Shyam Sankaran (shyamss@seas.upenn.edu) for assistance.

1 Visual Studio Code (VS Code)

Visual Studio Code (VS Code) is a free, powerful, and widely used code editor. We will be using VS Code as our primary environment for writing and running Python code during the workshop. It provides a user-friendly interface, excellent Python support, and helpful features for coding and debugging.

1. Download and install VS Code from this [link](#). Choose the installer appropriate for your operating system (Windows, macOS, or Linux).
2. Follow the installation instructions provided on the download page.
3. Once installed, launch VS Code.

2 Windows Subsystem for Linux (WSL) (Windows Users)

Note for Windows Users: While optional, we strongly recommend installing the Windows Subsystem for Linux (WSL). Many AI and machine learning tools and libraries are originally developed for Linux environments and often work most seamlessly there. WSL allows you to run a Linux environment directly on your Windows machine without the need for virtual machines or dual booting. This provides a more efficient and integrated development experience, especially when working with command-line tools and remote servers, which are common in AI development be it across cloud services or larger supercomputers maintained by national labs. Even if you are not familiar with Linux, getting a basic setup with WSL will be beneficial for this workshop and future AI explorations.

2.1 WSL Installation

1. Follow the instructions from [the official WSL installation guide](#). You may need to restart your PC multiple times during this process.
2. By default, the Ubuntu distribution of Linux will be installed. This is a popular and user-friendly Linux distribution, and is a good choice for this workshop.
3. After the installation is complete and you restart your PC, WSL will launch and prompt you to create a user account for your Linux environment. Follow the on-screen instructions to create a username and password. These can be different from your Windows username and password.
4. Once you are logged into your WSL Ubuntu environment, you will be at the Linux command line (also known as the "terminal" or "bash"). To verify your installation, type the following command and press Enter:

```
echo "Hello World"
```

If the installation is successful, you should see "Hello World" printed on the next line.

2.2 Linking WSL to VS Code

To seamlessly work with your WSL Linux environment within VS Code, you need to install the "Remote - WSL" extension in VS Code. This allows you to open folders and run commands directly within your WSL Linux environment while using the familiar VS Code interface.

1. If you haven't already, install VS Code following the instructions in Section [2.2](#).
2. Open VS Code. On the left-hand side activity bar, click on the Extensions icon (it looks like four squares, one slightly detached). You can also use the keyboard shortcut **Ctrl+Shift+X** (or **Cmd+Shift+X** on macOS).
3. In the Extensions view search bar, type "Remote - WSL".
4. Find the "Remote - WSL" extension published by Microsoft and click on the "Install" button.
5. Once the extension is installed, you may need to restart VS Code. VS Code might prompt you to reload - click "Reload".
6. To connect to your WSL environment, you can either:
 - Open a folder in your WSL environment directly from VS Code: In VS Code, press **Ctrl+Shift+P** (or **Cmd+Shift+P** on macOS) to open the Command Palette. Type "Remote-WSL: New Window" and select this option. This will open a new VS Code window connected to your WSL environment. You can then open folders within your WSL file system (e.g., your Linux home directory) using "File" > "Open Folder".
 - Open an existing VS Code window in WSL: If you already have a VS Code window open, press **Ctrl+Shift+P** (or **Cmd+Shift+P** on macOS) and type "Remote-WSL: Reopen Folder in WSL" and select this option. This will reopen the current folder in your WSL environment.

Troubleshooting WSL and VS Code Remote

WSL Installation Issues: If you encounter errors during WSL installation, carefully review the [official WSL installation guide](#).

3 Anaconda

Anaconda is a free and open-source distribution of Python and R, specifically designed for data science and machine learning. It simplifies the process of installing Python, managing different project environments (to avoid conflicts between different project dependencies), and installing essential Python libraries (packages). We will use Anaconda to set up a dedicated environment for this workshop and install the necessary packages.

1. Follow [this page](#) to download and install Anaconda. Choose the Python 3 installer appropriate for your operating system (Windows, macOS, or Linux). If you are using WSL, make sure to install the Linux version of Anaconda *within* your WSL Ubuntu environment (not on Windows directly, if you intend to use WSL for the workshop).
2. Follow the installation instructions on the Anaconda website.
3. Once Anaconda is installed, open a terminal or command prompt.
 - **Windows (with WSL):** Open your WSL Ubuntu terminal.
 - **macOS/Linux:** Open your default terminal application.
4. Create a dedicated environment for this workshop. In the terminal, type the following command and press Enter:

```
conda create -n ai_workshop python=3.11
```

This command creates a new Anaconda environment named `ai_workshop` with Python version 3.11. You can choose a different environment name if you prefer.

5. Activate the newly created environment. In the terminal, type:

```
conda activate ai_workshop
```

When the environment is activated, you will see the environment name (e.g., `(ai_workshop)`) at the beginning of your terminal prompt. To deactivate the environment later, you can type `conda deactivate`.

6. Install the Python packages we will use in the workshop. With the `ai_workshop` environment activated, run the following commands one by one in your terminal:

```
pip install torch torchvision torchaudio
pip install wandb
pip install 'huggingface_hub[cli,torch]'
```

These commands install the following packages:

- **torch, torchvision, torchaudio:** PyTorch libraries for deep learning, including functionalities for neural networks, computer vision, and audio processing.
 - **wandb:** Weights & Biases library for tracking and visualizing machine learning experiments.
 - **huggingface_hub:** Hugging Face Hub library for accessing pre-trained models and datasets, along with optional dependencies for command-line interface (`cli`) and PyTorch (`torch`) integration.
7. In VS Code, we need to ensure that it uses the Python environment you just created. Open VS Code.

8. Install the recommended VS Code extensions for Python development. On the left-hand side activity bar, click on the Extensions icon. Search for and install the following extensions published by Microsoft:

- **Python**
- **Pylance**
- **Jupyter**

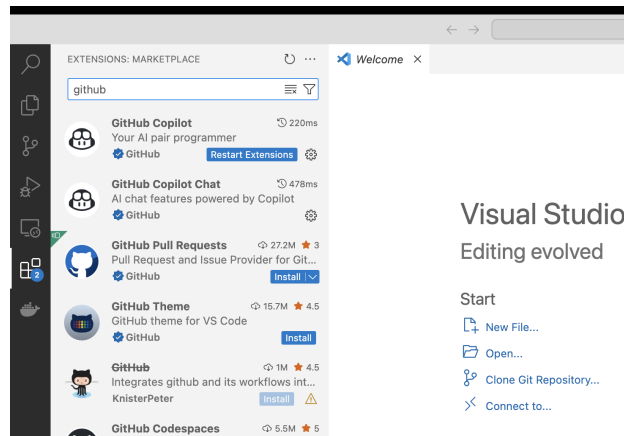


Figure 1: VS Code Extensions View. Search for and install the "Python", "Pylance", and "Jupyter" extensions published by Microsoft.

9. Open a folder in VS Code where you want to work on workshop materials (e.g., "File" > "Open Folder"). You can create a new folder for this purpose.

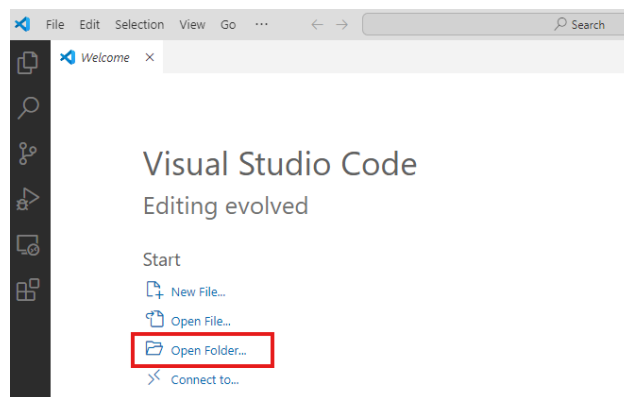


Figure 2: Open a folder in VS Code using "File" > "Open Folder".

10. Create a new Jupyter Notebook file within this folder. In the VS Code Explorer (usually on the left side), right-click in the folder area, select "New File", and name the file something like `hello.ipynb`. Make sure to include the `.ipynb` extension.

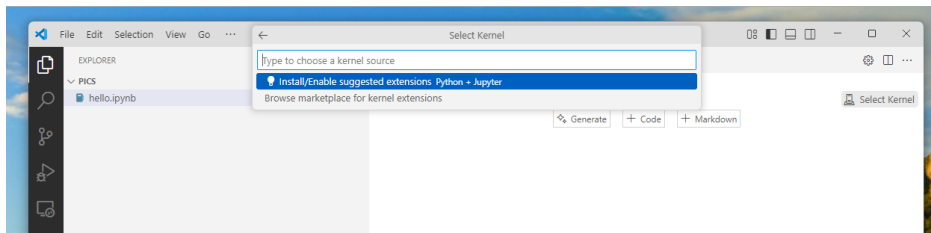
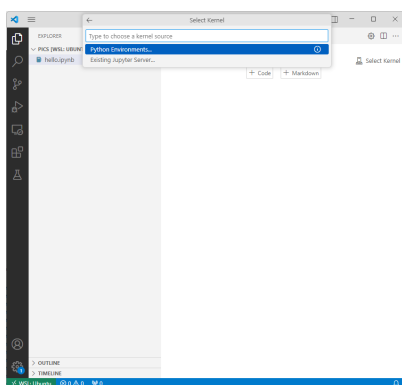
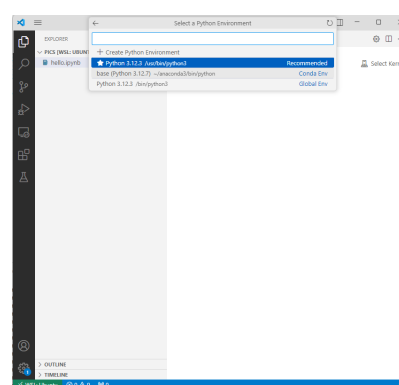


Figure 3: Creating a new Jupyter Notebook file named `hello.ipynb` in VS Code.

11. Open the `hello.ipynb` notebook. In the upper right corner of the notebook, you will see a "Select Kernel" button. Click on it.
12. VS Code will detect available Python environments. You should see your newly created Anaconda environment `ai_workshop` (or whatever name you chose) listed. Select the `ai_workshop` environment.



(a) Initial Kernel Selection.



(b) Selecting the `ai_workshop` environment.

Figure 4: Selecting the Python environment (Kernel) for your Jupyter Notebook in VS Code. Choose the `ai_workshop` environment you created with Anaconda.

Troubleshooting Anaconda and Python Environment Setup

‘conda’ command not found: If you get an error like “‘conda’ is not recognized...” in your terminal, it means Anaconda is not correctly added to your system’s PATH environment variable. You may need to manually add Anaconda to your PATH. Refer to the Anaconda installation documentation for instructions specific to your operating system.

4 GitHub Copilot

GitHub Copilot is an AI-powered code completion tool that can assist you in writing code more quickly and efficiently. It is available as a VS Code extension. Students with a valid `.edu` email address can apply for a free GitHub Education account, which often includes GitHub Copilot.

1. If you don’t already have one, create a GitHub account at github.com.
2. If you are a student with a `.edu` email address, you can apply for GitHub Education at [here](https://education.github.com). Follow the instructions on the GitHub Education website to verify your student status. It may take some time for your application to be processed.

3. Once you have a GitHub account and (optionally) GitHub Education, open VS Code.
4. In the Extensions view (see Figure 1), search for “GitHub Copilot”.
5. Find the “GitHub Copilot” extension and click “Install”.
6. After installation, you may need to sign in to GitHub within VS Code to activate Copilot. VS Code will likely prompt you to sign in. Follow the on-screen instructions to link your GitHub account. You can also find instructions for linking your account [here](#).

Troubleshooting GitHub Copilot

GitHub Education Verification Delays: It may take some time for GitHub to verify your student status for GitHub Education. Copilot might not be available immediately after applying.

Compatibility Issues: Ensure you are using a recent version of VS Code.

5 Weights & Biases (W&B)

Weights & Biases (W&B, often shortened to **wandb**) is a powerful platform for tracking, visualizing, and collaborating on machine learning experiments. It helps you monitor metrics like loss and accuracy during model training, compare different experiments, and gain insights into your model’s performance.

1. Sign up for a free account at wandb.com.
2. After signing up and logging in, navigate to your user settings or profile page on the W&B website. Look for a section related to “API Keys” or “Authorization”.
3. Create or find your API key. Copy this API key. You will need it if you want to use W&B to track your experiments during the workshop. Keep this API key in a safe place or easily accessible for the workshop.

6 Hugging Face

Hugging Face is a central hub for the AI community, particularly for Natural Language Processing (NLP) and related fields. It hosts a vast collection of pre-trained AI models, datasets, and tools. Hugging Face makes it easier to access and utilize state-of-the-art AI models without needing to train them from scratch.

1. Create a free account at huggingface.co.
2. After signing up and logging in, you can explore the Hugging Face Hub to discover models and datasets.
3. For this workshop, you may need to log in to Hugging Face from your terminal if you intend to download models using code. In your terminal (with your `ai_workshop` environment activated), run the command:

```
huggingface-cli login
```

Follow the prompts to log in using your Hugging Face account credentials. This will store an authentication token on your computer, allowing you to access Hugging Face resources.

7 Google Colab

Google Colaboratory (Colab) is a free, cloud-based Jupyter Notebook environment that runs entirely in your web browser. It provides access to computational resources, including GPUs and TPUs, making it suitable for machine learning tasks. Colab is a great alternative if you prefer not to install any software locally or if you encounter difficulties with local installations. Using Google Colab is an *alternative option* to the local setup described above.

7.1 Getting Started with Google Colab

1. Open your web browser and go to colab.research.google.com.
2. Sign in with your Google account (preferably a personal account).
3. To create a new notebook, go to "File" > "New Notebook". This will open a new Jupyter Notebook in your browser.
4. To use a GPU for faster computation (if needed for workshop exercises), go to "Runtime" > "Change runtime type" > "Hardware accelerator" and select "GPU". Click "Save".

When using Google Colab, you will be working in a cloud environment. You will need to upload any data files required for the workshop to your Colab environment and download any results you generate.

Before the Workshop Checklist

Please ensure you have completed the following steps *before* the workshop begins:

- ☐ (Recommended for Windows Users) Installed Windows Subsystem for Linux (WSL).
- ☐ Installed Visual Studio Code.
- ☐ Created a Github account and applied for Github Education and enabled Github Copilot.
- ☐ Installed Anaconda and created the `ai_workshop` environment with Python 3.11.
- ☐ Installed the required Python packages in the `ai_workshop` environment: `torch torchvision torchaudio wandb huggingface_hub[cli,torch]`.
- ☐ Created accounts on Weights & Biases (wandb.com) and Hugging Face (huggingface.co). Located and noted down your Weights & Biases API key (if applicable).
- ☐ (Alternative Option) Familiarized yourself with Google Colab (colab.research.google.com) and created a test notebook.