## Part 1: Local Environment Setup

All the commands in the following process are executed in the Visual Studio Code Terminal.

Python version 3.12

### 1. Cloning the Repository

git clone https://github.com/openai/point-e.git

cd point-e

### 2. Setting Up Virtual Environment

Using a virtual environment to manage dependencies:

python -m venv venv

.\venv\Scripts\activate # On Windows

### 3. Installing Dependencies

Installed required dependencies along with Open3D:

python.exe -m pip install --upgrade pip

python -m pip install --upgrade pip setuptools wheel

pip install -e .

pip install open3d==0.19.0

### 4. Verifying Installation

To confirm a successful installation:

python -c "import point\_e; print('Point-E installed successfully')"

### 5. Challenges Encountered & Resolutions

**Version Conflicts:**

Faced issues with **torch** and **numpy** version compatibility.

Resolved by pinning compatible versions:

pip install numpy==1.26.0

python -m pip install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cpu

**6. Visualization Tools:**

For viewing .ply files and 3D point clouds:

pip install trimesh

### ****7. JupyterLab &**** ipykernel ****Installation:****

Installed to provide an interactive development environment:

pip install jupyterlab

pip install notebook ipykernel

Build and launch JupyterLab by running below command

jupyter lab build

    jupyter lab

## Part 2: Running a Minimal Example

### 1. Executing Text-to-Point Cloud Generation

* Ran **text2pointcloud**.ipynb using JupyterLab with the prompt:

prompt = 'a red motorcycle'

* Successfully generated a 3D point cloud of a red motorcycle.

#### Observations:

* **Runtime:** ~28 minutes for 130 iterations on CPU.
* **GPU Usage:** Unavailable due to driver issues.
* **Output:** Displayed multiple 3D views of the generated point cloud.

### 2. Executing Image-to-Point Cloud Generation

* Ran **image2pointcloud**.ipynb using JupyterLab with the input image:  
  example\_data/cube\_stack.jpg
* Successfully generated a 3D point cloud of stacked cubes(a red cube on top of a blue cube).

**Observations:**

* **Runtime:** ~48 minutes for 130 iterations on CPU.
* **GPU Usage:** Unavailable due to driver issues.
* **Output:** Displayed multiple 3D views of the generated point cloud, accurately reflecting the spatial arrangement of the cubes.

## 3. Issues Encountered

* **NVIDIA GPU Driver Issue:**

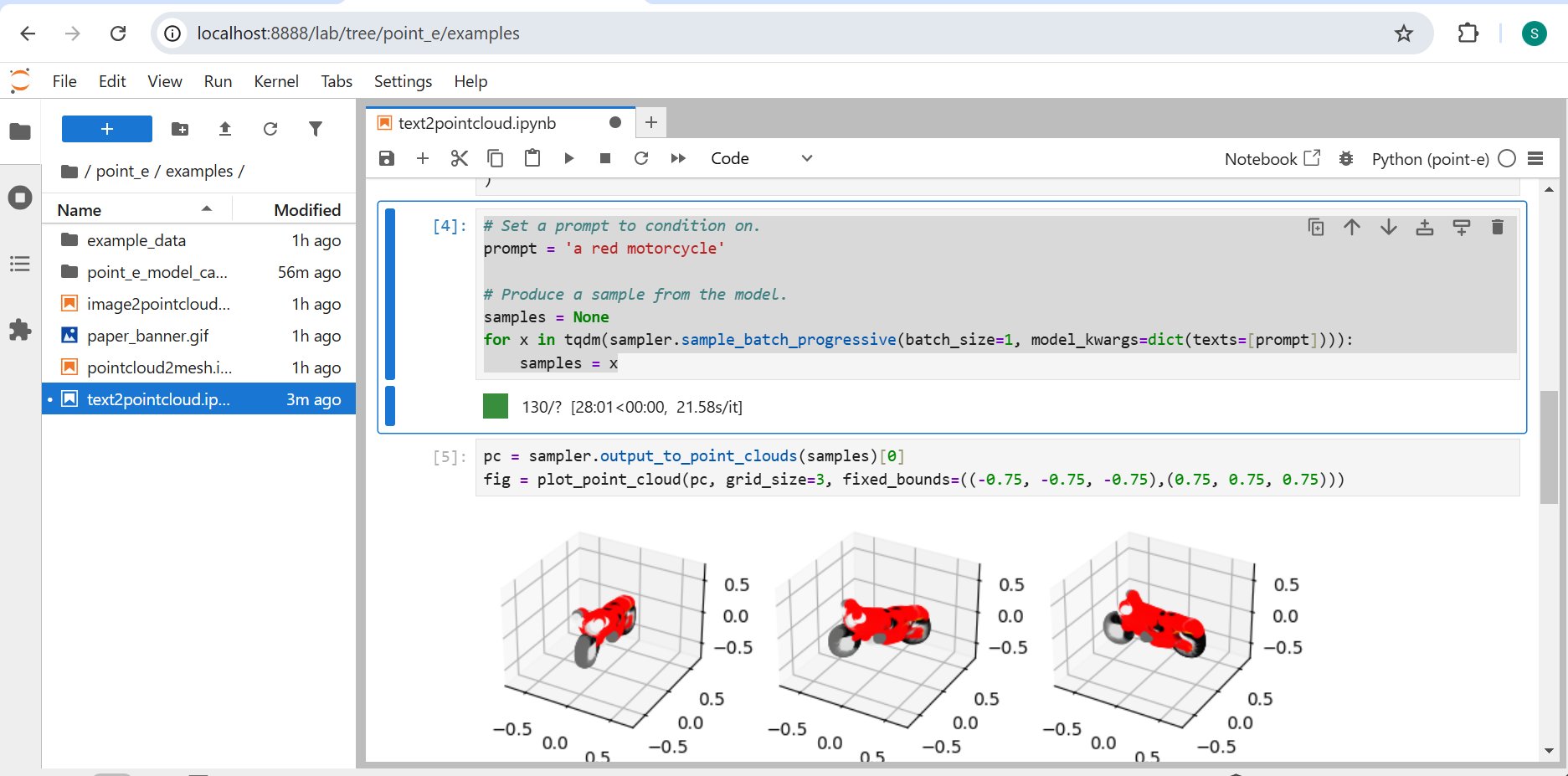
No NVIDIA GPU is detected on your system.

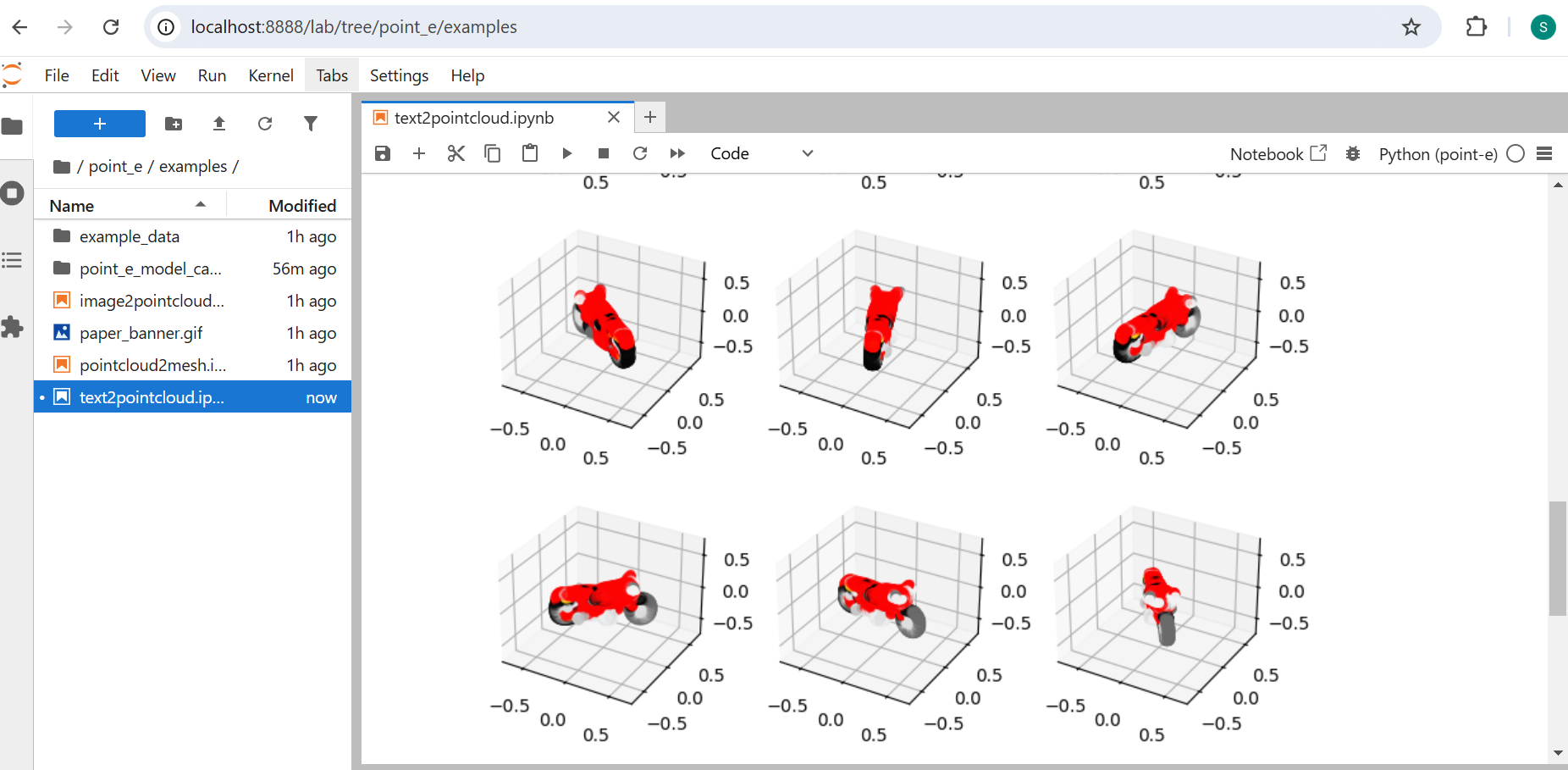
This graphics driver could not find compatible graphics hardware.

* **Root Cause:** No compatible NVIDIA GPU on the system.
* **Resolution:** Ran the example on CPU instead.
* **Next Steps:** Explore alternative drivers for GPU acceleration.

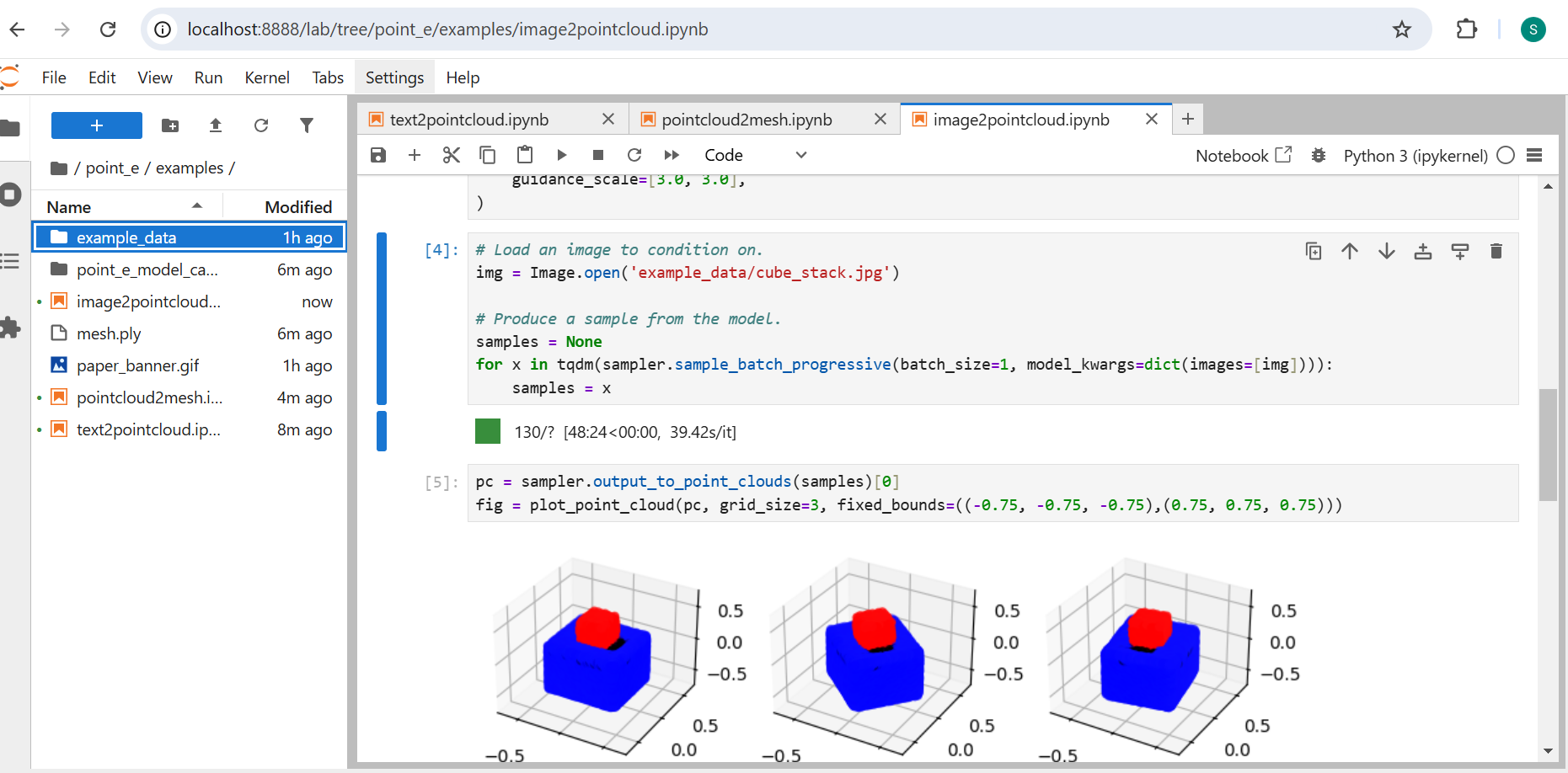
## 4. Proof of Successful Run

* **Output Screenshots:** Text-to-Point Cloud Generation





* **Output Screenshots:** Image-to-Point Cloud Generation



## 

## 

## Notes and Observations

* Running on **CPU** significantly increased runtime.
* Model output was visually accurate and matched the input prompt.
* **Next Steps:** Resolve GPU driver issue for improved performance.