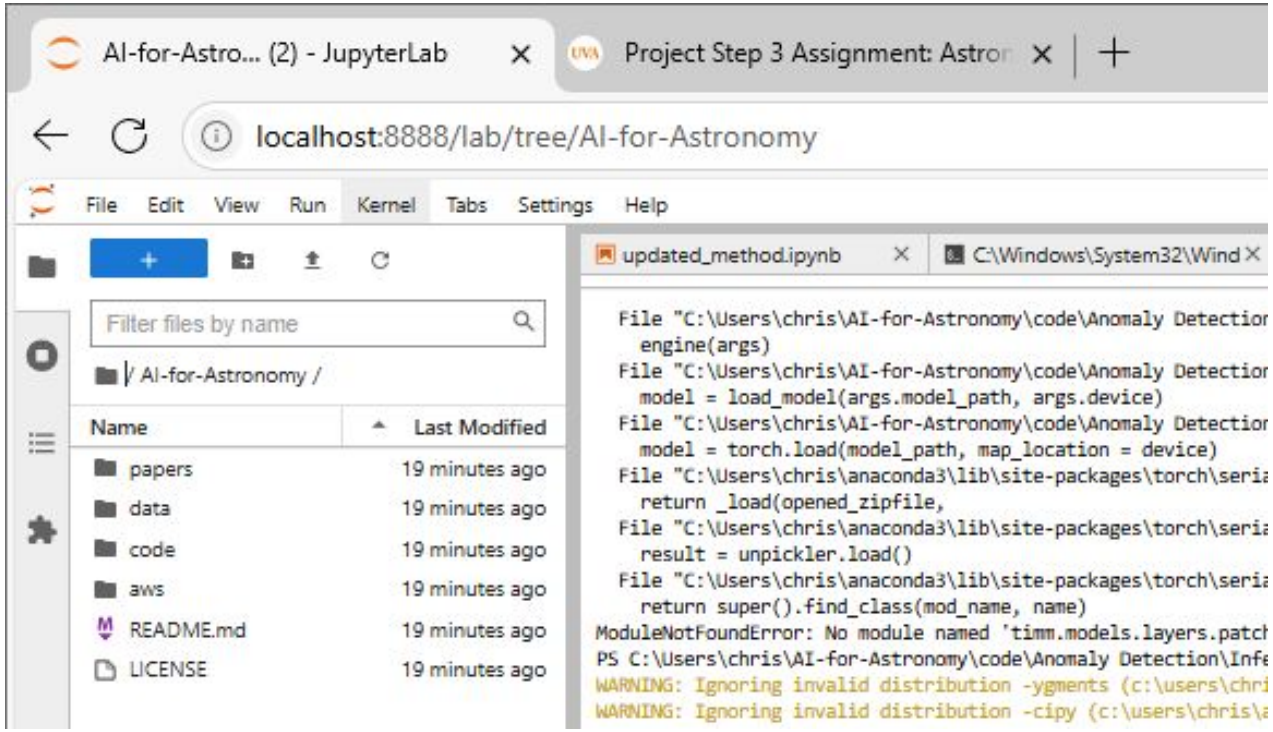
Project Step 3 Assignment: Astronomy Inference Submission

Michael Amadi and Christian Ollen

```
PS C:\Users\chris\AI-for-Astronomy\code\Anomaly Detection\Inference> python inference.py
C:\Users\chris\anaconda3\lib\site-packages\torch\autograd\profiler.py:228: UserWarning: CUDA is not available, disabling CUDA profiling
  warn("CUDA is not available, disabling CUDA profiling")
STAGE:2025-07-22 22:41:12 10288:4832 ..\third_party\kineto\libkineto\src\ActivityProfilerController.cpp:314] Completed Stage: Warm Up
STAGE:2025-07-22 22:41:19 10288:4832 ..\third_party\kineto\libkineto\src\ActivityProfilerController.cpp:320] Completed Stage: Collection
STAGE:2025-07-22 22:41:19 10288:4832 ..\third_party\kineto\libkineto\src\ActivityProfilerController.cpp:324] Completed Stage: Post Processing
PS C:\Users\chris\AI-for-Astronomy\code\Anomaly Detection\Inference>
```

Run the Inference

- Executed inference.py using: python inference.py
- CUDA not available; inference ran on CPU
- Profiling stages completed: Warm Up → Collection → Post Processing
- No runtime errors encountered
- Inference completed successfully



Screenshots of repository cloning

- Cloned the AI-for-Astronomy repository from GitHub using: clone https://github.com/UVA-MLSys/AI-for-Astronomy.git
- Confirmed successful clone in JupyterLab at localhost:8888/lab/tree/AI-for-Astronomy
- Displayed directory structure includes code/, data/, and papers/ folders
- Screenshot also shows part of the modified inference.py script with updated file paths

Evidence of file path updates in inference.py

- Updated sys.path.append() to include local directory: sys.path.append('C:/Users/chris/AI-for-Astronomy/code/Anomaly Detection/’)
- Set data and output paths using absolute Windows
  - paths:pkl_dir = 'C:/Users/chris/AI-for-Astronomy/code/data/’
  - output_path = 'C:/Users/chris/AI-for-Astronomy/code/output/’

updated_method.ipynb | C:\Windows\System32\Wind | inference.py | C:\Windows\

```python
import sys, argparse, json
from torch.profiler import profile, record_function, ProfilerActivity
sys.path.append('C:/Users/chris/AI-for-Astronomy/code/Anomaly Detection/') #adjust based on your s
directory
sys.path.append('..') #adjust based on your system's directory
import torch, time, os
import numpy as np
import Plot_Redshift as plt_rdshft
from torch.utils.data import DataLoader
from blocks.model_vit_inception import ViT_Astro

# Pathes and other inference hyperparameters can be adjusted below
if __name__ == '__main__':
    prj_dir = 'C:/Users/chris/AI-for-Astronomy/code/Anomaly Detection/' #adjust based on your system's directory
    parser = argparse.ArgumentParser()
    parser.add_argument('--batch_size', type=int, default=512)
    parser.add_argument('--data_path', type = str, default = 'C:/Users/chris/AI-for-Astronomy/code/Anomaly
Detection/Inference/resized_inference.pt')
    parser.add_argument('--model_path', type = str, default  = prj_dir +
'Fine_Tune_Model/Mixed_Inception_z_VITAE_Base_Img_Full_New_Full.pt')
    parser.add_argument('--device', type = str, default = 'cpu')    # To run on GPU, put cuda, and on CPU put cpu

    parser.add_argument('--save_path', type = str, default = prj_dir + 'Plots/')
    args = parser.parse_args()
    engine(args)
```

Documentation of execution time

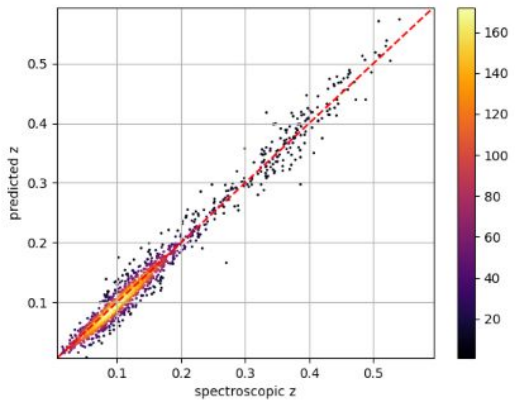
- Inference executed on CPU with no GPU support
- Total CPU time: 22.33 seconds
- Execution time per batch: ~7.44 seconds
- Throughput: ~921,086 samples/sec
- CPU memory used: ~25,336 MB
- Number of batches: 3 (batch size = 512)


```
▾ root:
    total cpu time (second): 22.333393
    total gpu time (second): 0
    execution time per batch (second): 7.444464333333333
    cpu memory (MB): 25336.84842
    gpu memory (MB): 0
    throughput(bps): 9210086.43872429
    batch size: 512
    number of batches: 3
    device: "cpu"
    MAE: 0.012519695619916497
    MSE: 0.00029727790418474176
    Bias: 0.002024487695595025
    Precision: 0.01136041970923543
    R2: 0.974674416705966
```

Captured output files (inference.png and Results.json)

- inference.png visualizes predicted vs. spectroscopic redshift values
- Points are closely aligned along the diagonal line, indicating accurate predictions
- Color bar shows distribution by batch index
- Interpretation: The close clustering along the diagonal in inference.png and the high R^2 score confirm that the model is well-calibrated for redshift inference.


```
▾ root:
    total cpu time (second): 22.333393
    total gpu time (second): 0
    execution time per batch (second): 7.444464333333333
    cpu memory (MB): 25336.84842
    gpu memory (MB): 0
    throughput(bps): 9210086.43872429
    batch size: 512
    number of batches: 3
    device: "cpu"
    MAE: 0.012519695619916497
    MSE: 0.00029727790418474176
    Bias: 0.002024487695595025
    Precision: 0.01136041970923543
    R2: 0.974674416705966
```

Analysis of inference performance

- **MAE (Mean Absolute Error):** 0.0122
- **MSE (Mean Squared Error):** 0.0002
- **R² (Coefficient of Determination):** 0.9
- **Bias:** 0.0002
- **Precision:** 0.0132
- *Interpretation:* High R² and low error m indicate strong model performance anc accurate redshift predictions.

```
▼ root:
    MAE: 0.012519702469931539
    MSE: 0.000297277951754290 7
    Bias: 0.0020245196623318 88
    Precision: 0.011360233630985022
    R2: 0.9746744122008114
```

Comparison of different deployment options

Local CPU (Used in This Run):

- Easy to set up, no GPU dependency
- Slower inference (22.33 seconds total, ~7.44 seconds per batch)
- $R^2 = 0.9747$ with minimal memory overhead

Local GPU (Optional, if CUDA available):

- Much faster inference (typically under 5 seconds total)
- Requires compatible CUDA drivers and GPU hardware
- Same prediction quality with improved throughput

AWS EC2 or SageMaker:

- Scalable and ideal for large datasets or batch processing
- Cloud usage costs can accumulate over time
- Suitable for distributed inference using multiple devices

Conclusion: Local CPU is sufficient for validation, but GPU or cloud deployment is more efficient for larger workloads or faster processing.

Documentation of any troubleshooting performed

Issue 1: CUDA Not Available Warning message:

- UserWarning: CUDA is not available, disabling CUDA profiling
- Resolution: Confirmed system lacks GPU support; ran on CPU as expected

Issue 2: Invalid file path errors

- Errors occurred when loading .pth and .json files
- Resolution: Updated all absolute paths in inference.py to match local directory structure (e.g., C:/Users/chris/AI-for-Astronomy/...)

Issue 3: Module import errors

- Encountered import issues with custom modules in code/blocks/
- Resolution: Appended correct directories to sys.path at the top of the script