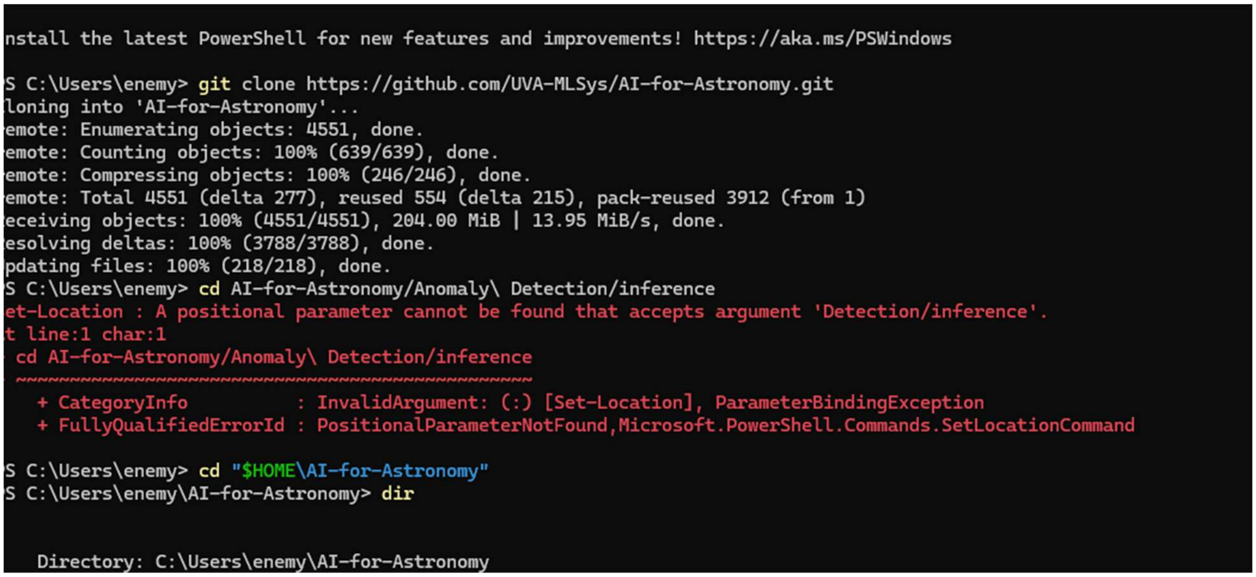
Final Project Step 3

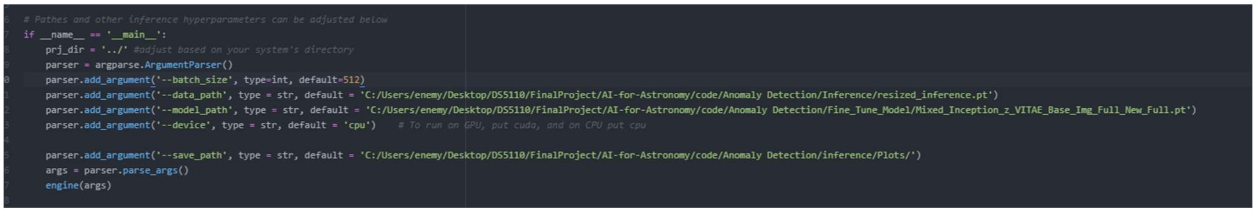
Group 6: Dae Hwang & Pratham Choksi

1. Repository Cloning



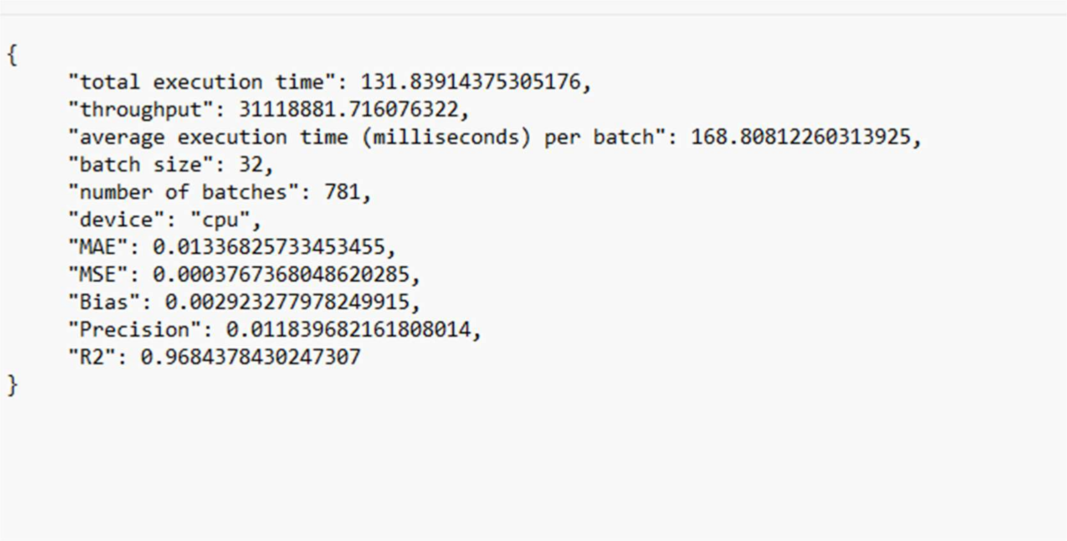
```
nstall the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

S C:\Users\enemy> git clone https://github.com/UVA-MLSys/AI-for-Astronomy.git
loning into 'AI-for-Astronomy'...
emote: Enumerating objects: 4551, done.
emote: Counting objects: 100% (639/639), done.
emote: Compressing objects: 100% (246/246), done.
emote: Total 4551 (delta 277), reused 554 (delta 215), pack-reused 3912 (from 1)
eceiving objects: 100% (4551/4551), 204.00 MiB | 13.95 MiB/s, done.
esolving deltas: 100% (3788/3788), done.
pdating files: 100% (218/218), done.
S C:\Users\enemy> cd AI-for-Astronomy/Anomaly\ Detection/inference
et-Location : A positional parameter cannot be found that accepts argument 'Detection/inference'.
t line:1 char:1
 cd AI-for-Astronomy/Anomaly\ Detection/inference
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
   + CategoryInfo          : InvalidArgument: (:) [Set-Location], ParameterBindingException
   + FullyQualifiedErrorId : PositionalParameterNotFound,Microsoft.PowerShell.Commands.SetLocationCommand

S C:\Users\enemy> cd "$HOME\AI-for-Astronomy"
S C:\Users\enemy\AI-for-Astronomy> dir


   Directory: C:\Users\enemy\AI-for-Astronomy
```

2. File Path Updates



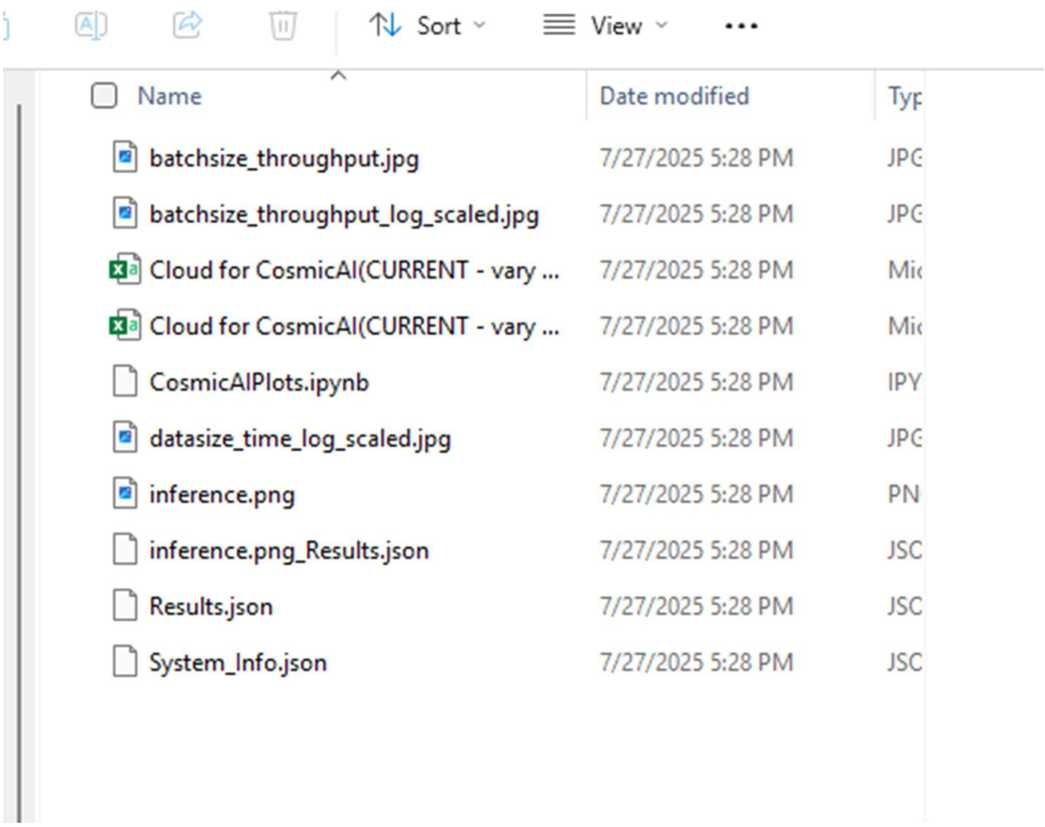
```python
# Pathes and other inference hyperparameters can be adjusted below
if __name__ == '__main__':
    prj_dir = '../' #adjust based on your system's directory
    parser = argparse.ArgumentParser()
    parser.add_argument('--batch_size', type=int, default=512)
    parser.add_argument('--data_path', type = str, default = 'C:/Users/enemy/Desktop/DS5110/FinalProject/AI-for-Astronomy/code/Anomaly Detection/Inference/resized_inference.pt')
    parser.add_argument('--model_path', type = str, default = 'C:/Users/enemy/Desktop/DS5110/FinalProject/AI-for-Astronomy/code/Anomaly Detection/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 = 'C:/Users/enemy/Desktop/DS5110/FinalProject/AI-for-Astronomy/code/Anomaly Detection/inference/Plots/')
    args = parser.parse_args()
    engine(args)
```

3. Documentation of Execution Time



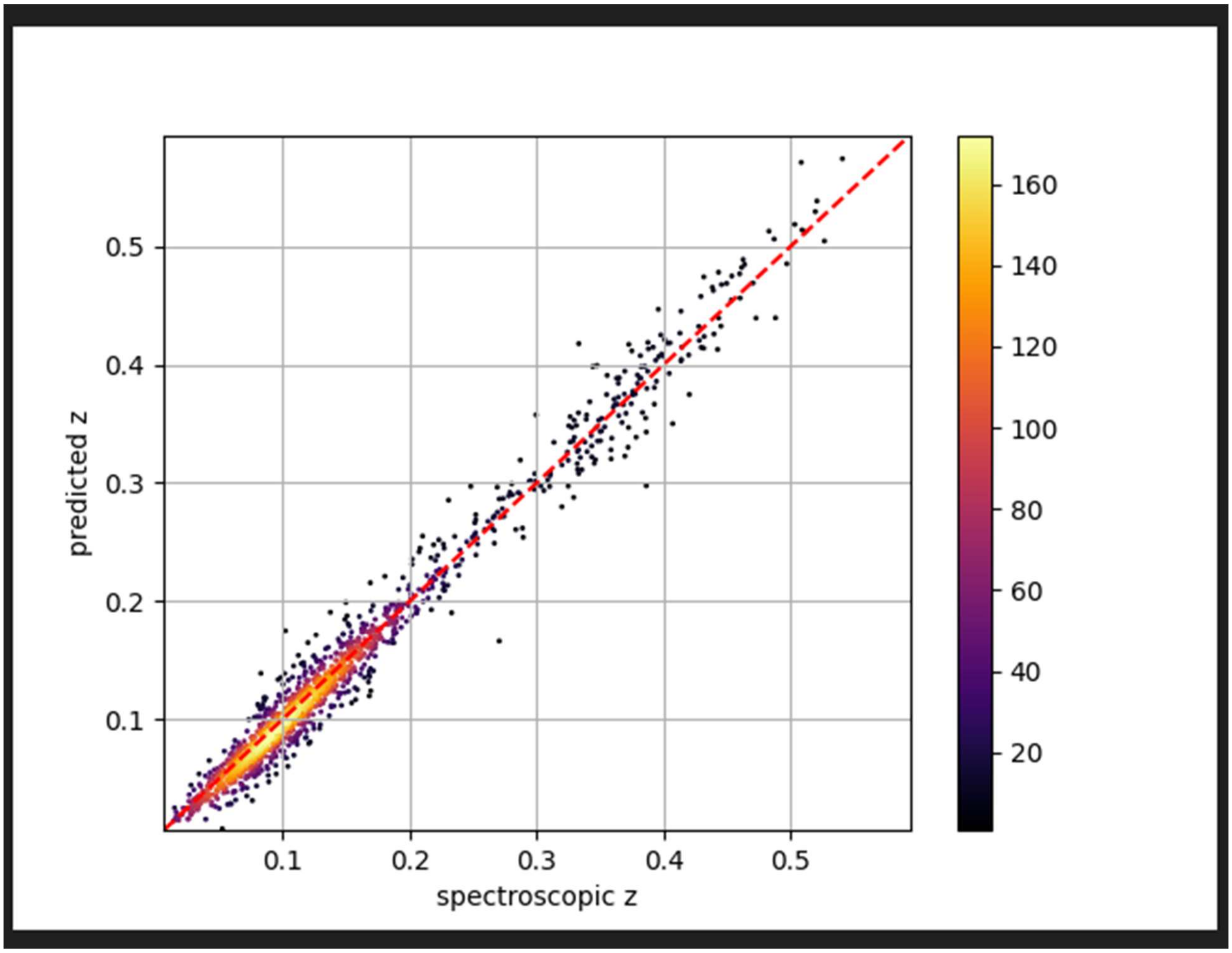
```
{
     "total execution time": 131.83914375305176,
     "throughput": 31118881.716076322,
     "average execution time (milliseconds) per batch": 168.80812260313925,
     "batch size": 32,
     "number of batches": 781,
     "device": "cpu",
     "MAE": 0.01336825733453455,
     "MSE": 0.0003767368048620285,
     "Bias": 0.002923277978249915,
     "Precision": 0.011839682161808014,
     "R2": 0.9684378430247307
}
```

4. Output files

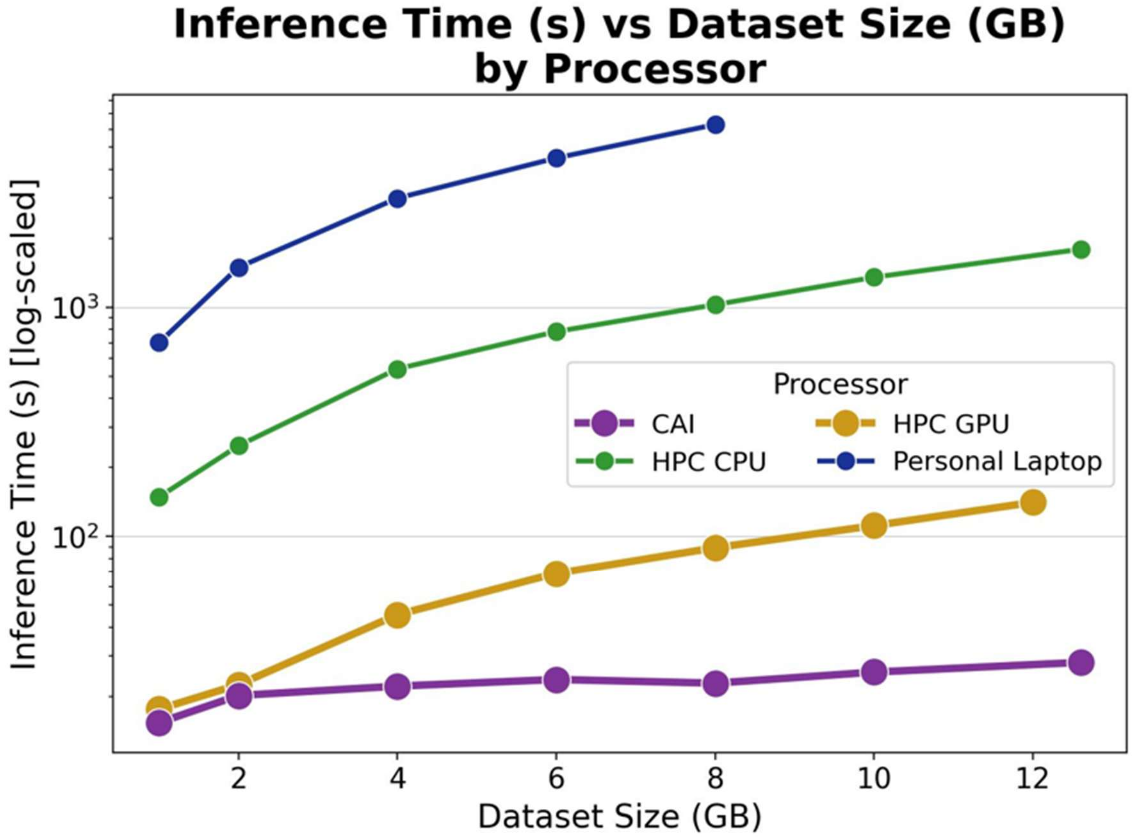


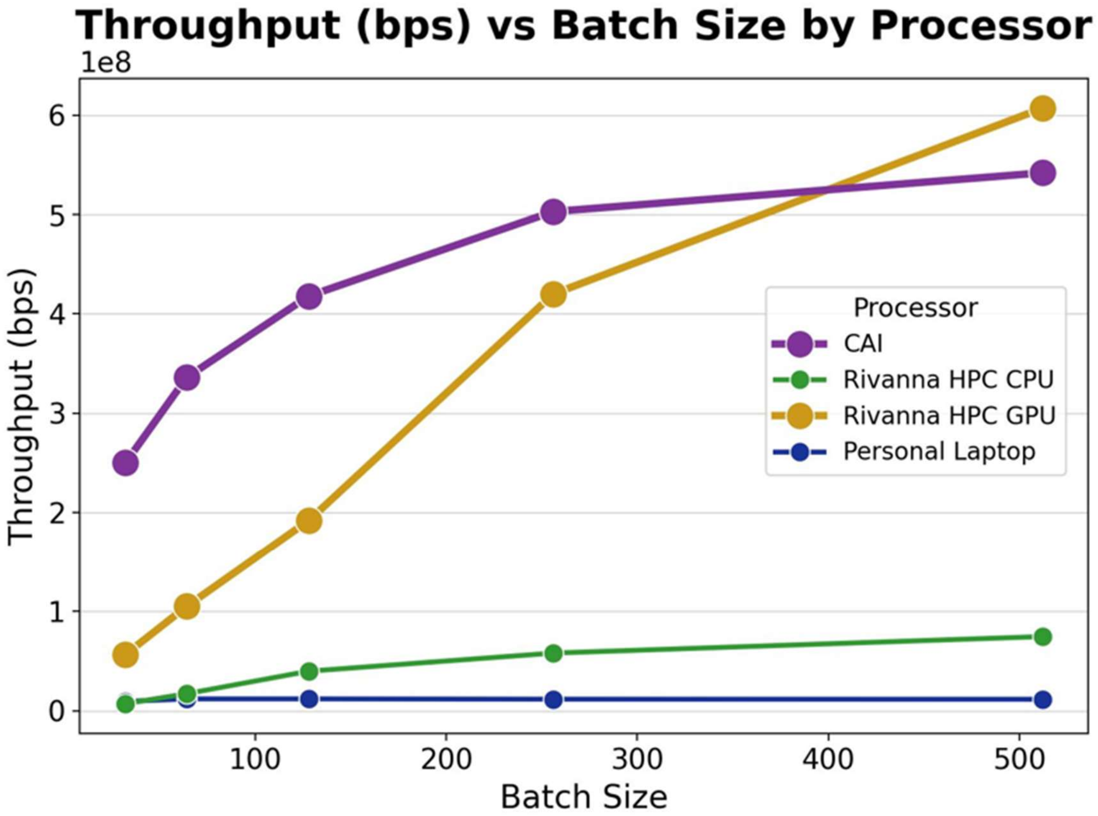
Name	Date modified	Typ
batchsize_throughput.jpg	7/27/2025 5:28 PM	JPG
batchsize_throughput_log_scaled.jpg	7/27/2025 5:28 PM	JPG
Cloud for CosmicAI(CURRENT - vary ...	7/27/2025 5:28 PM	Mic
Cloud for CosmicAI(CURRENT - vary ...	7/27/2025 5:28 PM	Mic
CosmicAIPlots.ipynb	7/27/2025 5:28 PM	IPY
datasize_time_log_scaled.jpg	7/27/2025 5:28 PM	JPG
inference.png	7/27/2025 5:28 PM	PN
inference.png_Results.json	7/27/2025 5:28 PM	JSO
Results.json	7/27/2025 5:28 PM	JSO
System_Info.json	7/27/2025 5:28 PM	JSO

5. Analysis of inference performance

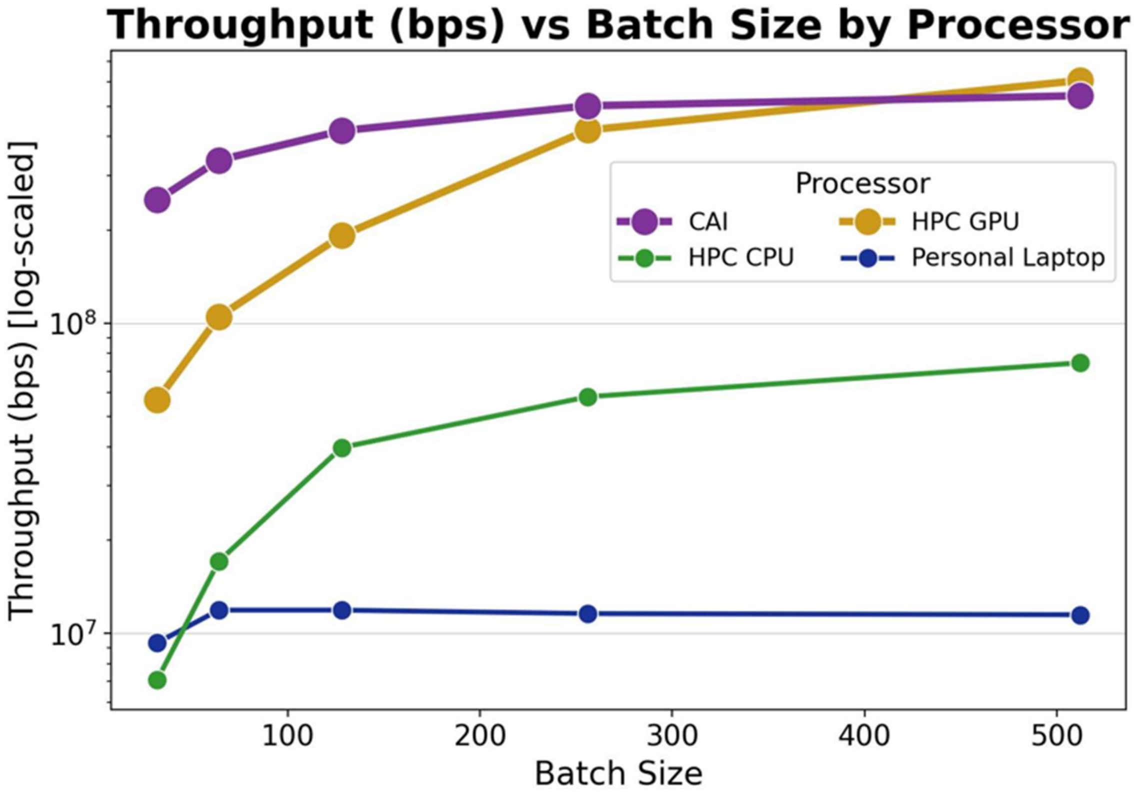

predicted z
0.1
0.2
0.3
0.4
0.5
spectroscopic z
0.1
0.2
0.3
0.4
0.5
20
40
60
80
100
120
140
160

6. Comparison of different deployment options





Throughput (bps) vs Batch Size by Processor



Throughput (bps) vs Batch Size by Processor

7. Analysis

The inference was performed on my personal computer using a CPU. The model processed the dataset in about 132 seconds with a batch size of 32, averaging around 169 milliseconds per batch. The Predicted vs Spectroscopic z chart shows how closely the output values align with the true values, indicating that the model's predictions follow the expected pattern. The Inference Time vs Dataset Size chart illustrates that prediction time increases as dataset size grows. The CAI system completed predictions faster than the other options, while the personal computer was the slowest. The Throughput vs Batch Size chart reveals that more powerful systems can handle larger batch sizes more efficiently, leading to higher processing rates. The Log-Scaled Throughput vs Batch Size chart confirms this trend by showing improved throughput across systems, especially for CAI and HPC GPU, as batch size increases.

While working on the project, I ran into a few issues that I needed to fix. First, I had trouble accessing a folder because the path had space in it, which I corrected by using quotes. I also saw errors due to software version mismatches on my system, especially related to the PyTorch library. Because of changes in newer versions of PyTorch, I had to update the load_model and load_data functions to include additional options and adjust how files were loaded. These changes helped the code run correctly on my setup. I also made sure the batch size was reasonable for my system and confirmed that all required output files were saved in the right location for submission.