# Space Weather Center

Andrew Martin, Autumn Nippert, Jonathan Ranger, Kristopher Fierro-Marin, Mitch Tilton
Sponsor: Matthew Argall
*Department of Computer Science, University of New Hampshire, Durham, NH 03824*

## Introduction

Space weather impacts various infrastructures, from power grids to satellite communications. The Space Weather Center processes and visualizes real-time space weather data to support research, forecasting, and industry decision-making. Our project integrates data from NOAA's (National Oceanic and Atmospheric Administration) Space Weather Prediction Center, utilizing data from two satellites: DSCOVR, and its next-generation successor SWFO-L1, launching in 2025.

By developing a real-time data pipeline and visualization tools, we aim to improve the understand of solar wind interactions with the Earth's magnetic field, enhance space weather predictions, and compare NOAA's calculations with UNH's independent analyses.

## Requirements

**Functional:**

- Real-Time Data Retrieval – Continuously fetch space weather data from satellites for up-to-date analysis and predictions.
- Data Visualizations – Generate real-time plots to aid in space weather forecasting.
- Public Accessibility – Display prediction plots on a public platform for easy access by researchers and the public.

**Non-functional:**

- Usability - The web API will be intuitive and easy to use, allowing users to view real-time plots by simply visiting the platform.
- Performance – Prediction plots will be processed and displayed within 256 seconds of receiving new data.

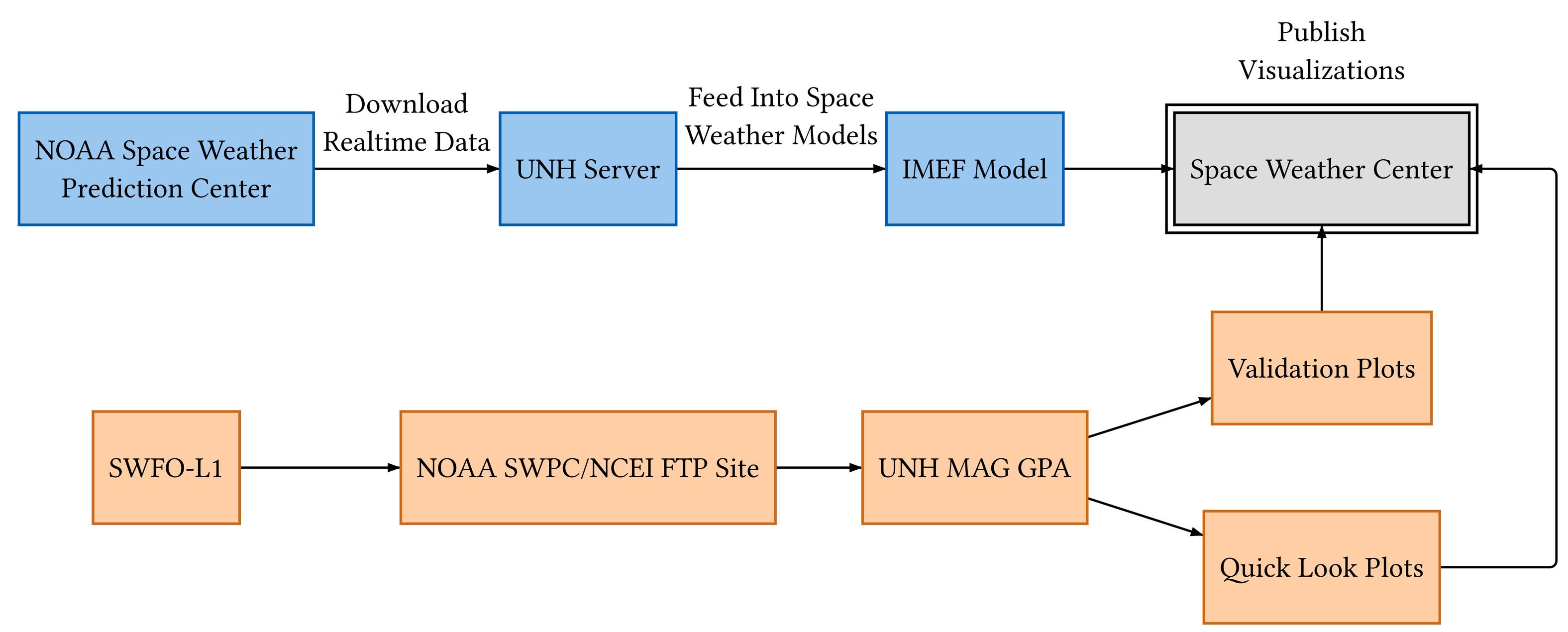
## Design

**DSCOVR:**
The workflow for DSCOVR starts at the NOAA Space Weather Prediction Center, where we download the real-time data from the satellite onto a server owned by UNH. From here, we send the data through the Inner Magnetospheric Electric Field (IMEF) Model to predict and produce predictions that we then serve to the UNH Space Weather Center website.

**SWFO:**
Since SWFO has not yet launched, its workflow differs from DSCOVR. NOAA publishes test data on an SFTP site, which our program downloads. These files are extracted and processed using UNH software to generate Quick-look/Comparison plots. Once SWFO is operational, this data will be integrated into the existing DSCOVR models to produce visualizations for the UNH Space Weather Center.

## Pipeline Design Flow Chart

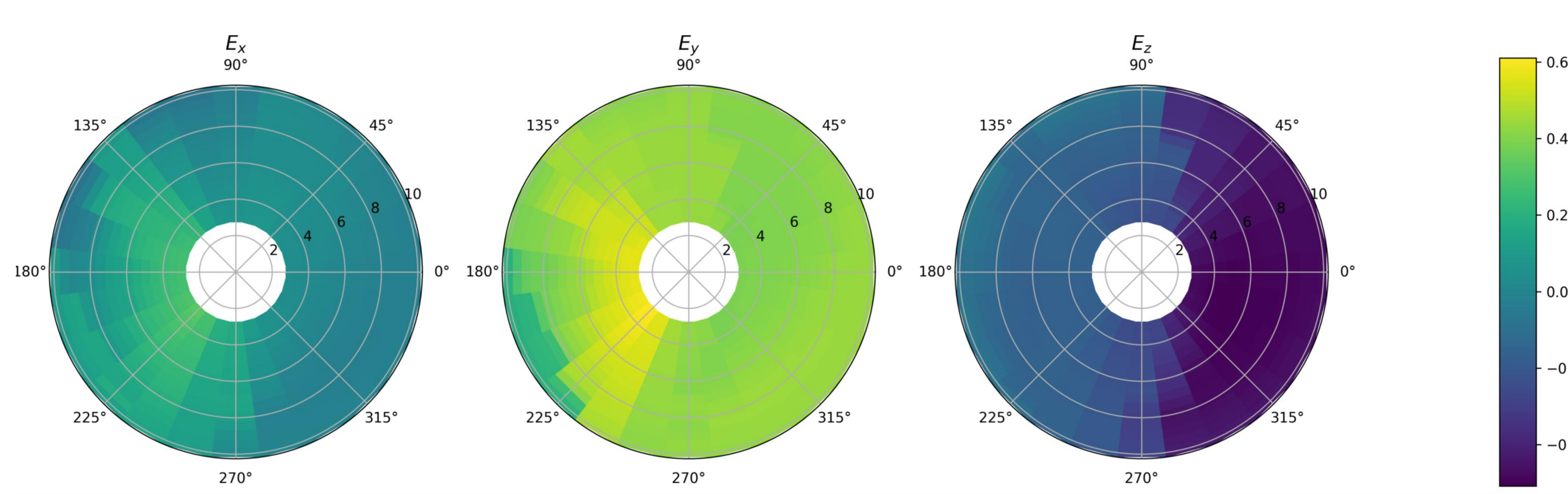
UNH Space Weather Forecasting




Data from DSCOVR and SWFO satellites are downloaded, processed, and turned into plots. Results are then published to UNH's Space Weather Center website within 256 seconds of collection

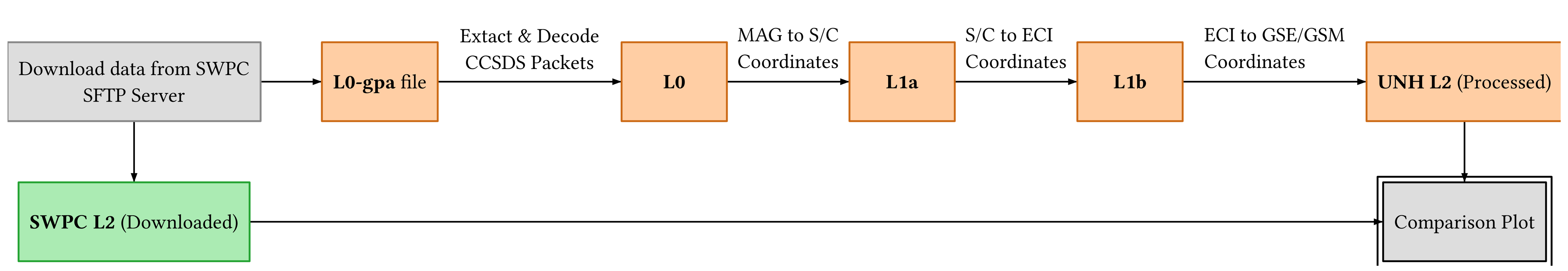
## DSCOVR Heatmap

Electric Field Predictions in Earth's Magnetosphere.



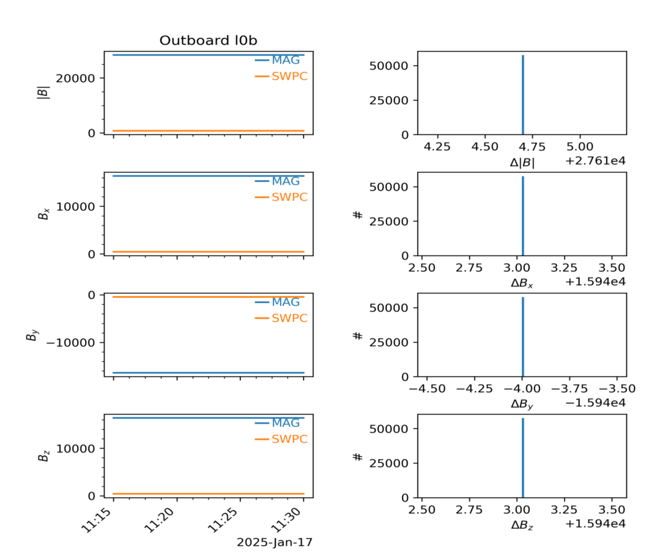

The real-time monitoring system transforms solar wind and geomagnetic data into global electric field predictions through machine learning. Solar wind conditions are processed, scaled and fed into a neural network, which generates predictions for all locations in the magnetosphere.

## SWFO Design Flow Chart




The SWFO pipeline transforms raw satellite telemetry into science-ready data through multiple processing levels.

Data is unpacked, validated, calibrated, and corrected for spacecraft effects before being projected into scientific coordinate systems. Our primary goal is to produce comparison plots that verify the consistency between UNH's processed L2 data (transformed from raw L0-gpa) and NOAA's L2 files. These visual comparisons help ensure the correctness and reliability of our data processing workflow.




## Implementation

**SWFO:**
Python automates downloading and processing SWFO telemetry data. A script connects to NOAA's SFTP site, retrieves new files, and processes them through multiple levels to generate calibrated datasets. We then produce verification plots to compare NOAA's processed data to our own.

**DSCOVR:**
Python was used to develop an Artificial Neural Network (ANN) for predicting the Earth's magnetic field based on space weather. The model utilizes various libraries such as xarray for preparing data, PyTorch for training, and Pandas for data manipulation.

**API:**
The API is backed by Python Flask and Gunicorn. It is built to be run on a Rocky Linux Server with multiple Systemd services providing robust backend functionality for processing and accessing data.

## Testing

**DSCOVR**: We have written API tests to ensure that the interface behaves as intended, including testing robustness in the case of missing data. Our model has been tested using metrics such as validation loss to ensure accurate predictions. We also leveraged our sponsors knowledge of the field to assess model performance.

**SWFO**: We implemented a mock data pipeline to simulate real-time data ingestion. Mock files are generated with updated timestamps to imitate new data uploads, allowing us to verify that the most recent files are correctly detected, downloaded, processed from L0-gpa to L2, and compared against NOAA's L2 data. This ensures our system's automated comparison and plotting workflow performs as expected.

## Evaluation & Conclusions

Our project has successfully met all our stated success criteria. We have implemented data processing pipelines for DSCOVR and SWFO-L1, enabling real-time data processing and visualization within 256 seconds. Visualizations made use of interactive plotting software for user friendly access, enabling ease of public use.

The DSCOVR model training was slow due to limited hardware capabilities. Future work could include adding a model translation layer to accommodate different satellite inputs.

While the SWFO-L1 satellite is not yet operational, these pipelines have been proven functional with mock data successfully outputting comparison plots. These plots lay the groundwork for future validation and will help identify discrepancies between NOAA and UNH processing pipelines.

## Acknowledgements

We would like to thank:

- UNH Drupal Team for their assistance on publicizing our visualizations.
- Brianna N. Isola for their assistance on our training model
- Craig Smith, our advisor.