HelioIndex

A Python Library for Reproducible Construction of Machine-Learning Datasets from Solar Observations and Event Catalogs

Authors: Balaji Kannan, Robin Chataut, Chetraj Pandey

THE PROBLEM

Without HelioIndex	With HelioIndex
– Ad-hoc scripts per group	✓ One unified pipeline
– No shared standards	✓ Shared, reproducible
– Data leakage risk	✓ Leakage prevention
– Non-reproducible splits	✓ Comparable results
– Results incomparable	✓ Config-driven splits

Every group rebuilds the same pipeline independently — *incompatibly, without publishing code, producing incomparable results.*

WHAT IS HELIOINDEX?

An open-source Python library that standardizes ML-ready dataset construction for solar event forecasting — from raw observation tables and event catalogs.

Observation Table + **Event Catalog** ⟶ **PyTorch Dataset**

→ Flexible observation windows — single frame to multi-frame sequences
→ Configurable prediction horizons & event-labeling rules
→ Missing observation handling & explicit temporal gap tokens
→ Reproducible chronological and active-region-aware splits

DATA SOURCES

HelioIndex ingests user-provided tables — no hardcoded downloads. Works with any timestamped observation series.

SDO / HMI & AIA (NASA)
Magnetogram & EUV timestamps; FITS paths

GOES X-Ray Flare Catalog (NOAA)
Event start/peak/end; flare class A–X

NOAA Active Region Lists (HARP)
Region ID, centroid, HALE classification

Future: CME & SEP Catalogs
Extensible to DONKI, CDAW, SEP event lists

END-TO-END PIPELINE

`obs_table + event_catalog → HelioIndex.createIndex() → Indexed samples → PyTorch-ready splits`

Inputs
Observation Table
+ Event Catalog

→

Cadence Norm.
Normalize timestamps
fill / flag gaps

→

Windowing
Build observation windows
+ prediction horizon

→

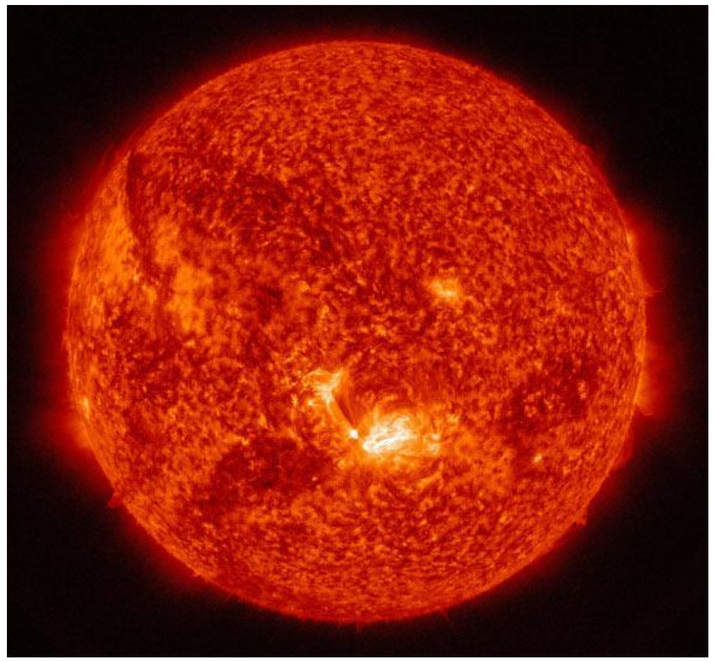
Event Labeling
Assign flare class
configurable rules

→

Dataset Splitting
Time-based / class-based
/ AR-aware

→

Dataset
train / evaluate / test
DataLoader ready



USE CASE EXAMPLES

Full-Disk Flare Forecasting
- 24-hour prediction horizon from observation cutoff
- Strict chronological train / evaluate / test split
- Active-region grouping prevents leakage by location

Active-Region Classification
- NOAA region-level sequences from HARP patches
- AR-aware split — same region never in train + test
- Multi-class target: B / C / M / X peak flare class
- Region grouping prevents temporal leakage by AR

Sequence Modeling with Gaps
- Explicit missing-image tokens preserve sequence length
- Temporal gap flags maintain causal order during outages
- Tri-monthly cross-validation for seasonal robustness
- Handles SDO/GOES data gaps & reduced-cadence periods

KEY BENEFITS

1 Eliminates Redundancy
No group rebuilds the same pipeline. One standardized library replaces dozens of incompatible ad-hoc scripts across the community.

2 Uniform & Shareable Splits
Reproduce any group's exact dataset partition from a config file. Identical splits enable fair, apples-to-apples comparisons.

3 Flexible Pipeline Filtering
Drop specific images, exclude low-quality frames, or swap event catalogs without rewriting any pipeline logic.

4 Prevents Data Leakage
Chronological and AR-aware splits are first-class features — rigorous experimental design by default, not by accident.

CONTACT

Balaji Kannan
Dept. of Computer Science · Texas Christian University
balaji.kannan@tcu.edu
github.com/balajiofficial

ACKNOWLEDGEMENTS

This work is supported by the Department of Computer Science at Texas Christian University. My advisor for this research is Dr. Chetraj Pandey, Assistant Professor of Computer Science in Texas Christian University.