Convection Cell Analytics

Physics-informed lightning waveform classification for edge deployment

TECHNICAL REPORT V0.1.0

PROJECT: CONVECTION-CELL-ANALYTICS

Overview

Convection Cell Analytics (CCA) is a physics-informed machine learning system for real-time classification of lightning discharge waveforms. The system distinguishes four discharge types: positive intra-cloud (+IC), negative intra-cloud (-IC), positive cloud-to-ground (+CG), and negative cloud-to-ground (-CG). Classification accuracy targets 97% overall with F1 ≥ 0.98 for safety-critical CG events.

The architecture combines a lightweight 1D CNN with explicit physics feature extraction. Waveforms (512 samples at 465 kHz, spanning 1100 µs) are processed through parallel branches: a CNN extracts learned representations while a physics branch computes 13 parametric features encoding electromagnetic domain knowledge. These are fused via concatenation before a classification head.

> **DESIGN TENETS**
>
> Physics-Informed Classification: Leverage electromagnetic signatures inherent in lightning waveforms. Edge Deployment: Operate on resource-constrained environments with strict latency (<0.08ms), memory (<60KB), and model size (<0.5MB) constraints.

Contents

1. Physics Features

The system extracts 13 parametric features from each waveform, encoding domain knowledge about electromagnetic signatures of different discharge types. Core features (8) provide primary discrimination; extended features (5) improve +CG vs -IC separation.

FEATURE	DESCRIPTION	DISCRIMINATIVE VALUE
Peak Amplitude	Signed maximum intensity	Polarity discrimination
Time to Peak	Rise time in μs	CG ~5μs, IC ~20μs
FWHM	Full width at half maximum	Narrow CG, broad IC
Decay Constant	Exponential tail τ	CG ~100μs, IC ~500μs
Symmetry Index	E+ / E- energy ratio	CG unipolar, IC bipolar
Bipolarity Index	Zero-crossing frequency	IC oscillatory character
Energy Bands	1-10kHz, 10-100kHz	Spectral signatures

Extended features: peak curvature, zero crossing count, mid-band energy (10-50kHz), envelope slope, pulse width.

2. CNN Architecture

The network is optimized for CPU inference, avoiding slow dilated convolutions. Three strided convolutions progressively downsample while extracting hierarchical features.

Network Layers

- Stem: 2→16 ch, k=7, s=4
- Block 1: 16→24 ch, k=5, s=2
- Block 2: 24→32 ch, k=5, s=2
- Global avg pool + Physics MLP
- Head: 48→4 (softmax)

Design Choices

- No dilated convolutions
- BatchNorm fused at export
- Focal loss for +CG imbalance
- Total: 6 ops vs 40+ baseline

> **FUSION STRATEGY**
>
> `output = head(concat(cnn, physics))` — Physics branch provides interpretable baseline; CNN captures subtle patterns.

3. Performance Constraints

Edge deployment requires strict adherence to resource constraints. The following invariants are defined in `mission.yaml` and enforced through automated testing.

Classification Accuracy

97% OVERALL ACCURACY	**≥0.98** +CG F1 SCORE
≥0.98 -CG F1 SCORE	**4** CLASSES

Cloud-to-ground events are prioritized for safety. F1 ≥ 0.98 required for both +CG and -CG.

Resource Constraints

<0.5MB MODEL SIZE (INT8)	**<0.08ms** INFERENCE LATENCY
<60KB PEAK MEMORY	**>10k/s** THROUGHPUT

Latency (<0.08ms) enables real-time processing. Memory (<60KB) fits microcontrollers. Throughput (>10k/s) handles storm-scale events.

Test Matrix

TEST	INVARIANT	PARAMETERS
T1	Four-class discrimination ≥97%	n=400, train=300
T2	CG F1 threshold ≥0.98	stratified=true
T3	Physics features (13)	—
T4-T7	Resource constraints	n_iter=100
T8-T9	Architecture validation	—

4. Project Structure

The codebase is a UV workspace with two libraries: `waveform` (domain models, physics, classifiers) and `ml-utils` (training infrastructure).

waveform

- `model.py` — Domain types
- `physics/` — Feature extraction
- `adapters/` — Classifiers
- `data/` — Dataset loaders
- `infra/` — ONNX export

ml-utils

- `model.py` — Training loop
- `callback.py` — Callbacks
- `tracker/` — Experiment tracking
- `loss.py` — Focal loss
- `benchmark.py` — Performance

Data Pipeline

Waveforms stored as NumPy arrays (512 samples, float64). Pipeline: normalize (z-score) → derivatives (central diff) → physics features → batch. Masked aggregation handles 1-5 waveforms per pulse.

> **DEPENDENCIES**
>
> `numpy`, `scipy`, `torch`, `einops`, `attrs`, `expression`. Dev: `pytest`, `hypothesis`.

Summary

CCA combines physics-informed feature extraction with a lightweight CNN for real-time lightning classification on edge devices. The hybrid architecture ensures interpretability through explicit physics features while capturing subtle patterns via learned representations. Strict resource constraints enable deployment on embedded systems and WebAssembly. All invariants are codified in `mission.yaml` and enforced through automated testing.