

NN-POLAN: Physics-Informed Neural Network Ionogram Inversion

Architecture · Training · Validation · Sporadic-E Extension

S. Chakraborty *et al.*

Embry-Riddle Aeronautical University — pynasonde project

April 15, 2026

`pynasonde.vipir.analysis.nn_inversion` — github.com/shibaji7/pynasonde

The Ionogram Inversion Problem

What we measure vs. what we want

Measure: virtual-height trace $h'(f)$ — apparent reflection height

Want: true electron density profile $N(h)$

Abel integral (forward model):

$$h'(f) = h_0 + \int_{h_0}^{h_r(f)} \frac{dh}{\sqrt{1 - f_p^2(h)/f^2}}$$

where $f_p = \sqrt{N_e/12441}$ [MHz] is the plasma frequency.

Inversion = find $N(h)$ such that $\text{Abel}(N) = h'(f)$

Classical solver: **POLAN** (iterative, 1–5 s/ionogram)

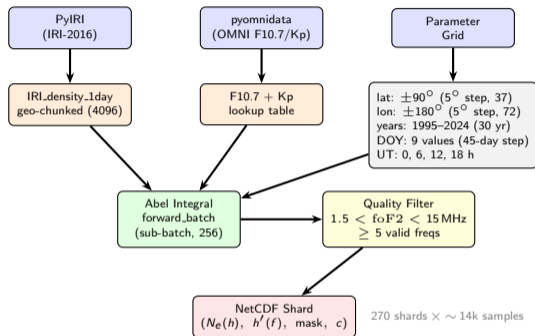
POLAN limitations

- × Sequential, slow per ionogram
- × Requires clean, scaled traces
- × Fails on spread-F, oblique echoes
- × No uncertainty estimate

NN-POLAN targets

- ✓ <1 ms/ionogram (GPU batch)
- ✓ Works on raw/noisy traces
- ✓ Physics-constrained via Abel loss
- ✓ Generalises across stations & seasons

Synthetic Data Generation Pipeline



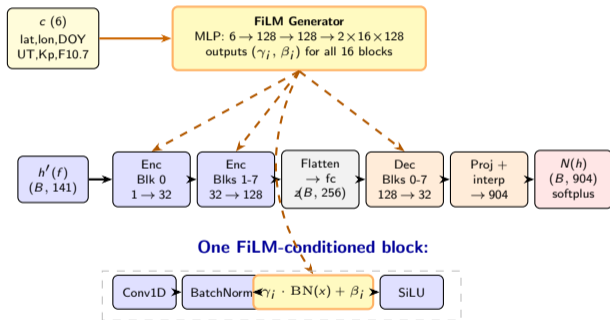
Conditioning vector c (6 values)

[lat, lon, DOY, UT, Kp, F10.7] Real solar flux from OMNI;
Kp derived from hourly SymH

Grid specs

Height: 60–511.5 km, 0.5 km step → **904** pts
Freq: 1.0–15.0 MHz, 0.1 MHz step → **141** pts

NNPolan Architecture — Overview



Key specifications

Component	Details
Input	$h'(f)$: ($B, 1, 141$)
Encoder	8 Conv1D blocks
Channels	$1 \rightarrow 32 \rightarrow 64 \rightarrow 128$
Stride-2	4 of 8 blocks ($141 \rightarrow 9$)
Bottleneck	flatten($B, 128, 9$) \rightarrow ($B, 1152$)
Latent z	($B, 256$) via fc $1152 \rightarrow 256$
Decoder	8 TConv blocks
Upsamp.	$9 \rightarrow 18 \rightarrow \dots \rightarrow 144$
Final Output	interp to 904 pts softplus ≥ 0
FiLM MLP	$6 \rightarrow 128 \rightarrow 128$
FiLM out	(γ_i, β_i) per block
Total params	1.43 M

Why FiLM conditioning?

A *single* model covers all:
latitudes, seasons, solar activity,
and local times via $c = (\gamma, \beta)$

Encoder (8 Conv1D blocks)

Purpose: compress $h'(f)$ into a latent representation

Blk	Operation	Ch	Stride
0	Conv1D $k = 7, \text{pad}=3$	$1 \rightarrow 32$	1
1	Conv1D $k = 5, \text{pad}=2$	$32 \rightarrow 32$	1
2	Conv1D $k = 5, \text{pad}=2$	$32 \rightarrow 64$	2
3	Conv1D $k = 3, \text{pad}=1$	$64 \rightarrow 64$	1
4	Conv1D $k = 3, \text{pad}=1$	$64 \rightarrow 128$	2
5–7	Conv1D $k = 3, \text{pad}=1$	$128 \rightarrow 128$	1 or 2

Each block: Conv1D \rightarrow BN \rightarrow FiLM \rightarrow SiLU

Flatten: $(B, 128, 9) \rightarrow (B, 1152)$

fc_out: $(B, 1152) \rightarrow (B, 256)$

No global-average-pool — preserves frequency-position for Abel gradients

FiLM Generator MLP

Input: conditioning vector $c \in \mathbb{R}^6$

Hidden: $128 \rightarrow 128$ (SiLU)

Output: $2 \times 16 \times 128$ scalars

$\Rightarrow \gamma_i \in \mathbb{R}^{128}, \beta_i \in \mathbb{R}^{128}$ for each of 16 blocks

$\tilde{x} = (1 + \gamma_i) \cdot \text{BN}(x) + \beta_i$ (*multiplicative residual form*)

Decoder (8 TConv blocks)

Purpose: map latent z back to profile space

Blk	Operation	Size
–	fc_in: $256 \rightarrow 128 \times 9$	
–	reshape $(B, 128, 9)$	9
0	TConv $k = 4, \text{stride}=2$	18
1	Conv $k = 3, \text{stride}=1$	18
2	TConv $k = 4, \text{stride}=2$	36
3	Conv $k = 3, \text{stride}=1$	36
4–7	similar $\times 2$ ups.	144
–	proj: $32 \rightarrow 1$	144
–	interp (linear) $\rightarrow 904$	904

Each block: TConv \rightarrow BN \rightarrow FiLM \rightarrow SiLU

Final activation: $\text{softplus}(x) = \ln(1 + e^x) \geq 0$

Why softplus output?

Forces $N(h) \geq 0$ (physical constraint) with smooth gradients near zero (unlike ReLU). Prevents hard cutoffs during gradient descent.

Combined Stage-1 loss

$$\mathcal{L} = \underbrace{\lambda_b \mathcal{L}_{\text{bg}}}_{\text{IRI prior}} + \underbrace{\lambda_\phi \mathcal{L}_{\text{Abel}}}_{\text{physics}} + \underbrace{\lambda_m \mathcal{L}_{\text{mono}}}_{\text{reg.}}$$

Abel inversion loss (log₁₀-space):

$$\mathcal{L}_{\text{Abel}} = \frac{1}{|\mathcal{B}|} \sum_{h \in \mathcal{B}} \left(\log_{10} \hat{N}(h) - \log_{10} N_{\text{abel}}(h) \right)^2$$

$N_{\text{abel}}(h)$: Ne from Abel quadrature inversion of $h'(f)$

\mathcal{B} : bottomside height mask (within observed trace)

Monotone: penalises $\uparrow f_p$ above hmF2 only

Background: $\text{MSE}(\hat{N}_n, N_n^{\text{IRI}})$ in log-norm space

Why Abel *inversion*, not forward?

Forward Abel: $h'_{\text{pred}} = \int \mu' dh$

$$\mu' = 1/\sqrt{1 - f_p^2/f^2} \rightarrow \infty \text{ near foF2}$$

⇒ gradient spike up to 2500× larger near foF2

⇒ overwhelming bg gradient ⇒ Ne collapse

Inversion: $h_r(f_p) = \frac{1}{N_q} \sum_k h'(f_p \sin \theta_k)$

Singularity *absent*: $f_p \sin \theta \leq f_p < f$ always

⇒ well-conditioned gradients everywhere

Pool bottleneck: GlobalAvgPool mixes all freq-band gradients ⇒ Abel signal lost in noise

Fixed by: flatten $(B, 128, 9) \rightarrow (B, 1152)$

Default weights (config.toml)

$\lambda_b = 1.0$ $\lambda_\phi = 10^{-4}$ (ramped) $\lambda_m = 0.1$ $N_w = 10$ warmup

Problem: the forward Abel singularity

Training goal: minimise $\|\mathcal{H}[\hat{N}] - h'\|^2$

$$\mathcal{H}[\hat{N}](f) = h_{\text{base}} + \int \frac{dh}{\sqrt{1 - f_p^2/f^2}}$$

Near foF2: $f_p \rightarrow f \Rightarrow \mu' \rightarrow \infty$

Discrete grid: $\mu'_{\text{max}} \approx 50$ (clamped at 50)

Gradient $\partial\mathcal{H}/\partial N \propto \mu'^2 \sim 2500\times$

Observed failure:

Config B (Abel-only) showed FLAT/INCREASING across MSE, log, and cumulative Abel formulations

Even after global-avg-pool was removed

Root cause: singularity in the forward graph cannot be fully tamed by clamping or reweighting

Solution: singularity-free inversion

Substitution $f = f_p \sin \theta$ in the Abel identity:

$$h_r(f_p) = \frac{2}{\pi} \int_0^{\pi/2} h'(f_p \sin \theta) d\theta$$

Midpoint quadrature ($N_q = 64$ nodes):

$$h_r(f_p) \approx \frac{1}{N_q} \sum_k h'(f_p \sin \theta_k)$$

Since $f_p \sin \theta_k \leq f_p < f$ always:

the $1/\sqrt{f^2 - \xi^2}$ kernel *never appears*

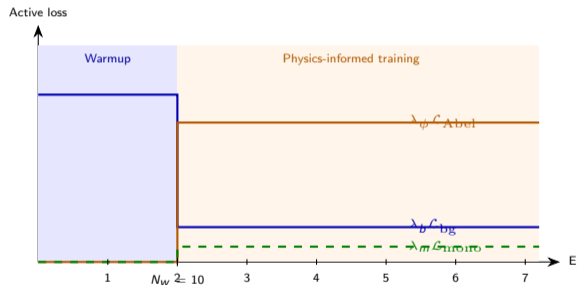
Ne at reflection: $N_{\text{abel}}(h_r) = f_p^2 \times 12441 \text{ cm}^{-3}$

Interpolate $\{(h_r, N_{\text{abel}})\}$ to $H_{\text{GRID_KM}}$

\Rightarrow fixed target computed with `torch.no_grad()`

Result: Config B now shows DECREASING Abel loss $\mathcal{O}(0.1)$ at convergence (log₁₀-space)

Training Curriculum: Warmup Prevents Ne Collapse



The $N_e \rightarrow 0$ collapse problem

At random init: $N(h) \approx 0$

$\Rightarrow h'_{\text{pred}} \approx 511$ km (top of grid)

$h'_{\text{obs}} \approx 200\text{--}400$ km

Forward Abel gradient pushes N further toward 0.

Warmup fix: bg loss first teaches a non-trivial $N(h)$ shape from IRI prior; Abel inversion then corrects physics consistency.

Ramp: λ_ϕ increases linearly from 0 to 10^{-4} over 10 post-warmup epochs.

Stage 2 (real data)

No background term ($\lambda_b = 0$)

Physics-only: Abel inversion + monotone

Optional: freeze FiLM generator

\Rightarrow preserves Stage 1 geophysical prior

Stage-1 Training: Loss Convergence

Warmup run (bg-only, 5 shards):

Ep	Train bg	Val bg	Train Abel	LR
0	8.5×10^{-4}	4.0×10^{-5}	0.0	3.0×10^{-4}
1	5.3×10^{-5}	2.5×10^{-5}	0.0	2.99×10^{-4}
2	4.0×10^{-5}	2.5×10^{-5}	0.0	2.98×10^{-4}
3	3.2×10^{-5}	1.5×10^{-5}	0.0	2.97×10^{-4}
4	2.5×10^{-5}	8.4×10^{-6}	0.0	2.95×10^{-4}

- Abel = 0 during warmup ($N_W = 10$ epochs)
- Val bg already $\mathcal{O}(10^{-5})$ after 4 epochs
- BG loss converges rapidly in log-norm Ne space
- Abel inversion ramp begins after warmup
- Full run: 270 shards, 50–100 epochs on VEGA

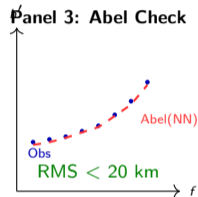
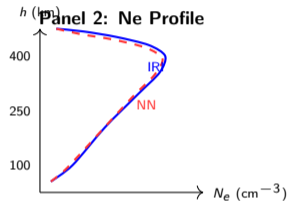
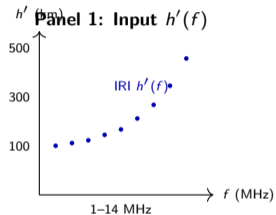
Component breakdown (Abel inversion path)

Term	Target	Notes
\mathcal{L}_{bg}	$< 10^{-4}$	log-norm MSE
\mathcal{L}_{Abel}	< 0.05	\log_{10} MSE
\mathcal{L}_{mono}	$\rightarrow 0$	topside only
Abel λ	10^{-4}	ramped
Warmup	10 ep	bg-only
Ramp	10 ep	$0 \rightarrow \lambda_\phi$

Full run (VEGA HPC)

Data: 270 shards (PBS array, --shard 0-269)
Train: 50–100 epochs on V100/A100 GPU node
Arch: 1.43 M params; spatial-latent encoder

Validation: Inference Test (3-Panel Diagnostic)



Panel 1

Input: IRI-generated $h'(f)$
Masked above foF2 (no reflection)

Panel 2

Target: IRI $N_e(h)$ (blue)
Predicted: NN output (red dashed)

Panel 3

Check: Abel(NN) vs observed $h'(f)$
Goal: RMS < 20 km after 50 epochs

What is Sporadic-E (Es)?

Thin layers ($\lesssim 1\text{--}2$ km) of enhanced ionization at **90–130 km** altitude, composed predominantly of metallic ions (Fe^+ , Mg^+ , Na^+) from meteor ablation.

Key signatures on ionograms:

- Flat trace at low virtual heights ($\sim 100\text{--}120$ km)
- High echo intensity at 1–5 MHz
- **Blanketing Es (EsB)**: reflects *all* HF — masks F layer
- **Transparent Es**: partial propagation — dual echoes

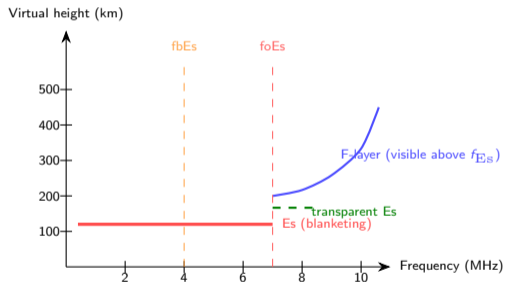
Impact on NN-POLAN

IRI does *not* model Es \Rightarrow synthetic training data has no Es
Blanketing Es: $h'(f)$ shows E-layer reflection, not F-layer
Model may invert the wrong layer — need Es detection first

Formation physics

1. **Meteor ablation** (ion source):
 ~ 100 tonnes/day of meteoric material ablates at 80–120 km, deposits Fe, Mg, Na
 2. **Wind-shear convergence** (primary mechanism):
Tidal and gravity waves create horizontal wind shear $\partial u / \partial z$ at ~ 100 km
 \Rightarrow **convergent ion motion** compresses metallic ions into thin layers
 3. **$\mathbf{E} \times \mathbf{B}$ drift** (electric field modulation):
Polarization electric fields from neutral wind dynamo enhance/suppress Es
- Diurnal pattern:** peaks in summer daytime midlatitudes (strongest tidal wind shear)

Sporadic-E: Ionogram Morphology & NN Extension Plan



Types of Es on ionograms

Type	Description
EsB (blanketing)	Strong reflection; blocks F-layer echoes below f_oE_s
EsT (transparent)	Partial reflections; F-layer visible above f_bE_s
EsC (cusp)	Isolated patches; retardation echoes
EsF (flat)	Perfectly flat trace; ionosphere nearly homogeneous

Proposed NN-POLAN extension

Step 1 — Es Detector (classification):

Binary CNN: Is Es present? What type?

Step 2 — Es-Aware Inversion:

Add Es flag to conditioning vector c
Separate loss branch for E-layer peak

Step 3 — Es Synthetic Data:

Augment PylRI profiles with Chapman Es layer
Vary h_{E_s} , N_{mE_s} , half-thickness ~ 1 km

Stage 1 — Foundation model

- ✓ FiLM encoder-decoder; **spatial-latent** (flatten, not pool)
- ✓ Synthetic data: PyIRI + OMNI (1995–2024)
- ✓ **Abel inversion loss** (singularity-free, \log_{10} MSE)
- ✓ Warmup+ramp curriculum; topside-only monotone
- ✓ config.toml single source of truth; loguru logging
- ✓ Convergence diagnostics (`diagnose_convergence.py`)
- Full VEGA run**: 270 shards, 50–100 epochs
- Validate**: \log_{10} Abel MSE < 0.05; Ne shape

Stage 2 — Station assimilation

- ✓ Abel inversion loss path implemented (`trainer_stage2.py`)
- Fine-tune on WI937 RIQ traces (pending Stage 1 best.pt)
- Per-station `.npz` in `weights/` directory
- Zero-dependency inference via `network.py`

Sporadic-E extension

- Es detector: classify EsB / EsT / none
- Augment synthetic data with artificial Es layers
- Add Es flag/intensity to conditioning vector `c`
- E-layer peak loss branch (h_{Es} , N_{mEs})

Open questions

- How many real ionograms for Stage 2?
(~500 clean traces/station is working estimate)
- CuPy for Abel GPU path? (installed, pending)
- Dual-polarisation (O-mode + X-mode) as input?
- Stage 2 convergence criterion?

Longer-term

Ensemble UQ — hmF2/foF2 head — Real-time API — JGR: ML manuscript

NN-POLAN: Where We Stand

Physics

Abel integral as core supervision
— not just post-hoc

Data

30-yr IRI library with real
F10.7/Kp from OMNI

Architecture

FiLM conditioning: one model,
all geophysical contexts

Es Extension

Sporadic-E detector + Es-aware
conditioning planned

Key engineering fixes

- **Flatten bottleneck:** pool → flatten preserves freq-position → Abel gradients can propagate
- **Abel inversion loss:** singularity removed from graph \log_{10} -MSE vs N_{abel} on bottomside only
- Warmup + ramp curriculum → no Ne collapse
- Topside-only monotone (not full profile)
- Full `config.toml` consolidation; loguru logging
- Convergence diagnostics: 4-config gradient health check

Immediate action items

- 1 Submit 270-shard PBS array on VEGA (Stage 1)
- 2 Train 50–100 epochs; monitor Abel \log_{10} MSE
- 3 Validate inference: `test_inference.py` 3-panel check
- 4 Begin Stage 2 data prep (WI937 RIQ files)
- 5 Update whitepaper & slides (in progress)
- 6 Prototype Es extension (future v2.0)

Thank you

Questions & Discussion

Code: `github.com/shibaji7/pynasonde`

Module: `pynasonde.vipir.analysis.nn_inversion`

Config: `pynasonde/config.toml` → `[nn_inversion.*]`