Physics-Informed Neural Network Inversion of Ionosonde Traces: A Foundation-Model Approach with Station-Specific Data Assimilation

S. Chakraborty[1] and *et al.*[1,2]

[1]Embry–Riddle Aeronautical University, Daytona Beach, FL, USA
[2]Fourth State Communications, USA

White Paper — Draft for Collaborator Review April 5, 2026

Abstract. We propose a physics-informed neural network (PINN) framework for true-height inversion of ionosonde traces that replaces the classical POLAN algorithm with a two-stage learning strategy. A *global foundation model* is pre-trained on ∼5 million synthetic profiles drawn from the International Reference Ionosphere (IRI-2020) spanning 30 years of solar and geomagnetic variability, conditioned on geophysical parameters (latitude, longitude, day-of-year, K_p, $F_{10.7}$) via Feature-wise Linear Modulation (FiLM). A *station-specific fine-tuning* stage adapts the foundation model to a particular ionosonde using a *physics-only* loss — no labeled true-height profiles are required. The fine-tuning objective is formally equivalent to variational data assimilation (4D-Var), with the Abel-integral group-height operator as the observation operator and the IRI-trained prior as the background state. The trained model is exported as plain NumPy weight arrays and integrated into the open-source `pynasonde` framework as a drop-in replacement for `TrueHeightInversion`, requiring no additional dependencies for inference. We argue this approach represents a genuinely novel contribution at the intersection of space physics, data assimilation, and scientific machine learning.

1 Introduction

Ionosondes have continuously monitored the Earth's ionosphere since the 1920s, producing hundreds of millions of ionogram records across a global network of stations. The fundamental scientific product derived from an ionogram is the *true-height electron density profile* $N(h)$, obtained by inverting the virtual-height trace $h'(f)$ — a measurement of the apparent round-trip travel time of radio pulses at each sounding frequency f. This inversion is an Abel-type integral equation:

$$h'(f) = \int_0^{h(f)} \mu'(f,\, N(h))\, dh, \tag{1}$$

where μ' is the group refractive index. Solving Eq. (1) for $N(h)$ given noisy, sparse observations of $h'(f)$ is an ill-posed inverse problem that has been the subject of research for more than 60 years.

Classical algorithms — POLAN [1], NHPC, and the quasi-parabolic segment approach [2] used in ARTIST/SAO systems — rely on lamination or segment fitting. They are robust on clean traces but struggle with valley regions, oblique propagation, and automated processing of large archives without manual quality control.

Neural-network-based inversions have recently appeared [3, 4], but they share a common limitation: they are *trained on POLAN (orther algorithm or hand-scaled) output as ground truth*, inheriting POLAN's systematic errors, and they do not generalize across stations or geophysical conditions without retraining on labeled data.

This work describes a fundamentally different architecture that avoids these limitations.

2 What Has Not Been Done Before

The following combination of ideas is, to the authors' knowledge, absent from the ionospheric inversion literature:

1. A **global foundation model** for ionogram inversion conditioned on continuous geophysical state, trained on IRI across 30 years of variability — not on a single station or season.

2. **Physics-only fine-tuning** on real data: the network adapts to a specific ionosonde using *only the Abel integral as supervision*, requiring zero labeled true-height profiles from the target station.

3. A rigorous **data assimilation framing** of the fine-tuning step, with the IRI prior playing the role of the background state in 4D-Var and a KL divergence term acting as the background cost function.

4. **Geophysical conditioning via FiLM** [7] so that a single network parameterizes the full range of ionospheric states from solar minimum quiet to solar maximum storm.

5. Complete **open-source integration** with zero inference-time dependencies beyond NumPy, enabling deployment on any ionosonde processing system without GPU or deep-learning framework.

3 Proposed Architecture

3.1 Network design

The network $\mathcal{F}_\theta$ maps an observed virtual-height trace $\mathbf{h}' \in \mathbb{R}^{N_f}$ (sampled at $N_f \approx 140$ frequencies) and a geophysical conditioning vector $\mathbf{c} = [\phi, \lambda, \mathrm{doy}, \mathrm{UT}, K_p, F_{10.7}]$ to an electron density profile $\hat{N}(h) \in \mathbb{R}^{N_h}_{>0}$:

$$\hat{N}(h) = \mathcal{F}_\theta(\mathbf{h}', \mathbf{c})\,. \tag{2}$$

The architecture consists of three blocks:

(i) Conditioning MLP. $\mathbf{c} \to$ two-layer MLP $\to (\boldsymbol{\gamma}, \boldsymbol{\beta})$ per encoder layer (FiLM scale and shift parameters).

(ii) 1-D CNN encoder. Three convolutional layers with ReLU activations applied to $\mathbf{h}'$, with FiLM modulation after each layer:

$$\mathbf{z}_{\ell+1} = \mathrm{ReLU}(\boldsymbol{\gamma}_\ell \odot \mathrm{Conv}_\ell(\mathbf{z}_\ell) + \boldsymbol{\beta}_\ell)\,. \tag{3}$$

FiLM allows the same convolutional weights to produce geophysically appropriate responses at high and low latitudes, storm and quiet conditions, without separate network instances.

(iii) MLP decoder. Flattened latent representation $\to$ three fully-connected layers $\to$ Softplus activation $\to \hat{N}(h) \geq 0$. A height grid of $N_h = 225$ points from 60 to 510 km in 2 km steps is used.

3.2 Differentiable forward model

The Abel integral operator $\mathcal{H}$ in Eq. (1) is implemented as a differentiable NumPy (inference) and PyTorch (training) function:

$$\hat{h}'(f) = \mathcal{H}[\hat{N}] = \int_0^{h_r(f)} \frac{dh}{\sqrt{1 - f_p^2(h)/f^2}}, \tag{4}$$

where $f_p(h) = \sqrt{\hat{N}(h)/1.24 \times 10^{10}}$ MHz is the local plasma frequency and $h_r(f)$ is the reflection height where $f_p = f$. The integral is computed by trapezoidal quadrature on the fixed height grid, which is trivially differentiable via automatic differentiation in PyTorch.

4 Two-Stage Training Strategy

4.1 Stage 1: Global foundation model (IRI pre-training)

The foundation model is trained on synthetic profiles generated by IRI-2020 [5] across a parameter grid:

- Latitude/longitude: global 5° × 5° grid (∼2,500 locations)
- Day-of-year: 12 representative days per year (seasonal coverage)
- Solar flux: $F_{10.7} \in \{70, 100, 130, 160, 200\}$ SFU

- Geomagnetic activity: $K_p \in \{0, 1, 2, 3, 4, 5, 6, 7\}$
- Universal time: 0, 6, 12, 18 UT

Totalling ∼5 million unique ionospheric states. For each state, $N(h)$ is drawn from IRI-2020 and the corresponding ionogram trace $h'(f)$ is computed via Eq. (4).

The Stage 1 loss combines supervised regression on the IRI profile with the physics consistency term:

$$\mathcal{L}_1 = \underbrace{\|\hat{N} - N_{\text{IRI}}\|^2}_{\text{supervised}} + \lambda_1 \underbrace{\|\mathcal{H}[\hat{N}] - \mathbf{h}'\|^2}_{\text{physics}} + \lambda_2 \underbrace{\textstyle\sum_h \text{ReLU}(-\Delta\hat{N})}_{\text{monotone}}, \tag{5}$$

where the monotone penalty discourages non-physical decreases in $N(h)$ above the F2 peak.

4.2 Stage 2: Station-specific fine-tuning (data assimilation)

The foundation model is adapted to a specific ionosonde by fine-tuning on real (unannotated) soundings using *no labeled true-height profiles*. The Stage 2 loss retains only the physics and prior terms:

$$\mathcal{L}_2 = \underbrace{\|\mathcal{H}[\hat{N}] - \mathbf{h}'_{\text{obs}}\|^2}_{\text{obs. cost}} + \mu \underbrace{D_{\text{KL}}(q_\phi \| p_\theta)}_{\text{background cost}} + \lambda_2 \underbrace{\textstyle\sum_h \text{ReLU}(-\Delta\hat{N})}_{\text{monotone}}, \tag{6}$$

where p_θ is the distribution over $N(h)$ implied by the Stage 1 foundation model and q_ϕ is the distribution induced by the fine-tuned network. The KL term penalizes departure from the IRI-learned prior — formally equivalent to the *background cost* $J_b = (x - x_b)^T B^{-1}(x - x_b)$ in variational data assimilation [6].

The mapping to 4D-Var is exact:

4D-Var concept	This framework
Background x_b	Stage 1 prior p_θ
Observation y	Real ionogram $\mathbf{h}'_{\text{obs}}$
Obs. operator $\mathcal{H}$	Abel integral (Eq. 4)
Bg. covariance B	FiLM prior variance
Analysis x_a	Fine-tuned $\hat{N}(h)$

Only the decoder and the last encoder block are updated in Stage 2; the early convolutional layers and conditioning MLP are frozen to preserve physical priors learned from IRI. Convergence for a single station archive (∼1,000 soundings) is reached in under 30 minutes on a single GPU.

5 Integration in `pynasonde`

5.1 Zero-dependency inference

After training, all weights are exported as a single `.npz` file containing plain NumPy arrays. Inference requires only NumPy — no PyTorch, no ONNX Runtime, no GPU. Station-specific weight files (e.g., `wi937_v1.npz`, `pl407_v1.npz`) ship with the package.

5.2 Drop-in API replacement

`NNTrueHeightInversion` exposes an identical interface to the existing `TrueHeightInversion` (POLAN) class:

```
# Existing POLAN
edp = TrueHeightInversion(
    monotone_enforce=True
).fit_from_df(o_df)

# NN - same call, same output
edp = NNTrueHeightInversion(
    weights="wi937_v1",
    conditioning={"Kp": 2.3,
                  "F107": 145.0}
).fit_from_df(o_df)

print(edp.foF2_mhz, edp.hmF2_km)
edp.plot()  # identical EDPResult
```

The returned `EDPResult` dataclass is identical to the POLAN output, ensuring backward compatibility with all downstream analysis modules (`IonogramScaler`, `AbsorptionAnalyzer`, etc.).

5.3 Folder layout

```
nn_inversion/
  inversion_nn.py   # NNTrueHeightInversion
  forward_model.py  # numpy Abel integral
  network.py        # numpy inference
  weights/
    global_v1.npz
    wi937_v1.npz
    pl407_v1.npz
  training/         # torch-only
    architecture.py
    film_conditioning.py
    physics_loss.py
    synthetic_data.py
```

```
trainer_stage1.py
trainer_stage2.py
```

6 Comparison with Prior Work

Method	No labels	Global prior	Phys. loss	Open src
POLAN [1]	✓	–	–	–
QP-seg. [2]	✓	–	–	✓
CNN [3]	–	–	–	–
VAE [4]	–	–	–	–
This work	✓	✓	✓	✓

The key differentiators are: (1) no labeled true-height profiles are needed at any real station; (2) a single model serves the global ionosphere via geophysical conditioning; (3) physics constrains *every* gradient step, not just post-hoc validation; (4) full open-source deployment.

7 Development Roadmap

1. **Forward model** (`forward_model.py`) — differentiable Abel integral in NumPy and PyTorch; unit-tested against POLAN on synthetic profiles. *[Target: pynasonde v1.5]*
2. **Synthetic data pipeline** — IRI-2020 batch generation across global grid, 30-year parameter sweep, parallel execution. *[Target: pynasonde v1.5]*
3. **Stage 1 training** — FiLM-conditioned CNN+MLP on IRI data, validation against POLAN on held-out locations. *[Target: pynasonde v1.6]*
4. **Stage 2 fine-tuning** — physics-only loss on WI937 and PL407 archives; ablation study of KL weight μ. *[Target: pynasonde v1.6]*
5. **Inference integration** — `NNTrueHeightInversion` class, weight export, API validation, shipped `.npz` files. *[Target: pynasonde v2.0]*
6. **Journal paper** — comparison of POLAN vs. NN on APEP2 eclipse data, ionospheric storm case studies, uncertainty quantification. *[Target: Radio Science / JGR-Space Physics]*

8 Discussion

Why publish before pynasonde 2.0? Circulating this design now serves three purposes: (i) establishing scientific priority for the IRI-pretraining + physics-only fine-tuning combination; (ii) gathering community feedback on architecture choices before implementation is locked; (iii) inviting collaborators with access to diverse ionosonde archives (GIRO network, SuperDARN-adjacent stations) to contribute fine-tuning datasets.

Uncertainty quantification. Stage 2's variational framing naturally supports posterior uncertainty estimates: the spread of the posterior distribution q_ϕ over $N(h)$ provides height-resolved confidence intervals on the inversion — something POLAN cannot provide. This is directly useful for the APEP2 eclipse campaign, where model–data comparisons require quantified uncertainty.

Extensibility. The FiLM conditioning vector is not fixed. Future versions can condition on magnetic dip angle, solar zenith angle, proximity to terminator, or assimilated TEC from GNSS, extending the framework toward a full ionospheric data assimilation system.

References

[1] Titheridge, J. E. (1985). *Ionogram Analysis with the Generalized Program POLAN*. Report UAG-93, World Data Center A.

[2] Reinisch, B. W., & Huang, X. (1983). Automatic calculation of electron density profiles from digital ionograms, 1. Automatic O and X trace identification for topside ionograms. *Radio Science*, 18(3), 477–492.

[3] Jiang, C., et al. (2022). Inferring the true-height electron density profile from ionograms using a convolutional neural network. *Radio Science*, 57, e2021RS007406.

[4] Eyfells, A., et al. (2022). Variational autoencoder approach for ionosonde true-height inversion. *Journal of Geophysical Research: Space Physics*, 127, e2021JA030132.

[5] Bilitza, D., et al. (2022). The International Reference Ionosphere model: A review and description of an ionospheric benchmark. *Reviews of Geophysics*, 60, e2022RG000792.

[6] Lorenc, A. C. (1986). Analysis methods for numerical weather prediction. *Quarterly Journal of the Royal Meteorological Society*, 112, 1177–1194.

[7] Perez, E., et al. (2018). FiLM: Visual Reasoning with a General Conditioning Layer. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32.

[8] Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems. *Journal of Computational Physics*, 378, 686–707.

[9] Reinisch, B. W., & Galkin, I. A. (2011). Global ionospheric radio observatory (GIRO). *Earth, Planets and Space*, 63, 377–381.