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 15, 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* (NN-POLAN) is pre-trained on ∼5 million synthetic electron density 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, universal time, 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 at any real station. The physics loss is formulated via *Abel inversion* of the observed virtual-height trace, which is both singularity-free and gradient-stable, resolving a fundamental training instability inherent in forward-Abel-based loss functions. The fine-tuning objective is formally equivalent to variational data assimilation (4D-Var), with the IRI-trained prior as the background state. The trained model is integrated into the open-source `pynasonde` framework as a drop-in replacement for `TrueHeightInversion`, requiring no additional dependencies for inference.

1 Introduction

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

$$h'(f) = h_{\text{base}} + \int_{h_{\text{base}}}^{h_r(f)} \mu'(f,\, N(h))\, dh, \quad (1)$$

where $\mu' = [1 - f_p^2(h)/f^2]^{-1/2}$ is the O-mode group refractive index, $f_p(h) = \sqrt{N(h)/1.2441 \times 10^4}$ MHz is the plasma frequency, and $h_r(f)$ is the reflection height where $f_p = f$. Solving Eq. (**??**) for $N(h)$ given noisy, sparse observations of $h'(f)$ is an ill-posed inverse problem that has been the subject of research for over 60 years.

Classical algorithms — POLAN [?], NHPC, and the quasi-parabolic approach [?] used in ARTIST/SAO systems — rely on analytical profile families (Chapman, parabolic) or lamination. 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 [? ?], but they share a common limitation: they are trained on POLAN or other algorithm output as ground truth, inheriting 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, with explicit documentation of the non-trivial training engineering challenges encountered and how they were resolved.

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 a single station or season.
2. **Singularity-free Abel inversion loss**: a physics supervision signal derived by analytically inverting the observed $h'(f)$ trace, bypassing the $1/\sqrt{f^2 - f_p^2}$ singularity that makes forward Abel losses untrainable.
3. **Physics-only fine-tuning** on real data: the network adapts to a specific ionosonde using only the Abel inversion as supervision, requiring zero labeled true-height profiles.
4. A rigorous **data assimilation framing** of the fine-tuning step, with the IRI prior as the background state in 4D-Var.
5. **Geophysical conditioning via FiLM** [?] so that a single network parameterizes the full range of ionospheric states from solar minimum quiet to solar maximum storm.
6. **Spatial-latent encoder** that preserves frequency-position information through the bottleneck via flatten rather than global pooling, enabling well-conditioned Abel gradients throughout the network.

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 $N_f = 141$ frequencies, 1.0–15.0 MHz at 0.1 MHz step) and a geophysical conditioning vector $\mathbf{c} = [\phi,\, \lambda,\, \mathrm{DOY},\, \mathrm{UT},\, K_p,\, F_{10.7}]$ to an electron density profile $\hat{N}(h) \in \mathbb{R}^{N_h}_{>0}$:

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

The architecture consists of four components.

(i) Conditioning MLP (FiLM generator). $\mathbf{c} \to$ shared two-layer MLP ($6 \to 128 \to 128$, SiLU) $\to$ per-layer projection heads, each outputting $(\boldsymbol{\gamma}_\ell, \boldsymbol{\beta}_\ell)$ for the corresponding encoder or decoder block. Two separate FiLM generators are instantiated: one for the encoder and one for the decoder.

(ii) 1-D CNN encoder. Eight convolutional blocks applied to $\mathbf{h}'$, with FiLM modulation (multiplicative residual form) after each BatchNorm:

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

Four of the eight blocks use stride-2 convolutions, progressively halving the sequence length: $N_f = 141 \to 71 \to 36 \to 18 \to 9$. Channel progression: $1 \to 32 \to 64 \to 128$ (held at 128 for the last four blocks). The final feature map $(B, 128, 9)$ is *flattened* to $(B, 1152)$ and projected through a linear layer to the latent vector $\mathbf{z} \in \mathbb{R}^{256}$.

(iii) MLP decoder. A symmetric transposed-convolution stack expanding $\mathbf{z}$ to a spatial sequence of length 144, followed by a $1{\times}1$ projection layer and linear interpolation to $N_h = 904$ output points. The height grid spans 60–511.5 km at 0.5 km step ($N_h = 904$). Softplus activation enforces $\hat{N}(h) \geq 0$. FiLM modulation is applied after each BatchNorm in the decoder as well.

The total model has ~1.43 million trainable parameters.

3.2 Why flatten, not pool?

A critical design decision — and one that was not obvious *a priori* — is replacing global-average-pooling with spatial flattening at the encoder bottleneck.

The Abel integral has a fundamentally frequency-dependent gradient structure: near foF2 the integrand diverges and gradients are large; at low frequencies they are small and smooth. After 4 stride-2 layers the encoder retains a 9-point spatial feature map where each spatial position corresponds to a different frequency band. A global-average-pool collapses this to a single vector, uniformly mixing information from all frequency bands. As a result, the gradient from the Abel physics loss — which originates at specific frequency cells — is averaged away before it can reach the weights responsible for those frequency bands. We observed this as complete failure of the Abel loss to decrease during training (convergence diagnostic: Config B flat/increasing, confirmed across MSE, log, and cumulative Abel loss formulations).

Replacing the pool with $\mathbf{x}$.flatten(1) preserves the frequency-position ordering in the latent vector, enabling Abel-loss gradients to flow directly back to the specific encoder weights that processed the corresponding frequency bands.

3.3 Differentiable forward model

For validation and self-consistency diagnostics, the Abel integral operator $\mathcal{H}$ in Eq. (**??**) is implemented as a differentiable PyTorch function (`torch_forward_batch` in `physics_loss.py`):

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

computed by trapezoidal quadrature on the fixed height grid.

This operator is retained in the codebase for use in convergence diagnostics (`diagnose_physics.py`) and inference visualization (`test_inference.py`). It is *not* used in the training gradient path — see Section **??** for why, and how this was resolved.

3.4 Abel inversion operator

The substitution $f = f_p \sin\theta$ in the Abel integral identity yields a singularity-free expression for the reflection height:

$$h_r(f_p) = \frac{2}{\pi} \int_0^{\pi/2} h'(f_p \sin\theta)\, d\theta \;\approx\; \frac{1}{N_q} \sum_{k=1}^{N_q} h'(f_p \sin\theta_k)\,, \tag{5}$$

where $\theta_k = (k - \frac{1}{2})\pi/(2N_q)$ are midpoint-rule quadrature nodes ($N_q = 64$ in practice). At the reflection height $h_r(f_p)$, the plasma frequency equals f_p by definition, so:

$$\hat{N}_{\text{abel}}(h_r(f_p)) = f_p^2 \times 12441 \text{ cm}^{-3}, \tag{6}$$

where the constant $12441 \text{ cm}^{-3}\,\text{MHz}^{-2}$ converts MHz^2 to electron density. Interpolating the set of $\{(h_r(f_p),\, \hat{N}_{\text{abel}}(h_r))\}$ pairs onto $H_{\text{GRID\_KM}}$ gives the *Abel-inverted Ne target* for the observed bottomside.

Crucially, the integrand in Eq. (**??**) is $h'(f_p \sin\theta)$, which samples the observed virtual-height trace at sub-critical frequencies where $f_p \sin\theta \le f_p < f$ everywhere — the $1/\sqrt{f^2 - \xi^2}$ singularity of the classical kernel is entirely absent.

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 [**?**] across a parameter grid:

- Latitude / longitude: global 5°×5° grid (∼2,500 locations)
- Years: 1995–2024 (30 years, temporal split below)
- Day-of-year: 9 representative days per year (45-day step)
- Universal time: 0, 6, 12, 18 UT
- Solar flux and K_p: real OMNI values from `pyomnidata`

Totalling ∼5 million unique ionospheric states. For each state, IRI-2020 provides $N(h)$; the corresponding virtual-height trace $h'(f)$ is computed by the NumPy forward batch model (`forward_batch` in `forward_model.py`). Profiles failing quality filters (foF2 $\notin$ [1.5, 15.0] MHz or fewer than 5 valid frequency cells) are discarded.

Data are partitioned temporally to avoid leakage: training $\leq$2016, validation 2017–2020, test $>$2020.

The Stage 1 loss combines supervised regression on the IRI profile with an Abel inversion consistency term and a topside monotonicity regulariser:

$$\begin{aligned}\mathcal{L}_1 = \lambda_{\text{bg}} & \underbrace{\|\hat{N}_n - N_n^{\text{IRI}}\|^2}_{\text{background (IRI prior)}} \\ & + \lambda_\phi \underbrace{\frac{1}{|\mathcal{B}|}\sum_{h\in\mathcal{B}}\Big(\log_{10}\hat{N}(h) - \log_{10}\hat{N}_{\text{abel}}(h)\Big)^2}_{\text{Abel inversion physics}} \\ & + \lambda_m \underbrace{\frac{1}{N_h}\sum_{h}\text{ReLU}(\Delta f_p(h)+\epsilon)\cdot\mathbf{1}[h>h_{\text{peak}}]}_{\text{topside monotone regulariser}},\end{aligned} \tag{7}$$

where $\hat{N}_n$ is the network output in normalised log-space, $\mathcal{B}$ is the set of height grid points within the observed bottomside (determined by the Abel inversion, Eq. **??**), and the monotone term penalises upward plasma-frequency excursions only above the F2 peak height. Default weights: $\lambda_{\text{bg}} = 1.0$, $\lambda_\phi = 10^{-4}$ (ramped linearly from 0 over 10 post-warmup epochs), $\lambda_m = 0.1$.

A *warmup curriculum* runs the background-only loss ($\lambda_\phi = \lambda_m = 0$) for 10 epochs to establish a non-degenerate Ne prior before enabling the Abel and monotone terms. Without warmup, a near-zero initial Ne pushes the forward integral to the top of the grid, and the background term is needed to rescue the profile into a physically plausible range first.

4.2 Training challenges and their resolution

Getting each loss term to decrease simultaneously was non-trivial and required two distinct architectural and algorithmic interventions. We document them explicitly because they are relevant to any physics-informed inversion network operating near integral singularities.

Challenge 1: Global-average-pooling destroys Abel gradients. In the initial architecture, the encoder bottleneck used `AdaptiveAvgPool1d(1)`, reducing the spatial feature map $(B, 128, 9)$ to a single vector $(B, 128)$ before projection. The Abel loss gradient with respect to a given frequency band is localised in spatial position after stride-2 downsampling. Global average pooling uniformly mixes gradients from all nine spatial positions, attenuating the frequency-specific Abel gradient by approximately 1/9 while mixing it with gradients from unrelated frequency bands. The net effect was that Config B (Abel-only training, convergence diagnostic) showed the loss as flat or increasing across all three Abel loss formulations tested (MSE, log-ratio, and cumulative frequency integral).

Resolution: replace `AdaptiveAvgPool1d(1)` with `flatten(1)`, concatenating the full $(B, 128, 9)$ map to $(B, 1152)$ and projecting via a $1152 \rightarrow 256$ linear layer. This preserves the spatial ordering of frequency-band information in the latent vector, allowing Abel gradients to propagate directly back to the encoder weights that processed each frequency band.

Challenge 2: Forward Abel loss has a gradient singularity near foF2. Even with the spatial latent fix, training with the forward Abel loss $\|\mathcal{H}[\hat{N}] - \mathbf{h}'\|^2$ was partially unstable. The integrand $\mu' = [1 - f_p^2(h)/f^2]^{-1/2}$ diverges as $f_p \rightarrow f$ (i.e., near foF2), producing group refractive index values up to $\mu'_{\max} = 50$ on a discrete grid. The gradient $\partial\mathcal{H}/\partial N \propto \mu'^2$ is therefore up to 2500× larger at near-foF2 heights than at low-frequency heights. In mixed training, this dominates the background-loss gradient and drives the Ne profile toward collapse in the early physics epochs. Clamping μ' at 50 reduced but did not eliminate the instability.

Resolution: replace the forward Abel loss entirely with the Abel inversion loss (Eq. **??**–**??**). The observed $h'(f)$ is analytically inverted to a Ne target $\hat{N}_{\text{abel}}$ with no forward model in the computational graph. The training loss is then a $\log_{10}$-space MSE between the network prediction and this fixed target on the observed bottomside:

$$\mathcal{L}_{\text{abel}} = \frac{1}{|\mathcal{B}|}\sum_{h\in\mathcal{B}}\Big(\log_{10}\hat{N}(h) - \log_{10}\hat{N}_{\text{abel}}(h)\Big)^2. \tag{8}$$

Because $\hat{N}_{\text{abel}}$ is pre-computed with `torch.no_grad()`, the Abel singularity is entirely absent from the gradient graph. The $\log_{10}$ metric is scale-invariant across 4+ orders of magnitude of Ne and yields gradients of $\mathcal{O}(1)$ at convergence.

A convergence diagnostic tool (`diagnose_convergence.py`) runs four independent gradient-step experiments from the same checkpoint: background-only (Config A), Abel-only (Config B), monotone-only (Config C), and Abel+BG (Config D). A healthy Config B shows a **DECREASING** Abel loss; a pool-based encoder or forward Abel loss produces **FLAT/INCREASING**.

4.3 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. Real virtual-height observations $\mathbf{h}'_{\text{obs}}$ from filtered echo DataFrames (RIQ files via `EchoExtractor` + `IonogramFilter`) are binned into the same $N_f = 141$ frequency grid and Abel-inverted using Eq. (**??**) to produce the observation-based Ne target.

The Stage 2 loss retains only the physics and monotone terms:

$$\mathcal{L}_2 = \underbrace{\lambda_\phi \frac{1}{|\mathcal{B}|} \sum_{h\in\mathcal{B}} \Big(\log_{10}\hat{N}(h) - \log_{10}\hat{N}_{\text{abel}}(h)\Big)^2}_{\text{Abel inversion obs. cost}} + \underbrace{\lambda_m \frac{1}{N_h} \sum_{h} \text{ReLU}(\Delta f_p + \epsilon)\cdot \mathbf{1}[h > h_{\text{peak}}]}_{\text{topside monotone}}, \tag{9}$$

with no background term ($\lambda_{\text{bg}} = 0$) — the Stage 1 prior is implicitly encoded in the initial weights and preserved by the low fine-tuning learning rate.

The mapping to 4D-Var is exact:

4D-Var concept	This framework
Background x_b	Stage 1 prior (IRI-trained weights)
Observation y	Real ionogram $\mathbf{h}'_{\text{obs}}$
Obs. operator	Abel inversion (Eq. **??**)
Bg. covariance B	FiLM prior, fine-tuning LR
Analysis x_a	Fine-tuned $\hat{N}(h)$

During Stage 2, the FiLM generator parameters can optionally be frozen (`--freeze_film`) to preserve the global geophysical conditioning learned from IRI; only the encoder/decoder weights are updated. 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`) will ship with the package.

5.2 Drop-in API replacement

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

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

# NN-POLAN   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/
  config.py          # typed NNCfg dataclass
  forward_model.py   # F_GRID_MHZ, H_GRID_KM,
                     #   forward_scalar/batch
  inversion_nn.py    # NNTrueHeightInversion [
    planned]
  network.py         # NumPy-only inference   [
    planned]
  training/
    architecture.py         # NNPolan (1.43 M
    params)
    physics_loss.py         # PhysicsLoss,
```

```
                          #
  abel_invert_batch,
                          #
  torch_forward_batch
  synthetic_data.py       # IRI shard
  generation
  trainer_stage1.py       # Stage-1 training
  loop
  trainer_stage2.py       # Stage-2 real-echo
  fine-tuning
  diagnose_convergence.py # per-loss gradient
  health check
  diagnose_physics.py     # forward-Abel self-
  consistency
  test_inference.py       # 3-panel visual
  validation
weights/                  # [planned, post-
  training]
  global_v1.npz
  wi937_v1.npz
```

6 Comparison with Prior Work

Method	No labels	Global prior	Phys. loss	Open src
POLAN [?]	✓	–	–	–
QP-seg. [?]	✓	–	–	✓
CNN [?]	–	–	–	–
VAE [?]	–	–	–	–
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 FiLM conditioning; (3) the Abel inversion physics loss is gradient-stable across all frequencies including near-foF2; (4) full open-source deployment.

7 Development Roadmap

1. **Synthetic data pipeline** — IRI-2020 batch generation across global grid, 30-year parameter sweep, temporal train/val/test split. `synthetic_data.py`. *[Complete]*
2. **Stage 1 training** — FiLM-conditioned CNN encoder-decoder on IRI data; Abel inversion loss + warmup curriculum. Full run on VEGA HPC (V100/A100 nodes, 270 shards, 50–100 epochs). *[In progress]*
3. **Stage 2 fine-tuning** — physics-only loss on WI937 and PL407 archives. *[Planned, pending Stage 1 best.pt]*
4. **Inference integration** — NumPy-only `NNTrueHeightInversion` class (`network.py`, `inversion_nn.py`), weight export (`.npz`), API validation. *[Planned]*
5. **Sporadic-E extension** — dedicated E-region module capable of representing thin ionized layers at 90–130 km. Requires (i) synthetic Es profile generation (Gaussian peak on IRI background), (ii) monotone regulariser restricted to F-region only ($h > 150$ km), (iii) Es-aware Abel inversion splitting the trace at the blanketing frequency f_bE_s, and (iv) possible dual-decoder branch for Es parameters (hE_s, foE_s, $\mathrm{f_bE_s}$). *[Future, v2.0]*
6. **Journal paper** — comparison of POLAN vs. NN-POLAN on APEP2 eclipse data, ionospheric storm case studies, uncertainty quantification. *[Target: JGR: Machine Learning and Computation / Radio Science]*

8 Discussion

Why the Abel inversion loss is a non-trivial contribution. The combination of a near-singular forward operator and a deep feature-space bottleneck creates a training regime that fails silently: the network converges on the background loss, producing visually plausible Ne profiles, but the Abel physics constraint is effectively inactive (gradient magnitude near foF2 overwhelms or the pool layer severs it entirely). Standard remedies — learning-rate tuning, loss reweighting, gradient clipping — do not resolve the root cause. The Abel inversion loss is the structural fix: by moving the singularity outside the computational graph entirely, gradients are well-conditioned at every frequency up to foF2.

Why publish before `pynasonde` v2.0? Circulating this design now serves three purposes: (i) establishing scientific priority for the IRI-pretraining + Abel-inversion physics supervision combination; (ii) gathering community feedback on architecture choices before implementation is locked; (iii) inviting collaborators with diverse ionosonde archives (GIRO network, SuperDARN-adjacent stations) to contribute Stage 2 fine-tuning datasets.

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

Extensibility. The FiLM conditioning vector is not fixed: future versions can condition on magnetic dip angle, solar zenith angle, assimilated TEC from GNSS, or absorption measurements from riometers, extending the framework toward a full ionospheric data assimilation system. Sporadic-E support is the most immediate extension target (Section 7, item 5), requiring relatively contained changes to the data generation, loss function, and decoder, while keeping the foundation model architecture unchanged.

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.