Analysis of Galactic Rotation Curves using Godelian Logical Flow Models and Massless Topological Effect Models

Paul Chun-Kit Lee, MD
dr.paul.c.lee@gmail.com
Division of Cardiology, New York University, New York

September 2, 2024

Abstract

This paper presents an analysis of galactic rotation curves using novel models, including the Godelian Logical Flow (GLF) framework and Richard Lieu's massless topological defect models. The study utilizes data from the HI Nearby Galaxy Survey (THINGS) to test these models' ability to describe the observed rotation curves without invoking dark matter. The GLF framework, which incorporates concepts from Gödel's incompleteness theorems, offers an alternative explanation for galactic dynamics by suggesting that the logical structure of spacetime influences these phenomena. Lieu's model, on the other hand, proposes that massless topological defects in the form of concentric spherical shells can account for the flat rotation curves typically attributed to dark matter.

We demonstrate that the Lieu Multi-Shell model and the GLF3 Exponential model consistently outperform the standard dark matter model across a sample of galaxies. The results indicate statistically significant differences in model performance, with the alternative models providing incremental improvements that capture important aspects of galactic dynamics.

This study highlights the potential of these alternative models to provide new insights into the nature of galactic rotation curves and challenges the traditional dark matter paradigm. The findings encourage further theoretical development and observational validation of these models, with implications for our understanding of cosmic structure formation and dynamics.

Introduction

The study of galactic rotation curves has been a cornerstone in our understanding of cosmic structure and dynamics. Traditional models often invoke dark matter to explain the observed flat rotation curves of galaxies. However, alternative theories have emerged that challenge this paradigm. This paper explores two such novel approaches: the Godelian Logical Flow (GLF) framework [18] and Richard Lieu's massless topological defect models [21].

The GLF framework, inspired by Gödel's incompleteness theorems [9], proposes that the logical structure of spacetime itself may influence galactic dynamics. This approach suggests that incompleteness in our physical theories might manifest as apparent anomalies in galactic rotation curves, potentially offering an alternative to dark matter [18].

Richard Lieu's model [21], on the other hand, introduces the concept of massless topological defects in the form of concentric spherical shells. This innovative approach proposes that these defects could account for the flat rotation curves typically attributed to dark matter, without requiring additional mass [21].

Our study utilizes data from the HI Nearby Galaxy Survey (THINGS) to test these models' ability to describe observed rotation curves. THINGS provides high-resolution 21-cm HI observations of 34 nearby galaxies, offering precise measurements of rotation curves that serve as an ideal testbed for these theories.

Through rigorous statistical analysis, including model fitting and comparison using the Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC), we evaluate the performance of these models against the standard dark matter paradigm. Our aim is to assess whether these alternative approaches can provide new insights into galactic dynamics and challenge our current understanding of cosmic structure.

Mathematical Derivations for Modeling HI Nearby Galaxy Survey

1. GLF Models

The Godelian Logical Flow (GLF) framework proposes that the logical structure of spacetime, as inspired by Gödel's incompleteness theorems, may influence galactic dynamics. This approach has evolved through several iterations:

a) Original GLF model

The initial GLF model was formulated as:

$$v(r) = \sqrt{\alpha c^2 \left(1 + \beta \log(1+r) G(r) + \delta (1+r)^\gamma G(r)\right)}$$

where

$$G(r) = G_0 \exp\left(-\frac{kr^2}{1+r^2}\right)$$

Here, $v(r)$ represents the rotational velocity at radius r, c is the speed of light, and α, β, δ, γ, G_0, and k are parameters to be fitted. The function $G(r)$ represents a Gödelian factor, introducing incompleteness effects into the model.

b) Improved GLF model

An enhanced version of the GLF model was later developed:

$$v(r) = v_0 \sqrt{\left(1 + \left(\frac{r}{r_s}\right)^\alpha\right)^{-\beta} \cdot \left[\left(1 + \frac{r}{r_l}\right)^\gamma\right]^\mu + \left[\left(1 + \frac{r}{r_f}\right)^\delta\right]^\epsilon \cdot \nu}^{\frac{1}{\xi}}$$

This model introduces additional parameters (v_0, r_s, r_l, r_f, α, β, γ, δ, μ, ϵ, ν, ξ) to capture more complex dynamics. The structure allows for different scaling behaviors at various radii, potentially accounting for diverse galactic features.

c) GLF3 Models (Three Flavors)

The latest iteration of GLF models, known as GLF3, comes in three variants:

Exponential:

$$v(r) = v_0 \sqrt{\alpha \exp\left(-\frac{r}{r_s}\right) + \beta \left(\frac{r}{r_s}\right)^\gamma}$$

Logarithmic:

$$v(r) = v_0 \sqrt{\alpha \log\left(1 + \frac{r}{r_s}\right) + \beta \left(\frac{r}{r_s}\right)^\gamma}$$

Polynomial:

$$v(r) = v_0 \sqrt{\alpha \left(\frac{r}{r_s}\right) + \beta \left(\frac{r}{r_s}\right)^2 + \gamma \left(\frac{r}{r_s}\right)^3 + \delta \left(\frac{r}{r_s}\right)^4}$$

These models combine different functional forms to capture various aspects of galactic rotation curves. The exponential term could represent inner galactic regions, while power-law terms might describe outer regions. The logarithmic variant introduces a gradual transition between these regimes.

2. Lieu Models

Richard Lieu's approach introduces massless topological defects as an alternative explanation for galactic rotation curves. His model has been implemented in several forms:

a) Theoretical Foundation

Lieu's model starts with the gravitational Poisson equation in an isotropic environment:

$$\nabla^2\Phi = \frac{1}{r^2}\frac{d}{dr}\left(r^2\frac{d\Phi}{dr}\right) = 4\pi G\rho(r)$$

where Φ is the gravitational potential, G is the gravitational constant, and $\rho(r)$ is the mass density.
Lieu introduces a novel source term $\rho(r)$ that combines a Dirac delta function and its derivative:

$$\rho(r) = \frac{c^2}{8\pi G}\left[\frac{2\alpha s}{r^2}\delta(R-r) + \frac{2\alpha s}{r}\delta'(R-r)\right]$$

where c is the speed of light, α is a dimensionless constant, s is a length scale ($s \to 0^+$), and R is the radius of the shell.

This configuration produces an inward radial force on a test particle of:

$$F = -\frac{d\Phi}{dr} = -\frac{\alpha s c^2}{r}\delta(R-r) = -\frac{GM}{r}\delta(R-r)$$

where $M = \frac{\alpha s c^2}{G}$ is defined for notational convenience but does not represent actual mass.

b) Simple Lieu model

Based on this theoretical foundation, a simplified model for rotation curves was developed:

$$v(r) = \sqrt{\frac{\alpha c^2}{r}}\exp\left(-\frac{r}{s}\right)$$

This model combines the $1/r$ dependence from the force equation with an exponential decay term.

c) Strict Lieu model

A more stringent interpretation of Lieu's theory led to:

$$v(r) = \sqrt{\alpha c^2}\frac{\exp\left(-\frac{(r-R)^2}{2\cdot 0.01^2}\right)}{0.01\sqrt{2\pi}}$$

This model represents a single shell at radius R with a Gaussian profile, closely adhering to the delta function in the original theory.

d) Multi-shell Lieu model

Extending the concept to multiple shells:

$$v(r) = \sqrt{\alpha c^2}\sum_{R_i\in\{R_1,R_2,R_3\}}\exp\left(-\frac{(r-R_i)^2}{2\cdot \text{width}^2}\right)$$

This model allows for multiple concentric shells, each contributing to the rotation curve.

e) Advanced Lieu model

The most sophisticated implementation incorporates multiple shells with increasing spacing, a logarithmic gravitational potential, and a simplified baryonic component. This model aims to capture the full complexity of Lieu's theory while remaining computationally tractable for fitting to observational data.

These models represent different interpretations and implementations of Lieu's theoretical framework, allowing for a comprehensive exploration of its ability to explain galactic rotation curves.

Methods: Data and Model Implementation

Our study utilizes data from the HI Nearby Galaxy Survey (THINGS) [28], which provides high-resolution 21-cm HI observations of 0 nearby galaxies. The THINGS dataset offers precise measurements of rotation curves, making it an ideal testbed for comparing different models of galactic dynamics.

The Godelian Logical Flow (GLF) models [18] and Richard Lieu's massless topological defect models [21] were implemented as Python functions and fitted to the data using the *scipy.optimize.curve_fit* function. These models were compared to the standard dark matter paradigm using Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) to evaluate their performance across galaxies [1, 2].

Implementation in the Program

The code implements these models as Python functions, allowing for fitting to the THINGS data. Key aspects of the implementation include:

1. **Data loading:** The code reads THINGS data for various galaxies, extracting radius, velocity, and velocity error information.
2. **Model fitting:** It uses scipy.optimize.curve fit to fit each model to the data, determining the best-fit parameters.
3. **Statistical analysis:** The code calculates Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) for model comparison.
4. **Comparative analysis:** The program compares the performance of different models across multiple galaxies, providing insights into which models best describe the observed rotation curves.

Result

The analysis of galaxy rotation curves using various models reveals several key findings:

1. Model Performance

- The Lieu Multi model emerges as the best performing Lieu model.
- GLF3 Exp (exponential) is the top-performing GLF model.

2. Comparative Analysis

(a) **Lieu Multi vs GLF3 Exp:**

- Statistically significant difference (p = 0.0000)
- Small effect size (Cohen's d = -0.48)
- Lieu Multi outperforms GLF3 Exp in 75% of galaxies (42/56)

(b) **Lieu Multi vs Standard Dark Matter (DM) model:**

- Statistically significant difference (p = 0.0000)
- Small effect size (Cohen's d = -0.36)
- Lieu Multi outperforms DM in 86.7% of galaxies (52/60)

(c) **GLF3 Exp vs DM:**

- Statistically significant difference (p = 0.0000)
- Small effect size (Cohen's d = -0.35)
- GLF3 Exp outperforms DM in 91.1% of galaxies (51/56)

3. Model Ranking (based on AIC and BIC)

- Lieu Multi consistently ranks first (AIC: 1.95, BIC: 1.75)
- GLF3 Exp ranks second (AIC: 3.05, BIC: 2.71)
- Lieu Advanced and GLF3 Log follow closely
- Standard DM model ranks relatively low (AIC: 6.78, BIC: 6.40)

Table 1: Model Ranking Based on AIC and BIC (lower ranks are better)

Model	**AIC Rank**	**BIC Rank**
Lieu Multi	1.95	1.75
GLF3 Exp	3.05	2.71
Lieu Advanced	4.37	3.75
GLF3 Log	4.27	3.83
GLF Improved	4.15	5.98
GLF Original	5.05	5.18
GLF3 Poly	6.47	6.53
DM	6.78	6.40
Lieu Strict	8.30	8.28
Lieu Simple	9.90	9.90

Discussion of Result

1. Model Performance

The Lieu Multi model and GLF3 Exp model consistently outperform other models, including the standard Dark Matter model. This suggests that these alternative approaches to understanding galactic dynamics merit serious consideration. The Lieu Multi model, based on massless topological defects, performs particularly well, indicating that it captures important aspects of galactic rotation curves not accounted for in traditional models.

2. Statistical Significance

The Wilcoxon tests show highly significant differences between the top models and the standard DM model ($p = 0.0000$ in all cases). This strongly suggests that these alternative models are not merely fitting noise but are capturing genuine patterns in the data.

3. Effect Sizes

While the differences between models are statistically significant, the effect sizes are small (Cohen's d ranging from -0.35 to -0.48). This indicates that while the improvements are consistent, they are not dramatically large. This could suggest that all models are capturing similar underlying physics, with the alternative models providing incremental improvements.

4. Consistency Across Galaxies

Both Lieu Multi and GLF3 Exp models outperform the DM model in a large majority of galaxies (86.7% and 91.1% respectively). This consistency across different galactic systems lends credibility to these alternative approaches and suggests they may be capturing universal aspects of galactic dynamics.

5. Model Complexity and Parsimony

The high performance of the Lieu Multi model, which ranks first in both AIC and BIC, is particularly noteworthy. AIC and BIC penalize model complexity, suggesting that the Lieu Multi model achieves its superior fit without undue complexity. This parsimony is a strong point in its favor.

Discussion of Ongoing works

The study of galactic dynamics and cosmic structure formation continues to challenge our understanding of the universe. Our analysis, which spans from large-scale cosmological observations to galactic rotation curves, reveals a complex picture that suggests the need for novel approaches to explain observed phenomena.

Cosmological Implications from BAO DESI Analysis

Our previous work on the BAO DESI data revealed intriguing results regarding the performance of different cosmological models. The Gödelian Logical Flow (GLF) model, which incorporates logical structures inspired by Gödel's incompleteness theorems, showed superior performance in fitting the BAO data compared to both the Ricci Flow model and the standard Lambda-CDM model. This suggests that incorporating concepts from logic and geometric flows into cosmological models may provide a more accurate description of large-scale structure formation and evolution.

The success of the GLF model at cosmological scales motivated us to explore its applicability to smaller-scale phenomena, particularly galactic dynamics. However, the transition from cosmological to galactic scales is not straightforward, and the model requires careful adaptation to address the specific challenges posed by galactic rotation curves.

Galactic Dynamics and Model Comparison

While the GLF model showed promise in cosmological applications, directly applying it to galactic rotation curves proved challenging. The complexity of galactic dynamics, including the interplay between baryonic matter, dark matter (if it exists), and potentially unknown physical phenomena, necessitated a more flexible approach.

Concurrently, Dr. Richard Lieu's model of massless topological defects offered an intriguing alternative explanation for galactic rotation curves. However, applying Lieu's model to the BAO DESI data would require significant extensions and modifications, potentially compromising its elegant simplicity and physical interpretability.

Given these considerations, we decided to focus our analysis on the THINGS (The HI Nearby Galaxy Survey) dataset. This high-resolution data on nearby galaxies provides an ideal testbed for comparing different models of galactic dynamics, including various iterations of the GLF model and Lieu's topological defect model.

Model Performance and Implications

Our analysis of the THINGS data reveals that both the GLF models (particularly the GLF3 Exp variant) and Lieu's models (especially the Lieu Multi model) outperform the standard dark matter model in explaining galactic rotation curves. This suggests that alternative approaches to galactic dynamics merit serious consideration.

The strong performance of Lieu's model, which posits massless topological defects as an explanation for flat rotation curves, is particularly noteworthy. Its ability to fit the data without invoking dark matter challenges the conventional wisdom and opens up new avenues for understanding galactic structure.

However, it's important to note that the success of these models does not necessarily mean they are mutually exclusive. The GLF approach, with its roots in logical structures and incompleteness, and Lieu's topological defect model may be capturing different aspects of a more complex underlying reality. The possibility that these models are complementary rather than competing should not be overlooked.

Implications for Dark Matter and Fundamental Physics

The success of alternative models in explaining galactic rotation curves without invoking dark matter raises important questions about the nature of dark matter and our understanding of fundamental physics. While these results don't necessarily negate the existence of dark matter, they suggest that the dark matter paradigm may need revision or that alternative explanations for galactic dynamics are viable.

Furthermore, the potential complementarity of the GLF and Lieu models hints at a deeper connection between logical structures, topological defects, and the fabric of spacetime. This connection could have profound implications for our understanding of gravity, quantum mechanics, and the fundamental nature of reality.

Conclusion

In conclusion, our work on both cosmological and galactic scales reveals the richness and complexity of cosmic structure formation and evolution. The success of alternative models in explaining observed phenomena challenges us to reconsider our fundamental assumptions about the universe and opens up exciting new avenues for theoretical and observational exploration.

References

[1] Alam, S., Ata, M., Bailey, S., *et al.* (2017). The Completed SDSS-IV extended Baryon Oscillation Spectroscopic Survey: Cosmological Implications from Two Decades of Spectroscopic Surveys at the Apache Point Observatory. *Monthly Notices of the Royal Astronomical Society*, 470(3), 2617-2652.

[2] Ahumada, R., Allende Prieto, C., Almeida, A., *et al.* (2020). The 16th Data Release of the Sloan Digital Sky Surveys: First Release of MaNGA Derived Quantities, Data Visualization Tools, and Stellar Library. *The Astrophysical Journal Supplement Series*, 249(1), 3.

[3] Awodey, S. (2010). *Category Theory* (2nd ed.). Oxford University Press.

[4] Baez, J. C., & Dolan, J. (1998). Higher-dimensional algebra and topological quantum field theory. *Journal of Mathematical Physics*, 36(11), 6073-6105.

[5] Cheng, E. (2015). *Cakes, Custard and Category Theory: Easy Recipes for Understanding Complex Maths*. Profile Books.

[6] Cubitt, T. S., Perez-Garcia, D., & Wolf, M. M. (2015). Undecidability of the spectral gap. *Nature*, 528(7581), 207-211.

[7] Forman, R. (1998). Morse theory for cell complexes. *Advances in Mathematics*, 134(1), 90-145.

[8] B. Fornal and B. Grinstein, "Dark Matter Interpretation of the Neutron Decay Anomaly," *Physical Review Letters*, vol. 120, no. 19, p. 191801, 2018.

[9] Gödel, K. (1931). Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. *Monatshefte für Mathematik und Physik*, 38(1), 173-198.

[10] Hofstadter, D. R. (1979). *Gödel, Escher, Bach: An Eternal Golden Braid*. Basic Books.

[11] Hou, J., Zhu, G. B., Tinker, J. L., *et al.* (2021). The Completed SDSS-IV extended Baryon Oscillation Spectroscopic Survey: BAO and RSD Measurements from Luminous Red Galaxies in the Final Sample. *Monthly Notices of the Royal Astronomical Society*, 500(1), 1201-1221.

[12] B. Koch and F. Hummel, "An Exciting Hint Towards the Solution of the Neutron Lifetime Puzzle?," *arXiv preprint arXiv:2403.00914*, 2024.

[13] Kitaev, A. (2003). Fault-tolerant quantum computation by anyons. *Annals of Physics*, 303(1), 2-30.

[14] Lawvere, F. W. (1963). Functorial semantics of algebraic theories. *Proceedings of the National Academy of Sciences*, 50(5), 869-872.

[15] Lee, P. C. K. (2024a). Higher Categorical Structures in Gödelian Incompleteness: Towards a Topos-Theoretic Model of Metamathematical Limitations. *viXra.org e-Print archive*, viXra:2407.0164.

[16] Lee, P. C. K. (2024b). The Geometry of Gödelian Categorical Singularities: A Refined Mathematical Framework for Incompleteness Phenomena (Part 2: Extending the Topological and Geometric Aspects). *viXra.org e-Print archive*, viXra:2408.0049.

[17] Lee, P. C. K. (2024c). Ricci Flow Techniques in General Relativity and Quantum Gravity: A Perelman-Inspired Approach to Spacetime Dynamics. *viXra.org e-Print archive*, viXra:2407.0165.

[18] Lee, P. C. K. (2024d). A Ricci Flow-Inspired Model for Cosmic Expansion: New Insights from BAO Measurements. *viXra.org e-Print archive*, https://vixra.org/abs/2408.0067.

[19] Lee, P. C. K. (2024e). Gödelian Index Theorem on Smooth Manifolds: Extending the Atiyah-Singer Framework and Its Cosmological Implications (Part 3 of Categorical Gödelian Incompleteness Series). *viXra.org e-Print archive*, https://vixra.org/abs/2408.0075.

[20] Lee, P. C. K. (2024f). Gödelian Index Theorem in Discrete Manifolds: A Unified Framework for Logical Complexity Across Cosmic and Quantum Scales (Part 4). *viXra.org e-Print archive*, http://viXra.org/abs/2408.0086.

[21] Lieu, R. (2024). The Binding of Cosmological Structures by Massless Topological Defects. *Preprint: MNRAS*, available at https://arxiv.org/abs/2406.04355.

[22] Lurie, J. (2009). *Higher Topos Theory*. Princeton University Press.

[23] Mac Lane, S., & Moerdijk, I. (1992). *Sheaves in Geometry and Logic: A First Introduction to Topos Theory*. Springer-Verlag.

[24] Penrose, R. (1989). *The Emperor's New Mind: Concerning Computers, Minds, and the Laws of Physics*. Oxford University Press.

[25] Ross, A. J., Samushia, L., Howlett, C., *et al.* (2017). The Clustering of the SDSS DR7 Main Galaxy Sample: A 4 per cent Distance Measure at $z = 0.15$. *Monthly Notices of the Royal Astronomical Society*, 464(1), 1168-1184.

[26] Sachdev, S. (2011). *Quantum Phase Transitions* (2nd ed.). Cambridge University Press.

[27] The Univalent Foundations Program. (2013). *Homotopy Type Theory: Univalent Foundations of Mathematics*. Institute for Advanced Study.

[28] Walter, F., Brinks, E., de Blok, W. J. G., Bigiel, F., Kennicutt, R. C. Jr., Thornley, M. D., Leroy, A. K. (2008). THINGS: The HI Nearby Galaxy Survey. *The Astronomical Journal*, 136(6), 2563-2647.

[29] Watson, J. D., Onorati, E., & Cubitt, T. S. (2021). Uncomputably complex renormalisation group flows. *arXiv preprint arXiv:2102.05145*.

[30] Wen, X. G. (2004). *Quantum Field Theory of Many-Body Systems: From the Origin of Sound to an Origin of Light and Electrons*. Oxford University Press.

[31] Witten, E. (1989). Quantum field theory and the Jones polynomial. *Communications in Mathematical Physics*, 121(3), 351-399.

[32] Wolfram, S. (2002). *A New Kind of Science*. Wolfram Media.

[33] A. T. Yue et al., "Improved Determination of the Neutron Lifetime," *Physical Review Letters*, vol. 111, no. 22, p. 222501, 2013.

Python Code

```
#!/usr/bin/env python3


# based on https://www2.mpia-hd.mpg.de/THINGS/Data.html, moment 1
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit
from astropy.table import Table
from astropy.io import fits
import warnings
import os
from tabulate import tabulate
from glob import glob
from astropy.io import fits
from astropy.wcs import WCS
import pandas as pd
```

```
from scipy import stats

warnings.filterwarnings('ignore')

# Constants
G = 6.67430e-11 # Gravitational constant in m^3 kg^-1 s^-2
c = 299792458  # Speed of light in m/s

# Original simplified Lieu model
def lieu_model_simple(r, alpha, s):
    return np.sqrt(alpha * c**2 / r) * np.exp(-r/s)

# Improved Lieu model based strictly on the paper
def lieu_model_strict(r, alpha, R):
    v = np.sqrt(alpha * c**2)
    delta = np.exp(-(r - R)**2 / (2 * 0.01**2)) / (0.01 * np.sqrt(2 * np.pi))
    return v * delta

# Practical multi-shell Lieu model
def lieu_model_multi(r, alpha, R1, R2, R3, width):
    v = np.sqrt(alpha * c**2)
    shell_effects = np.sum([np.exp(-(r - R)**2 / (2 * width**2)) for R in [R1, R2, R3]], axis=0)
    return v * shell_effects

def glf_model_original(r, alpha, beta, gamma, delta, G0, k):
    """Original Godelian Logical Flow (GLF) model for rotation curves."""
    G_r = G0 * np.exp(-k * r**2 / (1 + r**2))
    logical_term = np.log(1 + r) * G_r
    flow_term = (1 + r)**gamma * G_r
    return np.sqrt(alpha * c**2 * (1 + beta * logical_term + delta * flow_term))

def glf_model_improved(r, v0, r_s, alpha, beta, r_l, gamma, r_f, delta, epsilon, mu, nu, xi):
    """Improved Godelian Logical Flow (GLF) model for rotation curves."""
    G_r = (1 + (r/r_s)**alpha)**(-beta)
    logical_term = (1 + r/r_l)**gamma * G_r
    flow_term = (1 + (r/r_f)**delta)**epsilon * G_r
    combined_term = (logical_term**mu + flow_term**nu)**(1/xi)
    return v0 * np.sqrt(combined_term)

def dm_model(r, v_max, r_s):
    """Standard dark matter model (NFW profile) for rotation curves."""
    x = r / r_s
    return v_max * np.sqrt((np.log(1 + x) - x / (1 + x)) / (x * (np.log(1 + x) - x / (1 + x) + 0.5 / (1 +
        x)**2)))

def load_galaxy_data(file_path):
    """
    Load rotation curve data from a THINGS moment 1 FITS file.

    Args:
    file_path (str): Path to the FITS file

    Returns:
    tuple: Arrays of radius (kpc), velocity (km/s), and velocity error (km/s)
    """
    print(f"Loading file: {file_path}")
    with fits.open(file_path) as hdul:
        header = hdul[0].header
        data = hdul[0].data[0, 0, :, :] # Extract 2D data

        ny, nx = data.shape

        wcs = WCS(header).celestial # Get 2D WCS

        y, x = np.mgrid[:ny, :nx]
        ra, dec = wcs.wcs_pix2world(x, y, 0)

        center_ra, center_dec = wcs.wcs_pix2world(nx/2, ny/2, 0)

        # Calculate radial distance
        dx = (ra - center_ra) * np.cos(np.radians(center_dec))
        dy = dec - center_dec
        r = np.sqrt(dx**2 + dy**2)
```

```
        # Convert to physical units
        pixscale = np.abs(wcs.wcs.cdelt[0])
        r_kpc = r * 3600 * pixscale * 60 # Assuming 1 arcmin = 1 kpc

        # Flatten and clean data
        r_flat = r_kpc.flatten()
        v_flat = data.flatten()

        mask = ~np.isnan(v_flat)
        r_clean = r_flat[mask]
        v_clean = v_flat[mask]

        # Reduce data points
        n_points = 1000 # Adjust this number as needed
        if len(r_clean) > n_points:
            indices = np.linspace(0, len(r_clean) - 1, n_points, dtype=int)
            r_clean = r_clean[indices]
            v_clean = v_clean[indices]

        # Sort data by radius
        sort_idx = np.argsort(r_clean)
        radius = r_clean[sort_idx]
        velocity = v_clean[sort_idx]

        # Convert velocity to km/s if necessary
        if header.get('BUNIT', '').upper() == 'METR/SEC':
            velocity /= 1000

        # Estimate velocity error (10% of the velocity range)
        velocity_error = np.ones_like(velocity) * (np.max(velocity) - np.min(velocity)) * 0.1
        # Convert to physical units (assuming 1 arcmin = 1 kpc)
        pixscale = np.abs(wcs.wcs.cdelt[0])
        r_kpc = r * 3600 * pixscale

        # Convert velocity to km/s if necessary
        if header.get('BUNIT', '').upper() == 'M/S':
            velocity /= 1000
    print(f"Loaded data: {len(radius)} points")
    return radius, velocity, velocity_error
import matplotlib as mpl

def plot_rotation_curves(galaxies):
    """
    Plot rotation curves for all galaxies.

    Args:
    galaxies (dict): Dictionary with galaxy data
    """
    # Increase the chunk size to handle more complex paths
    mpl.rcParams['agg.path.chunksize'] = 10000

    for galaxy, data in galaxies.items():
        plt.figure(figsize=(12, 8))
        for weighting in ['NA', 'RO']:
            if data[weighting] is not None:
                radius, velocity, velocity_error = data[weighting]
                plt.errorbar(radius, velocity, yerr=velocity_error, fmt='o', alpha=0.5, capsize=3, label=f'{
                    galaxy} {weighting}')

        plt.xlabel('Radius (kpc)', fontsize=12)
        plt.ylabel('Velocity (km/s)', fontsize=12)
        plt.title(f'Rotation Curve - {galaxy}', fontsize=14)
        plt.legend(fontsize=10)
        plt.grid(True, linestyle='--', alpha=0.7)
        plt.tick_params(axis='both', which='major', labelsize=10)
        plt.tight_layout()
        plt.savefig(f'{galaxy}_rotation_curve.png', dpi=300)
        plt.close()

    # Plot all galaxies on one figure
    plt.figure(figsize=(15, 10))
    for galaxy, data in galaxies.items():
```

```
        for weighting in ['NA', 'RO']:
            if data[weighting] is not None:
                radius, velocity, _ = data[weighting]
                plt.plot(radius, velocity, '-', label=f'{galaxy} {weighting}', linewidth=1, alpha=0.7)

    plt.xlabel('Radius (kpc)', fontsize=12)
    plt.ylabel('Velocity (km/s)', fontsize=12)
    plt.title('Rotation Curves - All Galaxies', fontsize=14)
    plt.legend(fontsize=10, loc='center left', bbox_to_anchor=(1, 0.5))
    plt.grid(True, linestyle='--', alpha=0.7)
    plt.tick_params(axis='both', which='major', labelsize=10)
    plt.tight_layout()
    plt.savefig('all_galaxies_rotation_curves.png', dpi=300, bbox_inches='tight')
    plt.close()
def lieu_model(r, alpha, s):
    """Lieu's model for rotation curves based on massless topological defects."""
    return np.sqrt(alpha * c**2 / r) * np.exp(-r/s)



def fit_model(model, r, v, v_err, p0):
    try:
        popt, pcov = curve_fit(model, r, v, p0=p0, sigma=v_err, absolute_sigma=True, maxfev=50000)
        return popt, np.sqrt(np.diag(pcov))
    except RuntimeError as e:
        print(f"Fit failed for {model.__name__}: {str(e)}")
        return p0, np.ones_like(p0) * np.inf # Return initial guess and infinite errors
def compute_aic(model, params, r, v, v_err):
    """Compute the Akaike Information Criterion (AIC) for a model."""
    residuals = v - model(r, *params)
    sse = np.sum((residuals / v_err)**2)
    n = len(v)
    k = len(params)
    aic = 2*k + n*np.log(sse/n)
    return aic

def compute_bic(model, params, r, v, v_err):
    """Compute the Bayesian Information Criterion (BIC) for a model."""
    residuals = v - model(r, *params)
    sse = np.sum((residuals / v_err)**2)
    n = len(v)
    k = len(params)
    bic = k*np.log(n) + n*np.log(sse/n)
    return bic


def load_all_galaxies(data_dir):
    galaxies = {}

    # Use glob to find all FITS files
    fits_files = glob(os.path.join(data_dir, '**', '*MOM1_THINGS.FITS'), recursive=True)

    for file_path in fits_files:
        galaxy_name = extract_galaxy_name(file_path)

        if galaxy_name not in galaxies:
            galaxies[galaxy_name] = {'NA': None, 'RO': None}

        try:
            if 'NA' in file_path:
                galaxies[galaxy_name]['NA'] = load_galaxy_data(file_path)
            elif 'RO' in file_path:
                galaxies[galaxy_name]['RO'] = load_galaxy_data(file_path)
        except Exception as e:
            print(f"Error loading data for {galaxy_name} ({file_path}):")
            print(f"Error type: {type(e).__name__}")
            print(f"Error message: {str(e)}")

    return galaxies
def extract_galaxy_name(file_path):
    import re
    """
    Extract the galaxy name from the file path.
```

```
    Args:
    file_path (str): Path to the FITS file

    Returns:
    str: Extracted galaxy name
    """
    # Extract the filename from the path
    filename = os.path.basename(file_path)

    # Use regex to match galaxy name patterns
    match = re.match(r'(NGC_?\d+|DDO_?\d+|[A-Za-z]+_?\d+)_', filename)

    if match:
        return match.group(1)
    else:
        # If no match found, use the directory name as a fallback
        return os.path.basename(os.path.dirname(file_path))

# Now, update your analyze_galaxy function to include these new models:
def analyze_galaxy(file_path):
    galaxy_name = extract_galaxy_name(file_path)
    try:
        r, v, v_err = load_galaxy_data(file_path)
    except Exception as e:
        print(f"Error loading data for {galaxy_name}: {str(e)}")
        return None

    # Fit models
    models_to_fit = [
        ('lieu_simple', lieu_model_simple, [1e-6, 1e20]),
        ('lieu_strict', lieu_model_strict, [1e-6, 20]),
        ('lieu_multi', lieu_model_multi, [1e-6, 10, 20, 30, 5]),
        ('lieu_advanced', lieu_model_advanced, [1e-6, 1.0, 0.1, 10, 0.15]),
        ('glf_original', glf_model_original, [1e-6, 1, 1, 1e-6, 1, 1e-2]),
        ('glf_improved', glf_model_improved, [200, 10, 1, 1, 10, 1, 10, 1, 1, 1, 1, 1]),
        ('glf3_exp', glf3_exp, [200, 0.1, 0.1, 2, 10]),
        ('glf3_log', glf3_log, [200, 0.1, 0.1, 2, 10]),
        ('glf3_poly', glf3_poly, [200, 0.1, 0.1, 0.1, 0.1, 10]),
        ('dm', dm_model, [200, 20])
    ]

    results = {'galaxy': galaxy_name}
    for name, model, p0 in models_to_fit:
        popt, perr = fit_model(model, r, v, v_err, p0)
        if popt is not None:
            aic = compute_aic(model, popt, r, v, v_err)
            bic = compute_bic(model, popt, r, v, v_err)
            results[name] = {'params': popt, 'errors': perr, 'aic': aic, 'bic': bic}
        else:
            results[name] = None

    # Plot results
    plt.figure(figsize=(12, 8))
    plt.errorbar(r, v, yerr=v_err, fmt='o', alpha=0.5, label='Data')
    r_smooth = np.linspace(r.min(), r.max(), 500)
    for name, model, _ in models_to_fit:
        if results[name] is not None:
            plt.plot(r_smooth, model(r_smooth, *results[name]['params']), label=f'{name} model')
    plt.xlabel('Radius (kpc)')
    plt.ylabel('Velocity (km/s)')
    plt.legend()
    plt.title(f'Rotation curve models for {galaxy_name}')
    plt.savefig(f'{galaxy_name}_rotation_curve.png')
    plt.close()

    return results


def analyze_model_performance(data):
    # Convert the data to a pandas DataFrame
```

```
df = pd.DataFrame(data, columns=["Galaxy",
                              "Lieu Simple AIC", "Lieu Simple BIC",
                              "Lieu Strict AIC", "Lieu Strict BIC",
                              "Lieu Multi AIC", "Lieu Multi BIC",
                              "Lieu Advanced AIC", "Lieu Advanced BIC",
                              "GLF Original AIC", "GLF Original BIC",
                              "GLF Improved AIC", "GLF Improved BIC",
                              "GLF3 Exp AIC", "GLF3 Exp BIC",
                              "GLF3 Log AIC", "GLF3 Log BIC",
                              "GLF3 Poly AIC", "GLF3 Poly BIC",
                              "DM AIC", "DM BIC"])

# Convert AIC and BIC columns to float, replacing 'N/A' with NaN
for col in df.columns[1:]: # Skip the 'Galaxy' column
    df[col] = pd.to_numeric(df[col], errors='coerce')

narrative = "Detailed Statistical Analysis of Galaxy Model Performance\n\n"

# 1. Friedman Test
models = ['Lieu Simple', 'Lieu Strict', 'Lieu Multi', 'Lieu Advanced',
         'GLF Original', 'GLF Improved', 'GLF3 Exp', 'GLF3 Log', 'GLF3 Poly', 'DM']
friedman_data = df[[f'{model} AIC' for model in models]].dropna()

if len(friedman_data) > 1: # Ensure we have enough data for the test
    statistic, p_value = stats.friedmanchisquare(*[friedman_data[col] for col in friedman_data.columns
        ])

    narrative += "1. Overall Comparison of Models (Friedman Test):\n"
    narrative += f" Statistic: {statistic:.2f}, p-value: {p_value:.4f}\n"
    if p_value < 0.05:
        narrative += " There are significant differences among the models' performances.\n"
        narrative += " Pairwise Wilcoxon signed-rank tests:\n"
        for i, model1 in enumerate(models):
            for j, model2 in enumerate(models):
                if i < j:
                    valid_data = df[[f'{model1} AIC', f'{model2} AIC']].dropna()
                    if len(valid_data) > 1:
                        _, p = stats.wilcoxon(valid_data[f'{model1} AIC'], valid_data[f'{model2} AIC'])
                        narrative += f" - {model1} vs {model2}: {'Significant' if p < 0.05 else 'Not
                            significant'} (p = {p:.4f})\n"
                    else:
                        narrative += f" - {model1} vs {model2}: Insufficient data for comparison\n"
    else:
        narrative += " There are no significant differences among the models' performances.\n"
else:
    narrative += "1. Overall Comparison of Models (Friedman Test):\n"
    narrative += " Insufficient data for Friedman test\n"

# 2. Best Model Identification
lieu_models = ['Lieu Simple', 'Lieu Strict', 'Lieu Multi', 'Lieu Advanced']
glf_models = ['GLF Original', 'GLF Improved', 'GLF3 Exp', 'GLF3 Log', 'GLF3 Poly']

best_lieu = min(lieu_models, key=lambda m: df[f'{m} AIC'].mean())
best_glf = min(glf_models, key=lambda m: df[f'{m} AIC'].mean())

narrative += f"\n2. Best Performing Models:\n"
narrative += f" - Best Lieu model: {best_lieu}\n"
narrative += f" - Best GLF model: {best_glf}\n"

# 3. Detailed Comparison of Best Models
narrative += "\n3. Detailed Comparison of Best Models:\n"

def compare_models(model1, model2):
    valid_data = df[[f'{model1} AIC', f'{model2} AIC']].dropna()
    if len(valid_data) > 1:
        wilcoxon_statistic, wilcoxon_p_value = stats.wilcoxon(valid_data[f'{model1} AIC'], valid_data[f
            '{model2} AIC'])
        effect_size = (valid_data[f'{model1} AIC'].mean() - valid_data[f'{model2} AIC'].mean()) / np.
            std(valid_data[f'{model1} AIC'] - valid_data[f'{model2} AIC'])
        n_model1_wins = sum(valid_data[f'{model1} AIC'] < valid_data[f'{model2} AIC'])
        binomial_p_value = stats.binom_test(n_model1_wins, len(valid_data), p=0.5)

        comparison = f" {model1} vs {model2}:\n"
```

```
            comparison += f" - Wilcoxon test: {'Significant' if wilcoxon_p_value < 0.05 else 'Not
                significant'} (p = {wilcoxon_p_value:.4f})\n"
            comparison += f" - Effect size (Cohen's d): {effect_size:.2f} ({'Small' if abs(effect_size) <
                0.5 else 'Medium' if abs(effect_size) < 0.8 else 'Large'})\n"
            comparison += f" - {model1} outperforms {model2} in {n_model1_wins}/{len(valid_data)} galaxies
                (Binomial test p = {binomial_p_value:.4f})\n"
        else:
            comparison = f" {model1} vs {model2}: Insufficient data for comparison\n"
        return comparison

    narrative += compare_models(best_lieu, best_glf)
    narrative += compare_models(best_lieu, 'DM')
    narrative += compare_models(best_glf, 'DM')

    # 4. Model Ranking
    aic_ranks = df.iloc[:, 1::2].rank(axis=1, method='min')
    bic_ranks = df.iloc[:, 2::2].rank(axis=1, method='min')

    narrative += "\n4. Model Ranking:\n"
    narrative += " Average ranks (lower is better):\n"
    for model in models:
        avg_aic_rank = aic_ranks[f'{model} AIC'].mean()
        avg_bic_rank = bic_ranks[f'{model} BIC'].mean()
        narrative += f" - {model}: AIC rank = {avg_aic_rank:.2f}, BIC rank = {avg_bic_rank:.2f}\n"

    # 5. AIC and BIC Differences
    narrative += "\n5. AIC and BIC Differences:\n"
    for model in models:
        aic_diff = df[f'{model} AIC'] - df[f'{best_lieu} AIC']
        bic_diff = df[f'{model} BIC'] - df[f'{best_lieu} BIC']
        narrative += f" {model} vs {best_lieu}:\n"
        narrative += f" - Mean AIC difference: {aic_diff.mean():.2f} (std: {aic_diff.std():.2f})\n"
        narrative += f" - Mean BIC difference: {bic_diff.mean():.2f} (std: {bic_diff.std():.2f})\n"

    return narrative
def main():
    data_dir = '/Users/quantmann/Documents/THINGS_data'
    galaxies = load_all_galaxies(data_dir)

    # Plot rotation curves
    plot_rotation_curves(galaxies)

    results = []
    for galaxy, data in galaxies.items():
        print(f"Analyzing {galaxy}:")
        for weighting in ['NA', 'RO']:
            if data[weighting] is not None:
                radius, velocity, velocity_error = data[weighting]

                # Construct file path using the original galaxy name
                file_path = glob(os.path.join(data_dir, '**', f'{galaxy}*{weighting}*MOM1_THINGS.FITS'),
                    recursive=True)
                if file_path:
                    result = analyze_galaxy(file_path[0])
                    if result:
                        results.append(result)
                    print(f" {weighting} data analyzed")
                else:
                    print(f" No {weighting} data file found for {galaxy}")
            else:
                print(f" No {weighting} data")
        print()

    # Summarize results
    print(f"\nAnalyzed {len(results)} galaxy datasets")
    print("\nSummary of results:")
    table_data = []
    for result in results:
        row = [result['galaxy']]
        for model in ['lieu_simple', 'lieu_strict', 'lieu_multi', 'lieu_advanced', 'glf_original', '
             glf_improved', 'glf3_exp', 'glf3_log', 'glf3_poly', 'dm']:
            if model in result and result[model]:
                row.extend([f"{result[model]['aic']:.2f}", f"{result[model]['bic']:.2f}"])
```

```
            else:
                row.extend(["N/A", "N/A"])
        table_data.append(row)

    headers = ["Galaxy",
               "Lieu Simple AIC", "Lieu Simple BIC",
               "Lieu Strict AIC", "Lieu Strict BIC",
               "Lieu Multi AIC", "Lieu Multi BIC",
               "Lieu Advanced AIC", "Lieu Advanced BIC",
               "GLF Original AIC", "GLF Original BIC",
               "GLF Improved AIC", "GLF Improved BIC",
               "GLF3 Exp AIC", "GLF3 Exp BIC",
               "GLF3 Log AIC", "GLF3 Log BIC",
               "GLF3 Poly AIC", "GLF3 Poly BIC",
               "DM AIC", "DM BIC"]

    if table_data:
        print(tabulate(table_data, headers=headers, tablefmt="grid"))
    else:
        print("No valid results to display in the table.")

    # Perform statistical analysis
    narrative = analyze_model_performance(table_data)
    print("\nStatistical Analysis:")
    print(narrative)

def delta_approx(r, R, epsilon=1e-6):
    """Approximation of delta function."""
    return np.where(np.abs(r - R) < epsilon, 1/(2*epsilon), 0)

def lieu_model_advanced(r, alpha, R_0, spacing_factor, n_shells, baryon_fraction):
    """
    Advanced Lieu model with multiple shells, variable spacing, and baryonic component.

    This model aims to more closely represent the theoretical model described in Lieu's paper.
    It includes multiple shells with increasing spacing, a logarithmic gravitational potential,
    and a simplified baryonic component.

    Parameters:
    r : array-like, radial distances (kpc)
    alpha : float, model parameter related to the strength of the gravitational effect
    R_0 : float, radius of the first shell (kpc)
    spacing_factor : float, factor by which shell spacing increases
    n_shells : int, number of shells
    baryon_fraction : float, fraction of mass in baryons

    Returns:
    array-like, rotation velocities (km/s)
    """
    v = np.sqrt(alpha * c**2)

    # Calculate shell radii with increasing spacing
    shell_radii = R_0 * np.cumprod(np.full(int(n_shells), 1 + spacing_factor))

    # Sum effects of all shells
    shell_effects = np.sum([delta_approx(r, R) for R in shell_radii], axis=0)

    # Gravitational potential term
    potential_term = np.log(1 + r/R_0)

    # Baryonic term (simplified model assuming NFW-like profile)
    baryonic_term = baryon_fraction * (np.log(1 + r/R_0) - r/(r + R_0))

    return v * (shell_effects + potential_term + baryonic_term)

def glf3_exp(r, v0, alpha, beta, gamma, r_s):
    """GLF3 model with exponential term"""
    return v0 * np.sqrt(alpha * np.exp(-r/r_s) + beta * (r/r_s)**gamma)

def glf3_log(r, v0, alpha, beta, gamma, r_s):
    """GLF3 model with logarithmic term"""
    return v0 * np.sqrt(alpha * np.log(1 + r/r_s) + beta * (r/r_s)**gamma)
```

```
def glf3_poly(r, v0, alpha, beta, gamma, delta, r_s):
    """GLF3 model with polynomial terms"""
    return v0 * np.sqrt(alpha * (r/r_s) + beta * (r/r_s)**2 + gamma * (r/r_s)**3 + delta * (r/r_s)**4)

def fit_glf3_model(r, v, v_err, model):
    """Fit the GLF3 model to data"""
    if model.__name__ == 'glf3_poly':
        p0 = [100, 0.1, 0.1, 0.1, 0.1, 10] # Initial guess for poly model
    else:
        p0 = [100, 0.1, 0.1, 2, 10] # Initial guess for exp and log models

    popt, pcov = curve_fit(model, r, v, p0=p0, sigma=v_err, absolute_sigma=True, maxfev=10000)
    return popt, pcov
def plot_glf3_model(r, v, v_err, popt, model):
    """Plot the data and the fitted GLF3 model"""
    plt.figure(figsize=(10, 6))
    plt.errorbar(r, v, yerr=v_err, fmt='o', label='Data')
    r_smooth = np.linspace(r.min(), r.max(), 500)
    v_fit = model(r_smooth, *popt)
    plt.plot(r_smooth, v_fit, label=f'GLF3 {model.__name__} Model')
    plt.xlabel('Radius (kpc)')
    plt.ylabel('Velocity (km/s)')
    plt.legend()
    plt.title(f'GLF3 {model.__name__} Model Fit')
    plt.show()

if __name__ == "__main__":
    main()
```