UQGPF Development Roadmap (English Version)

Body instructions: UQGPF Development Roadmap & Gantt Chart (English Version)

Development Roadmap for UQGPF

Phase 1 – Connecting Model to Real Data (6–12 months)

Goal: Make key model components testable quickly.

1) Fit to Planck 2018 & DES Data - Objective: Establish a concrete workflow to infer model parameters from observational datasets, enabling rapid iteration and validation. - Data sources: - Planck 2018 CMB temperature and polarization spectra (TT, TE, EE, BB where applicable, including nuisance parameter treatments and covariance matrices). - DES (Dark Energy Survey) data products relevant for late-time structure growth, such as galaxy clustering and weak lensing two-point statistics, along with corresponding covariance matrices. - Parameter extraction targets: - λ (lambda): A coupling-like parameter in the UQGPF framework that modulates the strength of the proposed interaction or potential term. - ρ_0 (rho_0): A present-day density normalization that sets baseline contributions to the model's energy budget or field amplitude. - g_a (g_a): A coupling parameter associated with axion-like components or axion-gauge sector interactions within the UQGPF framework. - Methodology: - Implement Bayesian inference pipelines using Markov Chain Monte Carlo (MCMC) or Nested Sampling (e.g., MultiNest, dynesty) to obtain posterior distributions for (λ, ρ_0, g_a) and nuisance parameters. - Define a forward model: - Translate model parameters into predicted CMB spectra and matter power spectra. - Include stress-energy contributions, potential terms, and any non-standard transfer functions specific to UQGPF. - Likelihood construction: - For

Planck: use official likelihoods (Plik, Commander, SimAll as appropriate) with their covariance matrices and masks; include calibration and beam uncertainties as nuisance parameters. - For DES: assemble likelihoods for galaxy clustering and weak lensing, including intrinsic alignment, photo-z uncertainties, shear calibration biases, and redshift distribution errors. - Priors: - Physically motivated priors (e.g., non-negative energy densities, perturbative coupling bounds). - Consider broad, uninformative priors for exploratory phases, followed by more informative priors once preliminary fits indicate viable regions. - Validation: - Consistency checks across datasets (Planck vs DES) and cross-validation using subsets of the data. - Posterior predictive checks to assess how well the model reproduces observed statistics.

2) Compute Model's CMB Signature from `6.lamda correction 4.py` and compare with Planck spectrum - Objective: Validate the model's CMB imprint against canonical Planck measurements by directly computing the predicted spectrum from a referenced Python module/script. - Tasks: - Access and review the script 6.lamda correction 4.py to understand the corrections, the parameterization, and any numerical schemes used to compute CMB angular power spectra (C_ℓ). - Ensure the script produces C_ℓ predictions given (λ, ρ_0, g_a) along with standard cosmological parameters (Ω_b h^2, Ω_c h^2, H0, A_s, n_s, τ_reio, etc.) or through a wrapper that integrates with a Boltzmann code (e.g., CLASS or CAMB) if needed. - Implement a compatibility layer: - If the script outputs C_ℓ in a fluctuating ℓ-range, ensure it covers low-ℓ to high-ℓ with appropriate resolution. - If necessary, modify the script to output C_ℓ and error bars or covariance for a likelihood comparison. - Comparison metrics: - Compute χ^2 or likelihoods comparing model C_ℓ to Planck TT, TE, EE (and potentially BB) spectra, optionally using Planck's official covariance information. - Visualizations: plot residuals (data − model) across multipoles, and the ratio or fractional difference A_ℓ = (C_ℓ^model − C_ℓ^Planck)/σ_ℓ. - Uncertainty integration: - Propagate posterior samples of (λ, ρ_0, g_a) into C_ℓ predictions to obtain a distribution of C_ℓ curves and credible bands. - Validation: - Ensure numerical stability for high-ℓ and low-ℓ regimes, handle potential degeneracies with dust and foregrounds where relevant.

3) Link predictions for rc (axion core radius) to galactic halo data (SPARC / Gaia) - Objective: Connect particle-physics parameters to observable astrophysical structures in galaxies. - Definitions: - rc: The axion core radius in the halo, representing the scale where the axion dark matter density profile transitions to a

cored distribution rather than a cusp. This is a phenomenological parameter in several ultra-light axion cosmologies (fuzzy dark matter-like scenarios) and can be related to the particle mass m_a and the surrounding environment through quantum pressure support and halo formation processes. - Data sources: - SPARC (Spitzer Photometry and Accurate Rotation Curves): Rotation curves for a large sample of disk galaxies with high-quality rotation velocity measurements and detailed baryonic mass models. - Gaia: Stellar kinematics data for Milky Way and nearby galaxies, enabling dynamical inferences of dark matter distribution and possible core/cusp distinctions. - Methodology: - Build a mapping from model parameters (λ, ρ_0, g_a) to a predicted rc using physical relations from axion halo models: - Quantum pressure-derived core radius scales with axion mass m_a and halo properties: $rc \propto (\hbar^2/(G\ m\_a^2\ \rho\_c))^{1/4}$ in some fuzzy DM models, but adapt to your specific UQGPF framework including possible scale-dependent interactions. - If the model modifies the halo profile (e.g., soliton-like core inside Navarro–Frenk–White outer profile), implement the appropriate density profile and derive rc from the matching radius or density criterion. - For each galaxy in SPARC and Gaia datasets, fit the rotation curve with a composite model: - Baryonic components: stellar disk, gas, and bulge contributions using mass-to-light ratios and gas measurements. - Dark matter component: axion-core profile with rc as a parameter tied to the underlying particle physics. - Inference strategy: - Use hierarchical modeling to connect galaxy-scale rc distributions to global UQGPF parameters, or perform per-galaxy fits depending on data quality. - Incorporate measurement uncertainties in rotation curves, distance estimates, and baryonic mass modeling. - Validation: - Check for consistency between rc inferred from rotation curves and the rc predicted by the fitted (λ, ρ_0, g_a) in the context of the axion model. - Explore correlations between rc and halo properties (e.g., halo mass, concentration) to assess physical plausibility. - Visualization: - Produce rotation curve fits with and without axion core, residuals, and the posterior distribution of rc across the galaxy sample. - Forecasts: - Identify galaxies with the highest constraining power on rc and plan targeted observations or data re-analysis to tighten constraints.

Phase 2 – Code & Simulation Infrastructure Upgrade (12–18 months)

Goal: Prepare the code for large-scale scientific analysis.

1) Refactor Python scripts using Astropy, CLASS, Cobaya - Objectives: - Modernize the software stack to leverage community-standard libraries for cosmology and data analysis. - Achieve modular, maintainable, and reproducible code with clear interfaces between components. - Tasks: - Replace ad hoc numerical routines with Astropy-compatible utilities for units, cosmological calculations, and data I/O. - Integrate CLASS (via PyCLASS or a similar interface) for Boltzmann solver functionality when computing CMB and matter power spectra, ensuring compatibility with custom UQGPF terms. - Use Cobaya as the primary framework for Bayesian inference, cosmological parameter estimation, and sampling, leveraging its likelihoods, samplers, and plugin architecture. - Create a clean, well-documented API: - Core models module: defines UQGPF parameters, priors, and the forward model mappings to observables. - Likelihood module: wraps Planck, DES, and any supplementary data into Cobaya-compatible likelihoods with proper nuisance parameter handling. - Data module: handles loading, preprocessing, and masking of datasets (Planck, DES), including systematics treatment. - Diagnostics module: implements convergence checks (R-hat, effective sample size), posterior predictive checks, and plotting utilities. - Testing and CI: - Develop unit tests for critical components (parameter transformations, likelihood evaluations, spectral predictions). - Set up continuous integration to ensure code health across Python versions and platforms. - Performance: - Profile bottlenecks in the forward model and likelihood, and optimize with vectorization, parallelization, or just-in-time compilation (e.g., Numba) where appropriate. - Implement parallelized likelihood evaluations across multiple cores or compute nodes during sampling.

2) Add Λ(a) and interaction terms in a format compatible with BAO & SN Ia datasets - Objective: Ensure the model's non-standard Λ(a) functional form and interactions can be consistently evaluated alongside standard cosmological probes. - Tasks: - Define Λ(a): - Decide on the functional form Λ(a) as a function of the scale factor a (or redshift z) that captures the model's dark energy or effective potential evolution within UQGPF. - Implement Λ(a) as a modular component that can be toggled (e.g., Λ0, or a dynamical term with parameters controlling evolution). - Interaction terms: - Codify any coupling between UQGPF components (e.g., axion sector interacting with dark energy or dark matter) into the background evolution and perturbation equations if required by the model. - Ensure energy-momentum conservation is maintained, or clearly specify the non-conservation and its observable consequences. - Compatibility with BAO & SN Ia: - BAO data constrain the expansion history; SN Ia data provide luminosity

distances. Implement distance duals (D_V, D_A, H(z), μ(z)) within the forward model. - Extend the likelihood to include BAO measurements (e.g., from SDSS/BOSS/eBOSS) and SN Ia compilations (e.g., Pantheon) with their covariances. - Testing: - Compare standard ΛCDM predictions with Λ(a) cases to ensure codes recover ΛCDM in the limit of no interaction or when relevant parameters are set to baseline values. - Documentation: - Provide clear descriptions of the equations implemented, the parameter definitions, and the intended observational probes.

3) Test performance on parallel computing for parameter scanning - Objective: Achieve scalable parameter estimation across high-performance compute environments. - Tasks: - Implement parallel task distribution: - Use Cobaya's built-in parallelization capabilities to run multiple chains concurrently. - Employ MPI or multi-threading for the forward model evaluations if computationally intensive (e.g., when solving Boltzmann equations with CLASS coupled to the UQGPF terms). - Job management: - Create scripts for submitting sampling jobs to HPC clusters (e.g., Slurm, PBS), including resource requests and job arrays for independent runs. - Checkpointing and resilience: - Enable periodic checkpoints so long-running runs can resume after failures. - Reproducibility: - Record random seeds, environment specifications (Conda/venv), and exact package versions in a reproducible environment (e.g., environment.yml or requirements.txt). - Scalability experiments: - Benchmark sampling speed versus the number of walkers/samples, and quantify scaling behavior to guide future resource planning.

Phase 3 – Path to Experimental Testing (2–5 years)

Goal: Make the model experimentally verifiable.

1) Collaborate with LiteBIRD, PIXIE, CMB-S4 to check for B-mode & 4.5 GHz signatures - Objectives: - Identify and exploit potential signatures stemming from the UQGPF framework that could manifest as B-mode polarization or other CMB observables at 4.5 GHz or related observational windows. - Tasks: - Signature mapping: - Theorize how UQGPF could imprint B-modes: via primordial gravitational waves, axion-induced polarization mixing, or modified recombination-era physics. - Define expected amplitudes, spectral shapes, and angular dependence for the proposed signatures to guide data analysis. - Observational collaboration: - Engage with LiteBIRD, PIXIE, and CMB-S4 teams to

incorporate UQGPF-specific templates into their data analysis pipelines, ensuring compatibility with their instrument models, noise properties, and systematics. - Data analysis plan: - Propose joint likelihood analyses or cross-correlations with other probes to maximize sensitivity to predicted signatures. - Forecasts: - Use instrument specs to forecast the detectability of predicted signals over the mission lifetimes, including constraints on parameter combinations (λ, ρ_0, g_a). - Mitigation of systematics: - Develop strategies to separate potential UQGPF signals from foregrounds (galactic dust, synchrotron) and instrumental systematics.

2) Design joint neutrino–gravitational wave analysis for measurable time delays - Objective: Explore a multi-messenger pathway to test the model by correlating neutrino and gravitational-wave signals. - Tasks: - Model predictions: - Determine how the UQGPF framework could influence neutrino emission times, propagation effects, or gravitational wave propagation, potentially leading to measurable time delays in multi-messenger events. - Data sources: - Neutrino detectors (e.g., IceCube, ANTARES) and gravitational-wave detectors (LIGO/Virgo/KAGRA) for joint event analyses. - Analysis pipeline: - Construct a Bayesian framework to infer time-delay parameters, accounting for astrophysical source models and propagation effects. - Sensitivity estimates: - Provide forecasts for expected constraints given event rates and detector capabilities.

3) Simulate axion halo collisions & compare to LIGO/Virgo/KAGRA event data - Objective: Assess whether axion halo interactions could leave observable imprints in gravitational-wave event catalogs. - Tasks: - Theoretical modeling: - Develop simplified models for axion halo collisions within galactic environments and predict any gravitational-wave signatures or perturbations. - Simulations: - Run numerical simulations (or semi-analytical models) of halo collisions and propagate predictions to gravitational-wave observables where applicable. - Data comparison: - Compare predictions with LIGO/Virgo/KAGRA events, focusing on anomalies, time delays, waveform modulations, or stochastic background contributions.

Phase 4 – Publishing & Scientific Validation (in parallel with other phases)

Goal: Gain peer-reviewed feedback.

1) Draft a short preprint focused on testable UQGPF results - Content: - Clear statement of model, key parameters (λ, ρ_0, g_a), and the physical motivation. - Summary of the data used (Planck 2018, DES) and the analysis approach (MCMC/Nested Sampling, forward modeling, $\Lambda(a)$). - Main results including posterior constraints on parameters and initial predictions for rc and CMB signatures. - Description of the linking of rc to SPARC/Gaia data with initial fits or upper/lower bounds. - Discussion of systematic uncertainties and limitations, along with planned improvements for Phase 2.

2) Submit to arXiv under cosmology/astro-ph - Action steps: - Prepare arXiv-formatted manuscript with figures, tables, and appendices detailing methodology, prior choices, convergence diagnostics, and model equations. - Include supplementary material with code snippets, data processing steps, and instructions to reproduce results (where permissible), or provide a link to a repository.

3) Create a formal GitHub documentation with a polished README, run examples, and datasets - Components: - README: overview of project goals, repository structure, and how to get started. - Tutorials: step-by-step examples showing how to run: - The forward model for a given parameter set to produce CMB spectra. - The Bayesian inference pipeline on Planck+DES data using Cobaya. - The rc inference pipeline from SPARC/Gaia data with axion-core profiles. - API documentation: modules, classes, functions, and expected input/output formats. - Datasets: clear guidance on data formats, preprocessing steps, and any licensing considerations. - Running instructions: environment setup (conda/virtualenv), required dependencies, and reproducible commands. - Tests: how to run unit and integration tests; instructions for contributors. - Contribution guidelines: code style, review process, and issue/PR templates. - Changelog: record major updates and changes across versions.

Phase 5 – Long-term Theoretical Development (5–10 years)

Goal: Complete the advanced theoretical framework.

- Tasks:

- Extend UQCMF equations to fully solvable numerical systems:
  - Derive and implement a comprehensive set of coupled equations for the UQGPF framework, including background evolution and linear perturbations, ensuring numerical solvability and stability.
- Investigate symmetry-breaking links to the Standard Model:
  - Explore potential connections between the UQGPF sector and electroweak or QCD symmetries, examining possible effective field theory realizations.
- Design conceptual experiments for MultiUniverse & UltraLight physics:
  - Propose thought experiments or observational tests capable of probing ultra-light fields, multi-universe concepts, or related phenomena within the model's scope.

Main Recommendation

Focus Phase 1 fully on UQGPF.pdf + 6.lamda correction 4.py, since they offer both a complete framework and immediately testable predictions, enabling faster feedback from the scientific community.

Gantt Chart Timeline (Summary)

Phase	2025-Q3	2025-Q4	2026-Q1	2026-Q2	2026-Q3	2027-Q1	20
1 Connect to Data	████	████					
2 Code Upgrade			████	████			
3 Experimental				████	████	████	████
4 Publication	████	████	████	████	████	████	████
5 Theoretical LT					████	████	████

End of document.