Wideband PSD Estimation and Occupancy Detection

PSD Estimation via KL + GP

Data model. Let $\{f_i\}_{i=1}^M$ be a frequency grid with quadrature weights $\{w_i\}_{i=1}^M$ (w.r.t. the Lebesgue measure unless otherwise specified).

For wideband spectrum sensing, the default choice is a uniform grid of weights since FFT-based measurements naturally produce uniform frequency grids with spacing Δf, that is $w_i{=}\Delta f$ for all $i = 1, \ldots, M$. This avoids scaling errors in high-dimensional problems and ensures $\sum_{i=1}^M w_i = M\Delta f = B_{\text{total}}$. Trapezoidal weights, i.e., $w_i{=}(f_{i+1} - f_{i-1})/2$ with boundary adjustments, are preferred when high accuracy is needed for narrow-band features or non-uniform grids.

We observe subsampled measurements

$$\boldsymbol{y} = \boldsymbol{\Theta}\,\boldsymbol{s} + \boldsymbol{\varepsilon}, \quad \boldsymbol{y} \in \mathbb{R}^N, \qquad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_\varepsilon), \tag{1}$$

where $\boldsymbol{s}{\in}\mathbb{R}^M$ is the (unknown) PSD over the grid, $\boldsymbol{\Theta}{\in}\mathbb{R}^{N\times M}$ is a known linear sampling/sensing operator, and $\boldsymbol{\Sigma}_\varepsilon$ is the noise covariance (often $\Sigma^2\boldsymbol{I}$).

KL basis construction. Choose a positive definite kernel $k_f{:}\mathcal{F} \times \mathcal{F}{\to}\mathbb{R}$ on frequency. Form the weighted Gram matrix

$$[\boldsymbol{K}_w]_{ij} = k_f(f_i, f_j)\,\sqrt{w_i w_j}, \quad i, j = 1, \ldots, M. \tag{2}$$

Compute its eigendecomposition (retain $R{\ll}M$ dominant modes):

$$\boldsymbol{K}_w\,\boldsymbol{v}_n = \lambda_n\,\boldsymbol{v}_n, \quad \lambda_1{\geq}\cdots{\geq}\lambda_R{>}0, \quad n = 1, \ldots, R \tag{3}$$

Define discrete eigenfunctions on the grid

$$\phi_n(f_i) \approx [\boldsymbol{v}_n]_i/\sqrt{w_i}, \qquad i = 1, \ldots, M. \tag{4}$$

For off-grid evaluation $f{\in}\mathcal{F}$ (Nyström extension):

$$\phi_n(f) \approx \frac{1}{\lambda_n} \sum_{j=1}^{M} k_f(f, f_j)\, w_j\, \phi_n(f_j). \tag{5}$$

KL feature matrix and low-rank field model. Define the KL feature matrix $\mathbf{\Phi}_{\text{KL}}{\in}\mathbb{R}^{M\times R}$ by

$$\mathbf{\Phi}_{\text{KL}} = \sqrt{\lambda_n}\,\phi_n(f_i), \qquad i = 1, \ldots, M,\ n = 1, \ldots, R. \tag{6}$$

Approximate the PSD by the rank-R KL expansion

$$\boldsymbol{s} \approx \mathbf{\Phi}_{\text{KL}}\,\boldsymbol{\xi}, \qquad \boldsymbol{\xi} \in \mathbb{R}^R. \tag{7}$$

With the prior $\boldsymbol{\xi}{\sim}\mathcal{N}(\mathbf{0}, \boldsymbol{I}_R)$, this induces (approximately) a truncated Mercer/KL covariance:

$$C(\boldsymbol{s}) \approx \mathbf{\Phi}_{\text{KL}}\mathbf{\Phi}_{\text{KL}}^\top = \sum_{n=1}^{R} \lambda_n\,\boldsymbol{\phi}_n\boldsymbol{\phi}_n^\top$$

[Single-window] Static Bayesian linear regression. Let $\mathbf{A}{:=}\boldsymbol{\Theta}\,\boldsymbol{\Phi}_{\mathrm{KL}}{\in}\mathbb{R}^{N\times R}$ so that $\boldsymbol{y}{=}\boldsymbol{A}\,\boldsymbol{\xi}{+}\boldsymbol{\varepsilon}$. With $\boldsymbol{\xi}{\sim}\mathcal{N}(\mathbf{0},\boldsymbol{I}_R)$ and $\boldsymbol{\varepsilon}{\sim}\mathcal{N}(\mathbf{0},\boldsymbol{\Sigma}_\varepsilon)$, the posterior is

$$\boldsymbol{\Sigma}_{\xi|y} = \Big(\boldsymbol{I}_R + \boldsymbol{A}^\top \boldsymbol{\Sigma}_\varepsilon^{-1}\boldsymbol{A}\Big)^{-1}, \tag{8a}$$

$$\boldsymbol{\mu}_{\xi|y} = \boldsymbol{\Sigma}_{\xi|y}\,\boldsymbol{A}^\top \boldsymbol{\Sigma}_\varepsilon^{-1}\boldsymbol{y}. \tag{8b}$$

Within a single window, the PSD posterior (on the grid) is Gaussian with

$$\boldsymbol{\mu}_{s|y} = \boldsymbol{\Phi}_{\mathrm{KL}}\,\boldsymbol{\mu}_{\xi|y}, \tag{9a}$$

$$\boldsymbol{\Sigma}_{s|y} = \boldsymbol{\Phi}_{\mathrm{KL}}\,\boldsymbol{\Sigma}_{\xi|y}\,\boldsymbol{\Phi}_{\mathrm{KL}}^\top. \tag{9b}$$

[Multiple-windows] Dynamic GP prior on coefficients. For windows $t{=}1,\ldots,T$, let $\boldsymbol{y}_t{=}\boldsymbol{\Theta}_t\,\boldsymbol{s}_t + \boldsymbol{\varepsilon}_t$ and assume $\boldsymbol{s}_t{\approx}\boldsymbol{\Theta}_{\mathrm{KL}}\boldsymbol{\xi}_t$ with $\boldsymbol{\xi}_t{\in}\mathbb{R}^R$.

Place independent GP priors across time on coefficients:

$$\xi_n(\cdot) \sim \mathcal{GP}(0, k_n(\cdot,\cdot)), \qquad n = 1,\ldots,R, \tag{10}$$

e.g., Matérn kernels ($\nu \in \{1/2, 3/2, 5/2\}$).

State-space representations reformulate GP regression using Kalman filtering theory, achieving linear-time $\mathcal{O}(n)$ complexity for Markovian processes.

Use state-space realizations for scalable inference: for $\nu{=}1/2$ (Ornstein–Uhlenbeck state-space form), the discrete-time model for step Δt_t is

$$\xi_{n,t} = a_{n,t}\,\xi_{n,t-1} + \eta_{n,t}, \quad a_{n,t} = e^{-\alpha_n \Delta t_t}, \tag{11a}$$

$$\eta_{n,t} \sim \mathcal{N}\Big(0,\ q_n \frac{1 - e^{-2\alpha_n \Delta t_t}}{2\alpha_n}\Big), \tag{11b}$$

with hyperparameters $\alpha_n, q_n{>}0$.

This state-space representation enables efficient $\mathcal{O}(TR^2)$ inference via Kalman filtering/smoothing

For $\nu{\in}\{3/2, 5/2\}$, use standard p-dimensional Markov state embeddings with known $(\boldsymbol{F}_n, \boldsymbol{L}_n, \boldsymbol{Q}_n)$ (omitted for brevity).

The observation equation is

$$\boldsymbol{y}_t = \boldsymbol{H}_t\,\boldsymbol{\xi}_t + \boldsymbol{\varepsilon}_t, \qquad \boldsymbol{H}_t := \boldsymbol{\Theta}_t\,\boldsymbol{\Phi}_{\mathrm{KL}}, \qquad \boldsymbol{\varepsilon}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\varepsilon,t}). \tag{12}$$

Perform Kalman filtering and Rauch–Tung–Striebel smoothing across t and across modes n, producing smoothed $\boldsymbol{\mu}_{\xi|y,t}$ and $\boldsymbol{\Sigma}_{\xi|y,t}$, and reconstruct

$$\boldsymbol{\mu}_{s|y,t} = \boldsymbol{\Phi}_{\mathrm{KL}}\,\boldsymbol{\mu}_{\xi|y,t}, \qquad \boldsymbol{\Sigma}_{s|y,t} = \boldsymbol{\Phi}_{\mathrm{KL}}\,\boldsymbol{\Sigma}_{\xi|y,t}\,\boldsymbol{\Phi}_{\mathrm{KL}}^\top. \tag{13}$$

Outputs: PSD credible bands and occupancy probabilities. For a grid index i, the posterior $s_i|y{\sim}\mathcal{N}(\mu_{s|y,i}, \Sigma^2_{s|y,i})$ with $\mu_{s|y,i}{=}[\boldsymbol{\mu}_{s|y}]_i$ and $\Sigma^2_{s|y,i}{=}[\boldsymbol{\Sigma}_{s|y}]_{ii}$. A $(1{-}\alpha)$ pointwise credible interval is

$$\left[\mu_{s|y,i} - z_{\alpha/2}\,\Sigma_{s|y,i},\ \mu_{s|y,i} + z_{\alpha/2}\,\Sigma_{s|y,i}\right]. \tag{14}$$

Define band power over $B{\subset}\{1,\ldots,M\}$ (e.g., weighted sum or average):

$$p_B := \boldsymbol{c}_B^\top \boldsymbol{s}, \qquad \boldsymbol{c}_B \in \mathbb{R}^M,\ [\boldsymbol{c}_B]_i = \begin{cases} w_i / \sum_{j\in B} w_j, & i \in B, \\ 0, & \text{otherwise}. \end{cases} \tag{15}$$

where w_j is the set of quadrature weights.

Then $p_B|y \sim \mathcal{N}(\mu_B, \Sigma_B^2)$ with

$$\mu_B = \boldsymbol{c}_B^\top \boldsymbol{\mu}_{s|y}, \qquad \Sigma_B^2 = \boldsymbol{c}_B^\top \boldsymbol{\Sigma}_{s|y}\, \boldsymbol{c}_B. \tag{16}$$

For a threshold τ_B, the occupancy probability is

$$\mathbb{P}(p_B > \tau_B \mid \boldsymbol{y}) \;=\; 1 - \Psi\left(\frac{\tau_B - \mu_B}{\Sigma_B}\right), \tag{17}$$

where $\Psi(\cdot)$ is the standard normal CDF, $p_B := \boldsymbol{c}_B^\top \boldsymbol{s}$ is the band power, $p_B := \boldsymbol{c}_B^\top \boldsymbol{s}$ is the power threshold or a decision boundary such that: If $p_B > \tau_B$: The frequency band B is considered occupied, or if $p_B \leq \tau_B$: The frequency band B is considered vacant (noise only). For practical implementation, τ_B can be formulated as:

$$\tau_B = \sigma_\varepsilon^2\,(1+\gamma)$$

where σ_ε^2 is the noise variance, and γ is a threshold factor that can be set based on false alarm requirements: $\gamma = \sqrt{\frac{2}{M}} Q^{-1}(P_{fa})$ with P_{fa} as the target false alarm probability and $Q^{-1}(\cdot)$ as the inverse Q-function.

Lastly, the posterior distribution of band power is Gaussian: $p_B|\mathbf{y} \sim \mathcal{N}(\mu_B, \Sigma_B^2)$, where $\mu_B = \mathbf{c}_B^\top \boldsymbol{\mu}_{s|y}$, and $\Sigma_B^2 = \mathbf{c}_B^\top \boldsymbol{\Sigma}_{s|y} \mathbf{c}_B$. The occupancy probability is then:

$$P(p_B > \tau_B|\mathbf{y}) = 1 - \Phi\left(\frac{\tau_B - \mu_B}{\Sigma_B}\right)$$

where $\Psi(\cdot)$ is the standard normal cumulative distribution function.

For pointwise (per-bin) occupancy, a similar expression applies:

$$P(s_i > \tau_i|\mathbf{y}) = 1 - \Phi\left(\frac{\tau_i - \mu_{s|y,i}}{\Sigma_{s|y,i}}\right)$$

where $\mu_{s|y,i} = [\boldsymbol{\mu} s|y]i$ and $\Sigma^2 s|y,i = [\boldsymbol{\Sigma} s|y]_{ii}$.

Hyperparameters and learning. Kernel hyperparameters for k_f (frequency), dynamic priors k_n (time), and noise $\boldsymbol{\Sigma}_\varepsilon$ can be learned by maximizing the (marginal) likelihood or via cross-validation.

Static model: marginal likelihood optimization. In the static model with $\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, the log marginal likelihood is

$$\log p(\boldsymbol{y} \mid \boldsymbol{\theta}) = -\tfrac{1}{2}\boldsymbol{y}^\top \big(\boldsymbol{A}\boldsymbol{A}^\top + \boldsymbol{\Sigma}_\varepsilon\big)^{-1}\boldsymbol{y} - \tfrac{1}{2}\log\big|\boldsymbol{A}\boldsymbol{A}^\top + \boldsymbol{\Sigma}_\varepsilon\big| - \tfrac{N}{2}\log(2\pi), \tag{18}$$

where $\boldsymbol{\theta}$ collects all hyperparameters: frequency kernel parameters (e.g., lengthscale ℓ_f, variance σ_f^2), and noise level σ_ε^2.

Optimization strategy: Use gradient-based methods (L-BFGS-B, Adam) with automatic differentiation. The gradient w.r.t. θ_k is:

$$\frac{\partial \log p(\boldsymbol{y} \mid \boldsymbol{\theta})}{\partial \theta_k} = \tfrac{1}{2}\mathrm{tr}\Big[\big(\boldsymbol{\alpha}\boldsymbol{\alpha}^\top - \boldsymbol{C}^{-1}\big)\frac{\partial \boldsymbol{C}}{\partial \theta_k}\Big], \tag{19}$$

where $\boldsymbol{C} = \boldsymbol{A}\boldsymbol{A}^\top + \boldsymbol{\Sigma}_\varepsilon$ and $\boldsymbol{\alpha} = \boldsymbol{C}^{-1}\boldsymbol{y}$.

Computational cost: Each likelihood evaluation requires $\mathcal{O}(N^3)$ for Cholesky decomposition or $\mathcal{O}(NR^2)$ using the Woodbury identity:

$$\boldsymbol{C}^{-1} = \boldsymbol{\Sigma}_\varepsilon^{-1} - \boldsymbol{\Sigma}_\varepsilon^{-1}\boldsymbol{A}\big(\boldsymbol{I}_R + \boldsymbol{A}^\top \boldsymbol{\Sigma}_\varepsilon^{-1}\boldsymbol{A}\big)^{-1}\boldsymbol{A}^\top \boldsymbol{\Sigma}_\varepsilon^{-1}. \tag{20}$$

Initialization guidelines. **Frequency kernel k_f:**

- *Lengthscale* ℓ_f: Initialize to capture expected correlation bandwidth. For Matérn kernels, use $\ell_f \approx 0.1 \times B_{\text{total}}$ (10% of total bandwidth) as a starting point. For narrowband signals, use smaller values ($\ell_f \approx 0.01 \times B_{\text{total}}$).
- *Variance* σ_f^2: Initialize to the empirical variance of a rough PSD estimate (e.g., from periodogram): $\sigma_f^2 \approx \mathrm{Var}(\hat{\boldsymbol{s}}_{\text{init}})$.
- *Smoothness* ν (for Matérn): Start with $\nu = 3/2$ (once differentiable) for typical RF spectra. Use $\nu = 5/2$ for smoother spectra, $\nu = 1/2$ for rough/discontinuous.

Noise variance σ_ε^2: Initialize from hardware specs or empirical noise floor:

$$\sigma_\varepsilon^2 \approx k_B T B_{\text{sensor}} F_{\text{noise}}, \tag{21}$$

where k_B is Boltzmann's constant, T is temperature, B_{sensor} is sensor bandwidth, and F_{noise} is the noise figure.

Temporal kernel k_n (for dynamic model):

- *Lengthscale* ℓ_t or $1/\alpha_n$: Initialize to expected correlation time. For slowly-varying spectra (e.g., cognitive radio), use $\ell_t \approx 10 \times T_{\text{window}}$. For rapidly-varying (e.g., frequency-hopping), use $\ell_t \approx T_{\text{window}}$.
- *Variance* q_n: Initialize to allow moderate temporal variation: $q_n \approx 0.1$ (relative to unit-variance prior on ξ).

Identifiability and constraints. **Key identifiability issue:** The frequency kernel variance σ_f^2 and the KL coefficient prior variance (fixed at 1) interact through the eigenvalues $\{\lambda_n\}$. Similarly, temporal kernel variance q_n trades off with the frequency kernel.

Recommended constraints:

- **Normalize frequency kernel:** Fix $k_f(0,0) = 1$ (unit variance at origin) to avoid redundancy with coefficient scaling. Equivalently, constrain $\sigma_f^2 = 1$ and absorb signal power into the data likelihood.
- **Mode-specific temporal variances:** Allow different q_n for each KL mode n, but regularize via a hierarchical prior:

$$q_n \sim \mathrm{InverseGamma}(a_0, b_0), \tag{22}$$

  with $a_0 = b_0 = 1$ (weak prior encouraging $q_n \approx 1$).
- **Noise floor constraint:** Enforce $\sigma_\varepsilon^2 \geq \sigma_{\min}^2$ based on hardware noise floor to prevent overfitting.

Cross-validation strategy. For finite datasets with T windows, use **time-series cross-validation**:

Expanding window CV:

1. Split data into training windows $\{1, \ldots, t_{\text{train}}\}$ and validation $\{t_{\text{train}} + 1, \ldots, t_{\text{train}} + t_{\text{val}}\}$.
2. Train model on training set, predict on validation set.
3. Compute validation score: negative log predictive density (NLPD):

$$\mathrm{NLPD} = -\sum_{t=t_{\text{train}}+1}^{t_{\text{train}}+t_{\text{val}}} \log p(\boldsymbol{y}_t \mid \boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}). \tag{23}$$

4. Increment t_{train} and repeat, averaging scores across folds.

Alternative: leave-future-out CV. Fix training window size, slide forward in time, always predict future observations.

Multi-scale optimization. For large problems, use a **coarse-to-fine** strategy:

1. **Stage 1 (coarse grid):** Learn hyperparameters on a downsampled frequency grid ($M' \ll M$) to get rough estimates quickly.
2. **Stage 2 (full grid):** Refine hyperparameters on the full grid using Stage 1 values as initialization.
3. **Stage 3 (mode-specific):** Fix frequency kernel, optimize temporal parameters $\{\alpha_n, q_n\}$ per mode independently via mode-specific likelihoods.

Practical recipe.

1. **Initialize:** Use domain knowledge (bandwidth correlations, expected smoothness, noise floor).
2. **Optimize frequency kernel:** Maximize marginal likelihood on a single representative window, holding noise variance fixed.
3. **Optimize noise variance:** With frequency kernel fixed, optimize σ_ε^2 (or learn per-window if heteroscedastic).
4. **Optimize temporal kernels:** For the dynamic model, use expanding-window CV to tune $\{\alpha_n, q_n\}$.
5. **Joint refinement:** Optionally, perform joint optimization of all parameters, using the above as initialization.
6. **Validate:** Check that learned parameters are physically reasonable (e.g., ℓ_f matches expected signal bandwidths, σ_ε^2 aligns with SNR).

Diagnostic checks.

- **Residual analysis:** Standardized residuals $\boldsymbol{r} = \boldsymbol{\Sigma}_\varepsilon^{-1/2}(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\mu}_{\xi|y})$ should be approximately $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. Check via Q-Q plots.
- **Predictive checks:** Simulate data from the posterior predictive $p(\boldsymbol{y}^* \mid \boldsymbol{y})$ and compare to held-out observations. Large discrepancies indicate model misspecification.
- **Hyperparameter stability:** Repeat optimization from multiple random initializations. If results vary significantly, consider stronger priors or more data.

Complexity and benefits. The KL truncation yields rank-R features and reduces complexity:

- Static window (posterior of $\boldsymbol{\xi}$): forming and solving involves $\mathcal{O}(NR^2 + R^3)$.
- Streaming with Kalman: per step $\mathcal{O}(NR^2 + R^3)$ (often dominated by $\mathbf{H}_t$ multiplies and a size-R Riccati update).
- Nyström eigendecomposition on M-grid: typically $\mathcal{O}(M^3)$ once, then reused; for large M, randomized/Nyström acceleration may be used.

Benefits include robustness at low SNR, principled uncertainty quantification, and efficient sub-Nyquist recovery driven by low-rank spectral structure.

Implementation notes.

- Weight consistency checklist.
  - Verify $\sum_{i=1}^{M} w_i = B_{\text{total}}$ (total bandwidth)
  - Use identical $\{w_i\}$ in Gram matrix construction, eigenfunction definition, Nyström extension, and band power calculation
  - For uniform grid: $w_i = \Delta f = B_{\text{total}}/M$
  - For non-uniform grid: $w_i = (f_{i+1} - f_{i-1})/2$ with boundary corrections
  - Test orthonormality: $\sum_i \phi_n(f_i)\phi_m(f_i)w_i \approx \delta_{nm}$
  - Verify Nyström: $|\phi_n(f_k) - (1/\lambda_n)\sum_j k_f(f_k, f_j)w_j\phi_n(f_j)| < \epsilon$
- Choose R via cumulative variance: $\sum_{n=1}^{R} \lambda_n / \sum_{n=1}^{M} \lambda_n \in [0.95, 0.99]$.
- Ensure weighted inner products in projections and Nyström consistency to avoid scaling errors.
- Composite kernels on frequency (e.g., Matérn + periodic) can capture narrowband and cyclostationary patterns.
- **Sub-Nyquist Recovery Conditions.** The wideband PSD estimation framework enables recovery under sub-Nyquist sampling when the following key conditions are satisfied:

  Low-Rank Spectral Structure: The PSD can be well-approximated by a truncated Karhunen-Loève (KL) expansion:

  $$\mathbf{s} \approx \mathbf{\Phi}_{\text{KL}}\boldsymbol{\xi}, \quad \boldsymbol{\xi} \in \mathbb{R}^R \tag{24}$$

  where $R \ll M$ (the rank is much smaller than the number of frequency bins).

  Kernel Expressivity Condition: The chosen frequency kernel k_f must be sufficiently expressive to capture the spectral characteristics, with appropriate parameters:

  - Lengthscale: $\ell_f \approx 0.1 \times B_{\text{total}}$ for typical signals
  - Smoothness parameter (for Matérn): $\nu \in \{1/2, 3/2, 5/2\}$ matching signal regularity

  Sampling Operator Requirements: The linear sampling/sensing operator $\mathbf{\Theta} \in \mathbb{R}^{N \times M}$ must satisfy:

  - Sub-Nyquist condition: $N < M$
  - Sufficient incoherence with the KL basis
  - Adequate restricted isometry properties when combined with the KL basis

  Measurement SNR Condition: The noise covariance $\mathbf{\Sigma}_\varepsilon$ must permit reliable signal detection:

  $$\sigma_\varepsilon^2 < \text{min signal power of interest} \tag{25}$$

  KL Mode Selection Criterion: The retained KL modes should capture most of the signal variance:

  $$\frac{\sum_{n=1}^{R} \lambda_n}{\sum_{n=1}^{M} \lambda_n} \in [0.95, 0.99] \tag{26}$$

  Effective Dimensionality Reduction: The computational complexity benefits are realized when:

  $$O(NR^2 + R^3) \ll O(NM^2 + M^3) \tag{27}$$

  which is satisfied when $R \ll M$ and the recovery conditions above hold.

Next Steps for Implementation

The mathematical framework presented in the document provides a solid foundation for wideband PSD estimation and occupancy detection. The following steps outline a structured implementation approach:

1. **Core Algorithm Implementation**
   - Implement weighted Gram matrix construction: $[K_w]_{ij} = k_f(f_i, f_j)\sqrt{w_i w_j}$
   - Develop efficient eigendecomposition for KL basis extraction: $K_w v_n = \lambda_n v_n$
   - Construct the KL feature matrix: $\Phi_{\mathrm{KL}} = \sqrt{\lambda_n}\phi_n(f_i)$
   - Implement the posterior inference engine:

$$\mu_{s|y} = \Phi_{\mathrm{KL}}\mu_{\xi|y} \tag{28a}$$

$$\Sigma_{s|y} = \Phi_{\mathrm{KL}}\Sigma_{\xi|y}\Phi_{\mathrm{KL}}^{\top} \tag{28b}$$

2. **Dynamic Model Components**
   - Implement state-space representations for GP priors
   - Develop Kalman filtering and RTS smoothing infrastructure
   - Create mode-specific temporal parameter handling
3. **Hyperparameter Optimization**
   - Implement marginal likelihood optimization: $\log p(y|\theta)$
   - Develop gradient computation using automatic differentiation
   - Create the multi-scale optimization pipeline (coarse-to-fine strategy)
   - Implement time-series cross-validation for dynamic models
4. **Occupancy Detection Module**
   - Implement band power calculation: $p_B = c_B^{\top} s$
   - Develop threshold determination based on false alarm rate
   - Create occupancy probability calculator: $P(p_B > \tau_B|y) = 1 - \Psi\left(\frac{\tau_B - \mu_B}{\Sigma_B}\right)$
   - Implement visualization tools for occupancy maps and uncertainty
5. **Testing and Validation**
   - Develop synthetic data generators with known ground truth
   - Implement performance metrics (ROC curves, RMSE, KL divergence)
   - Create diagnostic tools for model validation
   - Design test cases for sub-Nyquist recovery scenarios
6. **Deployment Optimization**
   - Optimize computational bottlenecks (eigendecomposition, Kalman filtering)
   - Implement parallel processing for multi-band analysis
   - Develop streaming capabilities for real-time processing
   - Create caching mechanisms for reusable components (KL basis)
7. **Integration and API Development**

- Design modular API for component reusability
- Develop configuration system for hyperparameter management
- Create adapters for various input data formats
- Implement export functionality for results

Each implementation step should be accompanied by appropriate unit tests and validation procedures to ensure correctness and numerical stability. The implementation should prioritize numerical robustness, especially in the eigendecomposition and matrix inversion operations that are central to the framework.