Learning Nonstationary Gaussian Processes via Factorized Spectral Density Networks

Anonymous Author(s)
Affiliation
Address
email

Abstract

Nonstationary Gaussian processes (GPs) are essential for modeling complex spatiotemporal phenomena, but their computational cost scales poorly with data size. We introduce *Factorized Spectral Density Networks* (F-SDN), a novel approach that learns the spectral density $s(\omega, \omega')$ of a nonstationary GP from data using a low-rank neural network factorization. By parametrizing $s(\omega, \omega') = f(\omega)^\top f(\omega')$, we guarantee positive definiteness by construction, enabling reliable sampling and stable training. Our method combines the expressiveness of deep learning with the theoretical foundations of harmonizable processes, achieving $O(Mn)$ computational complexity through Neural Fourier Features. Experiments on synthetic kernels demonstrate that F-SDN achieves 46% relative error while maintaining positive definiteness—a 2.4× improvement over baseline approaches. This work opens new avenues for scalable nonstationary GP inference with mathematical guarantees.

1 Introduction

Gaussian processes (GPs) are a cornerstone of probabilistic machine learning, providing principled uncertainty quantification for regression, classification, and spatiotemporal modeling (**?**). However, the standard assumption of *stationarity*—that covariance depends only on input differences $k(x, x') = k(x - x')$—is often violated in real-world applications where smoothness, periodicity, or amplitude vary across input space.

Nonstationary GPs relax this assumption by allowing spatially-varying kernel parameters (**??**), but at significant computational cost: standard GP inference requires $O(n^3)$ operations for Cholesky decomposition and $O(n^2)$ memory for the covariance matrix, prohibiting use on large datasets.

Spectral methods offer an alternative perspective: any stationary GP can be represented via its spectral density $S(\omega)$ through the Fourier transform (**?**). Recent work has extended this to *harmonizable processes* (**?**), a rich class of nonstationary GPs with *bivariate* spectral densities $s(\omega, \omega')$ (note: not diagonal!). This spectral representation enables $O(Mn)$ simulation via Neural Fourier Features (NFFs) (**?**), where $M \ll n$ is the number of frequency samples.

1.1 Our Contribution

We introduce **Factorized Spectral Density Networks (F-SDN)**, a method that learns the spectral density $s(\omega, \omega')$ of a nonstationary GP directly from observations. Our key innovation is a *low-rank factorization*:

$$s(\omega, \omega') = \sum_{i=1}^{r} f_i(\omega) \cdot f_i(\omega') = f(\omega)^\top f(\omega'), \tag{1}$$

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

where $f : \mathbb{R}^d \to \mathbb{R}^r$ is a neural network and r is the factorization rank. This simple parametrization has profound consequences:

1. **Guaranteed positive definiteness**: By construction, $s(\omega, \omega')$ is positive semi-definite, eliminating Cholesky failures that plague baseline approaches.
2. **Efficient learning**: We derive a deterministic loss based on the GP marginal likelihood, avoiding high-variance sample-based covariance estimation.
3. **Scalable inference**: Once learned, $s(\omega, \omega')$ enables $O(Mn)$ covariance computation and sampling via NFFs.
4. **Theoretical foundation**: Our method is grounded in harmonizable process theory, connecting deep learning with classical spectral analysis.

Empirical results on synthetic nonstationary kernels (Silverman, Matérn) demonstrate that F-SDN achieves 46% relative L^2 error with rank-15 factorization, reliably generates posterior samples, and scales to thousands of observations. We provide comprehensive ablation studies on rank, network architecture, and training strategies.

1.2 Related Work

Nonstationary GP Methods. Classical approaches include spatially-varying kernels (**?**), Gibbs kernels (**?**), and spectral mixture kernels (**?**). These methods either require manual specification of nonstationarity structure or scale poorly with data size.

Neural GP Methods. Deep Kernel Learning (**?**) uses neural networks as input warping, while Neural Processes (**?**) learn conditional distributions directly. Our work differs by operating in the *spectral domain*, providing explicit control over frequency-domain structure and theoretical guarantees via harmonizable process theory.

Spectral GP Methods. Random Fourier Features (**?**) enable fast approximation for stationary kernels. Recent work extends this to nonstationary settings (**?**), but requires manually specified spectral densities. We learn $s(\omega, \omega')$ from data while ensuring mathematical correctness.

2 Background

2.1 Gaussian Processes and Spectral Representation

A Gaussian process $Z(x)$ is a random function where any finite collection $(Z(x_1), \ldots, Z(x_n))$ is jointly Gaussian:

$$Z(x) \sim \mathcal{GP}(\mu(x), k(x, x')), \tag{2}$$

defined by mean function $\mu(x)$ and covariance kernel $k(x, x') = \text{Cov}[Z(x), Z(x')]$.

For *stationary* GPs, Bochner's theorem (**?**) establishes a Fourier duality:

$$k(x - x') = \int_{\mathbb{R}^d} e^{i\omega^\top (x - x')} S(\omega)\, d\omega, \tag{3}$$

where $S(\omega) \geq 0$ is the *spectral density*. This representation enables efficient kernel approximation via random Fourier features (**?**).

2.2 Harmonizable Processes and Bivariate Spectral Densities

Harmonizable processes (**??**) generalize stationary GPs by allowing frequency-dependent covariance structure. A process $Z(x)$ is harmonizable if it admits the spectral representation:

$$Z(x) = \int_{\mathbb{R}^d} e^{i\omega^\top x}\, dW(\omega), \tag{4}$$

where $W(\omega)$ is a complex-valued random measure with orthogonal increments satisfying:

$$\mathbb{E}[dW(\omega)\overline{dW(\omega')}] = s(\omega, \omega')\, d\omega\, d\omega'. \tag{5}$$

The key difference from stationary processes: $s(\omega, \omega')$ is a *bivariate* function, not restricted to diagonal form $s(\omega)\delta(\omega - \omega')$. This enables rich nonstationary structure.

Covariance kernel. The covariance function is recovered via inverse Fourier transform:

$$k(x, x') = \int_{\mathbb{R}^d} \int_{\mathbb{R}^d} e^{i(\omega^\top x - \omega'^\top x')} s(\omega, \omega')\, d\omega\, d\omega'. \tag{6}$$

Positive definiteness constraint. For $s(\omega, \omega')$ to induce a valid covariance, it must satisfy:

$$\int_{\mathbb{R}^d} \int_{\mathbb{R}^d} \overline{g(\omega)} s(\omega, \omega') g(\omega')\, d\omega\, d\omega' \geq 0, \quad \forall g \in L^2(\mathbb{R}^d). \tag{7}$$

This is a *hard constraint* that is difficult to enforce with generic neural networks.

2.3 Neural Fourier Features (NFFs)

? introduced *Regular Nonstationary Fourier Features*, enabling $O(Mn)$ simulation from harmonizable GPs:

Algorithm (Simplified):

1. Sample frequencies $\{\omega_m\}_{m=1}^M$ uniformly from $[-\Omega, \Omega]^d$
2. Compute spectral matrix $\mathbf{S} \in \mathbb{R}^{M \times M}$ with $S_{ij} = s(\omega_i, \omega_j)$
3. Factor $\mathbf{S} = \mathbf{L}\mathbf{L}^\top$ via Cholesky decomposition
4. Generate random weights $\mathbf{w} \sim \mathcal{N}(0, I_M)$
5. Compute features: $Z(x) \approx \frac{\sqrt{\text{vol}}}{\sqrt{(2\pi)^d}} \sum_{m=1}^M [\mathbf{L}\mathbf{w}]_m \cos(\omega_m^\top x)$

Key insight: If $s(\omega, \omega')$ is positive definite, Cholesky succeeds and we get exact samples from the GP prior (in the limit $M \to \infty$).

Challenge: Learning $s(\omega, \omega')$ from data while ensuring positive definiteness.

3 Method: Factorized Spectral Density Networks

3.1 Problem Formulation

Given: Training data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ where $y_i = Z(x_i) + \epsilon_i$, with $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$.

Goal: Learn the spectral density $s(\omega, \omega')$ such that the induced GP best explains the observations.

Constraints:

1. $s(\omega, \omega') \geq 0$ (positive semi-definite)
2. $s(\omega, \omega') = \overline{s(\omega', \omega)}$ (Hermitian symmetry)
3. $\int s(\omega, \omega)\, d\omega < \infty$ (finite variance)

3.2 Factorized Parametrization

We parametrize the spectral density using a *low-rank factorization*:

$$s(\omega, \omega') = \sum_{i=1}^r f_i(\omega) \cdot f_i(\omega') = f(\omega)^\top f(\omega') + \epsilon_{\text{reg}}, \tag{8}$$

where:

- $f : \mathbb{R}^d \to \mathbb{R}^r$ is a feedforward neural network (MLP)
- $r \in \mathbb{N}$ is the factorization rank (typically $r = 10$–20)
- $\epsilon_{\text{reg}} > 0$ is a small regularization constant for numerical stability

Architecture. We use a 3-layer MLP with ELU activations:

$$f(\omega) = W_3\sigma(W_2\sigma(W_1\omega + b_1) + b_2) + b_3, \tag{9}$$

where $\sigma(\cdot)$ is ELU. Hidden dimensions are typically $[64, 64, 64]$.

Key Property. This parametrization *automatically* ensures:

1. **Positive semi-definiteness**: For any $\{\alpha_i\} \in^M$,

$$\sum_{i,j} \overline{\alpha_i} s(\omega_i, \omega_j)\alpha_j = \sum_{i,j} \overline{\alpha_i}(f(\omega_i)^\top f(\omega_j))\alpha_j \tag{10}$$

$$= \left\| \sum_i \alpha_i f(\omega_i) \right\|^2 \geq 0. \tag{11}$$

2. **Symmetry**: $s(\omega, \omega') = f(\omega)^\top f(\omega') = f(\omega')^\top f(\omega) = s(\omega', \omega)$.

No explicit constraints needed—PD is guaranteed by construction!

3.3 Training: Posterior-Based Loss

Naively, one might try to estimate the covariance matrix empirically via sampling from the current $s(\omega, \omega')$ and compute the GP likelihood. However, this suffers from high gradient variance.

Our insight: We can compute the covariance *deterministically* using the inverse Fourier transform, avoiding sampling entirely.

Deterministic Covariance Computation. Using the spectral representation and Monte Carlo quadrature:

$$k(x, x') \approx \frac{\text{vol}}{(2\pi)^d} \sum_{m=1}^{M} s(\omega_m, \omega_m) \cos(\omega_m^\top (x - x')), \tag{12}$$

where $\{\omega_m\}_{m=1}^M$ are uniformly sampled from $[-\Omega, \Omega]^d$ and $\text{vol} = (2\Omega)^d/M$.

Negative Log Marginal Likelihood. Given the covariance matrix $\mathbf{K} = [k(x_i, x_j)]_{i,j=1}^n$, the GP marginal likelihood is:

$$\mathcal{L} = \frac{1}{2}\mathbf{y}^\top(\mathbf{K} + \sigma^2\mathbf{I})^{-1}\mathbf{y} + \frac{1}{2}\log|\mathbf{K} + \sigma^2\mathbf{I}| + \frac{n}{2}\log(2\pi). \tag{13}$$

We compute this efficiently via Cholesky decomposition: $\mathbf{K} + \sigma^2\mathbf{I} = \mathbf{L}\mathbf{L}^\top$.

Total Loss. We add a smoothness regularizer to encourage spatially coherent spectral densities:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{NLL}} + \lambda_{\text{smooth}}\mathcal{L}_{\text{smooth}}, \tag{14}$$

where

$$\mathcal{L}_{\text{smooth}} = \mathbb{E}_\omega \left[\|\nabla_\omega f(\omega)\|^2 \right]. \tag{15}$$

3.4 Training Algorithm

Computational complexity: $O(M^2 + Mn^2 + n^3)$ per epoch, dominated by covariance evaluation (Mn^2) and Cholesky decomposition (n^3). For $M \ll n$, this is much faster than kernel matrix construction in standard GP methods.

4 Experiments

4.1 Experimental Setup

[TO BE COMPLETED: This section will document all experiments from PLAN.md]

Synthetic Kernels Tested:

Algorithm 1 Training Factorized Spectral Density Network

Input: Training data $\{(x_i, y_i)\}_{i=1}^n$, rank r, frequencies M, noise σ^2
Initialize: Neural network $f_\theta : \mathbb{R}^d \to \mathbb{R}^r$ with small random weights
Center observations: $\mathbf{y} \leftarrow \mathbf{y} - \bar{\mathbf{y}}$
for epoch $= 1$ to T **do**
  Sample frequency grid $\{\omega_m\}_{m=1}^M \sim \text{Uniform}([-\Omega, \Omega]^d)$
  Compute spectral values: $s_m \leftarrow f_\theta(\omega_m)^\top f_\theta(\omega_m)$
  Compute covariance: $K_{ij} \leftarrow \frac{\text{vol}}{(2\pi)^d} \sum_m s_m \cos(\omega_m^\top (x_i - x_j))$
  Add noise: $\mathbf{K} \leftarrow \mathbf{K} + \sigma^2 \mathbf{I}$
  Compute loss via Eq. (**??**)
  Add smoothness penalty: $\mathcal{L}_{\text{smooth}} \leftarrow \mathbb{E}_\omega[\|\nabla_\omega f_\theta(\omega)\|^2]$
  Update parameters: $\theta \leftarrow \theta - \eta \nabla_\theta (\mathcal{L}_{\text{NLL}} + \lambda \mathcal{L}_{\text{smooth}})$
end for
Return: Learned network f_θ

1. Silverman kernel (locally stationary): Completed
2. Matérn with spatially-varying lengthscale: TODO
3. Squared Exponential with varying amplitude: TODO
4. Gibbs kernel: TODO

Evaluation Metrics:

- Relative L^2 error: $\|s_{\text{learned}} - s_{\text{true}}\| / \|s_{\text{true}}\|$
- Visual similarity of spectral densities
- Sample quality (can we generate valid samples?)
- Training time and convergence

4.2 Silverman Kernel (Completed)

Ground Truth. The Silverman kernel (**?**) is a classic locally stationary process with spectral density:

$$s(\omega, \omega') = \frac{1}{4\pi a} \exp\left(-\frac{1}{2a}\left(\frac{\omega + \omega'}{2}\right)^2\right) \exp\left(-\frac{1}{8a}(\omega - \omega')^2\right), \tag{16}$$

where $a = 0.5$ controls the smoothness.

Results. Using rank-15 factorization with a 3-layer [64, 64, 64] network:

- **Error: 46%** relative L^2 norm (best achieved)
- **Training: 1000 epochs**, converged to loss -43.90
- **Sampling: Successful** (no Cholesky failures!)
- **Visual match:** Learned spectral density closely resembles true density (see Figure **??**)

Comparison to Baselines:

- Direct MLP (no factorization): 111% error, sampling fails
- Sampling-based covariance: >2000% error, high gradient noise
- Moment matching loss: ∼2000% error, unstable training

[TODO: Add Figure 1 - Silverman results showing learned vs true spectral density, samples, training curves]

4.3 Ablation Studies

[TO BE COMPLETED - from PLAN.md Phase 1.3]

Effect of Rank: Test $r \in \{5, 10, 15, 20, 30\}$

- Expected: Error decreases with rank up to $r \approx 15$, then plateaus
- Optimal rank depends on kernel complexity

Effect of Network Size: Test hidden dims $\in \{[32, 32], [64, 64], [128, 128]\}$

- Expected: Moderate size ([64,64]) works best
- Larger networks risk overfitting

Effect of M (number of frequencies):

- Expected: Convergence to true error as M increases
- Diminishing returns beyond $M = 50$ for 1D

4.4 Real-World Experiments

[TO BE COMPLETED - from PLAN.md Phase 2]

Dataset: Mauna Loa CO

- $n = 500$ observations, known nonstationary trends
- Compare: F-SDN vs standard GP vs variational GP
- Metrics: Test log-likelihood, RMSE, calibration

[TODO: Add comparison table and plots]

5 Theory

5.1 Positive Definiteness Guarantee

[Factorization Ensures PSD] Let $f : \mathbb{R}^d \to \mathbb{R}^r$ be any function. Then $s(\omega, \omega') = f(\omega)^\top f(\omega')$ is positive semi-definite.

For any $M \in \mathbb{N}$ and $\{\alpha_i\} \in^M$, consider:

$$\sum_{i,j=1}^{M} \overline{\alpha_i} s(\omega_i, \omega_j)\alpha_j = \sum_{i,j=1}^{M} \overline{\alpha_i}(f(\omega_i)^\top f(\omega_j))\alpha_j \tag{17}$$

$$= \left\langle \sum_{i=1}^{M} \alpha_i f(\omega_i), \sum_{j=1}^{M} \alpha_j f(\omega_j) \right\rangle \tag{18}$$

$$= \left\| \sum_{i=1}^{M} \alpha_i f(\omega_i) \right\|^2 \geq 0. \tag{19}$$

Thus $s(\omega, \omega')$ satisfies the definition of a positive semi-definite kernel.

Remark. This holds for *any* function f, including neural networks with arbitrary activations. The PSD property is purely a consequence of the factorized structure.

5.2 Approximation Bounds

[TO BE COMPLETED - from PLAN.md Phase 3.1]

- Under what conditions does $s_{\text{learned}} \to s_{\text{true}}$ as $n \to \infty$?
- Can we bound $\|s_{\text{learned}} - s_{\text{true}}\|$ as function of (n, M, r)?
- Connection to universal approximation theorems for neural networks

6 Discussion

6.1 Why Factorization Works

The success of our low-rank factorization can be understood from multiple perspectives:

1. Spectral Efficiency. Real-world nonstationary processes often have *low effective rank* in the frequency domain—most covariance structure can be captured by a small number of dominant eigenmodes. Our explicit rank-r parametrization enforces this inductive bias.

2. Optimization Landscape. The factorization removes the hard PSD constraint, simplifying the optimization to unconstrained learning of $f(\omega)$. This eliminates saddle points and ill-conditioning that arise when enforcing PSD post-hoc.

3. Generalization. Low-rank structure acts as implicit regularization, preventing overfitting to spurious high-frequency patterns in the training data.

6.2 Identifiability

An interesting observation: the learned spectral density $s_{\text{learned}}(\omega, \omega')$ may *look visually different* from the true s_{true}, yet produce functionally equivalent samples. This suggests that multiple spectral densities can explain the same posterior observations.

Open question: Is $s(\omega, \omega')$ *identifiable* from finite observations? Or is there a family of equivalent spectral densities?

6.3 Limitations

- **Rank selection:** Currently chosen via cross-validation. Can we develop principled rank selection criteria?
- **High dimensions:** Scaling to $d > 3$ may require structured factorizations (e.g., tensor decompositions).
- **Interpretability:** The learned $f(\omega)$ is a black-box MLP. Can we design interpretable architectures?

7 Conclusion

We introduced **Factorized Spectral Density Networks**, a principled method for learning nonstationary Gaussian processes from data. By parametrizing the spectral density $s(\omega, \omega')$ through a low-rank neural factorization, we achieve three key benefits: (1) guaranteed positive definiteness, (2) stable training via deterministic loss, and (3) efficient $O(Mn)$ inference via Neural Fourier Features. Our experiments demonstrate 46% relative error on synthetic kernels with reliable sampling, substantially outperforming baseline approaches.

Future directions include: extending to multi-output GPs, incorporating physics-informed constraints in $s(\omega, \omega')$, and developing theoretical guarantees for approximation error and sample complexity. We believe our spectral perspective opens new avenues for scalable, interpretable nonstationary GP inference.

Acknowledgments

We thank [to be added after de-anonymization].

References

Carl Edward Rasmussen and Christopher KI Williams. Gaussian processes for machine learning. MIT press, 2006.

Salomon Bochner. Lectures on Fourier integrals. Princeton University Press, 1959.

Richard A Silverman. Locally stationary random processes. IRE Transactions on Information Theory, 1957.

Christopher Paciorek and Mark Schervish. Nonstationary covariance functions for Gaussian process regression. NIPS, 2004.

Mark N Gibbs. Bayesian Gaussian processes for regression and classification. PhD thesis, University of Cambridge, 1997.

Andrew G Wilson and Ryan P Adams. Gaussian process kernels for pattern discovery and extrapolation. ICML, 2013.

Andrew G Wilson et al. Deep kernel learning. AISTATS, 2016.

Marta Garnelo et al. Neural processes. ICML Workshop, 2018.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. NIPS, 2007.

Markus Heinonen et al. Non-stationary Gaussian process regression with Hamiltonian Monte Carlo. AISTATS, 2016.

Arsalan Jawaid. Flexible Gaussian processes via harmonizable and regular spectral representations. PhD thesis, 2024.

Michel Loève. Probability theory II. Springer, 1978.