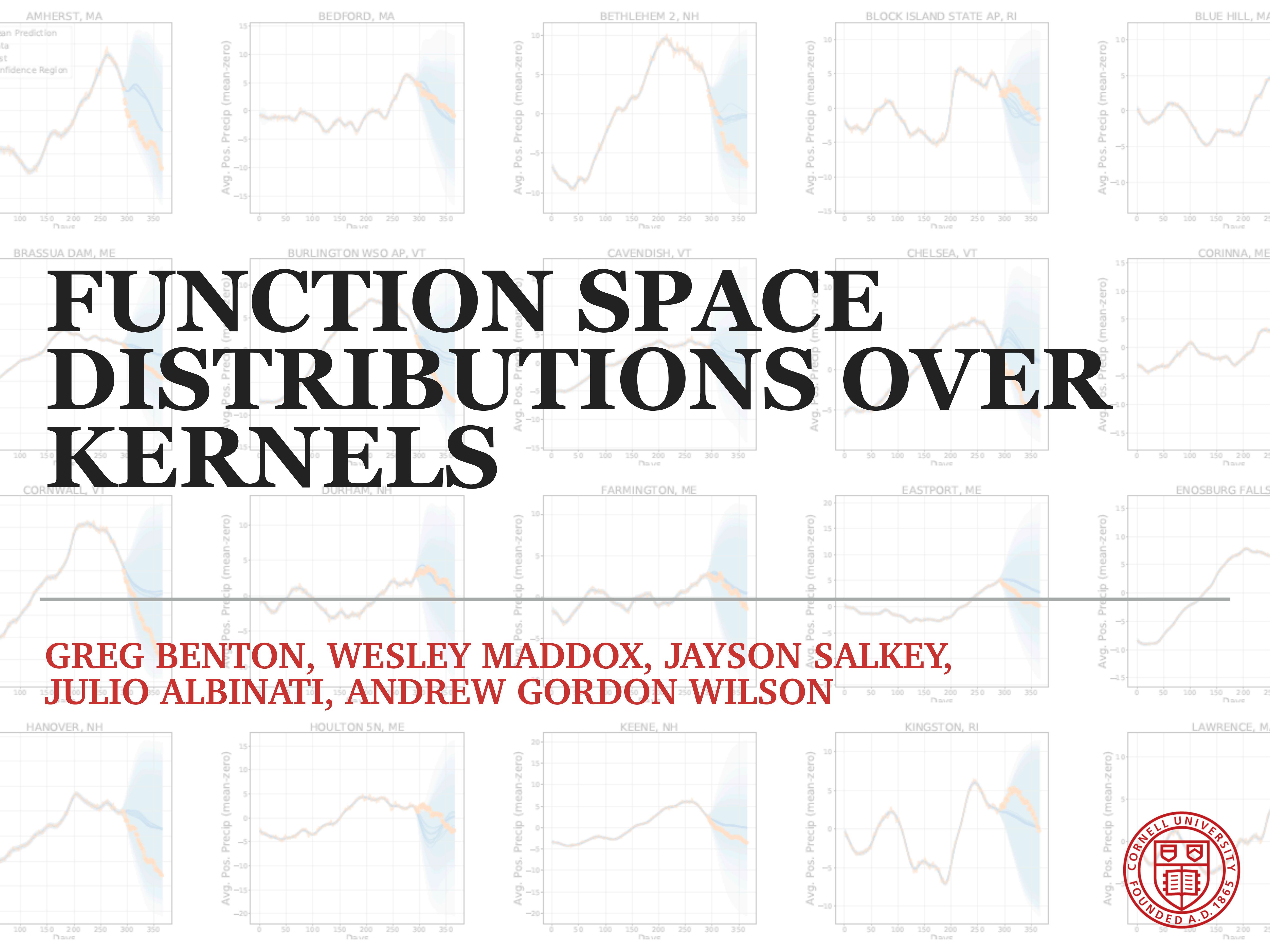


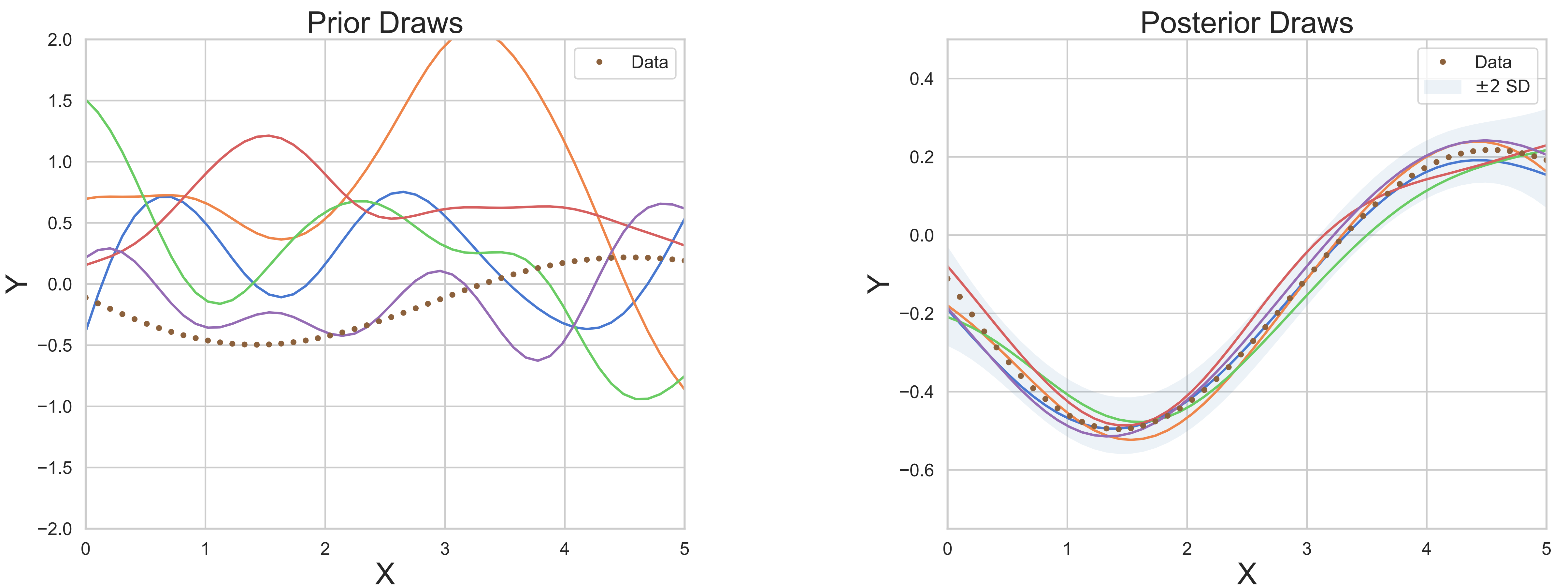
FUNCTION SPACE DISTRIBUTIONS OVER KERNELS

GREG BENTON, WESLEY MADDOX, JAYSON SALKEY, JULIO ALBINATI, ANDREW GORDON WILSON

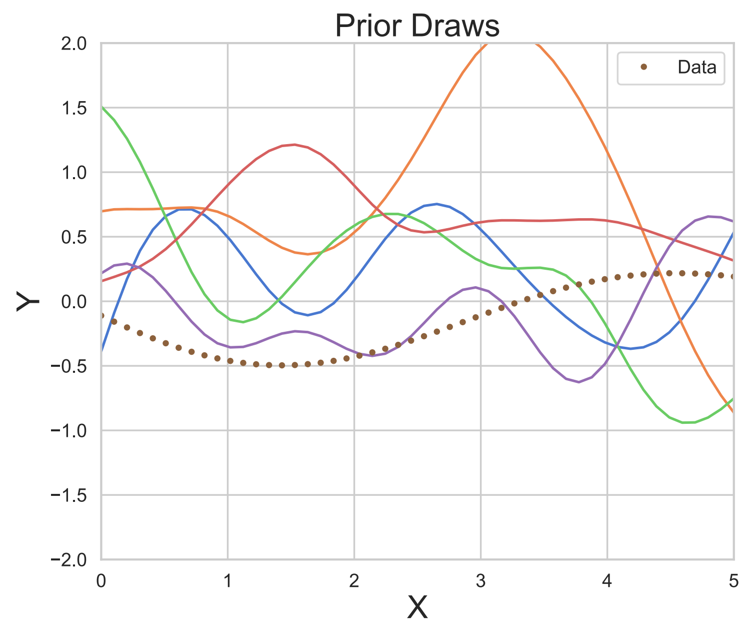
HIGH LEVEL IDEA

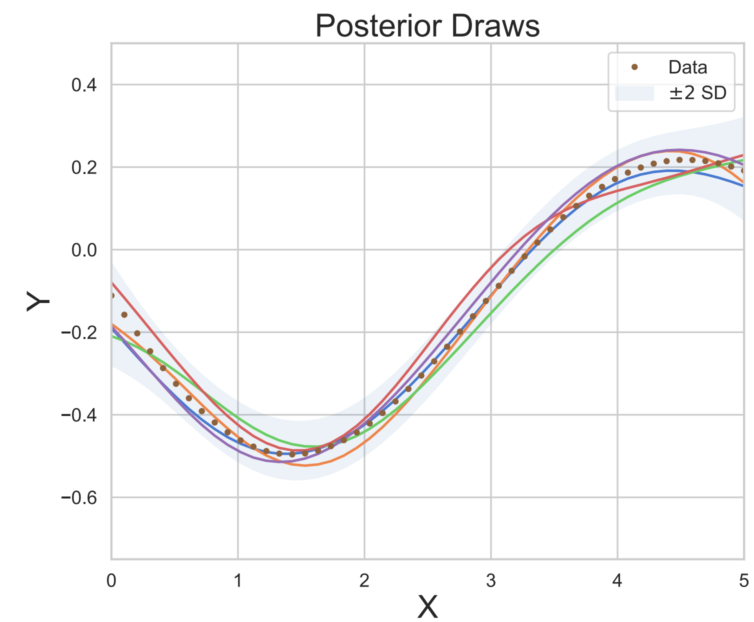
- Gaussian Process (GP): stochastic process for which any finite collection of points is jointly normal
- $k(x, x')$ a kernel function describing covariance

$$y(x) \sim GP(\mu(x), k(x, x'))$$

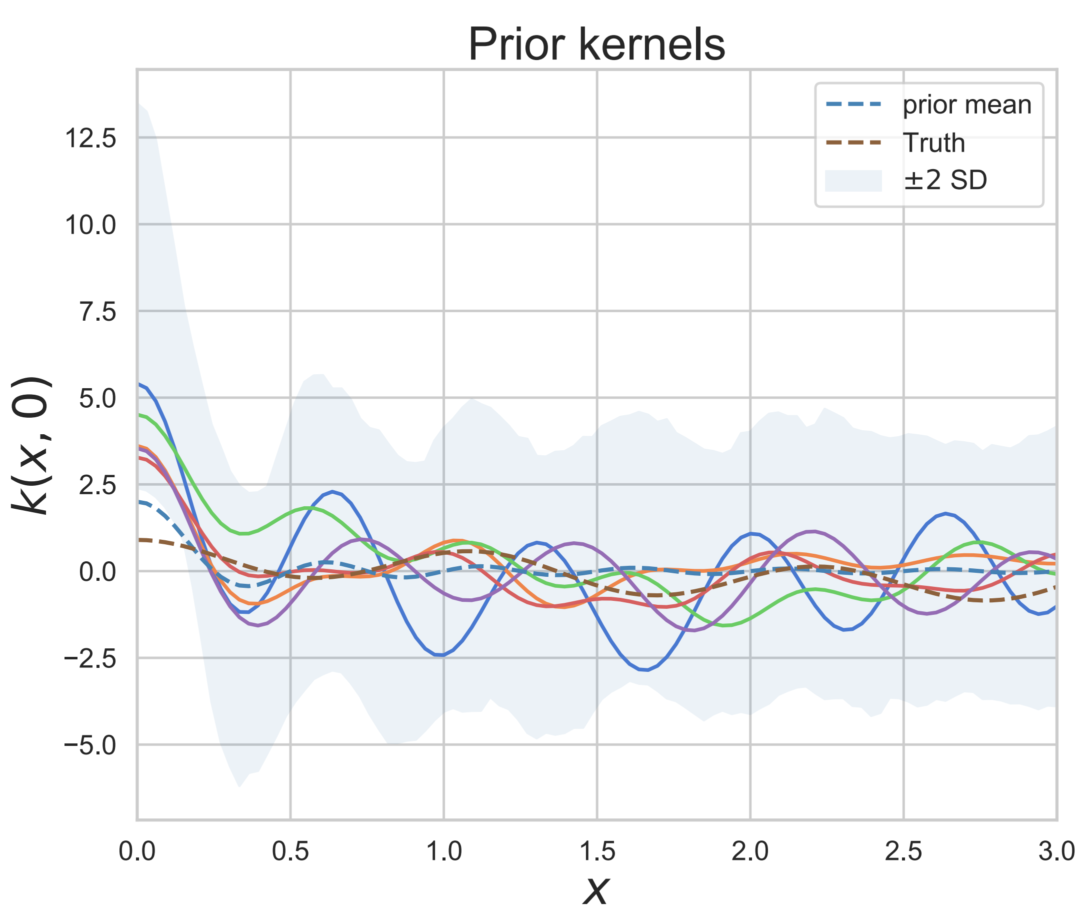


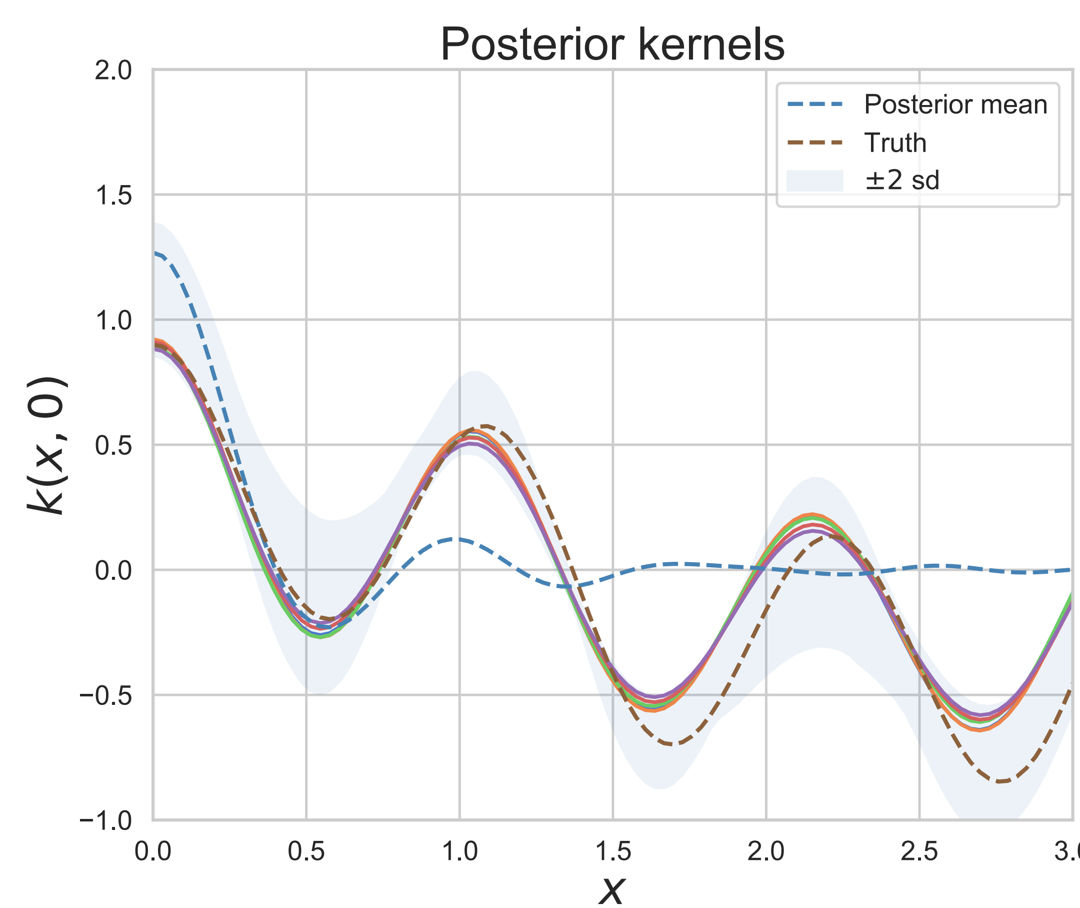
HIGH LEVEL IDEA





$$y(x) \sim GP(\mu(x), k(x, x'))$$





OUTLINE

- Introduction
    - Mathematical Foundation
    - Model Specification
    - Inference Procedure

OUTLINE

- Introduction
- Experimental Results
  - Recovery of known kernels
  - Interpolation and extrapolation of real data

OUTLINE

- Introduction
- Experimental Results
- Extension to multi-task time-series
    - Precipitation data

BOCHNER'S THEOREM

- If $k(x, x') = k(\tau)$ then we can represent $k(\tau)$ via its *spectral density:*

$$k(\tau) = \int_{\mathbb{R}} e^{2\pi i \omega \tau} S(\omega) d\omega$$

- Learning the spectral representation of $k(\tau)$ is sufficient to learn the entire kernel

BOCHNER'S THEOREM

- If $k(x, x') = k(\tau)$ then we can represent $k(\tau)$ via its *spectral density:*

$$k(\tau) = \int_{\mathbb{R}} e^{2\pi i \omega \tau} S(\omega) d\omega$$

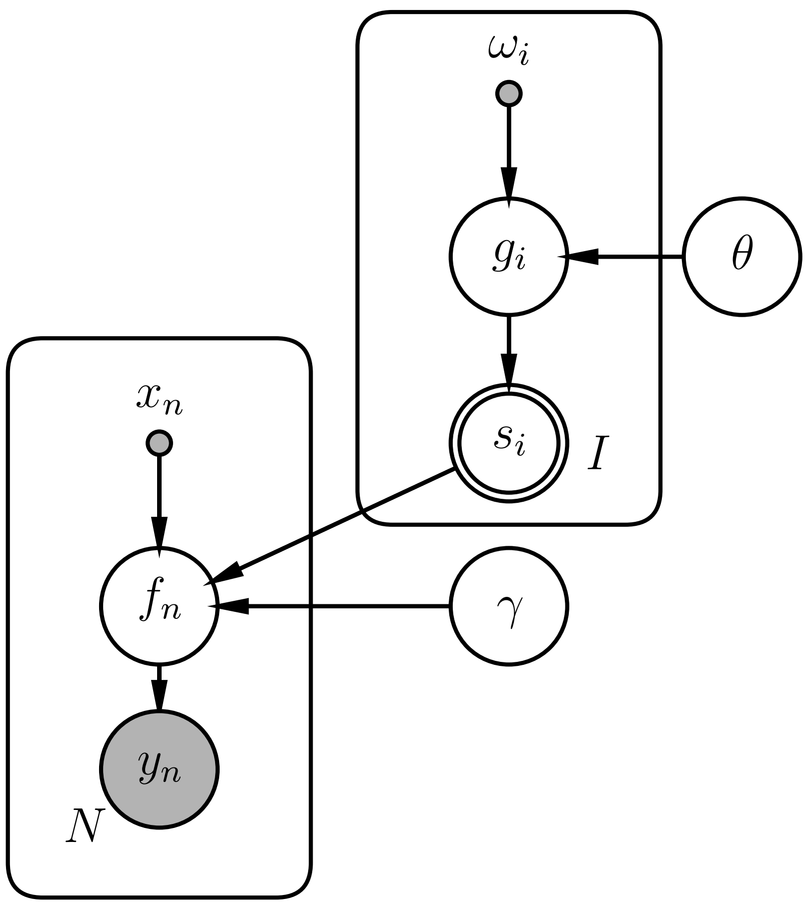
- Learning the spectral representation of $k(\tau)$ is sufficient to learn the entire kernel

- Assuming $k(\tau)$ is symmetric and data are finitely sampled, the reconstruction simplifies to:

$$k(\tau) = \int_{[0,\pi/\Delta)} \cos(2\pi\tau\omega) S(\omega) d\omega$$

FUNCTIONAL KERNEL LEARNING

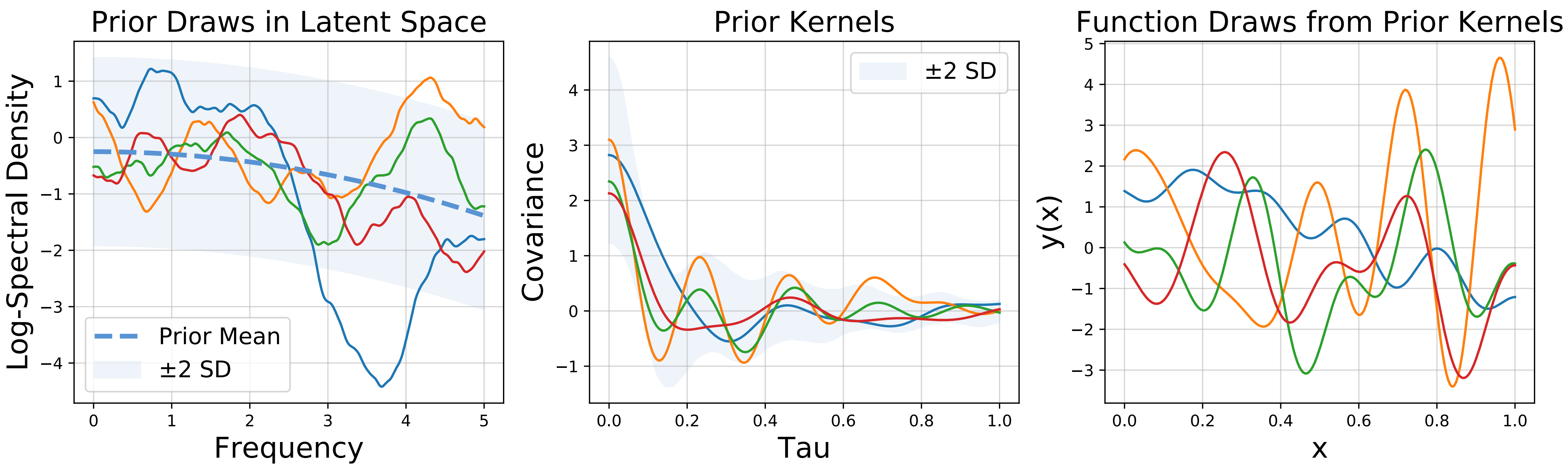
Graphical Model



Hyper-prior	$p(\phi) = p(\theta, \gamma)$
Latent GP	$g(\omega) \,\vert\, \theta \sim GP\left(\mu(\omega;\theta), k_g(\omega, \omega';\theta)\right)$
Spectral Density	$S(\omega) = \exp\{g(\omega)\}$
Data GP	$f(x) \,\vert\, S(\omega), \gamma \sim GP(\gamma_0, k(\tau; S(\omega)))$

FUNCTIONAL KERNEL LEARNING

Hyper-prior	$p(\phi) = p(\theta, \gamma)$
Latent GP	$g(\omega) \mid \theta \sim GP\left(\mu(\omega;\theta), k_g(\omega, \omega';\theta)\right)$
Spectral Density	$S(\omega) = \exp\{g(\omega)\}$
Data GP	$f(x) \mid S(\omega), \gamma \sim GP(\gamma_0, k(\tau; S(\omega)) + \gamma_1 \delta_{\tau=0})$



LATENT MODEL

- Mean of latent GP is log of RBF spectral density

$$\mu(\omega;\theta) = \theta_0 - \frac{\omega^2}{2\tilde{\theta}_1^{\,2}}$$

- Covariance is Matérn with $\nu = 1.5$

$$k_g(\omega,\omega';\theta) = \frac{2^{1-\nu}}{\Gamma(\nu)}\left(\sqrt{2\nu}\frac{|\omega-\omega'|}{\tilde{\theta}_2}\right) K_\nu\left(\sqrt{2\nu}\frac{|\omega-\omega'|}{\tilde{\theta}_2}\right) + \tilde{\theta}_3\delta_{\tau=0}$$

$\tilde{\theta}_i = \mathrm{softmax}(\theta_i)$

INFERENCE

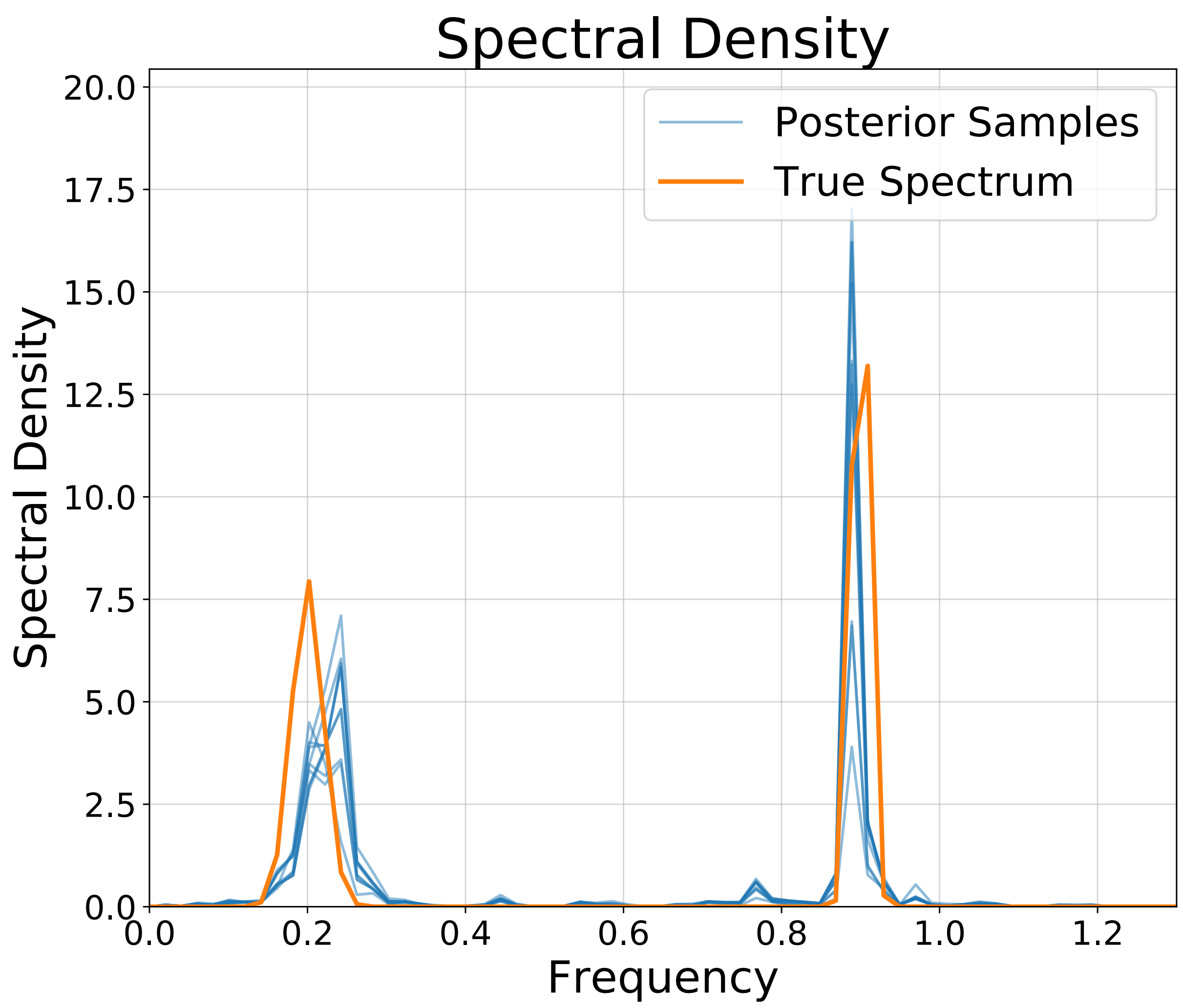
- Need to update the hyper parameters ϕ and the latent GP $g(\omega)$
- Initialize $g(\omega)$ to the log-periodogram of the data
- Alternate:
  - Fix $g(\omega)$ and use Adam to update ϕ
  - Fix ϕ and use elliptical slice sampling to draw samples of $g(\omega)$

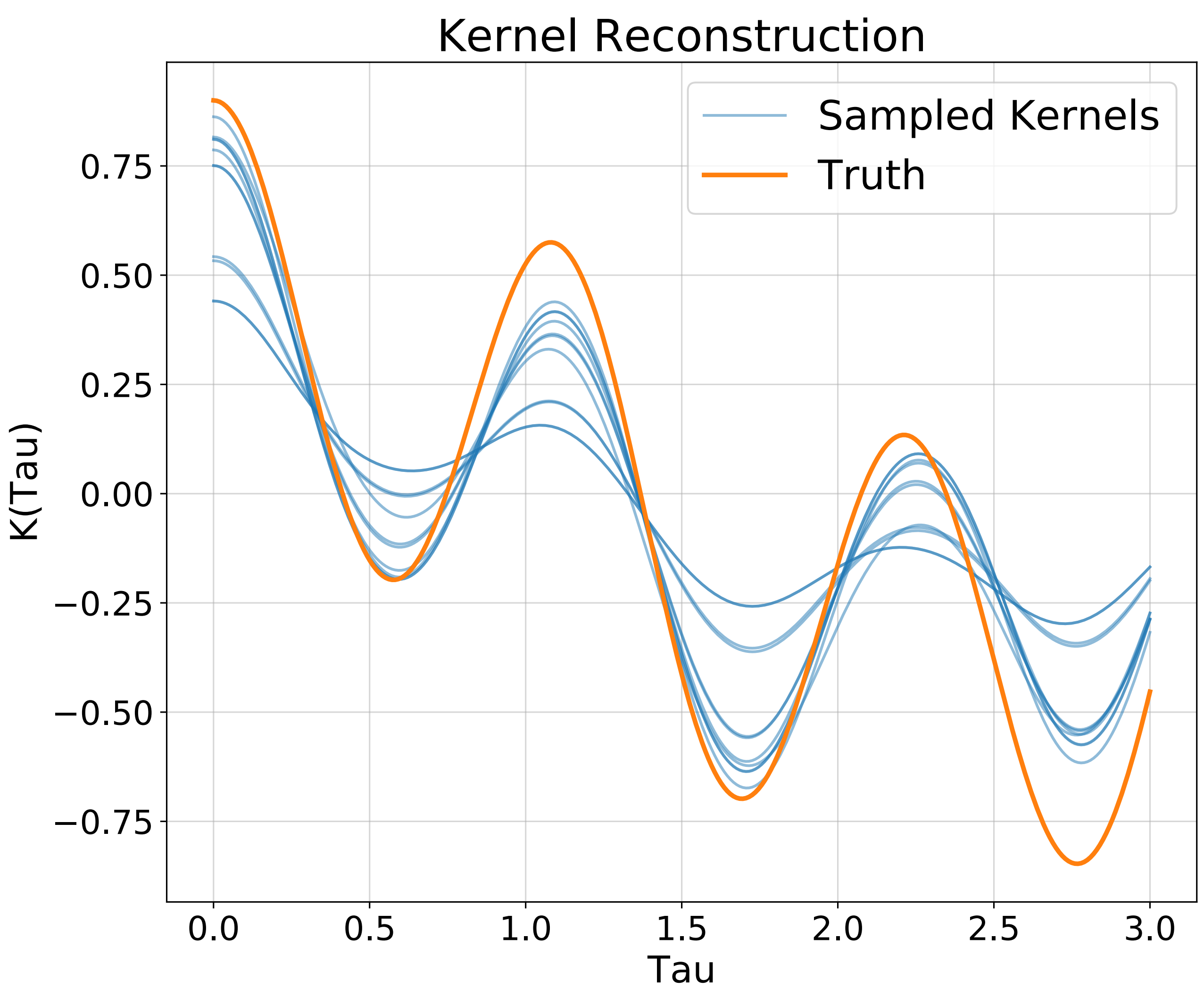
OUTLINE

- Introduction
- Experimental Results
  - Recovery of known kernels
  - Interpolation and extrapolation of real data

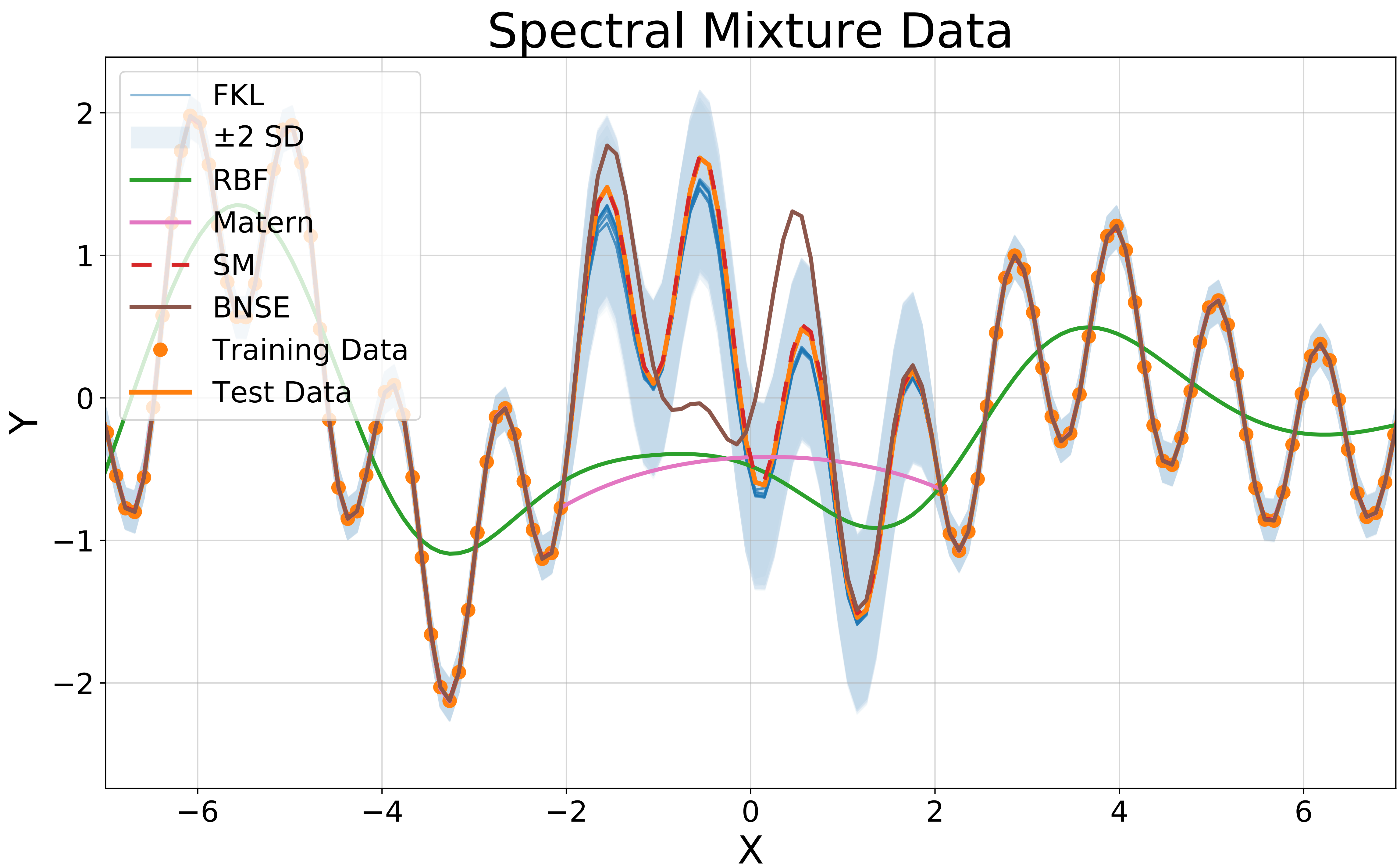
DATA FROM A SPECTRAL MIXTURE KERNEL

- Generative kernel has mixture of Gaussians as spectral density

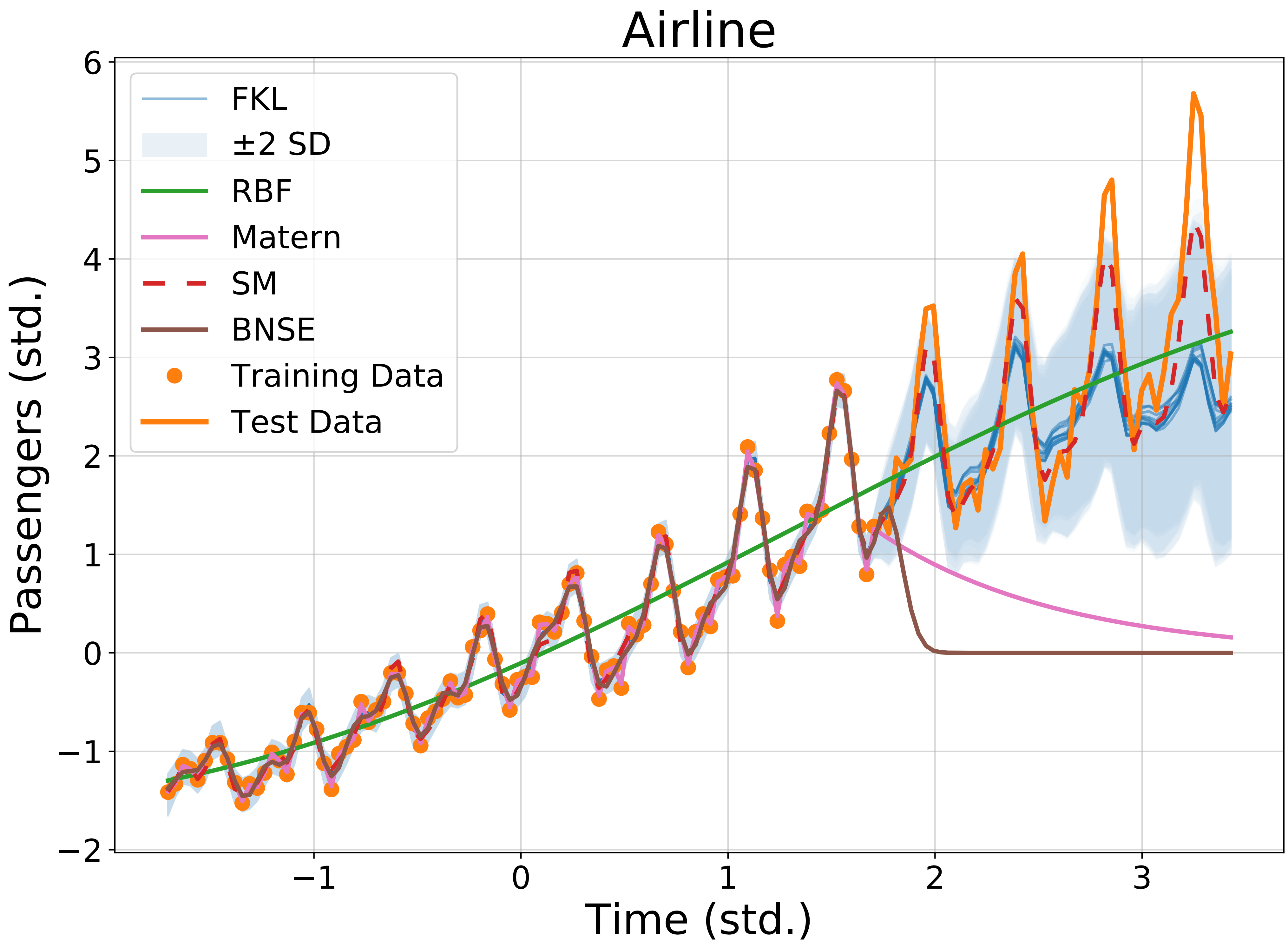




DATA FROM A SPECTRAL MIXTURE KERNEL



AIRLINE PASSENGER DATA



OUTLINE

- Introduction
- Experimental Results
- Extension to multi-task time-series
  - Precipitation data

MULTIPLE TIME SERIES

- Can 'link' multiple time series by sharing the latent GP across outputs
- Let $g^t(\omega)$ denote the t^{th} realization of the latent GP and $f_t(x)$ be the GP over the t^{th} time-series

Hyper-prior	$p(\phi) = p(\theta, \gamma)$
Latent GP	$g(\omega) \mid \theta \sim GP\left(\mu(\omega; \theta), k_g(\omega, \omega'; \theta)\right)$
t^{th} Spectral Density	$S^t(\omega) = \exp\{g^t(\omega)\}$
GP for t^{th} task	$f_t(x) \mid S(\omega), \gamma \sim GP(\gamma_0, k(\tau; S^t(\omega)) + \gamma_1 \delta_{\tau=0})$

MULTIPLE TIME SERIES

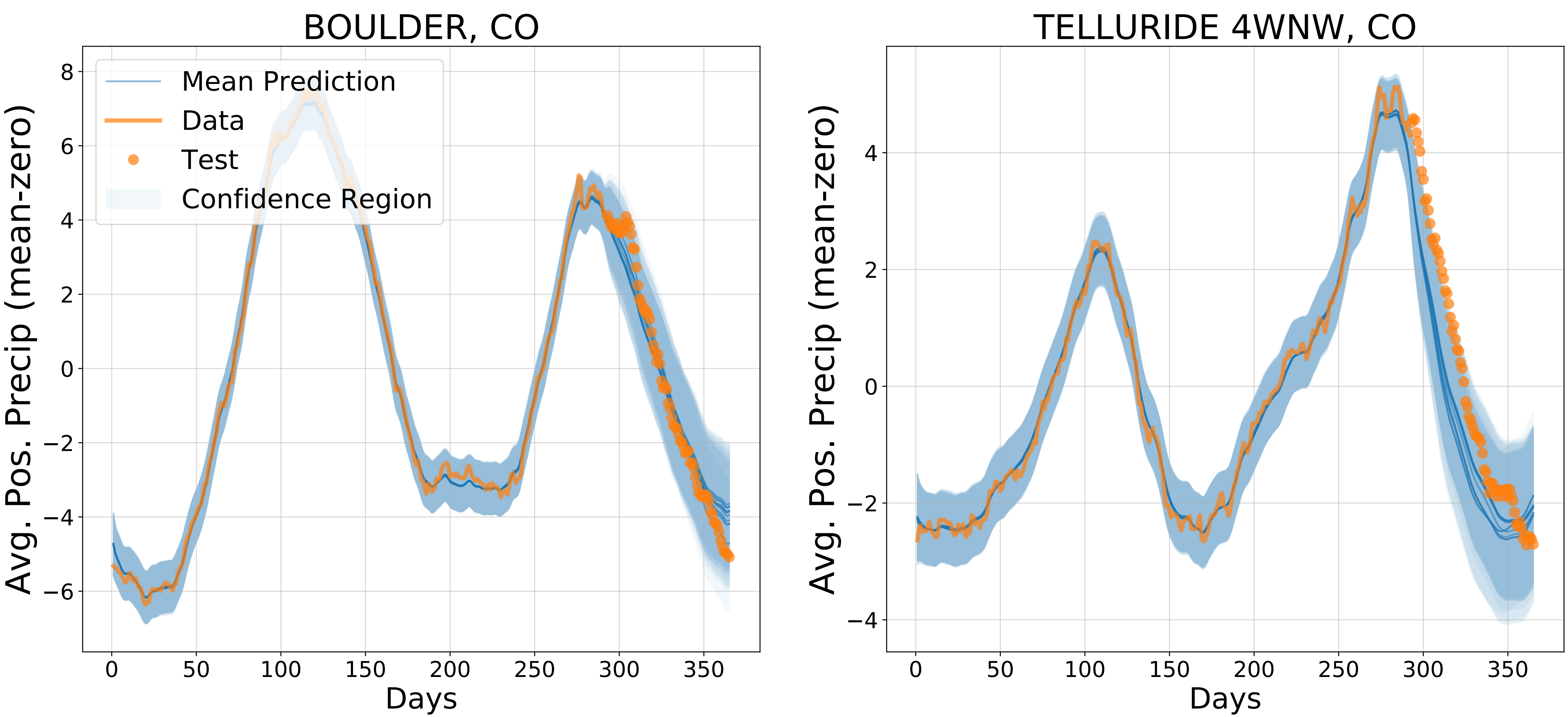
- Can 'link' multiple time series by sharing the latent GP across outputs
- Let $g^t(\omega)$ denote the t^{th} realization of the latent GP and $f_t(x)$ be the GP over the t^{th} time-series

Hyper-prior	$p(\phi) = p(\theta, \gamma)$
Latent GP	$g(\omega) \mid \theta \sim GP\left(\mu(\omega;\theta), k_g(\omega, \omega'; \theta)\right)$
t^{th} Spectral Density	$S^t(\omega) = \exp\{g^t(\omega)\}$
GP for t^{th} task	$f_t(x) \mid S(\omega), \gamma \sim GP(\gamma_0, k(\tau; S^t(\omega)) + \gamma_1 \delta_{\tau=0})$

- Test this on data from USHCN, daily precipitation values from continental US
    - Inductive bias: yearly precipitation for climatologically similar regions should have similar covariance, similar spectral densities

PRECIPITATION DATA

Ran on two climatologically similar locations

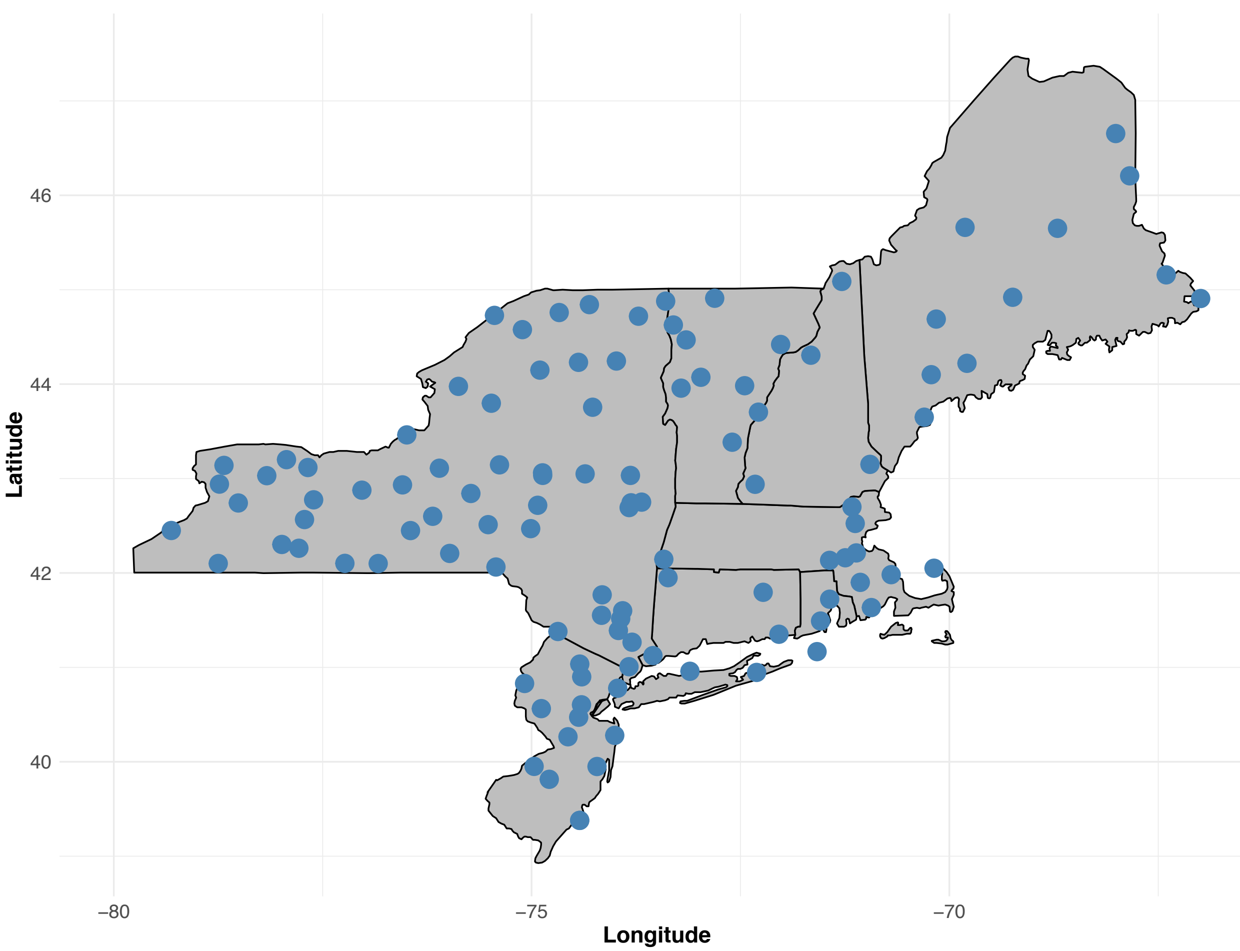


PRECIPITATION DATA

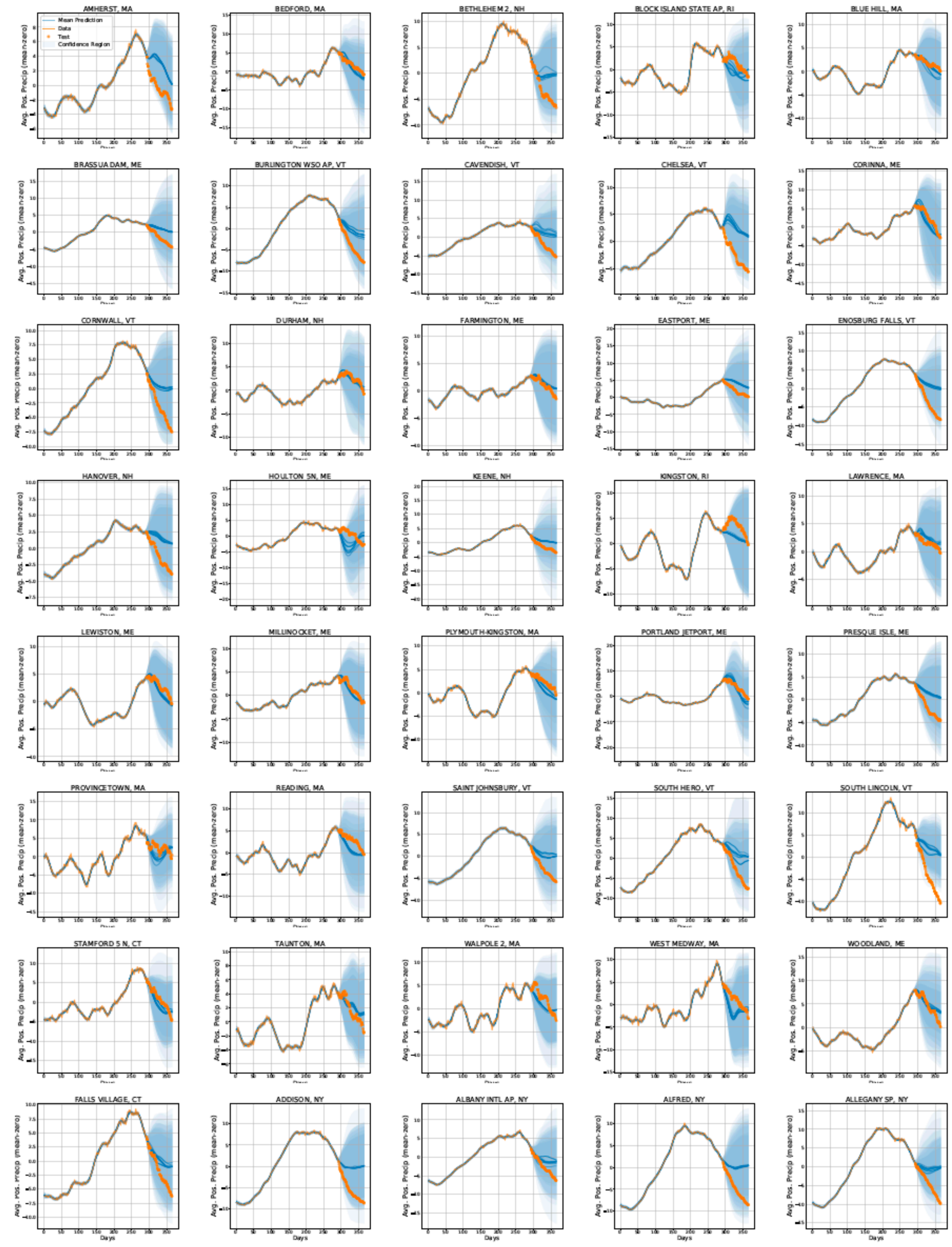
Used 108 locations across the Northeast USA

Each station, n = 300

Total: 300 * 108 = 32,400 data points



Locations Used



Here's 48 of them…

CONCLUSION

- FKL: Nonparametric, function-space view of kernel learning
- Can express *any* stationary kernel with uncertainty representation
- GPyTorch Code: https://github.com/wjmaddox/spectralgp

Link to Code

CONCLUSION

- FKL: Nonparametric, function-space view of kernel learning
- Can express *any* stationary kernel with uncertainty representation
- GPyTorch Code: https://github.com/wjmaddox/spectralgp

Link to Code

QUESTIONS?

- Poster 52

REFERENCES

Spectral Mixture Kernels: Wilson, Andrew, and Ryan Adams. "Gaussian process kernels for pattern discovery and extrapolation." *International Conference on Machine Learning*. 2013.

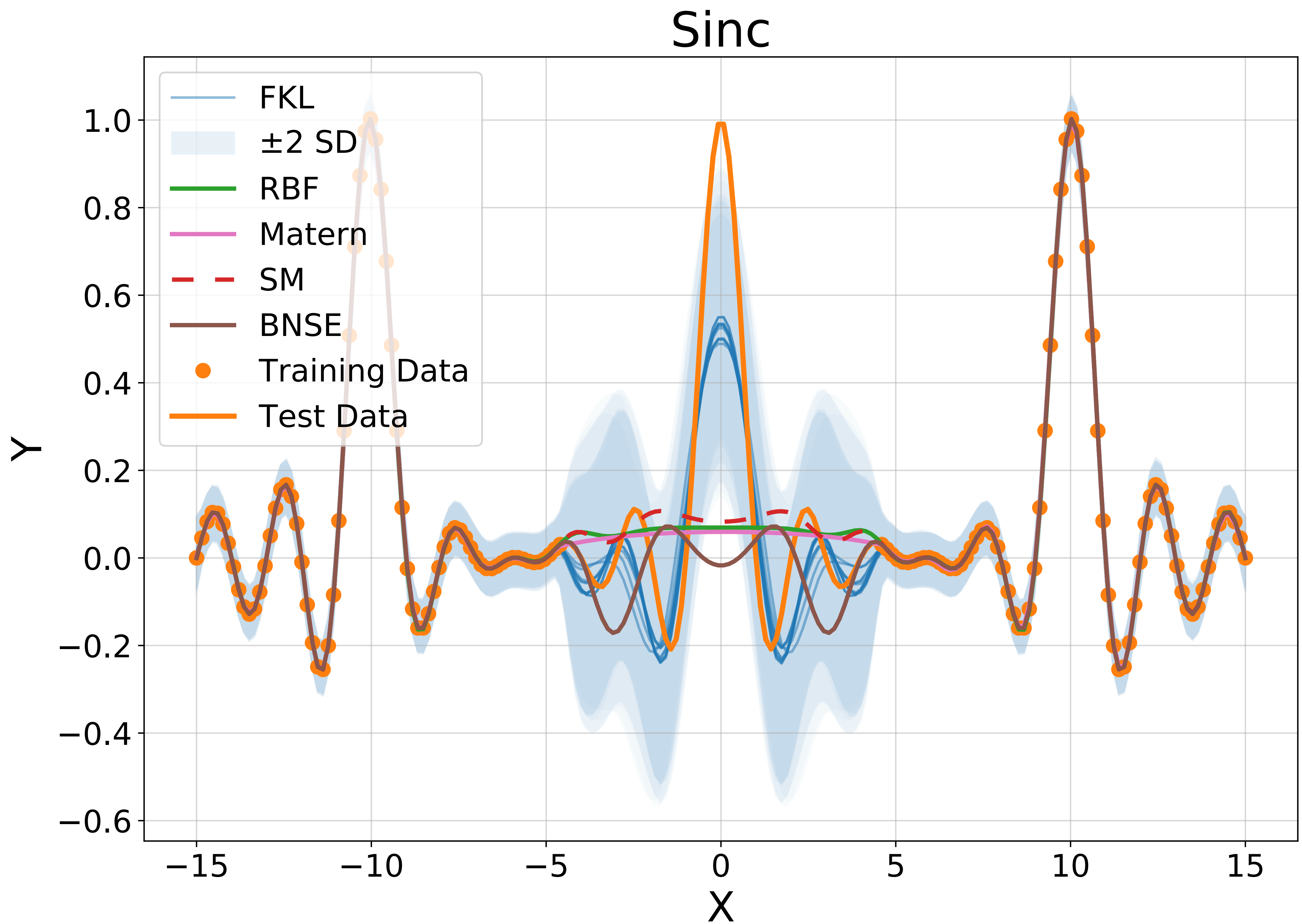
BNSE: Tobar, Felipe. "Bayesian Nonparametric Spectral Estimation." *Advances in Neural Information Processing Systems*. 2018.

Elliptical Slice Sampling: Murray, Iain, Ryan Adams, and David MacKay. "Elliptical slice sampling." *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. 2010.

GPyTorch: Gardner, Jacob, et al. "Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration." Advances in Neural Information Processing Systems. 2018.

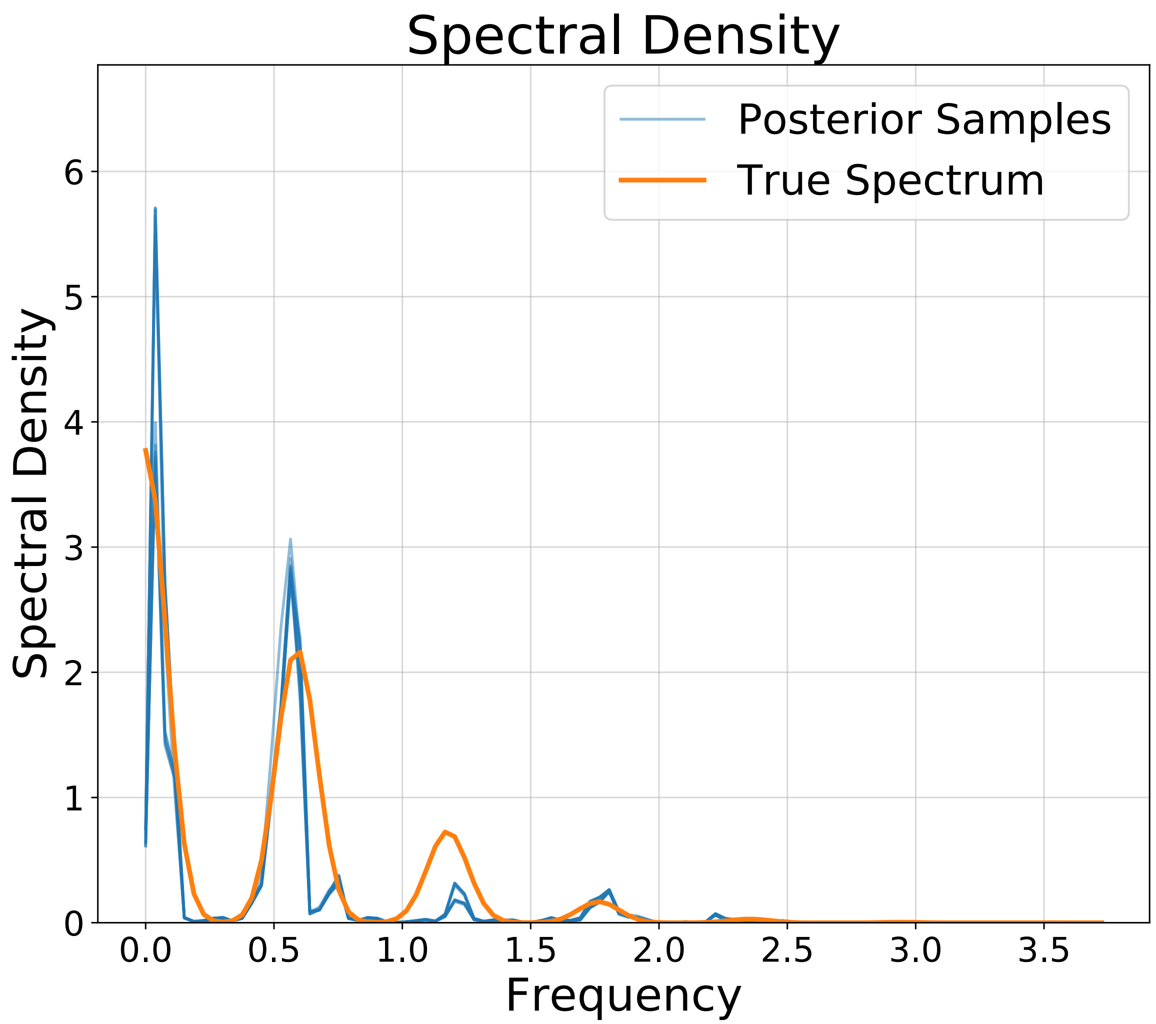
SINC DATA

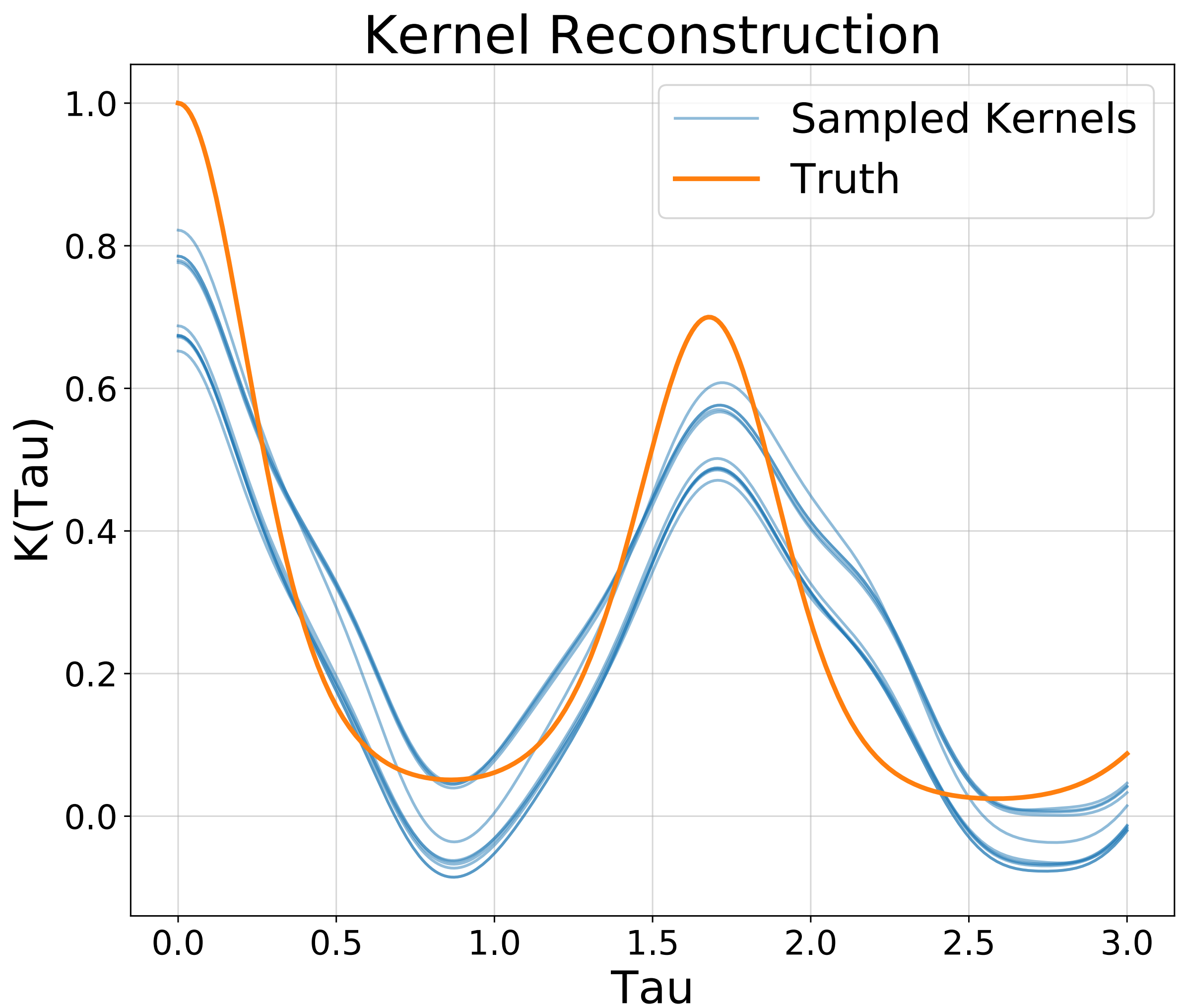
$\mathrm{sinc}(x) = \sin(\pi x)/(\pi x)$



QUASI-PERIODIC DATA

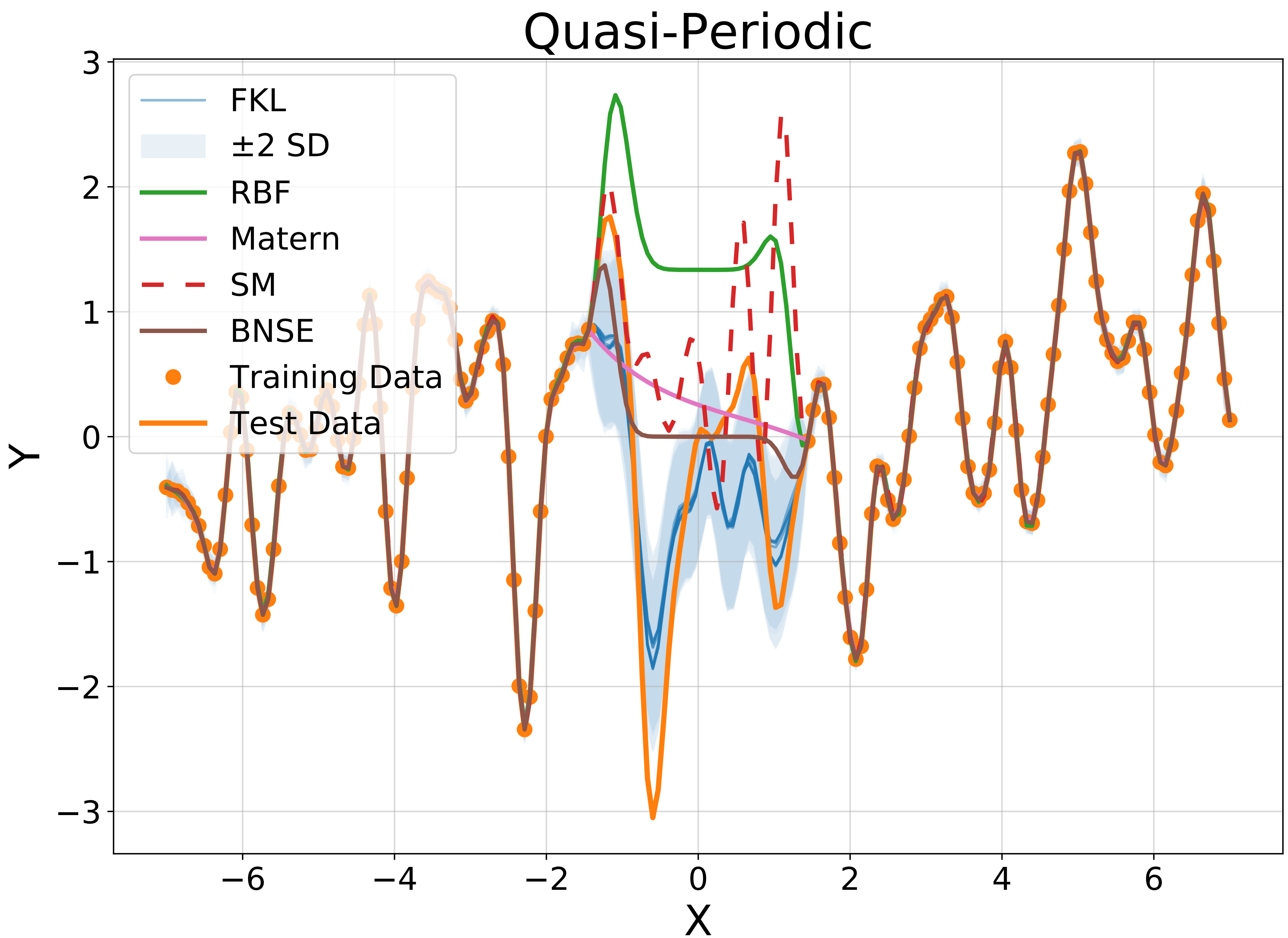
- Generative kernel is product of RBF and periodic kernels





QUASI-PERIODIC DATA

- Generative kernel is product of RBF and periodic kernels



ELLIPTICAL SLICE SAMPLING (MURRAY, ADAMS, MACKAY, 2010)

Sample zero mean Gaussians

Re-parameterize for non-zero mean

Input: current state $\mathbf{f}$, a routine that samples from $\mathcal{N}(0, \Sigma)$, log-likelihood function $\log L$.

Output: a new state $\mathbf{f}'$. When $\mathbf{f}$ is drawn from $p^\star(\mathbf{f}) \propto \mathcal{N}(\mathbf{f}; 0, \Sigma)\, L(\mathbf{f})$, the marginal distribution of $\mathbf{f}'$ is also $p^\star$.

1. Choose ellipse: $\boldsymbol{\nu} \sim \mathcal{N}(0, \Sigma)$
2. Log-likelihood threshold:

$$u \sim \mathrm{Uniform}[0, 1]$$
$$\log y \leftarrow \log L(\mathbf{f}) + \log u$$

3. Draw an initial proposal, also defining a bracket:

$$\theta \sim \mathrm{Uniform}[0, 2\pi]$$
$$[\theta_{\min}, \theta_{\max}] \leftarrow [\theta - 2\pi, \theta]$$

4. $\mathbf{f}' \leftarrow \mathbf{f} \cos\theta + \boldsymbol{\nu} \sin\theta$
5. **if** $\log L(\mathbf{f}') > \log y$ **then:**
6. Accept: **return** $\mathbf{f}'$
7. **else:**
   Shrink the bracket and try a new point:
8. **if** $\theta < 0$ **then:** $\theta_{\min} \leftarrow \theta$ **else:** $\theta_{\max} \leftarrow \theta$
9. $\theta \sim \mathrm{Uniform}[\theta_{\min}, \theta_{\max}]$
10. **GoTo** 4.

Figure 2: The elliptical slice sampling algorithm.

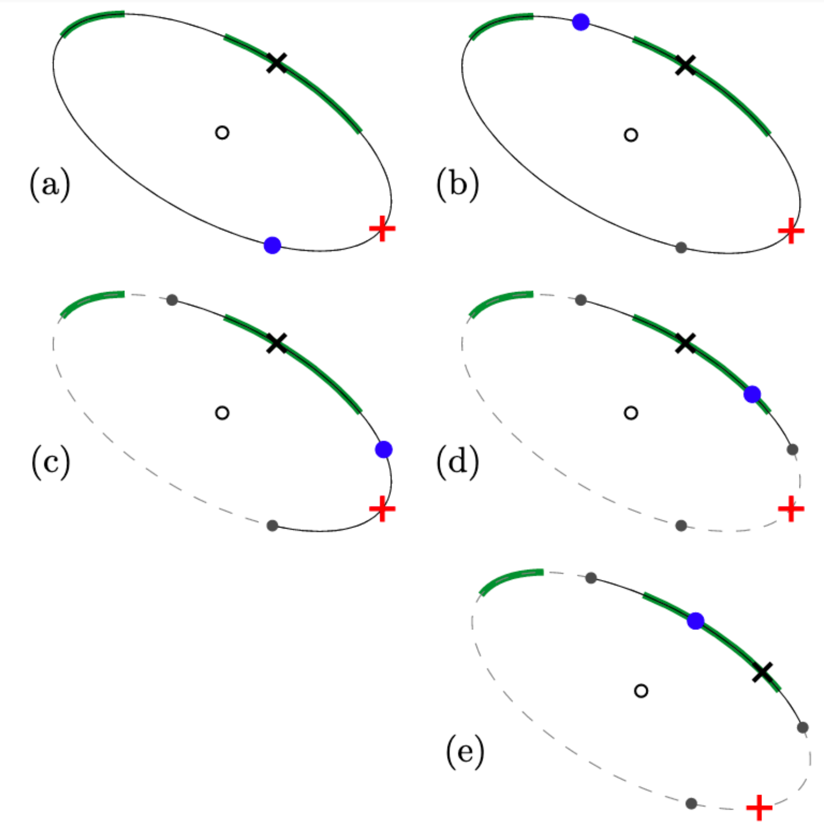


Figure 3: **(a)** The algorithm receives $\mathbf{f}$=✕ as input. Step 1 draws auxiliary variate $\boldsymbol{\nu}$=+, defining an ellipse centred at the origin (∘). Step 2: a likelihood threshold defines the 'slice' (▬). Step 3: an initial proposal ● is drawn, in this case not on the slice. **(b)** The first proposal defined both edges of the $[\theta_{\min}, \theta_{\max}]$ bracket; the second proposal (●) is also drawn from the whole range. **(c)** One edge of the bracket (—) is moved to the last rejected point such that ✕ is still included. Proposals are made with this shrinking rule until one lands on the slice. **(d)** The proposal here (●) is on the slice and is returned as $\mathbf{f}'$. **(e)** Shows the reverse configuration discussed in Section 2.3: ✕ is the input $\mathbf{f}'$, which with auxiliary $\boldsymbol{\nu}'$=+ defines the same ellipse. The brackets and first three proposals (•) are the same. The final proposal (●) is accepted, a move back to $\mathbf{f}$.