NeurIPS 2020

Task-Agnostic Amortized Inference of Gaussian Process Hyperparameters

Sulin Liu
Princeton University

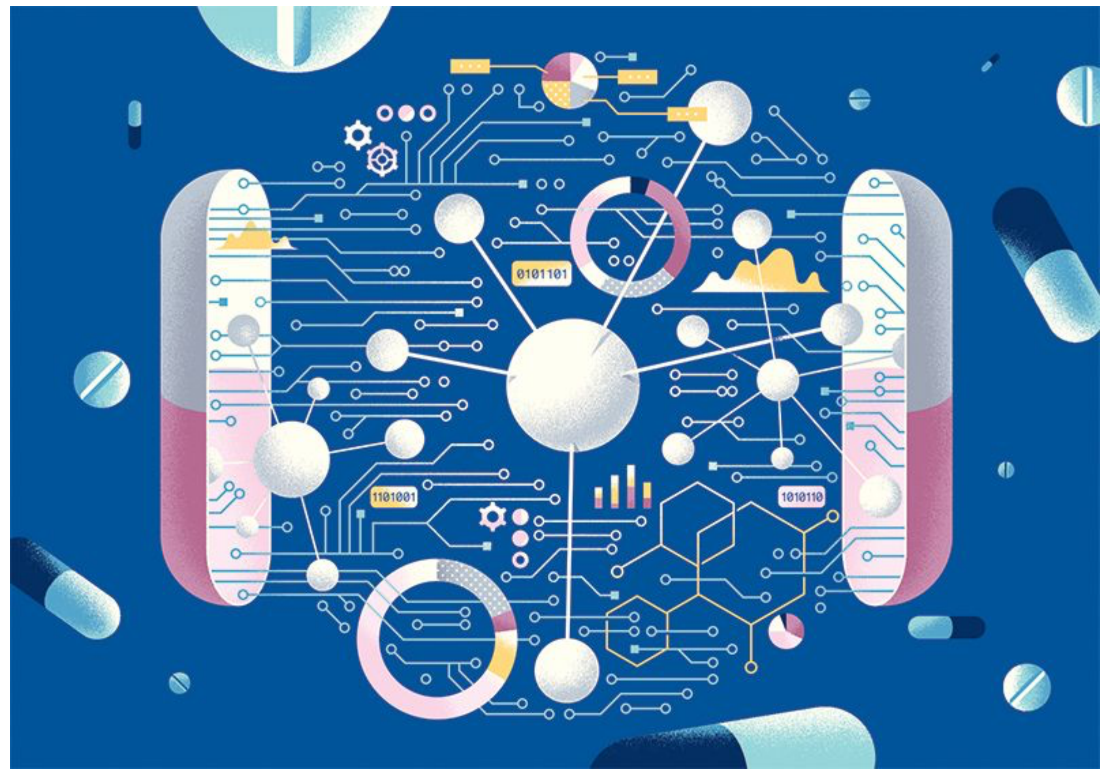
Joint work with Xingyuan Sun, Peter J. Ramadge, Ryan P. Adams

TL;DR: A single neural net is able to perform comparably well on unseen GP tasks, while being ~100 times faster than conventional procedures

PRINCETON
LABORATORY FOR INTELLIGENT PROBABILISTIC SYSTEMS

Gaussian processes

- GPs are **flexible** priors for modeling functions with **tractable** inference



BayesOpt for drug discovery

BayesOpt for hyperparameter tuning of DNN

Planning and control

Model selection problem in GPs

- GP prediction performance depends critically on its **kernel function** accurately **reflecting the structure of the problem**
- But finding a good kernel function is:
  - **Tricky**: figuring out what kernel function to use, Squared Exponential or Matérn? Periodicity?
  - **Costly**: searching for good kernel hyperparameters, usually through costly marginal likelihood maximization (**$O(n^3)$** per iteration)

Motivation

Motivation

- Goal: ~~tricky~~ **automatic** and ~~costly~~ **lightweight** model selection procedure for GP

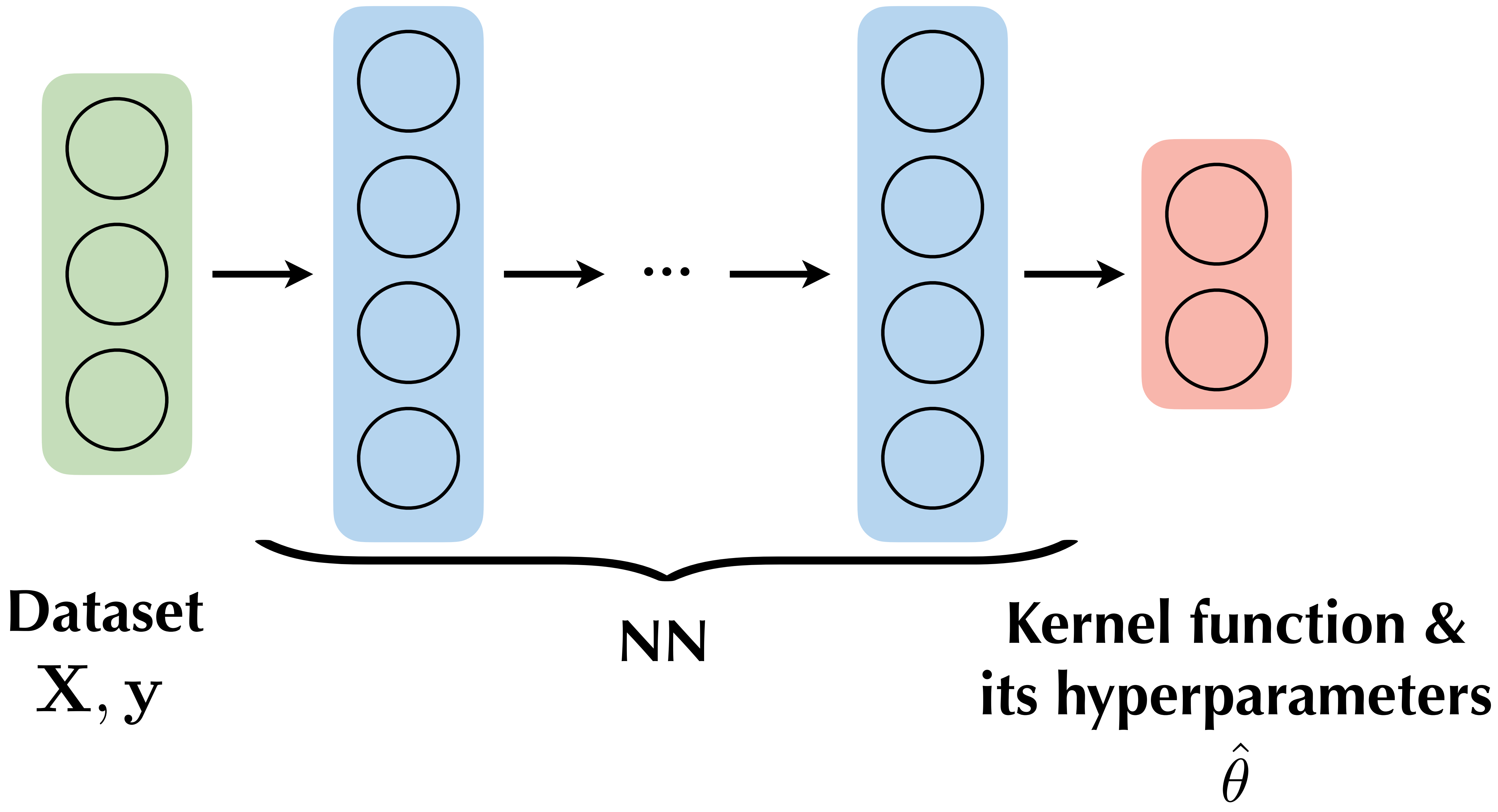
Motivation

- Goal: ~~tricky~~ **automatic** and ~~costly~~ **lightweight** model selection procedure for GP
- Idea: can we **amortize** this expensive model selection procedure with direct estimate from a neural network?
  - Inspiration from amortized variational inference literature (e.g. VAE)

Motivation

- Goal: ~~tricky~~ **automatic** and ~~costly~~ **lightweight** model selection procedure for GP

- Idea: can we **amortize** this expensive model selection procedure with direct estimate from a neural network?
    - Inspiration from amortized variational inference literature (e.g. VAE)

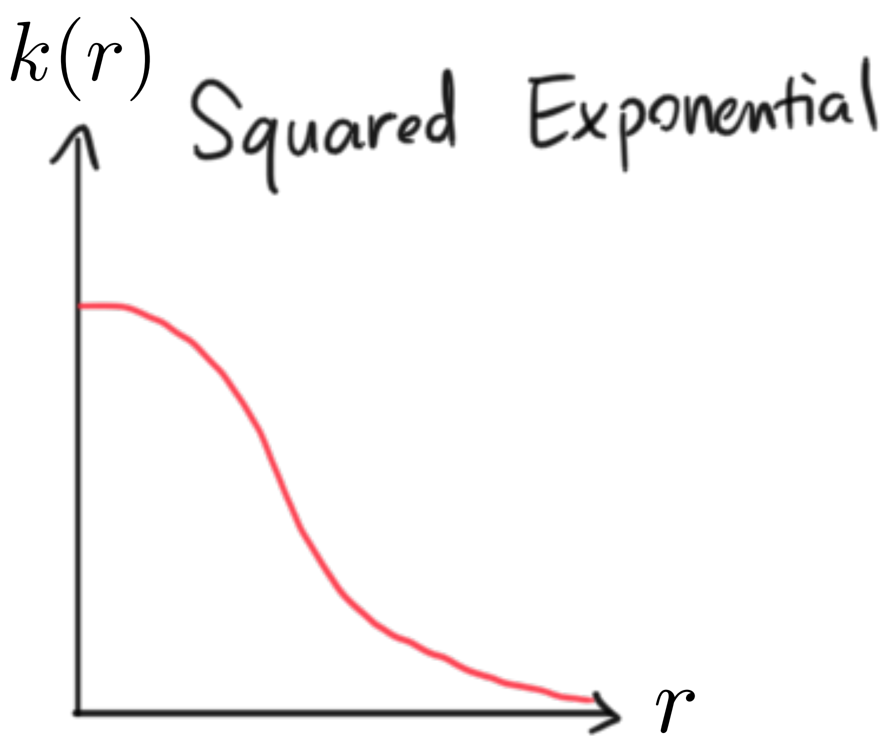


Amortized Type II Maximum Likelihood:

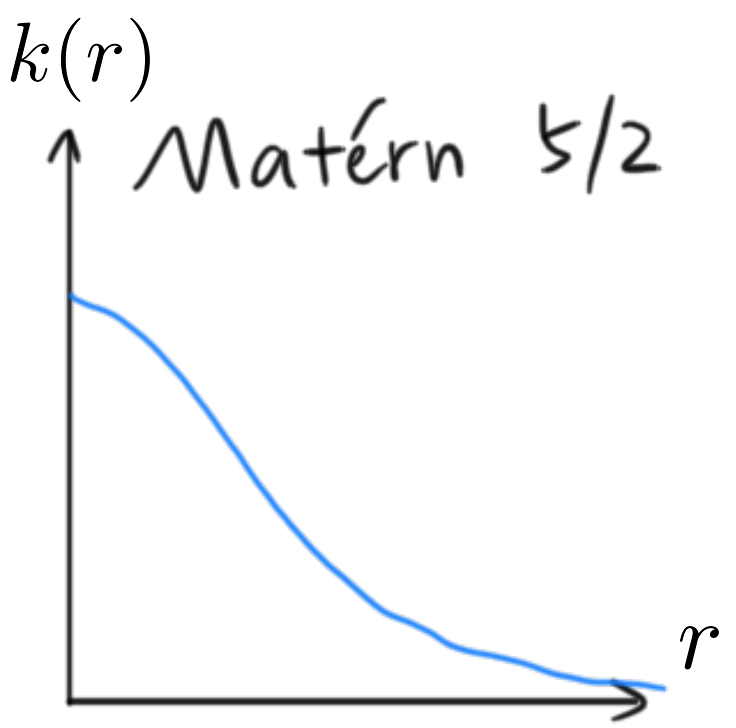
$$\hat{\theta} \approx \arg\max_{\theta} \; \log p(\mathbf{y}|\mathbf{X}, \theta)$$

How to parametrize the space of kernel functions?

How to parametrize the space of kernel functions?



kern_type = 0, lengthscale = 2



kern_type = 1, lengthscale = 3

Compositional kernel modeling

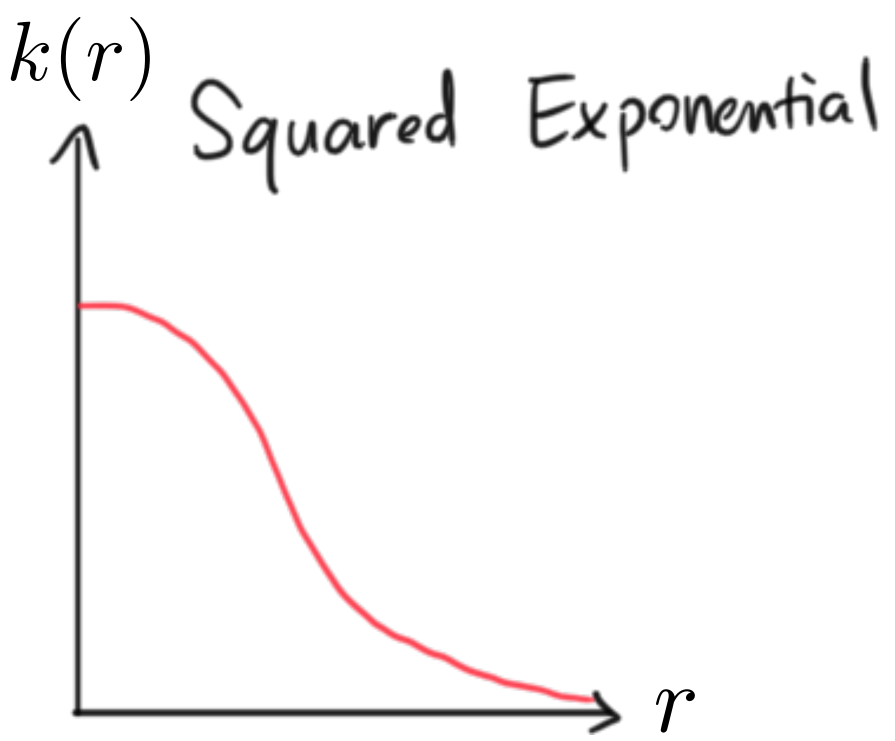
Use composition rules to build more complicated ones

- $k_0 + k_1 + \ldots$
- $k_0 * k_1 * \ldots$

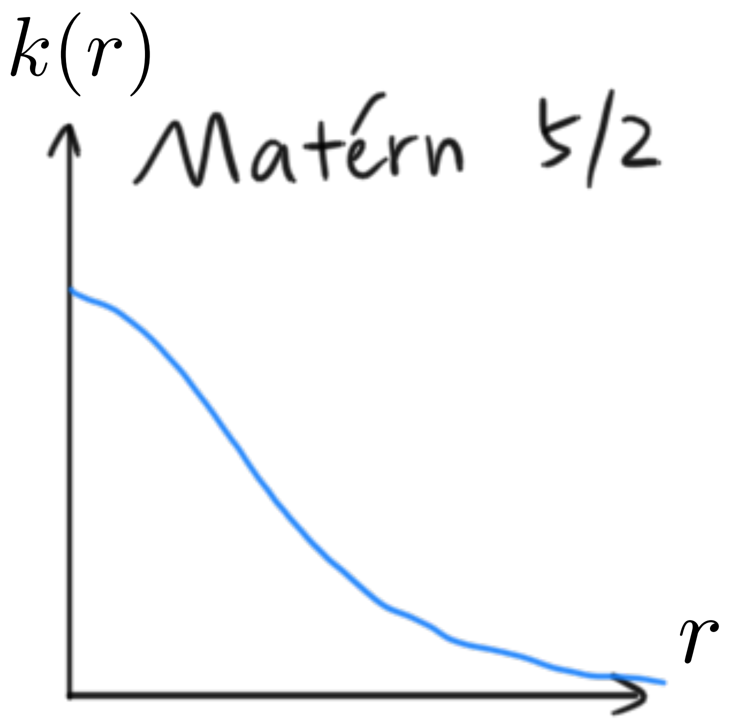
A **very hard** learning problem!

1. a mix of discrete and continuous variables
2. intercorrelated variables: such as kern_type and lengthscale in this case

How to parametrize the space of kernel functions?



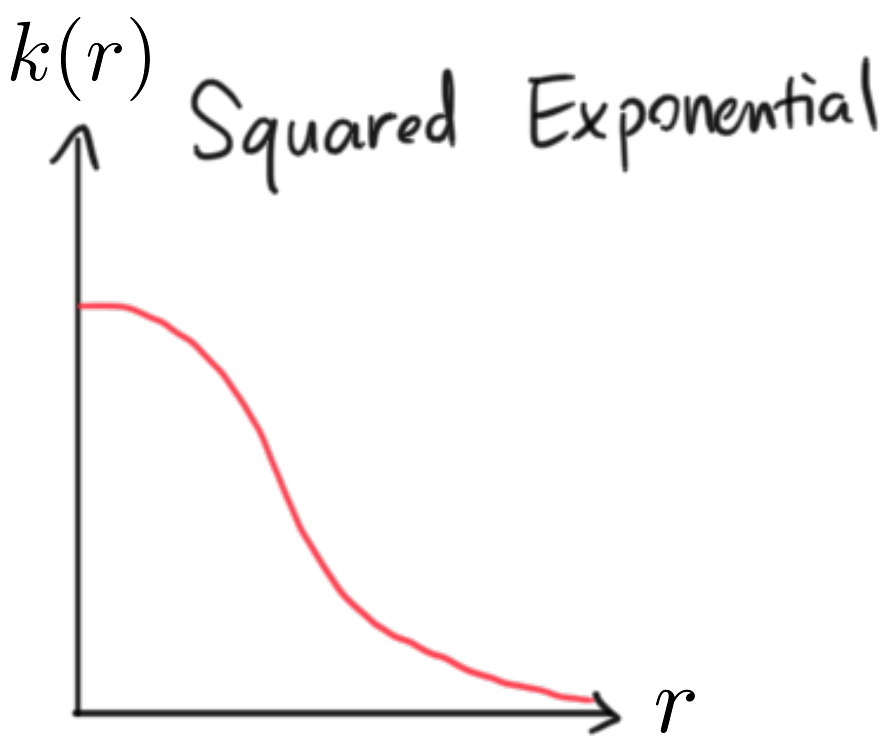
kern_type = 0, lengthscale = 2



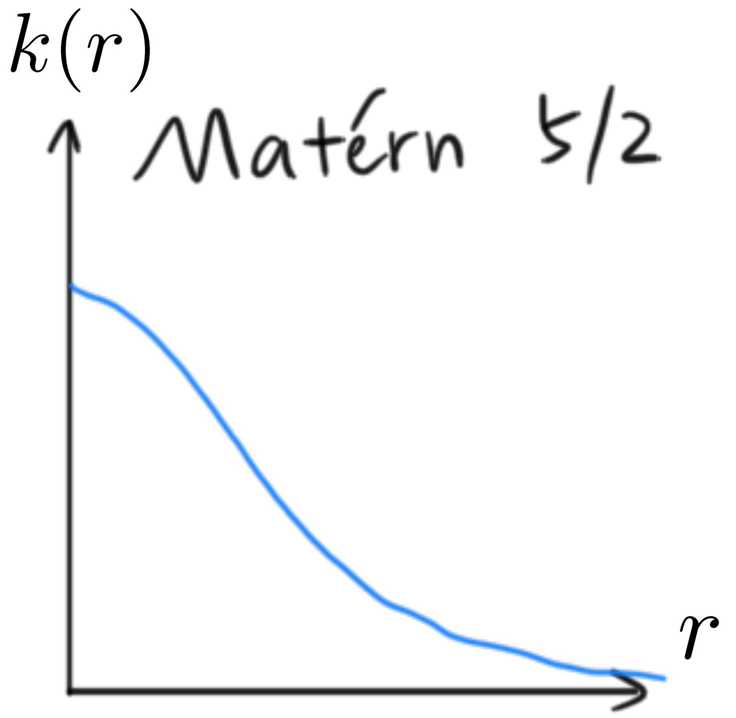
kern_type = 1, lengthscale = 3

Compositional kernel modeling

How to parametrize the space of kernel functions?



kern_type = 0, lengthscale = 2

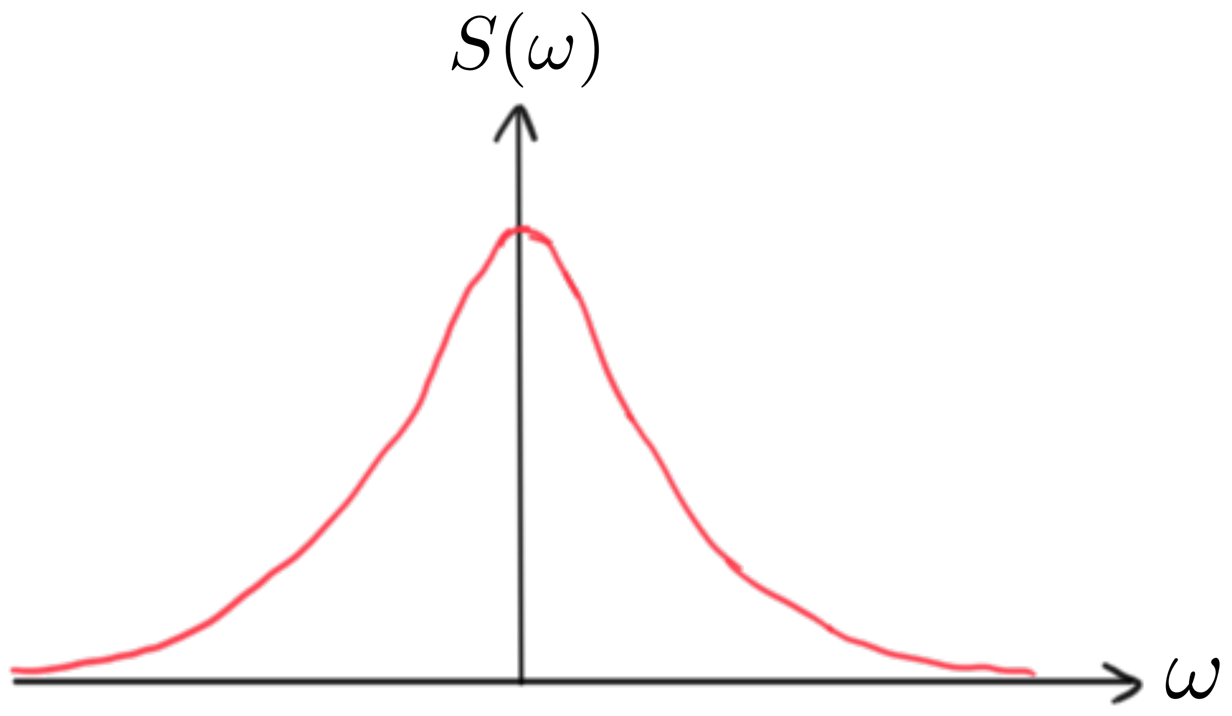


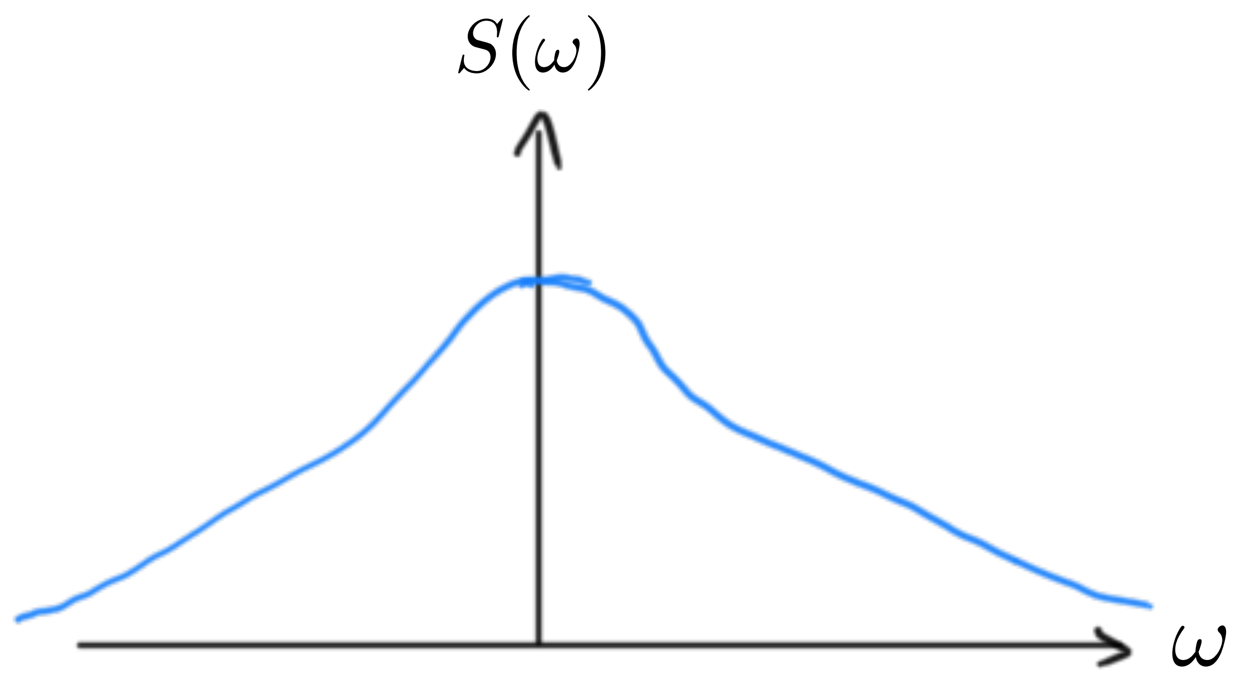
kern_type = 1, lengthscale = 3

Bochner's Theorem

$$k(r) \Leftrightarrow S(\omega)$$

$$S(\omega) = \int k(r) e^{-2\pi i \omega r} dr$$

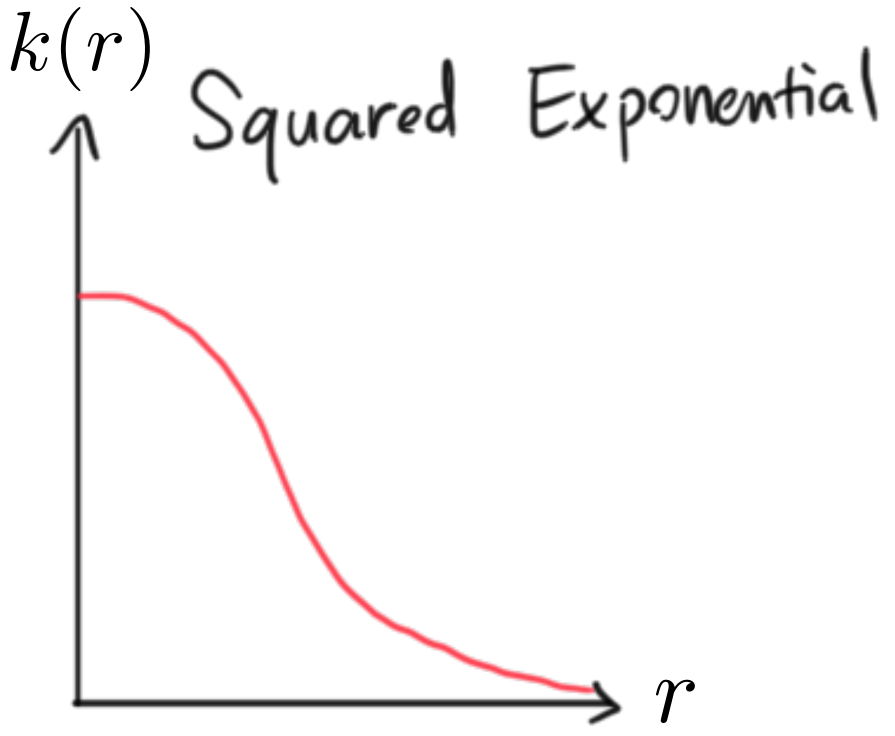




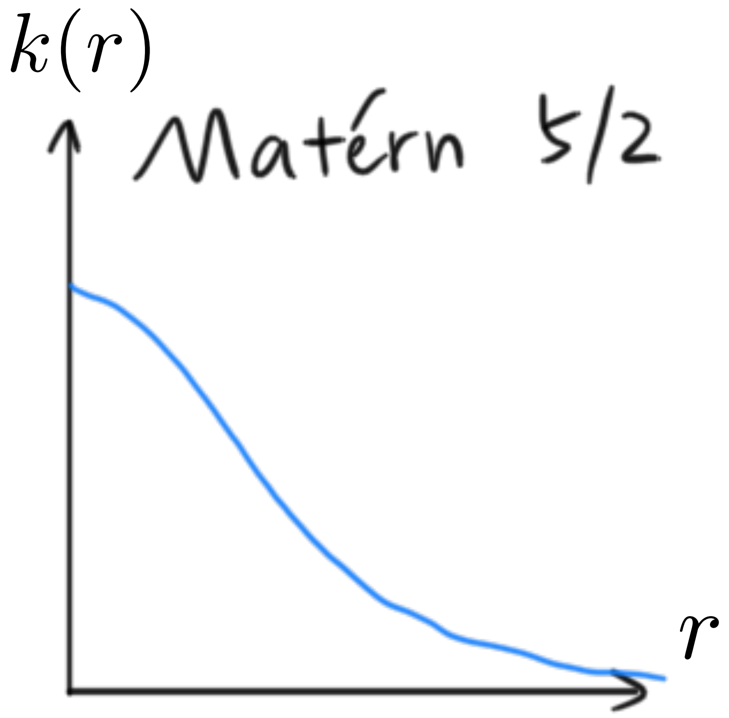
Compositional kernel modeling

Spectral density modeling

How to parametrize the space of kernel functions?



kern_type = 0, lengthscale = 2



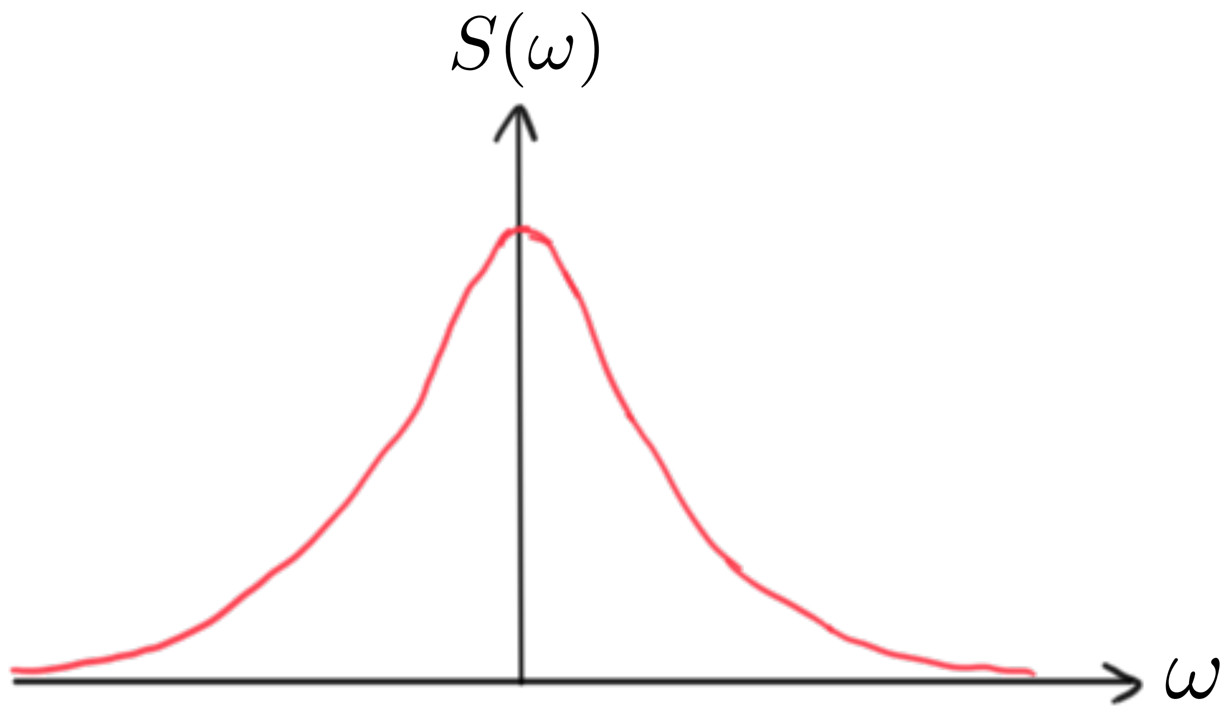
kern_type = 1, lengthscale = 3

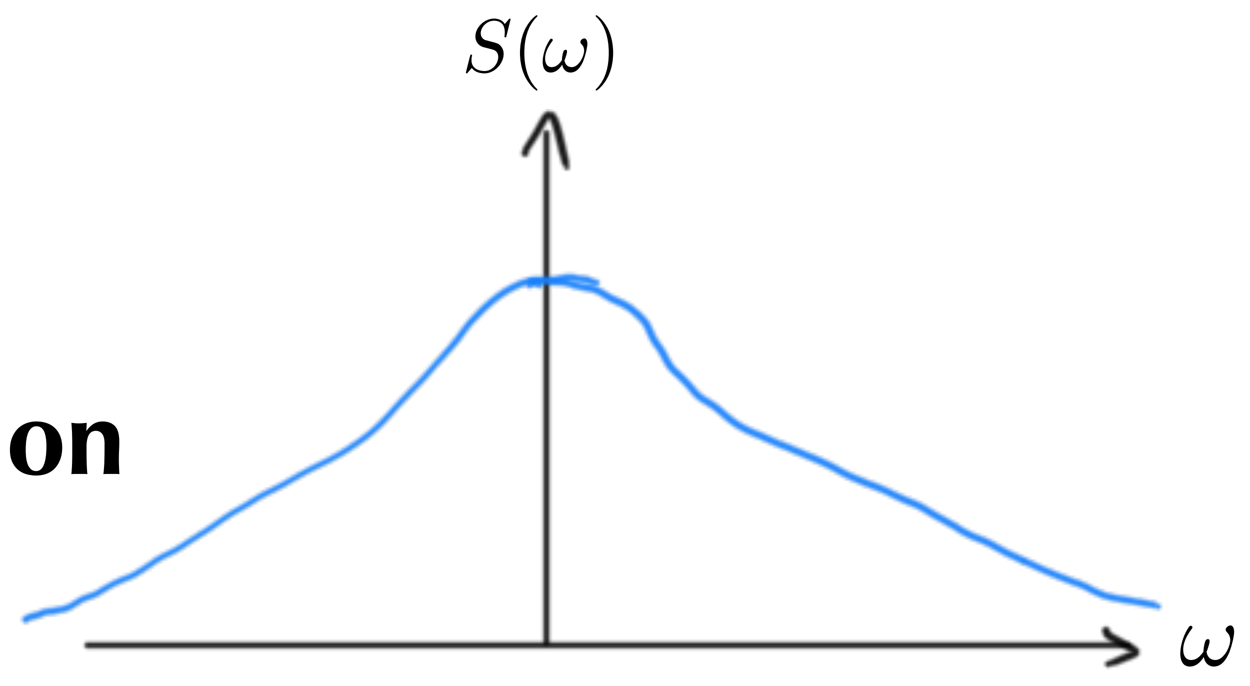
Bochner's Theorem

$$k(r) \Leftrightarrow S(\omega)$$

$$S(\omega) = \int k(r) e^{-2\pi i \omega r} dr$$

For any stationary kernel function





Compositional kernel modeling

Spectral density modeling

Amortized hyperparameter inference for Gaussian processes (AHGP)

Amortized hyperparameter inference for Gaussian processes (AHGP)

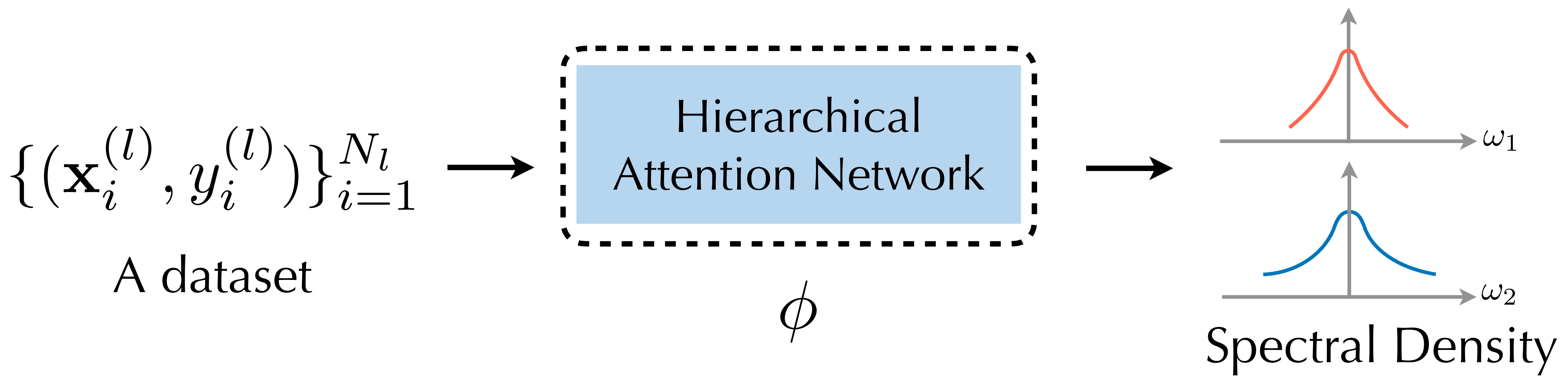
$\{(\mathbf{x}_i^{(l)}, y_i^{(l)})\}_{i=1}^{N_l}$

A dataset

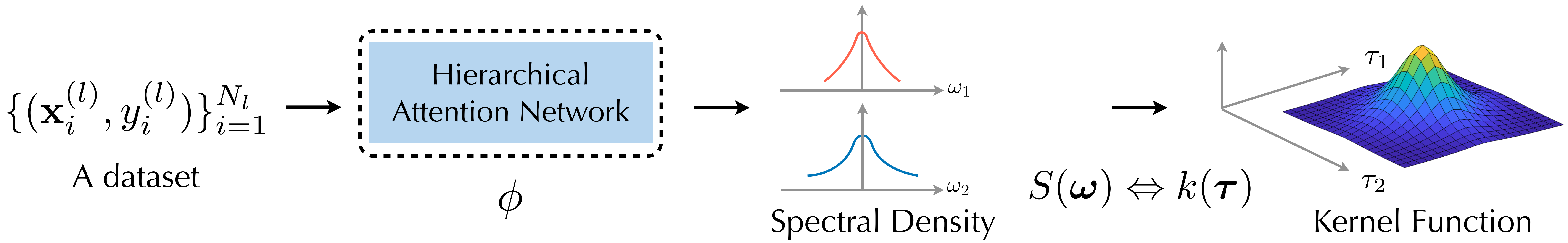
Amortized hyperparameter inference for Gaussian processes (AHGP)

$$\underbrace{\{(\mathbf{x}_i^{(l)}, y_i^{(l)})\}_{i=1}^{N_l}}_{\text{A dataset}} \longrightarrow \underbrace{\boxed{\text{Hierarchical Attention Network}}}_{\phi}$$

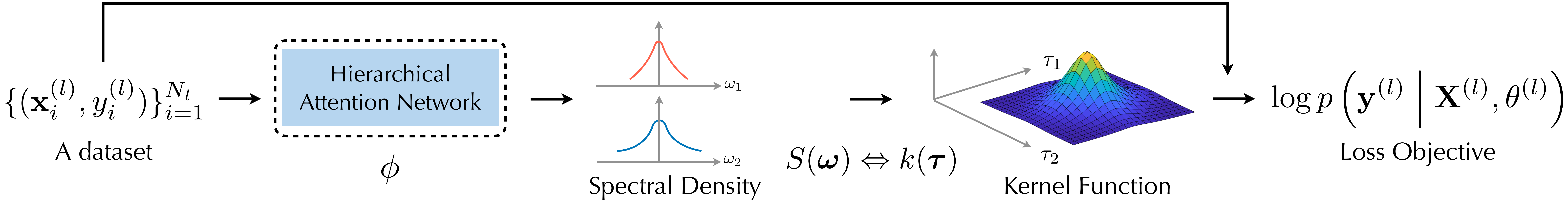
Amortized hyperparameter inference for Gaussian processes (AHGP)



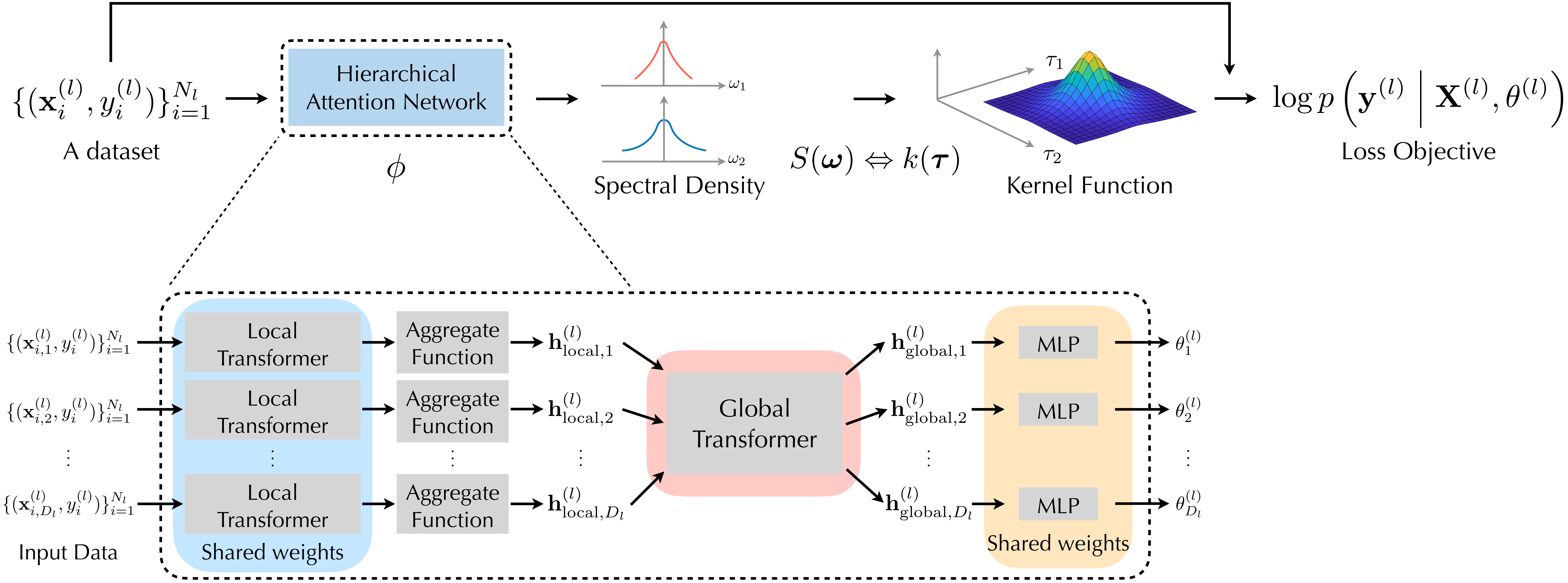
Amortized hyperparameter inference for Gaussian processes (AHGP)



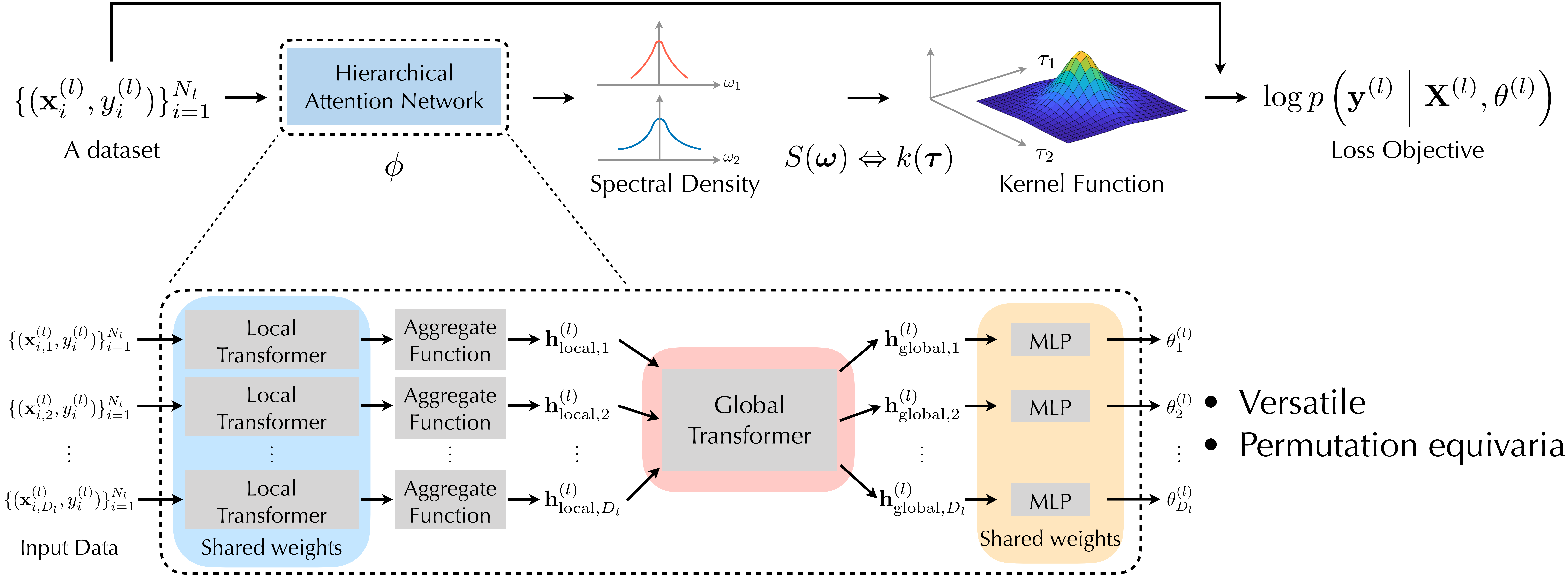
Amortized hyperparameter inference for Gaussian processes (AHGP)



Amortized hyperparameter inference for Gaussian processes (AHGP)



Amortized hyperparameter inference for Gaussian processes (AHGP)



- Versatile
- Permutation equivariant

Experimental results

- Training data: 5k **synthetic** datasets generated from GP priors with stationary kernels
- Training objective: $\mathcal{L}\left(\phi, \left\{\mathcal{D}^{(l)}\right\}_{l=1}^{L}\right) = -\frac{1}{L}\sum_{l=1}^{L}\frac{1}{N_l}\log p\left(\mathbf{y}^{(l)} \,\middle|\, \mathbf{X}^{(l)}, \theta^{(l)}\right)$
- Training method: Adam with fixed learning rate, dropout on self-attention encoders
- Results:
  - **One trained model for all**: a **single** trained neural net is able to produce hyperparameters of comparable quality to MLL-opt methods across different **unseen** real-world benchmarks (regression, BayesOpt, BayesQuadrature)
  - **Super lightweight**: ~100 times faster than MLL-opt
  - Even performs slightly **better** on relatively **smaller** dataset (due to implicit regularization in training)

Interested in AHGP?
Come to our poster session :)

Code: https://github.com/PrincetonLIPS/AHGP

Sulin Liu (sulinl@princeton.edu) @liu_sulin