# Open Problems in GP Approximation and Benchmarking

Mark van der Wilk

Bayesian Decision-making and Uncertainty @ NeurIPS

University of Oxford Department of Computer Science

🔗 https://mvdw.uk

@markvanderwilk

14 December 2024

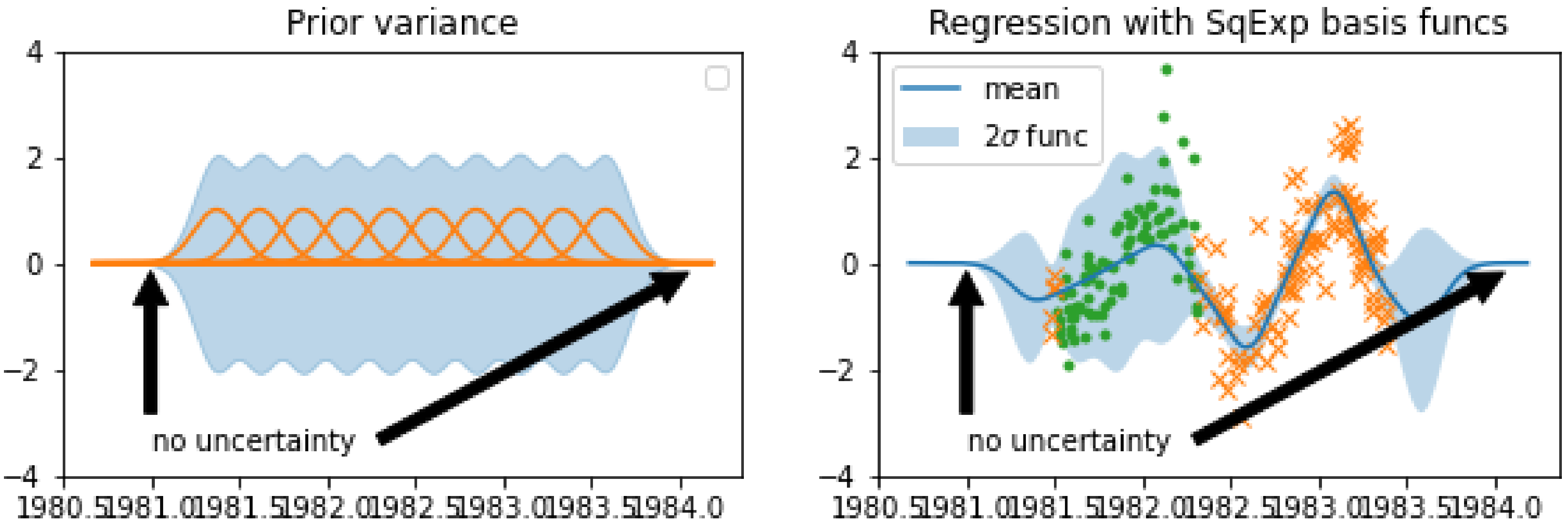
# The Promises of Gaussian Processes

# The Promises of Gaussian Processes

1. Good uncertainty estimates, from infinite basis functions

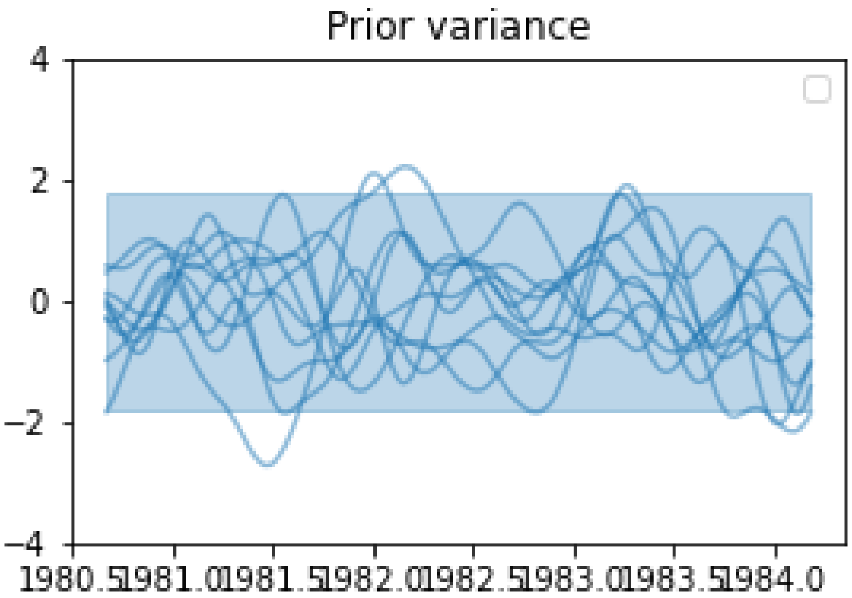
# The Promises of Gaussian Processes

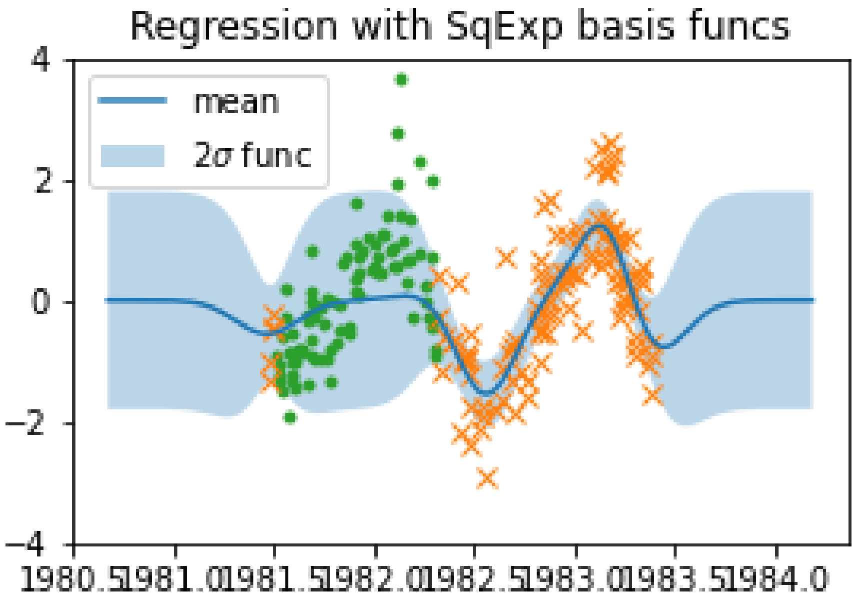
1. Good uncertainty estimates, from infinite basis functions



# The Promises of Gaussian Processes

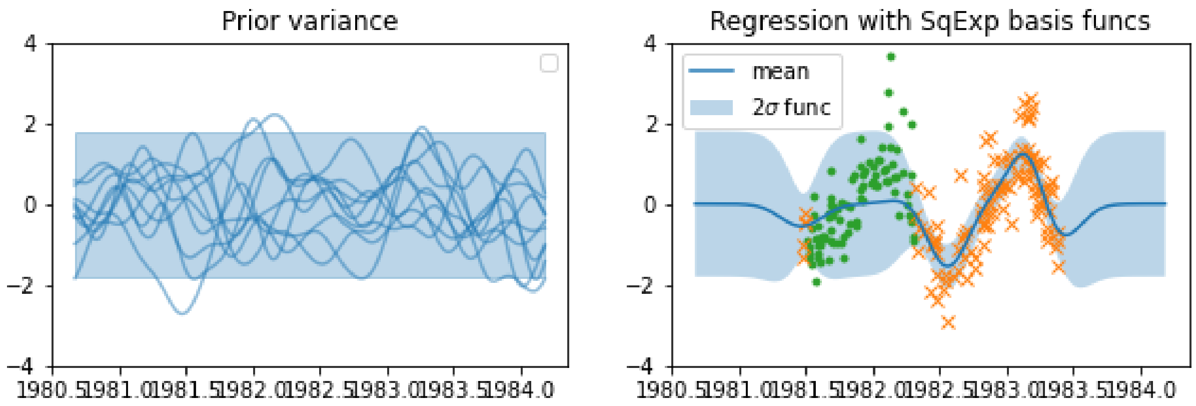
1. Good uncertainty estimates, from infinite basis functions



# The Promises of Gaussian Processes

1. Good uncertainty estimates, from infinite basis functions

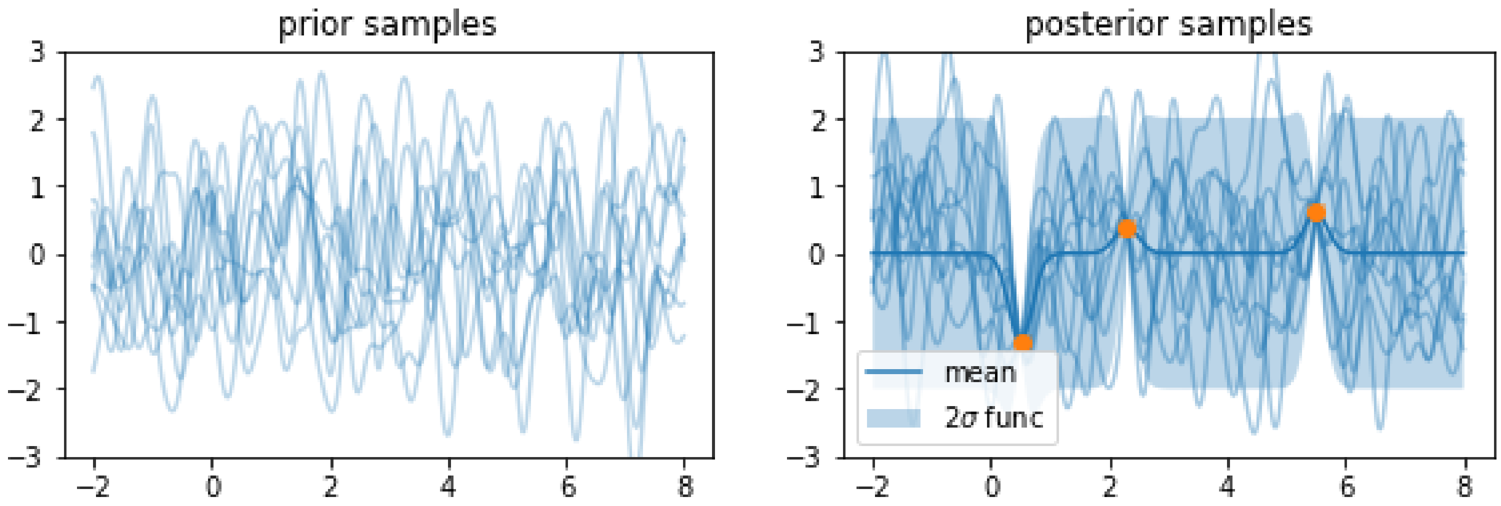



$$p(y^*|x^*, (\boldsymbol{y}, X), (\theta, \sigma)) = \mathcal{N}\Big(y^*; \quad k^\theta_{*X}\left(K^\theta_{XX} + \sigma^2\right)^{-1}\boldsymbol{y}$$

$$k^\theta_{**} - k^\theta_{*X}\left(K^\theta_{XX} + \sigma^2\right)^{-1}k^\theta_{X*}\Big)$$

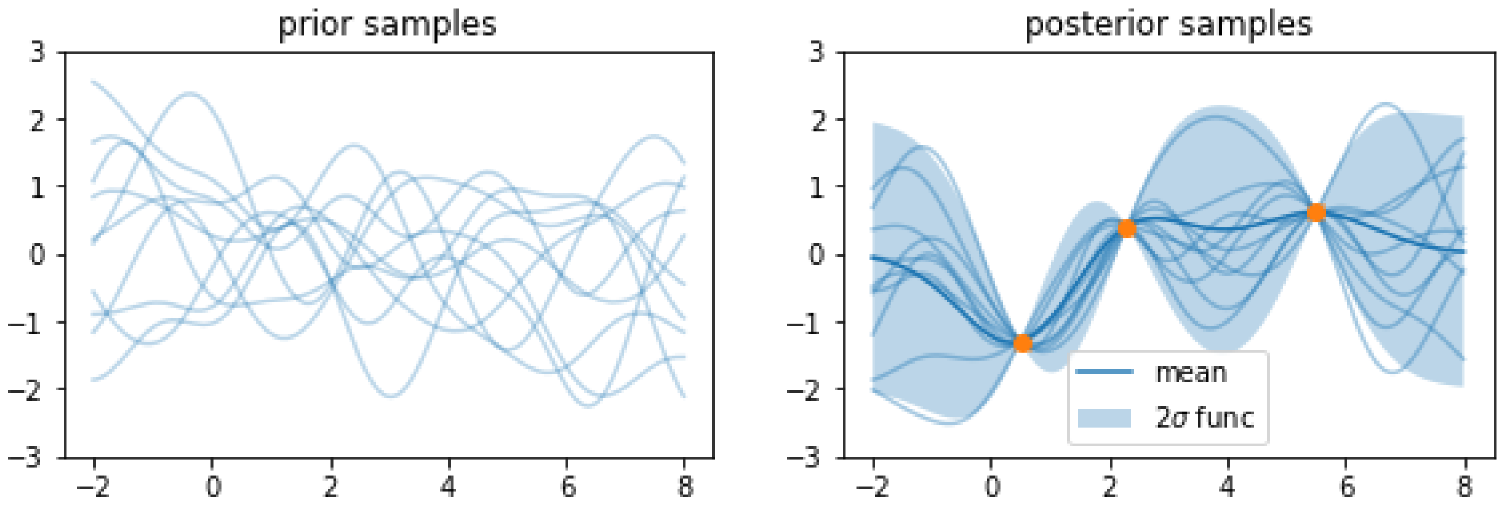
# The Promises of Gaussian Processes

Which kernel should we use?



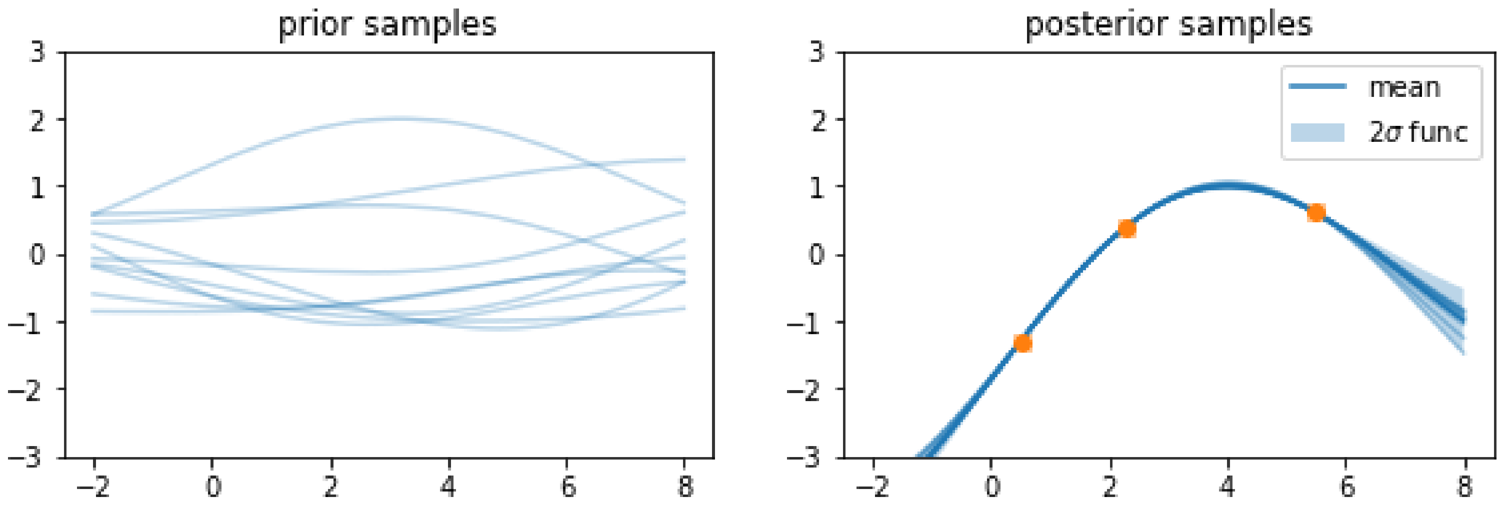
# The Promises of Gaussian Processes

Which kernel should we use?



# The Promises of Gaussian Processes

Which kernel should we use?



# The Promises of Gaussian Processes

2. Automatic hyperparameter tuning

$$\underset{\theta,\sigma}{\operatorname{argmax}} \log p(\boldsymbol{y}|X, (\theta, \sigma)) = \underset{\theta,\sigma}{\operatorname{argmax}} \log \mathcal{N}\left(\boldsymbol{y}; 0, K_{XX}^{\theta} + \sigma^2 I\right)$$

# The Promises of Gaussian Processes

Can even discover sophisticated structure in data!

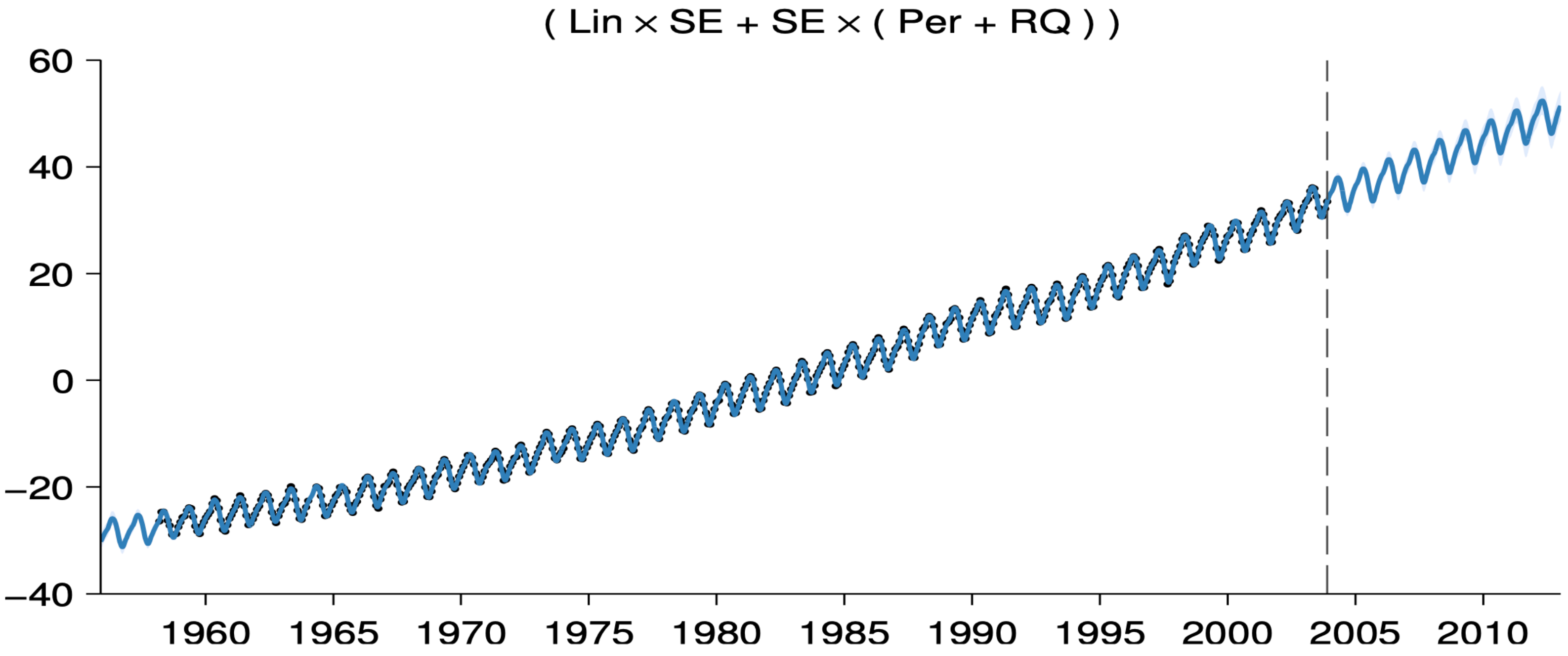



Figure 1: “Automatic Statistician” (Duvenaud et al., 2013)

# The Promises of Gaussian Processes

Can even discover sophisticated structure in data!

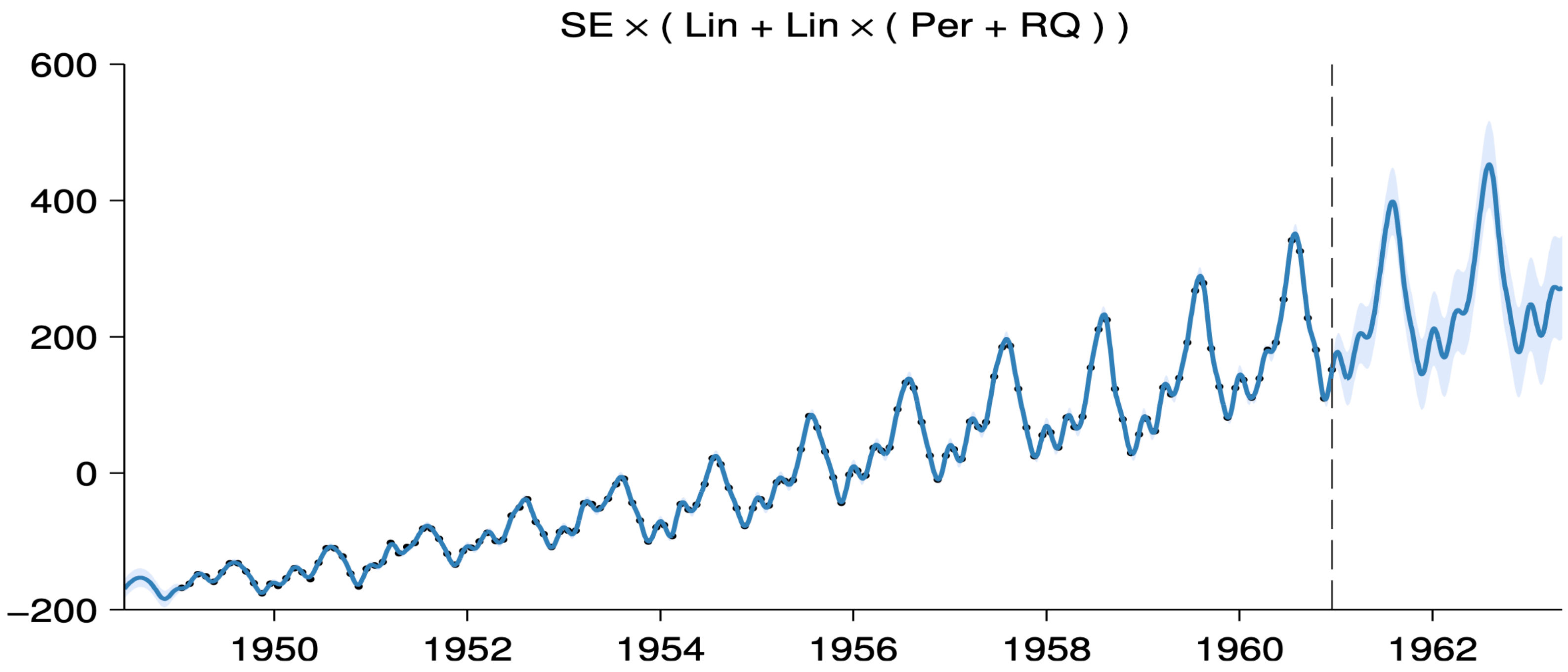



Figure 2: “Automatic Statistician” (Duvenaud et al., 2013)

# The Promises of Gaussian Processes

1. Good uncertainty estimates
   from infinite basis functions.
2. Automatic hyperparameter / kernel selection
   from Bayesian model selection.

**GPs should be robust, no-nonsense tools!**

Practitioners benefit from:

- trustworthy predictions, due to uncertainty,
- ease-of-use, due to automatic tuning to dataset.

GPs are a **silent workhorse** in data science & stats!

**What should I do if my dataset has 100,000 datapoints?**

# Decades of Progress in GP Approximations

- **Eigenfunction approximation**: Zhu et al. (1997), Ferrari-Trecate et al. (1998)
- **Finite basis functions**: Silverman (1985), Smola & Schölkopf (2000)
- **Inducing points**: Csató & Opper (2002), Seeger et al. (2003), Snelson & Ghahramani (2005), Titsias (2009)
- **Conjugate gradients**: Gibbs & MacKay (1996), Davies (2015), Wang et al. (2019)
- **Grid structures**: Saatçi (2011), Nickson et al. (2015), Wilson & Nickisch (2015)

… and many many more.

## What should I do if my dataset has 100,000 datapoints?

Practitioner expects:

- `gp_predict(X, Y)` ⇒ `gp_predict_approx(X, Y)`
- accurate predictions, similar behaviour to full GP
- … maybe `gp_predict_approx(X, Y, prediction_sacrifice="1%")`

Practitioner gets:

- "Well, which approximation do you want to use?" Too much choice!
- We need *fewer* answers to this question, not more!
- `gp_predict_approx_variational(X, Y, num_inducing=100, inducing_locations=???, jitter=1e-6, min_lengthscale=1e-3)`

**We don’t currently provide a black-box answer on how to approximate**

🎯 **Goal: Near-exact approximation, without thinking too hard**

So how do current approximations match up to this?

# Example: Variational Inducing Points (Titsias, 2009)

Approximation parameters:

- Number of inducing points
- How to pick the inducing points
- Jitter value, for numerical stability

**⚠ Relies on manual tuning**

- Can't know correct $M$ ahead of time for a new dataset
- Different advice on IP locations
- Jitter to make algorithm run

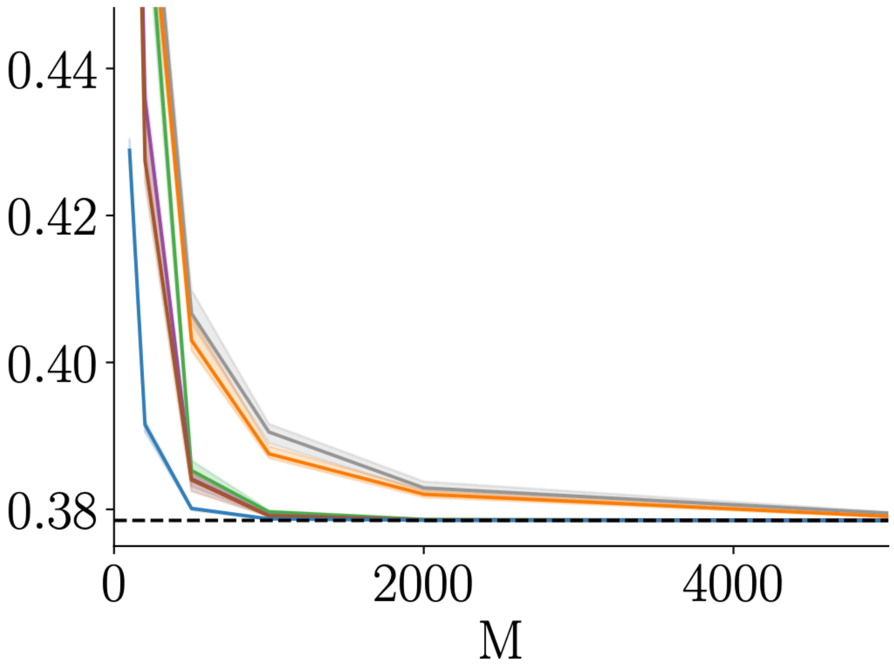



Figure 3: RMSE for `elevators` dataset

# Example: Conjugate Gradients (Wang et al., 2019)

Approximation parameters:

- Number of probe vectors
- Lower noise limits
- CG termination criterion
- …

**⚠ Relies on Manual Tuning**

Getting parameters wrong leads to underperformance, or even bad divergence.

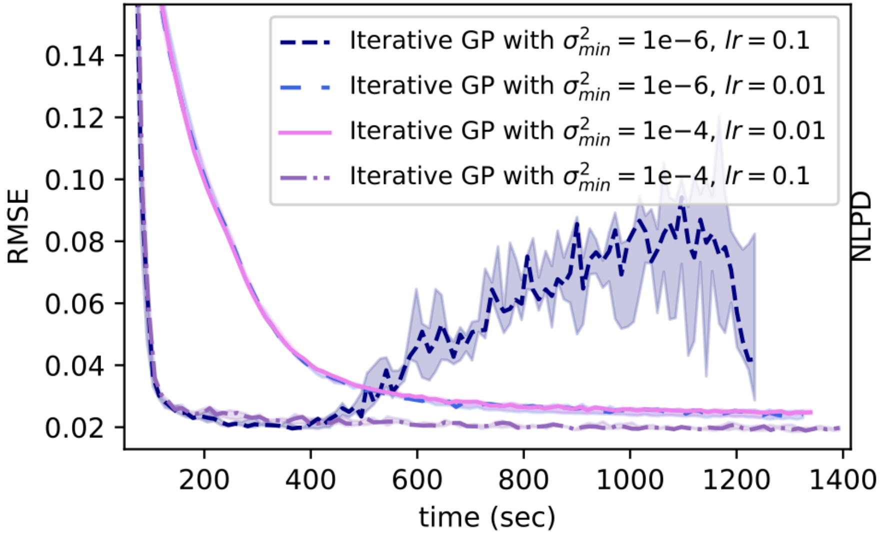



Figure 4: RMSE for `bike` dataset. Noise-free dataset, so full GP gets 0.000 RMSE.

⚠ **Methods work in the paper, but current default approximation parameters don't work well for all datasets**

"I tried <approximation method>,

but the results were bad

… so GPs must be bad. "

# Decades of Progress in GP Approximations

… have brought us

- many methods, but little clarity on which one to use, and when
- approximations that need *tuning*, negating promise #2!

⚠ **Approximate GPs should be robust, no-nonsense tools!**

GPs are a **silent workhorse** in data science & stats!

… but approximate scalable GPs are not!

# Decades of Progress in GP Approximations

… have also brought us
scalable methods, that are extremely accurate, *when tuned correctly*.

Case study:
Variational Inducing Points (Titsias, 2009)
vs
Conjugate Gradients (Davies, 2015; Gibbs & MacKay, 1996; Wang et al., 2019)

**So which method is better?**

Keeping in mind: both are arbitrarily accurate, if tuned correctly.

# Common Benchmarking

Some self-criticism (Artemev et al., 2021), but common practice.

- Pick a few datasets
- Run various approximations, possibly with various tuning parameters
- Measure predictive performance, present in a table, **bold** == publish.

| | | LML | | NLPD | | RMSE | |
|---|---|---|---|---|---|---|---|
| | | Approx | Cholesky | Approx | Cholesky | Approx | Cholesky |
| bike<br>n=17379, d=17 | Iterative GP | 30992.8 | 31319.1 | -2.016 | -3.257 | 0.020 | 0.014 |
| | SGPR-4096 | 30502.5 | 32814.2 | -3.280 | -3.336 | 0.010 | 0.010 |
| | CGLB-4096 | 37732.7 | **42023.0** | **-4.216** | -4.329 | 0.004 | 0.004 |
| | CGLB-2048 | 34102.8 | 38936.7 | -3.811 | -3.972 | **0.003** | 0.003 |
| | CGLB-1024 | 30493.9 | 35351.8 | -3.403 | -3.615 | 0.005 | 0.005 |
| elevators<br>n=16599, d=18 | Iterative GP | -4709.0 | -4705.1 | 0.407 | 0.384. | **0.353** | 0.353 |
| | SGPR-4096 | -4675.3 | **-4653.3** | **0.386** | 0.386 | 0.354 | 0.354 |
| | CGLB-4096 | -4669.8 | -4659.1 | **0.386** | 0.386 | 0.354 | 0.354 |

# Benchmarking Problems

⚠ **All these methods *can be* arbitrarily accurate**

**All** convergent approximations, if tuned correctly, should give **exactly the same** results.

$\Rightarrow$ **Any** performance difference, is **purely** down to tuning approximation parameters!

- So we are **not** measuring intrinsic quality of the approximation.
- Instead, we measure the quality of our tuning of the **approximation parameter**.
- Time-quality trade-off is only thing that matters, but not tested!

# How Approximations should Work

Currently, approximations work by:

- making a choice for the approximation parameters,
- and then *measuring* the resulting performance.

**We should develop methods which**

- take a desired tolerance on predictive performance,
- the method runs until this guarantee is satisfied,
- **we measure how *long* it takes to reach this**.

# How we should be Benchmarking

If approximations worked in this way, benchmarking would be easier too.

**Measure time until accuracy target is reached**

| | Method 1 | Method 2 | Method 3 |
|---|---|---|---|
| Dataset #1 | 11s | 102s | 3600s |
| Dataset #2 | 2345s | 3134s | 3714s |
| Dataset #3 | 142s | 10s | 343s |

Table 1: Time until with 10% of optimal predictive accuracy

⚠ **For this benchmarking to make sense,**
**methods need to converge to the right answer!**

🎯 **Goal: Near-exact approximation, without thinking too hard**

- Compute time is compared to human intervention.
- Methods should be set up such that more time makes them get continuously better, without human intervention.

❓ **How can we make methods convergent?**

# Making Variational Inducing Points Convergent

We know that as $M \to N$, we converge to the true posterior (Burt et al., 2019; 2020).

- To remove *all* tuning, we need to steadily increase the number of inducing points during training.
- Slow due to many repeated training runs, but does remove all tuning!
- Much closer to how method is used in practice!
- This cost **should** be measured in benchmarking!

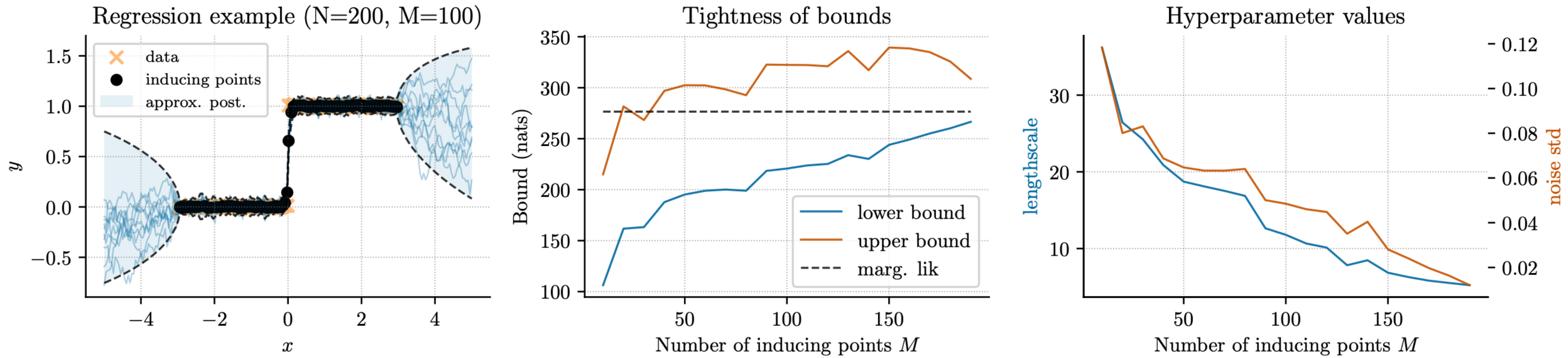
⚠ **This is a difficult and a pain!**

Takes lots of effort, but this is the problem we are faced with.

# Approximation and Model Specification are Dependent

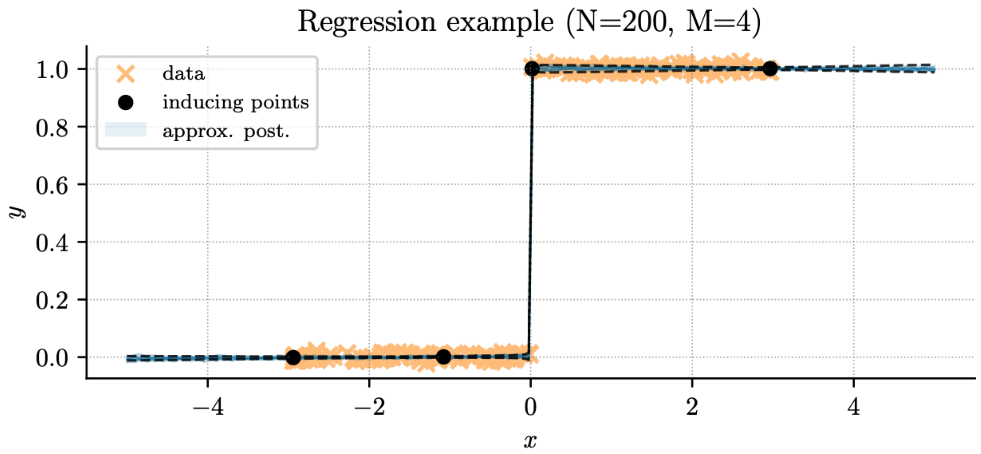
# Approximation and Model Specification are Dependent

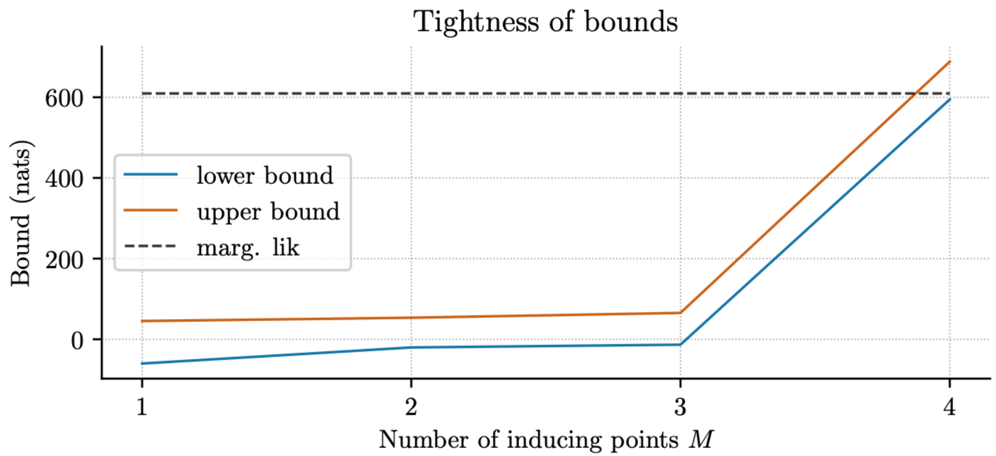
Approximations will behave strangely, if the true GP they are approximating behave strangely.



# Approximation and Model Specification are Dependent

But this is fixed if model misspecification is removed!




**⚠ Kernel search and approximation are related problems**

… and should probably be studied together.

# Conclusion

# Conclusion

Good approximations to GPs already exist… if you tune them correctly

**Next frontier: Make approximations *transparent* to the user!**

Procedures should converge to exact solution as they run longer.

**Benchmark the time it takes to reach a level of acccuracy**

**Kernel search and approximation are related problems**

… and should probably be studied together.

For more: *Recommendations for Baselines and Benchmarking Approx GPs* (Ober et al., 2024)

# Outlook

Lots of interesting problems are still open:

- **Mathematical**: Can we find *proofs* on how to scale approximation tuning parameters to *guarantee* convergence to an exact solution?
- **Statistical**: How do we solve the statistical and computational problems *together*?
- **Software**: How do we build tools that practitioners can easily use to solve their prediction problems? (Huge current bottleneck!)

# Bibliography

Artemev, A., Burt, D. R., & Wilk, M. van der. (2021). Tighter bounds on the log marginal likelihood of Gaussian process regression using conjugate gradients. *International Conference on Machine Learning*, 362–372.

Burt, D. R., Rasmussen, C. E., & Wilk, M. van der. (2020). Convergence of Sparse Variational Inference in Gaussian Processes Regression. *Journal of Machine Learning Research*, *21*(131), 1–63. **http://jmlr.org/papers/v21/19-1015.html**

Burt, D., Rasmussen, C. E., & Van Der Wilk, M. (2019). Rates of Convergence for Sparse Variational Gaussian Process Regression. In K. Chaudhuri & R. Salakhutdinov (Eds.), *Proceedings of the 36th International Conference on Machine Learning: Vol. 97. Proceedings of the 36th International Conference on Machine Learning*. **https://proceedings.mlr.press/v97/burt19a.html**

Csató, L., & Opper, M. (2002). Sparse on-line Gaussian processes. *Neural Computation*, *14*(3).

Davies, A. J. (2015). *Effective implementation of Gaussian process regression for machine learning*.

Duvenaud, D., Lloyd, J., Grosse, R., Tenenbaum, J., & Zoubin, G. (2013, ). Structure discovery in nonparametric regression through compositional kernel search. *International Conference on Machine Learning*.

Ferrari-Trecate, G., Williams, C., & Opper, M. (1998). Finite-dimensional approximation of Gaussian processes. *Advances in Neural Information Processing Systems*.

Gibbs, M., & MacKay, D. (1996). *Efficient implementation of gaussian processes.*

Nickson, T., Gunter, T., Lloyd, C., Osborne, M. A., & Roberts, S. (2015). Blitzkriging: Kronecker-structured stochastic Gaussian processes. *Arxiv Preprint Arxiv:1510.07965.*

Ober, S. W., Artemev, A., Wagenländer, M., Grobins, R., & Wilk, M. van der. (2024, ). *Recommendations for Baselines and Benchmarking Approximate Gaussian Processes.* **https://arxiv.org/abs/2402.09849**

Saatçi, Y. (2011). *Scalable inference for structured Gaussian process models.*

Seeger, M. W., Williams, C. K., & Lawrence, N. D. (2003, ). Fast forward selection to speed up sparse Gaussian process regression. *International Workshop on Artificial Intelligence and Statistics.*

Silverman, B. W. (1985). Some aspects of the spline smoothing approach to non-parametric regression curve fitting. *Journal of the Royal Statistical Society: Series B (Methodological), 47*(1).

Smola, A. J., & Schölkopf, B. (2000, ). Sparse Greedy Matrix Approximation for Machine Learning. *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000).*

Snelson, E., & Ghahramani, Z. (2005). Sparse Gaussian processes using pseudo-inputs. *Advances in Neural Information Processing Systems, 18.*

Titsias, M. (2009, ). Variational learning of inducing variables in sparse Gaussian processes. *Artificial Intelligence and Statistics.*

Wang, K., Pleiss, G., Gardner, J., Tyree, S., Weinberger, K. Q., & Wilson, A. G. (2019, ). Exact Gaussian Processes on a Million Data Points. *Advances in Neural Information Processing Systems.*

Wilson, A., & Nickisch, H. (2015, ). Kernel interpolation for scalable structured Gaussian processes (KISS-GP). *International Conference on Machine Learning.*

Zhu, H., Williams, C. K., Rohwer, R., & Morciniec, M. (1997). *Gaussian regression and optimal finite dimensional linear models.*