Neural Network Surrogate Priors for Efficient Bayesian Optimization

Connie Xu[1], Auralee Edelen[2], Ryan Roussel[2]
[1]Duke University, Durham, NC 27708
[2]SLAC National Accelerator Laboratory, 2575 Sand Hill Rd, Menlo Park, CA 94025

SLAC NATIONAL ACCELERATOR LABORATORY

Introduction

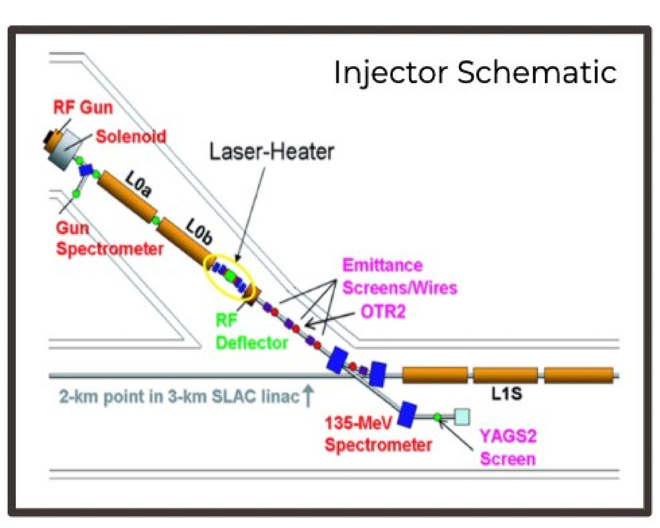
Bayesian Optimization (BO) is a time effective method of automating accelerator tuning for LCLS because we sample as few points as possible to find input values that achieve a good optimum. A problem with this BO application is that its performance worsens exponentially in higher dimensions due to poor scaling. A good prior mean in **Gaussian Processes** (GPs) can help with scaling. Therefore, this **study's objective** is to explore cheap and more expressive (non-constant) prior means for Gaussian Processes to optimize Bayesian Optimization for tuning the injector.

Objective Function

Inputs:	Outputs:
SOL1:solenoid_field_scale (kG*m)	-emittance×bmag (mm-mrad)
CQ01:b1_gradient (kG)	
SQ01:b1_gradient (kG)	
QA01:b1_gradient (kG)	
QA02:b1_gradient (kG)	
QE01:b1_gradient (kG)	
QE02:b1_gradient (kG)	
QE03:b1_gradient (kG)	
QE04:b1_gradient (kG)	

$$\beta_{mag} = \frac{1}{2}\left(\frac{\bar{\beta}}{\beta} + \frac{\beta}{\bar{\beta}}\right) + \frac{1}{2}\left(\alpha\sqrt{\frac{\bar{\beta}}{\beta}} - \bar{\alpha}\sqrt{\frac{\beta}{\bar{\beta}}}\right)^2$$

$$\beta = \frac{\sigma^2}{\varepsilon} \quad [1]$$



- We use BO during accelerator tuning to find input values to the injector that minimize emittance×bmag (**maximize -emittance×bmag**).
- The objective function employs a Neural Net (NN) surrogate model of the accelerator injector to prototype this optimization approach.

Gaussian Processes

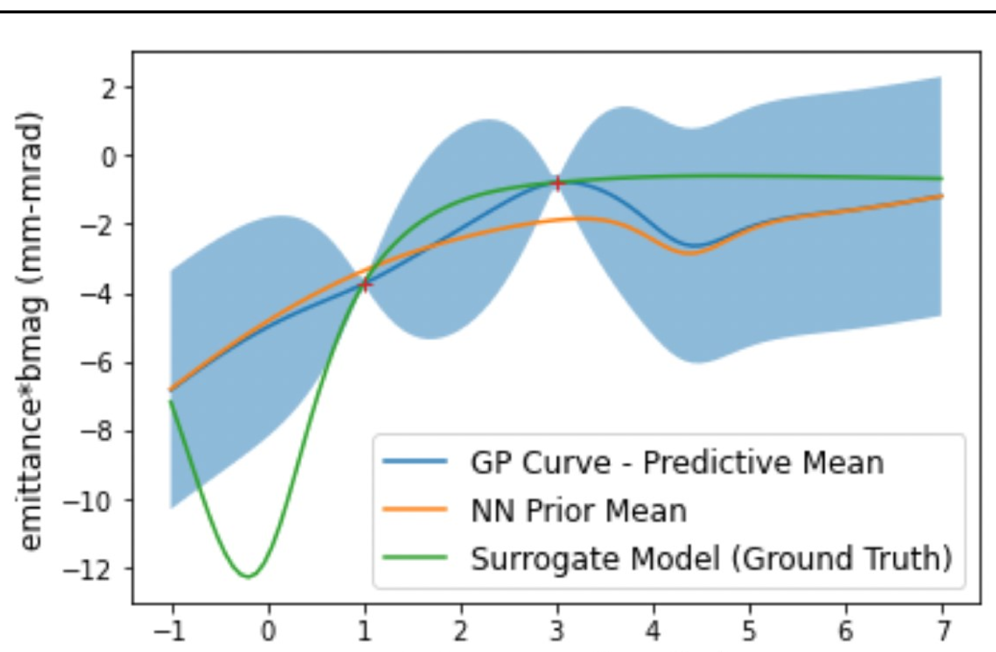


Figure 1: Visualization of a GP's posterior curve (blue) fitted to two data samples (red) compared to the GP's prior mean curve (orange).

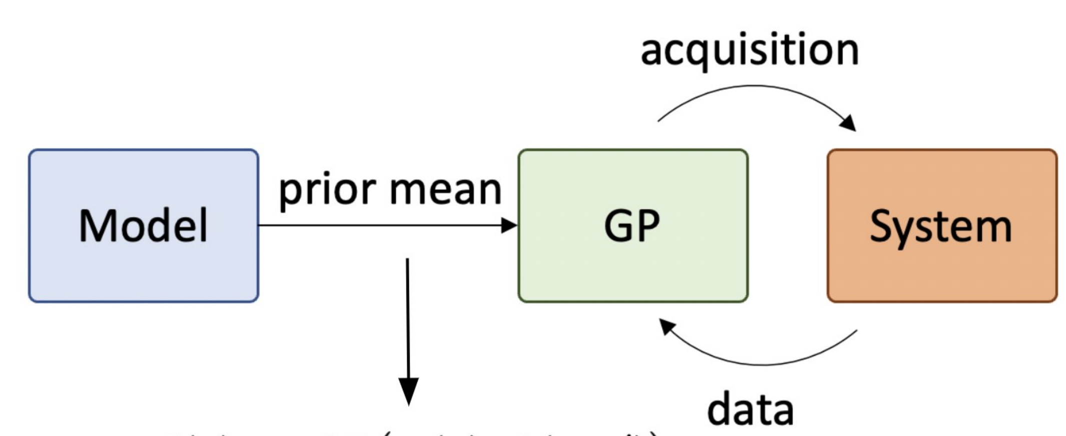
The GP and predictive mean function (blue line) are defined as:

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}),\ k(\mathbf{x},\mathbf{x}'))$$
$$\bar{\mathbf{f}}_* = \mathbf{m}(X_*) + K(X_*,X)K_y^{-1}(\mathbf{y} - \mathbf{m}(X))$$
$$\text{where } K_y = K + \sigma_n^2 I \quad [2]$$

- The GP's **predictive mean** fits through ground truth points and then returns to the prior mean in areas with no data samples.

Implementing Custom Priors

- Using a NN model prior mean that includes prior information of the objective function, BO should be able to find better optima faster.



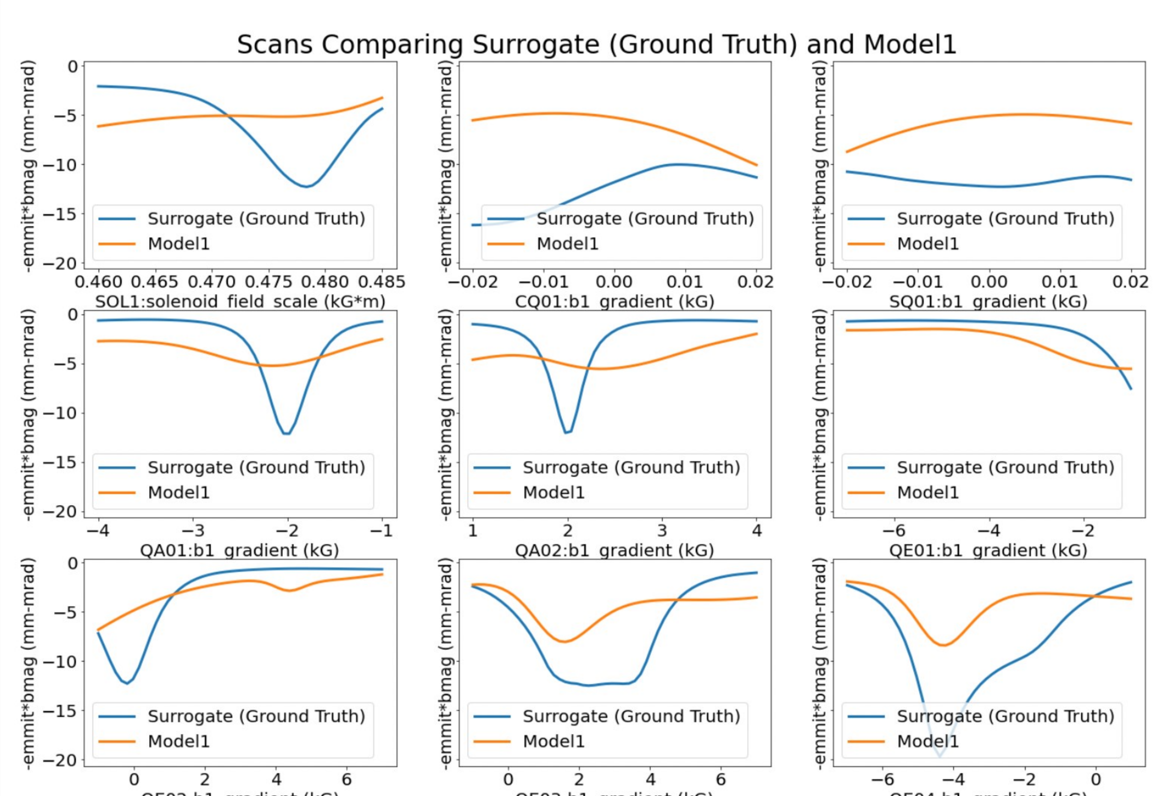
$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}),\ k(\mathbf{x},\mathbf{x}'))$$
$$\bar{\mathbf{f}}_* = \mathbf{m}(X_*) + K(X_*,X)K_y^{-1}(\mathbf{y} - \mathbf{m}(X))$$
$$\text{where } K_y = K + \sigma_n^2 I \quad [2]$$

Figure 2: A model is specified as the mean function of the GP and represents the initial behavior of the objective function. The system gives the GP data samples. The GP then generates a posterior distribution and predictive mean function, which the acquisition function (Upper Confidence Bound) uses to find the inputs most likely to yield an optimum and return it to the system.

- We use PyTorch for all models and BO implementations to aid end-to-end differentiability.

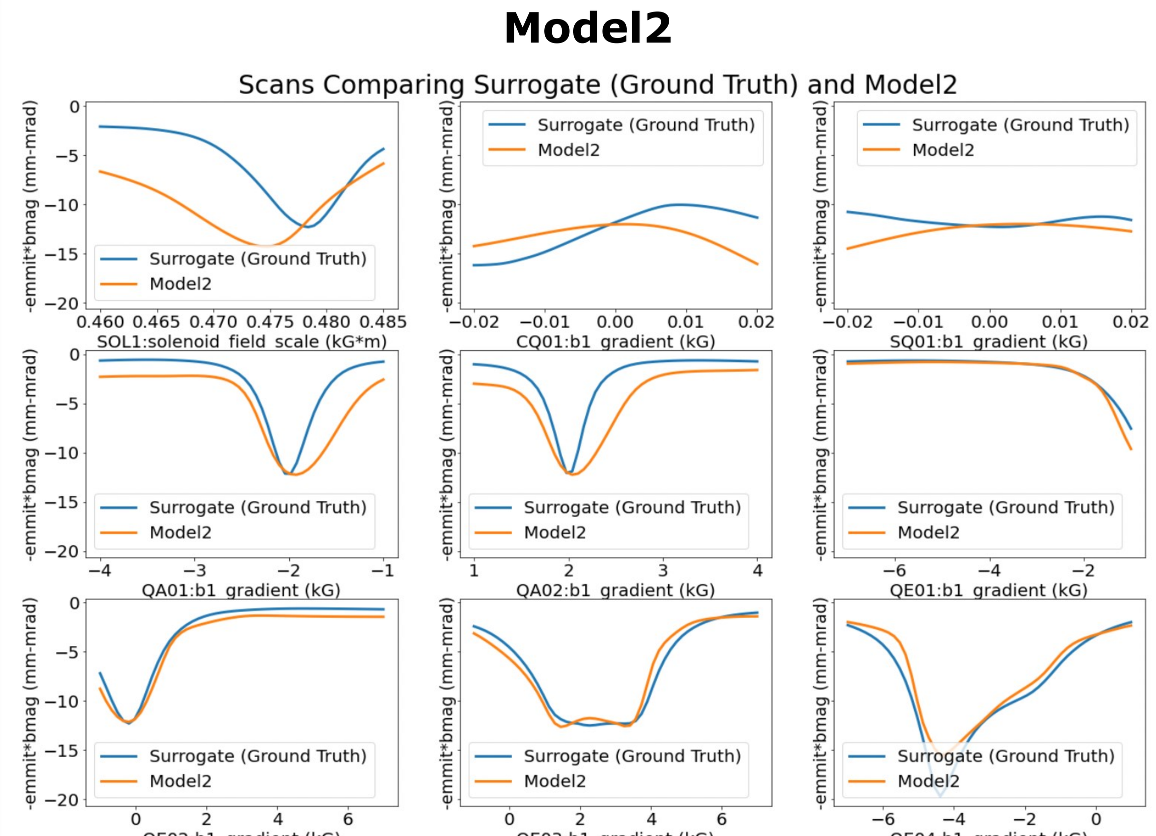
Neural Net Accuracies

Model1



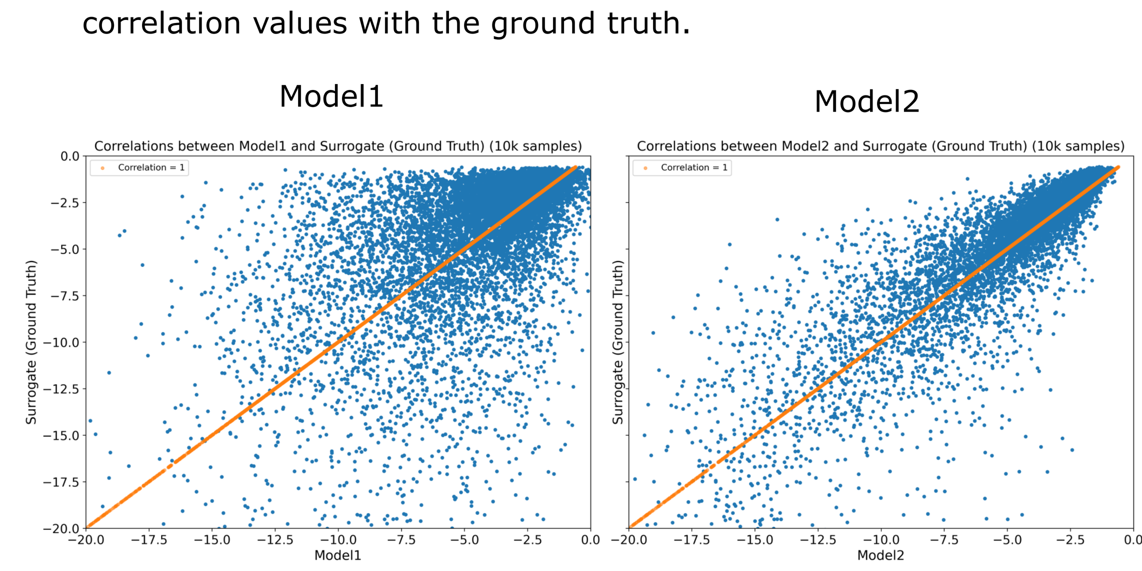
Correlation with ground truth: **0.547**

Model2



Correlation with ground truth: **0.889**

Figure 4: Accuracies of models trained on a mesh grid of 3^9 (top – Model1) and 4^9 (bottom – Model2) data samples are shown with parameter scans of the 9 input variables as well as the correlation values with the ground truth.



BO Comparison Results

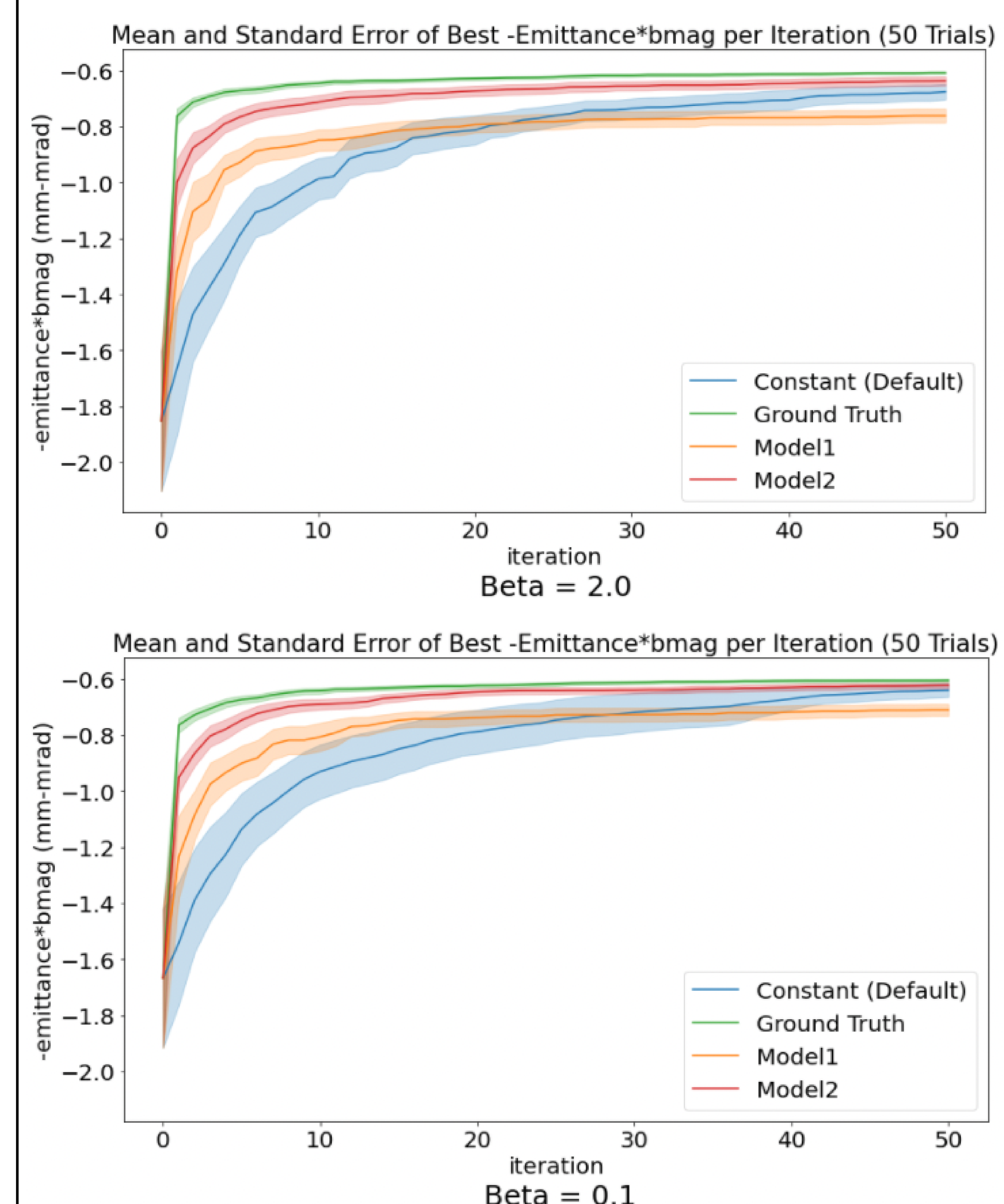


Figure 5: BO comparisons of the constant prior, injector surrogate "ground truth" prior, Model1, and Model2.

- In the trivial case, where the true model is used as a prior, the solution is found almost immediately
- Using better prior models demonstrates better initial performance vs. worse priors
- Both priors initially improve performance over uniform prior
- However, allowing the prior to be retrained at every iteration improves late-stage optimization performance

Conclusions

- Our goal was to determine how accurate a model must be as a prior mean in order to get a performance gain in BO.
- Including prior information in the GP's prior mean always leads to an improvement in BO performance during **coarse tuning**.
- During **fine tuning**, we need a more accurate model to give better BO performance.

Next Steps

- We will explore methods of improving BO performance during the fine-tuning stage.
- We will perform an experimental demonstration with the real accelerator.

References

[1] F. -J. Decker et al., "Dispersion and Betatron Matching into the Linac," Conference Record of the 1991 IEEE Particle Accelerator Conference, 1991, pp. 905-907 vol.2, doi: 10.1109/PAC.1991.164485
[2] Rasmussen, C. E. and Williams, C. K. I. *Gaussian Processes for Machine Learning*. MIT Press, 2006.