Bayesian Optimal Experimental Design for Adaptive Training of Neural Network Surrogate Models

Aniket Jivani, Xun Huan

University of Michigan

March 4, 2025

SIAM CSE 2025

Background: Optimal Experimental Design

A framework to answer: what design conditions or knobs for running an experiment are the "best"?

The notion of best needs to be tailored to the goals of the experimenter. Examples:

- reduction in uncertainty of free parameters
- reduction in uncertainty of QoIs based on model parameters

Background: Optimal Experimental Design

A framework to answer: what design conditions or knobs for running an experiment are the "best"?

The notion of best needs to be tailored to the goals of the experimenter. Examples:

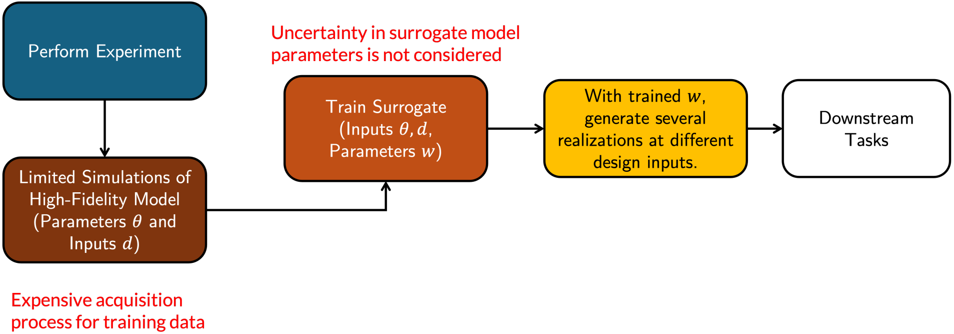
- reduction in uncertainty of free parameters
- reduction in uncertainty of QoIs based on model parameters
- calibration to minimize distance to observed data
- improve correlations between models of multiple fidelities

  and many more . . .

Background

When high-fidelity model evaluations are too expensive, we typically:

- build a surrogate model after collecting data in one shot
- use evaluated surrogate as a replacement for the forward model in future inference tasks.



Objective

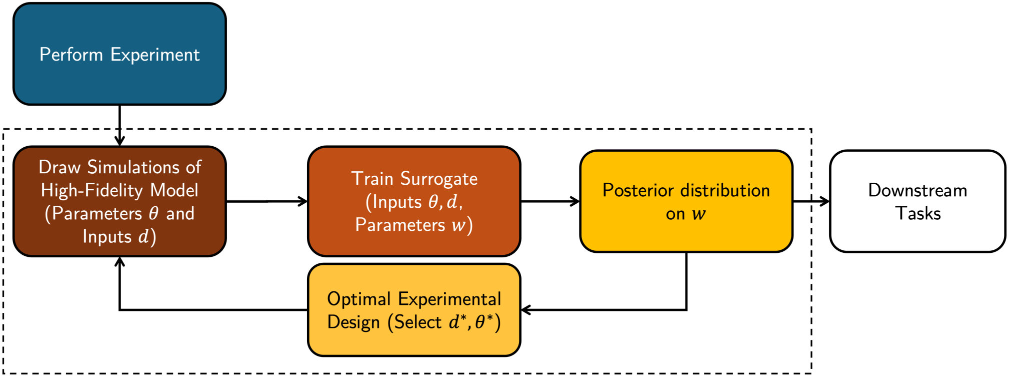
Can we setup training of surrogate models as an OED problem?

Proposed Approach:

- Quantify uncertainty in parameters of surrogate model via Bayesian Inference

- Iterate between inference and maximization of EIG criterion to select new training points.

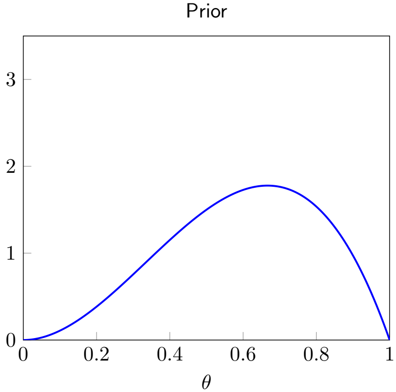
Proposed Workflow

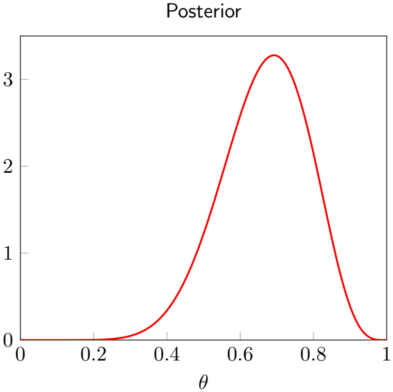


Definition

Quantify and update uncertainties in parameters θ after collecting new observations via Bayes' rule:

$$p(w|y) = \frac{p(y|w)p(w)}{p(y)}$$





Setting up a BNN

Variational Inference (VI)

Approximate a target distribution $p(w|x,y)$, using a simpler $q^*_\phi(w|x,y)$ from a known family, minimize the KL divergence:

$$q^* = \arg\min_{q \in \mathcal{Q}} [D_{\mathsf{KL}}(q||p)]$$

- [Blundell et al. 2015] BayesByBackprop: Mean-field VI - maximize evidence lower bound (ELBO) and learn ϕ for independent Gaussian distributions on weights via backpropagation.
- Improved variance reduction for minibatches by Flipout-based VI [Wen et al., 2018]

Stein Variational Gradient Descent

[Liu and Wang, 2016] Removes restrictive assumptions regarding independence of weights by transporting a set of particles to approximate the true posterior density.

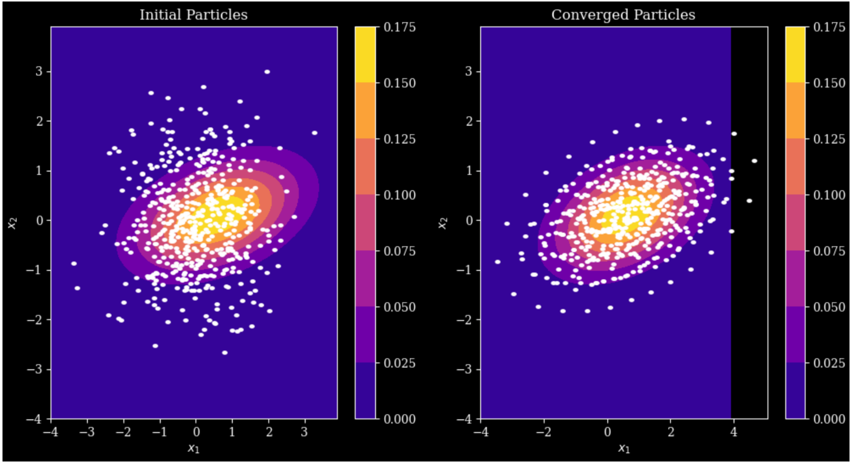
For n particles initialized from $q(w)$, update positions at every iteration ℓ via:

$$w_i^{\ell+1} \leftarrow w_i^{\ell} + \epsilon \hat{\phi}(w_i^{\ell})$$

where:

$$\hat{\phi}^*(w_i^{\ell}) = \frac{1}{n} \sum_{i=1}^{n} \Big(\underbrace{k(w_j^{\ell}, w_i^{\ell}) \nabla_{w_j^{\ell}} \log p(w_j^{\ell} | G, d)}_{\text{Push towards high-density regions of target}} + \underbrace{\nabla_{w_j^{\ell}} k(w_j^{\ell}, x_i^{\ell})}_{\text{Repulsive force preventing mode collapse}} \Big)$$

Example of SVGD



Setup for OED

$$y = G(\theta_{\text{model}}, d) + \varepsilon$$

$$G = \tilde{G}(\theta_{\text{model}}, d, w) + \eta$$

y, ϵ: observations and noise model

G, d: forward model and design variables

θ, w: weights of forward model and surrogate

$\tilde{G}$: surrogate model parametrized by w

η: Discrepancy term

Choosing η

Since we assume no stochasticity in our simulator output, a reasonable choice for the discrepancy is:

$$\eta \sim \mathcal{GP}(0, K_\phi((\theta_{\text{model}}, d), (\theta'_{\text{model}}, d')))$$

We use a GP with hyperparameters ϕ on the covariance kernel. This effectively models the residual or correction term and reduces the uncertainty near the training data.

EIG on Surrogate Model Parameters

For networks built with training snapshots varying with d, select d for next experiment via:

$$d^* = \arg\max_d U(d) = \arg\max_d \mathbb{E}_{G|d}\left[D_{\mathsf{KL}}\left(p(w|G,d)||p(w)\right)\right]$$

$$\begin{aligned}
U(d) &= \int_{\mathcal{G}}\int_{\mathcal{W}} \ln\left[\frac{p(w|G,d)}{p(w)}\right] p(G,w|d)dGdw \\
&= \int_{\mathcal{G}}\int_{\mathcal{W}} \left[\ln[p(G|w,d)] - \ln[p(G|w)]\right] p(G|w,d)p(w)dwdG \\
&\approx \frac{1}{N_{\text{out}}}\sum_{i=1}^{N_{\text{out}}}\left[\ln[p(G^{(i)}|w^{(i)},d] - \ln\left[\frac{1}{N_{\text{in}}}\sum_{j=1}^{N_{\text{in}}}(p(G^{(i)}|w^{(i,j)},d))\right]\right]
\end{aligned}$$

Posterior comparisons

$$G(d) = 0.4\sin(4d) + 0.5\cos(12d)$$

$\tilde{G}$ is a 2-hidden layer network with 8 units each.

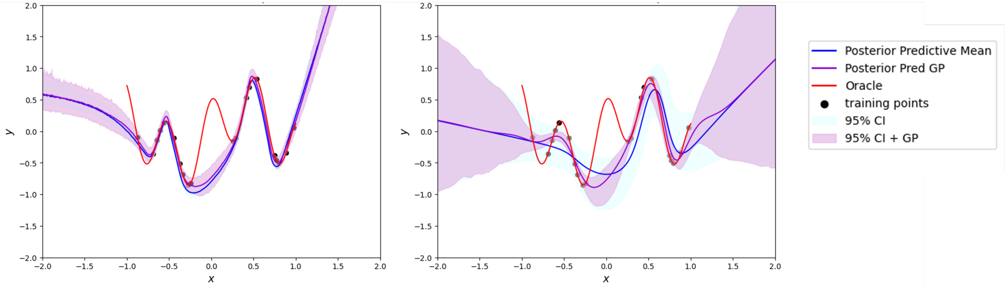
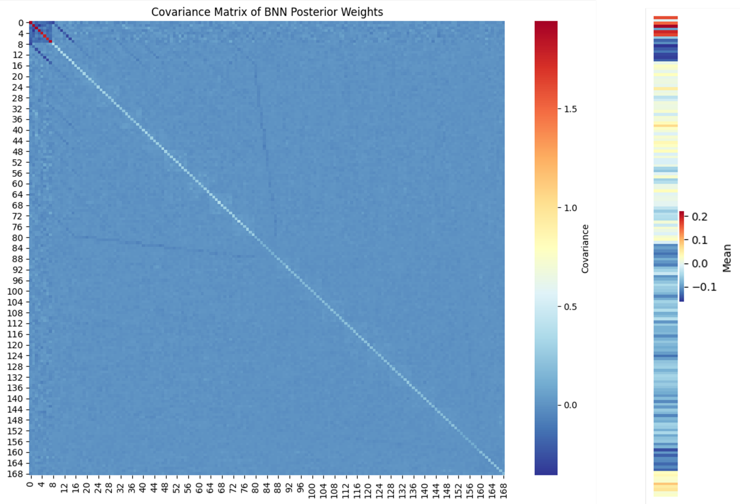
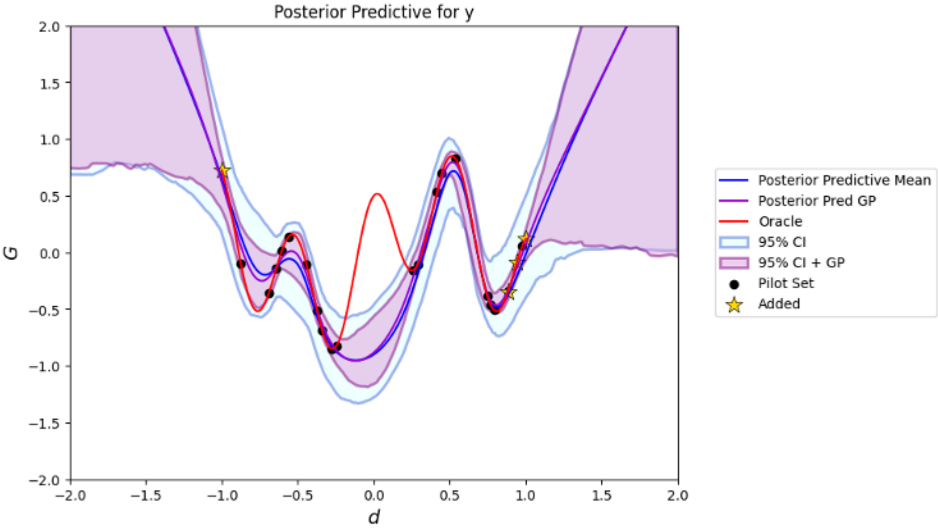


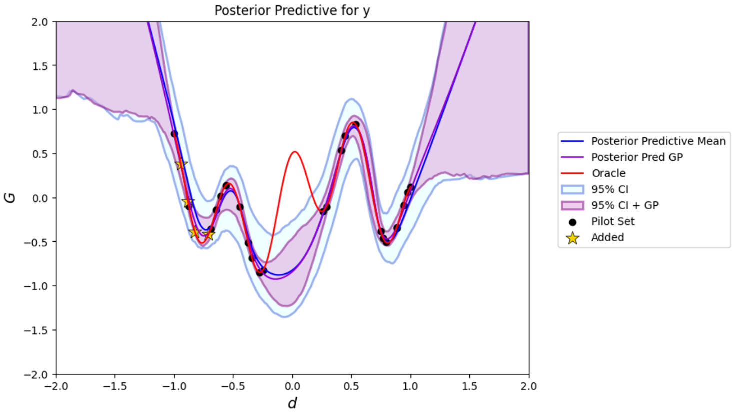
Figure: Results from BNN training with MFVI and SVGD on pilot set.

SVGD Posterior on Weights



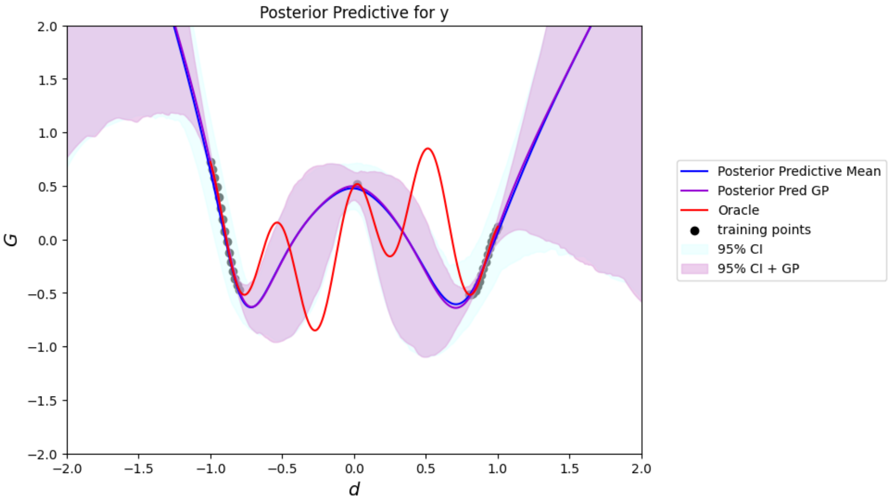
Retraining with Acquired Points



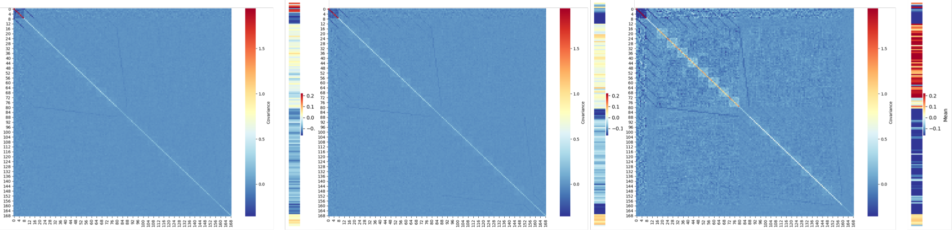


The quality of the fitted NN strongly depends on the pilot training set - here it smoothly interpolates in the middle of the domain where we have gaps in the training data.

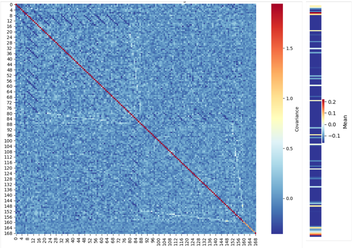
Initializing with a Different Pilot Set



Batch to batch weight posteriors



Random Design Covariance



Setting up goal-oriented design

Assuming $z = H(\tilde{G}(w, d^*))$ i.e. evaluating a function of the NN predictions at a fixed d, the MC estimator for GO-OED is (similar to formulation of Zhong et al. 2024):

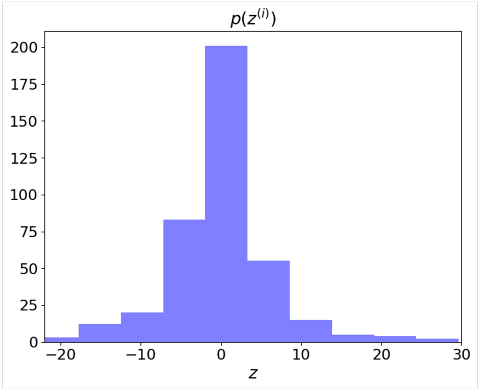
$$
\begin{aligned}
U(d) &= \int_{\mathcal{G}} \int_{\mathcal{Z}} p(z|G,d) \ln \frac{p(z|G,d)}{p(z)} p(G|d) dz dG \\
&= \int_{\mathcal{G}} \int_{\mathcal{Z}} \ln \frac{p(z|G,\theta)}{p(z)} p(z, G|\theta) dz dG \\
&= \int_{\mathcal{G}} \int_{\mathcal{Z}} \int_{\mathcal{W}} \ln \frac{p(z|G,\theta)}{p(z)} p(w, G, z|d) dw dz dG \\
&= \int_{\mathcal{G}} \int_{\mathcal{Z}} \int_{\mathcal{W}} \ln \frac{p(z|G,d)}{p(z)} p(G|w) p(w|G,d) p(z|w) dw dz dG \\
&\approx \frac{1}{N_{\text{out}}} \sum_{i=1}^{N_{\text{out}}} \left[\frac{1}{N_{\text{in}}} \sum_{i=1}^{N_{\text{in}}} \ln \left[p(z^{(i,j)}|G^{(i)}, d) \right] - \ln \left[p(z^{(i)}) \right] \right]
\end{aligned}
$$

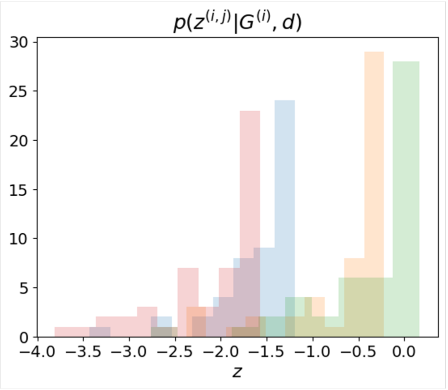
SVGD-based GO-OED

1. Prior samples $w^{(i)}$ can be perturbed to generate initial state of particles for inner loop rather than reusing samples from outer loop; to speed up generation of posterior samples.
2. PDFs of prior predictive $p(z^{(i)})$ and posterior predictive $p(z^{(i,j)})$ are estimated via KDE.

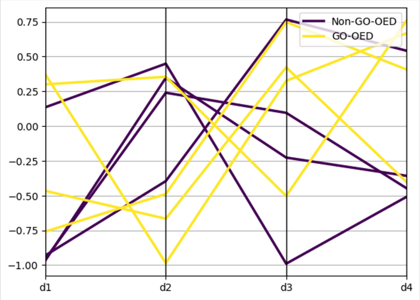
Summary of Prior and Posterior Predictive

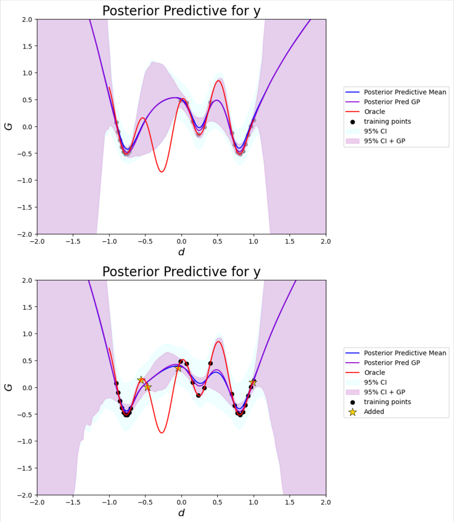
$$z = \tilde{G}(w, d = -0.5)$$





Results of batch design from U_{param} and U_{pred}





Conclusions

1. We looked at strategies to adaptively train neural network surrogates.
2. For a small example, we perform OED targetting uncertainty reduction in NN parameters as well as show a limited setup to target downstream QoIs.

   Scaling up the latter will require more efficient EIG estimation in the goal-oriented setting as well as inference methods for BNNs.

Next Steps

1. Adapting trained NNs for goal-oriented designs with prediction model $H(\theta, d)$ *different from but correlated to* the high-fidelity model G.
2. Tailoring kernel functions for common higher-dimensional inputs to NNs like image data, for instance, convolutional GP kernels [van der Wilk et al. 2017]
3. Determining resource allocations as a trade-off between surrogate building and subsequent outer-loop tasks.[1]

[1] For a sneak peek at balancing pilot sampling and estimation for stochastic optimization in multifidelity frameworks, go to Thomas Coons' talk - MS114, Room 106, 4:40 pm

Acknowledgement:

This work relates to Department of Navy award N00014-23-1-2735 issued by the Office of Naval Research.

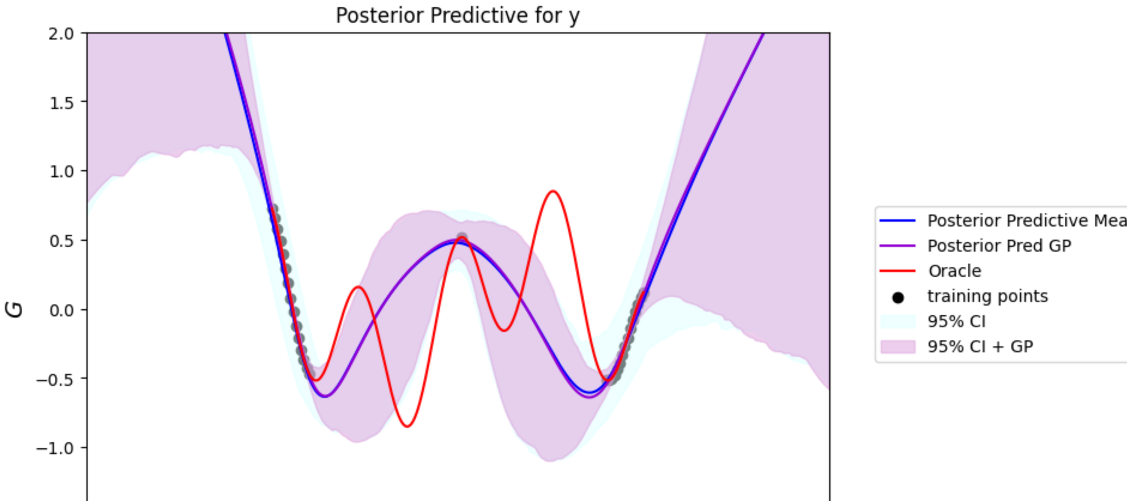
This material is based upon work supported in part by Sandia National Laboratories Advanced Science and Technology Laboratory Directed Research and Development program. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

SIAM Travel Award for CSE25.

Thank you!

Contact: ajivani@umich.edu

Initializing with a Different Pilot Set



OED Classification