Data-Efficient Scientific Design Optimization with Neural Network Surrogates

Jayaraman J. Thiagarajan, Rushil Anirudh, Yamen Mubarka, Irene Kim, Peer-Timo Bremer, J. Luc Peterson, Brian Spears
Lawrence Livermore National Laboratory
Livermore, CA 94550, USA

Vivek Narayanaswamy
Arizona State University
Tempe, AZ 85281, USA

Abstract

Design optimization is a central problem in the sciences, wherein computationally extensive simulations or expensive experiments are used to characterize physical systems. This paper focuses on advancing sequential optimization in these applications, particularly at low data regimes, using neural network surrogates. To this end, we introduce $\Delta-$UQ, a new uncertainty estimator for neural networks, based on stochastic data centering. Using empirical studies with synthetic data and a real-world Inertial Confinement Fusion (ICF) simulator, we demonstrate the effectiveness of the proposed approach in recovering optima from complex response surfaces.

Keywords: Design optimization, scientific simulators, uncertainty quantification, black-box optimization, deep neural networks

1. Introduction

At the core of AI-powered scientific discovery lies the need to perform design optimization for maximizing a chosen target objective, and to enable automated exploration in high-dimensional parameter spaces. Formally, denoting the underlying scientific process (*e.g.*, a simulation code or experiment) using a high-dimensional function $\mathrm{f} : \mathcal{D} \rightarrow \mathbb{R}$, our goal is to solve the following optimization problem: $\mathrm{x}^* = arg\max_{\mathrm{x}\in\mathcal{D}} \mathrm{f}(\mathrm{x})$. Here, $\mathcal{D}$ refers to a *bounded* design space comprising D different parameters with their corresponding value ranges $[\ell_d, h_d], \forall d = 1 \cdots D$. While f can be explicitly evaluated for any $\mathrm{x} \in \mathcal{D}$, its first and second-order information are unknown, thus making such a global optimization challenging. Commonly referred to as *black-box optimization* (Audet and Hare, 2017) (shortly BBO), this formulation is adopted in a wide-range of applications (Schneider et al., 2020; Wang et al., 2020; Gonzalvez et al., 2019; Ren et al., 2021).

Bayesian Optimization (BO) based on statistical surrogates form an important class of solutions for BBO (Shahriari et al., 2015). Given an initial experiment design and their function evaluation, these techniques incrementally select candidates to effectively achieve the so-called *exploration-exploitation* trade-off, and to identify optimal designs in $\mathcal{D}$. With deep neural networks (DNNs) becoming the standard for approximating complex scientific processes, this paper studies their use in design optimization. While deep models have produced state-of-the-art results in active learning

. This work was performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under Contract No. DE-AC52-07NA27344, Lawrence Livermore National Security, LLC, and was supported by the LLNL-LDRD Program under Project No. 21-ERD-028. LLNL-CONF-835898.

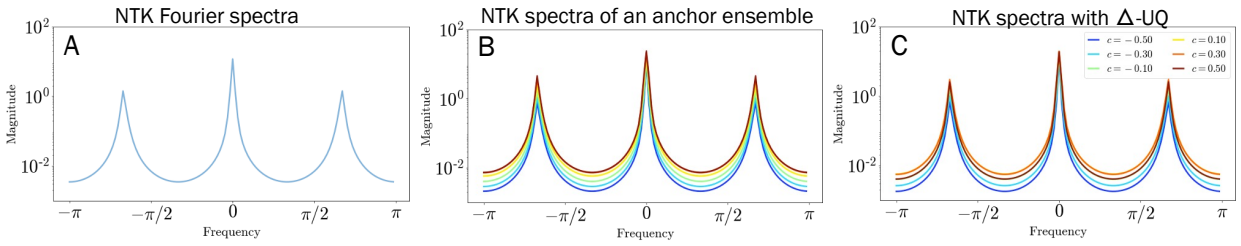


Figure 1: Fourier spectrum of an NTK for an MLP model (A); spectra of an anchor ensemble (B); and NTK spectra from $\Delta-$UQ (C). Note, the inputs are pre-processed through a sinusoidal PE. We make two key observations – a) shifts in the input domain cause the *effective* NTK to be distinct as a function of the shift c; and b) $\Delta-$UQ achieves a similar effect but with a single model.

for classification (Ash et al., 2019), obtaining reliable uncertainties and understanding their utility in BBO are ongoing topics of research. In particular, our focus is on scientific problems, where the computational or financial cost of obtaining a new evaluation of the function f can be prohibitively expensive. In practice, existing solutions tend to be ineffective, even in moderately high-dimensional design spaces (4-10), due to the lack of meaningful uncertainty estimates with such low sample sizes.

Proposed Work. Our goal is to advance DNN-powered design optimization in scientific applications. To this end, we present a novel uncertainty estimator, $\Delta-$UQ for DNNs, that is effective even in low data regimes. Conceptually, $\Delta-$UQ uses a stochastic data centering strategy to sample from the hypothesis-space of solutions. For this study, we consider the problem of optimizing design parameters for an inertial confinement fusion (ICF) (Betti and Hurricane, 2016) simulator. The physics of ICF fusion ignition are predicated on interactions between multiple strongly nonlinear physics mechanisms that have multivariate dependence on a number of controllable parameters. This presents the designer with a complicated response function that has sharp, nonlinear features, while real experiments can often cost upwards of millions of dollars (Moses, 2010). Using empirical comparisons to widely adopted uncertainty estimation techniques, we demonstrate the efficacy of our approach in recovering the optima with standard Bayesian optimization.

2. $\Delta-$UQ: Epistemic Uncertainties via Stochastic Data Centering

Background. Accurately estimating epistemic uncertainties in a deep neural network (DNN) is critical for enabling effective design optimization. Existing approaches include Bayesian methods (Wilson and Izmailov, 2020; He et al., 2020; Neal, 2012; Blundell et al., 2015), Monte Carlo approximations such as MC Dropout (Gal and Ghahramani, 2016), and empirical methods such as Deep Ensembles (DEns) (Lakshminarayanan et al., 2017). Recent advances in the neural tangent kernel (NTK) theory (Jacot et al., 2018; Arora et al., 2019; Bietti and Mairal, 2019; Lee et al., 2019) provide a convenient framework for analyzing deep uncertainty estimators. The basic idea of NTK is that, when the width of a neural network tends to infinity and the learning rate of SGD tends to zero, the function $f(\mathrm{x};\boldsymbol{\theta})$ converges to a solution obtained by kernel regression using the NTK defined as $\mathbf{K}_{\mathrm{x}_i\mathrm{x}_j} = \mathbb{E}_{\boldsymbol{\theta}}\left\langle \frac{\partial f(\mathrm{x}_i,\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}, \frac{\partial f(\mathrm{x}_j,\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right\rangle$. When the samples $\mathrm{x}_i, \mathrm{x}_j \in \mathcal{S}^{d-1}$, i.e., points on the hypersphere and have unit norm, the NTK for a simple 2 layer ReLU MLP can be simplified as a dot product kernel (Arora et al., 2019; Bietti and Mairal, 2019; Lee et al., 2019): $\mathbf{K}_{\mathrm{x}_i\mathrm{x}_j} = h_{\mathrm{NTK}}(\mathrm{x}_i^\top \mathrm{x}_j) = \frac{1}{2\pi}\mathrm{x}_i^\top \mathrm{x}_j(\pi - \cos^{-1}(\mathrm{x}_i^\top \mathrm{x}_j))$.

2.1 Anchor Ensembles: Ensembling via Data Centering

We use training data $\mathcal{D} = \{(\mathrm{x}_i, y_i)\}_{i=1}^n$, where $\mathrm{x}_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$, to train a model $f(\boldsymbol{\theta}) \in \mathcal{H}$ with randomly initialized weights $\boldsymbol{\theta}_0$ and hypothesis space $\mathcal{H}$. Our idea is to center the dataset around an *anchor* c, *i.e.*, a sample randomly drawn from $\mathcal{D}$, and train the model f_{c}. If the NTK induced by f_c is shift-invariant, we would obtain identical models for different anchors, *i.e.*, $f_{\mathrm{c}_1} = \cdots = f_{\mathrm{c}_k}$. However, NTKs for MLPs or CNNs are not inherently shift-invariant (Lee et al., 2019), and hence we find that the variation across the predictions is a strong indicator of epistemic uncertainties.

Effect of shifted training on NTK: Assuming $\mathrm{x}_i - \mathrm{c}$ and $\mathrm{x}_j - \mathrm{c}$ are unit norm, we simplify $h_{\mathrm{NTK}}((\mathrm{x}_i - \mathrm{c})^\top(\mathrm{x}_j - \mathrm{c}))$ using a Taylor series expansion for $\cos^{-1}$: $\cos^{-1}(\mathrm{u} - \mathrm{c}) \approx \cos^{-1}(\mathrm{u}) + \frac{\mathrm{c}}{\sqrt{1-(\mathrm{u}-\mathrm{c})^2}}$. Expanding $(\mathrm{x}_i - \mathrm{c})^\top(\mathrm{x}_j - \mathrm{c})$ as $\mathrm{x}_i^\top \mathrm{x}_j - \mathrm{c}^\top(\mathrm{x}_i + \mathrm{x}_j - \mathrm{c})$ and letting $\mathrm{v} = (\mathrm{x}_i + \mathrm{x}_j - \mathrm{c})$, we obtain the expression for h_{NTK} under a shifted domain as follows:

$$\mathbf{K}_{(\mathrm{x}_i-\mathrm{c})(\mathrm{x}_j-\mathrm{c})} \approx \frac{1}{2\pi}\mathrm{x}_i^\top \mathrm{x}_j(\pi - \cos^{-1}(\mathrm{x}_i^\top \mathrm{x}_j)) - \frac{1}{2\pi}\mathrm{c}^\top \mathrm{v}(\pi - \cos^{-1}(\mathrm{x}_i^\top \mathrm{x}_j)) - \frac{\mathrm{c}(\mathrm{x}_i^\top \mathrm{x}_j - \mathrm{c}^\top \mathrm{v})}{2\pi\sqrt{1-(\mathrm{x}_i^\top \mathrm{x}_j - \mathrm{c}^\top \mathrm{v})^2}}$$

$$= \mathbf{K}_{\mathrm{x}_i\mathrm{x}_j} - \Gamma_{\mathrm{x}_i,\mathrm{x}_j,\mathrm{c}}, \tag{1}$$

where we combine all terms dependent on c into $\Gamma_{\mathrm{x}_i,\mathrm{x}_j,\mathrm{c}}$, which also behaves as a dot product kernel. From (1), we note that a trivial shift in the domain results in a non-trivial shift in the NTK function itself. In other words, (1) outlines the *effective* NTK as a function of c. We also note that $\boldsymbol{\Gamma}$ does not affect the spectral properties of the original NTK, as we observe in Figure 1.

In contrast to existing ensembling approaches, even for a fixed $\boldsymbol{\theta}_0$, we find that one can make the NTK stochastic (in c) via stochastic centering. To demonstrate this, we compute the Fourier spectra using the same MLP on several shifted domains in Figure 1(B). The original spectrum for the MLP without any shift in the training domain is shown for comparison in 1(A). Note that, we constructed positional embeddings (PE), based on sinusoidal functions, prior to building the MLP model. We notice that each individual shift leads to a different NTK.

2.2 $\Delta-$UQ: Rolling Anchor Ensembles into a Single Model

Since different models in an anchoring-based ensemble are trained with the same initialization, we present a new technique to approximate the uncertainties using a single model. We perform a simple coordinate transformation by lifting the domain to a higher dimension as $\mathcal{E} : \mathrm{x} \to \{\mathrm{c}, \mathrm{x} - \mathrm{c}\}$, we refer to the residual by $\Delta = \mathrm{x} - \mathrm{c}$. This transformation allows the use of multiple representations (w.r.t. many anchors) for the same x, *i.e.*, $f_\Delta(\{\mathrm{c}_1, \mathrm{x} - \mathrm{c}_1\}) = f_\Delta(\{\mathrm{c}_2, \mathrm{x} - \mathrm{c}_2\}) = \cdots = f_\Delta(\{\mathrm{c}_k, \mathrm{x} - \mathrm{c}_k\})$, where f_Δ refers to the $\Delta-$UQ model that takes the tuple $(\{\mathrm{c}_k, \mathrm{x} - \mathrm{c}_k\})$ and predicts the target y.

```
for inputs, targets in trainloader:
    A = Shuffle(inputs) %% Anchors
    D = inputs—A %% Delta
    X_d = torch.cat([A, D],axis=1)
    y_d = model(X_d) %% prediction
    loss = criterion(y_d,targets)
```

Figure 2: $\Delta-$UQ Training.

Training. During training, for every input x_i we choose an anchor as random sample from the training dataset. Subsequently, we obtain the coordinate transformation $\{[\mathrm{c}, \mathrm{x}_i - c], y_i\}$, using which we train the model. With vector-valued data, this is implemented as a simple concatenation. We show simple a Pytorch snippet for training $\Delta-$UQ . Over the course of training, every training pair gets combined with a large number of anchors. Since the prediction on this training pair – regardless of anchor choice – must always be the same, this places a consistency in the predictions across different anchor choices. This consistency can trade-off with diversity in functions that can be learned when compared to an anchor ensemble, where the models are trained independently.

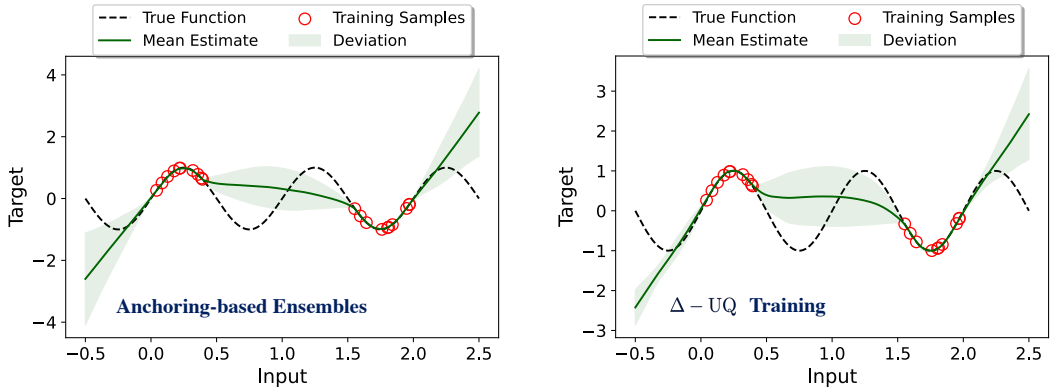


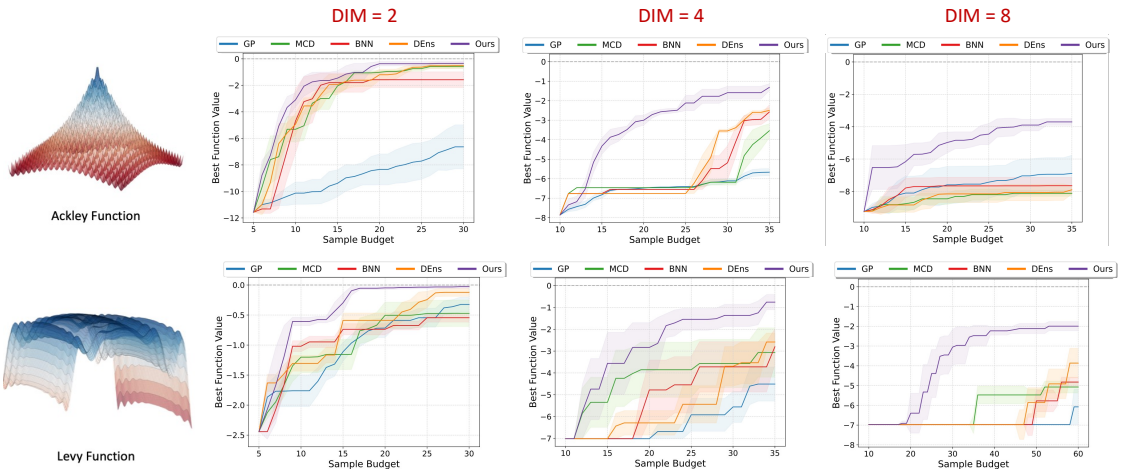
Figure 3: Comparing anchor ensembles and $\Delta-$UQ in function fitting with an MLP. As expected, we see that the disagreement between models in an anchor ensemble correlate strongly with the epistemic uncertainty, and that $\Delta-$UQ , with a single model, matches this behavior very closely.

This can be seen in the comparisons of the NTK spectra for $\Delta-$UQ with anchor ensembles in Figures 1(C). In practice, we find that the diversity from this single model is still sufficiently large, to estimate good quality uncertainties. We hypothesize that, since $\Delta-$UQ uses simple data manipulation to sample from the hypothesis space of the DNN, it can be sample efficient, unlike competing methods that require more data to be able to produce meaningful uncertainties.

Inference. For a test sample x_t, we obtain the prediction from $\Delta-$UQ as the mean prediction across several randomly chosen anchors; and the standard deviation around these predictions is our estimate for the epistemic uncertainty. In other words, we marginalize out the effect of anchors to obtain the final prediction mean and uncertainty. Formally, the predictive distribution is given by $p(y_t|\mathrm{x}_t) = \int_{\mathrm{c}\in\mathbf{X}} p(y_t|\mathrm{x}_t, \mathrm{c}, \boldsymbol{\theta})p(\mathrm{c})d\mathrm{c}$. In Figure 3, we show an 1D regression example using 20 training examples along with the predicted mean and estimated uncertainties. As it can be seen, both the anchoring-based ensemble (left) and $\Delta-$UQ training show high uncertainties around regions with no training samples, though the former requires training 20 networks.

3. Experiments

Setup. We use the following baseline uncertainty estimation approaches in our study: (i) Gaussian processes (GP); (ii) Monte-Carlo dropout (MCD); (iii) Bayesian neural networks (BNN) trained via variational inferencing; and (iv) deep ensembles (DEns). For all neural network surrogates, we computed positional embeddings (sinusoidal) of the raw parameter inputs prior to building a fully-connected network with 4 hidden layers each containing 128 neurons and ReLU activation. All methods were trained with the same set of hyperparameters: Adam optimizer learning rate $1e-4$ and 500 epochs, except for BNN, which required 1000 epochs for convergence. With MCD, we used 50 forward passes for each sample to obtain the uncertainties. Finally, for $\Delta-$UQ, we set the number of anchors for inferencing as $\min(20, n)$, where n is the number of samples in the observed dataset in any iteration. The DEns model was constructed using 5 constituent members, each trained with a different initialization. The number of initial samples (*Init.*) and number of steps (*Steps*) in the optimization were fixed for all methods. In each round of BO, we used $10,000$ samples for initialization and 15 restarts (i.e., multistart acquisition function optimization), and finally one candidate was evaluated with the black-box function. Since the goal is to reach the global optima with the fewest number of samples, we use the widely adopted area under the *iteration vs best achieved function value* curve to obtain a holistic evaluation of different approaches.



Function	Dim.	Init.	Steps	GP	MCD	BNN	DEns	Ours
Ackley	2	5	25	0.23 ± 0.08	0.76 ± 0.03	0.71 ± 0.1	0.75 ± 0.04	0.83 ± 0.03
Levy	2	5	25	0.57 ± 0.18	0.61 ± 0.32	0.55 ± 0.16	0.59 ± 0.16	0.83 ± 0.03
Ackley	4	10	25	0.17 ± 0.02	0.06 ± 0.04	0.1 ± 0.03	0.14 ± 0.02	0.59 ± 0.05
Levy	4	10	25	0.1 ± 0.08	0.4 ± 0.3	0.27 ± 0.21	0.21 ± 0.1	0.62 ± 0.2
Ackley	8	10	50	0.06 ± 0.09	0.09 ± 0.13	0.08 ± 0.12	0.11 ± 0.05	0.36 ± 0.09
Levy	8	10	50	0.12 ± 0.04	0.16 ± 0.08	0.11 ± 0.07	0.13 ± 0.07	0.47 ± 0.03

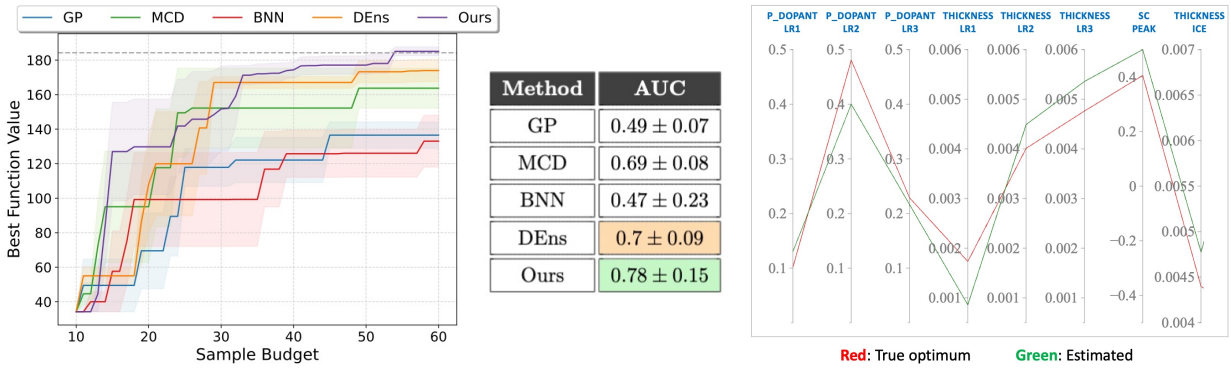
Figure 4: **Convergence curves obtained with different uncertainty estimators:** We show the best function value achieved for two different functions at dimensions 2, 4 and 8 respectively (for 1 random seed). We also include the table for AUC scores (averaged across 5 random seeds).

Synthetic Data. In order to analyze the behavior of the proposed approach, we use a standard Bayesian optimization setup (implemented using BoTorch), and use the popular expected improvement (EI) score to perform candidate selection. We first consider two benchmark functions, namely Ackley and Levy, in varying dimensions. From Figure 4, we find that $\Delta-$UQ produces significantly higher AUC scores in comparison to existing baselines. While MCD and DEns behave reasonably well in low dimensions, their performance suffers as dimension increases. Further, we find that the performance of BNN is generally lower due to the low small samples sizes that we operate in.

3.1 Use-Case: Inertial Confinement Fusion

In this section, we test our method's performance on a real-world scientific application: controlled nuclear fusion via inertial confinement. In the indirect-drive approach to inertial confinement fusion (ICF), a small millimeter-sized spherical capsule filled with hydrogen isotopes (such as deuterium and tritium) are compressed with high-energy x-rays to high temperatures and densities, at which conditions the isotopes can fuse together and release large quantities of energy. The goal of ICF research is to find a design (e.g. capsule material composition and geometry) that when compressed will produce more energy than it consumes (such that it yields positive net energy gain). However, a major challenge is that ICF experiments are costly and the design space vast, making optimization via experimentation difficult. Furthermore, computer models that could be used in digital design and *in silico* engineering require significant computational resources. As such, a data-efficient algorithm for design optimization could significantly accelerate the pursuit of ICF.

The Hydra Simulator As a test of our proposed methods, we created a database of 35,000 ICF simulations using the radiation-hydrodynamics code Hydra (Marinak et al., 2001). The workflow,



Method	AUC
GP	0.49 ± 0.07
MCD	0.69 ± 0.08
BNN	0.47 ± 0.23
DEns	0.7 ± 0.09
Ours	0.78 ± 0.15

Figure 5: **Sequential optimization results with the *Hydra* simulator in ICF.** In each case, we start with 10 initial samples and run 50 rounds of adaptive sampling. We show the convergence curves and corresponding AUC scores for all methods. We also illustrate the optimal design identified by our method along with the known optima (inferred using a large experiment design.)

built with Merlin (Peterson et al., 2022), varied eight design parameters, variations of the National Ignition Facility (NIF) (Moses, 2010) experiment N210808. Seven of the parameters specify the geometry of the capsule, which is comprised of three layers of varying thickness of high-density carbon doped with a varying percentage of tungsten. Six of the design parameters (P_DOPANT_LR[1,2,3] and THICKNESS_LR[1,2,3]) define these layers' atomic percentage of dopant and thickness in centimeters, respectively. Another design variable (THICKNESS_ICE) defines the thickness (cm) of the frozen deuterium-tritium ice fuel layer inside the capsule. The final design variable (SC_PEAK) adjusts the length of final x-ray drive on the capsule (in nanoseconds) around the baseline design, such that negative(positive) values indicate a shorter(longer) compressing drive and consequently less(more) energy put into the experiment.

To construct the dataset, we created a 1000-point latin-hypercube stencil of the seven capsule parameters and moved this stencil linearly through the eighth SC_PEAK variable, such that the same capsule designs were simulated at both low and high energy. For these 35,000 simulations, the best design produced slightly more than 18 mega-joules of energy yield (180 simulator units of energy), which would correspond to a energy gain of approximately 9, if realized in an experiment. The brute-force optimal design can be found in red in the right of Figure 5. The 35,000 simulations consumed approximately 8,750 core-hours on the Lassen supercomputer at LLNL (Computing, 2022). Note, we first fit a surrogate model to the data, and then perform a series of sequential optimization experiments on that surrogate.

Results. From the results in Figure 5, we find that even in this real-world setting, the proposed approach consistently leads to faster convergence to the optimum (indicated by higher AUC scores). Both MCD and DEns perform comparably, but tend to converge to other local optima in the multimodal response surface of Hydra. We also plot the design identified using our approach to the known optima inferred using a large experiment design ($\sim$ 35K samples).

4. Conclusion

We introduced a new method to estimate epistemic uncertainty in deep neural networks, and demonstrated its utility in sequential optimization with scientific simulators. The efficacy of our approach even in low sample regimes warrants further analysis of its behavior and investigating extensions to other model architectures such as graph neural networks and transformers.

References

Sanjeev Arora, Simon Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In *International Conference on Machine Learning*, pages 322–332. PMLR, 2019.

Jordan T Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. In *International Conference on Learning Representations*, 2019.

Charles Audet and Warren Hare. *Derivative-free and blackbox optimization*, volume 2. Springer, 2017.

R Betti and OA Hurricane. Inertial-confinement fusion with lasers. *Nature Physics*, 12(5):435–448, 2016.

Alberto Bietti and Julien Mairal. On the inductive bias of neural tangent kernels. *Advances in Neural Information Processing Systems*, 32, 2019.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International Conference on Machine Learning*, pages 1613–1622. PMLR, 2015.

Livermore Computing. Lassen. `https://hpc.llnl.gov/hardware/compute-platforms/lassen`, 2022. Accessed: 2022-06-01.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.

Joan Gonzalvez, Edmond Lezmi, Thierry Roncalli, and Jiali Xu. Financial applications of gaussian processes and bayesian optimization. *arXiv preprint arXiv:1903.04841*, 2019.

Bobby He, Balaji Lakshminarayanan, and Yee Whye Teh. Bayesian deep ensembles via the neural tangent kernel. *Advances in Neural Information Processing Systems*, 33:1010–1022, 2020.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

Jaehoon Lee, Lechao Xiao, Samuel Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. *Advances in neural information processing systems*, 32, 2019.

M. M. Marinak, G. D. Kerbel, N. A. Gentile, O. Jones, D. Munro, S. Pollaine, T. R. Dittrich, and S. W. Haan. Three-dimensional HYDRA simulations of National Ignition Facility targets. *Physics of Plasmas*, 8(4):22755, April 2001.

EI Moses. The national ignition facility and the national ignition campaign. *IEEE Transactions on Plasma Science*, 38(4):684–689, 2010.

Radford M Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.

J. Luc Peterson, Ben Bay, Joe Koning, Peter Robinson, Jessica Semler, Jeremy White, Rushil Anirudh, Kevin Athey, Peer-Timo Bremer, Francesco Di Natale, David Fox, Jim A. Gaffney, Sam A. Jacobs, Bhavya Kailkhura, Bogdan Kustowski, Steven Langer, Brian Spears, Jayaraman Thiagarajan, Brian Van Essen, and Jae-Seung Yeom. Enabling machine learning-ready HPC ensembles with Merlin. *Future Generation Computer Systems*, 131:255–268, 2022.

Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Xiaojiang Chen, and Xin Wang. A comprehensive survey of neural architecture search: Challenges and solutions. *ACM Computing Surveys (CSUR)*, 54(4):1–34, 2021.

Petra Schneider, W Patrick Walters, Alleyn T Plowright, Norman Sieroka, Jennifer Listgarten, Robert A Goodnow, Jasmin Fisher, Johanna M Jansen, José S Duca, Thomas S Rush, et al. Rethinking drug design in the artificial intelligence era. *Nature Reviews Drug Discovery*, 19(5): 353–364, 2020.

Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando De Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1): 148–175, 2015.

Chengcheng Wang, XP Tan, SB Tor, and CS Lim. Machine learning in additive manufacturing: State-of-the-art and perspectives. *Additive Manufacturing*, 36:101538, 2020.

Andrew G Wilson and Pavel Izmailov. Bayesian deep learning and a probabilistic perspective of generalization. *Advances in neural information processing systems*, 33:4697–4708, 2020.