ACTIVE LEARNING OF DEEP SURROGATES FOR PDES

Anonymous authors
Paper under double-blind review

ABSTRACT

Surrogate models for partial-differential equations are widely used in the design of metamaterials to rapidly evaluate the behavior of composable components. However, the training cost of accurate surrogates by machine learning can rapidly increase with the number of variables. We present an active learning algorithm and apply it to train deep surrogates of Helmholtz's equation and linear elasticity in solid mechanics. For the two problems of interest, our algorithm reduces the number of simulations required compared to uniform random samples by more than an order of magnitude for a neural-network surrogate model and by four, respectively. Results show that the surrogate evaluation is faster than a direct solve by over two orders of magnitude and over five orders of magnitude, respectively.

1 INTRODUCTION

We present an active-learning (AL) approach—in which training points are selected based on an error measure—that can reduce the number of training points significantly for a neural-network (NN) surrogate model of partial-differential equations (PDEs). We applied the algorithm to train deep surrogates of Helmholtz's equation and linear elasticity in solid mechanics and demonstrate a reduction of the number of simulations required compared to uniform random samples by more than an order of magnitude for a neural-network surrogate model and by four, respectively. Further, we show how such a surrogate can be exploited to speed up simulations by $> 10^2 \times$ and $> 10^5 \times$, respectively. This significant speedup is particularly useful in the context of inverse design of metamaterials (Pestourie et al., 2020). When the structure is difficult to parameterize, inverse design can also be achieved via physics-informed neural networks (Lu et al., 2021), where the model learns both the design geometry and the solution to the PDE.

Active learning (AL) is connected with the field of uncertainty quantification (UQ), because AL consists of adding the most uncertain points to training set in an iterative way Fig. 1) and hence it requires a measure of uncertainty. Our approach to UQ is based on the NN-ensemble idea of Ref. 5 due to its simplicity. There are many other approaches for UQ (Tagasovska & Lopez-Paz, 2019, Sec. 5), but Ref. 5 demonstrated performance and scalability advantages of the NN-ensemble approach. This approach is an instance of Bayesian deep learning (Tagasovska & Lopez-Paz, 2019). In contrast, Bayesian optimization relies on Gaussian processes that scale poorly ($\sim N^3$ where N is the number of training samples) (Lookman et al., 2019; Bassman et al., 2018). The work presented here achieves training time efficiency (we show an order of magnitude reduction sample complexity), evaluation time efficiency (the actively learned surrogate model is at least two orders of magnitude faster than solving Maxwell's equations).

2 RESULTS

2.1 SURROGATE MODELS

Surrogate models for optics and mechanics For problems that have a parameterized geometry $\mathbf{p}$ and a low dimensional solution $t(\mathbf{p})$, training a surrogate model can dramatically speed the evaluations of the solution at run time compared to solving the equations numerically. When training a surrogate models, we want to fit $t(\mathbf{p})$ to a surrogate model

$$\tilde{t}(\mathbf{p}) \approx t(\mathbf{p}) \tag{1}$$

where we have considered two PDEs: Helmholtz's equation in optics and linear elasticity equations in solid mechanics, for which the quantity of interest are the complex transmission and the Young's modulus, respectively.

In this paper, the surrogate model for each of the real and imaginary parts of the complex transmission is an ensemble of $J = 5$ independent neural networks (NNs) with the same training data but different random batches (Goodfellow et al., 2016) on each training step. Each of NN i is trained to output a prediction $\mu_i(\mathbf{p})$ and an error estimate $\sigma_i(\mathbf{p})$ for every set of parameters $\mathbf{p}$. To obtain these μ_i and σ_i from training data $y(\mathbf{p})$ (from brute-force offline Maxwell solves) we minimize (Lakshminarayanan et al., 2017):

$$-\sum_{\mathbf{p}} \log p_{\mathbf{\Theta}_i}(y|\mathbf{p}) \propto \sum_{\mathbf{p}} \left[\log \sigma_i(\mathbf{p}) + \frac{(y(\mathbf{p}) - \mu_i(\mathbf{p}))^2}{2\sigma_i(\mathbf{p})^2} \right] \tag{2}$$

over the parameters $\mathbf{\Theta}_i$ of NN i. Eq. 2 is motivated by problems in which y was sampled from a Gaussian distribution for each $\mathbf{p}$, in which case μ_i and σ_i^2 could be interpreted as mean and hetero-skedastic variance, respectively (Lakshminarayanan et al., 2017). Given this ensemble of J NNs, the final prediction μ_* (for the real or imaginary part of $t(\mathbf{p})$) and its associated error estimate σ_* are the pooled mean and variance.

2.2 ACTIVE-LEARNING ALGORITHM

We present an online algorithm to choose training points that is significantly better at reducing the error than choosing points from a random uniform distribution. As described below, we select the training points where the estimated model error is largest, given the estimated error σ_*.

The online algorithm used to train each of the real and imaginary parts is outlined in Fig. 1. Initially we choose n_{init} uniformly distributed random points $\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_{n_{\text{init}}}$ to train a first iteration $\tilde{t}^0(\mathbf{p})$ over 50 epochs (Goodfellow et al., 2016). Then, given the model at iteration i, we evaluate $\tilde{t}^i(\mathbf{p})$ (which is orders of magnitude faster than the Maxwell solver) at $M \times K$ points sampled uniformly at random and choose the K points that correspond to the largest σ_*^2. We perform the expensive PDE solves only for these K points, and add the newly labeled data to the training set. We train $\tilde{t}^{i+1}(\mathbf{p})$ with the newly expended training set, using $\tilde{t}^i$ as a warm start. We repeat this process T times.

Essentially, the method works because the error estimate σ_* is updated every time the model is retrained with an expended dataset. In this way, model tells us where it does poorly by setting a large σ_* for parameters $\mathbf{p}$ where the estimation would be bad in order to minimize Eq. 2.

2.3 ACTIVE-LEARNING RESULTS

For Helmholtz's equation, we considered the parameterized geometries of a multilayer optical structure with ten holes of air of independent widths embedded in Silica Fig. 2A(inset). We compute the complex transmission with periodic boundary conditions for the left and right side of the computational domain with period 400 nm, and a perfectly matched layer for the bottom and top. We trained a surrogate model with an ensemble of five hetero-skedastic Gaussian models to predict the transmission for red green and blue lights. We compared the fractional errors of a NN surrogate model trained using uniform random samples with an identical NN trained using an active-learning approach. With the notation of our algorithm in Fig. 1, the baseline corresponds to $T = 0$, and n_{init} equal to the total number of training points. This corresponds to no active learning at all, because the n_{init} points are chosen from a random uniform distribution. In the case of active learning, $n_{\text{init}} = 2000$, $M = 4$, and we computed for $K = 500, 1000, 2000, 4000, 8000, 16000, 32000,$ $64000,$ and 128000. Although three orders of magnitude on the log-log plot is too small to determine if the apparent linearity indicates a power law, Fig. 2 shows that the lower the desired fractional error, the greater the reduction in training cost compared to the baseline algorithm; the slope of the active-learning fractional error (-0.2) is about 30% steeper that that of baseline (-0.15). The active-learning algorithm achieves a reasonable fractional error of 0.07 in twelve times less points than the baseline, which corresponds to more than one order of magnitude saving in training data (much less expensive Maxwell solves). This advantage would presumably increase for a lower error tolerance, though computational costs prohibited us from collecting orders of magnitude more training data to explore this in detail. For comparison and completeness, Fig. 2 shows fractional errors using
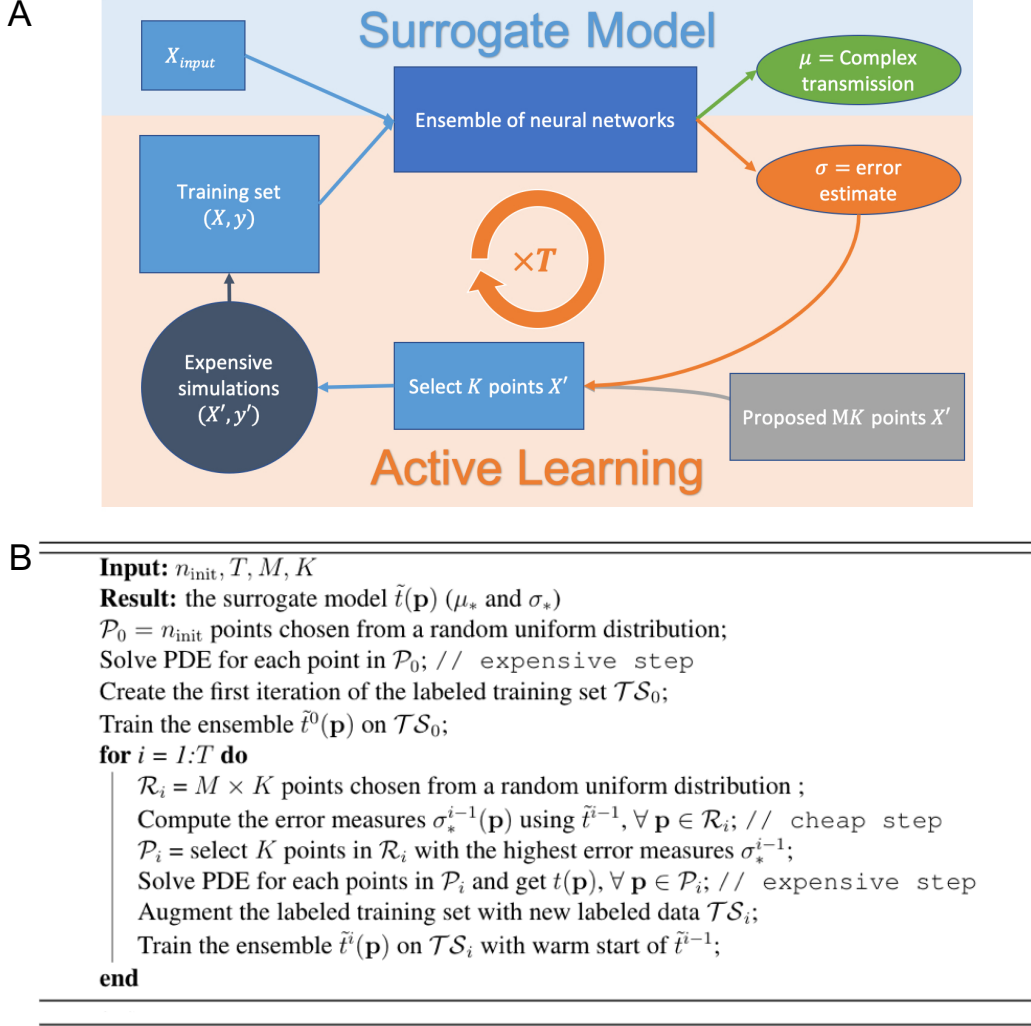
Figure 1: **Active-learning algorithm and surrogate model.** A) Diagram of the surrogate model (blue background), and the active-learning algorithm (orange background), the circle arrow signifies that the algorithm iterates T times. The fast evaluation of the surrogate is used both to create predictions of the surrogate model, and to compute the error measure that selects the points to add to the training set; B) pseudo-code of the algorithm.

Chebyshev interpolation (for the blue frequency only). Chebyshev interpolation has a much worse fractional error for a similar number of training points. Chebyshev interpolation suffers from the curse of dimensionality—the number of training points is exponential with the number of variables. The two fractional errors shown are for three and four interpolation points in each of the dimensions, respectively. In contrast, NNs are known to mitigate the curse of dimensionality (Cheridito et al., 2019).

For the solid mechanics application, we consider the PDEs of linear elasticity, consisting in the combination of equilibrium, constitutive and compatibility equations. The material is isotropic, fully characterised by both its Poisson's ratio $\nu = 0.28$ and Young's modulus $E = 1$. We consider a 2D square of length 10 m with a geometry parameterized by 4 circular pores (i.e holes in the material, as in Fig. 2B(inset)) whose centers are fixed, but with varying radii picked uniformly in the interval $[0.2; 2.0]$. We apply boundary conditions in order to obtain rollers on the left, bottom and top sides, while a Dirichlet boundary condition on the right side stretches the material by 10%. We run finite elements simulations to obtain the displacement and stress fields within the whole

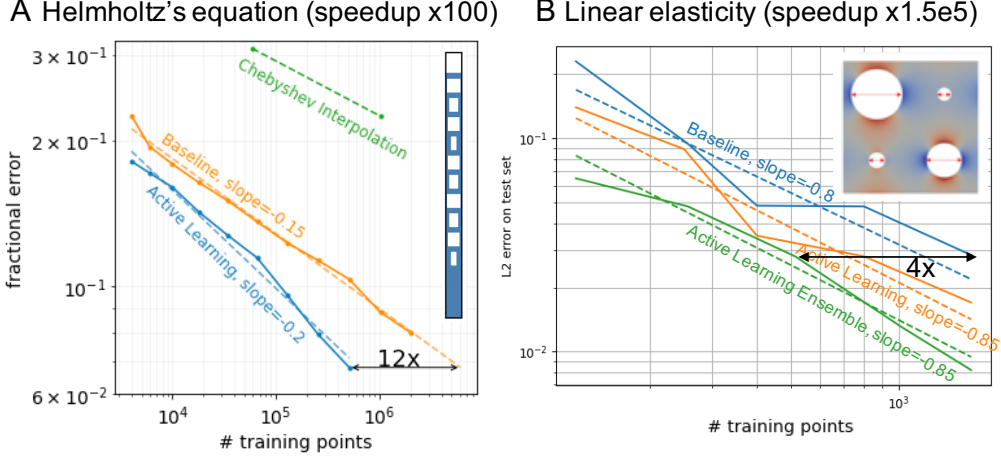A Helmholtz's equation (speedup x100)   B Linear elasticity (speedup x1.5e5)

Figure 2: **Active learning reduces the number of training points needed for a given accuracy. The corresponding surrogate models evaluate two orders of magnitude faster than the numerical solvers** A) when computing the complex transmission through a parameterized unit cell with 10 parameters (inset), the actively learned ensemble surrogate model achieved the same accuracy as the baseline with twelve times less points, and evaluates a hundred times faster than solving the equation numerically. B) when computing the effective Young's modulus in linear elasticity of a parameterized geometry (inset), the actively learned ensemble surrogate model achieved the same accuracy as the baseline with four times less points, and evaluated the Young's modulus around 1.5×10^5 faster than the finite elements simulations.

domain, and easily calculate the effective Young's modulus of the porous structure as the average ratio of stress over strain in the direction of the stretching. The Young's modulus being a proxy for the rigidity of structures, it is expected that structures with larger holes will yield to lower effective Young's moduli. Note that the Young's modulus parameter is a material characteristic independent of the geometry, we refer here to the *effective* Young's modulus of a structure which depends on its geometry. We train an ensemble of five hetero-skedastic Gaussian models predicting the effective Young's modulus for a given set of 4 pores radii. We train the network with a number of training points $n_{training} = 128, 256, 512, 1024, 1600$. The baseline is obtained with $T = 0$ where all the points of the training set are picked randomly, and the active learning model is run with $T = 3$, $M = 4$, and $n_{init} = K = \frac{n_{training}}{4}$. To assess the performance of the method, Fig. 2B shows that a given fractionnal error can be obtained with 4 times less training points, substantially cutting the cost of expensive PDEs solvers required to obtain the labeled data.

3 DISCUSSION

In this paper, we present an active-learning algorithm for composite materials which reduces the training time of the surrogate model for a physical response significantly. The simulation time is reduced by at least two orders of magnitude using the surrogate model compared to solving the partial differential equations numerically. We showed results for the scalar Helmholtz's equation (electromagnetism) and the governing equations of linear elasticity (solid mechanics).

We used an ensemble of NNs for interpolation in a regime that is seldom considered in the machine-learning literature—machine learning models are mostly trained from noisy measurements, whereas here the data are obtained from smooth functions. In this regime, it would be instructive to have a deeper understanding of the relationship between NNs and traditional approximation theory (e.g. with polynomials and rational functions (Boyd, 2001; Trefethen, 2019)). In the future, we plan to benchmark the active learning algorithm against diverse state-of-the-art uncertainty quantification techniques. We believe that the method presented in this paper will greatly extend the reach of surrogate-model based optimization of composite materials and other applications requiring moderate-accuracy high-dimensional interpolation.

REFERENCES

Lindsay Bassman, Pankaj Rajak, Rajiv K Kalia, Aiichiro Nakano, Fei Sha, Jifeng Sun, David J Singh, Muratahan Aykol, Patrick Huck, Kristin Persson, et al. Active learning for accelerated design of layered materials. *NPJ Comput. Mater.*, 4(1):1–9, 2018.

John P Boyd. *Chebyshev and Fourier Spectral Methods.* Dover Publications, Inc., Mineola, 2nd edition, 2001.

Patrick Cheridito, Arnulf Jentzen, and Florian Rossmannek. Efficient approximation of high-dimensional functions with deep neural networks. pp. Preprint at https://arxiv.org/abs/1912.04310, 2019.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning.* MIT Press, Cambridge, 2016.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Proc. 31st Int. Conf. Advances in Neural Information Processing Systems*, pp. 6402–6413. NIPS, 2017.

Turab Lookman, Prasanna V Balachandran, Dezhen Xue, and Ruihao Yuan. Active learning in materials science with emphasis on adaptive sampling using uncertainties for targeted design. *NPJ Comput. Mater.*, 5(1):1–17, 2019.

Lu Lu, Raphael Pestourie, Wenjie Yao, Zhicheng Wang, Francesc Verdugo, and Steven G Johnson. Physics-informed neural networks with hard constraints for inverse design. *arXiv preprint arXiv:2102.04626*, 2021.

Raphaël Pestourie, Youssef Mroueh, Thanh V Nguyen, Payel Das, and Steven G Johnson. Active learning of deep surrogates for pdes: Application to metasurface design. *npj Computational Materials*, 6(1):1–7, 2020.

Natasa Tagasovska and David Lopez-Paz. Single-model uncertainties for deep learning. In *Proc. 33rd Int. Conf. Advances in Neural Information Processing Systems*, pp. 6417–6428. NIPS, 2019.

Lloyd N Trefethen. *Approximation Theory and Approximation Practice.* SIAM, Philadelphia, extended edition, 2019.

A DEFINITION OF POOLED MEAN AND VARIANCE

Given this ensemble of J NNs, the final prediction μ_* (for the real or imaginary part of $t(\mathbf{p})$) and its associated error estimate σ_* are combined as (Lakshminarayanan et al., 2017):

$$\mu_*(\mathbf{p}) = \frac{1}{J} \sum_{i=1}^{J} \mu_i(\mathbf{p}) \tag{3}$$

$$\sigma_*^2(\mathbf{p}) = \frac{1}{J} \sum_{i=1}^{J} \left(\sigma_i^2(\mathbf{p}) + \left(\mu_i^2(\mathbf{p}) - \mu_*^2(\mathbf{p}) \right) \right) . \tag{4}$$

B FRACTIONAL ERROR

The fractional error FE between two vectors of complex values $\mathbf{u}_{\text{estimate}}$ and $\mathbf{v}_{\text{true}}$ is

$$FE = \frac{|\mathbf{u}_{\text{estimate}} - \mathbf{v}_{\text{true}}|}{|\mathbf{v}_{\text{true}}|} \tag{5}$$

where $|\cdot|$ is the L2-norm for complex vectors.