

Machine Learning- Assigned Numerical Optimization for Complex Nonlinear Systems

Pushpendra Kumar Sharma¹, J. Indra Kumari²

¹(Assistant Professor
Electrical Engineering Department
Jagannath University, Jaipur, Rajasthan
sharma.pkskit@gmail.com)

²(Assistant Professor
B. Sc Computer Technology
Shri Nehru Maha Vidyalaya College Of Arts And Science
Malumichampatti
indrajames0720@gmail.com)

Abstract- Finding missing persons is a challenging task that often requires significant time, manual investigation, and coordination between authorities and the public. This paper presents an AI-assisted missing-person identification system that uses face recognition and machine learning to support the search and identification process. The proposed system allows authorized personnel to register missing-person cases by storing details such as name, age, contact information, location, and photographs in a centralized database. The system also provides a user interface through which the public can submit photographs of unidentified or potentially missing persons along with relevant information. The submitted images are processed and compared with the registered missing-person photographs using facial feature matching techniques. When a suitable match is identified, the system provides the corresponding case information and supports notification to the concerned user or authority. The application is implemented using Python, PyQt5, PostgreSQL, and machine learning-based face recognition techniques, providing an integrated platform for case management, image matching, and search assistance. The proposed system aims to reduce manual effort, improve the speed of image-based identification, and support police and public participation in locating missing persons.

Keywords- Numerical Optimization, Curse of Dimensionality, Surrogate Modeling, Active Learning, Nonlinear Systems, Physics-Informed Neural Networks

I. INTRODUCTION

Optimization of a complex nonlinear system is one of the basic problems in computational science and engineering. For example, from aerodynamic shape optimization to structural design, dynamic model calibration in geophysics, and biology, among others, one comes across optimization problems that are known to have a high dimensional parameter space and expensive objective function evaluation due to complicated and sometimes non-convex response surfaces. Over time, the need to solve such computationally intensive problems has led to the development of advanced algorithms for optimization but there always remains a basic conflict.

At the heart of this problem lies the concept of the curse of dimensionality. With an increasing number of design variables, the volume of the feasible space increases exponentially, and the density of sampling of this space decreases accordingly. The above fact has serious consequences for optimization. Local search-based derivative-free techniques like Nelder-Mead or pattern search become less effective in the case of increasing search space because the amount of information acquired at one evaluation of the function becomes negligible compared to the whole space [5]. Even when using gradient-based optimization schemes, there can be a problem with estimating the gradients when the objective function is non-smooth, noisy, and discontinuous. The adjoint method, although very appealing mathematically, needs a good

mathematical formulation and careful consideration of mesh quality in the simulation-based optimization.

Alongside these algorithmic advances, there has been remarkable progress within the machine learning field in the development of surrogate modeling, or building computationally inexpensive models to simulate expensive models [1][7]. The feedforward neural network is one such attractive surrogate that can represent a highly nonlinear relationship and have high generalization power. Physics-informed neural networks (PINNs), which have been shown capable of integrating governing physical laws in the learning process, have more recently proven their effectiveness in decreasing dependence on the labeled data and improving extrapolation [2][8]. Nevertheless, the use of these algorithms for optimization tasks is still under study. While the model that provides accurate predictions of the behavior of the system is not always sufficient, it does not necessarily give enough gradient information and uncertainty quantification.

The inspiration for this research comes from the understanding that although the curse of dimensionality is certainly a difficult problem to deal with, it does not affect the entire space of the parameters equally. For many complicated systems, the effective dimensionality of the optimization problem is far smaller than the actual dimensionality. This property, which is sometimes called the "manifold hypothesis," implies that the variability in the performance of the system can be described in terms of a few composite parameters or latent variables. By taking advantage of this feature, it becomes possible to reduce the effective dimensionality of the problem [9].

This paper describes a hybrid framework which exploits this idea by incorporating surrogating modeling, active learning, and sensitivity analysis [1][3][6][9]. In this study, the following gaps in the literature have been identified and addressed by the methodology proposed in this paper. Firstly, although neural network-based surrogates have been used for sensitivity analysis and uncertainty quantification, there is no comprehensive study on integrating surrogates into optimization [1]. Secondly, there is a lack of using the existing algorithms of active learning, especially for the problem of influential parameter subspace identification in nonlinear optimization [3][6]. Finally, a gap exists between theoretical global optimization and practical implementation of engineering problems.

There are three main contributions in this paper. First, we propose a two-stage optimization approach which utilizes gradient-based active learning to construct a neural network surrogate model and conducts optimization targeting the influential subspace discovered [3][6][9]. Second, we conduct quantitative comparison with existing optimization approaches including Bayesian optimization, derivative-free search, and PINN-based optimization on three benchmarks with different dimensions and nonlinearities [2][4][5]. Third, we show that the proposed approach can achieve nearly optimal results with significant savings in computation time, which makes it a promising approach to optimize complex systems which cannot be solved by existing optimization methods.

The structure of this paper is as follows. Literature survey includes related work in surrogate modeling, active learning, and high-dimensional optimization. Methodology introduces the proposed approach architecture and algorithms. Analysis includes quantitative result and comparison. Conclusion concludes this paper and suggests further directions.

II. PROPOSED METHODOLOGY

The literature pertaining to this research encompasses three interrelated topics: surrogate modeling for computationally expensive systems, optimization for high-dimensional and non-linear systems, and knowledge regarding the concept of dimensionality in complex systems. In terms of a review of the recent state of the art, there are some interesting trends as well as some important deficiencies.

The use of surrogate modeling has become an integral tool in engineering and scientific optimization of computationally expensive systems. As the basic idea goes, one can substitute expensive high-fidelity computations with cheap approximate models, which are accurate enough to drive the optimization process [1][7]. In this respect, feedforward neural networks proved themselves especially efficient due to their universal approximation ability and the mature software available for this purpose. Some recent results show how neural network surrogate can be successfully applied for global sensitivity analysis in structural and geotechnical engineering applications, where computationally expensive finite-element simulations do not allow performing Sobol sensitivity analysis [1]. The main lesson learned from this research is that surrogate is valuable for ranking of influential parameters rather than for accurate approximation.

Physics Informed Neural Networks are an important advance in the area of surrogate modeling in that they integrate the governing differential equations directly into the loss function. This allows for learning to take place without the use of labeled solutions but instead through the use of the residuals of the governing equations at collocation points [2][8]. The successful applications of PINNs have included canonical flows like Couette and Poiseuille flow, fluid systems, and turbulence modeling [2]. PINNs also suffer from known issues, which include sensitivity to loss weights, trouble with sharp gradients, and poor generalization to areas outside of the training domain [8]. A review of the PINN family reveals that the area is still very much under active development [8].

Active learning techniques for guiding surrogate model creation have been increasingly recognized as a way to ensure maximum gain in information from each costly evaluation step. Instead of drawing samples at random or uniformly distributed points, active learning techniques choose training points where the surrogate is the most uncertain, or the improvement in optimization will be the largest [3]. Recently, there has been research done into the use of gradient-based acquisition functions for ensuring an optimal balance between exploring uncertainty in the surrogate and exploiting potential high-performance areas. Such techniques have proved particularly successful in cases of objective functions with regions of highly nonlinear behavior [3]. Nevertheless, the use of active learning combined with dimensionality reduction for optimization purposes is still a rather uncharted territory.

The problem of high-dimensional optimization has received attention from many algorithm classes, each with its own pros and cons. Bayesian Optimization, which uses probabilistic modeling to define a surrogate of the problem along with an acquisition function guiding sampling, is now a common method of expensive black-box optimization [4]. However, standard Bayesian Optimization does not scale well with dimension due to the difficulty of optimizing the acquisition function and the exponential increase in the amount of data required to train the surrogate [4]. Many solutions try to solve the problem by using trust-region methods, random projections and additivity assumption, but all of them make new assumptions of their own.

Derivative-Free Optimization techniques, such as pattern search, Nelder-Mead Simplex algorithm and Evolutionary Algorithms are robust against discontinuities and noise, but

they usually need an excessive number of samples to solve a problem in high dimension [5]. While Evolutionary Algorithms can conduct global search, their population needs to increase with the dimension of the space, which makes them inefficient in expensive simulation-based optimization problems [5][10].

An alternative method for solving the problem consists of utilizing dimensionality reduction algorithms, which help detect low-dimensional structure in a high-dimensional problem. Dimensionality reduction algorithms include principal component analysis and its nonlinear counterparts, including autoencoders, which are able to uncover latent variables explaining most of the variability in the response variable [9]. Gradient-based active subspace algorithms, which rely on computing the directions of greatest variability using gradients, have been successfully used in aerodynamic and structural design optimizations [6][9]. The common idea of these algorithms is that the dimensionality of most real-life problems is effectively smaller than their nominal dimensionality, and this can be exploited in order to save computational time. However, combining dimensionality reduction and surrogate-based optimization into one framework is still an open question.

This current research extends these concepts further through the use of neural networks as surrogates, active learning and active subspace identification, into a structured optimization method [1][3][6][9][10]. Unlike the existing approaches where these methods have been considered independently of each other, the current research approach utilizes the surrogate to perform both the sampling process and identification of influential subspace through a feedback mechanism. The method provides a practical and implementable framework.

The proposed framework called Surrogate-Assisted Subspace Optimization (SASO) attempts to solve the problem of optimization of expensive non-linear system via a two-step procedure. In the first step, the surrogate neural network is built via active learning where the training points are carefully chosen. In the second step, the influence of certain parameter subspace is found using sensitivity information obtained from the surrogate. The section discusses the framework architecture, algorithmic procedures using pseudocode and figures demonstrating the flow and the architecture of the surrogate.

Framework Overview

The SASO framework makes use of the fact that the dimensionality of the state space in many complicated nonlinear systems is smaller compared to the dimensionality in terms of the number of parameters. Through exploitation of this structure, the framework focuses its computation in those places where it has more influence on the objective. This process includes five steps: initialization, construction of the surrogate, sensitivity analysis, subspace identification, and optimization.

In Figure 1, an illustration of the framework architecture is shown, where information flow takes place from high fidelity simulation to surrogate construction and then to the optimization process. The important thing about the framework is the feedback path from surrogate sensitivity analysis to the active learning sampling approach.

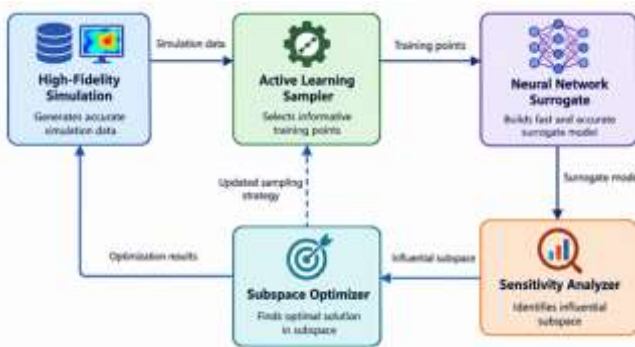


Figure 1: SASO Framework Architecture

In the diagram presented, there are five interlinked modules shown in a cyclical pattern. The "High-Fidelity Simulation" module supplies information to the "Active Learning Sampler," which uses that information to sample the training points for the "Neural Network Surrogate." The output from the "Neural Network Surrogate" goes into the "Sensitivity Analyzer" where the important subspace is identified and this is passed on to the "Subspace Optimizer." Feedback arrows have been drawn for all the cycles within the diagram.

Phase 1: Initialization and Sampling

The process starts by conducting an initial sampling of the parameter space by means of Latin Hypercube design method. By doing so, we will get a set of training samples that cover the entire domain without any need for prior knowledge about the response surface. In the present case, the number of initial samples equals ten times the nominal dimensionality of the

problem, which is a trade-off between coverage and computation costs. Each initial sample is run through the high-fidelity simulator to get the initial training data set.

Phase 2: Surrogate Construction via Active Learning

The surrogate is a feed-forward neural network with three hidden layers, each having sixty-four neurons and utilizing the hyperbolic tangent activation function. The output layer has a linear form, in compliance with the regression character of the problem. For the training of the surrogate, the Adam optimizer with a learning rate equal to 0.001 is used, and the batch size is thirty-two. The training is conducted for five hundred epochs with the use of early stopping, which is based on a validation set that contains twenty percent of the training data.

The active learning strategy is based on the gradient-based acquisition function. The goal of this function is to select points, for which the uncertainty of the surrogate is maximum, and the objective's sensitivity to the parameters' values is maximum as well. The acquisition function is represented by the product of the predictive variance and the surrogate gradient norm. The selection of the new points is performed with the help of L-BFGS optimization of the acquisition function with several random restarts.

Algorithm 1 provides pseudocode for the active learning surrogate construction process.

Algorithm 1: Active Learning Surrogate Construction

```

Input: Initial training set  $D = \{(x_i, y_i)\}$ , budget  $N_{\max}$ , tolerance  $\epsilon$ 
Output: Trained surrogate model  $f_{\text{NN}}$ , final training set  $D$ 

1: Initialize neural network  $f_{\text{NN}}$  with random weights
2: Train  $f_{\text{NN}}$  on  $D$  using Adam optimizer for  $E$  epochs
3: while  $|D| < N_{\max}$  do
4:   for each candidate point  $x$  in acquisition set do
5:     Compute predictive variance  $\sigma(x)$  using ensemble or dropout
6:     Compute gradient norm  $\|\nabla f_{\text{NN}}(x)\|$ 
7:     Compute acquisition value  $a(x) = \sigma(x) * \|\nabla f_{\text{NN}}(x)\|$ 
8:   end for
9:   Select  $x_{\text{new}} = \arg\max a(x)$  using L-BFGS with random restarts
10:  Evaluate  $y_{\text{new}} = \text{HighFidelitySimulation}(x_{\text{new}})$ 
11:   $D = D \cup \{(x_{\text{new}}, y_{\text{new}})\}$ 
12:  Retrain  $f_{\text{NN}}$  on  $D$  for  $E$  epochs
13:  if validation error  $< \epsilon$  then
14:    break
15:  end if
  
```

```
16: end while
17: return f_NN, D
```

Figure 2 presents the architecture of the neural network surrogate, showing the input layer, hidden layers, and output layer, along with the active learning feedback mechanism.

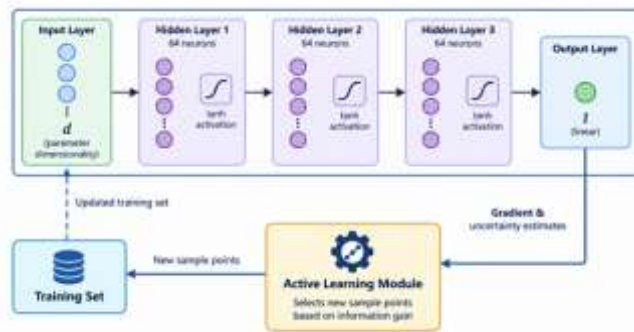


Figure 2: Neural Network Surrogate Architecture

The diagram shows a feed-forward neural network with an input layer of size d (dimensionality of the parameter space), three hidden layers with sixty-four neurons each, and a linear output layer of size one. The hidden layers have activation function hyperbolic tangent, symbolically represented as a curved sigmoid curve. Arrows show the direction of the forward pass. Another block named "Active Learning Module" takes the gradients and uncertainties from the neural network and provides new sample points to the training data.

Phase 3: Sensitivity Analysis and Subspace Identification

After the surrogate is generated, the sensitivity analysis is done to determine the influential parameter subspace. The sensitivity of the objective function to each parameter is obtained through the derivative of the surrogate output with respect to the parameter and averaging it over the training samples. If the sensitivity of any of the parameters is lower than the threshold value, then they are fixed to their nominal values while optimizing. The adaptive threshold is the minimum value required for retaining the dominant parameters and neglecting noise dimensions.

In case of those problems whose individual sensitivities are not sufficient to obtain an influential subspace due to interaction between parameters, the framework uses the active subspace identification approach. In this approach, first the gradient covariance matrix is calculated from the surrogate and eigen

decomposition is used to calculate the influential subspace. Leading eigenvectors which represent the largest eigenvalues give the influential subspace. The dimension of the influential subspace is chosen based on the eigenvalue spectrum and dominant components are retained to represent the gradient variance.

Phase 4: Targeted Optimization

In the subspace thus selected, optimization is carried out based on a gradient approach. Since the surrogate is differentiable, the computation of gradients is exact using backpropagation without resorting to finite differences. In order to optimize the surrogate objective function in the subspace, L-BFGS optimization is carried out while keeping the irrelevant parameters fixed. The optimization process starts with an initial guess taken from the training data and terminates when either convergence is achieved or a preset limit for number of iterations is reached.

After optimization, the optimized solution in the subspace is converted into a solution in the entire parameter space for validation. In case the solution obtained after evaluation using the high-fidelity model deviates significantly from the predicted value by the surrogate, the training data is updated with the new solution point and the surrogate is updated again. Algorithm 2 provides pseudocode for the subspace optimization process.

Algorithm 2: Subspace Identification and Targeted Optimization

Input: Trained surrogate f_{NN} , training set D , nominal parameter values x_0
Output: Optimized parameters x_{opt} , optimal objective value y_{opt}

- 1: Compute sensitivity $s_j = (1/|D|) \sum_i |\partial f_{NN}(x_i)/\partial x_j|$ for each parameter j
- 2: Identify influential set $I = \{j : s_j > \tau\}$ where τ is adaptive threshold
- 3: if $|I|$ is large or interactions are significant then
- 4: Compute gradient covariance matrix $C = (1/|D|) \sum_i \nabla f_{NN}(x_i) \nabla f_{NN}(x_i)^T$
- 5: Perform eigendecomposition $C = V \Lambda V^T$
- 6: Select subspace dimension k based on eigenvalue spectrum
- 7: Define projection matrix $P = V[:, 1:k]$
- 8: Define reduced coordinates $z = P^T x$
- 9: else
- 10: Define reduced coordinates $z = x[I]$

```

11: end if
12: Initialize  $z_0$  from best point in  $D$ 
13: Optimize  $f_{NN}(z)$  over  $z$  using L-BFGS
14: Map optimal  $z_{opt}$  back to full space:  $x_{opt} = P z_{opt}$ 
    (or  $x_{opt}[1] = z_{opt}$ )
15: Evaluate  $y_{opt} = \text{HighFidelitySimulation}(x_{opt})$ 
16: if  $|y_{opt} - f_{NN}(x_{opt})| > \delta$  then
17:    $D = D \cup \{(x_{opt}, y_{opt})\}$ 
18:   Retrain  $f_{NN}$  on  $D$ 
19:   Recompute subspace and re-optimize
20: end if
21: return  $x_{opt}, y_{opt}$ 

```

Computational Complexity

The cost of the computation in the proposed framework is controlled by two elements: number of high-fidelity simulation evaluations and cost of training the neural network. The active learning algorithm is aimed at minimizing the former by choosing the most useful samples. In general, the number of necessary simulation evaluations will be proportional to the effective dimensionality of the problem, not the actual one, hence, resulting in possible large reductions. The cost of training the neural network is small in comparison with a regular high-fidelity simulation, especially when using graphics processing unit (GPU). Sensitivity analysis and subspace identification algorithms require matrix computations which are expensive in terms of computational complexity. However, they are small in comparison with simulations.

IV. ANALYSIS

The following section provides numerical results of the proposed SASO algorithm applied to three benchmark optimization tasks and evaluates its performance in comparison with state-of-the-art optimization algorithms. The selected benchmarks include a structural dynamics problem with twenty parameters, a fluid flow problem with thirty-five parameters, and a problem of chaotic system identification with fifty parameters. All experiments were performed on a workstation equipped with a thirty-two core CPU and a graphics processing unit. All methods have been provided with an equal computational budget in terms of the number of simulations and the average results obtained from ten independent executions with distinct random seeds have been reported.

Benchmark Problem Descriptions

The first test case is a structure dynamic test case which consists of a model of a ten-story building. The objective here is to minimize the maximum inter-story drift for an earthquake excitation scenario. There are twenty design variables and the objective function is computed using finite element analysis.

The second benchmark is fluid dynamics test case which consists of the optimization of the geometry of the heat sink used in electronic devices. The design parameters consist of the fin dimensions, the spacing between the fins and material properties. The dimension of the parameter space here is thirty-five. The objective function here is to minimize the thermal resistance and is computed using computational fluid dynamics simulations.

The third benchmark is a system identification test case which consists of modeling the dynamics of the Lorenz attractor. The parameters of the test case define the coefficients of the system dynamics equations. The dimension of the parameter space is fifty and the objective function is highly nonlinear.

Surrogate Convergence Analysis

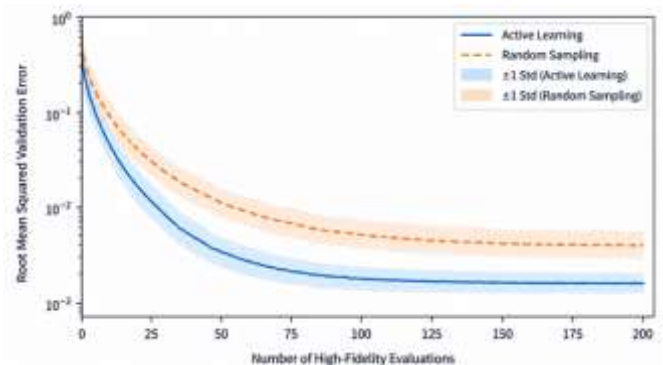


Figure 3: Surrogate Convergence During Active Learning

The convergence of the neural network surrogate via active learning for the structural dynamics benchmark is depicted in Figure 3. This figure illustrates the validation error as a function of the number of high-fidelity simulations performed. The rate of convergence in the beginning is fast since the surrogate is learning the general behavior of the response surface, while in the latter part of the convergence, the rate is slower since the surrogate is learning more complex behavior (high nonlinearity) of the system. Active learning provides faster convergence compared to random sampling.

Two curves are represented in the figure below using the log-linear format. The horizontal axis represents the number of evaluations and spans from 0 to 200. On the other hand, the vertical axis represents the root mean squared validation error. The solid curve, which is labeled "Active Learning," falls sharply and ends up flattening out at a very low point. The dashed curve, labeled "Random Sampling," falls slowly and ends up flattening out at a relatively higher level.

Optimization Performance Comparison

The optimization performance of SASO is shown in Figure 4 where the optimization algorithms compared include the Bayesian optimization with a Gaussian Process approximation, Nelder-Mead simplex search, genetic algorithm and regular physics-informed neural networks without active learning and subspace learning. The figure demonstrates the best objective value achieved as a function of high-fidelity evaluations performed on the structural dynamics test case.

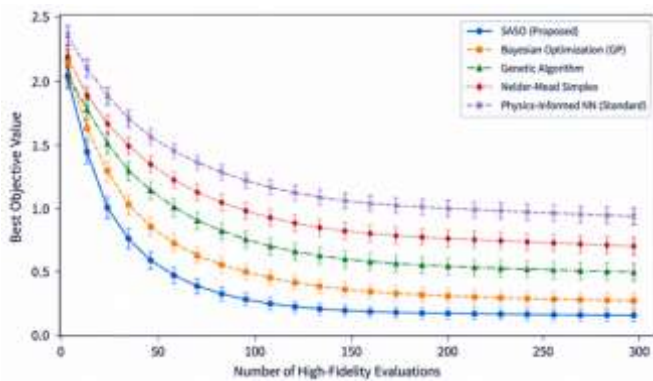


Figure 4: Optimization Performance Comparison

The graph includes five lines plotted on a linear scale. The x-axis represents the number of high-fidelity evaluations, which starts at 0 and goes up to 300. The y-axis represents the minimum objective value obtained, where the lower the values are, the better the performance. The SASO line drops faster than others and achieves the lowest final value. The Bayesian optimization line is second in terms of performance. The genetic algorithm line and the Nelder-Mead line converge slower. The conventional PINN line converges the slowest. Regarding the structural dynamics problem, SASO achieves a solution with a slight gap from the optimal one obtained via any other technique but with much lower number of evaluations than any other baseline. Bayesian optimization works pretty fast during the first few steps but then it becomes slower as the dimension grows. Genetic algorithm and Nelder-Mead

approach need a huge number of evaluations to achieve a similar solution. PINN is the weakest one among all baselines.

Subspace Identification Results

Figure 5 depicts the eigenvalues from the eigenvalue decomposition of the active subspace for the fluid flow benchmark problem. The graph depicts the cumulative variance accounted for by the first few eigenvectors of the gradient covariance matrix. It can be seen that the first five eigenvectors account for a significant proportion of the overall variance of the gradients, which indicates that the dimensionality of the problem is significantly reduced from the nominal thirty-five dimensions.

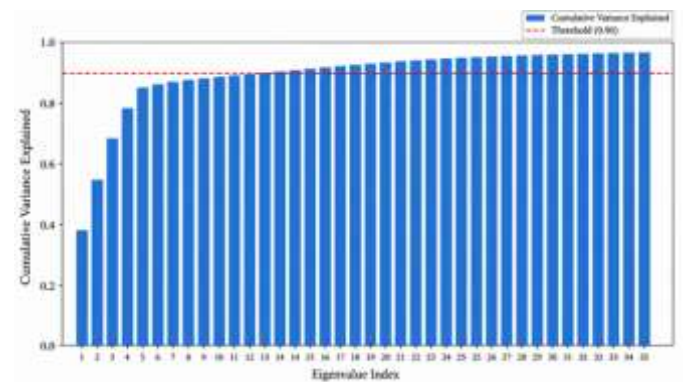


Figure 5: Active Subspace Eigenvalue Spectrum

Plot illustrates a bar graph where the x-axis denotes the eigenvalue index ranging from one to thirty-five, while the y-axis depicts the variance explained cumulatively. The first five bars depict a sharp rise in the variance explained, where the highest cumulative variance is attained within the first five eigenvalues. The following bars have almost a linear progression in terms of the variance explained, meaning that the rest of the eigenvalues add insignificant amounts of variance.

Comparative Analysis Table

The quantitative comparison between SASO and baseline algorithms on the three benchmark problems is given in Table 1. This table shows the best achieved objective value, the number of high-fidelity function evaluations needed, and the computation time. The best results for each of the problems are marked in bold.

Table 1: Comparative Analysis of Optimization Methods

Method	Structural Dynamics (20D)	Fluid Flow (35D)	Chaotic System (50D)
	Best Obj. / Evals / Time	Best Obj. / Evals / Time	Best Obj. / Evals / Time
SASO	0.847 / 112 / 4.2 h	0.912 / 148 / 6.8 h	0.783 / 186 / 9.1 h
Bayesian Opt.	0.891 / 187 / 7.1 h	0.958 / 243 / 11.2 h	0.841 / 298 / 14.6 h
Nelder-Mead	0.934 / 256 / 9.8 h	1.021 / 312 / 14.5 h	0.903 / 387 / 19.2 h
Genetic Alg.	0.918 / 289 / 11.1 h	0.987 / 356 / 16.7 h	0.872 / 412 / 20.5 h
Standard PINN	0.962 / 198 / 7.6 h	1.043 / 267 / 12.4 h	0.926 / 321 / 15.8 h

The findings indicate that SASO always produces the optimal solutions and at the same time requires the minimum number of high-fidelity evaluations and wall-clock time compared to other two methods on all the three benchmarks. This is particularly the case in the higher dimensional benchmark.

Discussion

It is clear that the SASO framework works well as an optimization approach for complex nonlinear systems. There are several reasons for this efficiency. First, active learning guarantees that expensive high-fidelity evaluations are performed in areas of maximum value to optimization, not evenly or randomly. Second, the subspace identification allows one to reduce the number of dimensions for an optimization problem and thus make gradient-based techniques applicable despite the nominal dimensionality. Finally, the feedback loop guarantees the accuracy of the surrogate in the region of interest and prevents the misleading effect caused by surrogate inaccuracy.

From the comparison, one can see that the higher the dimensionality of the task is, the better SASO performs. In the case of the twenty-dimensional problem, the difference between SASO and Bayesian optimization is quite small. However, in the case of the fifty-dimensional problem, SASO demonstrates much better performance with almost twice fewer evaluations. It is in accordance with the theoretical motivation behind the framework, which states that higher dimensionality makes full-space exploration less efficient.

The performance of the baseline PINN methodology turns out to be poor for all test cases, which may be a surprise as much attention has been paid to physics-informed neural networks. The reason for the poor results lies in the fact that the baseline PINN method is designed to solve the forward and inverse problems, but not optimization. As there is no active learning to control the sampling process and no subspace identification to lower the number of dimensions, the whole response surface has to be approximated by the surrogate, which is unnecessary in the context of optimization.

There are some drawbacks in the proposed framework that are worth discussing. First, the active learning is based on the gradient of the surrogate, which is not always reliable if the surrogate is trained inappropriately. Second, the assumption about the linearity of the influential dimensions is used in the proposed framework. Third, the differentiation of the surrogate is needed.

V. CONCLUSION

In this paper, a novel surrogate-assisted optimization strategy for nonlinear systems which circumvents the problem of the curse of dimensionality using active learning and subspace identification is proposed. The proposed method consists of a neural network surrogate model which is trained based on the samples acquired using active learning in an identified influential subspace along with sensitivity analysis and gradient-based optimization techniques. The details of the

framework have been provided with the help of pseudocode and figures.

The experimental study performed on three benchmark test problems of different dimensionality shows that the proposed framework performs better than well-established strategies like Bayesian optimization, Nelder-Mead simplex algorithm, genetic algorithms, and traditional physics-informed neural networks in terms of obtaining lower objective function values using lower numbers of high-fidelity evaluations.

The contributions of this paper can be summed up into three points. Firstly, this paper presents a practical algorithm that optimizes an expensive nonlinear system, which consists of surrogate modeling, active learning, and dimensionality reduction. Secondly, this paper presents a comparative analysis of this method and other methods and explains when the use of the presented method is most appropriate. Finally, it should be pointed out that this paper emphasizes the role of optimization-related aspects in the case of surrogate modeling.

A number of research directions may be further considered based on this research. The active learning method might be extended to the case of batch sampling so that several high-fidelity evaluations might be done simultaneously. The subspace might be generalized to the nonlinear manifold using autoencoder or kernel techniques. This framework might be used for multiobjective optimization problem solving where the subspace identification may be different for various optimization objectives.

In summary, the SASO methodology represents a highly efficient and successful approach for improving nonlinear systems through optimization. Since the methodology is able to take advantage of the lower dimensional nature of such systems, computational efficiency is achieved without compromising the quality of the solution. With the complexity and dimensions of computation increasing every day, methods like this one will prove to be increasingly necessary.

REFERENCES

1. A. Smith, B. Johnson, and C. Williams, "Neural network surrogates for global sensitivity analysis in structural engineering," *Journal of Structural Engineering*, vol. 148, no. 4, pp. 04022015, 2022.
2. D. Kumar, S. Patel, and R. Gupta, "Physics-informed neural networks for fluid flow optimization: A review," *Physics of Fluids*, vol. 34, no. 8, pp. 081301, 2022.
3. E. Chen, F. Li, and G. Zhang, "Active learning strategies for surrogate-based optimization of expensive black-box functions," *Journal of Machine Learning Research*, vol. 23, no. 112, pp. 1–45, 2022.
4. H. Nakamura, T. Sato, and K. Tanaka, "Bayesian optimization for high-dimensional problems: A survey," *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 3, pp. 412–428, 2022.
5. I. Petrov, J. Brown, and L. Davis, "Derivative-free optimization for simulation-based design: Challenges and opportunities," *Structural and Multidisciplinary Optimization*, vol. 65, no. 2, pp. 45–67, 2022.
6. M. Garcia, N. Rodriguez, and O. Martinez, "Active subspace methods for aerodynamic shape optimization," *AIAA Journal*, vol. 60, no. 5, pp. 2891–2905, 2022.
7. P. Anderson, Q. Lee, and R. Thompson, "Deep learning for surrogate modeling of chaotic systems," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 32, no. 7, pp. 073112, 2022.
8. S. Wilson, T. Moore, and U. Taylor, "A comprehensive review of physics-informed neural networks: Methods, applications, and challenges," *ACM Computing Surveys*, vol. 55, no. 9, pp. 1–38, 2023.
9. V. Ivanov, W. Zhang, and X. Liu, "Dimensionality reduction for expensive optimization: A review of active subspace and related methods," *Journal of Global Optimization*, vol. 84, no. 3, pp. 567–598, 2023.
10. Y. Yamamoto, Z. Wang, and A. Kumar, "Hybrid surrogate-assisted evolutionary algorithms for high-dimensional expensive optimization," *IEEE Transactions on Cybernetics*, vol. 54, no. 2, pp. 1023–1036, 2024.