Proceedings of the ASME 2022
International Design Engineering Technical Conferences and
Computers and Information in Engineering Conference
IDETC/CIE 2022
August 14-17, 2022, St. Louis, Missouri

DETC2022-90065

MEAN SQUARED ERROR MAY LEAD YOU ASTRAY WHEN OPTIMIZING YOUR INVERSE DESIGN METHODS

Shai Bernard
Dept. of Mechanical Engineering
University of Maryland
College Park, MD 20742
Email: sbernar2@umd.edu

Jun Wang
Dept. of Mechanical Engineering
University of Maryland
College Park, MD 20742
Email: jwang38@umd.edu

Mark Fuge[*]
Dept. of Mechanical Engineering
University of Maryland
College Park, MD 20742
Email: fuge@umd.edu

ABSTRACT

*When performing time-intensive optimization tasks, such as those in Topology Optimization or Shape Optimization, researchers have turned to Machine Learned (ML) Inverse Design methods—*i.e.*, algorithms that predict the optimized geometry from input conditions—to either replace or warm start traditional optimizers. Almost exclusively, such methods are trained and optimized to reduce the Mean Squared Error between a method's output and a ground truth training dataset of optimized designs, this being the obvious choice for traditional supervised learning. While convenient, we show that this choice may be myopic. Specifically, we compare two methods of optimizing the hyper-parameters of both a random forest (RF) and k-nearest neighbors (KNN) model for predicting the optimal topology in a 2D heat sink example.*

We show that under both direct Inverse Design as well as when warm starting further Topology Optimization (TO), using typical Mean Squared Error metrics produces less performant models than a proposed metric that directly evaluates the objective function, though both methods produce designs that are almost one order of magnitude better than a control condition that uses a uniform initialization common in TO. We also illustrate how warm starting TO with predicted solutions impacts both the convergence time, the type of solutions obtained during optimization, and the final designs. Sensitivity analyses on various model parameters demonstrate that the results are not dependent on other model hyperparameters. Overall, our initial results portend that researchers may need to revisit common choices for evaluating ID methods that subtly trade-off factors in how an ID method will actually be used. We hope our open-source dataset and evaluation environment will spur additional research in those directions

INTRODUCTION

Design optimization, such as in Topology Optimization (TO) or Shape Optimization, frequently requires expensive (in both time and computing resource) iterations to converge. For example, following the implementation of the governing equations and required parameters in a computational environment, TO problems typically require the iterative solving of these equations followed by updates to the problem conditions after each iteration. These computational expenses become significant, and sometimes prohibitive, in cases that require large numbers of calculations per iteration, large numbers of iterations, where computational resources are limited, or where a good solution is needed in a short duration of time.

In response, researchers have tried to circumvent this iterative process via *Inverse Design* (ID)—training a Machine Learning model to directly output an optimal design for a new problem, given a dataset of past (typically expensive) physics-based opti-

[*] Corresponding Author

mizations [1, 2]. In cases where such a dataset is available and one needs to evaluate many new input conditions or requirements quickly, ID methods can often provide significant time savings compared to optimizing a design for each bespoke input condition [1, 2].

But are ID methods any good? Such models are typically trained to minimize the Pointwise Mean Square Error (PMSE) of how well the ID model predicts the optimized geometry for the input condition. This standard choice results from formulating ID as a supervised learning problem—input conditions in and optimized designs out—and measuring the output's discrepancy with training or test samples. Researchers typically optimize any hyper-parameters of such models in similar fashion, prostrating the entire model upon the alter of the MSE.

However, an ID method's ultimate goal can differ from the above aim. Are we using the predictions to capture, as accurately as possible, the geometry or design itself? Or do we care just about outputting high performance designs, irrespective of how closely they match the training set? More importantly, are we using the predicted designs as-is, or using them to accelerate further optimization (*i.e.*, warm starting)? Does warm starting with ID methods actually help, and if so, how and when? Is Mean Squared Error always the best thing to optimize? This paper asks some of these questions and answers (tentatively and perhaps predictably) "it depends!" It casts doubt on MSE's monotheism and begs the impious question "what else might we optimize our ID models for?"

In this work, we attempt to demonstrate using relatively simple, fast, and easily reproducible Machine Learning models (Random Forests and k-Nearest Neighbors) how ID predictions impact the topology optimization of a 2D conductive heat sink governed by the Poisson Equation, subject to several parameters and boundary conditions, and optimized using interior point methods [3]. We examine multiple measures of ID performance, how the ID predictions modify the optimization process compared to a standard benchmark, and what, if any, effects altering the simple ML models has on our results. The overall contributions of this paper are:

1. We formulate an inverse design problem for the design of 2-dimensional heat sinks based on the problem described in [3] and shown in Fig. 1. This results in a dataset and ID evaluation environment that we make available for the research community, along with performance diagnostics that shed light onto how optimizers are affected by ID methods.
2. We compare the performance of k-nearest neighbors and random forest models on this inverse design problem across multiple metrics including MSE and objective function value, both for the initial prediction and as a warm start to an adjoint optimizer. We provide both aggregated results (Fig. 2) as well as illustrative examples (Fig. 3) that shed light on how adjoint optimizers adjust to warm starting by ID methods.
3. We compare two methods for optimizing the hyperparameters of those ID models, specifically minimizing the PMSE and another that minimizes the Objective Function value of the ID predictions at iteration 0, which we call the Prediction Objective Function Minimization Method (POFMM). We show in Fig. 2 and Tables 1 and 2 that models optimized with POFMM outperform those trained using PMSE.
4. We also provide sensitivity analyses of each model's performance with respect to key hyper-parameters of each ML model, noting that perturbations from those used in our experiments do not fundamentally alter model performance.

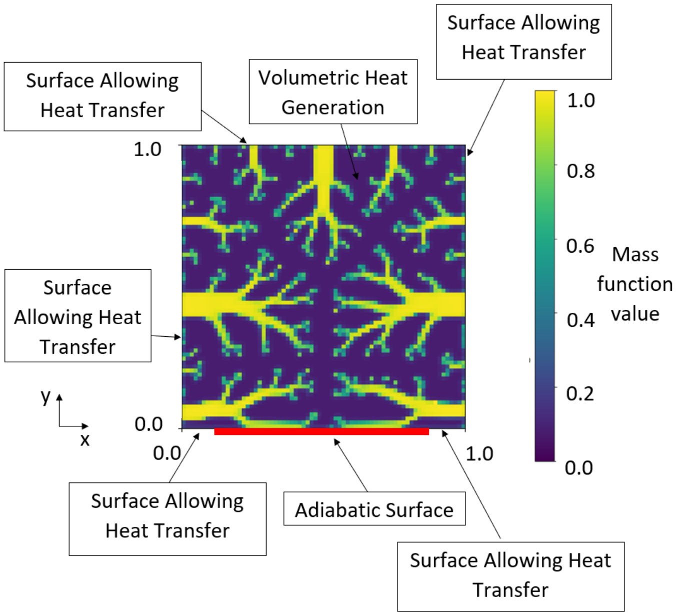

Surface Allowing Heat Transfer
Volumetric Heat Generation
Surface Allowing Heat Transfer
1.0
Surface Allowing Heat Transfer
y
x
0.0
0.0
1.0
1.0
0.8
0.6
0.4
0.2
0.0
Mass function value
Surface Allowing Heat Transfer
Adiabatic Surface
Surface Allowing Heat Transfer

FIGURE 1: The physical layout of this paper's TO problem that we will test ID methods on, adapted from [3, 4]. Note that the colorbar refers to the value of the mass function at a given point.

BACKGROUND AND RELATED WORK

In this section, we provide background on ID problems in general, the specific ID methods we use, and the needed background on the physical problem the paper addresses.

Inverse Design

Inverse design problems are a subset of inverse problems in general [5]. Inverse problems are defined by the inference of model parameters from observed measurements [6]. These measurements can be obtained via the observation of physical or simulated systems. This is in contrast with *forward modeling*, wherein one discovers physical principles that allow for the forecasting of system behavior [6]. Mathematically, an inverse problem can be described by

$$y = F(x) + e \tag{1}$$

where y ∈ Y is system observations (data), x ∈ X is a parameter of interest, F is the mapping between relevant function spaces (forward operator F: X → Y), and e is observational noise [7].

Unfortunately, in many applications, the attainment of a forward model is often impeded by gaps in understanding of underlying principles governing the system [7]. Furthermore, even when a forward model offers a high degree of accuracy in describing the behavior of a physical system, such a model often incurs a high computational cost to implement [7]. It is therefore often desirable to develop a mapping between some input parameter and the response of interest via use of input and output data without the costly and limited implementation of the forward model [7]. This is the essence of inverse problems and, as it relates to the engineering of systems, inverse design.

Various machine learning modeling techniques have demonstrated significant promise in efficiently providing such mappings [7].

Currently, in the field of mechanical engineering, a large amount of work has been done investigating the utility of ID in efficiently designing and characterizing materials (particularly nanomaterials and metamaterials) and microstructures [1, 5, 8, 9, 10]. The area of design for electromagnetic wave manipulation (e.g. nanophotonics) is particularly active [11, 12, 13, 14, 15]. ID methods have also been applied to problems in areas including molecular discovery [16], additive manufacturing [17], airfoil design [5, 18], and imaging [19].

Due to their typically greater efficiencies relative to forward or surrogate modeling techniques, ID methods have also been explored as a means of accelerating the optimization process, often by providing good initialization points [5, 13, 20, 21, 22].

While researchers have studied Inverse Design using a wide variety of predictive models, in this paper we chose two common, simple, and fast Machine Learning models: K-Nearest Neighbors (KNN) and Random Forests (RF). We chose these models since they can provide a readily reproducible benchmark for future research in this area while allowing us to rigorously study the fundamental questions on interest regarding the effects of ID metrics and warm start behavior. More advanced methods, such as Convolutional Neural Networks, Graph Neural Network models, or their variants would likely improve upon the raw numerical errors we show in later sections and would serve as interesting comparisons in the future. With this in mind, we will now provide some brief background on KNN and RF models, both of which have recently been applied to inverse design problems, and in particular, topology optimization problems [23, 24].

K- Nearest Neighbors K-nearest neighbors is an algorithm that classifies (or, in the case of regression problems, assigns a value to) unknown data points based on the characteristics of the data points in the training set that are closest in proximity to the unknown data point [25]. The Euclidean distance is often used to assess proximity between points:

$$d(a,b) = \sqrt{\sum_{i=1}^{N}(a_i - b_i)^2} \tag{2}$$

wherein a and b are two points, d is the Euclidean distance between them, and N is the dimensionality of the data [25]. The selection of k, the number of neighboring points considered, can have a large affect on model performance [25]. Other parameters, such as the weighting of different points considered, can also have an impact on performance [26]. K-nearest neighbors is relatively efficient in the training and testing process [25].

Random Forests The random forest technique is an ensemble method that employs decision trees to classify (or, in the case of regression, assign a value to) a data point [25]. This method selects random features to determine the optimal “split point” via calculating the Gini-index cost function [25]. Once trees are established, a classification (or regression) is made via passing a data point to all trees, and for classification, the class with the greatest number of trees assigning the point to said class is output as the classification [25].

Background on Our Specific Topology Optimization Problem

One type of problem that lends itself to inverse design approaches is that of topology optimization. Via topology optimization, one seeks to determine the distribution of material in a space that best satisfies certain performance criteria [27].

A particular subset of topology optimization problems are those in which an optimal distribution of material is sought to facilitate the transfer of heat through a domain while satisfying certain criteria. In more sophisticated cases, this can encompass the optimization of heat exchanger geometry subject to convective heat transfer in turbulent flow [28].

The problem we consider is a simpler one of the topology optimization of a 2D heat sink subject to pure conduction derived from a demonstration provided in [3]. In this problem, we seek to minimize the thermal compliance

$$\int_{\Omega} fT + \alpha \int_{\Omega} \nabla a \cdot \nabla a \tag{3}$$

wherein Ω is the unit square, f is the heat source term, T is the temperature, α is a regularization term, and a is the mass distribution function (a = 1 for material, a = 0 for no material) [3].

This problem has constraints “subject to the Poisson equation with mixed Dirichlet–Neumann conditions” [3]. It is also subject to the constraints of a ∈ [0,1] and the volume bound V such that:

$$\int_{\Omega} a \leq V \qquad (4)$$

for the entirety of Ω [3]. This problem is described in greater detail in example 1 of [4]. Following the example in [3], we employ an interior point optimization method described in [29] to generate data for model training and evaluation purposes, also described further later in the paper.

METHODOLOGY

To address the contributions mentioned in the introduction, our methodology has the following main steps: (1) generating the Dataset we use for later testing of ID methods and the TO warm starting, (2) how we train and optimize our specific ID methods, and (3) how we measure and evaluate the results from the ID methods.

Dataset Creation via 2D Heat Exchanger Topology Optimization

To create realistic but feasible benchmark TO problem upon which we could conduct our ID experiments, we built upon a classical thermal compliance example with Poisson Equation constraints from [4] and described further below.

Specifically, in this optimization problem, we sought to minimize the thermal compliance of the geometry given a limit on the volume of conducting material that can be used, and an adiabatic region of a prescribed length on one side of the problem space, which is a unit square. (See above background section and Fig. 1).

For our dataset generation, we explored an input space of two parameters: the upper volume limit on the material present in the unit square (*i.e.*, Volume Fraction) and the length of the adiabatic region on one side of the square. We selected values for the volume limit between 0.3 and 0.6, since the interior point solver we used (IPOPT) could not reliably produce converged results outside that range, and values within that range showed sufficient topological variability. We selected values for the adiabatic region length between 0 (corresponding to an absence of an adiabatic region) to 1 (corresponding to an entire side of the unit square being adiabatic). We then divided each parameter range into 20 segments, resulting in 21 values of interest for each (*i.e.*, the volume bounds sampled were 0.3, 0.315, 0.33, [. . .], 0.57, 0.585, 0.6, and the lengths sampled were 0.0, 0.05, 0.1, [. . .], 0.9, 0.95, 1.0). We then generate optimized topologies for each combination of volume limit and adiabatic region length by running the interior point solver to convergence within a small tolerance with respect to the objective function and design variables, as is standard.

We used a 70 x 70 mesh as the design domain, since this was large enough to provide fine details within the design space typical of topologically optimized solutions to this problem (see Fig. 1), without requiring a needlessly large increase in computational running time for the solver.

For each combination of parameters, we ran three consecutive optimization runs wherein IPOPT exited upon the satisfaction of a tolerance of 1.0e-3. The first run was initialized with a constant mass distribution set to the value of the volume limit. The two following runs were initialized with the output mass distribution of the preceding run. We used Dolfin-Adjoint [30, 31] and IPOPT [29] for finite element optimization.

Upon the completion of the third run for a set of parameters, the resulting distribution was discretized to convert the data into a form that is amenable to the simple regression models used in this paper. This was accomplished by dividing the unit square into a 70 x 70 grid and taking the mass function value at each intersection point, resulting in 71*71 = 5,041 points taken per topology. This was done to capture all of the information contained in the output of the optimizer. Therefore, each sampled point had five values associated with it: its x-coordinate, its y-coordinate, the volume bound, the length of the adiabatic region, and the corresponding value of the mass function.

Model Training and Cross Validation

In our optimization of model hyperparameters, we employed a methodology wherein for each test-train split, we randomly selected two unique values of volume limit and two unique values of adiabatic region length to exclude from the training set. In other words, all data points corresponding to topologies defined by possessing any of these values for volume limit and adiabatic region width were excluded from the training set. To find the Pointwise MSE, a model’s predictive abilities were then tested on data points corresponding to the topologies defined by having a volume limit and an adiabatic region length that were excluded from the training set. For our Prediction Objective Function Minimization Method, we followed the same training regime followed by testing on the four topologies with both adiabatic region length and volume limit excluded from the training set. In doing so, the model was tested on its ability to predict mass distributions simultaneously for values of both volume limit and adiabatic region width that it had not been trained on. We used the implementations of KNN and RF models available in the scikit-learn library [32]. However, each model is defined by its own set of unique hyperparameters, which we trained and tested via the above cross validation way ten times.

To select the final models for KNN and RF, however, we had to select which metric to use on the cross validation results to select the best model. Herein lies a major difference between the standard way of selecting ID model—picking the model that minimizes the Pointwise MSE—and one that evaluates the model directly on the objective function of interest—what we refer to later in the paper as Prediction Objective Function Minimization. We will describe each approach in turn, and then show in the results section how they impact ID performance.

Hyperparameter Optimization: PMSEM Our Pointwise Mean Squared Error Method (PMSEM) for model performance evaluation employs the following steps for comparing predictions to the corresponding topology within the test set:

1. Divide the experiment volume into a 70 x 70 grid
2. Find the model prediction at each intersection point in the grid, resulting in a 71 x 71 array of predictions
3. Find the mean squared error between these prediction and their corresponding points in the topology from the test set

Mathematically, the PMSEM in a given trial can be described as a minimization of the PMSEM error measure E with respect to model hyperparameters H, where E is defined as

$$E = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}(P_{i,j} - R_{i,j})^2 \tag{5}$$

wherein M is the number of images tested, N are the number of points per topology, i is the index of the topology, j is the index of the point in a given topology, P is the predicted value of the mass function at said point, and R is the actual value from the test set.

Physically, PMSEM compares the similarity of the mass distributions predicted by a model to the ground truth distributions.

Hyperparameter Optimization: POFMM In contrast to PMSEM, our Prediction Objective Function Minimization Method (POFMM) uses an alternative approach to model hyperparameter optimization. As one intention of using a model in an inverse design problem is to produce a prediction that is as close to an optimal design as possible and is therefore a good initialization point for future iterations, it is therefore desirable to find hyperparameter values which enable the model to yield predictions with good objective function values at iteration 0. In our problem, this means minimizing the objective function (thermal compliance) value of the model's predictions at iteration 0. Rather than comparing the predictions to the corresponding mass distributions in the test set, the model hyperparameters H can be optimized solely with respect to this objective function value F. Note that unlike PMSE, a model optimized in this way may not produce designs that are as close in the design space (*i.e.*, have the same geometry) as the training set compared to the PMSE method, yet should in principle still be able to produce results with high performance. In practice, because the primary parameters of the model (whether KNN or RF) are trained via MSE, these differences only affect model choice at the hyper-parameter level.

Optimized Model Evaluation Process

For a given hyperparameter setting, we can now train the correspondent models via the train-test split scheme described above. We then use the optimized KNN and RF models to generate predictions for each tested combination of volume limit and adiabatic region length.

For each combination of volume bound, adiabatic region length, and model type, we initialized the interior point solver (IPOPT) with the corresponding prediction. To guard against possible premature convergence, we ran IPOPT four consecutive times, with the second, third, and fourth runs initialized using the output of the previous run. The optimizer exited each run upon the satisfaction of the tolerance of 1.0e-3.

As a control, we used an uniform initialization with a constant mass distribution equal to the volume fraction, since this is the most common initialization for Solid Isotropic Material with Penalisation (SIMP)-based density TO methods. Three runs per combination of volume limit and adiabatic region length were performed in the same manner as for the warm started run sequences.

In total, 200 combinations of volume limit and adiabatic region width were tested with KNN initializations, 200 were tested with RF initializations, and 200 were tested with the control initializations.

Data Postprocessing Following the conclusion of optimized model evaluation process, thermal compliance trajectory results were normalized and averaged. This was done to prevent any individual combinations of volume limit and adiabatic region length from having a disproportionally large or small effect on the averaged thermal compliance trajectory. Specifically, the postprocessing procedure is as follows:

1. Normalize each value in each objective function trajectory with respect to the optimal (minimum) value obtained in said trajectory.
2. Find the mean of these normalized values at each iteration number for the PMSEM-optimized KNN models, the PMSEM-optimized RF models, POFMM-optimized KNN models, and the POFMM-optimized RF models.
3. Render all trajectories uniform in the number of iterations considered by extending runs that terminate before the maximum number of iterations attained among any runs. This extension is achieved by conservatively extrapolating the final value reached in each run over the remaining iterations.

RESULTS

Following the above methodology, this section first reviews the optimal models that we found for our specific 2D heat sink problem, then presents the main quantitative results on the impact of the initialization methods on ID performance. Following these, we provide qualitative comparisons of the final designs produced under each method, and an example trajectory that helps shed light how the ID method influences the warm start behavior of further Topology Optimization. For specific experiment details, access to our dataset, reproducible experiment code, and the simulation environment see: `https://github.com/IDEALLab/ID_Conduction_IDETC2022`.

Note that in all subsequent plots with shaded regions, these shaded areas depict the 95% confidence intervals of their corresponding plotted functions. We approximated the data as having a t-distribution given the limited number of samples.

Optimal Model Hyperparameters

For KNN models, we tested combinations of the number of neighbors and weightings. Specifically, we tested models with 1, 2, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, and 100 neighbors with either “uniform” or “distance” weightings.

For RF models, we tested combinations of the number of estimators and the minimum samples per leaf. Specifically, we tested models with 1, 5, 10, 15, 20, 25, and 30 estimators with 1, 5, 10, 15, and 20 minimum samples per leaf.

PMSEM Using PMSEM, we found that the optimal hyperparameters for KNN models were 50 neighbors and uniform weighting. Similarly, for RF models we found that 20 estimators with a minimum of 20 samples per leaf were ideal. We provide sensitivity analysis plots for these parameters in Appendix A in Fig. 6 and 7, for the KNN and RF models, respectively.

POFMM Using POFMM, we found that the optimal hyperparameters for KNN models were 1 neighbor with either “distance” or “uniform” weighting. We selected “uniform” weighting for the optimized model evaluations given their equivalent performances by the POFMM metric. We found that for RF models, 1 estimator with a minimum of 1 sample per leaf was optimal. We provide sensitivity analysis plots for these parameters in Appendix A in Fig. 8 and 9, for the KNN and RF models, respectively.

Impact of Different Initialization Methods on Prediction Performance and Trajectory Acceleration

Using these optimized models, we can now compare how they perform at both predicting the optimal geometry as well as how they act as a warm start to further Topology Optimization, compared to a control (uniform initialization).

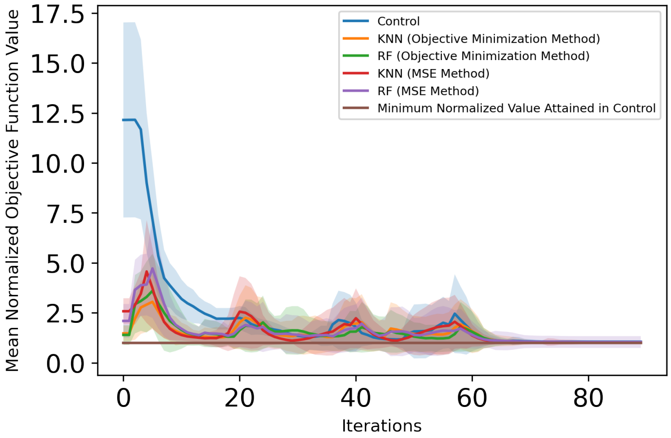


FIGURE 2: Mean normalized optimization trajectories for the tested initialization techniques

We found that, on average, all model types tested with either hyperparameter optimization method (POFMM or PMSEM) produced predictions with thermal compliance values significantly less than that of the control (Figure 2). We also found that initializing the IPOPT optimization process using these methods also, on average, offered an acceleration for low evaluation numbers, despite the fact that the optimizer increases the thermal compliance in early iterations of warm starting (Figure 2)—we show why this occurs in the next section.

Beyond around 20 interations, the control begins to reach comparable performance to that of the warm started optimizers. We note that each of the “bumps” in performance that occur roughly every 20 iterations correspond to restarting the IPOPT solver after initial convergence. This does not appear to help any of the warm started models (see Fig. 2 around iteration 20), however restarting the solver was needed to help the control method achieve better performance (otherwise the solver converges to a solution with significant optimality gap).

Specifically, on average, KNN models optimized using POFMM generated predictions that, at iteration 0, performed at 147.2% of the minimum thermal compliance reached in the corresponding control run, whereas predictions generated using KNN models optimized using PMSEM averaged at 257.8% (Table 1). For RF models, these figures were 139.3% and 209.3% using POFMM and PMSEM, respectively (Table 1). For comparison, a constant mass distribution set the volume limit (the control for this experiment) averaged at 1,215% (Table 1), almost an order of magnitude larger than the ID methods.

Furthermore, on average, the use of ID-generated initialization accelerated the topological optimization process up until iteration 13 (recall that these iterations and evaluations are 0-indexed) even despite the initial increase in thermal compliance values observed for lower iteration numbers. As illustrated in

TABLE 1: Mean Normalized Thermal Compliance (MNTC) at the zeroth iteration for tested initialization schemes

Model Type	MNTC at Iteration 0
KNN PMSEM	2.578
RF PMSEM	2.093
KNN POFMM	1.472
RF POFMM	1.393
Control	12.15

TABLE 2: Performance of Different Model Types at Iteration 13

Model Type	MNTC at Iteration 13
KNN PMSEM	1.254
RF PMSEM	1.382
KNN POFMM	1.277
RF POFMM	1.336
Control	2.627

Table 2, all optimizations using model-generated initializations on average outperformed the control (Table 2). We observed that KNN models averaged at 127.7% and 125.4% of the minimum thermal compliance attained in the corresponding control run using POFMM and PMSEM, respectively (Table 2). For RF models, these values are 133.6% and 138.2% for POFMM and PMSEM, respectively (Table 2).

Why Does Thermal Compliance Increase After Warmstarting?

A typical non-normalized optimization trajectory for the warmstarted optimizations is shown in Figure 3. In particular, Figure 3 displays the trajectory taken by the optimization of a mass distribution subjected to a volume limit of 0.315, an adiabatic region length of 0.4, and an initialization produced by a KNN model optimized using POFMM (Figure 3).

In this case for the KNN-initialized trajectory, it appears that the increase in thermal compliance that peaks at iteration 4 is due to the mass distribution becoming less defined for iterations at and around this value. In other words, the optimizer is subtracting some of the material predicted by the model.

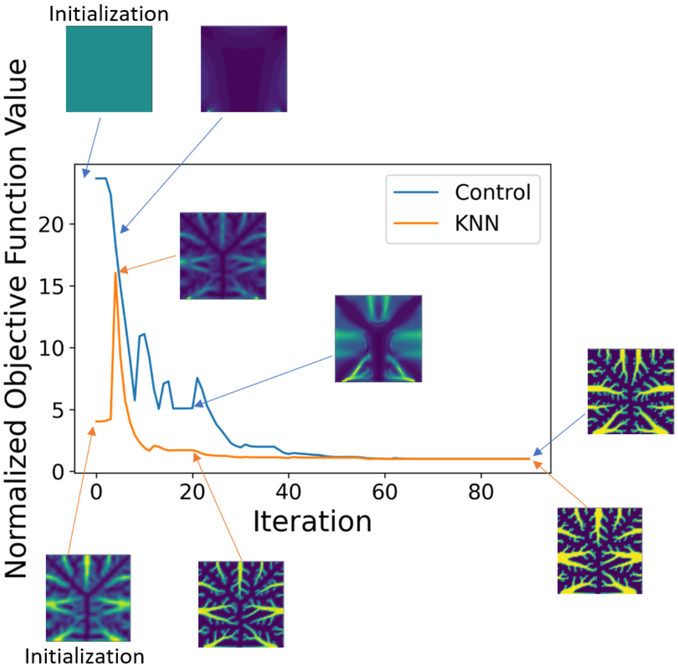


FIGURE 3: The evolution of 2D heat exchanger designs over the course of the optimization process. Here, the trajectory referenced as "KNN" refers to the trajectory initialized with the prediction of a KNN model optimized using PMSEM. The control trajectory is initialized with a constant distribution set to the volume limit.

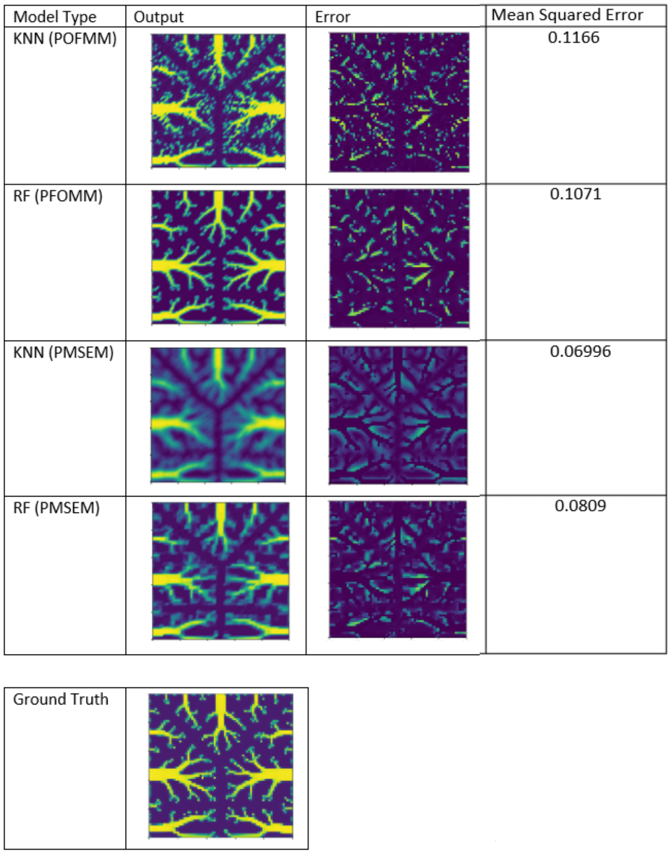
In contrast, the control trajectory is significantly smoother and lacks this degree of loss of definition in its mass distribution.

A Visual Comparison Between Predictions and Ground Truths

None of the trained models predicted an identical mass distribution for an designs in the test set. Nevertheless, as the case depicted in Figure 4 shows, the model predictions resembled the "ground truth" test design and illuminate how the different ID models and hyper-parameter optimization schemes qualitatively affected the predicted designs.

Impact of Different Initializations on Final Optimized Designs

We also observed qualitative differences between the 2D heat sink designs produced from optimization processes run with different initializations. This is expected due to the non-convexity of the problem, as in such problems differing initializations are often expected to result in a numerical optimizer arriving at different local minima. Our expectations of the close



Model Type	Output	Error	Mean Squared Error
KNN (POFMM)			0.1166
RF (PFOMM)			0.1071
KNN (PMSEM)			0.06996
RF (PMSEM)			0.0809

Ground Truth	

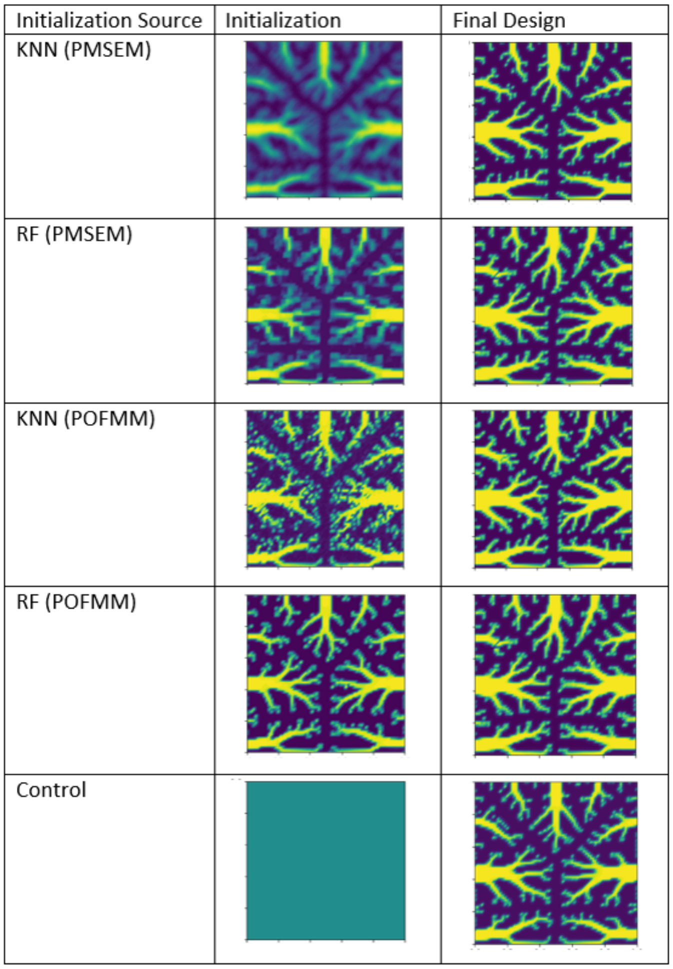
FIGURE 4: There are areas in this example case that all models tested struggle to predict, as shown by the lighter areas in the distributions in the "Error" column. This example is for a design with a volume fraction of 0.315 and an adiabatic region width of 0.9.

similarity of objective function values of these final designs were also met in numerous instances, including the cases presented in Figure 5.

Figure 5 displays the results of four consecutive optimization runs using different initializations. All runs used a volume fraction of 0.315 and an adiabatic region length of 0.90. Note that the optimizer exited upon the satisfaction of the tolerance of 1.0e-3. There are small differences in the structures of the designs that are visible upon inspection. The designs nevertheless appear to retain their dendritic character in this case, and share substantial resemblance to each other.

DISCUSSION, LIMITATIONS, AND FUTURE WORK

While the specific ID models we tested were simple, they nevertheless highlight interesting phenomena that may generalize to other problems or ID methods. Here we review possible limitations or areas of future work that may affect our stated contributions.



Initialization Source	Initialization	Final Design
KNN (PMSEM)		
RF (PMSEM)		
KNN (POFMM)		
RF (POFMM)		
Control		

FIGURE 5: There are numerous structural differences between optimized designs that used different initializations.

Alternative training and testing metrics Changing how we optimized the hyper-parameters of the ID models (from MSE-driven to Objective Function-driven) affected not only quantitative convergence measures (per Fig. 2), but also qualitative predictions (per Fig. 5), even though the underlying primary training measures were identical (*i.e.*, both the KNN and RF models minimize the design construction error during training).

This raises questions not only about how we evaluate ID methods, but also points toward alternative training procedures. For example, strategies such as training over joint losses that include both reconstruction-error-type losses as well as objective-function-derived losses. Moreover, if we know that the goal of an ID method is to act as a warm start for additional TO, then perhaps predicting the “peak“ distribution seen in Fig. 3 may be faster than either of the PMSEM or POFMM approaches used in this paper. Understanding under what conditions and ID applications different methods excel would be a fruitful avenue of future work.

Limitations in the Chosen Problem Some of these limitations and potential areas directly involve the physical problem that we considered—*i.e.*, the 2D heat sink. The results of this work are limited to this particular physical problem with a volume limit able to vary between 0.3 and 0.6 and an adiabatic region width able to vary between 0.0 and 1.0. In addition, we only tested the models on a mesh size of 70x70; it is certainly possible that higher or lower dimensional problems may introduce variability into the phenomena we have observed.

While we have no reason to believe that the fundamental behavior seen in our 2D examples would not extend to more complex or 3D SIMP/density-based TO examples, there may well be differences in other TO respresentations (levels sets, spectral representations), and for other shape optimization approaches more broadly.

Variations in ID methods and dataset size Lastly, using more advanced ID methods may alter some of our observations, for example, by eliminating the need for an adjoint optimizer to “erase and reoptimize” portions of the ID prediction as seen in Fig. 2 if the ID predictions lie sufficiently close to the global optima. Likewise, we did not investigate here how ID performance is modulated with the size of the training data set; uncovering transition points where ID methods become performant remains an open question worthy of future study in general.

CONCLUSIONS

We compared several simple Inverse Design models (KNNs and Random Forests) and two approaches for model hyperparameter optimization with standard uniform initialization using the SIMP-based TO on a 2D heat sink problem. We described a benchmark environment and dataset and showed how those models affected both the initial predictions from the ID methods as well as downstream accelerations when warm starting optimization.

Our findings indicate that both methods of hyperparameter optimization yield KNN and RF models that can substantially accelerate the optimization process when their predictions initialize an interior point solver. These predictions also tend to have thermal compliances close to the minimum obtained in a corresponding control optimization run, and tend to significantly outperform initialization with a uniform mass distribution—a common TO initialization method. Moreover, we showed how optimizing ID methods solely with respect to Mean Squared Error on the reconstruction of the test set designs is not necessarily the best strategy. Instead optimizing for models that produce lower objective function values can outperform standard MSE derived hyper-parameter optimization methods. Although we investigated a specific physical problem, two model hyperparameter optimization methods, and two ID model types (KNN and RF), there remains a large space of future work in both different physical problems and in different computational approaches to modeling and model optimization. Overall, our results highlight the nuances in evaluating ID methods—that your end goal in Inverse Design, whether that be direct prediction, distribution matching, or warm starting an optimizer, can affect both your evaluation approach and how you optimize your models.

Acknowledgements

This research was supported in part by funding from the U.S. Department of Energy’s Advanced Research Projects Agency-Energy (ARPA-E) DIFFERENTIATE funding opportunity through award DE-AR0001216.

REFERENCES

[1] Lee, X. Y., Balu, A., Stoecklein, D., Ganapathysubramanian, B., and Sarkar, S., 2019. “A case study of deep reinforcement learning for engineering design: Application to microfluidic devices for flow sculpting”. *Journal of Mechanical Design,* ***141***(11).

[2] Shi, X., Qiu, T., Wang, J., Zhao, X., and Qu, S., 2020. “Metasurface inverse design using machine learning approaches”. *Journal of Physics D: Applied Physics,* ***53***(27), p. 275105.

[3] Topology optimisation of heat conduction problems governed by the poisson equation. http://www.dolfin-adjoint.org/en/latest/documentation/poisson-topology/poisson-topology.html. Accessed: 2022-02-10.

[4] Bendsoe, M. P., and Sigmund, O., 2003. *Topology optimization: theory, methods, and applications*. Springer Science & Business Media.

[5] Chen, Q., Wang, J., Pope, P., Chen, W. W., and Fuge, M., 2021. “Inverse design of 2d airfoils using conditional generative models and surrogate log-likelihoods”. *Journal of Mechanical Design*, pp. 1–22.

[6] Tarantola, A., 2005. *Inverse problem theory and methods for model parameter estimation*. SIAM.

[7] Arridge, S., Maass, P., Öktem, O., and Schönlieb, C.-B., 2019. “Solving inverse problems using data-driven models”. *Acta Numerica,* ***28***, pp. 1–174.

[8] Kim, B., Lee, S., and Kim, J., 2020. “Inverse design of porous materials using artificial neural networks”. *Science advances,* ***6***(1), p. eaax9324.

[9] Kim, S., Noh, J., Gu, G. H., Aspuru-Guzik, A., and Jung, Y., 2020. "Generative adversarial networks for crystal structure prediction". *ACS central science,* ***6***(8), pp. 1412–1420.

[10] Challapalli, A., Patel, D., and Li, G., 2021. "Inverse machine learning framework for optimizing lightweight metamaterials". *Materials & Design,* ***208***, p. 109937.

[11] Huang, Z., Liu, X., and Zang, J., 2019. "The inverse design of structural color using machine learning". *Nanoscale,* ***11***(45), pp. 21748–21758.

[12] Liu, Z., Zhu, D., Raju, L., and Cai, W., 2021. "Tackling photonic inverse design with machine learning". *Advanced Science,* ***8***(5), p. 2002923.

[13] Wiecha, P. R., Arbouet, A., Girard, C., and Muskens, O. L., 2021. "Deep learning in nano-photonics: inverse design and beyond". *Photonics Research,* ***9***(5), pp. B182–B200.

[14] So, S., and Rho, J., 2019. "Designing nanophotonic structures using conditional deep convolutional generative adversarial networks". *Nanophotonics,* ***8***(7), pp. 1255–1261.

[15] Jiang, J., and Fan, J. A., 2020. "Simulator-based training of generative neural networks for the inverse design of metasurfaces". *Nanophotonics,* ***9***(5), pp. 1059–1069.

[16] Sanchez-Lengeling, B., Outeiral, C., Guimaraes, G. L., and Aspuru-Guzik, A., 2017. "Optimizing distributions over molecular space. an objective-reinforced generative adversarial network for inverse-design chemistry (organic)".

[17] Jin, Z., Zhang, Z., Demir, K., and Gu, G. X., 2020. "Machine learning for advanced additive manufacturing". *Matter,* ***3***(5), pp. 1541–1556.

[18] Sekar, V., Zhang, M., Shu, C., and Khoo, B. C., 2019. "Inverse design of airfoil using a deep convolutional neural network". *Aiaa Journal,* ***57***(3), pp. 993–1003.

[19] Ongie, G., Jalal, A., Metzler, C. A., Baraniuk, R. G., Dimakis, A. G., and Willett, R., 2020. "Deep learning techniques for inverse problems in imaging". *IEEE Journal on Selected Areas in Information Theory,* ***1***(1), pp. 39–56.

[20] Kim, I., Park, S. J., Jeong, C., Shim, M., Kim, D. S., Kim, G.-T., and Seok, J., 2022. "Simulator acceleration and inverse design of fin field-effect transistors using machine learning". *Scientific Reports,* ***12***(1), pp. 1–9.

[21] Hegde, R., 2021. "Sample-efficient deep learning for accelerating photonic inverse design". *OSA Continuum,* ***4***(3), pp. 1019–1033.

[22] Klaučo, M., Kalúz, M., and Kvasnica, M., 2019. "Machine learning-based warm starting of active set methods in embedded model predictive control". *Engineering Applications of Artificial Intelligence,* ***77***, pp. 1–8.

[23] Jiang, X., Wang, H., Li, Y., and Mo, K., 2020. "Machine learning based parameter tuning strategy for mmc based topology optimization". *Advances in Engineering Software,* ***149***, p. 102841.

[24] Li, J. K., and Zhang, Y. M., 2011. "Method of continuum structural topology optimization with information functional materials based on k nearest neighbor". In Advanced Materials Research, Vol. 321, Trans Tech Publ, pp. 200–203.

[25] Singh, A., Halgamuge, M. N., and Lakshmiganthan, R., 2017. "Impact of different data types on classifier performance of random forest, naive bayes, and k-nearest neighbors algorithms".

[26] Chernoff, K., and Nielsen, M., 2010. "Weighting of the k-nearest-neighbors". In 2010 20th International Conference on Pattern Recognition, IEEE, pp. 666–669.

[27] Sigmund, O., and Maute, K., 2013. "Topology optimization approaches". *Structural and Multidisciplinary Optimization,* ***48***(6), pp. 1031–1055.

[28] Dilgen, S. B., Dilgen, C. B., Fuhrman, D. R., Sigmund, O., and Lazarov, B. S., 2018. "Density based topology optimization of turbulent flow heat transfer systems". *Structural and Multidisciplinary Optimization,* ***57***(5), pp. 1905–1918.

[29] Wächter, A., and Biegler, L. T., 2006. "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming". *Mathematical programming,* ***106***(1), pp. 25–57.

[30] Mitusch, S. K., Funke, S. W., and Dokken, J. S., 2019. "dolfin-adjoint 2018.1: automated adjoints for fenics and firedrake". *Journal of Open Source Software,* ***4***(38), p. 1292.

[31] Funke, S. W., and Farrell, P. E., 2013. "A framework for automated pde-constrained optimisation". *arXiv preprint arXiv:1302.3894*.

[32] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E., 2011. "Scikit-learn: Machine learning in Python". *Journal of Machine Learning Research,* ***12***, pp. 2825–2830.

APPENDIX A: SENSITIVITY ANALYSIS OF KNN AND RF HYPER-PARAMETERS

Fig. 6 and 7 show how the test MSE changed as a function of hyper-parameter for variations of the KNN and RF models, respectively. Fig. 8 and 9 show how the objective function value minimization changed as a function of hyper-parameter, for the KNN and RF models, respectively.

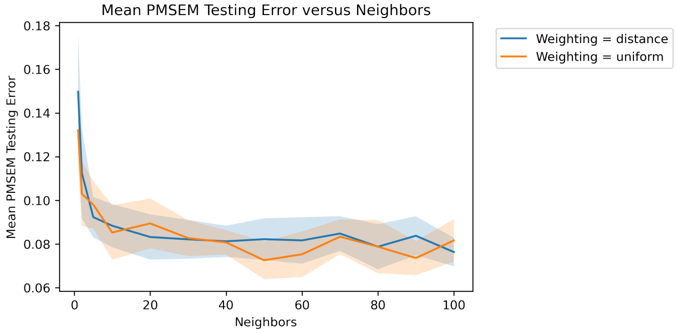


FIGURE 6: The PMSEM-optimized KNN model has 50 estimators using uniform weighting

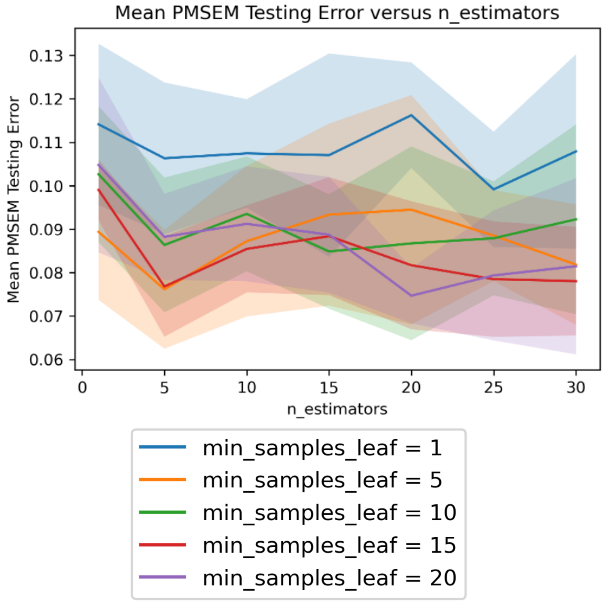


FIGURE 7: The PMSEM-optimized RF model has 20 estimators with a minimum of 20 samples per leaf

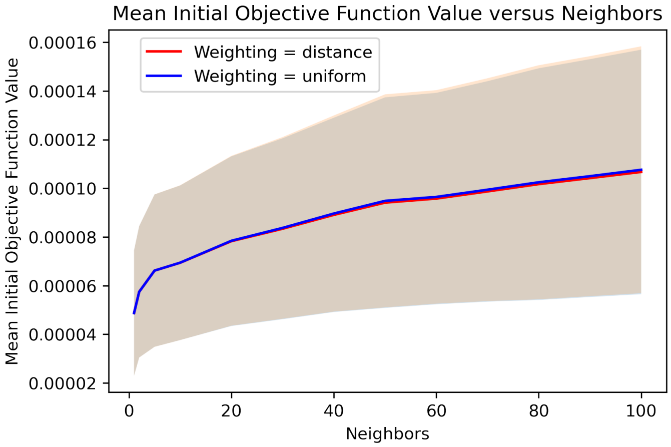


FIGURE 8: The POFMM-optimized KNN model has 1 neighbor using either weighting

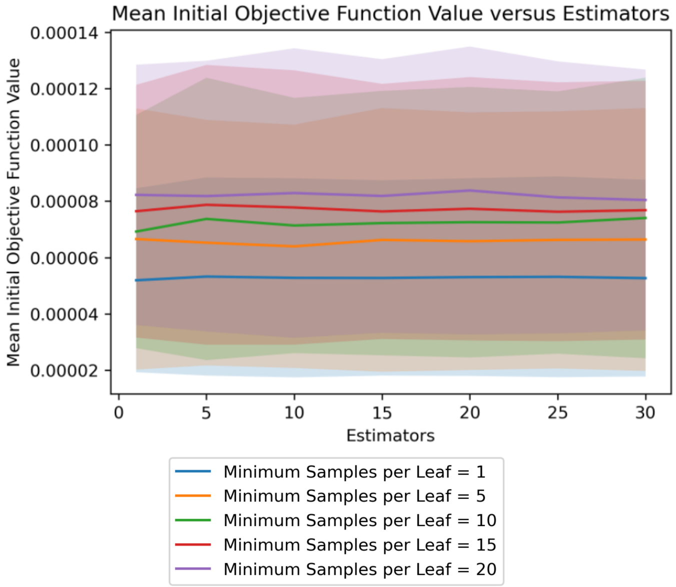


FIGURE 9: The POFMM-optimized RF model has 1 estimator with a minimum of 1 sample per leaf