Using Sparse Gaussian Processes for Predicting Robust Inertial Confinement Fusion Implosion Yields

Peter Hatfield, Steven Rose, Robbie Scott, Ibrahim Almosallam, Stephen Roberts, and Matt Jarvis

Abstract—Here, we present the application of an advanced sparse Gaussian process-based machine learning algorithm to the challenge of predicting the yields of inertial confinement fusion (ICF) experiments. The algorithm is used to investigate the parameter space of an extremely robust ICF design for the National Ignition Facility, the “Simplest Design”; deuterium–tritium gas in a plastic ablator with a Gaussian, Planckian drive. In particular, we show that: 1) GPz has the potential to decompose uncertainty on predictions into uncertainty from lack of data and shot-to-shot variation; 2) it permits the incorporation of science-goal-specific cost-sensitive learning (CSL), e.g., focusing on the high-yield parts of parameter space; and 3) it is very fast and effective in high dimensions.

Index Terms—Gaussian processes; inertial confinement.

I. INTRODUCTION

Inertial confinement fusion (ICF), in which deuterium–tritium (DT) fuel is compressed to temperatures and densities exceeding that found in the Sun, is one of the main potential pathways to nuclear fusion as a source of energy. The world’s leading ICF facility is the National Ignition Facility (NIF) at the Lawrence Livermore National Laboratory (LLNL). NIF uses indirect drive in which lasers first hit a hohlraum (typically a gold can), which then emits a thermal radiation field that drives the implosion, in contrast to direct drive ICF in which laser beams themselves drive the implosion.

Although huge progress has been made, NIF has been unable to reach the yields originally hoped for. This has led to an interest in using modern machine learning (ML) techniques to produce new designs and quantify uncertainties on predictions. Reference [1] presented an early ensemble of thousands of ICF implosions and used Gaussian processes (GPs) to model the parameter space. Reference [2] developed a novel neural network (NN) ML algorithm called “deep jointly informed NNs” (DJINNs) that used random forests to construct appropriate NN architectures with relatively little human input, which was used in [3] to identify a novel nonspherically symmetric design for NIF. Reference [4] used transfer learning1 with DJINN to update the ML predictions based on seeing real experiments. Finally, [5] used genetic algorithms in an even wider parameter space to produce ICF designs almost from scratch.

In this article, we use a GP-based ML algorithm GPz, [6], [7] to build surrogate models of a robust ICF design. GPz has the following characteristics.

1) It has a flexible cost-sensitive learning (CSL) feature that permits optimization for the specific science goal at hand.
2) It models heteroscedastic noise, permitting the uncertainty to be decomposed into uncertainty from shot-to-shot variation and uncertainty from lack of data.
3) It uses a sparse framework that lets it run quickly even in high dimensions. We compare it to DJINN’s performance on the same data and discuss future approaches to building ICF surrogates.

II. PROBLEM FORMULATION AND METHODOLOGY

A. Simplest Design

The point design for NIF is an indirect drive design, with a capsule of DT gas inside an ablator shell (plastic, CH, or a few other options) with a thin layer of solid DT ice on the inside. In the conventional implosion, the ablation compresses the DT gas to a relatively high convergence ratio (the ratio of the DT radius at the start of the experiment to the radius at peak compression) and extremely high temperatures. If certain criteria are met, sufficient nuclear reactions take place that alpha heating (heating of the DT plasma by alpha particles produced in the nuclear reactions) dominates, and the gas ignites. Then, this starts a burn wave propagating through the

1Transfer learning is a family of ML methods that seek to let an algorithm apply information/knowledge gained from one problem to the task of solving a similar but different problem.
DT ice layer, from which most of the neutron yield comes from. Most designs use a series of shockwaves to fine-tune the implosion, with several variants studied.

The Simplest Design removes some of the more complex/challenging aspects of ICF where the physics is more uncertain: 1) delicate pulse timing; 2) the burn wave through the DT ice; and 3) high convergence ratios. This design simply has DT gas (but of a much higher density), with a CH ablator, and a Gaussian drive (sketch of the capsule design shown in Fig. 1). It is unlikely that the Simplest Design will be able to lead to ignition at NIF, but it does represent a pathway to very predictable robust 1-D implosions. We use a thermal X-ray drive with a Gaussian temperature time dependence.

We parameterize the Simplest Design with five parameters and investigate the parameter space within the following limits:

1. $50 \text{ eV} < T_{\text{peak}} < 400 \text{ eV}$—the peak temperature of the drive;
2. $0.01 \text{ ns} < \sigma < 5 \text{ ns}$—the standard deviation of the time dependence of the temperature of the drive;
3. $0.1 \text{ mm} < r_1 < 1.5 \text{ mm}$—the radius of the DT gas;
4. $0.05 \text{ mm} < r_2 < 1 \text{ mm}$—the radius of the CH ablator;
5. $10 \text{ mg/cc} < \rho < 200 \text{ mg/cc}$—the density of the DT gas (mg/cc).

Note that the gas fill is much higher than typical for ICF (normally closer to $\sim 0.1 \text{ mg/cc}$)—meaning that the implosions have a much lower convergence ratio, but that we would not expect the design to be able to reach yields comparable to that which are in principle possible with the point design on NIF. This is because the point design has low-density gas that is (comparatively) easily compressed. At peak compression, the hotspot (the low-density gas that has been compressed) and DT ice are approximately isobaric, but the density is highly nonuniform, thus resulting in a small region of high temperature in the hotspot, with a low temperature in the ice. The temperatures in the low-density hotspot are in principle sufficient to initiate significant burn within the hotspot (which hopefully drives a burn wave through the DT ice to give extremely high yields). For an equivalent DT mass arranged uniformly (as a function of radius), the energy required to compress it sufficiently to heat it to temperatures high enough to initiate fusion would be far higher than that available on NIF. The design space is chosen to roughly correspond to what is achievable with a gold hohlraum on NIF; the main restrictions are total capsule radius and total energy in drive.

The Simplest Design shares some design philosophy with the two-shock design of [8] and [9]. The design of shot N161004 described in [9] still has a DT ice layer like more typical designs, but has a relatively high gas fill density of $5 \text{ mg/cc}$ (much closer in log space to the densities that we then consider to the point design), and a correspondingly more modest convergence ratio. They also have a relatively simple two-shock drive. These features are experimentally observed to make the implosion much more 1-D and much closer to the predictions of simulations. The Simplest Design goes a step further, with an even higher gas fill, even lower convergence ratio, and an even simpler one-shock drive, and thus also has the potential to give a robust 1-D implosion that closely matches simulation.

### B. Data

Typically, ML methods give better predictions in parts of parameter space with lots of data and vice versa. We use here a Monte Carlo sampling, but rather than sampling uniformly in the parameter under consideration, points are sampled from a multivariate Gaussian centered on where a preliminary estimate for an optimal design was. Reference [3] conversely use Latin hypercube sampling (LHS) to achieve good coverage of the parameter space, which may be a valuable alternative sampling approach for future work. This base design was $T_{\text{peak}} = 300 \text{ eV}$, $\sigma = 2 \text{ ns}$, $r_1 = 1 \text{ mm}$, $r_2 = 0.15 \text{ mm}$, and $\rho = 55 \text{ mg/cc}$; 5000 simulations were run, and we divided it into 30% training, 30% validation, and 40% testing data. The training data are used to infer a large number of parameters that make up the ML predictive model, the validation data are used to infer model hyper-parameters (essentially, the complexity of the model), and the test data are held back to use as a final test of performance. We simulate shot-to-shot variation by artificially adding Gaussian scatter to the calculated log-yields in a number of neutrons. This is chosen to be $\rho$-dependent; the scatter is chosen to have a standard deviation of $0.1 \times \rho/(55 \text{ mg/cc})$ dfig.

Our implosion simulations are performed using the HYADES [10] radiation-hydrodynamics simulation code, which is well benchmarked and widely used for the simulation of inertial fusion and high energy density physics applications [11]–[14]. HYADES models hydrodynamics within a Lagrangian framework. Electron and ion thermal energy transport is described by a flux-limited Spitzer–Härm thermal conductivity model. Equations-of-state either use the Los Alamos SESAME tables [15] or QEOS [16]. Ionization levels come from a hydrogenic average-atom model or self-consistently from QEOS. Radiation transport uses the multigroup diffusion

\[ \text{in log space} \]

\[ \text{Optimal sampling in the parameter space likely depends on the end goal, e.g., if planning to eventually implement transfer learning, it may still be valuable to sample the low-yield parts of parameter space.} \]

\[ \text{Log-yields always measured in logarithms of base ten of number of neutrons} \]
approximation; here, we use 60 groups. A 1-D spherically symmetric geometry is employed.

Neither laser–plasma interactions nor hohlraum physics are modeled; instead, we use an incoming X-ray drive imposed at the outside of the grid. The capsules are modeled within a 5-mm helium container. Simulations start in cryogenic conditions at $1.551 \times 10^{-3}$ eV = 18 K. The HYADES runs were performed on SCARF at the Central Laser Facility, Rutherford Appleton Laboratory, using 500 CPUs. The modeling used here is likely appropriate for the design investigated here,\(^5\) but the next level of sophistication of modeling would be to: 1) move to 2-D/3-D simulations (e.g., so Rayleigh–Taylor instabilities are incorporated) and 2) start to include laser/hohlraum physics and non-Planckian drives (e.g., simulate the laser light being converted to X-rays, rather than just assuming an incoming X-ray drive). See [17] and [18] for the overviews of the wide range of physics involved in ICF and associated issues concerning which physics to include in simulations.

C. GPz

GPz is an ML regression algorithm originally developed for the problem in astrophysics of calculating the photometric redshifts of galaxies: the details of the algorithm and the key developments in the ML theory are described in [6] and [7] and applied to photometric redshift calculation in [19] and [20], and to orbital dynamics in [21]. The algorithm is “GP-based”; a GP is a stochastic process with a random variable defined at each point in a space of interest, such that any finite subset of the random variables has a multivariate normal distribution (equivalently, any linear combination of random variables from different points has a normal distribution). A GP is essentially an unparameterised continuous function defined everywhere with Gaussian uncertainties. A GP-based ML algorithm will typically take a set of data over the parameter space of interest and, in some sense, try and find the function in the function space defined by the GP that was most likely to have produced the data—and then make predictions for other parts of parameter space based on that.

GPz is a sparse GP-based code, a fast and a scalable approximation of a full GP [22], with the added feature of being able to produce input-dependent variance estimations (heteroscedastic noise). For the full details of the algorithm, see [6], [7], and [23], but we summarize the main details here. The model assumes that the probability of observing a target variable $y$ given the vector input $x$ is $p(y|x) = \mathcal{N}(\mu(x), \sigma(x)^2)$. The mean function $\mu(x)$ and the variance function $\sigma(x)$ are both linear combinations\(^6\) of “basis functions” that take the following form:

\[
\begin{align*}
\mu(x) &= \sum_{i=1}^{m} \phi_i(x) w_i \\
\sigma(x) &= \exp \left( \sum_{i=1}^{m} \phi_i(x) \nu_i \right)
\end{align*}
\]

where $\{\phi_i(x), w_i, \nu_i\}_{i=1}^{m}$ are sets of $m$ basis functions and their associated weights, respectively.\(^7\) Basis function models (BFMs), for specific classes of basis functions such as the squared exponential, have the advantage of being universal approximators, i.e., there exists a function of that form that can approximate any function, with mild assumptions, to any desired degree of accuracy; i.e., a one size fits all function. BFM is a form of sparse GPs [23]. The most general form of the squared exponential is

\[
\phi_i(x) = \exp \left( -\frac{1}{2} (x - \mu_i)^T \Lambda_i^{-1} (x - \mu_i) \right).
\]

The goal of GPz essentially is to find the optimal parameters $\{\mu_i, \Lambda_i, w_i, \nu_i\}_{i=1}^{m}$, such that the mean and variance functions $\mu(x)$ and $\sigma(x)$ are the most likely functions to have generated the data—using Bayesian inference. These optimal parameters are found with a limited-memory Broyden–Fletcher–Goldfarb–Shanno algorithm [22] (a hill-climbing optimization algorithm for differentiable nonlinear problems). During the training stage of the algorithm, GPz infers the locations and spreads that describe the basis functions; during the validation stage, it infers the appropriate complexity of the model, essentially how many basis functions to use.

The key features introduced by GPz include: 1) implementation of a sparse GP framework, allowing the algorithm to run in $O(nm^2)$ instead of $O(n^3)$ (where $n$ is the number of samples in the data and $m$ is the number of basis functions); 2) a “CSL” framework where the algorithm can be tailored for the precise science goal; and 3) properly accounts for uncertainty contributions from both variance in the data as well as uncertainty from lack of data in a given part of parameter space (by marginalizing over the functions supported by the GP that could have produced the data).

Unless otherwise stated, we use the settings in Table I (see [6] and [7] for precise definitions and interpretations). GPz requires a very little fine-tuning. The most important parameter is $m$, the number of basis functions. A higher $m$ corresponds to higher model complexity and longer training times. Fig. 2 shows the algorithm performance as a function of $m$; the best performance is achieved for $m \approx 10 \sim 100$, in line with findings in [6] and [7].

GPz permits custom CSL, e.g., a specific science goal specified to the algorithm. In the application of calculating galaxy redshifts, this might typically be something like specifying that low-redshift galaxies are a low priority. In the ICF case, we can also set particular parts of parameter space of interest, e.g., areas around a cliff, parts of parameter space that are possible with a particular facility, and so on. In this article, we will use the weighting $w = \sqrt{Y}$ (where $Y$ is the simulated yield in a number of neutrons), up-weighting the cost of

\(^5\)It is in some sense hard to definitively know that this is the case until a real shot has been performed, so this statement must remain conditional without experimental data. However, the aforementioned design of [9] finds that the 1-D models are in reasonable agreement with the data; and the implosions described here should be “even more 1-D.”

\(^6\)Note that this is a different sense of linear combination to that which can be used in the definition of a GP. The definition of a GP involves the addition of random variables at different points; the linear combination discussed here is an addition of multivariable functions to build up a representation of $\mu(x)$ and $\sigma(x)$.

\(^7\)In practice, GPz actually uses $\sigma^2 = \exp \left( \sum_{i=1}^{m} \phi_i(x) \nu_i + b \right)$, where the addition of the bias term “$b$” is used for practical reasons but is not required theoretically.
A. Predictions and Optimal Design

The choice of $w = \sqrt{Y}$ was motivated by requiring a function that was: 1) monotonically increasing with $Y$ (so that higher yield parameter space has a higher weight than lower yield parameter space); 2) a power law (so that the weighting is invariant under rescaling of $Y$); and 3) has a “reasonable” dynamical range for the design space considered (here, yield spans $\sim 10^{10} - 10^{16}$, so the ratio of weightings of different parts of parameter space can reach up to $\sim \sqrt{10^{16}/10^{10}} = 1000$).

The 5-D space considered here is of relatively modest dimensionality, but GPz has been shown to be effective and fast running in $\gtrsim$10D (see [19] and Section III-E).

We also trialed our data with DJINN [2]. The DJINN solution to difficulties in designing NN structure is to use a novel mapping from decision tree to network structure, giving a very user-friendly algorithm that works with a very little human input in a wide variety of circumstances. Parameters used for DJINN are shown in Table II, with choices motivated by [2] (improved performance may be achievable with further parameter fine-tuning).

III. RESULTS

A. Predictions and Optimal Design

Fig. 3 shows the yield predicted by the ML algorithms for the test data compared with the Hyades yields. GPz is largely able to correctly predict the yield for most of the test data with reasonably realistic uncertainties. Fig. 4 shows the yield as a function of two of the parameters and the corresponding uncertainty on the predictions. The part of parameter space with maximal yield within design constraints (e.g., what is feasible for a given facility) can easily be extracted, or a similar stability test as in [3] can be used to find the design with the best combination of yield and stability. For example, say, we fix the drive at $T_{\text{peak}} = 300$ eV and $\sigma = 1.5$ ns and restrict interest to capsules smaller than 1.5 mm (e.g., $r_1 + r_2 < 1.5$ mm). The capsule with the highest predicted yield is easily found to be $r_1 \approx 1.2$ mm, $r_2 \approx 0.3$ mm, and $\rho \approx 25$ mg/cc, giving a yield of $Y \sim 4 \times 10^{15}$. However, suppose we were only interested in making sure that the design robustly had a yield above $Y = 10^{15}$ and instead wanted to minimize the capsule radius. We define $P(Y > 10^{15}|\delta = 0.2)$ as the fraction of designs that still have a yield above $Y = 10^{15}$ when each parameter is perturbed by an amount sampled from a Gaussian with a standard deviation of 20% (see [3]). We can find the design that minimizes $r_1 + r_2$ with $P(Y > 10^{15}|\delta = 0.2) > 0.9$, leading to a slightly different design, $r_1 \approx 0.95$ mm, $r_2 \approx 0.3$ mm, and $\rho \approx 25$ mg/cc ($Y \sim 2.5 \times 10^{15}$).

B. Cost-Sensitive Learning

Fig. 3 shows a comparison of GPz $w = 1$ (“normal” in [6]) and using CSL ($w = \sqrt{Y}$). Figs. 5 and 6 show the bias (mean of $Y_{\text{test}} - Y_{\text{prediction}}$) and root-mean-squared error (RMSE) on the predictions, illustrating that the performance in higher yield parts of parameter space ($\log Y \sim 15 - 17$) is indeed improved at the cost of performance in lower yield regions. In general, CSL can be linked to a specific science goal of a study, e.g., for the goal of achieving ignition at the NIF, we probably have a particular interest in having low RMSE close to cliff edges but are relatively insensitive to RMSE or bias both far above and far below this boundary. We would note that CSL is a method that can, in some circumstances, obtain slightly better statistical properties in certain parts of parameter space [20]; however, it cannot extract information from the data that

<p>|</p>
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m$</td>
<td>100</td>
<td>Number of basis functions (the $\phi_i$); complexity of GP</td>
</tr>
<tr>
<td>maxIter</td>
<td>500</td>
<td>Maximum number of iterations (comparisons with the validation data) permitted</td>
</tr>
<tr>
<td>maxAttempts</td>
<td>50</td>
<td>Maximum number of iterations to attempt if there is no progress on the validation set</td>
</tr>
<tr>
<td>method</td>
<td>GPVC</td>
<td>Type of bespoke covariances (the $\Lambda_i$ in equation 3) used on each basis function (see [7] for the different options)</td>
</tr>
<tr>
<td>normalize</td>
<td>True</td>
<td>Pre-process the input by subtracting the means and dividing by the standard deviations</td>
</tr>
<tr>
<td>joint</td>
<td>True</td>
<td>Jointly learn a prior linear mean-function (learn the function means and variances jointly)</td>
</tr>
<tr>
<td>heteroscedastic</td>
<td>True</td>
<td>Model noise as well as point estimates</td>
</tr>
</tbody>
</table>

TABLE I

PARAMETER SETTING OF GPz

TABLE II

PARAMETER SETTING OF DJINN

Fig. 2. Performance of the algorithm as a function of the number of basis functions used. The y-axis is the fraction of test data within 0.2 dex of the prediction.
Fig. 3. Surrogate yield and Hyades yield compared for different surrogate setups. GPz-Normal is shown in red (left), GPz-CSL in blue (center), and DJINN in green (right). The diagonal filled line shows equality, and the dashed lines show one dex discrepancies.

Fig. 4. 2-D slice through the 5-D parameter space (with the other three parameters set to $r_1 = 1$ mm, $T_{\text{peak}} = 300$ eV, and $\sigma = 0.5$ ns), showing the surrogates yield (left), the uncertainty on the prediction (center), and the uncertainty from lack of data divided by uncertainty from shot-to-shot variation (right).

simply is not there (e.g., a really extreme weighting scheme will still fail to improve predictions in parts of parameter space with almost no data).

C. Comparison With DJINN

We also applied (Bayesian) DJINN to our data, also shown in Figs. 3, 5, and 6. The intention is not to do a rigorous code comparison, as the codes were developed for different ML goals and it is nontrivial to directly compare model complexity, but simply to illustrate that GPs are comparably viable tools for building ICF surrogates.

DJINN also performs well in producing a surrogate model. The algorithm trained and predicted in $\sim 80$ and $\sim 2$ s, respectively, compared with $\sim 5$ and $\sim 0.02$ s for GPz on a laptop.9

Some authors have found that the best results are achieved using a committee of a variety of ML methods, so it is the possible best results that could be achieved using a combination of neural nets and GPs with an ensemble averaging [24] or a hierarchical Bayesian approach [20]. This would also start a move toward non-Gaussian pdfs, as predictions near cliffs are likely to be multimodal (an experiment either ignites or it

916-GB RAM, 3.1-GHz Intel Core i5.
does not). Greater precision in key parts of parameter space can, of course, also be achieved by doing more simulations in that part of parameter space, but that requires advanced knowledge of which part of parameter space is interesting. Future work could couple the process of sampling parameter space and building the surrogate. It might also be interesting to consider how the CSL-like methods might be implemented within DJINN, e.g., loss-calibrated learning as per [25].

D. Uncertainty Decomposition

Fig. 4 (right) shows \((v/\beta_s^{-1})^{1/2}\), the ratio of uncertainty from lack of data to uncertainty from intrinsic variation. \(v\) is the variance from lack of data, defined as \(v = \phi(x)\Sigma^{-1}\phi(x)^T\), where \(\phi\) is the vector of nonlinear basis functions that the prediction mapping is constructed from, \(x\) is the test data, and \(\Sigma\) is the covariance matrix of uncertainty on the weights applied to the basis functions when constructing the posterior mean. It is essentially the uncertainty on \(\mu(x)\) in (1). \(\beta_s^{-1}\) is the input-dependent noise variance. It is essentially \(\sigma(x)^2\) in (2). The total variance is \(\sigma^2_{\text{total}} = v + \beta_s^{-1}\). See [7, eqs. (3.13) and (5.10)] for a more in-depth explanation of the calculation and interpretation of these quantities and [26] for a more general background. The plot shows that GPz correctly identifies that for \(\rho < 10\) mg/cc, more of the uncertainty is coming from lack of data rather than shot-to-shot variation (as there were no simulations done in that part of parameter space). This shows that GPz can help understand what is the dominant source of uncertainty in different parts of the parameter space. This uncertainty decomposition does, however, come with the caveat that uncertainty is still likely underestimated at the edge of the domain and anywhere far from the data, due to the use of only a finite number of basis functions. Estimates of shot-to-shot variation can also be incorporated into design optimization, e.g., suppose we want to find the design with a yield above \(Y = 10^{15}\), but with minimal shot-to-shot variation, we can find the design with the smallest \(\beta_s^{-1}\) value that meets this yield criterion, which is \(r_1 \approx 1.05\) mm, \(r_2 \approx 0.15\) mm, and \(\rho \approx 15\) mg/cc (for the same drive as in Section III-A).

The shot-to-shot variation considered here is not quite identical to the real problem in ICF; here, we added scatter to the thousands of simulations, whereas more realistically, we might have a large number of simulations with no scatter/uncertainty and then just a few experiments with some shot-to-shot variation. Nonetheless, an approach similar to that detailed here could be effective in folding shot-to-shot variation into a surrogate building. For example, one might simulate a large number of shots varying both parameters that the designer controls (e.g., capsule shell thickness) as well as ones they do not (e.g., imperfections on the surface of the shell). The outputs from these simulations could then be used to build a “noisy” surrogate as a function of the controlled parameters. Another useful feature is that understanding the shot-to-shot variation permits the model to say that there is a substantial amount of uncertainty in the prediction of the yield for an individual shot, but also to say that there is no need to do any further experiments in that part of parameter space, as all the uncertainty is from shot-to-shot variation. Features of GPz in development that could be useful for ICF surrogate building in the future include: 1) incorporating uncertainty on predictions due to uncertainty on input parameters (e.g., if there are error bars associated with the gas fill densities and shell thicknesses) and 2) coping with incomplete data (e.g., training on the data where a subset of the shots do not have some of the experimental properties recorded).

E. Scalability

Thus far, we have explored the applicability of GPz to building the ICF surrogates in the context of the Simplest Design, as a possible pathway to extremely robust implosions. However, other ICF design space data sets, in general, might have more complex features, in particular: 1) sharp ignition cliffs; 2) higher dimensionality; and 3) a larger number of simulations. Future work will investigate more fully the performance of GPz in more complex design spaces. Here, however, we briefly consider a simple analytic model, to see if GPz is likely to have the capacity to scale well (for example, the simple analytic model considered in [4, Sec. 2]). We consider a 10-D hypercube with sides going from 0 to 1 and sample 100000 points randomly in this volume (each dimension sampled uniformly independently). We calculate a mock-yield of \(Y = r^5(1 + 100000 \times (1 + \text{erf}(10 \times (r - 2))))\), where \(r\) is the radius from the origin in this 10-D space (the error function giving an ignitionlike cliff). We then train GPz (with \(m = 100\) basis functions; the same settings as in Table I used except heteroscedastic = false) and DJINN on 90% of these data (90000 points) and test on the other 10% (10000 points); the results are shown in Fig. 7. Training time was 415 s for GPz and 450 s for DJINN. Predicting time was 0.1 s for GPz and 7 s for DJINN. Both capture the design space well, with each doing slightly better or worse at different aspects of the prediction, e.g., GPz has a few outliers that DJINN does not, but it is less biased for high and low \(r\). Which statistical properties are most desirable are in general likely to depend on the specific science goal at hand. In summary, it appears that GPz should be able to perform well
and run in reasonable amounts of time for higher dimensional spaces with larger quantities of data when there are steep cliffs in the design space.

IV. CONCLUSION

Our results show that Gaussian processes have the potential to be useful ICF surrogates, and that, in particular, GPz is shown to be effective for the task as an easy-to-use algorithm that can cope with huge amounts of data in a high number of dimensions with realistic uncertainties. In particular, GPz may be useful for modeling shot-to-shot variation alongside uncertainty from lack of data in integrated experiments.

Future work will seek to combine experiment and simulation, either through transfer learning as per [4], or possibly through scaling the surrogate as per [27], taking care to understand biases induced by differences between the target distribution and the training set.

Key Findings From This Study:

1) GPz can make highly effective surrogate models for predicting the outcomes of ICF experiments.

2) CSL can help improve the statistical properties of predictions in the most important parts of parameter space.

3) It is possible to distinguish between uncertainty from shot-to-shot variation and uncertainty from lack of data.

4) The Simplest Design should be able to produce yields of order $10^{15}$ neutrons extremely robustly on NIF.

ACKNOWLEDGMENT

The authors would like to thank W. Garbett at the Atomic Weapons Establishment, Aldermaston, K. McGlinchey and J. Chittenden at Imperial College, London, J. Gaffney, L. Peterson, K. Humbird, and B. Spears at the Lawrence Livermore National Laboratory, and C. Bridge at the Massachusetts General Hospital for useful discussions.

REFERENCES


Fig. 7. Top: GPz (red) and DJINN (green) predictions as a function of radius for our analytic model, with the analytic result shown with a black line. Note that these are results in 10-D; we plot the 1-D $r$ and $Y$ values of the test data for ease of visualization. Middle: RMSE on the predictions; GPz-Normal shown in red (solid line) and DJINN in green (dotted line). Bottom: RMSE on the predictions; GPz-Normal shown in red (solid line) and DJINN in green (dotted line).
Peter Hatfield is an astrophysicist and plasma physicist at the University of Oxford. His doctoral research focused on modeling the large-scale structure of the Universe and he is now a postdoctoral researcher working on machine learning for laboratory astrophysics, astronomy and inertial confinement fusion.

Steven Rose is currently a Professor with Imperial College London, London, U.K., and the University of Oxford, Oxford, U.K., where he is also the Co-Director of the Center for Inertial Fusion Studies and was the Vice-Dean of the Faculty of Natural Sciences from 2011 to 2014. His current research interests include plasmas produced using high-powered lasers, with special interest in astrophysical plasmas and inertial confinement fusion.

Robbie Scott received the Ph.D. degree from Imperial College London, London, U.K.

He has spent a year at the Lawrence Livermore National Laboratory, Livermore, CA, USA. He is currently a Theoretical Plasma Physicist with the Central Laser Facility, STFC Rutherford Appleton Laboratory, Didcot, U.K. His current research interests include inertial confinement fusion (including fast ignition), laser-plasma interactions, and future neutron sources.

Ibrahim Almosallam received the D.Phil. degree in machine learning from the University of Oxford, Oxford, U.K., in 2017.

He was an Assistant Research Professor with King Abdullah City for Science and Technology, Riyadh, Saudi Arabia. He is an experienced Researcher with more than ten years of experience in researching artificial intelligence and machine learning.

Stephen Roberts is currently the Royal Academy of Engineering/Man Group Professor of machine learning with the University of Oxford, Oxford, U.K., and the Co-Founder of the Oxford machine learning spin-out company, Mind Foundry, Oxford. His research interests include the theory and methodology of machine learning for large-scale real-world problems, especially those in which noise and uncertainty abound. He has successfully applied these approaches to a wide range of problem domains, including astronomy, biology, finance, engineering, control, sensor networks, and system monitoring. His current major interests include the application of machine learning in the physical sciences, ecology and environmental monitoring, engineering industry, and finance.

Matt Jarvis was with the University of Hertfordshire, Hatfield, U.K., and the University of Leiden, Leiden, The Netherlands. He is currently a Professor of astrophysics with the University of Oxford, Oxford, U.K., and a fellow of St Cross College, Oxford. He is an Adjunct Professor with the University of the Western Cape, Bellville, South Africa. He is also the Cosmology Lead of the Oxford Hintze Center for Astrophysical Surveys, University of Oxford, and has extensive experience across galaxy evolution, cosmology, astronomical surveys, and data science.