Abstract

Finite elements provide accurate calculations of stress intensity factors, at the expense of significant computational cost. In contrast, handbook solutions, such as the Raju-Newman equations, provide a rapid prediction of stress intensity factors, retaining much of the accuracy from finite elements. This research employs Bingo to automate the generation of handbook solutions. Adopting a mechanics-based approach, the stress intensity factor solution is decomposed into multiple easily understandable functions. This methodology enables the Bingo models to surpass both the Raju-Newman equations and black box models in terms of accuracy and evaluation efficiency.

Keywords: Interpretable Machine learning, GPSR, semi-elliptical surface crack, mechanics based training

1. Introduction

Finite element analysis (FEA) is effective at computing stress intensity factors (SIFs) for cracks in complex geometries. However, despite its accuracy, FEA has notable limitations. One significant drawback is its demanding computational requirements, as well as the need for a skilled analyst to achieve reliable results. These factors lead to added time and resource investments, especially for fatigue and uncertainty assessment. To address these challenges, engineers in industry rely on handbook solutions. These solutions offer a practical and efficient way to estimate SIFs for idealized and parameterized crack scenarios and, when applied correctly, they can yield accurate results.

The Raju Newman equations are a widely used set of handbook SIF solutions for various crack shapes and loading conditions [1]. These handbook are easy to implement in existing workflows as they are closed-form solutions. However, their ease of use sometimes leads to their application in situations that do not align well with the assumptions and limitations of the original idealized model, resulting in inaccurate predictions. This research focuses on the Raju-Newman solution for a semi-elliptical surface crack in a finite plate subjected to mode-I tension. Since its introduction alternative models have been proposed that allow for more complex loading conditions such as the models created by [2, 3]. The model crated by Wang and Lambert Follows the same approach used in the Raju-Newman equations of breaking the model into multiple sub-functions. They modified these sub-functions to allow a linear stress function [2]. Pommier, et al. dropped the useage of Raju-Newman like subfunctions instead opting for a function that directly calculates the boundary [3]. **********

Machine learning (ML) can play a pivotal role in automating the creation of SIF solutions for more accurate models and more complex geometries. Notably, ML has been successfully employed to produce accurate SIF solutions, as documented in studies such as in [4, 5, 6, 7, 8]. The advantage of ML over other techniques, like those found in [1, 3, 2], is its ability to be trained automatically, resulting in repeatable accurate models. It is worth noting that

many commonly used ML algorithms generate "black-box" models, which lack inherent interpretability. In engineering, inherent interpratability in the form of close-form equations allows engineers to easily implement these models into existing modeling codes. inherent interpretability also adds a level of trust to the models. This is why handbook solutions, such as those in [1], continue to be used since they inherently offer interpretability.

ML models have been developed using black box methods as demonstrated by [6, 7, 8]. These ML models, utilizing methods like Gaussian processes regression and neural networks, offer higher accuracy but produce less interpretable models. Notably, the ML models mentioned could predict SIFs at only a single point along the crack front, whereas the Raju-Newman equations can estimate SIFs across the entire crack length. **********

While ML models provide improved predictive capabilities, their lack of interpretability can be a drawback in certain contexts, especially when it comes to explaining and trusting the predicted SIF. The interpretability of the Raju and Newman equations gives engineers a clear understanding of how SIFs are derived, aiding decision-making and instilling confidence in the results. Raju and Newman devised a method of breaking down the SIF solution into multiple sub-functions, where each sub-function modifies an analytically solvable case. In the case of a semi-elliptical surface crack the analytical solution to an embedded ellipse in an infinite volume is used. Raju and Newman's approach is detailed in the background section. The method for breaking down the equation provides a framework that can be applied to other crack cases, enhancing the inherent explainability of an analytical solution.

This research will utilize an interpretable machine learning code Bingo, developed by researchers at NASA and the University of Utah [9]. Bingo generates closed-form mathematical expressions to predict SIFs along the entire crack length, offering improved accuracy and simplicity compared to the Raju-Newman equations while preserving interpretability. Thanks to the closed-form nature of Bingo's models, they can be easily integrated into existing Linear Elastic Fracture Mechanics (LEFM) software applications like NASGRO, AFGROW, and SMART|DT [10, 11, 12].

final paragraph **********

1.1. Background

In linear elastic fracture mechanics (LEFM), the exact solution for the SIF along an elliptical crack in an infinite volume is given by the equation:

$$K_{ee} = \sigma\sqrt{\frac{\pi a}{Q}}\left(\sin^2\phi + \frac{a^2}{c^2}\cos^2\phi\right)^{1/4} \text{ for } a \leq c, \tag{1}$$

where a represents the length of the minor axis of the ellipse, c represents the length of the major axis, σ is the far-field stress, ϕ is the parametric angle of the ellipse, and Q is the square of the complete elliptic integral of the second kind. This expression is only valid when $a \leq c$. In the context of a semi-elliptical surface crack, the values of a and c differ from the minor and major axes, instead representing the crack depth and half-crack surface length, respectively, as illustrated in Fig. 1 [13]. To accommodate cases where a/c will exceed 1, Raju and Newman employed a modified version of equation 1. This modification, detailed in Eq. 2, allows for the use of all values of a/c.

$$f_\phi = \begin{cases} \left[\left(\frac{a}{c}\right)^2 \cos^2\phi + \sin^2\phi\right]^{1/4} & \text{if } \frac{a}{c} \leq 1 \\ \\ \left[\cos^2\phi + \left(\frac{c}{a}\right)^2 \sin^2\phi\right]^{1/4} & \text{if } \frac{a}{c} > 1, \end{cases} \tag{2}$$

Using Eq. 1 with Eq. 2 results in Eq. 3

$$K_{ee} = \sigma \frac{\sqrt{\pi a}}{E} f_\phi. \tag{3}$$

Using Eq. 3 and multiplying by the correction factors f_w, M, and g leads to Eq. 4,

$$K = \frac{\sqrt{\pi a}}{E} f_\phi f_w M g, \tag{4}$$

which models SIFs along the front of a semi-elliptical surface crack in a finite plate. The finite width correction factor f_w,

$$f_w = \sqrt{\sec\left(\frac{\pi c}{2b}\sqrt{\frac{a}{t}}\right)}, \tag{5}$$

is a 3D extension for the equation developed in [14], which accounts for the finite width and thickness of the plate when $a/c = 1$ and $\phi = \pi/2$. The thickness of the plate is denoted as t and the half-width of the plate is denoted as b Fig. 1. The function M accounts for changes in SIF due to the aspect ratio of the crack at the point along the crack where $\phi = \pi/2$ and is given by

$$M = M_1 + M_2\left(\frac{a}{t}\right)^2 + M_3\left(\frac{a}{t}\right)^4, \text{ where} \tag{6}$$

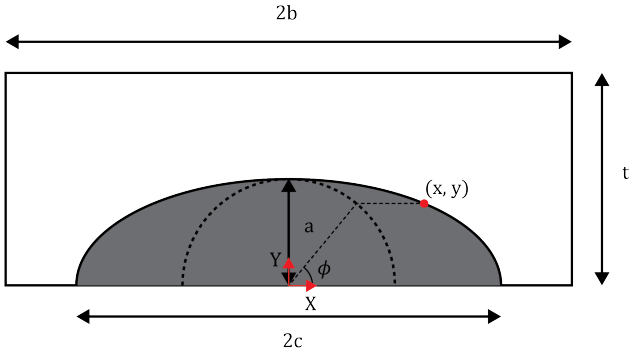
$$M_1 = \begin{cases} 1.13 - 0.09\left(\frac{a}{c}\right) & \text{if } \frac{a}{c} \leq 1 \\ \\ \sqrt{\frac{c}{a}}\left(1 + 0.04\frac{c}{a}\right) & \text{if } \frac{a}{c} > 1 \end{cases} \tag{7}$$

$$M_2 = \begin{cases} -0.54 + \frac{0.89}{0.2 + \left(\frac{a}{c}\right)} & \text{if } \frac{a}{c} \leq 1 \\ \\ 0.2\left(\frac{c}{a}\right)^4 & \text{if } \frac{a}{c} > 1 \end{cases} \tag{8}$$

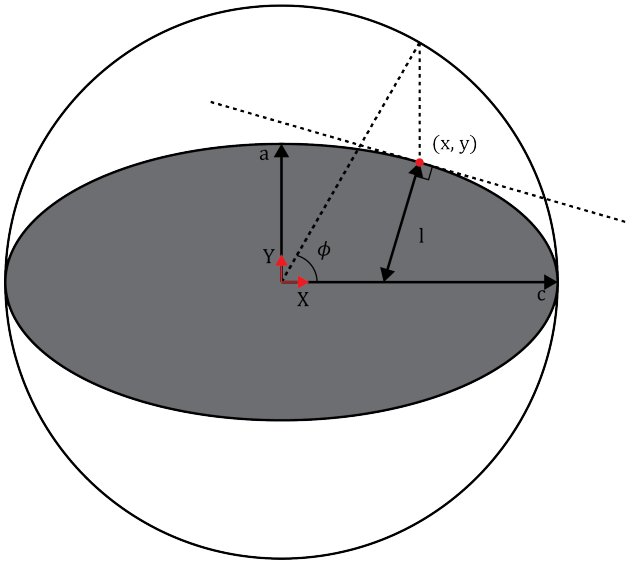
$$M_3 = \begin{cases} 0.5 - \frac{1}{0.65 + \frac{a}{c}} + 14\left(1 - \frac{a}{c}\right)^{24} & \text{if } \frac{a}{c} \leq 1 \\ \\ -0.11\left(\frac{c}{a}\right)^4 & \text{if } \frac{a}{c} > 1. \end{cases} \tag{9}$$

The function g corrects for free surface effects and is denoted by

$$g = \begin{cases} 1 + \left[0.1 + 0.35\left(\frac{a}{t}\right)^2\right](1 - \sin\phi)^2 & \text{if } \frac{a}{c} \leq 1 \\ \\ 1 + \left[0.1 + 0.35\left(\frac{c}{a}\right)\left(\frac{a}{t}\right)^2\right](1 - \sin\phi) & \text{if } \frac{a}{c} > 1. \end{cases} \tag{10}$$



(a) Crack dimensions



(b) Ellipse dimensions

Figure 1: (a) Crack parameters with a being the crack depth and $2c$ being the surface crack length. (b) ϕ is defined by the angle to the inscribed circle projected to the ellipse. l is defined as the distance perpendicular to the tangent line from the point of interest to the nearest axis.

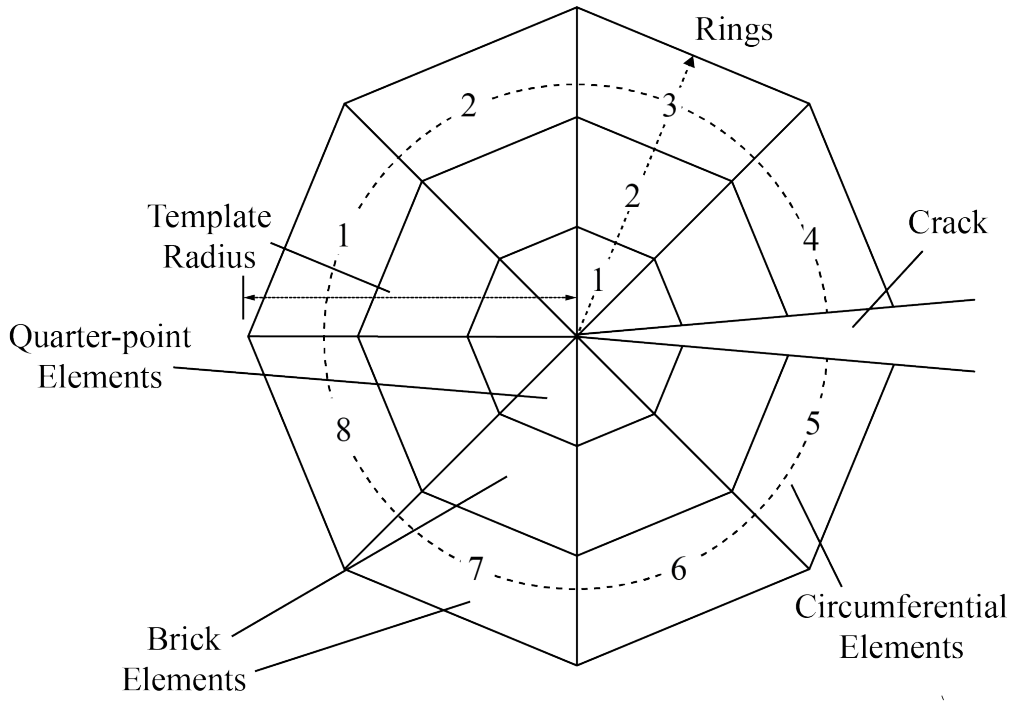


Figure 2: FRANC3D crack mesh containing 8 circumferential elements and 3 rings. The template radius is the radius of an inscribed circle

The function g is sinusoidal, having a value of 1 at $\phi = \pi/2$. By using this methodical approach of breaking down the problem into sub-functions that each account for a different aspect of the crack, allowed Raju and Newman to develop accurate equations that build upon the explainablity from the analytical solution of the embedded ellipse.

2. Methods

2.1. Computational Fracture Mechanics

The stress intensity factors used in this research were simulated using finite element analysis (FEA) using FRANC3D and Abaqus software [15, 16]. Abaqus served as the primary tool for creating the geometry, initial mesh discretization, application of BCs, and solution. FRANC3D was employed for tasks related to crack insertion and SIF computation. Abaqus' Python interface was used to generate the model geometries.

A local-global sub-modeling approach was employed within FRANC3D which involved dividing the geometry into two components: the global model, which encompassed the boundary conditions, and the local model, containing the region where the crack would be inserted. The local model is used in FRANC3D, where the crack is inserted using a crack front template consisting of rings of hexagonal elements, with an inner ring of quarter-point elements surrounding the crack front, allowing for SIF computations Fig. ??. After crack insertion the local model was re-meshed, preserving the nodal locations on the cut faces (i.e. local-global boundary) for later coherency with the global model. The complete model, with the inserted crack, was subsequently solved using Abaqus. This process is illustrated in Fig. **??**.

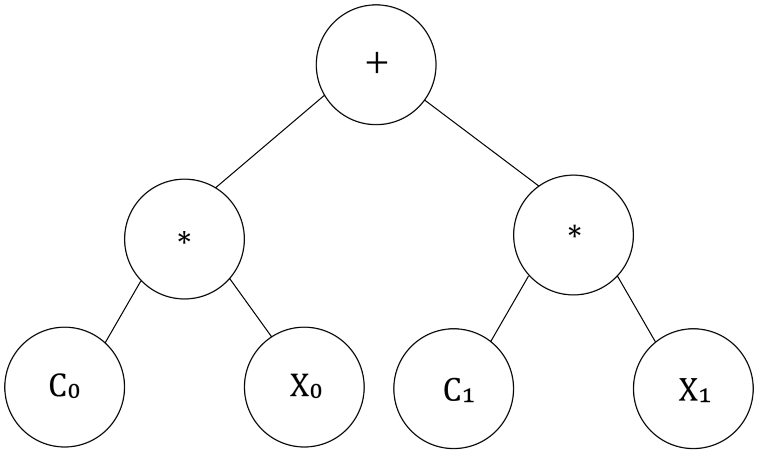


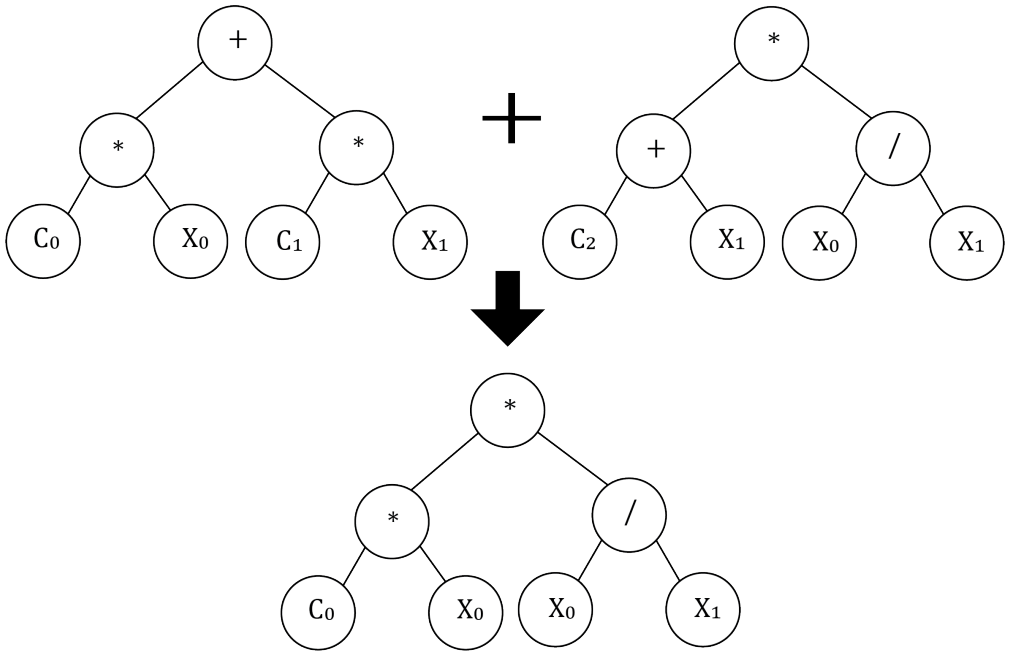
Figure 3: Example Agraph for for equation; $C_0X_0 + C_1X_1$, where the complexity is the sum of the nodes in this case 7.

Integral methods are geometrically more accurate than displacement methods, such as displacement correlation, for the computation of SIFs. The J-integral, developed by Rice [17], is a commonly used method for SIF calculation. However, it has a limitation, it cannot separate the SIFs into the three cracking modes, except for very simplified crack geometries, as noted by Banks-Sills et al. [18]. Banks-Sills et al. developed the M-Integral formulation to overcome this limitation allowing for the calculation of accurarate SIFs for all three cracking modes [18]

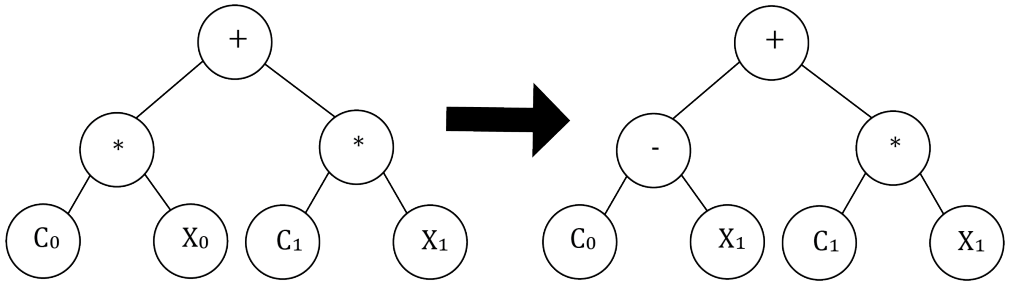
2.2. Genetic Programming Based Symbolic Regression

Symbolic Regression (SR) is a machine learning technique aimed at discovering closed-form analytical expressions that model training data and known physics [**?** **?**]. Presently, the most effective optimization approach for SR is genetic programming (GP) [19]. The implementation of genetic programming based symbolic regression (GPSR) employed in this research is an open-source Python package, Bingo [9]. Bingo enables learning real-valued mathematical expressions represented as acyclic graphs (Agraphs). The equation's complexity is determined by the number of nodes in the Agraph, Fig. **??** shows an Agraph for the equation $C_0X_0+C_1X_1$, that has a complexity of 7 corresponding to the 7 nodes in the Agraph. The training begins with an initial assortment of randomly generated equations, varying in complexity. During each generation, equations undergo randomized evolutions, employing a combination of crossover and mutation. Crossover entails exchanging nodes from two Agraphs along with their associated branches and leaves, generating two evolved equations. Mutation introduces random changes to nodes in the Agraph as shown in Fig. 4 [20]. After crossover and mutation has occurred the equations are ranked based on a defined fitness function e.g. MAE, RMSE, etc. The most fit equations are then used in the next generation where the cycle continues until a fitness threshold or maximum number of generations has been satisfied.

The most fit equations, at various complexity values, from the final population are taken and presented as a Pareto front Fig. **??**. Equations with high complexity have better fitness values; however, they often have reduced interpretabiliy from the engineer's perspective.



(a)



(b)

Figure 4: (a) crossover, where two parent graphs combine features to create a child graph. (b) where part of a graph is randomly changed creating a new graph.

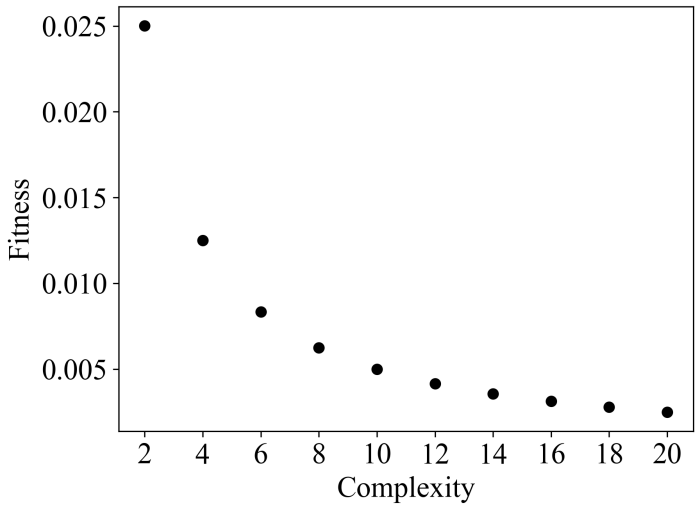


Figure 5: Example Pareto front where the horizontal axis is the model complexity and the vertical axis is the model fitness. Each point on the graph represents a different model in the population.

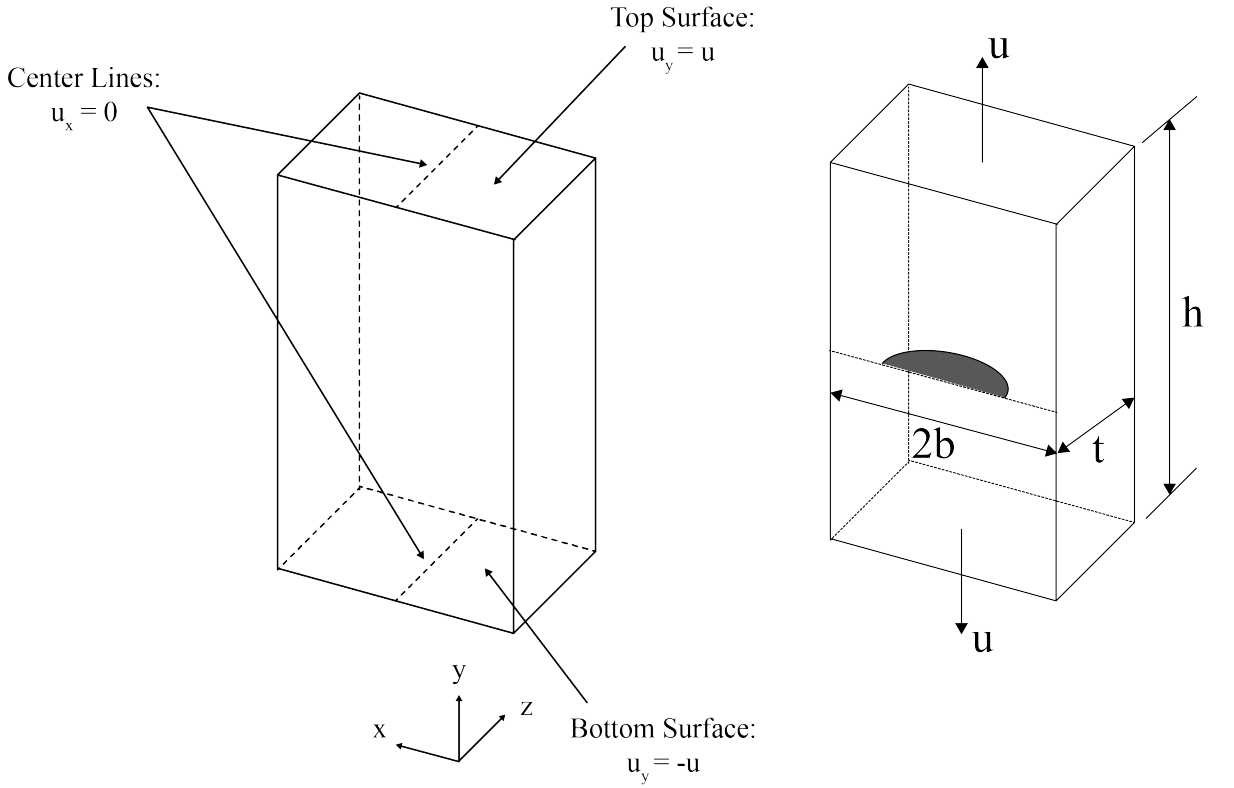
Equations with low complexity tend to be easily interpreteble at the expense of decreased fitness. Lacking additional criteria, a user must decide what equation is the best for the application. Typically, there is a point at which increasing the complexity yields little gain on fitness Fig. ??, which serves as a balance between accuracy and complexity. ***anotate plot****

3. Numerical Experiments

3.1. Training Data Generation

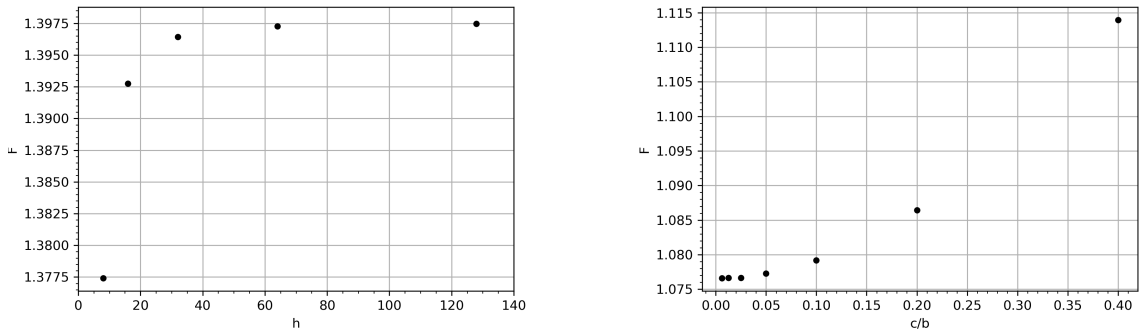
The finite element models were designed to be as consistent as possible with the FE models created by Raju and Newman [1], while using modern software, enabling the comparison between GPSR and the equations created in [1]. In [1], Raju and Newman neglected effects due to finite height by assuming the height of the plate to be large relative to the thickness of the plate. To be consistent with this assumption, a constant value of $h/t = 64$ was chosen to remove any effects due to the height of the plate as shown in figure 7a. The cracked finite element models are entirely defined by the variables a, c, b, t, h, and u shown in figure 6. The far field stress σ is extracted from the reaction forces at the top and bottom surfaces. The variables a and c are crack parameters and the variables b, t, and h define the geometry of the plate. A constant displacement of $u = 0.001$ units was applied on the top and bottom surfaces in the direction of the surface normals. Additionally the center-lines of the top and bottom surfaces were held fixed in the x-direction to removed rigid body motion. While a single point was held fixed in the y-direction. This application of BCs most accurately reproduced the conditions imposed in the quarter-symmetry model of Raju and Newman [1].

The crack mesh template created by FRANC3D is shown in **??**. The mesh contains a ring of quarter points elements around the crack front followed by rings of brick elements. The



(a) Applied boundary conditions (b) Model geometry

Figure 6: (a) Top and bottom surfaces with displacement in $+y$ and $-y$ respectively. The center lines of the top and bottom faces are held at zero displacement in x. (b) Model geometry plate height, h, plate width, $2b$, and plate thickness, t.



(a) Convergence of h/t (b) Convergence of c/b

Figure 7: (a) The convergence of h/t is plotted using the mean *******boundary correction factor. (b) The convergence of c/b is plotted using the mean boundary correction factor.

Feature	min	step	max
a/c	0.2	0.05	2
a/t	0.2	0.05	0.8
c/b	0.01	0.05	0.4
ϕ	0	$\pi/150$	π

Table 1: Max and min values for each of the features used when creating the cracked FE models

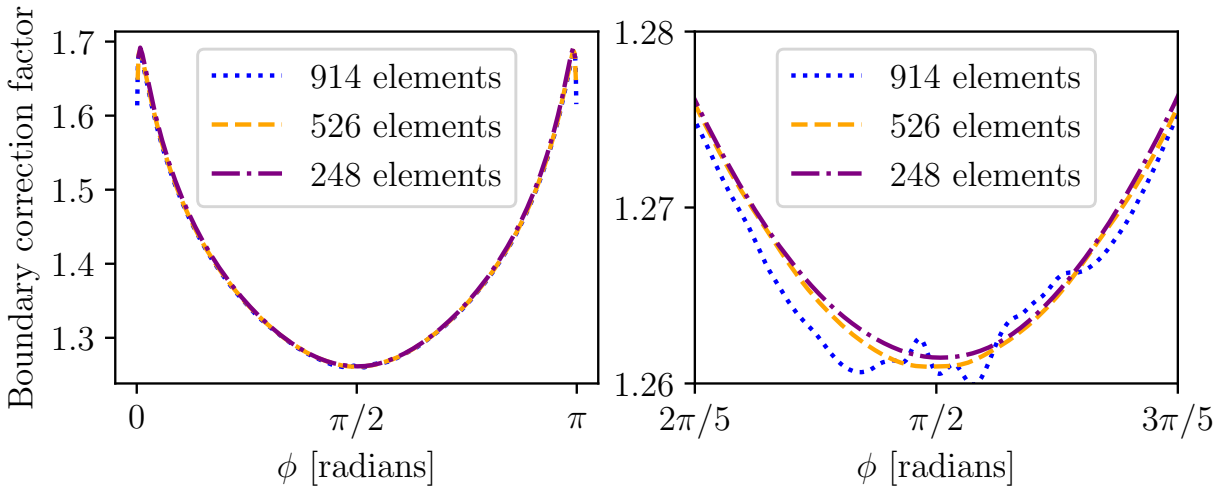


Figure 8: (left) entire SIF plot for differing number of elements along the crack front. (right) plot zoomed in at $\pi/2$ to highlight numerical noise with high element count.

parameters for the crack mesh are the template radius which specifies the size of each element; the number of rings of elements around the crack; and the number of circumferential elements in each ring. Instead of using a fixed template radius a number of elements along the entire crack was specified. This produced better results with varying crack aspect ratios. The best SIF values were computed with 300-600 elements along the crack, 8 rings of elements, and 14 circumferential elements per ring as shown in table 2. The average SIF value for each model is relatively insensitive to the crack mesh, however certain crack geometries display numerical noise for certain crack mesh sizes as shown in figure ??. The models that displayed the most numerical noise for the most crack sizes were the models with values close to the following: $a/c = 0.5$, $a/t = 0.6$, and $c/b = 0.2$. The crack mesh parameters were chosen based off of these models and then applied to the rest of the models. Before running the entire set of training simulations, a random sample of 10% of the total models was computed to check for convergence. After convergence was confirmed on those models the entire training set was computed.

The converged values are tabulated in table 2

After the converged model parameters were found all 2500 of the models were run to calculate 300-450 SIFs per model. Figure ?? shows the calculated SIF values from a few of the models compared to the SIFs calculated from [21].

Thickness	1
h/t	64
Global seed	max(0.25, min(b/50, 0.5))
Local seed	0.25
Number of rings	8
Number of circumferential elements	14
Number crack front elements	∼300

Table 2: Optimized model and mesh values

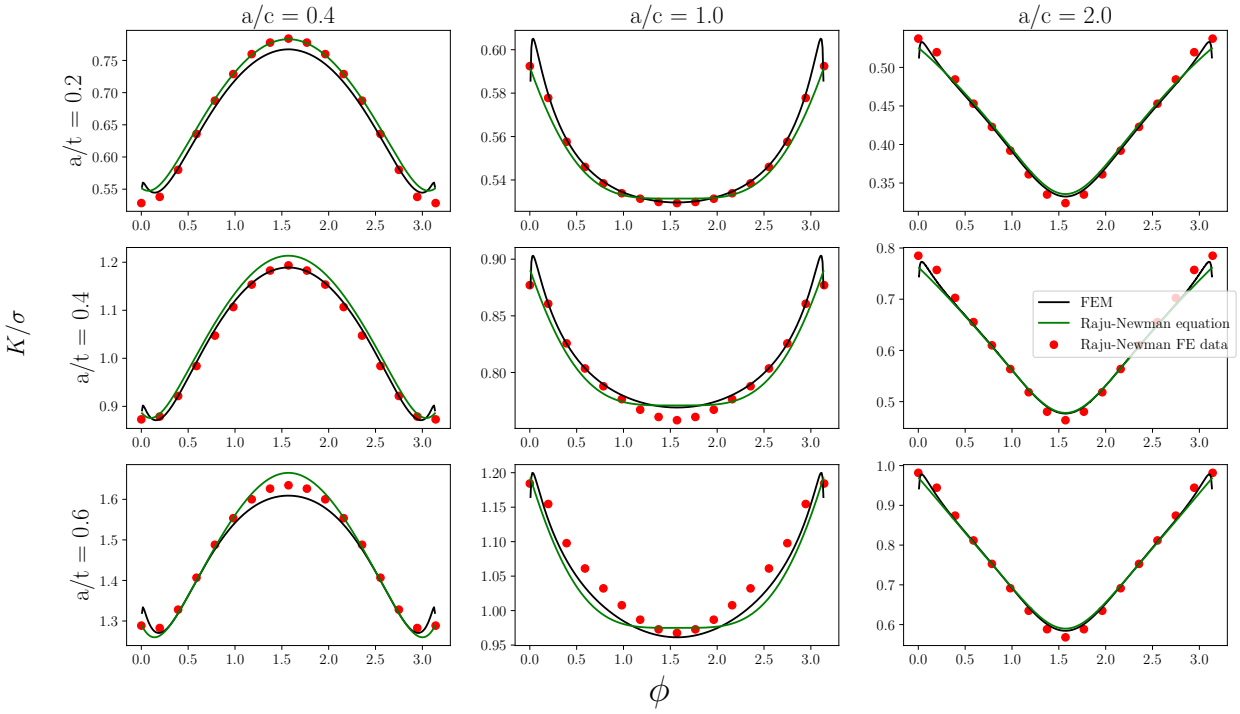


Figure 9: SIF values from a selection of models plotted with the Raju-Newman equation.

3.2. Training Data Subdivision

The 2500 training simulations were used for training and model selection, while a additional isolated set of 400 models were generated for testing to allow for better generalization error assessment. Instead of simply training on all of the SIF data at once we use a mechanics based approach to improve the training process and the explainablity of the final produced GPSR SIF model. The general approach used by Raju and Newman [1] was used to decompose the model for K into two parts: an a priori known part and a learned part. The known part includes Eq. 1, which is the solution to the embedded ellipse in an infinite volume. The learned part of the model are the boundary correction factors which are f_w, M, and g each accounting for a specific correction to the known part. The finite width correction factor, f_w, accounts for the effect of finite width and thickness at $\phi = \pi/2$ in a semi-circular crack. M accounts for the aspect ratio of the crack. The two functions f_w and M together comprise the boundary correction factor at $\phi = \pi/2$ along a semi-elliptical crack in a finite plate. Finally g applies this correction at $\phi = \pi/2$ along the crack front allowing for SIFs to be modeled along the entire crack front.

Two modifications were made to the approach used in [1] to both simplify and generalize the training process. First, we extend the definition of the major and minor axes of an ellipse by using the function l from [13] and modifying it to allow for values of $a/c > 1$ Eq. 11.

$$l = a\left(\frac{c}{\alpha}\right)^2 \sqrt{\left(\frac{a}{c}\right)^2 \cos^2\phi + \sin^2\phi}, \tag{11}$$

where a is the crack depth, c is half-crack surface length, and α is the length of the major axis of the ellipse. The function l is a measure of the perpendicular distance from the tangent line at the point of interest to the nearest axis seen in figure 1. The benefit of modifying the embedded ellipse equation by using equation 11 is that it removes the need for piece-wise functions, Eqns. 2 and **??**. This approach reduces the number of models that have to be trained and lowers the complexity of the final K solution resulting in a more interpretable equation.

The second modification is the formulation of f_w. While Raju and Newman [1] used a finite width correction factor for a 3d through crack from [14], this work uses a finite width correction factor for the case of a semi-elliptical surface crack. f_w is still a finite width and thickness correction when $\phi = \pi/2$ and $a/c = 1$. However, unlike Raju and Newman [1], by noting that when $\phi = \pi/2$ and $a/c = 1$ the value of K converges to the embedded ellipse equation as a/t and c/b go to zero, Fig **??**.

The final equation for K is then

$$f_w\left(\frac{c}{b}, \frac{a}{t}\right) = \frac{K\left(\frac{a}{c} = 1, \phi = \frac{\pi}{2}\right)}{\sigma \frac{2}{\pi}\sqrt{\pi a}}, \tag{12}$$

$$K = \sigma \frac{\sqrt{\pi l}}{E} f_w M g, \tag{13}$$

where f_w, M, and g are the boundary correction functions learned by using Bingo and $\sigma\sqrt{\pi l}/E$ is the modified embedded ellipse solution.

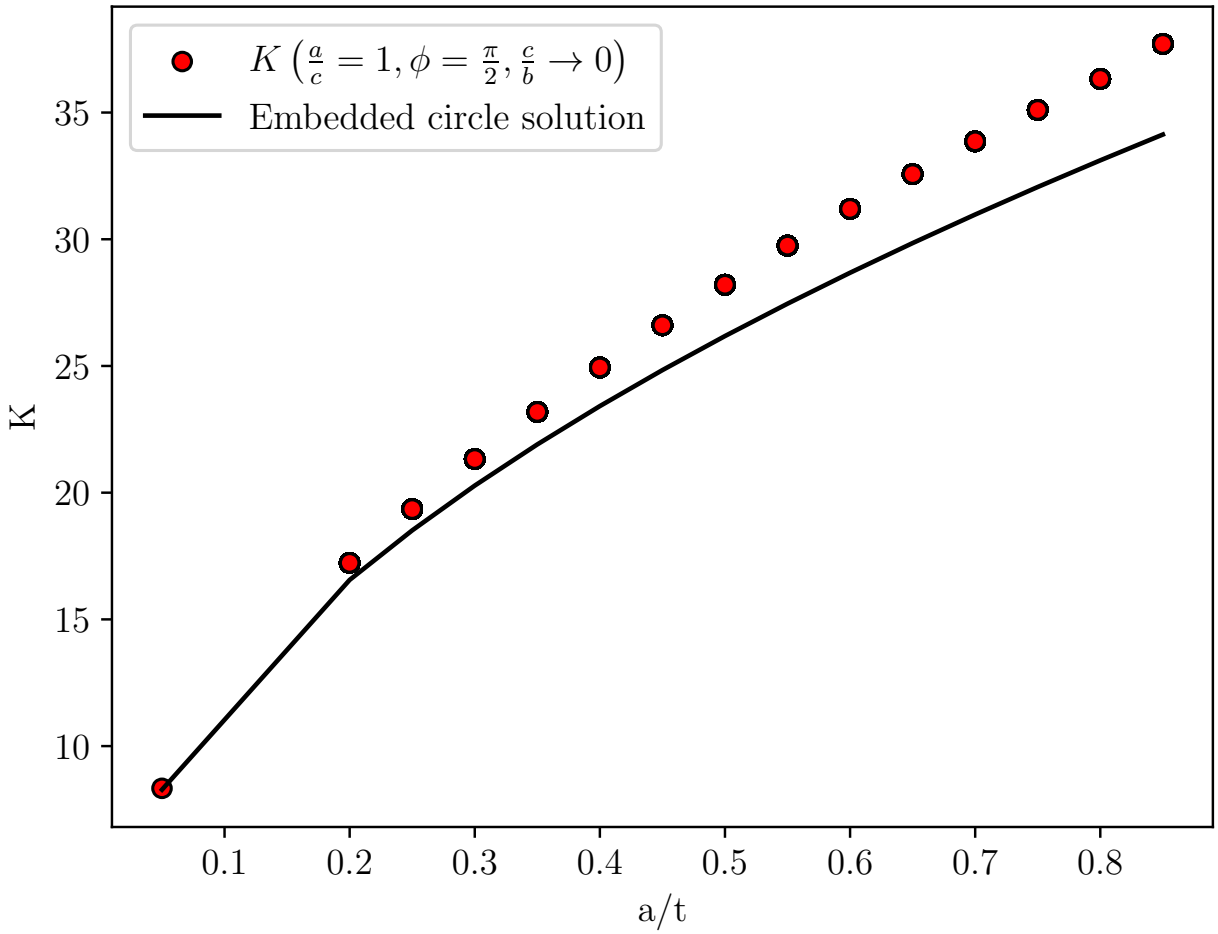


Figure 10: Comparison of K from the center of a surface crack and the solution to an embedded circle crack in an infinite volume as a function of a/t where c/b = 0.01. 7

A decomposition of training data will now follow the defined correction functions and they dependence on geometric variables as

$$K\left(\frac{a}{c},\frac{a}{t},\frac{c}{b},\phi,\sigma\right)=K_{ee}\left(\frac{a}{c},\sigma,\phi\right)f_w\left(\frac{a}{t},\frac{c}{b}\right)M\left(\frac{a}{t},\frac{a}{c}\right)g\left(\frac{a}{t},\frac{a}{c},\phi\right) \tag{14}$$

The next step is to extract the training data for M, which is computed similar to f_w. Specifically, The SIF training data from FEA simulations are taken at $c/b=0$ and $\phi=\pi/2$ then normalized by Eqns 1 and 12 as

$$M\left(\frac{a}{t},\frac{a}{c}\right)=\frac{K_{FE}\left(\frac{c}{b}\to 0,\phi=\frac{\pi}{2}\right)}{f_w\left(\frac{c}{b}\to 0\right)K_{ee}\left(\phi=\frac{\pi}{2}\right)}. \tag{15}$$

The final step is to extract the training data for g. This is done by normalizing by Eqns?? as

$$g\left(\frac{a}{t},\frac{a}{c},\phi\right)=\frac{K_{FE}\left(\frac{c}{b}\to 0\right)}{K_{ee}f_w\left(\frac{c}{b}\to 0\right)M}. \tag{16}$$

Forcing g to only be defined at $c/b=0$ matches the Raju-Newman equations. Additional models were trained where g is allowed to be a function of c/b for a more accurate SIF model.

3.3. Training Bingo Models

Due to the mechanics based break down of training data in Eqns. ?? a reduced subdomain of the feature space is used for training. The function f_w is trained on data where $a/c=1$ and $\phi=\pi/2$. The function M is trained on data where $c/b=0$ and $\phi=\pi/2$. The function g is trained on data where $c/b=0$. Fig. ?? shows this training data reduction. In this figure each point represents a FE simulation, each having SIF values along the crack front, ϕ. One of the simplifications Raju and Newman used when decomposing the SIF into the correction functions was that the effects of c/b could be modeled by the fw equation, neglecting interaction between c/b and ϕ. To understand the effect of this assumption an additional set of Bingo models was trained that allowed c/b to be used in g to model c/b interactions with ϕ. These additional g models were trained using the entire (not reduced) data set. The data set reduction was only possible with the assumption that g did not include c/b

During the training process, two techniques were employed to improve accuracy and interpretability of the trained f_w, M, and g models. The first of these techniques involved the use of custom fitness functions to enforce the known constraints for each model [? ? 9]. For instance, as the values of a/t and c/b approach zero, the value of f_w converges to 1. A fitness function that calculates the loss when this condition is met and assigns an infinite loss when it is not met enforces the equation's limits. Similar limit constraints apply to M and g, such as $M(a/c=1)=1$ and $g(\phi=\pi/2)=1$. The other method used to modify the training process was the addition of seed functions in the initial population. In the initial generation of the Bingo algorithm, a population of randomly generated equations is created. Using seeded equations allows the user to designate a fraction of these randomly generated equations as pre-set equations. This enables users to provide guidance to the algorithm. For example, upon visual examination of the training data for g, it was observed

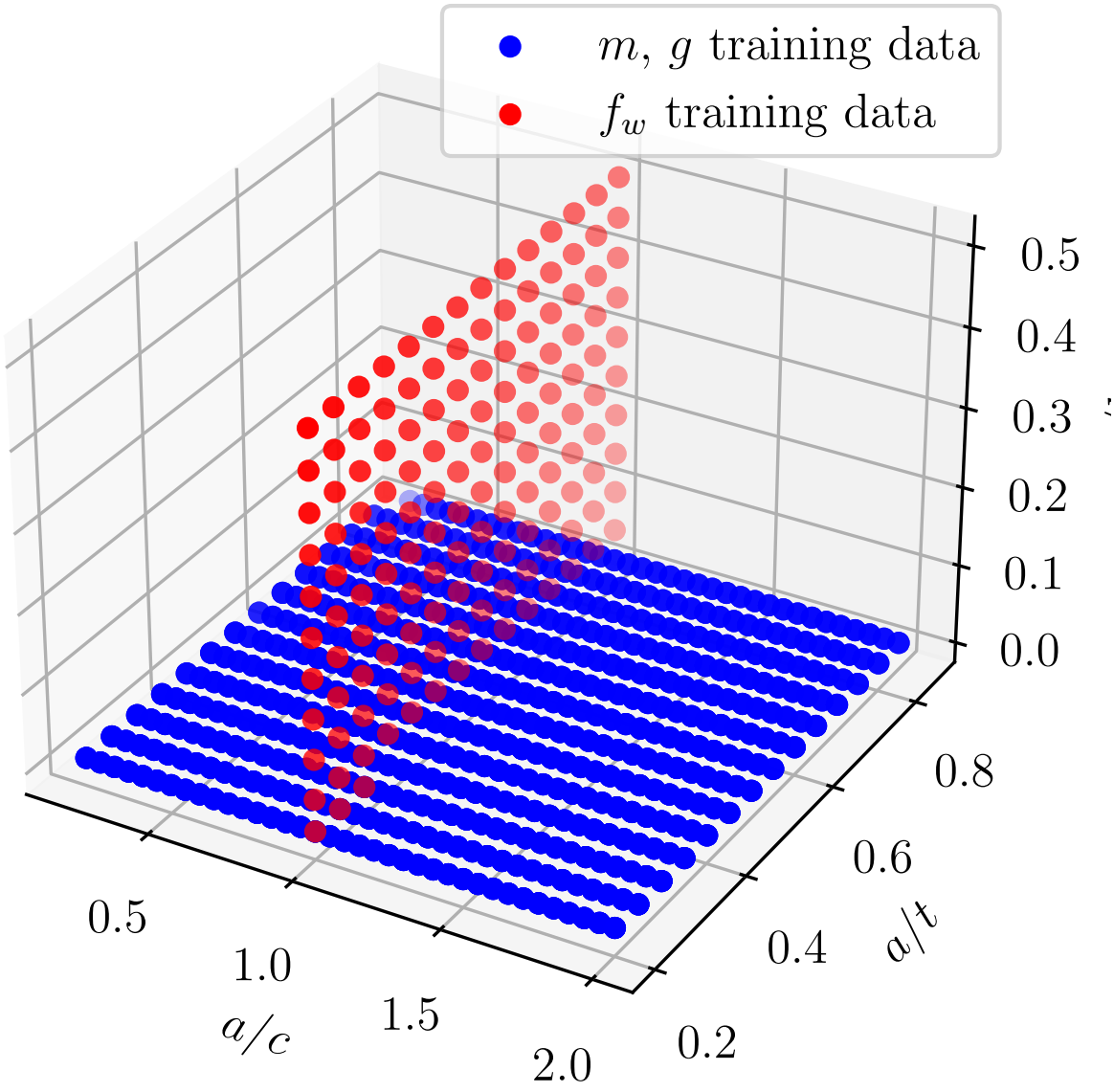


Figure 11: Visualization of the training data reduction allowed by the mechanics based approach. Each point represents a model in the training data-set. f_w was trained when $\phi = \pi/2$ and $a/c = 1$, M was trained when $\phi = \pi/2$ and $c/b = 0$, and g was trained when $c/b = 0$

that $(1-\sin\phi)^{a/t}$ appeared to fit the data well. Additional Bingo models were trained with this function seeded into the initial population. Other Bingo models were trained with only $1-\sin\phi$ while other had no seeding at all.

The base fitness function used was mean absolute error. Custom fitness functions were then applied for specific values of a/c, a/t, c/b, and phi. For instance as a/t and c/b approach 0 the values of f_w must approach 1. This is enforced by checking if the equation inforces the condition and if so it returns the MAE otherwise it returns an infinite loss. The same approach is used for the M and g equations to enforce the known constraints of each model.

The other method of modifying the training process was by the introduction of seed functions [**?**] in the initial population, a population entirely of randomly generated models. For example from the model created by Raju and Newman for g the function $1-sin\phi$ modeled the bounding conditions on g, and we found that $(1-\sin\phi)^{a/t}$ modeled the data across its domain. While these seed functions are not required, their definition when known can improve performance. To assess this, we train bingo models with and without these seed functions.

Once the training data has been generated with the described splits and reductions for, f_w, M, and g, can be trained simultaneously. Multiple models were trained for each of the functions, due to the randomness of genetic programming this gave new equations that could be compared to one another to find the best combinations of f_w, M, and g that predict SIF. The training process is done by first taking the SIF data computed from the FE model and then extracting the f_w, M, and g training data-sets. Then bingo equations can be found for all three equations, these equation are then combined with the embedded ellipse solution from equation 13 to create a prediction for SIFS, this process is visualized in figure **??**.

4. Results

Training Bingo on the functions f_w and M was relatively fast only requiring 1-2 days due to the small training data-set, which only contained one data point per model when $\phi = \pi/2$. Additionally, these functions only required the discovery of relationships between two variables, namely, c/b and a/t for f_w, and a/c and a/t for M. However, training the function g presented a different challenge due to the substantial increase in the number of data points as this function is now trained on every value of ϕ in each model. The Bingo algorithm tends to slow down as more data points are added. This results in a data-set comprising 300-400 data points per model, in contrast to the single data point per model for f_w and M. To expedite the training of the g function, two strategies were employed. The first involved down-sampling the ϕ data to reduce it to only 20 data points per model. The second strategy made use of a Bingo feature called the fitness predictor island (FPI). FPI aims to identify the optimal subset of the data that best represents the entire data-set. The size of this subset is determined by a hyper-parameter that specifies the fraction the subset. Even with theses two methods of reducing the number of data-points Bingo still took 5-7 days to train the g models that did not contain c/b and up to 13 days for the models that did contain c/b.

Upon training completion, the GPSR population of models varies in accuracy and complexity, The Pareto front of theses models is defined b y the best fit model at each complexity

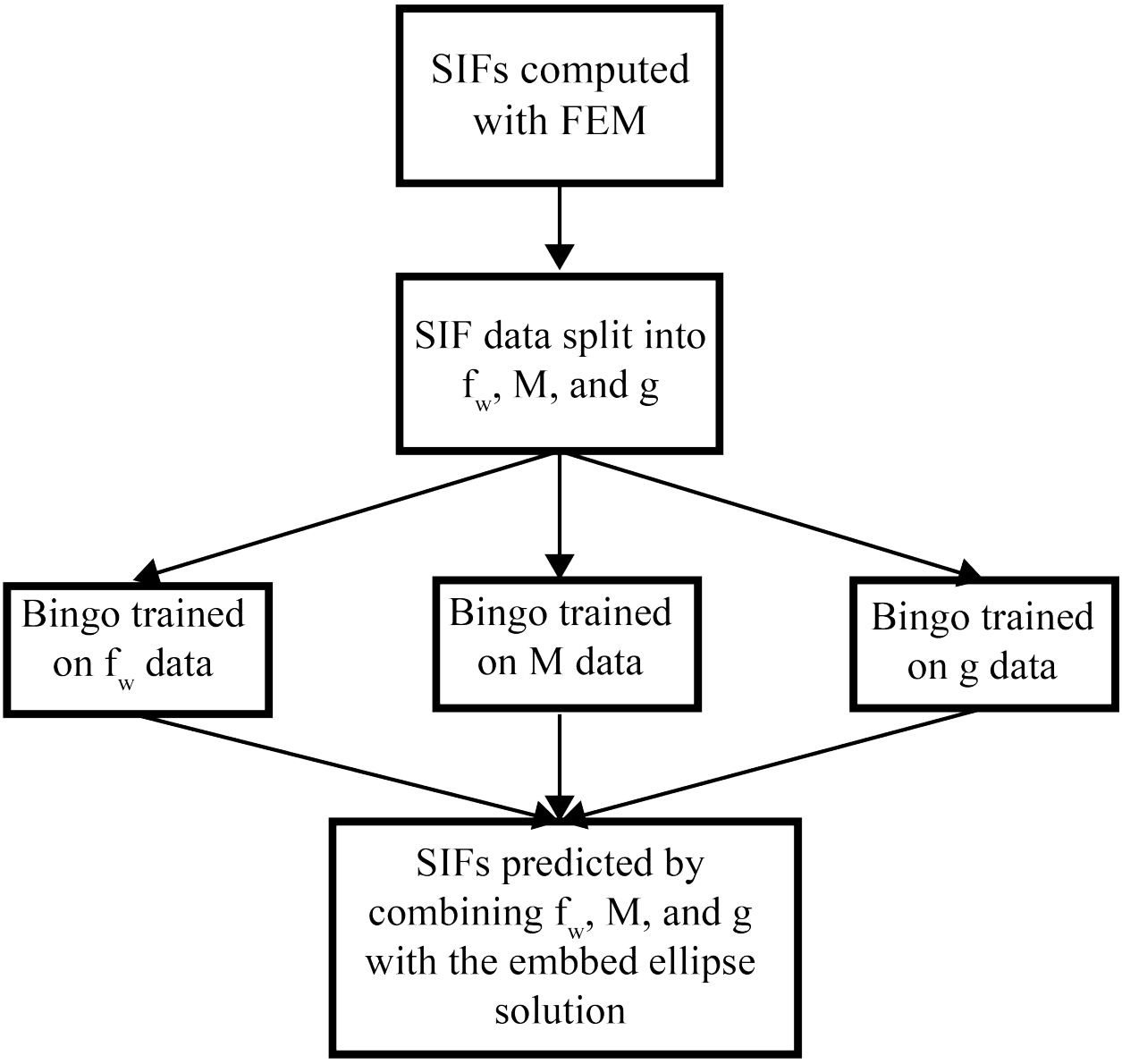


Figure 12: Flow chart illustrating the training process of the three sub-functions f_w, M, and g

level. Ultimately, the, models are selected by the user from the Pareto front. From the approach outlined in the previous section a Pareto front the model is produced for f_w, M, and g. Two methods were used to define the SIF model from the separate Pareto fronts for f_w , M, and g. The first method to select a single model for f_w, M, and g independently and then combining them into a model for SINMF. The second method is to define another Pareto front for SIF using all pssible combinations fo f_w, M, and g. This Pareto front can then be used to select the SIF model.

Four SIF models were chosen using each of the model selection methods. This is done to create various comparisons between GPSR and Raju-Newman models [1]. To lower the number of models presented the same f_w, Eq. 17, and M, Eq. 18, models are used for all four SIF models with only the g model changing.

$$f_w = \frac{\left(\frac{a}{t} + \cos\left(\frac{a}{t}\right)\right)^{\frac{1}{4}}}{\left(\cos\left(\frac{c}{b}\right)\right)^{\frac{1}{2}}} \tag{17}$$

$$M = 1 + 0.06\left(\frac{a}{t}\right)^2\left(\frac{a}{c} - 1\right) + \left(1 - \frac{c}{a}\right)\left(0.0069\frac{c}{a} - 0.28\frac{a}{t}\right) \tag{18}$$

The four models for g are given by Eqns 19-22. These four models were selected to make specific comparisons to the Raju-Newman equations [1]. Eqn 19 was selected to have similar error to the Raju-Newman equation, and Eqn 20 was selected to have similar complexity to the Raju-Newman equations.

results in an a model with the same overall complexity as the Raju-Newman equation. The first model uses the data where g was trained at $\frac{c}{b} = 0$ and

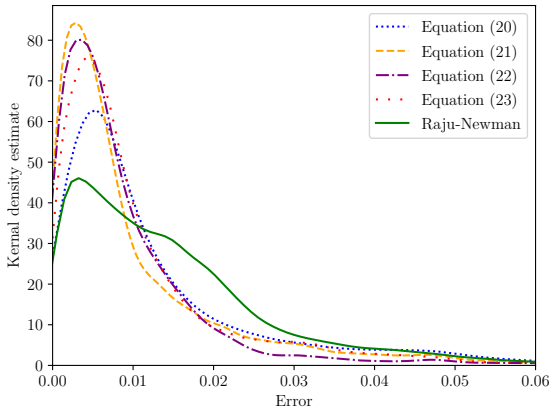
Many Bingo runs were done with different parameters. The candidate equations can be selected from the combined Pareto front of all the bingo runs.

Bingo equations are selected to compare with the Raju-Newman equations. First by matching the complexity of the equation and second by matching the error of the equations

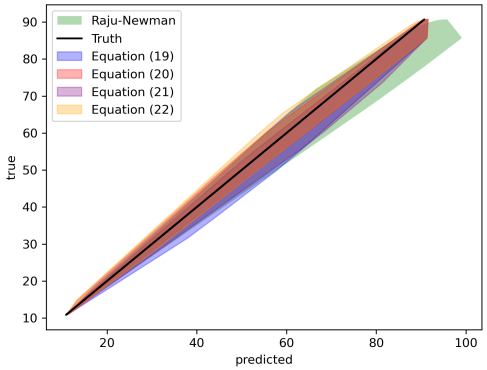
the equations f_w and M have a smaller effect on the value of KI than g, thus for consistency the same equations for f_w and M will be used for multiple models. Equation 19 is the g equation that produces a model with comparable error to the Raju-Newman equation. Equation 20 results in a model with the same complexity as the Raju-Newman equation. Equation 21 also matches the overall complexity; however, the g model was allowed to be trained on c/b. Equation 22 has a balance of error and complexity.

$$g = 1 + \frac{a}{t}\frac{1 - \sqrt{\sin(\phi)}}{\frac{a}{c} + 1} \tag{19}$$

$$g = 1 + 0.0316\frac{a}{c}\left(\left(\sin(\phi) - 0.53\right)\left(\left(\frac{a}{c}\right)^1.54 - 1.18\frac{a}{t} + 1.35\right) - 2.6\right)\left(\sin^{\frac{a}{t}+0.175}(\phi) - 1\right)$$
$$\frac{0.53\left(\frac{a}{c}\right)^1.54 - 0.625\frac{a}{t} + 3.32}{\left(0.739\left(\frac{a}{c}\right)^1.54 - 0.87\frac{a}{t} + 1\right)^2} \tag{20}$$



(a) Distributions of error



(b) Parity plots

Figure 13: (a) Density plot for the errors of each of the equations. (b) Parity plot with each of the different equations.

$$g = 1 + \frac{\left(0.78\left(0.66 - 1.37\sin\left(0.32\frac{a}{c}\frac{a}{t}\left(\frac{a}{t} - \frac{c}{b}\right) - \frac{a}{t} + 1.76\right)\right)(\sin(\phi) + 6.1) + 4.02\right)(\sin(\phi) - 1)}{\left(0.656 - 1.37\sin\left(0.32\frac{a}{c}\frac{a}{t}\left(\frac{a}{t} - \frac{c}{b}\right) - \frac{a}{t} + 1.76\right)\right)(\sin(\phi) + 6.1)} \tag{21}$$

$$g = 1 + \frac{0.4\frac{a}{c}\left(1 - \sin(\phi)^{\frac{a}{t}}\right)}{\left(\frac{a}{c}\right)^2 - \frac{a}{t} + 1} \tag{22}$$

Figure ?? shows the Pareto fronts for all 3 equations. The distributions of the errors for each of the four models along with with Raju-Newman equation are shown in figure 13. The parity plots for each model is also shown in figure 13.

The Pareto front for the model selection using the Pareto front generated using all possible combinations of the models f_w, M, and g can be seen in figure ??. Equations [23-25] are the equations gathered from the Pareto front of all possible equations. The error distributions and parity plots for these models can be seen in figure 15.

$$\begin{cases} f_w = 0.13\frac{c}{b} + 0.13\frac{a}{t} + 1.0 \\ M = -0.223\frac{a}{t} + 1 + \frac{0.223\frac{a}{t}}{\frac{a}{c}\frac{a}{c}} \\ g = 0.554\frac{a}{t}\left(1 - \sin^{0.491\frac{a}{c}}(\phi)\right) \end{cases} \tag{23}$$

$$\begin{cases} f_w = \frac{-63.94\frac{c}{b} + 51.875\frac{a}{t} + 38.855}{-63.94\frac{c}{b} + 43.336\frac{a}{t} + 38.855} \\ M = \frac{-0.028\frac{a}{c}^3 + \frac{a}{c}^2\left(\frac{a}{c} + 0.028\right) - 0.644\frac{a}{c}\frac{a}{t}\left(\frac{a}{c}\frac{a}{t} - 0.044\right)\left(-\frac{a}{c}\left(\frac{a}{t} - 1.103\right) + 0.044\right) + 0.644\frac{a}{t}\left(\frac{a}{c}\frac{a}{t} - 0.044\right)\left(-\frac{a}{c}\left(\frac{a}{t} - 1.103\right) + 0.044\right)}{\frac{a}{c}^3} \\ g = -\frac{0.127\frac{a}{c}\left(\sin^{\frac{a}{t}}(\phi) - 1\right)}{\left(\cos\left(\frac{c}{b}\right) - 0.66\right)\left(\frac{a}{c}^{1.811} - 1.066\frac{a}{t} + 0.992\right)} \end{cases} \tag{24}$$

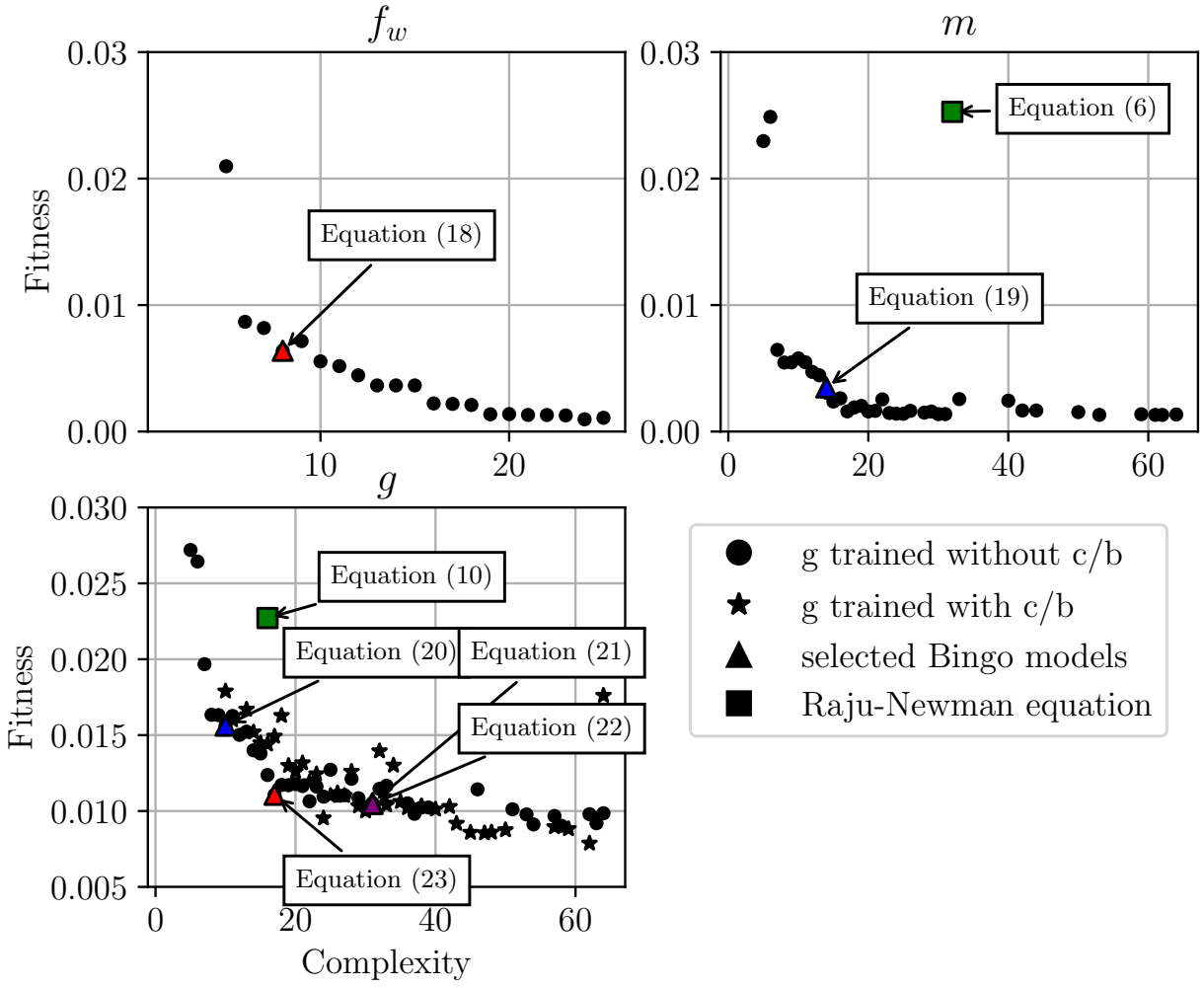
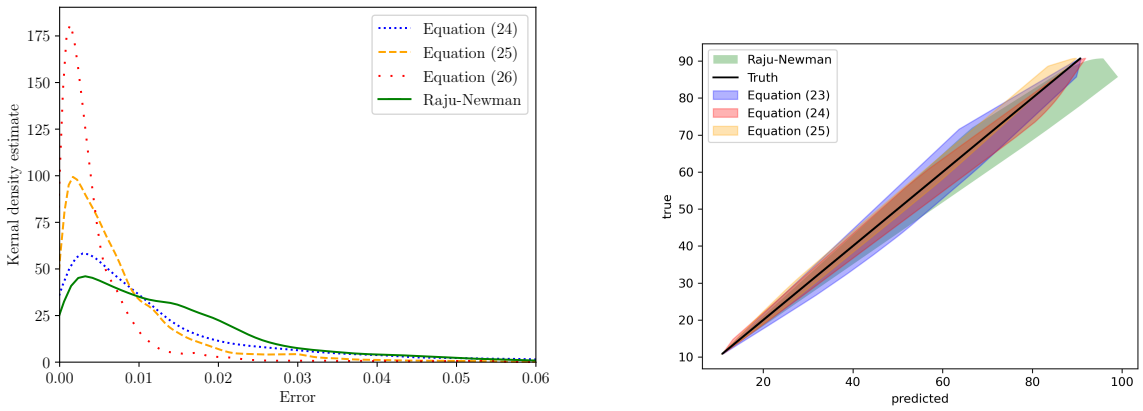


Figure 14: Pareto fronts for f_w, M, and g. f_w and M have the same equation for all four models. Both the Pareto fronts for g with and without c/b are shown



(a) Distributions of error

(b) Parity plots

Figure 15: (a) Density plot for the errors of each of the equations from the combined selection. (b) Parity plot with each of the different equations from the combined selection.

$$\begin{cases} f_w = -\frac{a}{t}\left(\left(1.216\frac{c}{b}^2 + 0.11\right)\left(\frac{a}{t} - \sqrt{-\frac{c}{b}^2 - 0.751}\right) - 0.119\right) + 1.0 \\ M = \frac{a}{c}\frac{-0.037\frac{a}{c} - 0.046\frac{a}{t}\left(\frac{a}{c}^2 - 15.413\right)\left(\frac{a}{t} - 1.114\right)\left(1.919\frac{a}{t} - 0.489\right) - 0.06}{\frac{a}{c} + 0.599} \\ g = -\frac{1.232\left(\frac{a}{c}\left(\frac{a}{c} + 0.06\right)\left(0.103(1-\sin(\phi))^{\frac{3}{2}} + 0.017\right) - 0.36\left(\frac{a}{t} - 0.187\right)\left(\frac{a}{c}\left(\left(\frac{a}{c} + 0.06\right)\left(0.017\frac{a}{c} + 0.033\frac{a}{t} - 0.409\frac{c}{b}^2\right) - 0.646\right)\left(\frac{a}{c} + 2\frac{a}{t}\right) + 2.501\frac{a}{c} + 0.\right.\right.}{\frac{a}{c}\left(\frac{a}{c} + 0.06\right)} \end{cases} \tag{25}$$

Finally figure ?? shows all eight models on the same Pareto front.

5. Discussion

GPSR offers the advantage of generating a population of equations, providing users with the flexibility to choose the equation that best suits their specific needs. Typically, the optimal equation strikes a balance between simplicity and accuracy. However, certain scenarios may necessitate either maximum accuracy or very simple equations. This study presents various models with varying levels of accuracy and complexity. The SIF equation was derived using two methods from the f_w, m, and g equations. The first method involved using fixed f_w and m equations, as the g equation has the most significant impact on overall SIF accuracy. The equations obtained through this method are illustrated in Figure ??. The second method employed every possible combination of the f_w, m, and g equations generated by Bingo, with the extracted equations depicted in Figure ??. To facilitate comparison with the Raju and Newman equation, models were selected based on similar errors and complexities, as shown in Figure ??. The comparison revealed that Bingo outperforms the Raju and Newman equation in both complexity and error. Models with similar complexity demonstrated lower errors, while models with similar errors exhibited considerably lower complexities.

When applying the same approach as Raju and Newman to decompose K using f_w, M, and g, the influence of c/b is not entirely captured. The finite width of the model is considered only in the function f_w, which is not dependent on ϕ. Consequently, there is no

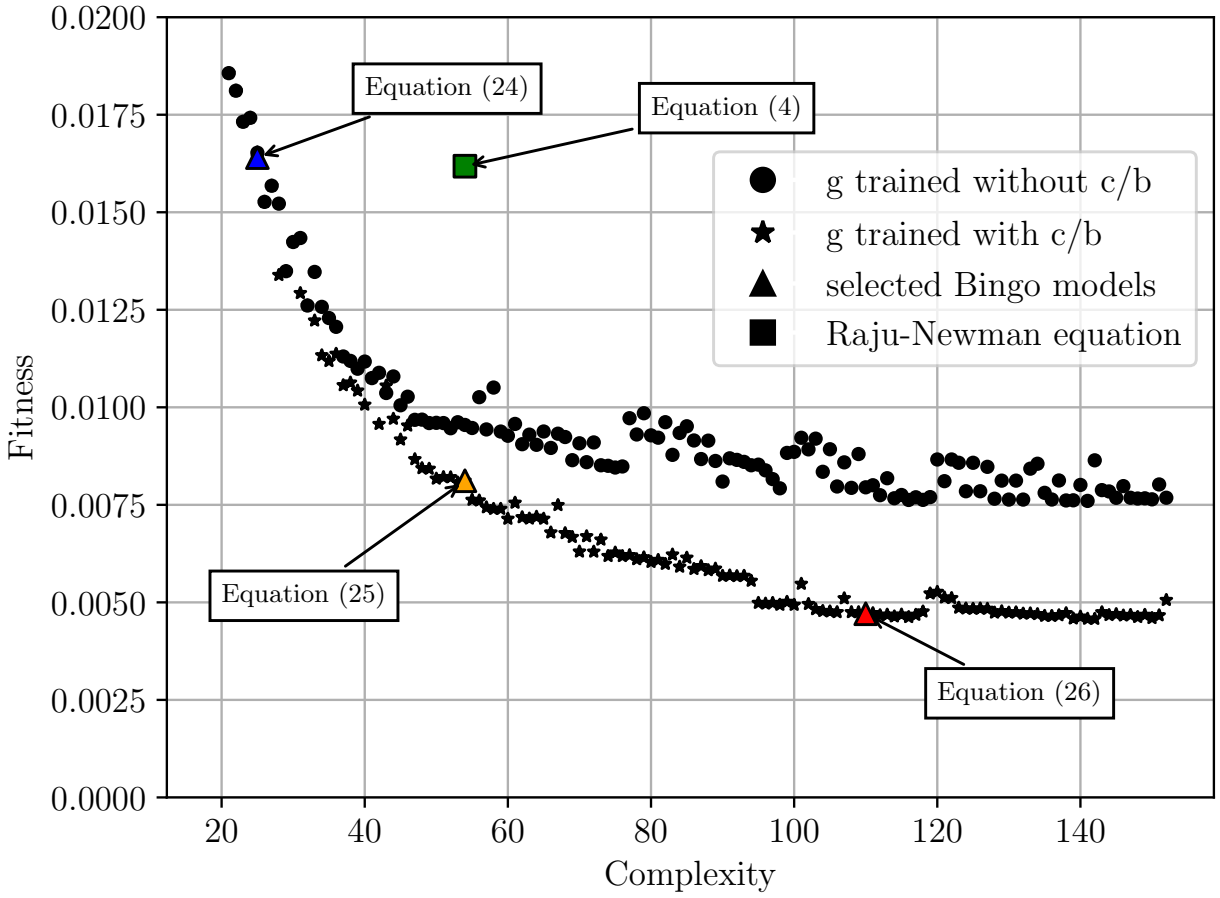


Figure 16: Pareto front for the models selected using all possible combinations.

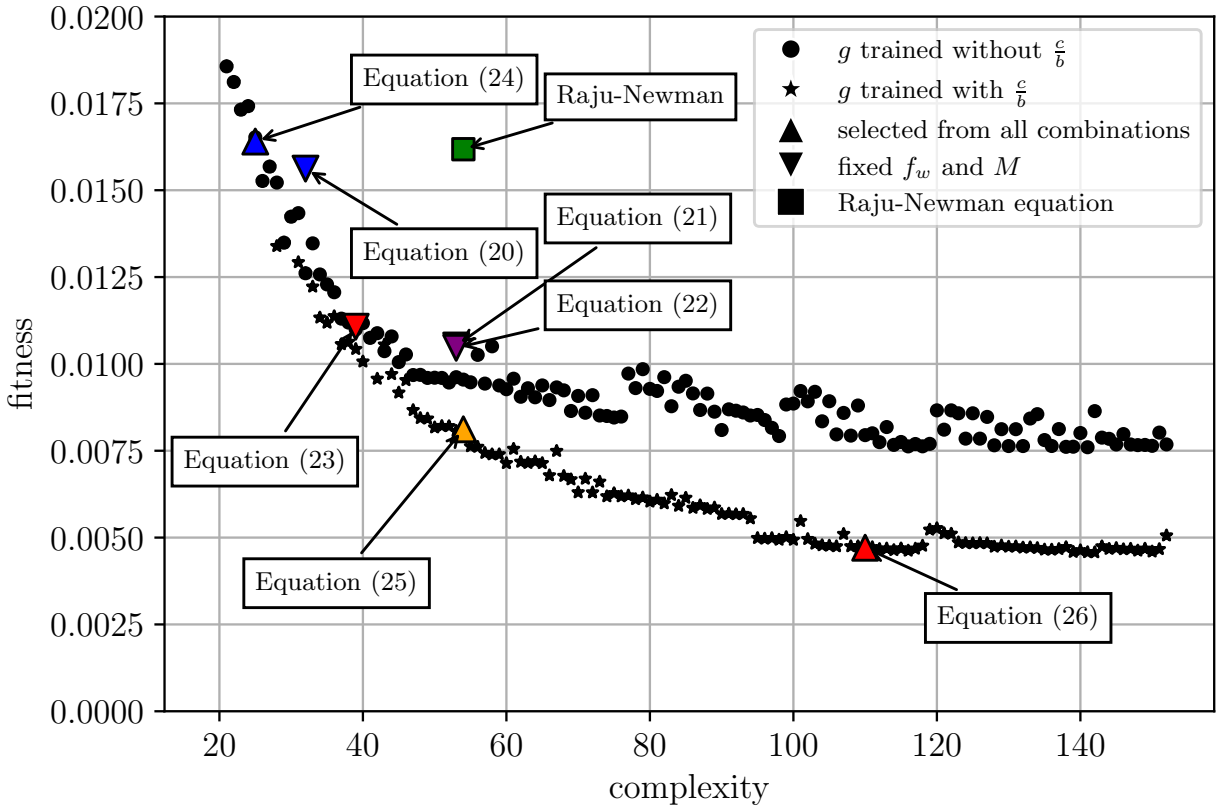


Figure 17: Pareto front with all models the triangle markers correspond to the models that were individually selected and the square markers are the models selected using all possible combinations.

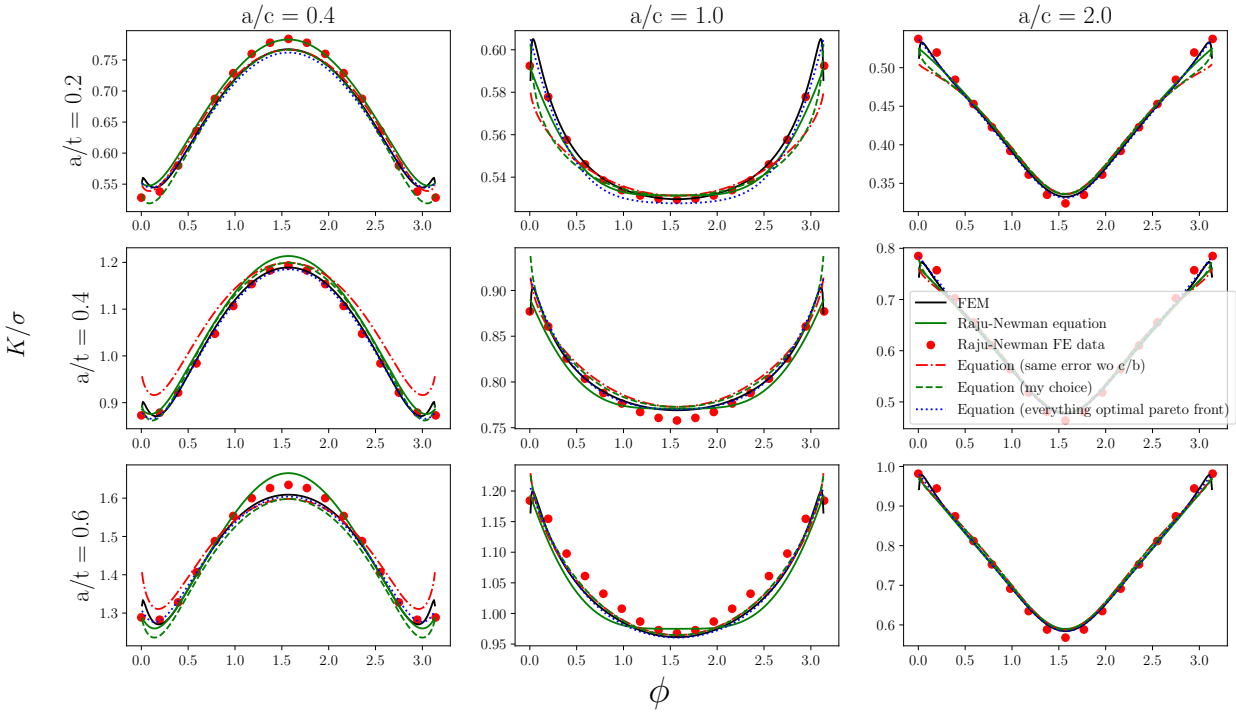


Figure 18: A sample of models comparing the SIFs collected from FRANC3D with the bingo equation, the Raju-Newman equation, and the Raju-Newman FE data.

coupling between ϕ and c/b. This is evident in Figure ??, where models allowing c/b in g outperform those that do not, particularly beyond a complexity of approximately 40. This discrepancy is observed specifically in models derived from every combination of f_w, M, and g. However, in the case of fixed f_w and M functions, whether or not c/b is utilized in g does not seem to alter the results, as shown in Figure ??. This is attributed to the specific f_w equation chosen not being influenced by c/b in g. Neglecting the interaction between ϕ and c/b allows for the rapid creation of simpler models, significantly reducing the number of FE models required for training. Nonetheless, if a highly accurate model is essential, employing the entire domain for training enables the attainment of higher accuracy.

5.1. Model limitations

It was observed that, across all trained Bingo models, the highest errors occurred at the crack surface where ϕ is close to 0 or π and when a/c is at its smallest value of 0.2. At the crack surface where $\phi = \pi, 0$, the SIF values exhibit a sharp decline due to the influence of plane stress vs. plane strain conditions. The Bingo model struggles to fully capture this region because of a lack of data at the surfaces. It might be possible to address these surface effects by introducing an additional correction factor that specifically accounts for the model's surfaces. The reason for the highest errors occurring at $a/c = 0.2$ is that this represents the lower bound of the training data, and there is also a significant change in the SIF data in this region. Without information about SIFs for values smaller than $a/c = 0.2$ in the training data, Bingo fits this region of the data less accurately. Including smaller aspect

ratios in the training data could likely reduce the maximum error of the model. However, it's worth noting that $a/c = 0.2$ is already a relatively small aspect ratio, and the rest of the domain with aspect ratios above $a/c = 0.2$ have very low error.

5.2. Comparison with Black Box methods

Comparing the results shown here from Bingo with other ML methods such as the ones used in [**?**]. Figure ?? shows Bingo and the Raju-Newman equation compared with two commonly used ML methods for a data-set of this size, support vector machine regression (SVM) and random forest regression (RFR). Using the optimal hyper-parameters from [**?**] and training directly on K results in the square markers on the plot. Training the models again using the mechanics bases approach used in this research, results in the circle points. From this figure, the error of the RFR is reduced by using the mechanics based approach, while the SVM produces worse error when using the mechanics based approach. One possible reason that the SVM produces a worse model when using the mechanics based approach, is that it cannot fit to the smaller data-sets, while Bingo and RFR are easily able to fit to fewer data-points. It is shown in [**?**] that as the data-points decrease to 100 that SVM performs worse. Being able to produce accurate models with fewer data-points is a strength as having many training examples is rarely feasible. Comparing the evaluation times of each of the ML models shown in ?? shows that bingo and the Raju-Newman have similar evaluation times, however, the black box models are slower with the random forest taking 3 seconds to evaluate the data-set, and SVM taking 40 seconds.

Bingo outperforms SVM and RFR both in evaluation time and error, while also producing an explainable closed form equation. Some of the Bingo models only needed a small subset of the data to be able to predict SIF for the entire domain. Bingo does, however, have a downside in its long training time on the order of hours to days, while SVM and RFR can be trained very quickly on the order of seconds to minutes. Bingo takes more effort than SVM and RFR to create a model for this data-set, however, the final model created by Bingo outperforms SVM, RFR, and the Raju-Newman equation in all metrics.

6. Conclusion

By incorporating knowledge of fracture mechanics into the training process, GPSR is able to create models that can outperform traditional ML methods in both accuracy and explainability. The main limitation of GPSR is its expensive training process. This reduces the number of parameters that can be used making it more difficult to apply to more complex problems. By using the framework proposed here it is possible to create a library of SIF equations for many crack cases that currently have no handbook solutions.

References

[1] J. C. Newman and I. S. Raju. *Stress-intensity factor equations for cracks in three-dimensional finite bodies*, volume 4. 1983.

[2] X Wang and S B Lambert. Local weight functions for semi-elliptical surface cracks in finite thickness plates, 1995.

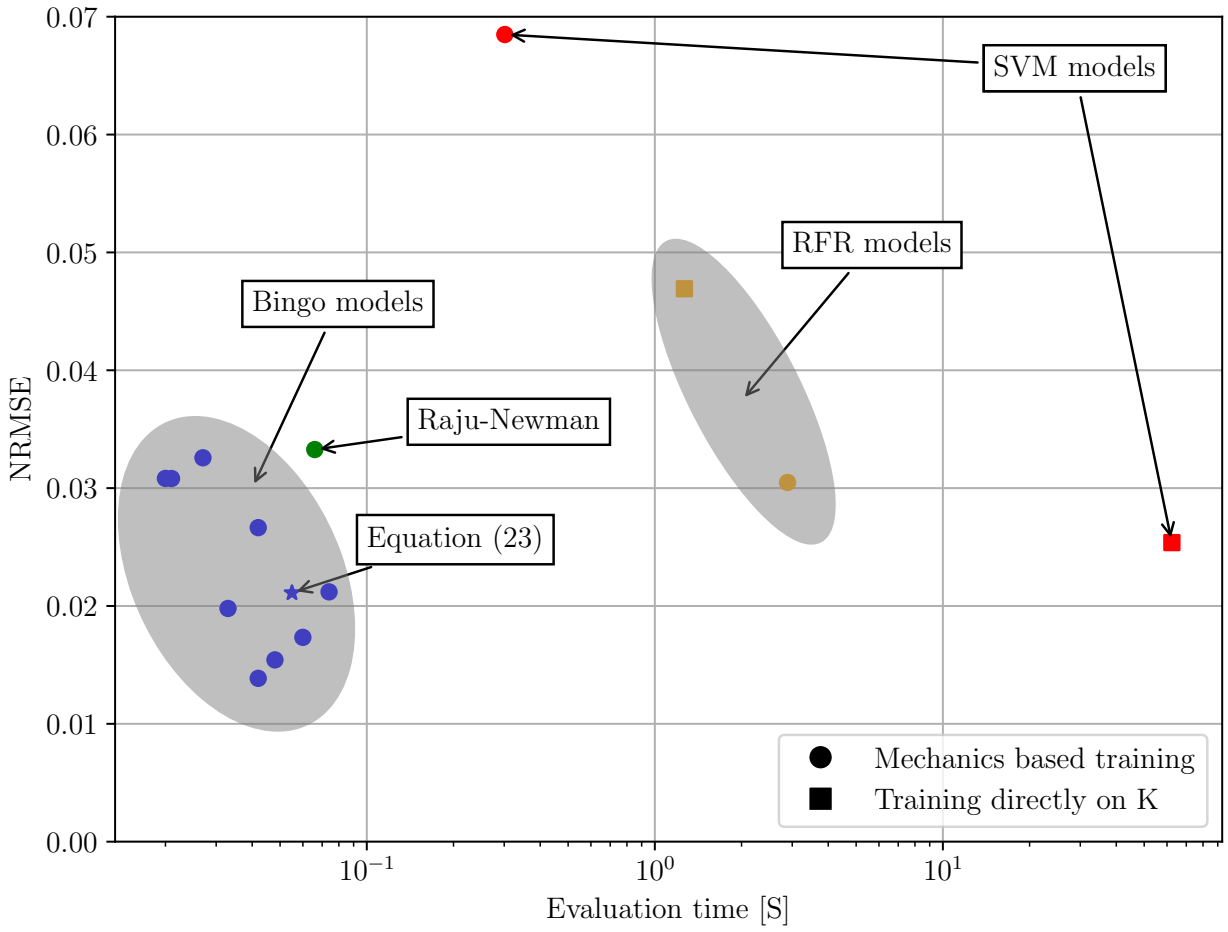


Figure 19: Plot comparing training times and errors between Bingo, Raju-Newman equation, RFR, and SVM. The circular markers were all trained using the mechanics based approach presented in this research, the square makers were trained directly on K. The evaluation times were measured on one core of an Intel i7-4790 with 32Gb DDR3 RAM at 1600 MHz on a data-set containing 150k data-points.

[3] S Pommier, C Sakae, and Y Murakami. An empirical stress intensity factor set of equations for a semi-elliptical crack in a semi-infinite body subjected to a polynomial stress distribution, 1999.

[4] Xinyu Zhang, Tingting Zhao, Yifan Liu, Qingqing Chen, Zhiyong Wang, and Zhihua Wang. A data-driven model for predicting the mixed-mode stress intensity factors of a crack in composites. *Engineering Fracture Mechanics*, 288, 8 2023.

[5] J. C. Sobotka and R. C. McClung. Stress-intensity factors solutions for straight through cracks in c-sections. *Engineering Fracture Mechanics*, 271, 8 2022.

[6] Arvind Keprate, R. M.Chandima Ratnayake, and Shankar Sankararaman. Comparison of various surrogate models to predict stress intensity factor of a crack propagating in offshore piping. *Journal of Offshore Mechanics and Arctic Engineering*, 139, 12 2017.

[7] Tongge Xu, Shuiting Ding, Huimin Zhou, and Guo Li. Machine learning-based efficient stress intensity factor calculation for aeroengine disk probabilistic risk assessment under polynomial stress fields. *Fatigue and Fracture of Engineering Materials and Structures*, 45:451–465, 2 2022.

[8] Mohamed El Amine Ben Seghier, Hermes Carvalho, Behrooz Keshtegar, José A.F.O. Correia, and Filippo Berto. Novel hybridized adaptive neuro-fuzzy inference system models based particle swarm optimization and genetic algorithms for accurate prediction of stress intensity factor. *Fatigue and Fracture of Engineering Materials and Structures*, 43:2653–2667, 11 2020.

[9] David L. Randall, Tyler S. Townsend, Jacob D. Hochhalter, and Geoffrey F. Bomarito. Bingo: A customizable framework for symbolic regression with genetic programming. pages 2282–2288. Association for Computing Machinery, Inc, 7 2022.

[10] Southwest Research Institute. Nasgro.

[11] TODO. Todo.

[12] TODO. Smart|dt.

[13] Hiroshi Tada, Paul C. Paris, and George R. Irwin. *Stress Analysis of Cracks Handbook*. American Society of Mechanical Engineers (ASME), 3rd edition, 1985.

[14] J. E. Srawley W. F. Brown. *Plane Strain Crack Toughness Testing of High Strength Metallic Materials*. ASTM International, 1966.

[15] Fracture Analysis Consultants, Inc. Franc3d.

[16] Dassault Systèmes Simulia Corp. Abaqus.

[17] J R Rice. A path independent integral and the approximate analysis of strain concentration by notches and cracks, 1968.

[18] Leslie Banks-Sills, Itai Hershkovitz, Paul A. Wawrzynek, Rami Eliasi, and Anthony R. Ingraffea. Methods for calculating stress intensity factors in anisotropic materials: Part i - z = 0 is a symmetric plane. *Engineering Fracture Mechanics*, 72:2328–2358, 10 2005.

[19] F. O. de Franca, M. Virgolin, M. Kommenda, M. S. Majumder, M. Cranmer, G. Espada, L. Ingelse, A. Fonseca, M. Landajuela, B. Petersen, R. Glatt, N. Mundhenk, C. S. Lee, J. D. Hochhalter, D. L. Randall, P. Kamienny, H. Zhang, G. Dick, A. Simon, B. Burlacu, Jaan Kasak, Meera Machado, Casper Wilstrup, and W. G. La Cava. Interpretable symbolic regression for data science: Analysis of the 2022 competition. 4 2023.

[20] Michael Schmidt and Hod Lipson. *Comparison of Tree and Graph Encodings as a Function of Problem Complexity.* Association for Computing Machinery, 2007.

[21] Isadore Newman and I. S. Raju. Three dimensional finite-element analysis of finite-thickness fracture specimens, 1977.