Accelerating continuum-scale brittle fracture simulations with machine learning

M. Giselle Fernández-Godino, Nishant Panda, Daniel O’Malley, Kevin Larkin, Abigail Hunter, Raphael T. Haftka, and Gowri Srinivasan

January 31, 2020

Abstract

Simulating the behavior of macroscale brittle materials is extremely computationally expensive if every microcrack is explicitly simulated since a very fine grid is required to resolve these cracks. Continuum scale models that lump these microcracks into elastic moduli are much less computationally expensive but are prone to neglecting the physics of the subgrid-scale microcracks. We develop an approach that leverages data from an expensive simulator (HOSS) that explicitly resolves the behavior of microcracks to build a machine learning model of the impact of the microcracks on the elastic moduli. We then combine this machine learning model with a continuum scale model (FLAG) to produce a model (which we call ML-FLAG) that provides accuracy comparable to HOSS but is four orders of magnitude faster than HOSS. We validate our ML-FLAG model by comparing it to a flyer plate experiment.

1 Introduction

The existence, evolution, interaction, and coalescence of individual cracks is key to modeling strength and damage behavior in several brittle materials such as granite, concrete, ceramics or Beryllium. The physical problem of interest for this work is the study of the strength and damage behavior of Beryllium S200F under dynamic loading conditions. The simulated problem is a two-dimensional flyer plate impact against a target specimen. Both, the flyer plate and the target, are modeled as high-strength Beryllium. The sample has a width of 28.8mm, the height of the target is 4mm, and the height of the flyer plate is 2mm. The flyer plate has an initial velocity of 0.721km/s and the simulation time is 1.2μs. The geometry and setup were chosen to recreate the available flyer plate experiment as close as possible for future validation. For more information on flyer plate experiments, the reader can refer to [9, 10]. Ideally, predicting damage evolution in practical applications must account for the presence of individual cracks, but such simulations using finite and/or discrete element simulations are computationally very expensive. In this work,
the evolution of the discrete crack network is first modeled with a high-fidelity implementation of the finite-discrete-element method (FDEM) which is called the Hybrid Optimization Software Suite (HOSS) [28, 19, 18]. In the case of the flyer plate test simulations used in this work, each run required 2.5 hours on 64 processors. The computational burden is thus intractable for real-world applications, such as optimization or uncertainty quantification, and surrogate or reduced-order models are desirable.

Predicting damage evolution in practical applications must account for the presence of cracks, but such simulations using finite and/or discrete element simulations are computationally very expensive. Therefore, in this work, we seek to correct the low-fidelity (LF) model compliance tensor, tensor that relates stress with strain, using the cracks information provided by the high-fidelity (HF) model. Current correcting approaches use empirical models or statistical averaging techniques to inform damage to continuum models [17]. However, detailed information about the material damage evolution is lost during this homogenization process [20, 21]. The current work takes into account the detailed information available in the HF model to account for damage evolution in the LF model, which was not considered previously.

While the detailed implementation of the FDEM method can be found in references [28, 19], we describe here briefly the setup of the problem considered. The cells in HOSS simulations are jointed by four shear and four tensile cohesive points that act like nonlinear springs. Under sufficient shear and/or tensile stress the cell edges are able to separate until they reach a critical separation and the points are broken.

The considered LF model is FLAG [35, 34, 14], which is a continuum hydrodynamics research code developed at Los Alamos National Laboratory. A FLAG flyer plate test simulation requires 0.1 hours on a single processor. To simulate brittle damage in FLAG, the stress compliance tensor is corrected using the information of the evolution of cracks [16], as described in Section 2.4.

Machine learning (ML) methodologies have shown great promise recently in modeling materials behavior, particularly for the problem considered [32, 15, 22, 31]. In particular, in this work we use a recurrent neural network (RNN)
This paper is organized as follows. Section 2 includes a description of the HF model, HOSS, of the low fidelity model, FLAG, and of the ML model. This section also shows how FLAG damage model is improved using crack statistics from HOSS. Section 3 describes how the ML quantities of interest (QoIs) were chosen and also shows comparing plots between the validation data from experiments and ML-FLAG results. Finally, Section 4 summarizes this work including also a brief discussion of the future work.

2 Methods

2.1 Hybrid Optimization Software Suite (HOSS) Model

For this work, we have chosen as an HF model the Hybrid Optimization Software Suite (HOSS) [28, 19, 18]. HOSS simulates the evolution of the material crack network which will be used to inform the ML approach described in Section 2.3. HOSS is a hybrid multi-physics model based on the combined finite discrete element method (FDEM). The FDEM approach combines finite element techniques to describe the deformation of the material with discrete element-based transient dynamics, contact detection, and contact interaction solutions, so it can account for both damage evolution, and catastrophic fracture or fragmentation. Detailed information about HOSS is out of the scope of this work but the interested reader can refer to the following references for a comprehensive
In the HOSS FDEM model, the discrete elements are further divided into finite elements. The governing equations are based on conservation of mass, momentum, and energy along with Newton’s laws [23, 25, 26, 27, 29, 24]. These equations are solved using an explicit central difference time integration scheme [30]. In HOSS FDEM model, cracks form along the boundaries of the finite elements. In order to capture fine mechanisms, such as crack nucleation, coalescence, propagation, branching, reorientation, etc., the crack network must be finely resolved spatially, with dozens to hundreds of finite elements along the length of each crack [26]. As a result, the outputs of simulations involving laboratory sized samples with thousands of incipient micro-cracks can result in petabytes of data. In addition, the need for a highly resolved mesh combined with the explicit time integration scheme can result in a need for high-performance computing resources during an extended period to model the damage evolution and failure of laboratory sized samples and larger.

Since the problem of interest will be dominated by Mode I crack growth, we focus this discussion on the key details as to how HOSS accounts for Mode I crack growth. However, it must be pointed out that the Mode II crack growth is handled in a similar way to the Mode I case, except different sets of parameters are applied. Also, in a HOSS simulation of a pure tension problem, many of the element edges will not be oriented orthogonally to the applied load. Hence, although globally Mode I failure dominates this problem, both shear and opening can occur at a local mesh element scale. Between the interface of any two finite elements, there are a user specified number of cohesive points, which are modeled as springs. As the two elements undergo tensile load and are pulled apart, the springs within the interface are strained resulting in a small space opening between the elements. Similarly, for shear, or Mode II deformation, there will be cohesive points that can deform to allow one element to slide relative to another.

The simulated problem is a two-dimensional flyer plate which impacts against a target specimen. Both, the flyer plate and the target, are modeled as high-strength beryllium. The sample has a width of 28.8\,mm, the height of the target is 4\,mm, and the height of the flyer plate is 2\,mm. The flyer plate has an initial velocity of 0.721\,km/s. The total number of HOSS HF simulations available is 100, having each 480 time steps. The simulation time is 1.2\,µs and the time step progression is 0.0025\,µs. To accurately simulate the crack dynamics, the HF model requires 31,000 elements, resulting in 23\,GBs of data per simulation. The target specimen has initially 200 cracks. The initial crack lengths are between 0.1\,mm and 0.3\,mm and have a random orientation. The initial distribution of cracks is based on a power-law function that is inspired by the preexisting crack length distribution on materials. As stated before, to calculate the statistical crack evolution information, each HOSS simulation required a highly resolved mesh and took hours to complete on hundreds of processors. The initial position, orientation, and length of the preexisting cracks are changed randomly in every simulation. At each of the 480 output files produced by a single HOSS simulation, we compute the lengths and orientations of every crack in the sample.
and compile the time-dependent crack PDF. The target specimen has initially 200 cracks. The PDF considered to generate the cracks lengths is the following power-law function

\[
PDF (x) = \frac{g x^{(g-1)}}{b^g - a^g},
\]

where \( g = -3, \ a = 0.1, \ b = 0.3, \) and \( x \) is a vector in the range \([a, b]\). Hence,

\[
PDF (x) \approx \frac{x^{-4}}{321}.
\]

The cumulative density function (CDF) correspondent to this PDF is used to obtain samples. We can write the CDF as

\[
CDF (x) = \int_a^x PDF (t) \, dt = \frac{x^g - a^g}{b^g - a^g} \approx \frac{1000 - x^g}{963}.
\]

The CDF of Eq. (3) is used to get a 200 sample crack length distribution for each simulation. Therefore given a random number \( r \) in the interval \([0, 1]\) each sample is obtained from

\[
x = [a^g + (b^g - a^g) r]^{1/g} \approx \frac{1}{\sqrt[6]{1000 + 963 r}}.
\]

Figure 3(a) shows the shock wave velocity at initial time (t=0) and final time (1.2 \( \mu \)s). As the figure shows the failure is produced in the middle section of the target plate. This is because the target plate is twice the width of the flyer plate.

Similarly, Figure 3(b) shows the damage at the initial time (t=0) and final time (1.2 \( \mu \)s). The flyer plate is modeled to not be damaged throughout the simulation. The target plate has initially 200 cracks which evolve until the final state (1.2 \( \mu \)s) shown in the figure.
Figure 3: HOSS simulation captures.

(a) Shock wave velocity at initial (up) and final (below) time.

(b) Damage at initial (up) and final time (below).
2.2 FLAG Model

FLAG is a fully parallel multiphysics hydrocode developed and maintained by Los Alamos National Laboratory [5, 7, 6]. Unstructured polyhedral cell-centered or staggered-grid finite volume elements, coupled with arbitrary Lagrangian-Eulerian (ALE) remapping [8] or adaptive mesh refinement (AMR) mesh optimization techniques, allow for the simulation of complex multidimensional dynamics problems in FLAG. FLAG is equipped with a large array of physics packages such as the various equation of state models as well as material strength, plasticity, and damage models that make FLAG especially useful for simulating shock physics problems. Previous researchers have used FLAG to study impact cratering [11], ejecta formation and transport [14], flyer plate impact experiments [12], shock-driven multiphase instabilities [3], and detonation shock dynamics [2]. We aim to use FLAG to simulate damage in flyer plate impact experiments using a statistical crack growth model, informed by crack data generated using machine learning techniques.

The Be on Be flyer plate experiment depicted in figure 1 is simulated in FLAG using a 2D Lagrangian framework. The flyer plate and target plate were meshed using 10368 and 20352 quad elements (zones), respectively. The left and right edges of the flyer and target plates are constrained to move only in the y-direction and a sideline boundary condition is placed between the plates to avoid interpenetration of the two meshed objects. The Gruneisen analytical equation of state [13] and the Steinburg-Guinan plasticity model [1] are used to simulate the material properties of beryllium. The parameters for these models can be found in tables 1 and 2, respectively. The damage model used to simulate crack propagation in the target is based on a modified version of the statistical effective moduli model of Ju and Chen [16]. This model relies on crack length and orientation data as well as the maximum tensile strength at each time step to approximate the effective compliance of a cracked material. This damage model and its connection to machine learning are discussed in more detail in section 2.4. The velocity at the rear, center of the target plate was recorded for comparison with the HOSS model and experimental VISAR data.

| Parameter | Description                  | Value       |
|-----------|------------------------------|-------------|
| \( \rho_0 \) | reference density           | 1.845g/cm³  |
| \( c \)   | bulk sound speed            | 0.799cm/µs  |
| \( s_1 \) | linear Hugoniot coefficient | 1.13        |
| \( \gamma_0 \) | Gruneisen \( \gamma \) at initial density | 1.11        |
| \( a \)   | ramp parameter              | 0.16        |
| \( t_0 \) | reference temperature       | 293K        |
| \( c_v \) | specific heat               | 1.82J/gK    |
Table 2: Steinberg-Guinan model parameters for Beryllium [33]

| Parameter | Description               | Value       |
|-----------|---------------------------|-------------|
| $\rho_0$  | reference density         | 1.845 $g/cm^3$ |
| $G_0$     | initial shear modulus     | 1.51 $Mbar$ |
| $Y_0$     | initial flow stress       | 0.0033 $Mbar$ |
| $Y_{\max}$ | max work hardening        | 0.0131 $Mbar$ |
| $\beta$  | work hardening parameter | 26          |
| $n$       | work hardening exponent   | 0.78        |
| $A$       | pressure dependence multiplier | 0 |
| $B$       | temperature dependence multiplier | 0 |
| $q_y$     | flow stress pressure dependence factor | 1.0 |
| $f_g$     | melt shaping for shear modulus | 0 |
| $f_y$     | melt shaping for flow stress | 0 |
| $\rho_{\text{cr}}$ | crushed-up density | 1.845 $g/cm^3$ |

2.3 Machine Learning Model

Building a model that could predict the length of the maximum crack (QoI1) and the maximum tensile stress (QoI2) turned out to be challenging in our case. We have a total of 100 simulations, each simulation gives us the quantities of interest (QoIs) for 480 time steps. We chose to use 70 simulations for training the model and 30 for testing the model. As Figure 8 shows, in the QoIs obtained from the simulations the variability from simulation to simulation is small. This small variation is because we are modeling the same experiment changing only the initial cracks orientation and cracks length from simulation to simulation to obtain statistical variation. The small variability in the data helps because it will be easier for the network to predict given the small amount of data. The drawback of the small availability is that we are predicting from the initial time step, hence if the model cannot tell the difference between simulation to simulation, it will not predict better than the mean, which is our baseline model.

The data was trained using multiple models and the best performance was achieved by the Encoder-Decoder LSTM (Long Short Term Memory) model described in detail in Figure 4. This model was based on the one presented in chapter 20, Section 8 from reference [4] and its architecture is described in Figure 4. The model predicts a sequence output. The LSTM model is used in the decoder, allowing it to both know what was predicted for the prior time step in the sequence and accumulate internal state while outputting the sequence. We define an LSTM hidden layer with 10 units. This is the decoder model that will read the input sequence and will output a 10 element vector (one output per unit) that captures features from the input sequence. We will use the initial time step as input. We will use a simple encoder-decoder architecture that is easy to implement in Keras that has a lot of similarity to the architecture of an LSTM autoencoder. First, the internal representation of the input sequence is repeated multiple times, once for each time step in the output sequence. This sequence of
vectors will be presented to the LSTM decoder. We then define another decoder as an LSTM hidden layer with 10 units. Importantly, the decoder will output the entire sequence, not just the output at the end of the sequence. This means that each of the 10 units will output a value for each of the 100 time steps that the model will be trained to predict each time, representing the basis for what to predict for each time step in the output sequence.

We will then use a fully connected layer to interpret each time step in the output sequence before the final output layer. Importantly, the output layer predicts a single step in the output sequence. This means that we will use the same layers applied to each step in the output sequence. It means that the same fully connected layer and the output layer will be used to process each time step provided by the decoder.

To achieve this, we wrap the interpretation layer and the output layer in a TimeDistributed wrapper that allows the wrapped layers to be used for each time step from the decoder.

We use the mean squared error (MSE) loss function as it is a good match for our chosen error metric, the root mean squared error (RMSE). We will use the efficient Adam implementation of stochastic gradient descent and the model for 10 epochs with a batch size of 14. The small batch size and the stochastic nature of the algorithm mean that the same model will learn a slightly different mapping of inputs to outputs each time it is trained. This means results may vary when the model is evaluated.

Literature research told us that to predict few "days" of data we needed dozens of "years" of data. We calculated that with roughly 10,000 simulations we could train the model successfully. The problem was, of course, we could not afford so many simulations and it would be more efficient to use HOSS instead of the ML model to speed up the process if so. After trying without success to perform better than the mean using the available data. We then tried the following three approaches to improve the performance of the network.

1. The first approach was to increase the number of epochs from 10 to 100. The number of epochs is the number of times that the model is fed with the data. For example, if we have 100 training simulations, and our batch size is 20, then it will take 5 iterations to complete 1 epoch. The batch data set changes randomly from epoch to epoch.

2. The second approach was to add artificial noise to each simulation. For this, we used the mean standard deviation for each QoI and then we randomly generated 100 extra artificial simulations per simulation used for training. The added simulations were taken from the following uniform distribution:

\[
\text{QoI}_i^k(t) \sim \text{QoI}_i^j(t) + U[-\bar{\sigma}_i, \bar{\sigma}_i]
\]

where QoI_i^k(t) is the artificial simulation k with k = 2, 3, …, 100 corresponding to the original simulation j with j = 1, 2, …, 70 for the ith QoI,
Figure 4: Structure of each network that makes up the recursive neural network.

with $i = 1, 2$. $\mathcal{U}[-\bar{\sigma}_i, \bar{\sigma}_i]$ is the random uniform distribution with lower limit $-\bar{\sigma}_i$ and upper limit $\bar{\sigma}_i$, where $\bar{\sigma}_i$ is the mean standard deviation for $\text{QoI}_i$.

3. The last approach was to shuffle the data and add it as training data 100 times (i.e 7,000 data points in total).

Figure 5 shows the range averaged mean RMSE (RMSE for simplicity) for both QoIs comparing the error of the baseline model (the mean) against the error of the ML model. Figure 5 shows in black the RMSE of the baseline model and in red the RMSE associated with the ML model. Figure 5(a) shows the RMSE for $\text{QoI}_1$ as a function of time and Figure 5(b) for $\text{QoI}_2$. Later, in Figure 8(a) we show the training data for $\text{QoI}_1$ and it can be seen the challenging part about training $\text{QoI}_1$ is the crack length jump at approximately $0.8 \mu$s, here is where it lays most of the variability. Before and after this time $\text{QoI}_1$ is almost constant. As we see in Figure 5(a), the ML model outperforms the baseline model around that time. Similarly, in Figure 8(b) we show the training data for $\text{QoI}_2$ and it can be seen the challenging part about training $\text{QoI}_2$ is in the time range $0.8 \mu s$. 


to 1.2 \(\mu s\), here is where it lays most of the variability. As we see in Figure 5(b), the ML model outperforms the baseline model in this range.

Figure 5: RMSE as a function of time calculated using the test data. In black is shown the RMSE of the baseline model and in red the RMSE of the ML model. Where the data variation is higher the ML model outperform the baseline model.
2.4 Approach to improve FLAG damage model using crack statistics

FLAG, being a continuum model, does a poor job in modeling stresses for brittle material. One of the reasons, as alluded in the previous paragraphs, is due to the presence of micro-cracks present in the material. The continuum model does not account for damage occurring from the evolution of micro-cracks and hence various upscaling procedures must be developed for correcting the stresses predicted by FLAG.

We take a statistical approach to approximating the damage caused by crack evolution in the target plate utilized in FLAG. In such a framework, Vaughn et al. [?] developed a modified version of the effective moduli model of Ju and Chen [16] which approximates the compliance of the damaged material as follows

\[ S = S^0 + S^* \]  

(6)

where \( S^0 \) is the elastic compliance tensor, which in two dimensions is defined as

\[
S^0 = \frac{(1 + \nu)}{E} \begin{bmatrix} 1 - \nu & -\nu & 0 \\ -\nu & 1 - \nu & 0 \\ 0 & 0 & 2 \end{bmatrix},
\]

(7)

and \( S^* \) is a compliance tensor correction that is added to the elastic compliance tensor to account for micro-crack effects which has the following expression

\[
S^* = \bar{L} \pi \frac{(1 - \nu^2)n}{E} \int_\alpha \int_\Theta a^2 M_0 f(a) f(\theta) d\theta da
\]

(8)

where \( f(a) \) and \( f(\theta) \) are PDFs of crack length and orientation for a given time step, \( n \) is the number of cracks at the given time step, and \( a \) is the crack length. The coordinate transformation tensor \( M_0 \) relates the local coordinates of the cracks to the global coordinate system. The interested reader can refer to [16] for a detailed derivation of \( M_0 \). The dimensionless parameter \( \bar{L} \) added by Vaughn et al. [?] accounts for the finite length of the material domain and is determined using the following relation.

\[
\bar{L} = \left( \frac{L}{L - \bar{a}} \right)
\]

(9)

where \( L \) is the width of the target plate and \( \bar{a} \) is the horizontal projection of the longest crack in the domain.

The nature of fracture in flyer plate experiments leads to a non-homogeneous damage distribution where a concentrated region of damage forms across the midspan of the target plate while the majority of the plate remains relatively undamaged. To account for this a stress-based damage evolution scheme developed by Larkin et al. 2020 is employed to evaluate the damage level of each material zone. The primary failure mechanism for the target plate is mode I...
crack opening. Therefore, the maximum tensile stress from the statistical data set is correlated to a compliance correction tensor generated at the same time step. The appropriate compliance correction is identified for each zone by comparing a stress estimate to the maximum stress statistics. Then the stress value is updated using the identified effective moduli using the following relations.

\[ \sigma = C : \epsilon \]  
\[ C = (S)^{-1} \]

where \( \sigma \) is the corrected stress, \( \epsilon \) is the elastic strain, and \( C \) is the effective stiffness tensor.

Previously, crack length and orientation data, as well as maximum tensile stress data, were extracted directly from the mesoscale simulations computed using HOSS. The crack data were used to generate the distributions \( f(a) \) and \( f(\theta) \) in equation 8 to inform the effective moduli model in FLAG. However, running a mesoscale simulation like HOSS in real-time is prohibitively expensive. Thus, in this work, we will use machine learning to emulate mesoscale tensile stresses and the crack statistics for the distributions \( f(a) \) and \( f(\theta) \). Our machine learning model will emulate how maximum stresses evolve in time in the presence of micro-cracks using data from HOSS simulation corresponding to various initial conditions. We will also model how each crack length evolves in time. This will enable us to estimate the PDF \( f(a) \).

3 Results

3.1 Dimensionality reduction

As discussed in section 2.4, we will use machine learning to emulate mesoscale tensile stresses and the crack statistics for the distributions \( f(a) \) and \( f(\theta) \). This will require emulating the maximum tensile stresses and the evolution of cracks in the mesoscale simulation given by HOSS.

Figure 6 shows the crack length PDF for one simulation (blue histogram), the crack length PDF for 100 simulations (orange histogram), and the initial crack length distribution (continuous red line). Subfigure 6(a) corresponds to the initial time (\( t = 0 \)), Subfigure 6(b) and Subfigure 6(c) correspond to the intermediate times (0.4\( \mu s \) and 0.8\( \mu s \), respectively), and Subfigure 6(d) to final time (1.2\( \mu s \)). At the initial time, the 200 crack lengths vary between 0.1\( mm \) and 0.3\( mm \). Until 0.8\( \mu s \) it cannot be observed crack growth. When the material failure occurs (\( \approx 0.8\mu s \)) we observe crack nucleation (generation of approximately 150 new cracks per simulation) and also crack growth. Because cracks coalesce at later times, at the final time we observe only around 160 cracks. As can be seen in Figure 6, regardless of the time, the initial distribution of cracks maintains and only a few cracks per simulation grow (less than five cracks) more than 0.3 mm. For clarity purposes the plots are shown on the same scale, however, after \( \approx 0.8\mu s \) the longest crack has a length of \( \approx 28.8mm \), which is the
width of the target plate. Nevertheless, Subfigure 6(c) and Subfigure 6(d) show the mentioned crack length growth (lengths longer than 0.3 mm).

![Graphs showing crack length distribution](image)

Figure 6: Evolution of the crack length distribution in the interval $[0, 0.5]\text{mm}$. Near the time when the material fails ($\approx 0.8\mu\text{s}$) we can see substantial crack nucleation and crack length growth.

While emulating the evolution of maximum tensile stresses can be challenging, emulating the evolution of each crack is a particularly difficult task for the following reasons:

1. The number of cracks changes as a function of time due to nucleation and coalescence of cracks.
2. The distribution changes substantially in the tails.
3. The number of training examples to simulate a full-time series starting from just initial condition need to sufficiently rich.

A close look at the distribution of cracks in Figure 3 shows that the majority of the cracks remain small, hence the main distribution does not change
substantially as a function of time. However, there are a few cracks that grow until, of course, the material fails.

Thus, emulating the complete distribution of crack lengths \( f(a) \) is extremely difficult. Instead, by reasons hinted above, we will only focus on the longest crack. This can be thought of as a dimension reduction technique where we replace a PDF by a Dirac measure \( \delta_{LC}(x) \) centered at the longest crack (denoted here as LC). In terms of the correction tensor discussed in Section 4, equation 8 is modified as follows:

\[
\hat{S}^* = \bar{L} \left( 1 - \frac{\nu^2}{n} \right) \int_a \int_\Theta a^2 M_0 \delta_{LC}(a) f(\theta) d\theta da
\]  

Figure 7 shows a summary of the reduced QoIs.

Figure 7 shows a summary of the reduced QoIs.

Figure 7: Dimensionality reduction of the problem.

Figure 8 shows the QoIs chosen to train the ML model as a function of time. Figure 8(a) shows the evolution of the length of the longest crack. As can be seen in the figure, the variability from simulation to simulation occurs close to 0.8 \( \mu s \). Figure 8(b) shows the maximum tensile stress as a function of time. This quantity shows more variability from simulation to simulation in time. Having this variability will allow the ML model to have better predictability.
Figure 8: Evolution of crack lengths distributions used to train, validate and test the neural network. Colored lines are the 150 randomly generated distributions for each time step $t$, black lines represent the mean given the 150 samples per time step.
3.2 Experimental validation of ML combining FLAG

The first step was to test if only having information about the longest crack will give predictive results. For this, we compare FLAG output using HOSS information of all cracks against FLAG output using HOSS information only of the longest crack. As Figure 9 shows, the performance of FLAG using single crack information from HOSS compares with HOSS and with the experimental data.

![Figure 9: Evolution of the shock wave velocity at the middle rear of the target plate. The performance of FLAG using single crack information from HOSS compares with HOSS and with the experimental data.](image)

The second step was to use the ML model train with HOSs instead of HOSS information directly. As Figure 10 shows, we found that FLAG + ML gives results almost identical to FLAG + HOSS (longest crack) and that we capture the shock bounce as HOSS does, improving remarkably the predictions given by FLAG (no corrections).
Figure 10: Evolution of the shock wave velocity at the middle rear of the target plate. FLAG + ML improves remarkably the predictions given by FLAG (no corrections).
4 Conclusion

This work is aimed at leveraging machine learning to simulate materials undergoing brittle fracturing in such a way that the accuracy is comparable to a model (such as HOSS) that explicitly simulates the evolution of each crack while maintaining the speed of a model (such as FLAG) that accounts for the effect of cracks using elastic moduli. By informing FLAG directly with HOSS data, we found that the machine learning model only needed to accurately predict the length of the longest crack and the maximum $y y$ component of the stress. This significantly reduces the dimensionality of the predictions that the ML model must make, since the number of cracks, the lengths of each crack, and their orientations are ostensibly needed. Our approach relied on data from HOSS simulations to conceptualize (via the dimensionality reduction) and train an appropriate machine learning model.

The machine learning model can then be used to inform a FLAG simulation. The result of combining the machine learning model with FLAG retains an accuracy that is comparable to HOSS, but is $\sim 10^4$ times faster. This speed-up is critical since the HOSS simulations are very computationally expensive and many simulations are often required to, e.g., estimate uncertainty or perform calibration. We validated the combination of machine learning and FLAG on a flyer plate experiment. We note that the flyer plate experiment was similar to the HOSS simulations used to train the machine learning model. In future work, we aim to explore the performance of this approach under more diverse loading conditions.

References

[1] A constitutive model for metals applicable at high-strain rate 51(3), 1498–1504 (1980)

[2] Aida, T., Walter, J.W., Aslam, T.D., Short, M.: Verification of 2-d detonation shock dynamics in conjunction with los alamos lagrangian hydrocode. Tech. Rep. LA-UR-12-20792, Los Alamos National Lab, Los Alamos, NM, USA (2013)

[3] Black, W., Denissen, N., McFarland, J.: Evaporation effects in shock-driven multiphase instabilities. ASME J. Fluids Eng. 139(7), 071204 (2017)

[4] Brownlee, J.: Deep learning for time series forecasting (2018)

[5] Burton, D.: Connectivity structures and differencing techniques for staggered-grid free-lagrange hydrodynamics. Tech. Rep. UCRL-JC-110555, Lawrence Livermore National Laboratory, Livermore, CA (1992)

[6] Burton, D.: Consistent finite-volume discretization of hydrodynamic conservation laws for unstructured grids. Tech. Rep. UCRL-JC-118788, Lawrence Livermore National Laboratory, Livermore, CA (1994)
[7] Burton, D.: Multidimensional discretization of conservation laws for unstructured polyhedral grids. Tech. Rep. UCRL-JC-118306, Lawrence Livermore National Laboratory, Livermore, CA (1994)

[8] Burton, D., Morgan, N., Charest, M., Kenamond, M., Fung, J.: Compatible, energy conserving, bounds preserving remap of hydrodynamic fields for an extended ale scheme 355, 492–533 (2018)

[9] Cady, C., Adams, C., Prime, M., Hull, L., Addessio, F., Bronkhorst, C., Brown, E., Liu, C., Sisneros, T., Brown, D., et al.: Characterization of S200-F beryllium using shock loading and quasi-static experiments. Tech. rep., LA-UR-11-06976, Los Alamos National Laboratory (2011)

[10] Cady, C.M., Adams, C.D., Hull, L.M., Gray, G., Prime, M.B., Addessio, F.L., Wynn, T.A., Papin, P., Brown, E.: Characterization of shocked beryllium. In: EPJ Web of Conferences, vol. 26, p. 01009. EDP Sciences (2012)

[11] Caldwell, W.K., Hunter, A., Plesko, C.S., Wirkus, S.: Verification and validation of the flag hydrocode for impact cratering simulations. Journal of the American Ceramic Society 3(3), 031004 (2018)

[12] Cooley, J., Olson, R., Oro, D.: Modeling and analysis of high-explosive driven perturbed plate experiments at los alamos. Journal of Physics: Conference Series 500(15), 152003 (2014)

[13] Fredenburg, D., T.D., A., L.S., B.: A gruneisen equation of state for tpx: Application in flag. Tech. Rep. LA-UR-15-28533, Los Alamos National Laboratory, Los Alamos, NM (2015)

[14] Fung, J., Harrison, A.K., Chitanvis, S., Margulies, J.: Ejecta source and transport modeling in the flag hydrocode. Computers & Fluids 83, 177–186 (2013)

[15] Hunter, A., Moore, B.A., Mudunuru, M.K., Chau, V.T., Miller, R., Tchoua, R., Nyshadham, C., Karra, S., Malley, D.O., Rougier, E., et al.: Reduced-order modeling through machine learning approaches for brittle fracture applications. arXiv preprint arXiv:1806.01949 (2018)

[16] Ju, J., Chen, T.: Effective elastic moduli of two-phase composites containing randomly dispersed spherical inhomogeneities. Acta Mechanica 103(1-4), 123–144 (1994)

[17] Kachanov, L.: Introduction to continuum damage mechanics, vol. 10. Springer Science & Business Media (2013)

[18] Knight, E., Rougier, E., Lei, Z.: Hybrid optimization software suite (hoss)-educational version. Tech. rep., Technical Report LA-UR-15-27013, Los Alamos National Laboratory (2015)
[19] Knight, E., Rougier, E., Munjiza, A.: Lanl-csm: Consortium proposal for the advancement of hoss. Presentation, LA-UR-13-23409 pp. 05–09 (2013)

[20] Krajcinovic, D., Lemaitre, J.: Continuum damage mechanics: theory and applications. Springer (1987)

[21] Krajcinovic, D., Mastilovic, S.: Some fundamental issues of damage mechanics. Mechanics of Materials 21(3), 217–230 (1995)

[22] Moore, B.A., Rougier, E., OMalley, D., Srinivasan, G., Hunter, A., Viswanathan, H.: Predictive modeling of dynamic fracture growth in brittle materials with machine learning. Computational Materials Science 148, 46–53 (2018)

[23] Munjiza, A.: Discrete elements in transient dynamics of fractured media. Ph.D. thesis, Swansea University (1992)

[24] Munjiza, A., Knight, E.E., Rougier, E.: Large strain finite element method: a practical course. John Wiley & Sons (2015)

[25] Munjiza, A., Owen, D., Bicanic, N.: A combined finite-discrete element method in transient dynamics of fracturing solids. Engineering Computations 12(2), 145–174 (1995)

[26] Munjiza, A.A.: The combined finite-discrete element method. John Wiley & Sons (2004)

[27] Munjiza, A.A., Knight, E.E., Rougier, E.: Computational mechanics of discontinua. John Wiley & Sons (2011)

[28] Rougier, E., Knight, E., Munjiza, A.: Lanl-csm: Hoss-munrou technology overview. Presentation, LA-UR-13-23422 pp. 05–10 (2013)

[29] Rougier, E., Knight, E.E., Broome, S.T., Sussman, A.J., Munjiza, A.: Validation of a three-dimensional finite-discrete element method using experimental results of the split hopkinson pressure bar test. International Journal of Rock Mechanics and Mining Sciences 70, 101–108 (2014)

[30] Rougier, E., Munjiza, A., John, N.: Numerical comparison of some explicit time integration schemes used in dem, fem/dem and molecular dynamics. International Journal for Numerical Methods in Engineering 61(6), 856–879 (2004)

[31] Schwarzer, M., Rogan, B., Ruan, Y., Song, Z., Lee, D.Y., Percus, A.G., Chau, V.T., Moore, B.A., Rougier, E., Viswanathan, H.S., et al.: Learning to fail: Predicting fracture evolution in brittle material models using recurrent graph convolutional neural networks. Computational Materials Science 162, 322–332 (2019)
[32] Srinivasan, G., Hyman, J.D., Osthus, D.A., Moore, B.A., OMalley, D., Karra, S., Rougier, E., Hagberg, A.A., Hunter, A., Viswanathan, H.S.: Quantifying topological uncertainty in fractured systems using graph theory and machine learning. Scientific reports 8(1), 11665 (2018)

[33] Steinberg, D.J.: Equation of state and strength properties of selected materials. Tech. Rep. UCRL-Ma-10639, Lawrence Livermore National Laboratory (1996)

[34] Tonks, D., Bronkhorst, C.A., Bingert, J.: A comparison of calculated damage from square waves and triangular waves. In: AIP Conference Proceedings, vol. 1426, pp. 1045–1048. AIP (2012)

[35] Tonks, D.L., Bingert, J., Livescu, V., Luo, S., Bronkhorst, C.: Mesoscale polycrystal calculations of damage in spallation in metals. In: EPJ Web of Conferences, vol. 10, p. 00006. EDP Sciences (2010)

**Funding Sources**

This work was partially performed under U.S. Government contract 89233218CNA000001 for Los Alamos National Laboratory (LANL), which is operated by Triad National Security, LLC for the U.S. Department of Energy/National Nuclear Security Administration. Approved for public release LA-UR-20-20148.