# Laser Pulse Shape Prediction on the OMEGA EP Laser System Using a Convolutional Neural Network

**Michael Yu**

Pittsford Mendon High School

Advisor: Mark Guardalben

Laboratory for Laser Energetics

University of Rochester

October 2026

## Abstract:

Future laser systems relevant to Inertial Fusion Energy (IFE) will operate at higher repetition rates than current fusion-class laser systems, requiring real-time feedback control and optimization of laser pulse shape and energy. In addition, these laser systems will rapidly generate enormous amounts of data that must be strategically selected to maintain calibration of laser performance models and to identify unsafe operating conditions. Accurate laser predictive models must therefore be developed that support the higher repetition rates. Laser performance prediction in the long-pulse beamlines of the OMEGA EP laser system is currently done using a physics-based model, *PSOPS*. A convolutional neural network was developed and trained using simulated data from *PSOPS* to predict both pulse shape and energy and achieved a mean absolute energy error of 1.89% and an RMS energy error of 4.85% relative to PSOPS predictions, with approximately 100x faster processing speed. These results are an encouraging step in the development of laser predictive models that support future IFE laser facilities.

# 1. Introduction

The mission of the University of Rochester Laboratory for Laser Energetics (LLE) is to support the development of technologies in pursuit of producing a sustainable energy source through inertial confinement fusion. While current laser systems at the National Ignition Facility (NIF) and LLE fire singular shots, a future inertial fusion energy system would require shots to fire in rapid succession.

At LLE, the OMEGA EP laser system [1] is a 4-beam addition to the OMEGA Laser Facility (Fig. 1). This addition allows for over 1 kJ of energy to be delivered to a target in a short nanosecond pulse. The pulses on the OMEGA EP laser system are created from a continuous laser beam using a waveform generator. This pulse is then amplified through front-end amplifiers before being amplified several times by the main beamline amplifiers that use flashlamp-pumped Nd:glass. Before the pulse reaches the target, the frequency of the beam is tripled from infrared to ultraviolet. The amplification and frequency conversion processes of the pulse are highly nonlinear and therefore require a specialized predictive model to accurately create specific pulse shapes and energies for OMEGA EP.

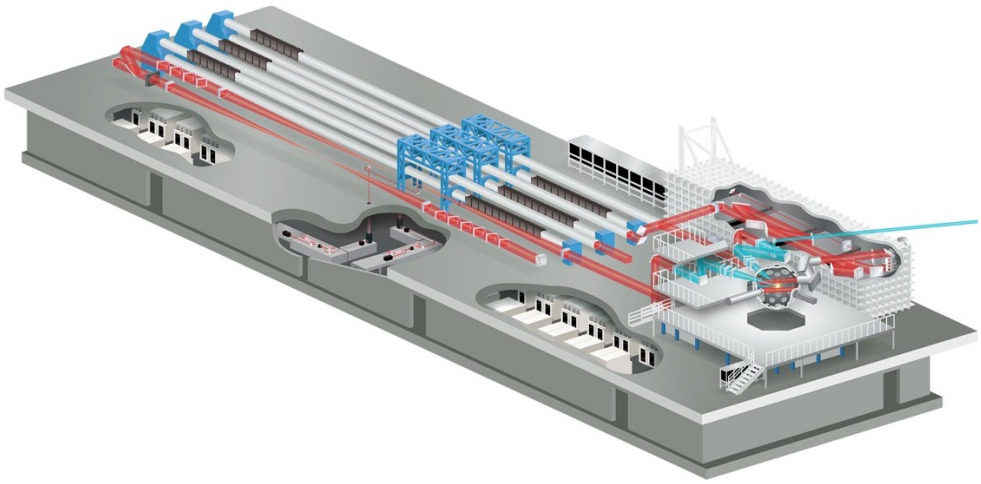


Figure 1: The *OMEGA EP Laser System with its four NIF-scale beamlines. From Ref. 2*

Currently, the OMEGA EP laser system uses a physics-based predictive model PSOPS to do both the forward and backward prediction of the laser pulse shape and energy. *PSOPS* uses *MATLAB* to provide real time optimization of the OMEGA EP shots. A backwards prediction is needed to obtain an initial estimate of the input pulse shape and energy needed to be input. Then a forward and backward prediction optimization loop can be repeated to optimize the input pulse so that the desired output pulse shape and energy are achieved [3].

Although *PSOPS* is accurate and is relatively quick, taking 5 seconds to predict a single shot, faster predictions are needed to sustain a future inertial confinement fusion energy system. Future inertial fusion laser systems will operate at higher repetition rates than current fusion-class laser systems—such as the NIF—requiring real-time feedback control and optimization of laser pulse shape and energy. These laser systems will also rapidly generate enormous amounts of data that must be strategically selected to maintain laser performance models and to identify unsafe operating conditions.

Neural networks are a type of model which uses non-linear functions called activation functions to help the model learn complex patterns beyond linear relations. A neural network consists of nodes, weights, biases, and activation functions (Fig. 2). Each of the nodes serves as a connection point in the neural network. A neural network functions by weighing the different connections between the different nodes and inputs. Then, at each node there is an activation function which introduces non-linearity into the model. Some examples of activation functions are leaky ReLU, ReLU, sigmoid, logistic, and tanh. By capturing these mathematical nuances into the model, a neural network could be a suitable choice to do the forward and backward prediction that *PSOPS* currently does.

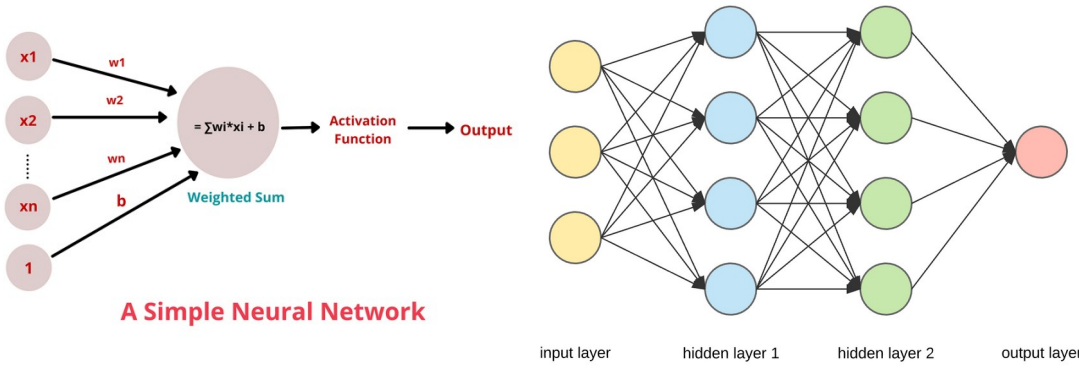



Figure 2: *A generalized simple neural network structure showing the pieces such as the weights, nodes, activation function, and hidden layers. From Ref. 4.*

These neural networks must first be trained on prelabeled data so the model can recognize patterns and intricacies that we want the model to predict. This is done through an optimization process using partial derivatives where the model is attempting to optimize the weights and biases so that it can best predict the data, without encountering a phenomenon known as overfitting. Overfitting generally occurs when too much training has occurred, and the model has learned to recognize the specific data in the dataset instead of the features of

the dataset. This can be seen by the model only performing well on the training data, and poorly on all other data.

One special type of neural network is called a convolutional neural network (CNN). This type of neural network uses a small piece of the data called a kernel, which is then used to help identify special sections of the input data called features (Fig. 3). While the input matrix might be 1024 x 1024 for example, the kernel would be a smaller subset of this matrix such as a 3 x 3 chunk. The kernel would start in the top left and slide across the dataset, multiplying the input data by the kernel's weights to form a feature. This process is called a convolution and is continued until the entire dataset has been traversed. When we have all our features, it forms a feature map. Then, combining all our feature maps together we get a hidden layer. We can then repeat this convolution process on the new hidden layers we formed to capture higher level details. If our input matrix was a 1024 x 1024-pixel image, these earlier feature maps could pick up smaller features such as vertical lines or horizontal edges, while further convolution can pick up complex details such as eyes, wheels, or other objects. The process of identifying features gives CNNs a unique advantage compared to more traditional neural networks by allowing CNNs to better learn and understand complex and distinctive patterns within the data.

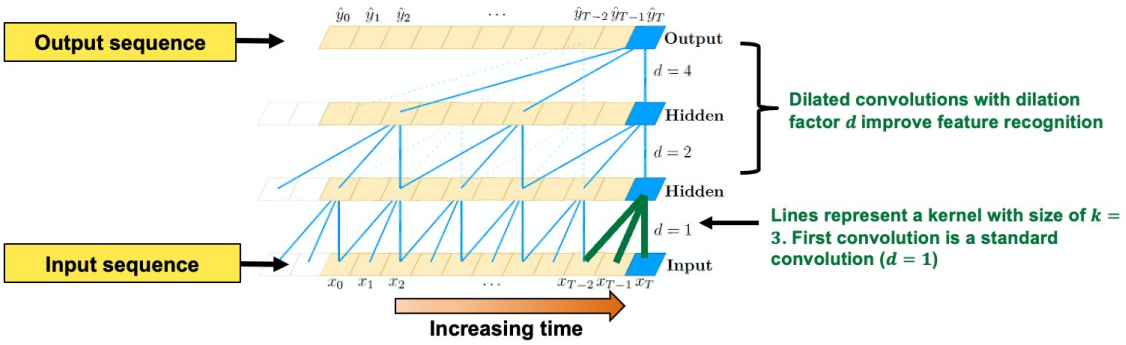



Figure 3: *A convolutional neural network broken down into smaller pieces, showing how the kernel connects layers of the neural network together.*

To optimize the training of a machine learning model, there are special parameters that we can optimize called hyperparameters. Some examples of hyperparameters include numbers like epochs, dropout factor, the initial learning rate, training/test data split, and the minibatch size. There are also special hyperparameters that are specifically for CNNs such as the number of segments, blocks, filter size, and number of filters. One of the most important factors is the number of epochs, which is the number of times the model is run through the training dataset. Logically the model will perform better with a higher number of epochs, but a balance must be found between training time, results, and overfitting. One of the other hyperparameters, dropout factors, can help with overfitting. The dropout factors are used to randomly drop a connection between nodes in the CNN, which prevents the model from becoming too reliant on a certain connection in the model, therefore helping the model become more resistant to overfitting. The training/test data split is a factor that splits the dataset, so the model can be tested on data that it has never seen before. The initial learning rate primarily affects how long training takes. With a higher initial learning rate the solution is likely to converge faster initially but may oscillate or diverge around the optimal solution.

In contrast, with a smaller initial learning rate, the algorithm will take longer to reach the minimum and has a higher risk of getting stuck in a local minimum rather than the global minimum. The minibatch size affects both the training speed and the overfitting of the model: with a higher minibatch size we can parallelize our training data, but this generally results in a smoother gradient which can cause poor generalization. While smaller minibatches result in computationally slow training, they also have a noisier gradient which results in less overfitting and better results on the unseen data.

## 2. Architecture of Prediction Pipeline

A convolutional neural network was investigated to be used for pulse shape and energy prediction on the OMEGA EP laser system. Previously multi-layer perceptron has been investigated for its use in energy prediction on the OMEGA EP laser system [5]. Multi-layer perceptron is like CNNs with the largest difference being the lack of convolution and that each neuron—with a neuron being a value in the layer—is fully connected to every neuron in the previous layer. This previous research showed a 1000x increase in processing speed with an error of 0.05%. To train the CNN, there is a process of forward and backward propagation. First, the model does the forward propagation process where it is fed the shape and energy of an infrared (IR) pulse as input to the CNN for prediction of the laser output. The CNN results in an output ultraviolet (UV) pulse from which we can extract the output energy and shape. Next, we perform backward propagation to improve the model. This is done by finding the loss between the predicted laser output from the CNN and the ground truth data—the prediction from *PSOPS*. After the error is found using a certain loss function, a gradient of the weights and biases is then calculated in a process known as gradient descent to attempt to minimize this error. The weights and biases are then updated according to this

gradient. The loss function that we are using is called mean squared error (MSE), which effectively sums the squares of the differences between the predicted and actual pulses. The number of differences summed depends on the minibatch size. The model will then try to optimize this by attempting to minimize the MSE (Fig. 4).

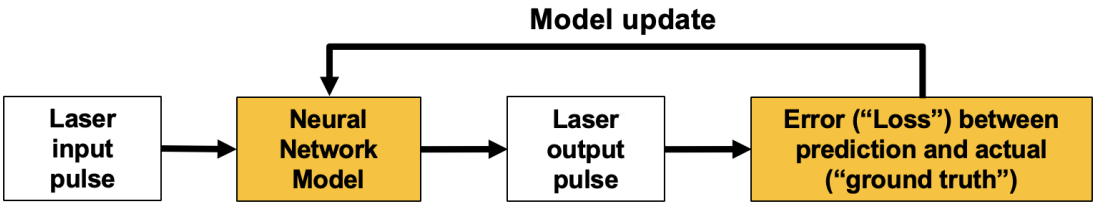



Figure 4: *A simplified model demonstrating how a neural network is trained on laser pulse data.*

A training/test split of 80/20 was chosen with the learning rate dropping every 900 epochs. The models were trained and tested on the GPU node of the University of Rochester's Deluge high performance cluster (HPC).

## 3. Predictions on Normalized Pulses

To start, the model was first trained on a normalized version of the pulses used in OMEGA EP. This makes it significantly easier to train the models because the pulses all have their power normalized to 1 GW. *MATLAB* was used to build and test our CNN. There is an existing suite of machine learning tools in *MATLAB* called the Deep Learning Toolbox which allows us to utilize built-in functions to assist with the creation of the model.

Although the pulses are normalized, the pulses still differ in time. For the normalized pulses, good results were found with just 100 epochs of training, with further epochs

resulting in stagnation in error reduction. Two random pulses from the dataset were sampled to highlight the average pulse prediction done by the CNN (Fig. 5).

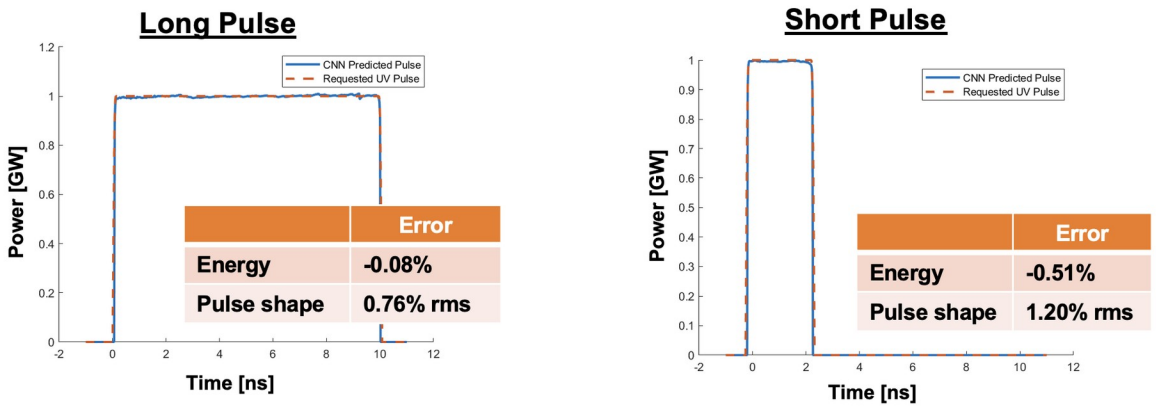



Figure 5: *A comparison between predicted normalized long and short pulses from the CNN and the expected shape and energy.*

It can be observed that the prediction is quite good even just upon visual inspection. When looking into the percentage error of the energy prediction—which is found through integrating the pulse—and the pulse shape compared to the ground truth, there is less than 1% error for the longer pulse and around 1% for the shorter pulse. This result makes sense, as even minor deviations have a larger impact on shorter pulses since these pulses have fewer data points to predict on. This result took 25 minutes of training on the HPC cluster. One observation that can be made is that both predictions have noise in their result. Overall, the CNN model was able to successfully predict normalized pulses compared with the PSOPS ground truth data.

# 4. Predictions on Unnormalized Pulses

## 4.1 Predictions using an Unweighted Loss Function

After the model has been demonstrated to work well on the normalized pulses, the next step is to train the model on unnormalized pulses. These pulses can vary in power from 100 to over 1000 GW so more epochs are needed to adequately capture the more nuanced features of the unnormalized pulses. The unnormalized pulses were trained with 500 epochs and different hyperparameters such as a lower dropout factor and learning drop rate. After performing multiple rounds of hyperparameter tuning, the model is quite good at predicting the short, high-power pulse, but significantly worse at predicting the long, low-power pulse (Fig. 6).

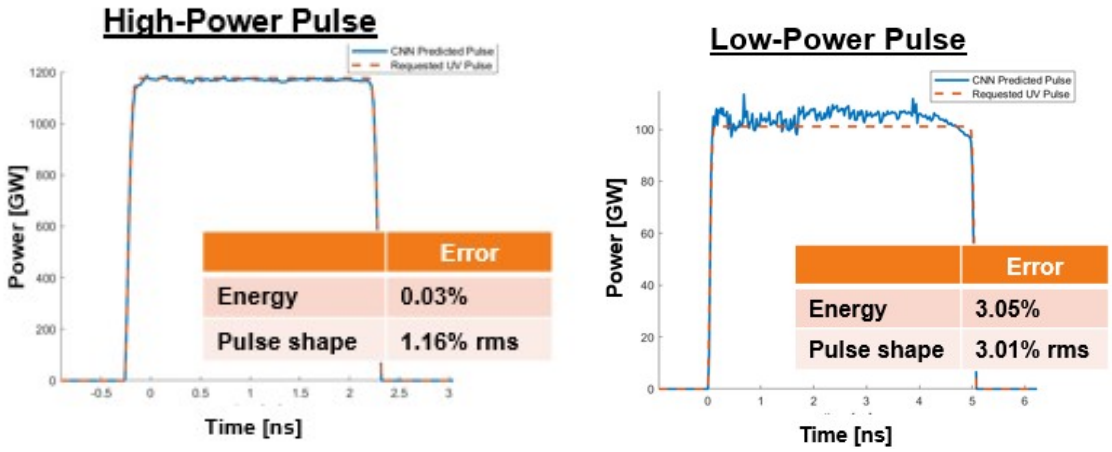



Figure 6: *A comparison between predicted unnormalized long (low-power) and short (high power) pulses from the CNN and the expected shape and energy.*

Upon further investigation, the root cause of this problem was discovered—the loss function. This is a result of the noise that was seen with the normalized pulse prediction. The MSE function uses squared power differences rather than relative differences. A 5 GW

deviation therefore contributes the same squared error for a 1200 GW pulse as for a 100 GW pulse, even though its relative size is very different. For the higher-power pulse, 5 GW is ~0.42% of the max power while 5 GW is ~5% of the max power for the lower-power pulse. This leads to the conclusion that an alternative loss function could be used to help reduce the noise for lower-energy pulses.

## 4.2 Predictions using a Weighted Loss Function

Several loss functions were considered, including mean absolute percentage error. The peak-normalized formulation illustrated in Fig. 7 divides the prediction difference at each time step by the peak power of the corresponding PSOPS pulse before squaring and averaging. This is equivalent to dividing that pulse's MSE by the square of its peak power. It gives equal relative deviations on low- and high-power pulses comparable contributions to the loss. The resulting weighted mean squared error (WMSE) is dimensionless.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(p_i - t_i)^2 \qquad\qquad \mathrm{WMSE} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{p_i - t_i}{P_{\max}}\right)^2$$

$p_i$: predicted power  $t_i$: PSOPS power  $n$: number of time samples
$P_{\max} = \max_i(t_i)$ for each pulse; the loss is then averaged across pulses

Figure 7: *The illustrated peak-normalized loss divides each power difference by the pulse peak power before squaring, equivalent to dividing the MSE by the square of the peak power.*

There are clear improvements from the addition of the weighted loss function (Fig. 8). Just from visual inspection, the weighted loss function's prediction is closer to the actual requested pulse compared to the unweighted function. The magnitude of the energy

prediction error decreased from 3.05% to 0.78% (the weighted prediction had a signed error of -0.78%) and the error of the pulse shape prediction improved from 3.01% to 2.14%. Both predictions still suffer from a problem of noise, which can be seen by the wavy nature of the CNN-predicted pulse and the still-high pulse shape error of ~2%. The pulse shape error is a statistic that allows us to quantify the amount of noise in the predicted pulse.

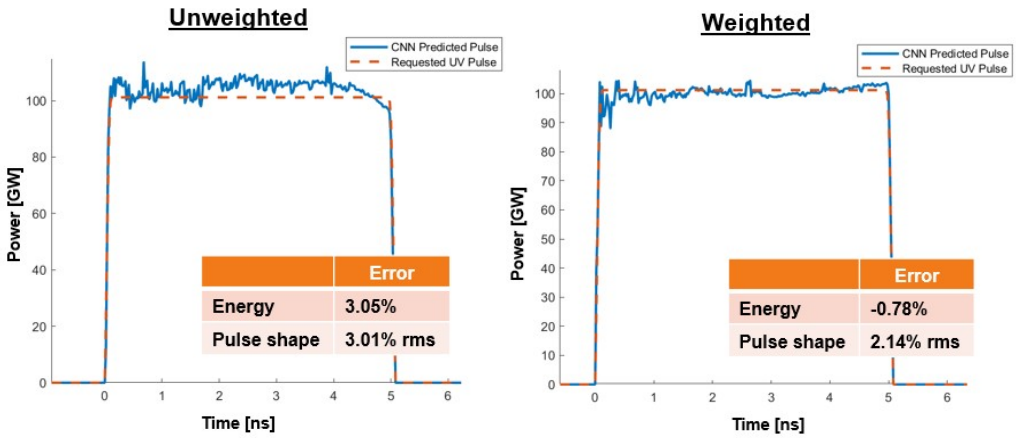



Figure 8: *A comparison of expected shape and energy between weighted and unweighted loss functions on unnormalized long pulses from the CNN.*

Individual pulse examples are useful for investigating specific cases, while the 90 saved predictions per model reveal overall trends (Fig. 9). For each pulse, the signed percentage energy error is $e = 100\dfrac{E_{CNN} - E_{PSOPS}}{E_{PSOPS}}$. The signed means reported in Fig. 9 are +1.52% for the unweighted model and +0.23% for the weighted model. Each is calculated as $\bar{e} = \dfrac{1}{N}\sum_{i=1}^{N} e_i$, where N = 90 is the number of pulses. They measure bias, but allow positive and negative errors to cancel.

To quantify error magnitude, the mean absolute error is $MAE = \frac{1}{N}\sum_{i=1}^{N}|e_i|$, and the RMS energy error is $RMS = \sqrt{\frac{1}{N}\sum_{i=1}^{N} e_i^2}$. The unweighted and weighted result sets have mean absolute energy errors of 2.22% and 1.89%, respectively, and RMS energy errors of 4.46% and 4.85%. Thus, the weighted result set has lower average bias and mean absolute error, but higher RMS energy error. The saved sets contain different PSOPS energy distributions, so a controlled comparison also requires evaluation on a common held-out set.

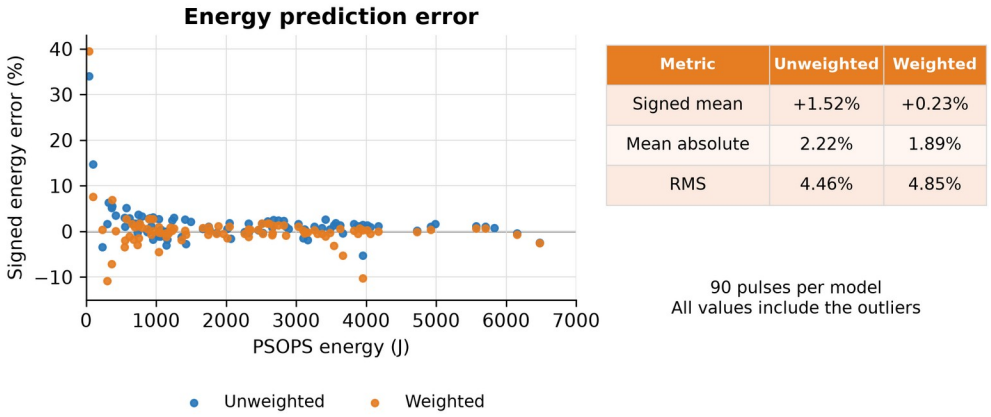
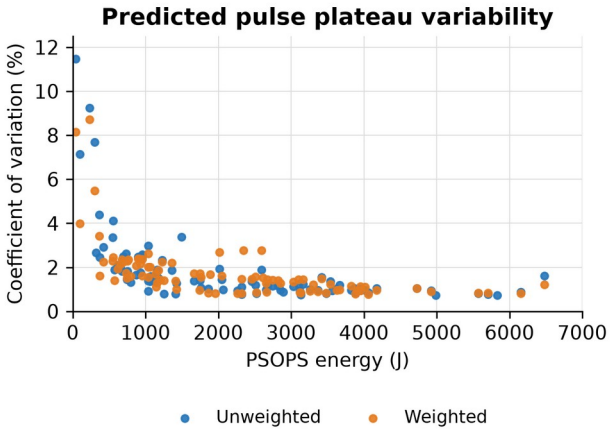



Figure 9: *Signed percentage energy errors for 90 pulses per model, with signed mean, mean absolute, and RMS energy errors. All summary statistics include the outliers.*

The saved results also quantify variability in the predicted pulse plateau (Fig. 10). This statistic is the coefficient of variation (CV): the standard deviation of the predicted power divided by its mean, expressed as a percentage. The plateau includes time samples where the PSOPS power exceeds 80% of its peak. The mean CV across the 90 pulses decreased from 1.84% for the unweighted model to 1.75% for the weighted model. This describes variation within the predicted plateau; it is distinct from RMS differences between

predicted and PSOPS pulse shapes and can also reflect real variation in the requested pulse. The individual pulse examples in Fig. 8 still show residual fluctuations.




| Model | Mean CV |
|---|---|
| Unweighted | 1.84% |
| Weighted | 1.75% |

CV = standard deviation / mean
Calculated on the pulse plateau
90 pulses per model

Figure 10: *Coefficient of variation of the predicted pulse plateau for 90 pulses per model. The table gives the mean CV across pulses; it does not give RMS pulse-shape prediction error.*

## 5. Processing Speed

*PSOPS* is a physics-based model and requires numerous complex computations and simulations to perform a prediction. Performing the optimization with *PSOPS* can take up to 5 seconds for the prediction of a single pulse. Laser systems for future inertial confinement fusion would have to run around 1 Hz, which requires processing improvements with the performance of laser pulse shape and energy prediction.

Neural networks are promising in this respect. Since the intensive aspect of neural networks is done beforehand with the training of the model, we expect a significant reduction in processing time of pulse shape and energy prediction compared to *PSOPS*. This was

observed, with the final CNN taking around 50 ms to make a single prediction. This demonstrates a nearly 100-fold increase in performance.

This performance results from the hours of training that neural networks require—ranging from 2-6 hours depending on the exact hyperparameters such as minibatch size used for training. There can be further optimizations such as pruning to allow for even faster pulse shape prediction. Pruning works through removing certain connections in the neural network to increase prediction speed while also decreasing the model size. Generally, pruning allows us to make a less computationally expensive model. There are numerous pruning algorithms such as variational dropout, regularization methods (L0 or Hoyer), weight reintroduction (RigL), and gradual magnitude pruning (GMP). Out of these methods, GMP generally appears to be the best, matching or beating all other approaches [6].

## 6. Conclusion

Using simulated data from PSOPS, a CNN was trained to predict both pulse shape and energy. The weighted model made predictions approximately 100x faster than PSOPS. Across the 90 saved predictions per model, the weighted result set had a signed mean energy error of +0.23% and a mean absolute energy error of 1.89%, compared with +1.52% and 2.22% for the unweighted result set. However, its RMS energy error was higher: 4.85% compared with 4.46%. These values quantify agreement with PSOPS simulations rather than accuracy against experimental measurements. Evaluation of both models on a common held-out set is needed to isolate the effect of weighting.

With previous research highlighting a multilayer perceptron being used for energy prediction and this paper demonstrating the use of a convolutional neural network for both

energy and shape prediction, machine learning models demonstrate a lot of promise for future laser prediction models—especially in light of machine learning's rapid growth in recent years. Future research could investigate issues such as noise in lower energy predictions and further increases in accuracy. Inertial fusion energy laser systems of the future could have their physics-based prediction systems completely replaced by machine learning models with their rapid prediction speed and adaptability, pushing us towards a future with cleaner energy.

## 7. Acknowledgements

I would like to thank my advisor Mark Guardalben for supporting me throughout the past summer. I would also like to thank Dr. Stephen Craxton for his work to make the High School Summer Research Program possible. Additionally, I would like to thank my physics teacher, Mr. Madden for inspiring me to apply and pursue this program. Lastly, I would like to thank my parents for supporting me and giving me the resources, so I was able to pursue opportunities like working at LLE.

# References

[1] Waxer, L. J., et al. “The OMEGA EP High-Energy, Short-Pulse Laser System.” CLEO/QELS 2008, OSA Technical Digest, Paper JThB1 (2008). https://opg.optica.org/abstract.cfm?uri=QELS-2008-JThB1.

[2] “Omega EP Laser System.” Laboratory for Laser Energetics, 4 Nov. 2024, www.lle.rochester.edu/omega-laser-facility-2/6271-2/.

[3] Guardalben, M. J., et al. “Laser-System Model for Enhanced Operational Performance and Flexibility on OMEGA EP.” High Power Laser Science and Engineering 8, e8 (2020). doi:10.1017/hpl.2020.6.

[4] Biswas, B. "Decoding Activation Functions in Neural Networks: Are You Choosing the Right One?" Medium, 06 Feb. 2024, https://medium.com/@balaka2605/decoding-activation-functions-in-neural-networks-are-you-choosing-the-right-one-24f792c9f4c6.

[5] Villani, D. “Energy Prediction on the OMEGA EP Laser System Using Neural Networks.” LLE Summer High School Research Program (2022).

[6] ODSC Community. “What Is Pruning in Machine Learning?” Open Data Science - Your News Source for AI, Machine Learning & More, 28 Oct. 2020, opendatascience.com/what-is-pruning-in-machine-learning/.